

*IBM SPSS Modeler 18.2.2 Modeling  
Nodes*



**Note**

Before you use this information and the product it supports, read the information in [“Notices” on page 349](#).

**Product Information**

This edition applies to version 18, release 2, modification 2 of IBM® SPSS® Modeler and to all subsequent releases and modifications until otherwise indicated in new editions.

© **Copyright International Business Machines Corporation .**

US Government Users Restricted Rights – Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.

---

# Contents

<b>Preface.....</b>	<b>xi</b>
About IBM Business Analytics.....	xi
Technical support.....	xi
 <b>Chapter 1. About IBM SPSS Modeler .....</b>	<b>1</b>
IBM SPSS Modeler Products.....	1
IBM SPSS Modeler .....	1
IBM SPSS Modeler Server .....	1
IBM SPSS Modeler Administration Console .....	2
IBM SPSS Modeler Batch .....	2
IBM SPSS Modeler Solution Publisher .....	2
IBM SPSS Modeler Server Adapters for IBM SPSS Collaboration and Deployment Services .....	2
IBM SPSS Modeler Editions.....	2
Documentation.....	3
SPSS Modeler Professional Documentation.....	3
SPSS Modeler Premium Documentation.....	4
Application examples.....	4
Demos Folder.....	4
License tracking.....	4
 <b>Chapter 2. Introduction to Modeling.....</b>	<b>5</b>
Building the Stream.....	6
Browsing the Model.....	10
Evaluating the Model.....	13
Scoring records.....	16
Summary.....	16
 <b>Chapter 3. Modeling Overview.....</b>	<b>19</b>
Overview of modeling nodes.....	19
Building Split Models.....	24
Splitting and Partitioning.....	24
Modeling nodes supporting split models.....	25
Features Affected by Splitting.....	25
Modeling Node Fields Options.....	26
Using Frequency and Weight Fields.....	27
Modeling Node Analyze Options.....	29
Propensity Scores.....	30
Misclassification Costs.....	31
Model Nuggets.....	31
Model Links.....	32
Replacing a model.....	33
The models palette.....	34
Browsing model nuggets.....	35
Model Nugget Summary / Information.....	36
Predictor Importance.....	37
Ensemble Viewer.....	38
Model Nuggets for Split Models.....	40
Using Model Nuggets in Streams.....	41
Regenerating a modeling node.....	41
Importing and exporting models as PMML.....	41

Publishing models for a scoring adapter.....	43
Unrefined Models.....	44
<b>Chapter 4. Screening Models.....</b>	<b>45</b>
Screening Fields and Records.....	45
Feature Selection node.....	45
Feature Selection Model Settings.....	46
Feature Selection Options.....	46
Feature Selection Model Nuggets.....	47
Feature Selection Model Results.....	47
Selecting Fields by Importance.....	48
Generating a Filter from a Feature Selection Model.....	48
Anomaly Detection Node.....	48
Anomaly Detection Model Options.....	49
Anomaly Detection Expert Options.....	50
Anomaly Detection Model Nuggets.....	50
Anomaly Detection model details.....	51
Anomaly Detection Model Summary.....	51
Anomaly Detection Model Settings.....	51
<b>Chapter 5. Automated Modeling Nodes.....</b>	<b>53</b>
Automated Modeling Node Algorithm Settings.....	54
Automated Modeling Node Stopping Rules.....	54
Auto Classifier node.....	54
Auto Classifier Node Model Options.....	55
Auto Classifier Node Expert Options.....	56
Misclassification Costs.....	59
Auto Classifier Node Discard Options.....	59
Auto Classifier Node Settings Options.....	60
Auto Numeric node.....	60
Auto Numeric Node Model Options.....	61
Auto Numeric Node Expert Options.....	62
Auto Numeric Node Settings Options.....	64
Auto Cluster node.....	64
Auto Cluster Node Model Options.....	65
Auto Cluster Node Expert Options.....	66
Auto Cluster Node Discard Options.....	67
Automated Model Nuggets.....	67
Generating Nodes and Models.....	68
Generating Evaluation Charts.....	69
Evaluation Graphs.....	69
<b>Chapter 6. Decision Trees.....</b>	<b>71</b>
Decision Tree Models.....	71
The Interactive Tree Builder.....	72
Growing and Pruning the Tree.....	73
Defining Custom Splits.....	74
Split Details and Surrogates.....	74
Customizing the Tree View.....	75
Gains.....	75
Risks.....	78
Saving Tree Models and Results.....	78
Generating Filter and Select Nodes.....	81
Generating a Rule Set from a Decision Tree.....	81
Building a Tree Model Directly.....	82
Decision Tree Nodes.....	82
C&R Tree Node.....	83

CHAID Node.....	84
QUEST Node.....	84
Decision Tree Node Fields Options.....	85
Decision Tree Node Build Options.....	85
Decision Tree Node Model Options.....	90
C5.0 Node.....	91
C5.0 Node Model Options.....	92
Tree-AS node.....	93
Tree-AS node fields options.....	94
Tree-AS node build options.....	94
Tree-AS node model options.....	96
Tree-AS model nugget.....	96
Random Trees node.....	98
Random Trees node fields options.....	98
Random Trees node build options.....	99
Random Trees node model options.....	100
Random Trees model nugget.....	101
C&R Tree, CHAID, QUEST, and C5.0 decision tree model nuggets.....	103
Single Tree Model Nuggets.....	104
Model nuggets for boosting, bagging, and very large datasets.....	108
C&R Tree, CHAID, QUEST, C5.0, and Apriori rule set model nuggets.....	109
Rule Set Model Tab.....	110
Importing Projects from AnswerTree 3.0.....	111
<b>Chapter 7. Bayesian Network Models.....</b>	<b>113</b>
Bayesian Network Node.....	113
Bayesian Network Node Model Options.....	114
Bayesian Network Node Expert Options.....	115
Bayesian Network Model Nuggets.....	117
Bayesian Network Model Settings.....	117
Bayesian Network Model Summary.....	118
<b>Chapter 8. Neural networks.....</b>	<b>119</b>
The neural networks model.....	119
Using neural networks with legacy streams.....	120
Objectives .....	121
Basics .....	122
Stopping rules .....	123
Ensembles .....	124
Advanced .....	125
Model Options .....	126
Model Summary .....	127
Predictor Importance .....	128
Predicted By Observed .....	129
Classification .....	130
Network .....	131
Settings .....	132
<b>Chapter 9. Decision List.....</b>	<b>133</b>
Decision List Model Options.....	134
Decision List Node Expert Options.....	135
Decision List Model Nugget.....	135
Decision List Model Nugget Settings.....	136
Decision List Viewer .....	136
Working Model Pane.....	136
Alternatives Tab.....	138
Snapshots Tab.....	138

Working with Decision List Viewer .....	139
---	-----

## **Chapter 10. Statistical Models..... 151**

Linear Node.....	152
Linear models.....	152
Linear-AS Node.....	158
Linear-AS models.....	158
Logistic Node.....	161
Logistic Node Model Options.....	162
Adding Terms to a Logistic Regression Model.....	164
Logistic Node Expert Options.....	165
Logistic Regression Convergence Options.....	165
Logistic Regression Advanced Output.....	166
Logistic Regression Stepping Options.....	166
Logistic Model Nugget.....	167
Logistic Nugget Model Details.....	167
Logistic Model Nugget Summary.....	168
Logistic Model Nugget Settings.....	168
Logistic Model Nugget Advanced Output.....	169
PCA/Factor Node.....	170
PCA/Factor Node Model Options.....	170
PCA/Factor Node Expert Options.....	171
PCA/Factor Node Rotation Options.....	172
PCA/Factor Model Nugget.....	172
PCA/Factor Model Nugget Equations.....	172
PCA/Factor Model Nugget Summary.....	172
PCA/Factor Model Nugget Advanced Output.....	173
Discriminant node.....	173
Discriminant Node Model Options.....	173
Discriminant Node Expert Options.....	174
Discriminant Node Output Options.....	174
Discriminant Node Stepping Options.....	175
Discriminant Model Nugget.....	176
GenLin Node.....	177
GenLin Node Field Options.....	177
GenLin Node Model Options.....	178
GenLin Node Expert Options.....	178
Generalized Linear Models Iterations.....	180
Generalized Linear Models Advanced Output.....	181
GenLin Model Nugget.....	182
Generalized Linear Mixed Models.....	183
GLMM Node.....	183
GLE Node.....	196
Target .....	197
Model effects .....	199
Weight and Offset .....	200
Build options .....	201
Estimation .....	201
Model selection .....	202
Model options .....	203
GLE model nugget.....	203
Cox Node.....	204
Cox Node Fields Options.....	205
Cox Node Model Options.....	205
Cox Node Expert Options.....	206
Cox Node Settings Options.....	207
Cox Model Nugget.....	208

<b>Chapter 11. Clustering models.....</b>	<b>211</b>
Kohonen node.....	212
Kohonen Node Model Options.....	213
Kohonen Node Expert Options.....	214
Kohonen Model Nuggets.....	214
Kohonen Model Summary.....	214
K-Means Node.....	215
K-Means Node Model Options.....	215
K-Means Node Expert Options.....	216
K-Means Model Nuggets .....	216
K-Means Model Summary.....	216
TwoStep Cluster node.....	216
TwoStep Cluster Node Model Options.....	217
TwoStep Cluster Model Nuggets.....	218
TwoStep Model Summary.....	218
TwoStep-AS Cluster node.....	218
Twostep-AS cluster analysis.....	218
TwoStep-AS Cluster Model Nuggets.....	222
TwoStep-AS Cluster Model Nugget Settings.....	223
K-Means-AS node.....	223
K-Means-AS node Fields.....	223
K-Means-AS node Build Options.....	223
The Cluster Viewer.....	224
Cluster Viewer - Model Tab .....	225
Navigating the Cluster Viewer.....	228
Generating Graphs from Cluster Models.....	229
 <b>Chapter 12. Association Rules.....</b>	 <b>231</b>
Tabular versus Transactional Data.....	232
Apriori node.....	233
Apriori Node Model Options.....	233
Apriori Node Expert Options.....	234
CARMA Node.....	235
CARMA Node Fields Options.....	235
CARMA Node Model Options.....	236
CARMA Node Expert Options.....	237
Association Rule Model Nuggets.....	237
Association Rule model nugget details.....	237
Association Rule Model Nugget Settings.....	240
Association Rule Model Nugget Summary.....	242
Generating a Rule Set from an Association Model Nugget.....	242
Generating a Filtered Model.....	242
Scoring Association Rules.....	243
Deploying Association Models.....	244
Sequence node.....	246
Sequence Node Fields Options.....	246
Sequence Node Model Options.....	247
Sequence Node Expert Options.....	247
Sequence Model Nuggets.....	248
Association Rules node.....	252
Association Rules - Fields Options.....	253
Association Rules - Rule building.....	253
Association Rules - Transformations.....	254
Association Rules - Output.....	255
Association Rules - Model Options.....	256
Association Rules Model Nuggets.....	257

<b>Chapter 13. Time Series Models.....</b>	<b>259</b>
Why forecast?.....	259
Time series data.....	259
Characteristics of time series.....	259
Autocorrelation and partial autocorrelation functions.....	263
Series transformations.....	264
Predictor series.....	264
Spatio-Temporal Prediction modeling node.....	264
Spatio-Temporal Prediction - Fields Options.....	265
Spatio-Temporal Prediction - Time Intervals.....	266
Spatio-Temporal Prediction - Basic Build Options.....	267
Spatio-Temporal Prediction - Advanced Build Options.....	267
Spatio-Temporal Prediction - Output.....	267
Spatio-Temporal Prediction - Model Options.....	268
Spatio-Temporal Prediction Model Nugget.....	269
TCM Node.....	269
Temporal Causal Models.....	269
TCM Model Nugget.....	279
Temporal Causal Model Scenarios.....	279
Time Series node.....	284
Time Series node - field options.....	285
Time Series node - data specification options.....	285
Time Series node - build options.....	288
Time Series node - model options.....	292
Time Series model nugget.....	294
 <b>Chapter 14. Self-Learning Response Node Models.....</b>	 <b>299</b>
SLRM node.....	299
SLRM Node Fields Options.....	299
SLRM Node Model Options.....	299
SLRM Node Settings Options.....	300
SLRM Model Nuggets.....	301
SLRM Model Settings.....	302
 <b>Chapter 15. Support Vector Machine Models.....</b>	 <b>305</b>
About SVM.....	305
How SVM Works.....	305
Tuning an SVM Model.....	306
SVM node.....	307
SVM Node Model Options.....	307
SVM Node Expert Options.....	307
SVM Model Nugget.....	308
SVM Model Settings.....	309
LSVM Node.....	309
LSVM Node Model Options.....	309
LSVM Build Options.....	310
LSVM Model Nugget (interactive output).....	310
LSVM Model Settings.....	311
 <b>Chapter 16. Nearest Neighbor Models.....</b>	 <b>313</b>
KNN node.....	313
KNN Node Objectives Options.....	313
KNN Node Settings .....	314
KNN Model Nugget.....	317
Nearest Neighbor Model View.....	318
KNN Model Settings.....	320



<b>Chapter 17. Python nodes.....</b>	<b>321</b>
SMOTE node.....	322
SMOTE node Settings.....	322
XGBoost Linear node.....	323
XGBoost Linear node Fields.....	323
XGBoost Linear node Build Options.....	323
XGBoost Linear node Model Options.....	324
XGBoost Tree node.....	324
XGBoost Tree node Fields.....	325
XGBoost Tree node Build Options.....	325
XGBoost Tree node Model Options.....	327
t-SNE node.....	327
t-SNE node Expert options.....	327
t-SNE node Output options.....	329
t-SNE model nuggets.....	329
Gaussian Mixture node.....	330
Gaussian Mixture node Fields.....	330
Gaussian Mixture node Build Options.....	330
Gaussian Mixture node Model Options.....	331
KDE nodes.....	331
KDE Modeling node and KDE Simulation node Fields.....	332
KDE nodes Build Options.....	332
KDE Modeling node and KDE Simulation node Model Options.....	333
Random Forest node.....	333
Random Forest node Fields.....	334
Random Forest node Build Options.....	334
Random Forest node Model Options.....	335
Random Forest model nuggets.....	335
HDBSCAN node.....	336
HDBSCAN node Fields.....	336
HDBSCAN node Build Options.....	336
HDBSCAN node Model Options.....	338
One-Class SVM node.....	338
One-Class SVM node Fields.....	339
One-Class SVM node Expert.....	339
One-Class SVM node Options.....	340
<b>Chapter 18. Spark nodes.....</b>	<b>341</b>
Isotonic-AS node.....	341
Isotonic-AS node Fields.....	341
Isotonic-AS node Build Options.....	342
Isotonic-AS model nuggets.....	342
XGBoost-AS node.....	342
XGBoost-AS node Fields.....	342
XGBoost-AS node Build Options.....	343
XGBoost-AS node Model Options.....	345
K-Means-AS node.....	345
K-Means-AS node Fields.....	345
K-Means-AS node Build Options.....	345
MultiLayerPerceptron-AS node.....	346
MultiLayerPerceptron-AS node Fields.....	346
MultiLayerPerceptron-AS node Build Options.....	347
MultiLayerPerceptron node Model Options.....	347
<b>Notices.....</b>	<b>349</b>
Trademarks.....	350

Terms and conditions for product documentation.....	350
<b>Glossary.....</b>	<b>353</b>
A.....	353
B.....	353
C.....	353
F.....	353
H.....	354
K.....	354
L.....	354
M.....	354
N.....	355
O.....	355
R.....	355
S.....	356
T.....	357
U.....	357
V.....	358
W.....	358
<b>Index.....</b>	<b>359</b>

## Preface

---

IBM SPSS Modeler is the IBM Corp. enterprise-strength data mining workbench. SPSS Modeler helps organizations to improve customer and citizen relationships through an in-depth understanding of data. Organizations use the insight gained from SPSS Modeler to retain profitable customers, identify cross-selling opportunities, attract new customers, detect fraud, reduce risk, and improve government service delivery.

SPSS Modeler's visual interface invites users to apply their specific business expertise, which leads to more powerful predictive models and shortens time-to-solution. SPSS Modeler offers many modeling techniques, such as prediction, classification, segmentation, and association detection algorithms. Once models are created, IBM SPSS Modeler Solution Publisher enables their delivery enterprise-wide to decision makers or to a database.

## About IBM Business Analytics

---

IBM Business Analytics software delivers complete, consistent and accurate information that decision-makers trust to improve business performance. A comprehensive portfolio of business intelligence, predictive analytics, financial performance and strategy management, and analytic applications provides clear, immediate and actionable insights into current performance and the ability to predict future outcomes. Combined with rich industry solutions, proven practices and professional services, organizations of every size can drive the highest productivity, confidently automate decisions and deliver better results.

As part of this portfolio, IBM SPSS Predictive Analytics software helps organizations predict future events and proactively act upon that insight to drive better business outcomes. Commercial, government and academic customers worldwide rely on IBM SPSS technology as a competitive advantage in attracting, retaining and growing customers, while reducing fraud and mitigating risk. By incorporating IBM SPSS software into their daily operations, organizations become predictive enterprises – able to direct and automate decisions to meet business goals and achieve measurable competitive advantage. For further information or to reach a representative visit <http://www.ibm.com/spss>.

## Technical support

---

Technical support is available to maintenance customers. Customers may contact Technical Support for assistance in using IBM Corp. products or for installation help for one of the supported hardware environments. To reach Technical Support, see the IBM Corp. web site at <http://www.ibm.com/support>. Be prepared to identify yourself, your organization, and your support agreement when requesting assistance.



---

# Chapter 1. About IBM SPSS Modeler

IBM SPSS Modeler is a set of data mining tools that enable you to quickly develop predictive models using business expertise and deploy them into business operations to improve decision making. Designed around the industry-standard CRISP-DM model, IBM SPSS Modeler supports the entire data mining process, from data to better business results.

IBM SPSS Modeler offers a variety of modeling methods taken from machine learning, artificial intelligence, and statistics. The methods available on the Modeling palette allow you to derive new information from your data and to develop predictive models. Each method has certain strengths and is best suited for particular types of problems.

SPSS Modeler can be purchased as a standalone product, or used as a client in combination with SPSS Modeler Server. A number of additional options are also available, as summarized in the following sections. For more information, see <https://www.ibm.com/analytics/us/en/technology/spss/>.

---

## IBM SPSS Modeler Products

The IBM SPSS Modeler family of products and associated software comprises the following.

- IBM SPSS Modeler
- IBM SPSS Modeler Server
- IBM SPSS Modeler Administration Console (included with IBM SPSS Deployment Manager)
- IBM SPSS Modeler Batch
- IBM SPSS Modeler Solution Publisher
- IBM SPSS Modeler Server adapters for IBM SPSS Collaboration and Deployment Services

### IBM SPSS Modeler

SPSS Modeler is a functionally complete version of the product that you install and run on your personal computer. You can run SPSS Modeler in local mode as a standalone product, or use it in distributed mode along with IBM SPSS Modeler Server for improved performance on large data sets.

With SPSS Modeler, you can build accurate predictive models quickly and intuitively, without programming. Using the unique visual interface, you can easily visualize the data mining process. With the support of the advanced analytics embedded in the product, you can discover previously hidden patterns and trends in your data. You can model outcomes and understand the factors that influence them, enabling you to take advantage of business opportunities and mitigate risks.

SPSS Modeler is available in two editions: SPSS Modeler Professional and SPSS Modeler Premium. See the topic “[IBM SPSS Modeler Editions](#)” on page 2 for more information.

### IBM SPSS Modeler Server

SPSS Modeler uses a client/server architecture to distribute requests for resource-intensive operations to powerful server software, resulting in faster performance on larger data sets.

SPSS Modeler Server is a separately-licensed product that runs continually in distributed analysis mode on a server host in conjunction with one or more IBM SPSS Modeler installations. In this way, SPSS Modeler Server provides superior performance on large data sets because memory-intensive operations can be done on the server without downloading data to the client computer. IBM SPSS Modeler Server also provides support for SQL optimization and in-database modeling capabilities, delivering further benefits in performance and automation.

## IBM SPSS Modeler Administration Console

The Modeler Administration Console is a graphical user interface for managing many of the SPSS Modeler Server configuration options, which are also configurable by means of an options file. The console is included in IBM SPSS Deployment Manager, can be used to monitor and configure your SPSS Modeler Server installations, and is available free-of-charge to current SPSS Modeler Server customers. The application can be installed only on Windows computers; however, it can administer a server installed on any supported platform.

## IBM SPSS Modeler Batch

While data mining is usually an interactive process, it is also possible to run SPSS Modeler from a command line, without the need for the graphical user interface. For example, you might have long-running or repetitive tasks that you want to perform with no user intervention. SPSS Modeler Batch is a special version of the product that provides support for the complete analytical capabilities of SPSS Modeler without access to the regular user interface. SPSS Modeler Server is required to use SPSS Modeler Batch.

## IBM SPSS Modeler Solution Publisher

SPSS Modeler Solution Publisher is a tool that enables you to create a packaged version of an SPSS Modeler stream that can be run by an external runtime engine or embedded in an external application. In this way, you can publish and deploy complete SPSS Modeler streams for use in environments that do not have SPSS Modeler installed. SPSS Modeler Solution Publisher is distributed as part of the IBM SPSS Collaboration and Deployment Services - Scoring service, for which a separate license is required. With this license, you receive SPSS Modeler Solution Publisher Runtime, which enables you to execute the published streams.

For more information about SPSS Modeler Solution Publisher, see the IBM SPSS Collaboration and Deployment Services documentation. The IBM SPSS Collaboration and Deployment Services Knowledge Center contains sections called "IBM SPSS Modeler Solution Publisher" and "IBM SPSS Analytics Toolkit."

## IBM SPSS Modeler Server Adapters for IBM SPSS Collaboration and Deployment Services

A number of adapters for IBM SPSS Collaboration and Deployment Services are available that enable SPSS Modeler and SPSS Modeler Server to interact with an IBM SPSS Collaboration and Deployment Services repository. In this way, an SPSS Modeler stream deployed to the repository can be shared by multiple users, or accessed from the thin-client application IBM SPSS Modeler Advantage. You install the adapter on the system that hosts the repository.

## IBM SPSS Modeler Editions

---

SPSS Modeler is available in the following editions.

### SPSS Modeler Professional

SPSS Modeler Professional provides all the tools you need to work with most types of structured data, such as behaviors and interactions tracked in CRM systems, demographics, purchasing behavior and sales data.

### SPSS Modeler Premium

SPSS Modeler Premium is a separately-licensed product that extends SPSS Modeler Professional to work with specialized data and with unstructured text data. SPSS Modeler Premium includes IBM SPSS Modeler Text Analytics:

**IBM SPSS Modeler Text Analytics** uses advanced linguistic technologies and Natural Language Processing (NLP) to rapidly process a large variety of unstructured text data, extract and organize the key concepts, and group these concepts into categories. Extracted concepts and categories can be combined

with existing structured data, such as demographics, and applied to modeling using the full suite of IBM SPSS Modeler data mining tools to yield better and more focused decisions.

### IBM SPSS Modeler Subscription

IBM SPSS Modeler Subscription provides all the same predictive analytics capabilities as the traditional IBM SPSS Modeler client. With the Subscription edition, you can download product updates regularly.

## Documentation

---

Documentation is available from the Help menu in SPSS Modeler. This opens the online Knowledge Center, which is always available outside the product.

Complete documentation for each product (including installation instructions) is also available in PDF format, in a separate compressed folder, as part of the product download. Or the latest PDF documents can be downloaded from the web at <https://www.ibm.com/support/pages/spss-modeler-1822-documentation>.

### SPSS Modeler Professional Documentation

The SPSS Modeler Professional documentation suite (excluding installation instructions) is as follows.

- **IBM SPSS Modeler User's Guide.** General introduction to using SPSS Modeler, including how to build data streams, handle missing values, build CLEM expressions, work with projects and reports, and package streams for deployment to IBM SPSS Collaboration and Deployment Services or IBM SPSS Modeler Advantage.
- **IBM SPSS Modeler Source, Process, and Output Nodes.** Descriptions of all the nodes used to read, process, and output data in different formats. Effectively this means all nodes other than modeling nodes.
- **IBM SPSS Modeler Modeling Nodes.** Descriptions of all the nodes used to create data mining models. IBM SPSS Modeler offers a variety of modeling methods taken from machine learning, artificial intelligence, and statistics.
- **IBM SPSS Modeler Applications Guide.** The examples in this guide provide brief, targeted introductions to specific modeling methods and techniques. An online version of this guide is also available from the Help menu. See the topic “[Application examples](#)” on [page 4](#) for more information.
- **IBM SPSS Modeler Python Scripting and Automation.** Information on automating the system through Python scripting, including the properties that can be used to manipulate nodes and streams.
- **IBM SPSS Modeler Deployment Guide.** Information on running IBM SPSS Modeler streams as steps in processing jobs under IBM SPSS Deployment Manager.
- **IBM SPSS Modeler CLEF Developer's Guide.** CLEF provides the ability to integrate third-party programs such as data processing routines or modeling algorithms as nodes in IBM SPSS Modeler.
- **IBM SPSS Modeler In-Database Mining Guide.** Information on how to use the power of your database to improve performance and extend the range of analytical capabilities through third-party algorithms.
- **IBM SPSS Modeler Server Administration and Performance Guide.** Information on how to configure and administer IBM SPSS Modeler Server.
- **IBM SPSS Deployment Manager User Guide.** Information on using the administration console user interface included in the Deployment Manager application for monitoring and configuring IBM SPSS Modeler Server.
- **IBM SPSS Modeler CRISP-DM Guide.** Step-by-step guide to using the CRISP-DM methodology for data mining with SPSS Modeler.
- **IBM SPSS Modeler Batch User's Guide.** Complete guide to using IBM SPSS Modeler in batch mode, including details of batch mode execution and command-line arguments. This guide is available in PDF format only.

## SPSS Modeler Premium Documentation

The SPSS Modeler Premium documentation suite (excluding installation instructions) is as follows.

- **SPSS Modeler Text Analytics User's Guide.** Information on using text analytics with SPSS Modeler, covering the text mining nodes, interactive workbench, templates, and other resources.

## Application examples

---

While the data mining tools in SPSS Modeler can help solve a wide variety of business and organizational problems, the application examples provide brief, targeted introductions to specific modeling methods and techniques. The data sets used here are much smaller than the enormous data stores managed by some data miners, but the concepts and methods that are involved are scalable to real-world applications.

To access the examples, click **Application Examples** on the Help menu in SPSS Modeler.

The data files and sample streams are installed in the Demos folder under the product installation directory. For more information, see [“Demos Folder”](#) on page 4.

**Database modeling examples.** See the examples in the *IBM SPSS Modeler In-Database Mining Guide*.

**Scripting examples.** See the examples in the *IBM SPSS Modeler Scripting and Automation Guide*.

## Demos Folder

---

The data files and sample streams that are used with the application examples are installed in the Demos folder under the product installation directory (for example: C:\Program Files\IBM\SPSS\Modeler\<version>\Demos). This folder can also be accessed from the IBM SPSS Modeler program group on the Windows Start menu, or by clicking Demos on the list of recent directories in the **File > Open Stream** dialog box.

## License tracking

---

When you use SPSS Modeler, license usage is tracked and logged at regular intervals. The license metrics that are logged are *AUTHORIZED\_USER* and *CONCURRENT\_USER*, and the type of metric that is logged depends on the type of license that you have for SPSS Modeler.

The log files that are produced can be processed by the IBM License Metric Tool, from which you can generate license usage reports.

The license log files are created in the same directory where SPSS Modeler Client log files are recorded (by default, %ALLUSERSPROFILE%\IBM\SPSS\Modeler\<version>\log).



# Chapter 2. Introduction to Modeling

A model is a set of rules, formulas, or equations that can be used to predict an outcome based on a set of input fields or variables. For example, a financial institution might use a model to predict whether loan applicants are likely to be good or bad risks, based on information that is already known about past applicants.

The ability to predict an outcome is the central goal of predictive analytics, and understanding the modeling process is the key to using IBM SPSS Modeler.

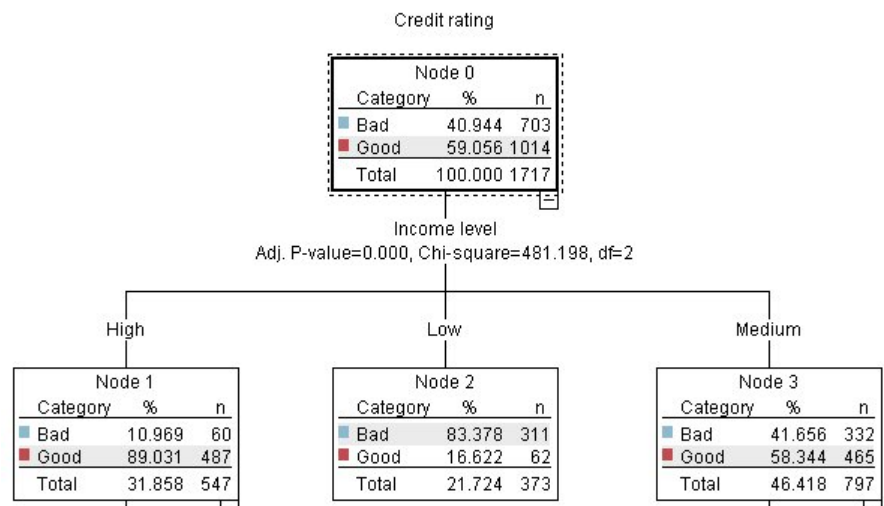


Figure 1. A simple decision tree model

This example uses a **decision tree** model, which classifies records (and predicts a response) using a series of decision rules, for example:

```
IF income = Medium
AND cards <5
THEN -> 'Good'
```

While this example uses a CHAID (Chi-squared Automatic Interaction Detection) model, it is intended as a general introduction, and most of the concepts apply broadly to other modeling types in IBM SPSS Modeler.

To understand any model, you first need to understand the data that go into it. The data in this example contain information about the customers of a bank. The following fields are used:

Field name	Description
Credit_rating	Credit rating: 0=Bad, 1=Good, 9=missing values
Age	Age in years
Income	Income level: 1=Low, 2=Medium, 3=High
Credit_cards	Number of credit cards held: 1=Less than five, 2=Five or more
Education	Level of education: 1=High school, 2=College
Car_loans	Number of car loans taken out: 1=None or one, 2=More than two

The bank maintains a database of historical information on customers who have taken out loans with the bank, including whether or not they repaid the loans (Credit rating = Good) or defaulted (Credit rating =

Bad). Using this existing data, the bank wants to build a model that will enable them to predict how likely future loan applicants are to default on the loan.

Using a decision tree model, you can analyze the characteristics of the two groups of customers and predict the likelihood of loan defaults.

This example uses the stream named *modelingintro.str*, available in the *Demos* folder under the *streams* subfolder. The data file is *tree\_credit.sav*. See the topic “[Demos Folder](#)” on [page 4](#) for more information.

Let's take a look at the stream.

1. Choose the following from the main menu:

**File > Open Stream**

2. Click the gold nugget icon on the toolbar of the Open dialog box and choose the Demos folder.
3. Double-click the *streams* folder.
4. Double-click the file named *modelingintro.str*.

## Building the Stream

---

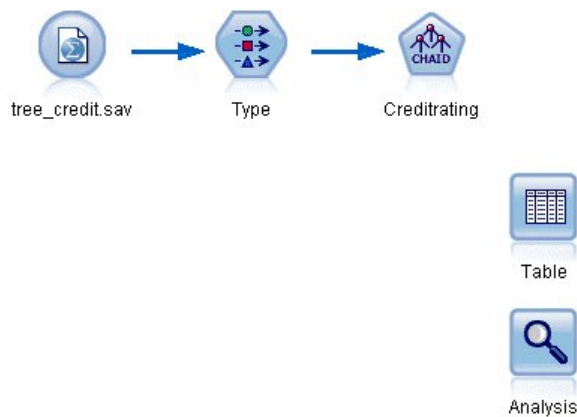


Figure 2. Modeling stream

To build a stream that will create a model, we need at least three elements:

- A source node that reads in data from some external source, in this case an IBM SPSS Statistics data file.
- A source or Type node that specifies field properties, such as measurement level (the type of data that the field contains), and the role of each field as a target or input in modeling.
- A modeling node that generates a model nugget when the stream is run.

In this example, we're using a CHAID modeling node. CHAID, or Chi-squared Automatic Interaction Detection, is a classification method that builds decision trees by using a particular type of statistics known as chi-square statistics to work out the best places to make the splits in the decision tree.

If measurement levels are specified in the source node, the separate Type node can be eliminated. Functionally, the result is the same.

This stream also has Table and Analysis nodes that will be used to view the scoring results after the model nugget has been created and added to the stream.

The Statistics File source node reads data in IBM SPSS Statistics format from the *tree\_credit.sav* data file, which is installed in the *Demos* folder. (A special variable named *\$CLEO\_DEMOS* is used to reference this folder under the current IBM SPSS Modeler installation. This ensures the path will be valid regardless of the current installation folder or version.)

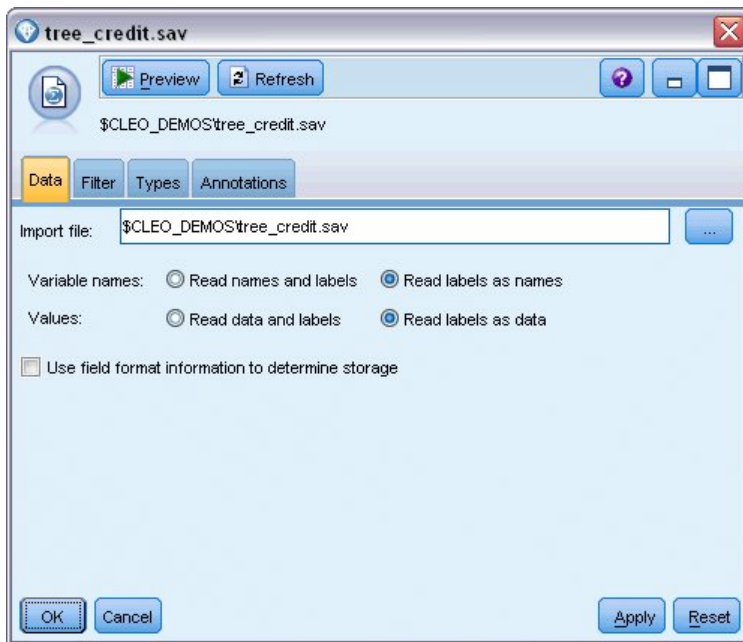


Figure 3. Reading data with a Statistics File source node

The Type node specifies the **measurement level** for each field. The measurement level is a category that indicates the type of data in the field. Our source data file uses three different measurement levels.

A **Continuous** field (such as the *Age* field) contains continuous numeric values, while a **Nominal** field (such as the *Credit rating* field) has two or more distinct values, for example *Bad*, *Good*, or *No credit history*. An **Ordinal** field (such as the *Income level* field) describes data with multiple distinct values that have an inherent order—in this case *Low*, *Medium* and *High*.

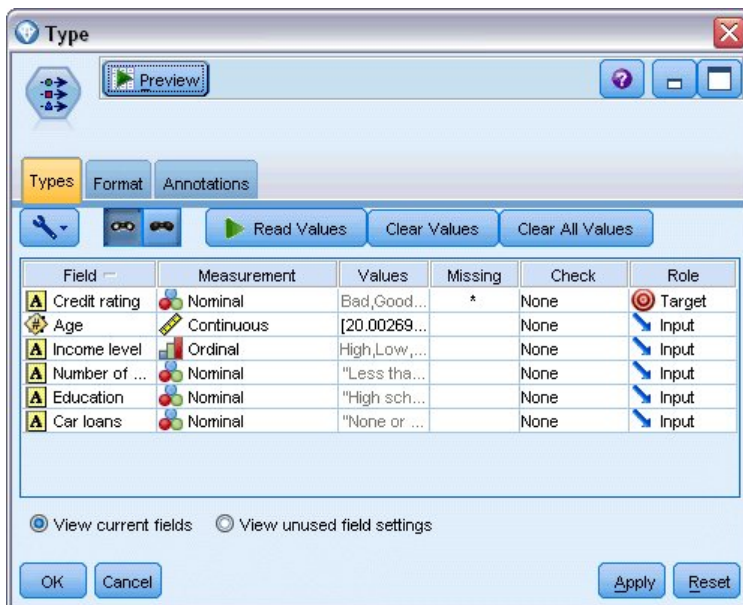


Figure 4. Setting the target and input fields with the Type node

For each field, the Type node also specifies a **role**, to indicate the part that each field plays in modeling. The role is set to *Target* for the field *Credit rating*, which is the field that indicates whether or not a given customer defaulted on the loan. This is the **target**, or the field for which we want to predict the value.

Role is set to *Input* for the other fields. Input fields are sometimes known as **predictors**, or fields whose values are used by the modeling algorithm to predict the value of the target field.

The CHAID modeling node generates the model.

On the Fields tab in the modeling node, the option **Use predefined roles** is selected, which means the target and inputs will be used as specified in the Type node. We could change the field roles at this point, but for this example we'll use them as they are.

1. Click the Build Options tab.

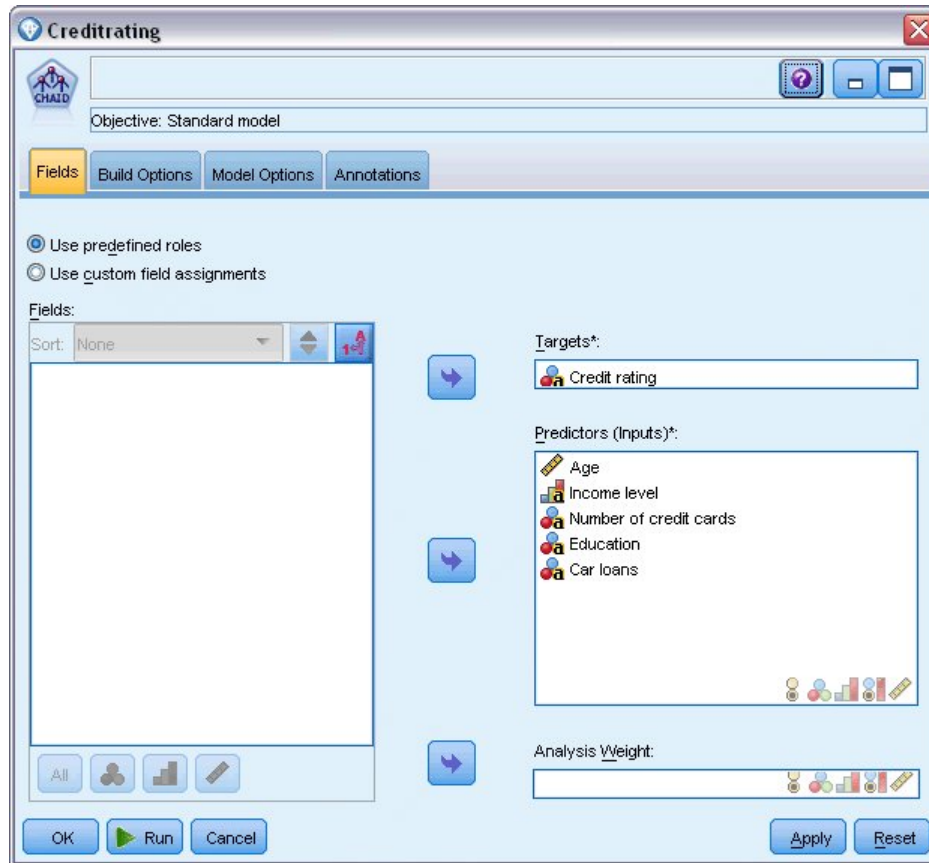


Figure 5. CHAID modeling node, Fields tab

Here there are several options where we could specify the kind of model we want to build.

We want a brand-new model, so we'll use the default option **Build new model**.

We also just want a single, standard decision tree model without any enhancements, so we'll also leave the default objective option **Build a single tree**.

While we can optionally launch an interactive modeling session that allows us to fine-tune the model, this example simply generates a model using the default mode setting **Generate model**.

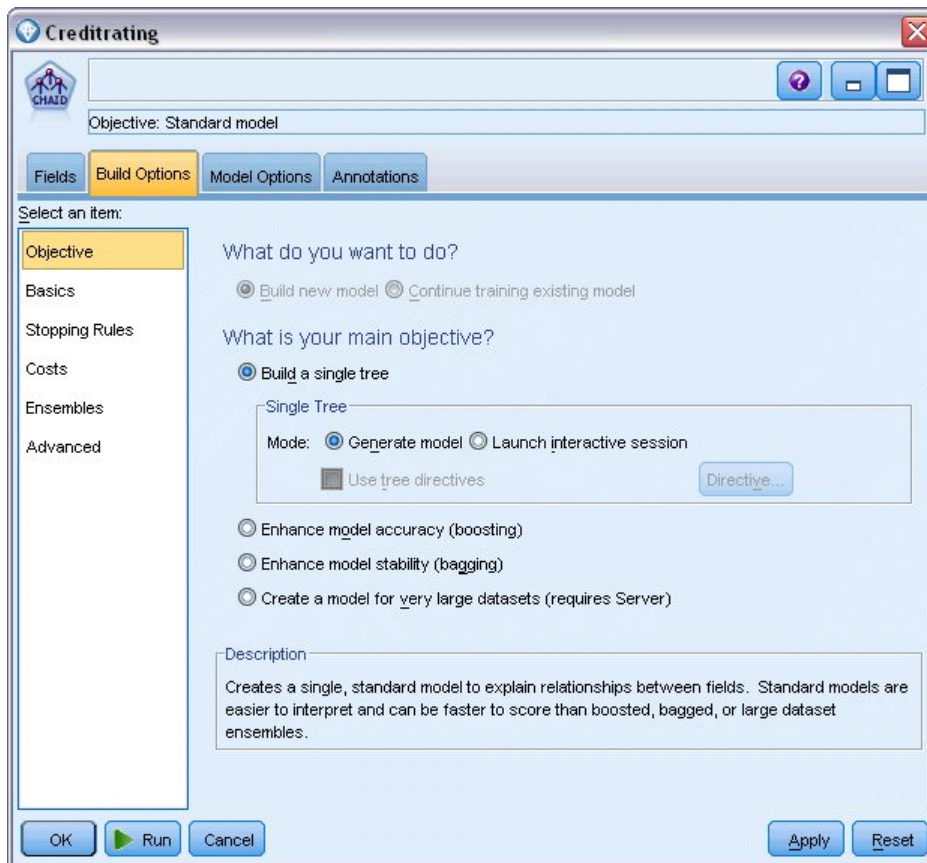


Figure 6. CHAID modeling node, Build Options tab

For this example, we want to keep the tree fairly simple, so we'll limit the tree growth by raising the minimum number of cases for parent and child nodes.

2. On the Build Options tab, select **Stopping Rules** from the navigator pane on the left.
3. Select the **Use absolute value** option.
4. Set **Minimum records in parent branch** to 400.
5. Set **Minimum records in child branch** to 200.

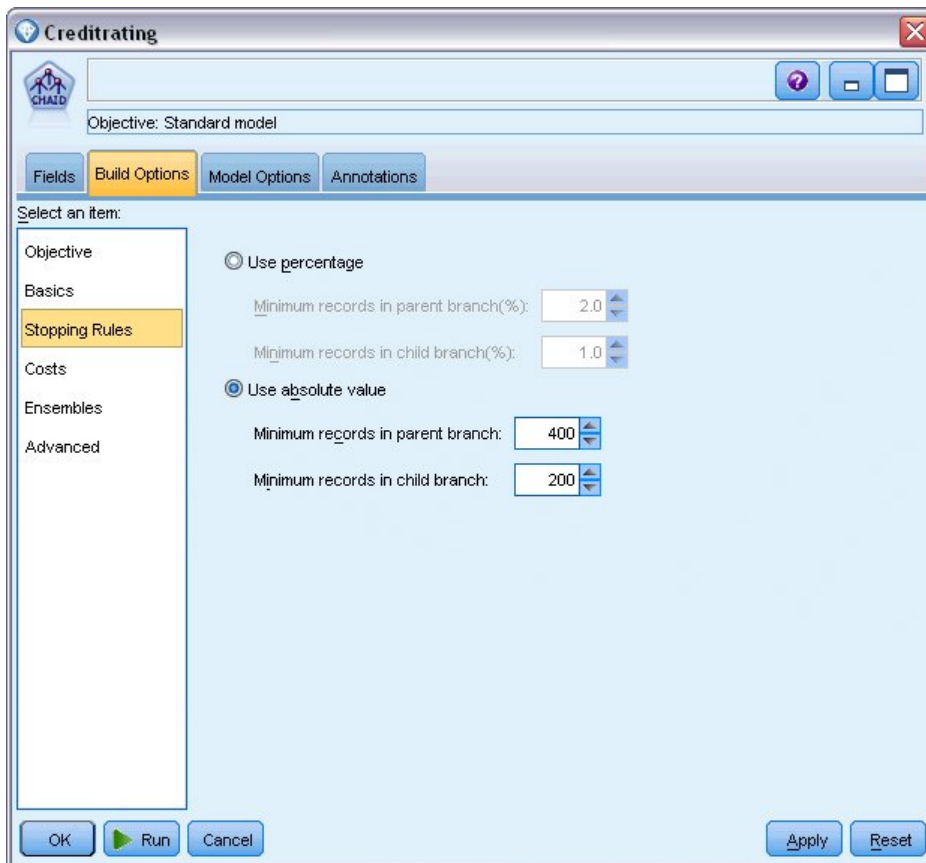


Figure 7. Setting the stopping criteria for decision tree building

We can use all the other default options for this example, so click **Run** to create the model. (Alternatively, right-click on the node and choose **Run** from the context menu, or select the node and choose **Run** from the Tools menu.)

## Browsing the Model

When execution completes, the model nugget is added to the Models palette in the upper right corner of the application window, and is also placed on the stream canvas with a link to the modeling node from which it was created. To view the model details, right-click on the model nugget and choose **Browse** (on the models palette) or **Edit** (on the canvas).

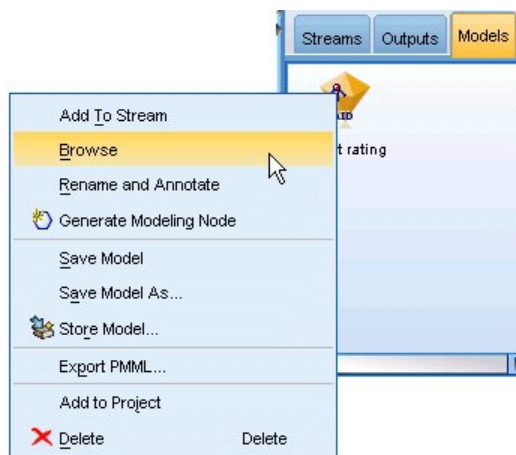


Figure 8. Models palette

In the case of the CHAID nugget, the Model tab displays the details in the form of a rule set--essentially a series of rules that can be used to assign individual records to child nodes based on the values of different input fields.

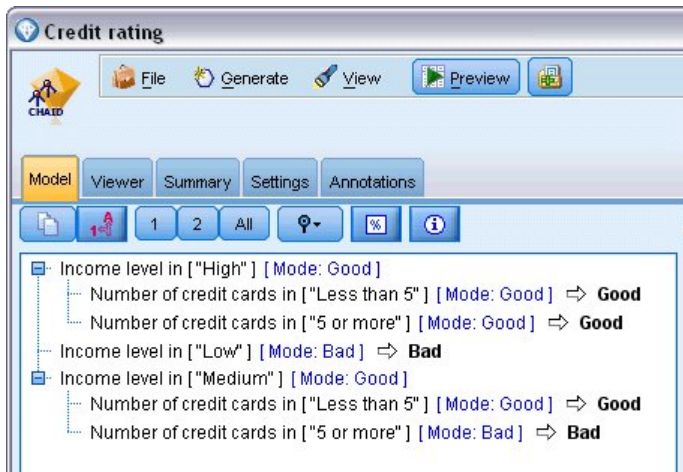


Figure 9. CHAID model nugget, rule set

For each decision tree terminal node--meaning those tree nodes that are not split further--a prediction of *Good* or *Bad* is returned. In each case the prediction is determined by the **mode**, or most common response, for records that fall within that node.

To the right of the rule set, the Model tab displays the Predictor Importance chart, which shows the relative importance of each predictor in estimating the model. From this we can see that *Income level* is easily the most significant in this case, and that the only other significant factor is *Number of credit cards*.

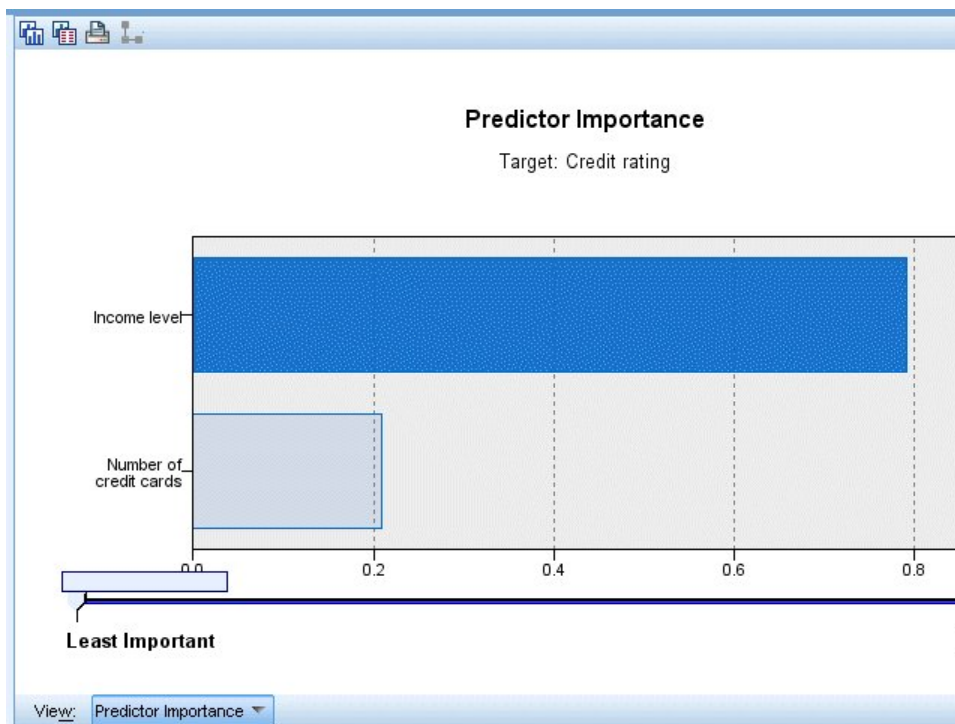


Figure 10. Predictor Importance chart

The Viewer tab in the model nugget displays the same model in the form of a tree, with a node at each decision point. Use the Zoom controls on the toolbar to zoom in on a specific node or zoom out to see the more of the tree.



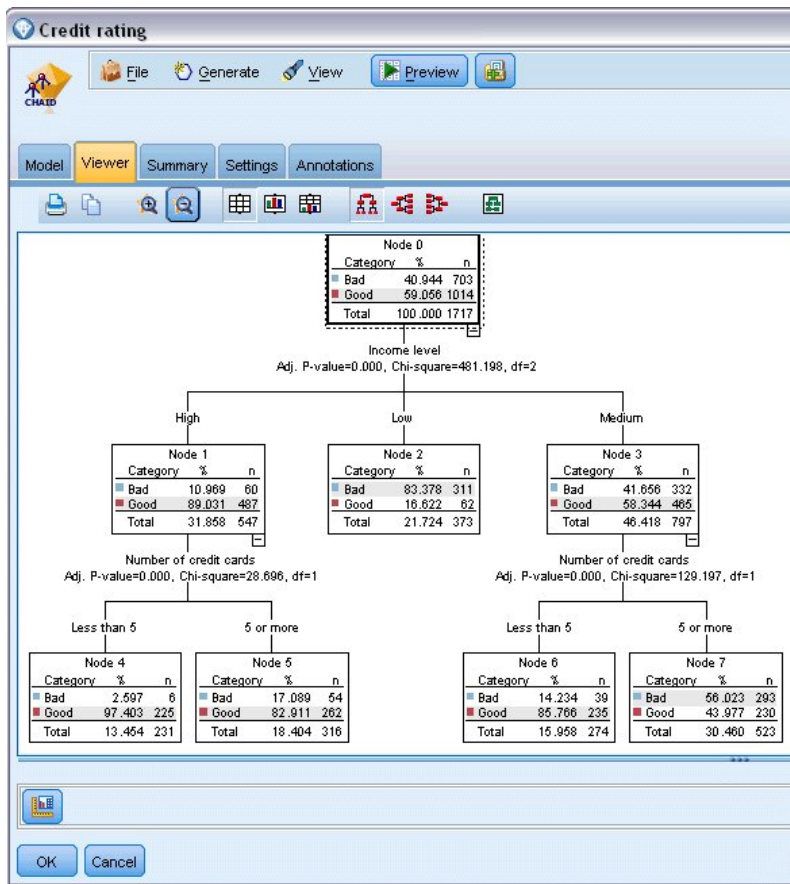


Figure 11. Viewer tab in the model nugget, with zoom out selected

Looking at the upper part of the tree, the first node (Node 0) gives us a summary for all the records in the data set. Just over 40% of the cases in the data set are classified as a bad risk. This is quite a high proportion, so let's see if the tree can give us any clues as to what factors might be responsible.

We can see that the first split is by *Income level*. Records where the income level is in the *Low* category are assigned to Node 2, and it's no surprise to see that this category contains the highest percentage of loan defaulters. Clearly lending to customers in this category carries a high risk.

However, 16% of the customers in this category actually *didn't* default, so the prediction won't always be correct. No model can feasibly predict every response, but a good model should allow us to predict the *most likely* response for each record based on the available data.

In the same way, if we look at the high income customers (Node 1), we see that the vast majority (89%) are a good risk. But more than 1 in 10 of these customers has also defaulted. Can we refine our lending criteria to minimize the risk here?

Notice how the model has divided these customers into two sub-categories (Nodes 4 and 5), based on the number of credit cards held. For high-income customers, if we lend only to those with fewer than 5 credit cards, we can increase our success rate from 89% to 97%--an even more satisfactory outcome.



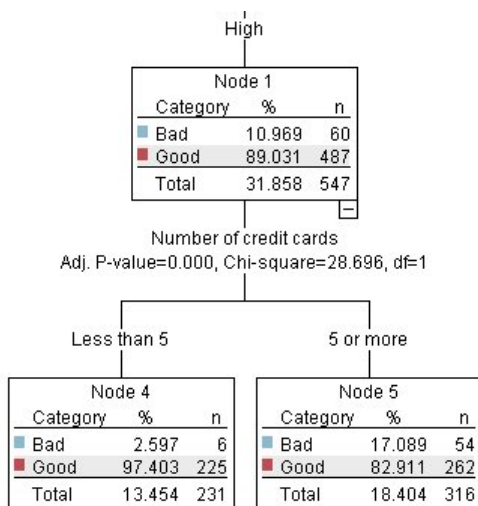


Figure 12. Tree view of high-income customers

But what about those customers in the Medium income category (Node 3)? They're much more evenly divided between Good and Bad ratings.

Again, the sub-categories (Nodes 6 and 7 in this case) can help us. This time, lending only to those medium-income customers with fewer than 5 credit cards increases the percentage of Good ratings from 58% to 85%, a significant improvement.

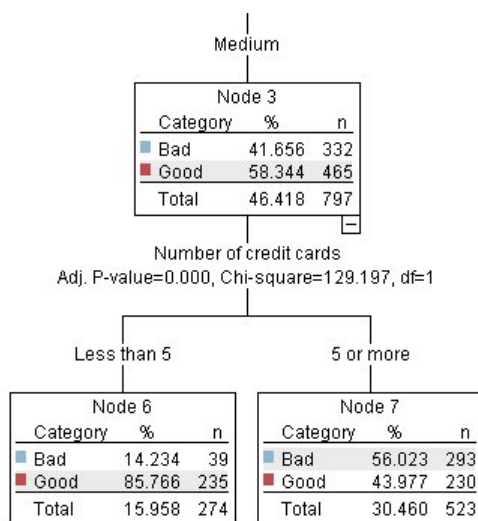


Figure 13. Tree view of medium-income customers

So, we've learnt that every record that is input to this model will be assigned to a specific node, and assigned a prediction of *Good* or *Bad* based on the most common response for that node.

This process of assigning predictions to individual records is known as **scoring**. By scoring the same records used to estimate the model, we can evaluate how accurately it performs on the training data—the data for which we know the outcome. Let's look at how to do this.

## Evaluating the Model

We've been browsing the model to understand how scoring works. But to evaluate *how accurately* it works, we need to score some records and compare the responses predicted by the model to the actual results. We're going to score the same records that were used to estimate the model, allowing us to compare the observed and predicted responses.

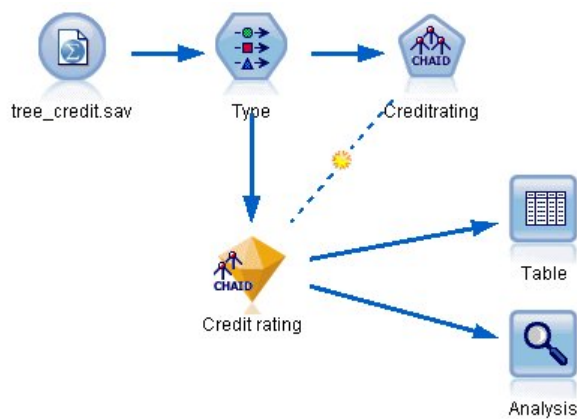


Figure 14. Attaching the model nugget to output nodes for model evaluation

1. To see the scores or predictions, attach the Table node to the model nugget, double-click the Table node and click **Run**.

The table displays the predicted scores in a field named *\$R-Credit rating*, which was created by the model. We can compare these values to the original *Credit rating* field that contains the actual responses.

By convention, the names of the fields generated during scoring are based on the target field, but with a standard prefix. Prefixes *\$G* and *\$GE* are generated by the Generalized Linear Model, *\$R* is the prefix used for the prediction generated by the CHAID model in this case, *\$RC* is for confidence values, *\$X* is typically generated by using an ensemble, and *\$XR*, *\$XS*, and *\$XF* are used as prefixes in cases where the target field is a Continuous, Categorical, Set, or Flag field, respectively. Different model types use different sets of prefixes. A **confidence value** is the model's own estimation, on a scale from 0.0 to 1.0, of how accurate each predicted value is.

Number of credit cards	Education	Car loans	\$R-Credit rating	\$RC-Credit rating
5 or more	College	More than 2	Bad	0.560
5 or more	College	More than 2	Bad	0.560
5 or more	High school	More than 2	Bad	0.832
5 or more	College	None or 1	Bad	0.832
5 or more	College	More than 2	Bad	0.560
5 or more	College	More than 2	Bad	0.560
5 or more	College	More than 2	Bad	0.560
5 or more	High school	More than 2	Bad	0.832
5 or more	High school	More than 2	Bad	0.832
5 or more	College	More than 2	Bad	0.560
5 or more	College	More than 2	Bad	0.832
5 or more	High school	More than 2	Bad	0.832
5 or more	High school	More than 2	Bad	0.560
5 or more	College	None or 1	Bad	0.832
5 or more	High school	More than 2	Bad	0.832
5 or more	College	More than 2	Bad	0.832
5 or more	College	More than 2	Bad	0.560
5 or more	College	More than 2	Bad	0.560
5 or more	College	More than 2	Good	0.827

Figure 15. Table showing generated scores and confidence values

As expected, the predicted value matches the actual responses for many records but not all. The reason for this is that each CHAID terminal node has a mix of responses. The prediction matches the

most common one, but will be wrong for all the others in that node. (Recall the 16% minority of low-income customers who did not default.)

To avoid this, we could continue splitting the tree into smaller and smaller branches, until every node was 100% pure—all *Good* or *Bad* with no mixed responses. But such a model would be extremely complicated and would probably not generalize well to other datasets.

To find out exactly how many predictions are correct, we could read through the table and tally the number of records where the value of the predicted field *\$R-Credit rating* matches the value of *Credit rating*. Fortunately, there's a much easier way--we can use an Analysis node, which does this automatically.

2. Connect the model nugget to the Analysis node.
3. Double-click the Analysis node and click **Run**.

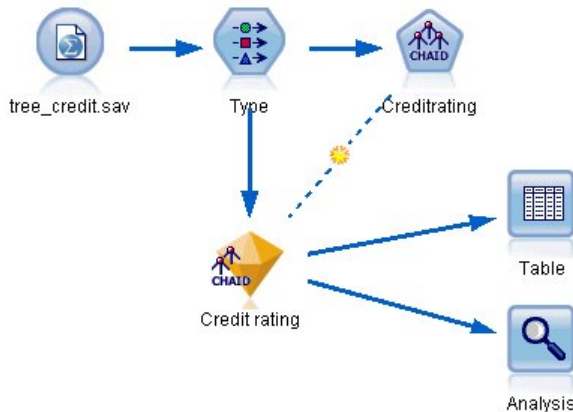


Figure 16. Attaching an Analysis node

The analysis shows that for 1899 out of 2464 records--over 77%--the value predicted by the model matched the actual response.

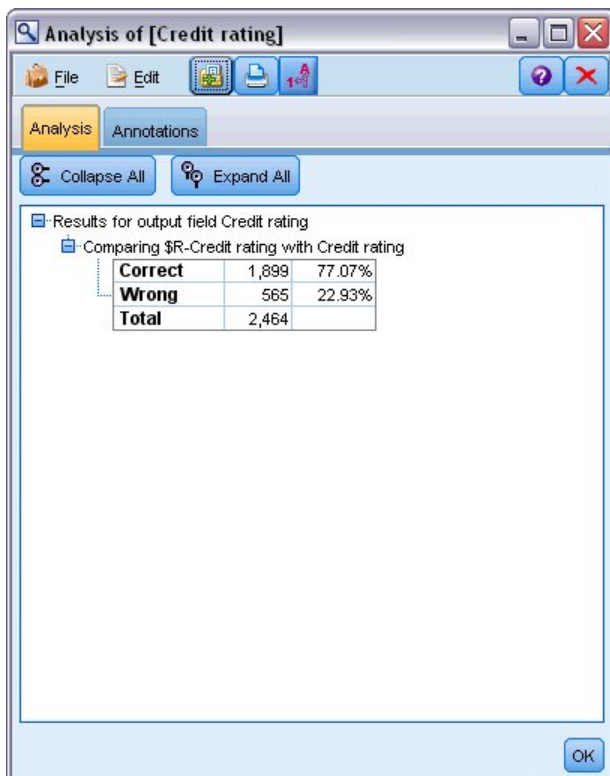


Figure 17. Analysis results comparing observed and predicted responses

This result is limited by the fact that the records being scored are the same ones used to estimate the model. In a real situation, you could use a Partition node to split the data into separate samples for training and evaluation.

By using one sample partition to generate the model and another sample to test it, you can get a much better indication of how well it will generalize to other datasets.

The Analysis node allows us to test the model against records for which we already know the actual result. The next stage illustrates how we can use the model to score records for which we don't know the outcome. For example, this might include people who are not currently customers of the bank, but who are prospective targets for a promotional mailing.

## Scoring records

Earlier, we scored the same records used to estimate the model in order to evaluate how accurate the model was. Now we're going to see how to score a different set of records from the ones used to create the model. This is the goal of modeling with a target field: Study records for which you know the outcome, to identify patterns that will allow you to predict outcomes you don't yet know.

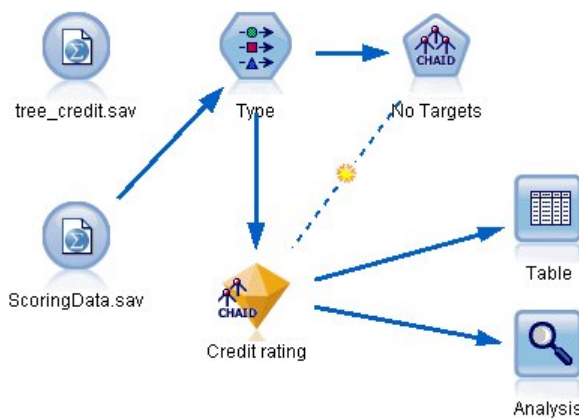


Figure 18. Attaching new data for scoring

You could update the Statistics File source node to point to a different data file, or you could add a new source node that reads in the data you want to score. Either way, the new dataset must contain the same input fields used by the model (*Age*, *Income level*, *Education* and so on) but not the target field *Credit rating*.

Alternatively, you could add the model nugget to any stream that includes the expected input fields. Whether read from a file or a database, the source type doesn't matter as long as the field names and types match those used by the model.

You could also save the model nugget as a separate file, or export the model in PMML format for use with other applications that support this format, or store the model in an IBM SPSS Collaboration and Deployment Services repository, which offers enterprise-wide deployment, scoring, and management of models.

Regardless of the infrastructure used, the model itself works in the same way.

## Summary

This example demonstrates the basic steps for creating, evaluating, and scoring a model.

- The modeling node estimates the model by studying records for which the outcome is known, and creates a model nugget. This is sometimes referred to as training the model.
- The model nugget can be added to any stream with the expected fields to score records. By scoring the records for which you already know the outcome (such as existing customers), you can evaluate how well it performs.

- Once you are satisfied that the model performs acceptably well, you can score new data (such as prospective customers) to predict how they will respond.
- The data used to train or estimate the model may be referred to as the analytical or historical data; the scoring data may also be referred to as the operational data.



---

# Chapter 3. Modeling Overview

## Overview of modeling nodes

---

IBM SPSS Modeler offers a variety of modeling methods taken from machine learning, artificial intelligence, and statistics. The methods available on the Modeling palette allow you to derive new information from your data and to develop predictive models. Each method has certain strengths and is best suited for particular types of problems.

The *IBM SPSS Modeler Applications Guide* provides examples for many of these methods, along with a general introduction to the modeling process. This guide is available as an online tutorial, and also in PDF format. See the topic [“Application examples”](#) on page 4 for more information.

Modeling methods are divided into these categories:

- Supervised
- Association
- Segmentation

### Supervised Models

*Supervised models* use the values of one or more **input** fields to predict the value of one or more output, or **target**, fields. Some examples of these techniques are: decision trees (C&R Tree, QUEST, CHAID and C5.0 algorithms), regression (linear, logistic, generalized linear, and Cox regression algorithms), neural networks, support vector machines, and Bayesian networks.

Supervised models help organizations to predict a known result, such as whether a customer will buy or leave or whether a transaction fits a known pattern of fraud. Modeling techniques include machine learning, rule induction, subgroup identification, statistical methods, and multiple model generation.

#### *Supervised nodes*



The Auto Classifier node creates and compares a number of different models for binary outcomes (yes or no, churn or do not churn, and so on), allowing you to choose the best approach for a given analysis. A number of modeling algorithms are supported, making it possible to select the methods you want to use, the specific options for each, and the criteria for comparing the results. The node generates a set of models based on the specified options and ranks the best candidates according to the criteria you specify.



The Auto Numeric node estimates and compares models for continuous numeric range outcomes using a number of different methods. The node works in the same manner as the Auto Classifier node, allowing you to choose the algorithms to use and to experiment with multiple combinations of options in a single modeling pass. Supported algorithms include neural networks, C&R Tree, CHAID, linear regression, generalized linear regression, and support vector machines (SVM). Models can be compared based on correlation, relative error, or number of variables used.



The Classification and Regression (C&R) Tree node generates a decision tree that allows you to predict or classify future observations. The method uses recursive partitioning to split the training records into segments by minimizing the impurity at each step, where a node in the tree is considered “pure” if 100% of cases in the node fall into a specific category of the target field. Target and input fields can be numeric ranges or categorical (nominal, ordinal, or flags); all splits are binary (only two subgroups).



The QUEST node provides a binary classification method for building decision trees, designed to reduce the processing time required for large C&R Tree analyses while also reducing the tendency found in classification tree methods to favor inputs that allow more splits. Input fields can be numeric ranges (continuous), but the target field must be categorical. All splits are binary.



The CHAID node generates decision trees using chi-square statistics to identify optimal splits. Unlike the C&R Tree and QUEST nodes, CHAID can generate nonbinary trees, meaning that some splits have more than two branches. Target and input fields can be numeric range (continuous) or categorical. Exhaustive CHAID is a modification of CHAID that does a more thorough job of examining all possible splits but takes longer to compute.



The C5.0 node builds either a decision tree or a rule set. The model works by splitting the sample based on the field that provides the maximum information gain at each level. The target field must be categorical. Multiple splits into more than two subgroups are allowed.



The Decision List node identifies subgroups, or segments, that show a higher or lower likelihood of a given binary outcome relative to the overall population. For example, you might look for customers who are unlikely to churn or are most likely to respond favorably to a campaign. You can incorporate your business knowledge into the model by adding your own custom segments and previewing alternative models side by side to compare the results. Decision List models consist of a list of rules in which each rule has a condition and an outcome. Rules are applied in order, and the first rule that matches determines the outcome.



Linear regression models predict a continuous target based on linear relationships between the target and one or more predictors.



The PCA/Factor node provides powerful data-reduction techniques to reduce the complexity of your data. Principal components analysis (PCA) finds linear combinations of the input fields that do the best job of capturing the variance in the entire set of fields, where the components are orthogonal (perpendicular) to each other. Factor analysis attempts to identify underlying factors that explain the pattern of correlations within a set of observed fields. For both approaches, the goal is to find a small number of derived fields that effectively summarizes the information in the original set of fields.



The Feature Selection node screens input fields for removal based on a set of criteria (such as the percentage of missing values); it then ranks the importance of remaining inputs relative to a specified target. For example, given a data set with hundreds of potential inputs, which are most likely to be useful in modeling patient outcomes?



Discriminant analysis makes more stringent assumptions than logistic regression but can be a valuable alternative or supplement to a logistic regression analysis when those assumptions are met.





Logistic regression is a statistical technique for classifying records based on values of input fields. It is analogous to linear regression but takes a categorical target field instead of a numeric range.



The Generalized Linear model expands the general linear model so that the dependent variable is linearly related to the factors and covariates through a specified link function. Moreover, the model allows for the dependent variable to have a non-normal distribution. It covers the functionality of a wide number of statistical models, including linear regression, logistic regression, loglinear models for count data, and interval-censored survival models.



A generalized linear mixed model (GLMM) extends the linear model so that the target can have a non-normal distribution, is linearly related to the factors and covariates via a specified link function, and so that the observations can be correlated. Generalized linear mixed models cover a wide variety of models, from simple linear regression to complex multilevel models for non-normal longitudinal data.



The Cox regression node enables you to build a survival model for time-to-event data in the presence of censored records. The model produces a survival function that predicts the probability that the event of interest has occurred at a given time ( $t$ ) for given values of the input variables.



The Support Vector Machine (SVM) node enables you to classify data into one of two groups without overfitting. SVM works well with wide data sets, such as those with a very large number of input fields.



The Bayesian Network node enables you to build a probability model by combining observed and recorded evidence with real-world knowledge to establish the likelihood of occurrences. The node focuses on Tree Augmented Naïve Bayes (TAN) and Markov Blanket networks that are primarily used for classification.



The Self-Learning Response Model (SLRM) node enables you to build a model in which a single new case, or small number of new cases, can be used to reestimate the model without having to retrain the model using all data.



The Time Series node estimates exponential smoothing, univariate Autoregressive Integrated Moving Average (ARIMA), and multivariate ARIMA (or transfer function) models for time series data and produces forecasts of future performance. This Time Series node is similar to the previous Time Series node that was deprecated in SPSS Modeler version 18. However, this newer Time Series node is designed to harness the power of IBM SPSS Analytic Server to process big data, and display the resulting model in the output viewer that was added in SPSS Modeler version 17.



The  $k$ -Nearest Neighbor (KNN) node associates a new case with the category or value of the  $k$  objects nearest to it in the predictor space, where  $k$  is an integer. Similar cases are near each other and dissimilar cases are distant from each other.



The Spatio-Temporal Prediction (STP) node uses data that contains location data, input fields for prediction (predictors), a time field, and a target field. Each location has numerous rows in the data that represent the values of each predictor at each time of measurement. After the data is analyzed, it can be used to predict target values at any location within the shape data that is used in the analysis.

## Association Models

*Association models* find patterns in your data where one or more entities (such as events, purchases, or attributes) are associated with one or more other entities. The models construct rule sets that define these relationships. Here the fields within the data can act as both inputs and targets. You could find these associations manually, but association rule algorithms do so much more quickly, and can explore more complex patterns. Apriori and Carma models are examples of the use of such algorithms. One other type of association model is a sequence detection model, which finds sequential patterns in time-structured data.

Association models are most useful when predicting multiple outcomes—for example, customers who bought product X also bought Y and Z. Association models associate a particular conclusion (such as the decision to buy something) with a set of conditions. The advantage of association rule algorithms over the more standard decision tree algorithms (C5.0 and C&RT) is that associations can exist between any of the attributes. A decision tree algorithm will build rules with only a single conclusion, whereas association algorithms attempt to find many rules, each of which may have a different conclusion.

### Association nodes



The Apriori node extracts a set of rules from the data, pulling out the rules with the highest information content. Apriori offers five different methods of selecting rules and uses a sophisticated indexing scheme to process large data sets efficiently. For large problems, Apriori is generally faster to train; it has no arbitrary limit on the number of rules that can be retained, and it can handle rules with up to 32 preconditions. Apriori requires that input and output fields all be categorical but delivers better performance because it is optimized for this type of data.



The CARMA model extracts a set of rules from the data without requiring you to specify input or target fields. In contrast to Apriori the CARMA node offers build settings for rule support (support for both antecedent and consequent) rather than just antecedent support. This means that the rules generated can be used for a wider variety of applications—for example, to find a list of products or services (antecedents) whose consequent is the item that you want to promote this holiday season.



The Sequence node discovers association rules in sequential or time-oriented data. A sequence is a list of item sets that tends to occur in a predictable order. For example, a customer who purchases a razor and aftershave lotion may purchase shaving cream the next time he shops. The Sequence node is based on the CARMA association rules algorithm, which uses an efficient two-pass method for finding sequences.



The Association Rules Node is similar to the Apriori Node; however, unlike Apriori, the Association Rules Node can process list data. In addition, the Association Rules Node can be used with IBM SPSS Analytic Server to process big data and take advantage of faster parallel processing.

## Segmentation Models

*Segmentation models* divide the data into segments, or clusters, of records that have similar patterns of input fields. As they are only interested in the input fields, segmentation models have no concept of

output or target fields. Examples of segmentation models are Kohonen networks, K-Means clustering, two-step clustering and anomaly detection.

Segmentation models (also known as "clustering models") are useful in cases where the specific result is unknown (for example, when identifying new patterns of fraud, or when identifying groups of interest in your customer base). Clustering models focus on identifying groups of similar records and labeling the records according to the group to which they belong. This is done without the benefit of prior knowledge about the groups and their characteristics, and it distinguishes clustering models from the other modeling techniques in that there is no predefined output or target field for the model to predict. There are no right or wrong answers for these models. Their value is determined by their ability to capture interesting groupings in the data and provide useful descriptions of those groupings. Clustering models are often used to create clusters or segments that are then used as inputs in subsequent analyses (for example, by segmenting potential customers into homogeneous subgroups).

#### *Segmentation nodes*



The Auto Cluster node estimates and compares clustering models, which identify groups of records that have similar characteristics. The node works in the same manner as other automated modeling nodes, allowing you to experiment with multiple combinations of options in a single modeling pass. Models can be compared using basic measures with which to attempt to filter and rank the usefulness of the cluster models, and provide a measure based on the importance of particular fields.



The K-Means node clusters the data set into distinct groups (or clusters). The method defines a fixed number of clusters, iteratively assigns records to clusters, and adjusts the cluster centers until further refinement can no longer improve the model. Instead of trying to predict an outcome, *k*-means uses a process known as unsupervised learning to uncover patterns in the set of input fields.



The Kohonen node generates a type of neural network that can be used to cluster the data set into distinct groups. When the network is fully trained, records that are similar should be close together on the output map, while records that are different will be far apart. You can look at the number of observations captured by each unit in the model nugget to identify the strong units. This may give you a sense of the appropriate number of clusters.



The TwoStep node uses a two-step clustering method. The first step makes a single pass through the data to compress the raw input data into a manageable set of subclusters. The second step uses a hierarchical clustering method to progressively merge the subclusters into larger and larger clusters. TwoStep has the advantage of automatically estimating the optimal number of clusters for the training data. It can handle mixed field types and large data sets efficiently.



The Anomaly Detection node identifies unusual cases, or outliers, that do not conform to patterns of "normal" data. With this node, it is possible to identify outliers even if they do not fit any previously known patterns and even if you are not exactly sure what you are looking for.

#### **In-Database Mining Models**

IBM SPSS Modeler supports integration with data mining and modeling tools that are available from database vendors, including Oracle Data Miner and Microsoft Analysis Services. You can build, score, and store models inside the database—all from within the IBM SPSS Modeler application. For full details, see the *IBM SPSS Modeler In-Database Mining Guide*.

#### **IBM SPSS Statistics Models**

If you have a copy of IBM SPSS Statistics installed and licensed on your computer, you can access and run certain IBM SPSS Statistics routines from within IBM SPSS Modeler to build and score models.

## Building Split Models

---

Split modeling enables you to use a single stream to build separate models for each possible value of a flag, nominal, or continuous input field, with the resulting models all being accessible from a single model nugget. The possible values for the input fields could have very different effects on the model. With split modeling, you can easily build the best-fitting model for each possible field value in a single execution of the stream.

Note that interactive modeling sessions cannot use splitting. With interactive modeling you specify each model individually, so there would be no advantage in using splitting, which builds multiple models automatically.

Split modeling works by designating a particular input field as a split field. You can do this by setting the field role to **Split** in the Type specification.

You can designate only fields with a measurement level of **Flag**, **Nominal**, **Ordinal**, or **Continuous** as split fields.

You can assign more than one input field as a split field. In this case, however, the number of models created can be greatly increased. A model is built for each possible combination of the values of the selected split fields. For example, if three input fields, each having three possible values, are designated as split fields, this will result in the creation of 27 different models.

Even after you assign one or more fields as split fields, you can still choose whether to create split models or a single model, by means of a check box setting on the modeling node dialog.

If split fields are defined but the check box is not selected, only a single model is generated. Likewise if the check box is selected but no split field is defined, splitting is ignored and a single model is generated.

When you run the stream, separate models are built behind the scenes for each possible value of the split field or fields, but only a single model nugget is placed in the models palette and the stream canvas. A split-model nugget is denoted by the split symbol; this is two gray rectangles overlaid on the nugget image.

When you browse the split-model nugget, you see a list of all the separate models that have been built.

You can investigate an individual model from a list by double-clicking its nugget icon in the viewer. Doing so opens a standard browser window for the individual model. When the nugget is on the canvas, double-clicking a graph thumbnail opens the full-size graph. See the topic [“Split Model Viewer” on page 40](#) for more information.

Once a model has been created as a split model, you cannot remove the split processing from it, nor can you undo splitting further downstream from a split-modeling node or nugget.

**Example.** A national retailer wants to estimate sales by product category at each of its stores around the country. Using split modeling, they designate the Store field of their input data as a split field, enabling them to build separate models for each category at each store in a single operation. They can then use the resulting information to control stock levels much more accurately than they could with only a single model.

## Splitting and Partitioning

Splitting has some features in common with partitioning, but the two are used in very different ways.

**Partitioning** divides the dataset randomly into either two or three parts: training, testing and (optionally) validation, and is used to test the performance of a single model.

**Splitting** divides the dataset into as many parts as there are possible values for a split field, and is used to build multiple models.

Partitioning and splitting operate completely independently of each other. You can choose either, both or neither in a modeling node.

## Modeling nodes supporting split models

A number of modeling nodes can create split models. The exceptions are Auto Cluster, PCA/Factor, Feature Selection, SLRM, Random Trees, Tree-AS, Linear-AS, LSVM, the association models (Apriori, Carma and Sequence), the clustering models (K-Means, Kohonen, Two Step and Anomaly), Statistics models, and the nodes used for in-database modeling.

The modeling nodes that support split modeling are:

	C&R Tree		Bayes Net		Linear
	QUEST		GenLin		GLMM
	CHAID		KNN		STP
	C5.0		Cox		One-Class SVM
	Neural Net		Auto Classifier		XGBoost Tree
	Decision List		Auto Numeric		XGBoost Linear
	Regression		Logistic		HDBSCAN
	Discriminant		SVM		Time Series

## Features Affected by Splitting

The use of split models affects a number of IBM SPSS Modeler features in various ways. This section provides guidance on using split models with other nodes in a stream.

### Record Ops nodes

When you use split models in a stream that contains a Sample node, stratify records by the split field to achieve an even sampling of records. This option is available when you choose *Complex* as the sample method.

If the stream contains a Balance node, balancing applies to the overall set of input records, not to the subset of records inside a split.

When aggregating records by means of an Aggregate node, set the split fields to be key fields if you want to calculate aggregates for each split.

## Field Ops nodes

The Type node is where you specify which field or fields to use as split fields.

**Note:** While the Ensemble node is used to combine two or more model nuggets, it cannot be used to reverse the action of splitting, as the split models are contained inside a single model nugget.

## Modeling nodes

Split models do not support the calculation of predictor importance (the relative importance of the predictor input fields in estimating the model). Predictor importance settings are ignored when building split models.

**Note:** Adjusted propensity score settings are ignored when using a split model.

The KNN (nearest neighbor) node supports split models only if it is set to predict a target field. The alternative setting (only identify nearest neighbors) does not create a model. If the option **Automatically select k** is chosen, each of the split models might have a different number of nearest neighbors. Thus the overall model has a number of generated columns equal to the largest number of nearest neighbors that are found across all the split models. For those split models where the number of nearest neighbors is less than this maximum, there is a corresponding number of columns filled with \$null values. See the topic [“KNN node” on page 313](#) for more information.

## Database Modeling nodes

The in-database modeling nodes do not support split models.

## Model nuggets

**Export to PMML** from a split model nugget is not possible, as the nugget contains multiple models and PMML does not support such a packaging. Export to text or HTML is possible.

## Modeling Node Fields Options

---

All modeling nodes have a Fields tab, where you can specify the fields to be used in building the model.

Before you can build a model, you need to specify which fields you want to use as targets and as inputs. With a few exceptions, all modeling nodes will use field information from an upstream Type node. If you are using a Type node to select input and target fields, you don't need to change anything on this tab. (Exceptions include the Sequence node and the Text Extraction node, which require that field settings be specified in the modeling node.)

**Use type node settings.** This option tells the node to use field information from an upstream Type node. This is the default.

**Use custom settings.** This option tells the node to use field information specified here instead of that given in any upstream Type node(s). After selecting this option, specify the fields below as required.

*Note: Not all fields are displayed for all nodes.*

- **Use transactional format (Apriori, CARMA, MS Association Rules and Oracle Apriori nodes only).**

Select this check box if the source data is in **transactional format**. Records in this format have two fields, one for an ID and one for content. Each record represents a single transaction or item, and associated items are linked by having the same ID. Deselect this box if the data is in **tabular format**, in which items are represented by separate flags, where each flag field represents the presence or absence of a specific item and each record represents a complete set of associated items. See the topic [“Tabular versus Transactional Data” on page 232](#) for more information.

- **ID.** For transactional data, select an ID field from the list. Numeric or symbolic fields can be used as the ID field. Each unique value of this field should indicate a specific unit of analysis. For example, in a market basket application, each ID might represent a single customer. For a Web log analysis application, each ID might represent a computer (by IP address) or a user (by login data).



- **IDs are contiguous.** (Apriori and CARMA nodes only) If your data are presorted so that all records with the same ID are grouped together in the data stream, select this option to speed up processing. If your data are not presorted (or you are not sure), leave this option unselected and the node will sort the data automatically.

**Note:** If your data are not sorted and you select this option, you may get invalid results in your model.

- **Content.** Specify the content field(s) for the model. These fields contain the items of interest in association modeling. You can specify multiple flag fields (if data are in tabular format) or a single nominal field (if data are in transactional format).
- **Target.** For models that require one or more target fields, select the target field or fields. This is similar to setting the field role to *Target* in a Type node.
- **Evaluation.** (For Auto Cluster models only.) No target is specified for cluster models; however, you can select an evaluation field to identify its level of importance. In addition, you can evaluate how well the clusters differentiate values of this field, which in turn indicates whether the clusters can be used to predict this field. *Note* The evaluation field must be a string with more than one value.
- **Inputs.** Select the input field or fields. This is similar to setting the field role to *Input* in a Type node.
- **Partition.** This field allows you to specify a field used to partition the data into separate samples for the training, testing, and validation stages of model building. By using one sample to generate the model and a different sample to test it, you can get a good indication of how well the model will generalize to larger datasets that are similar to the current data. If multiple partition fields have been defined by using Type or Partition nodes, a single partition field must be selected on the Fields tab in each modeling node that uses partitioning. (If only one partition is present, it is automatically used whenever partitioning is enabled.) Also note that to apply the selected partition in your analysis, partitioning must also be enabled in the Model Options tab for the node. (Deselecting this option makes it possible to disable partitioning without changing field settings.)
- **Splits.** For split models, select the split field or fields. This is similar to setting the field role to *Split* in a Type node. You can designate only fields with a measurement level of **Flag**, **Nominal**, **Ordinal** or **Continuous** as split fields. Fields chosen as split fields cannot be used as target, input, partition, frequency or weight fields. See the topic [“Building Split Models” on page 24](#) for more information.
- **Use frequency field.** This option enables you to select a field as a frequency weight. Use this if the records in your training data represent more than one unit each--for example, if you are using aggregated data. The field values should be the number of units represented by each record. See the topic [“Using Frequency and Weight Fields” on page 27](#) for more information.

*Note:* If you see the error message **Metadata (on input/output fields) not valid**, ensure that you have specified all fields that are required, such as the frequency field.

- **Use weight field.** This option enables you to select a field as a case weight. Case weights are used to account for differences in variance across levels of the output field. See the topic [“Using Frequency and Weight Fields” on page 27](#) for more information.
- **Consequents.** For rule induction nodes (Apriori), select the fields to be used as consequents in the resulting rule set. (This corresponds to fields with role *Target* or *Both* in a Type node.)
- **Antecedents.** For rule induction nodes (Apriori), select the fields to be used as antecedents in the resulting rule set. (This corresponds to fields with role *Input* or *Both* in a Type node.)

Some models have a Fields tab that differs from those described in this section.

- See the topic [“Sequence Node Fields Options” on page 246](#) for more information.
- See the topic [“CARMA Node Fields Options” on page 235](#) for more information.

## Using Frequency and Weight Fields

Frequency and weight fields are used to give extra importance to some records over others, for example, because you know that one section of the population is under-represented in the training data (weight) or because one record represents a number of identical cases (frequency).

- Values for a frequency field should be positive integers. Records with a negative or zero frequency weight are excluded from the analysis. Non-integer frequency weights are rounded to the nearest integer.
- Case weight values should be positive but need not be integer values. Records with a negative or zero case weight are excluded from the analysis.

#### Scoring Frequency and Weight Fields

Frequency and weight fields are used in training models, but are not used in scoring, because the score for each record is based on its characteristics regardless of how many cases it represents. For example, suppose you have the data in the following table.

<i>Table 1. Data example</i>	
<b>Married</b>	<b>Responded</b>
Yes	Yes
Yes	Yes
Yes	Yes
Yes	No
No	Yes
No	No
No	No

Based on this, you conclude that three out of four married people respond to the promotion, and two out of three unmarried people didn't respond. So you will score any new records accordingly, as shown in the following table.

<i>Table 2. Scored records example</i>		
<b>Married</b>	<b>\$-Responded</b>	<b>\$RP-Responded</b>
Yes	Yes	0.75 (three/four)
No	No	0.67 (two/three)

Alternatively, you could store your training data more compactly, using a frequency field, as shown in the following table.

<i>Table 3. Scored records alternative example</i>		
<b>Married</b>	<b>Responded</b>	<b>Frequency</b>
Yes	Yes	3
Yes	No	1
No	Yes	1
No	No	2

Since this represents exactly the same dataset, you will build the same model and predict responses based solely on marital status. If you have ten married people in your scoring data, you will predict *Yes* for each of them regardless of whether they are presented as ten separate records, or one with a frequency value of 10. Weight, although generally not an integer, can be thought of as similarly indicating the importance of a record. This is why frequency and weight fields are not used when scoring records.

#### Evaluating and Comparing Models



Some model types support frequency fields, some support weight fields, and some support both. But in all cases where they apply, they are used only for model building and are not considered when evaluating models using an Evaluation node or Analysis node, or when ranking models using most of the methods supported by the Auto Classifier and Auto Numeric nodes.

- When comparing models (using evaluation charts, for example), frequency and weight values will be ignored. This enables a level comparison between models that use these fields and models that don't, but means that for an accurate evaluation, a dataset that accurately represents the population without relying on a frequency or weight field must be used. In practical terms, you can do this by making sure that models are evaluated using a testing sample in which the value of the frequency or weight field is always null or 1. (This restriction only applies when evaluating models; if frequency or weight values were always 1 for both training and testing samples, there would be no reason to use these fields in the first place.)
- If using Auto Classifier, frequency can be taken into account if ranking models based on Profit, so this method is recommended in that case.
- If necessary, you can split the data into training and testing samples using a Partition node.

## Modeling Node Analyze Options

---

Many modeling nodes include an Analyze tab that enables you to obtain predictor importance information along with raw and adjusted propensity scores.

### Model Evaluation

**Calculate predictor importance.** For models that produce an appropriate measure of importance, you can display a chart that indicates the relative importance of each predictor in estimating the model. Typically you will want to focus your modeling efforts on the predictors that matter most, and consider dropping or ignoring those that matter least. Note that predictor importance may take longer to calculate for some models, particularly when working with large datasets, and is off by default for some models as a result. Predictor importance is not available for decision list models. See [“Predictor Importance” on page 37](#) for more information.

### Propensity Scores

Propensity scores can be enabled in the modeling node, and on the Settings tab in the model nugget. This functionality is available only when the selected target is a flag field. See the topic [“Propensity Scores” on page 30](#) for more information.

**Calculate raw propensity scores.** Raw propensity scores are derived from the model based on the training data only. If the model predicts the *true* value (will respond), then the propensity is the same as  $P$ , where  $P$  is the probability of the prediction. If the model predicts the false value, then the propensity is calculated as  $(1 - P)$ .

- If you choose this option when building the model, propensity scores will be enabled in the model nugget by default. However, you can always choose to enable raw propensity scores in the model nugget whether or not you select them in the modeling node.
- When scoring the model, raw propensity scores will be added in a field with the letters *RP* appended to the standard prefix. For example, if the predictions are in a field named *\$R-churn*, the name of the propensity score field will be *\$RRP-churn*.

**Calculate adjusted propensity scores.** Raw propensities are based purely on estimates given by the model, which may be overfitted, leading to over-optimistic estimates of propensity. Adjusted propensities attempt to compensate by looking at how the model performs on the test or validation partitions and adjusting the propensities to give a better estimate accordingly.

- This setting requires that a valid partition field is present in the stream.
- Unlike raw confidence scores, adjusted propensity scores must be calculated when building the model; otherwise, they will not be available when scoring the model nugget.
- When scoring the model, adjusted propensity scores will be added in a field with the letters *AP* appended to the standard prefix. For example, if the predictions are in a field named *\$R-churn*, the

name of the propensity score field will be *\$RAP-churn*. Adjusted propensity scores are not available for logistic regression models.

- When calculating the adjusted propensity scores, the test or validation partition used for the calculation must not have been balanced. To avoid this, be sure the **Only balance training data** option is selected in any upstream Balance nodes. In addition, if a complex sample has been taken upstream this will invalidate the adjusted propensity scores.
- Adjusted propensity scores are not available for "boosted" tree and rule set models. See the topic [“Boosted C5.0 Models” on page 107](#) for more information.

**Based on.** For adjusted propensity scores to be computed, a partition field must be present in the stream. You can specify whether to use the testing or validation partition for this computation. For best results, the testing or validation partition should include at least as many records as the partition used to train the original model.

## Propensity Scores

For models that return a *yes* or *no* prediction, you can request propensity scores in addition to the standard prediction and confidence values. Propensity scores indicate the likelihood of a particular outcome or response. The following table contains an example.

Table 4. Propensity scores	
Customer	Propensity to respond
Joe Smith	35%
Jane Smith	15%

Propensity scores are available only for models with flag targets, and indicate the likelihood of the *True* value defined for the field, as specified in a source or Type node.

### Propensity Scores Versus Confidence Scores

Propensity scores differ from confidence scores, which apply to the current prediction, whether *yes* or *no*. In cases where the prediction is *no*, for example, a high confidence actually means a high likelihood *not* to respond. Propensity scores sidestep this limitation to enable easier comparison across all records. For example, a *no* prediction with a confidence of 0.85 translates to a raw propensity of 0.15 (or 1 minus 0.85).

Table 5. Confidence scores		
Customer	Prediction	Confidence
Joe Smith	Will respond	.35
Jane Smith	Won't respond	.85

### Obtaining Propensity Scores

- Propensity scores can be enabled on the Analyze tab in the modeling node or on the Settings tab in the model nugget. This functionality is available only when the selected target is a flag field. See the topic [“Modeling Node Analyze Options” on page 29](#) for more information.
- Propensity scores may also be calculated by the Ensemble node, depending on the ensemble method used.

### Calculating Adjusted Propensity Scores

Adjusted propensity scores are calculated as part of the process of building the model, and will not be available otherwise. Once the model is built, it is then scored using data from the test or validation partition, and a new model to deliver adjusted propensity scores is constructed by analyzing the original model's performance on that partition. Depending on the type of model, one of two methods may be used to calculate the adjusted propensity scores.

- For rule set and tree models, adjusted propensity scores are generated by recalculating the frequency of each category at each tree node (for tree models) or the support and confidence of each rule (for rule set models). This results in a new rule set or tree model which is stored with the original model, to be used whenever adjusted propensity scores are requested. Each time the original model is applied to new data, the new model can subsequently be applied to the raw propensity scores to generate the adjusted scores.
- For other models, records produced by scoring the original model on the test or validation partition are then binned by their raw propensity score. Next, a neural network model is trained that defines a non-linear function that maps from the mean raw propensity in each bin to the mean observed propensity in the same bin. As noted earlier for tree models, the resulting neural net model is stored with the original model, and can be applied to the raw propensity scores whenever adjusted propensity scores are requested.

**Caution regarding missing values in the testing partition.** Handling of missing input values in the testing/validation partition varies by model (see individual model scoring algorithms for details). The C5 model cannot compute adjusted propensities when there are missing inputs.

## Misclassification Costs

---

In some contexts, certain kinds of errors are more costly than others. For example, it may be more costly to classify a high-risk credit applicant as low risk (one kind of error) than it is to classify a low-risk applicant as high risk (a different kind of error). Misclassification costs allow you to specify the relative importance of different kinds of prediction errors.

Misclassification costs are basically weights applied to specific outcomes. These weights are factored into the model and may actually change the prediction (as a way of protecting against costly mistakes).

With the exception of C5.0 models, misclassification costs are not applied when scoring a model and are not taken into account when ranking or comparing models using an Auto Classifier node, evaluation chart, or Analysis node. A model that includes costs may not produce fewer errors than one that doesn't and may not rank any higher in terms of overall accuracy, but it is likely to perform better in practical terms because it has a built-in bias in favor of *less expensive* errors.

The cost matrix shows the cost for each possible combination of predicted category and actual category. By default, all misclassification costs are set to 1.0. To enter custom cost values, select **Use misclassification costs** and enter your custom values into the cost matrix.

To change a misclassification cost, select the cell corresponding to the desired combination of predicted and actual values, delete the existing contents of the cell, and enter the desired cost for the cell. Costs are not automatically symmetrical. For example, if you set the cost of misclassifying *A* as *B* to be 2.0, the cost of misclassifying *B* as *A* will still have the default value of 1.0 unless you explicitly change it as well.

**Note:** Only the Decision Trees model allows costs to be specified at build time.

## Model Nuggets

---



Figure 19. Model nugget

A model nugget is a container for a model, that is, the set of rules, formulas or equations that represent the results of your model building operations in SPSS Modeler. The main purpose of a nugget is for scoring data to generate predictions, or to enable further analysis of the model properties. Opening a model nugget on the screen enables you to see various details about the model, such as the relative importance of the input fields in creating the model. To view the predictions, you need to attach and execute a further process or output node. See the topic [“Using Model Nuggets in Streams”](#) on page 41 for more information.



Figure 20. Model link from modeling node to model nugget

When you successfully execute a modeling node, a corresponding model nugget is placed on the stream canvas, where it is represented by a gold, diamond-shaped icon (hence the name "nugget"). On the stream canvas, the nugget is shown with a connection (solid line) to the nearest suitable node before the modeling node, and a link (dotted line) to the modeling node itself.

The nugget is also placed in the Models palette in the upper right corner of the IBM SPSS Modeler window. From either location, nuggets can be selected and browsed to view details of the model.

Nuggets are always placed in the Models palette when a modeling node is successfully executed. You can set a user option to control whether the nugget is additionally placed on the stream canvas.

The following topics provide information on using model nuggets in IBM SPSS Modeler. For an in-depth understanding of the algorithms used, see the *IBM SPSS Modeler Algorithms Guide*, available as a PDF file as part of your product download.

## Model Links

By default, a nugget is shown on the canvas with a link to the modeling node that created it. This is especially useful in complex streams with several nuggets, enabling you to identify the nugget that will be updated by each modeling node. Each link contains a symbol to indicate whether the model is replaced when the modeling node is executed. See the topic [“Replacing a model” on page 33](#) for more information.

### Defining and Removing Model Links

You can define and remove links manually on the canvas. When you are defining a new link, the cursor changes to the link cursor.



Figure 21. Link cursor

Defining a new link (context menu)

1. Right-click on the modeling node from which you want the link to start.
2. Choose **Define Model Link** from the context menu.
3. Click the nugget where you want the link to end.

Defining a new link (main menu)

1. Click the modeling node from which you want the link to start.
2. From the main menu, choose:

**Edit > Node > Define Model Link**

3. Click the nugget where you want the link to end.

Removing an existing link (context menu)

1. Right-click on the nugget at the end of the link.
2. Choose **Remove Model Link** from the context menu.

Alternatively:

1. Right-click on the symbol in the middle of the link.
2. Choose **Remove Link** from the context menu.

Removing an existing link (main menu)

1. Click the modeling node or nugget from which you want to remove the link.
2. From the main menu, choose:

**Edit > Node > Remove Model Link**

### Copying and Pasting Model Links

If you copy a linked nugget, without its modeling node, and paste it into the same stream, the nugget is pasted with a link to the modeling node. The new link has the same model replacement status (see [“Replacing a model”](#) on page 33) as the original link.

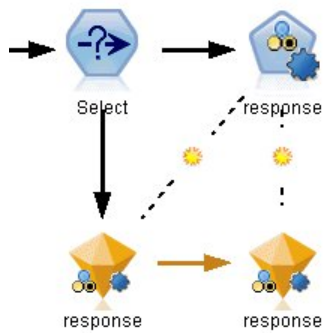


Figure 22. Copying and pasting a linked nugget

If you copy and paste a nugget together with its linked modeling node, the link is retained whether the objects are pasted into the same stream or a new stream.

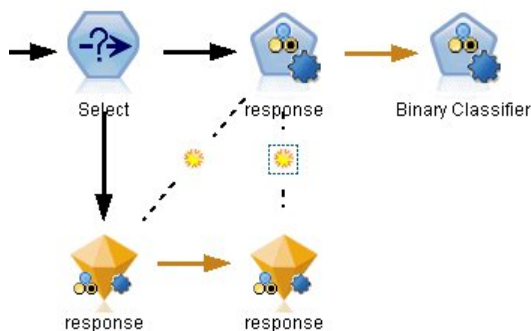


Figure 23. Copying and pasting a linked nugget

*Note:* If you copy a linked nugget, without its modeling node, and paste the nugget into a new stream (or into a SuperNode that does not contain the modeling node), the link is broken and only the nugget is pasted.

### Model Links and SuperNodes

If you define a SuperNode to include either the modeling node or the model nugget of a linked model (but not both), the link is broken. Expanding the SuperNode does not restore the link; you can only do this by undoing creation of the SuperNode.

## Replacing a model

You can choose whether to replace (that is, update) an existing nugget on re-execution of the modeling node that created the nugget. If you turn off the replacement option, a new nugget is created when you re-execute the modeling node.

Each link from modeling node to nugget contains a symbol to indicate whether the model is replaced when the modeling node is re-executed.



Figure 24. Model link with model replacement turned on

The link is initially shown with model replacement turned on, depicted by the small sunburst symbol in the link. In this state, re-executing the modeling node at one end of the link simply updates the nugget at the other end.



Figure 25. Model link with model replacement turned off

If model replacement is turned off, the link symbol is replaced by a gray dot. In this state, re-executing the modeling node at one end of the link adds a new, updated version of the nugget to the canvas.

In either case, in the Models palette the existing nugget is updated or a new nugget is added, depending on the setting of the **Replace previous model** system option.

### Order of execution

When you execute a stream with multiple branches containing model nuggets, the stream is first evaluated to make sure that a branch with model replacement turned on is executed before any branch that uses the resulting model nugget.

If your requirements are more complex, you can set the order of execution manually through scripting.

### Changing the model replacement setting

1. Right-click the symbol on the link.
2. Choose **Turn On(Off) Model Replacement** as desired.

**Note:** The model replacement setting on a model link overrides the setting on the Notifications tab of the User Options dialog (Tools > Options > User Options).

## The models palette

The models palette (on the Models tab in the managers window) enables you to use, examine, and modify model nuggets in various ways.

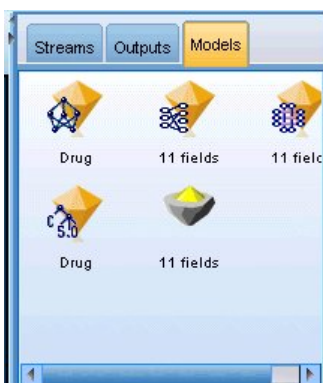


Figure 26. Models palette

Right-clicking a model nugget in the models palette opens a context menu with the following options:

- **Add To Stream.** Adds the model nugget to the currently active stream. If there is a selected node in the stream, the model nugget will be connected to the selected node when such a connection is possible, or otherwise to the nearest possible node. The nugget is displayed with a link to the modeling node that created the model, if that node is still in the stream.
- **Browse.** Opens the model browser for the nugget.
- **Rename and Annotate.** Allows you to rename the model nugget and/or modify the annotation for the nugget.
- **Generate Modeling Node.** If you have a model nugget that you want to modify or update and the stream used to create the model is not available, you can use this option to recreate a modeling node with the same options used to create the original model.
- **Save Model, Save Model As.** Saves the model nugget to an external generated model (.gm) binary file.
- **Store Model.** Stores the model nugget in IBM SPSS Collaboration and Deployment Services Repository.
- **Export PMML.** Exports the model nugget as predictive model markup language (PMML), which can be used for scoring new data outside of IBM SPSS Modeler. **Export PMML** is available for all generated model nodes.
- **Add to Project.** Saves the model nugget and adds it to the current project. On the Classes tab, the nugget will be added to the Generated Models folder. On the CRISP-DM tab, it will be added to the default project phase.
- **Delete.** Deletes the model nugget from the palette.

Right-clicking an unoccupied area in the models palette opens a context menu with the following options:

- **Open Model.** Loads a model nugget previously created in IBM SPSS Modeler.
- **Retrieve Model.** Retrieves a stored model from an IBM SPSS Collaboration and Deployment Services repository.
- **Load Palette.** Loads a saved models palette from an external file.
- **Retrieve Palette.** Retrieves a stored models palette from an IBM SPSS Collaboration and Deployment Services repository.
- **Save Palette.** Saves the entire contents of the models palette to an external generated models palette (.gen) file.
- **Store Palette.** Stores the entire contents of the models palette in an IBM SPSS Collaboration and Deployment Services repository.
- **Clear Palette.** Deletes all nuggets from the palette.
- **Add Palette To Project.** Saves the models palette and adds it to the current project. On the Classes tab, the nugget will be added to the Generated Models folder. On the CRISP-DM tab, it will be added to the default project phase.
- **Import PMML.** Loads a model from an external file. You can open, browse, and score PMML models created by IBM SPSS Statistics or other applications that support this format. See the topic [“Importing and exporting models as PMML” on page 41](#) for more information.

## Browsing model nuggets

The model nugget browsers enable you to examine and use the results of your models. From the browser, you can save, print, or export the generated model, examine the model summary, and view or edit annotations for the model. For some types of model nugget, you can also generate new nodes, such as Filter nodes or Rule Set nodes. For some models, you can also view model parameters, such as rules or cluster centers. For some types of models (tree-based models and cluster models), you can view a graphical representation of the structure of the model. Controls for using the model nugget browsers are described below.



## Menus

**File menu.** All model nuggets have a File menu, containing some subset of the following options:

- **Save Node.** Saves the model nugget to a node (.nod) file.
- **Store Node.** Stores the model nugget in an IBM SPSS Collaboration and Deployment Services repository.
- **Header and Footer.** Allows you to edit the page header and footer for printing from the nugget.
- **Page Setup.** Allows you to change the page setup for printing from the nugget.
- **Print Preview.** Displays a preview of how the nugget will look when printed. Select the information you want to preview from the submenu.
- **Print.** Prints the contents of the nugget. Select the information you want to print from the submenu.
- **Print View.** Prints the current view or all views.
- **Export Text.** Exports the contents of the nugget to a text file. Select the information you want to export from the submenu.
- **Export HTML.** Exports the contents of the nugget to an HTML file. Select the information you want to export from the submenu.
- **Export PMML.** Exports the model as predictive model markup language (PMML), which can be used with other PMML-compatible software. See the topic [“Importing and exporting models as PMML”](#) on page 41 for more information.
- **Export SQL.** Exports the model as structured query language (SQL), which can be edited and used with other databases.  
  
**Note:** SQL Export is available only from the following models: C5, C&RT, CHAID, QUEST, Linear Regression, Logistic Regression, Neural Net, PCA/Factor, and Decision List models.
- **Publish for Server Scoring Adapter.** Publishes the model to a database that has a scoring adapter installed, enabling model scoring to be performed within the database. See the topic [“Publishing models for a scoring adapter”](#) on page 43 for more information.

**Generate menu.** Most model nuggets also have a Generate menu, enabling you to generate new nodes based on the model nugget. The options available from this menu will depend on the type of model you are browsing. See the specific model nugget type for details about what you can generate from a particular model.

**View menu.** On the Model tab of a nugget, this menu enables you to display or hide the various visualization toolbars that are available in the current mode. To make the full set of toolbars available, select Edit Mode (the paintbrush icon) from the General toolbar.

**Preview button.** Some model nuggets have a Preview button, which enables you to see a sample of the model data, including the extra fields created by the modeling process. The default number of rows displayed is 10; however, you can change this in the stream properties.

**Add to Current Project button.** Saves the model nugget and adds it to the current project. On the Classes tab, the nugget will be added to the Generated Models folder. On the CRISP-DM tab, it will be added to the default project phase.

## Model Nugget Summary / Information

The Summary tab or Information view for a model nugget displays information about the fields, build settings, and model estimation process. Results are presented in a tree view that can be expanded or collapsed by clicking specific items.

**Analysis.** Displays information about the model. Specific details vary by model type, and are covered in the section for each model nugget. In addition, if you execute an Analysis node that is attached to this modeling node, information from that analysis is also displayed in this section.

**Fields.** Lists the fields that are used as the target and the inputs in building the model. For split models, also lists the fields that determined the splits.



**Note:** In the Information view for neural networks, linear, and other models with either boosting or bagging modes, the icon that is shown is the same (nominal icon), regardless of whether the type is a flag, nominal, or ordinal.

**Build Settings / Options.** Contains information about the settings that are used in building the model.

**Training Summary.** Shows the type of model, the stream that is used to create it, the user who created it, when it was built, and the elapsed time for building the model. Note that the elapsed time for building the model is only available on the Summary tab, not the Information view.

## Predictor Importance

Typically, you will want to focus your modeling efforts on the predictor fields that matter most and consider dropping or ignoring those that matter least. The predictor importance chart helps you do this by indicating the relative importance of each predictor in estimating the model. Since the values are relative, the sum of the values for all predictors on the display is 1.0. Predictor importance does not relate to model accuracy. It just relates to the importance of each predictor in making a prediction, not whether or not the prediction is accurate.

Predictor importance is available for models that produce an appropriate statistical measure of importance, including neural networks, decision trees (C&R Tree, C5.0, CHAID, and QUEST), Bayesian networks, discriminant, SVM, and SLRM models, linear and logistic regression, generalized linear, and nearest neighbor (KNN) models. For most of these models, predictor importance can be enabled on the Analyze tab in the modeling node. See the topic [“Modeling Node Analyze Options”](#) on page 29 for more information. For KNN models, see [“Neighbors”](#) on page 315.

*Note:* Predictor importance is not supported for split models. Predictor importance settings are ignored when building split models. See the topic [“Building Split Models”](#) on page 24 for more information.

Calculating predictor importance may take significantly longer than model building, particularly when using large datasets. It takes longer to calculate for SVM and logistic regression than for other models, and is disabled for these models by default. If using a dataset with a large number of predictors, initial screening using a Feature Selection node may give faster results (see below).

- Predictor importance is calculated from the test partition, if available. Otherwise the training data is used.
- For SLRM models, predictor importance is available but is computed by the SLRM algorithm. See the topic [“SLRM Model Nuggets”](#) on page 301 for more information.
- You can use IBM SPSS Modeler's graph tools to interact, edit, and save the graph.
- Optionally, you can generate a Filter node based on the information in the predictor importance chart. See the topic [“Filtering Variables Based on Importance”](#) on page 38 for more information.

### Predictor Importance and Feature Selection

The predictor importance chart displayed in a model nugget may seem to give results similar to the Feature Selection node in some cases. While feature selection ranks each input field based on the strength of its relationship to the specified target, independent of other inputs, the predictor importance chart indicates the relative importance of each input for *this* particular model. Thus feature selection will be more conservative in screening inputs. For example, if *job title* and *job category* are both strongly related to salary, then feature selection would indicate that both are important. But in modeling, interactions and correlations are also taken into consideration. Thus you might find that only one of two inputs is used if both duplicate much of the same information. In practice, feature selection is most useful for preliminary screening, particularly when dealing with large datasets with large numbers of variables, and predictor importance is more useful in fine-tuning the model.

### Predictor Importance Differences Between Single Models and Automated Modeling Nodes

Depending on whether you are creating a single model from an individual node, or using an automated modelling node to produce results, you may see slight differences in the predictor importance. Such differences in implementation are due to some engineering restrictions.

For example, with single classifiers such as CHAID the calculation applies a stopping rule and uses probability values when computing importance values. In contrast, the Auto Classifier does not use a stopping rule and uses predicted labels directly in the calculation. These differences can mean that if you produce a single model using Auto Classifier, the importance value can be considered as a rough estimation, compared with that computed for a single classifier. To obtain the most accurate predictor importance values we suggest using a single node instead of the automated modelling nodes.

### Filtering Variables Based on Importance

Optionally, you can generate a Filter node based on the information in the predictor importance chart.

Mark the predictors you want to include on the chart, if applicable, and from the menus choose:

#### Generate > Filter Node (Predictor Importance)

OR

#### > Field Selection (Predictor Importance)

**Top number of variables.** Includes or excludes the most important predictors up to the specified number.

**Importance greater than.** Includes or excludes all predictors with relative importance greater than the specified value.

## Ensemble Viewer

### Models for ensembles

The model for an ensemble provides information about the component models in the ensemble and the performance of the ensemble as a whole.

The main (view-independent) toolbar allows you to choose whether to use the ensemble or a reference model for scoring. If the ensemble is used for scoring you can also select the combining rule. These changes do not require model re-execution; however, these choices are saved to the model (nugget) for scoring and/or downstream model evaluation. They also affect PMML exported from the ensemble viewer.

**Combining Rule.** When scoring an ensemble, this is the rule used to combine the predicted values from the base models to compute the ensemble score value.

- Ensemble predicted values for **categorical** targets can be combined using voting, highest probability, or highest mean probability. **Voting** selects the category that has the highest probability most often across the base models. **Highest probability** selects the category that achieves the single highest probability across all base models. **Highest mean probability** selects the category with the highest value when the category probabilities are averaged across base models.
- Ensemble predicted values for **continuous** targets can be combined using the mean or median of the predicted values from the base models.

The default is taken from the specifications made during model building. Changing the combining rule recomputes the model accuracy and updates all views of model accuracy. The Predictor Importance chart also updates. This control is disabled if the reference model is selected for scoring.

**Show All Combining rules.** When selected, results for all available combining rules are shown in the model quality chart. The Component Model Accuracy chart is also updated to show reference lines for each voting method.

### Model Summary

The Model Summary view is a snapshot, at-a-glance summary of the ensemble quality and diversity.

**Quality.** The chart displays the accuracy of the final model, compared to a reference model and a naive model. Accuracy is presented in larger is better format; the "best" model will have the highest accuracy. For a categorical target, accuracy is simply the percentage of records for which the predicted value matches the observed value. For a continuous target, accuracy is 1 minus the ratio of the mean absolute

error in prediction (the average of the absolute values of the predicted values minus the observed values) to the range of predicted values (the maximum predicted value minus the minimum predicted value).

For bagging ensembles, the reference model is a standard model built on the whole training partition. For boosted ensembles, the reference model is the first component model.

The naive model represents the accuracy if no model were built, and assigns all records to the modal category. The naive model is not computed for continuous targets.

**Diversity.** The chart displays the "diversity of opinion" among the component models used to build the ensemble, presented in larger is more diverse format. It is a measure of how much predictions vary across the base models. Diversity is not available for boosted ensemble models, nor is it shown for continuous targets.

### *Predictor Importance*

Typically, you will want to focus your modeling efforts on the predictor fields that matter most and consider dropping or ignoring those that matter least. The predictor importance chart helps you do this by indicating the relative importance of each predictor in estimating the model. Since the values are relative, the sum of the values for all predictors on the display is 1.0. Predictor importance does not relate to model accuracy. It just relates to the importance of each predictor in making a prediction, not whether or not the prediction is accurate.

Predictor importance is not available for all ensemble models. The predictor set may vary across component models, but importance can be computed for predictors used in at least one component model.

### *Predictor Frequency*

The predictor set can vary across component models due to the choice of modeling method or predictor selection. The Predictor Frequency plot is a dot plot that shows the distribution of predictors across component models in the ensemble. Each dot represents one or more component models containing the predictor. Predictors are plotted on the y-axis, and are sorted in descending order of frequency; thus the topmost predictor is the one that is used in the greatest number of component models and the bottommost one is the one that was used in the fewest. The top 10 predictors are shown.

Predictors that appear most frequently are typically the most important. This plot is not useful for methods in which the predictor set cannot vary across component models.

### *Component Model Accuracy*

The chart is a dot plot of predictive accuracy for component models. Each dot represents one or more component models with the level of accuracy plotted on the y-axis. Hover over any dot to obtain information on the corresponding individual component model.

**Reference lines.** The plot displays color coded lines for the ensemble as well as the reference model and naïve models. A checkmark appears next to the line corresponding to the model that will be used for scoring.

**Interactivity.** The chart updates if you change the combining rule.

**Boosted ensembles.** A line chart is displayed for boosted ensembles.

### *Component Model Details*

The table displays information on component models, listed by row. By default, component models are sorted in ascending model number order. You can sort the rows in ascending or descending order by the values of any column.

**Model.** A number representing the sequential order in which the component model was created.

**Accuracy.** Overall accuracy formatted as a percentage.

**Method.** The modeling method.

**Predictors.** The number of predictors used in the component model.

**Model Size.** Model size depends on the modeling method: for trees, it is the number of nodes in the tree; for linear models, it is the number of coefficients; for neural networks, it is the number of synapses.

**Records.** The weighted number of input records in the training sample.

### *Automatic Data Preparation*

This view shows information about which fields were excluded and how transformed fields were derived in the automatic data preparation (ADP) step. For each field that was transformed or excluded, the table lists the field name, its role in the analysis, and the action taken by the ADP step. Fields are sorted by ascending alphabetical order of field names.

The action **Trim outliers**, if shown, indicates that values of continuous predictors that lie beyond a cutoff value (3 standard deviations from the mean) have been set to the cutoff value.

## **Model Nuggets for Split Models**

The model nugget for a split model provides access to all the separate models created by the splits.

A split-model nugget contains:

- a list of all the split models created, together with a set of statistics about each model
- information about the overall model

From the list of split models, you can open up individual models to examine them further.

### **Split Model Viewer**

The Model tab lists all the models contained in the nugget, and provides statistics in various forms about the split models. It has two general forms, depending on the modeling node.

**Sort by.** Use this list to choose the order in which the models are listed. You can sort the list based on the values of any of the display columns, in ascending or descending order. Alternatively, click on a column heading to sort the list by that column. Default is descending order of overall accuracy.

**Show/hide columns menu.** Click this button to display a menu from where you can choose individual columns to show or hide.

**View.** If you are using partitioning, you can choose to view the results for either the training data or the testing data.

For each split, the details shown are as follows:

**Graph.** A thumbnail indicating the data distribution for this model. When the nugget is on the canvas, double-click the thumbnail to open the full-size graph.

**Model.** An icon of the model type. Double-click the icon to open the model nugget for this particular split.

**Split fields.** The fields designated in the modeling node as split fields, with their various possible values.

**No. Records in Split.** The number of records involved in this particular split.

**No. Fields Used.** Ranks split models based on the number of input fields used.

**Overall Accuracy (%).** The percentage of records that is correctly predicted by the split model relative to the total number of records in that split.

**Split.** The column heading shows the field(s) used to create splits, and the cells are the split values. Double-click any split to open a Model Viewer for the model built for that split.

**Accuracy.** Overall accuracy formatted as a percentage.

**Model Size.** Model size depends on the modeling method: for trees, it is the number of nodes in the tree; for linear models, it is the number of coefficients; for neural networks, it is the number of synapses.

**Records.** The weighted number of input records in the training sample.

## Using Model Nuggets in Streams

Model nuggets are placed in streams to enable you to score new data and generate new nodes. **Scoring** data enables you to use the information gained from model building to create predictions for new records. To see the results of scoring, you need to attach a terminal node (that is, a processing or output node) to the nugget and execute the terminal node.

For some models, model nuggets can also give you additional information about the quality of the prediction, such as confidence values or distances from cluster centers. Generating new nodes enables you to easily create new nodes based on the structure of the generated model. For example, most models that perform input field selection enable you to generate Filter nodes that will pass only input fields that the model identified as important.

**Note:** There can be small differences in the scores assigned to a given case by a given model when executed in different versions of IBM SPSS Modeler. This is usually a result of enhancements to the software between versions.

To Use a Model Nugget for Scoring Data

1. Connect the model nugget to a data source or stream that will pass data to it.
2. Add or connect one or more processing or output nodes (such as a Table or Analysis node) to the model nugget.
3. Execute one of the nodes downstream from the model nugget.

*Note:* You cannot use the Unrefined Rule node for scoring data. To score data based on an association rule model, use the Unrefined Rule node to generate a Rule Set nugget, and use the Rule Set nugget for scoring. See the topic [“Generating a Rule Set from an Association Model Nugget”](#) on page 242 for more information.

To Use a Model Nugget for Generating Processing Nodes

1. On the palette, browse the model, or, on the stream canvas, edit the model.
2. Select the desired node type from the Generate menu of the model nugget browser window. The options available will vary, depending on the type of model nugget. See the specific model nugget type for details about what you can generate from a particular model.

## Regenerating a modeling node

If you have a model nugget that you want to modify or update and the stream used to create the model is not available, you can regenerate a modeling node with the same options used to create the original model.

To rebuild a model, right-click on the model in the models palette and choose **Generate Modeling Node**.

Alternatively, when browsing any model, choose **Generate Modeling Node** from the Generate menu.

The regenerated modeling node should be functionally identical to the one used to create the original model in most cases.

- For Decision Tree models, additional settings specified during the interactive session may also be stored with the node, and the **Use tree directives** option will be enabled in the regenerated modeling node.
- For Decision List models, the **Use saved interactive session information** option will be enabled. See the topic [“Decision List Model Options”](#) on page 134 for more information.
- For Time Series models, the **Continue estimation using existing model(s)** option is enabled, which enables you to regenerate the previous model with current data.

## Importing and exporting models as PMML

PMML, or predictive model markup language, is an XML format for describing data mining and statistical models, including inputs to the models, transformations used to prepare data for data mining, and the

parameters that define the models themselves. IBM SPSS Modeler can import and export PMML, making it possible to share models with other applications that support this format, such as IBM SPSS Statistics. For more information about PMML, see the Data Mining Group website (<http://www.dmg.org>).

### To export a model

PMML export is supported for most of the model types generated in IBM SPSS Modeler. See the topic [“Model types supporting PMML” on page 42](#) for more information.

1. Right-click a model nugget on the models palette. (Alternatively, double-click a model nugget on the canvas and select the File menu.)
2. On the menu, click **Export PMML**.
3. In the Export (or Save) dialog box, specify a target directory and a unique name for the model.

#### Note:

You can change options for PMML export in the User Options dialog box. On the main menu, click:

**Tools > Options > User Options**

and click the PMML tab.

### To import a model saved as PMML

Models exported as PMML from IBM SPSS Modeler or another application can be imported into the models palette. See the topic [“Model types supporting PMML” on page 42](#) for more information.

1. In the models palette, right-click the palette and select **Import PMML** from the menu.
2. Select the file to import and specify options for variable labels as required.
3. Click **Open**.

**Use variable labels if present in model.** The PMML may specify both variable names and variable labels (such as Referrer ID for *RefID*) for variables in the data dictionary. Select this option to use variable labels if they are present in the originally exported PMML.

If you have selected the variable label option but there are no variable labels in the PMML, the variable names are used as normal.

### Model types supporting PMML

#### PMML Export

**IBM SPSS Modeler models.** The following models created in IBM SPSS Modeler can be exported as PMML 4.0:

- C&R Tree
- QUEST
- CHAID
- Neural Net
- C5.0
- Logistic Regression
- Genlin
- SVM
- Apriori
- Carma
- K-Means
- Kohonen

- TwoStep
- TwoStep-AS
- GLMM (PMML is exported for all GLMM models, but the PMML has only fixed effects)
- Decision List
- Cox
- Sequence (scoring for Sequence PMML models is not supported)
- Random Trees
- Tree-AS
- Linear
- Linear-AS
- Regression
- Logistic
- GLE
- LSVM
- Anomaly Detection
- KNN
- Association Rules

**Database native models.** For models generated using database-native algorithms, PMML export is not available. Models created using Analysis Services from Microsoft or Oracle Data Miner cannot be exported.

### PMML Import

IBM SPSS Modeler can import and score PMML models generated by current versions of all IBM SPSS Statistics products, including models exported from IBM SPSS Modeler as well as model or transformation PMML generated by IBM SPSS Statistics 17.0 or later. Essentially, this means any PMML that the scoring engine can score, with the following exceptions:

- Apriori, CARMA, Anomaly Detection, Sequence, and Association Rules models cannot be imported.
- PMML models may not be browsed after importing into IBM SPSS Modeler even though they can be used in scoring. (Note that this includes models that were exported from IBM SPSS Modeler to begin with. To avoid this limitation, export the model as a generated model file [`*.gm`] rather than PMML.)
- Limited validation occurs on import, but full validation is performed on attempting to score the model. Thus it is possible for import to succeed but scoring to fail or produce incorrect results.

**Note:** For third party PMML imported into IBM SPSS Modeler, IBM SPSS Modeler will attempt to score valid PMML that can be recognized and scored. But it is not guaranteed that all PMML will score or that it will score in the same way as the application that generated it.

## Publishing models for a scoring adapter

You can publish models to a database server that has a scoring adapter installed. A scoring adapter enables model scoring to be performed within the database by using the user-defined function (UDF) capabilities of the database. Performing scoring in the database avoids the need to extract the data before scoring. Publishing to a scoring adapter also generates some example SQL to execute the UDF.

### To publish a scoring adapter

1. Double-click the model nugget to open it.
2. From the model nugget menu, choose:  
**File > Publish for Server Scoring Adapter**
3. Fill in the relevant fields on the dialog box and click **OK**.

**Database connection.** The connection details for the database you want to use for the model.

**Publish ID.** (Db2 for z/OS databases only) An identifier for the model. If you rebuild the same model and use the same publish ID, the generated SQL remains the same, so it is possible to rebuild a model without having to change the application that uses the SQL previously generated. (For other databases the generated SQL is unique to the model.)

**Generate Example SQL.** If selected, generates the example SQL into the file specified in the **File** field.

## Unrefined Models

An unrefined model contains information extracted from the data but is not designed for generating predictions directly. This means that it cannot be added to streams. Unrefined models are displayed as “diamonds in the rough” on the generated models palette.



*Figure 27. Unrefined model icon*

To see information about the unrefined rule model, right-click the model and choose **Browse** from the context menu. Like other models generated in IBM SPSS Modeler, the various tabs provide summary and rule information about the model created.

**Generating nodes.** The Generate menu enables you to create new nodes based on the rules.

- **Select Node.** Generates a Select node to select records to which the currently selected rule applies. This option is disabled if no rule is selected.
- **Rule set.** Generates a Rule Set node to predict values for a single target field. See the topic [“Generating a Rule Set from an Association Model Nugget”](#) on page 242 for more information.



---

# Chapter 4. Screening Models

## Screening Fields and Records

---

Several modeling nodes can be used during the preliminary stages of an analysis in order to locate fields and records that are most likely to be of interest in modeling. You can use the Feature Selection node to screen and rank fields by importance and the Anomaly Detection node to locate unusual records that do not conform to the known patterns of "normal" data.



The Feature Selection node screens input fields for removal based on a set of criteria (such as the percentage of missing values); it then ranks the importance of remaining inputs relative to a specified target. For example, given a data set with hundreds of potential inputs, which are most likely to be useful in modeling patient outcomes?



The Anomaly Detection node identifies unusual cases, or outliers, that do not conform to patterns of "normal" data. With this node, it is possible to identify outliers even if they do not fit any previously known patterns and even if you are not exactly sure what you are looking for.

Note that anomaly detection identifies unusual records or cases through cluster analysis based on the set of fields selected in the model without regard for any specific target (dependent) field and regardless of whether those fields are relevant to the pattern you are trying to predict. For this reason, you may want to use anomaly detection in combination with feature selection or another technique for screening and ranking fields. For example, you can use feature selection to identify the most important fields relative to a specific target and then use anomaly detection to locate the records that are the most unusual with respect to those fields. (An alternative approach would be to build a decision tree model and then examine any misclassified records as potential anomalies. However, this method would be more difficult to replicate or automate on a large scale.)

## Feature Selection node

---

Data mining problems may involve hundreds, or even thousands, of fields that can potentially be used as inputs. As a result, a great deal of time and effort may be spent examining which fields or variables to include in the model. To narrow down the choices, the Feature Selection algorithm can be used to identify the fields that are most important for a given analysis. For example, if you are trying to predict patient outcomes based on a number of factors, which factors are the most likely to be important?

Feature selection consists of three steps:

- **Screening.** Removes unimportant and problematic inputs and records, or cases such as input fields with too many missing values or with too much or too little variation to be useful.
- **Ranking.** Sorts remaining inputs and assigns ranks based on importance.
- **Selecting.** Identifies the subset of features to use in subsequent models—for example, by preserving only the most important inputs and filtering or excluding all others.

In an age where many organizations are overloaded with too much data, the benefits of feature selection in simplifying and speeding the modeling process can be substantial. By focusing attention quickly on the fields that matter most, you can reduce the amount of computation required; more easily locate small but important relationships that might otherwise be overlooked; and, ultimately, obtain simpler, more accurate, and more easily explainable models. By reducing the number of fields used in the model, you may find that you can reduce scoring times as well as the amount of data collected in future iterations.

**Example.** A telephone company has a data warehouse containing information about responses to a special promotion by 5,000 of the company's customers. The data includes a large number of fields

containing customers' ages, employment, income, and telephone usage statistics. Three target fields show whether or not the customer responded to each of three offers. The company wants to use this data to help predict which customers are most likely to respond to similar offers in the future.

**Requirements.** A single target field (one with its role set to *Target*), along with multiple input fields that you want to screen or rank relative to the target. Both target and input fields can have a measurement level of *Continuous* (numeric range) or *Categorical*.

## Feature Selection Model Settings

The settings on the Model tab include standard model options along with settings that enable you to fine-tune the criteria for screening input fields.

**Model name.** You can generate the model name automatically based on the target or ID field (or model type in cases where no such field is specified) or specify a custom name.

### Screening Input Fields

Screening involves removing inputs or cases that do not add any useful information with respect to the input/target relationship. Screening options are based on attributes of the field in question without respect to predictive power relative to the selected target field. Screened fields are excluded from the computations used to rank inputs and optionally can be filtered or removed from the data used in modeling.

Fields can be screened based on the following criteria:

- **Maximum percentage of missing values.** Screens fields with too many missing values, expressed as a percentage of the total number of records. Fields with a large percentage of missing values provide little predictive information.
- **Maximum percentage of records in a single category.** Screens fields that have too many records falling into the same category relative to the total number of records. For example, if 95% of the customers in the database drive the same type of car, including this information is not useful in distinguishing one customer from the next. Any fields that exceed the specified maximum are screened. This option applies to categorical fields only.
- **Maximum number of categories as a percentage of records.** Screens fields with too many categories relative to the total number of records. If a high percentage of the categories contains only a single case, the field may be of limited use. For example, if every customer wears a different hat, this information is unlikely to be useful in modeling patterns of behavior. This option applies to categorical fields only.
- **Minimum coefficient of variation.** Screens fields with a coefficient of variance less than or equal to the specified minimum. This measure is the ratio of the input field standard deviation to the mean of the input field. If this value is near zero, there is not much variability in the values for the variable. This option applies to continuous (numeric range) fields only.
- **Minimum standard deviation.** Screens fields with standard deviation less than or equal to the specified minimum. This option applies to continuous (numeric range) fields only.

**Records with missing data.** Records or cases that have missing values for the target field, or missing values for all inputs, are automatically excluded from all computations used in the rankings.

## Feature Selection Options

The Options tab allows you to specify the default settings for selecting or excluding input fields in the model nugget. You can then add the model to a stream to select a subset of fields for use in subsequent model-building efforts. Alternatively, you can override these settings by selecting or deselecting additional fields in the model browser after generating the model. However, the default settings make it possible to apply the model nugget without further changes, which may be particularly useful for scripting purposes.

See the topic [“Feature Selection Model Results” on page 47](#) for more information.

The following options are available:

**All fields ranked.** Selects fields based on their ranking as *important*, *marginal*, or *unimportant*. You can edit the label for each ranking as well as the cutoff values used to assign records to one rank or another.

**Top number of fields.** Selects the top  $n$  fields based on importance.

**Importance greater than.** Selects all fields with importance greater than the specified value.

The target field is always preserved regardless of the selection.

Importance Ranking Options

**All categorical.** When all inputs and the target are categorical, importance can be ranked based on any of four measures:

- **Pearson chi-square.** Tests for independence of the target and the input without indicating the strength or direction of any existing relationship.
- **Likelihood-ratio chi-square.** Similar to Pearson's chi-square but also tests for target-input independence.
- **Cramer's V.** A measure of association based on Pearson's chi-square statistic. Values range from 0, which indicates no association, to 1, which indicates perfect association.
- **Lambda.** A measure of association reflecting the proportional reduction in error when the variable is used to predict the target value. A value of 1 indicates that the input field perfectly predicts the target, while a value of 0 means the input provides no useful information about the target.

**Some categorical.** When some—but not all—inputs are categorical and the target is also categorical, importance can be ranked based on either the Pearson or likelihood-ratio chi-square. (Cramer's  $V$  and lambda are not available unless all inputs are categorical.)

**Categorical versus continuous.** When ranking a categorical input against a continuous target or vice versa (one or the other is categorical but not both), the  $F$  statistic is used.

**Both continuous.** When ranking a continuous input against a continuous target, the  $t$  statistic based on the correlation coefficient is used.

## Feature Selection Model Nuggets

---

Feature Selection model nuggets display the importance of each input relative to a selected target, as ranked by the Feature Selection node. Any fields that were screened out prior to the ranking are also listed. See the topic [“Feature Selection node”](#) on page 45 for more information.

When you run a stream containing a Feature Selection model nugget, the model acts as a filter that preserves only selected inputs, as indicated by the current selection on the Model tab. For example, you could select all fields ranked as important (one of the default options) or manually select a subset of fields on the Model tab. The target field is also preserved regardless of the selection. All other fields are excluded.

Filtering is based on the field name only; for example, if you select *age* and *income*, any field that matches either of these names will be preserved. The model does not update field rankings based on new data; it simply filters fields based on the selected names. For this reason, care should be used in applying the model to new or updated data. When in doubt, regenerating the model is recommended.

## Feature Selection Model Results

The Model tab for a Feature Selection model nugget displays the rank and importance of all inputs in the upper pane and enables you to select fields for filtering by using the check boxes in the column on the left. When you run the stream, only the selected fields are preserved; the other fields are discarded. The default selections are based on the options specified in the model-building node, but you can select or deselect additional fields as needed.

The lower pane lists inputs that have been excluded from the rankings based on the percentage of missing values or on other criteria specified in the modeling node. As with the ranked fields, you can choose to include or discard these fields by using the check boxes in the column on the left. See the topic [“Feature Selection Model Settings”](#) on page 46 for more information.

- To sort the list by rank, field name, importance, or any of the other displayed columns, click on the column header. Or, to use the toolbar, select the desired item from the Sort By list, and use the up and down arrows to change the direction of the sort.
- You can use the toolbar to check or uncheck all fields and to access the Check Fields dialog box, which enables you to select fields by rank or importance. You can also press the Shift and Ctrl keys while clicking on fields to extend the selection and use the spacebar to toggle on or off a group of selected fields. See the topic [“Selecting Fields by Importance”](#) on page 48 for more information.
- The threshold values for ranking inputs as important, marginal, or unimportant are displayed in the legend below the table. These values are specified in the modeling node. See the topic [“Feature Selection Options”](#) on page 46 for more information.

## Selecting Fields by Importance

When scoring data using a Feature Selection model nugget, all fields selected from the list of ranked or screened fields--as indicated by the check boxes in the column on the left--will be preserved. Other fields will be discarded. To change the selection, you can use the toolbar to access the Check Fields dialog box, which enables you to select fields by rank or importance.

**All fields marked.** Selects all fields marked as important, marginal, or unimportant.

**Top number of fields.** Allows you to select the top  $n$  fields based on importance.

**Importance greater than.** Selects all fields with importance greater than the specified threshold.

## Generating a Filter from a Feature Selection Model

Based on the results of a Feature Selection model, you can use the Generate Filter from Feature dialog box to generate one or more Filter nodes that include or exclude subsets of fields based on importance relative to the specified target. While the model nugget can also be used as a filter, this gives you the flexibility to experiment with different subsets of fields without copying or modifying the model. The target field is always preserved by the filter regardless of whether include or exclude is selected.

**Include/Exclude.** You can choose to include or exclude fields—for example, to include the top 10 fields or exclude all fields marked as unimportant.

**Selected fields.** Includes or excludes all fields currently selected in the table.

**All fields marked.** Selects all fields marked as important, marginal, or unimportant.

**Top number of fields.** Allows you to select the top  $n$  fields based on importance.

**Importance greater than.** Selects all fields with importance greater than the specified threshold.

## Anomaly Detection Node

Anomaly detection models are used to identify outliers, or unusual cases, in the data. Unlike other modeling methods that store rules about unusual cases, anomaly detection models store information on what normal behavior looks like. This makes it possible to identify outliers even if they do not conform to any known pattern, and it can be particularly useful in applications, such as fraud detection, where new patterns may constantly be emerging. Anomaly detection is an unsupervised method, which means that it does not require a training dataset containing known cases of fraud to use as a starting point.

While traditional methods of identifying outliers generally look at one or two variables at a time, anomaly detection can examine large numbers of fields to identify clusters or peer groups into which similar records fall. Each record can then be compared to others in its peer group to identify possible anomalies. The further away a case is from the normal center, the more likely it is to be unusual. For example, the algorithm might lump records into three distinct clusters and flag those that fall far from the center of any one cluster.

Each record is assigned an anomaly index, which is the ratio of the group deviation index to its average over the cluster that the case belongs to. The larger the value of this index, the more deviation the case has than the average. Under the usual circumstance, cases with anomaly index values less than 1 or even

1.5 would not be considered as anomalies, because the deviation is just about the same or a bit more than the average. However, cases with an index value greater than 2 could be good anomaly candidates because the deviation is at least twice the average.

Anomaly detection is an exploratory method designed for quick detection of unusual cases or records that should be candidates for further analysis. These should be regarded as *suspected* anomalies, which, on closer examination, may or may not turn out to be real. You may find that a record is perfectly valid but choose to screen it from the data for purposes of model building. Alternatively, if the algorithm repeatedly turns up false anomalies, this may point to an error or artifact in the data collection process.

Note that anomaly detection identifies unusual records or cases through cluster analysis based on the set of fields selected in the model without regard for any specific target (dependent) field and regardless of whether those fields are relevant to the pattern you are trying to predict. For this reason, you may want to use anomaly detection in combination with feature selection or another technique for screening and ranking fields. For example, you can use feature selection to identify the most important fields relative to a specific target and then use anomaly detection to locate the records that are the most unusual with respect to those fields. (An alternative approach would be to build a decision tree model and then examine any misclassified records as potential anomalies. However, this method would be more difficult to replicate or automate on a large scale.)

**Example.** In screening agricultural development grants for possible cases of fraud, anomaly detection can be used to discover deviations from the norm, highlighting those records that are abnormal and worthy of further investigation. You are particularly interested in grant applications that seem to claim too much (or too little) money for the type and size of farm.

**Requirements.** One or more input fields. Note that only fields with a role set to **Input** using a source or Type node can be used as inputs. Target fields (role set to **Target** or **Both**) are ignored.

**Strengths.** By flagging cases that do *not* conform to a known set of rules rather than those that do, Anomaly Detection models can identify unusual cases even when they don't follow previously known patterns. When used in combination with feature selection, anomaly detection makes it possible to screen large amounts of data to identify the records of greatest interest relatively quickly.

## Anomaly Detection Model Options

**Model name.** You can generate the model name automatically based on the target or ID field (or model type in cases where no such field is specified) or specify a custom name.

**Determine cutoff value for anomaly based on.** Specifies the method used to determine the cutoff value for flagging anomalies. The following options are available:

- **Minimum anomaly index level.** Specifies the minimum cutoff value for flagging anomalies. Records that meet or exceed this threshold are flagged.
- **Percentage of most anomalous records in the training data.** Automatically sets the threshold at a level that flags the specified percentage of records in the training data. The resulting cutoff is included as a parameter in the model. Note that this option determines how the cutoff value is set, *not* the actual percentage of records to be flagged during scoring. Actual scoring results may vary depending on the data.
- **Number of most anomalous records in the training data.** Automatically sets the threshold at a level that flags the specified number of records in the training data. The resulting threshold is included as a parameter in the model. Note that this option determines how the cutoff value is set, *not* the specific number of records to be flagged during scoring. Actual scoring results may vary depending on the data.

*Note:* Regardless of how the cutoff value is determined, it does not affect the underlying anomaly index value reported for each record. It simply specifies the threshold for flagging records as anomalous when estimating or scoring the model. If you later want to examine a larger or smaller number of records, you can use a Select node to identify a subset of records based on the anomaly index value (`$0 - AnomalyIndex > X`).

**Number of anomaly fields to report.** Specifies the number of fields to report as an indication of why a particular record is flagged as an anomaly. The most anomalous fields are reported, defined as those that show the greatest deviation from the field norm for the cluster to which the record is assigned.

## Anomaly Detection Expert Options

To specify options for missing values and other settings, set the mode to **Expert** on the Expert tab.

**Adjustment coefficient.** Value used to balance the relative weight given to continuous (numeric range) and categorical fields in calculating the distance. Larger values increase the influence of continuous fields. This must be a nonzero value.

**Automatically calculate number of peer groups.** Anomaly detection can be used to rapidly analyze a large number of possible solutions to choose the optimal number of peer groups for the training data. You can broaden or narrow the range by setting the minimum and maximum number of peer groups. Larger values will enable the system to explore a broader range of possible solutions; however, the cost is increased processing time.

**Specify number of peer groups.** If you know how many clusters to include in your model, select this option and enter the number of peer groups. Selecting this option will generally result in improved performance.

**Noise level and ratio.** These settings determine how outliers are treated during two-stage clustering. In the first stage, a cluster feature (CF) tree is used to condense the data from a very large number of individual records to a manageable number of clusters. The tree is built based on similarity measures, and when a node of the tree gets too many records in it, it splits into child nodes. In the second stage, hierarchical clustering commences on the terminal nodes of the CF tree. Noise handling is turned on in the first data pass, and it is off in the second data pass. The cases in the noise cluster from the first data pass are assigned to the regular clusters in the second data pass.

- **Noise level.** Specify a value between 0 and 0.5. This setting is relevant only if the CF tree fills during the growth phase, meaning that it cannot accept any more cases in a leaf node and that no leaf node can be split.

If the CF tree fills and the noise level is set to 0, the threshold will be increased and the CF tree regrown with all cases. After final clustering, values that cannot be assigned to a cluster are labeled outliers. The outlier cluster is given an identification number of -1. The outlier cluster is not included in the count of the number of clusters; that is, if you specify  $n$  clusters and noise handling, the algorithm will output  $n$  clusters and one noise cluster. In practical terms, increasing this value gives the algorithm more latitude to fit unusual records into the tree rather than assign them to a separate outlier cluster.

If the CF tree fills and the noise level is greater than 0, the CF tree will be regrown after placing any data in sparse leaves into their own noise leaf. A leaf is considered sparse if the ratio of the number of cases in the sparse leaf to the number of cases in the largest leaf is less than the noise level. After the tree is grown, the outliers will be placed in the CF tree if possible. If not, the outliers are discarded for the second phase of clustering.

- **Noise ratio.** Specifies the portion of memory allocated for the component that should be used for noise buffering. This value ranges between 0.0 and 0.5. If inserting a specific case into a leaf of the tree would yield tightness less than the threshold, the leaf is not split. If the tightness exceeds the threshold, the leaf is split, adding another small cluster to the CF tree. In practical terms, increasing this setting may cause the algorithm to gravitate more quickly toward a simpler tree.

**Impute missing values.** For continuous fields, substitutes the field mean in place of any missing values. For categorical fields, missing categories are combined and treated as a valid category. If this option is deselected, any records with missing values are excluded from the analysis.

## Anomaly Detection Model Nuggets

Anomaly Detection model nuggets contain all of the information captured by the Anomaly Detection model as well as information about the training data and estimation process.

When you run a stream containing an Anomaly Detection model nugget, a number of new fields are added to the stream, as determined by the selections made on the Settings tab in the model nugget. See the topic [“Anomaly Detection Model Settings” on page 51](#) for more information. New field names are based on the model name, prefaced by \$O, as summarized in the following table.

Table 6. New field name generation	
Field name	Description
<i>\$O-Anomaly</i>	Flag field indicating whether or not the record is anomalous.
<i>\$O-AnomalyIndex</i>	The anomaly index value for the record.
<i>\$O-PeerGroup</i>	Specifies the peer group to which the record is assigned.
<i>\$O-Field-n</i>	Name of the <i>n</i> th most anomalous field in terms of deviation from the cluster norm.
<i>\$O-FieldImpact-n</i>	Variable deviation index for the field. This value measures the deviation from the field norm for the cluster to which the record is assigned.

Optionally, you can suppress scores for non-anomalous records to make the results easier to read. See the topic “[Anomaly Detection Model Settings](#)” on page 51 for more information.

## Anomaly Detection model details

The Model tab for a generated Anomaly Detection model displays information about the peer groups in the model.

Note that the peer group sizes and statistics reported are estimates based on the training data and may differ slightly from actual scoring results even if run on the same data.

The **Residual of the unreported reasons** is 1 minus the sum of the mean anomaly index values for each of the anomaly columns for the records that are identified as anomalies. This percentage can serve as an indication of how much of the anomaly is explained by the fields that are reported. This might guide you in determining how many anomaly fields to report.

## Anomaly Detection Model Summary

The Summary tab for an Anomaly Detection model nugget displays information about the fields, build settings, and estimation process. The number of peer groups is also shown, along with the cutoff value used to flag records as anomalous.

## Anomaly Detection Model Settings

Use the Settings tab to specify options for scoring the model nugget.

**Indicate anomalous records with** Specifies how anomalous records are treated in the output.

- **Flag and index** Creates a flag field that is set to *True* for all records that exceed the cutoff value included in the model. The anomaly index is also reported for each record in a separate field. See the topic “[Anomaly Detection Model Options](#)” on page 49 for more information.
- **Flag only** Creates a flag field but without reporting the anomaly index for each record.
- **Index only** Reports the anomaly index without creating a flag field.

**Number of anomaly fields to report** Specifies the number of fields to report as an indication of why a particular record is flagged as an anomaly. The most anomalous fields are reported, defined as those that show the greatest deviation from the field norm for the cluster to which the record is assigned.

**Discard records** Select this option to discard all **Non anomalous** records from the stream, making it easier to focus on potential anomalies in any downstream nodes. Alternatively, you can choose to discard all **Anomalous** records in order to limit the subsequent analysis to those records that are not flagged as potential anomalies based on the model.



**Note:** Due to slight differences in rounding, the actual number of records flagged during scoring may not be identical to those flagged while training the model even if run on the same data.

**Generate SQL for this model** When using data from a database, SQL code can be pushed back to the database for execution, providing superior performance for many operations.

Select one of the following options to specify how SQL generation is performed.

- **Default: Score using Server Scoring Adapter (if installed) otherwise in process** If connected to a database with a scoring adapter installed, generates SQL using the scoring adapter and associated user defined functions (UDF) and scores your model within the database. When no scoring adapter is available, this option fetches your data back from the database and scores it in SPSS Modeler.
- **Score outside of the Database** If selected, this option fetches your data back from the database and scores it in SPSS Modeler.



---

## Chapter 5. Automated Modeling Nodes

The automated modeling nodes estimate and compare a number of different modeling methods, enabling you to try out a variety of approaches in a single modeling run. You can select the modeling algorithms to use, and the specific options for each, including combinations that would otherwise be mutually-exclusive. For example, rather than choose between the quick, dynamic, or prune methods for a Neural Net, you can try them all. The node explores every possible combination of options, ranks each candidate model based on the measure you specify, and saves the best for use in scoring or further analysis.

You can choose from three automated modeling nodes, depending on the needs of your analysis:



The Auto Classifier node creates and compares a number of different models for binary outcomes (yes or no, churn or do not churn, and so on), allowing you to choose the best approach for a given analysis. A number of modeling algorithms are supported, making it possible to select the methods you want to use, the specific options for each, and the criteria for comparing the results. The node generates a set of models based on the specified options and ranks the best candidates according to the criteria you specify.



The Auto Numeric node estimates and compares models for continuous numeric range outcomes using a number of different methods. The node works in the same manner as the Auto Classifier node, allowing you to choose the algorithms to use and to experiment with multiple combinations of options in a single modeling pass. Supported algorithms include neural networks, C&R Tree, CHAID, linear regression, generalized linear regression, and support vector machines (SVM). Models can be compared based on correlation, relative error, or number of variables used.



The Auto Cluster node estimates and compares clustering models, which identify groups of records that have similar characteristics. The node works in the same manner as other automated modeling nodes, allowing you to experiment with multiple combinations of options in a single modeling pass. Models can be compared using basic measures with which to attempt to filter and rank the usefulness of the cluster models, and provide a measure based on the importance of particular fields.

The best models are saved in a single composite model nugget, enabling you to browse and compare them, and to choose which models to use in scoring.

- For binary, nominal, and numeric targets only you can select multiple scoring models and combine the scores in a single model ensemble. By combining predictions from multiple models, limitations in individual models may be avoided, often resulting in a higher overall accuracy than can be gained from any one of the models.
- Optionally, you can choose to drill down into the results and generate modeling nodes or model nuggets for any of the individual models you want to use or explore further.

### Models and Execution Time

Depending on the dataset and the number of models, automated modeling nodes may take hours or even longer to execute. When selecting options, pay attention to the number of models being produced. When practical, you may want to schedule modeling runs during nights or weekends when system resources are less likely to be in demand.

- If necessary, a Partition or Sample node can be used to reduce the number of records included in the initial training pass. Once you have narrowed the choices to a few candidate models, the full dataset can be restored.

- To reduce the number of input fields, use Feature Selection. See the topic [“Feature Selection node”](#) on page 45 for more information. Alternatively, you can use your initial modeling runs to identify fields and options that are worth exploring further. For example, if your best-performing models all seem to use the same three fields, this is a strong indication that those fields are worth keeping.
- Optionally, you can limit the amount of time spent estimating any one model and specify the evaluation measures used to screen and rank models.

## Automated Modeling Node Algorithm Settings

---

For each model type, you can use the default settings, or you can choose options for each model type. The specific options are similar to those available in the separate modeling nodes, with the difference that rather than choosing one setting or another, you can choose as many as you want to apply in most cases. For example, if comparing Neural Net models, you can choose several different training methods, and try each method with and without a random seed. All possible combinations of the selected options will be used, making it very easy to generate many different models in a single pass. Use care, however, as choosing multiple settings can cause the number of models to multiply very quickly.

To choose options for each model type

1. On the automated modeling node, select the **Expert** tab.
2. Click in the **Model parameters** column for the model type.
3. From the drop-down menu, choose **Specify**.
4. On the **Algorithm settings** dialog, select options from the **Options** column.

*Note:* Further options are available on the Expert tab of the **Algorithm settings** dialog.

## Automated Modeling Node Stopping Rules

---

Stopping rules specified for automated modeling nodes relate to the overall node execution, not the stopping of individual models built by the node.

**Restrict overall execution time.** (Neural Network, K-Means, Kohonen, TwoStep, SVM, KNN, Bayes Net and C&R Tree models only) Stops execution after a specified number of hours. All models generated up to that point will be included in the model nugget, but no further models will be produced.

**Stop as soon as valid models are produced.** Stops execution when a model passes all criteria specified on the Discard tab (for the Auto Classifier or Auto Cluster node) or the Model tab (for the Auto Numeric node). See the topic [“Auto Classifier Node Discard Options”](#) on page 59 for more information. See the topic [“Auto Cluster Node Discard Options”](#) on page 67 for more information.

## Auto Classifier node

---

The Auto Classifier node estimates and compares models for either nominal (set) or binary (yes/no) targets, using a number of different methods, enabling you to try out a variety of approaches in a single modeling run. You can select the algorithms to use, and experiment with multiple combinations of options. For example, rather than choose between Radial Basis Function, polynomial, sigmoid, or linear methods for an SVM, you can try them all. The node explores every possible combination of options, ranks each candidate model based on the measure you specify, and saves the best models for use in scoring or further analysis. For more information, see [Chapter 5, “Automated Modeling Nodes,”](#) on page 53.

### Example

A retail company has historical data tracking the offers made to specific customers in past campaigns. The company now wants to achieve more profitable results by matching the right offer to each customer.

### Requirements

A target field with a measurement level of either *Nominal* or *Flag* (with the role set to **Target**), and at least one input field (with the role set to **Input**). For a flag field, the *True* value defined for the target is assumed to represent a hit when calculating profits, lift, and related statistics. Input fields can have a measurement level of *Continuous* or *Categorical*, with the limitation that some inputs may not be

appropriate for some model types. For example, ordinal fields used as inputs in C&R Tree, CHAID, and QUEST models must have numeric storage (not string), and will be ignored by these models if specified otherwise. Similarly, continuous input fields can be binned in some cases. The requirements are the same as when using the individual modeling nodes; for example a Bayes Net model works the same whether generated from the Bayes Net node or the Auto Classifier node.

### Frequency and weight fields

Frequency and weight are used to give extra importance to some records over others because, for example, the user knows that the build dataset under-represents a section of the parent population (Weight) or because one record represents a number of identical cases (Frequency). If specified, a frequency field can be used by C&R Tree, CHAID, QUEST, Decision List, and Bayes Net models. A weight field can be used by C&RT, CHAID, and C5.0 models. Other model types will ignore these fields and build the models anyway. Frequency and weight fields are used only for model building, and are not considered when evaluating or scoring models. For more information, see [“Using Frequency and Weight Fields”](#) on page 27.

### Prefixes

If you attach a table node to the nugget for the Auto Classifier Node, there are several new variables in the table with names that begin with a \$ prefix.

The names of the fields that are generated during scoring are based on the target field, but with a standard prefix. Different model types use different sets of prefixes.

For example, the prefixes \$G, \$R, \$C are used as the prefix for predictions that are generated by the Generalized Linear model, CHAID model, and C5.0 model, respectively. \$X is typically generated by using an ensemble, and \$XR, \$XS, and \$XF are used as prefixes in cases where the target field is a Continuous, Categorical, or Flag field, respectively.

\$.C prefixes are used for prediction confidence of a Categorical, or Flag target; for example, \$XFC is used as a prefix for ensemble Flag prediction confidence. \$RC and \$CC are the prefixes for a single prediction of confidence for a CHAID model and C5.0 model respectively.

### Supported Model Types

Supported model types include Neural Net, C&R Tree, QUEST, CHAID, C5.0, Logistic Regression, Decision List, Bayes Net, Discriminant, Nearest Neighbor, SVM, XGBoost Tree, and XGBoost-AS. See the topic [“Auto Classifier Node Expert Options”](#) on page 56 for more information.

## Auto Classifier Node Model Options

The Model tab of the Auto Classifier node enables you to specify the number of models to be created, along with the criteria used to compare models.

**Model name.** You can generate the model name automatically based on the target or ID field (or model type in cases where no such field is specified) or specify a custom name.

**Use partitioned data.** If a partition field is defined, this option ensures that data from only the training partition is used to build the model.

**Create split models.** Builds a separate model for each possible value of input fields that are specified as split fields. See [“Building Split Models”](#) on page 24 for more information.

**Rank models by.** Specifies the criteria used to compare and rank models. Options include overall accuracy, area under the ROC curve, profit, lift, and number of fields. Note that all of these measures will be available in the summary report regardless of which is selected here.

*Note:* For a nominal (set) target, ranking is restricted to either **Overall Accuracy** or **Number of Fields**.

When calculating profits, lift, and related statistics, the *True* value defined for the target field is assumed to represent a hit.

- **Overall accuracy** The percentage of records that is correctly predicted by the model relative to the total number of records.
- **Area under the ROC curve** The ROC curve provides an index for the performance of a model. The further the curve lies above the reference line, the more accurate the test.

- **Profit (Cumulative)** The sum of profits across cumulative percentiles (sorted in terms of confidence for the prediction), as computed based on the specified cost, revenue, and weight criteria. Typically, the profit starts near 0 for the top percentile, increases steadily, and then decreases. For a good model, profits will show a well-defined peak, which is reported along with the percentile where it occurs. For a model that provides no information, the profit curve will be relatively straight and may be increasing, decreasing, or level, depending on the cost/revenue structure that applies.
- **Lift (Cumulative)** The ratio of hits in cumulative quantiles relative to the overall sample (where quantiles are sorted in terms of confidence for the prediction). For example, a lift value of 3 for the top quantile indicates a hit rate three times as high as for the sample overall. For a good model, lift should start well above 1.0 for the top quantiles and then drop off sharply toward 1.0 for the lower quantiles. For a model that provides no information, the lift will hover around 1.0.
- **Number of fields** Ranks models based on the number of input fields used.

**Rank models using.** If a partition is in use, you can specify whether ranks are based on the training dataset or the testing set. With large datasets, use of a partition for preliminary screening of models may greatly improve performance.

**Number of models to use.** Specifies the maximum number of models to be listed in the model nugget produced by the node. The top-ranking models are listed according to the specified ranking criterion. Note that increasing this limit may slow performance. The maximum allowable value is 100.

**Calculate predictor importance.** For models that produce an appropriate measure of importance, you can display a chart that indicates the relative importance of each predictor in estimating the model. Typically you will want to focus your modeling efforts on the predictors that matter most, and consider dropping or ignoring those that matter least. Note that predictor importance may extend the time needed to calculate some models, and is not recommended if you simply want a broad comparison across many different models. It is more useful once you have narrowed your analysis to a handful of models that you want to explore in greater detail. See the topic [“Predictor Importance”](#) on page 37 for more information.

**Profit Criteria.** *Note.* Only for flag targets. Profit equals the revenue for each record minus the cost for the record. Profits for a quantile are simply the sum of profits for all records in the quantile. Profits are assumed to apply only to hits, but costs apply to all records.

- **Costs.** Specify the cost associated with each record. You can select **Fixed** or **Variable** costs. For fixed costs, specify the cost value. For variable costs, click the Field Chooser button to select a field as the cost field. (**Costs** is not available for ROC charts.)
- **Revenue.** Specify the revenue associated with each record that represents a hit. You can select **Fixed** or **Variable** costs. For fixed revenue, specify the revenue value. For variable revenue, click the Field Chooser button to select a field as the revenue field. (**Revenue** is not available for ROC charts.)
- **Weight.** If the records in your data represent more than one unit, you can use frequency weights to adjust the results. Specify the weight associated with each record, using **Fixed** or **Variable** weights. For fixed weights, specify the weight value (the number of units per record). For variable weights, click the Field Chooser button to select a field as the weight field. (**Weight** is not available for ROC charts.)

**Lift Criteria.** *Note.* Only for flag targets. Specifies the percentile to use for lift calculations. Note that you can also change this value when comparing the results. See the topic [“Automated Model Nuggets”](#) on page 67 for more information.

## Auto Classifier Node Expert Options

The Expert tab of the Auto Classifier node enables you to apply a partition (if available), select the algorithms to use, and specify stopping rules.

**Select models.** By default, all models are selected to be built; however, if you have Analytic Server, you can choose to restrict the models to those that can run on Analytic Server and preset them so that they either build split models or are ready to process very large data sets.

**Note:** Local building of Analytic Server models within the Auto Classifier node is not supported.

**Models used.** Use the check boxes in the column on the left to select the model types (algorithms) to include in the comparison. The more types you select, the more models will be created and the longer the processing time will be.

**Model type.** Lists the available algorithms (see below).

**Model parameters.** For each model type, you can use the default settings or select **Specify** to choose options for each model type. The specific options are similar to those available in the separate modeling nodes, with the difference that multiple options or combinations can be selected. For example, if comparing Neural Net models, rather than choosing one of the six training methods, you can choose all of them to train six models in a single pass.

**Number of models.** Lists the number of models produced for each algorithm based on current settings. When combining options, the number of models can quickly add up, so paying close attention to this number is strongly recommended, particularly when using large datasets.

**Restrict maximum time spent building a single model.** (K-Means, Kohonen, TwoStep, SVM, KNN, Bayes Net and Decision List models only) Sets a maximum time limit for any one model. For example, if a particular model requires an unexpectedly long time to train because of some complex interaction, you probably don't want it to hold up your entire modeling run.

**Note:** If the target is a nominal (set) field, the Decision List option is unavailable.

### Supported Algorithms



The Support Vector Machine (SVM) node enables you to classify data into one of two groups without overfitting. SVM works well with wide data sets, such as those with a very large number of input fields.



The *k*-Nearest Neighbor (KNN) node associates a new case with the category or value of the *k* objects nearest to it in the predictor space, where *k* is an integer. Similar cases are near each other and dissimilar cases are distant from each other.



Discriminant analysis makes more stringent assumptions than logistic regression but can be a valuable alternative or supplement to a logistic regression analysis when those assumptions are met.



The Bayesian Network node enables you to build a probability model by combining observed and recorded evidence with real-world knowledge to establish the likelihood of occurrences. The node focuses on Tree Augmented Naïve Bayes (TAN) and Markov Blanket networks that are primarily used for classification.



The Decision List node identifies subgroups, or segments, that show a higher or lower likelihood of a given binary outcome relative to the overall population. For example, you might look for customers who are unlikely to churn or are most likely to respond favorably to a campaign. You can incorporate your business knowledge into the model by adding your own custom segments and previewing alternative models side by side to compare the results. Decision List models consist of a list of rules in which each rule has a condition and an outcome. Rules are applied in order, and the first rule that matches determines the outcome.



Logistic regression is a statistical technique for classifying records based on values of input fields. It is analogous to linear regression but takes a categorical target field instead of a numeric range.



The CHAID node generates decision trees using chi-square statistics to identify optimal splits. Unlike the C&R Tree and QUEST nodes, CHAID can generate nonbinary trees, meaning that some splits have more than two branches. Target and input fields can be numeric range (continuous) or categorical. Exhaustive CHAID is a modification of CHAID that does a more thorough job of examining all possible splits but takes longer to compute.



The QUEST node provides a binary classification method for building decision trees, designed to reduce the processing time required for large C&R Tree analyses while also reducing the tendency found in classification tree methods to favor inputs that allow more splits. Input fields can be numeric ranges (continuous), but the target field must be categorical. All splits are binary.



The Classification and Regression (C&R) Tree node generates a decision tree that allows you to predict or classify future observations. The method uses recursive partitioning to split the training records into segments by minimizing the impurity at each step, where a node in the tree is considered “pure” if 100% of cases in the node fall into a specific category of the target field. Target and input fields can be numeric ranges or categorical (nominal, ordinal, or flags); all splits are binary (only two subgroups).



The C5.0 node builds either a decision tree or a rule set. The model works by splitting the sample based on the field that provides the maximum information gain at each level. The target field must be categorical. Multiple splits into more than two subgroups are allowed.



The Neural Net node uses a simplified model of the way the human brain processes information. It works by simulating a large number of interconnected simple processing units that resemble abstract versions of neurons. Neural networks are powerful general function estimators and require minimal statistical or mathematical knowledge to train or apply.



Linear regression models predict a continuous target based on linear relationships between the target and one or more predictors.



The Linear Support Vector Machine (LSVM) node enables you to classify data into one of two groups without overfitting. LSVM is linear and works well with wide data sets, such as those with a very large number of records.



The Random Trees node is similar to the existing C&RT node; however, the Random Trees node is designed to process big data to create a single tree and displays the resulting model in the output viewer that was added in SPSS Modeler version 17. The Random Trees tree node generates a decision tree that you use to predict or classify future observations. The method uses recursive partitioning to split the training records into segments by minimizing the impurity at each step, where a node in the tree is considered *pure* if 100% of cases in the node fall into a specific category of the target field. Target and input fields can be numeric ranges or categorical (nominal, ordinal, or flags); all splits are binary (only two subgroups).





The Tree-AS node is similar to the existing CHAID node; however, the Tree-AS node is designed to process big data to create a single tree and displays the resulting model in the output viewer that was added in SPSS Modeler version 17. The node generates a decision tree by using chi-square statistics (CHAID) to identify optimal splits. This use of CHAID can generate nonbinary trees, meaning that some splits have more than two branches. Target and input fields can be numeric range (continuous) or categorical. Exhaustive CHAID is a modification of CHAID that does a more thorough job of examining all possible splits but takes longer to compute.



XGBoost Tree<sup>®</sup> is an advanced implementation of a gradient boosting algorithm with a tree model as the base model. Boosting algorithms iteratively learn weak classifiers and then add them to a final strong classifier. XGBoost Tree is very flexible and provides many parameters that can be overwhelming to most users, so the XGBoost Tree node in SPSS Modeler exposes the core features and commonly used parameters. The node is implemented in Python.



XGBoost<sup>®</sup> is an advanced implementation of a gradient boosting algorithm. Boosting algorithms iteratively learn weak classifiers and then add them to a final strong classifier. XGBoost is very flexible and provides many parameters that can be overwhelming to most users, so the XGBoost-AS node in SPSS Modeler exposes the core features and commonly used parameters. The XGBoost-AS node is implemented in Spark.

**Note:** If you select Tree-AS to run on Analytic Server, it will fail to build a model when there is a Partition node upstream. In this case, to make Auto Classifier work with other modeling nodes on Analytic Server, deselect the Tree-AS model type.

## Misclassification Costs

In some contexts, certain kinds of errors are more costly than others. For example, it may be more costly to classify a high-risk credit applicant as low risk (one kind of error) than it is to classify a low-risk applicant as high risk (a different kind of error). Misclassification costs allow you to specify the relative importance of different kinds of prediction errors.

Misclassification costs are basically weights applied to specific outcomes. These weights are factored into the model and may actually change the prediction (as a way of protecting against costly mistakes).

With the exception of C5.0 models, misclassification costs are not applied when scoring a model and are not taken into account when ranking or comparing models using an Auto Classifier node, evaluation chart, or Analysis node. A model that includes costs may not produce fewer errors than one that doesn't and may not rank any higher in terms of overall accuracy, but it is likely to perform better in practical terms because it has a built-in bias in favor of *less expensive* errors.

The cost matrix shows the cost for each possible combination of predicted category and actual category. By default, all misclassification costs are set to 1.0. To enter custom cost values, select **Use misclassification costs** and enter your custom values into the cost matrix.

To change a misclassification cost, select the cell corresponding to the desired combination of predicted and actual values, delete the existing contents of the cell, and enter the desired cost for the cell. Costs are not automatically symmetrical. For example, if you set the cost of misclassifying A as B to be 2.0, the cost of misclassifying B as A will still have the default value of 1.0 unless you explicitly change it as well.

## Auto Classifier Node Discard Options

The Discard tab of the Auto Classifier node enables you to automatically discard models that do not meet certain criteria. These models will not be listed in the summary report.

You can specify a minimum threshold for overall accuracy and a maximum threshold for the number of variables used in the model. In addition, for flag targets, you can specify a minimum threshold for lift,

profit, and area under the curve; lift and profit are determined as specified on the Model tab. See the topic [“Auto Classifier Node Model Options”](#) on page 55 for more information.

Optionally, you can configure the node to stop execution the first time a model is generated that meets all specified criteria. See the topic [“Automated Modeling Node Stopping Rules”](#) on page 54 for more information.

## Auto Classifier Node Settings Options

The Settings tab of the Auto Classifier node enables you to pre-configure the score-time options that are available on the nugget.

**Filter out fields generated by ensembled models.** Removes from the output all of the additional fields generated by the individual models that feed into the Ensemble node. Select this check box if you are interested only in the combined score from all of the input models. Ensure that this option is deselected if, for example, you want to use an Analysis node or Evaluation node to compare the accuracy of the combined score with that of each of the individual input models.

## Auto Numeric node

---

The Auto Numeric node estimates and compares models for continuous numeric range outcomes using a number of different methods, enabling you to try out a variety of approaches in a single modeling run. You can select the algorithms to use, and experiment with multiple combinations of options. For example, you could predict housing values using neural net, linear regression, C&RT, and CHAID models to see which performs best, and you could try out different combinations of stepwise, forward, and backward regression methods. The node explores every possible combination of options, ranks each candidate model based on the measure you specify, and saves the best for use in scoring or further analysis. See the topic [Chapter 5, “Automated Modeling Nodes,”](#) on page 53 for more information.

### Example

A municipality wants to more accurately estimate real estate taxes and to adjust values for specific properties as needed without having to inspect every property. Using the Auto Numeric node, the analyst can generate and compare a number of models that predict property values based on building type, neighborhood, size, and other known factors.

### Requirements

A single target field (with the role set to **Target**), and at least one input field (with the role set to **Input**). The target must be a continuous (numeric range) field, such as *age* or *income*. Input fields can be continuous or categorical, with the limitation that some inputs may not be appropriate for some model types. For example, C&R Tree models can use categorical string fields as inputs, while linear regression models cannot use these fields and will ignore them if specified. The requirements are the same as when using the individual modeling nodes. For example, a CHAID model works the same whether generated from the CHAID node or the Auto Numeric node.

### Frequency and weight fields

Frequency and weight are used to give extra importance to some records over others because, for example, the user knows that the build dataset under-represents a section of the parent population (Weight) or because one record represents a number of identical cases (Frequency). If specified, a frequency field can be used by C&R Tree and CHAID algorithms. A weight field can be used by C&RT, CHAID, Regression, and GenLin algorithms. Other model types will ignore these fields and build the models anyway. Frequency and weight fields are used only for model building and are not considered when evaluating or scoring models. See the topic [“Using Frequency and Weight Fields”](#) on page 27 for more information.

### Prefixes

If you attach a table node to the nugget for the Auto Numeric Node, there are several new variables in the table with names that begin with a \$ prefix.

The names of the fields that are generated during scoring are based on the target field, but with a standard prefix. Different model types use different sets of prefixes.

For example, the prefixes \$G, \$R, \$C are used as the prefix for predictions that are generated by the Generalized Linear model, CHAID model, and C5.0 model, respectively. \$X is typically generated by



using an ensemble, and \$XR, \$XS, and \$XF are used as prefixes in cases where the target field is a Continuous, Categorical, or Flag field, respectively.

\$.E prefixes are used for the prediction confidence of a Continuous target; for example, \$XRE is used as a prefix for ensemble Continuous prediction confidence. \$GE is the prefix for a single prediction of confidence for a Generalized Linear model.

### Supported model types

Supported model types include Neural Net, C&R Tree, CHAID, Regression, GenLin, Nearest Neighbor, SVM, XGBoost Linear, GLE, and XGBoost-AS. For more information, see [“Auto Numeric Node Expert Options”](#) on page 62.

## Auto Numeric Node Model Options

The Model tab of the Auto Numeric node enables you to specify the number of models to be saved, along with the criteria used to compare models.

**Model name.** You can generate the model name automatically based on the target or ID field (or model type in cases where no such field is specified) or specify a custom name.

**Use partitioned data.** If a partition field is defined, this option ensures that data from only the training partition is used to build the model.

**Create split models.** Builds a separate model for each possible value of input fields that are specified as split fields. See [“Building Split Models”](#) on page 24 for more information.

**Rank models by.** Specifies the criteria used to compare models.

- **Correlation.** The Pearson Correlation between the observed value for each record and the value predicted by the model. The correlation is a measure of linear association between two variables, with values closer to 1 indicating a stronger relationship. (Correlation values range between  $-1$ , for a perfect negative relationship, and  $+1$  for a perfect positive relationship. A value of 0 indicates no linear relationship, while a model with a negative correlation would rank lowest of all.)
- **Number of fields.** The number of fields used as predictors in the model. Choosing models that use fewer fields may streamline data preparation and improve performance in some cases.
- **Relative error.** The relative error is the ratio of the variance of the observed values from those predicted by the model to the variance of the observed values from the mean. In practical terms, it compares how well the model performs relative to a **null** or **intercept** model that simply returns the mean value of the target field as the prediction. For a good model, this value should be less than 1, indicating that the model is more accurate than the null model. A model with a relative error greater than 1 is less accurate than the null model and is therefore not useful. For linear regression models, the relative error is equal to the square of the correlation and adds no new information. For nonlinear models, the relative error is unrelated to the correlation and provides an additional measure for assessing model performance.

**Rank models using.** If a partition is in use, you can specify whether ranks are based on the training partition or the testing partition. With large datasets, use of a partition for preliminary screening of models may greatly improve performance.

**Number of models to use.** Specifies the maximum number of models to be shown in the model nugget produced by the node. The top-ranking models are listed according to the specified ranking criterion. Increasing this limit will enable you to compare results for more models but may slow performance. The maximum allowable value is 100.

**Calculate predictor importance.** For models that produce an appropriate measure of importance, you can display a chart that indicates the relative importance of each predictor in estimating the model. Typically you will want to focus your modeling efforts on the predictors that matter most, and consider dropping or ignoring those that matter least. Note that predictor importance may extend the time needed to calculate some models, and is not recommended if you simply want a broad comparison across many different models. It is more useful once you have narrowed your analysis to a handful of models that you want to explore in greater detail. See the topic [“Predictor Importance”](#) on page 37 for more information.

**Do not keep models if.** Specifies threshold values for correlation, relative error, and number of fields used. Models that fail to meet any of these criteria will be discarded and will not be listed in the summary report.

- **Correlation less than.** The minimum correlation (in terms of absolute value) for a model to be included in the summary report.
- **Number of fields used is greater than.** The maximum number of fields to be used by any model to be included.
- **Relative error is greater than.** The maximum relative error for any model to be included.

Optionally, you can configure the node to stop execution the first time a model is generated that meets all specified criteria. See the topic [“Automated Modeling Node Stopping Rules”](#) on page 54 for more information.

## Auto Numeric Node Expert Options

The Expert tab of the Auto Numeric node enables you to select the algorithms and options to use and to specify stopping rules.

**Select models.** By default, all models are selected to be built; however, if you have Analytic Server, you can choose to restrict the models to those that can run on Analytic Server and preset them so that they either build split models or are ready to process very large data sets.

**Note:** Local building of Analytic Server models within the Auto Numeric node is not supported.

**Models used.** Use the check boxes in the column on the left to select the model types (algorithms) to include in the comparison. The more types you select, the more models will be created and the longer the processing time will be.

**Model type.** Lists the available algorithms (see below).

**Model parameters.** For each model type, you can use the default settings or select **Specify** to choose options for each model type. The specific options are similar to those available in the separate modeling nodes, with the difference that multiple options or combinations can be selected. For example, if comparing Neural Net models, rather than choosing one of the six training methods, you can choose all of them to train six models in a single pass.

**Number of models.** Lists the number of models produced for each algorithm based on current settings. When combining options, the number of models can quickly add up, so paying close attention to this number is strongly recommended, particularly when using large datasets.

**Restrict maximum time spent building a single model.** (K-Means, Kohonen, TwoStep, SVM, KNN, Bayes Net and Decision List models only) Sets a maximum time limit for any one model. For example, if a particular model requires an unexpectedly long time to train because of some complex interaction, you probably don't want it to hold up your entire modeling run.

## Supported algorithms



The Neural Net node uses a simplified model of the way the human brain processes information. It works by simulating a large number of interconnected simple processing units that resemble abstract versions of neurons. Neural networks are powerful general function estimators and require minimal statistical or mathematical knowledge to train or apply.



The Classification and Regression (C&R) Tree node generates a decision tree that allows you to predict or classify future observations. The method uses recursive partitioning to split the training records into segments by minimizing the impurity at each step, where a node in the tree is considered “pure” if 100% of cases in the node fall into a specific category of the target field. Target and input fields can be numeric ranges or categorical (nominal, ordinal, or flags); all splits are binary (only two subgroups).



The CHAID node generates decision trees using chi-square statistics to identify optimal splits. Unlike the C&R Tree and QUEST nodes, CHAID can generate nonbinary trees, meaning that some splits have more than two branches. Target and input fields can be numeric range (continuous) or categorical. Exhaustive CHAID is a modification of CHAID that does a more thorough job of examining all possible splits but takes longer to compute.



Linear regression is a common statistical technique for summarizing data and making predictions by fitting a straight line or surface that minimizes the discrepancies between predicted and actual output values.



The Generalized Linear model expands the general linear model so that the dependent variable is linearly related to the factors and covariates through a specified link function. Moreover, the model allows for the dependent variable to have a non-normal distribution. It covers the functionality of a wide number of statistical models, including linear regression, logistic regression, loglinear models for count data, and interval-censored survival models.



The  $k$ -Nearest Neighbor (KNN) node associates a new case with the category or value of the  $k$  objects nearest to it in the predictor space, where  $k$  is an integer. Similar cases are near each other and dissimilar cases are distant from each other.



The Support Vector Machine (SVM) node enables you to classify data into one of two groups without overfitting. SVM works well with wide data sets, such as those with a very large number of input fields.



Linear regression models predict a continuous target based on linear relationships between the target and one or more predictors.



The Linear Support Vector Machine (LSVM) node enables you to classify data into one of two groups without overfitting. LSVM is linear and works well with wide data sets, such as those with a very large number of records.



The Random Trees node is similar to the existing C&RT node; however, the Random Trees node is designed to process big data to create a single tree and displays the resulting model in the output viewer that was added in SPSS Modeler version 17. The Random Trees tree node generates a decision tree that you use to predict or classify future observations. The method uses recursive partitioning to split the training records into segments by minimizing the impurity at each step, where a node in the tree is considered *pure* if 100% of cases in the node fall into a specific category of the target field. Target and input fields can be numeric ranges or categorical (nominal, ordinal, or flags); all splits are binary (only two subgroups).



The Tree-AS node is similar to the existing CHAID node; however, the Tree-AS node is designed to process big data to create a single tree and displays the resulting model in the output viewer that was added in SPSS Modeler version 17. The node generates a decision tree by using chi-square statistics (CHAID) to identify optimal splits. This use of CHAID can generate nonbinary trees, meaning that some splits have more than two branches. Target and input fields can be numeric range (continuous) or categorical. Exhaustive CHAID is a modification of CHAID that does a more thorough job of examining all possible splits but takes longer to compute.



XGBoost Linear<sup>®</sup> is an advanced implementation of a gradient boosting algorithm with a linear model as the base model. Boosting algorithms iteratively learn weak classifiers and then add them to a final strong classifier. The XGBoost Linear node in SPSS Modeler is implemented in Python.



A GLE extends the linear model so that the target can have a non-normal distribution, is linearly related to the factors and covariates via a specified link function, and so that the observations can be correlated. Generalized linear mixed models cover a wide variety of models, from simple linear regression to complex multilevel models for non-normal longitudinal data.



XGBoost<sup>®</sup> is an advanced implementation of a gradient boosting algorithm. Boosting algorithms iteratively learn weak classifiers and then add them to a final strong classifier. XGBoost is very flexible and provides many parameters that can be overwhelming to most users, so the XGBoost-AS node in SPSS Modeler exposes the core features and commonly used parameters. The XGBoost-AS node is implemented in Spark.

## Auto Numeric Node Settings Options

The Settings tab of the Auto Numeric node enables you to pre-configure the score-time options that are available on the nugget.

**Filter out fields generated by ensembled models.** Removes from the output all of the additional fields generated by the individual models that feed into the Ensemble node. Select this check box if you are interested only in the combined score from all of the input models. Ensure that this option is deselected if, for example, you want to use an Analysis node or Evaluation node to compare the accuracy of the combined score with that of each of the individual input models.

**Calculate standard error.** For a continuous (numeric range) target, a standard error calculation is run by default to calculate the difference between the measured or estimated values and the true values; and to show how close those estimates matched.

## Auto Cluster node

---

The Auto Cluster node estimates and compares clustering models that identify groups of records with similar characteristics. The node works in the same manner as other automated modeling nodes, enabling you to experiment with multiple combinations of options in a single modeling pass. Models can be compared using basic measures with which to attempt to filter and rank the usefulness of the cluster models, and provide a measure based on the importance of particular fields.

Clustering models are often used to identify groups that can be used as inputs in subsequent analyses. For example you may want to target groups of customers based on demographic characteristics such as income, or based on the services they have bought in the past. This can be done without prior knowledge about the groups and their characteristics -- you may not know how many groups to look for, or what features to use in defining them. Clustering models are often referred to as unsupervised learning models, since they do not use a target field, and do not return a specific prediction that can be evaluated as true or false. The value of a clustering model is determined by its ability to capture interesting groupings in the

data and provide useful descriptions of those groupings. See [Chapter 11, “Clustering models,”](#) on page 211 for more information.

**Requirements.** One or more fields that define characteristics of interest. Cluster models do not use target fields in the same manner as other models, because they do not make specific predictions that can be assessed as true or false. Instead they are used to identify groups of cases that may be related. For example you cannot use a cluster model to predict whether a given customer will churn or respond to an offer. But you can use a cluster model to assign customers to groups based on their tendency to do those things. Weight and frequency fields are not used.

**Evaluation fields.** While no target is used, you can optionally specify one or more evaluation fields to be used in comparing models. The usefulness of a cluster model may be evaluated by measuring how well (or badly) the clusters differentiate these fields.

### Supported model types

Supported model types include TwoStep, K-Means, Kohonen, One-Class SVM, and K-Means-AS.

## Auto Cluster Node Model Options

The Model tab of the Auto Cluster node enables you to specify the number of models to be saved, along with the criteria used to compare models.

**Model name.** You can generate the model name automatically based on the target or ID field (or model type in cases where no such field is specified) or specify a custom name.

**Use partitioned data.** If a partition field is defined, this option ensures that data from only the training partition is used to build the model.

**Rank models by.** Specifies the criteria used to compare and rank models.

- **Silhouette.** An index measuring both cluster cohesion and separation. See *Silhouette Ranking Measure* below for more information.
- **Number of clusters.** The number of clusters in the model.
- **Size of smallest cluster.** The smallest cluster size.
- **Size of largest cluster.** The largest cluster size.
- **Smallest / largest cluster.** The ratio of the size of the smallest cluster to the largest cluster.
- **Importance.** The importance of the **Evaluation** field on the **Fields** tab. Note that this can only be calculated if an **Evaluation** field has been specified.

**Rank models using.** If a partition is in use, you can specify whether ranks are based on the training dataset or the testing set. With large datasets, use of a partition for preliminary screening of models may greatly improve performance.

**Number of models to keep.** Specifies the maximum number of models to be listed in the nugget produced by the node. The top-ranking models are listed according to the specified ranking criterion. Note that increasing this limit may slow performance. The maximum allowable value is 100.

### Silhouette Ranking Measure

The default ranking measure, Silhouette, has a default value of 0 because a value of less than 0 (i.e. negative) indicates that the average distance between a case and points in its assigned cluster is greater than the minimum average distance to points in another cluster. Therefore, models with a negative Silhouette can safely be discarded.

The ranking measure is actually a modified silhouette coefficient, which combines the concepts of cluster cohesion (favoring models which contain tightly cohesive clusters) and cluster separation (favoring models which contain highly separated clusters). The average Silhouette coefficient is simply the average over all cases of the following calculation for each individual case:

$$(B - A) / \max(A, B)$$

where  $A$  is the distance from the case to the centroid of the cluster which the case belongs to; and  $B$  is the minimal distance from the case to the centroid of every other cluster.

The Silhouette coefficient (and its average) range between -1 (indicating a very poor model) and 1 (indicating an excellent model). The average can be conducted on the level of total cases (which produces total Silhouette) or the level of clusters (which produces cluster Silhouette). Distances may be calculated using Euclidean distances.

## Auto Cluster Node Expert Options

The Expert tab of the Auto Cluster node enables you to apply a partition (if available), select the algorithms to use, and specify stopping rules.

**Select models.** By default, all models are selected to be built; however, if you have Analytic Server, you can choose to restrict the models to those that can run on Analytic Server and preset them so that they either build split models or are ready to process very large data sets.

**Note:** Local building of Analytic Server models within the Auto Cluster node is not supported.

**Models used.** Use the check boxes in the column on the left to select the model types (algorithms) to include in the comparison. The more types you select, the more models will be created and the longer the processing time will be.

**Model type.** Lists the available algorithms (see below).

**Model parameters.** For each model type, you can use the default settings or select **Specify** to choose options for each model type. The specific options are similar to those available in the separate modeling nodes, with the difference that multiple options or combinations can be selected. For example, if comparing Neural Net models, rather than choosing one of the six training methods, you can choose all of them to train six models in a single pass.

**Number of models.** Lists the number of models produced for each algorithm based on current settings. When combining options, the number of models can quickly add up, so paying close attention to this number is strongly recommended, particularly when using large datasets.

**Restrict maximum time spent building a single model.** (K-Means, Kohonen, TwoStep, SVM, KNN, Bayes Net and Decision List models only) Sets a maximum time limit for any one model. For example, if a particular model requires an unexpectedly long time to train because of some complex interaction, you probably don't want it to hold up your entire modeling run.

### Supported algorithms



The K-Means node clusters the data set into distinct groups (or clusters). The method defines a fixed number of clusters, iteratively assigns records to clusters, and adjusts the cluster centers until further refinement can no longer improve the model. Instead of trying to predict an outcome,  $k$ -means uses a process known as unsupervised learning to uncover patterns in the set of input fields.



The Kohonen node generates a type of neural network that can be used to cluster the data set into distinct groups. When the network is fully trained, records that are similar should be close together on the output map, while records that are different will be far apart. You can look at the number of observations captured by each unit in the model nugget to identify the strong units. This may give you a sense of the appropriate number of clusters.



The TwoStep node uses a two-step clustering method. The first step makes a single pass through the data to compress the raw input data into a manageable set of subclusters. The second step uses a hierarchical clustering method to progressively merge the subclusters into larger and larger clusters. TwoStep has the advantage of automatically estimating the optimal number of clusters for the training data. It can handle mixed field types and large data sets efficiently.





The One-Class SVM node uses an unsupervised learning algorithm. The node can be used for novelty detection. It will detect the soft boundary of a given set of samples, to then classify new points as belonging to that set or not. This One-Class SVM modeling node in SPSS Modeler is implemented in Python and requires the `scikit-learn`® Python library.



K-Means is one of the most commonly used clustering algorithms. It clusters data points into a predefined number of clusters. The K-Means-AS node in SPSS Modeler is implemented in Spark. For details about K-Means algorithms, see <https://spark.apache.org/docs/2.2.0/ml-clustering.html>. Note that the K-Means-AS node performs one-hot encoding automatically for categorical variables.

## Auto Cluster Node Discard Options

The Discard tab of the Auto Cluster node enables you to automatically discard models that do not meet certain criteria. These models will not be listed on the model nugget.

You can specify the minimum silhouette value, cluster numbers, cluster sizes, and the importance of the evaluation field used in the model. Silhouette and the number and size of clusters are determined as specified in the modeling node. See the topic [“Auto Cluster Node Model Options”](#) on page 65 for more information.

Optionally, you can configure the node to stop execution the first time a model is generated that meets all specified criteria. See the topic [“Automated Modeling Node Stopping Rules”](#) on page 54 for more information.

## Automated Model Nuggets

When an automated modeling node is executed, the node estimates candidate models for every possible combination of options, ranks each candidate model based on the measure you specify, and saves the best models in a composite automated model nugget. This model nugget actually contains a set of one or more models generated by the node, which can be individually browsed or selected for use in scoring. The model type and build time are listed for each model, along with a number of other measures as appropriate for the type of model. You can sort the table on any of these columns to quickly identify the most interesting models.

- To browse any of the individual model nuggets, double-click the nugget icon. From there you can then generate a modeling node for that model to the stream canvas, or a copy of the model nugget to the models palette.
- Thumbnail graphs give a quick visual assessment for each model type, as summarized below. You can double-click on a thumbnail to generate a full-sized graph. The full-sized plot shows up to 1000 points and will be based on a sample if the dataset contains more. (For scatterplots only, the graph is regenerated each time it is displayed, so any changes in the upstream data—such as updating of a random sample or partition if **Set Random Seed** is not selected—may be reflected each time the scatterplot is redrawn.)
- Use the toolbar to show or hide specific columns on the Model tab or to change the column used to sort the table. (You can also change the sort by clicking on the column headers.)
- Use the Delete button to permanently remove any unused models.
- To reorder columns, click on a column header and drag the column to the desired location.
- If a partition is in use, you can choose to view results for the training or testing partition as applicable.

The specific columns depend on the type of models being compared, as detailed below.

### Binary Targets

- For binary models, the thumbnail graph shows the distribution of actual values, overlaid with the predicted values, to give a quick visual indication of how many records were correctly predicted in each category.

- Ranking criteria match the options in the Auto Classifier modeling node. See the topic [“Auto Classifier Node Model Options”](#) on page 55 for more information.
- For the maximum profit, the percentile in which the maximum occurs is also reported.
- For cumulative lift, you can change the selected percentile using the toolbar.

#### Nominal Targets

- For nominal (set) models, the thumbnail graph shows the distribution of actual values, overlaid with the predicted values, to give a quick visual indication of how many records were correctly predicted in each category.
- Ranking criteria match the options in the Auto Classifier modeling node. See the topic [“Auto Classifier Node Model Options”](#) on page 55 for more information.

#### Continuous Targets

- For continuous (numeric range) models, the graph plots predicted against observed values for each model, providing a quick visual indication of the correlation between them. For a good model, points should tend to cluster along the diagonal rather than be scattered randomly across the graph.
- Ranking criteria match the options in the Auto Numeric modeling node. See the topic [“Auto Numeric Node Model Options”](#) on page 61 for more information.

#### Cluster Targets

- For cluster models, the graph plots counts against clusters for each model, providing a quick visual indication of cluster distribution.
- Ranking criteria match the options in the Auto Cluster modeling node. See the topic [“Auto Cluster Node Model Options”](#) on page 65 for more information.

#### Selecting Models for Scoring

The **Use?** column enables you to select the models to use in scoring.

- For binary, nominal, and numeric targets, you can select multiple scoring models and combine the scores in the single, ensembled model nugget. By combining predictions from multiple models, limitations in individual models may be avoided, often resulting in a higher overall accuracy than can be gained from any one of the models.
- For cluster models, only one scoring model can be selected at a time. By default, the top ranked one is selected first.

## Generating Nodes and Models

You can generate a copy of the composite automated model nugget, or the automated modeling node from which it was built. For example, this may be useful if you do not have the original stream from which the automated model nugget was built. Alternatively, you can generate a nugget or modeling node for any of the individual models listed in the automated model nugget.

#### Automated Modeling Nugget

From the Generate menu, select **Model to Palette** to add the automated model nugget to the Models palette. The generated model can be saved or used as is without rerunning the stream.

Alternatively, you can select **Generate Modeling Node** from the Generate menu to add the modeling node to the stream canvas. This node can be used to reestimate the selected models without repeating the entire modeling run.

#### Individual Modeling Nugget

1. In the **Model** menu, double-click on the individual nugget you require. A copy of that nugget opens in a new dialog.
2. From the Generate menu in the new dialog, select **Model to Palette** to add the individual modeling nugget to the Models palette.



3. Alternatively, you can select **Generate Modeling Node** from the Generate menu in the new dialog to add the individual modeling node to the stream canvas.

## Generating Evaluation Charts

For binary models only, you can generate evaluation charts that offer a visual way to assess and compare the performance of each model. Evaluation charts are not available for models generated by the Auto Numeric or Auto Cluster nodes.

1. Under the *Use?* column in the Auto Classifier automated model nugget, select the models that you want to evaluate.
2. From the Generate menu, choose **Evaluation Chart(s)**. The Evaluation Chart dialog box is displayed.
3. Select the chart type and other options as desired.

## Evaluation Graphs

On the Model tab of the automated model nugget you can drill down to display individual graphs for each of the models shown. For Auto Classifier and Auto Numeric nuggets, the Graph tab displays both a graph and predictor importance that reflect the results of all the models combined. See the topic [“Predictor Importance” on page 37](#) for more information.

For Auto Classifier a distribution graph is shown, whereas a multiplot (also known as a scatterplot) is shown for Auto Numeric.



---

# Chapter 6. Decision Trees

---

## Decision Tree Models

---

Use decision tree models to develop classification systems that predict or classify future observations based on a set of decision rules. If you have data divided into classes that interest you (for example, high-versus low-risk loans, subscribers versus nonsubscribers, voters versus nonvoters, or types of bacteria), you can use your data to build rules that you can use to classify old or new cases with maximum accuracy. For example, you might build a tree that classifies credit risk or purchase intent based on age and other factors.

This approach, sometimes known as *rule induction*, has several advantages. First, the reasoning process behind the model is clearly evident when browsing the tree. This is in contrast to other *black box* modeling techniques in which the internal logic can be difficult to work out.

Second, the process automatically includes in its rule only the attributes that really matter in making a decision. Attributes that do not contribute to the accuracy of the tree are ignored. This can yield very useful information about the data and can be used to reduce the data to relevant fields before training another learning technique, such as a neural net.

Decision tree model nuggets can be converted into a collection of if-then rules (a *rule set*), which in many cases show the information in a more comprehensible form. The decision-tree presentation is useful when you want to see how attributes in the data can *split*, or *partition*, the population into subsets relevant to the problem. The Tree-AS node output is different from the other Decision tree nodes because it includes a list of rules directly in the nugget without having to create a rule set. The rule set presentation is useful if you want to see how particular groups of items relate to a specific conclusion. For example, the following rule provides a *profile* for a group of cars that is worth buying:

```
IF tested = 'yes'  
AND mileage = 'low'  
THEN -> 'BUY'.
```

### Tree-building algorithms

Several algorithms are available for performing classification and segmentation analysis. These algorithms all perform basically the same thing, they examine all of the fields of your dataset to find the one that gives the best classification or prediction by splitting the data into subgroups. The process is applied recursively, splitting subgroups into smaller and smaller units until the tree is finished (as defined by certain stopping criteria). The target and input fields used in tree building can be continuous (numeric range) or categorical, depending on the algorithm used. If a continuous target is used, a regression tree is generated; if a categorical target is used, a classification tree is generated.



The Classification and Regression (C&R) Tree node generates a decision tree that allows you to predict or classify future observations. The method uses recursive partitioning to split the training records into segments by minimizing the impurity at each step, where a node in the tree is considered “pure” if 100% of cases in the node fall into a specific category of the target field. Target and input fields can be numeric ranges or categorical (nominal, ordinal, or flags); all splits are binary (only two subgroups).



The CHAID node generates decision trees using chi-square statistics to identify optimal splits. Unlike the C&R Tree and QUEST nodes, CHAID can generate nonbinary trees, meaning that some splits have more than two branches. Target and input fields can be numeric range (continuous) or categorical. Exhaustive CHAID is a modification of CHAID that does a more thorough job of examining all possible splits but takes longer to compute.



The QUEST node provides a binary classification method for building decision trees, designed to reduce the processing time required for large C&R Tree analyses while also reducing the tendency found in classification tree methods to favor inputs that allow more splits. Input fields can be numeric ranges (continuous), but the target field must be categorical. All splits are binary.



The C5.0 node builds either a decision tree or a rule set. The model works by splitting the sample based on the field that provides the maximum information gain at each level. The target field must be categorical. Multiple splits into more than two subgroups are allowed.



The Tree-AS node is similar to the existing CHAID node; however, the Tree-AS node is designed to process big data to create a single tree and displays the resulting model in the output viewer that was added in SPSS Modeler version 17. The node generates a decision tree by using chi-square statistics (CHAID) to identify optimal splits. This use of CHAID can generate nonbinary trees, meaning that some splits have more than two branches. Target and input fields can be numeric range (continuous) or categorical. Exhaustive CHAID is a modification of CHAID that does a more thorough job of examining all possible splits but takes longer to compute.



The Random Trees node is similar to the existing C&RT node; however, the Random Trees node is designed to process big data to create a single tree and displays the resulting model in the output viewer that was added in SPSS Modeler version 17. The Random Trees tree node generates a decision tree that you use to predict or classify future observations. The method uses recursive partitioning to split the training records into segments by minimizing the impurity at each step, where a node in the tree is considered *pure* if 100% of cases in the node fall into a specific category of the target field. Target and input fields can be numeric ranges or categorical (nominal, ordinal, or flags); all splits are binary (only two subgroups).

### General uses of tree-based analysis

The following are some general uses of tree-based analysis:

*Segmentation:* Identify persons who are likely to be members of a particular class.

*Stratification:* Assign cases into one of several categories, such as high-, medium-, and low-risk groups.

*Prediction:* Create rules and use them to predict future events. Prediction can also mean attempts to relate predictive attributes to values of a continuous variable.

*Data reduction and variable screening:* Select a useful subset of predictors from a large set of variables for use in building a formal parametric model.

*Interaction identification:* Identify relationships that pertain only to specific subgroups and specify these in a formal parametric model.

*Category merging and banding continuous variables:* Recode group predictor categories and continuous variables with minimal loss of information.

## The Interactive Tree Builder

---

You can generate a tree model automatically, where the algorithm decides the best split at each level, or you can use the interactive tree builder to take control, applying your business knowledge to refine or simplify the tree before saving the model nugget.

1. Create a stream and add one of the decision tree nodes C&R Tree, CHAID, or QUEST.

**Note:** Interactive tree building is not supported for either Tree-AS or C5.0 trees.

2. Open the node and, on the Fields tab, select target and predictor fields and specify additional model options as needed. For specific instructions, see the documentation for each tree-building node.
3. On the Objectives panel of the Build Options tab, select **Launch interactive session**.
4. Click **Run** to launch the tree builder.

The current tree is displayed, starting with the root node. You can edit and prune the tree level-by-level and access gains, risks, and related information before generating one or more models.

### Comments

- With the C&R Tree, CHAID, and QUEST nodes, any ordinal fields used in the model must have numeric storage (not string). If necessary, the Reclassify node can be used to convert them.
- Optionally, you can use a partition field to separate the data into training and test samples.
- As an alternative to using the tree builder, you can also generate a model directly from the modeling node as with other IBM SPSS Modeler models. See the topic [“Building a Tree Model Directly”](#) on page 82 for more information.

## Growing and Pruning the Tree

The Viewer tab in the tree builder enables you to view the current tree, starting with the root node.

1. To grow the tree, from the menus choose:

### **Tree > Grow Tree**

The system builds the tree by recursively splitting each branch until one or more stopping criteria are met. At each split, the best predictor is automatically selected based on the modeling method used.

2. Alternatively, select **Grow Tree One Level** to add a single level.
3. To add a branch below a specific node, select the node and select **Grow Branch**.
4. To choose the predictor used for a split, select the desired node and select **Grow Branch with Custom Split**. See the topic [“Defining Custom Splits”](#) on page 74 for more information.
5. To prune a branch, select a node and select **Remove Branch** to clear up the selected node.
6. To remove the bottom level from the tree, select **Remove One Level**.
7. For C&R Tree and QUEST trees only, select **Grow Tree and Prune** to prune based on a cost-complexity algorithm that adjusts the risk estimate based on the number of terminal nodes, typically resulting in a simpler tree. See the topic [“C&R Tree Node”](#) on page 83 for more information.

### Reading Split Rules on the Viewer Tab

When viewing split rules on the Viewer tab, square brackets mean that the adjacent value is included in the range whereas parentheses indicate that the adjacent value is excluded from the range. The expression (23,37] therefore means from 23 exclusive to 37 inclusive; that is, from just above 23 to 37. On the Model tab, the same condition would be displayed as:

Age > 23 and Age <= 37

**Interrupting tree growth.** To interrupt a tree-growing operation (if it is taking longer than expected, for example), click the Stop Execution button on the toolbar.



Figure 28. Stop Execution button

The button is enabled only during tree growth. It stops the current growing operation at its current point, leaving any nodes that have already been added, without saving changes or closing the window. The tree builder remains open, enabling you to generate a model, update directives, or export output in the appropriate format, as needed.

## Defining Custom Splits

The Define Split dialog box enables you to select the predictor and specify conditions for each split.

1. In the tree builder, select a node on the Viewer tab, and from the menus choose:

### **Tree > Grow Branch with Custom Split**

2. Select the desired predictor from the drop-down list, or click on the **Predictors** button to view details of each predictor. See the topic [“Viewing Predictor Details”](#) on page 74 for more information.
3. You can accept the default conditions for each split or select **Custom** to specify conditions for the split as appropriate.
  - For continuous (numeric range) predictors, you can use the **Edit Range Values** fields to specify the range of values that fall into each new node.
  - For categorical predictors, you can use the **Edit Set Values** or **Edit Ordinal Values** fields to specify the specific values (or range of values in case of an ordinal predictor) that map to each new node.
4. Select **Grow** to regrow the branch using the selected predictor.

The tree can generally be split using any predictor, regardless of stopping rules. The only exceptions are when the node is pure (meaning that 100% of cases fall into the same target class, thus there is nothing left to split) or the chosen predictor is constant (there is nothing to split against).

**Missing values info.** For CHAID trees only, if missing values are available for a given predictor, you have the option when defining a custom split to assign them to a specific child node. (With C&R Tree and QUEST, missing values are handled using surrogates as defined in the algorithm. See the topic [“Split Details and Surrogates”](#) on page 74 for more information.)

## Viewing Predictor Details

The Select Predictor dialog box displays statistics on available predictors (or "competitors" as they are sometimes called) that can be used for the current split.

- For CHAID and exhaustive CHAID, the chi-square statistic is listed for each categorical predictor; if a predictor is a numeric range, the *F* statistic is shown. The chi-square statistic is a measure of how independent the target field is from the splitting field. A high chi-square statistic generally relates to a lower probability, meaning that there is less chance that the two fields are independent—an indication that the split is a good one. Degrees of freedom are also included because these take into account the fact that it is easier for a three-way split to have a large statistic and small probability than it is for a two-way split.
- For C&R Tree and QUEST, the improvement for each predictor is displayed. The greater the improvement, the greater the reduction in impurity between the parent and child nodes if that predictor is used. (A pure node is one in which all cases fall into a single target category; the lower the impurity across the tree, the better the model fits the data.) In other words, a high improvement figure generally indicates a useful split for this type of tree. The impurity measure used is specified in the tree-building node.

## Split Details and Surrogates

You can select any node on the Viewer tab and select the split information button on the right side of the toolbar to view details about the split for that node. The split rule used is displayed, along with relevant statistics. For C&R Tree categorical trees, improvement and association are displayed. The association is a measure of correspondence between a surrogate and the primary split field, with the “best” surrogate generally being the one that most closely mimics the split field. For C&R Tree and QUEST, any surrogates used in place of the primary predictor are also listed.

To edit the split for the selected node, you can click the icon on the left side of the surrogates panel to open the Define Split dialog box. (As a shortcut, you can select a surrogate from the list before clicking the icon to select it as the primary split field.)

**Surrogates.** Where applicable, any surrogates for the primary split field are shown for the selected node. Surrogates are alternate fields used if the primary predictor value is missing for a given record. The

maximum number of surrogates allowed for a given split is specified in the tree-building node, but the actual number depends on the training data. In general, the more missing data, the more surrogates are likely to be used. For other decision tree models, this tab is empty.

**Note:** To be included in the model, surrogates must be identified during the training phase. If the training sample has no missing values, then no surrogates will be identified, and any records with missing values encountered during testing or scoring will automatically fall into the child node with the largest number of records. If missing values are expected during testing or scoring, be sure that values are missing from the training sample, as well. Surrogates are not available for CHAID trees.

Although surrogates are not used for CHAID trees, when defining a custom split you have the option to assign them to a specific child node. See the topic [“Defining Custom Splits” on page 74](#) for more information.

## Customizing the Tree View

The Viewer tab in the tree builder displays the current tree. By default, all branches in the tree are expanded, but you can expand and collapse branches and customize other settings as needed.

- Click the minus sign (–) at the bottom right corner of a parent node to hide all of its child nodes. Click the plus sign (+) at the bottom right corner of a parent node to display its child nodes.
- Use the View menu or the toolbar to change the orientation of the tree (top-down, left-to-right, or right-to-left).
- Click the "Display field and value labels" button on the main toolbar to show or hide field and value labels.
- Use the magnifying glass buttons to zoom the view in or out, or click the tree map button on the right side of the toolbar to view a diagram of the complete tree.
- If a partition field is in use, you can swap the tree view between training and testing partitions (**View > Partition**). When the testing sample is displayed, the tree can be viewed but not edited. (The current partition is displayed in the status bar in the lower right corner of the window.)
- Click the split information button (the "i" button on the far right of the toolbar) to view details on the current split. See the topic [“Split Details and Surrogates” on page 74](#) for more information.
- Display statistics, graphs, or both within each node (see below).

### Displaying Statistics and Graphs

**Node statistics.** For a categorical target field, the table in each node shows the number and percentage of records in each category and the percentage of the entire sample that the node represents. For a continuous (numeric range) target field, the table shows the mean, standard deviation, number of records, and predicted value of the target field.

**Node graphs.** For a categorical target field, the graph is a bar chart of percentages in each category of the target field. Preceding each row in the table is a color swatch that corresponds to the color that represents each of the target field categories in the graphs for the node. For a continuous (numeric range) target field, the graph shows a histogram of the target field for records in the node.

## Gains

The Gains tab displays statistics for all terminal nodes in the tree. Gains provide a measure of how far the mean or proportion at a given node differs from the overall mean. Generally speaking, the greater this difference, the more useful the tree is as a tool for making decisions. For example, an index or "lift" value of 148% for a node indicates that records in the node are about one-and-a-half times as likely to fall under the target category as for the dataset as a whole.

For C&R Tree and QUEST nodes where an overfit prevention set is specified, two sets of statistics are displayed:

- tree growing set - the training sample with the overfit prevention set removed
- overfit prevention set

For other C&R Tree and QUEST interactive trees, and for all CHAID interactive trees, only the tree growing set statistics are displayed.

The Gains tab enables you to:

- Display node-by-node, cumulative, or quantile statistics.
- Display gains or profits.
- Swap the view between tables and charts.
- Select the target category (categorical targets only).
- Sort the table in ascending or descending order based on the index percentage. If statistics for multiple partitions are displayed, sorts are always applied on the training sample rather than on the testing sample.

In general, selections made in the gains table will be updated in the tree view and vice versa. For example, if you select a row in the table, the corresponding node will be selected in the tree.

### Classification Gains

For classification trees (those with a categorical target variable), the gain index percentage tells you how much greater the proportion of a given target category at each node differs from the overall proportion.

#### Node-by-Node Statistics

In this view, the table displays one row for each terminal node. For example, if the overall response to your direct mail campaign was 10% but 20% of the records that fall into node X responded positively, the index percentage for the node would be 200%, indicating that respondents in this group are twice as likely to buy relative to the overall population.

For C&R Tree and QUEST nodes where an overfit prevention set is specified, two sets of statistics are displayed:

- tree growing set - the training sample with the overfit prevention set removed
- overfit prevention set

For other C&R Tree and QUEST interactive trees, and for all CHAID interactive trees, only the tree growing set statistics are displayed.

**Nodes.** The ID of the current node (as displayed on the Viewer tab).

**Node: n.** The total number of records in that node.

**Node (%).** The percentage of all records in the dataset that fall into this node.

**Gain: n.** The number of records with the selected target category that fall into this node. In other words, of all the records in the dataset that fall under the target category, how many are in this node?

**Gain (%).** The percentage of all records in the target category, across the entire dataset, that fall into this node.

**Response (%).** The percentage of records in the current node that fall under the target category. Responses in this context are sometimes referred to as "hits."

**Index (%).** The response percentage for the current node expressed as a percentage of the response percentage for the entire dataset. For example, an index value of 300% indicates that records in this node are three times as likely to fall under the target category as for the dataset as a whole.

#### Cumulative Statistics

In the cumulative view, the table displays one node per row, but statistics are cumulative, sorted in ascending or descending order by index percentage. For example if a descending sort is applied, the node with the highest index percentage is listed first, and statistics in the rows that follow are cumulative for that row and above.

The cumulative index percentage decreases row-by-row as nodes with lower and lower response percentages are added. The cumulative index for the final row is always 100% because at this point the entire dataset is included.



## Quantiles

In this view, each row in the table represents a quantile rather than a node. The quantiles are either quartiles, quintiles (fifths), deciles (tenths), vingtiles (twentieths), or percentiles (hundredths). Multiple nodes can be listed in a single quantile if more than one node is needed to make up that percentage (for example, if quartiles are displayed but the top two nodes contain fewer than 50% of all cases). The rest of the table is cumulative and can be interpreted in the same manner as the cumulative view.

## Classification Profits and ROI

For classification trees, gains statistics can also be displayed in terms of profit and ROI (return on investment). The Define Profits dialog box enables you to specify revenue and expenses for each category.

1. On the Gains tab, click the Profit button (labeled \$/\$) on the toolbar to access the dialog box.
2. Enter revenue and expense values for each category of the target field.

For example, if it costs you \$0.48 to mail an offer to each customer and the revenue from a positive response is \$9.95 for a three-month subscription, then each *no* response costs you \$0.48 and each *yes* earns you \$9.47 (calculated as  $9.95 - 0.48$ ).

In the gains table, **profit** is calculated as the sum of revenues minus expenditures for each of the records at a terminal node. **ROI** is total profit divided by total expenditure at a node.

### Comments

- Profit values affect only average profit and ROI values displayed in the gains table, as a way of viewing statistics in terms more applicable to your bottom line. They do not affect the basic tree model structure. Profits should not be confused with misclassification costs, which are specified in the tree-building node and are factored into the model as a way of protecting against costly mistakes.
- Profit specifications are not persisted between one interactive tree-building session and the next.

## Regression Gains

For regression trees, you can choose between node-by-node, cumulative node-by-node, and quantile views. Average values are shown in the table. Charts are available only for quantiles.

## Gains Charts

Charts can be displayed on the Gains tab as an alternative to tables.

1. On the Gains tab, select the Quantiles icon (third from left on the toolbar). (Charts are not available for node-by-node or cumulative statistics.)
2. Select the Charts icon.
3. Select the displayed units (percentiles, deciles, and so on) from the drop-down list as desired.
4. Select **Gains**, **Response**, or **Lift** to change the displayed measure.

### Gains Chart

The gains chart plots the values in the *Gains (%)* column from the table. Gains are defined as the proportion of hits in each increment relative to the total number of hits in the tree, using the equation:

$$(\text{hits in increment} / \text{total number of hits}) \times 100\%$$

The chart effectively illustrates how widely you need to cast the net to capture a given percentage of all the hits in the tree. The diagonal line plots the expected response for the entire sample, if the model were not used. In this case, the response rate would be constant, since one person is just as likely to respond as another. To double your yield, you would need to ask twice as many people. The curved line indicates how much you can improve your response by including only those who rank in the higher percentiles based on gain. For example, including the top 50% might net you more than 70% of the positive responses. The steeper the curve, the higher the gain.

### Lift Chart

The lift chart plots the values in the *Index (%)* column in the table. This chart compares the percentage of records in each increment that are hits with the overall percentage of hits in the training dataset, using the equation:

$$(\text{hits in increment} / \text{records in increment}) / (\text{total number of hits} / \text{total number of records})$$

Response Chart

The response chart plots the values in the *Response (%)* column of the table. The response is a percentage of records in the increment that are hits, using the equation:

$$(\text{responses in increment} / \text{records in increment}) \times 100\%$$

### Gains-Based Selection

The Gains-Based Selection dialog box enables you to automatically select terminal nodes with the best (or worst) gains based on a specified rule or threshold. You can then generate a Select node based on the selection.

1. On the Gains tab, select the node-by-node or cumulative view and select the target category on which you want to base the selection. (Selections are based on the current table display and are not available for quantiles.)
2. On the Gains tab, from the menus choose:

**Edit > Select Terminal Nodes > Gains-Based Selection**

**Select only.** You can select matching nodes *or* nonmatching nodes—for example, to select *all but* the top 100 records.

**Match by gains information.** Matches nodes based on gain statistics for the current target category, including:

- Nodes where the gain, response, or lift (index) matches a specified threshold—for example, response greater than or equal to 50%.
- The top *n* nodes based on the gain for the target category.
- The top nodes up to a specified number of records.
- The top nodes up to a specified percentage of training data.

3. Click **OK** to update the selection on the Viewer tab.
4. To create a new Select node based on the current selection on the Viewer tab, choose **Select Node** from the Generate menu. See the topic [“Generating Filter and Select Nodes” on page 81](#) for more information.

*Note:* Since you are actually selecting nodes rather than records or percentages, a perfect match with the selection criterion may not always be achieved. The system selects complete nodes *up to* the specified level. For example, if you select the top 12 cases and you have 10 in the first node and two in the second node, only the first node will be selected.

### Risks

Risks tell you the chances of misclassification at any level. The Risks tab displays a point risk estimate and (for categorical outputs) a misclassification table.

- For numeric predictions, the risk is a pooled estimate of the variance at each of the terminal nodes.
- For categorical predictions, the risk is the proportion of cases incorrectly classified, adjusted for any priors or misclassification costs.

### Saving Tree Models and Results

You can save or export the results of your interactive tree-building sessions in a number of ways, including:

- Generate a model based on the current tree (**Generate > Generate model**).

- Save the directives used to grow the current tree. The next time the tree-building node is executed, the current tree will automatically be regrown, including any custom splits that you have defined.
- Export model, gain, and risk information. See the topic [“Exporting Model, Gain, and Risk Information” on page 81](#) for more information.

From either the tree builder or a tree model nugget, you can:

- Generate a Filter or Select node based on the current tree. See [“Generating Filter and Select Nodes” on page 81](#) for more information.
- Generate a Rule Set nugget that represents the tree structure as a set of rules defining the terminal branches of the tree. See [“Generating a Rule Set from a Decision Tree” on page 81](#) for more information.
- In addition, for tree model nuggets only, you can export the model in PMML format. See [“The models palette” on page 34](#) for more information. If the model includes any custom splits, this information is not preserved in the exported PMML. (The split is preserved, but the fact that it is custom rather than chosen by the algorithm is not.)
- Generate a graph based on a selected part of the current tree. Note that this only works for a nugget when it is attached to other nodes in a stream. See [“Generating Graphs” on page 108](#) for more information.

*Note:* The interactive tree itself cannot be saved. To avoid losing your work, generate a model and/or update tree directives before closing the tree builder window.

### Generating a Model from the Tree Builder

To generate a model based on the current tree, from the tree builder menus choose:

#### Generate > Model

In the Generate New Model dialog box you can choose from the following options:

**Model name.** You can specify a custom name or generate the name automatically based on the name of the modeling node.

**Create node on.** You can add the node on the **Canvas**, **GM Palette**, or **Both**.

**Include tree directives.** To include the directives from the current tree in the generated model, select this box. This enables you to regenerate the tree, if required. See the topic [“Tree-Growing Directives” on page 79](#) for more information.

### Tree-Growing Directives

For C&R Tree, CHAID, and QUEST models, tree directives specify conditions for growing the tree, one level at a time. Directives are applied each time the interactive tree builder is launched from the node.

- Directives are most safely used as a way to regenerate a tree created during a previous interactive session. See the topic [“Updating Tree Directives” on page 81](#) for more information. You can also edit directives manually, but this should be done with care.
- Directives are highly specific to the structure of the tree they describe. Thus, any change to the underlying data or modeling options may cause a previously valid set of directives to fail. For example, if the CHAID algorithm changes a two-way split to a three-way split based on updated data, any directives based on the previous two-way split would fail.

*Note:* If you choose to generate a model directly (without using the tree builder), any tree directives are ignored.

#### Editing Directives

1. To view or edit saved directives, open the tree-building node and select the Objective panel of the Build Options tab.
2. Select **Launch interactive session** to enable the controls, select **Use tree directives**, and click **Directives**.

## Directive Syntax

Directives specify conditions for growing the tree, starting with the root node. For example to grow the tree one level:

```
Grow Node Index 0 Children 1 2
```

Since no predictor is specified, the algorithm chooses the best split.

Note that the first split must always be on the root node (Index 0) and the index values for both children must be specified (1 and 2 in this case). It is invalid to specify `Grow Node Index 2 Children 3 4` unless you first grew the root that created Node 2.

To grow the tree:

```
Grow Tree
```

To grow and prune the tree (C&R Tree only):

```
Grow_And_Prune Tree
```

To specify a custom split for a continuous predictor:

```
Grow Node Index 0 Children 1 2 Spliton  
  ( "EDUCATE", Interval ( NegativeInfinity, 12.5)  
    Interval ( 12.5, Infinity ))
```

To split on a nominal predictor with two values:

```
Grow Node Index 2 Children 3 4 Spliton  
  ( "GENDER", Group( "0.0" )Group( "1.0" ))
```

To split on a nominal predictor with multiple values:

```
Grow Node Index 6 Children 7 8 Spliton  
  ( "ORGS", Group( "2.0","4.0" )  
    Group( "0.0","1.0","3.0","6.0" ))
```

To split on an ordinal predictor:

```
Grow Node Index 4 Children 5 6 Spliton  
  ( "CHILDS", Interval ( NegativeInfinity, 1.0)  
    Interval ( 1.0, Infinity ))
```

*Note:* When specifying custom splits, field names and values (EDUCATE, GENDER, CHILDS, etc.) are case sensitive.

## Directives for CHAID Trees

Directives for CHAID trees are particularly sensitive to changes in the data or model because--unlike C&R Tree and QUEST--they are not constrained to use binary splits. For example, the following syntax looks perfectly valid but would fail if the algorithm splits the root node into more than two children:

```
Grow Node Index 0 Children 1 2  
Grow Node Index 1 Children 3 4
```

With CHAID, it is possible that Node 0 will have 3 or 4 children, which would cause the second line of syntax to fail.

## Using Directives in Scripts

Directives can also be embedded in scripts using triple quotation marks.

## Updating Tree Directives

To preserve your work from an interactive tree-building session, you can save the directives used to generate the current tree. Unlike saving a model nugget, which cannot be edited further, this enables you to regenerate the tree in its current state for further editing.

To update directives, from the tree builder menus choose:

### File > Update Directives

Directives are saved in the modeling node used to create the tree (either C&R Tree, QUEST, or CHAID) and can be used to regenerate the current tree. See the topic [“Tree-Growing Directives”](#) on page 79 for more information.

## Exporting Model, Gain, and Risk Information

From the tree builder, you can export model, gain, and risk statistics in text, HTML, or image formats as appropriate.

1. In the tree builder window, select the tab or view that you want to export.
2. From the menus choose:

### File > Export

3. Select **Text**, **HTML**, or **Graph** as appropriate, and select the specific items you want to export from the submenu.

Where applicable, the export is based on current selections.

**Exporting Text or HTML formats.** You can export gain or risk statistics for the training or testing partition (if defined). The export is based on the current selections on the Gains tab—for example, you can choose node-by-node, cumulative, or quantile statistics.

**Exporting graphics.** You can export the current tree as displayed on the Viewer tab or export gains charts for the training or testing partition (if defined). Available formats include *.JPEG*, *.PNG*, and *.BMP*. For gains, the export is based on current selections on the Gains tab (available only when a chart is displayed).

## Generating Filter and Select Nodes

In the tree builder window, or when browsing a decision tree model nugget, from the menus choose:

### Generate > Filter Node

or

### > Select Node

**Filter Node.** Generates a node that filters any fields not used by the current tree. This is a quick way to pare down the dataset to include only those fields that are selected as important by the algorithm. If there is a Type node upstream from this decision tree node, any fields with the role *Target* are passed on by the Filter model nugget.

**Select Node.** Generates a node that selects all records that fall into the current node. This option requires that one or more tree branches be selected on the Viewer tab.

The model nugget is placed on the stream canvas.

## Generating a Rule Set from a Decision Tree

You can generate a Rule Set model nugget that represents the tree structure as a set of rules that define the terminal branches of the tree. Rule sets can often retain most of the important information from a full decision tree but with a less complex model. The most important difference is that with a rule set, more than one rule might apply for any particular record or no rules at all might apply. For example, you might see all of the rules that predict a *no* outcome followed by all of the rules that predict *yes*. If multiple rules apply, each rule gets a weighted "vote" based on the confidence that is associated with that rule, and the final prediction is decided by combining the weighted votes of all of the rules that apply to the record in question. If no rule applies, a default prediction is assigned to the record.

**Note:** When you score a rule set you might notice differences in scoring compared to scoring against the tree; this is because each terminal branch in a tree is scored independently. One area where this difference might become noticeable is when there are missing values in your data.

Rule sets can be generated only from trees with categorical target fields (no regression trees).

In the tree builder window, or when you browse a decision tree model nugget, from the menus choose:

#### **Generate > Rule Set**

**Rule set name** Specify the name of the new Rule Set model nugget.

**Create node on** Controls the location of the new Rule Set model nugget. Select **Canvas**, **GM Palette**, or **Both**.

**Minimum instances** Specify the minimum number of instances (number of records to which the rule applies) to preserve in the Rule Set model nugget. Rules with support less than the specified value are not included in the new rule set.

**Minimum confidence** Specify the minimum confidence for rules to be preserved in the Rule Set model nugget. Rules with confidence less than the specified value are not included in the new rule set.

## Building a Tree Model Directly

---

As an alternative to using the interactive tree builder, you can build a decision tree model directly from the node when the stream is run. This is consistent with most other model-building nodes. For C5.0 tree and Tree-AS models, which are not supported by the interactive tree builder, this is the only method that can be used.

1. Create a stream and add one of the decision tree nodes—C&R Tree, CHAID, QUEST, C5.0, or Tree-AS.
2. For C&R Tree, QUEST or CHAID, on the Objective panel of the Build Options tab, choose one of the main objectives. If you choose **Build a single tree**, ensure that **Mode** is set to **Generate model**.

For C5.0, on the Model tab, set **Output type** to **Decision tree**.

For Tree-AS, on the Basics panel of the Build Options tab, select the **Tree growing algorithm** type.

3. Select target and predictor fields and specify additional model options, as needed. For specific instructions, see the documentation for each tree-building node.
4. Run the stream to generate the model.

Comments about tree building

- When generating trees using this method, tree-growing directives are ignored.
- Whether interactive or direct, both methods of creating decision trees ultimately generate similar models. It's just a question of how much control you want along the way.

## Decision Tree Nodes

---

The Decision Tree nodes in IBM SPSS Modeler provide access to the following tree-building algorithms:

- C&R Tree
- QUEST
- CHAID
- C5.0
- Tree-AS
- Random Trees

See the topic [“Decision Tree Models”](#) on page 71 for more information.

The algorithms are similar in that they can all construct a decision tree by recursively splitting the data into smaller and smaller subgroups. However, there are some important differences.

**Input fields.** The input fields (predictors) can be any of the following types (measurement levels): continuous, categorical, flag, nominal or ordinal.

**Target fields.** Only one target field can be specified. For C&R Tree, CHAID, Tree-AS, and Random Trees, the target can be continuous, categorical, flag, nominal or ordinal. For QUEST it can be categorical, flag or nominal. For C5.0 the target can be flag, nominal or ordinal.

**Type of split.** C&R Tree, QUEST, and Random Trees support only binary splits (that is, each node of the tree can be split into no more than two branches). By contrast, CHAID, C5.0, and Tree-AS support splitting into more than two branches at a time.

**Method used for splitting.** The algorithms differ in the criteria used to decide the splits. When C&R Tree predicts a categorical output, a dispersion measure is used (by default the Gini coefficient, though you can change this). For continuous targets, the least squared deviation method is used. CHAID and Tree-AS use a chi-square test; QUEST uses a chi-square test for categorical predictors, and analysis of variance for continuous inputs. For C5.0 an information theory measure is used, the information gain ratio.

**Missing value handling.** All algorithms allow missing values for the predictor fields, though they use different methods to handle them. C&R Tree and QUEST use substitute prediction fields, where needed, to advance a record with missing values through the tree during training. CHAID makes the missing values a separate category and enables them to be used in tree building. C5.0 uses a fractioning method, which passes a fractional part of a record down each branch of the tree from a node where the split is based on a field with a missing value.

**Pruning.** C&R Tree, QUEST and C5.0 offer the option to grow the tree fully and then prune it back by removing bottom-level splits that do not contribute significantly to the accuracy of the tree. However, all of the decision tree algorithms allow you to control the minimum subgroup size, which helps avoid branches with few data records.

**Interactive tree building.** C&R Tree, QUEST and CHAID provide an option to launch an interactive session. This enables you to build your tree one level at a time, edit the splits, and prune the tree before you create the model. C5.0, Tree-AS, and Random Trees do not have an interactive option.

**Prior probabilities.** C&R Tree and QUEST support the specification of prior probabilities for categories when predicting a categorical target field. Prior probabilities are estimates of the overall relative frequency for each target category in the population from which the training data are drawn. In other words, they are the probability estimates that you would make for each possible target value prior to knowing anything about predictor values. CHAID, C5.0, Tree-AS, and Random Trees do not support specifying prior probabilities.

**Rule sets.** Not available for Tree-AS or Random Trees. For models with categorical target fields, the decision tree nodes provide the option to create the model in the form of a rule set, which can sometimes be easier to interpret than a complex decision tree. For C&R Tree, QUEST and CHAID you can generate a rule set from an interactive session; for C5.0 you can specify this option on the modeling node. In addition, all decision tree models enable you to generate a rule set from the model nugget. See the topic [“Generating a Rule Set from a Decision Tree”](#) on page 81 for more information.

## C&R Tree Node

The Classification and Regression (C&R) Tree node is a tree-based classification and prediction method. Similar to C5.0, this method uses recursive partitioning to split the training records into segments with similar output field values. The C&R Tree node starts by examining the input fields to find the best split, measured by the reduction in an impurity index that results from the split. The split defines two subgroups, each of which is subsequently split into two more subgroups, and so on, until one of the stopping criteria is triggered. All splits are binary (only two subgroups).

### Pruning

C&R Trees give you the option to first grow the tree and then prune based on a cost-complexity algorithm that adjusts the risk estimate based on the number of terminal nodes. This method, which enables the tree to grow large before pruning based on more complex criteria, may result in smaller trees with better cross-validation properties. Increasing the number of terminal nodes generally reduces the risk for the current (training) data, but the actual risk may be higher when the model is generalized to unseen data. In

an extreme case, suppose you have a separate terminal node for each record in the training set. The risk estimate would be 0%, since every record falls into its own node, but the risk of misclassification for unseen (testing) data would almost certainly be greater than 0. The cost-complexity measure attempts to compensate for this.

**Example.** A cable TV company has commissioned a marketing study to determine which customers would buy a subscription to an interactive news service via cable. Using the data from the study, you can create a stream in which the target field is the intent to buy the subscription and the predictor fields include age, sex, education, income category, hours spent watching television each day, and number of children. By applying a C&R Tree node to the stream, you will be able to predict and classify the responses to get the highest response rate for your campaign.

**Requirements.** To train a C&R Tree model, you need one or more *Input* fields and exactly one *Target* field. Target and input fields can be continuous (numeric range) or categorical. Fields set to *Both* or *None* are ignored. Fields used in the model must have their types fully instantiated, and any ordinal (ordered set) fields used in the model must have numeric storage (not string). If necessary, the Reclassify node can be used to convert them.

**Strengths.** C&R Tree models are quite robust in the presence of problems such as missing data and large numbers of fields. They usually do not require long training times to estimate. In addition, C&R Tree models tend to be easier to understand than some other model types--the rules derived from the model have a very straightforward interpretation. Unlike C5.0, C&R Tree can accommodate continuous as well as categorical output fields.

## CHAID Node

CHAID, or Chi-squared Automatic Interaction Detection, is a classification method for building decision trees by using chi-square statistics to identify optimal splits.

CHAID first examines the crosstabulations between each of the input fields and the outcome, and tests for significance using a chi-square independence test. If more than one of these relations is statistically significant, CHAID will select the input field that is the most significant (smallest *p* value). If an input has more than two categories, these are compared, and categories that show no differences in the outcome are collapsed together. This is done by successively joining the pair of categories showing the least significant difference. This category-merging process stops when all remaining categories differ at the specified testing level. For nominal input fields, any categories can be merged; for an ordinal set, only contiguous categories can be merged.

Exhaustive CHAID is a modification of CHAID that does a more thorough job of examining all possible splits for each predictor but takes longer to compute.

**Requirements.** Target and input fields can be continuous or categorical; nodes can be split into two or more subgroups at each level. Any ordinal fields used in the model must have numeric storage (not string). If necessary, the Reclassify node can be used to convert them.

**Strengths.** Unlike the C&R Tree and QUEST nodes, CHAID can generate nonbinary trees, meaning that some splits have more than two branches. It therefore tends to create a wider tree than the binary growing methods. CHAID works for all types of inputs, and it accepts both case weights and frequency variables.

## QUEST Node

QUEST—or Quick, Unbiased, Efficient Statistical Tree—is a binary classification method for building decision trees. A major motivation in its development was to reduce the processing time required for large C&R Tree analyses with either many variables or many cases. A second goal of QUEST was to reduce the tendency found in classification tree methods to favor inputs that allow more splits, that is, continuous (numeric range) input fields or those with many categories.

- QUEST uses a sequence of rules, based on significance tests, to evaluate the input fields at a node. For selection purposes, as little as a single test may need to be performed on each input at a node. Unlike C&R Tree, all splits are not examined, and unlike C&R Tree and CHAID, category combinations are not tested when evaluating an input field for selection. This speeds the analysis.



- Splits are determined by running quadratic discriminant analysis using the selected input on groups formed by the target categories. This method again results in a speed improvement over exhaustive search (C&R Tree) to determine the optimal split.

**Requirements.** Input fields can be continuous (numeric ranges), but the target field must be categorical. All splits are binary. Weight fields cannot be used. Any ordinal (ordered set) fields used in the model must have numeric storage (not string). If necessary, the Reclassify node can be used to convert them.

**Strengths.** Like CHAID, but unlike C&R Tree, QUEST uses statistical tests to decide whether or not an input field is used. It also separates the issues of input selection and splitting, applying different criteria to each. This contrasts with CHAID, in which the statistical test result that determines variable selection also produces the split. Similarly, C&R Tree employs the impurity-change measure to both select the input field and to determine the split.

## Decision Tree Node Fields Options

On the Fields tab, you choose whether you want to use the field role settings already defined in upstream nodes, or make the field assignments manually.

**Use predefined roles** This option uses the role settings (targets, predictors and so on) from an upstream Type node (or the Types tab of an upstream source node).

**Use custom field assignments.** To manually assign targets, predictors and other roles, select this option.

**Fields.** Use the arrow buttons to assign items manually from this list to the various role fields on the right of the screen. The icons indicate the valid measurement levels for each role field.

To select all the fields in the list, click the **All** button, or click an individual measurement level button to select all fields with that measurement level.

**Target.** Select one field as the target for the prediction.

**Predictors (Inputs).** Choose one or more fields as inputs for the prediction.

**Analysis Weight.** (CHAID, C&RT, and Trees-AS only) To use a field as a case weight, specify the field here. Case weights are used to account for differences in variance across levels of the output field. See the topic [“Using Frequency and Weight Fields” on page 27](#) for more information.

## Decision Tree Node Build Options

The Build Options tab is where you set all the options for building the model. You can, of course, just click the **Run** button to build a model with all the default options, but normally you will want to customize the build for your own purposes.

The tab contains several different panes on which you set the customizations that are specific to your model.

### Decision Tree Nodes - Objectives

For the C&R Tree, QUEST, and CHAID nodes, in the Objectives pane on the Build Options tab, you can choose whether to build a new model or update an existing one. You also set the main objective of the node: to build a standard model, to build one with enhanced accuracy or stability, or to build one for use with very large datasets.

#### What do you want to do?

**Build new model.** (Default) Creates a completely new model each time you run a stream containing this modeling node.

**Continue training existing model.** By default, a completely new model is created each time a modeling node is executed. If this option is selected, training continues with the last model successfully produced by the node. This makes it possible to update or refresh an existing model without having to access the original data and may result in significantly faster performance since *only* the new or updated records are fed into the stream. Details on the previous model are stored with the modeling node, making it possible to use this option even if the previous model nugget is no longer available in the stream or Models palette.

**Note:** This option is activated only if you select **Build a single tree** (for C&R Tree, CHAID, and QUEST), **Create a standard model** (for Neural Net and Linear), or **Create a model for very large datasets** as the objective.

### What is your main objective?

- **Build a single tree.** Creates a single, standard decision tree model. Standard models are generally easier to interpret, and can be faster to score, than models built using the other objective options.

**Note:** For split models, to use this option with **Continue training existing model** you must be connected to Analytic Server.

**Mode.** Specifies the method used to build the model. **Generate model** creates a model automatically when the stream is run. **Launch interactive session** opens the tree builder, which enables you to build your tree one level at a time, edit splits, and prune as desired before creating the model nugget.

**Use tree directives.** Select this option to specify directives to apply when generating an interactive tree from the node. For example, you can specify the first- and second-level splits, and these would automatically be applied when the tree builder is launched. You can also save directives from an interactive tree-building session in order to re-create the tree at a future date. See the topic [“Updating Tree Directives”](#) on page 81 for more information.

- **Enhance model accuracy (boosting).** Choose this option if you want to use a special method, known as **boosting**, to improve the model accuracy rate. Boosting works by building multiple models in a sequence. The first model is built in the usual way. Then, a second model is built in such a way that it focuses on the records that were misclassified by the first model. Then a third model is built to focus on the second model's errors, and so on. Finally, cases are classified by applying the whole set of models to them, using a weighted voting procedure to combine the separate predictions into one overall prediction. Boosting can significantly improve the accuracy of a decision tree model, but it also requires longer training.
- **Enhance model stability (bagging).** Choose this option if you want to use a special method, known as **bagging** (bootstrap aggregating), to improve the stability of the model and to avoid overfitting. This option creates multiple models and combines them, in order to obtain more reliable predictions. Models obtained using this option can take longer to build and score than standard models.
- **Create a model for very large datasets.** Choose this option when working with datasets that are too large to build a model using any of the other objective options. This option divides the data into smaller data blocks and builds a model on each block. The most accurate models are then automatically selected and combined into a single model nugget. You can perform incremental model updating if you select the **Continue training existing model** option on this screen.

**Note:** This option for very large datasets requires a connection to IBM SPSS Modeler Server.

### Decision Tree Nodes - Basics

Specify the basic options about how the decision tree is to be built.

**Tree growing algorithm** (CHAID and Tree-AS only) Choose the type of **CHAID** algorithm you want to use. **Exhaustive CHAID** is a modification of CHAID that does a more thorough job of examining all possible splits for each predictor but takes longer to compute.

**Maximum tree depth** Specify the maximum number of levels below the root node (the number of times the sample will be split recursively). The default is 5; choose **Custom** and enter a value to specify a different number of levels.

### Pruning (C&RT and QUEST only)

**Prune tree to avoid overfitting** Pruning consists of removing bottom-level splits that do not contribute significantly to the accuracy of the tree. Pruning can help simplify the tree, making it easier to interpret and, in some cases, improving generalization. If you want the full tree without pruning, leave this option deselected.

- **Set maximum difference in risk (in Standard Errors)** Enables you to specify a more liberal pruning rule. The standard error rule enables the algorithm to select the simplest tree whose risk estimate is close to (but possibly greater than) that of the subtree with the smallest risk. The value indicates the size of the allowable difference in the risk estimate between the pruned tree and the tree with the smallest risk in terms of the risk estimate. For example, if you specify 2, a tree whose risk estimate is (2 × standard error) larger than that of the full tree could be selected.

**Maximum surrogates.** Surrogates are a method for dealing with missing values. For each split in the tree, the algorithm identifies the input fields that are most similar to the selected split field. Those fields are the *surrogates* for that split. When a record must be classified but has a missing value for a split field, its value on a surrogate field can be used to make the split. Increasing this setting will allow more flexibility to handle missing values but may also lead to increased memory usage and longer training times.

### Decision Tree Nodes - Stopping Rules

These options control how the tree is constructed. Stopping rules determine when to stop splitting specific branches of the tree. Set the minimum branch sizes to prevent splits that would create very small subgroups. **Minimum records in parent branch** prevents a split if the number of records in the node to be split (the *parent*) is less than the specified value. **Minimum records in child branch** prevents a split if the number of records in any branch created by the split (the *child*) would be less than the specified value.

- **Use percentage** Specify sizes in terms of percentage of overall training data.
- **Use absolute value** Specify sizes as the absolute numbers of records.

### Decision Tree Nodes - Ensembles

These settings determine the behavior of ensembling that occurs when boosting, bagging, or very large datasets are requested in Objectives. Options that do not apply to the selected objective are ignored.

**Bagging and Very Large Datasets.** When scoring an ensemble, this is the rule used to combine the predicted values from the base models to compute the ensemble score value.

- **Default combining rule for categorical targets.** Ensemble predicted values for categorical targets can be combined using voting, highest probability, or highest mean probability. **Voting** selects the category that has the highest probability most often across the base models. **Highest probability** selects the category that achieves the single highest probability across all base models. **Highest mean probability** selects the category with the highest value when the category probabilities are averaged across base models.
- **Default combining rule for continuous targets.** Ensemble predicted values for continuous targets can be combined using the mean or median of the predicted values from the base models.

Note that when the objective is to enhance model accuracy, the combining rule selections are ignored. Boosting always uses a weighted majority vote to score categorical targets and a weighted median to score continuous targets.

**Boosting and Bagging.** Specify the number of base models to build when the objective is to enhance model accuracy or stability; for bagging, this is the number of bootstrap samples. It should be a positive integer.

### C&R Tree and QUEST Nodes - Costs & Priors

#### Misclassification Costs

In some contexts, certain kinds of errors are more costly than others. For example, it may be more costly to classify a high-risk credit applicant as low risk (one kind of error) than it is to classify a low-risk applicant as high risk (a different kind of error). Misclassification costs allow you to specify the relative importance of different kinds of prediction errors.

Misclassification costs are basically weights applied to specific outcomes. These weights are factored into the model and may actually change the prediction (as a way of protecting against costly mistakes).

With the exception of C5.0 models, misclassification costs are not applied when scoring a model and are not taken into account when ranking or comparing models using an Auto Classifier node, evaluation chart,

or Analysis node. A model that includes costs may not produce fewer errors than one that doesn't and may not rank any higher in terms of overall accuracy, but it is likely to perform better in practical terms because it has a built-in bias in favor of *less expensive* errors.

The cost matrix shows the cost for each possible combination of predicted category and actual category. By default, all misclassification costs are set to 1.0. To enter custom cost values, select **Use misclassification costs** and enter your custom values into the cost matrix.

To change a misclassification cost, select the cell corresponding to the desired combination of predicted and actual values, delete the existing contents of the cell, and enter the desired cost for the cell. Costs are not automatically symmetrical. For example, if you set the cost of misclassifying *A* as *B* to be 2.0, the cost of misclassifying *B* as *A* will still have the default value of 1.0 unless you explicitly change it as well.

## Priors

These options allow you to specify prior probabilities for categories when predicting a categorical target field. **Prior probabilities** are estimates of the overall relative frequency for each target category in the population from which the training data are drawn. In other words, they are the probability estimates that you would make for each possible target value *prior* to knowing anything about predictor values. There are three methods of setting priors:

- **Based on training data.** This is the default. Prior probabilities are based on the relative frequencies of the categories in the training data.
- **Equal for all classes.** Prior probabilities for all categories are defined as  $1/k$ , where  $k$  is the number of target categories.
- **Custom.** You can specify your own prior probabilities. Starting values for prior probabilities are set as equal for all classes. You can adjust the probabilities for individual categories to user-defined values. To adjust a specific category's probability, select the probability cell in the table corresponding to the desired category, delete the contents of the cell, and enter the desired value.

The prior probabilities for all categories should sum to 1.0 (the **probability constraint**). If they do not sum to 1.0, a warning is displayed, with an option to automatically normalize the values. This automatic adjustment preserves the proportions across categories while enforcing the probability constraint. You can perform this adjustment at any time by clicking the **Normalize** button. To reset the table to equal values for all categories, click the **Equalize** button.

**Adjust priors using misclassification costs.** This option enables you to adjust the priors, based on misclassification costs (specified on the Costs tab). This enables you to incorporate cost information directly into the tree-growing process for trees that use the Twoing impurity measure. (When this option is not selected, cost information is used only in classifying records and calculating risk estimates for trees based on the Twoing measure.)

## CHAID Node - Costs

In some contexts, certain kinds of errors are more costly than others. For example, it may be more costly to classify a high-risk credit applicant as low risk (one kind of error) than it is to classify a low-risk applicant as high risk (a different kind of error). Misclassification costs allow you to specify the relative importance of different kinds of prediction errors.

Misclassification costs are basically weights applied to specific outcomes. These weights are factored into the model and may actually change the prediction (as a way of protecting against costly mistakes).

With the exception of C5.0 models, misclassification costs are not applied when scoring a model and are not taken into account when ranking or comparing models using an Auto Classifier node, evaluation chart, or Analysis node. A model that includes costs may not produce fewer errors than one that doesn't and may not rank any higher in terms of overall accuracy, but it is likely to perform better in practical terms because it has a built-in bias in favor of *less expensive* errors.

The cost matrix shows the cost for each possible combination of predicted category and actual category. By default, all misclassification costs are set to 1.0. To enter custom cost values, select **Use misclassification costs** and enter your custom values into the cost matrix.

To change a misclassification cost, select the cell corresponding to the desired combination of predicted and actual values, delete the existing contents of the cell, and enter the desired cost for the cell. Costs are not automatically symmetrical. For example, if you set the cost of misclassifying *A* as *B* to be 2.0, the cost of misclassifying *B* as *A* will still have the default value of 1.0 unless you explicitly change it as well.

### C&R Tree Node - Advanced

The advanced options enable you to fine-tune the tree-building process.

**Minimum change in impurity.** Specify the minimum change in impurity to create a new split in the tree. **Impurity** refers to the extent to which subgroups defined by the tree have a wide range of output field values within each group. For categorical targets, a node is considered “pure” if 100% of cases in the node fall into a specific category of the target field. The goal of tree building is to create subgroups with similar output values--in other words, to minimize the impurity within each node. If the best split for a branch reduces the impurity by less than the specified amount, the split will not be made.

**Impurity measure for categorical targets.** For categorical target fields, specify the method used to measure the impurity of the tree. (For continuous targets, this option is ignored, and the **least squared deviation** impurity measure is always used.)

- **Gini** is a general impurity measure based on probabilities of category membership for the branch.
- **Twoing** is an impurity measure that emphasizes the binary split and is more likely to lead to approximately equal-sized branches from a split.
- **Ordered** adds the additional constraint that only contiguous target classes can be grouped together, as is applicable only with ordinal targets. If this option is selected for a nominal target, the standard twoing measure is used by default.

**Overfit prevention set.** The algorithm internally separates records into a model building set and an overfit prevention set, which is an independent set of data records used to track errors during training in order to prevent the method from modeling chance variation in the data. Specify a percentage of records. The default is 30.

**Replicate results.** Setting a random seed enables you to replicate analyses. Specify an integer or click **Generate**, which will create a pseudo-random integer between 1 and 2147483647, inclusive.

### QUEST Node - Advanced

The advanced options enable you to fine-tune the tree-building process.

**Significance level for splitting.** Specifies the significance level (alpha) for splitting nodes. The value must be between 0 and 1. Lower values tend to produce trees with fewer nodes.

**Overfit prevention set.** The algorithm internally separates records into a model building set and an overfit prevention set, which is an independent set of data records used to track errors during training in order to prevent the method from modeling chance variation in the data. Specify a percentage of records. The default is 30.

**Replicate results.** Setting a random seed enables you to replicate analyses. Specify an integer or click **Generate**, which will create a pseudo-random integer between 1 and 2147483647, inclusive.

### CHAID Node - Advanced

The advanced options enable you to fine-tune the tree-building process.

**Significance level for splitting.** Specifies the significance level (alpha) for splitting nodes. The value must be between 0 and 1. Lower values tend to produce trees with fewer nodes.

**Significance level for merging.** Specifies the significance level (alpha) for merging categories. The value must be greater than 0 and less than or equal to 1. To prevent any merging of categories, specify a value of 1. For continuous targets, this means the number of categories for the variable in the final tree matches the specified number of intervals. This option is not available for Exhaustive CHAID.

**Adjust significance values using Bonferroni method.** Adjusts significance values when testing the various category combinations of a predictor. Values are adjusted based on the number of tests, which

directly relates to the number of categories and measurement level of a predictor. This is generally desirable because it better controls the false-positive error rate. Disabling this option will increase the power of your analysis to find true differences, but at the cost of an increased false-positive rate. In particular, disabling this option may be recommended for small samples.

**Allow resplitting of merged categories within a node.** The CHAID algorithm attempts to merge categories in order to produce the simplest tree that describes the model. If selected, this option enables merged categories to be resplit if that results in a better solution.

**Chi-square for categorical targets.** For categorical targets, you can specify the method used to calculate the chi-square statistic.

- **Pearson.** This method provides faster calculations but should be used with caution on small samples.
- **Likelihood ratio.** This method is more robust than Pearson but takes longer to calculate. For small samples, this is the preferred method. For continuous targets, this method is always used.

**Minimum change in expected cell frequencies.** When estimating cell frequencies (for both the nominal model and the row effects ordinal model), an iterative procedure (epsilon) is used to converge on the optimal estimate used in the chi-square test for a specific split. Epsilon determines how much change must occur for iterations to continue; if the change from the last iteration is smaller than the specified value, iterations stop. If you are having problems with the algorithm not converging, you can increase this value or increase the maximum number of iterations until convergence occurs.

**Maximum iterations for convergence.** Specifies the maximum number of iterations before stopping, whether convergence has taken place or not.

**Overfit prevention set.** (This option is only available when using the interactive tree builder.) The algorithm internally separates records into a model building set and an overfit prevention set, which is an independent set of data records used to track errors during training in order to prevent the method from modeling chance variation in the data. Specify a percentage of records. The default is 30.

**Replicate results.** Setting a random seed enables you to replicate analyses. Specify an integer or click **Generate**, which will create a pseudo-random integer between 1 and 2147483647, inclusive.

## Decision Tree Node Model Options

On the Model Options tab, you can choose whether to specify a name for the model, or generate a name automatically. You can also choose to obtain predictor importance information, as well as raw and adjusted propensity scores for flag targets.

**Model name.** You can generate the model name automatically based on the target or ID field (or model type in cases where no such field is specified) or specify a custom name.

### Model Evaluation

**Calculate predictor importance.** For models that produce an appropriate measure of importance, you can display a chart that indicates the relative importance of each predictor in estimating the model. Typically you will want to focus your modeling efforts on the predictors that matter most, and consider dropping or ignoring those that matter least. Note that predictor importance may take longer to calculate for some models, particularly when working with large datasets, and is off by default for some models as a result. Predictor importance is not available for decision list models. See [“Predictor Importance” on page 37](#) for more information.

### Propensity Scores

Propensity scores can be enabled in the modeling node, and on the Settings tab in the model nugget. This functionality is available only when the selected target is a flag field. See the topic [“Propensity Scores” on page 30](#) for more information.

**Calculate raw propensity scores.** Raw propensity scores are derived from the model based on the training data only. If the model predicts the *true* value (will respond), then the propensity is the same as P, where P is the probability of the prediction. If the model predicts the false value, then the propensity is calculated as  $(1 - P)$ .

- If you choose this option when building the model, propensity scores will be enabled in the model nugget by default. However, you can always choose to enable raw propensity scores in the model nugget whether or not you select them in the modeling node.
- When scoring the model, raw propensity scores will be added in a field with the letters *RP* appended to the standard prefix. For example, if the predictions are in a field named *\$R-churn*, the name of the propensity score field will be *\$RRP-churn*.

**Calculate adjusted propensity scores.** Raw propensities are based purely on estimates given by the model, which may be overfitted, leading to over-optimistic estimates of propensity. Adjusted propensities attempt to compensate by looking at how the model performs on the test or validation partitions and adjusting the propensities to give a better estimate accordingly.

- This setting requires that a valid partition field is present in the stream.
- Unlike raw confidence scores, adjusted propensity scores must be calculated when building the model; otherwise, they will not be available when scoring the model nugget.
- When scoring the model, adjusted propensity scores will be added in a field with the letters *AP* appended to the standard prefix. For example, if the predictions are in a field named *\$R-churn*, the name of the propensity score field will be *\$RAP-churn*. Adjusted propensity scores are not available for logistic regression models.
- When calculating the adjusted propensity scores, the test or validation partition used for the calculation must not have been balanced. To avoid this, be sure the **Only balance training data** option is selected in any upstream Balance nodes. In addition, if a complex sample has been taken upstream this will invalidate the adjusted propensity scores.
- Adjusted propensity scores are not available for "boosted" tree and rule set models. See the topic ["Boosted C5.0 Models" on page 107](#) for more information.

**Based on.** For adjusted propensity scores to be computed, a partition field must be present in the stream. You can specify whether to use the testing or validation partition for this computation. For best results, the testing or validation partition should include at least as many records as the partition used to train the original model.

## C5.0 Node

---

This feature is available in SPSS Modeler Professional and SPSS Modeler Premium.

This node uses the C5.0 algorithm to build either a **decision tree** or a **rule set**. A C5.0 model works by splitting the sample based on the field that provides the maximum **information gain**. Each subsample defined by the first split is then split again, usually based on a different field, and the process repeats until the subsamples cannot be split any further. Finally, the lowest-level splits are reexamined, and those that do not contribute significantly to the value of the model are removed or **pruned**.

*Note:* The C5.0 node can predict only a categorical target. When analyzing data with categorical (nominal or ordinal) fields, the node is more likely to group categories together than versions of C5.0 prior to release 11.0.

C5.0 can produce two kinds of models. A **decision tree** is a straightforward description of the splits found by the algorithm. Each terminal (or "leaf") node describes a particular subset of the training data, and each case in the training data belongs to exactly one terminal node in the tree. In other words, exactly one prediction is possible for any particular data record presented to a decision tree.

In contrast, a **rule set** is a set of rules that tries to make predictions for individual records. Rule sets are derived from decision trees and, in a way, represent a simplified or distilled version of the information found in the decision tree. Rule sets can often retain most of the important information from a full decision tree but with a less complex model. Because of the way rule sets work, they do not have the same properties as decision trees. The most important difference is that with a rule set, more than one rule may apply for any particular record, or no rules at all may apply. If multiple rules apply, each rule gets a weighted "vote" based on the confidence associated with that rule, and the final prediction is decided by combining the weighted votes of all of the rules that apply to the record in question. If no rule applies, a default prediction is assigned to the record.



**Example.** A medical researcher has collected data about a set of patients, all of whom suffered from the same illness. During their course of treatment, each patient responded to one of five medications. You can use a C5.0 model, in conjunction with other nodes, to help find out which drug might be appropriate for a future patient with the same illness.

**Requirements.** To train a C5.0 model, there must be one categorical (i.e., nominal or ordinal) *Target* field, and one or more *Input* fields of any type. Fields set to *Both* or *None* are ignored. Fields used in the model must have their types fully instantiated. A weight field can also be specified.

**Strengths.** C5.0 models are quite robust in the presence of problems such as missing data and large numbers of input fields. They usually do not require long training times to estimate. In addition, C5.0 models tend to be easier to understand than some other model types, since the rules derived from the model have a very straightforward interpretation. C5.0 also offers the powerful **boosting** method to increase accuracy of classification.

**Note:** C5.0 model building speed may benefit from enabling parallel processing.

## C5.0 Node Model Options

**Model name.** Specify the name of the model to be produced.

- **Auto.** With this option selected, the model name will be generated automatically, based on the target field name(s). This is the default.
- **Custom.** Select this option to specify your own name for the model nugget that will be created by this node.

**Use partitioned data.** If a partition field is defined, this option ensures that data from only the training partition is used to build the model.

**Create split models.** Builds a separate model for each possible value of input fields that are specified as split fields. See [“Building Split Models” on page 24](#) for more information.

**Output type.** Specify here whether you want the resulting model nugget to be a **Decision tree** or a **Rule set**.

**Group symbolics.** If this option is selected, C5.0 will attempt to combine symbolic values that have similar patterns with respect to the output field. If this option is not selected, C5.0 will create a child node for every value of the symbolic field used to split the parent node. For example, if C5.0 splits on a *COLOR* field (with values *RED*, *GREEN*, and *BLUE*), it will create a three-way split by default. However, if this option is selected, and the records where *COLOR* = *RED* are very similar to records where *COLOR* = *BLUE*, it will create a two-way split, with the *GREENs* in one group and the *BLUEs* and *REDs* together in the other.

**Use boosting.** The C5.0 algorithm has a special method for improving its accuracy rate, called **boosting**. It works by building multiple models in a sequence. The first model is built in the usual way. Then, a second model is built in such a way that it focuses on the records that were misclassified by the first model. Then a third model is built to focus on the second model's errors, and so on. Finally, cases are classified by applying the whole set of models to them, using a weighted voting procedure to combine the separate predictions into one overall prediction. Boosting can significantly improve the accuracy of a C5.0 model, but it also requires longer training. The **Number of trials** option enables you to control how many models are used for the boosted model. This feature is based on the research of Freund & Schapire, with some proprietary improvements to handle noisy data better.

**Cross-validate.** If this option is selected, C5.0 will use a set of models built on subsets of the training data to estimate the accuracy of a model built on the full dataset. This is useful if your dataset is too small to split into traditional training and testing sets. The cross-validation models are discarded after the accuracy estimate is calculated. You can specify the **number of folds**, or the number of models used for cross-validation. Note that in previous versions of IBM SPSS Modeler, building the model and cross-validating it were two separate operations. In the current version, no separate model-building step is required. Model building and cross-validation are performed at the same time.

**Mode.** For **Simple** training, most of the C5.0 parameters are set automatically. **Expert** training allows more direct control over the training parameters.



## Simple Mode Options

**Favor.** By default, C5.0 will try to produce the most accurate tree possible. In some instances, this can lead to overfitting, which can result in poor performance when the model is applied to new data. Select **Generality** to use algorithm settings that are less susceptible to this problem.

*Note:* Models built with the **Generality** option selected are not guaranteed to generalize better than other models. When generality is a critical issue, always validate your model against a held-out test sample.

**Expected noise (%).** Specify the expected proportion of noisy or erroneous data in the training set.

## Expert Mode Options

**Pruning severity.** Determines the extent to which the decision tree or rule set will be pruned. Increase this value to obtain a smaller, more concise tree. Decrease it to obtain a more accurate tree. This setting affects local pruning only (see "Use global pruning" below).

**Minimum records per child branch.** The size of subgroups can be used to limit the number of splits in any branch of the tree. A branch of the tree will be split only if two or more of the resulting subbranches would contain at least this many records from the training set. The default value is 2. Increase this value to help prevent **overtraining** with noisy data.

**Use global pruning.** Trees are pruned in two stages: First, a local pruning stage, which examines subtrees and collapses branches to increase the accuracy of the model. Second, a global pruning stage considers the tree as a whole, and weak subtrees may be collapsed. Global pruning is performed by default. To omit the global pruning stage, deselect this option.

**Winnow attributes.** If this option is selected, C5.0 will examine the usefulness of the predictors before starting to build the model. Predictors that are found to be irrelevant are then excluded from the model-building process. This option can be helpful for models with many predictor fields and can help prevent overfitting.

**Note:** C5.0 model building speed may benefit from enabling parallel processing.

## Tree-AS node

---

The Tree-AS node can be used with data in a distributed environment. In this node you can choose to build decision trees using either a CHAID or Exhaustive CHAID model.

CHAID, or Chi-squared Automatic Interaction Detection, is a classification method for building decision trees by using chi-square statistics to identify optimal splits.

CHAID first examines the crosstabulations between each of the input fields and the outcome, and tests for significance using a chi-square independence test. If more than one of these relations is statistically significant, CHAID will select the input field that is the most significant (smallest  $p$  value). If an input has more than two categories, these are compared, and categories that show no differences in the outcome are collapsed together. This is done by successively joining the pair of categories showing the least significant difference. This category-merging process stops when all remaining categories differ at the specified testing level. For nominal input fields, any categories can be merged; for an ordinal set, only contiguous categories can be merged.

Exhaustive CHAID is a modification of CHAID that does a more thorough job of examining all possible splits for each predictor but takes longer to compute.

**Requirements.** Target and input fields can be continuous or categorical; nodes can be split into two or more subgroups at each level. Any ordinal fields used in the model must have numeric storage (not string). If necessary, use the Reclassify node to convert them.

**Strengths.** CHAID can generate nonbinary trees, meaning that some splits have more than two branches. It therefore tends to create a wider tree than the binary growing methods. CHAID works for all types of inputs, and it accepts both case weights and frequency variables.

## Tree-AS node fields options

On the Fields tab, you choose whether you want to use the field role settings already defined in upstream nodes, or make the field assignments manually.

**Use predefined roles** This option uses the role settings (targets, predictors and so on) from an upstream Type node (or the Types tab of an upstream source node).

**Use custom field assignments.** To manually assign targets, predictors and other roles, select this option.

**Fields.** Use the arrow buttons to assign items manually from this list to the various role fields on the right of the screen. The icons indicate the valid measurement levels for each role field.

To select all the fields in the list, click the **All** button, or click an individual measurement level button to select all fields with that measurement level.

**Target.** Select one field as the target for the prediction.

**Predictors** Select one or more fields as inputs for the prediction.

**Analysis Weight** To use a field as a case weight, specify the field here. Case weights are used to account for differences in variance across levels of the output field. For more information, see [“Using Frequency and Weight Fields”](#) on page 27.

## Tree-AS node build options

The Build Options tab is where you set all the options for building the model. You can, of course, just click the **Run** button to build a model with all the default options, but normally you will want to customize the build for your own purposes.

The tab contains several different panes on which you set the customizations that are specific to your model.

### Tree-AS node - basics

Specify the basic options about how the decision tree is to be built.

**Tree growing algorithm** Select the type of **CHAID** algorithm you want to use. **Exhaustive CHAID** is a modification of CHAID that does a more thorough job of examining all possible splits for each predictor but takes longer to compute.

**Maximum tree depth** Specify the maximum number of levels below the root node (the number of times the sample is split recursively); the default is 5. The maximum number of levels (also referred to as *nodes*) is 50,000.

**Binning** If you use continuous data, you must bin the inputs. You can do this in a preceding node; however, the Tree-AS node automatically bins any continuous inputs. If you use the Tree-AS node to automatically bin the data, select the **Number of bins** into which the inputs are to be divided. Data is divided into bins with equal frequency; the available options are 2, 4, 5, 10, 20, 25, 50, or 100.

### Tree-AS node - growing

Use the growing options to fine-tune the tree-building process.

**Record threshold for switching from p-values to effect sizes** Specify the number of records at which the model will switch from using the **P-values settings** to the **Effect size settings** when building the tree. The default is 1,000,000.

**Significance level for splitting** Specify the significance level (alpha) for splitting nodes. The value must be between 0.01 and 0.99. Lower values tend to produce trees with fewer nodes.

**Significance level for merging** Specify the significance level (alpha) for merging categories. The value must be between 0.01 and 0.99. This option is not available for Exhaustive CHAID.

**Adjust significance values using Bonferroni method** Adjust significance values when you are testing the various category combinations of a predictor. Values are adjusted based on the number of tests, which directly relates to the number of categories and measurement level of a predictor. This is generally

desirable because it better controls the false-positive error rate. Disabling this option increases the power of your analysis to find true differences, but at the cost of an increased false-positive rate. In particular, disabling this option may be recommended for small samples.

**Effect size threshold (continuous targets only)** Set the effect size threshold to be used when splitting nodes and merging categories; when using a continuous target. The value must be between 0.01 and 0.99.

**Effect size threshold (categorical targets only)** Set the effect size threshold to be used when splitting nodes and merging categories; when using a categorical target. The value must be between 0.01 and 0.99.

**Allow resplitting of merged categories within a node** The CHAID algorithm attempts to merge categories in order to produce the simplest tree that describes the model. If selected, this option enables merged categories to be resplit if that results in a better solution.

**Significance level for grouping leaf nodes** Specify the significance level that determines how groups of leaf nodes are formed or how unusual leaf nodes are identified.

**Chi-square for categorical targets** For categorical targets, you can specify the method used to calculate the chi-square statistic.

- **Pearson** This method provides faster calculations but should be used with caution on small samples.
- **Likelihood ratio** This method is more robust than Pearson but takes longer to calculate. For small samples, this is the preferred method. For continuous targets, this method is always used.

### Tree-AS node - stopping rules

These options control how the tree is constructed. Stopping rules determine when to stop splitting specific branches of the tree. Set the minimum branch sizes to prevent splits that would create very small subgroups. **Minimum records in parent branch** prevents a split if the number of records in the node to be split (the *parent*) is less than the specified value. **Minimum records in child branch** prevents a split if the number of records in any branch created by the split (the *child*) would be less than the specified value.

- **Use percentage** Specify sizes in terms of percentage of overall training data.
- **Use absolute value** Specify sizes as the absolute numbers of records.

**Minimum change in expected cell frequencies** When estimating cell frequencies (for both the nominal model and the row effects ordinal model), an iterative procedure (epsilon) is used to converge on the optimal estimate used in the chi-square test for a specific split. Epsilon determines how much change must occur for iterations to continue; if the change from the last iteration is smaller than the specified value, iterations stop. If you are having problems with the algorithm not converging, you can increase this value or increase the maximum number of iterations until convergence occurs.

**Maximum iterations for convergence** Specifies the maximum number of iterations before stopping, whether convergence has taken place or not.

### Tree-AS node - costs

In some contexts, certain kinds of errors are more costly than others. For example, it may be more costly to classify a high-risk credit applicant as low risk (one kind of error) than it is to classify a low-risk applicant as high risk (a different kind of error). Misclassification costs allow you to specify the relative importance of different kinds of prediction errors.

Misclassification costs are basically weights applied to specific outcomes. These weights are factored into the model and may actually change the prediction (as a way of protecting against costly mistakes).

A model that includes costs might not produce fewer errors than one that doesn't, and might not rank any higher in terms of overall accuracy, but it is likely to perform better in practical terms because it has a built-in bias in favor of less expensive errors.

The cost matrix shows the cost for each possible combination of predicted category and actual category. By default, all misclassification costs are set to 1.0. To enter custom cost values, select **Use misclassification costs** and enter your custom values into the cost matrix.

To change a misclassification cost, select the cell corresponding to the desired combination of predicted and actual values, delete the existing contents of the cell, and enter the desired cost for the cell. Costs are not automatically symmetrical. For example, if you set the cost of misclassifying *A* as *B* to be 2.0, the cost of misclassifying *B* as *A* will still have the default value of 1.0 unless you explicitly change it as well.

For ordinal targets only, you can select the **Default cost increase for ordinal target** and set default values in the costs matrix. The available options are described in the following list.

- **No increase** - A default value of 1.0 for every correct prediction.
- **Linear** - Each successive incorrect prediction increases the cost by 1.
- **Square** - Each successive incorrect prediction is the square of the linear value. In this case, the values might be: 1, 4, 9, and so on.
- **Custom** - If you manually edit any values in the table, the drop-down option automatically changes to **Custom**. If you change the drop-down selection to any of the other options your edited values are replaced with the values for the selected option.

## Tree-AS node model options

On the Model Options tab, you can choose whether to specify a name for the model, or generate a name automatically. You can also choose to calculate confidence values and add an identifying ID during scoring of the model.

**Model name.** You can generate the model name automatically based on the target or ID field (or model type in cases where no such field is specified) or specify a custom name.

**Calculate confidences** To add a confidence field when the model is scored, select this check box.

**Rule identifier** To add a field when the model is scored that contains the ID of the leaf node a record was assigned to, select this check box.

## Tree-AS model nugget

### Tree-AS model nugget output

After you create a Tree-AS model, the following information is available in the Output viewer.

### Model Information table

The Model Information table provides key information about the model. The table identifies some high-level model settings, such as:

- The algorithm type used; either CHAID or Exhaustive CHAID.
- The name of the target field selected in either the Type node or the Tree-AS node Fields tab.
- The names of the fields selected as predictors in either the Type node or the Tree-AS node Fields tab.
- The number of records in the data. If you build a model with a frequency weight, this value is the weighted, valid count which represents the records on which the tree is based.
- The number of *leaf nodes* in the generated tree.
- The number of levels in the tree; that is, the tree depth.

### Predictor Importance

The Predictor Importance graph shows the importance of the top 10 inputs (predictors) in the model as a bar chart.

If there are more than 10 fields in the chart, you can change the selection of predictors that are included in the chart by using the slider beneath the chart. The indicator marks on the slider are a fixed width, and each mark on the slider represents 10 fields. You can move the indicator marks along the slider to display the next or previous 10 fields, ordered by predictor importance.

You can double-click the chart to open a separate dialog box in which you can edit the graph settings. For example, you can amend items such as the size of the graph, and the size and color of the fonts used. When you close this separate editing dialog box, the changes are applied to the chart that is displayed in the Output tab.

### Top Decision Rules table

By default, this interactive table displays the statistics of the rules for the top five leaf nodes in the output, based on the percentage of total records that are contained within the leaf node.

You can double-click the table to open a separate dialog box in which you can edit the rule information that is shown in the table. The information that is displayed, and the options that are available in the dialog box, depend on the data type of the target; such as, categorical, or continuous.

The following rule information is shown in the table:

- Rule ID
- The details of how the rule is applied and made up
- Record count for each rule. If you build a model with a frequency weight, this value is the weighted, valid count which represents the records on which the tree is based.
- Record percentage for each rule

In addition, for a continuous target, an extra column in the table shows the **Mean** value for each rule.

You can alter the rule table layout using the following **Table contents** options:

- **Top decision rules** The top five decision rules are sorted by the percentage of total records contained within the leaf nodes.
- **All rules** The table contains all of the leaf nodes produced by the model but only shows 20 rules per page. When you select this layout, you can search for a rule by using the additional options of **Find rule by ID** and **Page**.

In addition, for a categorical target, you can alter the rule table layout using the **Top rules by category** option. The top five decision rules are sorted by the percentage of total records for a **Target category** that you select.

If you change the layout of the rules table, you can copy the modified rules table back to the Output viewer by clicking the Copy to Viewer button at the upper-left of the dialog box.

### Tree-AS model nugget settings

On the Settings tab for a Tree-AS model nugget, you specify options for confidences and for SQL generation during model scoring. This tab is available only after the model nugget is added to a stream.

**Calculate confidences** To include confidences in scoring operations, select this check box. When you score models in the database, excluding confidences means that you can generate more efficient SQL. For regression trees, confidences are not assigned.

**Rule identifier** To add a field in the scoring output that indicates the ID for the terminal node to which each record is assigned, select this check box.

**Generate SQL for this model** When using data from a database, SQL code can be pushed back to the database for execution, providing superior performance for many operations.

Select one of the following options to specify how SQL is generated:

- **Default: Score using Server Scoring Adapter (if installed) otherwise in process** If connected to a database with a scoring adapter installed, generates SQL by using the scoring adapter and associated user-defined functions (UDF) and scores your model within the database. When no scoring adapter is available, this option fetches your data back from the database and scores it in SPSS Modeler.
- **Score outside of the Database** If selected, this option fetches your data back from the database and scores it in SPSS Modeler.

## Random Trees node

---

The Random Trees node can be used with data in a distributed environment. In this node, you build an ensemble model that consists of multiple decision trees.

The Random Trees node is a tree-based classification and prediction method that is built on Classification and Regression Tree methodology. As with C&R Tree, this prediction method uses recursive partitioning to split the training records into segments with similar output field values. The node starts by examining the input fields available to it to find the best split, which is measured by the reduction in an impurity index that results from the split. The split defines two subgroups, each of which is then split into two more subgroups, and so on, until one of the stopping criteria is triggered. All splits are binary (only two subgroups).

The Random Trees node uses bootstrap sampling with replacement to generate sample data. The sample data is used to grow a tree model. During tree growth, Random Trees will not sample the data again. Instead, it randomly selects part of the predictors and uses the best one to split a tree node. This process is repeated when splitting each tree node. This is the basic idea of growing a tree in random forest.

Random Trees uses C&R Tree-like trees. Since such trees are binary, each field for splitting results in two branches. For a categorical field with multiple categories, the categories are grouped into two groups based on the inner splitting criterion. Each tree grows to the largest extent possible (there is no pruning). In scoring, Random Trees combines individual tree scores by majority voting (for classification) or average (for regression).

Random Trees differ from C&R Trees as follows:

- Random Trees nodes randomly select a specified number of predictors and uses the best one from the selection to split a node. In contrast, C&R Tree finds the best one from all predictors.
- Each tree in Random Trees grows fully until each leaf node typically contains a single record. So the tree depth could be very large. But standard C&R Tree uses different stopping rules for tree growth, which usually leads to a much shallower tree.

Random Trees adds two features compared to C&R Tree:

- The first feature is *bagging*, where replicas of the training dataset are created by sampling with replacement from the original dataset. This action creates bootstrap samples that are of equal size to the original dataset, after which a *component model* is built on each replica. Together these component models form an ensemble model.
- The second feature is that, at each split of the tree, only a sampling of the input fields is considered for the impurity measure.

**Requirements.** To train a Random Trees model, you need one or more *Input* fields and one *Target* field. Target and input fields can be continuous (numeric range) or categorical. Fields that are set to either *Both* or *None* are ignored. Fields that are used in the model must have their types fully instantiated, and any ordinal (ordered set) fields that are used in the model must have numeric storage (not string). If necessary, the Reclassify node can be used to convert them.

**Strengths.** Random Trees models are robust when you are dealing with large data sets and numbers of fields. Due to the use of bagging and field sampling, they are much less prone to overfitting and thus the results that are seen in testing are more likely to be repeated when you use new data.

### Random Trees node fields options

On the Fields tab, you choose whether you want to use the field role settings already defined in upstream nodes, or make the field assignments manually.

**Use predefined roles** This option uses the role settings (targets, predictors and so on) from an upstream Type node (or the Types tab of an upstream source node).

**Use custom field assignments.** To manually assign targets, predictors and other roles, select this option.

**Fields.** Use the arrow buttons to assign items manually from this list to the various role fields on the right of the screen. The icons indicate the valid measurement levels for each role field.

To select all the fields in the list, click the **All** button, or click an individual measurement level button to select all fields with that measurement level.

**Target.** Select one field as the target for the prediction.

**Predictors** Select one or more fields as inputs for the prediction.

**Analysis Weight** To use a field as a case weight, specify the field here. Case weights are used to account for differences in variance across levels of the output field. For more information, see [“Using Frequency and Weight Fields”](#) on page 27.

## Random Trees node build options

The Build Options tab is where you set all the options for building the model. You can, of course, just click the **Run** button to build a model with all the default options, but normally you will want to customize the build for your own purposes.

The tab contains several different panes on which you set the customizations that are specific to your model.

### Random Trees node - basics

Specify basic options for how to build the decision tree.

**Number of models to build.** Specify the maximum number of trees that the node can build.

**Sample size.** By default, the size of the bootstrap sample is equal to the original training data. When dealing with large datasets, reducing the sample size can increase performance. It is a ratio from 0 to 1. For example, set the sample size to 0.6 to reduce it to 60% of the original training data size.

**Handle imbalanced data.** If the model's target is a flag outcome (for example, purchase or do not purchase) and the ratio of the desired outcome to non-desired is very small, the data is imbalanced and the bootstrap sampling that is conducted by the model could affect model accuracy. To improve accuracy select this check box; the model then captures a larger proportion of the desired outcome and generates a better model.

**Use weighted sampling for variable selection.** By default, variables for each leaf node are randomly selected with the same probability. To apply weighting to variables and improve the selection process, select this check box. The weight is calculated by the Random Trees node itself. The more important fields (with higher weight) are more likely to be selected as predictors.

**Maximum number of nodes.** Specify the maximum number of leaf nodes that are allowed in individual trees. If the number would be exceeded on the next split, tree growth is stopped before the split occurs.

**Maximum tree depth.** Specify the maximum number of levels *leaf nodes* below the root node; that is, the number of times the sample is split recursively).

**Minimum child node size.** Specify the minimum number of records that must be contained in a child node after the parent node is split. If a child node would contain fewer records than you enter, the parent node will not be split.

**Specify number of predictors to use for splitting.** If you are building split models, set the minimum number of predictors to be used to build each split. This prevents the split from creating excessively small subgroups. If you don't select this option, the default value is  $\lfloor \sqrt{M} \rfloor$  for classification and  $\lfloor M/3 \rfloor$  for regression, where  $M$  is the total number of predictor variables. If this option is selected, the specified number of predictors will be used.

**Note:** The number of predictors for splitting cannot be greater than the total number of predictors in the data.

**Stop building when accuracy can no longer be improved.** Random Trees uses a particular procedure for deciding when to stop training. Specifically, if the improvement of the current ensemble accuracy is smaller than a specified threshold, then it will stop adding new trees. This could result in a model with fewer trees than the value you specified for the **Number of models to build** option.

## Random Trees node - costs

In some contexts, certain kinds of errors are more costly than others. For example, it may be more costly to classify a high-risk credit applicant as low risk (one kind of error) than it is to classify a low-risk applicant as high risk (a different kind of error). Misclassification costs allow you to specify the relative importance of different kinds of prediction errors.

Misclassification costs are basically weights applied to specific outcomes. These weights are factored into the model and may actually change the prediction (as a way of protecting against costly mistakes).

A model that includes costs might not produce fewer errors than one that doesn't, and might not rank any higher in terms of overall accuracy, but it is likely to perform better in practical terms because it has a built-in bias in favor of less expensive errors.

The cost matrix shows the cost for each possible combination of predicted category and actual category. By default, all misclassification costs are set to 1.0. To enter custom cost values, select **Use misclassification costs** and enter your custom values into the cost matrix.

To change a misclassification cost, select the cell corresponding to the desired combination of predicted and actual values, delete the existing contents of the cell, and enter the desired cost for the cell. Costs are not automatically symmetrical. For example, if you set the cost of misclassifying *A* as *B* to be 2.0, the cost of misclassifying *B* as *A* will still have the default value of 1.0 unless you explicitly change it as well.

For ordinal targets only, you can select the **Default cost increase for ordinal target** and set default values in the costs matrix. The available options are described in the following list.

- **No increase** - A default value of 1.0 for every incorrect prediction.
- **Linear** - Each successive incorrect prediction increases the cost by 1.
- **Square** - Each successive incorrect prediction is the square of the linear value. In this case, the values might be: 1, 4, 9, and so on.
- **Custom** - If you manually edit any values in the table, the drop-down option automatically changes to **Custom**. If you change the drop-down selection to any of the other options your edited values are replaced with the values for the selected option.

## Random Trees node - advanced

Specify advanced options for how to build the decision tree.

**Maximum percentage of missing values.** Specify the maximum percentage of missing values allowed in any input. If the percentage exceeds this number, the input is excluded from model building.

**Exclude fields with a single category majority over.** Specify the maximum percentage of records that a single category can have within a field. If any category value represents a higher percentage of records than specified, the entire field is excluded from model building.

**Maximum number of field categories.** Specify the maximum number of categories that can be contained within a field. If the number of categories exceeds this number, the field is excluded from model building.

**Minimum field variation.** If the coefficient of variation of a continuous field is smaller than the value you specify here (in other words, the field is nearly constant), then field is excluded from model building.

**Number of bins.** Specify the number of equal frequency bins to be used for continuous inputs. The available options are: 2, 4, 5, 10, 20, 25, 50, or 100.

**Number of interesting rules to report.** Specify the number of rules to report (minimum of 1, maximum of 1000, with a default of 50).

## Random Trees node model options

On the Model Options tab, you can choose whether to specify a name for the model, or generate a name automatically. You can also choose to calculate the importance of the predictors during scoring of the model.



**Model name.** You can generate the model name automatically based on the target or ID field (or model type in cases where no such field is specified) or specify a custom name.

## Random Trees model nugget

### Random Trees model nugget output

After you create a Random Trees model, the following information is available in the Output viewer:

#### Model information table

The Model information table provides key information about the model. The table always includes the following high-level model settings:

- The name of the target field that is selected in either the Type node or the Random Trees node Fields tab.
- The model building method - Random Trees.
- The number of predictors input into the model.

The additional details that are shown in the table depend on whether you build a classification or regression model, and if the model is built to handle imbalanced data:

- Classification model (default settings)
  - Model accuracy
  - Misclassification rule
- Classification model (**Handle imbalanced data** selected)
  - Gmean
  - True positive rate, which is subdivided into classes.
- Regression model
  - Root mean squared error
  - Relative error
  - Variance explained

#### Records Summary

The summary shows how many records were used to fit the model, and how many were excluded. Both the number of records and the percentage of the whole number are shown. If the model was built to include frequency weight, the unweighted number of records that are included and excluded is also shown.

#### Predictor Importance

The Predictor Importance graph shows the importance of the top 10 inputs (predictors) in the model as a bar chart.

If there are more than 10 fields in the chart, you can change the selection of predictors that are included in the chart by using the slider beneath the chart. The indicator marks on the slider are a fixed width, and each mark on the slider represents 10 fields. You can move the indicator marks along the slider to display the next or previous 10 fields, ordered by predictor importance.

You can double-click the chart to open a separate dialog box in which you can edit the graph size. When you close this separate editing dialog box, the changes are applied to the chart that is displayed in the Output tab.

## Top Decision Rules table

By default, this interactive table displays the statistics of the top rules, which are sorted by interestingness.

You can double-click the table to open a separate dialog box in which you can edit the rule information that is shown in the table. The information that is displayed, and the options that are available in the dialog box, depend on the data type of the target; such as, categorical, or continuous.

The following rule information is shown in the table:

- The details of how the rule is applied and made up
- If the results are in the most frequent category
- Rule accuracy
- Trees accuracy
- Interestingness index

The interestingness index is calculated by using the following formula:

$$I_{index}(t) = P(A(t)) * P(B(t)) * (P(B(t)|A(t)) + P(\bar{B}(t)|\bar{A}(t)))$$

In this formula:

- $P(A(t))$  is the trees accuracy
- $P(B(t))$  is the rule accuracy
- $P(B(t)|A(t))$  represents correct predictions by both the trees and the node
- The remaining piece of the formula represents incorrect predictions by both the trees and the node.

You can alter the rule table layout by using the following **Table contents** options:

- **Top decision rules** The top five decision rules, which are sorted by the interestingness index.
- **All rules** The table contains all of the rules that are produced by the model but shows only 20 rules per page. When you select this layout, you can search for a rule by using the additional options of **Find rule by ID** and **Page**.

In addition, for a categorical target, you can alter the rule table layout by using the **Top rules by category** option. The top five decision rules are sorted by the percentage of total records for a **Target category** that you select.

**Note:** For categorical targets, the table is only available when **Handle imbalanced data** is not selected in the Basics tab of the Build Options.

If you change the layout of the rules table, you can copy the modified rules table back to the Output viewer by clicking the Copy to Viewer button at the upper left of the dialog box.

## Confusion Matrix

For classification models, the confusion matrix shows the number of predicted results versus the actual observed results, including the proportion of correct predictions.

**Note:** The confusion matrix is not available for regression models, nor when **Handle imbalanced data** is selected in the Basics tab of the Build Options.

## Random Trees model nugget settings

On the Settings tab for a Random Trees model nugget, you specify options for confidences and for SQL generation during model scoring. This tab is available only after the model nugget is added to a stream.

**Calculate confidences** To include confidences in scoring operations, select this check box. When you score models in the database, excluding confidences means that you can generate more efficient SQL. For regression trees, confidences are not assigned.

**Generate SQL for this model** When using data from a database, SQL code can be pushed back to the database for execution, providing superior performance for many operations.

Select one of the following options to specify how SQL is generated:

- **Default: Score using Server Scoring Adapter (if installed) otherwise in process** If connected to a database with a scoring adapter installed, generates SQL by using the scoring adapter and associated user-defined functions (UDF) and scores your model within the database. When no scoring adapter is available, this option fetches your data back from the database and scores it in SPSS Modeler.
- **Score outside of the Database** If selected, this option fetches your data back from the database and scores it in SPSS Modeler.

## C&R Tree, CHAID, QUEST, and C5.0 decision tree model nuggets

---

Decision tree model nuggets represent the tree structures for predicting a particular output field discovered by one of the decision tree modeling nodes (C&R Tree, CHAID, QUEST or C5.0). Tree models can be generated directly from the tree-building node, or indirectly from the interactive tree builder. See the topic [“The Interactive Tree Builder”](#) on page 72 for more information.

### Scoring Tree Models

When you run a stream containing a tree model nugget, the specific result depends on the type of tree.

- For classification trees (categorical target), two new fields, containing the predicted value and the confidence for each record, are added to the data. The prediction is based on the most frequent category for the terminal node to which the record is assigned; if a majority of respondents in a given node is yes, the prediction for all records assigned to that node is yes.
- For regression trees, only predicted values are generated; confidences are not assigned.
- Optionally, for CHAID, QUEST, and C&R Tree models, an additional field can be added that indicates the ID for the node to which each record is assigned.

The new field names are derived from the model name by adding prefixes. For C&R Tree, CHAID, and QUEST, the prefixes are *\$R-* for the prediction field, *\$RC-* for the confidence field, and *\$RI-* for the node identifier field. For C5.0 trees, the prefixes are *\$C-* for the prediction field and *\$CC-* for the confidence field. If multiple tree model nodes are present, the new field names will include numbers in the *prefix* to distinguish them if necessary—for example, *\$R1-* and *\$RC1-*, and *\$R2-*.

### Working with Tree Model Nuggets

You can save or export information related to the model in a number of ways.

**Note:** Many of these options are also available from the tree builder window.

From either the tree builder or a tree model nugget, you can:

- Generate a Filter or Select node based on the current tree. See [“Generating Filter and Select Nodes”](#) on page 81 for more information.
- Generate a Rule Set nugget that represents the tree structure as a set of rules defining the terminal branches of the tree. See [“Generating a Rule Set from a Decision Tree”](#) on page 81 for more information.
- In addition, for tree model nuggets only, you can export the model in PMML format. See [“The models palette”](#) on page 34 for more information. If the model includes any custom splits, this information is not preserved in the exported PMML. (The split is preserved, but the fact that it is custom rather than chosen by the algorithm is not.)
- Generate a graph based on a selected part of the current tree. Note that this only works for a nugget when it is attached to other nodes in a stream. See [“Generating Graphs”](#) on page 108 for more information.

- For boosted C5.0 models only, you can choose **Single Decision Tree (Canvas)** or **Single Decision Tree (GM Palette)** to create a new single rule set derived from the currently selected rule. See the topic [“Boosted C5.0 Models”](#) on page 107 for more information.

**Note:** Although the Build Rule node was replaced by the C&R Tree node, decision tree nodes in existing streams that were originally created using a Build Rule node will still function properly.

## Single Tree Model Nuggets

If you select **Build a single tree** as the main objective on the modeling node, the resulting model nugget contains the following tabs.

Table 7. Tabs on single tree nugget		
Tab	Description	Further Information
Model	Displays the rules that define the model.	See the topic <a href="#">“Decision Tree Model Rules”</a> on page 104 for more information.
Viewer	Displays the tree view of the model.	See the topic <a href="#">“Decision Tree Model Viewer”</a> on page 106 for more information.
Summary	Displays information about the fields, build settings, and model estimation process.	See the topic <a href="#">“Model Nugget Summary / Information”</a> on page 36 for more information.
Settings	Enables you to specify options for confidences and for SQL generation during model scoring.	See the topic <a href="#">“Decision Tree/Rule Set model nugget settings”</a> on page 106 for more information.
Annotation	Enables you to add descriptive annotations, specify a custom name, add tooltip text and specify search keywords for the model.	

### Decision Tree Model Rules

The Model tab for a decision tree nugget displays the rules that define the model. Optionally, a graph of predictor importance and a third panel with information about history, frequencies, and surrogates can also be displayed.

**Note:** If you select the **Create a model for very large datasets** option on the CHAID node Build Options tab (Objective panel), the Model tab displays the tree rule details only.

### Tree Rules

The left pane displays a list of conditions defining the partitioning of data discovered by the algorithm—essentially a series of rules that can be used to assign individual records to child nodes based on the values of different predictors.

Decision trees work by recursively partitioning the data based on input field values. The data partitions are called *branches*. The initial branch (sometimes called the *root*) encompasses all data records. The root is split into subsets, or *child branches*, based on the value of a particular input field. Each child branch can be further split into sub-branches, which can in turn be split again, and so on. At the lowest level of the tree are branches that have no more splits. Such branches are known as *terminal branches* (or *leaves*).

## Tree rule details

The rule browser shows the input values that define each partition or branch and a summary of output field values for the records in that split. For general information on using the model browser, see [“Browsing model nuggets” on page 35](#).

For splits based on numeric fields, the branch is shown by a line of the form:

```
fieldname relation value [summary]
```

where *relation* is a numeric relation. For example, a branch defined by values greater than 100 for the *revenue* field would be shown as:

```
revenue > 100 [summary]
```

For splits based on symbolic fields, the branch is shown by a line of the form:

```
fieldname = value [summary] or fieldname in [values] [summary]
```

where *values* represents the field values that define the branch. For example, a branch that includes records where the value of *region* can be *North*, *West*, or *South* would be represented as:

```
region in ["North" "West" "South"] [summary]
```

For terminal branches, a prediction is also given, adding an arrow and the predicted value to the end of the rule condition. For example, a leaf defined by *revenue* > 100 that predicts a value of *high* for the output field would be displayed as:

```
revenue > 100 [Mode: high] → high
```

The *summary* for the branch is defined differently for symbolic and numeric output fields. For trees with numeric output fields, the summary is the *average* value for the branch, and the *effect* of the branch is the difference between the average for the branch and the average of its parent branch. For trees with symbolic output fields, the summary is the *mode*, or the most frequent value, for records in the branch.

To fully describe a branch, you need to include the condition that defines the branch, plus the conditions that define the splits further up the tree. For example, in the tree:

```
revenue > 100
  region = "North"
  region in ["South" "East" "West"]
    revenue <= 200
```

the branch represented by the second line is defined by the conditions *revenue* > 100 and *region* = "North".

If you click **Show Instances/Confidence** on the toolbar, each rule will also show information about the number of records to which the rule applies (*Instances*) and the proportion of those records for which the rule is true (*Confidence*).

## Predictor Importance

Optionally, a chart that indicates the relative importance of each predictor in estimating the model may also be displayed on the Model tab. Typically you will want to focus your modeling efforts on the predictors that matter most and consider dropping or ignoring those that matter least.

**Note:** This chart is only available if **Calculate predictor importance** is selected on the Analyze tab before generating the model. See the topic [“Predictor Importance” on page 37](#) for more information.

## Additional Model Information

If you click **Show Additional Information Panel** on the toolbar, you will see a panel at the bottom of the window displaying detailed information for the selected rule. The information panel contains three tabs.

**History.** This tab traces the split conditions from the root node down to the selected node. This provides a list of conditions that determine when a record is assigned to the selected node. Records for which all of the conditions are true will be assigned to this node.

**Frequencies.** For models with symbolic target fields, this tab shows, for each possible target value, the number of records assigned to this node (in the training data) that have that target value. The frequency figure, expressed as a percentage (shown to a maximum of three decimal places) is also displayed. For models with numeric targets, this tab is empty.

**Surrogates.** Where applicable, any surrogates for the primary split field are shown for the selected node. Surrogates are alternate fields used if the primary predictor value is missing for a given record. The maximum number of surrogates allowed for a given split is specified in the tree-building node, but the actual number depends on the training data. In general, the more missing data, the more surrogates are likely to be used. For other decision tree models, this tab is empty.

**Note:** To be included in the model, surrogates must be identified during the training phase. If the training sample has no missing values, then no surrogates will be identified, and any records with missing values encountered during testing or scoring will automatically fall into the child node with the largest number of records. If missing values are expected during testing or scoring, be sure that values are missing from the training sample, as well. Surrogates are not available for CHAID trees.

## Effect

The effect of a node is the increase or decrease of the average value (predicted value compared to the parent node). For example, if the average of a node is 0.2 and the average of its parent is 0.6, then the effect for the node is  $0.2 - 0.6 = -0.4$ . This statistic only applies for a continuous target.

## Decision Tree Model Viewer

The Viewer tab for a decision tree model nugget resembles the display in the tree builder. The main difference is that when browsing the model nugget, you can not grow or modify the tree. Other options for viewing and customizing the display are similar between the two components. See the topic [“Customizing the Tree View”](#) on page 75 for more information.

**Note:** The Viewer tab is not displayed for CHAID model nuggets built if you select the **Create a model for very large datasets** option on the Build Options tab - Objective panel.

When viewing split rules on the Viewer tab, square brackets mean that the adjacent value is included in the range whereas parentheses indicate that the adjacent value is excluded from the range. The expression (23,37] therefore means from 23 exclusive to 37 inclusive, i.e. from just above 23 to 37. On the Model tab, the same condition would be displayed as:

```
Age > 23 and Age <= 37
```

## Decision Tree/Rule Set model nugget settings

The Settings tab for a decision tree or Rule Set model nugget enables you to specify options for confidences and for SQL generation during model scoring. This tab is available only after the model nugget has been added to a stream.

**Calculate confidences** Select to include confidences in scoring operations. When scoring models in the database, excluding confidences enables you to generate more efficient SQL. For regression trees, confidences are not assigned.

**Note:** If you select the **Create a model for very large datasets** option on the Build Options tab - Method panel for CHAID models, this checkbox is available only in the model nuggets for categorical targets of nominal or flag.

**Calculate raw propensity scores** For models with a flag target (which return a yes or no prediction), you can request propensity scores that indicate the likelihood of the true outcome specified for the target field. These are in addition to other prediction and confidence values that may be generated during scoring.

**Note:** If you select the **Create a model for very large datasets** option on the Build Options tab - Method panel for CHAID models, this checkbox is available only in model nuggets with a categorical target of flag.

**Calculate adjusted propensity scores** Raw propensity scores are based only on the training data and may be overly optimistic due to the tendency of many models to overfit this data. Adjusted propensities attempt to compensate by evaluating model performance against a test or validation partition. This option requires that a partition field be defined in the stream and adjusted propensity scores be enabled in the modeling node before generating the model.

**Note:** Adjusted propensity scores are not available for boosted tree and rule set models. See the topic [“Boosted C5.0 Models”](#) on page 107 for more information.

**Rule identifier** For CHAID, QUEST, and C&R Tree models, this option adds a field in the scoring output that indicates the ID for the terminal node to which each record is assigned.

**Note:** When this option is selected, SQL generation is not available.

**Generate SQL for this model** When using data from a database, SQL code can be pushed back to the database for execution, providing superior performance for many operations.

Select one of the following options to specify how SQL generation is performed.

- **Default: Score using Server Scoring Adapter (if installed) otherwise in process** If connected to a database with a scoring adapter installed, generates SQL using the scoring adapter and associated user defined functions (UDF) and scores your model within the database. When no scoring adapter is available, this option fetches your data back from the database and scores it in SPSS Modeler.
- **Score by converting to native SQL without missing value support** If selected, generates native SQL to score the model within the database, without the overhead of handling missing values. This option simply sets the prediction to null (\$null\$) when a missing value is encountered while scoring a case.

**Note:** This option is not available for CHAID models. For other model types, it is only available for decision trees (not rule sets).

- **Score by converting to native SQL with missing value support** For CHAID, QUEST, and C&R Tree models, you can generate native SQL to score the model within the database with full missing value support. This means that SQL is generated so that missing values are handled as specified in the model. For example, C&R Trees use surrogate rules and biggest child fallback.

**Note:** For C5.0 models, this option is only available for rule sets (not decision trees).

- **Score outside of the Database** If selected, this option fetches your data back from the database and scores it in SPSS Modeler.

## Boosted C5.0 Models

This feature is available in SPSS Modeler Professional and SPSS Modeler Premium.

When you create a boosted C5.0 model (either a rule set or a decision tree), you actually create a set of related models. The model rule browser for a boosted C5.0 model shows the list of models at the top level of the hierarchy, along with the estimated accuracy of each model and the overall accuracy of the ensemble of boosted models. To examine the rules or splits for a particular model, select that model and expand it as you would a rule or branch in a single model.

You can also extract a particular model from the set of boosted models and create a new Rule Set model nugget containing just that model. To create a new rule set from a boosted C5.0 model, select the rule set or tree of interest and choose either **Single Decision Tree (GM Palette)** or **Single Decision Tree (Canvas)** from the Generate menu.

## Generating Graphs

The Tree nodes provide a lot of information; however, it may not always be in an easily accessible format for business users. To provide the data in a way that can be easily incorporated into business reports, presentations, and so on, you can produce graphs of selected data. For example, from either the Model or the Viewer tabs of a model nugget, or from the Viewer tab of an interactive tree, you can generate a graph for a selected part of the tree, thereby only creating a graph for the cases in the selected tree or branch node.

*Note:* You can only generate a graph from a nugget when it is attached to other nodes in a stream.

Generate a graph

The first step is to select the information to be shown on the graph:

- On the Model tab of a nugget, expand the list of conditions and rules in the left pane and select the one in which you are interested.
- On the Viewer tab of a nugget, expand the list of branches and select the node in which you are interested.
- On the Viewer tab of an interactive tree, expand the list of branches and select the node in which you are interested.

*Note:* You cannot select the top node on either Viewer tab.

The way in which you create a graph is the same, regardless of how you select the data to be shown:

1. From the Generate menu, select **Graph (from selection)**; alternatively, on the Viewer tab, click the **Graph (from selection)** button in the bottom left corner. The Graphboard Basic tab is displayed.

*Note:* Only the Basic and Detailed tabs are available when you display the Graphboard in this way.

2. Using either the Basic or Detailed tab settings, specify the details to be displayed on the graph.
3. Click OK to generate the graph.

The graph heading identifies the nodes or rules that were chosen for inclusion.

## Model nuggets for boosting, bagging, and very large datasets

If you select **Enhance model accuracy (boosting)**, **Enhance model stability (bagging)**, or **Create a model for very large datasets** as the main objective on the modeling node, IBM SPSS Modeler builds an ensemble of multiple models. See the topic “Models for ensembles” on page 38 for more information.

The resulting model nugget contains the following tabs. The Model tab provides a number of different views of the model.

Table 8. Tabs available in model nugget			
Tab	View	Description	Further Information
Model	Model Summary	Displays a summary of the ensemble quality and (except for boosted models and continuous targets) diversity, a measure of how much the predictions vary across the different models.	See the topic “Model Summary” on page 38 for more information.
	Predictor Importance	Displays a chart indicating the relative importance of each predictor (input field) in estimating the model.	See the topic “Predictor Importance” on page 39 for more information.



Table 8. Tabs available in model nugget (continued)

Tab	View	Description	Further Information
	Predictor Frequency	Displays a chart showing the relative frequency with which each predictor is used in the set of models.	See the topic <a href="#">“Predictor Frequency”</a> on page 39 for more information.
	Component Model Accuracy	Plots a chart of the predictive accuracy of each of the different models in the ensemble.	
	Component Model Details	Displays information on each of the different models in the ensemble.	See the topic <a href="#">“Component Model Details”</a> on page 39 for more information.
	Information	Displays information about the fields, build settings, and model estimation process.	See the topic <a href="#">“Model Nugget Summary / Information”</a> on page 36 for more information.
Settings		Enables you to include confidences in scoring operations.	See the topic <a href="#">“Decision Tree/Rule Set model nugget settings”</a> on page 106 for more information.
Annotation		Enables you to add descriptive annotations, specify a custom name, add tooltip text and specify search keywords for the model.	

## C&R Tree, CHAID, QUEST, C5.0, and Apriori rule set model nuggets

A Rule Set model nugget represents the rules for predicting a particular output field discovered by the association rule modeling node (Apriori) or by one of the tree-building nodes (C&R Tree, CHAID, QUEST, or C5.0). For association rules, the rule set must be generated from an unrefined Rule nugget. For trees, a rule set can be generated from the interactive tree builder, from a C5.0 model-building node, or from any tree model nugget. Unlike unrefined Rule nuggets, Rule Set nuggets can be placed in streams to generate predictions.

When you run a stream containing a Rule Set nugget, two new fields are added to the stream containing the predicted value and the confidence for each record to the data. The new field names are derived from the model name by adding prefixes. For association rule sets, the prefixes are \$A- for the prediction field and \$AC- for the confidence field. For C5.0 rule sets, the prefixes are \$C- for the prediction field and \$CC- for the confidence field. For C&R Tree rule sets, the prefixes are \$R- for the prediction field and \$RC- for the confidence field. In a stream with multiple Rule Set nuggets in a series predicting the same output field(s), the new field names will include numbers in the *prefix* to distinguish them from each other. The first association Rule Set nugget in the stream will use the usual names, the second node will use names starting with \$A1- and \$AC1-, the third node will use names starting with \$A2- and \$AC2-, and so on.

**How rules are applied.** Rule Sets generated from association rules are unlike other model nuggets because for any particular record, more than one prediction can be generated, and those predictions may not all agree. There are two methods for generating predictions from rule sets.

**Note:** Rule Sets that are generated from decision trees return the same results regardless of which method is used, since the rules derived from a decision tree are mutually exclusive.

- **Voting.** This method attempts to combine the predictions of all of the rules that apply to the record. For each record, all rules are examined and each rule that applies to the record is used to generate a prediction and an associated confidence. The sum of confidence figures for each output value is computed, and the value with the greatest confidence sum is chosen as the final prediction. The confidence for the final prediction is the confidence sum for that value divided by the number of rules that fired for that record.
- **First hit.** This method simply tests the rules in order, and the first rule that applies to the record is the one used to generate the prediction.

The method used can be controlled in the stream options.

**Generating nodes.** The Generate menu enables you to create new nodes based on the rule set.

- **Filter Node** Creates a new Filter node to filter fields that are not used by rules in the rule set.
- **Select Node** Creates a new Select node to select records to which the selected rule applies. The generated node will select records for which all antecedents of the rule are true. This option requires a rule to be selected.
- **Rule Trace Node** Creates a new SuperNode that will compute a field indicating which rule was used to make the prediction for each record. When a rule set is evaluated using the first hit method, this is simply a symbol indicating the first rule that would fire. When the rule set is evaluated using the voting method, this is a more complex string showing the input to the voting mechanism.
- **Single Decision Tree (Canvas) / Single Decision Tree (GM Palette).** Creates a new single Rule Set nugget derived from the currently selected rule. Available only for **boosted** C5.0 models. See the topic [“Boosted C5.0 Models” on page 107](#) for more information.
- **Model to Palette** Returns the model to the models palette. This is useful in situations where a colleague may have sent you a stream containing the model and not the model itself.

**Note:** The Settings and Summary tabs in the Rule Set nugget are identical to those for decision tree models.

## Rule Set Model Tab

The Model tab for a Rule Set nugget displays a list of rules extracted from the data by the algorithm.

Rules are broken down by consequent (predicted category) and are presented in the following format:

```
if antecedent_1
and antecedent_2
...
and antecedent_n
then predicted value
```

where consequent and antecedent\_1 through antecedent\_n are all conditions. The rule is interpreted as "for records where antecedent\_1 through antecedent\_n are all true, consequent is also likely to be true." If you click the **Show Instances/Confidence** button on the toolbar, each rule will also show information on the number of records to which the rule applies--that is, for which the antecedents are true (**Instances**) and the proportion of those records for which the entire rule is true (**Confidence**).

Note that confidence is calculated somewhat differently for C5.0 rule sets. C5.0 uses the following formula for calculating the confidence of a rule:

$$\frac{(1 + \text{number of records where rule is correct})}{(2 + \text{number of records for which the rule's antecedents are true})}$$

This calculation of the confidence estimate adjusts for the process of generalizing rules from a decision tree (which is what C5.0 does when it creates a rule set).

## Importing Projects from AnswerTree 3.0

---

IBM SPSS Modeler can import projects saved in AnswerTree 3.0 or 3.1 using the standard File > Open dialog box, as follows:

1. From the IBM SPSS Modeler menus choose:

**File > Open Stream**

2. From the Files of Type drop-down list, select **AT Project Files (\*.atp, \*.ats)**.

Each imported project is converted into an IBM SPSS Modeler stream with the following nodes:

- One source node that defines the data source used (for example, an IBM SPSS Statistics data file or database source).
- For each tree in the project (there can be several), one Type node is created that defines properties for each field (variable), including type, role (input or predictor field versus output or predicted field), missing values, and other options.
- For each tree in the project, a Partition node is created that partitions the data for a training or test sample, and a tree-building node is created that defines parameters for generating the tree (either a C&R Tree, QUEST, or CHAID node).

3. To view the generated tree(s), run the stream.

### Comments

- Decision trees generated in IBM SPSS Modeler cannot be exported to AnswerTree; the import from AnswerTree to IBM SPSS Modeler is a one-way trip.
- Profits defined in AnswerTree are not preserved when the project is imported into IBM SPSS Modeler.



---

# Chapter 7. Bayesian Network Models

---

## Bayesian Network Node

---

The **Bayesian Network** node enables you to build a probability model by combining observed and recorded evidence with "common-sense" real-world knowledge to establish the likelihood of occurrences by using seemingly unlinked attributes. The node focuses on Tree Augmented Naïve Bayes (TAN) and Markov Blanket networks that are primarily used for classification.

Bayesian networks are used for making predictions in many varied situations; some examples are:

- Selecting loan opportunities with low default risk.
- Estimating when equipment will need service, parts, or replacement, based on sensor input and existing records.
- Resolving customer problems via online troubleshooting tools.
- Diagnosing and troubleshooting cellular telephone networks in real-time.
- Assessing the potential risks and rewards of research-and-development projects in order to focus resources on the best opportunities.

A Bayesian network is a graphical model that displays variables (often referred to as **nodes**) in a dataset and the probabilistic, or conditional, independencies between them. Causal relationships between nodes may be represented by a Bayesian network; however, the links in the network (also known as **arcs**) do not necessarily represent direct cause and effect. For example, a Bayesian network can be used to calculate the probability of a patient having a specific disease, given the presence or absence of certain symptoms and other relevant data, if the probabilistic independencies between symptoms and disease as displayed on the graph hold true. Networks are very robust where information is missing and make the best possible prediction using whatever information is present.

A common, basic, example of a Bayesian network was created by Lauritzen and Spiegelhalter (1988). It is often referred to as the "Asia" model and is a simplified version of a network that may be used to diagnose a doctor's new patients; the direction of the links roughly corresponding to causality. Each node represents a facet that may relate to the patient's condition; for example, "Smoking" indicates that they are a confirmed smoker, and "VisitAsia" shows if they recently visited Asia. Probability relationships are shown by the links between any nodes; for example, smoking increases the chances of the patient developing both bronchitis and lung cancer, whereas age only seems to be associated with the possibility of developing lung cancer. In the same way, abnormalities on an x-ray of the lungs may be caused by either tuberculosis or lung cancer, while the chances of a patient suffering from shortness of breath (dyspnea) are increased if they also suffer from either bronchitis or lung cancer.

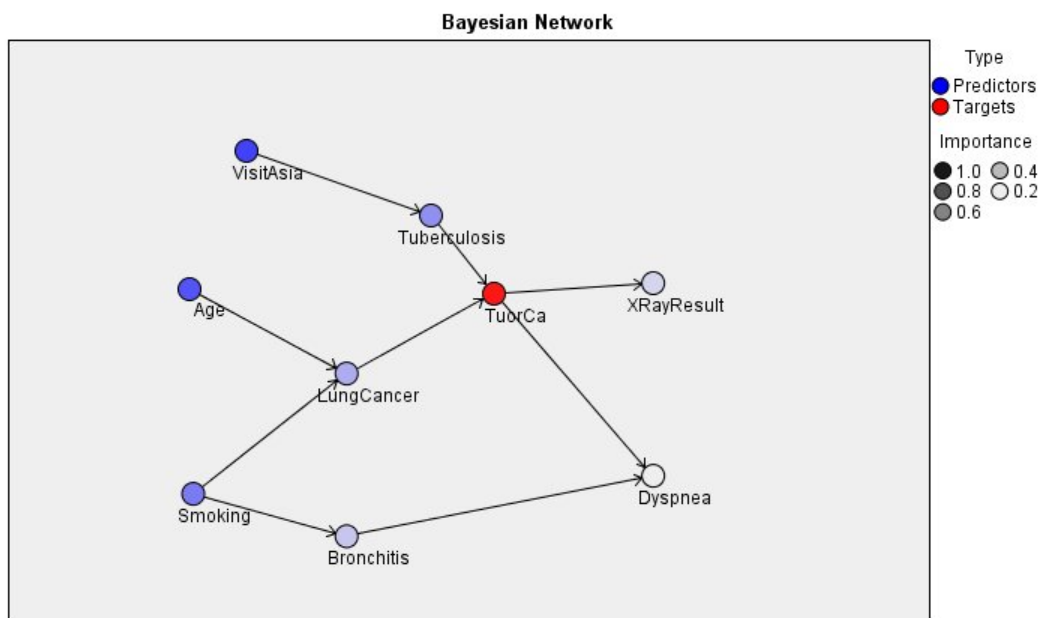


Figure 29. Lauritzen and Spiegelhalter's Asia network example

There are several reasons why you might decide to use a Bayesian network:

- It helps you learn about causal relationships. From this, it enables you to understand a problem area and to predict the consequences of any intervention.
- The network provides an efficient approach for avoiding the overfitting of data.
- A clear visualization of the relationships involved is easily observed.

**Requirements.** Target fields must be categorical and can have a measurement level of *Nominal*, *Ordinal*, or *Flag*. Inputs can be fields of any type. Continuous (numeric range) input fields will be automatically binned; however, if the distribution is skewed, you may obtain better results by manually binning the fields using a Binning node before the Bayesian Network node. For example, use Optimal Binning where the **Supervisor** field is the same as the Bayesian Network node **Target** field.

**Example.** An analyst for a bank wants to be able to predict customers, or potential customers, who are likely to default on their loan repayments. You can use a Bayesian network model to identify the characteristics of customers most likely to default, and build several different types of model to establish which is the best at predicting potential defaulters.

**Example.** A telecommunications operator wants to reduce the number of customers who leave the business (known as "churn"), and update the model on a monthly basis using each preceding month's data. You can use a Bayesian network model to identify the characteristics of customers most likely to churn, and continue training the model each month with the new data.

## Bayesian Network Node Model Options

**Model name.** You can generate the model name automatically based on the target or ID field (or model type in cases where no such field is specified) or specify a custom name.

**Use partitioned data.** If a partition field is defined, this option ensures that data from only the training partition is used to build the model.

**Build model for each split.** Builds a separate model for each possible value of input fields that are specified as split fields. See the topic "[Building Split Models](#)" on page 24 for more information.

**Partition.** This field enables you to specify a field used to partition the data into separate samples for the training, testing, and validation stages of model building. By using one sample to generate the model and a different sample to test it, you can get a good indication of how well the model will generalize to larger datasets that are similar to the current data. If multiple partition fields have been defined by using Type

or Partition nodes, a single partition field must be selected on the Fields tab in each modeling node that uses partitioning. (If only one partition is present, it is automatically used whenever partitioning is enabled.) Also note that to apply the selected partition in your analysis, partitioning must also be enabled in the Model Options tab for the node. (Deselecting this option makes it possible to disable partitioning without changing field settings.)

**Splits.** For split models, select the split field or fields. This is similar to setting the field role to *Split* in a Type node. You can designate only fields with a measurement level of **Flag**, **Nominal**, **Ordinal** or **Continuous** as split fields. Fields chosen as split fields cannot be used as target, input, partition, frequency or weight fields. See the topic [“Building Split Models” on page 24](#) for more information.

**Continue training existing model.** If you select this option, the results shown on the model nugget Model tab are regenerated and updated each time the model is run. For example, you would do this when you have added a new or updated data source to an existing model.

*Note:* This can only update the existing network; it cannot add or remove nodes or connections. Each time you retrain the model the network will be the same shape, only the conditional probabilities and predictor importance will change. If your new data are broadly similar to your old data then this does not matter since you expect the same things to be significant; however, if you want to check or update *what* is significant (as opposed to how significant it is), you will need to build a new model, that is, build a new network

**Structure type.** Select the structure to be used when building the Bayesian network:

- **TAN.** The Tree Augmented Naïve Bayes model (TAN) creates a simple Bayesian network model that is an improvement over the standard Naïve Bayes model. This is because it allows each predictor to depend on another predictor in addition to the target variable, thereby increasing the classification accuracy.
- **Markov Blanket.** This selects the set of nodes in the dataset that contain the target variable’s parents, its children, and its children’s parents. Essentially, a Markov blanket identifies all the variables in the network that are needed to predict the target variable. This method of building a network is considered to be more accurate; however, with large datasets there maybe a processing time-penalty due to the high number of variables involved. To reduce the amount of processing, you can use the **Feature Selection** options on the Expert tab to select the variables that are significantly related to the target variable.

**Include feature selection preprocessing step.** Selecting this box enables you to use the **Feature Selection** options on the Expert tab.

**Parameter learning method.** Bayesian network parameters refer to the conditional probabilities for each node given the values of its parents. There are two possible selections that you can use to control the task of estimating the conditional probability tables between nodes where the values of the parents are known:

- **Maximum likelihood.** Select this box when using a large dataset. This is the default selection.
- **Bayes adjustment for small cell counts.** For smaller datasets there is a danger of overfitting the model, as well as the possibility of a high number of zero-counts. Select this option to alleviate these problems by applying smoothing to reduce the effect of any zero-counts and any unreliable estimate effects.

## Bayesian Network Node Expert Options

The node expert options enable you to fine-tune the model-building process. To access the expert options, set Mode to **Expert** on the Expert tab.

**Missing values.** By default, IBM SPSS Modeler only uses records that have valid values for all fields used in the model. (This is sometimes called **listwise deletion** of missing values.) If you have a lot of missing data, you may find that this approach eliminates too many records, leaving you without enough data to generate a good model. In such cases, you can deselect the **Use only complete records** option. IBM SPSS Modeler then attempts to use as much information as possible to estimate the model, including records where some of the fields have missing values. (This is sometimes called *pairwise deletion* of

missing values.) However, in some situations, using incomplete records in this manner can lead to computational problems in estimating the model.

**Append all probabilities.** Specifies whether probabilities for each category of the output field are added to each record processed by the node. If this option is not selected, the probability of only the predicted category is added.

**Independence test.** A test of independence assesses whether paired observations on two variables are independent of each other. Select the type of test to be used, available options are:

- **Likelihood ratio.** Tests for target-predictor independence by calculating a ratio between the maximum probability of a result under two different hypotheses.
- **Pearson chi-square.** Tests for target-predictor independence by using a null hypothesis that the relative frequencies of occurrence of observed events follow a specified frequency distribution.

Bayesian network models conduct conditional tests of independence where additional variables are used beyond the tested pairs. In addition, the models explore not only the relations between the target and predictors, but also the relations among the predictors themselves

**Note:** The Independence test options are only available if you select either **Include feature selection preprocessing step** or a **Structure type** of Markov Blanket on the Model tab.

**Significance level.** Used in conjunction with the Independence test settings, this enables you to set a cut-off value to be used when conducting the tests. The lower the value, the fewer links remains in the network; the default level is 0.01.

**Note:** This option is only available if you select either **Include feature selection preprocessing step** or a **Structure type** of Markov Blanket on the Model tab.

**Maximal conditioning set size.** The algorithm for creating a Markov Blanket structure uses conditioning sets of increasing size to carry out independence testing and remove unnecessary links from the network. Because tests involving a high number of conditioning variables require more time and memory for processing you can limit the number of variables to be included. This can be especially useful when processing data with strong dependencies among many variables. Note however that the resulting network may contain some superfluous links.

Specify the maximal number of conditioning variables to be used for independence testing. The default setting is 5.

**Note:** This option is only available if you select either **Include feature selection preprocessing step** or a **Structure type** of Markov Blanket on the Model tab.

**Feature selection.** These options enable you to restrict the number of inputs used when processing the model in order to speed up the model building process. This is especially useful when creating a Markov Blanket structure due to the possible large number of potential inputs; it enables you to select the inputs that are significantly related to the target variable.

**Note:** The feature selection options are only available if you select **Include feature selection preprocessing step** on the Model tab.

- **Inputs always selected.** Using the Field Chooser (button to the right of the text field), select the fields from the dataset that are always to be used when building the Bayesian network model. The target field is always selected. Note that the Bayesian network may still drop the items from this list during the model building process if other tests do not consider it significant. So this option simply ensures that the items in the list are used in the model building process itself, not that they will absolutely appear in the resulting Bayesian model.
- **Maximum number of inputs.** Specify the total number of inputs from the dataset to be used when building the Bayesian network model. The highest number you can enter is the total number of inputs in the dataset.

**Note:** If the number of fields selected in **Inputs always selected** exceeds the value of **Maximum number of inputs**, an error message is displayed.



## Bayesian Network Model Nuggets

---

**Note:** If you select **Continue training existing parameters** on the modeling node Model tab, the information that is shown on the model nugget Model tab is updated each time that you regenerate the model.

The model nugget Model tab is split into two panes.

### Left Pane

**Basic** This view contains a network graph of nodes that displays the relationship between the target and its most important predictors, and the relationship between the predictors. The importance of each predictor is shown by the density of its color; a strong color shows an important predictor, and vice versa.

The bin values for nodes that represent a range are displayed in a tooltip when you hover the mouse pointer over the node.

You can use the graph tools in IBM SPSS Modeler to interact, edit, and save the graph. For example, for use in other applications such as MS Word.

**Tip:** If the network contains many nodes, you can click to select a node and drag it to make the graph more legible.

**Distribution** This view displays the conditional probabilities for each node in the network as a mini graph. Hover the mouse pointer over a graph to display its values in a tooltip.

### Right Pane

**Predictor Importance** This displays a chart that indicates the relative importance of each predictor in estimating the model. For more information, see [“Predictor Importance” on page 37](#).

**Conditional Probabilities** When you select a node or mini distribution graph in the left pane the associated conditional probabilities table is displayed in the right pane. This table contains the conditional probability value for each node value and each combination of values in its parent nodes. In addition, it includes the number of records that are observed for each record value and each combination of values in the parent nodes.

## Bayesian Network Model Settings

The Settings tab for a Bayesian Network model nugget specifies options for modifying the built model. For example, you may use the Bayesian Network node to build several different models using the same data and settings, then use this tab in each model to slightly modify the settings to see how that affects the results.

**Note:** This tab is only available after the model nugget has been added to a stream.

**Calculate raw propensity scores.** For models with a flag target (which return a yes or no prediction), you can request propensity scores that indicate the likelihood of the true outcome specified for the target field. These are in addition to other prediction and confidence values that may be generated during scoring.

**Calculate adjusted propensity scores.** Raw propensity scores are based only on the training data and may be overly optimistic due to the tendency of many models to overfit this data. Adjusted propensities attempt to compensate by evaluating model performance against a test or validation partition. This option requires that a partition field be defined in the stream and adjusted propensity scores be enabled in the modeling node before generating the model.

**Append all probabilities** Specifies whether probabilities for each category of the output field are added to each record processed by the node. If this option is not selected, the probability of only the predicted category is added.

The default setting of this check box is determined by the corresponding check box on the Expert tab of the modeling node. See the topic [“Bayesian Network Node Expert Options” on page 115](#) for more information.

**Generate SQL for this model** When using data from a database, SQL code can be pushed back to the database for execution, providing superior performance for many operations.

Select one of the following options to specify how SQL generation is performed.

- **Default: Score using Server Scoring Adapter (if installed) otherwise in process** If connected to a database with a scoring adapter installed, generates SQL using the scoring adapter and associated user defined functions (UDF) and scores your model within the database. When no scoring adapter is available, this option fetches your data back from the database and scores it in SPSS Modeler.
- **Score outside of the Database** If selected, this option fetches your data back from the database and scores it in SPSS Modeler.

## Bayesian Network Model Summary

The Summary tab of a model nugget displays information about the model itself (*Analysis*), fields used in the model (*Fields*), settings used when building the model (*Build Settings*), and model training (*Training Summary*).

When you first browse the node, the Summary tab results are collapsed. To see the results of interest, use the expander control to the left of an item to unfold it or click the **Expand All** button to show all results. To hide the results when you have finished viewing them, use the expander control to collapse the specific results that you want to hide or click the **Collapse All** button to collapse all results.

**Analysis.** Displays information about the specific model.

**Fields.** Lists the fields used as the target and the inputs in building the model.

**Build Settings.** Contains information about the settings used in building the model.

**Training Summary.** Shows the type of model, the stream used to create it, the user who created it, when it was built, and the elapsed time for building the model.

---

## Chapter 8. Neural networks

A **neural network** can approximate a wide range of predictive models with minimal demands on model structure and assumption. The form of the relationships is determined during the learning process. If a linear relationship between the target and predictors is appropriate, the results of the neural network should closely approximate those of a traditional linear model. If a nonlinear relationship is more appropriate, the neural network will automatically approximate the "correct" model structure.

The trade-off for this flexibility is that the neural network is not easily interpretable. If you are trying to explain an underlying process that produces the relationships between the target and predictors, it would be better to use a more traditional statistical model. However, if model interpretability is not important, you can obtain good predictions using a neural network.

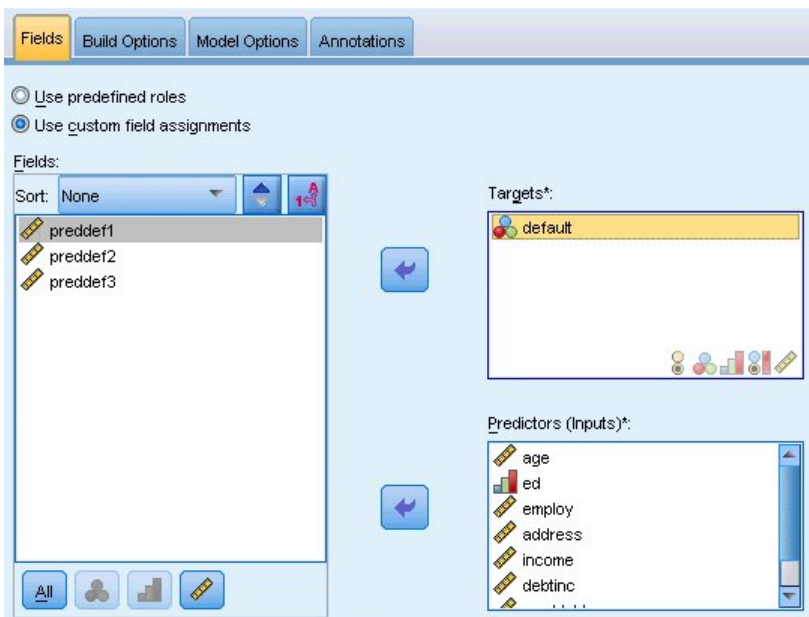


Figure 30. Fields tab

**Field requirements.** There must be at least one Target and one Input. Fields set to Both or None are ignored. There are no measurement level restrictions on targets or predictors (inputs). See [“Modeling Node Fields Options”](#) on page 26 for more information.

The initial weights assigned to neural networks during model building, and therefore the final models produced, depend on the order of the fields in the data. SPSS Modeler automatically sorts data by field name before presenting it to the neural network for training. This means that explicitly changing the order of the fields in the data upstream will not affect the generated neural net models when a random seed is set in the model builder. However, changing the input field names in a way that changes their sort order will produce different neural network models, even with a random seed set in the model builder. The model quality will not be affected significantly given different sort order of field names.

---

### The neural networks model

Neural networks are simple models of the way the nervous system operates. The basic units are **neurons**, which are typically organized into **layers**, as shown in the following figure.

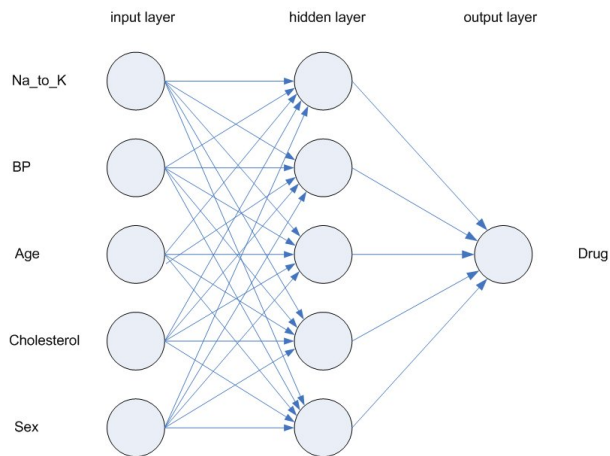


Figure 31. Structure of a neural network

A **neural network** is a simplified model of the way the human brain processes information. It works by simulating a large number of interconnected processing units that resemble abstract versions of neurons.

The processing units are arranged in layers. There are typically three parts in a neural network: an **input layer**, with units representing the input fields; one or more **hidden layers**; and an **output layer**, with a unit or units representing the target field(s). The units are connected with varying connection strengths (or **weights**). Input data are presented to the first layer, and values are propagated from each neuron to every neuron in the next layer. Eventually, a result is delivered from the output layer.

The network learns by examining individual records, generating a prediction for each record, and making adjustments to the weights whenever it makes an incorrect prediction. This process is repeated many times, and the network continues to improve its predictions until one or more of the stopping criteria have been met.

Initially, all weights are random, and the answers that come out of the net are probably nonsensical. The network learns through **training**. Examples for which the output is known are repeatedly presented to the network, and the answers it gives are compared to the known outcomes. Information from this comparison is passed back through the network, gradually changing the weights. As training progresses, the network becomes increasingly accurate in replicating the known outcomes. Once trained, the network can be applied to future cases where the outcome is unknown.

## Using neural networks with legacy streams

Version 14 of IBM SPSS Modeler introduced a new Neural Net node, supporting boosting and bagging techniques and optimization for very large datasets. Existing streams containing the old node may still build and score models in later releases. However, this support will be removed in a future release, so we recommend using the new version.

From version 13 onwards, fields with unknown values (that is, values not present in the training data) are no longer automatically treated as missing values, and are scored with the value \$null\$. Thus if you want to score fields with unknown values as non-null using an older (pre-13) Neural Net model in version 13 or later, you should mark unknown values as missing values (for example, by means of the Type node).

Note that, for compatibility, any legacy streams that still contain the old node may still be using the *Limit set size* option from **Tools > Stream Properties > Options**; this option only applies to Kohonen nets and K-Means nodes from version 14 onwards.

## Objectives

Fields Build Options Model Options Annotations

Select an item:

Objectives

Basics

Stopping Rules

Ensembles

Advanced

What do you want to do?

☒ Build new model

☐ Continue training existing model

What is your main objective?

☒ Create a standard model

☐ Enhance model accuracy (boosting)

☐ Enhance model stability (bagging)

☐ Optimize for very large datasets (requires Server)

Description

Creates a single, standard model to explain relationships between fields. Standard models are easier to interpret and can be faster to score than boosted, bagged, or large dataset ensembles. A standard model is always used for multiple targets.

Figure 32. Objectives settings

### What do you want to do?

- **Build a new model.** Build a completely new model. This is the usual operation of the node.
- **Continue training an existing model.** Training continues with the last model successfully produced by the node. This makes it possible to update or refresh an existing model without having to access the original data and may result in significantly faster performance since only the new or updated records are fed into the stream. Details on the previous model are stored with the modeling node, making it possible to use this option even if the previous model nugget is no longer available in the stream or Models palette.

**Note:** When this option is enabled, all other controls on the Fields and Build Options tabs are disabled.

### What is your main objective? Select the appropriate objective.

- **Create a standard model.** The method builds a single model to predict the target using the predictors. Generally speaking, standard models are easier to interpret and can be faster to score than boosted, bagged, or large dataset ensembles.

**Note:** For split models, to use this option with **Continue training an existing model** you must be connected to Analytic Server.

- **Enhance model accuracy (boosting).** The method builds an ensemble model using boosting, which generates a sequence of models to obtain more accurate predictions. Ensembles can take longer to build and to score than a standard model.

Boosting produces a succession of "component models", each of which is built on the entire dataset. Prior to building each successive component model, the records are weighted based on the previous component model's residuals. Cases with large residuals are given relatively higher analysis weights so that the next component model will focus on predicting these records well. Together these component models form an ensemble model. The ensemble model scores new records using a combining rule; the available rules depend upon the measurement level of the target.

- **Enhance model stability (bagging).** The method builds an ensemble model using bagging (bootstrap aggregating), which generates multiple models to obtain more reliable predictions. Ensembles can take longer to build and to score than a standard model.

Bootstrap aggregation (bagging) produces replicates of the training dataset by sampling with replacement from the original dataset. This creates bootstrap samples of equal size to the original dataset. Then a "component model" is built on each replicate. Together these component models form an ensemble model. The ensemble model scores new records using a combining rule; the available rules depend upon the measurement level of the target.

- **Create a model for very large datasets.** The method builds an ensemble model by splitting the dataset into separate data blocks. Choose this option if your dataset is too large to build any of the models above, or for incremental model building. This option can take less time to build, but can take longer to score than a standard model.

When there are multiple targets, this method will only create a standard model, regardless of the selected objective.

## Basics

Figure 33. Basics settings

**Neural network model.** The type of model determines how the network connects the predictors to the targets through the hidden layer(s). The **multilayer perceptron (MLP)** allows for more complex relationships at the possible cost of increasing the training and scoring time. The **radial basis function (RBF)** may have lower training and scoring times, at the possible cost of reduced predictive power compared to the MLP.

**Hidden Layers.** The hidden layer(s) of a neural network contains unobservable units. The value of each hidden unit is some function of the predictors; the exact form of the function depends in part upon the network type. A multilayer perceptron can have one or two hidden layers; a radial basis function network can have one hidden layer.

- **Automatically compute number of units.** This option builds a network with one hidden layer and computes the "best" number of units in the hidden layer.
- **Customize number of units.** This option allows you to specify the number of units in each hidden layer. The first hidden layer must have at least one unit. Specifying 0 units for the second hidden layer builds a multilayer perceptron with a single hidden layer.

**Note:** You should choose values so that the number of nodes does not exceed the number of continuous predictors plus the total number of categories across all categorical (flag, nominal, and ordinal) predictors.

## Stopping rules

The screenshot shows a software window with four tabs: 'Fields', 'Build Options', 'Model Options', and 'Annotations'. The 'Build Options' tab is selected. On the left, a sidebar lists 'Objectives', 'Basics', 'Stopping Rules' (which is highlighted), 'Ensembles', and 'Advanced'. The main area displays an information icon and text: 'Stopping rules are available for the Multilayer Perceptron model and apply to training the standard model (or component models if ensembles are in effect). Other stopping rules apply.' Below this, there are three settings: 1) 'Use maximum training time (per component model)' is checked, with a 'Minutes' spinner set to 15. 2) 'Customize number of maximum training cycles' is unchecked, with a 'Maximum number of cycles' spinner set to 100. 3) 'Use minimum accuracy' is unchecked, with an 'Accuracy (%)' spinner set to 90.0.

Figure 34. Stopping Rules settings

These are the rules that determine when to stop training multilayer perceptron networks; these settings are ignored when the radial basis function algorithm is used. Training proceeds through at least one cycle (data pass), and can then be stopped according to the following criteria.

**Use maximum training time (per component model).** Choose whether to specify a maximum number of minutes for the algorithm to run. Specify a number greater than 0. When an ensemble model is built, this is the training time allowed for each component model of the ensemble. Note that training may go a bit beyond the specified time limit in order to complete the current cycle.

**Customize number of maximum training cycles.** The maximum number of training cycles allowed. If the maximum number of cycles is exceeded, then training stops. Specify an integer greater than 0.

**Use minimum accuracy.** With this option, training will continue until the specified accuracy is attained. This may never happen, but you can interrupt training at any point and save the net with the best accuracy achieved so far.

The training algorithm will also stop if the error in the overfit prevention set does not decrease after each cycle, if the relative change in the training error is small, or if the ratio of the current training error is small compared to the initial error.

## Ensembles

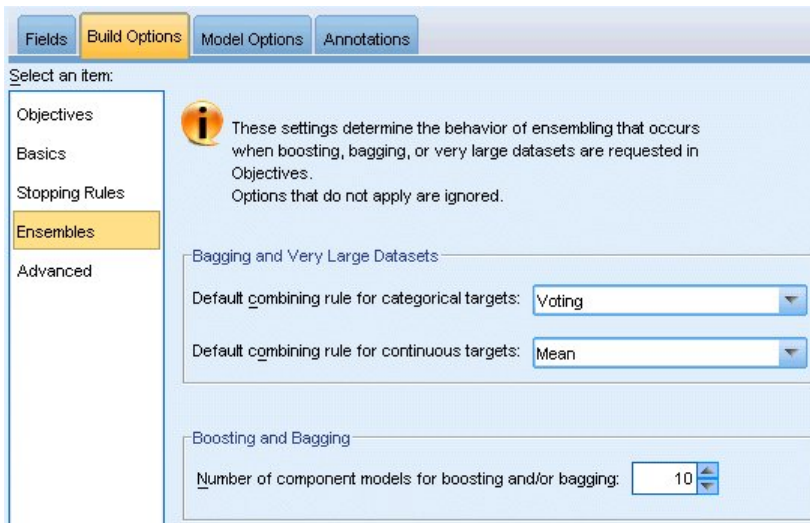


Figure 35. Ensembles settings

These settings determine the behavior of ensembling that occurs when boosting, bagging, or very large datasets are requested in Objectives. Options that do not apply to the selected objective are ignored.

**Bagging and Very Large Datasets.** When scoring an ensemble, this is the rule used to combine the predicted values from the base models to compute the ensemble score value.

- **Default combining rule for categorical targets.** Ensemble predicted values for categorical targets can be combined using voting, highest probability, or highest mean probability. **Voting** selects the category that has the highest probability most often across the base models. **Highest probability** selects the category that achieves the single highest probability across all base models. **Highest mean probability** selects the category with the highest value when the category probabilities are averaged across base models.
- **Default combining rule for continuous targets.** Ensemble predicted values for continuous targets can be combined using the mean or median of the predicted values from the base models.

Note that when the objective is to enhance model accuracy, the combining rule selections are ignored. Boosting always uses a weighted majority vote to score categorical targets and a weighted median to score continuous targets.

**Boosting and Bagging.** Specify the number of base models to build when the objective is to enhance model accuracy or stability; for bagging, this is the number of bootstrap samples. It should be a positive integer.



Fields Build Options Model Options Annotations

Select an item:

- Objectives
- Basics
- Stopping Rules
- Ensembles
- Advanced

Neural Network modeling internally separates records into a model building set and an overfit prevention set. Specify a percentage of records for the overfit prevention set.

Overfit prevention set (%): 30

☒ Replicate Results

Generate

Random seed: 229176228

Missing values in predictors:

- ☒ Delete listwise
- ☐ Impute missing values

Figure 36. Advanced settings

Advanced settings provide control over options that do not fit neatly into other groups of settings.

**Overfit prevention set.** The neural network method internally separates records into a model building set and an overfit prevention set, which is an independent set of data records used to track errors during training in order to prevent the method from modeling chance variation in the data. Specify a percentage of records. The default is 30.

**Replicate results.** Setting a random seed allows you to replicate analyses. Specify an integer or click **Generate**, which will create a pseudo-random integer between 1 and 2147483647, inclusive. By default, analyses are replicated with seed 229176228.

**Missing values in predictors.** This specifies how to treat missing values. **Delete listwise** removes records with missing values on predictors from model building. **Impute missing values** will replace missing values in predictors and use those records in the analysis. Continuous fields impute the average of the minimum and maximum observed values; categorical fields impute the most frequently occurring category. Note that records with missing values on any other field specified on the Fields tab are always removed from model building.

## Model Options

Fields Build Options **Model Options** Annotations

Model Name: ☒ Automatic ☐ Custom

Make Available for Scoring

Predicted value and confidence are always available for scoring.

Confidence is based on:

☒ The probability of the predicted value

☐ The increase in probability from the next most likely value

☒ Predicted probability for categorical targets

Maximum categories to save:

☒ Propensity scores for flag targets

Figure 37. Model Options tab

**Model Name.** You can generate the model name automatically based on the target fields or specify a custom name. The automatically generated name is the target field name. If there are multiple targets, then the model name is the field names in order, connected by ampersands. For example, if *field1 field2 field3* are targets, then the model name is: *field1 & field2 & field3*.

**Make Available for Scoring.** When the model is scored, the selected items in this group should be produced. The predicted value (for all targets) and confidence (for categorical targets) are always computed when the model is scored. The computed confidence can be based on the probability of the predicted value (the highest predicted probability) or the difference between the highest predicted probability and the second highest predicted probability.

- **Predicted probability for categorical targets.** This produces the predicted probabilities for categorical targets. A field is created for each category.
- **Propensity scores for flag targets.** For models with a flag target (which return a yes or no prediction), you can request propensity scores that indicate the likelihood of the true outcome specified for the target field. The model produces raw propensity scores; if partitions are in effect, the model also produces adjusted propensity scores based on the testing partition.

## Model Summary

Model Summary	
Target	Previously defaulted
Model	Multilayer Perceptron
Stopping Rule Used	Error cannot be further decreased
Hidden Layer 1 Neurons	4

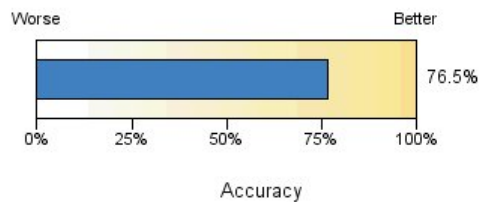


Figure 38. Neural Networks Model Summary view

The Model Summary view is a snapshot, at-a-glance summary of the neural network predictive or classification accuracy.

**Model summary.** The table identifies the target, the type of neural network trained, the stopping rule that stopped training (shown if a multilayer perceptron network was trained), and the number of neurons in each hidden layer of the network.

**Neural Network Quality.** The chart displays the accuracy of the final model, which is presented in larger is better format. For a categorical target, this is simply the percentage of records for which the predicted value matches the observed value. For a continuous target, the accuracy is given as the  $R^2$  value.

**Multiple targets.** If there are multiple targets, then each target is displayed in the **Target** row of the table. The accuracy displayed in the chart is the average of the individual target accuracies.

## Predictor Importance

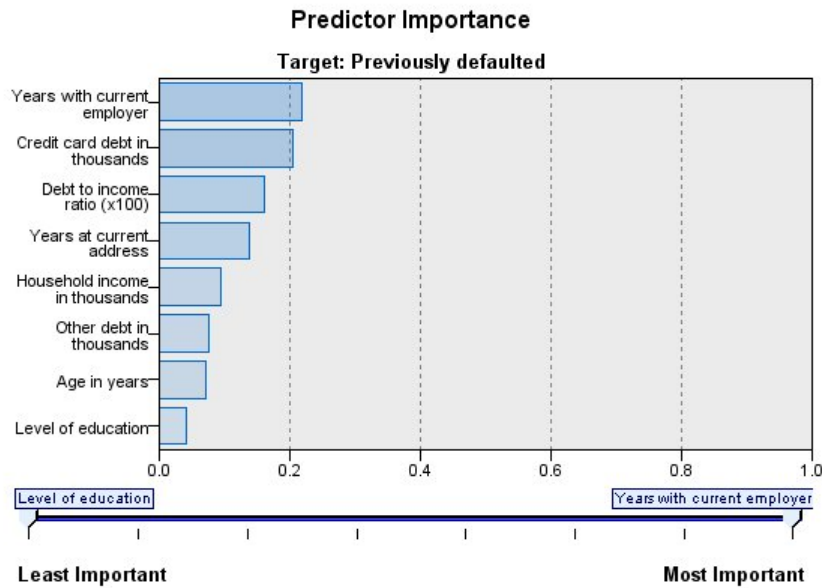
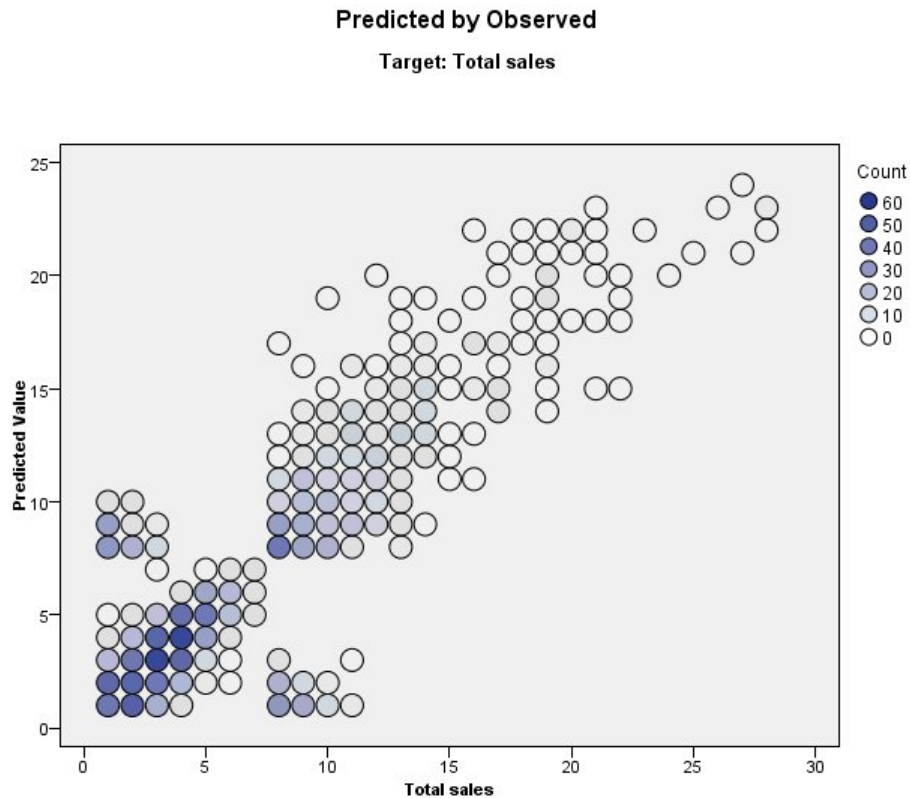


Figure 39. Predictor Importance view

Typically, you will want to focus your modeling efforts on the predictor fields that matter most and consider dropping or ignoring those that matter least. The predictor importance chart helps you do this by indicating the relative importance of each predictor in estimating the model. Since the values are relative, the sum of the values for all predictors on the display is 1.0. Predictor importance does not relate to model accuracy. It just relates to the importance of each predictor in making a prediction, not whether or not the prediction is accurate.

**Multiple targets.** If there are multiple targets, then each target is displayed in a separate chart and there is a **Target** drop-down that controls which target to display.

## Predicted By Observed



Target:

Figure 40. Predicted By Observed view

For continuous targets, this displays a binned scatterplot of the predicted values on the vertical axis by the observed values on the horizontal axis.

**Multiple targets.** If there are multiple continuous targets, then each target is displayed in a separate chart and there is a **Target** drop-down that controls which target to display.

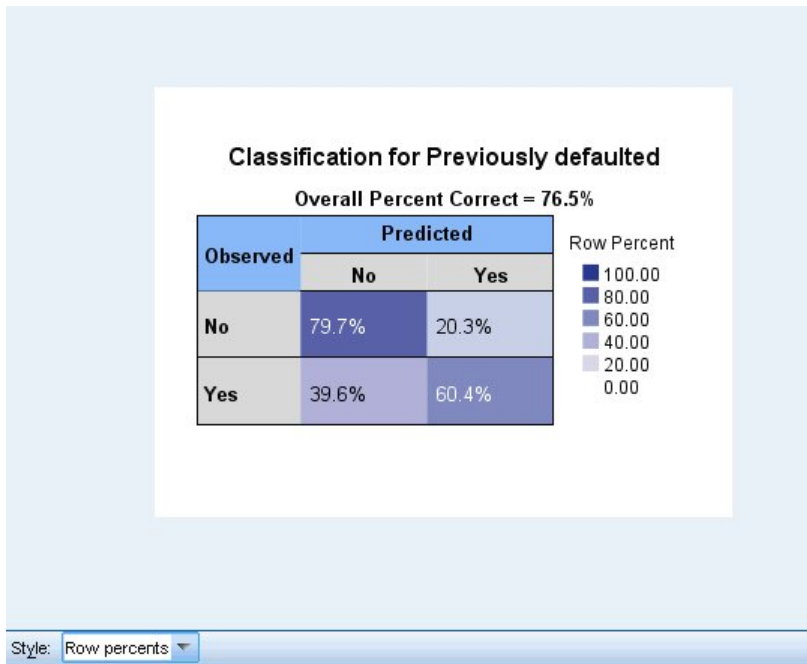


Figure 41. Classification view, row percents style

For categorical targets, this displays the cross-classification of observed versus predicted values in a heat map, plus the overall percent correct.

**Table styles.** There are several different display styles, which are accessible from the **Style** dropdown list.

- **Row percents.** This displays the row percentages (the cell counts expressed as a percent of the row totals) in the cells. This is the default.
- **Cell counts.** This displays the cell counts in the cells. The shading for the heat map is still based on the row percentages.
- **Heat map.** This displays no values in the cells, just the shading.
- **Compressed.** This displays no row or column headings, or values in the cells. It can be useful when the target has a lot of categories.

**Missing.** If any records have missing values on the target, they are displayed in a **(Missing)** row under all valid rows. Records with missing values do not contribute to the overall percent correct.

**Multiple targets.** If there are multiple categorical targets, then each target is displayed in a separate table and there is a **Target** dropdown list that controls which target to display.

**Large tables.** If the displayed target has more than 100 categories, no table is displayed.

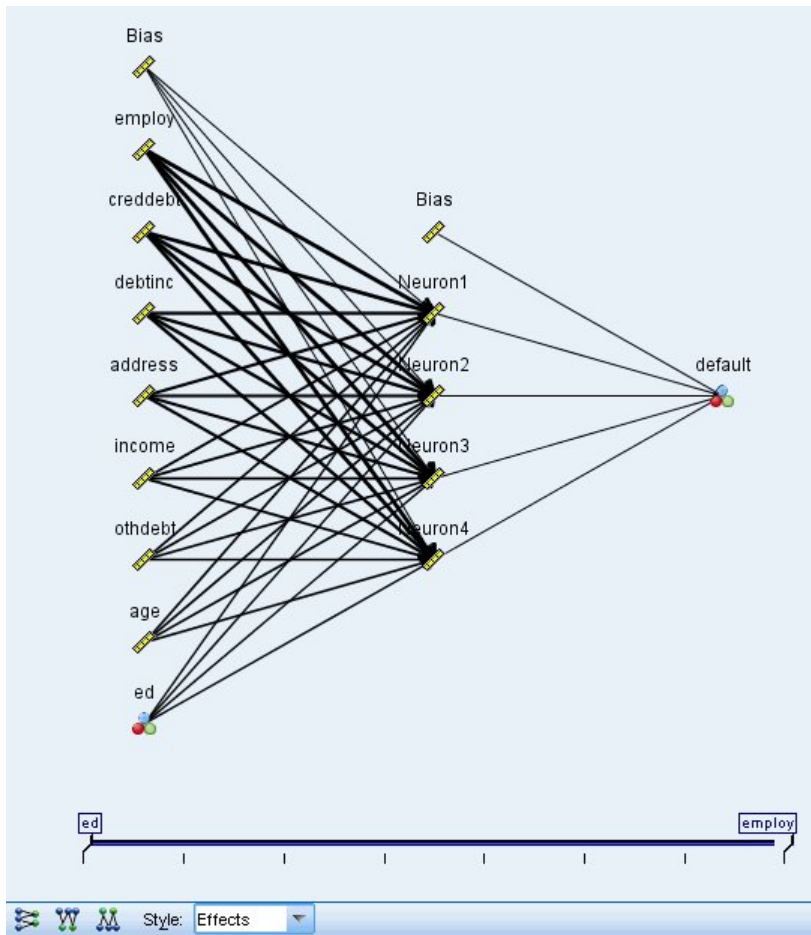


Figure 42. Network view, inputs on the left, effects style

This displays a graphical representation of the neural network.

**Chart styles.** There are two different display styles, which are accessible from the **Style** drop-down.

- **Effects.** This displays each predictor and target as one node in the diagram irrespective of whether the measurement scale is continuous or categorical. This is the default.
- **Coefficients.** This displays multiple indicator nodes for categorical predictors and targets. The connecting lines in the coefficients-style diagram are colored based on the estimated value of the synaptic weight.

**Diagram orientation.** By default, the network diagram is arranged with the inputs on the left and the targets on the right. Using toolbar controls, you can change the orientation so that inputs are on top and targets on the bottom, or inputs on the bottom and targets on top.

**Predictor importance.** Connecting lines in the diagram are weighted based on predictor importance, with greater line width corresponding to greater importance. There is a Predictor Importance slider in the toolbar that controls which predictors are shown in the network diagram. This does not change the model, but simply allows you to focus on the most important predictors.

**Multiple targets.** If there are multiple targets, all targets are displayed in the chart.

## Settings

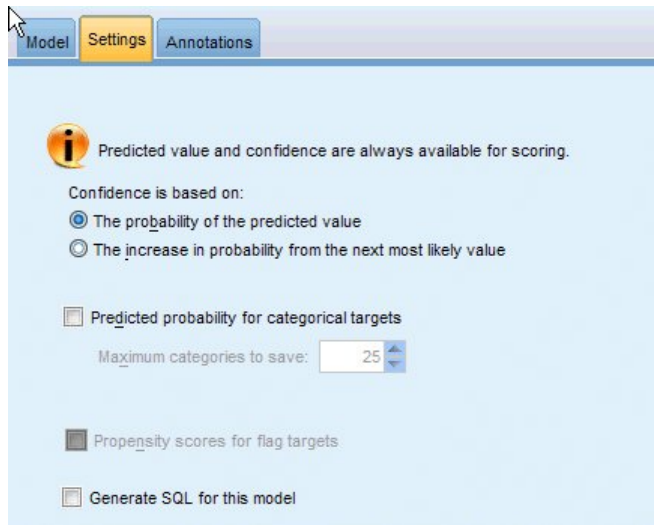


Figure 43. Settings tab

When the model is scored, the selected items in this tab should be produced. The predicted value (for all targets) and confidence (for categorical targets) are always computed when the model is scored. The computed confidence can be based on the probability of the predicted value (the highest predicted probability) or the difference between the highest predicted probability and the second highest predicted probability.

- **Predicted probability for categorical targets.** This produces the predicted probabilities for categorical targets. A field is created for each category.
- **Propensity scores for flag targets.** For models with a flag target (which return a yes or no prediction), you can request propensity scores that indicate the likelihood of the true outcome specified for the target field. The model produces raw propensity scores; if partitions are in effect, the model also produces adjusted propensity scores based on the testing partition.

**Generate SQL for this model** When using data from a database, SQL code can be pushed back to the database for execution, providing superior performance for many operations.

**Default: Score using Server Scoring Adapter (if installed) otherwise in process** If connected to a database with a scoring adapter installed, generates SQL using the scoring adapter and associated user defined functions (UDF) and scores your model within the database. When no scoring adapter is available, this option fetches your data back from the database and scores it in SPSS Modeler.

**Score by converting to native SQL** If selected, generates native SQL to score the model within the database.

**Note:** Although this option can provide quicker results, the size and complexity of the native SQL increases as the complexity of the model increases.

**Score outside of the Database** If selected, this option fetches your data back from the database and scores it in SPSS Modeler.



---

## Chapter 9. Decision List

Decision List models identify subgroups or **segments** that show a higher or lower likelihood of a binary (yes or no) outcome relative to the overall sample. For example, you might look for customers who are least likely to churn or most likely to say yes to a particular offer or campaign. The Decision List Viewer gives you complete control over the model, enabling you to edit segments, add your own business rules, specify how each segment is scored, and customize the model in a number of other ways to optimize the proportion of hits across all segments. As such, it is particularly well-suited for generating mailing lists or otherwise identifying which records to target for a particular campaign. You can also use multiple **mining tasks** to combine modeling approaches—for example, by identifying high- and low-performing segments within the same model and including or excluding each in the scoring stage as appropriate.

### Segments, Rules, and Conditions

A model consists of a list of segments, each of which is defined by a rule that selects matching records. A given rule may have multiple conditions; for example:

```
RFM_SCORE > 10 and  
MONTHS_CURRENT <= 9
```

Rules are applied in the order listed, with the first matching rule determining the outcome for a given record. Taken independently, rules or conditions may overlap, but the order of rules resolves ambiguity. If no rule matches, the record is assigned to the remainder rule.

### Complete Control over Scoring

The Decision List Viewer enables you to view, modify, and reorganize segments and to choose which to include or exclude for purposes of scoring. For example, you can choose to exclude one group of customers from future offers and include others and immediately see how this affects your overall hit rate. Decision List models return a score of Yes for included segments and *\$null\$* for everything else, including the remainder. This direct control over scoring makes Decision List models ideal for generating mailing lists, and they are widely used in customer relationship management, including call center or marketing applications.

### Mining Tasks, Measures, and Selections

The modeling process is driven by **mining tasks**. Each mining task effectively initiates a new modeling run and returns a new set of alternative models to choose from. The default task is based on your initial specifications in the Decision List node, but you can define any number of custom tasks. You can also apply tasks iteratively—for example, you can run a high probability search on the entire training set and then run a low probability search on the remainder to weed out low-performing segments.

### Data Selections

You can define data selections and custom model measures for model building and evaluation. For example, you can specify a data selection in a mining task to tailor the model to a specific region and create a custom measure to evaluate how well that model performs on the whole country. Unlike mining tasks, measures don't change the underlying model but provide another lens to assess how well it performs.

### Adding Your Business Knowledge

By fine-tuning or extending the segments identified by the algorithm, the Decision List Viewer enables you to incorporate your business knowledge right into the model. You can edit the segments generated by the model or add additional segments based on rules that you specify. You can then apply the changes and preview the results.

For further insight, a dynamic link with Excel enables you to export your data to Excel, where it can be used to create presentation charts and to calculate custom measures, such as complex profit and ROI, which can be viewed in the Decision List Viewer while you are building the model.

**Example.** The marketing department of a financial institution wants to achieve more profitable results in future campaigns by matching the right offer to each customer. You can use a Decision List model to identify the characteristics of customers most likely to respond favorably based on previous promotions and to generate a mailing list based on the results.

**Requirements.** A single categorical target field with a measurement level of type *Flag* or *Nominal* that indicates the binary outcome you want to predict (yes/no), and at least one input field. When the target field type is *Nominal*, you must manually choose a single value to be treated as a **hit**, or **response**; all the other values are lumped together as **not hit**. An optional frequency field may also be specified. Continuous date/time fields are ignored. Continuous numeric range inputs are automatically binned by the algorithm as specified on the Expert tab in the modeling node. For finer control over binning, add an upstream binning node and use the binned field as input with a measurement level of *Ordinal*.

## Decision List Model Options

---

**Model name.** You can generate the model name automatically based on the target or ID field (or model type in cases where no such field is specified) or specify a custom name.

**Use partitioned data.** If a partition field is defined, this option ensures that data from only the training partition is used to build the model.

**Create split models.** Builds a separate model for each possible value of input fields that are specified as split fields. See [“Building Split Models” on page 24](#) for more information.

**Mode.** Specifies the method used to build the model.

- **Generate model.** Automatically generates a model on the models palette when the node is executed. The resulting model can be added to streams for purposes of scoring but cannot be further edited.
- **Launch interactive session.** Opens the interactive Decision List Viewer modeling (output) window, enabling you to pick from multiple alternatives and repeatedly apply the algorithm with different settings to progressively grow or modify the model. See the topic [“Decision List Viewer” on page 136](#) for more information.
- **Use saved interactive session information.** Launches an interactive session using previously saved settings. Interactive settings can be saved from the Decision List Viewer using the Generate menu (to create a model or modeling node) or the File menu (to update the node from which the session was launched).

**Target value.** Specifies the value of the target field that indicates the outcome you want to model. For example, if the target field churn is coded 0 = no and 1 = yes, specify 1 to identify rules that indicate which records are likely to churn.

**Find segments with.** Indicates whether the search for the target variable should look for a **High probability** or **Low probability** of occurrence. Finding and excluding them can be a useful way to improve your model and can be particularly useful when the remainder has a low probability.

**Maximum number of segments.** Specifies the maximum number of segments to return. The top *N* segments are created, where the best segment is the one with the highest probability or, if more than one model has the same probability, the highest coverage. The minimum allowed setting is 1; there is no maximum setting.

**Minimum segment size.** The two settings below dictate the minimum segment size. The larger of the two values takes precedence. For example, if the percentage value equates to a number higher than the absolute value, the percentage setting takes precedence.

- **As percentage of previous segment (%).** Specifies the minimum group size as a percentage of records. The minimum allowed setting is 0; the maximum allowed setting is 99.9.
- **As absolute value (N).** Specifies the minimum group size as an absolute number of records. The minimum allowed setting is 1; there is no maximum setting.

**Segment rules.**

**Maximum number of attributes.** Specifies the maximum number of conditions per segment rule. The minimum allowed setting is 1; there is no maximum setting.

- **Allow attribute re-use.** When enabled, each cycle can consider all attributes, even those that have been used in previous cycles. The conditions for a segment are built up in cycles, where each cycle adds a new condition. The number of cycles is defined using the **Maximum number of attributes** setting.

**Confidence interval for new conditions (%).** Specifies the confidence level for testing segment significance. This setting plays a significant role in the number of segments (if any) that are returned as well as the number-of-conditions-per-segment rule. The higher the value, the smaller the returned result set. The minimum allowed setting is 50; the maximum allowed setting is 99.9.

## Decision List Node Expert Options

---

Expert options enable you to fine-tune the model-building process.

**Binning method.** The method used for binning continuous fields (equal count or equal width).

**Number of bins.** The number of bins to create for continuous fields. The minimum allowed setting is 2; there is no maximum setting.

**Model search width.** The maximum number of model results per cycle that can be used for the next cycle. The minimum allowed setting is 1; there is no maximum setting.

**Rule search width.** The maximum number of rule results per cycle that can be used for the next cycle. The minimum allowed setting is 1; there is no maximum setting.

**Bin merging factor.** The minimum amount by which a segment must grow when merged with its neighbor. The minimum allowed setting is 1.01; there is no maximum setting.

- **Allow missing values in conditions.** True to allow the IS MISSING test in rules.
- **Discard intermediate results.** When True, only the final results of the search process are returned. A final result is a result that is not refined any further in the search process. When False, intermediate results are also returned.

**Maximum number of alternatives.** Specifies the maximum number of alternatives that can be returned upon running the mining task. The minimum allowed setting is 1; there is no maximum setting.

Note that the mining task will only return the actual number of alternatives, up to the maximum specified. For example, if the maximum is set to 100 and only 3 alternatives are found, only those 3 are shown.

## Decision List Model Nugget

---

A model consists of a list of **segments**, each of which is defined by a **rule** that selects matching records. You can easily view or modify the segments before generating the model and choose which ones to include or exclude. When used in scoring, Decision List models return Yes for included segments and \$null\$ for everything else, including the remainder. This direct control over scoring makes Decision List models ideal for generating mailing lists, and they are widely used in customer relationship management, including call center or marketing applications.

When you run a stream containing a Decision List model, the node adds three new fields containing the score, either 1 (meaning Yes) for included fields or \$null\$ for excluded fields, the probability (hit rate) for the segment within which the record falls, and the ID number for the segment. The names of the new fields are derived from the name of the output field being predicted, prefixed with \$D- for the score, \$DP- for the probability, and \$DI- for the segment ID.

The model is scored based on the target value specified when the model was built. You can manually exclude segments so that they score as \$null\$. For example, if you run a low probability search to find segments with lower than average hit rates, these “low” segments will be scored as Yes unless you manually exclude them. If necessary, nulls can be recoded as No using a Derive or Filler node.

### PMML

A Decision List model can be stored as a PMML RuleSetModel with a “first hit” selection criterion. However, all of the rules are expected to have the same score. To allow for changes to the target field or the target value, multiple rule set models can be stored in one file to be applied in order, cases not

matched by the first model being passed to the second, and so on. The algorithm name *DecisionList* is used to indicate this non-standard behavior, and only rule set models with this name are recognized as Decision List models and scored as such.

## Decision List Model Nugget Settings

The Settings tab for a Decision List model nugget enables you to obtain propensity scores and to enable or disable SQL optimization. This tab is available only after adding the model nugget to a stream.

**Calculate raw propensity scores.** For models with a flag target (which return a yes or no prediction), you can request propensity scores that indicate the likelihood of the true outcome specified for the target field. These are in addition to other prediction and confidence values that may be generated during scoring.

**Calculate adjusted propensity scores.** Raw propensity scores are based only on the training data and may be overly optimistic due to the tendency of many models to overfit this data. Adjusted propensities attempt to compensate by evaluating model performance against a test or validation partition. This option requires that a partition field be defined in the stream and adjusted propensity scores be enabled in the modeling node before generating the model.

**Generate SQL for this model** When using data from a database, SQL code can be pushed back to the database for execution, providing superior performance for many operations.

Select one of the following options to specify how SQL generation is performed.

- **Default: Score using Server Scoring Adapter (if installed) otherwise in process** If connected to a database with a scoring adapter installed, generates SQL using the scoring adapter and associated user defined functions (UDF) and scores your model within the database. When no scoring adapter is available, this option fetches your data back from the database and scores it in SPSS Modeler.
- **Score by converting to native SQL** If selected, generates native SQL to score the model within the database.  
**Note:** Although this option can provide quicker results, the size and complexity of the native SQL increases as the complexity of the model increases.
- **Score outside of the Database** If selected, this option fetches your data back from the database and scores it in SPSS Modeler.

## Decision List Viewer

---

The easy-to-use, task-based Decision List Viewer graphical interface takes the complexity out of the model building process, freeing you from the low-level details of data mining techniques and enabling you to devote your full attention to those parts of the analysis requiring user intervention, such as setting objectives, choosing target groups, analyzing the results, and selecting the optimal model.

### Working Model Pane

The working model pane displays the current model, including mining tasks and other actions that apply to the working model.

**ID.** Identifies the sequential segment order. Model segments are calculated, in sequence, according to their ID number.

**Segment Rules.** Provides the segment name and defined segment conditions. By default, the segment name is the field name or concatenated field names used in the conditions, with a comma as a separator.

**Score.** Represents the field that you want to predict, whose value is assumed to be related to the values of other fields (the predictors).

*Note:* The following options can be toggled to display via the [“Organizing Model Measures”](#) on page 145 dialog.

**Cover.** The pie chart visually identifies the coverage each segment has in relation to the entire cover.

**Cover (n).** Lists the coverage for each segment in relation to the entire cover.

**Frequency.** Lists the number of hits received in relation to the cover. For example, when the cover is 79 and the frequency is 50, that means that 50 out of 79 responded for the selected segment.

**Probability.** Indicates the segment probability. For example, when the cover is 79 and the frequency is 50, that means that the probability for the segment is 63.29% (50 divided by 79).

**Error.** Indicates the segment error.

The information at the bottom of the pane indicates the cover, frequency, and probability for the entire model.

#### Working Model Toolbar

The working model pane provides the following functions via a toolbar.

*Note:* Some functions are also available by right-clicking a model segment.







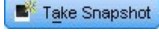






Table 9. Working model toolbar buttons	
Toolbar Button	Description
	Launches the <a href="#">Generate New Model</a> dialog, which provides options for creating a new model nugget.
	Saves the current state of the interactive session. The Decision List modeling node is updated with the current settings, including mining tasks, model snapshots, data selections, and custom measures. To restore a session to this state, check the <b>Use saved session information</b> box on the Model tab of the modeling node and click <b>Run</b> .
	Displays the Organize Model Measures dialog. See the topic <a href="#">“Organizing Model Measures”</a> on page 145 for more information.
	Displays the Organize Data Selections dialog. See the topic <a href="#">“Organizing Data Selections”</a> on page 141 for more information.
	Displays the Snapshots tab. See the topic <a href="#">“Snapshots Tab”</a> on page 138 for more information.
	Displays the Alternatives tab. See the topic <a href="#">“Alternatives Tab”</a> on page 138 for more information.
	Takes a snapshot of the current model structure. Snapshots display on the Snapshots tab and are commonly used for model comparison purposes.
	Launches the <a href="#">Inserting Segments</a> dialog, which provides options for creating new model segments.
	Launches the <a href="#">Editing Segment Rules</a> dialog, which provides options for adding conditions to model segments or changing previously defined model segment conditions.
	Moves the selected segment up in the model hierarchy.
	Moves the selected segment down in the model hierarchy.
	Deletes the selected segment.

Table 9. Working model toolbar buttons (continued)

	Toggles whether the selected segment is included in the model. When excluded, the segment results are added to the remainder. This differs from deleting a segment in that you have the option of reactivating the segment.
---	---

## Alternatives Tab

Generated when you click **Find Segments**, the Alternatives tab lists all alternative mining results for the selected model or segment on the working model pane.

To promote an alternative to be the working model, highlight the required alternative and click **Load**; the alternative model is displayed in the working model pane.

*Note:* The Alternatives tab is only displayed if you have set **Maximum number of alternatives** on the Decision List modeling node Expert tab to create more than one alternative.

Each generated model alternative displays specific model information:

**Name.** Each alternative is sequentially numbered. The first alternative usually contains the best results.

**Target.** Indicates the target value. For example: 1, which equals "true".

**No. of Segments.** The number of segment rules used in the alternative model.

**Cover.** The coverage of the alternative model.

**Freq.** The number of hits in relation to the cover.

**Prob.** Indicates the probability percentage of the alternative model.

*Note:* Alternative results are not saved with the model; results are valid only during the active session.

## Snapshots Tab

A snapshot is a view of a model at a specific point in time. For example, you could take a model snapshot when you want to load a different alternative model into the working model pane but do not want to lose the work on the current model. The Snapshots tab lists all model snapshots manually taken for any number of working model states.

*Note:* Snapshots are saved with the model. We recommend that you take a snapshot when you load the first model. This snapshot will then preserve the original model structure, ensuring that you can always return to the original model state. The generated snapshot name displays as a timestamp, indicating when it was generated.

Create a Model Snapshot

1. Select an appropriate model/alternative to display in the working model pane.
2. Make any necessary changes to the working model.
3. Click **Take Snapshot**. A new snapshot is displayed on the Snapshots tab.

**Name.** The snapshot name. You can change a snapshot name by double-clicking the snapshot name.

**Target.** Indicates the target value. For example: 1, which equals "true".

**No. of Segments.** The number of segment rules used in the model.

**Cover.** The coverage of the model.

**Freq.** The number of hits in relation to the cover.

**Prob.** Indicates the probability percentage of the model.

4. To promote a snapshot to be the working model, highlight the required snapshot and click **Load**; the snapshot model is displayed in the working model pane.
5. You can delete a snapshot by clicking **Delete** or by right-clicking the snapshot and choosing **Delete** from the menu.



## Working with Decision List Viewer

A model that will best predict customer response and behavior is built in various stages. When Decision List Viewer launches, the working model is populated with the defined model segments and measures, ready for you start a mining task, modify the segments/measures as required, and generate a new model or modeling node.

You can add one or more segment rules until you have developed a satisfactory model. You can add segment rules to the model by running mining tasks or by using the **Edit Segment Rule** function.

In the model building process, you can assess the performance of the model by validating the model against measure data, by visualizing the model in a chart, or by generating custom Excel measures.

When you feel certain about the model's quality, you can generate a new model and place it on the IBM SPSS Modeler canvas or Model palette.

### Mining Tasks

A **mining task** is a collection of parameters that determines the way new rules are generated. Some of these parameters are selectable to provide you with the flexibility to adapt models to new situations. A task consists of a task template (type), a target, and a build selection (mining dataset).

The following sections detail the various mining task operations:

- [“Running Mining Tasks” on page 139](#)
- [“Creating and Editing a Mining Task” on page 140](#)
- [“Organizing Data Selections” on page 141](#)

### *Running Mining Tasks*

Decision List Viewer enables you to manually add segment rules to a model by running mining tasks or by copying and pasting segment rules between models. A mining task holds information on how to generate new segment rules (the data mining parameter settings, such as the search strategy, source attributes, search width, confidence level, and so on), the customer behavior to predict, and the data to investigate. The goal of a mining task is to search for the best possible segment rules.

To generate a model segment rule by running a mining task:

1. Click the **Remainder** row. If there are already segments displayed on the working model pane, you can also select one of the segments to find additional rules based on the selected segment. After selecting the remainder or segment, use one of the following methods to generate the model, or alternative models:
  - From the Tools menu, choose **Find Segments**.
  - Right-click the **Remainder** row/segment and choose **Find Segments**.
  - Click the **Find Segments** button on the working model pane.

While the task is processing, the progress is displayed at the bottom of the workspace and informs you when the task has completed. Precisely how long a task takes to complete depends on the complexity of the mining task and the size of the dataset. If there is only a single model in the results it is displayed on the working model pane as soon as the task completes; however, where the results contain more than one model they are displayed on the Alternatives tab.

*Note:* A task result will either complete with models, complete with no models, or fail.

The process of finding new segment rules can be repeated until no new rules are added to the model. This means that all significant groups of customers have been found.

It is possible to run a mining task on any existing model segment. If the result of a task is not what you are looking for, you can choose to start another mining task on the same segment. This will provide additional found rules based on the selected segment. Segments that are "below" the selected segment (that is, added to the model later than the selected segment) are replaced by the new segments because each segment depends on its predecessors.

## Creating and Editing a Mining Task

A mining task is the mechanism that searches for the collection of rules that make up a data model. Alongside the search criteria defined in the selected template, a task also defines the target (the actual question that motivated the analysis, such as how many customers are likely to respond to a mailing), and it identifies the datasets to be used. The goal of a mining task is to search for the best possible models.

Create a mining task

To create a mining task:

1. Select the segment from which you want to mine additional segment conditions.
2. Click **Settings**. The Create/Edit Mining Task dialog opens. This dialog provides options for defining the mining task.
3. Make any necessary changes and click **OK** to return to the working model pane. Decision List Viewer uses the settings as the defaults to run for each task until an alternative task or settings is selected.
4. Click **Find Segments** to start the mining task on the selected segment.

Edit a mining task

The Create/Edit Mining Task dialog provides options for defining a new mining task or editing an existing one.

Most parameters available for mining tasks are similar to those offered in the Decision List node. The exceptions are shown below. See the topic [“Decision List Model Options” on page 134](#) for more information.

**Load Settings:** When you have created more than one mining task, select the required task.

**New...** Click to create a new mining task based on the settings of the task currently displayed.

Target

**Target Field:** Represents the field that you want to predict, whose value is assumed to be related to the values of other fields (the predictors).

**Target value.** Specifies the value of the target field that indicates the outcome you want to model. For example, if the target field churn is coded 0 = no and 1 = yes, specify 1 to identify rules that indicate which records are likely to churn.

Simple Settings

**Maximum number of alternatives.** Specifies the number of alternatives that will be displayed upon running the mining task. The minimum allowed setting is 1; there is no maximum setting.

Expert Settings

**Edit...** Opens the **Edit Advanced Parameters** dialog that enables you to define the advanced settings. See the topic [“Edit Advanced Parameters” on page 140](#) for more information.

Data

**Build selection.** Provides options for specifying the evaluation measure that Decision List Viewer should analyze to find new rules. The listed evaluation measures are created/edited in the Organize Data Selections dialog.

**Available fields.** Provides options for displaying all fields or manually selecting which fields to display.

**Edit...** If the **Custom** option is selected, this opens the **Customize Available Fields** dialog that enables you to select which fields are available as segment attributes found by the mining task. See the topic [“Customize Available Fields” on page 141](#) for more information.

*Edit Advanced Parameters*

The Edit Advanced Parameters dialog provides the following configuration options.

**Binning method.** The method used for binning continuous fields (equal count or equal width).



**Number of bins.** The number of bins to create for continuous fields. The minimum allowed setting is 2; there is no maximum setting.

**Model search width.** The maximum number of model results per cycle that can be used for the next cycle. The minimum allowed setting is 1; there is no maximum setting.

**Rule search width.** The maximum number of rule results per cycle that can be used for the next cycle. The minimum allowed setting is 1; there is no maximum setting.

**Bin merging factor.** The minimum amount by which a segment must grow when merged with its neighbor. The minimum allowed setting is 1.01; there is no maximum setting.

- **Allow missing values in conditions.** True to allow the IS MISSING test in rules.
- **Discard intermediate results.** When True, only the final results of the search process are returned. A final result is a result that is not refined any further in the search process. When False, intermediate results are also returned.

#### *Customize Available Fields*

The Customize Available Fields dialog enables you to select which fields are available as segment attributes found by the mining task.

**Available.** Lists the fields that are currently available as segment attributes. To remove fields from the list, select the appropriate fields and click **Remove >>**. The selected fields move from the Available list to the Not Available list.

**Not Available.** Lists the fields that are not available as segment attributes. To include the fields in the available list, select the appropriate fields and click **<< Add**. The selected fields move from the Not Available list to the Available list.

#### *Organizing Data Selections*

By organizing data selections (a mining dataset), you can specify which evaluation measures Decision List Viewer should analyze to find new rules and select which data selections are used as the basis for measures.

To organize data selections:

1. From the Tools menu, choose **Organize Data Selections**, or right-click a segment and choose the option. The Organize Data Selections dialog opens.

*Note:* The Organize Data Selections dialog also enables you to edit or delete existing data selections.

2. Click the **Add new data selection** button. A new data selection entry is added to the existing table.
3. Click **Name** and enter an appropriate selection name.
4. Click **Partition** and select an appropriate partition type.
5. Click **Condition** and select an appropriate condition option. When **Specify** is selected, the Specify Selection Condition dialog opens, providing options for defining specific field conditions.
6. Define the appropriate condition and click **OK**.

The data selections are available from the Build Selection drop-down list in the Create/Edit Mining Task dialog. The list enables you to select which evaluation measure is used for a particular mining task.

#### **Segment Rules**

You find model segment rules by running a mining task based on a task template. You can manually add segment rules to a model using the Insert Segment or Edit Segment Rule functions.

If you choose to mine for new segment rules, the results, if any, are displayed on the Viewer tab of the Interactive List dialog. You can quickly refine your model by selecting one of the alternative results from the Model Albums dialog and clicking **Load**. In this way, you can experiment with differing results until you are ready to build a model that accurately describes your optimum target group.

## Inserting Segments

You can manually add segment rules to a model using the Insert Segment function.

To add a segment rule condition to a model:

1. In the Interactive List dialog, select a location where you want to add a new segment. The new segment will be inserted directly above the selected segment.

2. From the Edit menu, choose **Insert Segment**, or access this selection by right-clicking a segment.

The Insert Segment dialog opens, enabling you to insert new segment rule conditions.

3. Click **Insert**. The Insert Condition dialog opens, enabling you to define the attributes for the new rule condition.

4. Select a field and an operator from the drop-down lists.

*Note:* If you select the **Not in** operator, the selected condition will function as an exclusion condition and displays in red in the Insert Rule dialog. For example, when the condition `region = 'TOWN'` displays in red, it means that TOWN is excluded from the result set.

5. Enter one or more values or click the **Insert Value** icon to display the Insert Value dialog. The dialog enables you to choose a value defined for the selected field. For example, the field **married** will provide the values **yes** and **no**.

6. Click **OK** to return to the Insert Segment dialog. Click **OK** a second time to add the created segment to the model.

The new segment will display in the specified model location.

## Editing Segment Rules

The Edit Segment Rule functionality enables you to add, change, or delete segment rule conditions.

To change a segment rule condition:

1. Select the model segment that you want to edit.

2. From the Edit menu, choose **Edit Segment Rule**, or right-click on the rule to access this selection.

The Edit Segment Rule dialog opens.

3. Select the appropriate condition and click **Edit**.

The Edit Condition dialog opens, enabling you to define the attributes for the selected rule condition.

4. Select a field and an operator from the drop-down lists.

*Note:* If you select the **Not in** operator, the selected condition will function as an exclusion condition and displays in red in the Edit Segment Rule dialog. For example, when the condition `region = 'TOWN'` displays in red, it means that TOWN is excluded from the result set.

5. Enter one or more values or click the **Insert Value** button to display the Insert Value dialog. The dialog enables you to choose a value defined for the selected field. For example, the field **married** will provide the values **yes** and **no**.

6. Click **OK** to return to the Edit Segment Rule dialog. Click **OK** a second time to return to the working model.

The selected segment will display with the updated rule conditions.

## Deleting Segment Rule Conditions

**To delete a segment rule condition:**

1. Select the model segment containing the rule conditions that you want to delete.

2. From the Edit menu, choose **Edit Segment Rule**, or right-click on the segment to access this selection.

The Edit Segment Rule dialog opens, enabling you to delete one or more segment rule conditions.

3. Select the appropriate rule condition and click **Delete**.

4. Click **OK**.

Deleting one or more segment rule conditions causes the working model pane to refresh its measure metrics.

### **Copying Segments**

Decision List Viewer provides you with a convenient way to copy model segments. When you want to apply a segment from one model to another model, simply copy (or cut) the segment from one model and paste it into another model. You can also copy a segment from a model displayed in the Alternative Preview panel and paste it into the model displayed in the working model pane. These cut, copy, and paste functions use a system clipboard to store or retrieve temporary data. This means that in the clipboard the conditions and target are copied. The clipboard contents are not solely reserved to be used in Decision List Viewer but can also be pasted in other applications. For example, when the clipboard contents are pasted in a text editor, the conditions and target are pasted in XML-format.

To copy or cut model segments:

1. Select the model segment that you want to use in another model.
2. From the Edit menu, choose **Copy** (or **Cut**), or right-click on the model segment and select **Copy** or **Cut**.
3. Open the appropriate model (where the model segment will be pasted).
4. Select one of the model segments, and click **Paste**.

*Note:* Instead of the **Cut**, **Copy**, and **Paste** commands you can also use the key combinations: **Ctrl+X**, **Ctrl+C**, and **Ctrl+V**.

The copied (or cut) segment is inserted above the previously selected model segment. The measures of the pasted segment and segments below are recalculated.

*Note:* Both models in this procedure must be based on the same underlying model template and contain the same target, otherwise an error message is displayed.

### **Alternative Models**

Where there is more than one result, the Alternatives tab displays the results of each mining task. Each result consists of the conditions in the selected data that most closely match the target, as well as any "good enough" alternatives. The total number of alternatives shown depends on the search criteria used in the analysis process.

To view alternative models:

1. Click on an alternative model on the Alternatives tab. The alternative model segments display, or replace the current model segments, in the Alternative Preview panel.
2. To work with an alternative model in the working model pane, select the model and click **Load** in the Alternative Preview panel or right-click an alternative name on the Alternatives tab and choose **Load**.

*Note:* Alternative models are not saved when you generate a new model.

### **Customizing a Model**

Data are not static. Customers move, get married, and change jobs. Products lose market focus and become obsolete.

Decision List Viewer offers business users the flexibility to adapt models to new situations easily and quickly. You can change a model by editing, prioritizing, deleting, or inactivating specific model segments.

### **Prioritizing Segments**

You can rank model rules in any order you choose. By default, model segments are displayed in order of priority, the first segment having the highest priority. When you assign a different priority to one or more of the segments, the model is changed accordingly. You may alter the model as required by moving segments to a higher or lower priority position.

To prioritize model segments:

1. Select the model segment to which you want to assign a different priority.
2. Click one of the two arrow buttons on the working model pane toolbar to move the selected model segment up or down the list.

After prioritization, all previous assessment results are recalculated and the new values are displayed.

### **Deleting Segments**

To delete one or more segments:

1. Select a model segment.
2. From the Edit menu, choose **Delete Segment**, or click the delete button on the toolbar of the working model pane.

The measures are recalculated for the modified model, and the model is changed accordingly.

### **Excluding Segments**

As you are searching for particular groups, you will probably base business actions on a selection of the model segments. When deploying a model, you may choose to exclude segments within a model. Excluded segments are scored as null values. Excluding a segment does not mean the segment is not used; it means that all records matching this rule are excluded from the mailing list. The rule is still applied but differently.

To exclude specific model segments:

1. Select a segment from the working model pane.
2. Click the **Toggle Segment Exclusion** button on the toolbar of the working model pane. **Excluded** is now displayed in the selected Target column of the selected segment.

*Note:* Unlike deleted segments, excluded segments remain available for reuse in the final model. Excluded segments affect chart results.

### **Change Target Value**

The Change Target Value dialog enables you to change the target value for the current target field.

Snapshots and session results with a different target value than the Working Model are identified by changing the table background for that row to yellow. This indicates that snapshot/session result is outdated.

The **Create/Edit Mining Task** dialog displays the target value for the current working model. The target value is not saved with the mining task. It is instead taken from the Working Model value.

When you promote a saved model to the Working Model that has a different target value from the current working model (for example, by editing an alternative result or editing a copy of a snapshot), the target value of the saved model is changed to be the same as the working model (the target value shown in the Working Model pane is not changed). The model metrics are reevaluated with the new target.

### **Generate New Model**

The Generate New Model dialog provides options for naming the model and selecting where the new node is created.

**Model name.** Select **Custom** to adjust the auto-generated name or to create a unique name for the node as displayed on the stream canvas.

**Create node on.** Selecting **Canvas** places the new model on the working canvas; selecting **GM Palette** places the new model on the Models palette; selecting **Both** places the new model on both the working canvas and the Models palette.

**Include interactive session state.** When enabled, the interactive session state is preserved in the generated model. When you later generate a modeling node from the model, the state is carried over and used to initialize the interactive session. Regardless of whether the option is selected, the model itself scores new data identically. When the option is not selected, the model is still able to create a build node,

but it will be a more generic build node that starts a new interactive session rather than pick up where the old session left off. If you change the node settings but execute with a saved state, the settings you have changed are ignored in favor of the settings from the saved state.

*Note:* The standard metrics are the only metrics that remain with the model. Additional metrics are preserved with the interactive state. The generated model does not represent the saved interactive mining task state. Once you launch the Decision List Viewer, it displays the settings originally made through the Viewer.

See the topic [“Regenerating a modeling node”](#) on page 41 for more information.

## Model Assessment

Successful modeling requires the careful assessment of the model before implementation in the production environment takes place. Decision List Viewer provides a number of statistical and business measures that can be used to assess the impact of a model in the real world. These include gains charts and full interoperability with Excel, thus enabling cost/benefit scenarios to be simulated for assessing the impact of deployment.

You can assess your model in the following ways:

- Using the predefined statistical and business model measures available in Decision List Viewer (probability, frequency).
- Evaluating measures imported from Microsoft Excel.
- Visualizing the model using a gains chart.

## Organizing Model Measures

Decision List Viewer provides options for defining the measures that are calculated and displayed as columns. Each segment can include the default cover, frequency, probability, and error measures represented as columns. You can also create new measures that will be displayed as columns.

### Defining Model Measures

To add a measure to your model or to define an existing measure:

1. From the Tools menu, choose **Organize Model Measures**, or right-click the model to make this selection. The Organize Model Measures dialog opens.
2. Click the **Add new model measure** button (to the right of the Show column). A new measure is displayed in the table.
3. Provide a measure name and select an appropriate type, display option, and selection. The Show column indicates whether the measure will display for the working model. When defining an existing measure, select an appropriate metric and selection and indicate if the measure will display for the working model.
4. Click **OK** to return to the Decision List Viewer workspace. If the Show column for the new measure was checked, the new measure will display for the working model.

### Custom Metrics in Excel

See the topic [“Assessment in Excel”](#) on page 146 for more information.

## Refreshing Measures

In certain cases, it may be necessary to recalculate the model measures, such as when you apply an existing model to a new set of customers.

To recalculate (refresh) the model measures:

From the Edit menu, choose **Refresh All Measures**.

or

Press F5.

All measures are recalculated, and the new values are shown for the working model.

## Assessment in Excel

Decision List Viewer can be integrated with Microsoft Excel, enabling you to use your own value calculations and profit formulas directly within the model building process to simulate cost/benefit scenarios. The link with Excel enables you to export data to Excel, where it can be used to create presentation charts, calculate custom measures, such as complex profit and ROI measures, and view them in Decision List Viewer while building the model.

*Note:* In order for you to work with an Excel spreadsheet, the analytical CRM expert has to define configuration information for the synchronization of Decision List Viewer with Microsoft Excel. The configuration is contained in an Excel spreadsheet file and indicates which information is transferred from Decision List Viewer to Excel, and vice versa.

The following steps are valid only when MS Excel is installed. If Excel is not installed, the options for synchronizing models with Excel are not displayed.

To synchronize models with MS Excel:

1. Open the model, run an interactive session, and choose **Organize Model Measures** from the Tools menu.
2. Select **Yes** for the **Calculate custom measures in Excel** option. The **Workbook** field activates, enabling you to select a preconfigured Excel workbook template.
3. Click the **Connect to Excel** button. The Open dialog opens, enabling you to navigate to the preconfigured template location on your local or network file system.
4. Select the appropriate Excel template and click **Open**. The selected Excel template launches; use the Windows taskbar (or press Alt-Tab) to navigate back to the Choose Inputs for Custom Measures dialog.
5. Select the appropriate mappings between the metric names defined in the Excel template and the model metric names and click **OK**.

Once the link is established, Excel starts with the preconfigured Excel template that displays the model rules in the spreadsheet. The results calculated in Excel are displayed as new columns in Decision List Viewer.

*Note:* Excel metrics do not remain when the model is saved; the metrics are valid only during the active session. However, you can create snapshots that include Excel metrics. The Excel metrics saved in the snapshot views are valid only for historical comparison purposes and do not refresh when reopened. See the topic [“Snapshots Tab” on page 138](#) for more information. The Excel metrics will not display in the snapshots until you reestablish a connection to the Excel template.

### MS Excel Integration Setup

The integration between Decision List Viewer and Microsoft Excel is accomplished through the use of a preconfigured Excel spreadsheet template. The template consists of three worksheets:

**Model Measures.** Displays the imported Decision List Viewer measures, the custom Excel measures, and the calculation totals (defined on the Settings worksheet).

**Settings.** Provides the variables to generate calculations based on the imported Decision List Viewer measures and the custom Excel measures.

**Configuration.** Provides options for specifying which measures are imported from Decision List Viewer and for defining the custom Excel measures.

**WARNING:** The structure of the Configuration worksheet is rigidly defined. Do **NOT** edit any cells in the green shaded area.

- **Metrics From Model.** Indicates which Decision List Viewer metrics are used in the calculations.
- **Metrics To Model.** Indicates which Excel-generated metric(s) will be returned to Decision List Viewer. The Excel-generated metrics display as new measure columns in Decision List Viewer.

*Note:* Excel metrics do not remain with the model when you generate a new model; the metrics are valid only during the active session.

### *Changing the Model Measures*

The following examples explain how to change Model Measures in several ways:

- Change an existing measure.
- Import an additional standard measure from the model.
- Export an additional custom measure to the model.

Change an existing measure

1. Open the template and select the Configuration worksheet.
2. Edit any **Name** or **Description** by highlighting and typing over them.

Note that if you want to change a measure--for example, to prompt the user for Probability instead of Frequency--you only need to change the name and description in **Metrics From Model** – this is then displayed in the model and the user can choose the appropriate measure to map.

Import an additional standard measure from the model

1. Open the template and select the Configuration worksheet.
2. From the menus choose:

**Tools > Protection > Unprotect Sheet**

3. Select cell A5, which is shaded yellow and contains the word **End**.
4. From the menus choose:

**Insert > Rows**

5. Type in the **Name** and **Description** of the new measure. For example, **Error** and **Error associated with segment**.
6. In cell C5, enter the formula **=COLUMN('Model Measures'!N3)**.
7. In cell D5, enter the formula **=ROW('Model Measures'!N3)+1**.

These formulae will cause the new measure to be displayed in column N of the Model Measures worksheet, which is currently empty.

8. From the menus choose:

**Tools > Protection > Protect Sheet**

9. Click **OK**.
10. On the Model Measures worksheet, ensure that cell N3 has **Error** as a title for the new column.
11. Select all of column N.
12. From the menus choose:  
**Format > Cells**
13. By default, all of the cells have a **General** number category. Click **Percentage** to change how the figures are displayed. This helps you check your figures in Excel; in addition, it enables you to utilize the data in other ways, for example, as an output to a graph.
14. Click **OK**.
15. Save the spreadsheet as an Excel 2003 template, with a unique name and the file extension **.xlt**. For ease of locating the new template, we recommend you save it in the preconfigured template location on your local or network file system.

Export an additional custom measure to the model

1. Open the template to which you added the Error column in the previous example; select the Configuration worksheet.
2. From the menus choose:

### **Tools > Protection > Unprotect Sheet**

3. Select cell A14, which is shaded yellow and contains the word **End**.
4. From the menus choose:

### **Insert > Rows**

5. Type in the **Name** and **Description** of the new measure. For example, **Scaled Error** and **Scaling applied to error from Excel**.
6. In cell C14, enter the formula **=COLUMN('Model Measures'!O3)**.
7. In cell D14, enter the formula **=ROW('Model Measures'!O3)+1**.  
These formulae specify that the column O will supply the new measure to the model.
8. Select the Settings worksheet.
9. In cell A17, enter the description **'- Scaled Error**.
10. In cell B17, enter the scaling factor of **10**.
11. On the Model Measures worksheet, enter the description **Scaled Error** in cell O3 as a title for the new column.
12. In cell O4, enter the formula **=N4\*Settings!\$B\$17**.
13. Select the corner of cell O4 and drag it down to cell O22 to copy the formula into each cell.
14. From the menus choose:

### **Tools > Protection > Protect Sheet**

15. Click **OK**.
16. Save the spreadsheet as an Excel 2003 template, with a unique name and the file extension *.xlt*. For ease of locating the new template, we recommend you save it in the preconfigured template location on your local or network file system.

When you connect to Excel using this template, the Error value is available as a new custom measure.

## **Visualizing Models**

The best way to understand the impact of a model is to visualize it. Using a gains chart, you can obtain valuable day-to-day insight into the business and technical benefit of your model by studying the effect of multiple alternatives in real time. The [“Gains Chart”](#) on page 148 section shows the benefit of a model over randomized decision-making and enables the direct comparison of multiple charts when there are alternative models.

### **Gains Chart**

The gains chart plots the values in the *Gains %* column from the table. Gains are defined as the proportion of hits in each increment relative to the total number of hits in the tree, using the equation:

$$(\text{hits in increment} / \text{total number of hits}) \times 100\%$$

Gains charts effectively illustrate how widely you need to cast the net to capture a given percentage of all of the hits in the tree. The diagonal line plots the expected response for the entire sample if the model is not used. In this case, the response rate would be constant, since one person is just as likely to respond as another. To double your yield, you would need to ask twice as many people. The curved line indicates how much you can improve your response by including only those who rank in the higher percentiles based on gain. For example, including the top 50% might net you more than 70% of the positive responses. The steeper the curve, the higher the gain.

To view a gains chart:

1. Open a stream that contains a Decision List node and launch an interactive session from the node.
2. Click the **Gains** tab. Depending on which partitions are specified, you may see one or two charts (two charts would display, for example, when both the training and testing partitions are defined for the model measures).



By default, the charts display as segments. You can switch the charts to display as quantiles by selecting **Quantiles** and then selecting the appropriate quantile method from the drop-down menu.

### *Chart Options*

The Chart Options feature provides options for selecting which models and snapshots are charted, which partitions are plotted, and whether or not segment labels display.

#### Models to Plot

**Current Models.** Enables you to select which models to chart. You can select the working model or any created snapshot models.

#### Partitions to Plot

**Partitions for left-hand chart.** The drop-down list provides options for displaying all defined partitions or all data.

**Partitions for right-hand chart.** The drop-down list provides options for displaying all defined partitions, all data, or only the left-hand chart. When **Graph only left** is selected, only the left chart is displayed.

**Display Segment Labels.** When enabled, each segment label is displayed on the charts.



---

## Chapter 10. Statistical Models

Statistical models use mathematical equations to encode information extracted from the data. In some cases, statistical modeling techniques can provide adequate models very quickly. Even for problems in which more flexible machine-learning techniques (such as neural networks) can ultimately give better results, you can use some statistical models as baseline predictive models to judge the performance of more advanced techniques.

The following statistical modeling nodes are available.



Linear regression models predict a continuous target based on linear relationships between the target and one or more predictors.



Logistic regression is a statistical technique for classifying records based on values of input fields. It is analogous to linear regression but takes a categorical target field instead of a numeric range.



The PCA/Factor node provides powerful data-reduction techniques to reduce the complexity of your data. Principal components analysis (PCA) finds linear combinations of the input fields that do the best job of capturing the variance in the entire set of fields, where the components are orthogonal (perpendicular) to each other. Factor analysis attempts to identify underlying factors that explain the pattern of correlations within a set of observed fields. For both approaches, the goal is to find a small number of derived fields that effectively summarizes the information in the original set of fields.



Discriminant analysis makes more stringent assumptions than logistic regression but can be a valuable alternative or supplement to a logistic regression analysis when those assumptions are met.



The Generalized Linear model expands the general linear model so that the dependent variable is linearly related to the factors and covariates through a specified link function. Moreover, the model allows for the dependent variable to have a non-normal distribution. It covers the functionality of a wide number of statistical models, including linear regression, logistic regression, loglinear models for count data, and interval-censored survival models.



A generalized linear mixed model (GLMM) extends the linear model so that the target can have a non-normal distribution, is linearly related to the factors and covariates via a specified link function, and so that the observations can be correlated. Generalized linear mixed models cover a wide variety of models, from simple linear regression to complex multilevel models for non-normal longitudinal data.



The Cox regression node enables you to build a survival model for time-to-event data in the presence of censored records. The model produces a survival function that predicts the probability that the event of interest has occurred at a given time ( $t$ ) for given values of the input variables.

## Linear Node

---

Linear regression is a common statistical technique for classifying records based on the values of numeric input fields. Linear regression fits a straight line or surface that minimizes the discrepancies between predicted and actual output values.

**Requirements.** Only numeric fields can be used in a linear regression model. You must have exactly one target field (with the role set to **Target**) and one or more predictors (with the role set to **Input**). Fields with a role of **Both** or **None** are ignored, as are non-numeric fields. (If necessary, non-numeric fields can be recoded using a Derive node.)

**Strengths.** Linear regression models are relatively simple and give an easily interpreted mathematical formula for generating predictions. Because linear regression is a long-established statistical procedure, the properties of these models are well understood. Linear models are also typically very fast to train. The Linear node provides methods for automatic field selection in order to eliminate nonsignificant input fields from the equation.

**Note:** In cases where the target field is categorical rather than a continuous range, such as **yes/no** or **churn/don't churn**, logistic regression can be used as an alternative. Logistic regression also provides support for non-numeric inputs, removing the need to recode these fields. See the topic [“Logistic Node”](#) on page 161 for more information.

### Linear models

Linear models predict a continuous target based on linear relationships between the target and one or more predictors.

Linear models are relatively simple and give an easily interpreted mathematical formula for scoring. The properties of these models are well understood and can typically be built very quickly compared to other model types (such as neural networks or decision trees) on the same dataset.

**Example.** An insurance company with limited resources to investigate homeowners' insurance claims wants to build a model for estimating claims costs. By deploying this model to service centers, representatives can enter claim information while on the phone with a customer and immediately obtain the "expected" cost of the claim based on past data.

**Field requirements.** There must be a Target and at least one Input. By default, fields with predefined roles of Both or None are not used. The target must be continuous (scale). There are no measurement level restrictions on predictors (inputs); categorical (flag, nominal, and ordinal) fields are used as factors in the model and continuous fields are used as covariates.

### Objectives

#### What do you want to do?

- **Build a new model.** Build a completely new model. This is the usual operation of the node.
- **Continue training an existing model.** Training continues with the last model successfully produced by the node. This makes it possible to update or refresh an existing model without having to access the original data and may result in significantly faster performance since only the new or updated records are fed into the stream. Details on the previous model are stored with the modeling node, making it possible to use this option even if the previous model nugget is no longer available in the stream or Models palette.

**Note:** When this option is enabled, all other controls on the Fields and Build Options tabs are disabled.

**What is your main objective?** Select the appropriate objective.

- **Create a standard model.** The method builds a single model to predict the target using the predictors. Generally speaking, standard models are easier to interpret and can be faster to score than boosted, bagged, or large dataset ensembles.

**Note:** For split models, to use this option with **Continue training an existing model** you must be connected to Analytic Server.

- **Enhance model accuracy (boosting).** The method builds an ensemble model using boosting, which generates a sequence of models to obtain more accurate predictions. Ensembles can take longer to build and to score than a standard model.

Boosting produces a succession of "component models", each of which is built on the entire dataset. Prior to building each successive component model, the records are weighted based on the previous component model's residuals. Cases with large residuals are given relatively higher analysis weights so that the next component model will focus on predicting these records well. Together these component models form an ensemble model. The ensemble model scores new records using a combining rule; the available rules depend upon the measurement level of the target.

- **Enhance model stability (bagging).** The method builds an ensemble model using bagging (bootstrap aggregating), which generates multiple models to obtain more reliable predictions. Ensembles can take longer to build and to score than a standard model.

Bootstrap aggregation (bagging) produces replicates of the training dataset by sampling with replacement from the original dataset. This creates bootstrap samples of equal size to the original dataset. Then a "component model" is built on each replicate. Together these component models form an ensemble model. The ensemble model scores new records using a combining rule; the available rules depend upon the measurement level of the target.

- **Create a model for very large datasets.** The method builds an ensemble model by splitting the dataset into separate data blocks. Choose this option if your dataset is too large to build any of the models above, or for incremental model building. This option can take less time to build, but can take longer to score than a standard model.

See ["Ensembles"](#) on page 154 for settings related to boosting, bagging, and very large datasets.

## Basics

**Automatically prepare data.** This option allows the procedure to internally transform the target and predictors in order to maximize the predictive power of the model; any transformations are saved with the model and applied to new data for scoring. The original versions of transformed fields are excluded from the model. By default, the following automatic data preparation are performed.

- **Date and Time handling.** Each date predictor is transformed into new a continuous predictor containing the elapsed time since a reference date (1970-01-01). Each time predictor is transformed into a new continuous predictor containing the time elapsed since a reference time (00:00:00).
- **Adjust measurement level.** Continuous predictors with less than 5 distinct values are recast as ordinal predictors. Ordinal predictors with greater than 10 distinct values are recast as continuous predictors.
- **Outlier handling.** Values of continuous predictors that lie beyond a cutoff value (3 standard deviations from the mean) are set to the cutoff value.
- **Missing value handling.** Missing values of nominal predictors are replaced with the mode of the training partition. Missing values of ordinal predictors are replaced with the median of the training partition. Missing values of continuous predictors are replaced with the mean of the training partition.
- **Supervised merging.** This makes a more parsimonious model by reducing the number of fields to be processed in association with the target. Similar categories are identified based upon the relationship between the input and the target. Categories that are not significantly different (that is, having a p-value greater than 0.1) are merged. If all categories are merged into one, the original and derived versions of the field are excluded from the model because they have no value as a predictor.

**Confidence level.** This is the level of confidence used to compute interval estimates of the model coefficients in the [Coefficients](#) view. Specify a value greater than 0 and less than 100. The default is 95.

## Model selection

**Model selection method.** Choose one of the model selection methods (details below) or **Include all predictors**, which simply enters all available predictors as main effects model terms. By default, **Forward stepwise** is used.

**Forward Stepwise Selection.** This starts with no effects in the model and adds and removes effects one step at a time until no more can be added or removed according to the stepwise criteria.

- **Criteria for entry/removal.** This is the statistic used to determine whether an effect should be added to or removed from the model. **Information Criterion (AICC)** is based on the likelihood of the training set given the model, and is adjusted to penalize overly complex models. **F Statistics** is based on a statistical test of the improvement in model error. **Adjusted R-squared** is based on the fit of the training set, and is adjusted to penalize overly complex models. **Overfit Prevention Criterion (ASE)** is based on the fit (average squared error, or ASE) of the overfit prevention set. The overfit prevention set is a random subsample of approximately 30% of the original dataset that is not used to train the model.

If any criterion other than **F Statistics** is chosen, then at each step the effect that corresponds to the greatest positive increase in the criterion is added to the model. Any effects in the model that correspond to a decrease in the criterion are removed.

If **F Statistics** is chosen as the criterion, then at each step the effect that has the smallest  $p$ -value less than the specified threshold, **Include effects with  $p$ -values less than**, is added to the model. The default is 0.05. Any effects in the model with a  $p$ -value greater than the specified threshold, **Remove effects with  $p$ -values greater than**, are removed. The default is 0.10.

- **Customize maximum number of effects in the final model.** By default, all available effects can be entered into the model. Alternatively, if the stepwise algorithm ends a step with the specified maximum number of effects, the algorithm stops with the current set of effects.
- **Customize maximum number of steps.** The stepwise algorithm stops after a certain number of steps. By default, this is 3 times the number of available effects. Alternatively, specify a positive integer maximum number of steps.

**Best Subsets Selection.** This checks "all possible" models, or at least a larger subset of the possible models than forward stepwise, to choose the best according to the best subsets criterion. **Information Criterion (AICC)** is based on the likelihood of the training set given the model, and is adjusted to penalize overly complex models. **Adjusted R-squared** is based on the fit of the training set, and is adjusted to penalize overly complex models. **Overfit Prevention Criterion (ASE)** is based on the fit (average squared error, or ASE) of the overfit prevention set. The overfit prevention set is a random subsample of approximately 30% of the original dataset that is not used to train the model.

The model with the greatest value of the criterion is chosen as the best model.

**Note:** Best subsets selection is more computationally intensive than forward stepwise selection. When best subsets is performed in conjunction with boosting, bagging, or very large datasets, it can take considerably longer to build than a standard model built using forward stepwise selection.

## Ensembles

These settings determine the behavior of ensembling that occurs when boosting, bagging, or very large datasets are requested in Objectives. Options that do not apply to the selected objective are ignored.

**Bagging and Very Large Datasets.** When scoring an ensemble, this is the rule used to combine the predicted values from the base models to compute the ensemble score value.

- **Default combining rule for continuous targets.** Ensemble predicted values for continuous targets can be combined using the mean or median of the predicted values from the base models.

Note that when the objective is to enhance model accuracy, the combining rule selections are ignored. Boosting always uses a weighted majority vote to score categorical targets and a weighted median to score continuous targets.

**Boosting and Bagging.** Specify the number of base models to build when the objective is to enhance model accuracy or stability; for bagging, this is the number of bootstrap samples. It should be a positive integer.

## Advanced

**Replicate results.** Setting a random seed allows you to replicate analyses. The random number generator is used to choose which records are in the overfit prevention set. Specify an integer or click **Generate**, which will create a pseudo-random integer between 1 and 2147483647, inclusive. The default is 54752075.

## Model Options

**Model Name.** You can generate the model name automatically based on the target fields or specify a custom name. The automatically generated name is the target field name.

Note that the predicted value is always computed when the model is scored. The name of the new field is the name of the target field, prefixed with  $SL$ -. For example, for a target field named *sales*, the new field would be named  $SL$ -*sales*.

## Model Summary

The Model Summary view is a snapshot, at-a-glance summary of the model and its fit.

### Table

The table identifies some high-level model settings, including:

- The name of the target specified on the [Fields](#) tab,
- Whether automatic data preparation was performed as specified on the [Basics](#) settings,
- The model selection method and selection criterion specified on the [Model Selection](#) settings. The value of the selection criterion for the final model is also displayed, and is presented in smaller is better format.

### Chart

The chart displays the accuracy of the final model, which is presented in larger is better format. The value is  $100 \times$  the adjusted  $R^2$  for the final model.

## Automatic Data Preparation

This view shows information about which fields were excluded and how transformed fields were derived in the automatic data preparation (ADP) step. For each field that was transformed or excluded, the table lists the field name, its role in the analysis, and the action taken by the ADP step. Fields are sorted by ascending alphabetical order of field names. The possible actions taken for each field include:

- **Derive duration: months** computes the elapsed time in months from the values in a field containing dates to the current system date.
- **Derive duration: hours** computes the elapsed time in hours from the values in a field containing times to the current system time.
- **Change measurement level from continuous to ordinal** recasts continuous fields with less than 5 unique values as ordinal fields.
- **Change measurement level from ordinal to continuous** recasts ordinal fields with more than 10 unique values as continuous fields.
- **Trim outliers** sets values of continuous predictors that lie beyond a cutoff value (3 standard deviations from the mean) to the cutoff value.
- **Replace missing values** replaces missing values of nominal fields with the mode, ordinal fields with the median, and continuous fields with the mean.
- **Merge categories to maximize association with target** identifies "similar" predictor categories based upon the relationship between the input and the target. Categories that are not significantly different (that is, having a  $p$ -value greater than 0.05) are merged.
- **Exclude constant predictor / after outlier handling / after merging of categories** removes predictors that have a single value, possibly after other ADP actions have been taken.

## Predictor Importance

Typically, you will want to focus your modeling efforts on the predictor fields that matter most and consider dropping or ignoring those that matter least. The predictor importance chart helps you do this by indicating the relative importance of each predictor in estimating the model. Since the values are relative, the sum of the values for all predictors on the display is 1.0. Predictor importance does not relate to model accuracy. It just relates to the importance of each predictor in making a prediction, not whether or not the prediction is accurate.

## Predicted By Observed

This displays a binned scatterplot of the predicted values on the vertical axis by the observed values on the horizontal axis. Ideally, the points should lie on a 45-degree line; this view can tell you whether any records are predicted particularly badly by the model.

## Residuals

This displays a diagnostic chart of model residuals.

**Chart styles.** There are different display styles, which are accessible from the **Style** drop-down.

- **Histogram.** This is a binned histogram of the studentized residuals with an overlay of the normal distribution. Linear models assume that the residuals have a normal distribution, so the histogram should ideally closely approximate the smooth line.
- **P-P Plot.** This is a binned probability-probability plot comparing the studentized residuals to a normal distribution. If the slope of the plotted points is less steep than the normal line, the residuals show greater variability than a normal distribution; if the slope is steeper, the residuals show less variability than a normal distribution. If the plotted points have an S-shaped curve, then the distribution of residuals is skewed.

## Outliers

This table lists records that exert undue influence upon the model, and displays the record ID (if specified on the Fields tab), target value, and Cook's distance. Cook's distance is a measure of how much the residuals of all records would change if a particular record were excluded from the calculation of the model coefficients. A large Cook's distance indicates that excluding a record from changes the coefficients substantially, and should therefore be considered influential.

Influential records should be examined carefully to determine whether you can give them less weight in estimating the model, or truncate the outlying values to some acceptable threshold, or remove the influential records completely.

## Effects

This view displays the size of each effect in the model.

**Styles.** There are different display styles, which are accessible from the **Style** dropdown list.

- **Diagram.** This is a chart in which effects are sorted from top to bottom by decreasing predictor importance. Connecting lines in the diagram are weighted based on effect significance, with greater line width corresponding to more significant effects (smaller  $p$ -values). Hovering over a connecting line reveals a tooltip that shows the  $p$ -value and importance of the effect. This is the default.
- **Table.** This is an ANOVA table for the overall model and the individual model effects. The individual effects are sorted from top to bottom by decreasing predictor importance. Note that by default, the table is collapsed to only show the results for the overall model. To see the results for the individual model effects, click the **Corrected Model** cell in the table.

**Predictor importance.** There is a Predictor Importance slider that controls which predictors are shown in the view. This does not change the model, but simply allows you to focus on the most important predictors. By default, the top 10 effects are displayed.

**Significance.** There is a Significance slider that further controls which effects are shown in the view, beyond those shown based on predictor importance. Effects with significance values greater than the slider value are hidden. This does not change the model, but simply allows you to focus on the most important effects. By default the value is 1.00, so that no effects are filtered based on significance.

## Coefficients

This view displays the value of each coefficient in the model. Note that factors (categorical predictors) are indicator-coded within the model, so that **effects** containing factors will generally have multiple associated **coefficients**; one for each category except the category corresponding to the redundant (reference) parameter.



**Styles.** There are different display styles, which are accessible from the **Style** dropdown list.

- **Diagram.** This is a chart which displays the intercept first, and then sorts effects from top to bottom by decreasing predictor importance. Within effects containing factors, coefficients are sorted by ascending order of data values. Connecting lines in the diagram are colored based on the sign of the coefficient (see the diagram key) and weighted based on coefficient significance, with greater line width corresponding to more significant coefficients (smaller  $p$ -values). Hovering over a connecting line reveals a tooltip that shows the value of the coefficient, its  $p$ -value, and the importance of the effect the parameter is associated with. This is the default style.
- **Table.** This shows the values, significance tests, and confidence intervals for the individual model coefficients. After the intercept, the effects are sorted from top to bottom by decreasing predictor importance. Within effects containing factors, coefficients are sorted by ascending order of data values. Note that by default the table is collapsed to only show the coefficient, significance, and importance of each model parameter. To see the standard error,  $t$  statistic, and confidence interval, click the **Coefficient** cell in the table. Hovering over the name of a model parameter in the table reveals a tooltip that shows the name of the parameter, the effect the parameter is associated with, and (for categorical predictors), the value labels associated with the model parameter. This can be particularly useful to see the new categories created when automatic data preparation merges similar categories of a categorical predictor.

**Predictor importance.** There is a Predictor Importance slider that controls which predictors are shown in the view. This does not change the model, but simply allows you to focus on the most important predictors. By default, the top 10 effects are displayed.

**Significance.** There is a Significance slider that further controls which coefficients are shown in the view, beyond those shown based on predictor importance. Coefficients with significance values greater than the slider value are hidden. This does not change the model, but simply allows you to focus on the most important coefficients. By default the value is 1.00, so that no coefficients are filtered based on significance.

### Estimated means

These are charts displayed for significant predictors. The chart displays the model-estimated value of the target on the vertical axis for each value of the predictor on the horizontal axis, holding all other predictors constant. It provides a useful visualization of the effects of each predictor's coefficients on the target.

**Note:** If no predictors are significant, no estimated means are produced.

### Model Building Summary

When a model selection algorithm other than **None** is chosen on the Model Selection settings, this provides some details of the model building process.

**Forward stepwise.** When forward stepwise is the selection algorithm, the table displays the last 10 steps in the stepwise algorithm. For each step, the value of the selection criterion and the effects in the model at that step are shown. This gives you a sense of how much each step contributes to the model. Each column allows you to sort the rows so that you can more easily see which effects are in the model at a given step.

**Best subsets.** When best subsets is the selection algorithm, the table displays the top 10 models. For each model, the value of the selection criterion and the effects in the model are shown. This gives you a sense of the stability of the top models; if they tend to have many similar effects with a few differences, then you can be fairly confident in the "top" model; if they tend to have very different effects, then some of the effects may be too similar and should be combined (or one removed). Each column allows you to sort the rows so that you can more easily see which effects are in the model at a given step.

### Settings

Note that the predicted value is always computed when the model is scored. The name of the new field is the name of the target field, prefixed with  $\$L$ -. For example, for a target field named *sales*, the new field would be named  $\$L$ -*sales*.

**Generate SQL for this model** When using data from a database, SQL code can be pushed back to the database for execution, providing superior performance for many operations.

Select one of the following options to specify how SQL generation is performed.

- **Default: Score using Server Scoring Adapter (if installed) otherwise in process** If connected to a database with a scoring adapter installed, generates SQL using the scoring adapter and associated user defined functions (UDF) and scores your model within the database. When no scoring adapter is available, this option fetches your data back from the database and scores it in SPSS Modeler.
- **Score by converting to native SQL** If selected, generates native SQL to score the model within the database.

**Note:** Although this option can provide quicker results, the size and complexity of the native SQL increases as the complexity of the model increases.

- **Score outside of the Database** If selected, this option fetches your data back from the database and scores it in SPSS Modeler.

## Linear-AS Node

---

IBM SPSS Modeler has two different versions of the Linear node:

- **Linear** is the traditional node that runs on the IBM SPSS Modeler Server.
- **Linear-AS** can run when connected to IBM SPSS Analytic Server.

Linear regression is a common statistical technique for classifying records based on the values of numeric input fields. Linear regression fits a straight line or surface that minimizes the discrepancies between predicted and actual output values.

**Requirements.** Only numeric fields and categorical predictors can be used in a linear regression model. You must have exactly one target field (with the role set to **Target**) and one or more predictors (with the role set to **Input**). Fields with a role of **Both** or **None** are ignored, as are non-numeric fields. (If necessary, non-numeric fields can be recoded using a Derive node.)

**Strengths.** Linear regression models are relatively simple and give an easily interpreted mathematical formula for generating predictions. Because linear regression is a long-established statistical procedure, the properties of these models are well understood. Linear models are also typically very fast to train. The Linear node provides methods for automatic field selection in order to eliminate nonsignificant input fields from the equation.

**Note:** In cases where the target field is categorical rather than a continuous range, such as **yes/no** or **churn/don't churn**, logistic regression can be used as an alternative. Logistic regression also provides support for non-numeric inputs, removing the need to recode these fields. See the topic [“Logistic Node”](#) on page 161 for more information.

## Linear-AS models

Linear models predict a continuous target based on linear relationships between the target and one or more predictors.

Linear models are relatively simple and give an easily interpreted mathematical formula for scoring. The properties of these models are well understood and can typically be built very quickly compared to other model types (such as neural networks or decision trees) on the same dataset.

**Example.** An insurance company with limited resources to investigate homeowners' insurance claims wants to build a model for estimating claims costs. By deploying this model to service centers, representatives can enter claim information while on the phone with a customer and immediately obtain the "expected" cost of the claim based on past data.

**Field requirements.** There must be a Target and at least one Input. By default, fields with predefined roles of Both or None are not used. The target must be continuous (scale). There are no measurement level restrictions on predictors (inputs); categorical (flag, nominal, and ordinal) fields are used as factors in the model and continuous fields are used as covariates.

## Basics

**Include intercept.** This option includes an offset on the y axis when the x axis is 0. The intercept is usually included in the model. If you can assume the data pass through the origin, you can exclude the intercept.

**Consider two way interaction.** This option tells the model to compare each possible pair of inputs to see if the trend of one affects the other. If it does, then those inputs are more likely to be included in the design matrix.

**Confidence interval for coefficient estimates (%).** This is the interval of confidence used to compute estimates of the model coefficients in the [Coefficients](#) view. Specify a value greater than 0 and less than 100. The default is 95.

**Sorting order for categorical predictors.** These controls determine the order of the categories for the factors (categorical inputs) for purposes of determining the "last" category. The sort order setting is ignored if the input is not categorical or if a custom reference category is specified.

## Model Selection

**Model selection method.** Choose one of the model selection methods (details below) or **Include all predictors**, which simply enters all available predictors as main effects model terms. By default, **Forward stepwise** is used.

**Forward Stepwise Selection.** This starts with no effects in the model and adds and removes effects one step at a time until no more can be added or removed according to the stepwise criteria.

- **Criteria for entry/removal.** This is the statistic used to determine whether an effect should be added to or removed from the model. **Information Criterion (AICC)** is based on the likelihood of the training set given the model, and is adjusted to penalize overly complex models. **F Statistics** is based on a statistical test of the improvement in model error. **Adjusted R-squared** is based on the fit of the training set, and is adjusted to penalize overly complex models. **Overfit Prevention Criterion (ASE)** is based on the fit (average squared error, or ASE) of the overfit prevention set. The overfit prevention set is a random subsample of approximately 30% of the original dataset that is not used to train the model.

If any criterion other than **F Statistics** is chosen, then at each step the effect that corresponds to the greatest positive increase in the criterion is added to the model. Any effects in the model that correspond to a decrease in the criterion are removed.

If **F Statistics** is chosen as the criterion, then at each step the effect that has the smallest  $p$ -value less than the specified threshold, **Include effects with  $p$ -values less than**, is added to the model. The default is 0.05. Any effects in the model with a  $p$ -value greater than the specified threshold, **Remove effects with  $p$ -values greater than**, are removed. The default is 0.10.

- **Customize maximum number of effects in the final model.** By default, all available effects can be entered into the model. Alternatively, if the stepwise algorithm ends a step with the specified maximum number of effects, the algorithm stops with the current set of effects.
- **Customize maximum number of steps.** The stepwise algorithm stops after a certain number of steps. By default, this is 3 times the number of available effects. Alternatively, specify a positive integer maximum number of steps.

**Best Subsets Selection.** This checks "all possible" models, or at least a larger subset of the possible models than forward stepwise, to choose the best according to the best subsets criterion. **Information Criterion (AICC)** is based on the likelihood of the training set given the model, and is adjusted to penalize overly complex models. **Adjusted R-squared** is based on the fit of the training set, and is adjusted to penalize overly complex models. **Overfit Prevention Criterion (ASE)** is based on the fit (average squared error, or ASE) of the overfit prevention set. The overfit prevention set is a random subsample of approximately 30% of the original dataset that is not used to train the model.

The model with the greatest value of the criterion is chosen as the best model.

**Note:** Best subsets selection is more computationally intensive than forward stepwise selection. When best subsets is performed in conjunction with boosting, bagging, or very large datasets, it can take considerably longer to build than a standard model built using forward stepwise selection.

## Model Options

**Model Name.** You can generate the model name automatically based on the target fields or specify a custom name. The automatically generated name is the target field name.

Note that the predicted value is always computed when the model is scored. The name of the new field is the name of the target field, prefixed with *\$L-*. For example, for a target field named *sales*, the new field would be named *\$L-sales*.

## Interactive Output

After running a Linear-AS model, the following output is available.

### Model Information

The Model Information view provides key information about the model. The table identifies some high-level model settings, such as:

- The name of the target specified on the [Fields](#) tab
- The regression weight field
- The model building method specified on the [Model Selection](#) settings
- The number of predictors input
- The number of predictors in the final model
- Akaike Information Criterion Corrected (AICC). AICC is a measure for selecting and comparing mixed models based on the -2 (Restricted) log likelihood. Smaller values indicate better models. The AICC "corrects" the AIC for small sample sizes. As the sample size increases, the AICC converges to the AIC.
- R Square. This is the goodness-of-fit measure of a linear model, sometimes called the coefficient of determination. It is the proportion of variation in the dependent variable explained by the regression model. It ranges in value from 0 to 1. Small values indicate that the model does not fit the data well.
- Adjusted R Square

### Records Summary

The Records Summary view provides information about the number and percentage of records (cases) included and excluded from the model.

### Predictor Importance

Typically, you will want to focus your modeling efforts on the predictor fields that matter most and consider dropping or ignoring those that matter least. The predictor importance chart helps you do this by indicating the relative importance of each predictor in estimating the model. Since the values are relative, the sum of the values for all predictors on the display is 1.0. Predictor importance does not relate to model accuracy. It just relates to the importance of each predictor in making a prediction, not whether or not the prediction is accurate.

### Predicted by Observed

This displays a binned scatterplot of the predicted values on the vertical axis by the observed values on the horizontal axis. Ideally, the points should lie on a 45-degree line; this view can tell you whether any records are predicted particularly badly by the model.

### Settings

Note that the predicted value is always computed when the model is scored. The name of the new field is the name of the target field, prefixed with *\$L-*. For example, for a target field named *sales*, the new field would be named *\$L-sales*.

**Generate SQL for this model** When using data from a database, SQL code can be pushed back to the database for execution, providing superior performance for many operations.

Select one of the following options to specify how SQL generation is performed.

- **Default: Score using Server Scoring Adapter (if installed) otherwise in process.** If connected to a database with a scoring adapter installed, generates SQL using the scoring adapter and associated user defined functions (UDF) and scores your model within the database. When no scoring adapter is available, this option fetches your data back from the database and scores it in SPSS Modeler.
- **Score outside of the Database.** If selected, this option fetches your data back from the database and scores it in SPSS Modeler.

## Logistic Node

---

**Logistic regression**, also known as **nominal regression**, is a statistical technique for classifying records based on values of input fields. It is analogous to linear regression but takes a categorical target field instead of a numeric one. Both binomial models (for targets with two discrete categories) and multinomial models (for targets with more than two categories) are supported.

Logistic regression works by building a set of equations that relate the input field values to the probabilities associated with each of the output field categories. Once the model is generated, it can be used to estimate probabilities for new data. For each record, a probability of membership is computed for each possible output category. The target category with the highest probability is assigned as the predicted output value for that record.

**Binomial example.** A telecommunications provider is concerned about the number of customers it is losing to competitors. Using service usage data, you can create a binomial model to predict which customers are liable to transfer to another provider and customize offers so as to retain as many customers as possible. A binomial model is used because the target has two distinct categories (likely to transfer or not).

**Note:** For binomial models only, string fields are limited to eight characters. If necessary, longer strings can be recoded using a Reclassify node or by using the Anonymize node.

**Multinomial example.** A telecommunications provider has segmented its customer base by service usage patterns, categorizing the customers into four groups. Using demographic data to predict group membership, you can create a multinomial model to classify prospective customers into groups and then customize offers for individual customers.

**Requirements.** One or more input fields and exactly one categorical target field with two or more categories. For a binomial model the target must have a measurement level of *Flag*. For a multinomial model the target can have a measurement level of *Flag*, or of *Nominal* with two or more categories. Fields set to *Both* or *None* are ignored. Fields used in the model must have their types fully instantiated.

**Strengths.** Logistic regression models are often quite accurate. They can handle symbolic and numeric input fields. They can give predicted probabilities for all target categories so that a second-best guess can easily be identified. Logistic models are most effective when group membership is a truly categorical field; if group membership is based on values of a continuous range field (for example, high IQ versus low IQ), you should consider using linear regression to take advantage of the richer information offered by the full range of values. Logistic models can also perform automatic field selection, although other approaches such as tree models or Feature Selection might do this more quickly on large datasets. Finally, since logistic models are well understood by many analysts and data miners, they may be used by some as a baseline against which other modeling techniques can be compared.

When processing large datasets, you can improve performance noticeably by disabling the likelihood-ratio test, an advanced output option. See the topic [“Logistic Regression Advanced Output” on page 166](#) for more information.

**Important:** If temporary disk space is low, Binomial Logistic Regression can fail to build, and will display an error. When building from a large data set (10GB or more), the same amount of free disk space is needed. You can use the environment variable SPSSTMPDIR to set the location of the temporary directory.

## Logistic Node Model Options

**Model name.** You can generate the model name automatically based on the target or ID field (or model type in cases where no such field is specified) or specify a custom name.

**Use partitioned data.** If a partition field is defined, this option ensures that data from only the training partition is used to build the model.

**Create split models.** Builds a separate model for each possible value of input fields that are specified as split fields. See [“Building Split Models”](#) on page 24 for more information.

**Procedure.** Specifies whether a binomial or multinomial model is created. The options available in the dialog box vary depending on which type of modeling procedure is selected.

- **Binomial.** Used when the target field is a flag or nominal field with two discrete values (dichotomous), such as *yes/no*, *on/off*, *male/female*.
- **Multinomial.** Used when the target field is a nominal field with more than two values. You can specify **Main effects**, **Full factorial**, or **Custom**.

**Include constant in equation.** This option determines whether the resulting equations will include a constant term. In most situations, you should leave this option selected.

### Binomial Models

For binomial models, the following methods and options are available:

**Method.** Specify the method to be used in building the logistic regression model.

- **Enter.** This is the default method, which enters all of the terms into the equation directly. No field selection is performed in building the model.
- **Forwards Stepwise.** The Forwards Stepwise method of field selection builds the equation in steps, as the name implies. The initial model is the simplest model possible, with no model terms (except the constant) in the equation. At each step, terms that have not yet been added to the model are evaluated, and if the best of those terms adds significantly to the predictive power of the model, it is added. In addition, terms that are currently in the model are reevaluated to determine if any of them can be removed without significantly detracting from the model. If so, they are removed. The process repeats, and other terms are added and/or removed. When no more terms can be added to improve the model, and no more terms can be removed without detracting from the model, the final model is generated.
- **Backwards Stepwise.** The Backwards Stepwise method is essentially the opposite of the Forwards Stepwise method. With this method, the initial model contains all of the terms as predictors. At each step, terms in the model are evaluated, and any terms that can be removed without significantly detracting from the model are removed. In addition, previously removed terms are reevaluated to determine if the best of those terms adds significantly to the predictive power of the model. If so, it is added back into the model. When no more terms can be removed without significantly detracting from the model, and no more terms can be added to improve the model, the final model is generated.

**Categorical inputs.** Lists the fields that are identified as categorical, that is, those with a measurement level of flag, nominal, or ordinal. You can specify the contrast and base category for each categorical field.

- **Field Name.** This column contains the field names of the categorical inputs. To add continuous or numerical inputs into this column, click the Add Fields icon to the right of the list and select the required inputs.
- **Contrast.** The interpretation of the regression coefficients for a categorical field depends on the contrasts that are used. The contrast determines how hypothesis tests are set up to compare the estimated means. For example, if you know that a categorical field has implicit order, such as a pattern or grouping, you can use the contrast to model that order. The available contrasts are:

**Indicator.** Contrasts indicate the presence or absence of category membership. This is the default method.

**Simple.** Each category of the predictor field, except the reference category, is compared to the reference category.

**Difference.** Each category of the predictor field, except the first category, is compared to the average effect of previous categories. Also known as reverse Helmert contrasts.

**Helmert.** Each category of the predictor field, except the last category, is compared to the average effect of subsequent categories.

**Repeated.** Each category of the predictor field, except the first category, is compared to the category that precedes it.

**Polynomial.** Orthogonal polynomial contrasts. Categories are assumed to be equally spaced. Polynomial contrasts are available for numeric fields only.

**Deviation.** Each category of the predictor field, except the reference category, is compared to the overall effect.

- **Base Category.** Specifies how the reference category is determined for the selected contrast type. Select **First** to use the first category for the input field—sorted alphabetically—or select **Last** to use the last category. The default base category applies to variables that are listed in the **Categorical inputs** area.

**Note:** This field is unavailable if the contrast setting is Difference, Helmert, Repeated, or Polynomial.

The estimate of each field's effect on the overall response is computed as an increase or decrease in the likelihood of each of the other categories relative to the reference category. This can help you identify the fields and values that are more likely to give a specific response.

The base category is shown in the output as 0.0. This is because comparing it to itself produces an empty result. All other categories are shown as equations relevant to the base category. See the topic [“Logistic Nugget Model Details”](#) on page 167 for more information.

## Multinomial Models

For multinomial models the following methods and options are available:

**Method.** Specify the method to be used in building the logistic regression model.

- **Enter.** This is the default method, which enters all of the terms into the equation directly. No field selection is performed in building the model.
- **Stepwise.** The Stepwise method of field selection builds the equation in steps, as the name implies. The initial model is the simplest model possible, with no model terms (except the constant) in the equation. At each step, terms that have not yet been added to the model are evaluated, and if the best of those terms adds significantly to the predictive power of the model, it is added. In addition, terms that are currently in the model are reevaluated to determine if any of them can be removed without significantly detracting from the model. If so, they are removed. The process repeats, and other terms are added and/or removed. When no more terms can be added to improve the model, and no more terms can be removed without detracting from the model, the final model is generated.
- **Forwards.** The Forwards method of field selection is similar to the Stepwise method in that the model is built in steps. However, with this method, the initial model is the simplest model, and only the constant and terms can be added to the model. At each step, terms not yet in the model are tested based on how much they would improve the model, and the best of those terms is added to the model. When no more terms can be added, or the best candidate term does not produce a large-enough improvement in the model, the final model is generated.
- **Backwards.** The Backwards method is essentially the opposite of the Forwards method. With this method, the initial model contains all of the terms as predictors, and terms can only be removed from the model. Model terms that contribute little to the model are removed one by one until no more terms can be removed without significantly worsening the model, yielding the final model.
- **Backwards Stepwise.** The Backwards Stepwise method is essentially the opposite of the Stepwise method. With this method, the initial model contains all of the terms as predictors. At each step, terms in the model are evaluated, and any terms that can be removed without significantly detracting from the model are removed. In addition, previously removed terms are reevaluated to determine if the best of those terms adds significantly to the predictive power of the model. If so, it is added back into the



model. When no more terms can be removed without significantly detracting from the model, and no more terms can be added to improve the model, the final model is generated.

**Note:** The automatic methods, including Stepwise, Forwards, and Backwards, are highly adaptable learning methods and have a strong tendency to overfit the training data. When using these methods, it is especially important to verify the validity of the resulting model either with new data or a hold-out test sample created using the Partition node.

**Base category for target.** Specifies how the reference category is determined. This is used as the baseline against which the regression equations for all other categories in the target are estimated. Select **First** to use the first category for the current target field—sorted alphabetically—or select **Last** to use the last category. Alternatively, you can select **Specify** to choose a specific category and select the desired value from the list. Available values can be defined for each field in a Type node.

Often you would specify the category in which you are least interested to be the base category, for example, a loss-leader product. The other categories are then related to this base category in a relative fashion to identify what makes them more likely to be in their own category. This can help you identify the fields and values that are more likely to give a specific response.

The base category is shown in the output as 0.0. This is because comparing it to itself produces an empty result. All other categories are shown as equations relevant to the base category. See the topic [“Logistic Nugget Model Details”](#) on page 167 for more information.

**Model type.** There are three options for defining the terms in the model. **Main Effects** models include only the input fields individually and do not test interactions (multiplicative effects) between input fields. **Full Factorial** models include all interactions as well as the input field main effects. Full factorial models are better able to capture complex relationships but are also much more difficult to interpret and are more likely to suffer from overfitting. Because of the potentially large number of possible combinations, automatic field selection methods (methods other than Enter) are disabled for full factorial models. **Custom** models include only the terms (main effects and interactions) that you specify. When selecting this option, use the Model Terms list to add or remove terms in the model.

**Model Terms.** When building a Custom model, you will need to explicitly specify the terms in the model. The list shows the current set of terms for the model. The buttons on the right side of the Model Terms list enable you to add and remove model terms.

- To add terms to the model, click the *Add new model terms* button.
- To delete terms, select the desired terms and click the *Delete selected model terms* button.

## Adding Terms to a Logistic Regression Model

When requesting a custom logistic regression model, you can add terms to the model by clicking the *Add new model terms* button on the Logistic Regression Model tab. The New Terms dialog box opens in which you can specify terms.

**Type of term to add.** There are several ways to add terms to the model, based on the selection of input fields in the Available fields list.

- **Single interaction.** Inserts the term representing the interaction of all selected fields.
- **Main effects.** Inserts one main effect term (the field itself) for each selected input field.
- **All 2-way interactions.** Inserts a 2-way interaction term (the product of the input fields) for each possible pair of selected input fields. For example, if you have selected input fields *A*, *B*, and *C* in the Available fields list, this method will insert the terms  $A * B$ ,  $A * C$ , and  $B * C$ .
- **All 3-way interactions.** Inserts a 3-way interaction term (the product of the input fields) for each possible combination of selected input fields, taken three at a time. For example, if you have selected input fields *A*, *B*, *C*, and *D* in the Available fields list, this method will insert the terms  $A * B * C$ ,  $A * B * D$ ,  $A * C * D$ , and  $B * C * D$ .
- **All 4-way interactions.** Inserts a 4-way interaction term (the product of the input fields) for each possible combination of selected input fields, taken four at a time. For example, if you have selected input fields *A*, *B*, *C*, *D*, and *E* in the Available fields list, this method will insert the terms  $A * B * C * D$ ,  $A * B * C * E$ ,  $A * B * D * E$ ,  $A * C * D * E$ , and  $B * C * D * E$ .



**Available fields.** Lists the available input fields to be used in constructing model terms.

**Preview.** Shows the terms that will be added to the model if you click **Insert**, based on the selected fields and term type.

**Insert.** Inserts terms in the model (based on the current selection of fields and term type) and closes the dialog box.

## Logistic Node Expert Options

If you have detailed knowledge of logistic regression, expert options enable you to fine-tune the training process. To access expert options, set Mode to **Expert** on the Expert tab.

**Scale (Multinomial models only).** You can specify a dispersion scaling value that will be used to correct the estimate of the parameter covariance matrix. **Pearson** estimates the scaling value by using the Pearson chi-square statistic. **Deviance** estimates the scaling value by using the deviance function (likelihood-ratio chi-square) statistic. You can also specify your own user-defined scaling value. It must be a positive numeric value.

**Append all probabilities.** If this option is selected, probabilities for each category of the output field will be added to each record processed by the node. If this option is not selected, the probability of only the predicted category is added.

For example, a table containing the results of a multinomial model with three categories will include five new columns. One column will list the probability of the outcome being correctly predicted, the next column will show the probability that this prediction is a hit or miss, and a further three columns will show the probability that each category's prediction is a miss or hit. See the topic [“Logistic Model Nugget”](#) on page 167 for more information.

*Note:* This option is always selected for binomial models.

**Singularity tolerance.** Specify the tolerance used in checking for singularities.

**Convergence.** These options enable you to control the parameters for model convergence. When you execute the model, the convergence settings control how many times the different parameters are repeatedly run through to see how well they fit. The more often the parameters are tried, the closer the results will be (that is, the results will converge). See the topic [“Logistic Regression Convergence Options”](#) on page 165 for more information.

**Output.** These options enable you to request additional statistics that will be displayed in the advanced output of the model nugget built by the node. See the topic [“Logistic Regression Advanced Output”](#) on page 166 for more information.

**Stepping.** These options enable you to control the criteria for adding and removing fields with the Stepwise, Forwards, Backwards, or Backwards Stepwise estimation methods. (The button is disabled if the Enter method is selected.) See the topic [“Logistic Regression Stepping Options”](#) on page 166 for more information.

## Logistic Regression Convergence Options

You can set the convergence parameters for logistic regression model estimation.

**Maximum iterations.** Specify the maximum number of iterations for estimating the model.

**Maximum step-halving.** Step-halving is a technique used by logistic regression to deal with complexities in the estimation process. Under normal circumstances, you should use the default setting.

**Log-likelihood convergence.** Iterations stop if the relative change in the log-likelihood is less than this value. The criterion is not used if the value is 0.

**Parameter convergence.** Iterations stop if the absolute change or relative change in the parameter estimates is less than this value. The criterion is not used if the value is 0.

**Delta (Multinomial models only).** You can specify a value between 0 and 1 to be added to each empty cell (combination of input field and output field values). This can help the estimation algorithm deal with

data where there are many possible combinations of field values relative to the number of records in the data. The default is 0.

## Logistic Regression Advanced Output

Select the optional output you want to display in the advanced output of the Regression model nugget. To view the advanced output, browse the model nugget and click the **Advanced** tab. See the topic [“Logistic Model Nugget Advanced Output”](#) on page 169 for more information.

### Binomial Options

Select the types of output to be generated for the model. See the topic [“Logistic Model Nugget Advanced Output”](#) on page 169 for more information.

**Display.** Select whether to display the results at each step, or to wait until all steps have been worked through.

**CI for exp(B).** Select the confidence intervals for each coefficient (shown as Beta) in the expression. Specify the level of the confidence interval (the default is 95%).

**Residual Diagnosis.** Requests a Casewise Diagnostics table of residuals.

- **Outliers outside (std. dev.).** List only residual cases for which the absolute standardized value of the listed variable is at least as large as the value you specify. The default value is 2.
- **All cases.** Include all cases in the Casewise Diagnostic table of residuals.

*Note:* Because this option lists each of the input records, it may result in an exceptionally large table in the report, with one line for every record.

**Classification cutoff.** This enables you to determine the cutpoint for classifying cases. Cases with predicted values that exceed the classification cutoff are classified as positive, while those with predicted values smaller than the cutoff are classified as negative. To change the default, enter a value between 0.01 and 0.99.

### Multinomial Options

Select the types of output to be generated for the model. See the topic [“Logistic Model Nugget Advanced Output”](#) on page 169 for more information.

*Note:* Selecting the **Likelihood ratio tests** option greatly increases the processing time required to build a logistic regression model. If your model is taking too long to build, consider disabling this option or utilize the Wald and Score statistics instead. See the topic [“Logistic Regression Stepping Options”](#) on page 166 for more information.

**Iteration history for every.** Select the step interval for printing iteration status in the advanced output.

**Confidence Interval.** The confidence intervals for coefficients in the equations. Specify the level of the confidence interval (the default is 95%).

## Logistic Regression Stepping Options

These options enable you to control the criteria for adding and removing fields with the Stepwise, Forwards, Backwards, or Backwards Stepwise estimation methods.

**Number of terms in model (Multinomial models only).** You can specify the minimum number of terms in the model for Backwards and Backwards Stepwise models and the maximum number of terms for Forwards and Stepwise models. If you specify a minimum value greater than 0, the model will include that many terms, even if some of the terms would have been removed based on statistical criteria. The minimum setting is ignored for Forwards, Stepwise, and Enter models. If you specify a maximum, some terms may be omitted from the model, even though they would have been selected based on statistical criteria. The **Specify Maximum** setting is ignored for Backwards, Backwards Stepwise, and Enter models.

**Entry criterion (Multinomial models only).** Select **Score** to maximize speed of processing. The **Likelihood Ratio** option may provide somewhat more robust estimates but take longer to compute. The default setting is to use the Score statistic.

**Removal criterion.** Select **Likelihood Ratio** for a more robust model. To shorten the time required to build the model, you can try selecting **Wald**. However, if you have complete or quasi-complete separation in the data (which you can determine by using the Advanced tab on the model nugget), the Wald statistic becomes particularly unreliable and should not be used. The default setting is to use the likelihood-ratio statistic. For binomial models, there is the additional option **Conditional**. This provides removal testing based on the probability of the likelihood-ratio statistic based on conditional parameter estimates.

**Significance thresholds for criteria.** This option enables you to specify selection criteria based on the statistical probability (the  $p$  value) associated with each field. Fields will be added to the model only if the associated  $p$  value is smaller than the **Entry** value and will be removed only if the  $p$  value is larger than the **Removal** value. The **Entry** value must be smaller than the **Removal** value.

**Requirements for entry or removal (Multinomial models only).** For some applications, it doesn't make mathematical sense to add interaction terms to the model unless the model also contains the lower-order terms for the fields involved in the interaction term. For example, it may not make sense to include  $A * B$  in the model unless  $A$  and  $B$  are also included in the model. These options let you determine how such dependencies are handled during stepwise term selection.

- **Hierarchy for discrete effects.** Higher-order effects (interactions involving more fields) will enter the model only if all lower-order effects (main effects or interactions involving fewer fields) for the relevant fields are already in the model, and lower-order effects will not be removed if higher-order effects involving the same fields are in the model. This option applies only to categorical fields.
- **Hierarchy for all effects.** This option works in the same way as the previous option, except that it applies to all input fields.
- **Containment for all effects.** Effects can be included in the model only if all of the effects contained in the effect are also included in the model. This option is similar to the **Hierarchy for all effects** option except that continuous fields are treated somewhat differently. For an effect to contain another effect, the contained (lower-order) effect must include *all* of the continuous fields involved in the containing (higher-order) effect, and the contained effect's categorical fields must be a subset of those in the containing effect. For example, if  $A$  and  $B$  are categorical fields and  $X$  is a continuous field, the term  $A * B * X$  contains the terms  $A * X$  and  $B * X$ .
- **None.** No relationships are enforced; terms are added to and removed from the model independently.

## Logistic Model Nugget

---

A Logistic model nugget represents the equation estimated by a Logistic node. It contains all of the information captured by the logistic regression model, as well as information about the model structure and performance. This type of equation may also be generated by other models such as Oracle SVM.

When you run a stream containing a Logistic model nugget, the node adds two new fields containing the model's prediction and the associated probability. The names of the new fields are derived from the name of the output field being predicted, prefixed with  $\$L$ - for the predicted category and  $\$LP$ - for the associated probability. For example, for an output field named *colorpref*, the new fields would be named  $\$L$ -*colorpref* and  $\$LP$ -*colorpref*. In addition, if you have selected the **Append all probabilities** option in the Logistic node, an additional field will be added for each category of the output field, containing the probability belonging to the corresponding category for each record. These additional fields are named based on the values of the output field, prefixed by  $\$LP$ -. For example, if the legal values of *colorpref* are *Red*, *Green*, and *Blue*, three new fields will be added:  $\$LP$ -*Red*,  $\$LP$ -*Green*, and  $\$LP$ -*Blue*.

**Generating a Filter node.** The Generate menu enables you to create a new Filter node to pass input fields based on the results of the model. Fields that are dropped from the model due to multicollinearity will be filtered by the generated node, as well as fields not used in the model.

## Logistic Nugget Model Details

For multinomial models, the Model tab in a Logistic model nugget has a split display with model equations in the left pane, and predictor importance on the right. For binomial models, the tab displays predictor importance only. See the topic [“Predictor Importance” on page 37](#) for more information.

Model Equations

For multinomial models, the left pane displays the actual equations estimated for the logistic regression model. There is one equation for each category in the target field, except the baseline category. The equations are displayed in a tree format. This type of equation may also be generated by certain other models such as Oracle SVM.

**Equation For.** Shows the regression equations used to derive the target category probabilities, given a set of predictor values. The last category of the target field is considered the **baseline category**; the equations shown give the log-odds for the other target categories relative to the baseline category for a particular set of predictor values. The predicted probability for each category of the given predictor pattern is derived from these log-odds values.

#### How Probabilities Are Calculated

Each equation calculates the log-odds for a particular target category, relative to the baseline category. The **log-odds**, also called the **logit**, is the ratio of the probability for the specified target category to that of the baseline category, with the natural logarithm function applied to the result. For the baseline category, the odds of the category relative to itself is 1.0, and thus the log-odds is 0. You can think of this as an implicit equation for the baseline category where all coefficients are 0.

To derive the probability from the log-odds for a particular target category, you take the logit value calculated by the equation for that category and apply the following formula:

$$P(\text{group } i) = \exp(g_i) / \sum_k \exp(g_k)$$

where  $g$  is the calculated log-odds,  $i$  is the category index, and  $k$  goes from 1 to the number of target categories.

#### Predictor Importance

Optionally, a chart that indicates the relative importance of each predictor in estimating the model may also be displayed on the Model tab. Typically you will want to focus your modeling efforts on the predictors that matter most and consider dropping or ignoring those that matter least. Note this chart is only available if **Calculate predictor importance** is selected on the Analyze tab before generating the model. See the topic [“Predictor Importance” on page 37](#) for more information.

*Note:* Predictor importance may take longer to calculate for logistic regression than for other types of models, and is not selected on the Analyze tab by default. Selecting this option may slow performance, particularly with large datasets.

## Logistic Model Nugget Summary

The summary for a logistic regression model displays the fields and settings used to generate the model. In addition, if you have executed an Analysis node attached to this modeling node, information from that analysis will also be displayed in this section. For general information on using the model browser, see [“Browsing model nuggets” on page 35](#).

## Logistic Model Nugget Settings

The Settings tab in a Logistic model nugget specifies options for confidences, probabilities, propensity scores, and SQL generation during model scoring. This tab is only available after the model nugget has been added to a stream and displays different options depending on the type of model and target.

### Multinomial Models

For multinomial models, the following options are available.

**Calculate confidences** Specifies whether confidences are calculated during scoring.

**Calculate raw propensity scores (flag targets only)** For models with flag targets only, you can request raw propensity scores that indicate the likelihood of the true outcome specified for the target field. These are in addition to standard prediction and confidence values. Adjusted propensity scores are not available. See the topic [“Modeling Node Analyze Options” on page 29](#) for more information.

**Append all probabilities** Specifies whether probabilities for each category of the output field are added to each record processed by the node. If this option is not selected, the probability of only the predicted category is added. For a nominal target with three categories, for example, the scoring output will include a column for each of the three categories, plus a fourth column indicating the probability for whichever category is predicted. For example if the probabilities for categories *Red*, *Green*, and *Blue* are 0.6, 0.3, and 0.1 respectively, the predicted category would be *Red*, with a probability of 0.6.

**Generate SQL for this model** When using data from a database, SQL code can be pushed back to the database for execution, providing superior performance for many operations.

Select one of the following options to specify how SQL generation is performed.

- **Default: Score using Server Scoring Adapter (if installed) otherwise in process** If connected to a database with a scoring adapter installed, generates SQL using the scoring adapter and associated user defined functions (UDF) and scores your model within the database. When no scoring adapter is available, this option fetches your data back from the database and scores it in SPSS Modeler.
- **Score by converting to native SQL** If selected, generates native SQL to score the model within the database.

**Note:** Although this option can provide quicker results, the size and complexity of the native SQL increases as the complexity of the model increases.

- **Score outside of the Database** If selected, this option fetches your data back from the database and scores it in SPSS Modeler.

**Note:** For multinomial models, SQL generation is unavailable if **Append all probabilities** has been selected, or—for models with nominal targets—if **Calculate confidences** has been selected. SQL generation with confidence calculations is supported for multinomial models with flag targets only. SQL generation is not available for binomial models.

### Binomial Models

For binomial models, confidences and probabilities are always enabled, and the settings that would enable you to disable these options are not available. SQL generation is not available for binomial models. The only setting that can be changed for binomial models is the ability to calculate raw propensity scores. As noted earlier for multinomial models, this applies to models with flag targets only. See the topic [“Modeling Node Analyze Options” on page 29](#) for more information.

## Logistic Model Nugget Advanced Output

The advanced output for logistic regression (also known as **nominal regression**) gives detailed information about the estimated model and its performance. Most of the information contained in the advanced output is quite technical, and extensive knowledge of logistic regression analysis is required to properly interpret this output.

**Warnings.** Indicates any warnings or potential problems with the results.

**Case processing summary.** Lists the number of records processed, broken down by each symbolic field in the model.

**Step summary (optional).** Lists the effects added or removed at each step of model creation, when using automatic field selection.

*Note:* Only shown for the Stepwise, Forwards, Backwards, or Backwards Stepwise methods.

**Iteration history (optional).** Shows the iteration history of parameter estimates for every  $n$  iterations beginning with the initial estimates, where  $n$  is the value of the print interval. The default is to print every iteration ( $n=1$ ).

**Model fitting information (Multinomial models).** Shows the likelihood-ratio test of your model (Final) against one in which all of the parameter coefficients are 0 (Intercept Only).

**Classification (optional).** Shows the matrix of predicted and actual output field values with percentages.

**Goodness-of-fit chi-square statistics (optional).** Shows Pearson's and likelihood-ratio chi-square statistics. These statistics test the overall fit of the model to the training data.

**Hosmer and Lemeshow goodness-of-fit (optional).** Shows the results of grouping cases into deciles of risk and comparing the observed probability with the expected probability within each decile. This goodness-of-fit statistic is more robust than the traditional goodness-of-fit statistic used in multinomial models, particularly for models with continuous covariates and studies with small sample sizes.

**Pseudo R-square (optional).** Shows the Cox and Snell, Nagelkerke, and McFadden *R*-square measures of model fit. These statistics are in some ways analogous to the *R*-square statistic in linear regression.

**Monotonicity measures (optional).** Shows the number of concordant pairs, discordant pairs, and tied pairs in the data, as well as the percentage of the total number of pairs that each represents. The Somers' D, Goodman and Kruskal's Gamma, Kendall's tau-a, and Concordance Index C are also displayed in this table.

**Information criteria (optional).** Shows Akaike's information criterion (AIC) and Schwarz's Bayesian information criterion (BIC).

**Likelihood ratio tests (optional).** Shows statistics testing of whether the coefficients of the model effects are statistically different from 0. Significant input fields are those with very small significance levels in the output (labeled *Sig.*).

**Parameter estimates (optional).** Shows estimates of the equation coefficients, tests of those coefficients, odds ratios derived from the coefficients labeled *Exp(B)*, and confidence intervals for the odds ratios.

**Asymptotic covariance/correlation matrix (optional).** Shows the asymptotic covariances and/or correlations of the coefficient estimates.

**Observed and predicted frequencies (optional).** For each covariate pattern, shows the observed and predicted frequencies for each output field value. This table can be quite large, especially for models with numeric input fields. If the resulting table would be too large to be practical, it is omitted, and a warning is displayed.

## PCA/Factor Node

---

The PCA/Factor node provides powerful data-reduction techniques to reduce the complexity of your data. Two similar but distinct approaches are provided.

- **Principal components analysis (PCA)** finds linear combinations of the input fields that do the best job of capturing the variance in the entire set of fields, where the components are orthogonal (perpendicular) to each other. PCA focuses on all variance, including both shared and unique variance.
- **Factor analysis** attempts to identify underlying concepts, or **factors**, that explain the pattern of correlations within a set of observed fields. Factor analysis focuses on shared variance only. Variance that is unique to specific fields is not considered in estimating the model. Several methods of factor analysis are provided by the Factor/PCA node.

For both approaches, the goal is to find a small number of derived fields that effectively summarize the information in the original set of fields.

**Requirements.** Only numeric fields can be used in a PCA-Factor model. To estimate a factor analysis or PCA, you need one or more fields with the role set to *Input* fields. Fields with the role set to *Target*, *Both*, or *None* are ignored, as are non-numeric fields.

**Strengths.** Factor analysis and PCA can effectively reduce the complexity of your data without sacrificing much of the information content. These techniques can help you build more robust models that execute more quickly than would be possible with the raw input fields.

### PCA/Factor Node Model Options

**Model name.** You can generate the model name automatically based on the target or ID field (or model type in cases where no such field is specified) or specify a custom name.



**Use partitioned data.** If a partition field is defined, this option ensures that data from only the training partition is used to build the model.

**Extraction Method.** Specify the method to be used for data reduction.

- **Principal Components.** This is the default method, which uses PCA to find components that summarize the input fields.
- **Unweighted Least Squares.** This factor analysis method works by finding the set of factors that is best able to reproduce the pattern of relationships (correlations) among the input fields.
- **Generalized Least Squares.** This factor analysis method is similar to unweighted least squares, except that it uses weighting to de-emphasize fields with a lot of unique (unshared) variance.
- **Maximum Likelihood.** This factor analysis method produces factor equations that are most likely to have produced the observed pattern of relationships (correlations) in the input fields, based on assumptions about the form of those relationships. Specifically, the method assumes that the training data follow a multivariate normal distribution.
- **Principal Axis Factoring.** This factor analysis method is very similar to the principal components method, except that it focuses on shared variance only.
- **Alpha Factoring.** This factor analysis method considers the fields in the analysis to be a sample from the universe of potential input fields. It maximizes the statistical reliability of the factors.
- **Image Factoring.** This factor analysis method uses data estimation to isolate the common variance and find factors that describe it.

## PCA/Factor Node Expert Options

If you have detailed knowledge of factor analysis and PCA, expert options enable you to fine-tune the training process. To access expert options, set Mode to **Expert** on the Expert tab.

**Missing values.** By default, IBM SPSS Modeler only uses records that have valid values for all fields used in the model. (This is sometimes called **listwise deletion** of missing values.) If you have a lot of missing data, you may find that this approach eliminates too many records, leaving you without enough data to generate a good model. In such cases, you can deselect the **Only use complete records** option. IBM SPSS Modeler then attempts to use as much information as possible to estimate the model, including records where some of the fields have missing values. (This is sometimes called **pairwise deletion** of missing values.) However, in some situations, using incomplete records in this manner can lead to computational problems in estimating the model.

**Fields.** Specify whether to use the correlation matrix (the default) or the covariance matrix of the input fields in estimating the model.

**Maximum iterations for convergence.** Specify the maximum number of iterations for estimating the model.

**Extract factors.** There are two ways to select the number of factors to extract from the input fields.

- **Eigenvalues over.** This option will retain all factors or components with eigenvalues larger than the specified criterion. **Eigenvalues** measure the ability of each factor or component to summarize variance in the set of input fields. The model will retain all factors or components with eigenvalues greater than the specified value when using the correlation matrix. When using the covariance matrix, the criterion is the specified value times the mean eigenvalue. That scaling gives this option a similar meaning for both types of matrix.
- **Maximum number.** This option will retain the specified number of factors or components in descending order of eigenvalues. In other words, the factors or components corresponding to the  $n$  highest eigenvalues are retained, where  $n$  is the specified criterion. The default extraction criterion is five factors/components.

**Component/factor matrix format.** These options control the format of the factor matrix (or component matrix for PCA models).

- **Sort values.** If this option is selected, factor loadings in the model output will be sorted numerically.

- **Hide values below.** If this option is selected, scores below the specified threshold will be hidden in the matrix to make it easier to see the pattern in the matrix.

**Rotation.** These options enable you to control the rotation method for the model. See the topic [“PCA/Factor Node Rotation Options”](#) on page 172 for more information.

## PCA/Factor Node Rotation Options

In many cases, mathematically rotating the set of retained factors can increase their usefulness and especially their interpretability. Select a rotation method:

- **No rotation.** The default option. No rotation is used.
- **Varimax.** An orthogonal rotation method that minimizes the number of fields with high loadings on each factor. It simplifies the interpretation of the factors.
- **Direct oblimin.** A method for oblique (non-orthogonal) rotation. When **Delta** equals 0 (the default), solutions are oblique. As delta becomes more negative, the factors become less oblique. To override the default delta of 0, enter a number less than or equal to 0.8.
- **Quartimax.** An orthogonal method that minimizes the number of factors needed to explain each field. It simplifies the interpretation of the observed fields.
- **Equamax.** A rotation method that is a combination of the Varimax method, which simplifies the factors, and the Quartimax method, which simplifies the fields. The number of fields that load highly on a factor and the number of factors needed to explain a field are minimized.
- **Promax.** An oblique rotation, which enables factors to be correlated. It can be calculated more quickly than a direct oblimin rotation, so it can be useful for large datasets. **Kappa** controls the obliqueness of the solution (the extent to which factors can be correlated).

## PCA/Factor Model Nugget

A PCA/Factor model nugget represents the factor analysis and principal component analysis (PCA) model created by a PCA/Factor node. They contain all of the information captured by the trained model, as well as information about the model's performance and characteristics.

When you run a stream containing a factor equation model, the node adds a new field for each factor or component in the model. The new field names are derived from the model name, prefixed by *\$F-* and suffixed by *-n*, where *n* is the number of the factor or component. For example, if your model is named *Factor* and contains three factors, the new fields would be named *\$F-Factor-1*, *\$F-Factor-2*, and *\$F-Factor-3*.

To get a better sense of what the factor model has encoded, you can do some more downstream analysis. A useful way to view the result of the factor model is to view the correlations between factors and input fields using a Statistics node. This shows you which input fields load heavily on which factors and can help you discover if your factors have any underlying meaning or interpretation.

You can also assess the factor model by using the information available in the advanced output. To view the advanced output, click the **Advanced** tab of the model nugget browser. The advanced output contains a lot of detailed information and is meant for users with extensive knowledge of factor analysis or PCA. See the topic [“PCA/Factor Model Nugget Advanced Output”](#) on page 173 for more information.

## PCA/Factor Model Nugget Equations

The Model tab for a Factor model nugget displays the factor score equation for each factor. Factor or component scores are calculated by multiplying each input field value by its coefficient and summing the results.

## PCA/Factor Model Nugget Summary

The Summary tab for a factor model displays the number of factors retained in the factor/PCA model, along with additional information on the fields and settings used to generate the model. See the topic [“Browsing model nuggets”](#) on page 35 for more information.



## PCA/Factor Model Nugget Advanced Output

The advanced output for factor analysis gives detailed information on the estimated model and its performance. Most of the information contained in the advanced output is quite technical, and extensive knowledge of factor analysis is required to properly interpret this output.

**Warnings.** Indicates any warnings or potential problems with the results.

**Communalities.** Shows the proportion of each field's variance that is accounted for by the factors or components. *Initial* gives the initial communalities with the full set of factors (the model starts with as many factors as input fields), and *Extraction* gives the communalities based on the retained set of factors.

**Total variance explained.** Shows the total variance explained by the factors in the model. *Initial Eigenvalues* shows the variance explained by the full set of initial factors. *Extraction Sums of Squared Loadings* shows the variance explained by factors retained in the model. *Rotation Sums of Squared Loadings* shows the variance explained by the rotated factors. Note that for oblique rotations, *Rotation Sums of Squared Loadings* shows only the sums of squared loadings and does not show variance percentages.

**Factor (or component) matrix.** Shows correlations between input fields and unrotated factors.

**Rotated factor (or component) matrix.** Shows correlations between input fields and rotated factors for orthogonal rotations.

**Pattern matrix.** Shows the partial correlations between input fields and rotated factors for oblique rotations.

**Structure matrix.** Shows the simple correlations between input fields and rotated factors for oblique rotations.

**Factor correlation matrix.** Shows correlations among factors for oblique rotations.

## Discriminant node

---

Discriminant analysis builds a predictive model for group membership. The model is composed of a discriminant function (or, for more than two groups, a set of discriminant functions) based on linear combinations of the predictor variables that provide the best discrimination between the groups. The functions are generated from a sample of cases for which group membership is known; the functions can then be applied to new cases that have measurements for the predictor variables but have unknown group membership.

**Example.** A telecommunications company can use discriminant analysis to classify customers into groups based on usage data. This allows them to score potential customers and target those who are most likely to be in the most valuable groups.

**Requirements.** You need one or more input fields and exactly one target field. The target must be a categorical field (with a measurement level of *Flag* or *Nominal*) with string or integer storage. (Storage can be converted using a Filler or Derive node if necessary. ) Fields set to *Both* or *None* are ignored. Fields used in the model must have their types fully instantiated.

**Strengths.** Discriminant analysis and Logistic Regression are both suitable classification models. However, Discriminant analysis makes more assumptions about the input fields—for example, they are normally distributed and should be continuous, and they give better results if those requirements are met, especially if the sample size is small.

## Discriminant Node Model Options

**Model name.** You can generate the model name automatically based on the target or ID field (or model type in cases where no such field is specified) or specify a custom name.

**Use partitioned data.** If a partition field is defined, this option ensures that data from only the training partition is used to build the model.

**Create split models.** Builds a separate model for each possible value of input fields that are specified as split fields. See [“Building Split Models” on page 24](#) for more information.

**Method.** The following options are available for entering predictors into the model:

- **Enter.** This is the default method, which enters all of the terms into the equation directly. Terms that do not add significantly to the predictive power of the model are not added.
- **Stepwise.** The initial model is the simplest model possible, with no model terms (except the constant) in the equation. At each step, terms that have not yet been added to the model are evaluated, and if the best of those terms adds significantly to the predictive power of the model, it is added.

*Note:* The Stepwise method has a strong tendency to overfit the training data. When using these methods, it is especially important to verify the validity of the resulting model with a hold-out test sample or new data.

## Discriminant Node Expert Options

If you have detailed knowledge of discriminant analysis, expert options allow you to fine-tune the training process. To access expert options, set **Mode** to **Expert** on the Expert tab.

**Prior Probabilities.** This option determines whether the classification coefficients are adjusted for a priori knowledge of group membership.

- **All groups equal.** Equal prior probabilities are assumed for all groups; this has no effect on the coefficients.
- **Compute from group sizes.** The observed group sizes in your sample determine the prior probabilities of group membership. For example, if 50% of the observations included in the analysis fall into the first group, 25% in the second, and 25% in the third, the classification coefficients are adjusted to increase the likelihood of membership in the first group relative to the other two.

**Use Covariance Matrix.** You can choose to classify cases using a within-groups covariance matrix or a separate-groups covariance matrix.

- *Within-groups.* The pooled within-groups covariance matrix is used to classify cases.
- *Separate-groups.* Separate-groups covariance matrices are used for classification. Because classification is based on the discriminant functions (not based on the original variables), this option is not always equivalent to quadratic discrimination.

**Output.** These options allow you to request additional statistics that will be displayed in the advanced output of the model nugget built by the node. See the topic [“Discriminant Node Output Options” on page 174](#) for more information.

**Stepping.** These options allow you to control the criteria for adding and removing fields with the Stepwise estimation method. (The button is disabled if the Enter method is selected.) See the topic [“Discriminant Node Stepping Options” on page 175](#) for more information.

## Discriminant Node Output Options

Select the optional output you want to display in the advanced output of the logistic regression model nugget. To view the advanced output, browse the model nugget and click the **Advanced** tab. See the topic [“Discriminant Model Nugget Advanced Output” on page 176](#) for more information.

**Descriptives.** Available options are means (including standard deviations), univariate ANOVAs, and Box's *M* test.

- *Means.* Displays total and group means, as well as standard deviations for the independent variables.
- *Univariate ANOVAs.* Performs a one-way analysis-of-variance test for equality of group means for each independent variable.
- *Box's M.* A test for the equality of the group covariance matrices. For sufficiently large samples, a nonsignificant p value means there is insufficient evidence that the matrices differ. The test is sensitive to departures from multivariate normality.

**Function Coefficients.** Available options are Fisher's classification coefficients and unstandardized coefficients.

- *Fisher's*. Displays Fisher's classification function coefficients that can be used directly for classification. A separate set of classification function coefficients is obtained for each group, and a case is assigned to the group for which it has the largest discriminant score (classification function value).
- *Unstandardized*. Displays the unstandardized discriminant function coefficients.

**Matrices.** Available matrices of coefficients for independent variables are within-groups correlation matrix, within-groups covariance matrix, separate-groups covariance matrix, and total covariance matrix.

- *Within-groups correlation*. Displays a pooled within-groups correlation matrix that is obtained by averaging the separate covariance matrices for all groups before computing the correlations.
- *Within-groups covariance*. Displays a pooled within-groups covariance matrix, which may differ from the total covariance matrix. The matrix is obtained by averaging the separate covariance matrices for all groups.
- *Separate-groups covariance*. Displays separate covariance matrices for each group.
- *Total covariance*. Displays a covariance matrix from all cases as if they were from a single sample.

**Classification.** The following output pertains to the classification results.

- *Casewise results*. Codes for actual group, predicted group, posterior probabilities, and discriminant scores are displayed for each case.
- *Summary table*. The number of cases correctly and incorrectly assigned to each of the groups based on the discriminant analysis. Sometimes called the "Confusion Matrix."
- *Leave-one-out classification*. Each case in the analysis is classified by the functions derived from all cases other than that case. It is also known as the "U-method."
- *Territorial map*. A plot of the boundaries used to classify cases into groups based on function values. The numbers correspond to groups into which cases are classified. The mean for each group is indicated by an asterisk within its boundaries. The map is not displayed if there is only one discriminant function.
- *Combined-groups*. Creates an all-groups scatterplot of the first two discriminant function values. If there is only one function, a histogram is displayed instead.
- *Separate-groups*. Creates separate-group scatterplots of the first two discriminant function values. If there is only one function, histograms are displayed instead.

**Stepwise. Summary of Steps** displays statistics for all variables after each step; **F for pairwise distances** displays a matrix of pairwise *F* ratios for each pair of groups. The *F* ratios can be used for significance tests of the Mahalanobis distances between groups.

## Discriminant Node Stepping Options

**Method.** Select the statistic to be used for entering or removing new variables. Available alternatives are Wilks' lambda, unexplained variance, Mahalanobis distance, smallest *F* ratio, and Rao's *V*. With Rao's *V*, you can specify the minimum increase in *V* for a variable to enter.

- *Wilks' lambda*. A variable selection method for stepwise discriminant analysis that chooses variables for entry into the equation on the basis of how much they lower Wilks' lambda. At each step, the variable that minimizes the overall Wilks' lambda is entered.
- *Unexplained variance*. At each step, the variable that minimizes the sum of the unexplained variation between groups is entered.
- *Mahalanobis distance*. A measure of how much a case's values on the independent variables differ from the average of all cases. A large Mahalanobis distance identifies a case as having extreme values on one or more of the independent variables.
- *Smallest F ratio*. A method of variable selection in stepwise analysis based on maximizing an *F* ratio computed from the Mahalanobis distance between groups.
- *Rao's V*. A measure of the differences between group means. Also called the Lawley-Hotelling trace. At each step, the variable that maximizes the increase in Rao's *V* is entered. After selecting this option, enter the minimum value a variable must have to enter the analysis.

**Criteria.** Available alternatives are **Use F value** and **Use probability of F**. Enter values for entering and removing variables.

- *Use F value.* A variable is entered into the model if its F value is greater than the Entry value and is removed if the F value is less than the Removal value. Entry must be greater than Removal, and both values must be positive. To enter more variables into the model, lower the Entry value. To remove more variables from the model, increase the Removal value.
- *Use probability of F.* A variable is entered into the model if the significance level of its F value is less than the Entry value and is removed if the significance level is greater than the Removal value. Entry must be less than Removal, and both values must be positive. To enter more variables into the model, increase the Entry value. To remove more variables from the model, lower the Removal value.

## Discriminant Model Nugget

Discriminant model nuggets represent the equations estimated by Discriminant nodes. They contain all of the information captured by the discriminant model, as well as information about the model structure and performance.

When you run a stream containing a Discriminant model nugget, the node adds two new fields containing the model's prediction and the associated probability. The names of the new fields are derived from the name of the output field being predicted, prefixed with *\$D-* for the predicted category and *\$DP-* for the associated probability. For example, for an output field named *colorpref*, the new fields would be named *\$D-colorpref* and *\$DP-colorpref*.

**Generating a Filter node.** The Generate menu allows you to create a new Filter node to pass input fields based on the results of the model.

### Predictor Importance

Optionally, a chart that indicates the relative importance of each predictor in estimating the model may also be displayed on the Model tab. Typically you will want to focus your modeling efforts on the predictors that matter most and consider dropping or ignoring those that matter least. Note this chart is only available if **Calculate predictor importance** is selected on the Analyze tab before generating the model. See the topic [“Predictor Importance” on page 37](#) for more information.

### Discriminant Model Nugget Advanced Output

The advanced output for discriminant analysis gives detailed information about the estimated model and its performance. Most of the information contained in the advanced output is quite technical, and extensive knowledge of discriminant analysis is required to properly interpret this output. See the topic [“Discriminant Node Output Options” on page 174](#) for more information.

### Discriminant Model Nugget Settings

The Settings tab in a Discriminant model nugget allows you to obtain propensity scores when scoring the model. This tab is available for models with flag targets only, and only after the model nugget has been added to a stream.

**Calculate raw propensity scores.** For models with a flag target (which return a yes or no prediction), you can request propensity scores that indicate the likelihood of the true outcome specified for the target field. These are in addition to other prediction and confidence values that may be generated during scoring.

**Calculate adjusted propensity scores.** Raw propensity scores are based only on the training data and may be overly optimistic due to the tendency of many models to overfit this data. Adjusted propensities attempt to compensate by evaluating model performance against a test or validation partition. This option requires that a partition field be defined in the stream and adjusted propensity scores be enabled in the modeling node before generating the model.

**Generate SQL for this model** When using data from a database, SQL code can be pushed back to the database for execution, providing superior performance for many operations.

Select one of the following options to specify how SQL generation is performed.

- **Default: Score using Server Scoring Adapter (if installed) otherwise in process** If connected to a database with a scoring adapter installed, generates SQL using the scoring adapter and associated user defined functions (UDF) and scores your model within the database. When no scoring adapter is available, this option fetches your data back from the database and scores it in SPSS Modeler.
- **Score outside of the Database** If selected, this option fetches your data back from the database and scores it in SPSS Modeler.

### Discriminant Model Nugget Summary

The Summary tab for a Discriminant model nugget displays the fields and settings used to generate the model. In addition, if you have executed an Analysis node attached to this modeling node, information from that analysis will also be displayed in this section. For general information on using the model browser, see [“Browsing model nuggets” on page 35](#).

## GenLin Node

---

The generalized linear model expands the general linear model so that the dependent variable is linearly related to the factors and covariates via a specified link function. Moreover, the model allows for the dependent variable to have a non-normal distribution. It covers widely used statistical models, such as linear regression for normally distributed responses, logistic models for binary data, loglinear models for count data, complementary log-log models for interval-censored survival data, plus many other statistical models through its very general model formulation.

**Examples.** A shipping company can use generalized linear models to fit a Poisson regression to damage counts for several types of ships constructed in different time periods, and the resulting model can help determine which ship types are most prone to damage.

A car insurance company can use generalized linear models to fit a gamma regression to damage claims for cars, and the resulting model can help determine the factors that contribute the most to claim size.

Medical researchers can use generalized linear models to fit a complementary log-log regression to interval-censored survival data to predict the time to recurrence for a medical condition.

Generalized linear models work by building an equation that relates the input field values to the output field values. Once the model is generated, it can be used to estimate values for new data. For each record, a probability of membership is computed for each possible output category. The target category with the highest probability is assigned as the predicted output value for that record.

**Requirements.** You need one or more input fields and exactly one target field (which can have a measurement level of *Continuous* or *Flag*) with two or more categories. Fields used in the model must have their types fully instantiated.

**Strengths.** The generalized linear model is extremely flexible, but the process of choosing the model structure is not automated and thus demands a level of familiarity with your data that is not required by "black box" algorithms.

### GenLin Node Field Options

In addition to the target, input, and partition custom options typically offered on modeling node Fields tabs (see [“Modeling Node Fields Options” on page 26](#)), the GenLin node offers the following extra functionality.

**Use weight field.** The scale parameter is an estimated model parameter related to the variance of the response. The scale weights are "known" values that can vary from observation to observation. If the scale weight variable is specified, the scale parameter, which is related to the variance of the response, is divided by it for each observation. Records with scale weight values that are less than or equal to 0 or are missing are not used in the analysis.

**Target field represents number of events occurring in a set of trials.** When the response is a number of events occurring in a set of trials, the target field contains the number of events and you can select an additional variable containing the number of trials. Alternatively, if the number of trials is the same across all subjects, then trials may be specified using a fixed value. The number of trials should be greater than

or equal to the number of events for each record. Events should be non-negative integers, and trials should be positive integers.

## GenLin Node Model Options

**Model name.** You can generate the model name automatically based on the target or ID field (or model type in cases where no such field is specified) or specify a custom name.

**Use partitioned data.** If a partition field is defined, this option ensures that data from only the training partition is used to build the model.

**Create split models.** Builds a separate model for each possible value of input fields that are specified as split fields. See [“Building Split Models”](#) on page 24 for more information.

**Model type.** There are two options for the type of model to build. **Main effects only** causes the model to include only the input fields individually, and not to test interactions (multiplicative effects) between input fields. **Main effects and all two-way interactions** includes all two-way interactions as well as the input field main effects.

**Offset.** The offset term is a "structural" predictor. Its coefficient is not estimated by the model but is assumed to have the value 1; thus, the values of the offset are simply added to the linear predictor of the target. This is especially useful in Poisson regression models, where each case may have different levels of exposure to the event of interest.

For example, when modeling accident rates for individual drivers, there is an important difference between a driver who has been at fault in one accident in three years of experience and a driver who has been at fault in one accident in 25 years! The number of accidents can be modeled as a Poisson or negative binomial response with a log link if the natural log of the experience of the driver is included as an offset term.

Other combinations of distribution and link types would require other transformations of the offset variable.

*Note:* If a variable offset field is used, the specified field should not also be used as an input. Set the role for the offset field to **None** in an upstream source or Type node if necessary.

### Base category for flag target.

For binary response, you can choose the reference category for the dependent variable. This can affect certain output, such as parameter estimates and saved values, but it should not change the model fit. For example, if your binary response takes values 0 and 1:

- By default, the procedure makes the last (highest-valued) category, or 1, the reference category. In this situation, model-saved probabilities estimate the chance that a given case takes the value 0, and parameter estimates should be interpreted as relating to the likelihood of category 0.
- If you specify the first (lowest-valued) category, or 0, as the reference category, then model-saved probabilities estimate the chance that a given case takes the value 1.
- If you specify the custom category and your variable has defined labels, you can set the reference category by choosing a value from the list. This can be convenient when, in the middle of specifying a model, you don't remember exactly how a particular variable was coded.

**Include intercept in model.** The intercept is usually included in the model. If you can assume the data pass through the origin, you can exclude the intercept.

## GenLin Node Expert Options

If you have detailed knowledge of generalized linear models, expert options allow you to fine-tune the training process. To access expert options, set **Mode** to **Expert** on the Expert tab.

Target Field Distribution and Link Function

### Distribution.

This selection specifies the distribution of the dependent variable. The ability to specify a non-normal distribution and non-identity link function is the essential improvement of the generalized linear model over the general linear model. There are many possible distribution-link function combinations, and several may be appropriate for any given dataset, so your choice can be guided by a priori theoretical considerations or which combination seems to fit best.

- **Binomial.** This distribution is appropriate only for variables that represent a binary response or number of events.
- **Gamma.** This distribution is appropriate for variables with positive scale values that are skewed toward larger positive values. If a data value is less than or equal to 0 or is missing, then the corresponding case is not used in the analysis.
- **Inverse Gaussian.** This distribution is appropriate for variables with positive scale values that are skewed toward larger positive values. If a data value is less than or equal to 0 or is missing, then the corresponding case is not used in the analysis.
- **Negative binomial.** This distribution can be thought of as the number of trials required to observe  $k$  successes and is appropriate for variables with non-negative integer values. If a data value is non-integer, less than 0, or missing, then the corresponding case is not used in the analysis. The fixed value of the negative binomial distribution's ancillary parameter can be any number greater than or equal to 0. When the ancillary parameter is set to 0, using this distribution is equivalent to using the Poisson distribution.
- **Normal.** This is appropriate for scale variables whose values take a symmetric, bell-shaped distribution about a central (mean) value. The dependent variable must be numeric.
- **Poisson.** This distribution can be thought of as the number of occurrences of an event of interest in a fixed period of time and is appropriate for variables with non-negative integer values. If a data value is non-integer, less than 0, or missing, then the corresponding case is not used in the analysis.
- **Tweedie.** This distribution is appropriate for variables that can be represented by Poisson mixtures of gamma distributions; the distribution is "mixed" in the sense that it combines properties of continuous (takes non-negative real values) and discrete distributions (positive probability mass at a single value, 0). The dependent variable must be numeric, with data values greater than or equal to zero. If a data value is less than zero or missing, then the corresponding case is not used in the analysis. The fixed value of the Tweedie distribution's parameter can be any number greater than one and less than two.
- **Multinomial.** This distribution is appropriate for variables that represent an ordinal response. The dependent variable can be numeric or string, and it must have at least two distinct valid data values.

### Link Functions.

The link function is a transformation of the dependent variable that allows estimation of the model. The following functions are available:

- **Identity.**  $f(x)=x$ . The dependent variable is not transformed. This link can be used with any distribution.
- **Complementary log-log.**  $f(x)=\log(-\log(1-x))$ . This is appropriate only with the binomial distribution.
- **Cumulative Cauchit.**  $f(x) = \tan(\pi (x - 0.5))$ , applied to the cumulative probability of each category of the response. This is appropriate only with the multinomial distribution.
- **Cumulative complementary log-log.**  $f(x)=\ln(-\ln(1-x))$ , applied to the cumulative probability of each category of the response. This is appropriate only with the multinomial distribution.
- **Cumulative logit.**  $f(x)=\ln(x / (1-x))$ , applied to the cumulative probability of each category of the response. This is appropriate only with the multinomial distribution.
- **Cumulative negative log-log.**  $f(x)=-\ln(-\ln(x))$ , applied to the cumulative probability of each category of the response. This is appropriate only with the multinomial distribution.
- **Cumulative probit.**  $f(x)=\Phi^{-1}(x)$ , applied to the cumulative probability of each category of the response, where  $\Phi^{-1}$  is the inverse standard normal cumulative distribution function. This is appropriate only with the multinomial distribution.
- **Log.**  $f(x)=\log(x)$ . This link can be used with any distribution.
- **Log complement.**  $f(x)=\log(1-x)$ . This is appropriate only with the binomial distribution.



- **Logit.**  $f(x)=\log(x / (1-x))$ . This is appropriate only with the binomial distribution.
- **Negative binomial.**  $f(x)=\log(x / (x+k^{-1}))$ , where  $k$  is the ancillary parameter of the negative binomial distribution. This is appropriate only with the negative binomial distribution.
- **Negative log-log.**  $f(x)=-\log(-\log(x))$ . This is appropriate only with the binomial distribution.
- **Odds power.**  $f(x)=[(x/(1-x))^{\alpha}-1]/\alpha$ , if  $\alpha \neq 0$ .  $f(x)=\log(x)$ , if  $\alpha=0$ .  $\alpha$  is the required number specification and must be a real number. This is appropriate only with the binomial distribution.
- **Probit.**  $f(x)=\Phi^{-1}(x)$ , where  $\Phi^{-1}$  is the inverse standard normal cumulative distribution function. This is appropriate only with the binomial distribution.
- **Power.**  $f(x)=x^{\alpha}$ , if  $\alpha \neq 0$ .  $f(x)=\log(x)$ , if  $\alpha=0$ .  $\alpha$  is the required number specification and must be a real number. This link can be used with any distribution.

**Parameters.** The controls in this group allow you to specify parameter values when certain distribution options are chosen.

- **Parameter for negative binomial.** For negative binomial distribution, choose either to specify a value or to allow the system to provide an estimated value.
- **Parameter for Tweedie.** For Tweedie distribution, specify a number between 1.0 and 2.0 for the fixed value.

**Parameter Estimation.** The controls in this group allow you to specify estimation methods and to provide initial values for the parameter estimates.

- **Method.** You can select a parameter estimation method. Choose between Newton-Raphson, Fisher scoring, or a hybrid method in which Fisher scoring iterations are performed before switching to the Newton-Raphson method. If convergence is achieved during the Fisher scoring phase of the hybrid method before the maximum number of Fisher iterations is reached, the algorithm continues with the Newton-Raphson method.
- **Scale Parameter Method.** You can select the scale parameter estimation method. Maximum-likelihood jointly estimates the scale parameter with the model effects; note that this option is not valid if the response has a negative binomial, Poisson, or binomial distribution. The deviance and Pearson chi-square options estimate the scale parameter from the value of those statistics. Alternatively, you can specify a fixed value for the scale parameter.
- **Covariance matrix.** The model-based estimator is the negative of the generalized inverse of the Hessian matrix. The robust (also called the Huber/White/sandwich) estimator is a "corrected" model-based estimator that provides a consistent estimate of the covariance, even when the specification of the variance and link functions is incorrect.

**Iterations.** These options allow you to control the parameters for model convergence. See the topic [“Generalized Linear Models Iterations”](#) on page 180 for more information.

**Output.** These options allow you to request additional statistics that will be displayed in the advanced output of the model nugget built by the node. See the topic [“Generalized Linear Models Advanced Output”](#) on page 181 for more information.

**Singularity Tolerance.** Singular (or non-invertible) matrices have linearly dependent columns, which can cause serious problems for the estimation algorithm. Even near-singular matrices can lead to poor results, so the procedure will treat a matrix whose determinant is less than the tolerance as singular. Specify a positive value.

## Generalized Linear Models Iterations

You can set the convergence parameters for estimating the generalized linear model.

**Iterations.** The following options are available:

- **Maximum iterations.** The maximum number of iterations the algorithm will execute. Specify a non-negative integer.
- **Maximum step-halving.** At each iteration, the step size is reduced by a factor of 0.5 until the log-likelihood increases or maximum step-halving is reached. Specify a positive integer.



- **Check for separation of data points.** When selected, the algorithm performs tests to ensure that the parameter estimates have unique values. Separation occurs when the procedure can produce a model that correctly classifies every case. This option is available for binomial responses with binary format .

**Convergence Criteria.** The following options are available

- **Change in parameter estimates.** When selected, the algorithm stops after an iteration in which the absolute or relative change in the parameter estimates is less than the value specified, which must be positive.
- **Change in log-likelihood.** When selected, the algorithm stops after an iteration in which the absolute or relative change in the log-likelihood function is less than the value specified, which must be positive.
- **Hessian convergence.** For the Absolute specification, convergence is assumed if a statistic based on the Hessian convergence is less than the positive value specified. For the Relative specification, convergence is assumed if the statistic is less than the product of the positive value specified and the absolute value of the log-likelihood.

## Generalized Linear Models Advanced Output

Select the optional output you want to display in the advanced output of the generalized linear model nugget. To view the advanced output, browse the model nugget and click the **Advanced** tab. See the topic [“GenLin Model Nugget Advanced Output” on page 182](#) for more information.

The following output is available:

- **Case processing summary.** Displays the number and percentage of cases included and excluded from the analysis and the Correlated Data Summary table.
- **Descriptive statistics.** Displays descriptive statistics and summary information about the dependent variable, covariates, and factors.
- **Model information.** Displays the dataset name, dependent variable or events and trials variables, offset variable, scale weight variable, probability distribution, and link function.
- **Goodness of fit statistics.** Displays deviance and scaled deviance, Pearson chi-square and scaled Pearson chi-square, log-likelihood, Akaike's information criterion (AIC), finite sample corrected AIC (AICC), Bayesian information criterion (BIC), and consistent AIC (CAIC).
- **Model summary statistics.** Displays model fit tests, including likelihood-ratio statistics for the model fit omnibus test and statistics for the Type I or III contrasts for each effect.
- **Parameter estimates.** Displays parameter estimates and corresponding test statistics and confidence intervals. You can optionally display exponentiated parameter estimates in addition to the raw parameter estimates.
- **Covariance matrix for parameter estimates.** Displays the estimated parameter covariance matrix.
- **Correlation matrix for parameter estimates.** Displays the estimated parameter correlation matrix.
- **Contrast coefficient (L) matrices.** Displays contrast coefficients for the default effects and for the estimated marginal means, if requested on the EM Means tab.
- **General estimable functions.** Displays the matrices for generating the contrast coefficient (L) matrices.
- **Iteration history.** Displays the iteration history for the parameter estimates and log-likelihood and prints the last evaluation of the gradient vector and the Hessian matrix. The iteration history table displays parameter estimates for every  $n^{\text{th}}$  iterations beginning with the 0<sup>th</sup> iteration (the initial estimates), where  $n$  is the value of the print interval. If the iteration history is requested, then the last iteration is always displayed regardless of  $n$ .
- **Lagrange multiplier test.** Displays Lagrange multiplier test statistics for assessing the validity of a scale parameter that is computed using the deviance or Pearson chi-square, or set at a fixed number, for the normal, gamma, and inverse Gaussian distributions. For the negative binomial distribution, this tests the fixed ancillary parameter.

**Model Effects.** The following options are available:

- **Analysis Type.** Specify the type of analysis to produce. Type I analysis is generally appropriate when you have a priori reasons for ordering predictors in the model, while Type III is more generally applicable. Wald or likelihood-ratio statistics are computed based upon the selection in the Chi-Square Statistics group.
- **Confidence Interval Level (%).** Specify a confidence level greater than 50 and less than 100. Wald intervals are based on the assumption that parameters have an asymptotic normal distribution; profile likelihood intervals are more accurate but can be computationally expensive. The tolerance level for profile likelihood intervals is the criteria used to stop the iterative algorithm used to compute the intervals.
- **Log-Likelihood Function.** This controls the display format of the log-likelihood function. The full function includes an additional term that is constant with respect to the parameter estimates; it has no effect on parameter estimation and is left out of the display in some software products.

## GenLin Model Nugget

A GenLin model nugget represents the equations estimated by a GenLin node. They contain all of the information captured by the model, as well as information about the model structure and performance.

When you run a stream containing a GenLin model nugget, the node adds new fields whose contents depend on the nature of the target field:

- **Flag target.** Adds fields containing the predicted category and associated probability and the probabilities for each category. The names of the first two new fields are derived from the name of the output field being predicted, prefixed with *\$G-* for the predicted category and *\$GP-* for the associated probability. For example, for an output field named *default*, the new fields would be named *\$G-default* and *\$GP-default*. The latter two additional fields are named based on the values of the output field, prefixed by *\$GP-*. For example, if the legal values of *default* are *Yes* and *No*, the new fields would be named *\$GP-Yes* and *\$GP-No*.
- **Continuous target.** Adds fields containing the predicted mean and standard error.
- **Continuous target, representing number of events in a series of trials.** Adds fields containing the predicted mean and standard error.
- **Ordinal target.** Adds fields containing the predicted category and associated probability for each value of the ordered set. The names of the fields are derived from the value of the ordered set being predicted, prefixed with *\$G-* for the predicted category and *\$GP-* for the associated probability.

**Generating a Filter node.** The Generate menu allows you to create a new Filter node to pass input fields based on the results of the model.

### Predictor Importance

Optionally, a chart that indicates the relative importance of each predictor in estimating the model may also be displayed on the Model tab. Typically you will want to focus your modeling efforts on the predictors that matter most and consider dropping or ignoring those that matter least. Note this chart is only available if **Calculate predictor importance** is selected on the Analyze tab before generating the model. See the topic [“Predictor Importance” on page 37](#) for more information.

### GenLin Model Nugget Advanced Output

The advanced output for generalized linear model gives detailed information about the estimated model and its performance. Most of the information contained in the advanced output is quite technical, and extensive knowledge of this type of analysis is required to properly interpret this output. See the topic [“Generalized Linear Models Advanced Output” on page 181](#) for more information.

### GenLin Model Nugget Settings

The Settings tab for a GenLin model nugget allows you to obtain propensity scores when scoring the model, and also for SQL generation during model scoring. This tab is available for models with flag targets only, and only after the model nugget has been added to a stream.

**Calculate raw propensity scores.** For models with a flag target (which return a yes or no prediction), you can request propensity scores that indicate the likelihood of the true outcome specified for the target field. These are in addition to other prediction and confidence values that may be generated during scoring.

**Calculate adjusted propensity scores.** Raw propensity scores are based only on the training data and may be overly optimistic due to the tendency of many models to overfit this data. Adjusted propensities attempt to compensate by evaluating model performance against a test or validation partition. This option requires that a partition field be defined in the stream and adjusted propensity scores be enabled in the modeling node before generating the model.

**Generate SQL for this model** When using data from a database, SQL code can be pushed back to the database for execution, providing superior performance for many operations.

Select one of the following options to specify how SQL generation is performed.

- **Default: Score using Server Scoring Adapter (if installed) otherwise in process** If connected to a database with a scoring adapter installed, generates SQL using the scoring adapter and associated user defined functions (UDF) and scores your model within the database. When no scoring adapter is available, this option fetches your data back from the database and scores it in SPSS Modeler.
- **Score outside of the Database** If selected, this option fetches your data back from the database and scores it in SPSS Modeler.

### GenLin Model Nugget Summary

The Summary tab for a GenLin model nugget displays the fields and settings used to generate the model. In addition, if you have executed an Analysis node attached to this modeling node, information from that analysis will also be displayed in this section. For general information on using the model browser, see [“Browsing model nuggets” on page 35](#).

## Generalized Linear Mixed Models

---

### GLMM Node

Use this node to create a generalized linear mixed model (GLMM).

#### Generalized linear mixed models

Generalized linear mixed models extend the linear model so that:

- The target is linearly related to the factors and covariates via a specified link function.
- The target can have a non-normal distribution.
- The observations can be correlated.

Generalized linear mixed models cover a wide variety of models, from simple linear regression to complex multilevel models for non-normal longitudinal data.

**Examples.** The district school board can use a generalized linear mixed model to determine whether an experimental teaching method is effective at improving math scores. Students from the same classroom should be correlated since they are taught by the same teacher, and classrooms within the same school may also be correlated, so we can include random effects at school and class levels to account for different sources of variability.

Medical researchers can use a generalized linear mixed model to determine whether a new anticonvulsant drug can reduce a patient's rate of epileptic seizures. Repeated measurements from the same patient are typically positively correlated so a mixed model with some random effects should be appropriate. The target field, the number of seizures, takes positive integer values, so a generalized linear mixed model with a Poisson distribution and log link may be appropriate.

Executives at a cable provider of television, phone, and internet services can use a generalized linear mixed model to know more about potential customers. Since possible answers have nominal measurement levels, the company analyst uses a generalized logit mixed model with a random intercept

to capture correlation between answers to the service usage questions across service types (tv, phone, internet) within a given survey responder's answers.

The Data Structure tab allows you to specify the structural relationships between records in your dataset when observations are correlated. If the records in the dataset represent independent observations, you do not need to specify anything on this tab.

**Subjects.** The combination of values of the specified categorical fields should uniquely define subjects within the dataset. For example, a single *Patient ID* field should be sufficient to define subjects in a single hospital, but the combination of *Hospital ID* and *Patient ID* may be necessary if patient identification numbers are not unique across hospitals. In a repeated measures setting, multiple observations are recorded for each subject, so each subject may occupy multiple records in the dataset.

A **subject** is an observational unit that can be considered independent of other subjects. For example, the blood pressure readings from a patient in a medical study can be considered independent of the readings from other patients. Defining subjects becomes particularly important when there are repeated measurements per subject and you want to model the correlation between these observations. For example, you might expect that blood pressure readings from a single patient during consecutive visits to the doctor are correlated.

All of the fields specified as **Subjects** on the Data Structure tab are used to define subjects for the residual covariance structure, and provide the list of possible fields for defining subjects for random-effects covariance structures on the [Random Effect Block](#).

**Repeated measures.** The fields specified here are used to identify repeated observations. For example, a single variable *Week* might identify the 10 weeks of observations in a medical study, or *Month* and *Day* might be used together to identify daily observations over the course of a year.

**Define covariance groups by.** The categorical fields specified here define independent sets of repeated effects covariance parameters; one for each category defined by the cross-classification of the grouping fields. All subjects have the same covariance type; subjects within the same covariance grouping will have the same values for the parameters.

**Spatial covariance coordinates.** The variables in this list specify the coordinates of the repeated observations when one of the spatial covariance types is selected for the repeated covariance type.

**Repeated covariance type.** This specifies the covariance structure for the residuals. The available structures are:

- First-order autoregressive (AR1)
- Autoregressive moving average (1,1) (ARMA11)
- Compound symmetry
- Diagonal
- Scaled identity
- Spatial: Power
- Spatial: Exponential
- Spatial: Gaussian
- Spatial: Linear
- Spatial: Linear-log
- Spatial: Spherical
- Toeplitz
- Unstructured
- Variance components

### **Target**

These settings define the target, its distribution, and its relationship to the predictors through the link function.

**Target.** The target is required. It can have any measurement level, and the measurement level of the target restricts which distributions and link functions are appropriate.

- **Use number of trials as denominator.** When the target response is a number of events occurring in a set of trials, the target field contains the number of events and you can select an additional field containing the number of trials. For example, when testing a new pesticide you might expose samples of ants to different concentrations of the pesticide and then record the number of ants killed and the number of ants in each sample. In this case, the field recording the number of ants killed should be specified as the target (events) field, and the field recording the number of ants in each sample should be specified as the trials field. If the number of ants is the same for each sample, then the number of trials may be specified using a fixed value.

The number of trials should be greater than or equal to the number of events for each record. Events should be non-negative integers, and trials should be positive integers.

- **Customize reference category.** For a categorical target, you can choose the reference category. This can affect certain output, such as parameter estimates, but it should not change the model fit. For example, if your target takes values 0, 1, and 2, by default, the procedure makes the last (highest-valued) category, or 2, the reference category. In this situation, parameter estimates should be interpreted as relating to the likelihood of category 0 or 1 *relative* to the likelihood of category 2. If you specify a custom category and your target has defined labels, you can set the reference category by choosing a value from the list. This can be convenient when, in the middle of specifying a model, you don't remember exactly how a particular field was coded.

**Target Distribution and Relationship (Link) with the Linear Model.** Given the values of the predictors, the model expects the distribution of values of the target to follow the specified shape, and for the target values to be linearly related to the predictors through the specified link function. Short cuts for several common models are provided, or choose a **Custom** setting if there is a particular distribution and link function combination you want to fit that is not on the short list.

- **Linear model.** Specifies a normal distribution with an identity link, which is useful when the target can be predicted using a linear regression or ANOVA model.
- **Gamma regression.** Specifies a Gamma distribution with a log link, which should be used when the target contains all positive values and is skewed towards larger values.
- **Loglinear.** Specifies a Poisson distribution with a log link, which should be used when the target represents a count of occurrences in a fixed period of time.
- **Negative binomial regression.** Specifies a negative binomial distribution with a log link, which should be used when the target and denominator represent the number of trials required to observe  $k$  successes.
- **Multinomial logistic regression.** Specifies a multinomial distribution, which should be used when the target is a multi-category response. It uses either a cumulative logit link (ordinal outcomes) or a generalized logit link (multi-category nominal responses).
- **Binary logistic regression.** Specifies a binomial distribution with a logit link, which should be used when the target is a binary response predicted by a logistic regression model.
- **Binary probit.** Specifies a binomial distribution with a probit link, which should be used when the target is a binary response with an underlying normal distribution.
- **Interval censored survival.** Specifies a binomial distribution with a complementary log-log link, which is useful in survival analysis when some observations have no termination event.

### Distribution

This selection specifies the distribution of the target. The ability to specify a non-normal distribution and non-identity link function is the essential improvement of the generalized linear mixed model over the linear mixed model. There are many possible distribution-link function combinations, and several may be appropriate for any given dataset, so your choice can be guided by a priori theoretical considerations or which combination seems to fit best.

#### Binomial

This distribution is appropriate only for a target that represents a binary response or number of events.

**Gamma**

This distribution is appropriate for a target with positive scale values that are skewed toward larger positive values. If a data value is less than or equal to 0 or is missing, then the corresponding case is not used in the analysis.

**Inverse Gaussian**

This distribution is appropriate for a target with positive scale values that are skewed toward larger positive values. If a data value is less than or equal to 0 or is missing, then the corresponding case is not used in the analysis.

**Multinomial**

This distribution is appropriate for a target that represents a multi-category response. The form of the model will depend on the measurement level of the target.

A **nominal** target will result in a nominal multinomial model in which a separate set of model parameters are estimated for each category of the target (except the reference category). The parameter estimates for a given predictor show the relationship between that predictor and the likelihood of each category of the target, relative to the reference category.

An **ordinal** target will result in an ordinal multinomial model in which the traditional intercept term is replaced with a set of **threshold** parameters that relate to the cumulative probability of the target categories.

**Negative binomial**

Negative binomial regression uses a negative binomial distribution with a log link, which should be used when the target represents a count of occurrences with high variance.

**Normal**

This is appropriate for a continuous target whose values take a symmetric, bell-shaped distribution about a central (mean) value.

**Poisson**

This distribution can be thought of as the number of occurrences of an event of interest in a fixed period of time and is appropriate for variables with non-negative integer values. If a data value is non-integer, less than 0, or missing, then the corresponding case is not used in the analysis.

**Link function**

The link function is a transformation of the target that allows estimation of the model. The following functions are available:

**Identity**

$f(x)=x$ . The target is not transformed. This link can be used with any distribution, except the multinomial.

**Complementary log-log**

$f(x)=\log(-\log(1-x))$ . This is appropriate only with the binomial or multinomial distribution.

**Cauchit**

$f(x) = \tan(\pi (x - 0.5))$ . This is appropriate only with the binomial or multinomial distribution.

**Log**

$f(x)=\log(x)$ . This link can be used with any distribution, except the multinomial.

**Log complement**

$f(x)=\log(1-x)$ . This is appropriate only with the binomial distribution.

**Logit**

$f(x)=\log(x / (1-x))$ . This is appropriate only with the binomial or multinomial distribution.

**Negative log-log**

$f(x)=-\log(-\log(x))$ . This is appropriate only with the binomial or multinomial distribution.

**Probit**

$f(x)=\Phi^{-1}(x)$ , where  $\Phi^{-1}$  is the inverse standard normal cumulative distribution function. This is appropriate only with the binomial or multinomial distribution.

## Power

$f(x)=x^\alpha$ , if  $\alpha \neq 0$ .  $f(x)=\log(x)$ , if  $\alpha=0$ .  $\alpha$  is the required number specification and must be a real number. This link can be used with any distribution, except the multinomial.

## Fixed Effects




Fixed effects factors are generally thought of as fields whose values of interest are all represented in the dataset, and can be used for scoring. By default, fields with the predefined input role that are not specified elsewhere in the dialog are entered in the fixed effects portion of the model. Categorical (flag, nominal, and ordinal) fields are used as factors in the model and continuous fields are used as covariates.

Enter effects into the model by selecting one or more fields in the source list and dragging to the effects list. The type of effect created depends upon which hotspot you drop the selection.

- **Main.** Dropped fields appear as separate main effects at the bottom of the effects list.
- **2-way.** All possible pairs of the dropped fields appear as 2-way interactions at the bottom of the effects list.
- **3-way.** All possible triplets of the dropped fields appear as 3-way interactions at the bottom of the effects list.
- **\***. The combination of all dropped fields appear as a single interaction at the bottom of the effects list.

Buttons to the right of the Effect Builder allow you to perform various actions.

Table 10. Effect builder button descriptions

Icon	Description
	Delete terms from the fixed effects model by selecting the terms you want to delete and clicking the delete button.
	Reorder the terms within the fixed effects model by selecting the terms you want to reorder and clicking the up or down arrow.
	Add nested terms to the model using the “Add a Custom Term” on page 187 dialog, by clicking on the Add a Custom Term button.

**Include Intercept.** The intercept is usually included in the model. If you can assume the data pass through the origin, you can exclude the intercept.

### Add a Custom Term

You can build nested terms for your model in this procedure. Nested terms are useful for modeling the effect of a factor or covariate whose values do not interact with the levels of another factor. For example, a grocery store chain may follow the spending habits of its customers at several store locations. Since each customer frequents only one of these locations, the *Customer* effect can be said to be **nested within** the *Store location* effect.

Additionally, you can include interaction effects, such as polynomial terms involving the same covariate, or add multiple levels of nesting to the nested term.

**Limitations:** Nested terms have the following restrictions:

- All factors within an interaction must be unique. Thus, if *A* is a factor, then specifying *A\*A* is invalid.
- All factors within a nested effect must be unique. Thus, if *A* is a factor, then specifying *A(A)* is invalid.
- No effect can be nested within a covariate. Thus, if *A* is a factor and *X* is a covariate, then specifying *A(X)* is invalid.

## Constructing a nested term

1. Select a factor or covariate that is nested within another factor, and then click the arrow button.
2. Click **(Within)**.
3. Select the factor within which the previous factor or covariate is nested, and then click the arrow button.
4. Click **Add Term**.

Optionally, you can include interaction effects or add multiple levels of nesting to the nested term.

## Random Effects

Random effects factors are fields whose values in the data file can be considered a random sample from a larger population of values. They are useful for explaining excess variability in the target. By default, if you have selected more than one subject in the Data Structure tab, a Random Effect block will be created for each subject beyond the innermost subject. For example, if you selected School, Class, and Student as subjects on the Data Structure tab, the following random effect blocks are automatically created:

- Random Effect 1: subject is school (with no effects, intercept only)
- Random Effect 2: subject is school \* class (no effects, intercept only)

You can work with random effects blocks in the following ways:

1. To add a new block, click **Add Block...** This opens the “Random Effect Block ” on page 188 dialog.
2. To edit an existing block, select the block you want to edit and click **Edit Block...** This opens the “Random Effect Block ” on page 188 dialog.
3. To delete one or more blocks, select the blocks you want to delete and click the delete button.




## Random Effect Block

Enter effects into the model by selecting one or more fields in the source list and dragging to the effects list. The type of effect created depends upon which hotspot you drop the selection. Categorical (flag, nominal, and ordinal) fields are used as factors in the model and continuous fields are used as covariates.

- **Main.** Dropped fields appear as separate main effects at the bottom of the effects list.
- **2-way.** All possible pairs of the dropped fields appear as 2-way interactions at the bottom of the effects list.
- **3-way.** All possible triplets of the dropped fields appear as 3-way interactions at the bottom of the effects list.
- **\***. The combination of all dropped fields appear as a single interaction at the bottom of the effects list.

Buttons to the right of the Effect Builder allow you to perform various actions.

Table 11. Effect builder button descriptions

Icon	Description
	Delete terms from the model by selecting the terms you want to delete and clicking the delete button.
	Reorder the terms within the model by selecting the terms you want to reorder and clicking the up or down arrow.
	Add nested terms to the model using the “Add a Custom Term ” on page 187 dialog, by clicking on the Add a Custom Term button.

**Include Intercept.** The intercept is not included in the random effects model by default. If you can assume the data pass through the origin, you can exclude the intercept.



**Display parameter predictions for this block.** Specifies to display the random-effects parameter estimates.

**Define covariance groups by.** The categorical fields specified here define independent sets of random effects covariance parameters; one for each category defined by the cross-classification of the grouping fields. A different set of grouping fields can be specified for each random effect block. All subjects have the same covariance type; subjects within the same covariance grouping will have the same values for the parameters.

**Subject combination.** This allows you to specify random effect subjects from preset combinations of subjects from the Data Structure tab. For example, if *School*, *Class*, and *Student* are defined as subjects on the Data Structure tab, and in that order, then the Subject combination dropdown list will have **None**, **School**, **School \* Class**, and **School \* Class \* Student** as options.

**Random effect covariance type.** This specifies the covariance structure for the residuals. The available structures are:

- First-order autoregressive (AR1)
- Heterogeneous autoregressive (ARH1)
- Autoregressive moving average (1,1) (ARMA11)
- Compound symmetry
- Heterogeneous compound symmetry (CSH)
- Diagonal
- Scaled identity
- Toeplitz
- Unstructured
- Variance components

### *Weight and Offset*

**Analysis weight.** The scale parameter is an estimated model parameter related to the variance of the response. The analysis weights are "known" values that can vary from observation to observation. If the analysis weight field is specified, the scale parameter, which is related to the variance of the response, is divided by the analysis weight values for each observation. Records with analysis weight values that are less than or equal to 0 or are missing are not used in the analysis.

**Offset.** The offset term is a "structural" predictor. Its coefficient is not estimated by the model but is assumed to have the value 1; thus, the values of the offset are simply added to the linear predictor of the target. This is especially useful in Poisson regression models, where each case may have different levels of exposure to the event of interest.

For example, when modeling accident rates for individual drivers, there is an important difference between a driver who has been at fault in one accident in three years of experience and a driver who has been at fault in one accident in 25 years! The number of accidents can be modeled as a Poisson or negative binomial response with a log link if the natural log of the experience of the driver is included as an offset term.

Other combinations of distribution and link types would require other transformations of the offset variable.

### *General Build Options*

These selections specify some more advanced criteria used to build the model.

#### **Sorting Order**

These controls determine the order of the categories for the target and factors (categorical inputs) for purposes of determining the "last" category. The target sort order setting is ignored if the target is not categorical or if a custom reference category is specified on the ["Target" on page 184 settings](#).

## Stopping Rules

You can specify the maximum number of iterations the algorithm will execute. The algorithm uses a doubly iterative process that consists of an inner loop and an outer loop. The value that is specified for the maximum number of iterations applies to both loops. Specify a non-negative integer. The default is 100.

## Post-Estimation Settings

These settings determine how some of the model output is computed for viewing.

### Confidence level (%)

This is the level of confidence used to compute interval estimates of the model coefficients. Specify a value greater than 0 and less than 100. The default is 95.

### Degrees of freedom

This specifies how degrees of freedom are computed for significance tests. Choose **Residual method** if your sample size is sufficiently large, or the data are balanced, or the model uses a simpler covariance type (for example, scaled identity or diagonal). This is the default setting. Choose **Satterthwaite approximation** if your sample size is small, or the data are unbalanced, or the model uses a complicated covariance type (for example, unstructured). Choose **Kenward-Roger approximation** if your sample size is small and you have a Restricted Maximum Likelihood (REML) model.

### Tests of fixed effects and coefficients

This is the method for computing the parameter estimates covariance matrix. Choose the robust estimate if you are concerned that the model assumptions are violated.

## Estimation

The model building algorithm uses a doubly iterative process that consists of an inner loop and an outer loop. The following settings apply to the inner loop.

## Sorting Order

These controls determine the order of the categories for the target and factors (categorical inputs) for purposes of determining the "last" category. The target sort order setting is ignored if the target is not categorical or if a custom reference category is specified on the ["Target" on page 184](#) settings.

## Parameter Convergence.

Convergence is assumed if the maximum absolute change or maximum relative change in the parameter estimates is less than the value specified, which must be non-negative. The criterion is not used if the value specified equals 0.

## Log-likelihood Convergence.

Convergence is assumed if the absolute change or relative change in the log-likelihood function is less than the value specified, which must be non-negative. The criterion is not used if the value specified equals 0.

## Hessian Convergence.

For the **Absolute** specification, convergence is assumed if a statistic based on the Hessian is less than the value specified. For the **Relative** specification, convergence is assumed if the statistic is less than the product of the value specified and the absolute value of the log-likelihood. The criterion is not used if the value specified equals 0.

## Maximum Fisher scoring steps.

Specify a non-negative integer. A value of 0 specifies the Newton-Raphson method. Values greater than 0 specify to use the Fisher scoring algorithm up to iteration number  $n$ , where  $n$  is the specified integer, and Newton-Raphson thereafter.

## Singularity tolerance.

This value is used as the tolerance in checking singularity. Specify a positive value.

## Stopping Rules

You can specify the maximum number of iterations the algorithm will execute. The algorithm uses a doubly iterative process that consists of an inner loop and an outer loop. The value that is specified for the maximum number of iterations applies to both loops. Specify a non-negative integer. The default is 100.

## Post-Estimation Settings

These settings determine how some of the model output is computed for viewing.

### Confidence level (%)

This is the level of confidence used to compute interval estimates of the model coefficients. Specify a value greater than 0 and less than 100. The default is 95.

### Degrees of freedom

This specifies how degrees of freedom are computed for significance tests. Choose **Residual method** if your sample size is sufficiently large, or the data are balanced, or the model uses a simpler covariance type (for example, scaled identity or diagonal). This is the default setting. Choose **Satterthwaite approximation** if your sample size is small, or the data are unbalanced, or the model uses a complicated covariance type (for example, unstructured). Choose **Kenward-Roger approximation** if your sample size is small and you have a Restricted Maximum Likelihood (REML) model.

### Tests of fixed effects and coefficients

This is the method for computing the parameter estimates covariance matrix. Choose the robust estimate if you are concerned that the model assumptions are violated.

**Note:** By default, Parameter Convergence is used, where the maximum **Absolute** change at a tolerance of 1E-6 is checked. This setting might produce results that differ from the results that are obtained in versions before version 22. To reproduce results from pre-22 versions, use **Relative** for the Parameter Convergence criterion and keep the default tolerance value of 1E-6.

## General

**Model Name.** You can generate the model name automatically based on the target fields or specify a custom name. The automatically generated name is the target field name. If there are multiple targets, then the model name is the field names in order, connected by ampersands. For example, if *field1 field2 field3* are targets, then the model name is: *field1 & field2 & field3*.

**Make Available for Scoring.** When the model is scored, the selected items in this group should be produced. The predicted value (for all targets) and confidence (for categorical targets) are always computed when the model is scored. The computed confidence can be based on the probability of the predicted value (the highest predicted probability) or the difference between the highest predicted probability and the second highest predicted probability.

- **Predicted probability for categorical targets.** This produces the predicted probabilities for categorical targets. A field is created for each category.
- **Propensity scores for flag targets.** For models with a flag target (which return a yes or no prediction), you can request propensity scores that indicate the likelihood of the true outcome specified for the target field. The model produces raw propensity scores; if partitions are in effect, the model also produces adjusted propensity scores based on the testing partition.

## Estimated Means

This tab allows you to display the estimated marginal means for levels of factors and factor interactions. Estimated marginal means are not available for multinomial models.

### Terms

The model terms in the Fixed Effects that are entirely comprised of categorical fields are listed here. Check each term for which you want the model to produce estimated marginal means.

### Contrast Type

This specifies the type of contrast to use for the levels of the contrast field.

#### None

No contrasts are produced.

#### Pairwise

Produces pairwise comparisons for all level combinations of the specified factors. This is the only available contrast for factor interactions.

**Deviation**

Contrasts compare each level of the factor to the grand mean.

**Simple**

Contrasts compare each level of the factor, except the last, to the last level. The "last" level is determined by the sort order for factors specified on the Build Options. Note that all of these contrast types are not orthogonal.

**Contrast Field**

This specifies a factor, the levels of which are compared using the selected contrast type. If **None** is selected as the contrast type, no contrast field can (or need) be selected.

**Continuous Fields**

The listed continuous fields are extracted from the terms in the Fixed Effects that use continuous fields. When computing estimated marginal means, covariates are fixed at the specified values. Select the mean or specify a custom value.

**Adjust for multiple comparisons using**

When performing hypothesis tests with multiple contrasts, the overall significance level can be adjusted from the significance levels for the included contrasts. This allows you to choose the adjustment method.

**Least significant difference**

This method does not control the overall probability of rejecting the hypotheses that some linear contrasts are different from the null hypothesis values.

**Sequential Bonferroni**

This is a sequentially step-down rejective Bonferroni procedure that is much less conservative in terms of rejecting individual hypotheses but maintains the same overall significance level.

**Sequential Sidak**

This is a sequentially step-down rejective Sidak procedure that is much less conservative in terms of rejecting individual hypotheses but maintains the same overall significance level.

The least significant difference method is less conservative than the sequential Sidak method, which in turn is less conservative than the sequential Bonferroni; that is, least significant difference will reject at least as many individual hypotheses as sequential Sidak, which in turn will reject at least as many individual hypotheses as sequential Bonferroni.

**Display estimated means in terms of**

This specifies whether to compute estimated marginal means based on the original scale of the target or based on the link function transformation.

**Original target scale**

Computes estimated marginal means for the target. Note that when the target is specified using the events/trials option, this gives the estimated marginal means for the events/trials proportion rather than for the number of events.

**Link function transformation**

Computes estimated marginal means for the linear predictor.

**Model view**

By default, the Model Summary view is shown. To see another model view, select it from the view thumbnails.

*Model Summary*

This view is a snapshot, at-a-glance summary of the model and its fit.

**Table.** The table identifies the target, probability distribution, and link function specified on the Target settings. If the target is defined by events and trials, the cell is split to show the events field and the trials field or fixed number of trials. Additionally the finite sample corrected Akaike information criterion (AICC) and Bayesian information criterion (BIC) are displayed.

- *Akaike Corrected*. A measure for selecting and comparing mixed models based on the -2 (Restricted) log likelihood. Smaller values indicate better models. The AICC "corrects" the AIC for small sample sizes. As the sample size increases, the AICC converges to the AIC.
- *Bayesian*. A measure for selecting and comparing models based on the -2 log likelihood. Smaller values indicate better models. The BIC also "penalizes" overparameterized models (complex models with a large number of inputs, for example), but more strictly than the AIC.

**Chart.** If the target is categorical, a chart displays the accuracy of the final model, which is the percentage of correct classifications.

### *Data Structure*

This view provides a summary of the data structure you specified, and helps you to check that the subjects and repeated measures have been specified correctly. The observed information for the first subject is displayed for each subject field and repeated measures field, and the target. Additionally, the number of levels for each subject field and repeated measures field is displayed.

### *Predicted by Observed*

For continuous targets, including targets specified as events/trials, this displays a binned scatterplot of the predicted values on the vertical axis by the observed values on the horizontal axis. Ideally, the points should lie on a 45-degree line; this view can tell you whether any records are predicted particularly badly by the model.

### *Classification*

For categorical targets, this displays the cross-classification of observed versus predicted values in a heat map, plus the overall percent correct.

**Table styles.** There are several different display styles, which are accessible from the **Style** dropdown list.

- **Row percents.** This displays the row percentages (the cell counts expressed as a percent of the row totals) in the cells. This is the default.
- **Cell counts.** This displays the cell counts in the cells. The shading for the heat map is still based on the row percentages.
- **Heat map.** This displays no values in the cells, just the shading.
- **Compressed.** This displays no row or column headings, or values in the cells. It can be useful when the target has a lot of categories.

**Missing.** If any records have missing values on the target, they are displayed in a **(Missing)** row under all valid rows. Records with missing values do not contribute to the overall percent correct.

**Multiple targets.** If there are multiple categorical targets, then each target is displayed in a separate table and there is a **Target** dropdown list that controls which target to display.

**Large tables.** If the displayed target has more than 100 categories, no table is displayed.

### *Fixed Effects*

This view displays the size of each fixed effect in the model.

**Styles.** There are different display styles, which are accessible from the **Style** dropdown list.

- **Diagram.** This is a chart in which effects are sorted from top to bottom in the order in which they were specified on the Fixed Effects settings. Connecting lines in the diagram are weighted based on effect significance, with greater line width corresponding to more significant effects (smaller *p*-values). This is the default.
- **Table.** This is an ANOVA table for the overall model and the individual model effects. The individual effects are sorted from top to bottom in the order in which they were specified on the Fixed Effects settings.

**Significance.** There is a Significance slider that controls which effects are shown in the view. Effects with significance values greater than the slider value are hidden. This does not change the model, but simply allows you to focus on the most important effects. By default the value is 1.00, so that no effects are filtered based on significance.

#### *Fixed Coefficients*

This view displays the value of each fixed coefficient in the model. Note that factors (categorical predictors) are indicator-coded within the model, so that **effects** containing factors will generally have multiple associated **coefficients**; one for each category except the category corresponding to the redundant coefficient.

**Styles.** There are different display styles, which are accessible from the **Style** dropdown list.

- **Diagram.** This is a chart which displays the intercept first, and then sorts effects from top to bottom in the order in which they were specified on the Fixed Effects settings. Within effects containing factors, coefficients are sorted by ascending order of data values. Connecting lines in the diagram are colored and weighted based on coefficient significance, with greater line width corresponding to more significant coefficients (smaller  $p$ -values). This is the default style.
- **Table.** This shows the values, significance tests, and confidence intervals for the individual model coefficients. After the intercept, the effects are sorted from top to bottom in the order in which they were specified on the Fixed Effects settings. Within effects containing factors, coefficients are sorted by ascending order of data values.

**Multinomial.** If the multinomial distribution is in effect, then the Multinomial drop-down list controls which target category to display. The sort order of the values in the list is determined by the specification on the Build Options settings.

**Exponential.** This displays exponential coefficient estimates and confidence intervals for certain model types, including Binary logistic regression (binomial distribution and logit link), Nominal logistic regression (multinomial distribution and logit link), Negative binomial regression (negative binomial distribution and log link), and Log-linear model (Poisson distribution and log link).

**Significance.** There is a Significance slider that controls which coefficients are shown in the view. Coefficients with significance values greater than the slider value are hidden. This does not change the model, but simply allows you to focus on the most important coefficients. By default the value is 1.00, so that no coefficients are filtered based on significance.

#### *Random Effect Covariances*

This view displays the random effects covariance matrix (**G**).

**Styles.** There are different display styles, which are accessible from the **Style** dropdown list.

- **Covariance values.** This is a heat map of the covariance matrix in which effects are sorted from top to bottom in the order in which they were specified on the Fixed Effects settings. Colors in the corrgram correspond to the cell values as shown in the key. This is the default.
- **Corrgram.** This is a heat map of the covariance matrix.
- **Compressed.** This is a heat map of the covariance matrix without the row and column headings.

**Blocks.** If there are multiple random effect blocks, then there is a Block dropdown list for selecting the block to display.

**Groups.** If a random effect block has a group specification, then there is a Group dropdown list for selecting the group level to display.

**Multinomial.** If the multinomial distribution is in effect, then the Multinomial drop-down list controls which target category to display. The sort order of the values in the list is determined by the specification on the Build Options settings.

## Covariance Parameters

This view displays the covariance parameter estimates and related statistics for residual and random effects. These are advanced, but fundamental, results that provide information on whether the covariance structure is suitable.

**Summary table.** This is a quick reference for the number of parameters in the residual (**R**) and random effect (**G**) covariance matrices, the rank (number of columns) in the fixed effect (**X**) and random effect (**Z**) design matrices, and the number of subjects defined by the subject fields that define the data structure.

**Covariance parameter table.** For the selected effect, the estimate, standard error, and confidence interval are displayed for each covariance parameter. The number of parameters shown depends upon the covariance structure for the effect and, for random effect blocks, the number of effects in the block. If you see that the off-diagonal parameters are not significant, you may be able to use a simpler covariance structure.

**Effects.** If there are random effect blocks, then there is an Effect dropdown list for selecting the residual or random effect block to display. The residual effect is always available.

**Groups.** If a residual or random effect block has a group specification, then there is a Group dropdown list for selecting the group level to display.

**Multinomial.** If the multinomial distribution is in effect, then the Multinomial drop-down list controls which target category to display. The sort order of the values in the list is determined by the specification on the Build Options settings.

### *Estimated Means: Significant Effects*

These are charts displayed for the 10 "most significant" fixed all-factor effects, starting with the three-way interactions, then the two-way interactions, and finally main effects. The chart displays the model-estimated value of the target on the vertical axis for each value of the main effect (or first listed effect in an interaction) on the horizontal axis; a separate line is produced for each value of the second listed effect in an interaction; a separate chart is produced for each value of the third listed effect in a three-way interaction; all other predictors are held constant. It provides a useful visualization of the effects of each predictor's coefficients on the target. Note that if no predictors are significant, no estimated means are produced.

**Confidence.** This displays upper and lower confidence limits for the marginal means, using the confidence level specified as part of the Build Options.

### *Estimated Means: Custom Effects*

These are tables and charts for user-requested fixed all-factor effects.

**Styles.** There are different display styles, which are accessible from the **Style** dropdown list.

- **Diagram.** This style displays a line chart of the model-estimated value of the target on the vertical axis for each value of the main effect (or first listed effect in an interaction) on the horizontal axis; a separate line is produced for each value of the second listed effect in an interaction; a separate chart is produced for each value of the third listed effect in a three-way interaction; all other predictors are held constant.

If contrasts were requested, another chart is displayed to compare levels of the contrast field; for interactions, a chart is displayed for each level combination of the effects other than the contrast field. For **pairwise** contrasts, it is a distance network chart; that is, a graphical representation of the comparisons table in which the distances between nodes in the network correspond to differences between samples. Yellow lines correspond to statistically significant differences; black lines correspond to non-significant differences. Hovering over a line in the network displays a tooltip with the adjusted significance of the difference between the nodes connected by the line.

For **deviation** contrasts, a bar chart is displayed with the model-estimated value of the target on the vertical axis and the values of the contrast field on the horizontal axis; for interactions, a chart is displayed for each level combination of the effects other than the contrast field. The bars show the difference between each level of the contrast field and the overall mean, which is represented by a black horizontal line.

For **simple** contrasts, a bar chart is displayed with the model-estimated value of the target on the vertical axis and the values of the contrast field on the horizontal axis; for interactions, a chart is displayed for each level combination of the effects other than the contrast field. The bars show the difference between each level of the contrast field (except the last) and the last level, which is represented by a black horizontal line.

- **Table.** This style displays a table of the model-estimated value of the target, its standard error, and confidence interval for each level combination of the fields in the effect; all other predictors are held constant.

If contrasts were requested, another table is displayed with the estimate, standard error, significance test, and confidence interval for each contrast; for interactions, there a separate set of rows for each level combination of the effects other than the contrast field. Additionally, a table with the overall test results is displayed; for interactions, there is a separate overall test for each level combination of the effects other than the contrast field.

**Confidence.** This toggles the display of upper and lower confidence limits for the marginal means, using the confidence level specified as part of the Build Options.

**Layout.** This toggles the layout of the pairwise contrasts diagram. The circle layout is less revealing of contrasts than the network layout but avoids overlapping lines.

### Settings

When the model is scored, the selected items in this tab should be produced. The predicted value (for all targets) and confidence (for categorical targets) are always computed when the model is scored. The computed confidence can be based on the probability of the predicted value (the highest predicted probability) or the difference between the highest predicted probability and the second highest predicted probability.

- **Predicted probability for categorical targets.** This produces the predicted probabilities for categorical targets. A field is created for each category.
- **Propensity scores for flag targets.** For models with a flag target (which return a yes or no prediction), you can request propensity scores that indicate the likelihood of the true outcome specified for the target field. The model produces raw propensity scores; if partitions are in effect, the model also produces adjusted propensity scores based on the testing partition.

**Generate SQL for this model** When using data from a database, SQL code can be pushed back to the database for execution, providing superior performance for many operations.

Select one of the following options to specify how SQL generation is performed.

- **Default: Score using Server Scoring Adapter (if installed) otherwise in process** If connected to a database with a scoring adapter installed, generates SQL using the scoring adapter and associated user defined functions (UDF) and scores your model within the database. When no scoring adapter is available, this option fetches your data back from the database and scores it in SPSS Modeler.
- **Score outside of the Database** If selected, this option fetches your data back from the database and scores it in SPSS Modeler.

## GLE Node

---

The GLE model identifies the dependent variable that is linearly related to the factors and covariates via a specified link function. Moreover, the model allows for the dependent variable to have a non-normal distribution. It covers widely used statistical models, such as linear regression for normally distributed responses, logistic models for binary data, loglinear models for count data, complementary log-log models for interval-censored survival data, plus many other statistical models through its very general model formulation

**Examples.** A shipping company can use generalized linear models to fit a Poisson regression to damage counts for several types of ships constructed in different time periods, and the resulting model can help determine which ship types are most prone to damage.



A car insurance company can use generalized linear models to fit a gamma regression to damage claims for cars, and the resulting model can help determine the factors that contribute the most to claim size.

Medical researchers can use generalized linear models to fit a complementary log-log regression to interval-censored survival data to predict the time to recurrence for a medical condition.

GLE models work by building an equation that relates the input field values to the output field values. Once the model is generated, it can be used to estimate values for new data.

For a categorical target, for each record, a probability of membership is computed for each possible output category. The target category with the highest probability is assigned as the predicted output value for that record.

**Requirements.** You need one or more input fields and exactly one target field (which can have a measurement level of *Continuous*, *Categorical*, or *Flag*) with two or more categories. Fields used in the model must have their types fully instantiated.

## Target

These settings define the target, its distribution, and its relationship to the predictors through the link function.

**Target** The target is required. It can have any measurement level, and the measurement level of the target affects which distributions and link functions are appropriate.

- **Use predefined target** To use the target settings from an upstream Type node (or the Types tab of an upstream source node), select this option.
- **Use custom target** To manually assign a target, select this option.
- **Use number of trials as denominator** When the target response is a number of events occurring in a set of trials, the target field contains the number of events and you can select an additional field containing the number of trials. For example, when testing a new pesticide you might expose samples of ants to different concentrations of the pesticide and then record the number of ants killed and the number of ants in each sample. In this case, the field recording the number of ants killed should be specified as the target (events) field, and the field recording the number of ants in each sample should be specified as the trials field. If the number of ants is the same for each sample, then the number of trials may be specified using a fixed value.

The number of trials should be greater than or equal to the number of events for each record. Events should be non-negative integers, and trials should be positive integers.

- **Customize reference category.** For a categorical target, you can choose the reference category. This can affect certain output, such as parameter estimates, but it should not change the model fit. For example, if your target takes values 0, 1, and 2, by default, the procedure makes the last (highest-valued) category, or 2, the reference category. In this situation, parameter estimates should be interpreted as relating to the likelihood of category 0 or 1 *relative* to the likelihood of category 2. If you specify a custom category and your target has defined labels, you can set the reference category by choosing a value from the list. This can be convenient when, in the middle of specifying a model, you don't remember exactly how a particular field was coded.

**Target Distribution and Relationship (Link) with the Linear Model** Given the values of the predictors, the model expects the distribution of values of the target to follow the specified shape, and for the target values to be linearly related to the predictors through the specified link function. Short cuts for several common models are provided, or choose a **Custom** setting if there is a particular distribution and link function combination you want to fit that is not on the short list.

- **Linear model** Specifies a normal distribution with an identity link, which is useful when the target can be predicted using a linear regression or ANOVA model.
- **Gamma regression** Specifies a Gamma distribution with a log link, which should be used when the target contains all positive values and is skewed towards larger values.
- **Loglinear** Specifies a Poisson distribution with a log link, which should be used when the target represents a count of occurrences in a fixed period of time.

- **Negative binomial regression** Specifies a negative binomial distribution with a log link, which should be used when the target and denominator represent the number of trials required to observe  $k$  successes.
- **Tweedie regression** Specifies a Tweedie distribution with identity, log, or power link functions and are useful for modeling responses that are a mixture of zeros and positive real values. These distributions are also called *compound Poisson*, *compound gamma*, and *Poisson-gamma* distributions.
- **Multinomial logistic regression** Specifies a multinomial distribution, which should be used when the target is a multi-category response. It uses either a cumulative logit link (ordinal outcomes) or a generalized logit link (multi-category nominal responses).
- **Binary logistic regression** Specifies a binomial distribution with a logit link, which should be used when the target is a binary response predicted by a logistic regression model.
- **Binary probit** Specifies a binomial distribution with a probit link, which should be used when the target is a binary response with an underlying normal distribution.
- **Interval censored survival** Specifies a binomial distribution with a complementary log-log link, which is useful in survival analysis when some observations have no termination event.
- **Custom** Specify your own combination of distribution and link function.

### Distribution

This selection specifies the **Distribution** of the target. The ability to specify a non-normal distribution and non-identity link function is the essential improvement of the generalized linear model over the linear model. There are many possible distribution-link function combinations, and several may be appropriate for any given dataset, so your choice can be guided by a priori theoretical considerations or which combination seems to fit best.

- **Automatic** If you are unsure which distribution to use, select this option; the node analyzes your data to estimate and apply the best distribution method.
- **Binomial** This distribution is appropriate only for a target that represents a binary response or number of events.
- **Gamma** This distribution is appropriate for a target with positive scale values that are skewed toward larger positive values. If a data value is less than or equal to 0 or is missing, then the corresponding case is not used in the analysis.
- **Inverse Gaussian** This distribution is appropriate for a target with positive scale values that are skewed toward larger positive values. If a data value is less than or equal to 0 or is missing, then the corresponding case is not used in the analysis.
- **Multinomial** This distribution is appropriate for a target that represents a multi-category response. The form of the model will depend on the measurement level of the target.

A **nominal** target will result in a nominal multinomial model in which a separate set of model parameters are estimated for each category of the target (except the reference category). The parameter estimates for a given predictor show the relationship between that predictor and the likelihood of each category of the target, relative to the reference category.

An **ordinal** target will result in an ordinal multinomial model in which the traditional intercept term is replaced with a set of **threshold** parameters that relate to the cumulative probability of the target categories.

- **Negative binomial** Negative binomial regression uses a negative binomial distribution with a log link, which should be used when the target represents a count of occurrences with high variance.
- **Normal** This is appropriate for a continuous target whose values take a symmetric, bell-shaped distribution about a central (mean) value.
- **Poisson** This distribution can be thought of as the number of occurrences of an event of interest in a fixed period of time and is appropriate for variables with non-negative integer values. If a data value is non-integer, less than 0, or missing, then the corresponding case is not used in the analysis.
- **Tweedie** This distribution is appropriate for variables that can be represented by Poisson mixtures of gamma distributions; the distribution is "mixed" in the sense that it combines properties of continuous (takes non-negative real values) and discrete distributions (positive probability mass at a single value,

0). The dependent variable must be numeric, with data values greater than or equal to zero. If a data value is less than zero or missing, then the corresponding case is not used in the analysis. The fixed value of the Tweedie distribution's parameter can be any number greater than one and less than two.

## Link functions

The **Link function** is a transformation of the target that allows estimation of the model. The following functions are available:

- **Automatic** If you are unsure which link to use, select this option; the node analyzes your data to estimate and apply the best link function.
- **Identity**  $f(x)=x$ . The target is not transformed. This link can be used with any distribution, except the multinomial.
- **Complementary log-log**  $f(x)=\log(-\log(1-x))$ . This is appropriate only with the binomial or multinomial distribution.
- **Cauchit**  $f(x) = \tan(\pi (x - 0.5))$ . This is appropriate only with the binomial or multinomial distribution.
- **Log**  $f(x)=\log(x)$ . This link can be used with any distribution, except the multinomial.
- **Log complement**  $f(x)=\log(1-x)$ . This is appropriate only with the binomial distribution.
- **Logit**  $f(x)=\log(x / (1-x))$ . This is appropriate only with the binomial or multinomial distribution.
- **Negative log-log**  $f(x)=-\log(-\log(x))$ . This is appropriate only with the binomial or multinomial distribution.
- **Probit**  $f(x)=\Phi^{-1}(x)$ , where  $\Phi^{-1}$  is the inverse standard normal cumulative distribution function. This is appropriate only with the binomial or multinomial distribution.
- **Power**  $f(x)=x^\alpha$ , if  $\alpha \neq 0$ .  $f(x)=\log(x)$ , if  $\alpha=0$ .  $\alpha$  is the required number specification and must be a real number. This link can be used with any distribution, except the multinomial.

**Parameter for Tweedie** Only available if you have selected either the **Tweedie regression** radio button or Tweedie as the **Distribution** method. Select a value between 1 and 2.

## Model effects

Fixed effects factors are generally thought of as fields whose values of interest are all represented in the dataset, and can be used for scoring. By default, fields with the predefined input role that are not specified elsewhere in the dialog are entered in the fixed effects portion of the model. Categorical (flag, nominal, and ordinal) fields are used as factors in the model and continuous fields are used as covariates.

Enter effects into the model by selecting one or more fields in the source list and dragging to the effects list. The type of effect created depends upon which hotspot you drop the selection.

- **Main** Dropped fields appear as separate main effects at the bottom of the effects list.
- **2-way** All possible pairs of the dropped fields appear as 2-way interactions at the bottom of the effects list.
- **3-way** All possible triplets of the dropped fields appear as 3-way interactions at the bottom of the effects list.
- **\*** The combination of all dropped fields appear as a single interaction at the bottom of the effects list.

Buttons to the right of the Effect Builder allow you to perform various actions.

Table 12. Effect builder button descriptions




Icon	Description
	Delete terms from the fixed effects model by selecting the terms you want to delete and clicking the delete button.

Table 12. Effect builder button descriptions (continued)

Icon	Description
	Reorder the terms within the fixed effects model by selecting the terms you want to reorder and clicking the up or down arrow.
	Add nested terms to the model using the Add a Custom Term dialog, by clicking on the Add a Custom Term button.

**Include Intercept** The intercept is usually included in the model. If you can assume the data pass through the origin, you can exclude the intercept.

### Add a custom term

You can build nested terms for your model in this procedure. Nested terms are useful for modeling the effect of a factor or covariate whose values do not interact with the levels of another factor. For example, a grocery store chain may follow the spending habits of its customers at several store locations. Since each customer frequents only one of these locations, the Customer effect can be said to be *nested within* the Store location effect.

Additionally, you can include interaction effects, such as polynomial terms involving the same covariate, or add multiple levels of nesting to the nested term.

**Limitations.** Nested terms have the following restrictions:

- All factors within an interaction must be unique. Thus, if *A* is a factor, then specifying *A\*A* is invalid.
- All factors within a nested effect must be unique. Thus, if *A* is a factor, then specifying *A(A)* is invalid.
- No effect can be nested within a covariate. Thus, if *A* is a factor and *X* is a covariate, then specifying *A(X)* is invalid.

### Constructing a nested term

1. Select a factor or covariate that is nested within another factor, and then click the arrow button.
2. Click **(Within)**.
3. Select the factor within which the previous factor or covariate is nested, and then click the arrow button.
4. Click **Add Term**.

Optionally, you can include interaction effects or add multiple levels of nesting to the nested term.

## Weight and Offset

**Analysis weight** The scale parameter is an estimated model parameter related to the variance of the response. The analysis weights are "known" values that can vary from observation to observation. If the **Analysis weight** field is specified, the scale parameter, which is related to the variance of the response, is divided by the analysis weight values for each observation. Records with analysis weight values that are less than or equal to 0, or are missing, are not used in the analysis.

**Offset** The offset term is a *structural* predictor. Its coefficient is not estimated by the model but is assumed to have the value 1; thus, the values of the offset are simply added to the linear predictor of the target. This is especially useful in Poisson regression models, where each case may have different levels of exposure to the event of interest.

For example, when modeling accident rates for individual drivers, there is an important difference between a driver who has been at fault in one accident in three years of experience and a driver who has been at fault in one accident in 25 years. The number of accidents can be modeled as a Poisson or

negative binomial response with a log link if the natural log of the experience of the driver is included as an offset term.

Other combinations of distribution and link types would require other transformations of the offset variable.

## Build options

These selections specify some more advanced criteria used to build the model.

**Sorting Order** These controls determine the order of the categories for the target and factors (categorical inputs) for purposes of determining the "last" category. The target sort order setting is ignored if the target is not categorical or if a custom reference category is specified on the ["Target "](#) on page 197 settings.

**Post-Estimation Settings** These settings determine how some of the model output is computed for viewing.

- **Confidence level %** This is the level of confidence used to compute interval estimates of the model coefficients. Specify a value greater than 0 and less than 100. The default is 95.
- **Degrees of freedom** This specifies how degrees of freedom are computed for significance tests. Choose **Fixed for all tests (Residual method)** if your sample size is sufficiently large, or the data are balanced, or the model uses a simpler covariance type; for example, scaled identity or diagonal. This is the default. Choose **Varied across tests (Satterthwaite approximation)** if your sample size is small, or the data are unbalanced, or the model uses a complicated covariance type; for example, unstructured.
- **Tests of fixed effects and coefficients.** This is the method for computing the parameter estimates covariance matrix. Choose the robust estimate if you are concerned that the model assumptions are violated.

**Detect influential outliers** For all distributions except multinomial distribution, select this option to identify influential outliers.

**Conduct trend analysis** For a scatter plot, select this option to conduct trend analysis.

## Estimation

**Method** Select the maximum likelihood estimation method to be used; the available options are:

- Fisher scoring
- Newton-Raphson
- Hybrid

**Maximum Fisher iterations** Specify a non-negative integer. A value of 0 specifies the Newton-Raphson method. Values greater than 0 specify to use the Fisher scoring algorithm up to iteration number  $n$ , where  $n$  is the specified integer, and Newton-Raphson thereafter.

**Scale parameter method** Select the method for the estimation of the scale parameter; the available options are:

- Maximum likelihood estimate
- Fixed value. You also set the **Value** to be used.
- Deviance
- Pearson Chi-square

**Negative Binomial method** Select the method for the estimation of the negative binomial ancillary parameter; the available options are:

- Maximum likelihood estimate
- Fixed value. You also set the **Value** to be used.

**Parameter Convergence** Convergence is assumed if the maximum absolute change or maximum relative change in the parameter estimates is less than the value specified, which must be non-negative. The criterion is not used if the value specified equals 0.

**Log-Likelihood Convergence** Convergence is assumed if the absolute change or relative change in the log-likelihood function is less than the value specified, which must be non-negative. The criterion is not used if the value specified equals 0.

**Hessian Convergence** For the **Absolute** specification, convergence is assumed if a statistic based on the Hessian is less than the value specified. For the **Relative** specification, convergence is assumed if the statistic is less than the product of the value specified and the absolute value of the log-likelihood. The criterion is not used if the value specified equals 0.

**Maximum iterations** You can specify the maximum number of iterations the algorithm will execute. The algorithm uses a doubly iterative process that consists of an inner loop and an outer loop. The value that is specified for the maximum number of iterations applies to both loops. Specify a non-negative integer. The default is 100.

**Singularity tolerance** This value is used as the tolerance in checking singularity. Specify a positive value.

**Note:** By default, **Parameter Convergence** is used, where the maximum **Absolute** change at a tolerance of 1E-6 is checked. This setting might produce results that differ from the results that are obtained in versions before version 17. To reproduce results from pre-17 versions, use **Relative** for the Parameter Convergence criterion and keep the default tolerance value of 1E-6.

## Model selection

**Use model selection or regularization** To activate the controls on this pane, select this check box.

**Method** Select the method of model selection or (if using **Ridge**) the regularization to be used. You can choose from the following options:

- **Lasso** Also known as L1 Regularization, this method is faster than Forward Stepwise if there are a large number of predictors. This method prevents overfitting by shrinking (that is, by imposing a penalty) on the parameters. It can shrink some parameters to zero, performing a variable selection lasso.
- **Ridge** Also known as L2 Regularization, this method prevents overfitting by shrinking (that is, by imposing a penalty) on the parameters. It shrinks all the parameters by the same proportions but eliminates none and is not a variable selection method.
- **Elastic Net** Also known as L1 + L2 Regularization, this method prevents overfitting by shrinking (that is, by imposing a penalty) on the parameters. It can shrink some parameters to zero, performing variable selection.
- **Forward Stepwise** This method starts with no effects in the model and adds or removes effects one step at a time until no more can be added or removed, according to the stepwise criteria.

**Automatically detect two-way interactions** To automatically detect two-way interactions, select this option.

### Penalty Parameters

These options are only available if you select either the Lasso or Elastic Net **Method**.

**Automatically select penalty parameters** If you are unsure what parameter penalties to set, select this check box and the node identifies and applies the penalties.

**Lasso penalty parameter** Enter the penalty parameter to be used by the Lasso model selection **Method**.

**Elastic net penalty parameter 1** Enter the L1 penalty parameter to be used by the Elastic Net model selection **Method**.

**Elastic net penalty parameter 2** Enter the L2 penalty parameter to be used by the Elastic Net model selection **Method**.

### Forward Stepwise

These options are only available if you select the Forward Stepwise **Method**.

**Include effects with p-value no less than** Specify the minimum probability value that effects can have to be included in the calculation.

**Remove effects with p-value greater than** Specify the maximum probability value that effects can have to be included in the calculation.

**Customize maximum number of effects in the final model** To activate the **Maximum number of effects** option, select this check box.

**Maximum number of effects** Specify the maximum number of effects when using the forward stepwise building method.

**Customize maximum number of steps** To activate the **Maximum number of steps** option, select this check box.

**Maximum number of steps** Specify the maximum number of steps when using the forward stepwise building method.

## Model options

**Model name** You can generate the model name automatically based on the target field, or specify a **Custom** name. The automatically generated name is the target field name. If there are multiple targets, then the model name is the field names in order, connected by ampersands. For example, if field1, field2, and field3 are targets, then the model name is: *field1 & field2 & field3*.

**Calculate predictor importance** For models that produce an appropriate measure of importance, you can display a chart that indicates the relative importance of each predictor in estimating the model. Typically you will want to focus your modeling efforts on the predictors that matter most, and consider dropping or ignoring those that matter least. Note that predictor importance may take longer to calculate for some models, particularly when working with large datasets, and is off by default for some models as a result.

For more information, see [“Predictor Importance” on page 37](#).

## GLE model nugget

### GLE model nugget output

After you create a GLE model, the following information is available in the output.

### Model Information table

The Model Information table provides key information about the model. The table identifies some high-level model settings, such as:

- The name of the target field selected in either the Type node or the GLE node Fields tab.
- The modeled and reference target category percentages.
- The probability distribution and associated link function.
- The model building method used.
- The number of predictors input and the number in the final model.
- The classification accuracy percentage.
- The model type.
- The percentage accuracy of the model, if the target is continuous.

### Records Summary

The summary table shows how many records were used to fit the model, and how many were excluded. The details shown include the number and percentage of the records included and excluded, as well as the unweighted number if you used frequency weighting.

## Predictor Importance

The Predictor Importance graph shows the importance of the top 10 inputs (predictors) in the model as a bar chart.

If there are more than 10 fields in the chart, you can change the selection of predictors that are included in the chart by using the slider beneath the chart. The indicator marks on the slider are a fixed width, and each mark on the slider represents 10 fields. You can move the indicator marks along the slider to display the next or previous 10 fields, ordered by predictor importance.

You can double-click the chart to open a separate dialog box in which you can edit the graph settings. For example, you can amend items such as the size of the graph, and the size and color of the fonts used. When you close this separate editing dialog box, the changes are applied to the chart that is displayed in the Output tab.

## Residual by Predicted plot

You can use this plot either to identify outliers, or to diagnose non-linearity or non constant error variance. An ideal plot will show the points randomly scattered about the zero line.

The expected pattern is that the distribution of standardized deviance residuals across predicted values of the linear predictor has a mean value of zero and a constant range. The expected pattern is a horizontal line through zero.

## GLE model nugget settings

On the Settings tab for a GLE model nugget, you specify options for raw propensity and for SQL generation during model scoring. This tab is available only after the model nugget is added to a stream.

**Calculate raw propensity scores** For models with flag targets only, you can request raw propensity scores that indicate the likelihood of the true outcome specified for the target field. These are in addition to standard prediction and confidence values. Adjusted propensity scores are not available.

**Generate SQL for this model** When using data from a database, SQL code can be pushed back to the database for execution, providing superior performance for many operations.

Select one of the following options to specify how SQL is generated:

- **Default: Score using Server Scoring Adapter (if installed) otherwise in process** If connected to a database with a scoring adapter installed, generates SQL by using the scoring adapter and associated user-defined functions (UDF) and scores your model within the database. When no scoring adapter is available, this option fetches your data back from the database and scores it in SPSS Modeler.
- **Score outside of the Database** If selected, this option fetches your data back from the database and scores it in SPSS Modeler.

## Cox Node

---

Cox Regression builds a predictive model for time-to-event data. The model produces a survival function that predicts the probability that the event of interest has occurred at a given time  $t$  for given values of the predictor variables. The shape of the survival function and the regression coefficients for the predictors are estimated from observed subjects; the model can then be applied to new cases that have measurements for the predictor variables. Note that information from censored subjects, that is, those that do not experience the event of interest during the time of observation, contributes usefully to the estimation of the model.

**Example.** As part of its efforts to reduce customer churn, a telecommunications company is interested in modeling the "time to churn" in order to determine the factors that are associated with customers who are quick to switch to another service. To this end, a random sample of customers is selected, and their time spent as customers (whether or not they are still active customers) and various demographic fields are pulled from the database.

**Requirements.** You need one or more input fields, exactly one target field, and you must specify a survival time field within the Cox node. The target field should be coded so that the "false" value indicates



survival and the "true" value indicates that the event of interest has occurred; it must have a measurement level of *Flag*, with string or integer storage. (Storage can be converted using a Filler or Derive node if necessary. ) Fields set to *Both* or *None* are ignored. Fields used in the model must have their types fully instantiated. The survival time can be any numeric field.

**Note:** On scoring a Cox Regression model, an error is reported if empty strings in categorical variables are used as input to model building. Avoid using empty strings as input.

**Dates & Times.** Date & Time fields cannot be used to directly define the survival time; if you have Date & Time fields, you should use them to create a field containing survival times, based upon the difference between the date of entry into the study and the observation date.

**Kaplan-Meier Analysis.** Cox regression can be performed with no input fields. This is equivalent to a Kaplan-Meier analysis.

## Cox Node Fields Options

**Survival time.** Choose a numeric field (one with a measurement level of *Continuous*) in order to make the node executable. Survival time indicates the lifespan of the record being predicted. For example, when modeling customer time to churn, this would be the field that records how long the customer has been with the organization. The date on which the customer joined or churned would not affect the model; only the duration of the customer's tenure would be relevant.

Survival time is taken to be a duration with no units. You must make sure that the input fields match the survival time. For example, in a study to measure churn by months, you would use sales per month as an input instead of sales per year. If your data has start and end dates instead of a duration, you must recode those dates to a duration upstream from the Cox node.

The remaining fields in this dialog box are the standard ones used throughout IBM SPSS Modeler. See the topic ["Modeling Node Fields Options"](#) on page 26 for more information.

## Cox Node Model Options

**Model name.** You can generate the model name automatically based on the target or ID field (or model type in cases where no such field is specified) or specify a custom name.

**Use partitioned data.** If a partition field is defined, this option ensures that data from only the training partition is used to build the model.

**Create split models.** Builds a separate model for each possible value of input fields that are specified as split fields. See ["Building Split Models"](#) on page 24 for more information.

**Method.** The following options are available for entering predictors into the model:

- **Enter.** This is the default method, which enters all of the terms into the model directly. No field selection is performed in building the model.
- **Stepwise.** The Stepwise method of field selection builds the model in steps, as the name implies. The initial model is the simplest model possible, with no model terms (except the constant) in the model. At each step, terms that have not yet been added to the model are evaluated, and if the best of those terms adds significantly to the predictive power of the model, it is added. In addition, terms that are currently in the model are reevaluated to determine if any of them can be removed without significantly detracting from the model. If so, they are removed. The process repeats, and other terms are added and/or removed. When no more terms can be added to improve the model, and no more terms can be removed without detracting from the model, the final model is generated.
- **Backwards Stepwise.** The Backwards Stepwise method is essentially the opposite of the Stepwise method. With this method, the initial model contains all of the terms as predictors. At each step, terms in the model are evaluated, and any terms that can be removed without significantly detracting from the model are removed. In addition, previously removed terms are reevaluated to determine if the best of those terms adds significantly to the predictive power of the model. If so, it is added back into the model. When no more terms can be removed without significantly detracting from the model, and no more terms can be added to improve the model, the final model is generated.

*Note:* The automatic methods, including Stepwise and Backwards Stepwise, are highly adaptable learning methods and have a strong tendency to overfit the training data. When using these methods, it is especially important to verify the validity of the resulting model either with new data or a hold-out test sample created using the Partition node.

**Groups.** Specifying a groups field causes the node to compute separate models for each category of the field. It can be any categorical field (Flag or Nominal) with string or integer storage.

**Model type.** There are two options for defining the terms in the model. **Main effects** models include only the input fields individually and do not test interactions (multiplicative effects) between input fields.

**Custom** models include only the terms (main effects and interactions) that you specify. When selecting this option, use the Model Terms list to add or remove terms in the model.

**Model Terms.** When building a Custom model, you will need to explicitly specify the terms in the model. The list shows the current set of terms for the model. The buttons on the right side of the Model Terms list allow you to add and remove model terms.

- To add terms to the model, click the *Add new model terms* button.
- To delete terms, select the desired terms and click the *Delete selected model terms* button.

### Adding Terms to a Cox Regression Model

When requesting a custom model, you can add terms to the model by clicking the *Add new model terms* button on the Model tab. A new dialog box opens in which you can specify terms.

**Type of term to add.** There are several ways to add terms to the model, based on the selection of input fields in the Available fields list.

- **Single interaction.** Inserts the term representing the interaction of all selected fields.
- **Main effects.** Inserts one main effect term (the field itself) for each selected input field.
- **All 2-way interactions.** Inserts a 2-way interaction term (the product of the input fields) for each possible pair of selected input fields. For example, if you have selected input fields *A*, *B*, and *C* in the Available fields list, this method will insert the terms  $A * B$ ,  $A * C$ , and  $B * C$ .
- **All 3-way interactions.** Inserts a 3-way interaction term (the product of the input fields) for each possible combination of selected input fields, taken three at a time. For example, if you have selected input fields *A*, *B*, *C*, and *D* in the Available fields list, this method will insert the terms  $A * B * C$ ,  $A * B * D$ ,  $A * C * D$ , and  $B * C * D$ .
- **All 4-way interactions.** Inserts a 4-way interaction term (the product of the input fields) for each possible combination of selected input fields, taken four at a time. For example, if you have selected input fields *A*, *B*, *C*, *D*, and *E* in the Available fields list, this method will insert the terms  $A * B * C * D$ ,  $A * B * C * E$ ,  $A * B * D * E$ ,  $A * C * D * E$ , and  $B * C * D * E$ .

**Available fields.** Lists the available input fields to be used in constructing model terms. Note that the list may include fields that are not legal input fields, so take care to ensure that all model terms include only input fields.

**Preview.** Shows the terms that will be added to the model if you click **Insert**, based on the selected fields and the term type selected above.

**Insert.** Inserts terms in the model (based on the current selection of fields and term type) and closes the dialog box.

## Cox Node Expert Options

**Convergence.** These options allow you to control the parameters for model convergence. When you execute the model, the convergence settings control how many times the different parameters are repeatedly run through to see how well they fit. The more often the parameters are tried, the closer the results will be (that is, the results will converge). See the topic [“Cox Node Convergence Criteria”](#) on page 207 for more information.

**Output.** These options allow you to request additional statistics and plots, including the survival curve, that will be displayed in the advanced output of the generated model built by the node. See the topic [“Cox Node Advanced Output Options”](#) on page 207 for more information.

**Stepping.** These options allow you to control the criteria for adding and removing fields with the Stepwise estimation method. (The button is disabled if the Enter method is selected.) See the topic [“Cox Node Stepping Criteria”](#) on page 207 for more information.

### Cox Node Convergence Criteria

**Maximum iterations.** Allows you to specify the maximum iterations for the model, which controls how long the procedure will search for a solution.

**Log-likelihood convergence.** Iterations stop if the relative change in the log-likelihood is less than this value. The criterion is not used if the value is 0.

**Parameter convergence.** Iterations stop if the absolute change or relative change in the parameter estimates is less than this value. The criterion is not used if the value is 0.

### Cox Node Advanced Output Options

**Statistics.** You can obtain statistics for your model parameters, including confidence intervals for  $\exp(B)$  and correlation of estimates. You can request these statistics either at each step or at the last step only.

**Display baseline function.** Allows you to display the baseline hazard function and cumulative survival at the mean of the covariates.

#### Plots

Plots can help you to evaluate your estimated model and interpret the results. You can plot the survival, hazard, log-minus-log, and one-minus-survival functions.

- *Survival.* Displays the cumulative survival function on a linear scale.
- *Hazard.* Displays the cumulative hazard function on a linear scale.
- **Log minus log.** Displays the cumulative survival estimate after the  $\ln(-\ln)$  transformation is applied to the estimate.
- *One minus survival.* Plots one-minus the survival function on a linear scale.

**Plot a separate line for each value.** This option is available only for categorical fields.

**Value to use for plots.** Because these functions depend on values of the predictors, you must use constant values for the predictors to plot the functions versus time. The default is to use the mean of each predictor as a constant value, but you can enter your own values for the plot using the grid. For categorical inputs, indicator coding is used, so there is a regression coefficient for each category (except the last). Thus, a categorical input has a mean value for each indicator contrast, equal to the proportion of cases in the category corresponding to the indicator contrast.

### Cox Node Stepping Criteria

**Removal criterion.** Select **Likelihood Ratio** for a more robust model. To shorten the time required to build the model, you can try selecting **Wald**. There is the additional option **Conditional**, which provides removal testing based on the probability of the likelihood-ratio statistic based on conditional parameter estimates.

**Significance thresholds for criteria.** This option allows you to specify selection criteria based on the statistical probability (the  $p$  value) associated with each field. Fields will be added to the model only if the associated  $p$  value is smaller than the **Entry** value and will be removed only if the  $p$  value is larger than the **Removal** value. The **Entry** value must be smaller than the **Removal** value.

## Cox Node Settings Options

**Predict survival at future times.** Specify one or more future times. Survival, that is, whether each case is likely to survive for at least that length of time (from now) without the terminal event occurring, is

predicted for each record at each time value, one prediction per time value. Note that survival is the "false" value of the target field.

- **Regular intervals.** Survival time values are generated from the specified **Time interval** and **Number of time periods to score**. For example, if 3 time periods are requested with an interval of 2 between each time, survival will be predicted for future times 2, 4, 6. Every record is evaluated at the same time values.
- **Time fields.** Survival times are provided for each record in the time field chosen (one prediction field is generated), thus each record can be evaluated at different times.

**Past survival time.** Specify the survival time of the record so far—for example, the tenure of an existing customer as a field. Scoring the likelihood of survival at a future time will be conditional on past survival time.

*Note:* The values of future and past survival times must be within range of survival times in the data used to train the model. Records whose times fall outside this range are scored as null.

**Append all probabilities.** Specifies whether probabilities for each category of the output field are added to each record processed by the node. If this option is not selected, the probability of only the predicted category is added. Probabilities are computed for each future time.

**Calculate cumulative hazard function.** Specifies whether the value of the cumulative hazard is added to each record. The cumulative hazard is computed for each future time.

## Cox Model Nugget

Cox regression models represent the equations estimated by Cox nodes. They contain all of the information captured by the model, as well as information about the model structure and performance.

When you run a stream containing a generated Cox regression model, the node adds two new fields containing the model's prediction and the associated probability. The names of the new fields are derived from the name of the output field being predicted, prefixed with *\$C-* for the predicted category and *\$CP-* for the associated probability, suffixed with the number of the future time interval or the name of the time field that defines the time interval. For example, for an output field named *churn* and two future time intervals defined at regular intervals, the new fields would be named *\$C-churn-1*, *\$CP-churn-1*, *\$C-churn-2*, and *\$CP-churn-2*. If future times are defined with a time field *tenure*, the new fields would be *\$C-churn\_tenure* and *\$CP-churn\_tenure*.

If you have selected the **Append all probabilities** settings option in the Cox node, two additional fields will be added for each future time, containing the probabilities of survival and failure for each record. These additional fields are named based on the name of the output field, prefixed by *\$CP-<false value>-* for the probability of survival and *\$CP-<true value>-* for the probability the event has occurred, suffixed with the number of the future time interval. For example, for an output field where the "false" value is 0 and the "true" value is 1, and two future time intervals defined at regular intervals, the new fields would be named *\$CP-0-1*, *\$CP-1-1*, *\$CP-0-2*, and *\$CP-1-2*. If future times are defined with a single time field *tenure*, the new fields would be *\$CP-0-1* and *\$CP-1-1*, since there is a single future interval.

If you have selected the **Calculate cumulative hazard function** settings option in the Cox Node, an additional field will be added for each future time, containing the cumulative hazard function for each record. These additional fields are named based on the name of the output field, prefixed by *\$CH-*, suffixed with the number of the future time interval or the name of the time field that defines the time interval. For example, for an output field named *churn* and two future time intervals defined at regular intervals, the new fields would be named *\$CH-churn-1* and *\$CH-churn-2*. If future times are defined with a time field *tenure*, the new field would be *\$CH-churn-1*.

### Cox Regression Output Settings

Except for SQL generation, the Settings tab of the nugget contains the same controls as the Settings tab of the model node. The default values of the nugget controls are determined by the values set in the model node. See the topic [“Cox Node Settings Options” on page 207](#) for more information.

**Generate SQL for this model** When using data from a database, SQL code can be pushed back to the database for execution, providing superior performance for many operations.

Select one of the following options to specify how SQL generation is performed.

- **Default: Score using Server Scoring Adapter (if installed) otherwise in process** If connected to a database with a scoring adapter installed, generates SQL using the scoring adapter and associated user defined functions (UDF) and scores your model within the database. When no scoring adapter is available, this option fetches your data back from the database and scores it in SPSS Modeler.
- **Score outside of the Database** If selected, this option fetches your data back from the database and scores it in SPSS Modeler.

### **Cox Regression Advanced Output**

The advanced output for Cox regression gives detailed information about the estimated model and its performance, including the survival curve. Most of the information contained in the advanced output is quite technical, and extensive knowledge of Cox regression is required to properly interpret this output.



## Chapter 11. Clustering models

Clustering models focus on identifying groups of similar records and labeling the records according to the group to which they belong. This is done without the benefit of prior knowledge about the groups and their characteristics. In fact, you may not even know exactly how many groups to look for. This is what distinguishes clustering models from the other machine-learning techniques—there is no predefined output or target field for the model to predict. These models are often referred to as **unsupervised learning** models, since there is no external standard by which to judge the model's classification performance. There are no *right* or *wrong* answers for these models. Their value is determined by their ability to capture interesting groupings in the data and provide useful descriptions of those groupings.

Clustering methods are based on measuring distances between records and between clusters. Records are assigned to clusters in a way that tends to minimize the distance between records belonging to the same cluster.

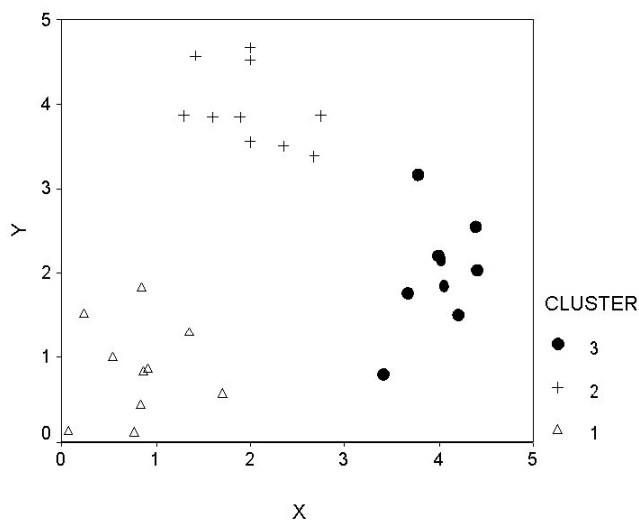


Figure 44. Simple clustering model

The following clustering methods are provided:



The K-Means node clusters the data set into distinct groups (or clusters). The method defines a fixed number of clusters, iteratively assigns records to clusters, and adjusts the cluster centers until further refinement can no longer improve the model. Instead of trying to predict an outcome, *k*-means uses a process known as unsupervised learning to uncover patterns in the set of input fields.



The TwoStep node uses a two-step clustering method. The first step makes a single pass through the data to compress the raw input data into a manageable set of subclusters. The second step uses a hierarchical clustering method to progressively merge the subclusters into larger and larger clusters. TwoStep has the advantage of automatically estimating the optimal number of clusters for the training data. It can handle mixed field types and large data sets efficiently.



The Kohonen node generates a type of neural network that can be used to cluster the data set into distinct groups. When the network is fully trained, records that are similar should be close together on the output map, while records that are different will be far apart. You can look at the number of observations captured by each unit in the model nugget to identify the strong units. This may give you a sense of the appropriate number of clusters.



Hierarchical Density-Based Spatial Clustering (HDBSCAN)<sup>®</sup> uses unsupervised learning to find clusters, or dense regions, of a data set. The HDBSCAN node in SPSS Modeler exposes the core features and commonly used parameters of the HDBSCAN library. The node is implemented in Python, and you can use it to cluster your dataset into distinct groups when you don't know what those groups are at first.

Clustering models are often used to create clusters or segments that are then used as inputs in subsequent analyses. A common example of this is the market segments used by marketers to partition their overall market into homogeneous subgroups. Each segment has special characteristics that affect the success of marketing efforts targeted toward it. If you are using data mining to optimize your marketing strategy, you can usually improve your model significantly by identifying the appropriate segments and using that segment information in your predictive models.

## Kohonen node

---

Kohonen networks are a type of neural network that perform clustering, also known as a **knet** or a **self-organizing map**. This type of network can be used to cluster the dataset into distinct groups when you don't know what those groups are at the beginning. Records are grouped so that records within a group or cluster tend to be similar to each other, and records in different groups are dissimilar.

The basic units are **neurons**, and they are organized into two layers: the **input layer** and the **output layer** (also called the **output map**). All of the input neurons are connected to all of the output neurons, and these connections have **strengths**, or **weights**, associated with them. During training, each unit competes with all of the others to "win" each record.

The output map is a two-dimensional grid of neurons, with no connections between the units.

Input data is presented to the input layer, and the values are propagated to the output layer. The output neuron with the strongest response is said to be the **winner** and is the answer for that input.

Initially, all weights are random. When a unit wins a record, its weights (along with those of other nearby units, collectively referred to as a **neighborhood**) are adjusted to better match the pattern of predictor values for that record. All of the input records are shown, and weights are updated accordingly. This process is repeated many times until the changes become very small. As training proceeds, the weights on the grid units are adjusted so that they form a two-dimensional "map" of the clusters (hence the term **self-organizing map**).

When the network is fully trained, records that are similar should be close together on the output map, whereas records that are vastly different will be far apart.

Unlike most learning methods in IBM SPSS Modeler, Kohonen networks do *not* use a target field. This type of learning, with no target field, is called **unsupervised learning**. Instead of trying to predict an outcome, Kohonen nets try to uncover patterns in the set of input fields. Usually, a Kohonen net will end up with a few units that summarize many observations (**strong** units), and several units that don't really correspond to any of the observations (**weak** units). The strong units (and sometimes other units adjacent to them in the grid) represent probable cluster centers.

Another use of Kohonen networks is in **dimension reduction**. The spatial characteristic of the two-dimensional grid provides a mapping from the  $k$  original predictors to two derived features that preserve the similarity relationships in the original predictors. In some cases, this can give you the same kind of benefit as factor analysis or PCA.



Note that the method for calculating default size of the output grid has changed from previous versions of IBM SPSS Modeler. The new method will generally produce smaller output layers that are faster to train and generalize better. If you find that you get poor results with the default size, try increasing the size of the output grid on the Expert tab. See the topic [“Kohonen Node Expert Options”](#) on page 214 for more information.

**Requirements.** To train a Kohonen net, you need one or more fields with the role set to *Input*. Fields with the role set to *Target*, *Both*, or *None* are ignored.

**Strengths.** You do not need to have data on group membership to build a Kohonen network model. You don't even need to know the number of groups to look for. Kohonen networks start with a large number of units, and as training progresses, the units gravitate toward the natural clusters in the data. You can look at the number of observations captured by each unit in the model nugget to identify the strong units, which can give you a sense of the appropriate number of clusters.

## Kohonen Node Model Options

**Model name.** You can generate the model name automatically based on the target or ID field (or model type in cases where no such field is specified) or specify a custom name.

**Use partitioned data.** If a partition field is defined, this option ensures that data from only the training partition is used to build the model.

**Continue training existing model.** By default, each time you execute a Kohonen node, a completely new network is created. If you select this option, training continues with the last net successfully produced by the node.

**Show feedback graph.** If this option is selected, a visual representation of the two-dimensional array is displayed during training. The strength of each node is represented by color. Red denotes a unit that is winning many records (a **strong** unit), and white denotes a unit that is winning few or no records (a **weak** unit). Feedback may not display if the time taken to build the model is relatively short. Note that this feature can slow training time. To speed up training time, deselect this option.

**Stop on.** The default stopping criterion stops training, based on internal parameters. You can also specify time as the stopping criterion. Enter the time (in minutes) for the network to train.

**Set random seed.** If no random seed is set, the sequence of random values used to initialize the network weights will be different every time the node is executed. This can cause the node to create different models on different runs, even if the node settings and data values are exactly the same. By selecting this option, you can set the random seed to a specific value so the resulting model is exactly reproducible. A specific random seed always generates the same sequence of random values, in which case executing the node always yields the same generated model.

**Note:** When using the **Set random seed** option with records read from a database, a Sort node may be required prior to sampling in order to ensure the same result each time the node is executed. This is because the random seed depends on the order of records, which is not guaranteed to stay the same in a relational database.

**Note:** If you want to include nominal (set) fields in your model but are having memory problems in building the model, or the model is taking too long to build, consider recoding large set fields to reduce the number of values, or consider using a different field with fewer values as a proxy for the large set. For example, if you are having a problem with a *product\_id* field containing values for individual products, you might consider removing it from the model and adding a less detailed *product\_category* field instead.

**Optimize.** Select options designed to increase performance during model building based on your specific needs.

- Select **Speed** to instruct the algorithm to never use disk spilling in order to improve performance.
- Select **Memory** to instruct the algorithm to use disk spilling when appropriate at some sacrifice to speed. This option is selected by default.

**Note:** When running in distributed mode, this setting can be overridden by administrator options specified in `options.cfg`.

**Append cluster label.** Selected by default for new models, but deselected for models loaded from earlier versions of IBM SPSS Modeler, this creates a single categorical score field of the same type that is created by both the K-Means and TwoStep nodes. This string field is used in the Auto Cluster node when calculating ranking measures for the different model types. See the topic [“Auto Cluster node”](#) on page 64 for more information.

## Kohonen Node Expert Options

For those with detailed knowledge of Kohonen networks, expert options allow you to fine-tune the training process. To access expert options, set the Mode to **Expert** on the Expert tab.

**Width and Length.** Specify the size (width and length) of the two-dimensional output map as number of output units along each dimension.

**Learning rate decay.** Select either linear or exponential learning rate decay. The **learning rate** is a weighting factor that decreases over time, such that the network starts off encoding large-scale features of the data and gradually focuses on more fine-level detail.

**Phase 1 and Phase 2.** Kohonen net training is split into two phases. Phase 1 is a rough estimation phase, used to capture the gross patterns in the data. Phase 2 is a tuning phase, used to adjust the map to model the finer features of the data. For each phase, there are three parameters:

- **Neighborhood.** Sets the starting size (radius) of the neighborhood. This determines the number of "nearby" units that get updated along with the winning unit during training. During phase 1, the neighborhood size starts at *Phase 1 Neighborhood* and decreases to  $(\text{Phase 2 Neighborhood} + 1)$ . During phase 2, neighborhood size starts at *Phase 2 Neighborhood* and decreases to 1.0. *Phase 1 Neighborhood* should be larger than *Phase 2 Neighborhood*.
- **Initial Eta.** Sets the starting value for learning rate **eta**. During phase 1, eta starts at *Phase 1 Initial Eta* and decreases to *Phase 2 Initial Eta*. During phase 2, eta starts at *Phase 2 Initial Eta* and decreases to 0. *Phase 1 Initial Eta* should be larger than *Phase 2 Initial Eta*.
- **Cycles.** Sets the number of cycles for each phase of training. Each phase continues for the specified number of passes through the data.

## Kohonen Model Nuggets

Kohonen model nuggets contain all of the information captured by the trained Kohonen network, as well as information about the network's architecture.

When you run a stream containing a Kohonen model nugget, the node adds two new fields containing the *X* and *Y* coordinates of the unit in the Kohonen output grid that responded most strongly to that record. The new field names are derived from the model name, prefixed by *\$KX-* and *\$KY-*. For example, if your model is named *Kohonen*, the new fields would be named *\$KX-Kohonen* and *\$KY-Kohonen*.

To get a better sense of what the Kohonen net has encoded, click the Model tab on the model nugget browser. This displays the Cluster Viewer, providing a graphical representation of clusters, fields, and importance levels. See the topic [“Cluster Viewer - Model Tab”](#) on page 225 for more information.

If you prefer to visualize the clusters as a grid, you can view the result of the Kohonen net by plotting the *\$KX-* and *\$KY-* fields using a Plot node. (You should select **X-Agitation** and **Y-Agitation** in the Plot node to prevent each unit's records from all being plotted on top of each other.) In the plot, you can also overlay a symbolic field to investigate how the Kohonen net has clustered the data.

Another powerful technique for gaining insight into the Kohonen network is to use rule induction to discover the characteristics that distinguish the clusters found by the network. See the topic [“C5.0 Node”](#) on page 91 for more information.

For general information on using the model browser, see [“Browsing model nuggets”](#) on page 35

## Kohonen Model Summary

The Summary tab for a Kohonen model nugget displays information about the architecture or topology of the network. The length and width of the two-dimensional Kohonen feature map (the output layer) are

shown as **\$KX-** *model\_name* and **\$KY-** *model\_name*. For the input and output layers, the number of units in that layer is listed.

## K-Means Node

---

The K-Means node provides a method of **cluster analysis**. It can be used to cluster the dataset into distinct groups when you don't know what those groups are at the beginning. Unlike most learning methods in IBM SPSS Modeler, K-Means models do *not* use a target field. This type of learning, with no target field, is called **unsupervised learning**. Instead of trying to predict an outcome, K-Means tries to uncover patterns in the set of input fields. Records are grouped so that records within a group or cluster tend to be similar to each other, but records in different groups are dissimilar.

K-Means works by defining a set of starting cluster centers derived from data. It then assigns each record to the cluster to which it is most similar, based on the record's input field values. After all cases have been assigned, the cluster centers are updated to reflect the new set of records assigned to each cluster. The records are then checked again to see whether they should be reassigned to a different cluster, and the record assignment/cluster iteration process continues until either the maximum number of iterations is reached, or the change between one iteration and the next fails to exceed a specified threshold.

*Note:* The resulting model depends to a certain extent on the order of the training data. Reordering the data and rebuilding the model may lead to a different final cluster model.

**Requirements.** To train a K-Means model, you need one or more fields with the role set to *Input*. Fields with the role set to *Output*, *Both*, or *None* are ignored.

**Strengths.** You do not need to have data on group membership to build a K-Means model. The K-Means model is often the fastest method of clustering for large datasets.

### K-Means Node Model Options

**Model name.** You can generate the model name automatically based on the target or ID field (or model type in cases where no such field is specified) or specify a custom name.

**Use partitioned data.** If a partition field is defined, this option ensures that data from only the training partition is used to build the model.

**Specified number of clusters.** Specify the number of clusters to generate. The default is 5.

**Generate distance field.** If this option is selected, the model nugget will include a field containing the distance of each record from the center of its assigned cluster.

**Cluster label.** Specify the format for the values in the generated cluster membership field. Cluster membership can be indicated as a **String** with the specified **Label prefix** (for example "Cluster 1", "Cluster 2", and so on), or as a **Number**.

*Note:* If you want to include nominal (set) fields in your model but are having memory problems in building the model or the model is taking too long to build, consider recoding large set fields to reduce the number of values, or consider using a different field with fewer values as a proxy for the large set. For example, if you are having a problem with a *product\_id* field containing values for individual products, you might consider removing it from the model and adding a less detailed *product\_category* field instead.

**Optimize.** Select options designed to increase performance during model building based on your specific needs.

- Select **Speed** to instruct the algorithm to never use disk spilling in order to improve performance.
- Select **Memory** to instruct the algorithm to use disk spilling when appropriate at some sacrifice to speed. This option is selected by default.

**Note:** When running in distributed mode, this setting can be overridden by administrator options specified in `options.cfg`.

## K-Means Node Expert Options

For those with detailed knowledge of *k*-means clustering, expert options allow you to fine-tune the training process. To access expert options, set the Mode to **Expert** on the Expert tab.

**Stop on.** Specify the stopping criterion to be used in training the model. The **Default** stopping criterion is 20 iterations or change < 0.000001, whichever occurs first. Select **Custom** to specify your own stopping criteria.

- **Maximum Iterations.** This option allows you to stop model training after the number of iterations specified.
- **Change tolerance.** This option allows you to stop model training when the largest change in cluster centers for an iteration is less than the level specified.

**Encoding value for sets.** Specify a value between 0 and 1.0 to use for recoding set fields as groups of numeric fields. The default value is the square root of 0.5 (approximately 0.707107), which provides the proper weighting for recoded flag fields. Values closer to 1.0 will weight set fields more heavily than numeric fields.

## K-Means Model Nuggets

---

K-Means model nuggets contain all of the information captured by the clustering model, as well as information about the training data and the estimation process.

When you run a stream containing a K-Means modeling node, the node adds two new fields containing the cluster membership and distance from the assigned cluster center for that record. The new field names are derived from the model name, prefixed by *\$KM-* for the cluster membership and *\$KMD-* for the distance from the cluster center. For example, if your model is named *Kmeans*, the new fields would be named *\$KM-Kmeans* and *\$KMD-Kmeans*.

A powerful technique for gaining insight into the K-Means model is to use rule induction to discover the characteristics that distinguish the clusters found by the model. See the topic [“C5.0 Node” on page 91](#) for more information. You can also click the Model tab on the model nugget browser to display the Cluster Viewer, providing a graphical representation of clusters, fields, and importance levels. See the topic [“Cluster Viewer - Model Tab” on page 225](#) for more information.

For general information on using the model browser, see [“Browsing model nuggets” on page 35](#)

## K-Means Model Summary

The Summary tab for a K-Means model nugget contains information about the training data, the estimation process, and the clusters defined by the model. The number of clusters is shown, as well as the iteration history. If you have executed an Analysis node attached to this modeling node, information from that analysis will also be displayed in this section.

## TwoStep Cluster node

---

The TwoStep Cluster node provides a form of **cluster analysis**. It can be used to cluster the dataset into distinct groups when you don't know what those groups are at the beginning. As with Kohonen nodes and K-Means nodes, TwoStep Cluster models do *not* use a target field. Instead of trying to predict an outcome, TwoStep Cluster tries to uncover patterns in the set of input fields. Records are grouped so that records within a group or cluster tend to be similar to each other, but records in different groups are dissimilar.

TwoStep Cluster is a two-step clustering method. The first step makes a single pass through the data, during which it compresses the raw input data into a manageable set of subclusters. The second step uses a hierarchical clustering method to progressively merge the subclusters into larger and larger clusters, without requiring another pass through the data. Hierarchical clustering has the advantage of not requiring the number of clusters to be selected ahead of time. Many hierarchical clustering methods start with individual records as starting clusters and merge them recursively to produce ever larger clusters.

Though such approaches often break down with large amounts of data, TwoStep's initial preclustering makes hierarchical clustering fast even for large datasets.

**Note:** The resulting model depends to a certain extent on the order of the training data. Reordering the data and rebuilding the model may lead to a different final cluster model.

**Requirements.** To train a TwoStep Cluster model, you need one or more fields with the role set to *Input*. Fields with the role set to *Target*, *Both*, or *None* are ignored. The TwoStep Cluster algorithm does not handle missing values. Records with blanks for any of the input fields will be ignored when building the model.

**Strengths.** TwoStep Cluster can handle mixed field types and is able to handle large datasets efficiently. It also has the ability to test several cluster solutions and choose the best, so you don't need to know how many clusters to ask for at the outset. TwoStep Cluster can be set to automatically exclude **outliers**, or extremely unusual cases that can contaminate your results.

IBM SPSS Modeler has two different versions of the TwoStep Cluster node:

- **TwoStep Cluster** is the traditional node that runs on the IBM SPSS Modeler Server.
- **TwoStep-AS Cluster** can run when connected to IBM SPSS Analytic Server.

## TwoStep Cluster Node Model Options

**Model name.** You can generate the model name automatically based on the target or ID field (or model type in cases where no such field is specified) or specify a custom name.

**Use partitioned data.** If a partition field is defined, this option ensures that data from only the training partition is used to build the model.

**Standardize numeric fields.** By default, TwoStep will standardize all numeric input fields to the same scale, with a mean of 0 and a variance of 1. To retain the original scaling for numeric fields, deselect this option. Symbolic fields are not affected.

**Exclude outliers.** If you select this option, records that don't seem to fit into a substantive cluster will be automatically excluded from the analysis. This prevents such cases from distorting the results.

Outlier detection occurs during the preclustering step. When this option is selected, subclusters with few records relative to other subclusters are considered potential outliers, and the tree of subclusters is rebuilt excluding those records. The size below which subclusters are considered to contain potential outliers is controlled by the **Percentage** option. Some of those potential outlier records can be added to the rebuilt subclusters if they are similar enough to any of the new subcluster profiles. The rest of the potential outliers that cannot be merged are considered outliers and are added to a "noise" cluster and excluded from the hierarchical clustering step.

When *scoring* data with a TwoStep model that uses outlier handling, new cases that are more than a certain threshold distance (based on the log-likelihood) from the nearest substantive cluster are considered outliers and are assigned to the "noise" cluster with the name -1.

**Cluster label.** Specify the format for the generated cluster membership field. Cluster membership can be indicated as a **String** with the specified **Label prefix** (for example, "Cluster 1", "Cluster 2", and so on) or as a **Number**.

**Automatically calculate number of clusters.** TwoStep cluster can very rapidly analyze a large number of cluster solutions to choose the optimal number of clusters for the training data. Specify a range of solutions to try by setting the **Maximum** and the **Minimum** number of clusters. TwoStep uses a two-stage process to determine the optimal number of clusters. In the first stage, an upper bound on the number of clusters in the model is selected based on the change in the Bayes Information Criterion (BIC) as more clusters are added. In the second stage, the change in the minimum distance between clusters is found for all models with fewer clusters than the minimum-BIC solution. The largest change in distance is used to identify the final cluster model.

**Specify number of clusters.** If you know how many clusters to include in your model, select this option and enter the number of clusters.

**Distance measure.** This selection determines how the similarity between two clusters is computed.

- **Log-likelihood.** The likelihood measure places a probability distribution on the variables. Continuous variables are assumed to be normally distributed, while categorical variables are assumed to be multinomial. All variables are assumed to be independent.
- **Euclidean.** The Euclidean measure is the "straight line" distance between two clusters. It can be used only when all of the variables are continuous.

**Clustering Criterion.** This selection determines how the automatic clustering algorithm determines the number of clusters. Either the Bayesian Information Criterion (BIC) or the Akaike Information Criterion (AIC) can be specified.

## TwoStep Cluster Model Nuggets

---

TwoStep cluster model nuggets contain all of the information captured by the clustering model, as well as information about the training data and the estimation process.

When you run a stream containing a TwoStep cluster model nugget, the node adds a new field containing the cluster membership for that record. The new field name is derived from the model name, prefixed by \$T-. For example, if your model is named *TwoStep*, the new field would be named *\$T-TwoStep*.

A powerful technique for gaining insight into the TwoStep model is to use rule induction to discover the characteristics that distinguish the clusters found by the model. See the topic [“C5.0 Node” on page 91](#) for more information. You can also click the Model tab on the model nugget browser to display the Cluster Viewer, providing a graphical representation of clusters, fields, and importance levels. See the topic [“Cluster Viewer - Model Tab” on page 225](#) for more information.

For general information on using the model browser, see [“Browsing model nuggets” on page 35](#)

## TwoStep Model Summary

The Summary tab for a TwoStep cluster model nugget displays the number of clusters found, along with information about the training data, the estimation process, and build settings used.

See the topic [“Browsing model nuggets” on page 35](#) for more information.

## TwoStep-AS Cluster node

---

IBM SPSS Modeler has two different versions of the TwoStep Cluster node:

- **TwoStep Cluster** is the traditional node that runs on the IBM SPSS Modeler Server.
- **TwoStep-AS Cluster** can run when connected to IBM SPSS Analytic Server.

## Twostep-AS cluster analysis

TwoStep Cluster is an exploratory tool that is designed to reveal natural groupings (or clusters) within a data set that would otherwise not be apparent. The algorithm that is employed by this procedure has several desirable features that differentiate it from traditional clustering techniques:

- **Handling of categorical and continuous variables.** By assuming variables to be independent, a joint multinomial-normal distribution can be placed on categorical and continuous variables.
- **Automatic selection of number of clusters.** By comparing the values of a model-choice criterion across different clustering solutions, the procedure can automatically determine the optimal number of clusters.
- **Scalability.** By constructing a cluster feature (CF) tree that summarizes the records, the TwoStep algorithm can analyze large data files.

For example, retail and consumer product companies regularly apply clustering techniques to information that describes their customers' buying habits, gender, age, income level, and other attributes. These companies tailor their marketing and product development strategies to each consumer group to increase sales and build brand loyalty.

## Fields tab

The Fields tab specifies which fields are used in the analysis.

**Use predefined roles.** All fields with a defined role of Input are selected.

**Use custom field assignments.** Add and remove fields regardless of their defined role assignments. You can select fields with any role and move them in or out of the **Predictors (Inputs)** list.

## Basics

### Number of Clusters

#### Determine automatically

The procedure determines the best number of clusters, within the specified range. The **Minimum** must be greater than 1. This is the default option.

#### Specify fixed

The procedure generates the specified number of clusters. The **Number** must be greater than 1.

### Clustering Criterion

This selection controls how the automatic clustering algorithm determines the number of clusters.

#### Bayesian Information Criterion (BIC)

A measure for selecting and comparing models based on the -2 log likelihood. Smaller values indicate better models. The BIC also "penalizes" overparameterized models (complex models with a large number of inputs, for example), but more strictly than the AIC.

#### Akaike Information Criterion (AIC)

A measure for selecting and comparing models based on the -2 log likelihood. Smaller values indicate better models. The AIC "penalizes" overparameterized models (complex models with a large number of inputs, for example).

### Automatic Clustering Method

If you select **Determine automatically**, choose from the following clustering methods used to automatically determine the number of clusters:

#### Use Clustering Criterion setting

Information criteria convergence is the ratio of information criteria corresponding to two current cluster solutions and the first cluster solution. The criterion used is the one selected in the Clustering Criterion group.

#### Distance jump

Distance jump is the ratio of distances corresponding to two consecutive cluster solutions.

#### Maximum

Combine results from the information criteria convergence method and the distance jump method to produce the number of clusters corresponding to the second jump.

#### Minimum

Combine results from the information criteria convergence method and the distance jump method to produce the number of clusters corresponding to the first jump.

### Feature Importance Method

**Feature Importance Method** determines how important the features (fields) are in the cluster solution. The output includes information about overall feature importance and the importance of each feature field in each cluster. Features that do not meet a minimum threshold are excluded.

#### Use Clustering Criterion setting.

This is the default method, based on the criterion that is selected in the Clustering Criterion group.

#### Effect size

Feature importance is based on effect size instead of significance values.

## Feature Tree Criteria

These settings determine how the cluster feature tree is built. By building a cluster feature tree and summarizing the records, the TwoStep algorithm can analyze large data files. In other words, TwoStep Cluster uses a cluster feature tree to build clusters, enabling it to process many cases.

## Distance Measure

This selection determines how the similarity between two clusters is computed.

## Log-likelihood

The likelihood measure places a probability distribution on the fields. Continuous fields are assumed to be normally distributed, while categorical fields are assumed to be multinomial. All fields are assumed to be independent.

## Euclidean

The Euclidean measure is the "straight line" distance between two clusters. Squared Euclidean measure and the Ward method are used to compute similarity between clusters. It can be used only when all of the fields are continuous.

## Outlier Clusters

### Include outlier clusters

Include clusters for cases that are outliers from the regular clusters. If this option is not selected, all cases are included in regular clusters.

### Number of cases in feature tree leaf is less than.

If the number of cases in the feature tree leaf is less than the specified value, the leaf is considered an outlier. The value must be an integer greater than 1. If you change this value, higher values are likely to result in more outlier clusters.

### Top percentage of outliers.

When the cluster model is built, outliers are ranked by outlier strength. The outlier strength that is required to be in the top percentage of outliers is used as the threshold for determining whether a case is classified as an outlier. Higher values mean that more cases are classified as outliers. The value must be between 1 - 100.

## Additional settings

### Initial distance change threshold

The initial threshold that is used to grow the cluster feature tree. If insertion of a leaf into a leaf of the tree yields tightness less than this threshold, the leaf is not split. If the tightness exceeds this threshold, the leaf is split.

### Leaf node maximum branches

The maximum number of child nodes that a leaf node can have.

### Non-leaf node maximum branches

The maximum number of child nodes that a non-leaf node can have.

### Maximum tree depth

The maximum number of levels that the cluster tree can have.

### Adjustment weight on measurement level

Reduces the influence of categorical fields by increasing the weight for continuous fields. This value represents a denominator for reducing the weight for categorical fields. So a default of 6, for example, gives categorical fields a weight of  $1/6$ .



## Memory allocation

The maximum amount of memory in megabytes (MB) that the cluster algorithm uses. If the procedure exceeds this maximum, it uses the disk to store information that does not fit in memory.

## Delayed split

Delay rebuilding of the cluster feature tree. The clustering algorithm rebuilds the cluster feature tree multiple times as it evaluates new cases. This option can improve performance by delaying that operation and reducing the number of times the tree is rebuilt.

## Standardize

The clustering algorithm works with standardized continuous fields. By default, all continuous fields are standardized. To save some time and computational effort, you can move continuous fields that are already standardized to the **Do not standardize** list.

## Feature Selection

On the Feature Selection screen, you can set rules that determine when fields are excluded. For example, you can exclude fields that have numerous missing values.

### Rules for Excluding Fields

#### Percentage of missing values is greater than.

Fields with a percentage of missing values greater than the specified value are excluded from the analysis. The value must be a positive number greater than zero and less than 100.

#### Number of categories for categorical fields is greater than.

Categorical fields with more than the specified number of categories are excluded from the analysis. The value must be a positive integer greater than 1.

#### Fields with a Tendency Toward a Single Value

##### Coefficient of variation for continuous fields is less than.

Continuous fields with a coefficient of variation less than the specified value are excluded from the analysis. The coefficient of variation is the ratio of the standard deviation to the mean. Lower values tend to indicate less variation in values. The value must be between 0 and 1.

##### Percentage of cases in a single category for categorical fields is greater than.

Categorical fields with a percentage of cases in a single category greater than the specified value are excluded from the analysis. The value must be greater than 0 and less than 100.

## Adaptive feature selection

This option runs an extra data pass to find and remove the least important fields.

## Model Output

### Model Building Summary

#### Model specifications

Summary of model specifications, number of clusters in the final model, and inputs (fields) included in the final model.

#### Record summary

Number and percentage of records (cases) included and excluded from the model.

#### Excluded inputs

For any fields not included in the final model, the reason the field was excluded.

## Evaluation

### Model quality

Table of goodness and importance for each cluster and overall model goodness of fit.

**Feature importance bar chart**

Bar chart of feature (field) importance across all clusters. Features (fields) with longer bars in the chart are more important than fields with shorter bars. They are also sorted in descending order of importance (the bar at the top is the most important).

**Feature importance word cloud**

Word cloud of feature (field) importance across all clusters. Features (fields) with larger text are more important than those with smaller text.

**Outlier clusters**

These options are disabled if you chose not to include outliers.

**Interactive table and chart**

Table and chart of outlier strength and the relative similarity of outlier clusters to regular clusters. Selecting different rows in the table displays information for different outlier clusters in the chart.

**Pivot table**

Table of outlier strength and the relative similarity of outlier clusters to regular clusters. This table contains the same information as the interactive display. This table supports all standard features for pivoting and editing tables.

**Maximum number**

The maximum number of outliers to display in the output. If there are more than twenty outlier clusters, a pivot table will be displayed instead.

**Interpretation****Across cluster feature importance profiles****Interactive table and chart.**

Table and charts of feature importance and cluster centers for each input (field) used in the cluster solution. Selecting different rows in the table displays a different chart. For categorical fields, a bar chart is displayed. For continuous fields, a chart of means and standard deviations is displayed.

**Pivot table.**

Table of feature importance and cluster centers for each input (field). This table contains the same information as the interactive display. This table supports all standard features for pivoting and editing tables.

**Within cluster feature importance**

For each cluster, the cluster center and feature importance for each input (field). There is a separate table for each cluster.

**Cluster distances**

A panel chart that displays the distances between clusters. There is a separate panel for each cluster.

**Cluster label****Text**

The label for each cluster is the value that is specified for **Prefix**, followed by a sequential number.

**Number**

The label for each cluster is a sequential number.

**Model Options**

**Model name.** You can generate the model name automatically based on the target or ID field (or model type in cases where no such field is specified) or specify a custom name.

## TwoStep-AS Cluster Model Nuggets

---

The TwoStep-AS model nugget displays details of the model in the Model tab of the Output Viewer. For more information on using the viewer, see the section titled "Working with Output" in the Modeler User's Guide (ModelerUsersGuide.pdf).

The TwoStep-AS cluster model nuggets contain all of the information captured by the clustering model, as well as information about the training data and the estimation process.

When you run a stream containing a TwoStep-AS cluster model nugget, the node adds a new field containing the cluster membership for that record. The new field name is derived from the model name, prefixed by **\$AS-**. For example, if your model is named TwoStep, the new field will be named **\$AS-TwoStep**.

A powerful technique for gaining insight into the TwoStep-AS model is to use rule induction to discover the characteristics that distinguish the clusters found by the model. See the topic [“C5.0 Node” on page 91](#) for more information.

For general information on using the model browser, see [“Browsing model nuggets” on page 35](#)

## TwoStep-AS Cluster Model Nugget Settings

The Settings tab provides additional options for the TwoStep-AS model nugget.

**Generate SQL for this model** When using data from a database, SQL code can be pushed back to the database for execution, providing superior performance for many operations.

Select one of the following options to specify how SQL generation is performed.

- **Default: Score using Server Scoring Adapter (if installed) otherwise in process** If connected to a database with a scoring adapter installed, generates SQL using the scoring adapter and associated user defined functions (UDF) and scores your model within the database. When no scoring adapter is available, this option fetches your data back from the database and scores it in SPSS Modeler.
- **Score by converting to native SQL** If selected, generates native SQL to score the model within the database.

**Note:** Although this option can provide quicker results, the size and complexity of the native SQL increases as the complexity of the model increases.

- **Score outside of the database** If selected, this option fetches your data back from the database and scores it in SPSS Modeler.

## K-Means-AS node

---

K-Means is one of the most commonly used clustering algorithms. It clusters data points into a predefined number of clusters.<sup>1</sup> The K-Means-AS node in SPSS Modeler is implemented in Spark.

For details about K-Means algorithms, see <https://spark.apache.org/docs/2.2.0/ml-clustering.html>.

Note that the K-Means-AS node performs one-hot encoding automatically for categorical variables.

<sup>1</sup> "Clustering." *Apache Spark*. MLlib: Main Guide. Web. 3 Oct 2017.

## K-Means-AS node Fields

The Fields tab specifies which fields are used in the analysis.

**Use predefined roles.** This option tells the node to use field information from an upstream Type node. It's selected by default.

**Use custom field assignments.** If you want to manually assign input fields, select this option and then select the input field or fields. Using this option is similar to setting the field role in **Input** in a Type node.

## K-Means-AS node Build Options

Use the Build Options tab to specify build options for the K-Means-AS node, including regular options for model building, initialization options for initializing cluster centers, and advanced options for the computing iteration and random seed. For more information, see the [JavaDoc for K-Means on SparkML](#).<sup>1</sup>

## Regular

**Model Name.** The name of the field generated after scoring to a specific cluster. Select **Auto** (default) or select **Custom** and type a name.

**Number of Clusters.** Specify the number of clusters to generate. The default is **5** and the minimum is **2**.

## Initialization

**Initialization Mode.** Specify the method for initializing the cluster centers. **K-Means||** is the default. For details about these two methods, see [Scalable K-Means++](#).<sup>2</sup>

**Initialization Steps.** If the **K-Means||** initialization mode is selected, specify the number of initialization steps. **2** is the default.

## Advanced

**Advanced Settings.** Select this option if you want to set advanced options as follows.

**Max Iteration.** Specify the maximum number of iterations to perform when searching cluster centers. **20** is the default.

**Tolerance.** Specify the convergence tolerance for iterative algorithms. **1.0E-4** is the default.

**Set Random Seed.** Select this option and click **Generate** to generate the seed used by the random number generator.

## Display

**Display Graph.** Select this option if you want a graph to be included in the output.

The following table shows the relationship between the settings in the SPSS Modeler K-Means-AS node and the K-Means Spark parameters.

Table 13. Node properties mapped to Spark parameters		
SPSS Modeler setting	Script name (property name)	K-Means SparkML parameter
Input Fields	features	
Number of Clusters	clustersNum	k
Initialization Mode	initMode	initMode
Initialization Steps	initSteps	initSteps
Max Iteration	maxIter	maxIter
Toleration	toleration	tol
Random Seed	randomSeed	seed

<sup>1</sup> "Class KMeans." *Apache Spark*. JavaDoc. Web. 3 Oct 2017.

<sup>2</sup> Bahmani, Moseley, et al. "Scalable K-Means++." Feb 28, 2012. <http://theory.stanford.edu/%7Eesergei/papers/vldb12-kmpar.pdf>.

## The Cluster Viewer

Cluster models are typically used to find groups (or clusters) of similar records based on the variables examined, where the similarity between members of the same group is high and the similarity between members of different groups is low. The results can be used to identify associations that would otherwise not be apparent. For example, through cluster analysis of customer preferences, income level, and buying habits, it may be possible to identify the types of customers who are more likely to respond to a particular marketing campaign.

There are two approaches to interpreting the results in a cluster display:

- Examine clusters to determine characteristics unique to that cluster. *Does one cluster contain all the high-income borrowers? Does this cluster contain more records than the others?*
- Examine fields across clusters to determine how values are distributed among clusters. *Does one's level of education determine membership in a cluster? Does a high credit score distinguish between membership in one cluster or another?*

Using the main views and the various linked views in the Cluster Viewer, you can gain insight to help you answer these questions.

The following cluster model nuggets can be generated in IBM SPSS Modeler:

- Kohonen net model nugget
- K-Means model nugget
- TwoStep cluster model nugget

To see information about the cluster model nuggets, right-click the model node and choose **Browse** from the context menu (or **Edit** for nodes in a stream). Alternatively, if you are using the Auto Cluster modeling node, double-click on the required cluster nugget within the Auto Cluster model nugget. See the topic [“Auto Cluster node”](#) on page 64 for more information.

## Cluster Viewer - Model Tab

The Model tab for cluster models shows a graphical display of summary statistics and distributions for fields between clusters; this is known as the **Cluster Viewer**.

*Note:* The Model tab is not available for models built in versions of IBM SPSS Modeler prior to 13.

The Cluster Viewer is made up of two panels, the main view on the left and the linked, or auxiliary, view on the right. There are two main views:

- Model Summary (the default). See the topic [“Model Summary View”](#) on page 225 for more information.
- Clusters. See the topic [“Clusters View”](#) on page 226 for more information.

There are four linked/auxiliary views:

- Predictor Importance. See the topic [“Cluster Predictor Importance View”](#) on page 227 for more information.
- Cluster Sizes (the default). See the topic [“Cluster Sizes View”](#) on page 227 for more information.
- Cell Distribution. See the topic [“Cell Distribution View”](#) on page 227 for more information.
- Cluster Comparison. See the topic [“Cluster Comparison View”](#) on page 227 for more information.

### Model Summary View

The Model Summary view shows a snapshot, or summary, of the cluster model, including a Silhouette measure of cluster cohesion and separation that is shaded to indicate poor, fair, or good results. This snapshot enables you to quickly check if the quality is poor, in which case you may decide to return to the modeling node to amend the cluster model settings to produce a better result.

The results of poor, fair, and good are based on the work of Kaufman and Rousseeuw (1990) regarding interpretation of cluster structures. In the Model Summary view, a good result equates to data that reflects Kaufman and Rousseeuw's rating as either reasonable or strong evidence of cluster structure, fair reflects their rating of weak evidence, and poor reflects their rating of no significant evidence.

The silhouette measure averages, over all records,  $(B-A) / \max(A,B)$ , where A is the record's distance to its cluster center and B is the record's distance to the nearest cluster center that it doesn't belong to. A silhouette coefficient of 1 would mean that all cases are located directly on their cluster centers. A value of -1 would mean all cases are located on the cluster centers of some other cluster. A value of 0 means, on average, cases are equidistant between their own cluster center and the nearest other cluster.

The summary includes a table that contains the following information:

- **Algorithm.** The clustering algorithm used, for example, "TwoStep".
- **Input Features.** The number of fields, also known as **inputs** or **predictors**.
- **Clusters.** The number of clusters in the solution.

## Clusters View

The Clusters view contains a cluster-by-features grid that includes cluster names, sizes, and profiles for each cluster.

The columns in the grid contain the following information:

- **Cluster.** The cluster numbers created by the algorithm.
- **Label.** Any labels applied to each cluster (this is blank by default). Double-click in the cell to enter a label that describes the cluster contents; for example, "Luxury car buyers".
- **Description.** Any description of the cluster contents (this is blank by default). Double-click in the cell to enter a description of the cluster; for example, "55+ years of age, professionals, earning over \$100,000".
- **Size.** The size of each cluster as a percentage of the overall cluster sample. Each size cell within the grid displays a vertical bar that shows the size percentage within the cluster, a size percentage in numeric format, and the cluster case counts.
- **Features.** The individual inputs or predictors, sorted by overall importance by default. If any columns have equal sizes they are shown in ascending sort order of the cluster numbers.

Overall feature importance is indicated by the color of the cell background shading; the most important feature is darkest; the least important feature is unshaded. A guide above the table indicates the importance attached to each feature cell color.

When you hover your mouse over a cell, the full name/label of the feature and the importance value for the cell is displayed. Further information may be displayed, depending on the view and feature type. In the Cluster Centers view, this includes the cell statistic and the cell value; for example: "Mean: 4.32". For categorical features the cell shows the name of the most frequent (modal) category and its percentage.

Within the Clusters view, you can select various ways to display the cluster information:

- Transpose clusters and features. See the topic [“Transpose Clusters and Features”](#) on page 226 for more information.
- Sort features. See the topic [“Sort Features”](#) on page 226 for more information.
- Sort clusters. See the topic [“Sort Clusters”](#) on page 227 for more information.
- Select cell contents. See the topic [“Cell Contents”](#) on page 227 for more information.

## Transpose Clusters and Features

By default, clusters are displayed as columns and features are displayed as rows. To reverse this display, click the **Transpose Clusters and Features** button to the left of the **Sort Features By** buttons. For example you may want to do this when you have many clusters displayed, to reduce the amount of horizontal scrolling required to see the data.

## Sort Features

The **Sort Features By** buttons enable you to select how feature cells are displayed:

- **Overall Importance.** This is the default sort order. Features are sorted in descending order of overall importance, and sort order is the same across clusters. If any features have tied importance values, the tied features are listed in ascending sort order of the feature names.
- **Within-Cluster Importance.** Features are sorted with respect to their importance for each cluster. If any features have tied importance values, the tied features are listed in ascending sort order of the feature names. When this option is chosen the sort order usually varies across clusters.
- **Name.** Features are sorted by name in alphabetical order.
- **Data order.** Features are sorted by their order in the dataset.

## Sort Clusters

By default clusters are sorted in descending order of size. The **Sort Clusters By** buttons enable you to sort them by name in alphabetical order, or, if you have created unique labels, in alphanumeric label order instead.

Features that have the same label are sorted by cluster name. If clusters are sorted by label and you edit the label of a cluster, the sort order is automatically updated.

## Cell Contents

The **Cells** buttons enable you to change the display of the cell contents for features and evaluation fields.

- **Cluster Centers.** By default, cells display feature names/labels and the central tendency for each cluster/feature combination. The mean is shown for continuous fields and the mode (most frequently occurring category) with category percentage for categorical fields.
- **Absolute Distributions.** Shows feature names/labels and absolute distributions of the features within each cluster. For categorical features, the display shows bar charts overlaid with categories ordered in ascending order of the data values. For continuous features, the display shows a smooth density plot which use the same endpoints and intervals for each cluster.

The solid red colored display shows the cluster distribution, whilst the paler display represents the overall data.

- **Relative Distributions.** Shows feature names/labels and relative distributions in the cells. In general the displays are similar to those shown for absolute distributions, except that relative distributions are displayed instead.

The solid red colored display shows the cluster distribution, while the paler display represents the overall data.

- **Basic View.** Where there are a lot of clusters, it can be difficult to see all the detail without scrolling. To reduce the amount of scrolling, select this view to change the display to a more compact version of the table.

## Cluster Predictor Importance View

The Predictor Importance view shows the relative importance of each field in estimating the model.

## Cluster Sizes View

The Cluster Sizes view shows a pie chart that contains each cluster. The percentage size of each cluster is shown on each slice; hover the mouse over each slice to display the count in that slice.

Below the chart, a table lists the following size information:

- The size of the smallest cluster (both a count and percentage of the whole).
- The size of the largest cluster (both a count and percentage of the whole).
- The ratio of size of the largest cluster to the smallest cluster.

## Cell Distribution View

The Cell Distribution view shows an expanded, more detailed, plot of the distribution of the data for any feature cell you select in the table in the Clusters main panel.

## Cluster Comparison View

The Cluster Comparison view consists of a grid-style layout, with features in the rows and selected clusters in the columns. This view helps you to better understand the factors that make up the clusters; it also enables you to see differences between clusters not only as compared with the overall data, but with each other.

To select clusters for display, click on the top of the cluster column in the Clusters main panel. Use either Ctrl-click or Shift-click to select or deselect more than one cluster for comparison.

*Note:* You can select up to five clusters for display.

Clusters are shown in the order in which they were selected, while the order of fields is determined by the **Sort Features By** option. When you select **Within-Cluster Importance**, fields are always sorted by overall importance .

The background plots show the overall distributions of each features:

- Categorical features are shown as dot plots, where the size of the dot indicates the most frequent/modal category for each cluster (by feature).
- Continuous features are displayed as boxplots, which show overall medians and the interquartile ranges.

Overlaid on these background views are boxplots for selected clusters:

- For continuous features, square point markers and horizontal lines indicate the median and interquartile range for each cluster.
- Each cluster is represented by a different color, shown at the top of the view.

## Navigating the Cluster Viewer


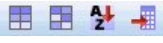
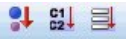

The Cluster Viewer is an interactive display. You can:

- Select a field or cluster to view more details.
- Compare clusters to select items of interest.
- Alter the display.
- Transpose axes.
- Generate Derive, Filter, and Select nodes using the Generate menu.

Using the Toolbars

You control the information shown in both the left and right panels by using the toolbar options. You can change the orientation of the display (top-down, left-to-right, or right-to-left) using the toolbar controls. In addition, you can also reset the viewer to the default settings, and open a dialog box to specify the contents of the Clusters view in the main panel.

The **Sort Features By**, **Sort Clusters By**, **Cells**, and **Display** options are only available when you select the **Clusters** view in the main panel. See the topic [“Clusters View” on page 226](#) for more information.

Table 14. Toolbar icons	
Icon	Topic
	See <a href="#">Transpose Clusters and Features</a>
	See <a href="#">Sort Features By</a>
	See <a href="#">Sort Clusters By</a>
	See <a href="#">Cells</a>

### Generating Nodes from Cluster Models

The Generate menu enables you to create new nodes based on the cluster model. This option is available from the Model tab of the generated model and enables you to generate nodes based on either the current display or selection (that is, all visible clusters or all selected ones). For example, you can select a single feature and then generate a Filter node to discard all other (nonvisible) features. The generated nodes are placed unconnected on the canvas. In addition, you can generate a copy of the model nugget to the models palette. Remember to connect the nodes and make any desired edits before execution.



- **Generate Modeling Node.** Creates a modeling node on the stream canvas. This would be useful, for example, if you have a stream in which you want to use these model settings but you no longer have the modeling node used to generate them.
- **Model to Palette.** Creates a the nugget on the Models palette. This is useful in situations where a colleague may have sent you a stream containing the model and not the model itself.
- **Filter Node.** Creates a new Filter node to filter fields that are not used by the cluster model, and/or not visible in the current Cluster Viewer display. If there is a Type node upstream from this Cluster node, any fields with the role *Target* are discarded by the generated Filter node.
- **Filter Node (from selection).** Creates a new Filter node to filter fields based on selections in the Cluster Viewer. Select multiple fields using the Ctrl-click method. Fields selected in the Cluster Viewer are discarded downstream, but you can change this behavior by editing the Filter node before execution.
- **Select Node.** Creates a new Select node to select records based on their membership in any of the clusters visible in the current Cluster Viewer display. A select condition is automatically generated.
- **Select Node (from selection).** Creates a new Select node to select records based on membership in clusters selected in the Cluster Viewer. Select multiple clusters using the Ctrl-click method.
- **Derive Node.** Creates a new Derive node, which derives a flag field that assigns records a value of *True* or *False* based on membership in all clusters visible in the Cluster Viewer. A derive condition is automatically generated.
- **Derive Node (from selection).** Creates a new Derive node, which derives a flag field based on membership in clusters selected in the Cluster Viewer. Select multiple clusters using the Ctrl-click method.

In addition to generating nodes, you can also create graphs from the Generate menu. See the topic [“Generating Graphs from Cluster Models”](#) on page 229 for more information.

#### Control Cluster View Display

To control what is shown in the Clusters view on the main panel, click the **Display** button; the Display dialog opens.

**Features.** Selected by default. To hide all input features, deselect the check box.

**Evaluation Fields.** Choose the evaluation fields (fields not used to create the cluster model, but sent to the model viewer to evaluate the clusters) to display; none are shown by default. *Note* The evaluation field must be a string with more than one value. This check box is unavailable if no evaluation fields are available.

**Cluster Descriptions.** Selected by default. To hide all cluster description cells, deselect the check box.

**Cluster Sizes.** Selected by default. To hide all cluster size cells, deselect the check box.

**Maximum Number of Categories.** Specify the maximum number of categories to display in charts of categorical features; the default is 20.

## Generating Graphs from Cluster Models

Cluster models provide a lot of information; however, it may not always be in an easily accessible format for business users. To provide the data in a way that can be easily incorporated into business reports, presentations, and so on, you can produce graphs of selected data. For example, from the Cluster Viewer you can generate a graph for a selected cluster, thereby only creating a graph for the cases in that cluster.

*Note:* You can only generate a graph from the Cluster Viewer when the model nugget is attached to other nodes in a stream.

Generate a graph

1. Open the model nugget containing the Cluster Viewer.
2. On the Model tab select *Clusters* from the **View** drop-down list.
3. In the main view, select the cluster, or clusters, for which you want to produce a graph.
4. From the Generate menu, select **Graph (from selection)**; the Graphboard Basic tab is displayed.

*Note:* Only the Basic and Detailed tabs are available when you display the Graphboard in this way.

5. Using either the Basic or Detailed tab settings, specify the details to be displayed on the graph.
6. Click OK to generate the graph.

The graph heading identifies the model type and cluster, or clusters, that were chosen for inclusion.

## Chapter 12. Association Rules

**Association rules** associate a particular conclusion (the purchase of a particular product, for example) with a set of conditions (the purchase of several other products, for example). For example, the rule

```
beer <= cannedveg & frozenmeal (173, 17.0%, 0.84)
```

states that *beer* often occurs when *cannedveg* and *frozenmeal* occur together. The rule is 84% reliable and applies to 17% of the data, or 173 records. Association rule algorithms automatically find the associations that you could find manually using visualization techniques, such as the Web node.

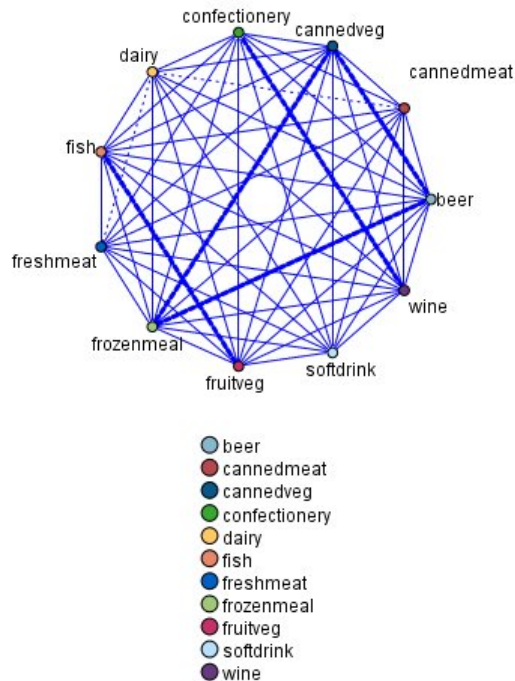


Figure 45. Web node showing associations between market basket items

The advantage of association rule algorithms over the more standard decision tree algorithms (C5.0 and C&R Trees) is that associations can exist between *any* of the attributes. A decision tree algorithm will build rules with only a single conclusion, whereas association algorithms attempt to find many rules, each of which may have a different conclusion.

The disadvantage of association algorithms is that they are trying to find patterns within a potentially very large search space and, hence, can require much more time to run than a decision tree algorithm. The algorithms use a **generate and test** method for finding rules--simple rules are generated initially, and these are validated against the dataset. The good rules are stored and all rules, subject to various constraints, are then specialized. **Specialization** is the process of adding conditions to a rule. These new rules are then validated against the data, and the process iteratively stores the best or most interesting rules found. The user usually supplies some limit to the possible number of antecedents to allow in a rule, and various techniques based on information theory or efficient indexing schemes are used to reduce the potentially large search space.

At the end of the processing, a table of the best rules is presented. Unlike a decision tree, this set of association rules cannot be used directly to make predictions in the way that a standard model (such as a decision tree or a neural network) can. This is due to the many different possible conclusions for the rules. Another level of transformation is required to transform the association rules into a classification rule set. Hence, the association rules produced by association algorithms are known as **unrefined models**. Although the user can browse these unrefined models, they cannot be used explicitly as classification

models unless the user tells the system to generate a classification model from the unrefined model. This is done from the browser through a Generate menu option.

Two association rule algorithms are supported:



The Apriori node extracts a set of rules from the data, pulling out the rules with the highest information content. Apriori offers five different methods of selecting rules and uses a sophisticated indexing scheme to process large data sets efficiently. For large problems, Apriori is generally faster to train; it has no arbitrary limit on the number of rules that can be retained, and it can handle rules with up to 32 preconditions. Apriori requires that input and output fields all be categorical but delivers better performance because it is optimized for this type of data.



The Sequence node discovers association rules in sequential or time-oriented data. A sequence is a list of item sets that tends to occur in a predictable order. For example, a customer who purchases a razor and aftershave lotion may purchase shaving cream the next time he shops. The Sequence node is based on the CARMA association rules algorithm, which uses an efficient two-pass method for finding sequences.

## Tabular versus Transactional Data

---

Data used by association rule models may be in transactional or tabular format, as described below. These are general descriptions; specific requirements may vary as discussed in the documentation for each model type. Note that when scoring models, the data to be scored must mirror the format of the data used to build the model. Models built using tabular data can be used to score only tabular data; models built using transactional data can score only transactional data.

### Transactional Format

Transactional data have a separate record for each transaction or item. If a customer makes multiple purchases, for example, each would be a separate record, with associated items linked by a customer ID. This is also sometimes known as **till-roll** format.

Customer	Purchase
1	jam
2	milk
3	jam
3	bread
4	jam
4	bread
4	milk

The Apriori, CARMA, and Sequence nodes can all use transactional data.

### Tabular Data

Tabular data (also known as **basket** or **truth-table** data) have items represented by separate flags, where each flag field represents the presence or absence of a specific item. Each record represents a complete set of associated items. Flag fields can be categorical or numeric, although certain models may have more specific requirements.

Customer	Jam	Bread	Milk
1	T	F	F

Customer	Jam	Bread	Milk
2	F	F	T
3	T	T	F
4	T	T	T

The Apriori, CARMA, GSAR, and Sequence nodes can all use tabular data.

## Apriori node

---

The Apriori node also discovers association rules in the data. Apriori offers five different methods of selecting rules and uses a sophisticated indexing scheme to efficiently process large data sets.

**Requirements.** To create an Apriori rule set, you need one or more *Input* fields and one or more *Target* fields. Input and output fields (those with the role *Input*, *Target*, or *Both*) must be symbolic. Fields with the role *None* are ignored. Fields types must be fully instantiated before executing the node. Data can be in tabular or transactional format. See the topic [“Tabular versus Transactional Data”](#) on page 232 for more information.

**Strengths.** For large problems, Apriori is generally faster to train. It also has no arbitrary limit on the number of rules that can be retained and can handle rules with up to 32 preconditions. Apriori offers five different training methods, allowing more flexibility in matching the data mining method to the problem at hand.

### Apriori Node Model Options

**Model name.** You can generate the model name automatically based on the target or ID field (or model type in cases where no such field is specified) or specify a custom name.

**Minimum antecedent support.** You can specify a support criterion for keeping rules in the rule set.

**Support** refers to the percentage of records in the training data for which the antecedents (the “if” part of the rule) are true. (Note that this definition of support differs from that used in the CARMA and Sequence nodes. See the topic [“Sequence Node Model Options”](#) on page 247 for more information. ) If you are getting rules that apply to very small subsets of the data, try increasing this setting.

**Note:** The definition of support for Apriori is based on the number of records with the antecedents. This is in contrast to the CARMA and Sequence algorithms for which the definition of support is based on the number of records with all the items in a rule (that is, both the antecedents and consequent). The results for association models show both the (antecedent) support and rule support measures.

**Minimum rule confidence.** You can also specify a confidence criterion. **Confidence** is based on the records for which the rule's antecedents are true and is the percentage of those records for which the consequent(s) are also true. In other words, it's the percentage of predictions based on the rule that are correct. Rules with lower confidence than the specified criterion are discarded. If you are getting too many rules, try increasing this setting. If you are getting too few rules (or no rules at all), try decreasing this setting.

**Note:** If necessary, you can highlight the value and type in your own value. Be aware that if you reduce the confidence value below 1.0, in addition to the process requiring a lot of free memory, you might find that the rules take an extremely long time to build.

**Maximum number of antecedents.** You can specify the maximum number of preconditions for any rule. This is a way to limit the complexity of the rules. If the rules are too complex or too specific, try decreasing this setting. This setting also has a large influence on training time. If your rule set is taking too long to train, try reducing this setting.

**Only true values for flags.** If this option is selected for data in tabular (truth table) format, then only true values will be included in the resulting rules. This can help make rules easier to understand. The option does not apply to data in transactional format. See the topic [“Tabular versus Transactional Data”](#) on page 232 for more information.

**Note:** The CARMA model building node ignores empty records when building a model if the field type is a flag, whereas the Apriori model building node includes empty records. Empty records are records where all the fields used in the model build have a false value.

**Optimize.** Select options designed to increase performance during model building based on your specific needs.

- Select **Speed** to instruct the algorithm to never use disk spilling in order to improve performance.
- Select **Memory** to instruct the algorithm to use disk spilling when appropriate at some sacrifice to speed. This option is selected by default.

**Note:** When running in distributed mode, this setting can be overridden by administrator options specified in the *options.cfg* file. For more information, see the *IBM SPSS Modeler Server Administrator's Guide*.

## Apriori Node Expert Options

For those with detailed knowledge of Apriori's operation, the following expert options allow you to fine-tune the induction process. To access expert options, set the Mode to **Expert** on the Expert tab.

**Evaluation measure.** Apriori supports five methods of evaluating potential rules.

- **Rule Confidence.** The default method uses the confidence (or accuracy) of the rule to evaluate rules. For this measure, the **Evaluation measure lower bound** is disabled, since it is redundant with the **Minimum rule confidence** option on the Model tab. See the topic [“Apriori Node Model Options”](#) on page 233 for more information.
- **Confidence Difference.** (Also called **absolute confidence difference to prior**.) This evaluation measure is the absolute difference between the rule's confidence and its prior confidence. This option prevents bias where the outcomes are not evenly distributed. This helps prevent "obvious" rules from being kept. For example, it may be the case that 80% of customers buy your most popular product. A rule that predicts buying that popular product with 85% accuracy doesn't add much to your knowledge, even though 85% accuracy may seem quite good on an absolute scale. Set the evaluation measure lower bound to the minimum difference in confidence for which you want rules to be kept.
- **Confidence Ratio.** (Also called **difference of confidence quotient to 1**.) This evaluation measure is the ratio of rule confidence to prior confidence (or, if the ratio is greater than one, its reciprocal) subtracted from 1. Like Confidence Difference, this method takes uneven distributions into account. It is especially good at finding rules that predict rare events. For example, suppose that there is a rare medical condition that occurs in only 1% of patients. A rule that is able to predict this condition 10% of the time is a great improvement over random guessing, even though on an absolute scale, 10% accuracy might not seem very impressive. Set the evaluation measure lower bound to the difference for which you want rules to be kept.
- **Information Difference.** (Also called **information difference to prior**.) This measure is based on the **information gain** measure. If the probability of a particular consequent is considered as a logical value (a **bit**), then the information gain is the proportion of that bit that can be determined, based on the antecedents. The information difference is the difference between the information gain, given the antecedents, and the information gain, given only the prior confidence of the consequent. An important feature of this method is that it takes support into account so that rules that cover more records are preferred for a given level of confidence. Set the evaluation measure lower bound to the information difference for which you want rules to be kept.

*Note:* Because the scale for this measure is somewhat less intuitive than the other scales, you may need to experiment with different lower bounds to get a satisfactory rule set.

- **Normalized Chi-square.** (Also called **normalized chi-squared measure**.) This measure is a statistical index of association between antecedents and consequents. The measure is normalized to take values between 0 and 1. This measure is even more strongly dependent on support than the information difference measure. Set the evaluation measure lower bound to the information difference for which you want rules to be kept.

*Note:* As with the information difference measure, the scale for this measure is somewhat less intuitive than the other scales, so you may need to experiment with different lower bounds to get a satisfactory rule set.

**Allow rules without antecedents.** Select to allow rules that include only the consequent (item or item set). This is useful when you are interested in determining common items or item sets. For example, *cannedveg* is a single-item rule without an antecedent that indicates purchasing *cannedveg* is a common occurrence in the data. In some cases, you may want to include such rules if you are simply interested in the most confident predictions. This option is off by default. By convention, antecedent support for rules without antecedents is expressed as 100%, and rule support will be the same as confidence.

## CARMA Node

The CARMA node uses an association rules discovery algorithm to discover association rules in the data. Association rules are statements in the form

```
if antecedent(s) then consequent(s)
```

For example, if a Web customer purchases a wireless card and a high-end wireless router, the customer is also likely to purchase a wireless music server if offered. The CARMA model extracts a set of rules from the data without requiring you to specify input or target fields. This means that the rules generated can be used for a wider variety of applications. For example, you can use rules generated by this node to find a list of products or services (antecedents) whose consequent is the item that you want to promote this holiday season. Using IBM SPSS Modeler, you can determine which clients have purchased the antecedent products and construct a marketing campaign designed to promote the consequent product.

**Requirements.** In contrast to Apriori, the CARMA node does not require *Input* or *Target* fields. This is integral to the way the algorithm works and is equivalent to building an Apriori model with all fields set to *Both*. You can constrain which items are listed only as antecedents or consequents by filtering the model after it is built. For example, you can use the model browser to find a list of products or services (antecedents) whose consequent is the item that you want to promote this holiday season.

To create a CARMA rule set, you need to specify an ID field and one or more content fields. The ID field can have any role or measurement level. Fields with the role *None* are ignored. Field types must be fully instantiated before executing the node. Like Apriori, data may be in tabular or transactional format. See the topic [“Tabular versus Transactional Data” on page 232](#) for more information.

**Strengths.** The CARMA node is based on the CARMA association rules algorithm. In contrast to Apriori, the CARMA node offers build settings for rule support (support for both antecedent and consequent) rather than antecedent support. CARMA also allows rules with multiple consequents. Like Apriori, models generated by a CARMA node can be inserted into a data stream to create predictions. See the topic [“Model Nuggets” on page 31](#) for more information.

## CARMA Node Fields Options

Before executing a CARMA node, you must specify input fields on the Fields tab of the CARMA node. While most modeling nodes share identical Fields tab options, the CARMA node contains several unique options. All options are discussed below.

**Use Type node settings.** This option tells the node to use field information from an upstream type node. This is the default.

**Use custom settings.** This option tells the node to use field information specified here instead of that given in any upstream Type node(s). After selecting this option, specify fields below according to whether you are reading data in transactional or tabular format.

**Use transactional format.** This option changes the field controls in the rest of this dialog box depending on whether your data are in transactional or tabular format. If you use multiple fields with transactional data, the items specified in these fields for a particular record are assumed to represent items found in a single transaction with a single timestamp. See the topic [“Tabular versus Transactional Data” on page 232](#) for more information.

Tabular data

If **Use transactional format** is not selected, the following fields are displayed.

- **Inputs.** Select the input field or fields. This is similar to setting the field role to *Input* in a Type node.
- **Partition.** This field allows you to specify a field used to partition the data into separate samples for the training, testing, and validation stages of model building. By using one sample to generate the model and a different sample to test it, you can get a good indication of how well the model will generalize to larger datasets that are similar to the current data. If multiple partition fields have been defined by using Type or Partition nodes, a single partition field must be selected on the Fields tab in each modeling node that uses partitioning. (If only one partition is present, it is automatically used whenever partitioning is enabled.) Also note that to apply the selected partition in your analysis, partitioning must also be enabled in the Model Options tab for the node. (Deselecting this option makes it possible to disable partitioning without changing field settings.)

Transactional data

If you select **Use transactional format**, the following fields are displayed.

- **ID.** For transactional data, select an ID field from the list. Numeric or symbolic fields can be used as the ID field. Each unique value of this field should indicate a specific unit of analysis. For example, in a market basket application, each ID might represent a single customer. For a Web log analysis application, each ID might represent a computer (by IP address) or a user (by login data).
- **IDs are contiguous.** (Apriori and CARMA nodes only) If your data are presorted so that all records with the same ID are grouped together in the data stream, select this option to speed up processing. If your data are not presorted (or you are not sure), leave this option unselected and the node will sort the data automatically.  
  
**Note:** If your data are not sorted and you select this option, you may get invalid results in your model.
- **Content.** Specify the content field(s) for the model. These fields contain the items of interest in association modeling. You can specify multiple flag fields (if data are in tabular format) or a single nominal field (if data are in transactional format).

## CARMA Node Model Options

**Model name.** You can generate the model name automatically based on the target or ID field (or model type in cases where no such field is specified) or specify a custom name.

**Minimum rule support (%).** You can specify a support criterion. **Rule support** refers to the proportion of IDs in the training data that contain the entire rule. (Note that this definition of support differs from antecedent support used in the Apriori nodes.) If you want to focus on more common rules, increase this setting.

**Minimum rule confidence (%).** You can specify a confidence criterion for keeping rules in the rule set. **Confidence** refers to the percentage of IDs where a correct prediction is made (out of all IDs for which the rule makes a prediction). It is calculated as the number of IDs for which the entire rule is found divided by the number of IDs for which the antecedents are found, based on the training data. Rules with lower confidence than the specified criterion are discarded. If you are getting uninteresting or too many rules, try increasing this setting. If you are getting too few rules, try decreasing this setting.

**Note:** If necessary, you can highlight the value and type in your own value. Be aware that if you reduce the confidence value below 1.0, in addition to the process requiring a lot of free memory, you might find that the rules take an extremely long time to build.

**Maximum rule size.** You can set the maximum number of distinct item sets (as opposed to items) in a rule. If the rules of interest are relatively short, you can decrease this setting to speed up building the rule set.

**Note:** The CARMA model building node ignores empty records when building a model if the field type is a flag, whereas the Apriori model building node includes empty records. Empty records are records where all the fields used in the model build have a false value.



## CARMA Node Expert Options

For those with detailed knowledge of the CARMA node's operation, the following expert options allow you to fine-tune the model-building process. To access expert options, set the mode to **Expert** on the Expert tab.

**Exclude rules with multiple consequents.** Select to exclude “two-headed” consequents—that is, consequents that contain two items. For example, the rule `bread & cheese & fish -> wine&fruit` contains a two-headed consequent, `wine&fruit`. By default, such rules are included.

**Set pruning value.** To conserve memory, the CARMA algorithm used periodically removes (**prunes**) infrequent item sets from its list of potential item sets during processing. Select this option to adjust the frequency of pruning, and the number you specify determines the frequency of pruning. Enter a smaller value to decrease the memory requirements of the algorithm (but potentially increase the training time required), or enter a larger value to speed up training (but potentially increase memory requirements). The default value is 500.

**Vary support.** Select to increase efficiency by excluding infrequent item sets that seem to be frequent when they are included unevenly. This is achieved by starting with a higher support level and tapering it down to the level specified on the Model tab. Enter a value for **Estimated number of transactions** to specify how quickly the support level should be tapered.

**Allow rules without antecedents.** Select to allow rules that include only the consequent (item or item set). This is useful when you are interested in determining common items or item sets. For example, `cannedveg` is a single-item rule without an antecedent that indicates purchasing `cannedveg` is a common occurrence in the data. In some cases, you may want to include such rules if you are simply interested in the most confident predictions. This option is unselected by default.

## Association Rule Model Nuggets

---

Association rule model nuggets represent the rules discovered by one of the following association rule modeling nodes:

- Apriori
- CARMA

The model nuggets contain information about the rules extracted from the data during model building.

**Note:** Association rule nugget scoring may be incorrect if you do not sort transactional data by ID.

### Viewing Results

You can browse the rules generated by association models (Apriori and CARMA) and Sequence models using the Model tab on the dialog box. Browsing a model nugget shows you the information about the rules and provides options for filtering and sorting results before generating new nodes or scoring the model.

### Scoring the Model

Refined model nuggets (Apriori, CARMA, and Sequence) may be added to a stream and used for scoring. See the topic [“Using Model Nuggets in Streams”](#) on page 41 for more information. Model nuggets used for scoring include an extra Settings tab on their respective dialog boxes. See the topic [“Association Rule Model Nugget Settings”](#) on page 240 for more information.

An unrefined model nugget cannot be used for scoring in its raw format. Instead, you can generate a rule set and use the rule set for scoring. See the topic [“Generating a Rule Set from an Association Model Nugget”](#) on page 242 for more information.

## Association Rule model nugget details

On the Model tab of an Association Rule model nugget, you can see a table containing the rules extracted by the algorithm. Each row in the table represents a rule. The first column represents the consequents (the “then” part of the rule), while the next column represents the antecedents (the “if” part of the rule). Subsequent columns contain rule information, such as confidence, support, and lift.

Association rules are often shown in the format in the following table.

Table 15. Example of an association rule	
Consequent	Antecedent
Drug = drugY	Sex = F BP = HIGH

The example rule is interpreted as *if Sex = "F" and BP = "HIGH," then Drug is likely to be drugY*; or to phrase it another way, *for records where Sex = "F" and BP = "HIGH," Drug is likely to be drugY*. Using the dialog box toolbar, you can choose to display additional information, such as confidence, support, and instances.

**Sort menu.** The Sort menu button on the toolbar controls the sorting of rules. Direction of sorting (ascending or descending) can be changed using the sort direction button (up or down arrow).

You can sort rules by:

- Support
- Confidence
- Rule Support
- Consequent
- Evaluation
- Lift
- Deployability

**Show/Hide menu.** The Show/Hide menu (criteria toolbar button) controls options for the display of rules.

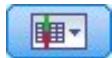


Figure 46. Show/Hide button

The following display options are available:

- **Rule ID** displays the rule ID assigned during model building. A rule ID enables you to identify which rules are being applied for a given prediction. Rule IDs also allow you to merge additional rule information, such as deployability, product information, or antecedents, at a later time.
- **Instances** displays information about the number of unique IDs to which the rule applies--that is, for which the antecedents are true. For example, given the rule *bread -> cheese*, the number of records in the training data that include the antecedent *bread* are referred to as **instances**.
- **Support** displays antecedent support—that is, the proportion of IDs for which the antecedents are true, based on the training data. For example, if 50% of the training data includes the purchase of bread, then the rule *bread -> cheese* will have an antecedent support of 50%. *Note:* Support as defined here is the same as the instances but is represented as a percentage.
- **Confidence** displays the ratio of rule support to antecedent support. This indicates the proportion of IDs with the specified antecedent(s) for which the consequent(s) is/are also true. For example, if 50% of the training data contains bread (indicating antecedent support) but only 20% contains both bread and cheese (indicating rule support), then confidence for the rule *bread -> cheese* would be  $\text{Rule Support} / \text{Antecedent Support}$  or, in this case, 40%.
- **Rule Support** displays the proportion of IDs for which the entire rule, antecedents, and consequent(s), are true. For example, if 20% of the training data contains both bread and cheese, then rule support for the rule *bread -> cheese* is 20%.
- **Evaluation** is included if you select one of the expert association rule criterion (Confidence Difference, Confidence Ratio, Information Difference, or Normalized Chi-Square). These expert criteria measures are compared to the **Evaluation measure lower bound** number set by the user (and only applies when

an expert criterion rule is selected). The Evaluation statistic has the following meanings for each expert association rule criterion:

- Confidence Difference: Posterior Confidence - Prior Confidence
- Confidence Ratio: (Posterior Confidence - Prior Confidence)/Posterior Confidence
- Information Difference: Information Gain Measure
- Normalized Chi-Square: Normalized Chi-Square Statistic

Each of these statistics are compared to the **Evaluation measure lower bound** number set by the user, and a rule is selected if the statistics exceed this number.

- **Lift** displays the ratio of confidence for the rule to the prior probability of having the consequent. For example, if 10% of the entire population buys bread, then a rule that predicts whether people will buy bread with 20% confidence will have a lift of  $20/10 = 2$ . If another rule tells you that people will buy bread with 11% confidence, then the rule has a lift of close to 1, meaning that having the antecedent(s) does not make a lot of difference in the probability of having the consequent. In general, rules with lift different from 1 will be more interesting than rules with lift close to 1.
- **Deployability** is a measure of what percentage of the training data satisfies the conditions of the antecedent but does not satisfy the consequent. In product purchase terms, it basically means what percentage of the total customer base owns (or has purchased) the antecedent(s) but has not yet purchased the consequent. The deployability statistic is defined as  $((\text{Antecedent Support in \# of Records} - \text{Rule Support in \# of Records}) / \text{Number of Records}) * 100$ , where *Antecedent Support* means the number of records for which the antecedents are true and *Rule Support* means the number of records for which both antecedents and the consequent are true.

**Filter button.** The Filter button (funnel icon) on the menu expands the bottom of the dialog box to show a panel where active rule filters are displayed. Filters are used to narrow the number of rules displayed on the Models tab.



Figure 47. Filter button

To create a filter, click the Filter icon to the right of the expanded panel. This opens a separate dialog box in which you can specify constraints for displaying rules. Note that the Filter button is often used in conjunction with the Generate menu to first filter rules and then generate a model containing that subset of rules. For more information, see [“Specifying Filters for Rules”](#) on page 240 below.

**Find Rule button.** The Find Rule button (binoculars icon) enables you to search the rules shown for a specified rule ID. The adjacent display box indicates the number of rules currently displayed out of the number available. Rule IDs are assigned by the model in the order of discovery at the time and are added to the data during scoring.



Figure 48. Find Rule button

To reorder rule IDs:

1. You can rearrange rule IDs in IBM SPSS Modeler by first sorting the rule display table according to the desired measurement, such as confidence or lift.
2. Then using options from the Generate menu, create a filtered model.
3. In the Filtered Model dialog box, select **Renumber rules consecutively starting with**, and specify a start number.

See [“Generating a Filtered Model”](#) on page 242 for more information.

## Specifying Filters for Rules

By default, rule algorithms, such as Apriori, CARMA, and Sequence, may generate a large and cumbersome number of rules. To enhance clarity when browsing or to streamline rule scoring, you should consider filtering rules so that consequents and antecedents of interest are more prominently displayed. Using the filtering options on the Model tab of a rule browser, you can open a dialog box for specifying filter qualifications.

**Consequents.** Select **Enable Filter** to activate options for filtering rules based on the inclusion or exclusion of specified consequents. Select **Includes any of** to create a filter where rules contain at least one of the specified consequents. Alternatively, select **Excludes** to create a filter excluding specified consequents. You can select consequents using the picker icon to the right of the list box. This opens a dialog box listing all consequents present in the generated rules.

*Note:* Consequents may contain more than one item. Filters will check only that a consequent contains one of the items specified.

**Antecedents.** Select **Enable Filter** to activate options for filtering rules based on the inclusion or exclusion of specified antecedents. You can select items using the picker icon to the right of the list box. This opens a dialog box listing all antecedents present in the generated rules.

- Select **Includes all of** to set the filter as an inclusionary one where all antecedents specified must be included in a rule.
- Select **Includes any of** to create a filter where rules contain at least one of the specified antecedents.
- Select **Excludes** to create a filter excluding rules that contain a specified antecedent.

**Confidence.** Select **Enable Filter** to activate options for filtering rules based on the level of confidence for a rule. You can use the **Min** and **Max** controls to specify a confidence range. When you are browsing generated models, confidence is listed as a percentage. When you are scoring output, confidence is expressed as a number between 0 and 1.

**Antecedent Support.** Select **Enable Filter** to activate options for filtering rules based on the level of antecedent support for a rule. Antecedent support indicates the proportion of training data that contains the same antecedents as the current rule, making it analogous to a popularity index. You can use the **Min** and **Max** controls to specify a range used to filter rules based on support level.

**Lift.** Select **Enable Filter** to activate options for filtering rules based on the lift measurement for a rule.

*Note:* Lift filtering is available only for association models built after release 8.5 or for earlier models that contain a lift measurement. Sequence models do not contain this option.

Click **OK** to apply all filters that have been enabled in this dialog box.

## Generating Graphs for Rules

The Association nodes provide a lot of information; however, it may not always be in an easily accessible format for business users. To provide the data in a way that can be easily incorporated into business reports, presentations, and so on, you can produce graphs of selected data. From the Model tab, you can generate a graph for a selected rule, thereby only creating a graph for the cases in that rule.

1. On the Model tab, select the rule in which you are interested.
2. From the Generate menu, select **Graph (from selection)**. The Graphboard Basic tab is displayed.

*Note:* Only the Basic and Detailed tabs are available when you display the Graphboard in this way.

3. Using either the Basic or Detailed tab settings, specify the details to be displayed on the graph.
4. Click OK to generate the graph.

The graph heading identifies the rule and antecedent details that were chosen for inclusion.

## Association Rule Model Nugget Settings

This Settings tab is used to specify scoring options for association models (Apriori and CARMA). This tab is available only after the model nugget has been added to a stream for purposes of scoring.

**Note:** The dialog box for browsing an unrefined model does not include the Settings tab, since it cannot be scored. To score the "unrefined" model, you must first generate a rule set. See the topic [“Generating a Rule Set from an Association Model Nugget”](#) on page 242 for more information.

**Maximum number of predictions** Specify the maximum number of predictions included for each set of basket items. This option is used in conjunction with Rule Criterion below to produce the “top” predictions, where *top* indicates the highest level of confidence, support, lift, and so on, as specified below.

**Rule Criterion** Select the measure used to determine the strength of rules. Rules are sorted by the strength of criteria selected here in order to return the top predictions for an item set. Available criteria are shown in the following list.

- Confidence
- Support
- Rule support (Support \* Confidence)
- Lift
- Deployability

**Allow repeat predictions** Select to include multiple rules with the same consequent when scoring. For example, selecting this option allows the following rules to be scored:

```
bread & cheese -> wine
cheese & fruit -> wine
```

Turn off this option to exclude repeat predictions when scoring.

**Note:** Rules with multiple consequents (bread & cheese & fruit -> wine & pate) are considered repeat predictions only if all consequents (wine & pate) have been predicted before.

**Ignore unmatched basket items** Select to ignore the presence of additional items in the item set. For example, when this option is selected for a basket that contains [tent & sleeping bag & kettle], the rule tent & sleeping bag -> gas\_stove will apply despite the extra item (kettle) present in the basket.

There may be some circumstances where extra items should be excluded. For example, it is likely that someone who purchases a tent, sleeping bag, and kettle may already have a gas stove, indicated by the presence of the kettle. In other words, a gas stove may not be the best prediction. In such cases, you should deselect **Ignore unmatched basket items** to ensure that rule antecedents exactly match the contents of a basket. By default, unmatched items are ignored.

**Check that predictions are not in basket.** Select to ensure that consequents are not also present in the basket. For example, if the purpose of scoring is to make a home furniture product recommendation, then it is unlikely that a basket that already contains a dining room table will be likely to purchase another one. In such a case, you should select this option. On the other hand, if products are perishable or disposable (such as cheese, baby formula, or tissue), then rules where the consequent is already present in the basket may be of value. In the latter case, the most useful option might be **Do not check basket for predictions** below.

**Check that predictions are in basket** Select this option to ensure that consequents are also present in the basket. This approach is useful when you are attempting to gain insight into existing customers or transactions. For example, you may want to identify rules with the highest lift and then explore which customers fit these rules.

**Do not check basket for predictions** Select to include all rules when scoring, regardless of the presence or absence of consequents in the basket.

**Generate SQL for this model** When using data from a database, SQL code can be pushed back to the database for execution, providing superior performance for many operations.

Select one of the following options to specify how SQL generation is performed.

- **Default: Score using Server Scoring Adapter (if installed) otherwise in process** If connected to a database with a scoring adapter installed, generates SQL using the scoring adapter and associated user defined functions (UDF) and scores your model within the database. When no scoring adapter is available, this option fetches your data back from the database and scores it in SPSS Modeler.
- **Score outside of the Database** If selected, this option fetches your data back from the database and scores it in SPSS Modeler.

## Association Rule Model Nugget Summary

The Summary tab of an association rule model nugget displays the number of rules discovered and the minimum and maximum for support, lift, confidence and deployability of rules in the rule set.

## Generating a Rule Set from an Association Model Nugget

Association model nuggets, such as Apriori and CARMA, can be used to score data directly, or you can first generate a subset of rules, known as a **rule set**. Rule sets are particularly useful when you are working with an unrefined model, which cannot be used directly for scoring. See the topic [“Unrefined Models”](#) on page 44 for more information.

To generate a rule set, choose **Rule set** from the Generate menu in the model nugget browser. You can specify the following options for translating the rules into a rule set:

**Rule set name.** Allows you to specify the name of the new generated Rule Set node.

**Create node on.** Controls the location of the new generated Rule Set node. Select **Canvas**, **GM Palette**, or **Both**.

**Target field.** Determines which output field will be used for the generated Rule Set node. Select a single output field from the list.

**Minimum support.** Specify the minimum support for rules to be preserved in the generated rule set. Rules with support less than the specified value will not be included in the new rule set.

**Minimum confidence.** Specify the minimum confidence for rules to be preserved in the generated rule set. Rules with confidence less than the specified value will not be included in the new rule set.

**Default value.** Allows you to specify a default value for the target field that is assigned to scored records for which no rule fires.

## Generating a Filtered Model

To generate a filtered model from an association model nugget, such as an Apriori, CARMA, or Sequence Rule Set node, choose **Filtered Model** from the Generate menu in the model nugget browser. This creates a subset model that includes only those rules currently displayed in the browser. *Note:* You cannot generate filtered models for unrefined models.

You can specify the following options for filtering rules:

**Name for New Model.** Allows you to specify the name of the new Filtered Model node.

**Create node on.** Controls the location of the new Filtered Model node. Select **Canvas**, **GM Palette**, or **Both**.

**Rule numbering.** Specify how rule IDs will be numbered in the subset of rules included in the filtered model.

- **Retain original rule ID numbers.** Select to maintain the original numbering of rules. By default, rules are given an ID that corresponds with their order of discovery by the algorithm. That order may vary depending on the algorithm employed.
- **Renumber rules consecutively starting with.** Select to assign new rule IDs for the filtered rules. New IDs are assigned based on the sort order displayed in the rule browser table on the Model tab, beginning with the number you specify here. You can specify the start number for IDs using the arrows to the right.

## Scoring Association Rules

Scores produced by running new data through an association rule model nugget are returned in separate fields. Three new fields are added for each prediction, with *P* representing the prediction, *C* representing confidence, and *I* representing the rule ID. The organization of these output fields depends on whether the input data are in transactional or tabular format. See [“Tabular versus Transactional Data”](#) on page 232 for an overview of these formats.

For example, suppose you are scoring basket data using a model that generates predictions based on the following three rules:

```
Rule_15 bread&wine -> meat (confidence 54%)
Rule_22 cheese -> fruit (confidence 43%)
Rule_5 bread&cheese -> frozveg (confidence 24%)
```

**Tabular data.** For tabular data, the three predictions (3 is the default) are returned in a single record.

Table 16. Scores in tabular format												
ID	Bread	Wine	Cheese	P1	C1	I1	P2	C2	I2	P3	C3	I3
Fred	1	1	1	meat	0.54	15	fruit	0.43	22	frozveg	.24	5

**Transactional data.** For transactional data, a separate record is generated for each prediction. Predictions are still added in separate columns, but scores are returned as they are calculated. This results in records with incomplete predictions, as shown in the sample output below. The second and third predictions (P2 and P3) are blank in the first record, along with the associated confidences and rule IDs. As scores are returned, however, the final record contains all three predictions.

Table 17. Scores in transactional format										
ID	Item	P1	C1	I1	P2	C2	I2	P3	C3	I3
Fred	bread	meat	0.54	14	\$null\$	\$null\$	\$null\$	\$null\$	\$null\$	\$null\$
Fred	cheese	meat	0.54	14	fruit	0.43	22	\$null\$	\$null\$	\$null\$
Fred	wine	meat	0.54	14	fruit	0.43	22	frozveg	0.24	5

To include only complete predictions for reporting or deployment purposes, use a Select node to select complete records.

*Note:* The field names used in these examples are abbreviated for clarity. During actual use, results fields for association models are named as shown in the following table.

Table 18. Names of results fields for association models	
New field	Example field name
Prediction	\$A-TRANSACTION_NUMBER-1
Confidence (or other criterion)	\$AC-TRANSACTION_NUMBER-1
Rule ID	\$A-Rule_ID-1

### Rules with Multiple Consequents

The CARMA algorithm allows rules with multiple consequents—for example:

```
bread -> wine&cheese
```

When you are scoring such “two-headed” rules, predictions are returned in the format displayed in the following table.

Table 19. Scoring results including a prediction with multiple consequents												
ID	Bread	Wine	Cheese	P1	C1	I1	P2	C2	I2	P3	C3	I3
Fred	1	1	1	meat&veg	0.54	16	fruit	0.43	22	frozveg	.24	5

In some cases, you may need to split such scores before deployment. To split a prediction with multiple consequents, you will need to parse the field using the CLEM string functions.

## Deploying Association Models

When scoring association models, predictions and confidences are output in separate columns (where *P* represents the prediction, *C* represents confidence, and *I* represents the rule ID). This is the case whether the input data are tabular or transactional. See the topic [“Scoring Association Rules”](#) on page 243 for more information.

When preparing scores for deployment, you might find that your application requires you to transpose your output data to a format with predictions in rows rather than columns (one prediction per row, sometimes known as “till-roll” format).

### Transposing Tabular Scores

You can transpose tabular scores from columns to rows using a combination of steps in IBM SPSS Modeler, as described in the steps that follow.

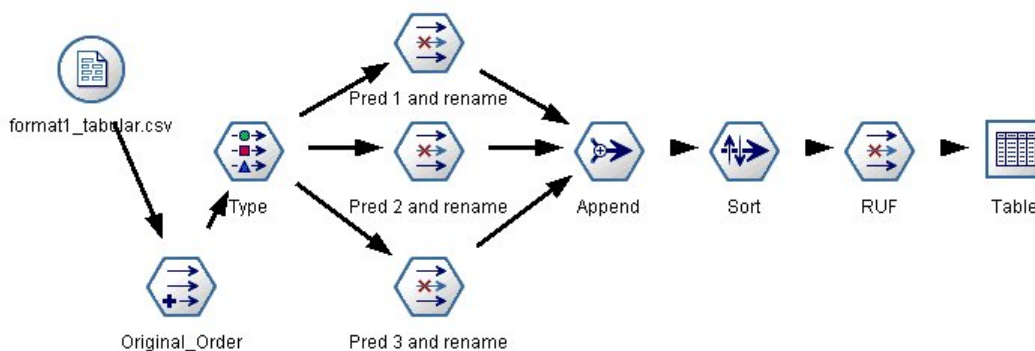


Figure 49. Example stream used to transpose tabular data into till-roll format

1. Use the @INDEX function in a Derive node to ascertain the current order of predictions and save this indicator in a new field, such as *Original\_order*.
2. Add a Type node to ensure that all fields are instantiated.
3. Use a Filter node to rename the default prediction, confidence, and ID fields (*P1*, *C1*, *I1*) to common fields, such as *Pred*, *Crit*, and *Rule\_ID*, which will be used to append records later on. You will need one Filter node for each prediction generated.



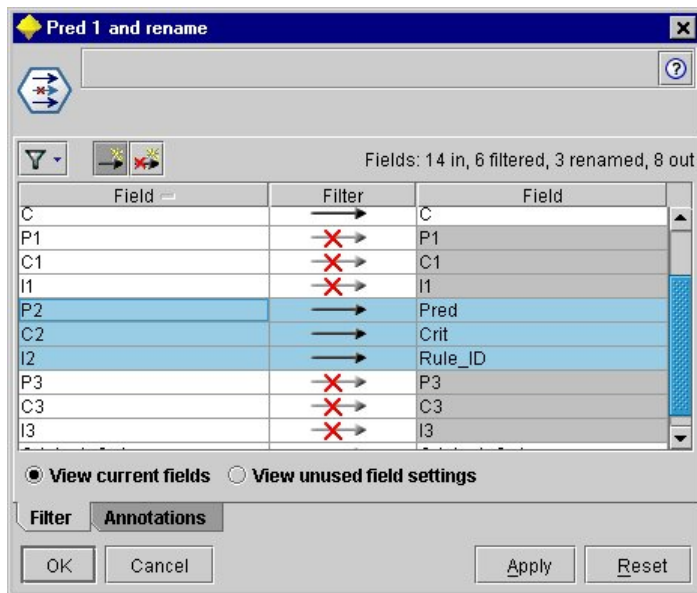


Figure 50. Filtering fields for predictions 1 and 3 while renaming fields for prediction 2.

4. Use an Append node to append values for the shared *Pred*, *Crit*, and *Rule\_ID*.
5. Attach a Sort node to sort records in ascending order for the field *Original\_order* and in descending order for *Crit*, which is the field used to sort predictions by criteria such as confidence, lift, and support.
6. Use another Filter node to filter the field *Original\_order* from the output.

At this point, the data are ready for deployment.

#### Transposing Transactional Scores

The process is similar for transposing transactional scores. For example, the stream shown below transposes scores to a format with a single prediction in each row as needed for deployment.

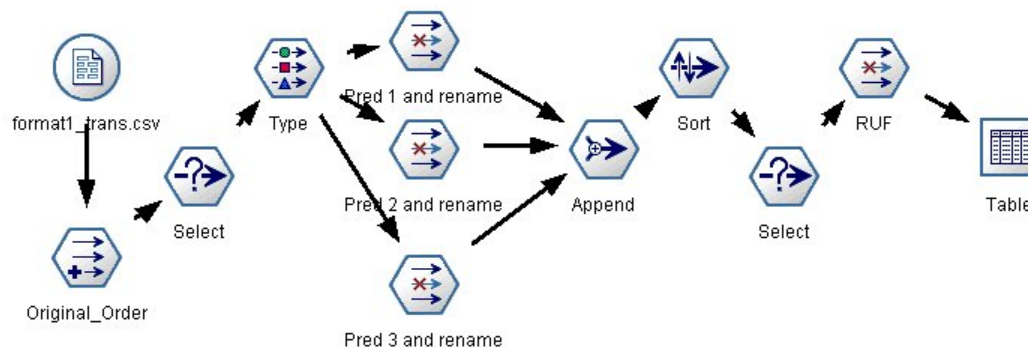


Figure 51. Example stream used to transpose transactional data into till-roll format

With the addition of two Select nodes, the process is identical to that explained earlier for tabular data.

- The first Select node is used to compare rule IDs across adjacent records and include only unique or undefined records. This Select node uses the CLEM expression to select records:  $ID \neq @OFFSET(ID, -1)$  or  $@OFFSET(ID, -1) = \text{undef}$ .
- The second Select node is used to discard extraneous rules, or rules where *Rule\_ID* has a null value. This Select node uses the following CLEM expression to discard records:  $\text{not} (@NULL(Rule\_ID))$ .

For more information on transposing scores for deployment, contact Technical Support.

## Sequence node

---

The Sequence node discovers patterns in sequential or time-oriented data, in the format bread -> cheese. The elements of a sequence are **item sets** that constitute a single transaction. For example, if a person goes to the store and purchases bread and milk and then a few days later returns to the store and purchases some cheese, that person's buying activity can be represented as two item sets. The first item set contains bread and milk, and the second one contains cheese. A **sequence** is a list of item sets that tend to occur in a predictable order. The Sequence node detects frequent sequences and creates a generated model node that can be used to make predictions.

**Requirements.** To create a Sequence rule set, you need to specify an ID field, an optional time field, and one or more content fields. Note that these settings must be made on the Fields tab of the modeling node; they cannot be read from an upstream Type node. The ID field can have any role or measurement level. If you specify a time field, it can have any role but its storage must be numeric, date, time, or timestamp. If you do not specify a time field, the Sequence node will use an implied timestamp, in effect using row numbers as time values. Content fields can have any measurement level and role, but all content fields must be of the same type. If they are numeric, they must be integer ranges (not real ranges).

**Strengths.** The Sequence node is based on the CARMA association rules algorithm, which uses an efficient two-pass method for finding sequences. In addition, the generated model node created by a Sequence node can be inserted into a data stream to create predictions. The generated model node can also generate SuperNodes for detecting and counting specific sequences and for making predictions based on specific sequences.

### Sequence Node Fields Options

Before executing a Sequence node, you must specify ID and content fields on the Fields tab of the Sequence node. If you want to use a time field, you also need to specify that here.

**ID field.** Select an ID field from the list. Numeric or symbolic fields can be used as the ID field. Each unique value of this field should indicate a specific unit of analysis. For example, in a market basket application, each ID might represent a single customer. For a Web log analysis application, each ID might represent a computer (by IP address) or a user (by login data).

- **IDs are contiguous.** If your data are presorted so that all records with the same ID are grouped together in the data stream, select this option to speed up processing. If your data are not presorted (or you are not sure), leave this option unselected, and the Sequence node will sort the data automatically.

*Note:* If your data are not sorted and you select this option, you may get invalid results in your Sequence model.

**Time field.** If you want to use a field in the data to indicate event times, select **Use time field** and specify the field to be used. The time field must be numeric, date, time, or timestamp. If no time field is specified, records are assumed to arrive from the data source in sequential order, and record numbers are used as time values (the first record occurs at time "1"; the second, at time "2"; and so on).

**Content fields.** Specify the content field(s) for the model. These fields contain the events of interest in sequence modeling.

The Sequence node can handle data in either tabular or transactional format. If you use multiple fields with transactional data, the items specified in these fields for a particular record are assumed to represent items found in a single transaction with a single timestamp. See the topic [“Tabular versus Transactional Data”](#) on page 232 for more information.

**Partition.** This field allows you to specify a field used to partition the data into separate samples for the training, testing, and validation stages of model building. By using one sample to generate the model and a different sample to test it, you can get a good indication of how well the model will generalize to larger datasets that are similar to the current data. If multiple partition fields have been defined by using Type or Partition nodes, a single partition field must be selected on the Fields tab in each modeling node that uses partitioning. (If only one partition is present, it is automatically used whenever partitioning is enabled.) Also note that to apply the selected partition in your analysis, partitioning must also be enabled

in the Model Options tab for the node. (Deselecting this option makes it possible to disable partitioning without changing field settings.)

## Sequence Node Model Options

**Model name.** You can generate the model name automatically based on the target or ID field (or model type in cases where no such field is specified) or specify a custom name.

**Use partitioned data.** If a partition field is defined, this option ensures that data from only the training partition is used to build the model.

**Minimum rule support (%)** You can specify a support criterion. *Rule support* refers to the proportion of IDs in the training data that contain the entire sequence. If you want to focus on more common sequences, increase this setting.

**Minimum rule confidence (%)** You can specify a confidence criterion for keeping sequences in the sequence set. *Confidence* refers to the percentage of the IDs where a correct prediction is made, out of all the IDs for which the rule makes a prediction. It is calculated as the number of IDs for which the entire sequence is found divided by the number of IDs for which the antecedents are found, based on the training data. Sequences with lower confidence than the specified criterion are discarded. If you are getting too many sequences or uninteresting sequences, try increasing this setting. If you are getting too few sequences, try decreasing this setting.

**Note:** If necessary, you can highlight the value and type in your own value. Be aware that if you reduce the confidence value below 1.0, in addition to the process requiring a lot of free memory, you might find that the rules take an extremely long time to build.

**Maximum sequence size** You can set the maximum number of distinct items in a sequence. If the sequences of interest are relatively short, you can decrease this setting to speed up building the sequence set.

**Predictions to add to stream** Specify the number of predictions to be added to the stream by the resulting generated Model node. For more information, see [“Sequence Model Nuggets” on page 248](#).

## Sequence Node Expert Options

For those with detailed knowledge of the Sequence node's operation, the following expert options allow you to fine-tune the model-building process. To access expert options, set the Mode to **Expert** on the Expert tab.

**Set maximum duration.** If this option is selected, sequences will be limited to those with a duration (the time between the first and last item set) less than or equal to the value specified. If you haven't specified a time field, the duration is expressed in terms of rows (records) in the raw data. If the time field used is a time, date, or timestamp field, the duration is expressed in seconds. For numeric fields, the duration is expressed in the same units as the field itself.

**Set pruning value.** The CARMA algorithm used in the Sequence node periodically removes (**prunes**) infrequent item sets from its list of potential item sets during processing to conserve memory. Select this option to adjust the frequency of pruning. The number specified determines the frequency of pruning. Enter a smaller value to decrease the memory requirements of the algorithm (but potentially increase the training time required), or enter a larger value to speed up training (but potentially increase memory requirements).

**Set maximum sequences in memory.** If this option is selected, the CARMA algorithm will limit its memory store of candidate sequences during model building to the number of sequences specified. Select this option if IBM SPSS Modeler is using too much memory during the building of Sequence models. Note that the maximum sequences value you specify here is the number of candidate sequences tracked internally as the model is built. This number should be much larger than the number of sequences you expect in the final model.

**Constrain gaps between item sets.** This option allows you to specify constraints on the time gaps that separate item sets. If selected, item sets with time gaps smaller than the **Minimum gap** or larger than the

**Maximum gap** that you specify will not be considered to form part of a sequence. Use this option to avoid counting sequences that include long time intervals or those that take place in a very short time span.

*Note:* If the time field used is a time, date, or timestamp field, the time gap is expressed in seconds. For numeric fields, the time gap is expressed in the same units as the time field.

For example, consider the following list of transactions.

Table 20. Example list of transactions		
ID	Time	Content
1001	1	apples
1001	2	bread
1001	5	cheese
1001	6	dressing

If you build a model on these data with the minimum gap set to 2, you would get the following sequences:

```
apples -> cheese
apples -> dressing
bread -> cheese
bread -> dressing
```

You would not see sequences such as apples -> bread because the gap between apples and bread is smaller than the minimum gap. Similarly, consider the following alternative data.

Table 21. Example list of transactions		
ID	Time	Content
1001	1	apples
1001	2	bread
1001	5	cheese
1001	20	dressing

If the maximum gap were set to 10, you would not see any sequences with dressing, because the gap between cheese and dressing is too large for them to be considered part of the same sequence.

## Sequence Model Nuggets

Sequence model nuggets represent the sequences found for a particular output field discovered by the Sequence node and can be added to streams to generate predictions.

When you run a stream containing a Sequence node, the node adds a pair of fields containing predictions and associated confidence values for each prediction from the sequence model to the data. By default, three pairs of fields containing the top three predictions (and their associated confidence values) are added. You can change the number of predictions generated when you build the model by setting the Sequence node model options at build time, as well as on the Settings tab after adding the model nugget to a stream. See the topic [“Sequence Model Nugget Settings”](#) on page 251 for more information.

The new field names are derived from the model name. The field names are *\$S-sequence-n* for the prediction field (where *n* indicates the *n*th prediction) and *\$SC-sequence-n* for the confidence field. In a stream with multiple Sequence Rules nodes in a series, the new field names will include numbers in the prefix to distinguish them from each other. The first Sequence Set node in the stream will use the usual names, the second node will use names starting with *\$S1-* and *\$SC1-*, the third node will use names

starting with *\$S2-* and *\$SC2-*, and so on. Predictions are displayed in order by confidence, so that *\$S-sequence-1* contains the prediction with the highest confidence, *\$S-sequence-2* contains the prediction with the next highest confidence, and so on. For records where the number of available predictions is smaller than the number of predictions requested, remaining predictions contain the value *\$null\$*. For example, if only two predictions can be made for a particular record, the values of *\$S-sequence-3* and *\$SC-sequence-3* will be *\$null\$*.

For each record, the rules in the model are compared to the set of transactions processed for the current ID so far, including the current record and any previous records with the same ID and earlier timestamp. The *k* rules with the highest confidence values that apply to this set of transactions are used to generate the *k* predictions for the record, where *k* is the number of predictions specified on the Settings tab after adding the model to the stream. (If multiple rules predict the same outcome for the transaction set, only the rule with the highest confidence is used.) See the topic [“Sequence Model Nugget Settings” on page 251](#) for more information.

As with other types of association rule models, the data format must match the format used in building the sequence model. For example, models built using tabular data can be used to score only tabular data. See the topic [“Scoring Association Rules” on page 243](#) for more information.

*Note:* When scoring data using a generated Sequence Set node in a stream, any tolerance or gap settings that you selected in building the model are ignored for scoring purposes.

### Predictions from Sequence Rules

The node handles the records in a time-dependent manner (or order-dependent, if no timestamp field was used to build the model). Records should be sorted by the ID field and timestamp field (if present). However, predictions are not tied to the timestamp of the record to which they are added. They simply refer to the most likely items to occur *at some point in the future*, given the history of transactions for the current ID up to the current record.

Note that the predictions for each record do not necessarily depend on that record's transactions. If the current record's transactions do not trigger a specific rule, rules will be selected based on the previous transactions for the current ID. In other words, if the current record doesn't add any useful predictive information to the sequence, the prediction from the last useful transaction for this ID is carried forward to the current record.

For example, suppose you have a Sequence model with the single rule

```
Jam -> Bread (0.66)
```

and you pass it the following records.

Table 22. Example records		
ID	Purchase	Prediction
001	jam	bread
001	milk	bread

Notice that the first record generates a prediction of *bread*, as you would expect. The second record also contains a prediction of *bread*, because there's no rule for *jam* followed by *milk*; therefore, the *milk* transaction doesn't add any useful information, and the rule *Jam -> Bread* still applies.

### Generating New Nodes

The Generate menu allows you to create new SuperNodes based on the sequence model.

- **Rule SuperNode.** Creates a SuperNode that can detect and count occurrences of sequences in scored data. This option is disabled if no rule is selected. See the topic [“Generating a Rule SuperNode from a Sequence Model Nugget” on page 251](#) for more information.
- **Model to Palette.** Returns the model to the Models palette. This is useful in situations where a colleague may have sent you a stream containing the model and not the model itself.

## Sequence Model Nugget Details

The Model tab for a Sequence model nugget displays the rules extracted by the algorithm. Each row in the table represents a rule, with the antecedent (the "if" part of the rule) in the first column followed by the consequent (the "then" part of the rule) in the second column.

Each rule is shown in the following format.

Table 23. Rule format	
Antecedent	Consequent
beer and cannedveg	beer
fish	fish
fish	

The first example rule is interpreted as *for IDs that had "beer" and "cannedveg" in the same transaction, there is likely a subsequent occurrence of "beer."* The second example rule can be interpreted as *for IDs that had "fish" in one transaction and then "fish" in another, there is a likely subsequent occurrence of "fish."* Note that in the first rule, *beer* and *cannedveg* are purchased at the same time; in the second rule, *fish* is purchased in two separate transactions.

**Sort menu.** The Sort menu button on the toolbar controls the sorting of rules. Direction of sorting (ascending or descending) can be changed using the sort direction button (up or down arrow).

You can sort rules by:

- Support %
- Confidence %
- Rule Support %
- Consequent
- First Antecedent
- Last Antecedent
- Number of Items (antecedents)

For example, the following table is sorted in descending order by number of items. Rules with multiple items in the antecedent set precede those with fewer items.

Table 24. Rules sorted by number of items	
Antecedent	Consequent
beer and cannedveg and frozenmeal	frozenmeal
beer and cannedveg	beer
fish	fish
fish	
softdrink	softdrink

**Show/hide criteria menu.** The Show/hide criteria menu button (grid icon) controls options for the display of rules. The following display options are available:

- **Instances** displays information about the number of unique IDs for which the *full sequence*—both antecedents and consequent—occurs. (Note this differs from Association models, for which the number of instances refers to the number of IDs for which *only* the antecedents apply.) For example, given the rule *bread -> cheese*, the number of IDs in the training data that include both *bread* and *cheese* are referred to as **instances**.

- **Support** displays the proportion of IDs in the training data for which the antecedents are true. For example, if 50% of the training data includes the antecedent *bread* then the support for the *bread* -> *cheese* rule would be 50%. (Unlike Association models, support is *not* based on the number of instances, as noted earlier.)
- **Confidence** displays the percentage of the IDs where a correct prediction is made, out of all the IDs for which the rule makes a prediction. It is calculated as the number of IDs for which the entire sequence is found divided by the number of IDs for which the antecedents are found, based on the training data. For example, if 50% of the training data contains *cannedveg* (indicating antecedent support) but only 20% contains both *cannedveg* and *frozenmeal*, then confidence for the rule *cannedveg* -> *frozenmeal* would be  $\text{Rule Support} / \text{Antecedent Support}$  or, in this case, 40%.
- **Rule Support** for Sequence models is based on instances and displays the proportion of training records for which the entire rule, antecedents, and consequent(s), are true. For example, if 20% of the training data contains both *bread* and *cheese*, then rule support for the rule *bread* -> *cheese* is 20%.

Note that the proportions are based on valid transactions (transactions with at least one observed item or true value) rather than total transactions. Invalid transactions—those with no items or true values—are discarded for these calculations.

**Filter button.** The Filter button (funnel icon) on the menu expands the bottom of the dialog box to show a panel where active rule filters are displayed. Filters are used to narrow the number of rules displayed on the Models tab.



Figure 52. Filter button

To create a filter, click the Filter icon to the right of the expanded panel. This opens a separate dialog box in which you can specify constraints for displaying rules. Note that the Filter button is often used in conjunction with the Generate menu to first filter rules and then generate a model containing that subset of rules. For more information, see [“Specifying Filters for Rules”](#) on page 240 below.

### Sequence Model Nugget Settings

The Settings tab for a Sequence model nugget displays scoring options for the model. This tab is available only after the model has been added to the stream canvas for scoring.

**Maximum number of predictions.** Specify the maximum number of predictions included for each set of basket items. The rules with the highest confidence values that apply to this set of transactions are used to generate predictions for the record up to the specified limit.

### Sequence Model Nugget Summary

The Summary tab for a sequence rule model nugget displays the number of rules discovered and the minimum and maximum for support and confidence in the rules. If you have executed an Analysis node attached to this modeling node, information from that analysis will also be displayed in this section.

See the topic [“Browsing model nuggets”](#) on page 35 for more information.

### Generating a Rule SuperNode from a Sequence Model Nugget

To generate a rule SuperNode based on a sequence rule:

1. On the Model tab for the sequence rule model nugget, click on a row in the table to select the desired rule.
2. From the rule browser menus choose:

#### Generate > Rule SuperNode

**Important:** To use the generated SuperNode, you must sort the data by ID field (and Time field, if any) before passing them into the SuperNode. The SuperNode will not detect sequences properly in unsorted data.

You can specify the following options for generating a rule SuperNode:



**Detect.** Specifies how matches are defined for data passed into the SuperNode.

- **Antecedents only.** The SuperNode will identify a match any time it finds the antecedents for the selected rule in the correct order within a set of records having the same ID, regardless of whether the consequent is also found. Note that this does not take into account timestamp tolerance or item gap constraint settings from the original Sequence modeling node. When the last antecedent item set is detected in the stream (and all other antecedents have been found in the proper order), all subsequent records with the current ID will contain the summary selected below.
- **Entire sequence.** The SuperNode will identify a match any time it finds the antecedents and the consequent for the selected rule in the correct order within a set of records having the same ID. This does not take into account timestamp tolerance or item gap constraint settings from the original Sequence modeling node. When the consequent is detected in the stream (and all antecedents have also been found in the correct order), the current record and all subsequent records with the current ID will contain the summary selected below.

**Display.** Controls how match summaries are added to the data in the Rule SuperNode output.

- **Consequent value for first occurrence.** The value added to the data is the consequent value predicted based on the first occurrence of the match. Values are added as a new field named *rule\_n\_consequent*, where *n* is the rule number (based on the order of creation of Rule SuperNodes in the stream).
- **True value for first occurrence.** The value added to the data is true if there is at least one match for the ID and false if there is no match. Values are added as a new field named *rule\_n\_flag*.
- **Count of occurrences.** The value added to the data is the number of matches for the ID. Values are added as a new field named *rule\_n\_count*.
- **Rule number.** The value added is the rule number for the selected rule. **Rule numbers** are assigned based on the order in which the SuperNode was added to the stream. For example, the first Rule SuperNode is considered *rule 1*, the second Rule SuperNode is considered *rule 2*, etc. This option is most useful when you will be including multiple Rule SuperNodes in your stream. Values are added as a new field named *rule\_n\_number*.
- **Include confidence figures.** If selected, this option will add the rule confidence to the data stream as well as the selected summary. Values are added as a new field named *rule\_n\_confidence*.

## Association Rules node

---

Association rules are statements of the following form.

For example, "If a customer purchases a razor and after shave, then that customer will purchase shaving cream with 80% confidence." The Association Rules node extracts a set of rules from the data, pulling out the rules with the highest information content. The Association Rules node is very similar to the Apriori node, however, there are some notable differences:

- The Association Rules node cannot process transactional data.
- The Association Rules node can process data that has the List storage type and the Collection measurement level.
- The Association Rules node can be used with IBM SPSS Analytic Server. This provides scalability and means that you can process big data and take advantage of faster parallel processing.
- The Association Rules node provides additional settings, such as the ability to restrict the number of rules that are generated, thereby increasing the processing speed.
- Output from the model nugget is shown in the Output Viewer.

**Note:** The Association Rules node does not support the Model Evaluation or Champion Challenger steps in IBM SPSS Collaboration and Deployment Services.

**Note:** The Association Rules node ignores empty records when building a model if the field type is a flag. Empty records are records where all the fields used in the model build have a false value.

A stream that shows a working example of using Association Rules, named *geospatial\_association.str*, and which references the data files *InsuranceData.sav*, *CountyData.sav*, and *ChicagoAreaCounties.shp* is available from the Demos directory of your IBM



SPSS Modeler installation. You can access the Demos directory from the IBM SPSS Modeler program group on the Windows Start menu. The `geospatial_association.str` file is in the `streams` directory.

## Association Rules - Fields Options

On the **Fields** tab, you choose whether you want to use the field role settings that are already defined in upstream nodes, such as a previous Type node, or make the field assignments manually.

### Use predefined roles

This option uses the role settings (such as targets, or predictors) from an upstream Type node (or the Types tab of an upstream source node). Fields with an input role are considered to be Conditions, fields with a target role are considered to be Predictions, and those fields that are used as inputs and targets are considered to have both roles.

### Use custom field assignments

Choose this option if you want to assign targets, predictors, and other roles manually on this screen.

### Fields

If you selected **Use custom field assignments**, use the arrow buttons to assign items manually from this list to the boxes on the right of the screen. The icons indicate the valid measurement levels for each field.

### Both (Condition or Prediction)

Fields added to this list can take either the condition or prediction role in rules that are generated by the model. This is on a rule by rule basis, so a field might be a condition in one rule and a prediction in another.

### Prediction only

Fields added to this list can appear only as a prediction (also known as a "consequent") of a rule. The presence of a field in this list does not mean that the field is used in any rules, only that if it is used it can be only a prediction.

### Condition only

Fields added to this list can appear only as a condition (also known as an "antecedent") of a rule. The presence of a field in this list does not mean that the field is used in any rules, only that if it is used it can be only a condition.

## Association Rules - Rule building

### Items per rule

Use these options to specify how many items, or values, can be used in each rule.

**Note:** The combined total of these two fields cannot exceed 10.

### Maximum conditions

Select the maximum number of conditions that can be included in a single rule.

### Maximum predictions

Select the maximum number of predictions that can be included in a single rule.

### Rule building

Use these options to specify the number and type of rules to build.

### Maximum number of rules

Specify the maximum number of rules that can be considered for use in the building of rules for your model.

### Rule criterion for top N

Select the criterion that is used to establish which are the top N rules, where N is the value that is entered in the **Maximum number of rules** field. You can choose from the following criterion.

- **Confidence**

- **Rule Support**
- **Condition Support**
- **Lift**
- **Deployability**

#### **Only true values for flags**

When your data is in tabular format, select this option to include only true values for flag fields in the resulting rules. Selecting true values can help make rules easier to understand. The option does not apply to data in transactional format. For more information, see [“Tabular versus Transactional Data” on page 232.](#)

#### **Rule criterion**

If you select **Enable rule criterion**, you can use these options to select the minimum strength that rules must meet to be considered for use in your model.

- **Confidence** Specify the minimum percentage value for the Confidence level for a rule that is produced by the model. If the model produces a rule with a level less than this amount, the rule is discarded.
- **Rule Support** Specify the minimum percentage value for the Rule Support level for a rule that is produced by the model. If the model produces a rule with a level less than this amount, the rule is discarded.
- **Condition Support** Specify the minimum percentage value for the Condition Support level for a rule that is produced by the model. If the model produces a rule with a level less than the specified amount, the rule is discarded.
- **Lift** Specify the minimum Lift value that is allowed for a rule that is produced by the model. If the model produces a rule with a value less than the specified amount, the rule is discarded.

#### **Exclude rules**

In some cases, the association between two or more fields is known or is self-evident, in such a case you can exclude rules where the fields predict each other. By excluding rules that contain both values, you reduce irrelevant input and increase the chances of finding useful results.

#### **Fields**

Select the associated fields that you do not want to use together in rule building. For example, associated fields might be Car Manufacturer and Car Model, or School Year and Age of Pupil. When the model creates rules, if the rule contains at least one of the fields selected on either side of the rule (condition or prediction), the rule is discarded.

## **Association Rules - Transformations**

#### **Binning**

Use these options to specify how continuous (numeric range) fields are binned.

#### **Number of bins**

Any continuous fields set to be automatically binned are divided into the number of equally spaced bins that you specify. You can select any number in the range 2 - 10.

#### **List fields**

#### **Maximum list length**

To restrict the number of items to be included in the model if the length of a list field is unknown, enter the maximum length of the list. You can select any number in the range 1 - 100. If a list is longer than the number you enter, the model will still use the field but only include values up to this number; any extra values in the field are ignored.

## Association Rules - Output

Use the options in this pane to control what output is generated when the model is built.

### Rules tables

Use these options to create one or more table types that display the best number of rules (based on a number you specify) for each selected criterion.

#### Confidence

Confidence is the ratio of rule support to condition support. Of the items with the listed condition values, the percentage that has the predicted consequent values. Creates a table that contains the best N association rules that are based on confidence to be included in the output (where N is the **Rules to display** value).

#### Rule Support

The proportion of items for which the entire rule, conditions, and predictions are true. For all items in the dataset, the percentage that is correctly accounted for and predicted by the rule. This measure gives an overall importance of the rule. Creates a table that contains the best N association rules that are based on rule support to be included in the output (where N is the **Rules to display** value).

#### Lift

The ratio of rule confidence and the prior probability of having the prediction. The ratio of the Confidence value for a rule versus the percentage the Consequent values occur in the overall population. This ratio gives a measure of how well the rule improves over chance. Creates a table that contains the best N association rules that are based on lift to be included in the output (where N is the **Rules to display** value).

#### Condition Support

The proportion of items for which the conditions are true. Creates a table that contains the best N association rules that are based on antecedent support to be included in the output (where N is the **Rules to display** value).

#### Deployability

A measure of what percentage of the training data satisfies the condition but not the prediction. This measure shows how often the rule misses. It is effectively the opposite of Confidence. Creates a table that contains the best N association rules that are based on deployability to be included in the output (N is the **Rules to display** value).

#### Rules to display

Set the maximum number of rules to display in the tables.

### Model information tables

Use one or more of these options to select which model tables to include in the output.

- **Field Transformations**
- **Records Summary**
- **Rule Statistics**
- **Most Frequent Values**
- **Most Frequent Fields**

### Sortable word cloud of rules.

Use these options to create a word cloud that displays the rules outputs. Words are displayed in increasing text sizes to indicate their importance.

#### Create a sortable word cloud.

Check this box to create a sortable word cloud in your output.

### Default sort

Select the sort type to be used to when initially creating the word cloud. The word cloud is interactive and you can change the criterion in the Model Viewer to see different rules and sorts. You can choose from the following sort options:

- Confidence.
- Rule Support
- Lift
- Condition Support.
- Deployability

### Max rules to display

Set the number of rules to be displayed in the word cloud; the maximum you can choose is 20.

## Association Rules - Model Options

Use the settings on this tab to specify the scoring options for Association Rules models.

**Model name** You can generate the model name that is automatically based on the target field (or model type in cases where no such field is specified), or specify a custom name.

**Maximum number of predictions** Specify the maximum number of predictions that are included in the score result. This option is used with the **Rule Criterion** entries to produce the “top” predictions, where “top” indicates the highest level of confidence, support, lift, and so on.

**Rule Criterion** Select the measure that is used to determine the strength of the rules. Rules are sorted by the strength of criteria that are selected here in order to return the top predictions for an item set. You can choose from 5 different criteria.

- **Confidence** Confidence is the ratio of rule support to condition support. Of the items with the listed condition values, the percentage that has the predicted consequent values.
- **Condition Support** The proportion of items for which the conditions are true.
- **Rule Support** The proportion of items for which the entire rule, conditions, and predictions are true. Calculated by multiplying the **Condition Support** value by the **Confidence** value.
- **Lift** The ratio of rule confidence and the prior probability of having the prediction.
- **Deployability** A measure of what percentage of the training data satisfies the condition but not the prediction.

**Allow repeat predictions** To include multiple rules with the same prediction during scoring, select this check box. For example, selecting this enables the following rules to be scored.

```
bread & cheese -> wine  
cheese & fruit -> wine
```

**Note:** Rules with multiple predictions (bread & cheese & fruit -> wine & pate) are considered repeat predictions only if all predictions (wine & pate) were predicted before.

**Only score rules when predictions are not present in the input** To ensure that predictions are not also present in the input, select this option. For example, if the purpose of scoring is to make a home furniture product recommendation, then it is unlikely that input that already contains a dining room table is likely to purchase another. In such a case, select this option. However, if products are perishable or disposable (such as cheese, baby formula, or tissue), then rules where the consequent is already present in the input might be of value. In the latter case, the most useful option might be **Score all rules**.

**Only score rules when predictions are present in the input** To ensure that predictions are also present in the input, select this option. This approach is useful when you are attempting to gain insight into existing customers or transactions. For example, you might want to identify rules with the highest lift and then explore which customers fit these rules.

**Score all rules** To include all rules during scoring, regardless of the presence or absence of predictions, select this option.

## Association Rules Model Nuggets

The model nugget contains information about the rules extracted from your data during model building.

### Viewing Results

You can browse the rules generated by Association Rules models by using the Model tab on the dialog box. Browsing a model nugget shows you the information about the rules before generating new nodes or scoring the model.

### Scoring the Model

Refined model nuggets may be added to a stream and used for scoring. See the topic [“Using Model Nuggets in Streams”](#) on page 41 for more information. Model nuggets used for scoring include an extra Settings tab on their respective dialog boxes. See the topic [“Association Rules Model Nugget Settings”](#) on page 257 for more information.

### Association Rules Model Nugget Details

The Association Rules model nugget displays details of the model in the Model tab of the Output Viewer. For more information on using the viewer, see the section titled "Working with Output" in the Modeler User's Guide (ModelerUsersGuide.pdf).

The GSAR modeling operation creates a number of new fields with the prefix \$A as shown in the following table.

Table 25. New fields created by the Association Rules modeling operation	
Field name	Description
\$A-<prediction>#	This field contains the prediction from the model for the scored records. <prediction> is the name of the field included in the Predictions role in the model, and # is a sequence of numbers for the output rules (for example, if the score is set to include 3 rules the sequence of numbers will be from 1 to 3).
\$AC-<prediction>#	This field contains the confidence in the prediction. <prediction> is the name of the field included in the Predictions role in the model, and # is a sequence of numbers for the output rules (for example, if the score is set to include 3 rules the sequence of numbers will be from 1 to 3).
\$A-Rule_ID#	This column contains the ID of the rule predicted for each record in the scored data set. # is a sequence of numbers for the output rules (for example, if the score is set to include 3 rules the sequence of numbers will be from 1 to 3).

### Association Rules Model Nugget Settings

The Settings tab for an Association Rules model nugget displays scoring options for the model. This tab is available only after the model is added to the stream canvas for scoring.

**Maximum number of predictions** Specify the maximum number of predictions that are included for each set of items. The rules with the highest confidence values that apply to this set of transactions are used to generate predictions for the record up to the specified limit. Use this option with the **Rule Criterion** option

to produce the “top” predictions, where *top* indicates the highest level of confidence, support, lift, and so on.

**Rule Criterion** Select the measure that is used to determine the strength of rules. Rules are sorted by the strength of criteria that are selected here in order to return the top predictions for an item set. You can choose from the following criteria.

- **Confidence**
- **Rule Support**
- **Lift**
- **Condition Support**
- **Deployability**

**Allow repeat predictions** To include multiple rules with the same consequent when scoring, select this check box. For example, selecting this option means that the following rule can be scored:

```
bread & cheese -> wine  
cheese & fruit -> wine
```

To exclude repeat predictions when scoring, clear the check box.

**Note:** Rules with multiple consequents (bread & cheese & fruit -> wine & pate) are considered repeat predictions only if all consequents (wine & pate) have been predicted before.

**Only score rules when predictions are not present in the input** Select to ensure that consequents are not also present in the input. For example, if the purpose of scoring is to make a home furniture product recommendation, then it is unlikely that input that already contains a dining room table is likely to purchase another one. In such a case, select this option. On the other hand, if products are perishable or disposable (such as cheese, baby formula, or tissue), then rules where the consequent is already present in the input might be of value. In the latter case, the most useful option might be **Score all rules**.

**Only score rules when predictions are present in the input** Select this option to ensure that consequents are also present in the input. This approach is useful when you are attempting to gain insight into existing customers or transactions. For example, you might want to identify rules with the highest lift and then explore which customers fit these rules.

**Score all rules** To include all rules when scoring, regardless of the presence or absence of consequents in the input, select this option.

---

# Chapter 13. Time Series Models

## Why forecast?

---

To forecast means to predict the values of one or more series over time. For example, you may want to predict the expected demand for a line of products or services in order to allocate resources for manufacturing or distribution. Because planning decisions take time to implement, forecasts are an essential tool in many planning processes.

Methods of modeling time series assume that history repeats itself—if not exactly, then closely enough that by studying the past, you can make better decisions in the future. To predict sales for next year, for example, you would probably start by looking at this year's sales and work backward to figure out what trends or patterns, if any, have developed in recent years. But patterns can be difficult to gauge. If your sales increase several weeks in a row, for example, is this part of a seasonal cycle or the beginning of a long-term trend?

Using statistical modeling techniques, you can analyze the patterns in your past data and project those patterns to determine a range within which future values of the series are likely to fall. The result is more accurate forecasts on which to base your decisions.

## Time series data

---

A **time series** is an ordered collection of measurements taken at regular intervals—for example, daily stock prices or weekly sales data. The measurements may be of anything that interests you, and each series can generally be classified as one of the following:

- **Dependent.** A series that you want to forecast.
- **Predictor.** A series that may help to explain the target—for example, using an advertising budget to predict sales. Predictors can only be used with ARIMA models.
- **Event.** A special predictor series used to account for predictable recurring incidents—for example, sales promotions.
- **Intervention.** A special predictor series used to account for one-time past incidents—for example, a power outage or employee strike.

The intervals can represent any unit of time, but the interval must be the same for all measurements. Moreover, any interval for which there is no measurement must be set to the missing value. Thus, the number of intervals with measurements (including those with missing values) defines the length of time of the historical span of the data.

## Characteristics of time series

Studying the past behavior of a series will help you identify patterns and make better forecasts. When plotted, many time series exhibit one or more of the following features:

- Trends
- Seasonal and nonseasonal cycles
- Pulses and steps
- Outliers

### Trends

A **trend** is a gradual upward or downward shift in the level of the series or the tendency of the series values to increase or decrease over time.

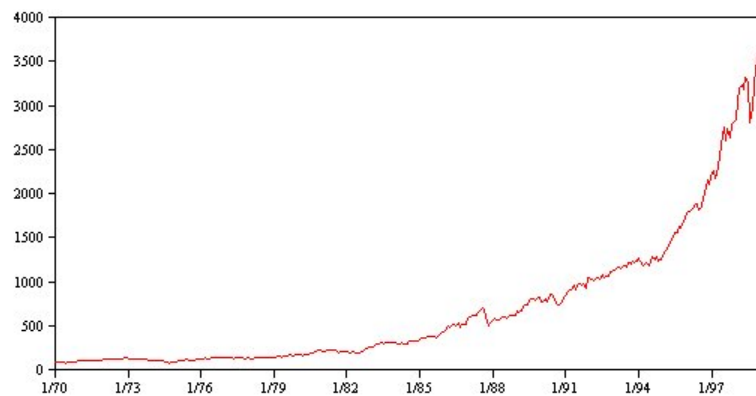


Figure 53. Trend

Trends are either **local** or **global**, but a single series can exhibit both types. Historically, series plots of the stock market index show an upward global trend. Local downward trends have appeared in times of recession, and local upward trends have appeared in times of prosperity.

Trends can also be either **linear** or **nonlinear**. Linear trends are positive or negative additive increments to the level of the series, comparable to the effect of simple interest on principal. Nonlinear trends are often multiplicative, with increments that are proportional to the previous series value(s).

Global linear trends are fit and forecast well by both exponential smoothing and ARIMA models. In building ARIMA models, series showing trends are generally differenced to remove the effect of the trend.

### Seasonal cycles

A **seasonal cycle** is a repetitive, predictable pattern in the series values.

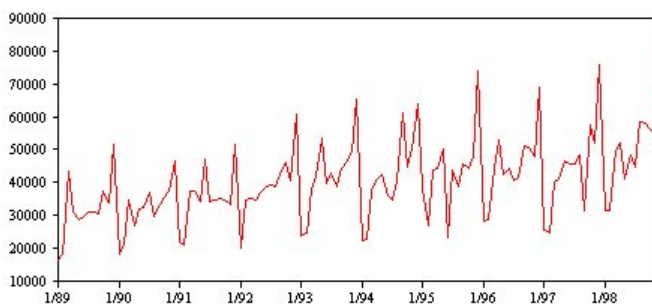


Figure 54. Seasonal cycle

Seasonal cycles are tied to the interval of your series. For instance, monthly data typically cycles over quarters and years. A monthly series might show a significant quarterly cycle with a low in the first quarter or a yearly cycle with a peak every December. Series that show a seasonal cycle are said to exhibit **seasonality**.

Seasonal patterns are useful in obtaining good fits and forecasts, and there are exponential smoothing and ARIMA models that capture seasonality.

### Nonseasonal cycles

A **nonseasonal cycle** is a repetitive, possibly unpredictable, pattern in the series values.



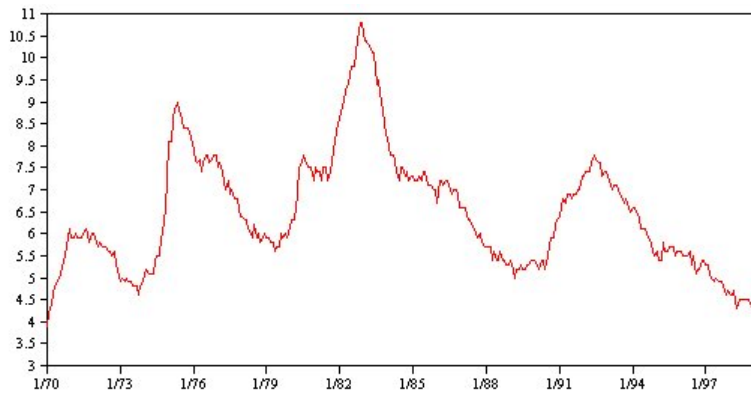


Figure 55. Nonseasonal cycle

Some series, such as unemployment rate, clearly display cyclical behavior; however, the periodicity of the cycle varies over time, making it difficult to predict when a high or low will occur. Other series may have predictable cycles but do not neatly fit into the Gregorian calendar or have cycles longer than a year. For example, the tides follow the lunar calendar, international travel and trade related to the Olympics swell every four years, and there are many religious holidays whose Gregorian dates change from year to year.

Nonseasonal cyclical patterns are difficult to model and generally increase uncertainty in forecasting. The stock market, for example, provides numerous instances of series that have defied the efforts of forecasters. All the same, nonseasonal patterns must be accounted for when they exist. In many cases, you can still identify a model that fits the historical data reasonably well, which gives you the best chance to minimize uncertainty in forecasting.

### Pulses and steps

Many series experience abrupt changes in level. They generally come in two types:

- A sudden, *temporary* shift, or **pulse**, in the series level
- A sudden, *permanent* shift, or **step**, in the series level

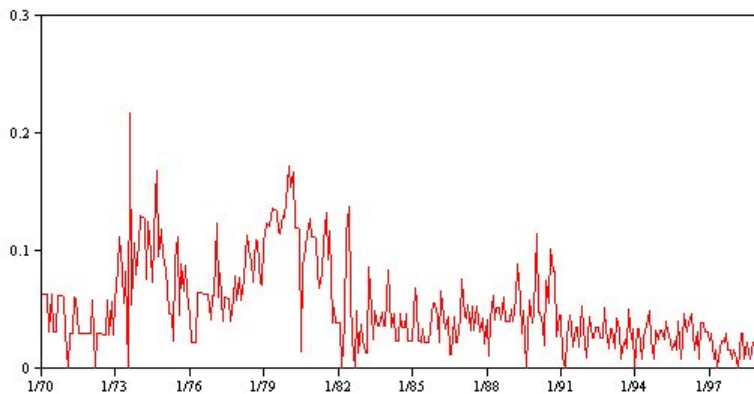


Figure 56. Series with a pulse

When steps or pulses are observed, it is important to find a plausible explanation. Time series models are designed to account for gradual, not sudden, change. As a result, they tend to underestimate pulses and be ruined by steps, which lead to poor model fits and uncertain forecasts. (Some instances of seasonality may appear to exhibit sudden changes in level, but the level is constant from one seasonal period to the next.)

If a disturbance can be explained, it can be modeled using an **intervention** or **event**. For example, during August 1973, an oil embargo imposed by the Organization of Petroleum Exporting Countries (OPEC) caused a drastic change in the inflation rate, which then returned to normal levels in the ensuing months. By specifying a **point intervention** for the month of the embargo, you can improve the fit of your model, thus indirectly improving your forecasts. For example, a retail store might find that sales were much

higher than usual on the day all items were marked 50% off. By specifying the 50%-off promotion as a recurring **event**, you can improve the fit of your model and estimate the effect of repeating the promotion on future dates.

## Outliers

Shifts in the level of a time series that cannot be explained are referred to as **outliers**. These observations are inconsistent with the remainder of the series and can dramatically influence the analysis and, consequently, affect the forecasting ability of the time series model.

The following figure displays several types of outliers commonly occurring in time series. The blue lines represent a series without outliers. The red lines suggest a pattern that might be present if the series contained outliers. These outliers are all classified as **deterministic** because they affect only the mean level of the series.

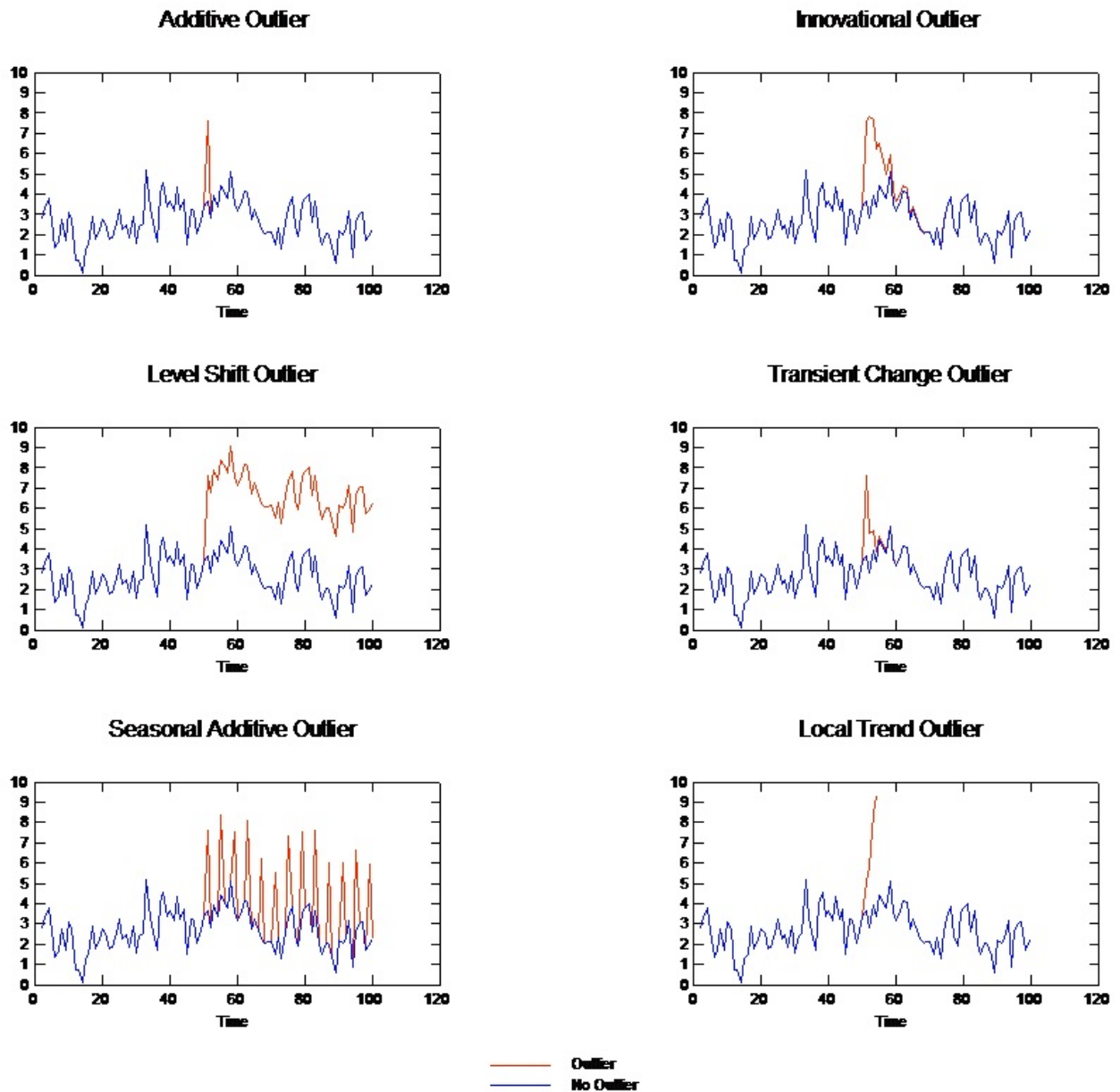


Figure 57. Outlier types

- **Additive Outlier.** An additive outlier appears as a surprisingly large or small value occurring for a single observation. Subsequent observations are unaffected by an additive outlier. Consecutive additive outliers are typically referred to as **additive outlier patches**.
- **Innovational Outlier.** An innovational outlier is characterized by an initial impact with effects lingering over subsequent observations. The influence of the outliers may increase as time proceeds.
- **Level Shift Outlier.** For a level shift, all observations appearing after the outlier move to a new level. In contrast to additive outliers, a level shift outlier affects many observations and has a permanent effect.
- **Transient Change Outlier.** Transient change outliers are similar to level shift outliers, but the effect of the outlier diminishes exponentially over the subsequent observations. Eventually, the series returns to its normal level.
- **Seasonal Additive Outlier.** A seasonal additive outlier appears as a surprisingly large or small value occurring repeatedly at regular intervals.
- **Local Trend Outlier.** A local trend outlier yields a general drift in the series caused by a pattern in the outliers after the onset of the initial outlier.

Outlier detection in time series involves determining the location, type, and magnitude of any outliers present. Tsay (1988) proposed an iterative procedure for detecting mean level change to identify deterministic outliers. This process involves comparing a time series model that assumes no outliers are present to another model that incorporates outliers. Differences between the models yield estimates of the effect of treating any given point as an outlier.

## Autocorrelation and partial autocorrelation functions

Autocorrelation and partial autocorrelation are measures of association between current and past series values and indicate which past series values are most useful in predicting future values. With this knowledge, you can determine the order of processes in an ARIMA model. More specifically,

- **Autocorrelation function (ACF).** At lag  $k$ , this is the correlation between series values that are  $k$  intervals apart.
- **Partial autocorrelation function (PACF).** At lag  $k$ , this is the correlation between series values that are  $k$  intervals apart, accounting for the values of the intervals between.

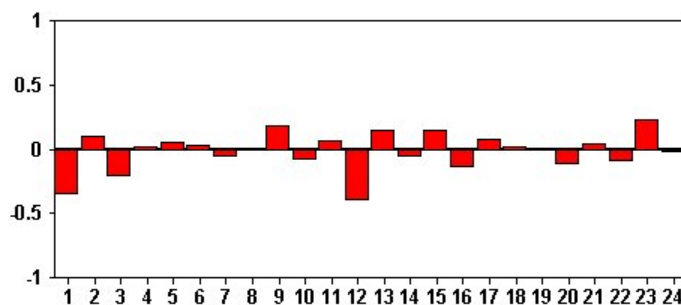


Figure 58. ACF plot for a series

The  $x$  axis of the ACF plot indicates the lag at which the autocorrelation is computed; the  $y$  axis indicates the value of the correlation (between  $-1$  and  $1$ ). For example, a spike at lag 1 in an ACF plot indicates a strong correlation between each series value and the preceding value, a spike at lag 2 indicates a strong correlation between each value and the value occurring two points previously, and so on.

- A positive correlation indicates that large current values correspond with large values at the specified lag; a negative correlation indicates that large current values correspond with small values at the specified lag.
- The absolute value of a correlation is a measure of the strength of the association, with larger absolute values indicating stronger relationships.

## Series transformations

Transformations are often useful for stabilizing a series before estimating models. This is particularly important for ARIMA models, which require series to be **stationary** before models are estimated. A series is stationary if the global level (mean) and average deviation from the level (variance) are constant throughout the series.

While most interesting series are not stationary, ARIMA is effective as long as the series can be made stationary by applying transformations, such as the natural log, differencing, or seasonal differencing.

**Variance stabilizing transformations.** Series in which the variance changes over time can often be stabilized using a natural log or square root transformation. These are also called functional transformations.

- **Natural log.** The natural logarithm is applied to the series values.
- **Square root.** The square root function is applied to the series values.

Natural log and square root transformations cannot be used for series with negative values.

**Level stabilizing transformations.** A slow decline of the values in the ACF indicates that each series value is strongly correlated with the previous value. By analyzing the change in the series values, you obtain a stable level.

- **Simple differencing.** The differences between each value and the previous value in the series are computed, except the oldest value in the series. This means that the differenced series will have one less value than the original series.
- **Seasonal differencing.** Identical to simple differencing, except that the differences between each value and the previous seasonal value are computed.

When either simple or seasonal differencing is simultaneously in use with either the log or square root transformation, the variance stabilizing transformation is always applied first. When simple and seasonal differencing are both in use, the resulting series values are the same whether simple differencing or seasonal differencing is applied first.

## Predictor series

---

Predictor series include related data that may help explain the behavior of the series to be forecast. For example, a Web- or catalog-based retailer might forecast sales based on the number of catalogs mailed, the number of phone lines open, or the number of hits to the company Web page.

Any series can be used as a predictor provided that the series extends as far into the future as you want to forecast and has complete data with no missing values.

Use care when adding predictors to a model. Adding large numbers of predictors will increase the time required to estimate models. While adding predictors may improve a model's ability to fit the historical data, it doesn't necessarily mean that the model does a better job of forecasting, so the added complexity may not be worth the trouble. Ideally, the goal should be to identify the simplest model that does a good job of forecasting.

As a general rule, it is recommended that the number of predictors should be less than the sample size divided by 15 (at most, one predictor per 15 cases).

**Predictors with missing data.** Predictors with incomplete or missing data cannot be used in forecasting. This applies to both historical data and future values. In some cases, you can avoid this limitation by setting the model's estimation span to exclude the oldest data when estimating models.

## Spatio-Temporal Prediction modeling node

---

Spatio-Temporal prediction (STP) has many potential applications such as energy management for buildings or facilities, performance analysis and forecasting for mechanical service engineers, or public transport planning. In these applications measurements such as energy usage are often taken over space and time. Questions that might be relevant to the recording of these measurements include what factors will affect future observations, what can be done to effect a wanted change, or to better manage the

system? To address these questions, you can use statistical techniques that can forecast future values at different locations, and can explicitly model adjustable factors to perform what-if analysis.

STP analysis uses data that contains location data, input fields for prediction (predictors), a time field, and a target field. Each location has numerous rows in the data that represent the values of each predictor at each time of measurement. After the data is analyzed, it can be used to predict target values at any location within the shape data that is used in the analysis. It can also forecast when input data for the future points in time are known.

**Note:** The STP node does not support the Model Evaluation or Champion Challenger steps in IBM SPSS Collaboration and Deployment Services.

A stream that shows a worked example of using STP, named `stp_server_demo.str`, and which references the data files `room_data.csv` and `score_data.csv` is available from the Demos directory of your IBM SPSS Modeler installation. You can access the Demos directory from the IBM SPSS Modeler program group on the Windows Start menu. The `stp_server_demo.str` file is in the `streams` directory.

## Spatio-Temporal Prediction - Fields Options

On the Fields tab, you choose whether you want to use the field role settings that are already defined in upstream nodes, or make the field assignments manually.

### Use predefined roles

This option uses the role settings (targets and predictors only) from an upstream Type node (or the Types tab of an upstream source node).

### Use custom field assignments

To assign targets, predictors, and other roles manually on this screen, select this option.

### Fields

Displays all of the fields in the data that can be selected. Use the arrow buttons to assign items manually from this list to the various boxes on the right of the screen. The icons indicate the valid measurement levels for each field.

**Note:** STP requires 1 record per location, per time interval to function correctly; therefore these are mandatory fields.

At the bottom of the **Fields** pane, either click the **All** button to select all fields, regardless of measurement level, or click an individual measurement level button to select all fields with that measurement level.

### Target

Select one field as the target for the prediction.

**Note:** You can only select fields with a measurement level of continuous.

### Location

Select the location type to be used in the model.

**Note:** You can only select fields with a measurement level of geospatial.

### Location label

Shape data often includes a field that shows the names of the features in the layer, for example, this might be the names of states or counties. Use this field to associate a name, or label, with a location by selecting a categorical field to label the chosen **Location** field in your output.

### Time Field

Select the time fields to use in your predictions.

**Note:** You can select only fields with a measurement level of continuous and a storage type of time, date, timestamp, or integer.

### Predictors (Inputs)

Choose one or more fields as inputs for the prediction.

**Note:** You can select only fields with a measurement level of continuous.

## Spatio-Temporal Prediction - Time Intervals

In the Time Intervals pane, you can select the options to set the time interval and any required aggregation over time.

Data preparation is required to convert time fields into an index before you can build an STP model; for conversion to be possible the time field must have a constant interval between records. If your data does not already contain this information, use the options in this pane to set this interval before you can use the modeling node.

**Time interval** Select the interval that you want the data set to be converted to. Available options depend on the storage type of the field that is chosen as the **Time Field** for the model on the Fields tab.

- **Periods** Only available for integer time fields; this is a series of intervals, with a uniform interval between each measurement, that does not match any of the other available intervals.
- **Years** Only available for Date or Timestamp time fields.
- **Quarters** Only available for Date or Timestamp time fields. If you choose this option, you are prompted to select the **Start month** of the first quarter.
- **Months** Only available for Date or Timestamp time fields.
- **Weeks** Only available for Date or Timestamp time fields.
- **Days** Only available for Date or Timestamp time fields.
- **Hours** Only available for Time or Timestamp time fields.
- **Minutes** Only available for Time or Timestamp time fields.
- **Seconds** Only available for Time or Timestamp time fields.

When you select the **Time interval**, you are prompted to complete further fields. The fields available depend on both the time interval and the storage type. The fields that might be displayed are shown in the following list.

- **Number of days per week**
- **Number of hours in a day**
- **Week begins on** The first day of the week
- **Day begins at** The time at which you consider a new day to start.
- **Interval value** You can choose one of the following options: 1, 2, 3, 4, 5, 6, 10, 12, 15, 20, or 30.
- **Start month** The month on which the financial year starts.
- **Starting period** If you are using **Periods**, select the starting period.

**Data matches specified time interval settings** If your data already contains the correct time intervals information and does not need to be converted, select this check box. When you select this box, the fields in the **Aggregation** area are unavailable.

### Aggregation

Only available if you clear the **Data matches specified time interval settings** check box; specify the options for aggregating fields to match the specified interval. For example, if you have a mix of weekly and monthly data, you might aggregate, or "roll up", the weekly values to achieve a uniform monthly interval. Select the default settings to be used for aggregation of different field types and create any custom settings that you want for any specific fields.

- **Continuous** Set the default aggregation method to be applied to all continuous fields that are not individually specified. You can choose from several methods:
  - Sum
  - Mean
  - Minimum
  - Maximum

- Median
- First quartile
- Third quartile

**Custom settings for specified fields** To apply a specific aggregation function to individual fields, select them in this table and choose the aggregation method.

- **Field** Use the **Add field** button to display the **Select Fields** dialog box and choose the required fields. The chosen fields are displayed in this column.
- **Aggregation function** From the drop-down list, select the aggregation function to convert the field to the specified time interval.

## Spatio-Temporal Prediction - Basic Build Options

Use the settings in this dialog box to set the basic model building options.

### Model Settings

#### Include intercept

Including the intercept (the constant term in the model) can increase the overall accuracy of the solution. If you can assume the data passes through the origin, you can exclude the intercept.

#### Maximum autoregressive order

Autoregressive orders specify which previous values are used to predict current values. Use this option to specify the number of previous records that are used to calculate a new value. You can choose any integer between 1 and 5.

### Spatial Covariance

#### Estimation method

Select the estimation method to be used; you can choose either **Parametric** or **Nonparametric**. For the **Parametric** method you can choose from one of three **Model** types:

- **Gaussian**
- **Exponential**
- **Powered Exponential** If you select this option, you must also specify the **Power** level to be used. This level can be any value between 1 and 2, changed in 0.1 increments.

## Spatio-Temporal Prediction - Advanced Build Options

Users with detailed knowledge of STP can use the following options to fine-tune the model building process.

#### Maximum percentage of missing values

Specify the maximum percentage of records that contain missing values that can be included in the model.

#### Significance level for hypotheses testing in model build

Specify the significance level value to be used for all tests of STP model estimation, including two Goodness of Fit tests, Effect F-tests and Coefficient T-tests. This level can be any value from 0 to 1, changed in 0.01 increments.

## Spatio-Temporal Prediction - Output

Before you build the model, use the options on this pane to select the output that you want to include in the output viewer.

### Model Information

#### Model specifications

Select this option to include model specification information in the model output.

**Temporal information summary**

Select this option to include a summary of temporal information in the model output.

**Evaluation****Model quality**

Select this option to include model quality in the model output.

**Test of effects in mean structure model**

Select this option to include test of effects information in the model output.

**Interpretation****Mean structure model coefficients**

Select this option to include mean structure model coefficients information in the model output.

**Autoregressive coefficients**

Select this option to include autoregressive coefficients information in the model output.

**Tests of decay over space**

Select this option to include test of spatial covariance, or decay over space, information in the model output.

**Parametric spatial covariance model parameters plot**

Select this option to include parametric spatial covariance model parameter plot information in the model output.

**Note:** This option is only available if you selected the **Parametric** estimation method on the Basics tab.

**Correlations heat map**

Select this option to include a map of the target values in the model output.

**Note:** If there are more than 500 locations in your model, the map output will not be created.

**Correlations map**

Select this option to include a map of correlations in the model output.

**Note:** If there are more than 500 locations in your model, the map output will not be created.

**Location clusters**

Select this option to include location clustering output in the model output. Only output which does not require access to map data is included as part of the cluster output.

**Note:** This output can only be created for a non-parametric spatial covariance model.

If you choose this option, you can set the following:

- **Similarity threshold** Select the threshold at which output clusters are to be considered as similar enough to be merged into a single cluster.
- **Maximum number of clusters to display** Set the upper limit for the number of clusters that can be included in the model output.

**Spatio-Temporal Prediction - Model Options**

**Model Name** You can generate the model name automatically, based on the target fields, or specify a custom name. The automatically generated name is the target field name.

**Uncertainty factor (%)** Uncertainty factor is a percentage value that represents the growth in uncertainty when you forecast into the future. The upper and lower limit of forecast uncertainty increase by this percentage each step into the future. Set the uncertainty factor to be applied to your model outputs; this sets the upper and lower boundaries for the predicted values.



## Spatio-Temporal Prediction Model Nugget

The Spatio-Temporal Prediction (STP) model nugget displays details of the model in the Model tab of the Output Viewer. For more information on using the viewer, see the section titled "Working with Output" in the Modeler User's Guide (ModelerUsersGuide.pdf).

The spatio-temporal prediction (STP) modeling operation creates a number of new fields with the prefix \$STP- as shown in the following table.

Table 26. New fields created by the STP modeling operation	
Field name	Description
\$STP-<Time>	The time field created as part of the model build. The settings on the Time Intervals pane of the Build Options tab determine how this field is created.  <Time> is the original name of the field selected as the <b>Time Field</b> on the Fields tab.  <b>Note:</b> This field is only created if you converted the original <b>Time Field</b> as part of the model build.
\$STP-<Target>	This field contains the predictions for the target value.  <Target> is the name of the original <b>Target</b> field for the model
\$STPVAR-<Target>	This field contains the VarianceOfPointPrediction values.  <Target> is the name of the original <b>Target</b> field for the model
\$STPLCI-<Target>	This field contains the LowerOfPredictionInterval values; that is, the lower bound of confidence.  <Target> is the name of the original <b>Target</b> field for the model
\$STPUCI-<Target>	This field contains the UpperOfPredictionInterval values; that is, the upper bound of confidence.  <Target> is the name of the original <b>Target</b> field for the model

### Spatio-Temporal Prediction Model Settings

Use the Settings tab to control the level of uncertainty that you deem acceptable in the modeling operation.

**Uncertainty factor (%)** Uncertainty factor is a percentage value that represents the growth in uncertainty when you forecast into the future. The upper and lower limit of forecast uncertainty increase by this percentage each step into the future. Set the uncertainty factor to be applied to your model outputs; this sets the upper and lower boundaries for the predicted values.

## TCM Node

Use this node to create a temporal causal model (TCM).

### Temporal Causal Models

Temporal causal modeling attempts to discover key causal relationships in time series data. In temporal causal modeling, you specify a set of target series and a set of candidate inputs to those targets. The procedure then builds an autoregressive time series model for each target and includes only those inputs that have a causal relationship with the target. This approach differs from traditional time series modeling where you must explicitly specify the predictors for a target series. Since temporal causal modeling typically involves building models for multiple related time series, the result is referred to as a *model system*.

In the context of temporal causal modeling, the term *causal* refers to Granger causality. A time series X is said to "Granger cause" another time series Y if regressing for Y in terms of past values of both X and Y results in a better model for Y than regressing only on past values of Y.

**Note:** The Temporal causal modeling node does not support the Model Evaluation or Champion Challenger steps in IBM SPSS Collaboration and Deployment Services.

## Examples

Business decision makers can use temporal causal modeling to uncover causal relationships within a large set of time-based metrics that describe the business. The analysis might reveal a few controllable inputs, which have the largest impact on key performance indicators.

Managers of large IT systems can use temporal causal modeling to detect anomalies in a large set of interrelated operational metrics. The causal model then allows going beyond anomaly detection and discovering the most likely root causes of the anomalies.

## Field requirements

There must be at least one target. By default, fields with a predefined role of None are not used.

## Data structure

Temporal causal modeling supports two types of data structures.

### Column-based data

For column-based data, each time series field contains the data for a single time series. This structure is the traditional structure of time series data, as used by the Time Series Modeler.

### Multidimensional data

For multidimensional data, each time series field contains the data for multiple time series. Separate time series, within a particular field, are then identified by a set of values of categorical fields referred to as *dimension* fields. For example, sales data for two different sales channels (retail and web) might be stored in a single *sales* field. A dimension field that is named *channel*, with values 'retail' and 'web', identifies the records that are associated with each of the two sales channels.

**Note:** To build a temporal causal model, you need enough data points. The product uses the constraint:

$$m > (L + KL + 1)$$

where *m* is the number of data points, *L* is the number of lags, and *K* is the number of predictors. Make sure your data set is big enough so that the number of data points (*m*) satisfies the condition.

## Time Series to Model

On the Fields tab, use the **Time Series** settings to specify the series to include in the model system.

Select the option for the data structure that applies to your data. For multidimensional data, click **Select Dimensions** to specify the dimension fields. The specified order of the dimension fields defines the order in which they appear in all subsequent dialogs and output. Use the up and down arrow buttons on the Select Dimensions subdialog to reorder the dimension fields.

For column-based data, the term *series* has the same meaning as the term *field*. For multidimensional data, fields that contain time series are referred to as *metric* fields. A time series, for multidimensional data, is defined by a metric field and a value for each of the dimension fields. The following considerations apply to both column-based and multidimensional data.

- Series that are specified as candidate inputs or as both target and input are considered for inclusion in the model for each target. The model for each target always includes lagged values of the target itself.
- Series that are specified as forced inputs are always included in the model for each target.
- At least one series must be specified as either a target or as both target and input.
- When **Use predefined roles** is selected, fields that have a role of Input are set as candidate inputs. No predefined role maps to a forced input.

## Multidimensional data

For multidimensional data, you specify metric fields and associated roles in a grid, where each row in the grid specifies a single metric and role. By default, the model system includes series for all combinations of the dimension fields for each row in the grid. For example, if there are dimensions for *region* and *brand* then, by default, specifying the metric *sales* as a target means that there is a separate sales target series for each combination of *region* and *brand*.

For each row in the grid, you can customize the set of values for any of the dimension fields by clicking the ellipsis button for a dimension. This action opens the Select Dimension Values subdialog. You can also add, delete, or copy grid rows.

The **Series Count** column displays the number of sets of dimension values that are currently specified for the associated metric. The displayed value can be larger than the actual number of series (one series per set). This condition occurs when some of the specified combinations of dimension values do not correspond to series contained by the associated metric.

### Select Dimension Values

For multidimensional data, you can customize the analysis by specifying which dimension values apply to a particular metric field with a particular role. For example, if *sales* is a metric field and *channel* is a dimension with values 'retail' and 'web,' you can specify that 'web' sales is an input and 'retail' sales is a target. You can also specify dimension subsets that apply to all metric fields used in the analysis. For example, if *region* is a dimension field that indicates geographical region, then you can limit the analysis to particular regions.

#### All values

Specifies that all values of the current dimension field are included. This option is the default.

#### Select values to include or exclude

Use this option to specify the set of values for the current dimension field. When **Include** is selected for the **Mode**, only values that are specified in the **Selected values** list are included. When **Exclude** is selected for the **Mode**, all values other than the values that are specified in the **Selected values** list are included.

You can filter the set of values from which to choose. Values that meet the filter condition appear in the **Matched** tab and values that do not meet the filter condition appear in the **Unmatched** tab of the **Unselected values** list. The **All** tab lists all unselected values, regardless of any filter condition.

- You can use asterisks (\*) to indicate wildcard characters when you specify a filter.
- To clear the current filter, specify an empty value for the search term on the Filter Displayed Values dialog.

## Observations

On the Fields tab, use the **Observations** settings to specify the fields that define the observations.

### Observations that are defined by date/times

You can specify that the observations are defined by a date, time, or timestamp field. In addition to the field that defines the observations, select the appropriate time interval that describes the observations. Depending on the specified time interval, you can also specify other settings, such as the interval between observations (increment) or the number of days per week. The following considerations apply to the time interval:

- Use the value **Irregular** when the observations are irregularly spaced in time, such as the time at which a sales order is processed. When **Irregular** is selected, you must specify the time interval that is used for the analysis, from the **Time Interval** settings on the Data Specifications tab.
- When the observations represent a date and time and the time interval is hours, minutes, or seconds, then use **Hours per day**, **Minutes per day**, or **Seconds per day**. When the observations represent a time (duration) without reference to a date and the time interval is hours, minutes, or seconds, then use **Hours (non-periodic)**, **Minutes (non-periodic)**, or **Seconds (non-periodic)**.

- Based on the selected time interval, the procedure can detect missing observations. Detecting missing observations is necessary since the procedure assumes that all observations are equally spaced in time and that there are no missing observations. For example, if the time interval is Days and the date 2014-10-27 is followed by 2014-10-29, then there is a missing observation for 2014-10-28. Values are imputed for any missing observations. Settings for handling missing values can be specified from the Data Specifications tab.
- The specified time interval allows the procedure to detect multiple observations in the same time interval that need to be aggregated together and to align observations on an interval boundary, such as the first of the month, to ensure that the observations are equally spaced. For example, if the time interval is Months, then multiple dates in the same month are aggregated together. This type of aggregation is referred to as *grouping*. By default, observations are summed when grouped. You can specify a different method for grouping, such as the mean of the observations, from the **Aggregation and Distribution** settings on the Data Specifications tab.
- For some time intervals, the additional settings can define breaks in the normal equally spaced intervals. For example, if the time interval is Days, but only weekdays are valid, you can specify that there are five days in a week, and the week begins on Monday.

### Observations that are defined by periods or cyclic periods

Observations can be defined by one or more integer fields that represent periods or repeating cycles of periods, up to an arbitrary number of cycle levels. With this structure, you can describe series of observations that don't fit one of the standard time intervals. For example, a fiscal year with only 10 months can be described with a cycle field that represents years and a period field that represents months, where the length of one cycle is 10.

Fields that specify cyclic periods define a hierarchy of periodic levels, where the lowest level is defined by the **Period** field. The next highest level is specified by a cycle field whose level is 1, followed by a cycle field whose level is 2, and so on. Field values for each level, except the highest, must be periodic with respect to the next highest level. Values for the highest level cannot be periodic. For example, in the case of the 10-month fiscal year, months are periodic within years and years are not periodic.

- The length of a cycle at a particular level is the periodicity of the next lowest level. For the fiscal year example, there is only one cycle level and the cycle length is 10 since the next lowest level represents months and there are 10 months in the specified fiscal year.
- Specify the starting value for any periodic field that does not start from 1. This setting is necessary for detecting missing values. For example, if a periodic field starts from 2 but the starting value is specified as 1, then the procedure assumes that there is a missing value for the first period in each cycle of that field.

### Time Interval for Analysis

The time interval that is used for the analysis can differ from the time interval of the observations. For example, if the time interval of the observations is Days, you might choose Months for the time interval for analysis. The data are then aggregated from daily to monthly data before the model is built. You can also choose to distribute the data from a longer to a shorter time interval. For example, if the observations are quarterly then you can distribute the data from quarterly to monthly data.

The available choices for the time interval at which the analysis is done depend on how the observations are defined and the time interval of those observations. In particular, when the observations are defined by cyclic periods, then only aggregation is supported. In that case, the time interval of the analysis must be greater than or equal to the time interval of the observations.

The time interval for the analysis is specified from the **Time Interval** settings on the Data Specifications tab. The method by which the data are aggregated or distributed is specified from the **Aggregation and Distribution** settings on the Data Specifications tab.

## Aggregation and Distribution

### Aggregation functions

When the time interval that is used for the analysis is longer than the time interval of the observations, the input data are aggregated. For example, aggregation is done when the time interval of the observations is Days and the time interval for analysis is Months. The following aggregation functions are available: mean, sum, mode, min, or max.

### Distribution functions

When the time interval that is used for the analysis is shorter than the time interval of the observations, the input data are distributed. For example, distribution is done when the time interval of the observations is Quarters and the time interval for analysis is Months. The following distribution functions are available: mean or sum.

### Grouping functions

Grouping is applied when observations are defined by date/times and multiple observations occur in the same time interval. For example, if the time interval of the observations is Months, then multiple dates in the same month are grouped and associated with the month in which they occur. The following grouping functions are available: mean, sum, mode, min, or max. Grouping is always done when the observations are defined by date/times and the time interval of the observations is specified as Irregular.

**Note:** Although grouping is a form of aggregation, it is done before any handling of missing values whereas formal aggregation is done after any missing values are handled. When the time interval of the observations is specified as Irregular, aggregation is done only with the grouping function.

### Aggregate cross-day observations to previous day

Specifies whether observations with times that cross a day boundary are aggregated to the values for the previous day. For example, for hourly observations with an eight-hour day that starts at 20:00, this setting specifies whether observations between 00:00 and 04:00 are included in the aggregated results for the previous day. This setting applies only if the time interval of the observations is Hours per day, Minutes per day or Seconds per day and the time interval for analysis is Days.

### Custom settings for specified fields

You can specify aggregation, distribution, and grouping functions on a field by field basis. These settings override the default settings for the aggregation, distribution, and grouping functions.

## Missing Values

Missing values in the input data are replaced with an imputed value. The following replacement methods are available:

### Linear interpolation

Replaces missing values by using a linear interpolation. The last valid value before the missing value and the first valid value after the missing value are used for the interpolation. If the first or last observation in the series has a missing value, then the two nearest non-missing values at the beginning or end of the series are used.

### Series mean

Replaces missing values with the mean for the entire series.

### Mean of nearby points

Replaces missing values with the mean of valid surrounding values. The span of nearby points is the number of valid values before and after the missing value that are used to compute the mean.

### Median of nearby points

Replaces missing values with the median of valid surrounding values. The span of nearby points is the number of valid values before and after the missing value that are used to compute the median.

### Linear trend

This option uses all non-missing observations in the series to fit a simple linear regression model, which is then used to impute the missing values.

Other settings:

**Maximum percentage of missing values (%)**

Specifies the maximum percentage of missing values that are allowed for any series. Series with more missing values than the specified maximum are excluded from the analysis.

**General Data Options****Maximum number of distinct values per dimension field**

This setting applies to multidimensional data and specifies the maximum number of distinct values that are allowed for any one dimension field. By default, this limit is set to 10000 but it can be increased to an arbitrarily large number.

**General Build Options****Confidence interval width (%)**

This setting controls the confidence intervals for both forecasts and model parameters. You can specify any positive value less than 100. By default, a 95% confidence interval is used.

**Maximum number of inputs for each target**

This setting specifies the maximum number of inputs that are allowed in the model for each target. You can specify an integer in the range 1 - 20. The model for each target always includes lagged values of itself, so setting this value to 1 specifies that the only input is the target itself.

**Model tolerance**

This setting controls the iterative process that is used for determining the best set of inputs for each target. You can specify any value that is greater than zero. The default is 0.001. Model tolerance is a stop criterion for predictor selection. It can affect the number of predictors that are included in the final model. But if a target can predict itself very well, other predictors may not be included in the final model. Some trial and error may be required (for example, if you have this value set to high, you can try setting it to a smaller value to see if other predictors can be included or not).

**Outlier threshold (%)**

An observation is flagged as an outlier if the probability, as calculated from the model, that it is an outlier exceeds this threshold. You can specify a value in the range 50 - 100.

**Number of Lags for Each Input**

This setting specifies the number of lag terms for each input in the model for each target. By default, the number of lag terms is automatically determined from the time interval that is used for the analysis. For example, if the time interval is months (with an increment of one month) then the number of lags is 12. Optionally, you can explicitly specify the number of lags. The specified value must be an integer in the range 1 - 20.

**Continue estimation using existing models**

If you already generated a temporal causal model, select this option to reuse the criteria settings that are specified for that model, rather than building a new model. In this way, you can save time by reestimating and producing a new forecast that is based on the same model settings as before but using more recent data.

**Series to Display**

These options specify the series (targets or inputs) for which output is displayed. The content of the output for the specified series is determined by the **Output Options** settings.

**Display targets associated with best-fitting models**

By default, output is displayed for the targets that are associated with the 10 best-fitting models, as determined by the R square value. You can specify a different fixed number of best-fitting models or you can specify a percentage of best-fitting models. You can also choose from the following goodness of fit measures:

**R square**

Goodness-of-fit measure of a linear model, sometimes called the coefficient of determination. It is the proportion of variation in the target variable explained by the model. It ranges in value from 0 to 1. Small values indicate that the model does not fit the data well.

**Root mean square percentage error**

A measure of how much the model-predicted values differ from the observed values of the series. It is independent of the units that are used and can therefore be used to compare series with different units.

**Root mean square error**

The square root of mean square error. A measure of how much a dependent series varies from its model-predicted level, expressed in the same units as the dependent series.

**BIC**

Bayesian Information Criterion. A measure for selecting and comparing models based on the -2 reduced log likelihood. Smaller values indicate better models. The BIC also "penalizes" overparameterized models (complex models with a large number of inputs, for example), but more strictly than the AIC.

**AIC**

Akaike Information Criterion. A measure for selecting and comparing models based on the -2 reduced log likelihood. Smaller values indicate better models. The AIC "penalizes" overparameterized models (complex models with a large number of inputs, for example).

**Specify Individual Series**

You can specify individual series for which you want output.

- For column-based data, you specify the fields that contain the series that you want. The order of the specified fields defines the order in which they appear in the output.
- For multidimensional data, you specify a particular series by adding an entry to the grid for the metric field that contains the series. You then specify the values of the dimension fields that define the series.
  - You can enter the value for each dimension field directly into the grid or you can select from the list of available dimension values. To select from the list of available dimension values, click the ellipsis button in the cell for the dimension that you want. This action opens the Select Dimension Value subdialog.
  - You can search the list of dimension values, on the Select Dimension Value subdialog, by clicking the binoculars icon and specifying a search term. Spaces are treated as part of the search term. Asterisks (\*) in the search term do not indicate wildcard characters.
  - The order of the series in the grid defines the order in which they appear in the output.

For both column-based data and multidimensional data, output is limited to 30 series. This limit includes individual series (inputs or targets) that you specify and targets that are associated with best-fitting models. Individually specified series take precedence over targets that are associated with best-fitting models.

**Output Options**

These options specify the content of the output. Options in the **Output for targets** group generate output for the targets that are associated with the best-fitting models on the **Series to Display** settings. Options in the **Output for series** group generate output for the individual series that are specified on the **Series to Display** settings.

**Overall model system**

Displays a graphical representation of the causal relations between series in the model system. Tables of both model fit statistics and outliers for the displayed targets are included as part of the output item. When this option is selected in the **Output for series** group, a separate output item is created for each individual series that is specified on the **Series to Display** settings.

Causal relations between series have an associated significance level, where a smaller significance level indicates a more significant connection. You can choose to hide relations with a significance level that is greater than a specified value.

### Model fit statistics and outliers

Tables of model fit statistics and outliers for the target series that are selected for display. These tables contain the same information as the tables in the Overall Model System visualization. These tables support all standard features for pivoting and editing tables.

### Model effects and model parameters

Tables of model effects tests and model parameters for the target series that are selected for display. Model effects tests include the F statistic and associated significance value for each input included in the model.

### Impact diagram

Displays a graphical representation of the causal relations between a series of interest and other series that it affects or that affect it. Series that affect the series of interest are referred to as *causes*. Selecting **Effects** generates an impact diagram that is initialized to display effects. Selecting **Causes** generates an impact diagram that is initialized to display causes. Selecting **Both causes and effects** generates two separate impact diagrams, one that is initialized to causes and one that is initialized to effects. You can interactively toggle between causes and effects in the output item that displays the impact diagram.

You can specify the number of levels of causes or effects to display, where the first level is just the series of interest. Each additional level shows more indirect causes or effects of the series of interest. For example, the third level in the display of effects consists of the series that contain series in the second level as a direct input. Series in the third level are then indirectly affected by the series of interest since the series of interest is a direct input to the series in the second level.

### Series plot

Plots of observed and predicted values for the target series that are selected for display. When forecasts are requested, the plot also shows the forecasted values and the confidence intervals for the forecasts.

### Residuals plot

Plots of the model residuals for the target series that are selected for display.

### Top inputs

Plots of each displayed target, over time, along with the top 3 inputs for the target. The top inputs are the inputs with the lowest significance value. To accommodate different scales for the inputs and target, the y axis represents the z score for each series.

### Forecast table

Tables of forecasted values and confidence intervals of those forecasts for the target series that are selected for display.

### Outlier root cause analysis

Determines which series are most likely to be the cause of each outlier in a series of interest. Outlier root cause analysis is done for each target series that is included in the list of individual series on the **Series to Display** settings.

### Output

#### Interactive outliers table and chart

Table and chart of outliers and root causes of those outliers for each series of interest. The table contains a single row for each outlier. The chart is an impact diagram. Selecting a row in the table highlights the path, in the impact diagram, from the series of interest to the series that most likely causes the associated outlier.

#### Pivot table of outliers

Table of outliers and root causes of those outliers for each series of interest. This table contains the same information as the table in the interactive display. This table supports all standard features for pivoting and editing tables.

### Causal levels

You can specify the number of levels to include in the search for the root causes. The concept of levels that is used here is the same as described for impact diagrams.



## Model fit across all models

Histogram of model fit for all models and for selected fit statistics. The following fit statistics are available:

### R square

Goodness-of-fit measure of a linear model, sometimes called the coefficient of determination. It is the proportion of variation in the target variable explained by the model. It ranges in value from 0 to 1. Small values indicate that the model does not fit the data well.

### Root mean square percentage error

A measure of how much the model-predicted values differ from the observed values of the series. It is independent of the units that are used and can therefore be used to compare series with different units.

### Root mean square error

The square root of mean square error. A measure of how much a dependent series varies from its model-predicted level, expressed in the same units as the dependent series.

### BIC

Bayesian Information Criterion. A measure for selecting and comparing models based on the -2 reduced log likelihood. Smaller values indicate better models. The BIC also "penalizes" overparameterized models (complex models with a large number of inputs, for example), but more strictly than the AIC.

### AIC

Akaike Information Criterion. A measure for selecting and comparing models based on the -2 reduced log likelihood. Smaller values indicate better models. The AIC "penalizes" overparameterized models (complex models with a large number of inputs, for example).

## Outliers over time

Bar chart of the number of outliers, across all targets, for each time interval in the estimation period.

## Series transformations

Table of any transformations that were applied to the series in the model system. The possible transformations are missing value imputation, aggregation, and distribution.

## Estimation Period

By default, the estimation period starts at the time of the earliest observation and ends at the time of the latest observation across all series.

### By start and end times

You can specify both the start and end of the estimation period or you can specify just the start or just the end. If you omit the start or the end of the estimation period, the default value is used.

- If the observations are defined by a date/time field, then enter values for start and end in the same format that is used for the date/time field.
- For observations that are defined by cyclic periods, specify a value for each of the cyclic periods fields. Each field is displayed in a separate column.

### By latest or earliest time intervals

Defines the estimation period as a specified number of time intervals that start at the earliest time interval or end at the latest time interval in the data, with an optional offset. In this context, the time interval refers to the time interval of the analysis. For example, assume that the observations are monthly but the time interval of the analysis is quarters. Specifying **Latest** and a value of 24 for the **Number of time intervals** means the latest 24 quarters.

Optionally, you can exclude a specified number of time intervals. For example, specifying the latest 24 time intervals and 1 for the number to exclude means that the estimation period consists of the 24 intervals that precede the last one.

## Model Options

### Model Name

You can specify a custom name for the model or accept the automatically generated name, which is *TCM*.

### Forecast

The option to **Extend records into the future** sets the number of time intervals to forecast beyond the end of the estimation period. The time interval in this case is the time interval of the analysis, which is specified on the Data Specifications tab. When forecasts are requested, autoregressive models are automatically built for any input series that are not also targets. These models are then used to generate values for those input series in the forecast period. There is no maximum limit for this setting.

### Interactive Output

The output from temporal causal modeling includes a number of interactive output objects. Interactive features are available by activating (double-clicking) the object in the Output Viewer.

### Overall model system

Displays the causal relations between series in the model system. All lines that connect a particular target to its inputs have the same color. The thickness of the line indicates the significance of the causal connection, where thicker lines represent a more significant connection. Inputs that are not also targets are indicated with a black square.

- You can display relations for top models, a specified series, all series, or models with no inputs. The top models are the models that meet the criteria that were specified for best-fitting models on the **Series to Display** settings.
- You can generate impact diagrams for one or more series by selecting the series names in the chart, right-clicking, and then choosing **Create Impact Diagram** from the context menu.
- You can choose to hide causal relations that have a significance level that is greater than a specified value. Smaller significance levels indicate a more significant causal relation.
- You can display relations for a particular series by selecting the series name in the chart, right-clicking, and then choosing **Highlight relations for series** from the context menu.

### Impact diagram

Displays a graphical representation of the causal relations between a series of interest and other series that it affects or that affect it. Series that affect the series of interest are referred to as *causes*.

- You can change the series of interest by specifying the name of the series that you want. Double-clicking any node in the impact diagram changes the series of interest to the series associated with that node.
- You can toggle the display between causes and effects and you can change the number of levels of causes or effects to display.
- Single-clicking any node opens a detailed sequence diagram for the series that is associated with the node.

### Outlier root cause analysis

Determines which series are most likely to be the cause of each outlier in a series of interest.

- You can display the root cause for any outlier by selecting the row for the outlier in the Outliers table. You can also display the root cause by clicking the icon for the outlier in the sequence chart.
- Single-clicking any node opens a detailed sequence diagram for the series that is associated with the node.

### Overall model quality

Histogram of model fit for all models, for a particular fit statistic. Clicking a bar in the bar chart filters the dot plot so that it displays only the models that are associated with the selected bar. You can find the model for a particular target series in the dot plot by specifying the series name.

### Outlier distribution

Bar chart of the number of outliers, across all targets, for each time interval in the estimation period. Clicking a bar in the bar chart filters the dot plot so that it displays only the outliers that are associated with the selected bar.

## TCM Model Nugget

The TCM modeling operation creates a number of new fields with the prefix \$TCM- as shown in the following table.

Table 27. New fields created by the TCM modeling operation	
Field name	Description
\$TCM- <i>colname</i>	The value forecasted by the model for each target series.
\$TCMLCI- <i>colname</i>	The lower confidence intervals for each forecasted series.
\$TSUCI- <i>colname</i>	The upper confidence intervals for each forecasted series.
\$TCMResidual- <i>colname</i>	The noise residual value for each column of the generated model data.

### TCM Model Nugget Settings

The Settings tab provides additional options for the TCM model nugget.

#### Forecast

The option to **Extend records into the future** sets the number of time intervals to forecast beyond the end of the estimation period. The time interval in this case is the time interval of the analysis, which is specified on the Data Specifications tab of the TCM node. When forecasts are requested, autoregressive models are automatically built for any input series that are not also targets. These models are then used to generate values for those input series in the forecast period.

#### Make available for scoring

**Create new fields for each model to be scored.** Enables you to specify the new fields to create for each model to be scored.

- **Noises Residuals.** If selected, creates a new field (with the default prefix \$TCM-) for the model residuals for each target field, together with a total of these values.
- **Upper and lower confidence limits.** If selected, creates new fields (with the default prefix \$TCM-) for the lower and upper confidence intervals, respectively, for each target field, together with totals of these values.

**Targets included for scoring.** Select available targets to include in the model score.

## Temporal Causal Model Scenarios

The Temporal Causal Model Scenarios procedure runs user-defined scenarios for a temporal causal model system, with the data from the active dataset. A *scenario* is defined by a time series, that is referred to as the *root series*, and a set of user-defined values for that series over a specified time range. The specified values are then used to generate predictions for the time series that are affected by the root series. The procedure requires a model system file that was created by the Temporal Causal Modeling procedure. It is assumed that the active dataset is the same data as was used to create the model system file.

### Example

Using the Temporal Causal Modeling procedure, a business decision maker discovered a key metric that affects a number of important performance indicators. The metric is controllable, so the decision maker wants to investigate the effect of various sets of values for the metric over the next quarter. The

investigation is easily done by loading the model system file into the Temporal Causal Model Scenarios procedure and specifying the sets of values for the key metric.

### Defining the Scenario Period

The scenario period is the period over which you specify the values that are used to run your scenarios. It can start before or after the end of the estimation period. You can optionally specify to predict beyond the end of the scenario period. By default, predictions are generated through the end of the scenario period. All scenarios use the same scenario period and specifications for how far to predict.

**Note:** Predictions start at the first time period after the beginning of the scenario period. For example, if the scenario period starts on 2014-11-01 and the time interval is months, then the first prediction is for 2014-12-01.

### Specify by start, end and predict through times

- If the observations are defined by a date/time field, then enter values for start, end, and predict through in the same format that is used for the date/time field. Values for date/time fields are aligned to the beginning of the associated time interval. For example, if the time interval of the analysis is months, then the value 10/10/2014 is adjusted to 10/01/2014, which is the beginning of the month.
- For observations that are defined by cyclic periods, specify a value for each of the cyclic periods fields. Each field is displayed in a separate column.

### Specify by time intervals relative to end of estimation period

Defines start and end in terms of the number of time intervals relative to the end of the estimation period, where the time interval is the time interval of the analysis. The end of the estimation period is defined as time interval 0. Time intervals before the end of the estimation period have negative values and intervals after the end of the estimation period have positive values. You can also specify how many intervals to predict beyond the end of the scenario period. The default is 0.

For example, assume that the time interval of the analysis is months and that you specify 1 for starting interval, 3 for ending interval, and 1 for how far to predict beyond that. The scenario period is then the 3 months that follow the end of the estimation period. Predictions are generated for the second and third month of the scenario period and for 1 more month past the end of the scenario period.

### Adding Scenarios and Scenario Groups

The Scenarios tab specifies the scenarios that are to be run. To define scenarios, you must first define the scenario period by clicking **Define Scenario Period**. Scenarios and scenario groups (applies only to multidimensional data) are created by clicking the associated **Add Scenario** or **Add Scenario Group** button. By selecting a particular scenario or scenario group in the associated grid, you can edit it, make a copy of it, or delete it.

### Column-based data

The **Root field** column in the grid specifies the time series field whose values are replaced with the scenario values. The **Scenario values** column displays the specified scenario values in the order of earliest to latest. If the scenario values are defined by an expression, then the column displays the expression.

### Multidimensional data

#### Individual Scenarios

Each row in the Individual Scenarios grid specifies a time series whose values are replaced by the specified scenario values. The series is defined by the combination of the field that is specified in the **Root Metric** column and the specified value for each of the dimension fields. The content of the **Scenario values** column is the same as for column-based data.

## Scenario Groups

A *scenario group* defines a set of scenarios that are based on a single root metric field and multiple sets of dimension values. Each set of dimension values (one value per dimension field), for the specified metric field, defines a time series. An individual scenario is then generated for each such time series, whose values are then replaced by the scenario values. Scenario values for a scenario group are specified by an expression, which is then applied to each time series in the group.

The **Series Count** column displays the number of sets of dimension values that are associated with a scenario group. The displayed value can be larger than the actual number of time series that are associated with the scenario group (one series per set). This condition occurs when some of the specified combinations of dimension values do not correspond to series contained by the root metric for the group.

As an example of a scenario group, consider a metric field *advertising* and two dimension fields *region* and *brand*. You can define a scenario group that is based on *advertising* as the root metric and that includes all combinations of *region* and *brand*. You might then specify *advertising*\*1.2 as the expression to investigate the effect of increasing *advertising* by 20 percent for each of the time series that are associated with the *advertising* field. If there are 4 values of *region* and 2 values of *brand*, then there are 8 such time series and thus 8 scenarios defined by the group.

## Scenario Definition

Settings for defining a scenario depend on whether your data are column-based or multidimensional.

### Root series

Specifies the root series for the scenario. Each scenario is based on a single root series. For column-based data, you select the field that defines the root series. For multidimensional data, you specify the root series by adding an entry to the grid for the metric field that contains the series. You then specify the values of the dimension fields that define the root series. The following apply to specifying the dimension values:

- You can enter the value for each dimension field directly into the grid or you can select from the list of available dimension values. To select from the list of available dimension values, click the ellipsis button in the cell for the dimension that you want. This action opens the Select Dimension Value subdialog.
- You can search the list of dimension values, on the Select Dimension Value subdialog, by clicking the binoculars icon and specifying a search term. Spaces are treated as part of the search term. Asterisks (\*) in the search term do not indicate wildcard characters.

### Specify affected targets

Use this option when you know specific targets that are affected by the root series and you want to investigate the effects on those targets only. By default, targets that are affected by the root series are automatically determined. You can specify the breadth of the series that are affected by the scenario with settings on the Options tab.

For column-based data, select the targets that you want. For multidimensional data, you specify target series by adding an entry to the grid for the target metric field that contains the series. By default, all series that are contained in the specified metric field are included. You can customize the set of included series by customizing the included values for one or more of the dimension fields. To customize the dimension values that are included, click the ellipsis button for the dimension that you want. This action opens the Select Dimension Values dialog.

The **Series Count** column (for multidimensional data) displays the number of sets of dimension values that are currently specified for the associated target metric. The displayed value can be larger than the actual number of affected target series (one series per set). This condition occurs when some of the specified combinations of dimension values do not correspond to series contained by the associated target metric.

### Scenario ID

Each scenario must have a unique identifier. The identifier is displayed in output that is associated with the scenario. There are no restrictions, other than uniqueness, on the value of the identifier.

### Specify scenario values for root series

Use this option to specify explicit values for the root series in the scenario period. You must specify a numeric value for each time interval that is listed in the grid. You can obtain the values of the root series (actual or forecasted) for each interval in the scenario period by clicking **Read**, **Forecast**, or **Read\Forecast**.

### Specify expression for scenario values for root series

You can define an expression for computing the values of the root series in the scenario period. You can enter the expression directly or click the calculator button and create the expression from the Scenario Values Expression Builder.

- The expression can contain any target or input in the model system.
- When the scenario period extends beyond the existing data, the expression is applied to forecasted values of the fields in the expression.
- For multidimensional data, each field in the expression specifies a time series that is defined by the field and the dimension values that were specified for the root metric. It is those time series that are used to evaluate the expression.

As an example, assume that the root field is *advertising* and the expression is *advertising\*1.2*. The values for *advertising* that are used in the scenario represent a 20 percent increase over the existing values.

**Note:** Scenarios are created by clicking **Add Scenario** on the Scenarios tab.

### Select Dimension Values

For multidimensional data, you can customize the dimension values that define the targets that are affected by a scenario or scenario group. You can also customize the dimension values that define the set of root series for a scenario group.

### All values

Specifies that all values of the current dimension field are included. This option is the default.

### Select values

Use this option to specify the set of values for the current dimension field. You can filter the set of values from which to choose. Values that meet the filter condition appear in the **Matched** tab and values that do not meet the filter condition appear in the **Unmatched** tab of the **Unselected values** list. The **All** tab lists all unselected values, regardless of any filter condition.

- You can use asterisks (\*) to indicate wildcard characters when you specify a filter.
- To clear the current filter, specify an empty value for the search term on the Filter Displayed Values dialog.

To customize dimension values for affected targets:

1. From the Scenario Definition or Scenario Group Definition dialog, select the target metric for which you want to customize dimension values.
2. Click the ellipsis button in the column for the dimension that you want to customize.

To customize dimension values for the root series of a scenario group:

1. From the Scenario Group Definition dialog, click the ellipsis button (in the root series grid) for the dimension that you want to customize.

## Scenario Group Definition

### Root Series

Specifies the set of root series for the scenario group. An individual scenario is generated for each time series in the set. You specify the root series by adding an entry to the grid for the metric field that contains the series that you want. You then specify the values of the dimension fields that define the set. By default, all series that are contained in the specified root metric field are included. You can customize the set of included series by customizing the included values for one or more of the

dimension fields. To customize the dimension values that are included, click the ellipsis button for a dimension. This action opens the Select Dimension Values dialog.

The **Series Count** column displays the number of sets of dimension values that are currently included for the associated root metric. The displayed value can be larger than the actual number of root series for the scenario group (one series per set). This condition occurs when some of the specified combinations of dimension values do not correspond to series contained by the root metric.

### Specify affected target series

Use this option when you know specific targets that are affected by the set of root series and you want to investigate the effects on those targets only. By default, targets that are affected by each root series are automatically determined. You can specify the breadth of the series that are affected by each individual scenario with settings on the Options tab.

You specify target series by adding an entry to the grid for the metric field that contains the series. By default, all series that are contained in the specified metric field are included. You can customize the set of included series by customizing the included values for one or more of the dimension fields. To customize the dimension values that are included, click the ellipsis button for the dimension that you want. This action opens the Select Dimension Values dialog.

The **Series Count** column displays the number of sets of dimension values that are currently specified for the associated target metric. The displayed value can be larger than the actual number of affected target series (one series per set). This condition occurs when some of the specified combinations of dimension values do not correspond to series contained by the associated target metric.

### Scenario ID prefix

Each scenario group must have a unique prefix. The prefix is used to construct an identifier that is displayed in output that is associated with each individual scenario in the scenario group. The identifier for an individual scenario is the prefix, followed by an underscore, followed by the value of each dimension field that identifies the root series. The dimension values are separated by underscores. There are no restrictions, other than uniqueness, on the value of the prefix.

### Expression for scenario values for root series

Scenario values for a scenario group are specified by an expression, which is then used to compute the values for each of the root series in the group. You can enter an expression directly or click the calculator button and create the expression from the Scenario Values Expression Builder.

- The expression can contain any target or input in the model system.
- When the scenario period extends beyond the existing data, the expression is applied to forecasted values of the fields in the expression.
- For each root series in the group, the fields in the expression specify time series that are defined by those fields and the dimension values that define the root series. It is those time series that are used to evaluate the expression. For example, if a root series is defined by `region='north'` and `brand='X'`, then the time series that are used in the expression are defined by those same dimension values.

As an example, assume that the root metric field is *advertising* and that there are two dimension fields *region* and *brand*. Also, assume that the scenario group includes all combinations of the dimension field values. You might then specify `advertising*1.2` as the expression to investigate the effect of increasing *advertising* by 20 percent for each of the time series that are associated with the *advertising* field.

**Note:** Scenario groups apply to multidimensional data only and are created by clicking **Add Scenario Group** on the Scenarios tab.

### Options

#### Maximum level for affected targets

Specifies the maximum number of levels of affected targets. Each successive level, up to the maximum of 5, includes targets that are more indirectly affected by the root series. Specifically, the first level includes targets that have the root series as a direct input. Targets in the second level have

targets in the first level as a direct input, and so on. Increasing the value of this setting increases the complexity of the computation and might affect performance.

#### **Maximum auto-detected targets**

Specifies the maximum number of affected targets that are automatically detected for each root series. Increasing the value of this setting increases the complexity of the computation and might affect performance.

#### **Impact diagram**

Displays a graphical representation of the causal relations between the root series for each scenario and the target series that it affects. Tables of both the scenario values and the predicted values for the affected targets are included as part of the output item. The graph includes plots of the predicted values of the affected targets. Single-clicking any node in the impact diagram opens a detailed sequence diagram for the series that is associated with the node. A separate impact diagram is generated for each scenario.

#### **Series plots**

Generates series plots of the predicted values for each of the affected targets in each scenario.

#### **Forecast and scenario tables**

Tables of predicted values and scenario values for each scenario. These tables contains the same information as the tables in the impact diagram. These tables support all standard features for pivoting and editing tables.

#### **Include confidence intervals in plots and tables**

Specifies whether confidence intervals for scenario predictions are included in both chart and table output.

#### **Confidence interval width (%)**

This setting controls the confidence intervals for the scenario predictions. You can specify any positive value less than 100. By default, a 95% confidence interval is used.

## **Time Series node**

---

The Time Series node can be used with data in either a local or distributed environment; in a distributed environment you can harness the power of IBM SPSS Analytic Server. With this node, you can choose to estimate and build exponential smoothing, univariate Autoregressive Integrated Moving Average (ARIMA), or multivariate ARIMA (or transfer function) models for time series, and produce forecasts based on the time series data.

**Exponential smoothing** is a method of forecasting that uses weighted values of previous series observations to predict future values. As such, exponential smoothing is not based on a theoretical understanding of the data. It forecasts one point at a time, adjusting its forecasts as new data come in. The technique is useful for forecasting series that exhibit trend, seasonality, or both. You can choose from various exponential smoothing models that differ in their treatment of trend and seasonality.

**ARIMA** models provide more sophisticated methods for modeling trend and seasonal components than do exponential smoothing models, and, in particular, they allow the added benefit of including independent (predictor) variables in the model. This involves explicitly specifying autoregressive and moving average orders as well as the degree of differencing. You can include predictor variables and define transfer functions for any or all of them, as well as specify automatic detection of outliers or an explicit set of outliers.

**Note:** In practical terms, ARIMA models are most useful if you want to include predictors that might help to explain the behavior of the series that is being forecast, such as the number of catalogs that are mailed or the number of hits to a company web page. Exponential smoothing models describe the behavior of the time series without attempting to understand why it behaves as it does. For example, a series that historically peaks every 12 months will probably continue to do so even if you don't know why.

An **Expert Modeler** option is also available, which attempts to automatically identify and estimate the best-fitting ARIMA or exponential smoothing model for one or more target variables, thus eliminating the need to identify an appropriate model through trial and error. If in doubt, use the Expert Modeler option.



If predictor variables are specified, the Expert Modeler selects those variables that have a statistically significant relationship with the dependent series for inclusion in ARIMA models. Model variables are transformed where appropriate using differencing and/or a square root or natural log transformation. By default, the Expert Modeler considers all exponential smoothing models and all ARIMA models and picks the best model among them for each target field. You can, however, limit the Expert Modeler only to pick the best of the exponential smoothing models or only to pick the best of the ARIMA models. You can also specify automatic detection of outliers.

## Time Series node - field options

On the Fields tab, you choose whether you want to use the field role settings already defined in upstream nodes, or make the field assignments manually.

**Use predefined roles** This option uses the role settings (targets, predictors and so on) from an upstream Type node (or the Types tab of an upstream source node).

**Use custom field assignments.** To manually assign targets, predictors and other roles, select this option.

**Fields.** Use the arrow buttons to assign items manually from this list to the various role fields on the right of the screen. The icons indicate the valid measurement levels for each role field.

To select all the fields in the list, click the **All** button, or click an individual measurement level button to select all fields with that measurement level.

**Targets.** Select one or more of the fields as your target for the prediction.

**Candidate inputs.** Select one or more fields as inputs for the prediction. You can select only fields with a measurement level of continuous.

**Events and Interventions.** Use this area to designate certain input fields as event or intervention fields. This designation identifies a field as containing time series data that can be affected by events (predictable recurring situations; for example, sales promotions) or interventions (one-time incidents; for example, power outage or employee strike). The fields you select must be flags that have integer storage.

## Time Series node - data specification options

The Data Specifications tab is where you set all the options for the data to be included in your model. As long as you specify both a **Date/time field** and **Time interval**, you can click the **Run** button to build a model with all the default options, but normally you will want to customize the build for your own purposes.

The tab contains several different panes on which you set the customizations that are specific to your model.

### Time Series node - observations

Use the settings in this pane to specify the fields that define the observations.

#### Observations that are specified by a date/time field

You can specify that the observations are defined by a date, time, or timestamp field. In addition to the field that defines the observations, select the appropriate time interval that describes the observations. Depending on the specified time interval, you can also specify other settings, such as the interval between observations (increment) or the number of days per week. The following considerations apply to the time interval:

- Use the value **Irregular** when the observations are irregularly spaced in time, such as the time at which a sales order is processed. When **Irregular** is selected, you must specify the time interval that is used for the analysis, from the **Time Interval** settings on the Data Specifications tab.
- When the observations represent a date and time and the time interval is hours, minutes, or seconds, then use **Hours per day**, **Minutes per day**, or **Seconds per day**. When the observations represent a time (duration) without reference to a date and the time interval is hours, minutes, or seconds, then use **Hours (non-periodic)**, **Minutes (non-periodic)**, or **Seconds (non-periodic)**.

- Based on the selected time interval, the procedure can detect missing observations. Detecting missing observations is necessary since the procedure assumes that all observations are equally spaced in time and that no observations are missing. For example, if the time interval is Days and the date 2015-10-27 is followed by 2015-10-29, then an observation is missing for 2015-10-28. Values are imputed for any missing observations; use the **Missing Value Handling** area of the Data Specifications tab to specify settings for handling missing values.
- The specified time interval allows the procedure to detect multiple observations in the same time interval that need to be aggregated together and to align observations on an interval boundary, such as the first of the month, to ensure that the observations are equally spaced. For example, if the time interval is Months, then multiple dates in the same month are aggregated together. This type of aggregation is referred to as *grouping*. By default, observations are summed when grouped. You can specify a different method for grouping, such as the mean of the observations, from the **Aggregation and Distribution** settings on the Data Specifications tab.
- For some time intervals, the additional settings can define breaks in the normal equally spaced intervals. For example, if the time interval is Days, but only weekdays are valid, you can specify that there are five days in a week, and the week begins on Monday.

### Observations that are defined as periods or cyclic periods

Observations can be defined by one or more integer fields that represent periods or repeating cycles of periods, up to an arbitrary number of cycle levels. With this structure, you can describe series of observations that don't fit one of the standard time intervals. For example, a fiscal year with only 10 months can be described with a cycle field that represents years and a period field that represents months, where the length of one cycle is 10.

Fields that specify cyclic periods define a hierarchy of periodic levels, where the lowest level is defined by the **Period** field. The next highest level is specified by a cycle field whose level is 1, followed by a cycle field whose level is 2, and so on. Field values for each level, except the highest, must be periodic with respect to the next highest level. Values for the highest level cannot be periodic. For example, in the case of the 10-month fiscal year, months are periodic within years and years are not periodic.

- The length of a cycle at a particular level is the periodicity of the next lowest level. For the fiscal year example, there is only one cycle level and the cycle length is 10 since the next lowest level represents months and there are 10 months in the specified fiscal year.
- Specify the starting value for any periodic field that does not start from 1. This setting is necessary for detecting missing values. For example, if a periodic field starts from 2 but the starting value is specified as 1, then the procedure assumes that there is a missing value for the first period in each cycle of that field.

### Time Series node - time interval for analysis

The time interval that you use for analysis can differ from the time interval of the observations. For example, if the time interval of the observations is Days, you might choose Months for the time interval for analysis. The data is then aggregated from daily to monthly data before the model is built. You can also choose to distribute the data from a longer to a shorter time interval. For example, if the observations are quarterly then you can distribute the data from quarterly to monthly data.

Use the settings in this pane to specify the time interval for the analysis. The method by which the data is aggregated or distributed is specified from the **Aggregation and Distribution** settings on the Data Specifications tab.

The available choices for the time interval at which the analysis is done depend on how the observations are defined and the time interval of those observations. In particular, when the observations are defined by cyclic periods, only aggregation is supported. In that case, the time interval of the analysis must be greater than or equal to the time interval of the observations.

### Time Series node - aggregation and distribution options

Use the settings in this pane to specify settings for aggregating or distributing the input data with respect to the time intervals of the observations.

### **Aggregation functions**

When the time interval that is used for the analysis is longer than the time interval of the observations, the input data are aggregated. For example, aggregation is done when the time interval of the observations is Days and the time interval for analysis is Months. The following aggregation functions are available: mean, sum, mode, min, or max.

### **Distribution functions**

When the time interval that is used for the analysis is shorter than the time interval of the observations, the input data are distributed. For example, distribution is done when the time interval of the observations is Quarters and the time interval for analysis is Months. The following distribution functions are available: mean or sum.

### **Grouping functions**

Grouping is applied when observations are defined by date/times and multiple observations occur in the same time interval. For example, if the time interval of the observations is Months, then multiple dates in the same month are grouped and associated with the month in which they occur. The following grouping functions are available: mean, sum, mode, min, or max. Grouping is always done when the observations are defined by date/times and the time interval of the observations is specified as Irregular.

**Note:** Although grouping is a form of aggregation, it is done before any handling of missing values whereas formal aggregation is done after any missing values are handled. When the time interval of the observations is specified as Irregular, aggregation is done only with the grouping function.

### **Aggregate cross-day observations to previous day**

Specifies whether observations with times that cross a day boundary are aggregated to the values for the previous day. For example, for hourly observations with an eight-hour day that starts at 20:00, this setting specifies whether observations between 00:00 and 04:00 are included in the aggregated results for the previous day. This setting applies only if the time interval of the observations is Hours per day, Minutes per day or Seconds per day and the time interval for analysis is Days.

### **Custom settings for specified fields**

You can specify aggregation, distribution, and grouping functions on a field by field basis. These settings override the default settings for the aggregation, distribution, and grouping functions.

### **Time Series node - missing value options**

Use the settings in this pane to specify how any missing values in the input data are to be replaced with an imputed value. The following replacement methods are available:

#### **Linear interpolation**

Replaces missing values by using a linear interpolation. The last valid value before the missing value and the first valid value after the missing value are used for the interpolation. If the first or last observation in the series has a missing value, then the two nearest non-missing values at the beginning or end of the series are used.

Replaces missing values by using a linear interpolation.

- For non-seasonal data, the last valid value before the missing value and the first valid value after the missing value are used for the interpolation. If missing values are at the beginning or the end of a time series, then a linear extrapolation method is used based on the two nearest valid values.
- For seasonal data, a missing value is linearly interpolated using the last valid value of the same period before the missing value and the first valid value of the same period after the missing value. If one of the two values of the same period can't be found for the missing value, then the data will be regarded as non-seasonal data and linear interpolation of non-seasonal data is used to impute the missing value.

#### **Series mean**

Replaces missing values with the mean for the entire series.

#### **Mean of nearby points**

Replaces missing values with the mean of valid surrounding values. The span of nearby points is the number of valid values before and after the missing value that are used to compute the mean.

### Median of nearby points

Replaces missing values with the median of valid surrounding values. The span of nearby points is the number of valid values before and after the missing value that are used to compute the median.

### Linear trend

This option uses all non-missing observations in the series to fit a simple linear regression model, which is then used to impute the missing values.

Other settings:

### Lowest data quality score (%)

Computes data quality measures for the time variable and for input data corresponding to each time series. If the data quality score is lower than this threshold, the corresponding time series will be discarded.

## Time Series node - estimation period

In the Estimation Period pane, you can specify the range of records to be used in model estimation. By default, the estimation period starts at the time of the earliest observation and ends at the time of the latest observation across all series.

### By start and end times

You can specify both the start and end of the estimation period or you can specify just the start or just the end. If you omit the start or the end of the estimation period, the default value is used.

- If the observations are defined by a date/time field, enter values for start and end in the same format that is used for the date/time field.
- For observations that are defined by cyclic periods, specify a value for each of the cyclic periods fields. Each field is displayed in a separate column.

### By latest or earliest time intervals

Defines the estimation period as a specified number of time intervals that start at the earliest time interval or end at the latest time interval in the data, with an optional offset. In this context, the time interval refers to the time interval of the analysis. For example, assume that the observations are monthly but the time interval of the analysis is quarters. Specifying **Latest** and a value of 24 for the **Number of time intervals** means the latest 24 quarters.

Optionally, you can exclude a specified number of time intervals. For example, specifying the latest 24 time intervals and 1 for the number to exclude means that the estimation period consists of the 24 intervals that precede the last one.

## Time Series node - build options

The Build Options tab is where you set all the options for building your model. You can, of course, just click the **Run** button to build a model with all the default options, but normally you will want to customize the build for your own purposes.

The tab contains two different panes on which you set the customizations that are specific to your model.

### Time Series node - general build options

The options available on this pane depend on which of the following three settings you choose from the **Method** list:

- **Expert Modeler.** Choose this option to use the Expert Modeler, which automatically finds the best-fitting model for each dependent series.
- **Exponential Smoothing.** Use this option to specify a custom exponential smoothing model.
- **ARIMA.** Use this option to specify a custom ARIMA model.

### Expert Modeler

Under **Model Type**, select the type of models you want to build:

- **All models.** The Expert Modeler considers both ARIMA and exponential smoothing models.
- **Exponential smoothing models only.** The Expert Modeler considers only exponential smoothing models.
- **ARIMA models only.** The Expert Modeler considers only ARIMA models.

**Expert Modeler considers seasonal models.** This option is only enabled if a periodicity is defined for the active dataset. When this option is selected, the Expert Modeler considers both seasonal and nonseasonal models. If this option is not selected, the Expert Modeler considers only nonseasonal models.

**Expert Modeler considers sophisticated exponential smoothing models.** When this option is selected, the Expert Modeler searches a total of 13 exponential smoothing models (7 of them existed in the original Time Series node, and 6 of them were added in version 18.1). If this option is not selected, the Expert Modeler only searches the original 7 exponential smoothing models.

Under **Outliers**, select from the following options

**Detect outliers automatically.** By default, automatic detection of outliers is not performed. Select this option to perform automatic detection of outliers, then select the desired outlier types.

Input fields must have a measurement level of *Flag*, *Nominal*, or *Ordinal* and must be numeric (for example, 1/0, not True/False, for a flag field), before they are included in this list.

The Expert Modeler considers only simple regression and not arbitrary transfer functions for inputs that are identified as event or intervention fields on the **Fields** tab.

## Exponential Smoothing

**Model Type.** Exponential smoothing models are classified as either seasonal or nonseasonal.<sup>1</sup> Seasonal models are only available if the periodicity defined by using the Time Intervals pane on the Data Specifications tab is seasonal. The seasonal periodicities are: cyclic periods, years, quarters, months, days per week, hours per day, minutes per day, and seconds per day. The following model types are available:

- **Simple.** This model is appropriate for a series in which there is no trend or seasonality. Its only relevant smoothing parameter is level. Simple exponential smoothing is most similar to an ARIMA with zero orders of autoregression, one order of differencing, one order of moving average, and no constant.
- **Holt's linear trend.** This model is appropriate for a series in which there is a linear trend and no seasonality. Its relevant smoothing parameters are level and trend, and, in this model, they are not constrained by each other's values. Holt's model is more general than Brown's model but may take longer to compute estimates for large series. Holt's exponential smoothing is most similar to an ARIMA with zero orders of autoregression, two orders of differencing, and two orders of moving average.
- **Damped trend.** This model is appropriate for a series in which there is a linear trend that is dying out and no seasonality. Its relevant smoothing parameters are level, trend, and damping trend. Damped exponential smoothing is most similar to an ARIMA with one order of autoregression, one order of differencing, and two orders of moving average.
- **Multiplicative trend.** This model is appropriate for a series in which there is a trend that changes with the magnitude of the series and no seasonality. Its relevant smoothing parameters are level and trend. Multiplicative trend exponential smoothing is not similar to any ARIMA model.
- **Brown's linear trend.** This model is appropriate for a series in which there is a linear trend and no seasonality. Its relevant smoothing parameters are level and trend, but, in this model, they are assumed to be equal. Brown's model is therefore a special case of Holt's model. Brown's exponential smoothing is most similar to an ARIMA with zero orders of autoregression, two orders of differencing, and two orders of moving average, with the coefficient for the second order of moving average equal to one half of the coefficient for the first order squared.
- **Simple seasonal.** This model is appropriate for a series in which there is no trend and a seasonal effect that is constant over time. Its relevant smoothing parameters are level and season. Seasonal

<sup>1</sup> Gardner, E. S. 1985. Exponential smoothing: The state of the art. *Journal of Forecasting*, 4, 1-28.

exponential smoothing is most similar to an ARIMA with zero orders of autoregression; one order of differencing; one order of seasonal differencing; and orders 1,  $p$ , and  $p+1$  of moving average, where  $p$  is the number of periods in a seasonal interval. For monthly data,  $p = 12$ .

- **Winters' additive.** This model is appropriate for a series in which there is a linear trend and a seasonal effect that is constant over time. Its relevant smoothing parameters are level, trend, and season. Winters' additive exponential smoothing is most similar to an ARIMA with zero orders of autoregression; one order of differencing; one order of seasonal differencing; and  $p+1$  orders of moving average, where  $p$  is the number of periods in a seasonal interval. For monthly data,  $p=12$ .
- **Damped trend with additive seasonal.** This model is appropriate for a series in which there is a linear trend that is dying out and a seasonal effect that is constant over time. Its relevant smoothing parameters are level, trend, damping trend, and season. Damped trend and additive seasonal exponential smoothing is not similar to any ARIMA model.
- **Multiplicative trend with additive seasonal.** This model is appropriate for a series in which there is a trend that changes with the magnitude of the series and a seasonal effect that is constant over time. Its relevant smoothing parameters are level, trend, and season. Multiplicative trend and additive seasonal exponential smoothing is not similar to any ARIMA model.
- **Multiplicative seasonal.** This model is appropriate for a series in which there is no trend and a seasonal effect that changes with the magnitude of the series. Its relevant smoothing parameters are level and season. Multiplicative seasonal exponential smoothing is not similar to any ARIMA model.
- **Winters' multiplicative.** This model is appropriate for a series in which there is a linear trend and a seasonal effect that changes with the magnitude of the series. Its relevant smoothing parameters are level, trend, and season. Winters' multiplicative exponential smoothing is not similar to any ARIMA model.
- **Damped trend with multiplicative seasonal.** This model is appropriate for a series in which there is a linear trend that is dying out and a seasonal effect that changes with the magnitude of the series. Its relevant smoothing parameters are level, trend, damping trend, and season. Damped trend and multiplicative seasonal exponential smoothing is not similar to any ARIMA model.
- **Multiplicative trend with multiplicative seasonal.** This model is appropriate for a series in which there are a trend and a seasonal effect that both change with the magnitude of the series. Its relevant smoothing parameters are level, trend, and season. Multiplicative trend and multiplicative seasonal exponential smoothing is not similar to any ARIMA model.

**Target Transformation.** You can specify a transformation to be performed on each dependent variable before it is modeled.

- **None.** No transformation is performed.
- **Square root.** Square root transformation is performed.
- **Natural log.** Natural log transformation is performed.

## ARIMA

Specify the structure of a custom ARIMA model.

**ARIMA Orders.** Enter values for the various ARIMA components of your model into the corresponding cells of the grid. All values must be non-negative integers. For autoregressive and moving average components, the value represents the maximum order. All positive lower orders are included in the model. For example, if you specify 2, the model includes orders 2 and 1. Cells in the Seasonal column are only enabled if a periodicity is defined for the active dataset.

- **Autoregressive (p).** The number of autoregressive orders in the model. Autoregressive orders specify which previous values from the series are used to predict current values. For example, an autoregressive order of 2 specifies that the value of the series two time periods in the past is used to predict the current value.
- **Difference (d).** Specifies the order of differencing applied to the series before estimating models. Differencing is necessary when trends are present (series with trends are typically nonstationary and ARIMA modeling assumes stationarity) and is used to remove their effect. The order of differencing

corresponds to the degree of series trend; first-order differencing accounts for linear trends, second-order differencing accounts for quadratic trends, and so on.

- **Moving Average (q).** The number of moving average orders in the model. Moving average orders specify how deviations from the series mean for previous values are used to predict current values. For example, moving-average orders of 1 and 2 specify that deviations from the mean value of the series from each of the last two time periods be considered when predicting current values of the series.

**Seasonal.** Seasonal autoregressive, moving average, and differencing components play the same roles as their nonseasonal counterparts. For seasonal orders, however, current series values are affected by previous series values that are separated by one or more seasonal periods. For example, for monthly data (seasonal period of 12), a seasonal order of 1 means that the current series value is affected by the series value 12 periods before the current one. A seasonal order of 1, for monthly data, is then the same as specifying a nonseasonal order of 12.

**Detect outliers automatically.** Select this option to perform automatic detection of outliers, and select one or more of the outlier types available.

**Type of Outliers to Detect.** Select the outlier type(s) you want to detect. The supported types are:

- Additive (default)
- Level shift (default)
- Innovational
- Transient
- Seasonal additive
- Local trend
- Additive patch

**Transfer Function Orders and Transformations.** To specify transformations and to define transfer functions for any or all of the input fields in your ARIMA model, click **Set**; a separate dialog box is displayed in which you enter the transfer and transformation details.

**Include constant in model.** Inclusion of a constant is standard unless you are sure that the overall mean series value is 0. Excluding the constant is recommended when differencing is applied.

### Further details

- For more information on types of outliers, see [“Outliers” on page 262](#).
- For more information on transfer and transformation functions, see [“Transfer and transformation functions” on page 291](#).

### *Transfer and transformation functions*

Use the Transfer Function Orders and Transformations dialog box to specify transformations and to define transfer functions for any or all of the input fields in your ARIMA model.

**Target Transformations.** In this pane you can specify a transformation to be performed on each target variable before it is modeled.

- **None.** No transformation is performed.
- **Square root.** Square root transformation is performed.
- **Natural log.** Natural log transformation is performed.

**Candidate Inputs Transfer functions and Transformation.** You use transfer functions to specify the manner in which past values of the input fields are used to forecast future values of the target series. The list on the left hand side of the pane shows all input fields. The remaining information in this pane is specific to the input field you select.

**Transfer Function Orders.** Enter values for the various components of the transfer function into the corresponding cells of the **Structure** grid. All values must be non-negative integers. For numerator and denominator components, the value represents the maximum order. All positive lower orders are

included in the model. In addition, order 0 is always included for numerator components. For example, if you specify 2 for numerator, the model includes orders 2, 1, and 0. If you specify 3 for denominator, the model includes orders 3, 2, and 1. Cells in the Seasonal column are only enabled if a periodicity is defined for the active dataset.

**Numerator.** The numerator order of the transfer function specifies which previous values from the selected independent (predictor) series are used to predict current values of the dependent series. For example, a numerator order of 1 specifies that the value of an independent series one time period in the past, in addition to the current value of the independent series, is used to predict the current value of each dependent series.

**Denominator.** The denominator order of the transfer function specifies how deviations from the series mean, for previous values of the selected independent (predictor) series, are used to predict current values of the dependent series. For example, a denominator order of 1 specifies that deviations from the mean value of an independent series one time period in the past is considered when predicting the current value of each dependent series.

**Difference.** Specifies the order of differencing applied to the selected independent (predictor) series before estimating models. Differencing is necessary when trends are present and is used to remove their effect.

**Seasonal.** Seasonal numerator, denominator, and differencing components play the same roles as their nonseasonal counterparts. For seasonal orders, however, current series values are affected by previous series values that are separated by one or more seasonal periods. For example, for monthly data (seasonal period of 12), a seasonal order of 1 means that the current series value is affected by the series value 12 periods prior to the current one. A seasonal order of 1, for monthly data, is then the same as specifying a nonseasonal order of 12.

**Delay.** Setting a delay causes the input field's influence to be delayed by the number of intervals specified. For example, if the delay is set to 5, the value of the input field at time  $t$  doesn't affect forecasts until five periods have elapsed ( $t + 5$ ).

**Transformation.** Specification of a transfer function for a set of independent variables also includes an optional transformation to be performed on those variables.

- **None.** No transformation is performed.
- **Square root.** Square root transformation is performed.
- **Natural log.** Natural log transformation is performed.

### Time Series node - build output options

**Maximum number of lags in ACF and PACF output.** Autocorrelation (ACF) and partial autocorrelation (PACF) are measures of association between current and past series values and indicate which past series values are most useful in predicting future values. You can set the maximum number of lags that are shown in tables and plots of autocorrelations and partial autocorrelations.

**Calculate predictor importance.** For models that produce an appropriate measure of importance, you can display a chart that indicates the relative importance of each predictor in estimating the model. Typically you want to focus your modeling efforts on the predictors that matter most, and consider dropping or ignoring the predictors that matter least. Predictor importance can take longer to calculate for some models, particularly when you are working with large datasets, and is off by default for some models as a result.

### Time Series node - model options

**Model name.** You can generate the model name automatically based on the target or ID field (or model type in cases where no such field is specified) or specify a custom name.

**Confidence limit width (%).** Confidence intervals are computed for the model predictions and residual autocorrelations. You can specify any positive value less than 100. By default, a 95% confidence interval is used.



**Continue estimation using existing model(s).** If you already generated a time series model, select this option to reuse the criteria settings that are specified for that model and generate a new model node in the Models palette, rather than building a new model from the beginning. In this way, you can save time by re-estimating and producing a new forecast that is based on the same model settings as before but using more recent data. Thus, for example, if the original model for a particular time series was Holt's linear trend, the same type of model is used for reestimating and forecasting for that data. The system does not reattempt to find the best model type for the new data.

**Build scoring model only.** To reduce the amount of data that is stored in the model, check this box. Using this option can improve performance when building models with large numbers of time series (tens of thousands). You can still score the data in the usual way.

**Extend records into the future.** Enables the following **Future Values to Use in Forecasting** section, where you can set the number of time intervals to forecast beyond the end of the estimation period. The time interval in this case is the time interval of the analysis, which you specify on the Data Specifications tab. There is no maximum limit for this setting. Using the following options, you can automatically compute future values of inputs, or you can manually specify forecasting values for one or more predictors.

### Future Values to Use in Forecasting

- **Compute future values of inputs** If you select this option, the forecast values for predictors, noise predictions, variance estimation, and future time values are calculated automatically. When forecasts are requested, autoregressive models are automatically built for any input series that are not also targets. These models are then used to generate values for those input series in the forecast period.
- **Select fields whose values you wish to add to the data.** For each record that you want to forecast (excluding holdouts), if you are using predictor fields (with the role set to Input), you can specify estimated values for the forecast period for each predictor. You can either specify values manually, or choose from a list.
  - **Field.** Click the field selector button and choose any fields that may be used as predictors. Note that fields selected here may or may not be used in modeling; to actually use a field as a predictor, it must be selected in a downstream modeling node. This dialog box simply gives you a convenient place to specify future values so they can be shared by multiple downstream modeling nodes without specifying them separately in each node. Also note that the list of available fields may be constrained by selections on the Build Options tab.

Note that if future values are specified for a field that is no longer available in the stream (because it has been dropped or because of updated selections made on the Build Options tab), the field is shown in red.

- **Values.** For each field, you can choose from a list of functions, or click **Specify** to either enter values manually or choose from a list of predefined values. If the predictor fields relate to items that are under your control, or which are otherwise knowable in advance, you should enter values manually. For example, if you are forecasting next month's revenues for a hotel based on the number of room reservations, you could specify the number of reservations you actually have for that period. Conversely, if a predictor field relates to something outside your control, such as a stock price, you could use a function such as the most recent value or the mean of recent points.

The available functions depend on the measurement level of the field.

Table 28. Functions available for measurement levels	
Measurement level	Functions
Continuous or Nominal field	Blank Mean of recent points Most recent value Specify

Table 28. Functions available for measurement levels (continued)	
Measurement level	Functions
Flag field	Blank Most recent value True False Specify

**Mean of recent points** calculates the future value from the mean of the last three data points.

**Most recent value** sets the future value to that of the most recent data point.

**True/False** sets the future value of a flag field to True or False as specified.

**Specify** opens a dialog box for specifying future values manually, or choosing them from a predefined list.

### Make Available for Scoring

You can set the default values here for the scoring options that appear on the dialog box for the model nugget.

- **Calculate upper and lower confidence limits.** If selected, this option creates new fields (with the default prefixes \$TSLCI- and \$TSUCI-) for the lower and upper confidence intervals, for each target field.
- **Calculate noise residuals.** If selected, this option creates a new field (with the default prefix \$TSResidual-) for the model residuals for each target field, together with a total of these values.

### Model Settings

**Maximum number of models to be displayed in output.** Specify the maximum number of models you want to include in the output. Note that if the number of models built exceeds this threshold, the models are not shown in the output but they're still available for scoring. Default value is 10. Displaying a large number of models may result in poor performance or instability.

## Time Series model nugget

### Time Series model nugget output

After you create a Time Series model, the following information is available in the Output viewer. Note that there's a limit of 10 models that can be displayed in the Output viewer for Time Series models.

### Temporal Information Summary

The summary shows the following information:

- The Time field
- The increment
- The start and end point
- The number of unique points

The summary applies to all targets.

### Model information table

Repeated for each target, the Model information table provides key information about the model. The table always includes the following high-level model settings:

- The name of the target field that is selected in either the Type node or the Time Series node Fields tab.

- The model building method - for example, Exponential Smoothing, or ARIMA.
- The number of predictors input into the model.
- The number of records that were used to fit the model type. Examples of the different types of models might include: RMSE, MAE, AIC, BIC, and R Square.

In addition, the Ljung-Box Q statistic might also be shown if your data meets the required conditions. This statistic is **not** available under the following conditions:

- If the number of non-missing data points is less than or equal to the number of summation terms desired (fixed at 18).
- If the number of parameters is greater than or equal to the number of summation terms desired.
- If the number of summation terms that are computed is less than the smallest acceptable k value (fixed at 7).
- If the table repeats for each target.

### Predictor Importance

Repeated for each target, the Predictor Importance graph shows the importance of the top 10 inputs (predictors) in the model as a bar chart.

If there are more than 10 fields in the chart, you can change the selection of predictors that are included in the chart by using the slider beneath the chart. The indicator marks on the slider are a fixed width, and each mark on the slider represents 10 fields. You can move the indicator marks along the slider to display the next or previous 10 fields, ordered by predictor importance.

You can double-click the chart to open a separate dialog box in which you can edit the graph settings. For example, you can amend items such as the size of the graph, and the size and color of the fonts used. When you close this separate editing dialog box, the changes are applied to the chart that is displayed in the Output tab.

### Correlogram

A correlogram, or autocorrelation plot, is shown for each target and shows the autocorrelation function (ACF), or partial autocorrelation function (PACF), of the residuals (the differences between expected and actual values) versus the time lags. The confidence interval is shown as a highlight across the chart.

### Parameter Estimates

Repeated for each target, the Parameter Estimates table shows (where applicable) the following details:

- Target name
- The transformation applied
- The lags used for this parameter in the model (ARIMA)
- The coefficient value
- The standard error of the parameter estimate
- The value of the parameter estimate divided by the standard error
- The significance level for the parameter estimate.

### Time Series model nugget settings

The Settings tab provides additional options for the Time Series model nugget.

### Forecast

The option to **Extend records into the future**, sets the number of time intervals to forecast beyond the end of the estimation period. The time interval in this case is the time interval of the analysis, which is specified on the Data Specifications tab of the Time Series node. When forecasts are requested,

autoregressive models are automatically built for any input series that are not also targets. These models are then used to generate values for those input series in the forecast period.

**Compute future values of inputs.** If you select this option, the forecast values for predictors, noise predictions, variance estimation, and future time values are calculated.

### Future Values to Use in Forecasting

- **Compute future values of inputs** If you select this option, the forecast values for predictors, noise predictions, variance estimation, and future time values are calculated automatically. When forecasts are requested, autoregressive models are automatically built for any input series that are not also targets. These models are then used to generate values for those input series in the forecast period.
- **Select fields whose values you wish to add to the data.** For each record that you want to forecast (excluding holdouts), if you are using predictor fields (with the role set to Input), you can specify estimated values for the forecast period for each predictor. You can either specify values manually, or choose from a list.
  - **Field.** Click the field selector button and choose any fields that may be used as predictors. Note that fields selected here may or may not be used in modeling; to actually use a field as a predictor, it must be selected in a downstream modeling node. This dialog box simply gives you a convenient place to specify future values so they can be shared by multiple downstream modeling nodes without specifying them separately in each node. Also note that the list of available fields may be constrained by selections on the Build Options tab.

Note that if future values are specified for a field that is no longer available in the stream (because it has been dropped or because of updated selections made on the Build Options tab), the field is shown in red.

- **Values.** For each field, you can choose from a list of functions, or click **Specify** to either enter values manually or choose from a list of predefined values. If the predictor fields relate to items that are under your control, or which are otherwise knowable in advance, you should enter values manually. For example, if you are forecasting next month's revenues for a hotel based on the number of room reservations, you could specify the number of reservations you actually have for that period. Conversely, if a predictor field relates to something outside your control, such as a stock price, you could use a function such as the most recent value or the mean of recent points.

The available functions depend on the measurement level of the field.

Table 29. Functions available for measurement levels	
Measurement level	Functions
Continuous or Nominal field	Blank Mean of recent points Most recent value Specify
Flag field	Blank Most recent value True False Specify

**Mean of recent points** calculates the future value from the mean of the last three data points.

**Most recent value** sets the future value to that of the most recent data point.

**True/False** sets the future value of a flag field to True or False as specified.

**Specify** opens a dialog box for specifying future values manually, or choosing them from a predefined list.

### **Make available for scoring**

**Create new fields for each model to be scored.** Enables you to specify the new fields to create for each model to be scored.

- **Noises Residuals.** If selected, creates a new field (with the default prefix \$TSResidual-) for the model residuals for each target field, together with a total of these values.
- **Upper and lower confidence limits.** If selected, this option creates new fields (with the default prefixes \$TSLCI- and \$TSUCI-) for the lower and upper confidence intervals respectively, for each target field, together with totals of these values.

**Targets included for scoring.** Select available targets to include in the model score.



---

# Chapter 14. Self-Learning Response Node Models

## SLRM node

---

The **Self-Learning Response Model** (SLRM) node enables you to build a model that you can continually update, or reestimate, as a dataset grows without having to rebuild the model every time using the complete dataset. For example, this is useful when you have several products and you want to identify which product a customer is most likely to buy if you offer it to them. This model allows you to predict which offers are most appropriate for customers and the probability of the offers being accepted.

The model can initially be built using a small dataset with randomly made offers and the responses to those offers. As the dataset grows, the model can be updated and therefore becomes more able to predict the most suitable offers for customers and the probability of their acceptance based upon other input fields such as age, gender, job, and income. The offers available can be changed by adding or removing them from within the node dialog box, instead of having to change the target field of the dataset.

When coupled with IBM SPSS Collaboration and Deployment Services, you can set up automatic regular updates to the model. This process, without the need for human oversight or action, provides a flexible and low-cost solution for organizations and applications where custom intervention by a data miner is not possible or necessary.

**Example.** A financial institution wants to achieve more profitable results by matching the offer that is most likely to be accepted to each customer. You can use a self-learning model to identify the characteristics of customers most likely to respond favorably based on previous promotions and to update the model in real time based on the latest customer responses.

### SLRM Node Fields Options

Before executing an SLRM node, you must specify both the target and target response fields on the Fields tab of the node.

**Target field.** Select the target field from the list; for example, a nominal (set) field containing the different products you want to offer to customers.

*Note:* The target field must have string storage, not numeric.

**Target response field.** Select the target response field from the list. For example, Accepted or Rejected.

*Note:* This field must be a Flag. The true value of the flag indicates offer acceptance and the false value indicates offer refusal.

The remaining fields in this dialog box are the standard ones used throughout IBM SPSS Modeler. See the topic [“Modeling Node Fields Options”](#) on page 26 for more information.

*Note:* If the source data includes ranges that are to be used as continuous (numeric range) input fields, you must ensure that the metadata includes both the minimum and maximum details for each range.

### SLRM Node Model Options

**Model name.** You can generate the model name automatically based on the target or ID field (or model type in cases where no such field is specified) or specify a custom name.

**Use partitioned data.** If a partition field is defined, this option ensures that data from only the training partition is used to build the model.

**Continue training existing model.** By default, a completely new model is created each time a modeling node is executed. If this option is selected, training continues with the last model successfully produced by the node. This makes it possible to update or refresh an existing model without having to access the original data and may result in significantly faster performance since *only* the new or updated records are

fed into the stream. Details on the previous model are stored with the modeling node, making it possible to use this option even if the previous model nugget is no longer available in the stream or Models palette.

**Target field values** By default this is set to **Use all**, which means that a model will be built that contains every offer associated with the selected target field value. If you want to generate a model that contains only some of the target field's offers, click **Specify** and use the **Add**, **Edit**, and **Delete** buttons to enter or amend the names of the offers for which you want to build a model. For example, if you chose a target that lists all of the products you supply, you can use this field to limit the offered products to just a few that you enter here.

**Model Assessment.** The fields in this panel are independent from the model in that they don't affect the scoring. Instead they enable you to create a visual representation of how well the model will predict results.

*Note:* To display the model assessment results in the model nugget you must also select the **Display model evaluation** box.

- **Include model assessment.** Select this box to create graphs that show the model's predicted accuracy for each selected offer.
- **Set random seed.** When estimating the accuracy of a model based on a random percentage, this option allows you to duplicate the same results in another session. By specifying the starting value used by the random number generator, you can ensure the same records are assigned each time the node is executed. Enter the desired seed value. If this option is not selected, a different sample will be generated each time the node is executed.
- **Simulated sample size.** Specify the number of records to be used in the sample when assessing the model. The default is 100.
- **Number of iterations.** This enables you to stop building the model assessment after the number of iterations specified. Specify the maximum number of iterations; the default is 20.

*Note:* Bear in mind that large sample sizes and high numbers of iterations will increase the amount of time it takes to build the model.

**Display model evaluation.** Select this option to display a graphical representation of the results in the model nugget.

## SLRM Node Settings Options

The node settings options allow you to fine-tune the model-building process.

**Maximum number of predictions per record.** This option allows you to limit the number of predictions made for each record in the dataset. The default is 3.

For example, you may have six offers (such as savings, mortgage, car loan, pension, credit card, and insurance), but you only want to know the best two to recommend; in this case you would set this field to 2. When you build the model and attach it to a table, you would see two prediction columns (and the associated confidence in the probability of the offer being accepted) per record. The predictions could be made up of any of the six possible offers.

**Level of randomization.** To prevent any bias—for example, in a small or incomplete dataset—and treat all potential offers equally, you can add a level of randomization to the selection of offers and the probability of their being included as recommended offers. Randomization is expressed as a percentage, shown as decimal values between 0.0 (no randomization) and 1.0 (completely random). The default is 0.0.

**Set random seed.** When adding a level of randomization to selection of an offer, this option allows you to duplicate the same results in another session. By specifying the starting value used by the random number generator, you can ensure the same records are assigned each time the node is executed. Enter the desired seed value. If this option is not selected, a different sample will be generated each time the node is executed.

**Note:** When using the **Set random seed** option with records read from a database, a Sort node may be required prior to sampling in order to ensure the same result each time the node is executed. This is



because the random seed depends on the order of records, which is not guaranteed to stay the same in a relational database.

**Sort order.** Select the order in which offers are to be displayed in the built model:

- **Descending.** The model displays offers with the highest scores first. These are the offers that have the greatest probability of being accepted.
- **Ascending.** The model displays offers with the lowest scores first. These are the offers that have the greatest probability of being rejected. For example, this may be useful when deciding which customers to remove from a marketing campaign for a specific offer.

**Preferences for target fields.** When building a model, there may be certain aspects of the data that you want to actively promote or remove. For example, if building a model that selects the best financial offer to promote to a customer, you may want to ensure that one particular offer is always included regardless of how well it scores against each customer.

To include an offer in this panel and edit its preferences, click **Add**, type the offer's name (for example, Savings or Mortgage), and click **OK**.

- **Value.** This shows the name of the offer that you added.
- **Preference.** Specify the level of preference to be applied to the offer. Preference is expressed as a percentage, shown as decimal values between 0.0 (not preferred) and 1.0 (most preferred). The default is 0.0.
- **Always include.** To ensure that a specific offer is always included in the predictions, select this box.

**Note:** If the **Preference** is set to 0.0, the **Always include** setting is ignored.

**Take account of model reliability.** A well-structured, data-rich model that has been fine-tuned through several regenerations should always produce more accurate results compared to a brand new model with little data. To take advantage of the more mature model's increased reliability, select this box.

## SLRM Model Nuggets

---

*Note:* Results are only shown on this tab if you select both **Include model assessment** and **Display model evaluation** on the Model options tab.

When you run a stream containing an SLRM model, the node estimates the accuracy of the predictions for each target field value (offer) and the importance of each predictor used.

*Note:* If you selected **Continue training existing model** on the modeling node Model tab, the information shown on the model nugget is updated each time you regenerate the model.

For models built using IBM SPSS Modeler 12.0 or later, the model nugget Model tab is divided into two columns:

### Left column.

- **View.** When you have more than one offer, select the one for which you want to display results.
- **Model Performance.** This shows the estimated model accuracy of each offer. The test set is generated through simulation.

### Right column.

- **View.** Select whether you want to display **Association with Response** or **Variable Importance** details.
- **Association with Response.** Displays the association (correlation) of each predictor with the target variable.
- **Predictor Importance.** Indicates the relative importance of each predictor in estimating the model. Typically you will want to focus your modeling efforts on the predictors that matter most and consider dropping or ignoring those that matter least. This chart can be interpreted in the same manner as for other models that display predictor importance, though in the case of SLRM the graph is generated through simulation by the SLRM algorithm. This is done by removing each predictor in turn from the model and seeing how this affects the model's accuracy. See the topic [“Predictor Importance” on page 37](#) for more information.

## SLRM Model Settings

The Settings tab for a SLRM model nugget specifies options for modifying the built model. For example, you may use the SLRM node to build several different models using the same data and settings, then use this tab in each model to slightly modify the settings to see how that affects the results.

**Note:** This tab is only available after the model nugget has been added to a stream.

**Maximum number of predictions per record.** This option allows you to limit the number of predictions made for each record in the dataset. The default is 3.

For example, you may have six offers (such as savings, mortgage, car loan, pension, credit card, and insurance), but you only want to know the best two to recommend; in this case you would set this field to 2. When you build the model and attach it to a table, you would see two prediction columns (and the associated confidence in the probability of the offer being accepted) per record. The predictions could be made up of any of the six possible offers.

**Level of randomization.** To prevent any bias—for example, in a small or incomplete dataset—and treat all potential offers equally, you can add a level of randomization to the selection of offers and the probability of their being included as recommended offers. Randomization is expressed as a percentage, shown as decimal values between 0.0 (no randomization) and 1.0 (completely random). The default is 0.0.

**Set random seed.** When adding a level of randomization to selection of an offer, this option allows you to duplicate the same results in another session. By specifying the starting value used by the random number generator, you can ensure the same records are assigned each time the node is executed. Enter the desired seed value. If this option is not selected, a different sample will be generated each time the node is executed.

**Note:** When using the **Set random seed** option with records read from a database, a Sort node may be required prior to sampling in order to ensure the same result each time the node is executed. This is because the random seed depends on the order of records, which is not guaranteed to stay the same in a relational database.

**Sort order.** Select the order in which offers are to be displayed in the built model:

- **Descending.** The model displays offers with the highest scores first. These are the offers that have the greatest probability of being accepted.
- **Ascending.** The model displays offers with the lowest scores first. These are the offers that have the greatest probability of being rejected. For example, this may be useful when deciding which customers to remove from a marketing campaign for a specific offer.

**Preferences for target fields.** When building a model, there may be certain aspects of the data that you want to actively promote or remove. For example, if building a model that selects the best financial offer to promote to a customer, you may want to ensure that one particular offer is always included regardless of how well it scores against each customer.

To include an offer in this panel and edit its preferences, click **Add**, type the offer's name (for example, Savings or Mortgage), and click **OK**.

- **Value.** This shows the name of the offer that you added.
- **Preference.** Specify the level of preference to be applied to the offer. Preference is expressed as a percentage, shown as decimal values between 0.0 (not preferred) and 1.0 (most preferred). The default is 0.0.
- **Always include.** To ensure that a specific offer is always included in the predictions, select this box.

**Note:** If the **Preference** is set to 0.0, the **Always include** setting is ignored.

**Take account of model reliability.** A well-structured, data-rich model that has been fine-tuned through several regenerations should always produce more accurate results compared to a brand new model with little data. To take advantage of the more mature model's increased reliability, select this box.

**Generate SQL for this model** When using data from a database, SQL code can be pushed back to the database for execution, providing superior performance for many operations.

Select one of the following options to specify how SQL generation is performed.

- **Default: Score using Server Scoring Adapter (if installed) otherwise in process** If connected to a database with a scoring adapter installed, generates SQL using the scoring adapter and associated user defined functions (UDF) and scores your model within the database. When no scoring adapter is available, this option fetches your data back from the database and scores it in SPSS Modeler.
- **Score outside of the Database** If selected, this option fetches your data back from the database and scores it in SPSS Modeler.



---

# Chapter 15. Support Vector Machine Models

## About SVM

---

Support Vector Machine (SVM) is a robust classification and regression technique that maximizes the predictive accuracy of a model without overfitting the training data. SVM is particularly suited to analyzing data with very large numbers (for example, thousands) of predictor fields.

SVM has applications in many disciplines, including customer relationship management (CRM), facial and other image recognition, bioinformatics, text mining concept extraction, intrusion detection, protein structure prediction, and voice and speech recognition.

## How SVM Works

---

SVM works by mapping data to a high-dimensional feature space so that data points can be categorized, even when the data are not otherwise linearly separable. A separator between the categories is found, then the data are transformed in such a way that the separator could be drawn as a hyperplane. Following this, characteristics of new data can be used to predict the group to which a new record should belong.

For example, consider the following figure, in which the data points fall into two different categories.

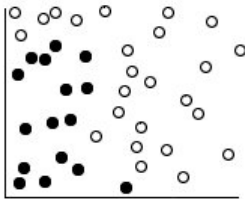


Figure 59. Original dataset

The two categories can be separated with a curve, as shown in the following figure.

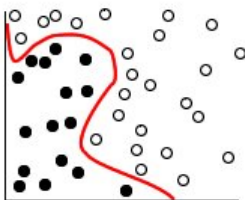


Figure 60. Data with separator added

After the transformation, the boundary between the two categories can be defined by a hyperplane, as shown in the following figure.

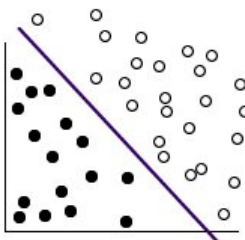


Figure 61. Transformed data

The mathematical function used for the transformation is known as the **kernel** function. SVM in IBM SPSS Modeler supports the following kernel types:

- Linear
- Polynomial
- Radial basis function (RBF)
- Sigmoid

A linear kernel function is recommended when linear separation of the data is straightforward. In other cases, one of the other functions should be used. You will need to experiment with the different functions to obtain the best model in each case, as they each use different algorithms and parameters.

## Tuning an SVM Model

Besides the separating line between the categories, a classification SVM model also finds marginal lines that define the space between the two categories.

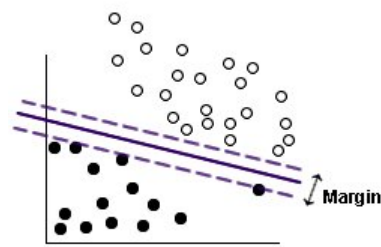


Figure 62. Data with a preliminary model

The data points that lie on the margins are known as the **support vectors**.

The wider the margin between the two categories, the better the model will be at predicting the category for new records. In the previous example, the margin is not very wide, and the model is said to be **overfitted**. A small amount of misclassification can be accepted in order to widen the margin; an example of this is shown in the following figure.

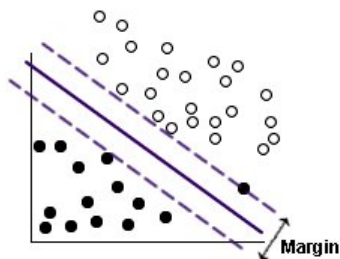


Figure 63. Data with an improved model

In some cases, linear separation is more difficult; an example of this is shown in the following figure.

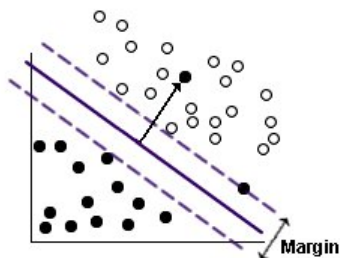


Figure 64. A problem for linear separation

In a case like this, the goal is to find the optimum balance between a wide margin and a small number of misclassified data points. The kernel function has a **regularization parameter** (known as C) which controls the trade-off between these two values. You will probably need to experiment with different values of this and other kernel parameters in order to find the best model.

## SVM node

---

The SVM node enables you to use a support vector machine to classify data. SVM is particularly suited for use with wide datasets, that is, those with a large number of predictor fields. You can use the default settings on the node to produce a basic model relatively quickly, or you can use the Expert settings to experiment with different types of SVM model.

When the model has been built, you can:

- Browse the model nugget to display the relative importance of the input fields in building the model.
- Append a Table node to the model nugget to view the model output.

**Example.** A medical researcher has obtained a dataset containing characteristics of a number of human cell samples extracted from patients who were believed to be at risk of developing cancer. Analysis of the original data showed that many of the characteristics differed significantly between benign and malignant samples. The researcher wants to develop an SVM model that can use the values of similar cell characteristics in samples from other patients to give an early indication of whether their samples might be benign or malignant.

### SVM Node Model Options

**Model name.** You can generate the model name automatically based on the target or ID field (or model type in cases where no such field is specified) or specify a custom name.

**Use partitioned data.** If a partition field is defined, this option ensures that data from only the training partition is used to build the model.

**Create split models.** Builds a separate model for each possible value of input fields that are specified as split fields. See [“Building Split Models”](#) on page 24 for more information.

### SVM Node Expert Options

If you have detailed knowledge of support vector machines, expert options allow you to fine-tune the training process. To access the expert options, set Mode to **Expert** on the Expert tab.

**Append all probabilities (valid only for categorical targets).** If selected (checked), specifies that probabilities for each possible value of a nominal or flag target field are displayed for each record processed by the node. If this option is not selected, the probability of only the predicted value is displayed for nominal or flag target fields. The setting of this check box determines the default state of the corresponding check box on the model nugget display.

**Stopping criteria.** Determines when to stop the optimization algorithm. Values range from 1.0E–1 to 1.0E–6; default is 1.0E–3. Reducing the value results in a more accurate model, but the model will take longer to train.

**Regularization parameter (C).** Controls the trade-off between maximizing the margin and minimizing the training error term. Value should normally be between 1 and 10 inclusive; default is 10. Increasing the value improves the classification accuracy (or reduces the regression error) for the training data, but this can also lead to overfitting.

**Regression precision (epsilon).** Used only if the measurement level of the target field is *Continuous*. Causes errors to be accepted provided that they are less than the value specified here. Increasing the value may result in faster modeling, but at the expense of accuracy.

**Kernel type.** Determines the type of kernel function used for the transformation. Different kernel types cause the separator to be calculated in different ways, so it is advisable to experiment with the various options. Default is **RBF** (Radial Basis Function).

**RBF gamma.** Enabled only if the kernel type is set to **RBF**. Value should normally be between  $3/k$  and  $6/k$ , where  $k$  is the number of input fields. For example, if there are 12 input fields, values between 0.25 and 0.5 would be worth trying. Increasing the value improves the classification accuracy (or reduces the regression error) for the training data, but this can also lead to overfitting.

**Gamma.** Enabled only if the kernel type is set to **Polynomial** or **Sigmoid**. Increasing the value improves the classification accuracy (or reduces the regression error) for the training data, but this can also lead to overfitting.

**Bias.** Enabled only if the kernel type is set to **Polynomial** or **Sigmoid**. Sets the `coef0` value in the kernel function. The default value 0 is suitable in most cases.

**Degree.** Enabled only if Kernel type is set to **Polynomial**. Controls the complexity (dimension) of the mapping space. Normally you would not use a value greater than 10.

## SVM Model Nugget

The SVM model creates a number of new fields. The most important of these is the **\$S-fieldname** field, which shows the target field value predicted by the model.

The number and names of the new fields created by the model depend on the measurement level of the target field (this field is indicated in the following tables by *fieldname*).

To see these fields and their values, add a Table node to the SVM model nugget and execute the Table node.

Table 30. Target field measurement level is 'Nominal' or 'Flag'	
New field name	Description
\$S-fieldname	Predicted value of target field.
\$SP-fieldname	Probability of predicted value.
\$SP-value	Probability of each possible value of nominal or flag (displayed only if <b>Append all probabilities</b> is checked on the Settings tab of the model nugget).
\$SRP-value	(Flag targets only) Raw (SRP) and adjusted (SAP) propensity scores, indicating the likelihood of a "true" outcome for the target field. These scores are displayed only if the corresponding check boxes are selected on the Analyze tab of the SVM modeling node before the model is generated. See the topic <a href="#">"Modeling Node Analyze Options"</a> on page 29 for more information.
\$SAP-value	

Table 31. Target field measurement level is 'Continuous'	
New field name	Description
\$S-fieldname	Predicted value of target field.

### Predictor Importance

Optionally, a chart that indicates the relative importance of each predictor in estimating the model may also be displayed on the Model tab. Typically you will want to focus your modeling efforts on the predictors that matter most and consider dropping or ignoring those that matter least. Note this chart is only available if **Calculate predictor importance** is selected on the Analyze tab before generating the model. See the topic ["Predictor Importance"](#) on page 37 for more information.

*Note:* Predictor importance may take longer to calculate for SVM than for other types of models, and is not selected on the Analyze tab by default. Selecting this option may slow performance, particularly with large datasets.



## SVM Model Settings

The Settings tab enables you to specify extra fields to be displayed when viewing the results (for example by executing a Table node attached to the nugget). You can see the effect of each of these options by selecting them and clicking the Preview button--scroll to the right of the Preview output to see the extra fields.

**Append all probabilities (valid only for categorical targets).** If this option is checked, probabilities for each possible value of a nominal or flag target field are displayed for each record processed by the node. If this option is unchecked, only the predicted value and its probability are displayed for nominal or flag target fields.

The default setting of this check box is determined by the corresponding check box on the modeling node.

**Calculate raw propensity scores.** For models with a flag target (which return a yes or no prediction), you can request propensity scores that indicate the likelihood of the true outcome specified for the target field. These are in addition to other prediction and confidence values that may be generated during scoring.

**Calculate adjusted propensity scores.** Raw propensity scores are based only on the training data and may be overly optimistic due to the tendency of many models to overfit this data. Adjusted propensities attempt to compensate by evaluating model performance against a test or validation partition. This option requires that a partition field be defined in the stream and adjusted propensity scores be enabled in the modeling node before generating the model.

**Generate SQL for this model** When using data from a database, SQL code can be pushed back to the database for execution, providing superior performance for many operations.

Select one of the following options to specify how SQL generation is performed.

- **Default: Score using Server Scoring Adapter (if installed) otherwise in process** If connected to a database with a scoring adapter installed, generates SQL using the scoring adapter and associated user defined functions (UDF) and scores your model within the database. When no scoring adapter is available, this option fetches your data back from the database and scores it in SPSS Modeler.
- **Score outside of the Database** If selected, this option fetches your data back from the database and scores it in SPSS Modeler.

## LSVM Node

---

The LSVM node enables you to use a linear support vector machine to classify data. LSVM is particularly suited for use with wide datasets, that is, those with a large number of predictor fields. You can use the default settings on the node to produce a basic model relatively quickly, or you can use the build options to experiment with different settings.

The LSVM node is similar to the SVM node, but it is linear and is better at handling a large number of records.

When the model has been built, you can:

- Browse the model nugget to display the relative importance of the input fields in building the model.
- Append a Table node to the model nugget to view the model output.

**Example.** A medical researcher has obtained a dataset containing characteristics of a number of human cell samples extracted from patients who were believed to be at risk of developing cancer. Analysis of the original data showed that many of the characteristics differed significantly between benign and malignant samples. The researcher wants to develop an LSVM model that can use the values of similar cell characteristics in samples from other patients to give an early indication of whether their samples might be benign or malignant.

## LSVM Node Model Options

**Model name.** You can generate the model name automatically based on the target or ID field (or model type in cases where no such field is specified) or specify a custom name.

**Calculate predictor importance.** For models that produce an appropriate measure of importance, you can display a chart that indicates the relative importance of each predictor in estimating the model. Typically you will want to focus your modeling efforts on the predictors that matter most, and consider dropping or ignoring those that matter least. Note that predictor importance may take longer to calculate for some models, particularly when working with large datasets, and is off by default for some models as a result. Predictor importance is not available for decision list models. See [“Predictor Importance” on page 37](#) for more information.

## LSVM Build Options

### Model Settings

**Include intercept.** Including the intercept (the constant term in the model) can increase the overall accuracy of the solution. If you can assume the data passes through the origin, you can exclude the intercept.

**Sorting order for categorical target.** Specifies the sorting order for the categorical target. This setting is ignored for continuous targets.

**Regression precision (epsilon).** Used only if the measurement level of the target field is *Continuous*. Causes errors to be accepted provided that they are less than the value specified here. Increasing the value may result in faster modeling, but at the expense of accuracy.

**Exclude records with any missing values.** When set to **True**, a record is excluded if any single value is missing.

### Penalty Settings

**Penalty function.** Specifies the type of penalty function that is used to reduce the likelihood of overfitting. The options are either **L1** or **L2**.

**L1** and **L2** reduce the chance of overfitting by adding a penalty on the coefficients. The difference between them is that when there is a large number of features, **L1** uses feature selection by setting some coefficients to 0 during model building. **L2** does not have this ability so should not be used when you have a large number of features.

**Penalty parameter (lambda).** Specifies the penalty (regularization) parameter. This setting is enabled if the **Penalty function** is set.

## LSVM Model Nugget (interactive output)

---

After running an LSVM model, the following output is available.

### Model Information

The Model Information view provides key information about the model. The table identifies some high-level model settings, such as:

- The name of the target specified on the Fields tab
- The model building method specified on the [Model Selection](#) settings
- The number of predictors input
- The number of predictors in the final model
- The regularization type (L1 or L2)
- The penalty parameter (lambda). This is the regularization parameter.
- The regression precision (epsilon). Errors are accepted if they are less than this value. A higher value may result in faster modeling, but at the expense of accuracy. It is only used only if the measurement level of the target field is *Continuous*.
- The classification accuracy percentage. This is only applicable for Classification.

- The average squared error. This is only applicable for Regression.

### Records Summary

The Records Summary view provides information about the number and percentage of records (cases) included and excluded from the model.

### Predictor Importance

Typically, you will want to focus your modeling efforts on the predictor fields that matter most and consider dropping or ignoring those that matter least. The predictor importance chart helps you do this by indicating the relative importance of each predictor in estimating the model. Since the values are relative, the sum of the values for all predictors on the display is 1.0. Predictor importance does not relate to model accuracy. It just relates to the importance of each predictor in making a prediction, not whether or not the prediction is accurate.

### Predicted by Observed

This displays a binned scatterplot of the predicted values on the vertical axis by the observed values on the horizontal axis. Ideally, the points should lie on a 45-degree line; this view can tell you whether any records are predicted particularly badly by the model.

**Note:** Predictor importance may take longer to calculate for LSVM and SVM than for other types of models. Selecting this option may slow performance, particularly with large datasets.

### Confusion Matrix

The confusion matrix, sometimes referred to as the summary table, shows the number of cases correctly and incorrectly assigned to each of the groups based on the LSVM analysis.

## LSVM Model Settings

On the Settings tab for an SVLM model nugget, you specify options for raw propensity and for SQL generation during model scoring. This tab is available only after the model nugget is added to a stream.

**Calculate raw propensity scores** For models with flag targets only, you can request raw propensity scores that indicate the likelihood of the true outcome specified for the target field. These are in addition to standard prediction and confidence values. Adjusted propensity scores are not available.

**Generate SQL for this model** When using data from a database, SQL code can be pushed back to the database for execution, providing superior performance for many operations.

Select one of the following options to specify how SQL is generated.

- **Default: Score using Server Scoring Adapter (if installed) otherwise in process.** If connected to a database with a scoring adapter installed, generates SQL using the scoring adapter and associated user defined functions (UDF) and scores your model within the database. When no scoring adapter is available, this option fetches your data back from the database and scores it in SPSS Modeler.
- **Score outside of the Database.** If selected, this option fetches your data back from the database and scores it in SPSS Modeler.



---

# Chapter 16. Nearest Neighbor Models

## KNN node

---

Nearest Neighbor Analysis is a method for classifying cases based on their similarity to other cases. In machine learning, it was developed as a way to recognize patterns of data without requiring an exact match to any stored patterns, or cases. Similar cases are near each other and dissimilar cases are distant from each other. Thus, the distance between two cases is a measure of their dissimilarity.

Cases that are near each other are said to be “neighbors.” When a new case (holdout) is presented, its distance from each of the cases in the model is computed. The classifications of the most similar cases – the nearest neighbors – are tallied and the new case is placed into the category that contains the greatest number of nearest neighbors.

You can specify the number of nearest neighbors to examine; this value is called  $k$ . The pictures show how a new case would be classified using two different values of  $k$ . When  $k = 5$ , the new case is placed in category 1 because a majority of the nearest neighbors belong to category 1. However, when  $k = 9$ , the new case is placed in category 0 because a majority of the nearest neighbors belong to category 0.

Nearest neighbor analysis can also be used to compute values for a continuous target. In this situation, the average or median target value of the nearest neighbors is used to obtain the predicted value for the new case.

## KNN Node Objectives Options

The Objectives tab is where you can choose either to build a model that predicts the value of a target field in your input data based on the values of its nearest neighbors, or to simply find which are the nearest neighbors for a particular case of interest.

What type of analysis do you want to perform?

**Predict a target field.** Choose this option if you want to predict the value of a target field based on the values of its nearest neighbors.

**Only identify the nearest neighbors.** Choose this option if you only want to see which are the nearest neighbors for a particular input field.

If you choose to identify only the nearest neighbors, the remaining options on this tab relating to accuracy and speed are disabled as they are relevant only for predicting targets.

What is your objective?

When predicting a target field, this group of options lets you decide whether speed, accuracy, or a blend of both, are the most important factors when predicting a target field. Alternatively you can choose to customize settings yourself.

If you choose the Balance, Speed, or Accuracy option, the algorithm preselects the most appropriate combination of settings for that option. Advanced users may wish to override these selections; this can be done on the various panels of the Settings tab.

**Balance speed and accuracy.** Selects the best number of neighbors within a small range.

**Speed.** Finds a fixed number of neighbors.

**Accuracy.** Selects the best number of neighbors within a larger range, and uses predictor importance when calculating distances.

**Custom analysis.** Choose this option to fine-tune the algorithm on the Settings tab.

**Note:** The size of the resulting KNN model, unlike most other models, increases linearly with the quantity of training data. If, when trying to build a KNN model, you see an error reporting an “out of memory” error, try increasing the maximum system memory used by IBM SPSS Modeler. To do so, choose

and enter the new size in the **Maximum memory** field. Changes made in the System Options dialog do not take effect until you restart IBM SPSS Modeler.

## KNN Node Settings

The Settings tab is where you specify the options that are specific to Nearest Neighbor Analysis. The sidebar on the left of the screen lists the panels that you use to specify the options.

### Model

The Model panel provides options that control how the model is to be built, for example, whether to use partitioning or split models, whether to transform numeric input fields so that they all fall within the same range, and how to manage cases of interest. You can also choose a custom name for the model.

**Note:** The **Use partitioned data** and **Use case labels** can not use the same field.

**Model name.** You can generate the model name automatically based on the target or ID field (or model type in cases where no such field is specified) or specify a custom name.

**Use partitioned data.** If a partition field is defined, this option ensures that data from only the training partition is used to build the model.

**Create split models.** Builds a separate model for each possible value of input fields that are specified as split fields. See [“Building Split Models”](#) on page 24 for more information.

**To select fields manually...** By default, the node uses the partition and split field settings (if any) from the Type node, but you can override those settings here. To activate the **Partition** and **Splits** fields, select the **Fields** tab and choose **Use Custom Settings**, then return here.

- **Partition.** This field allows you to specify a field used to partition the data into separate samples for the training, testing, and validation stages of model building. By using one sample to generate the model and a different sample to test it, you can get a good indication of how well the model will generalize to larger datasets that are similar to the current data. If multiple partition fields have been defined by using Type or Partition nodes, a single partition field must be selected on the Fields tab in each modeling node that uses partitioning. (If only one partition is present, it is automatically used whenever partitioning is enabled.) Also note that to apply the selected partition in your analysis, partitioning must also be enabled in the Model Options tab for the node. (Deselecting this option makes it possible to disable partitioning without changing field settings.)
- **Splits.** For split models, select the split field or fields. This is similar to setting the field role to *Split* in a Type node. You can designate only fields of type **Flag**, **Nominal** or **Ordinal** as split fields. Fields chosen as split fields cannot be used as target, input, partition, frequency or weight fields. See the topic [“Building Split Models”](#) on page 24 for more information.

**Normalize range inputs.** Check this box to normalize the values for continuous input fields. Normalized features have the same range of values, which can improve the performance of the estimation algorithm. Adjusted normalization,  $[2 \cdot (x - \min) / (\max - \min)] - 1$ , is used. Adjusted normalized values fall between -1 and 1.

**Use case labels.** Check this box to enable the drop-down list, from where you can choose a field whose values will be used as labels to identify the cases of interest in the predictor space chart, peers chart, and quadrant map in the model viewer. You can choose any field with a measurement level of *Nominal*, *Ordinal*, or *Flag* to use as the labeling field. If you do not choose a field here, records are displayed in the model viewer charts with nearest neighbors being identified by row number in the source data. If you will be manipulating the data at all after building the model, use case labels to avoid having to refer back to the source data each time to identify the cases in the display.

**Identify focal record.** Check this box to enable the drop-down list, which allows you to mark an input field of particular interest (for flag fields only). If you specify a field here, the points representing that field are initially selected in the model viewer when the model is built. Selecting a focal record here is optional; any point can temporarily become a focal record when selected manually in the model viewer.

## Neighbors

The Neighbors panel has a set of options that control how the number of nearest neighbors is calculated.

**Number of Nearest Neighbors (k).** Specify the number of nearest neighbors for a particular case. Note that using a greater number of neighbors will not necessarily result in a more accurate model.

If the objective is to predict a target, you have two choices:

- **Specify fixed k.** Use this option if you want to specify a fixed number of nearest neighbors to find.
- **Automatically select k.** You can alternatively use the **Minimum** and **Maximum** fields to specify a range of values and allow the procedure to choose the "best" number of neighbors within that range. The method for determining the number of nearest neighbors depends upon whether feature selection is requested on the Feature Selection panel:

If feature selection is in effect, then feature selection is performed for each value of  $k$  in the requested range, and the  $k$ , and accompanying feature set, with the lowest error rate (or the lowest sum-of-squares error if the target is continuous) is selected.

If feature selection is not in effect, then  $V$ -fold cross-validation is used to select the "best" number of neighbors. See the Cross-validation panel for control over assignment of folds.

**Distance Computation.** This is the metric used to specify the distance metric used to measure the similarity of cases.

- **Euclidean metric.** The distance between two cases,  $x$  and  $y$ , is the square root of the sum, over all dimensions, of the squared differences between the values for the cases.
- **City Block metric.** The distance between two cases is the sum, over all dimensions, of the absolute differences between the values for the cases. Also called Manhattan distance.

Optionally, if the objective is to predict a target, you can choose to weight features by their normalized importance when computing distances. Feature importance for a predictor is calculated by the ratio of the error rate or sum-of-squares error of the model with the predictor removed from the model, to the error rate or sum-of-squares error for the full model. Normalized importance is calculated by reweighting the feature importance values so that they sum to 1.

**Weight features by importance when computing distances.** (Displayed only if the objective is to predict a target.) Check this box to cause predictor importance to be used when calculating the distances between neighbors. Predictor importance will then be displayed in the model nugget, and used in predictions (and so will affect scoring). See the topic ["Predictor Importance" on page 37](#) for more information.

**Predictions for Range Target.** (Displayed only if the objective is to predict a target.) If a continuous (numeric range) target is specified, this defines whether the predicted value is computed based upon the mean or the median value of the nearest neighbors.

## Feature Selection

This panel is activated only if the objective is to predict a target. It allows you to request and specify options for feature selection. By default, all features are considered for feature selection, but you can optionally select a subset of features to force into the model.

**Perform feature selection.** Check this box to enable the feature selection options.

- **Forced entry.** Click the field chooser button next to this box and choose one or more features to force into the model.

**Stopping Criterion.** At each step, the feature whose addition to the model results in the smallest error (computed as the error rate for a categorical target and sum of squares error for a continuous target) is considered for inclusion in the model set. Forward selection continues until the specified condition is met.

- **Stop when the specified number of features have been selected.** The algorithm adds a fixed number of features in addition to those forced into the model. Specify a positive integer. Decreasing values of the number to select creates a more parsimonious model, at the risk of missing important features.

Increasing values of the number to select will capture all the important features, at the risk of eventually adding features that actually increase the model error.

- **Stop when the change in the absolute error ratio is less than or equal to the minimum.** The algorithm stops when the change in the absolute error ratio indicates that the model cannot be further improved by adding more features. Specify a positive number. Decreasing values of the minimum change will tend to include more features, at the risk of including features that do not add much value to the model. Increasing the value of the minimum change will tend to exclude more features, at the risk of losing features that are important to the model. The “optimal” value of the minimum change will depend upon your data and application. See the Feature Selection Error Log in the output to help you assess which features are most important. See the topic [“Predictor selection error log”](#) on page 320 for more information.

### Cross-Validation

This panel is activated only if the objective is to predict a target. The options on this panel control whether to use cross-validation when calculating the nearest neighbors.

Cross-validation divides the sample into a number of subsamples, or **folds**. Nearest neighbor models are then generated, excluding the data from each subsample in turn. The first model is based on all of the cases except those in the first sample fold, the second model is based on all of the cases except those in the second sample fold, and so on. For each model, the error is estimated by applying the model to the subsample excluded in generating it. The “best” number of nearest neighbors is the one which produces the lowest error across folds.

**Cross-Validation Folds.** V-fold cross-validation is used to determine the “best” number of neighbors. It is not available in conjunction with feature selection for performance reasons.

- **Randomly assign cases to folds.** Specify the number of folds that should be used for cross-validation. The procedure randomly assigns cases to folds, numbered from 1 to V, the number of folds.
- **Set random seed.** When estimating the accuracy of a model based on a random percentage, this option allows you to duplicate the same results in another session. By specifying the starting value used by the random number generator, you can ensure the same records are assigned each time the node is executed. Enter the desired seed value. If this option is not selected, a different sample will be generated each time the node is executed.
- **Use field to assign cases.** Specify a numeric field that assigns each case in the active dataset to a fold. The field must be numeric and take values from 1 to V. If any values in this range are missing, and on any split fields if split models are in effect, this will cause an error.

### Analyze

The Analyze panel is activated only if the objective is to predict a target. You can use it to specify whether the model is to include additional variables to contain:

- probabilities for each possible target field value
- distances between a case and its nearest neighbors
- raw and adjusted propensity scores (for flag targets only)

**Append all probabilities.** If this option is checked, probabilities for each possible value of a nominal or flag target field are displayed for each record processed by the node. If this option is unchecked, only the predicted value and its probability are displayed for nominal or flag target fields.

**Save distances between cases and k nearest neighbors.** For each focal record, a separate variable is created for each of the focal record’s k nearest neighbors (from the training sample) and the corresponding k nearest distances.

### Propensity Scores

Propensity scores can be enabled in the modeling node, and on the Settings tab in the model nugget. This functionality is available only when the selected target is a flag field. See the topic [“Propensity Scores”](#) on page 30 for more information.



**Calculate raw propensity scores.** Raw propensity scores are derived from the model based on the training data only. If the model predicts the *true* value (will respond), then the propensity is the same as P, where P is the probability of the prediction. If the model predicts the false value, then the propensity is calculated as (1 – P).

- If you choose this option when building the model, propensity scores will be enabled in the model nugget by default. However, you can always choose to enable raw propensity scores in the model nugget whether or not you select them in the modeling node.
- When scoring the model, raw propensity scores will be added in a field with the letters *RP* appended to the standard prefix. For example, if the predictions are in a field named *\$R-churn*, the name of the propensity score field will be *\$RRP-churn*.

**Calculate adjusted propensity scores.** Raw propensities are based purely on estimates given by the model, which may be overfitted, leading to over-optimistic estimates of propensity. Adjusted propensities attempt to compensate by looking at how the model performs on the test or validation partitions and adjusting the propensities to give a better estimate accordingly.

- This setting requires that a valid partition field is present in the stream.
- Unlike raw confidence scores, adjusted propensity scores must be calculated when building the model; otherwise, they will not be available when scoring the model nugget.
- When scoring the model, adjusted propensity scores will be added in a field with the letters *AP* appended to the standard prefix. For example, if the predictions are in a field named *\$R-churn*, the name of the propensity score field will be *\$RAP-churn*. Adjusted propensity scores are not available for logistic regression models.
- When calculating the adjusted propensity scores, the test or validation partition used for the calculation must not have been balanced. To avoid this, be sure the **Only balance training data** option is selected in any upstream Balance nodes. In addition, if a complex sample has been taken upstream this will invalidate the adjusted propensity scores.
- Adjusted propensity scores are not available for "boosted" tree and rule set models. See the topic [“Boosted C5.0 Models” on page 107](#) for more information.

## KNN Model Nugget

The KNN model creates a number of new fields, as shown in the following table. To see these fields and their values, add a Table node to the KNN model nugget and execute the Table node, or click the Preview button on the nugget.

Table 32. KNN model fields	
New field name	Description
<i>\$KNN-fieldname</i>	Predicted value of target field.
<i>\$KNNP-fieldname</i>	Probability of predicted value.
<i>\$KNNP-value</i>	Probability of each possible value of a nominal or flag field. Included only if <b>Append all probabilities</b> is checked on the Settings tab of the model nugget.
<i>\$KNN-neighbor-n</i>	The name of the <i>n</i> th nearest neighbor to the focal record. Included only if <b>Display Nearest</b> on the Settings tab of the model nugget is set to a non-zero value.
<i>\$KNN-distance-n</i>	The relative distance from the focal record of the <i>n</i> th nearest neighbor to the focal record. Included only if <b>Display Nearest</b> on the Settings tab of the model nugget is set to a non-zero value.

## Nearest Neighbor Model View

### Model View

The model view has a 2-panel window:

- The first panel displays an overview of the model called the main view.
- The second panel displays one of two types of views:

An auxiliary model view shows more information about the model, but is not focused on the model itself.

A linked view is a view that shows details about one feature of the model when the user drills down on part of the main view.

By default, the first panel shows the predictor space and the second panel shows the predictor importance chart. If the predictor importance chart is not available; that is, when **Weight features by importance** was not selected on the Neighbors panel of the Settings tab, the first available view in the View dropdown is shown.

When a view has no available information, it is omitted from the View dropdown.

### Predictor Space

The predictor space chart is an interactive graph of the predictor space (or a subspace, if there are more than 3 predictors). Each axis represents a predictor in the model, and the location of points in the chart show the values of these predictors for cases in the training and holdout partitions.

**Keys.** In addition to the predictor values, points in the plot convey other information.

- Shape indicates the partition to which a point belongs, either Training or Holdout.
- The color/shading of a point indicates the value of the target for that case; with distinct color values equal to the categories of a categorical target, and shades indicating the range of values of a continuous target. The indicated value for the training partition is the observed value; for the holdout partition, it is the predicted value. If no target is specified, this key is not shown.
- Heavier outlines indicate a case is focal. Focal records are shown linked to their  $k$  nearest neighbors.

**Controls and Interactivity.** A number of controls in the chart allow you explore the predictor space.

- You can choose which subset of predictors to show in the chart and change which predictors are represented on the dimensions.
- “Focal records” are simply points selected in the Predictor Space chart. If you specified a focal record variable, the points representing the focal records will initially be selected. However, any point can temporarily become a focal record if you select it. The “usual” controls for point selection apply; clicking on a point selects that point and deselects all others; Control-clicking on a point adds it to the set of selected points. Linked views, such as the Peers Chart, will automatically update based upon the cases selected in the Predictor Space.
- You can change the number of nearest neighbors ( $k$ ) to display for focal records.
- Hovering over a point in the chart displays a tooltip with the value of the case label, or case number if case labels are not defined, and the observed and predicted target values.
- A “Reset” button allows you to return the Predictor Space to its original state.

### *Changing the axes on the Predictor Space chart*

You can control which features are displayed on the axes of the Predictor Space chart.

### **To change the axis settings:**

1. Click the Edit Mode button (paintbrush icon) in the left-hand panel to select Edit mode for the Predictor Space.

2. Change the view (to anything) on the right-hand panel. The **Show zones** panel appears between the two main panels.
3. Click the **Show zones** check box.
4. Click any data point in the Predictor Space.
5. To replace an axis with a predictor of the same data type:
  - Drag the new predictor over the zone label (the one with the small X button) of the one you want to replace.
6. To replace an axis with a predictor of a different data type:
  - On the zone label of the predictor you want to replace, click the small X button. The predictor space changes to a two-dimensional view.
  - Drag the new predictor over the **Add dimension** zone label.
7. Click the Explore Mode button (arrowhead icon) in the left-hand panel to exit from Edit mode.

### Predictor Importance

Typically, you will want to focus your modeling efforts on the predictor fields that matter most and consider dropping or ignoring those that matter least. The predictor importance chart helps you do this by indicating the relative importance of each predictor in estimating the model. Since the values are relative, the sum of the values for all predictors on the display is 1.0. Predictor importance does not relate to model accuracy. It just relates to the importance of each predictor in making a prediction, not whether or not the prediction is accurate.

### Nearest Neighbor Distances

This table displays the  $k$  nearest neighbors and distances for focal records only. It is available if a focal record identifier is specified on the modeling node, and only displays focal records identified by this variable.

Each row of:

- The **Focal Record** column contains the value of the case labeling variable for the focal record; if case labels are not defined, this column contains the case number of the focal record.
- The  $i$ th column under the **Nearest Neighbors** group contains the value of the case labeling variable for the  $i$ th nearest neighbor of the focal record; if case labels are not defined, this column contains the case number of the  $i$ th nearest neighbor of the focal record.
- The  $i$ th column under the **Nearest Distances** group contains the distance of the  $i$ th nearest neighbor to the focal record

### Peers

This chart displays the focal cases and their  $k$  nearest neighbors on each predictor and on the target. It is available if a focal case is selected in the Predictor Space.

The Peers chart is linked to the Predictor Space in two ways.

- Cases selected (focal) in the Predictor Space are displayed in the Peers chart, along with their  $k$  nearest neighbors.
- The value of  $k$  selected in the Predictor Space is used in the Peers chart.

**Select Predictors.** Enables you to select the predictors to display in the Peers chart.

### Quadrant Map

This chart displays the focal cases and their  $k$  nearest neighbors on a scatterplot (or dotplot, depending upon the measurement level of the target) with the target on the y-axis and a scale predictor on the x-axis, paneled by predictors. It is available if there is a target and if a focal case is selected in the Predictor Space.

- Reference lines are drawn for continuous variables, at the variable means in the training partition.

**Select Predictors.** Enables you to select the predictors to display in the Quadrant Map.

#### *Predictor selection error log*

Points on the chart display the error (either the error rate or sum-of-squares error, depending upon the measurement level of the target) on the y-axis for the model with the predictor listed on the x-axis (plus all features to the left on the x-axis). This chart is available if there is a target and feature selection is in effect.

#### *Classification Table*

This table displays the cross-classification of observed versus predicted values of the target, by partition. It is available if there is a target and it is categorical (flag, nominal, or ordinal).

- The **(Missing)** row in the Holdout partition contains holdout cases with missing values on the target. These cases contribute to the Holdout Sample: Overall Percent values but not to the Percent Correct values.

#### *Error Summary*

This table is available if there is a target variable. It displays the error associated with the model; sum-of-squares for a continuous target and the error rate (100% – overall percent correct) for a categorical target.

## **KNN Model Settings**

The Settings tab enables you to specify extra fields to be displayed when viewing the results (for example by executing a Table node attached to the nugget). You can see the effect of each of these options by selecting them and clicking the Preview button--scroll to the right of the Preview output to see the extra fields.

**Append all probabilities (valid only for categorical targets).** If this option is checked, probabilities for each possible value of a nominal or flag target field are displayed for each record processed by the node. If this option is unchecked, only the predicted value and its probability are displayed for nominal or flag target fields.

The default setting of this check box is determined by the corresponding check box on the modeling node.

**Calculate raw propensity scores.** For models with a flag target (which return a yes or no prediction), you can request propensity scores that indicate the likelihood of the true outcome specified for the target field. These are in addition to other prediction and confidence values that may be generated during scoring.

**Calculate adjusted propensity scores.** Raw propensity scores are based only on the training data and may be overly optimistic due to the tendency of many models to overfit this data. Adjusted propensities attempt to compensate by evaluating model performance against a test or validation partition. This option requires that a partition field be defined in the stream and adjusted propensity scores be enabled in the modeling node before generating the model.

**Display nearest.** If you set this value to  $n$ , where  $n$  is a non-zero positive integer, the  $n$  nearest neighbors to the focal record are included in the model, together with their relative distances from the focal record.

---

## Chapter 17. Python nodes

SPSS Modeler offers nodes for using Python native algorithms. The **Python** tab on the Nodes Palette contains the following nodes you can use to run Python algorithms. These nodes are supported on Windows 64, Linux64, and Mac.



The Synthetic Minority Over-sampling Technique (SMOTE) node provides an over-sampling algorithm to deal with imbalanced data sets. It provides an advanced method for balancing data. The SMOTE process node in SPSS Modeler is implemented in Python and requires the `imbalanced-learn`® Python library.



XGBoost Linear® is an advanced implementation of a gradient boosting algorithm with a linear model as the base model. Boosting algorithms iteratively learn weak classifiers and then add them to a final strong classifier. The XGBoost Linear node in SPSS Modeler is implemented in Python.



XGBoost Tree® is an advanced implementation of a gradient boosting algorithm with a tree model as the base model. Boosting algorithms iteratively learn weak classifiers and then add them to a final strong classifier. XGBoost Tree is very flexible and provides many parameters that can be overwhelming to most users, so the XGBoost Tree node in SPSS Modeler exposes the core features and commonly used parameters. The node is implemented in Python.



t-Distributed Stochastic Neighbor Embedding (t-SNE) is a tool for visualizing high-dimensional data. It converts affinities of data points to probabilities. This t-SNE node in SPSS Modeler is implemented in Python and requires the `scikit-learn`® Python library.



A Gaussian Mixture® model is a probabilistic model that assumes all the data points are generated from a mixture of a finite number of Gaussian distributions with unknown parameters. One can think of mixture models as generalizing k-means clustering to incorporate information about the covariance structure of the data as well as the centers of the latent Gaussians. The Gaussian Mixture node in SPSS Modeler exposes the core features and commonly used parameters of the Gaussian Mixture library. The node is implemented in Python.



Kernel Density Estimation (KDE)® uses the Ball Tree or KD Tree algorithms for efficient queries, and combines concepts from unsupervised learning, feature engineering, and data modeling. Neighbor-based approaches such as KDE are some of the most popular and useful density estimation techniques. The KDE Modeling and KDE Simulation nodes in SPSS Modeler expose the core features and commonly used parameters of the KDE library. The nodes are implemented in Python.



The Random Forest node uses an advanced implementation of a bagging algorithm with a tree model as the base model. This Random Forest modeling node in SPSS Modeler is implemented in Python and requires the `scikit-learn`® Python library.



Hierarchical Density-Based Spatial Clustering (HDBSCAN)<sup>®</sup> uses unsupervised learning to find clusters, or dense regions, of a data set. The HDBSCAN node in SPSS Modeler exposes the core features and commonly used parameters of the HDBSCAN library. The node is implemented in Python, and you can use it to cluster your dataset into distinct groups when you don't know what those groups are at first.



The One-Class SVM node uses an unsupervised learning algorithm. The node can be used for novelty detection. It will detect the soft boundary of a given set of samples, to then classify new points as belonging to that set or not. This One-Class SVM modeling node in SPSS Modeler is implemented in Python and requires the `scikit-learn`<sup>®</sup> Python library.

## SMOTE node

---

The Synthetic Minority Over-sampling Technique (SMOTE) node provides an over-sampling algorithm to deal with imbalanced data sets. It provides an advanced method for balancing data. The SMOTE process node is implemented in Python and requires the `imbalanced-learn`<sup>®</sup> Python library. For details about the `imbalanced-learn` library, see <https://imbalanced-learn.org/stable/><sup>1</sup>.

The Python tab on the Nodes Palette contains the SMOTE node and other Python nodes.

<sup>1</sup>Lemaître, Nogueira, Aridas. "Imbalanced-learn: A Python Toolbox to Tackle the Curse of Imbalanced Datasets in Machine Learning." *Journal of Machine Learning Research*, vol. 18, no. 17, 2017, pp. 1-5. (<http://jmlr.org/papers/v18/16-365.html>)

### SMOTE node Settings

Define the following settings on the SMOTE node's **Settings** tab.

#### Target Setting

**Target Field.** Select the target field. All flag, nominal, ordinal, and discrete measurement types are supported. If the **Use partitioned data** option is selected in the Partition section, only training data will be over-sampled.

#### Over Sample Ratio

Select **Auto** to automatically select an over-sample ratio, or select **Set Ratio (minority over majority)** to set a custom ratio value. The ratio is the number of samples in the minority class over the number of samples in the majority class. The value must be greater than **0** and less than or equal to **1**.

#### Random Seed

**Set random seed.** Select this option and click **Generate** to generate the seed used by the random number generator.

#### Methods

**Algorithm Kind.** Select the type of SMOTE algorithm you wish to use.

#### Samples Rules

**K Neighbours.** Specify the number of the nearest neighbors to use for constructing synthetic samples

**M Neighbours.** Specify the number of nearest neighbors to use for determining if a minority sample is in danger. This will only be used if the **Borderline1** or **Borderline2** SMOTE algorithm type is selected.

## Partition

**Use partitioned data.** Select this option if you only want training data to be over-sampled.

The SMOTE node requires the `imbalanced-learn`® Python library. The following table shows the relationship between the settings in the SPSS Modeler SMOTE node dialog and the Python algorithm.

Table 33. Node properties mapped to Python library parameters		
SPSS Modeler setting	Script name (property name)	Python API parameter name
Over sample ratio (number input control)	sample_ratio_value	ratio
Random seed	random_seed	random_state
K_Neighbours	k_neighbours	k
M_Neighbours	m_neighbours	m
Algorithm kind	algorithm_kind	kind

## XGBoost Linear node

XGBoost Linear® is an advanced implementation of a gradient boosting algorithm with a linear model as the base model. Boosting algorithms iteratively learn weak classifiers and then add them to a final strong classifier. The XGBoost Linear node in SPSS Modeler is implemented in Python.

For more information about boosting algorithms, see the XGBoost Tutorials available at <http://xgboost.readthedocs.io/en/latest/tutorials/index.html>.<sup>1</sup>

Note that the XGBoost cross-validation function is not supported in SPSS Modeler. You can use the SPSS Modeler Partition node for this functionality. Also note that XGBoost in SPSS Modeler performs one-hot encoding automatically for categorical variables.

<sup>1</sup> "XGBoost Tutorials." *Scalable and Flexible Gradient Boosting*. Web. © 2015-2016 DMLC.

## XGBoost Linear node Fields

The Fields tab specifies which fields are used in the analysis.

**Use predefined roles.** This option uses the role settings (targets, predictors and so on) from an upstream Type node (or the Types tab of an upstream source node).

**Use custom field assignments.** To manually assign a target and predictors, select this option.

**Fields.** Use the arrow buttons to assign items manually from this list to the Target and Predictors role fields on the right of the screen. The icons indicate the valid measurement levels for each role field. To select all the fields in the list, click the **All** button, or click an individual measurement level button to select all fields with that measurement level.

**Target.** Select a field to use as the target for the prediction.

**Predictors.** Select one or more fields as inputs for the prediction.

## XGBoost Linear node Build Options

Use the Build Options tab to specify build options for the XGBoost Linear node, including **basic options** such as linear boost parameters and model building, and **learning task options** for objectives. For additional information about these options, see the following online resources:

- [XGBoost Parameter Reference](#)<sup>1</sup>
- [XGBoost Python API](#)<sup>2</sup>
- [XGBoost Home Page](#)<sup>3</sup>



## Basic

**Hyper-Parameter Optimization (Based on Rbfopt).** Select this option to enable Hyper-Parameter Optimization based on Rbfopt, which automatically discovers the optimal combination of parameters so that the model will achieve the expected or lower error rate on the samples. For details about Rbfopt, see [http://rbfopt.readthedocs.io/en/latest/rbfopt\\_settings.html](http://rbfopt.readthedocs.io/en/latest/rbfopt_settings.html).

**Alpha.** L1 regularization term on weights Increasing this value will make model more conservative.

**Lambda.** L2 regularization term on weights. Increasing this value will make the model more conservative.

**Lambda bias.** L2 regularization term on bias. (There is no L1 regularization term on bias because it is not important.)

**Number boost round.** The number of boosting iterations.

## Learning Task

**Objective.** Select from the following learning task objective types: **reg:linear**, **reg:logistic**, **reg:gamma**, **reg:tweedie**, **count:poisson**, **rank:pairwise**, **binary:logistic**, or **multi**.

**Random Seed.** You can click **Generate** to generate the seed used by the random number generator.

The following table shows the relationship between the settings in the SPSS Modeler XGBoost Linear node dialog and the Python XGBoost library parameters.

Table 34. Node properties mapped to Python library parameters		
SPSS Modeler setting	Script name (property name)	XGBoost parameter
Target	TargetField	
Predictors	InputFields	
Lambda	lambda	lambda
Alpha	alpha	alpha
Lambda bias	lambdaBias	lambda_bias
Num boost round	numBoostRound	num_boost_round
Objective	objectiveType	objective
Random Seed	random_seed	seed

<sup>1</sup> "XGBoost Parameters" *Scalable and Flexible Gradient Boosting*. Web. © 2015-2016 DMLC.

<sup>2</sup> "Plotting API" *Scalable and Flexible Gradient Boosting*. Web. © 2015-2016 DMLC.

<sup>3</sup> "Scalable and Flexible Gradient Boosting." Web. © 2015-2016 DMLC.

## XGBoost Linear node Model Options

**Model name.** You can generate the model name automatically based on the target or ID field (or model type in cases where no such field is specified) or specify a custom name.

## XGBoost Tree node

XGBoost Tree<sup>®</sup> is an advanced implementation of a gradient boosting algorithm with a tree model as the base model. Boosting algorithms iteratively learn weak classifiers and then add them to a final strong classifier. XGBoost Tree is very flexible and provides many parameters that can be overwhelming to most users, so the XGBoost Tree node in SPSS Modeler exposes the core features and commonly used parameters. The node is implemented in Python.

For more information about boosting algorithms, see the XGBoost Tutorials available at <http://xgboost.readthedocs.io/en/latest/tutorials/index.html>.<sup>1</sup>



Note that the XGBoost cross-validation function is not supported in SPSS Modeler. You can use the SPSS Modeler Partition node for this functionality. Also note that XGBoost in SPSS Modeler performs one-hot encoding automatically for categorical variables.

<sup>1</sup> "XGBoost Tutorials." *Scalable and Flexible Gradient Boosting*. Web. © 2015-2016 DMLC.

## XGBoost Tree node Fields

The Fields tab specifies which fields are used in the analysis.

**Use predefined roles.** This option uses the role settings (targets, predictors and so on) from an upstream Type node (or the Types tab of an upstream source node).

**Use custom field assignments.** To manually assign a target and predictors, select this option.

**Fields.** Use the arrow buttons to assign items manually from this list to the Target and Predictors role fields on the right of the screen. The icons indicate the valid measurement levels for each role field. To select all the fields in the list, click the **All** button, or click an individual measurement level button to select all fields with that measurement level.

**Target.** Select a field to use as the target for the prediction.

**Predictors.** Select one or more fields as inputs for the prediction.

## XGBoost Tree node Build Options

Use the Build Options tab to specify build options for the XGBoost Tree node, including **basic options** for model building and tree growth, **learning task options** for objectives, and **advanced options** for control overfitting and handling of imbalanced datasets. For additional information about these options, see the following online resources:

- [XGBoost Parameter Reference](#)<sup>1</sup>
- [XGBoost Python API](#)<sup>2</sup>
- [XGBoost Home Page](#)<sup>3</sup>

### Basic

**Hyper-Parameter Optimization (Based on Rbfopt).** Select this option to enable Hyper-Parameter Optimization based on Rbfopt, which automatically discovers the optimal combination of parameters so that the model will achieve the expected or lower error rate on the samples. For details about Rbfopt, see [http://rbfopt.readthedocs.io/en/latest/rbfopt\\_settings.html](http://rbfopt.readthedocs.io/en/latest/rbfopt_settings.html).

**Tree method.** Select the XGBoost tree construction algorithm to use.

**Num boost round.** Specify the number of boosting iterations.

**Max depth.** Specify the maximum depth for trees. Increasing this value will make the model more complex and likely to be overfitting.

**Min child weight.** Specify the minimum sum of instance weight (hessian) needed in a child. If the tree partition step results in a leaf node with the sum of instance weight less than this **Min child weight**, then the building process will stop further partitioning. In linear regression mode, this simply corresponds to minimum number of instances needed in each node. The larger the weight, the more conservative the algorithm will be.

**Max delta step.** Specify the maximum delta step to allow for each tree's weight estimation. If set to **0**, there is no constraint. If set to a positive value, it can help the update step be more conservative. Usually this parameter is not needed, but it may help in logistic regression when a class is extremely imbalanced.

### Learning Task

**Objective.** Select from the following learning task objective types: **reg:linear**, **reg:logistic**, **reg:gamma**, **reg:tweedie**, **count:poisson**, **rank:pairwise**, **binary:logistic**, or **multi**.

**Early stopping.** Select this option if you want to use the early stopping function. For the **stopping rounds**, validation errors must decrease at least every early stopping round(s) to continue training. The **Evaluation data ratio** is the ratio of input data used for validation errors.

**Random Seed.** You can click **Generate** to generate the seed used by the random number generator.

### Advanced

**Sub sample.** Sub sample is the ratio of the training instance. For example, if you set this to **0.5**, XGBoost will randomly collect half the data instances to grow trees and this will prevent overfitting.

**Eta.** The step size shrinkage used during the update step to prevent overfitting. After each boosting step, the weights of new features can be obtained directly. Eta also shrinks the feature weights to make the boosting process more conservative.

**Gamma.** The minimum loss reduction required to make a further partition on a leaf node of the tree. The larger the gamma setting, the more conservative the algorithm will be.

**Colsample by tree.** Sub sample ratio of columns when constructing each tree.

**Colsample by level.** Sub sample ratio of columns for each split, in each level.

**Lambda.** L2 regularization term on weights. Increasing this value will make the model more conservative.

**Alpha.** L1 regularization term on weights. Increasing this value will make model more conservative.

**Scale pos weight.** Control the balance of positive and negative weights. This is useful for unbalanced classes.

The following table shows the relationship between the settings in the SPSS Modeler XGBoost Tree node dialog and the Python XGBoost library parameters.

Table 35. Node properties mapped to Python library parameters		
SPSS Modeler setting	Script name (property name)	XGBoost parameter
Target	TargetField	
Predictors	InputFields	
Tree method	treeMethod	tree_method
Num boost round	numBoostRound	num_boost_round
Max depth	maxDepth	max_depth
Min child weight	minChildWeight	min_child_weight
Max delta step	maxDeltaStep	max_delta_step
Objective	objectiveType	objective
Early stopping	earlyStopping	early_stopping_rounds
stopping rounds	stoppingRounds	
Evaluation data ratio	evaluationDataRatio	
Random Seed	random_seed	seed
Sub sample	sampleSize	subsample
Eta	eta	eta
Gamma	gamma	gamma
Colsample by tree	colsSampleRatio	colsample_bytree
Colsample by level	colsSampleLevel	colsample_bylevel

Table 35. Node properties mapped to Python library parameters (continued)		
SPSS Modeler setting	Script name (property name)	XGBoost parameter
Lambda	lambda	lambda
Alpha	alpha	alpha
Scale pos weight	scalePosWeight	scale_pos_weight

<sup>1</sup> "XGBoost Parameters" *Scalable and Flexible Gradient Boosting*. Web. © 2015-2016 DMLC.

<sup>2</sup> "Plotting API" *Scalable and Flexible Gradient Boosting*. Web. © 2015-2016 DMLC.

<sup>3</sup> "Scalable and Flexible Gradient Boosting." Web. © 2015-2016 DMLC.

## XGBoost Tree node Model Options

**Model name.** You can generate the model name automatically based on the target or ID field (or model type in cases where no such field is specified) or specify a custom name.

## t-SNE node

t-Distributed Stochastic Neighbor Embedding (t-SNE)<sup>®</sup> is a tool for visualizing high-dimensional data. It converts affinities of data points to probabilities. The affinities in the original space are represented by Gaussian joint probabilities and the affinities in the embedded space are represented by Student's t-distributions. This allows t-SNE to be particularly sensitive to local structure and has a few other advantages over existing techniques: <sup>1</sup>

- Revealing the structure at many scales on a single map
- Revealing data that lie in multiple, different, manifolds, or clusters
- Reducing the tendency to crowd points together at the center

The t-SNE node in SPSS Modeler is implemented in Python and requires the `scikit-learn`<sup>®</sup> Python library. For details about t-SNE and the `scikit-learn` library, see:

- <https://scikit-learn.org/stable/modules/generated/sklearn.manifold.TSNE.html#sklearn.manifold.TSNE>
- <https://scikit-learn.org/stable/modules/manifold.html#t-sne>

The Python tab on the Nodes Palette contains this node and other Python nodes. The t-SNE node is also available on the Graphs tab.

<sup>1</sup> References:

van der Maaten, L.J.P.; Hinton, G. "Visualizing High-Dimensional Data using t-SNE." *Journal of Machine Learning Research*. 9:2579-2605, 2008.

van der Maaten, L.J.P. "t-Distributed Stochastic Neighbor Embedding."

van der Maaten, L.J.P. "Accelerating t-SNE using Tree-Based Algorithms." *Journal of Machine Learning Research*. 15(Oct):3221-3245, 2014.

## t-SNE node Expert options

Choose **Simple** mode or **Expert** mode depending on which options you want to set for the t-SNE node.

**Visualization type.** Select **2D** or **3D** to specify whether to draw the graph as two-dimensional or three-dimensional.

**Method.** Select **Barnes Hut** or **Exact**. By default, the gradient calculation algorithm uses Barnes-Hut approximation which runs much faster than the Exact method. Barnes-Hut approximation allows the t-SNE technique to be applied to large, real-world datasets. The Exact algorithm will do a better job of avoiding nearest-neighbor errors.

**Init.** Select **Random** or **PCA** for the initialization of embedding.

**Target Field.** Select the target field to show as a colormap on the output graph. The graph will use one color if no target field is specified here.

## Optimization

**Perplexity.** The perplexity is related to the number of nearest neighbors that are used in other manifold learning algorithms. Larger datasets usually required a larger perplexity. Consider selecting a value between **5** and **50**. Default is **30**, and the range is **2 - 9999999**.

**Early exaggeration.** This setting controls how tight natural clusters in the original space will be in the embedded space, and how much space will be between them. Default is **12**, and the range is **2 - 9999999**.

**Learning rate.** If the learning rate is too high, the data might look like a "ball" with any point approximately equidistant from its nearest neighbors. If the learning rate is too low, most points may look compressed in a dense cloud with few outliers. If the cost function gets stuck in a bad local minimum, increasing the learning rate may help. Default is **200**, and the range is **0 - 9999999**.

**Max iterations.** The maximum number of iterations for the optimization. Default is **1000**, and the range is **250 - 9999999**.

**Angular size.** The angular size of a distant node as measured from a point. Enter a value between **0** and **1**. Default is **0.5**.

## Random seed

**Set random seed.** Select this option and click **Generate** to generate the seed used by the random number generator.

## Optimization stop condition

**Max iterations without progress.** The maximum number of iterations without progress to perform before stopping the optimization, used after 250 initial iterations with early exaggeration. Note that progress is only checked every 50 iterations, so this value is rounded to the next multiple of 50. Default is **300**, and the range is **0 - 9999999**.

**Min gradient norm.** If the gradient norm is below this minimum threshold, the optimization will stop. Default is **1.0E-7**.

**Metric.** The metric to use when calculating distance between instances in a feature array. If the metric is a string, it must be one of the options allowed by `scipy.spatial.distance.pdist` for its metric parameter, or a metric listed in `pairwise.PAIRWISE_DISTANCE_FUNCTIONS`. Select one of the available metric types. Default is **euclidean**.

**When number of records greater than.** Specify a method for plotting large datasets. You can specify a maximum dataset size or use the default 2,000 points. Performance is enhanced for large datasets when you select the **Bin** or **Sample** options. Alternatively, you can choose to plot all data points by selecting **Use all data**, but you should note that this may dramatically decrease the performance of the software.

- **Bin.** Select to enable binning when the dataset contains more than the specified number of records. Binning divides the graph into fine grids before actually plotting and counts the number of connections that would appear in each of the grid cells. In the final graph, one connection is used per cell at the bin centroid (average of all connection points in the bin).
- **Sample.** Select to randomly sample the data to the specified number of records.

The following table shows the relationship between the settings on the Expert tab of the SPSS Modeler t-SNE node dialog and the Python t-SNE library parameters.

Table 36. Node properties mapped to Python library parameters		
SPSS Modeler setting	Script name (property name)	Python t-SNE parameter
Mode	mode_type	

Table 36. Node properties mapped to Python library parameters (continued)

SPSS Modeler setting	Script name (property name)	Python t-SNE parameter
Visualization type	n_components	n_components
Method	method	method
Initialization of embedding	init	init
Target	target_field	target_field
Perplexity	perplexity	perplexity
Early exaggeration	early_exaggeration	early_exaggeration
Learning rate	learning_rate	learning_rate
Max iterations	n_iter	n_iter
Angular size	angle	angle
Set random seed	enable_random_seed	
Random seed	random_seed	random_state
Max iterations without progress	n_iter_without_progress	n_iter_without_progress
Min gradient norm	min_grad_norm	min_grad_norm
Perform t-SNE with multiple perplexities	isGridSearch	

## t-SNE node Output options

Specify options for the t-SNE node output on the **Output** tab.

**Output name.** Specify the name of the output that is produced when the node runs. If you select **Auto**, the name of the output is automatically set.

**Output to screen.** Select this option to generate and display the output in a new window. The output is also added to the Output manager.

**Output to file.** Select this option to save the output to a file. Doing so enables the **File name** and **File type** fields. The t-SNE node requires access to this output file if you want to create plots using other fields for comparison purposes – or to use its output as predictors in classification or regression models. The t-SNE model creates a result file of x, y (and z) coordinate fields that is most easily accessed using a Fixed File source node.

## t-SNE model nuggets

t-SNE model nuggets contain all of the information captured by the t-SNE model. The following tabs are available.

### Graph

The **Graph** tab displays chart output for the t-SNE node. A pyplot scatter chart shows the low dimensions result. If you didn't select the **Perform t-SNE with multiple perplexities** option on the **Expert** tab of the t-SNE node, only one graph is included rather than six graphs with different perplexities.

### Text output

The **Text output** tab displays the results of the t-SNE algorithm. If you chose the **2D** visualization type on the **Expert** tab of the t-SNE node, the result here is the point value in two dimensions. If you chose **3D**, the result is the point value in three dimensions.

## Gaussian Mixture node

---

A Gaussian Mixture<sup>®</sup> model is a probabilistic model that assumes all the data points are generated from a mixture of a finite number of Gaussian distributions with unknown parameters. One can think of mixture models as generalizing k-means clustering to incorporate information about the covariance structure of the data as well as the centers of the latent Gaussians.<sup>1</sup>

The Gaussian Mixture node in SPSS Modeler exposes the core features and commonly used parameters of the Gaussian Mixture library. The node is implemented in Python.

For more information about Gaussian Mixture modeling algorithms and parameters, see the Gaussian Mixture documentation available at <http://scikit-learn.org/stable/modules/mixture.html> and <https://scikit-learn.org/stable/modules/generated/sklearn.mixture.GaussianMixture.html>.<sup>2</sup>

<sup>1</sup> "User Guide." *Gaussian mixture models*. Web. © 2007 - 2017. scikit-learn developers.

<sup>2</sup> Scikit-learn: Machine Learning in Python, Pedregosa *et al.*, JMLR 12, pp. 2825-2830, 2011.

### Gaussian Mixture node Fields

The Fields tab specifies which fields are used in the analysis.

**Use predefined roles.** This option uses the input settings from an upstream Type node (or the Types tab of an upstream source node).

**Use custom field assignments.** To manually assign inputs, select this option.

**Fields.** Use the arrow buttons to assign items manually from this list to the Predictors list on the right of the screen. The icons indicate the valid measurement levels for each field. To select all the fields in the list, click the **All** button, or click an individual measurement level button to select all fields with that measurement level.

**Predictors.** Select one or more fields as predictors.

### Gaussian Mixture node Build Options

Use the Build Options tab to specify build options for the Gaussian Mixture node, including **basic options** and **advanced options**. For details about these options not covered in this section, see the following online resources:

- [Gaussian mixture parameter reference](#)<sup>1</sup>
- [Gaussian mixture node user guide](#)<sup>2</sup>

#### Basic

**Covariance type.** Select one of the following covariance matrices:

- **Full.** Each component has its own general covariance matrix.
- **Tied.** All components share the same general covariance matrix.
- **Diag.** Each component has its own diagonal covariance matrix.
- **Spherical.** Each component has its own single variance.

**Number of components.** Specify the number of mixture components to use when building the model.

**Cluster Label.** Specify whether the cluster label is a number or a string. If you choose **String**, specify a prefix for the cluster label (for example, the default prefix is `cluster`, which results in cluster labels such as **cluster-1**, **cluster-2**, etc.).

**Random Seed.** Select this option and click **Generate** to generate the seed used by the random number generator.

#### Advanced

**Tolerance.** Specify the convergence threshold. Default value is **0.001**.

**Number of iterations.** Specify the maximum number of iterations to perform. Default value is **100**.

**Init parameters.** Select the initialization parameter **Kmeans** (responsibilities are initialized using k-means) or **Random** (responsibilities are initialized randomly).

**Warm start.** If you select **True**, the solution of the last fitting will be used as the initialization for the next fitting. This can speed up convergence when fitting is called several times on similar problems.

The following table shows the relationship between the settings in the SPSS Modeler Gaussian Mixture node dialog and the Python Gaussian Mixture library parameters.

Table 37. Node properties mapped to Python library parameters		
SPSS Modeler setting	Script name (property name)	Gaussian Mixture parameter
Use predefined roles / Use custom field assignments	role_use	
Inputs	predictors	
Use partitioned data	use_partition	
Covariance type	covariance_type	covariance_type
Number of components	number_component	n_components
Cluster Label	component_lable	
Label Prefix	label_prefix	
Set random seed	enable_random_seed	
Random Seed	random_seed	random_state
Tolerance	tol	tol
Number of iterations	max_iter	max_iter
Init parameters	init_params	init_params
Warm start	warm_start	warm_start

<sup>1</sup> Scikit-learn: Machine Learning in Python, Pedregosa *et al.*, JMLR 12, pp. 2825-2830, 2011.

<sup>2</sup> "User Guide." *Gaussian mixture models*. Web. © 2007 - 2017. scikit-learn developers.

## Gaussian Mixture node Model Options

**Model name.** You can generate the model name automatically based on the target or ID field (or model type in cases where no such field is specified) or specify a custom name.

## KDE nodes

Kernel Density Estimation (KDE)<sup>©</sup> uses the Ball Tree or KD Tree algorithms for efficient queries, and walks the line between unsupervised learning, feature engineering, and data modeling. Neighbor-based approaches such as KDE are some of the most popular and useful density estimation techniques. KDE can be performed in any number of dimensions, though in practice high dimensionality can cause a degradation of performance. The KDE Modeling and KDE Simulation nodes in SPSS Modeler expose the core features and commonly used parameters of the KDE library. The nodes are implemented in Python. <sup>1</sup>

To use a KDE node, you must set up an upstream Type node. The KDE node will read input values from the Type node (or the Types tab of an upstream source node).

The **KDE Modeling** node is available on SPSS Modeler's Modeling tab and Python tab. The KDE Modeling node generates a model nugget, and the nugget's scored values are kernel density values from the input data.

The **KDE Simulation** node is available on the Output tab and the Python tab. The KDE Simulation node generates a KDE Gen source node that can create some records that have the same distribution as the input data. The KDE Gen node includes a Settings tab where you can specify how many records the node will create (default is 1) and generate a random seed.

For more information about KDE, including examples, see the KDE documentation available at <http://scikit-learn.org/stable/modules/density.html#kernel-density-estimation>.<sup>1</sup>

<sup>1</sup> "User Guide." *Kernel Density Estimation*. Web. © 2007-2018, scikit-learn developers.

## KDE Modeling node and KDE Simulation node Fields

The Fields tab specifies which fields are used in the analysis.

**Use predefined roles.** This option uses the input settings from an upstream Type node (or the Types tab of an upstream source node).

**Use custom field assignments.** To manually assign inputs, select this option.

**Fields.** Use the arrow buttons to assign items manually from this list to the Inputs list on the right of the screen. The icons indicate the valid measurement levels for each field. To select all the fields in the list, click the **All** button, or click an individual measurement level button to select all fields with that measurement level.

**Inputs.** Select one or more fields as inputs for clustering. KDE can only deal with continuous fields.

## KDE nodes Build Options

Use the Build Options tab to specify build options for the KDE nodes, including **basic options** for kernel density parameters and cluster labels, and **advanced options** such as tolerance, leaf size, and whether to use a breadth-first approach. For additional information about these options, see the following online resources:

- [Kernel Density Estimation Python API Parameter Reference](#)<sup>1</sup>
- [Kernel Density Estimation User Guide](#)<sup>2</sup>

### Basic

**Bandwidth.** Specify the bandwidth of the kernel.

**Kernel.** Select the kernel to use. Available kernels for the KDE Modeling node are **Gaussian**, **Tophat**, **Epanechnikov**, **Eponential**, **Linear**, or **Cosine**. Available kernels for the KDE Simulation node are **Gaussian** or **Tophat**. For details about these available kernels, see the [Kernel Density Estimation User Guide](#).<sup>2</sup>

**Algorithm.** Select **Auto**, **Ball Tree** or **KD Tree** for the tree algorithm to use. For more information, see [Ball Tree](#)<sup>3</sup> and [KD Tree](#).<sup>4</sup>

**Metric.** Select a distance metric. Available metrics are **Euclidean**, **Braycurtis**, **Chebyshev**, **Canberra**, **Cityblock**, **Dice**, **Hamming**, **Infinity**, **Jaccard**, **L1**, **L2**, **Matching**, **Manhattan**, **P**, **Rogerstanimoto**, **Russellrao**, **Sokalmichener**, **Sokalsneath**, **Kulsinski**, or **Minkowski**. If you select **Minkowski**, set the **P Value** as desired.

The metrics available in this drop-down will vary depending on which algorithm you choose. Also note that the normalization of the density output is correct only for the Euclidean distance metric.

### Advanced

**Absolute Tolerance.** Specify the desired absolute tolerance of the result. A larger tolerance will generally result in faster run time. Default is **0.0**.

**Relative Tolerance.** Specify the desired relative tolerance of the result. A larger tolerance will generally result in faster run time. Default is **1E-8**.



**Leaf Size.** Specify the leaf size of the underlying tree. Default is **40**. Changing the leaf size may significantly impact performance and required memory. For more information about the Ball Tree and KD Tree algorithms, see [Ball Tree](#)<sup>3</sup> and [KD Tree](#).<sup>4</sup>

**Breadth first.** Select **True** if you want to use a breadth-first approach or **False** to use a depth-first approach.

The following table shows the relationship between the settings in the SPSS Modeler KDE node dialogs and the Python KDE library parameters.

Table 38. Node properties mapped to Python library parameters		
SPSS Modeler setting	Script name (property name)	KDE parameter
Inputs	inputs	
Bandwidth	bandwidth	bandwidth
Kernel	kernel	kernel
Algorithm	algorithm	algorithm
Metric	metric	metric
P Value	pValue	pValue
Absolute Tolerance	atol	atol
Relative Tolerance	rtol	Rtol
Leaf Size	leafSize	leafSize
Breadth first	breadthFirst	breadthFirst

<sup>1</sup> "API Reference." *sklearn.neighbors.KernelDensity*. Web. © 2007-2018, scikit-learn developers.

<sup>2</sup> "User Guide." *Kernel Density Estimation*. Web. © 2007-2018, scikit-learn developers.

<sup>3</sup> "Ball Tree." *Five balltree construction algorithms*. © 1989, Omohundro, S.M., International Computer Science Institute Technical Report.

<sup>4</sup> "K-D Tree." *Multidimensional binary search trees used for associative searching*. © 1975, Bentley, J.L., Communications of the ACM.

## KDE Modeling node and KDE Simulation node Model Options

**Model name.** You can generate the model name automatically based on the target or ID field (or model type in cases where no such field is specified) or specify a custom name.

## Random Forest node

Random Forest® is an advanced implementation of a bagging algorithm with a tree model as the base model. In random forests, each tree in the ensemble is built from a sample drawn with replacement (for example, a bootstrap sample) from the training set. When splitting a node during the construction of the tree, the split that is chosen is no longer the best split among all features. Instead, the split that is picked is the best split among a random subset of the features. Because of this randomness, the bias of the forest usually slightly increases (with respect to the bias of a single non-random tree) but, due to averaging, its variance also decreases, usually more than compensating for the increase in bias, hence yielding an overall better model.<sup>1</sup>

The Random Forest node in SPSS Modeler is implemented in Python. The Python tab on the Nodes Palette contains this node and other Python nodes.

For more information about random forest algorithms, see <https://scikit-learn.org/stable/modules/ensemble.html#forest>.

<sup>1</sup>L. Breiman, "Random Forests," *Machine Learning*, 45(1), 5-32, 2001.

## Random Forest node Fields

The Fields tab specifies which fields are used in the analysis.

**Use predefined roles.** This option uses the role settings (targets, predictors and so on) from an upstream Type node (or the Types tab of an upstream source node).

**Use custom field assignments.** To manually assign a target and predictors, select this option.

**Fields.** Use the arrow buttons to assign items manually from this list to the Target and Predictors role fields on the right of the screen. The icons indicate the valid measurement levels for each role field. To select all the fields in the list, click the **All** button, or click an individual measurement level button to select all fields with that measurement level.

**Target.** Select a field to use as the target for the prediction.

**Predictors.** Select one or more fields as inputs for the prediction.

## Random Forest node Build Options

Use the Build Options tab to specify build options for the Random Forest node, including **basic options** and **advanced options**. For more information about these options, see <https://scikit-learn.org/stable/modules/ensemble.html#forest>

### Basic

**Number of trees to build.** Select the number of trees in the forest.

**Specify max depth.** If not selected, nodes are expanded until all leaves are pure or until all leaves contain less than min\_samples\_split samples.

**Max depth.** The maximum depth of the tree.

**Minimum leaf node size.** The minimum number of samples required to be at a leaf node.

**Number of features to use for splitting.** The number of features to consider when looking for the best split:

- If auto, then  $\text{max\_features} = \sqrt{n\_features}$  for classifier and  $\text{max\_features} = \sqrt{n\_features}$  for regression.
- If sqrt, then  $\text{max\_features} = \sqrt{n\_features}$ .
- If log2, then  $\text{max\_features} = \log_2(n\_features)$ .

### Advanced

**Use bootstrap samples when building trees.** If selected, bootstrap samples are used when building trees.

**Use out-of-bag samples to estimate the generalization accuracy.** If selected, out-of-bag samples are used to estimate the generalization accuracy.

**Use extremely randomized trees.** If selected, extremely randomized trees are used instead of general random forests. In extremely randomized trees, randomness goes one step further in the way splits are computed. As in random forests, a random subset of candidate features is used, but instead of looking for the most discriminative thresholds, thresholds are drawn at random for each candidate feature and the best of these randomly-generated thresholds is picked as the splitting rule. This usually allows the variance of the model to be reduced a bit more, at the expense of a slightly greater increase in bias.<sup>1</sup>

**Replicate results.** If selected, the model building process is replicated to achieve the same scoring results.

**Random seed.** You can click **Generate** to generate the seed used by the random number generator.

**Hyper-Parameter Optimization (Based on Rbfopt).** Select this option to enable Hyper-Parameter Optimization based on Rbfopt, which automatically discovers the optimal combination of parameters so

that the model will achieve the expected or lower error rate on the samples. For details about Rbfopt, see [http://rbfopt.readthedocs.io/en/latest/rbfopt\\_settings.html](http://rbfopt.readthedocs.io/en/latest/rbfopt_settings.html).

**Target.** The objective function value (error rate of the model on the samples) you want to reach (i.e., the value of the unknown optimum). Set to an acceptable value such as 0.01.

**Max iterations.** The maximum number of iterations to try the model. Default is 1000.

**Max evaluations.** The maximum number of function evaluations in accurate mode, for trying the model. Default is 300.

The following table shows the relationship between the settings in the SPSS Modeler Random Forest node dialog and the Python Random Forest library parameters.

<i>Table 39. Node properties mapped to Python library parameters</i>		
<b>SPSS Modeler setting</b>	<b>Script name (property name)</b>	<b>Random Forest parameter</b>
Target	target	
Predictors	inputs	
Number of trees to build	n_estimators	n_estimators
Specify max depth	specify_max_depth	specify_max_depth
Max depth	max_depth	max_depth
Minimum leaf node size	min_samples_leaf	min_samples_leaf
Number of features to use for splitting	max_features	max_features
Use bootstrap samples when building trees	bootstrap	bootstrap
Use out-of-bag samples to estimate the generalization accuracy	oob_score	oob_score
Use extremely randomized trees	extreme	
Replicate results	use_random_seed	
Random seed	random_seed	random_seed
Hyper-Parameter Optimization (based on Rbfopt)	enable_hpo	
Target (for HPO)	target_objval	
Max iterations (for HPO)	max_iterations	
Max evaluations (for HPO)	max_evaluations	

<sup>1</sup>L. Breiman, "Random Forests," Machine Learning, 45(1), 5-32, 2001.

## Random Forest node Model Options

**Model name.** You can generate the model name automatically based on the target or ID field (or model type in cases where no such field is specified) or specify a custom name.

## Random Forest model nuggets

Random Forest model nuggets contain all of the information captured by the random forest model. The following sections are available.

## Model Information

This view provides key information about the model, including input fields, one-hot encoding values, and model parameters.

## Predictor Importance

This view displays a chart that indicates the relative importance of each predictor in estimating the model. For more information, see [“Predictor Importance” on page 37](#).

## HDBSCAN node

---

Hierarchical Density-Based Spatial Clustering (HDBSCAN)<sup>®</sup> uses unsupervised learning to find clusters, or dense regions, of a data set. The HDBSCAN node in SPSS Modeler exposes the core features and commonly used parameters of the HDBSCAN library. The node is implemented in Python, and you can use it to cluster your dataset into distinct groups when you don't know what those groups are at first. Unlike most learning methods in SPSS Modeler, HDBSCAN models do *not* use a target field. This type of learning, with no target field, is called *unsupervised learning*. Rather than trying to predict an outcome, HDBSCAN tries to uncover patterns in the set of input fields. Records are grouped so that records within a group or cluster tend to be similar to each other, but records in different groups are dissimilar. The HDBSCAN algorithm views clusters as areas of high density separate by areas of low density. Due to this rather generic view, clusters found by HDBSCAN can be any shape, as opposed to k-means which assumes that clusters are convex shaped. Outlier points that lie alone in low-density regions are also marked. HDBSCAN also supports scoring of new samples.<sup>1</sup>

To use the HDBSCAN node, you must set up an upstream Type node. The HDBSCAN node will read input values from the Type node (or the Types tab of an upstream source node).

For more information about HDBSCAN clustering algorithms, see the HDBSCAN documentation available at <http://hdbscan.readthedocs.io/en/latest/>.<sup>1</sup>

<sup>1</sup> "User Guide / Tutorial." *The hdbscan Clustering Library*. Web. © 2016, Leland McInnes, John Healy, Steve Astels.

## HDBSCAN node Fields

The Fields tab specifies which fields are used in the analysis.

**Important:** To train an HDBSCAN model, you must use one or more fields with the role set to **Input**. Fields with the role set to **Output**, **Both**, or **None** are ignored.

**Use predefined roles.** This option uses the input settings from an upstream Type node (or the Types tab of an upstream source node).

**Use custom field assignments.** To manually assign inputs, select this option.

**Fields.** Use the arrow buttons to assign items manually from this list to the Inputs list on the right of the screen. The icons indicate the valid measurement levels for each field. To select all the fields in the list, click the **All** button, or click an individual measurement level button to select all fields with that measurement level.

**Inputs.** Select one or more fields as inputs for clustering.

## HDBSCAN node Build Options

Use the Build Options tab to specify build options for the HDBSCAN node, including **basic options** for cluster parameters and cluster labels, and **advanced options** for advanced parameters and chart output options. For additional information about these options, see the following online resources:

- [HDBSCAN Python API Parameter Reference](#)<sup>1</sup>
- [HDBSCAN Home Page](#)<sup>2</sup>

## Basic

**Hyper-Parameter Optimization (Based on Rbfopt).** Select this option to enable Hyper-Parameter Optimization based on Rbfopt, which automatically discovers the optimal combination of parameters so that the model will achieve the expected or lower error rate on the samples. For details about Rbfopt, see [http://rbfopt.readthedocs.io/en/latest/rbfopt\\_settings.html](http://rbfopt.readthedocs.io/en/latest/rbfopt_settings.html).

**Min Cluster Size.** Specify the minimum size of clusters. Single linkage splits that contain fewer points than the value specified here will be considered points "falling out" of a cluster rather than a cluster splitting into two new clusters.

**Min Samples.** Specify the minimum number of samples in a neighborhood for a point to be considered a core point. If set to **0**, the default value is the minimum cluster size value.

**Algorithm.** Select the algorithm to use. HDBSCAN has variants that are specialized for different characteristics of the data. By default, **BEST** is used - which automatically chooses the best algorithm given the nature of the data. For details about these algorithm types, see the [HDBSCAN documentation](#).<sup>1</sup> Note that the algorithm you choose will impact performance. For example, for large data we recommend trying Boruvka KDTree or Boruvka BallTree.

**Metric for Distance.** Select the metric to use when calculating distance between instances in a feature array.

**Cluster Label.** Specify whether the cluster label is a number or a string. If you choose **String**, specify a prefix for the cluster label (for example, the default prefix is `cluster`, which results in cluster labels such as **cluster-1**, **cluster-2**, etc.).

## Advanced

**Approximate Minimum Spanning Tree.** Select **True** if you want to accept an approximate minimum spanning tree. For some algorithms, this can improve performance, but the resulting clustering might be of marginally lower quality. If you are willing to sacrifice speed for correctness, you may want to try the **False** option. In most cases, **True** is recommended.

**Method to Select Cluster.** Select which method to use for selecting clusters from the condensed tree. The standard approach for HDBSCAN is to use an Excess of Mass (**EOM**) algorithm to find the most persistent clusters. Or you can select the clusters at the leaves of the tree, which provides the most fine-grained and homogeneous clusters.

**Accept Single Cluster.** Change this setting to **True** to allow single cluster results only if this is a valid result for your dataset.

**P Value.** If using the Minkowski metric for distance (under **Basic** build options), you can change this p value if desired.

**Leaf Size.** If using a space tree algorithm (Boruvka KDTree or Boruvka BallTree), this is the number of points in a leaf node of the tree. This setting doesn't alter the resulting clustering, but it may impact the run time of the algorithm.

**Validity Index.** Select this option to include the Validity Index chart in the model nugget output.

**Condensed Tree.** Select this option to include the Condensed Tree chart in the model nugget output.

**Single Linkage Tree.** Select this option to include the Single Linkage Tree chart in the model nugget output.

**Min Span Tree.** Select this option to include the Min Span Tree chart in the model nugget output.

The following table shows the relationship between the settings in the SPSS Modeler HDBSCAN node dialog and the Python HDBSCAN library parameters.

Table 40. Node properties mapped to Python library parameters		
SPSS Modeler setting	Script name (property name)	HDBSCAN parameter
Inputs	inputs	inputs

Table 40. Node properties mapped to Python library parameters (continued)

SPSS Modeler setting	Script name (property name)	HDBSCAN parameter
Hyper-Parameter Optimization	useHPO	
Min Cluster Size	min_cluster_size	min_cluster_size
Min Samples	min_samples	min_samples
Algorithm	algorithm	algorithm
Metric for Distance	metric	metric
Cluster Label	useStringLabel	
Label Prefix	stringLabelPrefix	
Approximate Minimum Spanning Tree	approx_min_span_tree	approx_min_span_tree
Method to Select Cluster	cluster_selection_method	cluster_selection_method
Accept Single Cluster	allow_single_cluster	allow_single_cluster
P Value	p_value	p_value
Leaf Size	leaf_size	leaf_size
Validity Index	outputValidity	
Condensed Tree	outputCondensed	
Single Linkage Tree	outputSingleLinkage	
Min Span Tree	outputMinSpan	

<sup>1</sup> "API Reference." *The hdbscan Clustering Library*. Web. © 2016, Leland McInnes, John Healy, Steve Astels.

<sup>2</sup> "User Guide / Tutorial." *The hdbscan Clustering Library*. Web. © 2016, Leland McInnes, John Healy, Steve Astels.

## HDBSCAN node Model Options

**Model name.** You can generate the model name automatically based on the target or ID field (or model type in cases where no such field is specified) or specify a custom name.

## One-Class SVM node

The One-Class SVM<sup>®</sup> node uses an unsupervised learning algorithm. The node can be used for novelty detection. It will detect the soft boundary of a given set of samples, to then classify new points as belonging to that set or not. This One-Class SVM modeling node is implemented in Python and requires the *scikit-learn*<sup>®</sup> Python library. For details about the *scikit-learn* library, see <http://contrib.scikit-learn.org/imbalanced-learn/about.html><sup>1</sup>.

The Python tab on the Nodes Palette contains the One-Class SVM node and other Python nodes.

**Note:** One-Class SVM is used for unsupervised outlier and novelty detection. In most cases, we recommend using a known, "normal" dataset to build the model so the algorithm can set a correct boundary for the given samples. Parameters for the model – such as *nu*, *gamma*, and *kernel* – impact the result significantly. So you may need to experiment with these options until you find the optimal settings for your situation.

<sup>1</sup>Smola, Schölkopf. "A Tutorial on Support Vector Regression." *Statistics and Computing Archive*, vol. 14, no. 3, August 2004, pp. 199-222. (<http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.114.4288>)

## One-Class SVM node Fields

The Fields tab specifies which fields are used in the analysis.

**Use predefined roles.** Select this option to select all fields with a defined role of Input.

**Use custom field assignments.** To manually select fields, select this option and choose input fields and split fields:

**Inputs.** Select the input fields to use in the analysis. All storage types and measurement types are supported, except for typeless or unknown. If a field has a String storage type, the values of this field will be binarized in a one-vs-all fashion via a one-hot encoding algorithm.

**Split.** Select which field or fields to use as split fields. All flag, nominal, ordinal, and discrete measurement types are supported.

**Use partitioned data** If a partition field is defined, this option ensures that data from only the training partition is used to build the model.

## One-Class SVM node Expert

On the Expert tab of the One-Class SVM node, you can choose from **Simple** mode or **Expert** mode. If you choose **Simple**, all parameters are set with the default values as shown below. If you select **Expert**, you can specify custom values for these parameters. For further detail about these options, see <http://scikit-learn.org/stable/modules/generated/sklearn.svm.OneClassSVM.html#sklearn.svm.OneClassSVM>.

**Stopping criteria.** Specify the tolerance for stopping criteria. Default is **1.0E-3** (0.001).

**Regression precision (nu).** Bound on the fraction of training errors and support vectors. Default is **0.1**.

**Kernel type.** The kernel type to use in the algorithm. Options include **RBF**, **Polynomial**, **Sigmoid**, **Linear**, or **Precomputed**. Default is **RBF**.

**Specify Gamma.** Select this option to specify the Gamma. Otherwise, the auto gamma will be applied.

**Gamma.** The Gamma setting is only available for the RBF, Polynomial, and Sigmoid kernel types.

**Coef0.** Coef0 is only available for the Polynomial and Sigmoid kernel types.

**Degree.** Degree is only available for the Polynomial kernel type.

**Use the shrinking heuristic.** Select this option to use the shrinking heuristic. This option is deselected by default.

**Specify the size of the kernel cache (in MB).** Select this option to specify the size of the kernel cache. This option is deselected by default. When selected, the default value is **200 MB**.

**Hyper-Parameter Optimization (Based on Rbfopt).** Select this option to enable Hyper-Parameter Optimization based on Rbfopt, which automatically discovers the optimal combination of parameters so that the model will achieve the expected or lower error rate on the samples. For details about Rbfopt, see [http://rbfopt.readthedocs.io/en/latest/rbfopt\\_settings.html](http://rbfopt.readthedocs.io/en/latest/rbfopt_settings.html).

**Target.** The objective function value (error rate of the model on the samples) we want to reach (for example, the value of the unknown optimum). Set to an acceptable value such as **0.01**.

**Max Iterations.** Maximum number of iterations for trying the model. Default is **1000**.

**Max Evaluations.** Maximum number of function evaluations for trying the model, where the focus is accuracy over speed. Default is **300**.

The One-Class SVM node requires the `scikit-learn`® Python library. The following table shows the relationship between the settings in the SPSS Modeler SMOTE node dialog and the Python algorithm.

Table 41. Node properties mapped to Python library parameters		
Parameter name	Script name (property name)	Python API parameter name
Stopping criteria	stopping_criteria	tol

Table 41. Node properties mapped to Python library parameters (continued)

Parameter name	Script name (property name)	Python API parameter name
Regression precision	precision	nu
Kernel type	kernel	kernel
Gamma	gamma	gamma
Coef0	coef0	coef0
Degree	degree	degree
Use the shrinking heuristic	shrinking	shrinking
Specify the size of the kernel cache (number input box)	cache_size	cache_size
Random seed	random_seed	random_state

## One-Class SVM node Options

On the Options tab of the One-Class SVM node, you can set the following options.

**Type of Parallel Coordinates graphic.** SPSS Modeler draws parallel coordinates graphics to present the built model. Sometimes, the values for some data columns/features will be displayed much larger than others, which can make some other portions of the graph hard to see. For cases like this, you can choose the **Independent vertical axes** option to give all vertical axes a standalone axis scale, or select **General vertical axes** to force all vertical axes to share the same axes scale.

**Maximum lines on the graphic.** Specify the maximum number of data rows (lines) to display in graph output. The default is 100. For performance reasons, a maximum of 20 fields will be displayed.

**Draw all input fields on the graphic.** Select this option to show all input fields in the graph output. By default, each data field will be drawn as a vertical axis. For performance reasons, a maximum of 30 fields will be displayed.

**Custom fields to be drawn on the graphic.** Rather than showing all input fields in the graph output, you can select this option and choose a subset of fields to show. This can improve performance. For performance reasons, a maximum of 20 fields will be displayed.



---

## Chapter 18. Spark nodes

SPSS Modeler offers nodes for using Spark native algorithms. The **Spark** tab on the Nodes Palette contains the following nodes you can use to run Spark algorithms. These nodes are supported on Windows 64, Mac 64, and Linux 64. Note that these nodes don't support specifying an integer/double column as Flag/Nominal for building a model. To do this, you must convert the column value to 0/1 or 0,1,2,3,4...



Isotonic Regression belongs to the family of regression algorithms. The Isotonic-AS node in SPSS Modeler is implemented in Spark. For details about Isotonic Regression algorithms, see <https://spark.apache.org/docs/2.2.0/mllib-isotonic-regression.html>.



XGBoost® is an advanced implementation of a gradient boosting algorithm. Boosting algorithms iteratively learn weak classifiers and then add them to a final strong classifier. XGBoost is very flexible and provides many parameters that can be overwhelming to most users, so the XGBoost-AS node in SPSS Modeler exposes the core features and commonly used parameters. The XGBoost-AS node is implemented in Spark.



K-Means is one of the most commonly used clustering algorithms. It clusters data points into a predefined number of clusters. The K-Means-AS node in SPSS Modeler is implemented in Spark. For details about K-Means algorithms, see <https://spark.apache.org/docs/2.2.0/ml-clustering.html>. Note that the K-Means-AS node performs one-hot encoding automatically for categorical variables.



Multilayer perceptron is a classifier based on the feedforward artificial neural network and consists of multiple layers. Each layer is fully connected to the next layer in the network. The MultiLayerPerceptron-AS node in SPSS Modeler is implemented in Spark. For details about the multilayer perceptron classifier (MLPC), see <https://spark.apache.org/docs/latest/ml-classification-regression.html#multilayer-perceptron-classifier>.

---

### Isotonic-AS node

Isotonic Regression belongs to the family of regression algorithms. The Isotonic-AS node in SPSS Modeler is implemented in Spark.

For details about Isotonic Regression algorithms, see <https://spark.apache.org/docs/2.2.0/mllib-isotonic-regression.html>.<sup>1</sup>

<sup>1</sup> "Regression - RDD-based API." *Apache Spark*. MLib: Main Guide. Web. 3 Oct 2017.

### Isotonic-AS node Fields

The Fields tab specifies which fields are used in the analysis.

**Fields.** Lists all fields in the data source. Use the arrow buttons to assign items manually from this list to the Target, Input, and Weight fields on the right of the screen. The icons indicate the valid measurement levels for each role field. To select all the fields in the list, click the **All** button, or click an individual measurement level button to select all fields with that measurement level.

**Target.** Select a field to use as the target.

**Input.** Select the input field or fields.

**Weight.** Select a weight field for exponential weight. If not set, the default weight value of **1** will be used.

## Isotonic-AS node Build Options

Use the Build Options tab to specify build options for the Isotonic-AS node, including the feature index and isotonic type. For more information, see <http://spark.apache.org/docs/latest/api/java/org/apache/spark/ml/regression/IsotonicRegression.html>.<sup>1</sup>

**Input Fields Index.** Specify the index of the input fields. Default is **0**.

**Isotonic Type.** This setting determines whether the output sequence should be isotonic/increasing or antitonic/decreasing. Default is **Isotonic**.

<sup>1</sup> "Class IsotonicRegression." *Apache Spark*. JavaDoc. Web. 3 Oct 2017.

## Isotonic-AS model nuggets

Isotonic-AS model nuggets contain all of the information captured by the isotonic regression model. The following sections are available.

### Model Summary

This view provides key information about the model, including input fields, target field, and model building options.

### Model Chart

This view displays a scatter diagram.

## XGBoost-AS node

---

XGBoost<sup>®</sup> is an advanced implementation of a gradient boosting algorithm. Boosting algorithms iteratively learn weak classifiers and then add them to a final strong classifier. XGBoost is very flexible and provides many parameters that can be overwhelming to most users, so the XGBoost-AS node in SPSS Modeler exposes the core features and commonly used parameters. The XGBoost-AS node is implemented in Spark.

For more information about boosting algorithms, see the XGBoost Tutorials available at <http://xgboost.readthedocs.io/en/latest/tutorials/index.html>.<sup>1</sup>

Note that the XGBoost cross-validation function is not supported in SPSS Modeler. You can use the SPSS Modeler Partition node for this functionality. Also note that XGBoost in SPSS Modeler performs one-hot encoding automatically for categorical variables.

**Note:** On Mac, version 10.12.3 or higher is required for building XGBoost-AS models.

<sup>1</sup> "XGBoost Tutorials." *Scalable and Flexible Gradient Boosting*. Web. © 2015-2016 DMLC.

## XGBoost-AS node Fields

The Fields tab specifies which fields are used in the analysis.

**Use predefined roles.** This option uses the role settings (targets, predictors and so on) from an upstream Type node (or the Types tab of an upstream source node).

**Use custom field assignments.** To manually assign a target and predictors, select this option.

**Fields.** Use the arrow buttons to assign items manually from this list to the Target and Predictors role fields on the right of the screen. The icons indicate the valid measurement levels for each role field. To select all the fields in the list, click the **All** button, or click an individual measurement level button to select all fields with that measurement level.

**Target.** Select a field to use as the target for the prediction.

**Predictors.** Select one or more fields as inputs for the prediction.

## XGBoost-AS node Build Options

Use the Build Options tab to specify build options for the XGBoost-AS node, including **general options** for model building and handling imbalanced datasets, **learning task options** for objectives and evaluation metrics, and **booster parameters** for specific boosters. For more information about these options, see the following online resources:

- [XGBoost Home Page](#)<sup>1</sup>
- [XGBoost Parameter Reference](#)<sup>2</sup>
- [XGBoost Spark API](#)<sup>3</sup>

### General

**Number of Workers.** Number of workers used to train the XGBoost model.

**Number of Threads.** Number of threads used per worker.

**Use External Memory.** Whether to use external memory as cache.

**Booster Type.** The booster to use (**gbtree**, **gblinear**, or **dart**).

**Booster Rounds Number.** The number of rounds for boosting.

**Scale pos weight.** This setting controls the balance of positive and negative weights, and is useful for unbalanced classes.

**Random Seed.** Click **Generate** to generate the seed used by the random number generator.

### Learning Task

**Objective.** Select from the following learning task objective types: **reg:linear**, **reg:logistic**, **reg:gamma**, **reg:tweedie**, **rank:pairwise**, **binary:logistic**, or **multi**.

**Evaluation Metrics.** Evaluation metrics for validation data. A default metric will be assigned according to the objective (**rmse** for regression, **error** for classification, or **mean average precision** for ranking). Available options are **rmse**, **mae**, **logloss**, **error**, **merror**, **mlogloss**, **uac**, **ndcg**, **map**, or **gamma-deviance** (default is **rmse**).

### Booster Parameters

**Lambda.** L2 regularization term on weights. Increasing this value will make the model more conservative.

**Alpha.** L1 regularization term on weights. Increasing this value will make model more conservative.

**Lambda bias.** L2 regularization term on bias. (There is no L1 regularization term on bias because it is not important.)

**Tree method.** Select the XGBoost tree construction algorithm to use.

**Max depth.** Specify the maximum depth for trees. Increasing this value will make the model more complex and likely to be overfitting.

**Min child weight.** Specify the minimum sum of instance weight (hessian) needed in a child. If the tree partition step results in a leaf node with the sum of instance weight less than this **Min child weight**, then the building process will stop further partitioning. In linear regression mode, this simply corresponds to minimum number of instances needed in each node. The larger the weight, the more conservative the algorithm will be.

**Max delta step.** Specify the maximum delta step to allow for each tree's weight estimation. If set to **0**, there is no constraint. If set to a positive value, it can help the update step be more conservative. Usually this parameter is not needed, but it may help in logistic regression when a class is extremely imbalanced.

**Sub sample.** Sub sample is the ratio of the training instance. For example, if you set this to **0.5**, XGBoost will randomly collect half the data instances to grow trees and this will prevent overfitting.

**Eta.** The step size shrinkage used during the update step to prevent overfitting. After each boosting step, the weights of new features can be obtained directly. Eta also shrinks the feature weights to make the boosting process more conservative.

**Gamma.** The minimum loss reduction required to make a further partition on a leaf node of the tree. The larger the gamma setting, the more conservative the algorithm will be.

**Colsample by tree.** Sub sample ratio of columns when constructing each tree.

**Colsample by level.** Sub sample ratio of columns for each split, in each level.

**Normalization Algorithm.** The normalization algorithm to use when the dart booster type is selected under General options. Available options are **tree** or **forest** (default is **tree**).

**Sampling Algorithm.** The sampling algorithm to use when the dart booster type is selected under General options. The **uniform** algorithm uniformly selects dropped trees. The **weighted** algorithm selects dropped trees in proportion to weight. The default is **uniform**.

**Dropout Rate.** The dropout rate to use when the dart booster type is selected under General options.

**Probability of Skip Dropout.** The skip dropout probability to use when the dart booster type is selected under General options. If a dropout is skipped, new trees are added in the same manner as **gbtree**.

The following table shows the relationship between the settings in the SPSS Modeler XGBoost-AS node dialog and the XGBoost Spark parameters.

Table 42. Node properties mapped to Spark parameters		
SPSS Modeler setting	Script name (property name)	XGBoost Spark parameter
Target	target_fields	
Predictors	input_fields	
Lambda	lambda	lambda
Number of Workers	nWorkers	nWorkers
Number of Threads	numThreadPerTask	numThreadPerTask
Use External Memory	useExternalMemory	useExternalMemory
Booster Type	boosterType	boosterType
Boosting Round Number	numBoostRound	round
Scale Pos Weight	scalePosWeight	scalePosWeight
Objective	objectiveType	objective
Evaluation Metrics	evalMetric	evalMetric
Lambda	lambda	lambda
Alpha	alpha	alpha
Lambda bias	lambdaBias	lambdaBias
Tree Method	treeMethod	treeMethod
Max Depth	maxDepth	maxDepth
Min child weight	minChildWeight	minChildWeight
Max delta step	maxDeltaStep	maxDeltaStep
Sub sample	sampleSize	sampleSize
Eta	eta	eta
Gamma	gamma	gamma

Table 42. Node properties mapped to Spark parameters (continued)		
SPSS Modeler setting	Script name (property name)	XGBoost Spark parameter
Colsample by tree	colsSampleRation	colSampleByTree
Colsample by level	colsSampleLevel	colsSampleLevel
Normalization Algorithm	normalizeType	normalizeType
Sampling Algorithm	sampleType	sampleType
Dropout Rate	rateDrop	rateDrop
Probability of Skip Dropout	skipDrop	skipDrop

<sup>1</sup> "Scalable and Flexible Gradient Boosting." Web. © 2015-2016 DMLC.

<sup>2</sup> "XGBoost Parameters" *Scalable and Flexible Gradient Boosting*. Web. © 2015-2016 DMLC.

<sup>3</sup> "ml.dmlc.xgboost4j.scala.spark Params." *DMLC for Scalable and Reliable Machine Learning*. Web. 3 Oct 2017.

## XGBoost-AS node Model Options

**Model name.** You can generate the model name automatically based on the target or ID field (or model type in cases where no such field is specified) or specify a custom name.

## K-Means-AS node

K-Means is one of the most commonly used clustering algorithms. It clusters data points into a predefined number of clusters.<sup>1</sup> The K-Means-AS node in SPSS Modeler is implemented in Spark.

For details about K-Means algorithms, see <https://spark.apache.org/docs/2.2.0/ml-clustering.html>.

Note that the K-Means-AS node performs one-hot encoding automatically for categorical variables.

<sup>1</sup> "Clustering." *Apache Spark*. MLib: Main Guide. Web. 3 Oct 2017.

## K-Means-AS node Fields

The Fields tab specifies which fields are used in the analysis.

**Use predefined roles.** This option tells the node to use field information from an upstream Type node. It's selected by default.

**Use custom field assignments.** If you want to manually assign input fields, select this option and then select the input field or fields. Using this option is similar to setting the field role in **Input** in a Type node.

## K-Means-AS node Build Options

Use the Build Options tab to specify build options for the K-Means-AS node, including regular options for model building, initialization options for initializing cluster centers, and advanced options for the computing iteration and random seed. For more information, see the [JavaDoc for K-Means on SparkML](#).<sup>1</sup>

### Regular

**Model Name.** The name of the field generated after scoring to a specific cluster. Select **Auto** (default) or select **Custom** and type a name.

**Number of Clusters.** Specify the number of clusters to generate. The default is **5** and the minimum is **2**.

## Initialization

**Initialization Mode.** Specify the method for initializing the cluster centers. **K-Means||** is the default. For details about these two methods, see [Scalable K-Means++](#).<sup>2</sup>

**Initialization Steps.** If the **K-Means||** initialization mode is selected, specify the number of initialization steps. **2** is the default.

## Advanced

**Advanced Settings.** Select this option if you want to set advanced options as follows.

**Max Iteration.** Specify the maximum number of iterations to perform when searching cluster centers. **20** is the default.

**Tolerance.** Specify the convergence tolerance for iterative algorithms. **1.0E-4** is the default.

**Set Random Seed.** Select this option and click **Generate** to generate the seed used by the random number generator.

## Display

**Display Graph.** Select this option if you want a graph to be included in the output.

The following table shows the relationship between the settings in the SPSS Modeler K-Means-AS node and the K-Means Spark parameters.

Table 43. Node properties mapped to Spark parameters		
SPSS Modeler setting	Script name (property name)	K-Means SparkML parameter
Input Fields	features	
Number of Clusters	clustersNum	k
Initialization Mode	initMode	initMode
Initialization Steps	initSteps	initSteps
Max Iteration	maxIter	maxIter
Toleration	toleration	tol
Random Seed	randomSeed	seed

<sup>1</sup> "Class KMeans." *Apache Spark*. JavaDoc. Web. 3 Oct 2017.

<sup>2</sup> Bahmani, Moseley, et al. "Scalable K-Means++." Feb 28, 2012. <http://theory.stanford.edu/%7Esergei/papers/vldb12-kmpar.pdf>.

## MultiLayerPerceptron-AS node

Multilayer perceptron is a classifier based on the feedforward artificial neural network and consists of multiple layers. Each layer is fully connected to the next layer in the network. For details about the multilayer perceptron classifier (MLPC), see <https://spark.apache.org/docs/latest/ml-classification-regression.html#multilayer-perceptron-classifier>.<sup>1</sup>

The MultiLayerPerceptron-AS node in SPSS Modeler is implemented in Spark. To use a this node, you must set up an upstream Type node. The MultiLayerPerceptron-AS node will read input values from the Type node (or the Types tab of an upstream source node).

<sup>1</sup> "Multilayer perceptron classifier." *Apache Spark*. MLlib: Main Guide. Web. 5 Oct 2018.

## MultiLayerPerceptron-AS node Fields

The Fields tab specifies which fields are used in the analysis.

**Use predefined roles.** This option tells the node to use field information from an upstream Type node. This is the default.

**Use custom field assignments.** To manually assign target and predictors, select this option.

**Target.** Select a field to use as the target for the prediction.

**Predictors.** Select one or more fields to use as inputs for the prediction.

## MultiLayerPerceptron-AS node Build Options

Use the Build Options tab to specify build options for the MultiLayerPerceptron-AS node, including performance, modeling building, and expert options. For more information about these options, see <http://spark.apache.org/docs/latest/api/java/org/apache/spark/ml/classification/MultilayerPerceptronClassifier.html>.<sup>1</sup>

### Performance

**Perceptrons Layer.** Use this setting to define the number of perceptron layers to include. This value must be larger than the number of perceptron fields. The default value is **1**.

**Hidden Layers.** Specify the number of hidden layers. Use a comma between multiple hidden layers. The default value is **1**.

**Output Layer.** Specify the number of output layers. The default value is **1**.

**Random Seed.** Click **Generate** if you want to generate the seed used by the random number generator.

### Model Building

**Max iterations.** Specify the maximum number of iterations to perform. The default value is **10**.

### Expert Only

**BlockSize.** Select the **Expert Mode** option in the Model Building section if you want to specify the block size for stacking input data in matrices. This can speed up the computation. The default block size is **128**.

The following table shows the relationship between the settings in the SPSS Modeler MultiLayerPerceptron-AS node dialogs and the Spark KDE library parameters.

Table 44. Node properties mapped to Spark parameters		
SPSS Modeler setting	Script name (property name)	Spark parameter
Predictors	features	
Target	label	
Perceptrons Layer	layers[0]	layers[0]
Hidden Layers	layers[1...<latest-1>]	layers[1...<latest-1>]
Output Layer	layers[<latest>]	layers[<latest>]
Random Seed	seed	seed
Max iterations	maxiter	maxiter

<sup>1</sup> "Class MultilayerPerceptronClassifier." *Apache Spark*. JavaDoc. Web. 5 Oct 2018.

## MultiLayerPerceptron node Model Options

**Model name.** You can generate the model name automatically based on the target or ID field (or model type in cases where no such field is specified) or specify a custom name.





## Notices

---

This information was developed for products and services offered in the US. This material might be available from IBM in other languages. However, you may be required to own a copy of the product or product version in that language in order to access it.

IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not grant you any license to these patents. You can send license inquiries, in writing, to:

*IBM Director of Licensing  
IBM Corporation  
North Castle Drive, MD-NC119  
Armonk, NY 10504-1785  
US*

For license inquiries regarding double-byte (DBCS) information, contact the IBM Intellectual Property Department in your country or send inquiries, in writing, to:

*Intellectual Property Licensing  
Legal and Intellectual Property Law  
IBM Japan Ltd.  
19-21, Nihonbashi-Hakozakicho, Chuo-ku  
Tokyo 103-8510, Japan*

INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some jurisdictions do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM websites are provided for convenience only and do not in any manner serve as an endorsement of those websites. The materials at those websites are not part of the materials for this IBM product and use of those websites is at your own risk.

IBM may use or distribute any of the information you provide in any way it believes appropriate without incurring any obligation to you.

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact:

*IBM Director of Licensing  
IBM Corporation  
North Castle Drive, MD-NC119  
Armonk, NY 10504-1785  
US*

Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.

The licensed program described in this document and all licensed material available for it are provided by IBM under terms of the IBM Customer Agreement, IBM International Program License Agreement or any equivalent agreement between us.

The performance data and client examples cited are presented for illustrative purposes only. Actual performance results may vary depending on specific configurations and operating conditions.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

Statements regarding IBM's future direction or intent are subject to change or withdrawal without notice, and represent goals and objectives only.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to actual people or business enterprises is entirely coincidental.

## Trademarks

---

IBM, the IBM logo, and ibm.com are trademarks or registered trademarks of International Business Machines Corp., registered in many jurisdictions worldwide. Other product and service names might be trademarks of IBM or other companies. A current list of IBM trademarks is available on the web at "Copyright and trademark information" at [www.ibm.com/legal/copytrade.shtml](http://www.ibm.com/legal/copytrade.shtml).

Adobe, the Adobe logo, PostScript, and the PostScript logo are either registered trademarks or trademarks of Adobe Systems Incorporated in the United States, and/or other countries.

Intel, Intel logo, Intel Inside, Intel Inside logo, Intel Centrino, Intel Centrino logo, Celeron, Intel Xeon, Intel SpeedStep, Itanium, and Pentium are trademarks or registered trademarks of Intel Corporation or its subsidiaries in the United States and other countries.

Linux is a registered trademark of Linus Torvalds in the United States, other countries, or both.

Microsoft, Windows, Windows NT, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.

UNIX is a registered trademark of The Open Group in the United States and other countries.

Java and all Java-based trademarks and logos are trademarks or registered trademarks of Oracle and/or its affiliates.

## Terms and conditions for product documentation

---

Permissions for the use of these publications are granted subject to the following terms and conditions.

### **Applicability**

These terms and conditions are in addition to any terms of use for the IBM website.

### **Personal use**

You may reproduce these publications for your personal, noncommercial use provided that all proprietary notices are preserved. You may not distribute, display or make derivative work of these publications, or any portion thereof, without the express consent of IBM.

**Commercial use**

You may reproduce, distribute and display these publications solely within your enterprise provided that all proprietary notices are preserved. You may not make derivative works of these publications, or reproduce, distribute or display these publications or any portion thereof outside your enterprise, without the express consent of IBM.

**Rights**

Except as expressly granted in this permission, no other permissions, licenses or rights are granted, either express or implied, to the publications or any information, data, software or other intellectual property contained therein.

IBM reserves the right to withdraw the permissions granted herein whenever, in its discretion, the use of the publications is detrimental to its interest or, as determined by IBM, the above instructions are not being properly followed.

You may not download, export or re-export this information except in full compliance with all applicable laws and regulations, including all United States export laws and regulations.

IBM MAKES NO GUARANTEE ABOUT THE CONTENT OF THESE PUBLICATIONS. THE PUBLICATIONS ARE PROVIDED "AS-IS" AND WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING BUT NOT LIMITED TO IMPLIED WARRANTIES OF MERCHANTABILITY, NON-INFRINGEMENT, AND FITNESS FOR A PARTICULAR PURPOSE.



# Glossary

---

## A

---

### **AICC**

A measure for selecting and comparing mixed models based on the -2 (Restricted) log likelihood. Smaller values indicate better models. The AICC "corrects" the AIC for small sample sizes. As the sample size increases, the AICC converges to the AIC.

## B

---

### **Bayesian Information Criterion (BIC)**

A measure for selecting and comparing models based on the -2 log likelihood. Smaller values indicate better models. The BIC also "penalizes" overparameterized models (complex models with a large number of inputs, for example), but more strictly than the AIC.

### **Box's M test**

A test for the equality of the group covariance matrices. For sufficiently large samples, a nonsignificant p value means there is insufficient evidence that the matrices differ. The test is sensitive to departures from multivariate normality.

## C

---

### **Cases**

Codes for actual group, predicted group, posterior probabilities, and discriminant scores are displayed for each case.

### **Classification Results**

The number of cases correctly and incorrectly assigned to each of the groups based on the discriminant analysis. Sometimes called the "Confusion Matrix."

### **Combined-Groups Plots**

Creates an all-groups scatterplot of the first two discriminant function values. If there is only one function, a histogram is displayed instead.

### **Covariance**

An unstandardized measure of association between two variables, equal to the cross-product deviation divided by  $N-1$ .

## F

---

### **Fisher's**

Displays Fisher's classification function coefficients that can be used directly for classification. A separate set of classification function coefficients is obtained for each group, and a case is assigned to the group for which it has the largest discriminant score (classification function value).

## H

---

### ***Hazard Plot***

Displays the cumulative hazard function on a linear scale.

## K

---

### ***Kurtosis***

A measure of the extent to which there are outliers. For a normal distribution, the value of the kurtosis statistic is zero. Positive kurtosis indicates that the data exhibit more extreme outliers than a normal distribution. Negative kurtosis indicates that the data exhibit less extreme outliers than a normal distribution.

## L

---

### ***Leave-one-out Classification***

Each case in the analysis is classified by the functions derived from all cases other than that case. It is also known as the "U-method."

## M

---

### ***MAE***

Mean absolute error. Measures how much the series varies from its model-predicted level. MAE is reported in the original series units.

### ***Mahalanobis Distance***

A measure of how much a case's values on the independent variables differ from the average of all cases. A large Mahalanobis distance identifies a case as having extreme values on one or more of the independent variables.

### ***MAPE***

Mean Absolute Percentage Error. A measure of how much a dependent series varies from its model-predicted level. It is independent of the units used and can therefore be used to compare series with different units.

### ***MaxAE***

Maximum Absolute Error. The largest forecasted error, expressed in the same units as the dependent series. Like MaxAPE, it is useful for imagining the worst-case scenario for your forecasts. Maximum absolute error and maximum absolute percentage error may occur at different series points--for example, when the absolute error for a large series value is slightly larger than the absolute error for a small series value. In that case, the maximum absolute error will occur at the larger series value and the maximum absolute percentage error will occur at the smaller series value.

### ***MaxAPE***

Maximum Absolute Percentage Error. The largest forecasted error, expressed as a percentage. This measure is useful for imagining a worst-case scenario for your forecasts.

### ***Maximizing the Smallest F Ratio Method of Entry***

A method of variable selection in stepwise analysis based on maximizing an F ratio computed from the Mahalanobis distance between groups.

**Maximum**

The largest value of a numeric variable.

**Mean**

A measure of central tendency. The arithmetic average, the sum divided by the number of cases.

**Means**

Displays total and group means, as well as standard deviations for the independent variables.

**Median**

The value above and below which half of the cases fall, the 50th percentile. If there is an even number of cases, the median is the average of the two middle cases when they are sorted in ascending or descending order. The median is a measure of central tendency not sensitive to outlying values (unlike the mean, which can be affected by a few extremely high or low values).

**Minimize Wilks' Lambda**

A variable selection method for stepwise discriminant analysis that chooses variables for entry into the equation on the basis of how much they lower Wilks' lambda. At each step, the variable that minimizes the overall Wilks' lambda is entered.

**Minimum**

The smallest value of a numeric variable.

**Mode**

The most frequently occurring value. If several values share the greatest frequency of occurrence, each of them is a mode.

**N**

---

**Normalized BIC**

Normalized Bayesian Information Criterion. A general measure of the overall fit of a model that attempts to account for model complexity. It is a score based upon the mean square error and includes a penalty for the number of parameters in the model and the length of the series. The penalty removes the advantage of models with more parameters, making the statistic easy to compare across different models for the same series.

**O**

---

**One Minus Survival**

Plots one-minus the survival function on a linear scale.

**R**

---

**Range**

The difference between the largest and smallest values of a numeric variable, the maximum minus the minimum.

**Rao's V (Discriminant Analysis)**

A measure of the differences between group means. Also called the Lawley-Hotelling trace. At each step, the variable that maximizes the increase in Rao's V is entered. After selecting this option, enter the minimum value a variable must have to enter the analysis.

## **RMSE**

Root Mean Square Error. The square root of mean square error. A measure of how much a dependent series varies from its model-predicted level, expressed in the same units as the dependent series.

## **R-Squared**

Goodness-of-fit measure of a linear model, sometimes called the coefficient of determination. It is the proportion of variation in the dependent variable explained by the regression model. It ranges in value from 0 to 1. Small values indicate that the model does not fit the data well.

## **S**

---

### **Separate-Groups**

Separate-groups covariance matrices are used for classification. Because classification is based on the discriminant functions (not based on the original variables), this option is not always equivalent to quadratic discrimination.

### **Separate-Groups Covariance**

Displays separate covariance matrices for each group.

### **Separate-Groups Plots**

Creates separate-group scatterplots of the first two discriminant function values. If there is only one function, histograms are displayed instead.

### **Sequential Bonferroni**

This is a sequentially step-down rejective Bonferroni procedure that is much less conservative in terms of rejecting individual hypotheses but maintains the same overall significance level.

### **Sequential Sidak**

This is a sequentially step-down rejective Sidak procedure that is much less conservative in terms of rejecting individual hypotheses but maintains the same overall significance level.

### **Skewness**

A measure of the asymmetry of a distribution. The normal distribution is symmetric and has a skewness value of 0. A distribution with a significant positive skewness has a long right tail. A distribution with a significant negative skewness has a long left tail. As a guideline, a skewness value more than twice its standard error is taken to indicate a departure from symmetry.

### **standard deviation**

A measure of dispersion around the mean, equal to the square root of the variance. The standard deviation is measured in the same units as the original variable.

### **Standard Deviation**

A measure of dispersion around the mean. In a normal distribution, 68% of cases fall within one standard deviation of the mean and 95% of cases fall within two standard deviations. For example, if the mean age is 45, with a standard deviation of 10, 95% of the cases would be between 25 and 65 in a normal distribution.

### **Standard Error**

A measure of how much the value of a test statistic varies from sample to sample. It is the standard deviation of the sampling distribution for a statistic. For example, the standard error of the mean is the standard deviation of the sample means.



### ***Standard Error of Kurtosis***

The ratio of kurtosis to its standard error can be used as a test of normality (that is, you can reject normality if the ratio is less than -2 or greater than +2). A large positive value for kurtosis indicates that the tails of the distribution are longer than those of a normal distribution; a negative value for kurtosis indicates shorter tails (becoming like those of a box-shaped uniform distribution).

### ***Standard Error of Mean***

A measure of how much the value of the mean may vary from sample to sample taken from the same distribution. It can be used to roughly compare the observed mean to a hypothesized value (that is, you can conclude the two values are different if the ratio of the difference to the standard error is less than -2 or greater than +2).

### ***Standard Error of Skewness***

The ratio of skewness to its standard error can be used as a test of normality (that is, you can reject normality if the ratio is less than -2 or greater than +2). A large positive value for skewness indicates a long right tail; an extreme negative value indicates a long left tail.

### ***Stationary R-squared***

A measure that compares the stationary part of the model to a simple mean model. This measure is preferable to ordinary R-squared when there is a trend or seasonal pattern. Stationary R-squared can be negative with a range of negative infinity to 1. Negative values mean that the model under consideration is worse than the baseline model. Positive values mean that the model under consideration is better than the baseline model.

### ***Sum***

The sum or total of the values, across all cases with nonmissing values.

### ***Survival Plot***

Displays the cumulative survival function on a linear scale.

## **T**

---

### ***Territorial Map***

A plot of the boundaries used to classify cases into groups based on function values. The numbers correspond to groups into which cases are classified. The mean for each group is indicated by an asterisk within its boundaries. The map is not displayed if there is only one discriminant function.

### ***Total Covariance***

Displays a covariance matrix from all cases as if they were from a single sample.

## **U**

---

### ***Unexplained Variance***

At each step, the variable that minimizes the sum of the unexplained variation between groups is entered.

### ***Unique***

Evaluates all effects simultaneously, adjusting each effect for all other effects of any type.

### ***Univariate ANOVAs***

Performs a one-way analysis-of-variance test for equality of group means for each independent variable.

### ***Unstandardized***

Displays the unstandardized discriminant function coefficients.

### ***Use F Value***

A variable is entered into the model if its F value is greater than the Entry value and is removed if the F value is less than the Removal value. Entry must be greater than Removal, and both values must be positive. To enter more variables into the model, lower the Entry value. To remove more variables from the model, increase the Removal value.

### ***Use Probability of F***

A variable is entered into the model if the significance level of its F value is less than the Entry value and is removed if the significance level is greater than the Removal value. Entry must be less than Removal, and both values must be positive. To enter more variables into the model, increase the Entry value. To remove more variables from the model, lower the Removal value.

## **V**

---

### ***Valid***

Valid cases having neither the system-missing value, nor a value defined as user-missing.

### ***Variance***

A measure of dispersion around the mean, equal to the sum of squared deviations from the mean divided by one less than the number of cases. The variance is measured in units that are the square of those of the variable itself.

## **W**

---

### ***Within-Groups***

The pooled within-groups covariance matrix is used to classify cases.

### ***Within-Groups Correlation***

Displays a pooled within-groups correlation matrix that is obtained by averaging the separate covariance matrices for all groups before computing the correlations.

### ***Within-Groups Covariance***

Displays a pooled within-groups covariance matrix, which may differ from the total covariance matrix. The matrix is obtained by averaging the separate covariance matrices for all groups.

---

# Index

## A

- absolute confidence difference to prior
  - apriori evaluation measure [234](#)
- add model rules [141](#)
- additive outliers
  - patches [262](#)
- adjusted propensity scores
  - balancing data [30](#)
  - decision list models [136](#)
  - discriminant models [176](#)
  - generalized linear models [182](#)
- adjusted R-square
  - in linear models [153](#)
  - in linear-AS models [159](#)
- advanced output
  - Cox regression models [207](#)
  - Factor/PCA node [172](#)
- advanced parameters [140](#)
- Akaike information criterion
  - in linear models [153](#)
  - in linear-AS models [159](#)
- algorithms [31](#)
- alternative models [143](#)
- Alternative Rules pane [141](#)
- Alternatives tab [138](#)
- analysis of variance
  - in generalized linear mixed models [183](#)
- anomaly detection models
  - adjustment coefficient [50](#)
  - anomaly fields [49](#), [51](#)
  - anomaly index [49](#)
  - cutoff value [49](#), [51](#)
  - missing values [50](#)
  - noise level [50](#)
  - peer groups [50](#), [51](#)
  - scoring [50](#), [51](#)
- ANOVA
  - in linear models [156](#)
- antecedent
  - rules without [237](#)
- application examples [3](#)
- apriori models
  - evaluation measures [234](#)
  - expert options [234](#)
  - modeling node [233](#)
  - modeling node options [233](#)
  - tabular versus transactional data [26](#)
- ARIMA models
  - transfer functions [291](#)
- assess a model [145](#)
- assessment in Excel [146](#)
- association rule models
  - apriori [233](#)
  - CARMA [235](#)
  - deploying [244](#)
  - for sequences [246](#)
  - association rule models (*continued*)
    - generating a filtered model [242](#)
    - generating a rule set [242](#)
    - graph generation [240](#)
    - model nugget [237](#)
    - model nugget details [237](#)
    - model nugget summary [242](#)
    - scoring rules [243](#)
    - settings [240](#)
    - specifying filters [240](#)
    - transposing scores [244](#)
  - Association Rules [252](#)
  - Association Rules building [253](#)
  - Association Rules model options [256](#)
  - Association Rules models
    - field options [253](#)
    - model nugget [257](#)
    - model nugget details [257](#)
    - model nugget settings [257](#)
  - Association Rules node [252](#)
  - Association Rules output [255](#)
  - Association Rules transformations [254](#)
  - asymptotic correlations
    - logistic regression models [166](#), [169](#)
  - asymptotic covariance
    - logistic regression models [166](#)
  - auto classifier models
    - algorithm settings [54](#)
    - discarding models [59](#)
    - evaluation charts [69](#)
    - evaluation graphs [69](#)
    - generating modeling nodes and nuggets [68](#)
    - introduction [54](#)
    - model nugget [67](#)
    - model types [56](#)
    - modeling node [54](#), [55](#)
    - partitions [56](#)
    - ranking models [55](#)
    - results browser window [67](#)
    - settings [60](#)
    - stopping rules [54](#)
  - auto cluster models
    - algorithm settings [54](#)
    - discarding models [67](#)
    - evaluation charts [69](#)
    - generating modeling nodes and nuggets [68](#)
    - model nugget [67](#)
    - model types [66](#)
    - modeling node [65](#)
    - partitions [66](#)
    - ranking models [65](#)
    - results browser window [67](#)
    - stopping rules [54](#)
  - Auto Cluster models
    - modeling node [64](#)
  - auto numeric models
    - algorithm settings [54](#)

- auto numeric models (*continued*)
  - evaluation charts [69](#)
  - evaluation graphs [69](#)
  - generating modeling nodes and nuggets [68](#)
  - model nugget [67](#)
  - model types [62](#)
  - modeling node [60](#), [61](#)
  - modeling options [61](#)
  - results browser window [67](#)
  - settings [64](#)
  - stopping rules [54](#), [62](#)
- autocorrelation function
  - series [263](#)
- automated modeling nodes
  - auto classifier models [53](#)
  - auto cluster models [53](#)
  - auto numeric models [53](#)
- automatic data preparation
  - in linear models [155](#)
- available fields [141](#)

## B

- bagging
  - in linear models [152](#)
  - in neural networks [121](#)
- base category
  - Logistic node [162](#)
- basket data [243](#), [244](#)
- Bayesian network models
  - expert options [115](#)
  - model nugget [117](#)
  - model nugget settings [117](#)
  - model nugget summary [118](#)
  - model options [114](#)
  - modeling node [113](#)
- best subsets
  - in linear models [153](#)
  - in linear-AS models [159](#)
- binomial logistic regression models [161](#), [162](#)
- Bonferroni adjustment
  - CHAID node [89](#)
  - Tree-AS node [94](#)
- boosting
  - in linear models [152](#)
  - in neural networks [121](#)
- Box's M test
  - Discriminant node [174](#)
- build options for spatio-temporal prediction [267](#)
- Build Rule node [103](#)
- build selections
  - defining [140](#)
- building association rules [253](#)

## C

- C&R Tree models
  - build options [85](#)
  - case weights [26](#)
  - ensembling [87](#)
  - field options [85](#)
  - frequency weights [26](#)
  - graph generation from model nugget [108](#)

- C&R Tree models (*continued*)
  - impurity measures [89](#)
  - misclassification costs [87](#)
  - model nugget [103](#)
  - modeling node [72](#), [82](#), [83](#), [106](#)
  - objectives [85](#)
  - prior probabilities [87](#)
  - pruning [86](#)
  - stopping options [87](#)
  - surrogates [86](#)
  - tree depth [86](#)
- C5.0 models
  - boosting [92](#), [107](#)
  - graph generation from model nugget [108](#)
  - misclassification costs [92](#)
  - model nugget [103](#), [109](#), [110](#)
  - modeling node [91](#), [92](#), [106](#), [107](#)
  - options [92](#)
  - pruning [92](#)
- CARMA models
  - content field(s) [235](#)
  - data formats [235](#)
  - expert options [237](#)
  - field options [235](#)
  - ID field [235](#)
  - modeling node [235](#)
  - modeling node options [236](#)
  - multiple consequents [243](#)
  - tabular versus transactional data [237](#)
  - time field [235](#)
- CHAID models
  - build options [85](#)
  - ensembling [87](#)
  - exhaustive CHAID [86](#), [94](#)
  - field options [85](#)
  - graph generation from model nugget [108](#)
  - misclassification costs [88](#)
  - model nugget [103](#)
  - modeling node [72](#), [82](#), [84](#), [106](#)
  - objectives [85](#)
  - stopping options [87](#), [95](#)
  - tree depth [86](#), [94](#)
- change target value [144](#)
- chart options [149](#)
- chi-square
  - CHAID node [89](#)
  - feature selection [46](#)
  - Tree-AS node [94](#)
- classification gains
  - decision trees [76](#), [77](#)
- classification table
  - in Nearest Neighbor Analysis [320](#)
  - logistic regression models [166](#)
- classification trees [83](#), [84](#), [91](#), [93](#), [98](#)
- cluster analysis
  - anomaly detection [50](#)
  - number of clusters [217](#)
  - Twostep Cluster [218–222](#)
- cluster viewer
  - about cluster models [224](#)
  - basic view [227](#)
  - cell content display [227](#)
  - cell distribution view [227](#)
  - cluster centers view [226](#)

- cluster viewer (*continued*)
  - cluster comparison view [227](#)
  - cluster display sort [227](#)
  - cluster predictor importance view [227](#)
  - cluster sizes view [227](#)
  - clusters view [226](#)
  - comparison of clusters [227](#)
  - distribution of cells [227](#)
  - feature display sort [226](#)
  - flip clusters and features [226](#)
  - graph generation [229](#)
  - model summary [225](#)
  - overview [225](#)
  - predictor importance [227](#)
  - size of clusters [227](#)
  - sort cell contents [227](#)
  - sort clusters [227](#)
  - sort features [226](#)
  - summary view [225](#)
  - transpose clusters and features [226](#)
  - using [228](#)
- clustering
  - overall display [225](#)
  - viewing clusters [225](#)
- clustering nodes [223](#), [345](#)
- coefficient of variance
  - screening fields [46](#)
- combining rules
  - in linear models [154](#)
  - in neural networks [124](#)
- confidence
  - Apriori node [233](#)
  - association rules [237](#), [240](#), [250](#)
  - CARMA node [236](#)
  - for sequences [250](#)
  - Sequence node [247](#)
- confidence difference
  - apriori evaluation measure [234](#)
- confidence intervals
  - logistic regression models [166](#)
- confidence ratio
  - apriori evaluation measure [234](#)
- confidence scores [30](#)
- confidences
  - decision tree models [97](#), [102](#), [106](#)
  - GLE models [204](#)
  - logistic regression models [168](#)
  - rule sets [106](#)
- confusion matrix
  - LSVM models [310](#)
- consequent
  - multiple consequents [237](#)
- content field(s)
  - CARMA node [235](#)
  - Sequence node [246](#)
- contrast coefficients matrix
  - generalized linear models [181](#)
- convergence options
  - CHAID node [89](#)
  - Cox regression models [207](#)
  - generalized linear models [180](#)
  - logistic regression models [165](#)
  - Tree-AS node [95](#)
- copying model links [33](#)

- correlation matrix
  - generalized linear models [181](#)
- costs
  - decision trees [87](#), [88](#), [95](#), [100](#)
  - misclassification [31](#)
- covariance matrix
  - generalized linear models [181](#)
- Cox regression models
  - advanced output [207](#), [209](#)
  - convergence criteria [207](#)
  - expert options [206](#)
  - field options [205](#)
  - model nugget [208](#)
  - model options [205](#)
  - modeling node [204](#)
  - settings options [207](#)
  - stepping criteria [207](#)
- Cramér's V
  - feature selection [46](#)
- custom splits
  - decision trees [74](#)
- customize a model [143](#)

## D

- data reduction
  - PCA/factor models [170](#)
- decision list models
  - alternatives tab [138](#)
  - binning method [135](#)
  - expert options [135](#)
  - model options [134](#)
  - modeling node [133](#)
  - PMML [135](#)
  - requirements [133](#)
  - scoring [135](#)
  - search direction [134](#)
  - search width [135](#)
  - segments [135](#)
  - settings [136](#)
  - snapshots tab [138](#)
  - SQL generation [136](#)
  - target value [134](#)
  - viewer workspace [136](#)
  - working model pane [136](#)
  - working with viewer [139](#)
- decision tree models
  - custom splits [74](#)
  - exporting results [81](#)
  - gains [75–78](#)
  - generating [78](#), [79](#)
  - graph generation [108](#)
  - misclassification costs [87](#), [88](#), [95](#), [100](#)
  - modeling node [81](#)
  - predictors [74](#)
  - profits [77](#)
  - ROI [77](#)
  - surrogates [74](#)
  - viewer [106](#)
- deleting
  - model links [32](#)
- deployability measure [237](#)
- descriptive statistics
  - generalized linear models [181](#)

- difference of confidence quotient to 1
  - apriori evaluation measure [234](#)
- differencing transformation [264](#)
- dimension reduction [212](#)
- direct oblimin rotation
  - PCA/factor models [172](#)
- directives
  - decision trees [81](#)
- discriminant models
  - advanced output [174](#), [176](#)
  - convergence criteria [174](#)
  - expert options [174](#)
  - model form [173](#)
  - model nugget [176](#), [177](#)
  - modeling node [173](#)
  - propensity scores [176](#)
  - scoring [176](#)
  - stepping criteria (field selection) [175](#)
- documentation [3](#)
- DTD [41](#)

## E

- edit
  - advanced parameters [140](#)
- eigenvalues
  - PCA/factor models [171](#)
- ensemble viewer
  - automatic data preparation [40](#)
  - component model accuracy [39](#)
  - component model details [39](#)
  - model summary [38](#)
  - predictor frequency [39](#)
  - predictor importance [39](#)
- ensembles
  - in linear models [154](#)
  - in neural networks [124](#)
- epsilon for convergence
  - CHAID node [89](#)
  - Tree-AS node [95](#)
- equamax rotation
  - PCA/factor models [172](#)
- error summary
  - in Nearest Neighbor Analysis [320](#)
- evaluation charts
  - from auto classifier models [69](#)
  - from auto cluster models [69](#)
  - from auto numeric models [69](#)
- evaluation graphs
  - from auto classifier models [69](#)
  - from auto numeric models [69](#)
- evaluation measures
  - Apriori node [234](#)
- events
  - identifying [261](#)
- examples
  - Applications Guide [3](#)
  - overview [4](#)
- exhaustive CHAID [72](#), [86](#), [94](#)
- expert options
  - Apriori node [234](#)
  - Bayesian network node [115](#)
  - CARMA node [237](#)
  - Cox regression models [206](#)

- expert options (*continued*)
  - k-means models [216](#)
  - Kohonen models [214](#)
  - Sequence node [247](#)
- expert output
  - Cox regression models [207](#)
- exponential smoothing [284](#)
- exporting
  - model nuggets [34](#)
  - PMML [41](#), [42](#)
  - SQL [35](#)

## F

- F statistic
  - feature selection [46](#)
  - in linear models [153](#)
  - in linear-AS models [159](#)
- factor models
  - advanced output [173](#)
  - eigenvalues [171](#)
  - equations [172](#)
  - expert options [171](#)
  - factor scores [171](#)
  - iterations [171](#)
  - missing-value handling [171](#)
  - model nugget [172](#), [173](#)
  - model options [170](#)
  - modeling node [170](#)
  - number of factors [171](#)
  - rotation [172](#)
- feature selection models
  - generating Filter nodes [48](#)
  - importance [46](#), [47](#)
  - ranking predictors [46](#), [47](#)
  - screening predictors [46](#), [47](#)
- field importance
  - filtering fields [38](#)
  - model results [29](#), [37](#), [38](#)
  - ranking fields [46–48](#)
- field options
  - Cox node [205](#)
  - modeling nodes [26](#)
  - SLRM node [299](#)
- Filter node
  - generating from decision trees [81](#)
- filtering rules
  - association rules [240](#)
- first hit rule set [109](#)
- focal records [314](#)
- folds, cross-validation [316](#)
- forecasting
  - overview [259](#)
  - predictor series [264](#)
- forward stepwise
  - in linear models [153](#)
  - in linear-AS models [159](#)
- frequency fields [27](#)
- functional transformation [264](#)

## G

- gains

- gains (*continued*)
  - chart [148](#)
  - decision trees [75–77](#)
  - exporting [81](#)
- gains-based selection [78](#)
- Gaussian Mixture node
  - inputs [330](#)
- general estimable function(s)
  - generalized linear models [181](#)
- general linear model
  - generalized linear mixed models [183](#)
- generalized linear mixed models
  - analysis weight [189](#)
  - classification table [193](#)
  - covariance parameters [195](#)
  - custom terms [187](#)
  - data structure [193](#)
  - estimated marginal means [191](#)
  - estimated means [195](#)
  - fixed coefficients [194](#)
  - fixed effects [187](#), [193](#)
  - link function [184](#)
  - model summary [192](#)
  - model view [192](#)
  - offset [189](#)
  - predicted by observed [193](#)
  - random effect block [188](#)
  - random effect covariances [194](#)
  - random effects [188](#)
  - scoring options [191](#)
  - settings [196](#)
  - target distribution [184](#)
- generalized linear model
  - in generalized linear mixed models [183](#)
- generalized linear models
  - advanced output [181](#), [182](#)
  - convergence options [180](#)
  - expert options [178](#)
  - fields [177](#)
  - model form [178](#)
  - model nugget [182](#), [183](#)
  - modeling node [177](#), [196](#)
  - propensity scores [182](#)
- generate new model [144](#)
- generated sequence rule set [242](#)
- getting started [136](#)
- Gini impurity measure [89](#)
- GLE models
  - analysis weight [200](#)
  - build options [201](#)
  - custom terms [200](#)
  - link function [197](#)
  - model effects [199](#)
  - model information [203](#)
  - model selection options [202](#)
  - modeling node [204](#)
  - offset [200](#)
  - output [203](#)
  - predictor importance [203](#)
  - scoring options [203](#)
  - target distribution [197](#)
- GMM node
  - inputs [330](#)
- goodness-of-fit statistics

- goodness-of-fit statistics (*continued*)
  - generalized linear models [181](#)
  - logistic regression models [169](#)
- graph generation
  - association rules [240](#)

## H

- HDBSCAN node
  - inputs [336](#)
- hierarchical models
  - generalized linear mixed models [183](#)
- hits
  - decision tree gains [75](#)
- Hosmer and Lemeshow goodness-of-fit
  - logistic regression models [169](#)

## I

- IBM SPSS Modeler
  - documentation [3](#)
- IBM SPSS Modeler Server [1](#)
- ID field
  - CARMA node [235](#)
  - Sequence node [246](#)
- importance
  - filtering fields [38](#)
  - predictors in models [29](#), [37](#), [38](#)
  - ranking predictors [46–48](#)
- importing
  - PMML [34](#), [41](#), [42](#)
- impurity measures
  - C&R Tree node [89](#)
  - decision trees [89](#)
- index
  - decision tree gains [75](#)
- information criteria
  - in linear models [153](#)
  - in linear-AS models [159](#)
- information difference
  - apriori evaluation measure [234](#)
- innovational outliers [262](#)
- input fields
  - screening [46](#)
  - selecting for analysis [46](#)
- instances [237](#), [250](#)
- interactions
  - logistic regression models [164](#)
- interactive trees
  - custom splits [74](#)
  - exporting results [81](#)
  - gains [75–78](#)
  - generating models [78](#), [79](#)
  - graph generation [108](#)
  - profits [77](#)
  - ROI [77](#)
  - surrogates [74](#)
- interventions
  - identifying [261](#)
- Isotonic-AS model nuggets [342](#)
- Isotonic-AS node [341](#), [342](#)
- iteration history
  - generalized linear models [181](#)

iteration history (*continued*)  
logistic regression models [166](#)

## K

k-means models  
clustering [215](#), [216](#)  
distance field [215](#)  
encoding value for sets [216](#)  
expert options [216](#)  
model nugget [216](#)  
stopping criteria [216](#)  
K-Means models  
graph generation from model nugget [229](#)  
K-Means-AS node [223](#), [345](#)  
KDE Modeling node [331](#)  
KDE node  
inputs [332](#)  
kernel functions  
support vector machine models [305](#)  
KNN. See nearest neighbor models [313](#)  
Kohonen models  
binary set encoding option (removed) [213](#)  
expert options [214](#)  
feedback graph [213](#)  
graph generation from model nugget [229](#)  
learning rate [214](#)  
model nugget [214](#)  
modeling node [212](#)  
neighborhood [212](#), [214](#)  
neural networks [212](#), [214](#)  
stopping criteria [213](#)

## L

L matrix  
generalized linear models [181](#)  
labels  
value [41](#)  
variable [41](#)  
lag  
ACF and PACF [263](#)  
Lagrange multiplier test  
generalized linear models [181](#)  
lambda  
feature selection [46](#)  
level shift outliers [262](#)  
level stabilizing transformation [264](#)  
lift  
association rules [240](#)  
decision tree gains [75](#)  
lift charts  
decision tree gains [77](#)  
likelihood ratio test  
logistic regression models [166](#), [169](#)  
likelihood-ratio chi-square  
CHAID node [89](#)  
feature selection [46](#)  
Tree-AS node [94](#)  
linear kernel  
support vector machine models [305](#)  
linear models  
ANOVA table [156](#)

linear models (*continued*)  
automatic data preparation [153](#), [155](#)  
coefficients [156](#)  
combining rules [154](#)  
confidence level [153](#)  
ensembles [154](#)  
estimated means [157](#)  
information criterion [155](#)  
model building summary [157](#)  
model options [155](#)  
model selection [153](#)  
model summary [155](#)  
nugget settings [157](#)  
objectives [152](#)  
outliers [156](#)  
predicted by observed [156](#)  
predictor importance [155](#)  
R-square statistic [155](#)  
replicating results [154](#)  
residuals [156](#)  
linear regression models  
modeling node [152](#), [158](#)  
weighted least squares [26](#)  
linear support vector machine models  
build options [310](#)  
model nugget [310](#)  
model options [309](#)  
modeling node [309](#)  
settings [311](#)  
linear trends  
identifying [259](#)  
linear-AS models  
confidence interval [159](#)  
confidence level [159](#)  
consider two way interaction [159](#)  
include intercept [159](#)  
information criterion [160](#)  
model information [160](#)  
model options [160](#)  
model selection [159](#)  
nugget settings [160](#)  
output [160](#)  
predicted by observed [160](#)  
predictor importance [160](#)  
R-square statistic [160](#)  
records summary [160](#)  
sorting order for categorical predictors [159](#)  
linear-AS node [158](#)  
linearnode node [152](#)  
link function  
generalized linear mixed models [184](#)  
GLE models [197](#)  
links  
model [32](#)  
loading  
model nuggets [34](#)  
local trend outliers [262](#)  
log transformation  
Time Series Modeler [291](#)  
log-odds  
logistic regression models [167](#)  
logistic regression  
generalized linear mixed models [183](#)  
logistic regression models



logistic regression models (*continued*)

- adding terms [164](#)
- advanced output [166](#), [169](#)
- binomial options [162](#)
- convergence options [165](#)
- expert options [165](#)
- interactions [164](#)
- main effects [164](#)
- model equations [167](#)
- model nugget [167](#), [168](#)
- modeling node [161](#)
- multinomial options [162](#)
- predictor importance [167](#)
- stepping options [166](#)

loglinear analysis

- in generalized linear mixed models [183](#)

longitudinal models

- generalized linear mixed models [183](#)

LSVM models

- confusion matrix [310](#)
- model information [310](#)
- output [310](#)
- predicted by observed [310](#)
- predictor importance [310](#)
- records summary [310](#)

## M

main effects

- logistic regression models [164](#)

managers

- Models tab [34](#)

mining task

- starting [140](#)

mining tasks

- creating [140](#)
- editing [140](#)

misclassification costs

- C5.0 node [92](#)

missing data

- predictor series [264](#)

missing values

- CHAID trees [74](#)
- excluding from SQL [97](#), [102](#), [106](#), [204](#)
- screening fields [46](#)

mixed models

- generalized linear mixed models [183](#)

MLP (multilayer perceptron)

- in neural networks [122](#)

model fit

- logistic regression models [169](#)

model information

- generalized linear models [181](#)
- GLE models [203](#)
- linear-AS models [160](#)
- LSVM models [310](#)
- Random Trees models [101](#)
- Time Series models [294](#)
- Tree-AS models [96](#)

model links

- and SuperNodes [33](#)
- copying and pasting [33](#)
- defining and removing [32](#)

model measures

model measures (*continued*)

- defining [145](#)
- refresh [145](#)

model nuggets

- ensemble models [38](#)
- exporting [34](#), [35](#)
- generating processing nodes [41](#)
- menus [35](#)
- printing [35](#)
- saving [35](#)
- saving and loading [34](#)
- scoring data with [41](#)
- split models [40](#)
- Summary tab [36](#)
- using in streams [41](#)

model options

- Bayesian network node [114](#)
- Cox regression models [205](#)
- SLRM node [299](#)

model options for spatio-temporal prediction [268](#)

model refresh

- self-learning response models [299](#)

model view

- in generalized linear mixed models [192](#)
- in Nearest Neighbor Analysis [318](#)

modeling nodes [48](#), [91](#), [113](#), [212](#), [215](#), [216](#), [218](#), [223](#), [233](#), [246](#), [299](#), [341–343](#), [345–347](#)

models

- importing [34](#)
- replacing [33](#)
- split [24](#), [25](#)
- Summary tab [36](#)

models palette [31](#), [34](#)

MS Excel setup integration format [146](#)

multilayer perceptron (MLP)

- in neural networks [122](#)

MultiLayerPerceptron-AS node [346](#), [347](#)

multilevel models

- generalized linear mixed models [183](#)

multinomial logistic regression

- generalized linear mixed models [183](#)

multinomial logistic regression models [161](#), [162](#)

## N

natural log transformation

- Time Series Modeler [291](#)

Nearest Neighbor Analysis

- model view [318](#)

nearest neighbor distances

- in Nearest Neighbor Analysis [319](#)

nearest neighbor models

- about [313](#)
- analyze options [316](#)
- cross-validation options [316](#)
- feature selection options [315](#)
- model options [314](#)
- modeling node [313](#)
- neighbors options [315](#)
- objectives options [313](#)
- settings options [314](#)

neural network models

- field options [26](#)

neural networks

- neural networks (*continued*)
  - classification [130](#)
  - combining rules [124](#)
  - ensembles [124](#)
  - hidden layers [122](#)
  - missing values [125](#)
  - model options [126](#)
  - model summary [127](#)
  - multilayer perceptron (MLP) [122](#)
  - network [131](#)
  - nugget settings [132](#)
  - objectives [121](#)
  - overfit prevention [125](#)
  - predicted by observed [129](#)
  - predictor importance [128](#)
  - radial basis function (RBF) [122](#)
  - replicating results [125](#)
  - stopping rules [123](#)
- neuralnetwork node [119](#)
- nodeName node [183](#)
- nominal regression [161](#)
- nonlinear trends
  - identifying [259](#)
- nonparametric estimation [267](#)
- nonseasonal cycles [260](#)
- normalized chi-square
  - apriori evaluation measure [234](#)

## O

- One-Class SVM node [338–340](#)
- optimizing performance [233](#)
- ordered twoing impurity measure [89](#)
- organize data selections [141](#)
- outliers
  - additive patches [262](#)
  - deterministic [262](#)
  - in series [261](#)
  - innovational [262](#)
  - level shift [262](#)
  - local trend [262](#)
  - seasonal additive [262](#)
  - transient change [262](#)
- output from Association Rules [255](#)
- output from spatio-temporal prediction [267](#)
- overfit prevention
  - in neural networks [125](#)
- overfit prevention criterion
  - in linear models [153](#)
  - in linear-AS models [159](#)
- overfitting SVM model [306](#)

## P

- p value [46](#)
- parameter estimates
  - generalized linear models [181](#)
  - logistic regression models [169](#)
- parametric estimation [267](#)
- partial autocorrelation function
  - series [263](#)
- partitions
  - selecting [246](#)

- PCA models
  - advanced output [173](#)
  - eigenvalues [171](#)
  - equations [172](#)
  - expert options [171](#)
  - factor scores [171](#)
  - iterations [171](#)
  - missing-value handling [171](#)
  - model nugget [172](#), [173](#)
  - model options [170](#)
  - modeling node [170](#)
  - number of factors [171](#)
  - rotation [172](#)
- Pearson chi-square
  - CHAID node [89](#)
  - feature selection [46](#)
  - Tree-AS node [94](#)
- peer groups
  - anomaly detection [50](#)
- peers
  - in Nearest Neighbor Analysis [319](#)
- performance enhancements [166](#), [233](#)
- periodicity
  - Time Series Modeler [291](#)
- PMML
  - exporting models [34](#), [41](#), [42](#)
  - importing models [34](#), [41](#), [42](#)
- point interventions
  - identifying [261](#)
- Poisson regression
  - generalized linear mixed models [183](#)
- predicted by observed
  - linear-AS models [160](#)
  - LSVM models [310](#)
- predictor importance
  - discriminant models [176](#)
  - filtering fields [38](#)
  - generalized linear models [182](#)
  - GLE models [203](#)
  - in Nearest Neighbor Analysis [319](#)
  - linear models [155](#)
  - linear-AS models [160](#)
  - logistic regression models [167](#)
  - LSVM models [310](#)
  - model results [29](#), [37](#), [38](#)
  - neural networks [128](#)
  - Random Trees models [101](#)
  - Time Series models [294](#)
  - Tree-AS models [96](#)
- predictor selection
  - in Nearest Neighbor Analysis [320](#)
- predictor series
  - missing data [264](#)
- predictor space chart
  - in Nearest Neighbor Analysis [318](#)
- predictors
  - decision trees [74](#)
  - ranking importance [46–48](#)
  - screening [47](#), [48](#)
  - selecting for analysis [46–48](#)
  - surrogates [74](#)
- preview
  - model contents [35](#)
- principal components analysis. See PCA models [170](#), [172](#)

- prior probabilities
  - decision trees [87](#)
- probabilities
  - logistic regression models [167](#)
- probit analysis
  - generalized linear mixed models [183](#)
- profits
  - decision tree gains [77](#)
- promax rotation
  - PCA/factor models [172](#)
- propensity scores
  - balancing data [30](#)
  - decision list models [136](#)
  - discriminant models [176](#)
  - generalized linear models [182](#)
- pruning decision trees [83](#), [86](#)
- pseudo R-square
  - logistic regression models [169](#)
- pulses
  - in series [261](#)
- python nodes [322–325](#), [327](#), [329–336](#), [338–340](#)

## Q

- quadrant map
  - in Nearest Neighbor Analysis [319](#)
- quartimax rotation
  - PCA/factor models [172](#)
- QUEST models
  - build options [85](#)
  - ensembling [87](#)
  - field options [85](#)
  - graph generation from model nugget [108](#)
  - misclassification costs [87](#)
  - model nugget [103](#)
  - modeling node [72](#), [82](#), [84](#), [106](#)
  - objectives [85](#)
  - prior probabilities [87](#)
  - pruning [86](#)
  - stopping options [87](#)
  - surrogates [86](#)
  - tree depth [86](#)

## R

- R-square
  - in linear models [155](#), [160](#)
- radial basis function (RBF)
  - in neural networks [122](#)
- Random Forest model nuggets [335](#)
- Random Forest node [333–335](#)
- Random Trees models
  - advanced settings [100](#)
  - binning [100](#)
  - build options [99](#)
  - field options [98](#)
  - misclassification costs [100](#)
  - model information [101](#)
  - modeling node [98](#), [102](#)
  - output [101](#)
  - predictor importance [101](#)
  - sample size [99](#)
  - tree depth [99](#)

- ranking predictors [46–48](#)
- raw propensity scores [30](#)
- RBF (radial basis function)
  - in neural networks [122](#)
- records summary
  - linear-AS models [160](#)
  - LSVM models [310](#)
- reference category
  - Logistic node [162](#)
- refreshing measures [145](#)
- refreshing models
  - self-learning response models [299](#)
- regression gains
  - decision trees [77](#), [78](#)
- regression models
  - modeling node [152](#), [158](#)
- regression trees [83](#), [84](#), [93](#), [98](#)
- removing model links [32](#)
- replacing models [33](#)
- response charts
  - decision tree gains [75](#), [77](#)
- risk estimate
  - decision tree gains [78](#)
- risks
  - exporting [81](#)
- ROI
  - decision tree gains [77](#)
- rotation
  - PCA/factor models [172](#)
- rule ID [237](#)
- rule induction [83](#), [84](#), [91](#), [93](#), [98](#), [233](#)
- rule set
  - generating from decision trees [81](#)
- Rule SuperNode
  - generating from sequence rules [251](#)
- rules
  - association rules [233](#), [235](#)
  - rule support [237](#), [250](#)
- run a mining task [139](#)

## S

- score statistic [166](#)
- scoring data [41](#)
- screening input fields [46](#)
- screening predictors [47](#), [48](#)
- seasonal additive outliers [262](#)
- seasonal differencing transformation [264](#)
- seasonality
  - identifying [260](#)
- segment rule generation [139](#)
- segments
  - copy [143](#)
  - deleting [144](#)
  - deleting rule conditions [142](#)
  - editing [142](#)
  - excluding [144](#)
  - inserting [142](#)
  - prioritizing [143](#)
- Select node
  - generating from decision trees [81](#)
- self-learning response models
  - field options [299](#)
  - model nugget [301](#)

- self-learning response models (*continued*)
  - model refresh [299](#)
  - modeling node [299](#)
  - settings [302](#)
  - variable importance [301](#)
- self-organizing maps [212](#)
- sequence browser [251](#)
- sequence detection [246](#)
- sequence models
  - content field(s) [246](#)
  - data formats [246](#)
  - expert options [247](#)
  - field options [246](#)
  - generating a rule SuperNode [251](#)
  - ID field [246](#)
  - model nugget [248](#), [250](#), [251](#)
  - model nugget details [250](#)
  - model nugget settings [251](#)
  - model nugget summary [251](#)
  - modeling node [246](#)
  - options [247](#)
  - predictions [248](#)
  - sequence browser [251](#)
  - sorting [251](#)
  - tabular versus transactional data [247](#)
  - time field [246](#)
- series
  - transforming [264](#)
- settings options
  - Cox regression models [207](#)
  - SLRM node [300](#)
- significance levels
  - for merging [89](#), [94](#)
- SLRM. See self-learning response models [299](#)
- SMOTE node [322](#)
- snapshot
  - creating [138](#)
- Snapshots tab [138](#)
- spark nodes [223](#), [341–343](#), [345–347](#)
- Spatio-Temporal Prediction [264](#)
- spatio-temporal prediction advanced build options [267](#)
- spatio-temporal prediction model options [268](#)
- spatio-temporal prediction output [267](#)
- split models
  - building [24](#)
  - features affected by [25](#)
  - modeling nodes [25](#)
  - versus partitioning [24](#)
- split-model nuggets
  - Summary tab [36](#)
  - viewer [40](#)
- splits
  - decision trees [74](#)
- SQL
  - export [35](#)
  - GLE models [204](#)
  - logistic regression models [168](#)
  - Random Trees models [102](#)
  - rule sets [106](#)
  - Tree-AS CHAID models [97](#)
- square root transformation
  - Time Series Modeler [291](#)
- statistical models [151](#)
- step interventions (*continued*)
  - identifying [261](#)
- stepping options
  - Cox regression models [207](#)
  - logistic regression models [166](#)
- stepwise field selection
  - Discriminant node [175](#)
- STP models
  - field options [265](#)
  - model nugget [269](#)
  - time interval options [266](#)
- STP node [264](#)
- SuperNodes
  - and model links [33](#)
- support
  - antecedent support [237](#), [250](#)
  - Apriori node [233](#)
  - association rules [240](#)
  - CARMA node [236](#), [237](#)
  - for sequences [250](#)
  - rule support [237](#), [250](#)
  - Sequence node [247](#)
- support vector machine models
  - about [305](#)
  - expert options [307](#)
  - kernel functions [305](#)
  - model nugget [308](#), [317](#)
  - model options [307](#)
  - modeling node [307](#)
  - overfitting [306](#)
  - settings [309](#)
  - tuning [306](#)
- surrogates
  - decision trees [74](#), [86](#), [94](#)
- SVM. See support vector machine models [305](#)

## T

- t statistic
  - feature selection [46](#)
- t-SNE model nuggets [329](#)
- t-SNE node [327](#), [329](#)
- tabular data
  - Apriori node [26](#)
  - CARMA node [235](#)
  - Sequence node [246](#)
  - transposing [244](#)
- TCM models
  - model nugget [279](#)
  - model nugget settings [279](#)
  - modeling node [269](#)
- TCM node [269](#)
- temporal causal model scenarios [279–283](#)
- temporal causal modeling
  - model nugget [279](#)
  - model nugget settings [279](#)
- temporal causal models
  - modeling node [269](#)
- territorial map
  - Discriminant node [174](#)
- till-roll data [243](#), [244](#)
- time field
  - CARMA node [235](#)
  - Sequence node [246](#)

- Time Series models
  - aggregation and distribution options [286](#)
  - ARIMA [288](#), [291](#)
  - ARIMA models [284](#)
  - build options [288](#)
  - build output options [292](#)
  - data specification options [285](#)
  - estimation period [288](#)
  - exponential smoothing [284](#)
  - Exponential Smoothing [288](#)
  - field options [285](#)
  - general build options [288](#)
  - missing value options [287](#)
  - model information [294](#)
  - model nugget settings [295](#)
  - model options [292](#)
  - modeling node [284](#)
  - observation options [285](#)
  - output [294](#)
  - predictor importance [294](#)
  - time interval options [286](#)
  - transfer function order [291](#)
  - transformation [291](#)
- transactional data
  - Apriori node [26](#)
  - CARMA node [235](#)
  - MS Association Rules node [26](#)
  - Sequence node [246](#)
- transfer functions
  - delay [291](#)
  - denominator orders [291](#)
  - difference orders [291](#)
  - numerator orders [291](#)
  - seasonal orders [291](#)
- Transforming association rules [254](#)
- transforming series [264](#)
- transient change outliers [262](#)
- transposing tabular output [244](#)
- tree builder
  - custom splits [74](#)
  - exporting results [81](#)
  - gains [75–78](#)
  - generating models [78](#), [79](#)
  - graph generation [108](#)
  - predictors [74](#)
  - profits [77](#)
  - ROI [77](#)
  - surrogates [74](#)
- tree depth [86](#), [94](#), [99](#)
- tree directives
  - C&R Tree node [79](#)
  - CHAID node [79](#)
  - decision trees [81](#)
  - QUEST node [79](#)
- tree map
  - decision tree models [106](#)
  - graph generation [108](#)
- Tree-AS models
  - binning [94](#)
  - build options [85](#), [94](#)
  - field options [94](#)
  - misclassification costs [95](#)
  - model information [96](#)
  - modeling node [93](#), [97](#)

- Tree-AS models (*continued*)
  - output [96](#)
  - predictor importance [96](#)
  - stopping options [95](#)
  - tree depth [94](#)
- trends
  - identifying [259](#)
- truth-table data [243](#), [244](#)
- two-headed rules [237](#)
- twoing impurity measure [89](#)
- Twostep Cluster [218–222](#)
- TwoStep cluster models
  - clustering [218](#)
  - graph generation from model nugget [229](#)
  - model nugget [218](#)
  - modeling node [216](#)
  - number of clusters [217](#)
  - options [217](#)
  - outlier handling [217](#)
  - standardization of fields [217](#)
- TwoStep-AS cluster models
  - modeling node [218](#)
- TwoStep-AS models
  - model nugget [222](#)
  - model nugget settings [223](#)

## U

- unrefined models [44](#), [47](#), [48](#)
- unrefined rule models [237](#), [242](#)
- unsupervised learning [212](#)

## V

- variable importance
  - self-learning response models [301](#)
- variance stabilizing transformation [264](#)
- varimax rotation
  - PCA/factor models [172](#)
- viewer tab
  - decision tree models [106](#)
  - graph generation [108](#)
- visualization
  - clustering models [225](#)
  - decision trees [106](#)
  - graph generation [108](#), [229](#), [240](#)
- visualize a model [148](#)
- voting rule set [109](#)

## W

- Wald statistic [166](#)
- weight fields [26](#), [27](#)
- weighted least squares [26](#)
- working model pane [136](#)

## X

- XGBoost Linear node [323](#), [324](#)
- XGBoost Tree node [324](#), [325](#), [327](#)
- XGBoost-AS node [342](#), [343](#), [345](#)





