

IBM SPSS Modeler 18.1
Python 스크립팅 및 자동화 안내서

IBM

참고

이 정보와 이 정보가 지원하는 제품을 사용하기 전에, 377 페이지의 『주의사항』에 있는 정보를 확인하십시오.

제품 정보

이 개정판은 새 개정판에 별도로 명시하지 않는 한, IBM SPSS Modeler의 버전 18, 릴리스 0, 수정 0 및 모든 후속 릴리스와 수정에 적용됩니다.

목차

제 1 장 스크립팅 및 스크립팅 언어	1	스트림	31
스크립팅 개요	1	수퍼노드 스트림	31
스크립트의 유형	1	다이어그램	31
스트림 스크립트	2	스트림 실행	32
스트림 스크립트 예: 신경망 학습	3	스크립팅 컨텍스트	32
Jython 코드 크기 한계	4	기존 노드 참조	33
독립형 스크립트	4	노드 찾기	33
독립형 스크립트 예: 모델 저장 및 로드	5	특성 설정	34
독립형 스크립트 예: 필드선택 모델 생성	5	노드 작성 및 스트림 수정	35
수퍼노드 스크립트	6	노드 작성	35
수퍼노드 스크립트 예	7	노드 링크 및 링크 해제	35
스트림에서 루핑 및 조건부 실행	7	노드 가져오기, 바꾸기 및 삭제	37
스트림에서 루핑	8	스트림에서 노드 횡단	37
스트림에서의 조건부 실행	12	항목 지우기 또는 제거	38
스크립트 실행 및 중단	13	노드에 관한 정보 얻기	39
찾기 및 바꾸기	14		
 		제 4 장 스크립팅 API	41
제 2 장 스크립팅 언어	17	스크립팅 API 소개	41
스크립팅 언어 개요	17	예 1: 사용자 정의 필터를 사용한 노드 검색	41
Python과 Jython	17	예제 2: 권한에 따라 디렉토리 또는 파일 정보를 가져오도록 허용	41
Python 스크립팅	18	메타데이터: 데이터에 관한 정보	42
작업	18	생성된 오브젝트 액세스	45
목록	19	오류 처리	46
문자열	19	스트림, 세션 및 수퍼노드 모수	47
주석	21	글로벌 값	51
명령문 구문	21	다중 스트림에 대한 작업: 독립형 스크립트	52
식별자	21		
코드 블록	22	제 5 장 스크립팅 팁	53
스크립트로 인수 전달	22	스트림 실행 수정	53
예	23	노드 루핑	53
수학적 메소드	24	IBM SPSS Collaboration and Deployment Services Repository의 오브젝트 액세스	54
ASCII가 아닌 문자 사용	25	인코딩된 비밀번호 생성	56
오브젝트 지향 프로그래밍	26	스크립트 검사	57
클래스 정의	27	명령행에서 스크립팅	57
클래스 인스턴스 작성	27	이전 릴리스와의 호환성	57
클래스 인스턴스에 속성 추가	27	스트림 실행 결과 액세스	58
클래스 속성 및 메소드 정의	28	테이블 콘텐츠 모델	58
숨겨진 변수	28	XML 콘텐츠 모델	60
상속	29	JSON 콘텐츠 모델	61
		열 통계 콘텐츠 모델 및 대응별 통계 콘텐츠 모델	63
제 3 장 IBM SPSS Modeler에서의 스크립팅	31		
스크립트의 유형	31		
스트림, 수퍼노드 스트림 및 다이어그램	31		

제 6 장 명령행 인수	67
소프트웨어 호출	67
명령행 인수 사용	67
시스템 인수	68
모수 인수	69
서버 연결 인수	70
IBM SPSS Collaboration and Deployment Services Repository 연결 인수	71
IBM SPSS Analytic Server 연결 인수	72
다중 인수 결합	72
제 7 장 특성 참조	75
특성 참조 개요	75
특성 구문	75
노드 및 스트림 특성 예	77
노드 특성 개요	78
공통 노드 특성	78
제 8 장 스트림 특성	79
제 9 장 소스 노드 특성	83
소스 노드 공통 특성	83
asimport 특성	87
cognosimport 노드 특성	88
databasenode 특성	90
datacollectionimportnode 특성	92
excelimportnode 특성	94
extensionimportnode 특성	95
fixedfilenode 특성	98
gsdata_import 노드 특성	101
sasimportnode 특성	102
simgennode 특성	102
statisticsimportnode 특성	104
tm1oddataimport 노드 특성	105
tm1import 노드 특성(더 이상 사용되지 않음)	105
twcimport 노드 특성	106
userinputnode 특성	107
variablefilenode 특성	108
xmlimportnode 특성	111
dataviewimport 특성	112
제 10 장 레코드 작업 노드 특성	115
appendnode 특성	115
aggregatenode 특성	115
balancenode 특성	116
derive_stbnode 특성	117
distinctnode 특성	119
extensionprocessnode 특성	121

mergenode 특성	122
rfmaggregatenode 특성	124
samplenode 특성	126
selectnode 특성	128
sortnode 특성	128
spacetimeboxes 특성	129
streamingtimeseries 특성	130
cplexnode 특성	137
제 11 장 필드 작업 노드 특성	139
anonymizenode 특성	139
autodatapreprenode 특성	140
astimeintervalsnode 특성	143
binningnode 특성	143
derivenode 특성	146
ensemblenode 특성	148
fillernode 특성	149
filternode 특성	150
historynode 특성	151
partitionnode 특성	152
reclassifynode 특성	153
reordernode 특성	154
reprojectnode 특성	154
restructurenode 특성	155
rfmanalysisnode 특성	155
settoflagnode 특성	157
statisticstransformnode 특성	157
timeintervalsnode 특성(더 이상 사용되지 않 음)	158
transposenode 특성	162
typenode 특성	164
제 12 장 그래프 노드 특성	171
그래프 노드 공통 특성	171
collectionnode 특성	172
distributionnode 특성	173
evaluationnode 특성	174
graphboardnode 특성	176
histogramnode 특성	178
mapvisualization 특성	179
multiplotnode 특성	183
plotnode 특성	184
timeplotnode 특성	186
webnode 특성	188
제 13 장 모델링 노드 특성	191
공통 모델링 노드 특성	191

anomalydetectionnode 특성	191
apriorinode 특성	193
associationrulesnode 특성	194
autoclassifiernode 특성	196
알고리즘 특성 설정	198
autoclusternode 특성	199
autonumericnode 특성	200
bayesnetnode 특성	201
c50node 특성	203
carmanode 특성	205
cartnode 특성	206
chaidnode 특성	208
coxregnode 특성	210
decisionlistnode 특성	212
discriminantnode 특성	213
extensionmodelnode 특성	215
factornode 특성	217
featureselectionnode 특성	219
genlinnode 특성	221
glimmnode 특성	224
gle 특성	228
kmeansnode 특성	232
knnnode 특성	233
kohonennode 특성	235
linearnode 특성	236
linearasnode 특성	237
logregnode 특성	238
lsvmnode 특성	243
neuralnetnode 특성	244
neuralnetworknode 특성	246
questnode 특성	248
randomtrees 특성	250
regressionnode 특성	251
sequencenode 특성	253
slrmnode 특성	255
statisticsmodelnode 특성	256
stpnode 특성	256
svmnode 특성	260
tcmnode 특성	261
ts 특성	265
treeas 특성	271
twostepnode 특성	273
twostepAS 특성	274
제 14 장 모델 너깃 노드 특성	277
applyanomalydetectionnode 특성	277
applyapriorinode 특성	277

applyassociationrulesnode 특성	278
applyautoclassifiernode 특성	278
applyautoclusternode 특성	279
applyautonumericnode 특성	279
applybayesnetnode 특성	279
applyc50node 특성	280
applycarmanode 특성	280
applycartnode 특성	280
applychaidnode 특성	281
applycoxregnode 특성	281
applydecisionlistnode 특성	281
applydiscriminantnode 특성	282
applyextension 특성	282
applyfactornode 특성	284
applyfeatureselectionnode 특성	284
applygeneralizedlinearnode 특성	285
applyglimmnode 특성	285
applygle 특성	286
applykmeansnode 특성	286
applyknnnode 특성	286
applykohonennode 특성	286
applylinearnode 특성	287
applylinearasnode 특성	287
applylogregnode 특성	287
applylsvmnode 특성	288
applyneuralnetnode 특성	288
applyneuralnetworknode 특성	289
applyocsvmnode 특성	289
applyquestnode 특성	289
applyrandomtrees 특성	290
applyregressionnode 특성	291
applyselflearningnode 특성	291
applysequencenode 특성	291
applysvmnode 특성	291
applystpnode 특성	292
applytcmnode 특성	292
applyts 특성	292
applytimeseriesnode 특성(더 이상 사용되지 않 음)	293
applytreeas 특성	293
applytwostepnode 특성	293
applytwostepAS 특성	294
applyxgboosttreenode 특성	294
applyxgboostlinearnode 특성	294

제 15 장 데이터베이스 모델링 노드 특성	295
Microsoft 모델링에 대한 노드 특성	295

Microsoft 모델링 노드 특성	295
Microsoft 모델 너깃 특성	298
Oracle 모델링의 노드 특성	299
Oracle 모델링 노드 특성	299
Oracle 모델 너깃 특성	305
IBM Netezza Analytics 모델링의 노드 특성	306
Netezza 모델링 노드 특성	306
Netezza 모델 너깃 특성	316
제 16 장 출력 노드 특성.	317
analysisnode 특성	317
dataauditnode 특성	318
extensionoutputnode 특성	320
matrixnode 특성	321
meansnode 특성	323
reportnode 특성	324
setglobalsnode 특성	326
simevalnode 특성	326
simfitnode 특성	327
statisticsnode 특성	328
statisticsoutputnode 특성	329
tablnode 특성	329
transformnode 특성	332
제 17 장 내보내기 노드 특성	333
공통 내보내기 노드 특성	333
asexport 특성	333
cognosexportnode 특성	334
databaseexportnode 특성	336
datacollectionexportnode 특성	340
exelexportnode 특성	340
extensionexportnode 특성	341
outputfilenode 특성	342
sasexportnode 특성	343
statisticsexportnode 특성	344
tm1odataexport 노드 특성	344
tm1export 노드 특성(더 이상 사용되지 않음)	345
xmlexportnode 특성	347
제 18 장 IBM SPSS Statistics 노드 특성	349
statisticsimportnode 특성	349
statisticstransformnode 특성	349
statisticsmodelnode 특성	350
statisticsoutputnode 특성	351
statisticsexportnode 특성	351

제 19 장 Python 노드 특성	353
smotenode 특성	353
xgboosttreenode 특성	353
xboostlinearnode 특성	355
ocsvmnode 특성	355
제 20 장 슈퍼노드 특성	359
부록 A. 노드 이름 참조	361
모델 너깃 이름	361
중복 모델 이름 피하기	363
출력 유형 이름	363
부록 B. 레거시 스크립팅에서 Python 스크립팅	
으로 마이그레이션	365
레거시 스크립팅 마이그레이션 개요	365
일반 차이점	365
스크립팅 컨텍스트	365
명령 대 함수	365
리터럴 및 주석	366
연산자	367
조건부 및 루핑	367
변수	368
노드, 출력 및 모델 유형	368
특성 이름	369
노드 참조	369
특성 가져오기 및 설정	369
스트림 편집	370
노드 작업	371
루핑	371
스트림 실행	372
파일 시스템 및 리포지토리를 통한 오브젝트 액	
세스	373
스트림 작업	374
모델 작업	374
문서 출력 작업	374
레거시 스크립팅과 Python 스크립트 사이의 기	
타 차이	375
주의사항.	377
상표	378
제품 문서의 이용 약관	379
색인	381

제 1 장 스크립팅 및 스크립팅 언어

스크립팅 개요

IBM® SPSS® Modeler에서 사용하는 스크립팅은 사용자 인터페이스의 프로세스 자동화를 위한 강력한 도구입니다. 스크립트는 마우스나 키보드로 수행하는 것과 같은 유형의 동작을 수행할 수 있으며, 스크립트를 사용하여 반복성이 높거나 수동으로 하려면 시간이 많이 걸리는 작업을 자동화할 수 있습니다.

스크립트를 사용하여 다음을 수행할 수 있습니다.

- 스트림에 노드 실행을 위한 특정 순서를 강요합니다.
- 노드의 특성을 설정할 뿐 아니라 CLEM(Control Language for Expression Manipulation)의 서브셋을 사용하여 파생을 수행합니다.
- 정상적으로 사용자 상호작용을 포함하는 조치의 자동 순차규칙을 지정합니다. 예를 들어, 모델을 작성한 후 검정할 수 있습니다.
- 견실한 사용자 상호작용이 필요한 복잡한 프로세스를 설정합니다. 예를 들어, 반복되는 모델 생성 및 검정이 필요한 교차 검증 프로시저를 설정합니다.
- 스트림을 조작하는 프로세스를 설정합니다. 예를 들어 모델 훈련 스트림을 가져와서 실행하고 대응하는 모델 검정 스트림을 자동으로 생성할 수 있습니다.

이 장에서는 IBM SPSS Modeler 인터페이스에 있는 스트림 수준 스크립트, 독립형 스크립트, 슈퍼노드 내의 스크립트의 상위 수준 설명 및 예제를 제공합니다. 뒤에 오는 장에서 스크립팅 언어, 구문, 명령에 관한 자세한 정보가 제공됩니다.

참고:

IBM SPSS Modeler 내에 IBM SPSS Statistics에서 작성된 스크립트를 가져와서 실행할 수 없습니다.

스크립트의 유형

IBM SPSS Modeler는 다음 3가지 유형의 스크립트를 사용합니다.

- 스트림 스크립트는 스트림 특성으로 저장되므로 특정 스트림과 함께 저장 및 로드됩니다. 예를 들어, 모델 너جت 훈련 및 적용 프로세스를 자동화하는 스트림 스크립트를 작성할 수 있습니다. 또한 특정 스트림이 실행될 때마다 스트림의 캔버스 콘텐츠 대신 스크립트가 실행되도록 지정할 수도 있습니다.
- 독립형 스크립트는 어떤 특정 스트림과도 연관되지 않고 외부 텍스트 파일에 저장됩니다. 예를 들어 독립형 스크립트를 사용하여 다중 스트림을 함께 조작할 수 있습니다.
- 슈퍼노드 스크립트는 슈퍼노드 스트림 특성으로 저장됩니다. 슈퍼노드 스크립트는 터미널 슈퍼노드에서만 사용할 수 있습니다. 슈퍼노드 스크립트를 사용하여 슈퍼노드 콘텐츠의 실행 순서를 제어할

수 있습니다. 비터미널(소스 또는 프로세스) 슈퍼노드의 경우, 슈퍼노드 또는 사용자의 스트림 스크립트에서 직접 포함하는 노드에 대한 특성을 정의할 수 있습니다.

스트림 스크립트

스크립트를 사용하면 특정 스트림 내에서의 작업을 사용자 정의할 수 있으며, 스크립트는 해당 스트림과 함께 저장됩니다. 스트림 스크립트를 사용하여 스트림 내의 터미널 노드에 대한 특정 실행 순서를 지정할 수 있습니다. 스트림 스크립트 대화 상자를 사용하여 현재 스트림과 함께 저장되는 스크립트를 편집합니다.

스트림 특성 대화 상자에서 스트림 스크립트 탭에 액세스하려면 다음을 수행하십시오.

1. 도구 메뉴에서 다음을 선택하십시오.

스트림 특성 > 실행

2. 실행 탭을 클릭하여 현재 스트림에 대한 스크립트에 대해 작업하십시오.

스트림 스크립트 대화 상자 맨 위에 있는 도구 모음 아이콘을 사용하여 다음 작업을 수행할 수 있습니다.

- 이미 존재하는 독립형 스크립트의 내용을 창으로 가져옵니다.
- 스크립트를 텍스트 파일로 저장합니다.
- 스크립트를 인쇄합니다.
- 기본 스크립트를 붙여씹니다.
- 스크립트를 편집합니다(실행 취소, 잘라내기, 복사, 붙여넣기 및 기타 공통 편집 기능).
- 전체 현재 스크립트를 실행합니다.
- 스크립트에서 선택된 행을 실행합니다.
- 실행 중에 스크립트를 중단합니다. (이 아이콘은 스크립트가 실행 중일 때만 사용 가능합니다.)
- 스크립트의 구문을 검사하고, 오류가 발견되면 대화 상자의 아래 분할창에서 검토를 위해 오류를 표시합니다.

참고: 버전 16.0에서부터, SPSS Modeler는 Python 스크립팅 언어를 사용합니다. 16.0 이전의 모든 버전은 SPSS Modeler에 고유한 스크립팅 언어를 사용했으며, 지금은 레거시 스크립팅이라고 부릅니다. 작업 중인 스크립트의 유형에 따라서, 실행 탭이 기본(선택적 스크립트) 실행 모드를 선택한 후 **Python** 또는 **레거시**를 선택하십시오.

스트림을 실행할 때 스크립트 실행 여부를 지정할 수 있습니다. 스크립트의 실행 순서에 따라서 스트림이 실행될 때마다 스크립트를 실행하기 위해 이 **스크립트 실행**을 선택하십시오. 이 설정은 스트림 수준에서 빠른 모델 작성을 위한 자동화를 제공합니다. 그러나, 기본 설정은 스트림 실행 중에 이 스크립트를 무시하는 것입니다. 이 **스크립트 무시** 옵션을 선택하는 경우에도 항상 이 대화 상자에서 스크립트를 직접 실행할 수 있습니다.

스크립트 편집기는 스크립트 작성에 도움을 주는 다음 기능을 포함하고 있습니다.

- 구문 강조 표시: 키워드, 리터럴 값(문자열 및 숫자 같은) 및 주석이 강조 표시됩니다.
- 행 번호 지정.
- 블록 매치. 커서가 프로그램 블록의 시작에 위치할 때 대응하는 종료 블록도 강조 표시됩니다.
- 자동 완성 제안.

구문 강조 표시기에서 사용하는 색상과 텍스트 유형은 IBM SPSS Modeler 표시화면 기본 설정을 사용하여 사용자 정의할 수 있습니다. 표시 환경 설정에 액세스하려면 **도구 > 옵션 > 사용자 옵션**을 선택하고 **구문** 탭을 선택하십시오.

컨텍스트 메뉴에서 **자동 제안**을 선택하거나 Ctrl+스페이스바를 눌러서 제안되는 구문 완료의 목록에 액세스할 수 있습니다. 커서 키를 사용하여 목록을 위 아래로 이동한 후 Enter를 눌러서 선택된 텍스트를 삽입하십시오. 기존 텍스트를 수정하지 않고 자동 제안 모드를 종료하려면 Esc 키를 누르십시오.

디버그 탭은 디버깅 메시지를 표시하며 스크립트가 실행한 후에 스크립트 상태를 평가하기 위해 사용할 수 있습니다. **디버그** 탭은 읽기 전용 텍스트 영역과 단일 행 입력 텍스트 필드로 구성됩니다. 텍스트 영역은 스크립트에 의해 예를 들어 오류 메시지 텍스트를 통해 표준 출력이나 표준 오류로 전송되는 텍스트를 표시합니다. 입력 텍스트 필드는 사용자로부터 입력을 받습니다. 이 입력이 대화 상자 안에서 가장 최근에 실행된 스크립트의 컨텍스트(스크립팅 컨텍스트라고 함) 안에서 평가됩니다. 텍스트 영역은 명령과 결과 출력을 포함하므로 사용자는 명령의 추적을 볼 수 있습니다. 텍스트 입력 필드는 항상 명령 프롬프트(레거시 스크립팅의 경우 -->)를 포함하고 있습니다.

다음 상황에서는 새 스크립팅 컨텍스트가 작성됩니다.

- 이 스크립트 실행 또는 선택한 행 실행을 사용하여 스크립트를 실행합니다.
- 스크립팅 언어가 변경됩니다.

새 스크립팅 컨텍스트가 작성되는 경우 텍스트 영역이 지워집니다.

참고: 스크립트 분할창 외부에서 스트림을 실행하면 스크립트 분할창의 스크립트 컨텍스트가 수정되지 않습니다. 해당 실행의 일부로 작성된 모든 변수의 값은 스크립트 대화 상자 안에서 볼 수 없습니다.

스트림 스크립트 예: 신경망 학습

스트림을 사용하여 신경망 모델을 실행할 때 학습할 수 있습니다. 일반적으로 모델을 검정하기 위해 모델링 노드를 실행하여 모델을 스트림에 추가하고 적당한 연결을 작성하고 분석 노드를 실행할 수 있습니다.

IBM SPSS Modeler 스크립트를 사용하여 모델 너트를 작성한 후 검정하는 프로세스를 자동화할 수 있습니다. 예를 들어, druglearn.str 데모 스트림(IBM SPSS Modeler 설치의 /Demos/streams/ 폴더에서 사용 가능함)을 검정하기 위한 다음 스트림 스크립트는 스트림 특성 대화 상자(**도구 > 스트림 특성 > 스크립트**)에서 실행할 수 있습니다.

```

stream = modeler.script.stream()
neuralnetnode = stream.findByType("neuralnetwork", None)
results = []
neuralnetnode.run(results)
appliernode = stream.createModelApplierAt(results[0], "Drug", 594, 187)
analysisnode = stream.createAt("analysis", "Drug", 688, 187)
typenode = stream.findByType("type", None)
stream.linkBetween(appliernode, typenode, analysisnode)
analysisnode.run([])

```

다음 글머리 기호는 이 스크립트 예제의 각 행을 설명합니다.

- 첫 번째 행은 현재 스트림을 가리키는 변수를 정의합니다.
- 2행에서 스크립트는 신경망 작성기 노드를 찾습니다.
- 3행에서 스크립트는 실행 결과가 저장될 수 있는 목록을 작성합니다.
- 4행에서 신경망 모델 너깃이 작성됩니다. 이것은 3행에서 정의된 목록에 저장됩니다.
- 5행에서 모델 적용 노드가 모델 너깃에 대해 작성되고 스트림 캔버스에 배치됩니다.
- 6행에서 Drug라는 분석 노드가 작성됩니다.
- 7행에서 스크립트가 유형 노드를 찾습니다.
- 8행에서 스크립트는 유형 노드와 분석 노드 사이에 5행에서 작성된 모델 적용 노드를 연결합니다.
- 마지막으로, 분석 노드가 실행되어 분석 보고서를 생성합니다.

스크립트를 사용하여 빈 캔버스에서 시작하여 스크래치로부터 스트림을 작성하고 실행할 수 있습니다. 일반적으로 스크립팅 언어에 대해 자세히 알려면 스크립팅 언어 개요를 참조하십시오.

Jython 코드 크기 한계

Jython에서 각 스크립트를 Java 바이트 코드로 컴파일합니다. 그러면 JVM(Java Virtual Machine)에서 이 코드를 실행합니다. 그러나 Java에서는 단일 바이트 코드 파일의 크기에 제한을 둡니다. Jython에서 바이트 코드를 로드하려고 하면 JVM이 충돌할 수 있습니다. IBM SPSS Modeler에서 이 문제가 발생하지 않게 할 수 없습니다.

좋은 코딩 방법을 사용하여 Jython 스크립트를 작성하십시오(예: 공통 중간 값을 컴퓨팅하기 위해 변수나 함수를 사용하여 중복 코드를 최소화). 필요한 경우 여러 소스 파일에 코드를 분할하거나 모듈을 사용하여 코드를 정의해야 할 수도 있습니다. 모듈은 개별 바이트 코드 파일로 컴파일되기 때문입니다.

독립형 스크립트

독립형 스크립트 대화 상자는 텍스트 파일로 저장되는 스크립트를 작성 또는 편집하기 위해 사용됩니다. 파일의 이름을 표시하고 스크립트 로드, 저장, 가져오기, 실행을 위한 기능을 제공합니다.

독립형 스크립트 대화 상자에 액세스하려면 다음을 수행하십시오.

주 메뉴에서 다음을 선택하십시오.

도구 > 독립형 스크립트

스트림 스크립트의 경우와 동일한 도구 모음 및 스크립트 구문 검사 옵션을 독립형 스크립트에 사용할 수 있습니다. 자세한 정보는 2 페이지의 『스트림 스크립트』의 내용을 참조하십시오.

독립형 스크립트 예: 모델 저장 및 로드

독립형 스크립트는 스트림 조작에 유용합니다. 두 개의 스트림이 있는데, 하나는 모델을 작성하고 다른 하나는 그래프를 사용하여 기존 데이터 필드를 갖는 첫 번째 스트림에서 생성된 규칙 세트를 탐색한다고 가정하십시오. 이 시나리오의 독립형 스크립트는 다음과 유사할 수 있습니다.

```
taskrunner = modeler.script.session().getTaskRunner()

# Modify this to the correct Modeler installation Demos folder.
# Note use of forward slash and trailing slash.
installation = "C:/Program Files/IBM/SPSS/Modeler/18.1/Demos/"

# First load the model builder stream from file and build a model
druglearn_stream = taskrunner.openStreamFromFile(installation + "streams/druglearn.str", True)
results = []
druglearn_stream.findByType("c50", None).run(results)

# Save the model to file
taskrunner.saveModelToFile(results[0], "rule.gm")

# Now load the plot stream, read the model from file and insert it into the stream
drugplot_stream = taskrunner.openStreamFromFile(installation + "streams/drugplot.str", True)
model = taskrunner.openModelFromFile("rule.gm", True)
modelapplier = drugplot_stream.createModelApplier(model, "Drug")

# Now find the plot node, disconnect it and connect the
# model applier node between the derive node and the plot node
derivenode = drugplot_stream.findByType("derive", None)
plotnode = drugplot_stream.findByType("plot", None)
drugplot_stream.disconnect(plotnode)
modelapplier.setPositionBetween(derivenode, plotnode)
drugplot_stream.linkBetween(modelapplier, derivenode, plotnode)
plotnode.setPropertyValue("color_field", "$C-Drug")
plotnode.run([])
```

참고: 일반적으로 스크립팅 언어에 대해 자세히 알려면 스크립팅 언어 개요를 참조하십시오.

독립형 스크립트 예: 필드선택 모델 생성

이 예는 공백 캔버스에서 시작하여 필드선택 모델을 생성하는 스트림을 작성하고, 모델을 적용하고, 지정된 목표에 상대적인 15개의 가장 중요한 필드를 나열하는 테이블을 작성합니다.

```
stream = modeler.script.session().createProcessorStream("featureselection", True)

statisticsimportnode = stream.createAt("statisticsimport", "Statistics File", 150, 97)
statisticsimportnode.setPropertyValue("full_filename", "$CLEO_DEMOS/customer_dbase.sav")

typenode = stream.createAt("type", "Type", 258, 97)
typenode.setKeyedPropertyValue("direction", "response_01", "Target")

featureselectionnode = stream.createAt("featureselection", "Feature Selection", 366, 97)
featureselectionnode.setPropertyValue("top_n", 15)
featureselectionnode.setPropertyValue("max_missing_values", 80.0)
```

```

featureselectionnode.setPropertyValue("selection_mode", "TopN")
featureselectionnode.setPropertyValue("important_label", "Check Me Out!")
featureselectionnode.setPropertyValue("criteria", "Likelihood")

stream.link(statisticsimportnode, typenode)
stream.link(typenode, featureselectionnode)
models = []
featureselectionnode.run(models)

# Assumes the stream automatically places model apply nodes in the stream
applynode = stream.findByType("applyfeatureselection", None)
tablenode = stream.createAt("table", "Table", applynode.getXPosition() + 96, applynode.getYPosition())
stream.link(applynode, tablenode)
tablenode.run([])

```

스크립트는 데이터에서 읽을 소스 노드를 작성하고, 유형 노드를 사용하여 response_01 필드의 역할 (방향)을 Target으로 설정한 후, 필드선택 노드를 작성 및 실행합니다. 스크립트는 또한 노드를 연결하고 각각을 스트림 캔버스에 배치하여 읽을 수 있는 레이아웃을 생성합니다. 결과 모델 너트는 테이블 노드에 연결되는데, 이것은 selection_mode 및 top_n 특성에 의해 판별되는 15개의 가장 중요한 필드를 나열합니다. 자세한 정보는 219 페이지의 『featureselectionnode 특성』의 내용을 참조하십시오.

수퍼노드 스크립트

IBM SPSS Modeler의 스크립팅 언어를 사용하여 임의의 터미널 수퍼노드 내에서 스크립트를 작성하고 저장할 수 있습니다. 이들 스크립트는 터미널 수퍼노드에만 사용할 수 있으며 보통 템플릿 스트림을 작성할 때나 수퍼노드 콘텐츠를 위한 특별한 실행 순서를 강요하기 위해 사용합니다. 수퍼노드 스크립트를 사용하면 스트림 안에서 둘 이상의 스크립트를 실행할 수도 있습니다.

예를 들어, 복잡한 스트림에 대한 실행 순서를 지정할 필요가 있었고 수퍼노드가 SetGlobals 노드를 포함한 여러 노드를 포함하고 있다고 가정합니다. SetGlobals 노드는 구성 노드에서 사용하는 새 필드를 파생하기 전에 실행되어야 합니다. 이 경우, SetGlobals 노드를 처음으로 실행하는 수퍼노드 스크립트를 작성할 수 있습니다. 평균이나 표준 편차 같이 이 노드가 계산하는 값을 구성 노드가 실행될 때 사용할 수 있습니다.

수퍼노드 스크립트 내에서, 다른 스크립트와 같은 방식으로 노드 특성을 지정할 수 있습니다. 또는 스트림 스크립트로부터 직접 임의의 수퍼노드 또는 캡슐화 노드의 특성을 변경 및 정의할 수 있습니다. 자세한 정보는 359 페이지의 제 20 장 『수퍼노드 특성』의 내용을 참조하십시오. 이 방법은 소스 및 프로세스 수퍼노드뿐 아니라 터미널 수퍼노드에 대해 작동합니다.

참고: 터미널 수퍼노드만이 자신의 스크립트를 실행할 수 있으므로, 수퍼노드 대화 상자의 스크립트 탭은 터미널 수퍼노드에만 사용할 수 있습니다.

주 캔버스에서 수퍼노드 스크립트 대화 상자를 열려면 다음을 수행하십시오.

스트림 캔버스에서 터미널 수퍼노드를 선택하고, 수퍼노드 메뉴에서 다음을 선택하십시오.

수퍼노드 스크립트...

확대된 슈퍼노드 캔버스에서 슈퍼노드 스크립트 대화 상자를 열려면 다음을 수행하십시오.

슈퍼노드 캔버스를 마우스 오른쪽 단추로 클릭하고, 컨텍스트 메뉴에서 다음을 선택하십시오.

슈퍼노드 스크립트...

슈퍼노드 스크립트 예

다음 슈퍼노드 스크립트는 슈퍼노드 안에 있는 터미널 노드가 실행될 순서를 선언합니다. 이 순서는 전역값 설정 노드가 처음 실행되어 이 노드가 계산하는 값이 다른 노드가 실행될 때 사용할 수 있도록 보장합니다.

```
execute 'Set Globals'  
execute 'gains'  
execute 'profit'  
execute 'age v. $CC-pep'  
execute 'Table'
```

슈퍼노드 잠금 및 잠금 해제

다음 예에서는 슈퍼노드를 잠그고 잠금 해제할 수 있는 방법을 보여줍니다.

```
stream = modeler.script.stream()  
superNode=stream.findByID('id854RNTSD5MB')  
# unlock one super node  
print 'unlock the super node with password abcd'  
if superNode.unlock('abcd'):  
    print 'unlocked.'  
else:  
    print 'invalid password.'  
# lock one super node  
print 'lock the super node with password abcd'  
superNode.lock('abcd')
```

스트림에서 루핑 및 조건부 실행

버전 16.0에서부터, SPSS Modeler에서 스크립팅 언어로 직접 명령어를 쓸 필요 없이 다양한 대화 상자에서 값을 선택하여 스트림 내에서 몇 가지 기본 스크립트를 작성할 수 있습니다. 이 방식으로 작성할 수 있는 스크립트의 두 가지 기본 유형은 단순 루프와 조건이 충족된 경우에 노드를 실행하는 방법입니다.

스트림 안에서 루핑과 조건부 실행 규칙을 모두 결합할 수 있습니다. 예를 들어, 전세계 제조업체의 자동차 매출과 관련된 데이터가 있습니다. 스트림에서 데이터를 처리하는 루프를 설정하여 제조업체 국가별로 세부사항을 식별하고 모델별 매출액, 제조업체 및 엔진 크기별 배기량 수준 등과 같은 세부사항을 표시하는 여러 가지 그래프로 데이터를 출력할 수 있습니다. 유럽 정보 분석에만 관심이 있는 경우 미국 및 아시아에 기반한 제조업체에 대해 그래프가 작성되지 않도록 하는 루핑에 조건을 추가할 수도 있습니다.

참고: 루핑과 조건부 실행이 둘 다 백그라운드 스크립트에 기반하기 때문에 스트림이 실행될 때 전체 스트림에만 적용됩니다.

- 루핑 루핑을 사용하여 반복 작업을 자동화할 수 있습니다. 예를 들어, 주어진 수의 노드를 스트림에 추가하고 매번 하나의 노드 모수 변경을 의미할 수 있습니다. 또는 다음 예에서와 같이 주어진 횟수 동안 스트림 또는 분기의 실행을 계속해서 제어할 수 있습니다.
 - 스트림을 주어진 횟수만큼 실행하고 매번 소스를 변경합니다.
 - 스트림을 주어진 횟수만큼 실행하고 매번 변수의 값을 변경합니다.
 - 스트림을 주어진 횟수만큼 실행하고 각 실행에서 하나의 추가 필드를 입력합니다.
 - 모델을 주어진 횟수만큼 작성하고 매번 설정을 변경합니다.
- 조건부 실행 이것을 사용하여 사용자가 사전 정의하는 조건을 기반으로 터미널 노드가 실행하는 방법을 제어할 수 있습니다. 다음과 같은 예가 포함될 수 있습니다.
 - 주어진 값이 true 또는 false인지 여부를 기반으로, 노드가 실행되는지를 제어합니다.
 - 노드의 루핑이 병렬로 또는 순차적으로 실행되는지 여부를 정의합니다.

루핑과 조건부 실행은 둘 다 스트림 특성 대화 상자 내의 실행 탭에서 설정됩니다. 조건부 또는 루핑 요구 사항에서 사용하는 모든 노드는 스트림 캔버스에서 추가 기호가 첨부된 상태로 표시되어 루핑 및 조건부 실행에 참여하고 있음을 표시합니다.

3가지 방법 중 하나로 실행 탭에 액세스할 수 있습니다.

- 주 대화 상자의 맨 위에 있는 메뉴 사용:
 1. 도구 메뉴에서 다음을 선택하십시오.

스트림 특성 > 실행

2. 현재 스트림에 대한 스크립트에 대해 작업하려면 실행 탭을 클릭하십시오.
- 스트림 내에서:
 1. 노드를 마우스 오른쪽 단추로 클릭하고 루핑/조건부 실행을 선택하십시오.
 2. 관련 하위 메뉴 옵션을 선택하십시오.
 - 주 대화 상자의 맨 위에 있는 그래픽 도구 모음에서 스트림 특성 아이콘을 클릭하십시오.

처음으로 루핑 또는 조건부 실행 세부사항을 설정한 경우, 실행 탭에서 루핑/조건부 실행 실행 모드를 선택한 후 조건부 또는 루핑 하위 탭을 선택하십시오.

스트림에서 루핑

루핑을 사용하여 스트림에서 반복 작업을 자동화할 수 있습니다. 다음과 같은 예가 포함될 수 있습니다.

- 스트림을 주어진 횟수만큼 실행하고 매번 소스를 변경합니다.
- 스트림을 주어진 횟수만큼 실행하고 매번 변수의 값을 변경합니다.
- 스트림을 주어진 횟수만큼 실행하고 각 실행에서 하나의 추가 필드를 입력합니다.

- 모델을 주어진 횟수만큼 작성하고 매번 설정을 변경합니다.

스트림 실행 탭의 루핑 하위 탭에서 충족될 조건을 설정합니다. 하위 탭을 표시하려면 루핑/조건부 실행 실행 모드를 선택하십시오.

사용자가 정의하는 모든 루핑 요구 사항은 루핑/조건부 실행 실행 모드가 설정된 경우 스트림을 실행할 때 적용됩니다. 선택적으로 루핑 요구 사항을 위한 스크립트 코드를 생성하고 루핑 하위 탭의 오른쪽 하단 모서리에 있는 붙여넣기...를 클릭하여 이를 스크립트 편집기에 붙여넣을 수 있습니다. 기본 실행 탭 화면은 변경되어 탭의 맨 위쪽에 스크립트와 함께 기본(선택적 스크립트) 실행 모드를 표시합니다. 이것은 스크립트 편집기에서 추가로 사용자 정의할 수 있는 스크립트를 생성하기 전에 다양한 루핑 대화 상자 옵션을 사용하여 루핑 구조를 정의할 수 있음을 의미합니다. 붙여넣기...를 클릭할 때 사용자가 정의한 모든 조건부 실행 요구 사항도 생성된 스크립트에 표시됨을 참고하십시오.

중요사항: IBM SPSS Collaboration and Deployment Services 작업에서 스트림을 실행하는 경우 SPSS Modeler 스트림에서 설정하는 루핑 변수가 대체될 수 있습니다. 이것은 IBM SPSS Collaboration and Deployment Services 작업 편집기 항목이 SPSS Modeler 항목을 대체하기 때문입니다. 예를 들어, 각 루프에 대해 서로 다른 출력 파일 이름을 작성하기 위해 스트림에 루핑 변수를 설정하는 경우, 파일이 SPSS Modeler에서 올바르게 이름 지정되지만 IBM SPSS Collaboration and Deployment Services 배포 관리자의 결과 탭에 입력되는 수정 항목에 의해 대체됩니다.

루프 설정

1. 반복 키를 사용하여 스트림에서 수행될 주 루핑 구조를 정의하십시오. 자세한 정보는 반복 키 작성을 참조하십시오.
2. 필요한 경우 하나 이상의 반복 변수를 정의하십시오. 자세한 정보는 반복 변수 작성을 참조하십시오.
3. 사용자가 작성한 반복 및 모든 변수가 하위 탭의 주 본문에 표시됩니다. 기본적으로, 반복은 나타나는 순서대로 실행됩니다. 반복을 목록에서 위나 아래로 이동하려면 해당 항목을 클릭하여 선택한 후 하위 탭의 오른쪽 열에 있는 위로 또는 아래로 화살표를 사용하여 순서를 변경하십시오.

스트림에서 루핑을 위한 반복 키 작성

반복 키를 사용하여 스트림에서 수행될 주 루핑 구조를 정의합니다. 예를 들어, 자동차 영업을 분석 중인 경우 제조업체 국가 스트림 모수를 작성하고 이것을 반복 키로 사용할 수 있습니다. 해당 스트림이 실행될 때 이 키는 각 반복 동안 사용자 데이터에 있는 각각의 다른 국가 값으로 설정됩니다. 키를 설정하려면 반복 키 정의 대화 상자를 사용하십시오.

대화 상자를 열려면 루핑 하위 탭의 왼쪽 하단 모서리에 있는 반복 키... 단추를 선택하거나, 스트림에 있는 임의의 노드에서 마우스 오른쪽 단추를 클릭하고 루핑/조건부 실행 > 반복 키 정의(필드) 또는 루핑/조건부 실행 > 반복 키 정의(값)를 선택하십시오. 스트림에서 대화 상자를 여는 경우, 노드의 이름 같은 일부 필드는 자동으로 완료될 수 있습니다.

반복 키를 설정하려면 다음 필드를 완료하십시오.

반복 위치. 다음 옵션 중 하나를 선택할 수 있습니다.

- **스트림 모수 - 필드.** 기존 스트림 모수의 값을 차례로 각 지정된 필드로 설정하는 루프를 작성하려면 이 옵션을 사용하십시오.
- **스트림 모수 - 값.** 기존 스트림 모수의 값을 차례로 각 지정된 값으로 설정하는 루프를 작성하려면 이 옵션을 사용하십시오.
- **노드 특성 - 필드.** 노드 특성의 값을 차례로 각 지정된 필드로 설정하는 루프를 작성하려면 이 옵션을 사용하십시오.
- **노드 특성 - 값.** 노드 특성의 값을 차례로 각 지정된 값으로 설정하는 루프를 작성하려면 이 옵션을 사용하십시오.

설정 대상. 루프가 실행될 때마다 값을 갖는 항목을 선택하십시오. 다음 옵션 중 하나를 선택할 수 있습니다.

- **모수.** 스트림 모수 - 필드 또는 스트림 모수 - 값을 선택하는 경우에만 사용할 수 있습니다. 사용 가능 목록에서 필수 모수를 선택하십시오.
- **노드.** 노드 특성 - 필드 또는 노드 특성 - 값을 선택하는 경우에만 사용할 수 있습니다. 루프를 설정할 노드를 선택하십시오. 찾아보기 단추를 클릭하여 노드 선택 대화 상자를 열고 원하는 노드를 선택하십시오. 나열된 노드가 너무 많은 경우, 소스, 프로세스, 그래프, 모델링, 출력, 내보내기 또는 모델 적용 노드 범주 중 하나를 선택하여 노드의 특정 유형만 표시하도록 화면을 필터링할 수 있습니다.
- **특성.** 노드 특성 - 필드 또는 노드 특성 - 값을 선택하는 경우에만 사용할 수 있습니다. 사용 가능 목록에서 노드의 특성을 선택하십시오.

사용할 필드. 스트림 모수 - 필드 또는 노드 특성 - 값을 선택하는 경우에만 사용할 수 있습니다. 반복 값을 제공하기 위해 사용할 노드 내의 하나 이상의 필드를 선택하십시오. 다음 옵션 중 하나를 선택할 수 있습니다.

- **노드.** 스트림 모수 - 필드를 선택하는 경우에만 사용할 수 있습니다. 루프를 설정하려는 세부사항을 포함하는 노드를 선택하십시오. 찾아보기 단추를 클릭하여 노드 선택 대화 상자를 열고 원하는 노드를 선택하십시오. 나열된 노드가 너무 많은 경우, 소스, 프로세스, 그래프, 모델링, 출력, 내보내기 또는 모델 적용 노드 범주 중 하나를 선택하여 노드의 특정 유형만 표시하도록 화면을 필터링할 수 있습니다.
- **필드 목록.** 오른쪽 열의 목록 단추를 클릭하여 필드 선택 대화 상자를 표시하십시오. 그 안에서 반복 데이터를 제공하기 위한 노드의 필드를 선택합니다. 자세한 정보는 12 페이지의 『반복할 필드 선택』을 참조하십시오.

사용할 값. 스트림 모수 - 값 또는 노드 특성 - 값을 선택하는 경우에만 사용할 수 있습니다. 반복 값으로 사용할 선택된 필드 내에 있는 하나 이상의 값을 선택하십시오. 다음 옵션 중 하나를 선택할 수 있습니다.

- **노드.** 스트림 모수 - 값을 선택하는 경우에만 사용할 수 있습니다. 루프를 설정하려는 세부사항을 포함하는 노드를 선택하십시오. 찾아보기 단추를 클릭하여 노드 선택 대화 상자를 열고 원하는 노

드를 선택하십시오. 나열된 노드가 너무 많은 경우, 소스, 프로세스, 그래프, 모델링, 출력, 내보내기 또는 모델 적용 노드 범주 중 하나를 선택하여 노드의 특정 유형만 표시하도록 화면을 필터링할 수 있습니다.

- **필드 목록.** 반복 데이터를 제공할 노드의 필드를 선택하십시오.
- **값 목록.** 오른쪽 열의 목록 단추를 클릭하여 값 선택 대화 상자를 표시하십시오. 그 안에서 반복 데이터를 제공하기 위한 필드의 값을 선택합니다.

스트림에서 루핑을 위한 반복 변수 작성

반복 변수를 사용하여 루프가 실행될 때마다 스트림 변수의 값이나 스트림 내의 선택된 노드의 특성을 변경할 수 있습니다. 예를 들어, 스트림 루프가 자동차 영업 데이터를 분석하고 제조업체 국가를 반복 키로 사용 중인 경우, 모델별 매출을 표시하는 하나의 그래프 출력과 배기가스 정보를 표시하는 다른 그래프 출력을 가질 수 있습니다. 이들 경우에 스웨덴 차량 배기물과 모델별 일본차 매출 같은 결과 그래프에 대한 새 제목을 생성하는 반복 변수를 작성할 수 있습니다. 필요한 임의의 변수를 설정하려면 반복 변수 정의 대화 상자를 사용하십시오.

대화 상자를 열려면 루핑 하위 탭의 왼쪽 하단 모서리에 있는 **변수 추가...** 단추를 선택하거나 스트림의 임의의 노드를 마우스 오른쪽 단추로 클릭하고 **루핑/조건부 실행 > 반복 변수 정의**를 선택하십시오.

반복 변수를 설정하려면 다음 필드를 완료하십시오.

변경. 수정하려는 속성의 유형을 선택하십시오. **스트림 변수** 또는 **노드 특성** 중에서 선택할 수 있습니다.

- **스트림 변수**를 선택하는 경우, 필수 변수를 선택한 후 다음 옵션 중 하나를 사용하여(사용자 스트림에서 사용 가능한 경우) 루프의 각 반복에서 설정할 해당 변수의 값을 정의하십시오.
 - **글로벌 변수.** 스트림 변수가 설정될 글로벌 변수를 선택하십시오.
 - **테이블 출력 셀.** 테이블 출력 셀에서 값이 될 스트림 변수를 설정하려면 목록에서 테이블을 선택하고 사용할 **행과 열**을 입력하십시오.
 - **수동으로 입력.** 각 반복에서 사용할 이 변수의 값을 수동으로 입력하려는 경우 이것을 선택하십시오. 루핑 하위 탭으로 돌아갈 때 필수 텍스트를 입력하는 새 열이 작성됩니다.
- **노드 특성을** 선택하는 경우 필수 노드 및 해당 특성 중 하나를 선택한 후 해당 특성에 대해 사용할 값을 설정하십시오. 다음 옵션 중 하나를 사용하여 새 특성 값을 설정하십시오.
 - **단독.** 특성 값이 반복 키 값을 사용합니다. 자세한 정보는 9 페이지의 『스트림에서 루핑을 위한 반복 키 작성』을 참조하십시오.
 - **시스템에 대한 접두문자로.** 반복 키 값을 시스템 필드에 입력하는 것의 접두문자로 사용합니다.
 - **시스템에 대한 접미문자로.** 반복 키 값을 시스템 필드에 입력하는 것의 접미문자로 사용합니다.

접두문자 또는 접미문자 옵션을 선택하는 경우 시스템 필드에 추가 텍스트를 추가하라는 프롬프트가 표시됩니다. 예를 들어, 반복 키 값이 제조업체 국가이고 **시스템에 대한 접두문자로**를 선택하는 경우, 이 필드에 - 모델별 매출을 입력할 수 있습니다.

반복할 필드 선택

반복을 작성할 때 필드 선택 대화 상자를 사용하여 하나 이상의 필드를 선택할 수 있습니다.

정렬기준 다음 옵션 중 하나를 선택하여 볼 수 있는 필드를 정렬할 수 있습니다.

- 기본 데이터 스트림이 필드를 현재 노드로 전달한 대로 필드의 순서를 봅니다.
- 이름 영문자 순서를 사용하여 필드를 보도록 정렬합니다.
- 유형 측정 수준에 의해 정렬된 필드를 봅니다. 이 옵션은 특정 측정 수준을 갖는 필드를 선택할 때 유용합니다.

한 번에 하나씩 목록에서 필드를 선택하거나 Shift-클릭 및 Ctrl-클릭 방법을 사용하여 다중 필드를 선택하십시오. 또한 목록 아래의 단추를 사용하여 측정 수준을 기반으로 필드 그룹을 선택하거나 테이블의 모든 필드를 선택 또는 선택 취소할 수 있습니다.

선택할 수 있는 필드가 필터링되어 스트림 모수 또는 사용하려는 노드 특성에 적합한 필드만 표시함을 참고하십시오. 예를 들어, String의 저장 유형을 갖는 스트림 모수를 사용 중인 경우 String의 저장 유형을 갖는 필드만 표시됩니다.

스트림에서의 조건부 실행


조건부 실행을 사용하면 사용자가 정의하는 조건과 매치하는 스트림 내용을 기반으로 터미널 노드가 실행되는 방법을 제어할 수 있습니다. 예는 다음을 포함할 수 있습니다.

- 주어진 값이 true 또는 false인지 여부를 기반으로, 노드가 실행되는지를 제어합니다.
- 노드의 루핑이 병렬로 또는 순차적으로 실행되는지 여부를 정의합니다.

스트림 실행 탭의 조건부 하위 탭에서 충족될 조건을 설정합니다. 하위 탭을 표시하려면 루핑/조건부 실행 실행 모드를 선택하십시오.

사용자가 정의하는 모든 조건부 실행 요구 사항은 루핑/조건부 실행 실행 모드가 설정된 경우 스트림을 실행할 때 적용됩니다. 선택적으로 조건부 실행 요구 사항을 위한 스크립트 코드를 생성하고 조건부 하위 탭의 오른쪽 하단 모서리에 있는 붙여넣기...를 클릭하여 이를 스크립트 편집기에 붙여넣을 수 있습니다. 기본 실행 탭 화면은 변경되어 탭의 맨 위쪽에 스크립트와 함께 기본(선택적 스크립트) 실행 모드를 표시합니다. 이것은 스크립트 편집기에서 추가로 사용자 정의할 수 있는 스크립트를 생성하기 전에 다양한 루핑 대화 상자 옵션을 사용하여 조건을 정의할 수 있음을 의미합니다. 붙여넣기...를 클릭할 때 사용자가 정의한 모든 루핑 요구 사항도 생성된 스크립트에 표시됨을 참고하십시오.

조건을 설정하려면 다음을 수행하십시오.

1. 조건부 하위 탭의 오른쪽 열에서 새 조건 추가 단추  를 클릭하여 조건부 실행 명령문 추가 대화 상자를 여십시오. 이 대화 상자에서 노드가 실행되기 위해서 충족되어야 하는 조건을 지정합니다.
2. 조건부 실행 명령문 추가 대화 상자에서 다음을 지정하십시오.

- a. **노드.** 조건부 실행을 설정하려는 노드를 선택하십시오. 찾아보기 단추를 클릭하여 노드 선택 대화 상자를 열고 원하는 노드를 선택하십시오. 너무 많은 노드가 나열되는 경우 내보내기, 그래프, 모델링 또는 출력 노드 범주 중 하나로 노드를 표시하도록 화면을 필터링할 수 있습니다.
- b. **기준 조건.** 노드가 실행되기 위해 충족해야 하는 조건을 지정하십시오. **스트림 모수**, **글로벌 변수**, **테이블 출력 셀** 또는 **항상 참**의 4가지 옵션 중 하나를 선택할 수 있습니다. 대화 상자의 하반부에 입력하는 세부사항은 사용자가 선택하는 조건에 의해 제어됩니다.
 - **스트림 모수.** 사용 가능 목록에서 모수를 선택한 후 해당 모수에 대한 **연산자**를 선택하십시오. 예를 들어, 연산자는 More than(초과), Equals(같음), Less than(미만), Between(사이) 등일 수 있습니다. 그런 다음 연산자에 따라서 **값** 또는 **최소 및 최대값**을 입력합니다.
 - **글로벌 변수.** 사용 가능 목록에서 변수를 선택하십시오. 예를 들어, 평균, 합계, 최소값, 최대값 또는 표준 편차를 포함할 수 있습니다. 그런 다음 필요한 **연산자**와 **값**을 선택합니다.
 - **테이블 출력 셀.** 사용 가능 목록에서 테이블 노드를 선택한 후 테이블에서 **행과 열**을 선택하십시오. 그런 다음 필요한 **연산자**와 **값**을 선택합니다.
 - **항상 참.** 노드가 항상 실행되어야 하는 경우 이 옵션을 선택하십시오. 이 옵션을 선택하면 선택할 추가 모수가 없습니다.
3. 필요한 모든 조건을 설정할 때까지 필요한 대로 1단계와 2단계를 반복하십시오. 사용자가 선택한 노드 및 해당 노드가 실행되기 전에 충족되어야 하는 조건이 각각 **노드 실행** 및 **이 조건이 참인 경우 열**에 있는 하위 탭의 주 본문에 표시됩니다.
4. 기본적으로, 노드 및 조건은 나타나는 순서대로 실행됩니다. 노드 및 조건을 목록에서 위나 아래로 이동하려면 해당 항목을 클릭하여 선택한 후 하위 탭의 오른쪽 열에 있는 위로 또는 아래로 화살표를 사용하여 순서를 변경하십시오.

또한, 조건부 하위 탭의 맨 아래에서 다음 옵션을 설정할 수 있습니다.

- **순서대로 모두 평가.** 각 조건을 하위 탭에 표시되는 순서에 따라서 평가하려면 이 옵션을 선택하십시오. 조건이 "참"인 것으로 확인된 노드는 모두가 모든 조건이 평가된 후에 실행됩니다.
- **한 번에 하나씩 실행.** 순서대로 모두 평가가 선택된 경우에만 사용할 수 있습니다. 이것을 선택하는 것은 조건이 "참"으로 평가되는 경우 해당 조건과 연관된 노드가 다음 조건이 평가되기 전에 실행됨을 의미합니다.
- **첫 번째 적중이 발생할 때까지 평가.** 이것을 선택하는 것은 사용자가 지정한 조건에서 "참" 평가를 리턴하는 첫 번째 노드만 실행됨을 의미합니다.

스크립트 실행 및 중단

많은 스크립트 실행 방법을 사용할 수 있습니다. 예를 들어 스트림 스크립트나 독립형 스크립트 대화 상자에서 "이 스크립트 실행" 단추가 전체 스크립트를 실행합니다.



그림 1. 이 스크립트 실행

"선택된 행 실행" 단추는 스크립트에 선택한 단일 행 또는 인접한 행의 블록을 실행합니다.



그림 2. 선택된 행 실행 단추

다음 방법 중 하나를 사용하여 스크립트를 실행할 수 있습니다.

- 스트림 스크립트 또는 독립형 스크립트 대화 상자 안에서 "이 스크립트 실행" 또는 "선택된 행 실행" 단추를 클릭하십시오.
- 이 스크립트 실행이 기본 실행 방법으로 설정된 스트림을 실행하십시오.
- 대화식 모드에서 시작 시에 `-execute` 플래그를 사용하십시오. 자세한 정보는 67 페이지의 『명령 행 인수 사용』의 내용을 참조하십시오.

참고: 슈퍼노드 스크립트 대화 상자 안에서 이 스크립트 실행을 선택한 경우 슈퍼노드가 실행될 때 슈퍼노드 스크립트가 실행됩니다.

스크립트 실행 중단

스트림 스크립트 대화 상자 안에서 스크립트 실행 동안 빨간색 중단 단추가 활성화됩니다. 이 단추를 사용하여 스크립트 및 임의의 현재 스트림의 실행을 중단할 수 있습니다.

찾기 및 바꾸기

찾기/바꾸기 대화 상자는 스크립트 또는 표현식 텍스트를 편집하는 위치(스크립트 편집기, CLEM 표현식 작성기 포함)에서, 또는 보고서 노드에서 템플릿을 정의할 때 사용할 수 있습니다. 이러한 영역에서 편집하는 경우 `Ctrl+F`를 눌러 이 대화 상자에 액세스하고 커서가 텍스트 영역에서 초점을 갖고 있는지 확인하십시오. 예를 들어, 채움 노드에서 작업하는 경우 설정 탭의 텍스트 영역에서 또는 표현식 작성기의 텍스트 필드에서 대화 상자에 액세스할 수 있습니다.

1. 텍스트 영역에 커서를 두고 `Ctrl+F`를 눌러서 찾기/바꾸기 대화 상자에 액세스하십시오.
2. 검색할 텍스트를 입력하거나 최근에 검색한 항목 드롭 다운 목록에서 선택하십시오.
3. 대체 텍스트(있는 경우)를 입력하십시오.
4. 다음 찾기를 클릭하여 검색을 시작하십시오.
5. 바꾸기를 클릭하여 현재 선택을 바꾸거나 모두 바꾸기를 클릭하여 선택된 또는 모든 인스턴스를 업데이트하십시오.
6. 각 작업 후에 대화 상자가 닫힙니다. 텍스트 영역에서 `F3`을 눌러 마지막 찾기 작업을 반복하거나 `Ctrl+F`를 눌러 대화 상자에 다시 액세스하십시오.

검색 옵션

대소문자 구분. 찾기 작업이 대소문자를 구분하는지 여부를 지정합니다(예: *myvar*이 *myVar*과 매치하는지 여부). 대체 텍스트는 이 설정에 관계없이 항상 입력된 대로 정확하게 삽입됩니다.

전체 단어만. 찾기 작업이 단어 안에 임베드된 텍스트와 매치하는지 여부를 지정합니다. 선택된 경우, 예를 들어 *spider*에 대한 검색은 *spiderman* 또는 *spider-man*과 매치하지 않습니다.

정규식. 정규식 구문 사용 여부를 지정합니다(다음 절 참조). 선택된 경우에는 **전체 단어만** 옵션을 사용할 수 없으며 해당 값을 무시합니다.

선택된 텍스트만. 모두 바꾸기 옵션 사용 시 검색 범위를 제어합니다.

정규식 구문

정규식을 사용하여 탭이나 줄 바꾸기 문자와 같은 특수 문자, *a - d*와 같은 문자 클래스나 범위, 숫자나 비숫자, 행 시작이나 끝과 같은 경계에 대한 검색을 수행할 수 있습니다. 지원되는 표현식 유형은 다음과 같습니다.

표 1. 문자 매치.

문자	매치
x	x 문자
\\	백슬래시 문자
\\0n	8진 값 0n이 포함된 문자(0 <= n <= 7)
\\0nn	8진 값 0nn이 포함된 문자(0 <= n <= 7)
\\0mnn	8진 값 0mnn이 포함된 문자(0 <= m <= 3, 0 <= n <= 7)
\\xhh	16진 값 0xhh가 포함된 문자
\\uhhhh	16진 값 0xhhhh가 포함된 문자
\\t	탭 문자('\u0009')
\\n	줄 바꾸기(줄 바꾸기) 문자('\u000A')
\\r	캐리지 리턴 문자('\u000D')
\\f	용지 넘김 문자('\u000C')
\\a	경고(벨) 문자('\u0007')
\\e	이스케이프 문자('\u001B')
\\cx	x에 해당되는 제어 문자

표 2. 문자 클래스 매치.

문자 클래스	매치
[abc]	a, b 또는 c(단순 클래스)
[^abc]	a, b 또는 c를 제외한 모든 문자(뺀셈)
[a-zA-Z]	a - z 또는 A - Z(경계값 포함)
[a-d[m-p]]	a - d 또는 m - p(합집합). 또는 이를 [a-dm-p]로 지정할 수 있음
[a-z&&[def]]	a - z, d, e 또는 f(교집합)
[a-z&&[^bc]]	b와 c를 제외하고 a - z(뺀셈). 또는 이를 [ad-z]로 지정할 수 있음

표 2. 문자 클래스 매치 (계속).

문자 클래스	매치
[a-z&&[^m-p]]	m - p를 제외하고 a - z(뺄셈). 또는 이를 [a-lq-z]로 지정할 수 있음

표 3. 사전 정의된 문자 클래스.

사전 정의된 문자 클래스	매치
.	모든 문자(행 종결자와 매치하거나 매치하지 않을 수 있음)
\d	모든 숫자: [0-9]
\D	비숫자: [^0-9]
\s	공백 문자: [\t\n\x0B\f\r]
\S	비공백 문자: [^\s]
\w	단어 문자: [a-zA-Z_0-9]
\W	비단어 문자: [^\w]

표 4. 경계 매치.

경계 매치기	매치
^	행의 시작
\$	행의 끝
\b	단어 경계
\B	비단어 경계
\A	입력의 시작
\Z	입력의 끝이지만 최종 종결자로도 사용(해당하는 경우)
\z	입력의 끝

제 2 장 스크립팅 언어

스크립팅 언어 개요

IBM SPSS Modeler의 스크립팅 기능으로 SPSS Modeler 사용자 인터페이스에서 동작하고 출력 오브젝트를 조작하고 명령 구문을 실행하는 스크립트를 작성할 수 있습니다. SPSS Modeler 내에서 직접 스크립트를 실행할 수 있습니다.

IBM SPSS Modeler의 스크립트는 스크립팅 언어 Python으로 작성됩니다. IBM SPSS Modeler에서 사용하는 Python의 Java 기반 구현을 Jython이라고 합니다. 스크립팅 언어는 다음 기능으로 구성됩니다.

- 노드, 스트림, 프로젝트, 출력, 기타 IBM SPSS Modeler 오브젝트를 참조하는 형식.
- 이들 오브젝트를 조작하는 데 사용할 수 있는 스크립팅 명령문 또는 명령 세트.
- 변수의 값, 모수 및 기타 오브젝트 설정을 위한 스크립팅 표현식 언어.
- 주석, 연속, 리터럴 텍스트의 블록에 대한 지원.

다음 절은 Python 스크립팅 언어, Python의 Jython 구현, IBM SPSS Modeler 내에서 스크립팅을 시작하기 위한 기본 구문을 설명합니다. 특정 특성 및 명령에 관한 정보는 그 뒤에 오는 절에서 제공됩니다.

Python과 Jython

Jython은 Python 스크립팅 언어의 한 구현으로, Java 언어로 작성되고 Java 플랫폼에 통합됩니다. Python은 강력한 오브젝트 지향 스크립팅 언어입니다. Jython은 완성된 스크립팅 언어의 생산성 기능을 제공하고 Python과는 달리 Java 가상 머신(JVM)을 지원하는 모든 환경에서 실행하기 때문에 유용합니다. 이것은 JVM의 Java 라이브러리가 프로그램을 작성 중일 때 사용할 수 있음을 의미합니다. Jython을 사용하면 이 차이를 활용하고 Python 언어의 구문과 대부분의 기능을 사용할 수 있습니다.

스크립팅 언어로서, Python(및 그의 Jython 구현)은 배우기 쉽고 코딩하기에 효율적이며, 실행 프로그램을 작성하기 위한 최소 필수 구조를 갖고 있습니다. 코드는 대화식으로, 즉 한 번에 한 행씩 입력할 수 있습니다. Python은 해석되는 스크립팅 언어입니다. 즉, Java에서와 같이 사전 컴파일 단계가 없습니다. Python 프로그램은 입력되는 그대로(구문 오류에 대해 구문 분석된 후) 해석되는 단순 텍스트 파일입니다. 정의된 값 같은 단순 표현식뿐 아니라 함수 정의 같은 더 복잡한 조치가 즉시 실행되고 사용 가능합니다. 코드에 대해 작성되는 모든 변경은 빨리 검정할 수 있습니다. 그러나 스크립트 해석은 몇 가지 단점을 갖고 있습니다. 예를 들어, 정의되지 않은 변수의 사용이 컴파일러 오류가 아니므로, 해당 변수를 사용하는 명령문이 실행되는 경우(및 그 때)에만 발견됩니다. 이 경우 프로그램을 편집하고 실행하여 오류를 디버깅할 수 있습니다.

Python은 모든 데이터 및 코드를 포함한 모든 것을 오브젝트로 봅니다. 그러므로 코드 행으로 이들 오브젝트를 조작할 수 있습니다. 숫자 및 문자열 같은 일부 선택 유형은 오브젝트가 아니라 값으로 더 편리하게 간주됩니다. 이것이 Python에서 지원됩니다. 지원되는 하나의 널값이 있습니다. 이 널값은 None의 예약 이름을 갖습니다.

Python 및 Jython 스크립팅에 대한 더 깊이있는 소개 및 몇 가지 스크립트 예에 대해서는 <http://www.ibm.com/developerworks/java/tutorials/j-jython1/j-jython1.html> 및 <http://www.ibm.com/developerworks/java/tutorials/j-jython2/j-jython2.html>의 내용을 참조하십시오.

Python 스크립팅

Python 스크립팅 언어에 대한 이 안내서는 개념 및 프로그래밍 기초를 포함하여 IBM SPSS Modeler에서 스크립팅할 때 사용할 수 있는 구성요소에 대한 소개입니다. 이것은 IBM SPSS Modeler 내에서 사용할 사용자 자신의 Python 스크립팅 개발을 시작하기에 충분한 지식을 제공합니다.

작업

지정은 등호(=)를 사용하여 수행됩니다. 예를 들어 "x"라는 변수에 값 "3"을 지정하려면 다음 명령문을 사용합니다.

```
x = 3
```

등호는 또한 변수에 문자열 유형 데이터를 지정하는 데도 사용합니다. 예를 들어, 변수 "y"에 값 "a string value"를 지정하려면 다음 명령문을 사용합니다.

```
y = "a string value"
```

다음 테이블은 몇 가지 공통적으로 사용하는 비교 및 숫자 연산 및 해당 설명을 나열합니다.

표 5. 공통 비교 및 숫자 연산

연산	설명
$x < y$	x가 y보다 작습니까?
$x > y$	x가 y보다 큼습니까?
$x \leq y$	x가 y보다 작거나 같습니까?
$x \geq y$	x가 y보다 크거나 같습니까?
$x == y$	x와 y와 같습니까?
$x != y$	x가 y와 같지 않습니까?
$x \lt;> y$	x가 y와 같지 않습니까?
$x + y$	y를 x에 더합니다.
$x - y$	x에서 y를 뺍니다.
$x * y$	x에 y를 곱합니다.
x / y	x를 y로 나눕니다.
$x ** y$	x를 y 거듭제곱합니다.

목록

목록은 요소의 시퀀스입니다. 목록은 임의의 수의 요소를 포함할 수 있으며, 목록의 요소는 임의의 오브젝트 유형일 수 있습니다. 목록은 배열로 생각할 수도 있습니다. 목록에 있는 요소의 수는 요소가 추가, 제거 또는 바뀔 때 늘거나 줄어든 수 있습니다.

예

<code>[]</code>	임의의 빈 목록.
<code>[1]</code>	정수인 단일 요소를 갖는 목록.
<code>["Mike", 10, "Don", 20]</code>	2개의 문자열 요소와 2개의 정수 요소인 4개 요소를 갖는 목록.
<code>[[], [7], [8, 9]]</code>	목록의 목록입니다. 각 하위 목록은 빈 목록 또는 정수 요소의 목록입니다.
<code>x = 7; y = 2; z = 3;</code> <code>[1, x, y, x + y]</code>	정수의 목록입니다. 이 예는 변수 및 표현식의 사용을 보여줍니다.

목록을 변수에 지정할 수 있습니다. 예를 들어,

```
mylist1 = ["one", "two", "three"]
```

그런 다음 목록의 특정 요소에 액세스할 수 있습니다.

```
mylist[0]
```

출력은 다음과 같습니다.

```
one
```

대괄호([]) 안에 있는 숫자를 지수라고 하며 목록의 특정 요소를 의미합니다. 목록의 요소는 0에서 시작하여 지수가 지정됩니다.

목록의 요소 범위를 선택할 수도 있는데, 이를 조각이라고 합니다. 예를 들어, `x[1:3]`은 `x`의 두 번째 및 세 번째 요소를 선택합니다. 끝 지수는 선택을 하나 지난 것입니다.

문자열

문자열은 값으로 처리되는 문자의 불변 시퀀스입니다. 문자열은 새 문자열이 되는 모든 불변 시퀀스 함수 및 연산자를 지원합니다. 예를 들어, `"abcdef"[1:4]`는 출력 `"bcd"`가 됩니다.

Python에서 문자는 길이 1의 문자열로 표시됩니다.

문자열 리터럴은 단일 또는 3중 인용의 사용에 의해 정의됩니다. 작은따옴표를 사용하여 정의되는 문자열은 여러 행에 표시할 수 없는 반면, 3중 인용부호를 사용하여 정의되는 문자열은 여러 행에 표시할 수 있습니다. 문자열은 작은따옴표(') 또는 큰따옴표(")로 묶을 수 있습니다. 인용 문자는 이스케이프되지 않은 다른 인용 문자 또는 백슬래시(\) 문자가 선행되는 이스케이프된 인용 문자를 포함할 수 있습니다.

예

```
"This is a string"
'This is also a string'
"It's a string"
'This book is called "Python Scripting and Automation Guide".'
"This is an escape quote (\") in a quoted string"
```

공백으로 구분되는 다중 문자열은 Python 구문 분석기에 의해 자동으로 연결됩니다. 이것은 긴 문자열을 입력하고 단일 문자열에서 인용 유형을 혼합하기가 쉽게 합니다. 예를 들어,

```
"This string uses ' and " 'that string uses ".'
```

이 결과는 다음 출력이 됩니다.

```
This string uses ' and that string uses ".
```

문자열은 여러 가지 유용한 메소드를 지원합니다. 이들 메소드의 일부가 다음 표에서 제공됩니다.

표 6. 문자열 메소드

메소드	사용법
s.capitalize()	s를 처음 대문자로 만듭니다.
s.count(ss {,start {,end}})	s[start:end]에서 ss의 발생을 계수합니다.
s.startswith(str {, start {, end}}) s.endswith(str {, start {, end}})	s가 str로 시작하는지 여부를 검정합니다. s가 str로 끝나는지 여부를 검정합니다.
s.expandtabs({size})	탭을 공백으로 바꿉니다. 기본 size는 8입니다.
s.find(str {, start {, end}}) s.rfind(str {, start {, end}})	s에서 str의 첫 번째 지수를 찾습니다. 없는 경우 결과는 -1입니다. rfind는 오른쪽에서 왼쪽으로 검색합니다.
s.index(str {, start {, end}}) s.rindex(str {, start {, end}})	s에서 str의 첫 번째 지수를 찾습니다. 없는 경우 ValueError가 발생합니다. rindex는 오른쪽에서 왼쪽으로 검색합니다.
s.isalnum	문자열이 영숫자인지 여부를 검정합니다.
s.isalpha	문자열이 알파벳인지 여부를 검정합니다.
s.isnum	문자열이 숫자인지 여부를 검정합니다.
s.isupper	문자열이 모두 대문자인지 여부를 검정합니다.
s.islower	문자열이 모두 소문자인지 여부를 검정합니다.
s.isspace	문자열이 모두 공백인지 여부를 검정합니다.
s.istitle	문자열이 초기 캡 영숫자 문자열 시퀀스인지 여부를 검정합니다.
s.lower() s.upper() s.swapcase() s.title()	모두 소문자로 변환 모두 대문자로 변환 모두 반대 케이스로 변환 모두 제목 케이스로 변환
s.join(seq)	s를 구분 문자로 사용하여 seq의 문자열을 결합합니다.
s.splitlines({keep})	s를 행으로 분할합니다. keep가 true인 경우 새 행을 유지합니다.
s.split({sep {, max}})	s를 최대 max번 sep를 사용하여 "단어"로 분할합니다(기본 sep는 공백임).

표 6. 문자열 메소드 (계속)

메소드	사용법
s.ljust(width) s.rjust(width) s.center(width) s.zfill(width)	width 필드의 문자열을 가로로 왼쪽 맞춤 width 필드의 문자열을 가로로 오른쪽 맞춤 width 필드의 문자열을 가로로 가운데 맞춤 0으로 채웁니다.
s.lstrip() s.rstrip() s.strip()	선행 공백 제거 후미 공백 제거 선행 및 후미 공백 제거
s.translate(str {,delc})	delc에서 모든 문자를 제거한 후, 테이블을 사용하여 s를 변환합니다. str은 길이 == 256인 문자열이어야 합니다.
s.replace(old, new {, max})	문자열 old의 모든 또는 max 발생을 문자열 new로 바꿉니다.

주석

주석은 파운드(또는 해시) 부호(#)에 의해 도입되는 설명입니다. 동일한 행에서 파운드 부호 뒤에 오는 모든 텍스트는 주석의 일부로 간주되고 무시됩니다. 주석은 임의의 열에서 시작할 수 있습니다. 다음 예는 주석의 사용을 보여줍니다.

```
#The HelloWorld application is one of the most simple
print 'Hello World' # print the Hello World line
```

명령문 구문

Python의 명령문 구문은 매우 단순합니다. 일반적으로 각 소스 행이 단일 명령문입니다. expression 및 assignment문을 제외하면 각 명령문은 if 또는 for 같은 키워드 이름에 의해 도입됩니다. 공백 행이나 주석 행을 코드에 있는 임의의 명령문 사이에 삽입할 수 있습니다. 한 행에 둘 이상의 명령문이 있는 경우, 각 명령문은 세미콜론(;)으로 구분되어야 합니다.

매우 긴 명령문은 둘 이상의 행에서 계속될 수 있습니다. 이 경우 다음 행에서 계속할 명령문은 백슬래시(\)로 끝나야 합니다. 예를 들면,

```
x = "A loooooooooooooooooooooooooong string" + \
    "another loooooooooooooooooooooooooong string"
```

구조가 소괄호(()), 대괄호([]) 또는 중괄호({})로 묶이면, 명령문은 백슬래시를 삽입할 필요 없이 임의의 심표 뒤에 있는 새 행에서 계속될 수 있습니다. 예를 들면,

```
x = (1, 2, 3, "hello",
    "goodbye", 4, 5, 6)
```

식별자

식별자는 변수, 함수, 클래스 및 키워드에 이름을 지정하는 데 사용됩니다. 식별자는 임의의 길이일 수 있지만, 대문자나 소문자의 영문자나 밑줄 문자(_)로 시작해야 합니다. 밑줄로 시작하는 이름은 일반적으로 내부 또는 개인용 이름을 위해 예약됩니다. 첫 번째 문자 후에, 식별자는 임의의 숫자 및 영문자, 0 - 9 범위의 숫자, 밑줄 문자의 조합을 포함할 수 있습니다.

Jython에는 변수, 함수 또는 클래스의 이름으로 사용할 수 없는 몇 가지 예약어가 있습니다. 이들은 다음 범주에 해당합니다.

- **명령문 도입자:** assert, break, class, continue, def, del, elif, else, except, exec, finally, for, from, global, if, import, pass, print, raise, return, try while
- **모수 도입자:** as, import 및 in
- **연산자:** and, in, is, lambda, not 및 or

부적절한 키워드 사용은 일반적으로 SyntaxError가 됩니다.

코드 블록

코드 블록은 단일 명령문이 예상되는 위치에서 사용하는 명령문 그룹입니다. 코드 블록은 if, elif, else, for, while, try, except, def, class문 중 하나 뒤에 올 수 있습니다. 이들 명령문은 콜론 문자(:)를 사용하여 코드 블록을 도입합니다. 예를 들어,

```
if x == 1:
    y = 2
    z = 3
elif:
    y = 4
    z = 5
```

들여쓰기는(Java에서 사용하는 중괄호 대신) 코드 블록을 구분하기 위해 사용합니다. 블록의 모든 행은 동일한 위치로 들여써야 합니다. 이것은 들여쓰기의 변화가 코드 블록의 종료를 표시하기 때문입니다. 대개 수준당 4개의 공백만큼 들여씁니다. 행을 들여쓰기 위해 탭보다는 공백을 사용할 것을 권장합니다. 공백과 탭을 혼합해서는 안 됩니다. 모듈의 가장 바깥쪽 블록의 행은 1열에서 시작해야 하며, 그렇지 않으면 SyntaxError가 발생합니다.

코드 블록을 구성하는(및 콜론 뒤에 오는) 명령문은 세미콜론으로 구분되어 단일 행에 있을 수 있습니다. 예를 들어,

```
if x == 1: y = 2; z = 3;
```

스크립트로 인수 전달

스크립트로 인수 전달은 스크립트가 수정 없이 반복적으로 사용할 수 있음을 의미하므로 유용합니다. 명령행에서 전달되는 인수는 sys.argv 목록의 값으로 전달됩니다. 전달되는 값의 수는 len(sys.argv) 명령을 사용하여 얻을 수 있습니다. 예를 들어,

```
import sys
print "test1"
print sys.argv[0]
print sys.argv[1]
print len(sys.argv)
```

이 예에서 import 명령은 전체 sys 클래스를 가져오므로 이 클래스에 대해 존재하는 메소드(예: argv)를 사용할 수 있습니다.

이 예의 스크립트는 다음 행을 사용하여 호출할 수 있습니다.

```
/u/mjloos/test1 mike don
```

결과는 다음 출력과 유사합니다.

```
/u/mjloos/test1 mike don
test1
mike
don
3
```

예

print 키워드는 그 뒤에 바로 따라오는 인수를 인쇄합니다. 명령문이 쉼표 뒤에 오는 경우, 줄 바꾸기가 출력에 포함되지 않습니다. 예를 들어,

```
print "This demonstrates the use of a",
print " comma at the end of a print statement."
```

출력은 다음과 같습니다.

```
This demonstrates the use of a comma at the end of a print statement.
```

for 명령문은 코드 블록을 반복하는 데 사용합니다. 예를 들어,

```
mylist1 = ["one", "two", "three"]
for lv in mylist1:
    print lv
    continue
```

이 예에서는 세 개의 문자열이 mylist1 목록에 지정됩니다. 목록의 요소가 각 행에 하나의 요소를 갖고 인쇄됩니다. 출력은 다음과 같습니다.

```
one
two
three
```

이 예에서, for 루프가 각 요소에 대한 코드 블록을 구현하므로 반복자 lv는 mylist1 목록에 있는 각 요소의 값을 차례로 사용합니다. 반복자는 임의 길이의 모든 유효한 식별자일 수 있습니다.

if문은 조건문입니다. 조건을 평가하고 평가 결과에 따라서 true 또는 false를 리턴합니다. 예를 들어,

```
mylist1 = ["one", "two", "three"]
for lv in mylist1:
    if lv == "two"
        print "The value of lv is ", lv
    else
        print "The value of lv is not two, but ", lv
    continue
```

이 예에서, 반복자 lv의 값이 평가됩니다. lv의 값이 two인 경우, lv의 값이 two가 아닌 경우에 리턴되는 문자열로 다른 문자열이 리턴됩니다. 이 결과는 다음 출력이 됩니다.

```
The value of lv is not two, but one
The value of lv is two
The value of lv is not two, but three
```

수학적 메소드

math 모듈에서 수학적 메소드에 액세스할 수 있습니다. 이들 메소드의 일부가 다음 표에서 제공됩니다. 별도로 지정되지 않는 한, 모든 값은 Float로 리턴됩니다.

표 7. 수학적 메소드

메소드	사용법
math.ceil(x)	x의 상한을 Float로서 리턴하는데, x보다 크거나 같은 가장 작은 정수입니다.
math.copysign(x, y)	y의 부호를 갖는 x를 리턴합니다. copysign(1, -0.0)은 -1을 리턴합니다.
math.fabs(x)	x의 절대값을 리턴합니다.
math.factorial(x)	x 계승을 리턴합니다. x가 음수이거나 정수가 아닌 경우 ValueError가 발생합니다.
math.floor(x)	x의 바닥을 Float로 리턴하는데, 이것은 x보다 작거나 같은 가장 큰 정수입니다.
math.frexp(x)	x의 가수(m) 및 지수(e)를 (m, e) 쌍으로 리턴합니다. m은 Float이고 e는 정수이므로, 정확하게 $x == m * 2^{**e}$ 입니다. x가 0인 경우 (0.0, 0)을 리턴하고, 그렇지 않으면 $0.5 <= \text{abs}(m) < 1$ 을 리턴합니다.
math.fsum(iterable)	iterable에 있는 값의 정확한 부동 소수점 합계를 리턴합니다.
math.isinf(x)	Float x가 양 또는 음의 무한대인지 검사합니다.
math.isnan(x)	Float x가 NaN(숫자가 아님)인지 검사합니다.
math.ldexp(x, i)	$x * (2^{**i})$ 를 리턴합니다. 이것은 본질적으로 frexp 함수의 역입니다.
math.modf(x)	x의 소수 및 정수 부분을 리턴합니다. 두 결과 모두가 x의 부호를 갖고 Float입니다.
math.trunc(x)	Integral로 잘린 Real 값 x를 리턴합니다.
math.exp(x)	e^{**x} 를 리턴합니다.
math.log(x[, base])	base의 주어진 값에 대한 x의 대수를 리턴합니다. base가 지정되지 않으면 x의 자연로그가 리턴됩니다.
math.log1p(x)	$1+x$ (base e)의 자연로그를 리턴합니다.
math.log10(x)	x의 밑이 10인 대수를 리턴합니다.
math.pow(x, y)	y로 거듭제곱한 x를 리턴합니다. pow(1.0, x) 및 pow(x, 0.0)은 x가 0이나 NaN일 때도 항상 1을 리턴합니다.
math.sqrt(x)	x의 제곱근을 리턴합니다.

수학적 함수 외에, 몇 가지 유용한 삼각법 메소드가 있습니다. 이들 메소드가 다음 표에 표시됩니다.

표 8. 삼각법 메소드

메소드	사용법
math.acos(x)	x의 아크 코사인을 라디안 단위로 리턴합니다.
math.asin(x)	x의 아크 사인을 라디안 단위로 리턴합니다.
math.atan(x)	x의 아크 탄젠트를 라디안 단위로 리턴합니다.
math.atan2(y, x)	atan(y / x)를 라디안 단위로 리턴합니다.

표 8. 삼각법 메소드 (계속)

메소드	사용법
math.cos(x)	x의 코사인을 라디안 단위로 리턴합니다.
math.hypot(x, y)	유클리디안 표준 sqrt(x*x + y*y)를 리턴합니다. 이것은 원점부터 (x, y) 점까지의 벡터 길이입니다.
math.sin(x)	x의 사인을 라디안 단위로 리턴합니다.
math.tan(x)	x의 탄젠트를 라디안 단위로 리턴합니다.
math.degrees(x)	각 x를 라디안에서 도로 변환합니다.
math.radians(x)	각 x를 도에서 라디안으로 변환합니다.
math.acosh(x)	x의 역쌍곡 코사인을 리턴합니다.
math.asinh(x)	x의 역쌍곡 사인을 리턴합니다.
math.atanh(x)	x의 역쌍곡 탄젠트를 리턴합니다.
math.cosh(x)	x의 쌍곡 코사인을 리턴합니다.
math.sinh(x)	x의 쌍곡 사인을 리턴합니다.
math.tanh(x)	x의 쌍곡 탄젠트를 리턴합니다.

두 개의 수학 상수도 있습니다. math.pi의 값은 수학 상수 pi(원주율)입니다. math.e의 값은 수학 상수 e입니다.

ASCII가 아닌 문자 사용

ASCII가 아닌 문자를 사용하기 위해서 Python은 유니코드로의 문자열의 명시적 인코딩 및 디코딩이 필요합니다. IBM SPSS Modeler에서 Python 스크립트는 UTF-8으로 인코딩되는 것으로 가정되는데, 이것은 ASCII가 아닌 문자를 지원하는 표준 유니코드 인코딩입니다. 다음 스크립트는 Python 컴파일러가 SPSS Modeler에 의해 UTF-8으로 설정되었기 때문에 컴파일됩니다.

```
stream = modeler.script.stream()
filenode = stream.createAt("variablefile", "テストノード", 96, 64)
```

그러나, 결과 노드는 올바르지 않은 레이블을 갖습니다.



ãfã, 'ãf`ãf ãf'ãfãf

그림 3. ASCII가 아닌 문자를 포함하는 노드 레이블이 올바르지 않게 표시됨

문자열 리터럴 자체가 Python에 의해 ASCII 문자열로 변환되었기 때문에 레이블이 올바르지 않습니다.

Python은 문자열 리터럴 전에 u 문자 접두문자를 추가하여 유니코드 문자열 리터럴이 지정되도록 허용합니다.

```
stream = modeler.script.stream()
filenode = stream.createAt("variablefile", u"テストノード", 96, 64)
```

이것은 유니코드 문자열을 작성하며 레이블이 올바르게 나타납니다.



テストノード

그림 4. ASCII가 아닌 문자를 포함하는 노드 레이블이 올바르게 표시됨

Python 및 유니코드 사용은 이 문서의 범위를 벗어나는 주제입니다. 이 주제를 상세하게 다루는 많은 서적 및 온라인 자원이 사용 가능합니다.

오브젝트 지향 프로그래밍

오브젝트 지향 프로그래밍은 사용자 프로그램에서 대상 문제점의 모델 작성 개념을 기반으로 합니다. 오브젝트 지향 프로그래밍은 프로그래밍 오류를 줄였으며 코드의 재사용을 권장합니다. Python은 오브젝트 지향 언어입니다. Python에서 정의되는 오브젝트는 다음 기능을 갖습니다.

- **항등.** 각 오브젝트는 고유해야 하며 이것은 검정 가능해야 합니다. 이를 위한 is 및 is not 검정이 존재합니다.
- **상태.** 각 오브젝트는 상태를 저장할 수 있어야 합니다. 필드 및 인스턴스 변수 같은 속성이 이 목적을 위해 존재합니다.
- **작동.** 각 오브젝트는 상태를 조작할 수 있어야 합니다. 이를 위한 메소드가 존재합니다.

Python에는 오브젝트 지향 프로그래밍 지원을 위한 다음 기능이 포함되어 있습니다.

- **클래스 기반 오브젝트 작성.** 클래스는 오브젝트 작성을 위한 템플리트입니다. 오브젝트는 연관된 작동을 갖는 데이터 구조입니다.
- **다형성을 갖는 상속.** Python은 단일 및 다중 상속을 지원합니다. 모든 Python 인스턴스 메소드는 다형성이며 서브클래스로 대체할 수 있습니다.
- **데이터 숨김을 포함한 캡슐화.** Python은 속성을 숨길 수 있습니다. 숨긴 상태인 경우, 속성은 클래스의 메소드를 통해서만 클래스 외부에서 액세스할 수 있습니다. 클래스는 데이터를 수정하기 위해 메소드를 구현합니다.

클래스 정의

Python 클래스를 사용하면 변수와 메소드를 둘 다 정의할 수 있습니다. Java에서와는 달리, Python에서는 소스 파일(또는 모듈)당 임의 숫자의 공용 클래스를 정의할 수 있습니다. 그러므로 Python의 모듈을 Java에서의 패키지과 비슷하게 생각할 수 있습니다.

Python에서는 클래스가 `class`문을 사용하여 정의됩니다. `class`문은 다음 양식을 갖습니다.

```
class name (superclasses): statement
```

또는

```
class name (superclasses):  
    assignment  
    .  
    .  
    function  
    .  
    .
```

클래스를 정의할 때, 0개 이상의 *assignment*문을 제공하는 옵션이 있습니다. 이들은 클래스의 모든 인스턴스가 공유하는 클래스 속성을 작성합니다. 0개 이상의 함수 정의를 제공할 수도 있습니다. 이들 함수 정의는 메소드를 작성합니다. 슈퍼클래스 목록은 선택사항입니다.

클래스 이름은 동일한 범위, 즉 모듈, 함수 또는 클래스 내에서 고유해야 합니다. 동일한 클래스를 참조하는 다중 변수를 정의할 수 있습니다.

클래스 인스턴스 작성

클래스는 클래스(또는 공유) 속성을 보유하거나 클래스 인스턴스를 작성하는 데 사용합니다. 클래스의 인스턴스를 작성하기 위해 클래스를 함수인 것처럼 호출합니다. 예를 들어, 다음 클래스를 고려하십시오.

```
class MyClass:  
    pass
```

여기에서 클래스를 완료하기 위해 명령문이 필요하지만 프로그램식으로 필요한 조치가 없기 때문에 `pass`문을 사용합니다.

다음 명령문은 `MyClass` 클래스의 인스턴스를 작성합니다.

```
x = MyClass()
```

클래스 인스턴스에 속성 추가

Java에서와는 달리, Python에서는 클라이언트가 클래스의 인스턴스에 속성을 추가할 수 있습니다. 단 하나의 인스턴스만 변경됩니다. 예를 들어 인스턴스 `x`에 속성을 추가하려면, 해당 인스턴스에서 새 값을 설정하십시오.

```
x.attr1 = 1
x.attr2 = 2
.
.
x.attrN = n
```

클래스 속성 및 메소드 정의

클래스에 바인드되는 모든 변수가 클래스 속성입니다. 클래스 내에서 정의되는 모든 함수가 메소드입니다. 메소드는 관습적으로 `self`라고 하는 클래스의 인스턴스를 첫 번째 인수로 수신합니다. 예를 들어 일부 클래스 속성 및 메소드를 정의하기 위해 다음 코드를 입력할 수 있습니다.

```
class MyClass
    attr1 = 10          #class attributes
    attr2 = "hello"

    def method1(self):
        print MyClass.attr1  #reference the class attribute

    def method2(self):
        print MyClass.attr2  #reference the class attribute

    def method3(self, text):
        self.text = text      #instance attribute
        print text, self.text #print my argument and my attribute

    method4 = method3        #make an alias for method3
```

클래스 안에서, 클래스 이름과 함께 클래스 속성에 대한 모든 참조를 규정해야 합니다(예: `MyClass.attr1`). 인스턴스 속성에 대한 모든 참조는 `self` 변수로 규정되어야 합니다(예: `self.text`). 클래스 외부에서는 클래스 이름(예: `MyClass.attr1`) 또는 클래스의 인스턴스(예: `x.attr1`, 여기서 `x`는 클래스의 인스턴스임)로 클래스 속성에 대한 모든 참조를 규정해야 합니다. 클래스 외부에서, 인스턴스 변수에 대한 모든 참조는 클래스의 인스턴스로 규정되어야 합니다(예: `x.text`).

숨겨진 변수

데이터는 *Private* 변수를 작성하여 숨길 수 있습니다. 개인용 변수는 클래스 자체에 의해서만 액세스할 수 있습니다. `__xxx` 또는 `__xxx_yyy`(두 개의 선형 밑줄이 있음) 양식의 이름을 선언하는 경우, Python 구문 분석기는 자동으로 클래스 이름을 선언된 이름에 추가하여 숨겨진 변수를 작성합니다. 예를 들면,

```
class MyClass:
    __attr = 10      #private class attribute

    def method1(self):
        pass

    def method2(self, p1, p2):
        pass

    def __privateMethod(self, text):
        self.__text = text    #private attribute
```

Java에서와는 달리, Python에서는 인스턴스 변수에 대한 모든 참조가 `self`로 규정되어야 합니다. 즉, `this`의 내재된 사용은 없습니다.

상속

클래스로부터 상속하는 기능은 오브젝트 지향 프로그래밍에서 기본입니다. Python은 단일 및 다중 상속을 둘 다 지원합니다. 단일 상속은 단 하나의 슈퍼클래스가 있을 수 있음을 의미합니다. 다중 상속은 둘 이상의 슈퍼클래스가 있을 수 있음을 의미합니다.

상속은 다른 클래스를 서브클래싱하여 구현됩니다. 임의 숫자의 Python 클래스가 슈퍼클래스일 수 있습니다. Python의 Jython 구현에서는 단 하나의 Java 클래스가 직접 또는 간접적으로 상속될 수 있습니다. 슈퍼클래스가 제공될 필요는 없습니다.

슈퍼클래스의 모든 속성이나 메소드도 임의의 서브클래스에 있으며 클래스 자체 또는 속성이나 메소드가 숨겨지지 않는 한 임의의 클라이언트가 사용할 수 있습니다. 서브클래스의 모든 상속은 어디에서나 사용할 수 있으며 슈퍼클래스의 인스턴스를 사용할 수 있습니다. 이것이 다형성의 예입니다. 이들 기능은 확장의 재사용 및 용이성을 가능하게 합니다.

예제

```
class Class1: pass    #no inheritance

class Class2: pass

class Class3(Class1): pass    #single inheritance

class Class4(Class3, Class2): pass    #multiple inheritance
```

제 3 장 IBM SPSS Modeler에서의 스크립팅

스크립트의 유형

IBM SPSS Modeler에는 다음 세 가지 유형의 스크립트가 있습니다.

- 스트림 스크립트는 단일 스트림의 실행을 제어하는 데 사용하며 스트림 내에 저장됩니다.
- 슈퍼노드 스크립트는 슈퍼노드의 작동을 제어하는 데 사용됩니다.
- 독립형 또는 세션 스크립트는 많은 상이한 스트림 사이에서 실행을 조정하는 데 사용할 수 있습니다.

광범위한 SPSS Modeler 기능에 액세스할 수 있는 IBM SPSS Modeler의 스크립트에서 다양한 메소드를 사용할 수 있습니다. 이들 메소드는 고급 함수를 작성하기 위해 41 페이지의 제 4 장 『스크립팅 API』에서 사용할 수도 있습니다.

스트림, 슈퍼노드 스트림 및 다이어그램

대부분의 경우에 스트림이란 용어는 파일에서 로드되는 스트림이든 슈퍼노드 내에서 사용하는 스트림이든 간에 동일한 것을 의미합니다. 일반적으로 함께 연결되고 실행될 수 있는 노드의 컬렉션을 의미합니다. 그러나 스크립팅에서는 모든 작업이 모든 장소에서 지원되지는 않으며, 이는 스크립트 작성자가 사용할 스크립트 변형을 알고 있어야 함을 의미합니다.

스트림

스트림은 기본 IBM SPSS Modeler 문서 유형입니다. 스트림은 저장, 로드, 편집 및 실행할 수 있습니다. 스트림은 또한 모수, 글로벌 값, 스크립트 및 연관된 기타 정보를 가질 수 있습니다.

슈퍼노드 스트림

슈퍼노드 스트림은 슈퍼노드 내에서 사용하는 스트림의 유형입니다. 일반 스트림처럼, 함께 링크되는 노드를 포함하고 있습니다. 슈퍼노드 스트림은 일반 스트림과는 많은 차이가 있습니다.

- 모수 및 임의의 스크립트가 슈퍼노드 스트림 자체가 아니라 슈퍼노드 스트림을 소유하는 슈퍼노드와 연관됩니다.
- 슈퍼노드 스트림은 슈퍼노드의 유형에 따라서 추가 입력 및 출력 연결자 노드를 갖습니다. 이들 연결자 노드는 슈퍼노드 스트림 안쪽으로 정보를 이동시키는 데 사용하며 슈퍼노드가 작성될 때 자동으로 작성됩니다.

다이어그램

다이어그램이란 용어는 노드 추가 및 제거, 노드 사이의 연결 수정 같이 일반 스트림 및 슈퍼노드 스트림 모두에서 지원되는 함수를 포함합니다.

스트림 실행

다음 예는 스트림의 모든 실행 가능 노드를 실행하며 스트림 스크립트의 가장 간단한 유형입니다.

```
modeler.script.stream().runAll(None)
```

다음 예도 스트림의 모든 실행 가능 노드를 실행합니다.

```
stream = modeler.script.stream()
stream.runAll(None)
```

이 예에서, 스트림은 `stream`이라는 변수에 저장됩니다. 스크립트는 일반적으로 스트림 또는 스트림 내의 노드를 수정하는 데 사용하기 때문에 스트림을 변수에 저장하는 것은 유용합니다. 스트림을 저장하는 변수를 작성하면 간결한 스크립트가 됩니다.

스크립팅 컨텍스트

`modeler.script` 모듈은 스크립트가 실행되는 컨텍스트를 제공합니다. 모듈은 런타임 시에 자동으로 SPSS Modeler 스크립트로 가져옵니다. 모듈은 실행 환경에 대한 액세스를 갖는 스크립트를 제공하는 4개의 함수를 정의합니다.

- `session()` 함수가 스크립트에 대한 세션을 리턴합니다. 세션은 임의의 스트림을 실행하는 데 사용 중인 로케일 및 SPSS Modeler 백엔드(로컬 프로세스 또는 네트워크된 SPSS Modeler Server) 같은 정보를 정의합니다.
- `stream()` 함수는 스트림 및 슈퍼노드 스크립트와 함께 사용할 수 있습니다. 이 함수는 실행 중인 스트림 스크립트 또는 슈퍼노드 스크립트를 소유하는 스트림을 리턴합니다.
- `diagram()` 함수는 슈퍼노드 스크립트와 함께 사용할 수 있습니다. 이 함수는 슈퍼노드 내의 다이어그램을 리턴합니다. 기타 스크립트 유형의 경우 이 함수는 `stream()` 함수와 동일한 것을 리턴합니다.
- `supernode()` 함수는 슈퍼노드 스크립트와 함께 사용할 수 있습니다. 이 함수는 실행 중인 스크립트를 소유하는 슈퍼노드를 리턴합니다.

4개의 함수와 해당 출력이 다음 표에 요약되어 있습니다.

표 9. `modeler.script` 함수의 요약

스크립트 유형	<code>session()</code>	<code>stream()</code>	<code>diagram()</code>	<code>supernode()</code>
독립형	세션 리턴	스크립트가 호출될 때 현재 관리 스트림(예: 일괄처리 모드 <code>-stream</code> 옵션을 통해 전달되는 스트림) 또는 <code>None</code> 을 리턴합니다.	<code>stream()</code> 의 경우와 동일	적용되지 않음
스트림	세션 리턴	스트림 리턴	<code>stream()</code> 의 경우와 동일	적용되지 않음
슈퍼노드	세션 리턴	스트림 리턴	슈퍼노드 스트림 리턴	슈퍼노드 리턴

modeler.script 모듈은 또한 스크립트를 종료 코드와 함께 종료하는 방법도 정의합니다. `exit(exit-code)` 함수는 스크립트를 실행 중지하고 제공되는 정수 종료 코드를 리턴합니다.

스트림에 대해 정의되는 메소드 중 하나가 `runAll(List)`입니다. 이 메소드는 모든 실행 가능 노드를 실행합니다. 노드를 실행하여 생성되는 모든 모델이나 출력이 제공되는 목록에 추가됩니다.

스트림 실행의 경우 모델, 그래프 및 기타 출력 같은 출력을 생성하는 것이 공통적입니다. 이 출력을 캡처하기 위해 스크립트가 목록으로 초기화되는 변수를 제공할 수도 있습니다. 예를 들어,

```
stream = modeler.script.stream()
results = []
stream.runAll(results)
```

실행이 완료될 때, 실행에 의해 생성되는 모든 오브젝트를 `results` 목록에서 액세스할 수 있습니다.

기존 노드 참조

스트림은 종종 스트림이 실행되기 전에 수정되어야 하는 일부 모수와 함께 사전 작성됩니다. 이들 모수의 수정은 다음 작업을 포함합니다.

1. 관련 스트림에서 노드 찾기.
2. 노드 또는 스트림 설정(또는 둘 다) 변경.

노드 찾기

스트림은 기존 노드를 찾는 많은 방법을 제공합니다. 이들 메소드가 다음 표에 요약되어 있습니다.

표 10. 기존 노드 찾기를 위한 메소드

메소드	리턴 유형	설명
<code>s.findAll(type, label)</code>	요약도표	지정된 유형 및 레이블을 갖는 모든 노드의 목록을 리턴합니다. 유형이나 레이블이 None일 수 있으며, 이 경우 다른 모수를 사용합니다.
<code>s.findAll(filter, recursive)</code>	요약도표	지정된 필터에서 허용되는 모든 노드의 컬렉션을 리턴합니다. 재귀적 플래그가 True인 경우 지정된 스트림 내의 모든 슈퍼노드도 검색됩니다.
<code>s.findById(id)</code>	노드	제공되는 ID를 갖는 노드 또는 그런 노드가 없는 경우 None을 리턴합니다. 검색은 현재 스트림으로 제한됩니다.
<code>s.findByName(type, label)</code>	노드	제공되는 유형, 레이블 또는 둘 다 갖는 노드를 리턴합니다. 유형이나 이름이 None일 수 있으며, 이 경우 다른 모수를 사용합니다. 다중 노드가 매치하는 경우 임의의 하나가 선택되어 리턴됩니다. 매치하는 노드가 없는 경우 리턴값은 None입니다.

표 10. 기존 노드 찾기를 위한 메소드 (계속)

메소드	리턴 유형	설명
s.findDownstream(fromNodes)	요약도표	제공되는 노드 목록에서 검색하고 제공되는 노드의 노드 다운스트림 세트를 리턴합니다. 리턴되는 노드에는 원래 제공된 노드가 포함됩니다.
s.findUpstream(fromNodes)	요약도표	제공되는 노드 목록에서 검색하고 제공되는 노드의 노드 업스트림 세트를 리턴합니다. 리턴되는 노드에는 원래 제공된 노드가 포함됩니다.

예를 들어, 스트림이 스크립트가 액세스하기 위해 필요한 단일 필터 노드를 포함한 경우 필터 노드는 다음 스크립트를 사용하여 찾을 수 있습니다.

```
stream = modeler.script.stream()
node = stream.findByType("filter", None)
...
```

다른 경우로는, 노드의 ID(노드 대화 상자의 주석 탭에 표시되는)가 알려진 경우, 해당 ID를 사용하여 노드를 찾을 수 있습니다. 예를 들면,

```
stream = modeler.script.stream()
node = stream.findByID("id32FJT71G2") # the filter node ID
...
```

특성 설정

노드, 스트림, 모델 및 출력은 모두 액세스할 수 있고 대부분의 경우에 설정할 수 있는 특성을 갖습니다. 특성은 일반적으로 오브젝트의 작동이나 외형을 수정하는 데 사용합니다. 오브젝트 특성 액세스 및 설정에 사용할 수 있는 메소드가 다음 표에 요약되어 있습니다.

표 11. 오브젝트 특성 액세스 및 설정을 위한 메소드

메소드	리턴 유형	설명
p.getPropertyValue(propertyName)	Object	이름 지정된 특성의 값 또는 그런 특성이 없는 경우 None을 리턴합니다.
p.setPropertyValue(propertyName, value)	적용되지 않음	이름 지정된 특성의 값을 설정합니다.
p.setPropertyValues(properties)	적용되지 않음	이름 지정된 특성의 값을 설정합니다. 특성 맵의 각 항목은 특성 이름 및 해당 특성에 지정되어야 하는 값을 나타내는 키로 구성됩니다.
p.getKeyedPropertyValue(propertyName, keyName)	Object	이름 지정된 특성의 값과 연관된 키 또는 그런 특성이나 키가 없는 경우 None을 리턴합니다.
p.setKeyedPropertyValue(propertyName, keyName, value)	적용되지 않음	이름 지정된 특성 및 키의 값을 설정합니다.

예를 들어, 스트림의 시작에 있는 가변파일 노드의 값을 설정하려는 경우 다음 스크립트를 사용할 수 있습니다.

```
stream = modeler.script.stream()
node = stream.findByType("variablefile", None)
node.setPropertyValue("full_filename", "$CLEO/DEMOS/DRUG1n")
...
```

또는, 필터 노드에서 필드를 필터링하기 원할 수 있습니다. 이 경우에는 값도 필드 이름에 키로 입력됩니다. 예를 들어,

```
stream = modeler.script.stream()
# Locate the filter node ...
node = stream.findByType("filter", None)
# ... and filter out the "Na" field
node.setKeyedPropertyValue("include", "Na", False)
```

노드 작성 및 스트림 수정

일부 상황에서는 기존 스트림에 새 노드를 추가하기 원할 수 있습니다. 기존 스트림에 노드 추가는 일반적으로 다음 작업을 포함합니다.

1. 노드 작성.
2. 기존 스트림 플로우에 노드 링크.

노드 작성

스트림은 많은 노드 작성 방법을 제공합니다. 이들 메소드가 다음 표에 요약되어 있습니다.

표 12. 노드 작성용 메소드

메소드	리턴 유형	설명
s.create(nodeType, name)	노드	지정된 유형의 노드를 작성하고 이를 지정된 스트림에 추가합니다.
s.createAt(nodeType, name, x, y)	노드	지정된 유형의 노드를 작성하고 이를 지정된 위치에서 지정된 스트림에 추가합니다. $x < 0$ 또는 $y < 0$ 인 경우, 위치가 설정되지 않습니다.
s.createModelApplier(modelOutput, name)	노드	제공되는 모델 출력 오브젝트로부터 파생되는 모델 적용자 노드를 작성합니다.

예를 들어, 스트림에서 새 유형 노드를 작성하기 위해 다음 스크립트를 사용할 수 있습니다.

```
stream = modeler.script.stream()
# Create a new type node
node = stream.create("type", "My Type")
```

노드 링크 및 링크 해제

새 노드가 스트림 내에서 작성될 때, 사용할 수 있기 전에 노드의 시퀀스에 연결되어야 합니다. 스트림은 노드 링크 및 링크 해제를 위한 많은 메소드를 제공합니다. 이들 메소드가 다음 표에 요약되어 있습니다.

표 13. 노드 링크 및 링크 해제를 위한 메소드

메소드	리턴 유형	설명
s.link(source, target)	적용되지 않음	소스와 목표 노드 사이의 새 링크를 작성합니다.
s.link(source, targets)	적용되지 않음	소스 노드와 제공되는 목록의 각 목표 노드 사이에 새 링크를 작성합니다.
s.linkBetween(inserted, source, target)	적용되지 않음	두 개의 다른 노드 인스턴스(소스 및 목표 노드) 사이의 노드를 연결하고 삽입된 노드의 위치를 그들 사이로 설정합니다. 소스와 목표 노드 사이의 모든 직접 링크가 먼저 제거됩니다.
s.linkPath(path)	적용되지 않음	노드 인스턴스 사이에 새 경로를 작성합니다. 처음 노드가 두 번째에 연결되고, 두 번째가 세 번째에 연결되는 식입니다.
s.unlink(source, target)	적용되지 않음	소스와 목표 노드 사이의 모든 직접 링크를 제거합니다.
s.unlink(source, targets)	적용되지 않음	소스 노드와 목표 목록의 각 오브젝트 사이의 모든 직접 링크를 제거합니다.
s.unlinkPath(path)	적용되지 않음	노드 인스턴스 사이에 존재하는 모든 경로를 제거합니다.
s.disconnect(node)	적용되지 않음	제공되는 노드와 지정된 스트림에 있는 다른 모든 노드 사이의 모든 링크를 제거합니다.
s.isValidLink(source, target)	boolean	지정된 소스와 목표 노드 사이에 링크를 작성하는 것이 유효한 경우 True를 리턴합니다. 이 메소드는 두 오브젝트 모두가 지정된 스트림에 속하는지, 소스 노드가 링크를 공급하고 목표 노드가 링크를 받을 수 있는지 그리고 그런 링크를 작성해도 스트림에 순환성을 유발하지 않는지 검사합니다.

뒤에 오는 스크립트 예는 다음의 5가지 작업을 수행합니다.

1. 가변파일 입력 노드, 필터 노드 및 테이블 출력 노드를 작성합니다.
2. 노드를 서로 연결합니다.
3. 변수파일 입력 노드에서 파일 이름을 설정합니다.
4. 결과 출력에서 "Drug" 필드를 필터링합니다.
5. 테이블 노드를 실행합니다.

```
stream = modeler.script.stream()
filenode = stream.createAt("variablefile", "My File Input ", 96, 64)
filternode = stream.createAt("filter", "Filter", 192, 64)
tablenode = stream.createAt("table", "Table", 288, 64)
stream.link(filenode, filternode)
stream.link(filternode, tablenode)
```

```

filenode.setPropertyValue("full_filename", "$CLEO_DEMOS/DRUG1n")
filternode.setKeyedPropertyValue("include", "Drug", False)
results = []
tablenode.run(results)

```

노드 가져오기, 바꾸기 및 삭제

노드 작성 및 연결뿐 아니라, 종종 스트림에서 노드를 바꾸고 삭제해야 합니다. 노드 가져오기, 바꾸기 및 삭제에 사용 가능한 메소드가 다음 표에 요약되어 있습니다.

표 14. 노드 가져오기, 바꾸기 및 삭제를 위한 메소드

메소드	리턴 유형	설명
<code>s.replace(originalNode, replacementNode, discardOriginal)</code>	적용되지 않음	지정된 스트림에서 지정된 노드를 바꿉니다. 원래 노드와 대체 노드가 둘 다 지정된 스트림의 소유여야 합니다.
<code>s.insert(source, nodes, newIDs)</code>	List	제공되는 목록에 노드의 사본을 삽입합니다. 제공되는 목록의 모든 노드가 지정된 스트림에 포함된다고 가정합니다. <code>newIDs</code> 플래그는 각 노드에 대해 새 ID가 생성되어야 하는지 또는 기존 ID를 복사 및 사용할지 여부를 표시합니다. 스트림의 모든 노드가 고유 ID를 갖는다고 가정하므로, 소스 스트림이 지정된 스트림과 동일한 경우 이 플래그는 True로 설정되어야 합니다. 이 메소드는 새로 삽입된 노드의 목록을 리턴하며, 노드의 순서가 정의되지 않습니다(즉, 순서 지정이 반드시 입력 목록에 있는 노드의 순서와 같지는 않음).
<code>s.delete(node)</code>	적용되지 않음	지정된 스트림에서 지정된 노드를 삭제합니다. 지정된 스트림이 해당 노드를 소유해야 합니다.
<code>s.deleteAll(nodes)</code>	적용되지 않음	지정된 스트림에서 모든 지정된 노드를 삭제합니다. 컬렉션의 모든 노드가 지정된 스트림에 속합니다.
<code>s.clear()</code>	적용되지 않음	지정된 스트림에서 모든 노드를 삭제합니다.

스트림에서 노드 횡단

일반 요구사항은 특정 노드의 업스트림 또는 다운스트림인 노드를 식별하는 것입니다. 스트림은 이들 노드를 식별하는 데 사용할 수 있는 많은 메소드를 제공합니다. 이들 메소드가 다음 표에 요약되어 있습니다.

표 15. 업스트림 및 다운스트림 노드를 식별하는 메소드

메소드	리턴 유형	설명
<code>s.iterator()</code>	Iterator	지정된 스트림에 포함된 노드 오브젝트에 대한 반복자를 리턴합니다. 스트림이 <code>next()</code> 함수의 호출 사이에 수정되는 경우, 반복자의 작동은 정의되지 않습니다.

표 15. 업스트림 및 다운스트림 노드를 식별하는 메소드 (계속)

메소드	리턴 유형	설명
s.predecessorAt(node, index)	노드	제공되는 노드의 지정된 직전 선행자 또는 지수가 경계를 벗어난 경우 None을 리턴합니다.
s.predecessorCount(node)	int	제공되는 노드의 직전 선행자 수를 리턴합니다.
s.predecessors(node)	List	제공되는 노드의 직전 선행자를 리턴합니다.
s.successorAt(node, index)	노드	제공되는 노드의 지정된 직후 후속자 또는 지수가 경계를 벗어난 경우 None을 리턴합니다.
s.successorCount(node)	int	제공되는 노드의 직후 후속자 수를 리턴합니다.
s.successors(node)	List	제공되는 노드의 직후 후속자를 리턴합니다.

항목 지우기 또는 제거

레거시 스크립팅은 clear 명령의 다양한 사용을 지원합니다. 예:

- clear outputs는 관리자 팔레트에서 모든 출력 항목을 삭제합니다.
- clear generated palette는 모델 팔레트에서 모든 모델 너깃을 지웁니다.
- clear stream은 스트림의 내용을 제거합니다.

Python 스크립팅은 비슷한 함수 세트를 지원하는데, removeAll() 명령은 스트림, 출력, 모델 관리자를 지우는 데 사용합니다. 예를 들어,

- 스트림 관리자를 지우려면,

```
session = modeler.script.session()
session.getStreamManager.removeAll()
```

- 출력 관리자를 지우려면,

```
session = modeler.script.session()
session.getDocumentOutputManager().removeAll()
```

- 모델 관리자를 지우려면,

```
session = modeler.script.session()
session.getModelOutputManager().removeAll()
```

노드에 관한 정보 얻기

노드는 데이터 가져오기 및 내보내기 노드, 모델 작성 노드 및 다른 유형의 노드와 같은 여러 범주로 구분됩니다. 모든 노드는 노드에 관한 정보를 찾기 위해 사용할 수 있는 많은 메소드를 제공합니다.

노드의 ID, 이름 및 레이블을 얻는 데 사용할 수 있는 메소드가 다음 표에 요약되어 있습니다.

표 16. 노드의 ID, 이름 및 레이블을 얻기 위한 메소드

메소드	리턴 유형	설명
n.getLabel()	string	지정된 노드의 표시 레이블을 리턴합니다. 레이블은 특성이 비어 있지 않은 문자열이고 use_custom_name 특성이 설정되지 않은 경우에만 custom_name 특성의 값입니다. 그렇지 않으면 레이블은 getName()의 값입니다.
n.setLabel(label)	적용되지 않음	지정된 노드의 표시 레이블을 설정합니다. 새 레이블이 비어 있지 않은 문자열인 경우 custom_name 특성에 지정되며 false가 use_custom_name 특성에 지정되어 지정된 레이블이 우선권을 갖습니다. 그렇지 않으면 빈 문자열이 custom_name 특성에 지정되고 True가 use_custom_name에 지정됩니다.
n.getName()	string	지정된 노드의 이름을 리턴합니다.
n.getID()	string	지정된 노드의 ID를 리턴합니다. 새 노드가 작성될 때마다 새 ID가 작성됩니다. ID는 스트림의 일부로 저장될 때 노드와 함께 지속되어 스트림이 열릴 때 노드 ID가 유지됩니다. 그러나 저장된 노드가 스트림에 삽입되는 경우 삽입된 노드는 새 오브젝트인 것으로 간주되고 새 ID가 할당됩니다.

노드에 관한 기타 정보를 얻는 데 사용할 수 있는 메소드가 다음 표에 요약되어 있습니다.

표 17. 노드에 관한 정보를 얻기 위한 메소드

메소드	리턴 유형	설명
n.getTypeName()	string	이 노드의 스크립팅 이름을 리턴합니다. 이것은 이 노드의 새 인스턴스를 작성하는 데 사용할 수 있는 것과 동일한 이름입니다.
n.isInitial()	부울	스트림의 시작 시에 발생하는 노드인 초기 노드인 경우 True를 리턴합니다.
n.isInline()	부울	스트림 중간에 발생하는 노드인 인라인 노드인 경우 True를 리턴합니다.
n.isTerminal()	부울	스트림의 끝에서 발생하는 노드인 터미널 노드인 경우 True를 리턴합니다.

표 17. 노드에 관한 정보를 얻기 위한 메소드 (계속)

메소드	리턴 유형	설명
n.getXPosition()	int	스트림에서 노드의 x 위치 오프셋을 리턴합니다.
n.getYPosition()	int	스트림에서 노드의 y 위치 오프셋을 리턴합니다.
n.setXYPosition(x, y)	적용되지 않음	스트림에서 노드의 위치를 설정합니다.
n.setPositionBetween(source, target)	적용되지 않음	노드가 제공되는 노드 사이에 위치되도록 스트림에서 노드의 위치를 설정합니다.
n.isCacheEnabled()	부울	캐시가 사용 가능하면 True, 그렇지 않으면 False를 리턴합니다.
n.setCacheEnabled(val)	적용되지 않음	이 오브젝트에 대한 캐시를 사용 또는 사용할 수 없습니다. 캐시가 가득 차고 캐싱이 사용 불가능하게 되면 캐시가 비워집니다.
n.isCacheFull()	부울	캐시가 가득 차면 True, 그렇지 않으면 False를 리턴합니다.
n.flushCache()	적용되지 않음	이 노드의 캐시를 비웁니다. 캐시가 사용 가능하지 않거나 가득 차지 않은 경우 효과가 없습니다.

제 4 장 스크립팅 API

스크립팅 API 소개

스크립팅 API는 광범위한 SPSS Modeler 기능에 대한 액세스를 제공합니다. 지금까지 설명한 모든 메소드는 이 API의 일부이며 추가 가져오기 없이 스크립트 내에서 내재적으로 액세스될 수 있습니다. 그러나 API 클래스를 참조하려는 경우 다음 명령문으로 API를 명시적으로 가져와야 합니다.

```
import modeler.api
```

이 가져오기 명령문은 많은 스크립팅 API 예에서 필요합니다.

스크립팅 API를 통해 사용할 수 있는 클래스, 메소드 및 모수에 대한 전체 안내서는 *IBM SPSS Modeler Python Scripting API Reference Guide*에서 찾을 수 있습니다.

예 1: 사용자 정의 필터를 사용한 노드 검색

33 페이지의 『노드 찾기』 절에 노드의 유형 이름을 검색 기준으로 사용하여 스트림에서 노드를 검색하는 예가 포함되어 있습니다. 일부 상황에서는 더 일반적인 검색이 필요하며 이것은 `NodeFilter` 클래스 및 `findAll()` 메소드를 사용하여 구현될 수 있습니다. 이런 종류의 검색에는 다음 두 단계가 포함됩니다.

1. `NodeFilter`를 확장하고 `accept()` 메소드의 사용자 정의 버전을 구현하는 새 클래스 작성.
2. 이 새 클래스의 인스턴스로 스트림 `findAll()` 메소드 호출. 이것은 `accept()` 메소드에서 정의된 기준을 만족하는 모든 노드를 리턴합니다.

다음 예는 노드 캐시가 사용 가능한 스트림에서 노드를 검색하는 방법을 보여줍니다. 노드의 리턴된 목록을 사용하여 이들 노드의 캐시를 비우거나 사용 안할 수 있습니다.

```
import modeler.api
```

```
class CacheFilter(modeler.api.NodeFilter):  
    """A node filter for nodes with caching enabled"""  
    def accept(this, node):  
        return node.isCacheEnabled()
```

```
cachingnodes = modeler.script.stream().findAll(CacheFilter(), False)
```

예제 2: 권한에 따라 디렉토리 또는 파일 정보를 가져오도록 허용

사용자에게 PSAPI가 열리지 않게 하려면 `session.getServerFileSystem()`라는 메소드를 통해 PSAPI 함수를 호출하여 파일 시스템 오브젝트를 작성할 수 있습니다.

다음 예에서는 IBM SPSS Modeler Server에 연결하는 사용자의 권한을 기반으로 디렉토리나 파일 정보를 가져올 수 있는 방법을 보여줍니다.

```

import modeler.api
stream = modeler.script.stream()
sourceNode = stream.findByID('')
session = modeler.script.session()
fileSystem = session.getServerFileSystem()
parameter = stream.getParameterValue('VPATH')
serverDirectory = fileSystem.getServerFile(parameter)
files = fileSystem.GetFiles(serverDirectory)
for f in files:
    if f.isDirectory():
        print 'Directory:'
    else:
        print 'File:'
        sourceNode.setPropertyValue('full_filename',f.getPath())
        break
    print f.getName(),f.getPath()
stream.execute()

```

메타데이터: 데이터에 관한 정보

노드가 스트림에서 서로 연결되기 때문에, 각 노드에서 사용 가능한 열이나 필드에 관한 정보가 사용 가능합니다. 예를 들어, Modeler UI에서 이것을 사용하여 정렬 또는 통합할 필드를 선택할 수 있습니다. 이 정보를 데이터 모델이라고 합니다.

스크립트는 또한 노드로 들어오거나 나가는 필드를 보고 데이터 모델에 액세스할 수 있습니다. 일부 노드의 경우, 입력 및 출력 데이터 모델은 동일합니다. 예를 들어 정렬 노드는 레코드를 단순히 다시 정렬하지만 데이터 모델을 변경하지 않습니다. 파생 노드 같은 일부 노드는 새 필드를 추가할 수 있습니다. 필터 노드 같은 다른 노드는 필드를 이름 변경 또는 제거할 수 있습니다.

다음 예에서 스크립트는 표준 IBM SPSS Modeler druglearn.str 스트림을 취하고, 각 필드에 대해 입력 필드 중 하나가 삭제된 모델을 작성합니다. 다음 방법으로 이를 수행합니다.

1. 유형 노드에서 출력 데이터 모델 액세스.
2. 출력 데이터 모델의 각 필드를 루핑.
3. 각 입력 필드에 대한 필터 노드 수정.
4. 작성될 모델의 이름 변경.
5. 모델 작성 노드 실행.

참고: druglearn.str 스트림에서 스크립트를 실행하기 전에, 스크립팅 언어를 Python으로 설정하는 것을 기억하십시오(스트림이 IBM SPSS Modeler의 이전 버전에서 작성되었으므로 스트림 스크립팅 언어가 레거시로 설정됨).

```

import modeler.api

stream = modeler.script.stream()
filternode = stream.findByType("filter", None)
typenode = stream.findByType("type", None)
c50node = stream.findByType("c50", None)
# Always use a custom model name
c50node.setPropertyValue("use_model_name", True)

```



```

lastRemoved = None
fields = typenode.getOutputDataModel()
for field in fields:
    # If this is the target field then ignore it
    if field.getModelingRole() == modeler.api.ModelingRole.OUT:
        continue

    # Re-enable the field that was most recently removed
    if lastRemoved != None:
        filternode.setKeyedPropertyValue("include", lastRemoved, True)

    # Remove the field
    lastRemoved = field.getColumnname()
    filternode.setKeyedPropertyValue("include", lastRemoved, False)

    # Set the name of the new model then run the build
    c50node.setPropertyValue("model_name", "Exclude " + lastRemoved)
    c50node.run([])

```

DataModel 오브젝트는 데이터 모델 내의 필드 및 열에 관한 정보에 액세스하기 위한 많은 메소드를 제공합니다. 이들 메소드가 다음 표에 요약되어 있습니다.

표 18. 필드 또는 열에 관한 정보 액세스를 위한 DataModel 오브젝트 메소드

메소드	리턴 유형	설명
d.getColumnCount()	int	데이터 모델의 열 수를 리턴합니다.
d.columnIterator()	Iterator	"기본" 삽입 순서로 각 열을 리턴하는 반복자를 리턴합니다. 반복자는 열의 인스턴스를 리턴합니다.
d.nameIterator()	Iterator	"기본" 삽입 순서로 각 열의 이름을 리턴하는 반복자를 리턴합니다.
d.contains(name)	부울	제공된 이름의 열이 이 DataModel에 존재하는 경우 True, 그렇지 않으면 False를 리턴합니다.
d.getColumn(name)	Column	지정된 이름의 열을 리턴합니다.
d.getColumnGroup(name)	ColumnGroup	이름 지정된 열 그룹 또는 그런 열 그룹이 존재하지 않는 경우 None을 리턴합니다.
d.getColumnGroupCount()	int	이 데이터 모델에 있는 열 그룹의 수를 리턴합니다.
d.columnGroupIterator()	Iterator	각 열 그룹을 차례로 리턴하는 반복자를 리턴합니다.
d.toArray()	Column[]	데이터 모델을 열의 배열로서 리턴합니다. 열은 "기본" 삽입 순서로 정렬됩니다.

각 필드(열 오브젝트)는 열에 관한 정보에 액세스하기 위한 많은 메소드를 포함합니다. 아래 테이블은 이들의 선택을 보여줍니다.

표 19. 열에 관한 정보 액세스를 위한 열 오브젝트 메소드

메소드	리턴 유형	설명
c.getColumnName()	string	열의 이름을 리턴합니다.
c.getColumnLabel()	string	열의 레이블 또는 열과 연관된 레이블이 없는 경우 빈 문자열을 리턴합니다.
c.getMeasureType()	MeasureType	열의 측도 유형을 리턴합니다.
c.getStorageType()	StorageType	열에 대한 저장 유형을 리턴합니다.
c.isMeasureDiscrete()	부울	열이 이산형인 경우 True를 리턴합니다. 세트 또는 플래그인 열이 이산형으로 간주됩니다.
c.isModelOutputColumn()	부울	열이 모델 출력 열인 경우 True를 리턴합니다.
c.isStorageDatetime()	부울	열의 저장 공간이 시간, 날짜 또는 시간 소인 값인 경우 True를 리턴합니다.
c.isStorageNumeric()	부울	열의 저장 공간이 정수 또는 실수인 경우 True를 리턴합니다.
c.isValidValue(value)	부울	지정된 값이 이 저장 공간에 유효한 경우 True를 리턴하고, 유효한 열 값이 알려질 때 valid를 리턴합니다.
c.getModelingRole()	ModelingRole	열에 대한 모델링 역할을 리턴합니다.
c.getSetValues()	Object[]	열에 대한 유효한 값의 배열 또는 값이 알려지지 않거나 열이 세트가 아닌 경우 None을 리턴합니다.
c.getValueLabel(value)	string	열에 있는 값에 대한 레이블 또는 값과 연관된 레이블이 없는 경우 빈 문자열을 리턴합니다.
c.getFalseFlag()	Object	열에 대한 "false" 표시기 값 또는 값이 알려지지 않거나 열이 플래그가 아닌 경우 None을 리턴합니다.
c.getTrueFlag()	Object	열에 대한 "true" 표시기 값 또는 값이 알려지지 않거나 열이 플래그가 아닌 경우 None을 리턴합니다.
c.getLowerBound()	Object	열에 있는 값에 대한 하한 값 또는 값이 알려지지 않거나 열이 연속형이 아닌 경우 None을 리턴합니다.
c.getUpperBound()	Object	열에 있는 값에 대한 상한 값 또는 값이 알려지지 않거나 열이 연속형이 아닌 경우 None을 리턴합니다.

열에 관한 정보에 액세스하는 대부분의 메소드는 DataModel 오브젝트 자체에서 정의되는 동등한 메소드를 갖고 있음을 참고하십시오. 예를 들어 다음 두 개의 명령문은 동등합니다.

```
dataModel.getColumn("someName").getModelingRole()
dataModel.getModelingRole("someName")
```

생성된 오브젝트 액세스

스트림 실행은 일반적으로 추가 출력 오브젝트 생성을 포함합니다. 이들 추가 오브젝트는 새 모델 또는 후속 실행에서 사용할 정보를 제공하는 출력의 일부일 수 있습니다.

아래 예에서, `druglearn.str` 스트림은 스트림의 시작점으로서 다시 사용됩니다. 이 예에서 스트림의 모든 노드가 실행되고 그 결과는 목록에 저장됩니다. 그런 다음 스크립트는 결과를 순환하고, 실행 결과인 모든 모델 출력이 IBM SPSS Modeler 모델(.gm) 파일로서 저장되고 모델은 PMML 내보내집니다.

```
import modeler.api

stream = modeler.script.stream()

# Set this to an existing folder on your system.
# Include a trailing directory separator
modelFolder = "C:/temp/models/"

# Execute the stream
models = []
stream.runAll(models)

# Save any models that were created
taskrunner = modeler.script.session().getTaskRunner()
for model in models:
    # If the stream execution built other outputs then ignore them
    if not(isinstance(model, modeler.api.ModelOutput)):
        continue

    label = model.getLabel()
    algorithm = model.getModelDetail().getAlgorithmName()

    # save each model...
    modelFile = modelFolder + label + algorithm + ".gm"
    taskrunner.saveModelToFile(model, modelFile)

    # ...and export each model PMML...
    modelFile = modelFolder + label + algorithm + ".xml"
    taskrunner.exportModelToFile(model, modelFile, modeler.api.FileFormat.XML)
```

작업 실행기 클래스가 다양한 공통 작업을 실행하는 편리한 방법을 제공합니다. 이 클래스에서 사용할 수 있는 메소드가 다음 표에 요약되어 있습니다.

표 20. 공통 작업 수행을 위한 작업 실행기 클래스의 메소드

메소드	리턴 유형	설명
<code>t.createStream(name, autoConnect, autoManage)</code>	스트림	새 스트림을 작성하고 리턴합니다. 스트림을 사용자에게 보이게 하지 않고 스트림을 개인용으로 작성해야 하는 코드는 <code>autoManage</code> 플래그를 <code>False</code> 로 설정해야 합니다.
<code>t.exportDocumentToFile(documentOutput, filename, fileFormat)</code>	적용되지 않음	스트림 설명을 지정된 파일 형식을 사용하여 파일로 내보냅니다.

표 20. 공통 작업 수행을 위한 작업 실행기 클래스의 메소드 (계속)

메소드	리턴 유형	설명
t.exportModelToFile(modelOutput, filename, fileFormat)	적용되지 않음	모델을 지정된 파일 형식을 사용하여 파일로 내보냅니다.
t.exportStreamToFile(stream, filename, fileFormat)	적용되지 않음	스트림을 지정된 파일 형식을 사용하여 파일로 내보냅니다.
t.insertNodeFromFile(filename, diagram)	노드	지정된 파일에서 노드를 읽고 해당 노드를 제공된 다이어그램에 삽입하여 리턴합니다. 이것을 사용하면 노드 및 수퍼노드 오브젝트를 둘 다 읽을 수 있습니다.
t.openDocumentFromFile(filename, autoManage)	DocumentOutput	지정된 파일에서 문서를 읽고 리턴합니다.
t.openModelFromFile(filename, autoManage)	ModelOutput	지정된 파일에서 모델을 읽고 리턴합니다.
t.openStreamFromFile(filename, autoManage)	스트림	지정된 파일에서 스트림을 읽고 리턴합니다.
t.saveDocumentToFile(documentOutput, filename)	적용되지 않음	문서를 지정된 파일 위치에 저장합니다.
t.saveModelToFile(modelOutput, filename)	적용되지 않음	모델을 지정된 파일 위치에 저장합니다.
t.saveStreamToFile(stream, filename)	적용되지 않음	스트림을 지정된 파일 위치에 저장합니다.

오류 처리

Python 언어는 try...except 코드 블록을 통해 오류 처리를 제공합니다. 이것은 스크립트 내에서 사용하여 예외를 트랩하고 그렇지 않으면 스크립트가 종료하게 만드는 문제점을 처리할 수 있습니다.

아래 스크립트 예에서는 IBM SPSS Collaboration and Deployment Services Repository로부터 모델을 검색하려고 시도합니다. 이 작업으로 인해 예외가 발생할 수 있습니다. 예를 들어, 리포지토리 로그인 신임 정보가 올바르게 설정되지 않았거나 리포지토리 경로가 올바르지 않습니다. 스크립트에서 이것은 ModelerException이 발생하게 만들 수 있습니다(IBM SPSS Modeler에 의해 생성되는 모든 예외는 modeler.api.ModelerException에서 파생됨).

```
import modeler.api

session = modeler.script.session()
try:
    repo = session.getRepository()
    m = repo.retrieveModel("/some-non-existent-path", None, None, True)
    # print goes to the Modeler UI script panel Debug tab
    print "Everything OK"
except modeler.api.ModelerException, e:
    print "An error occurred:", e.getMessage()
```

참고: 일부 스크립팅 작업은 표준 Java 예외가 발생하게 만들 수 있습니다. 이들은 ModelerException에서 파생되지 않습니다. 이들 예외를 처리하기 위해, 추가 예외 블록을 사용하여 모든 Java 예외를 처리(catch)할 수 있습니다. 예:

```

import modeler.api

session = modeler.script.session()
try:
    repo = session.getRepository()
    m = repo.retrieveModel("/some-non-existent-path", None, None, True)
    # print goes to the Modeler UI script panel Debug tab
    print "Everything OK"
except modeler.api.ModelerException, e:
    print "An error occurred:", e.getMessage()
except java.lang.Exception, e:
    print "A Java exception occurred:", e.getMessage()

```

스트림, 세션 및 수퍼노드 모수

모수는 값을 스크립트에 직접 하드 코딩하는 대신 런타임 시에 값을 전달하는 유용한 방법을 제공합니다. 모수 및 해당 값은 스트림의 경우와 동일한 방법으로, 즉 스트림 또는 수퍼노드의 모수 표에 있는 항목이나 명령행의 모수로서 정의됩니다. 스트림 및 수퍼노드 클래스는 다음 표에 표시된 것처럼 `ParameterProvider` 오브젝트에 의해 정의되는 함수 세트를 구현합니다. 세션은 해당 함수를 정의하는 오브젝트를 리턴하는 `getParameters()` 호출을 제공합니다.

표 21. `ParameterProvider` 오브젝트에 의해 정의되는 함수

메소드	리턴 유형	설명
<code>p.parameterIterator()</code>	Iterator	이 오브젝트에 대한 모수 이름의 반복자를 리턴합니다.
<code>p.getParameterDefinition(parameterName)</code>	ParameterDefinition	지정된 이름의 모수에 대한 모수 정의 또는 그런 모수가 이 제공자에 존재하지 않는 경우 None을 리턴합니다. 결과는 메소드가 호출될 때 정의의 스냅샷일 수 있으며 이 제공자를 통해 모수에 작성된 모든 후속 수정을 반영할 필요가 없습니다.
<code>p.getParameterLabel(parameterName)</code>	string	이름 지정된 모수의 레이블 또는 그런 모수가 없는 경우 None을 리턴합니다.
<code>p.setParameterLabel(parameterName, label)</code>	적용되지 않음	이름 지정된 모수의 레이블을 설정합니다.
<code>p.getParameterStorage(parameterName)</code>	ParameterStorage	이름 지정된 모수의 저장 공간 또는 그런 모수가 없는 경우 None을 리턴합니다.
<code>p.setParameterStorage(parameterName, storage)</code>	적용되지 않음	이름 지정된 모수의 저장 공간을 설정합니다.
<code>p.getParameterType(parameterName)</code>	ParameterType	이름 지정된 모수의 유형 또는 그런 모수가 없는 경우 None을 리턴합니다.
<code>p.setParameterType(parameterName, type)</code>	적용되지 않음	이름 지정된 모수의 유형을 설정합니다.
<code>p.getParameterValue(parameterName)</code>	Object	이름 지정된 모수의 값 또는 그런 모수가 없는 경우 None을 리턴합니다.
<code>p.setParameterValue(parameterName, value)</code>	적용되지 않음	이름 지정된 모수의 값을 설정합니다.

다음 예에서, 스크립트는 가장 낮은 평균 수입 데이터를 갖는 영역을 찾기 위해 일부 Telco 데이터를 통합합니다. 그런 다음 스트림 모수가 이 영역으로 설정됩니다. 이탈 모델이 나머지에서 작성되기 전에 해당 스트림 모수가 해당 영역을 데이터에서 제외하기 위해 선택 노드에서 사용됩니다.

스크립트가 선택 노드 자체를 생성하므로 올바른 값을 선택 노드 표현식에 직접 생성했을 수 있기 때문에 이 예는 인위적입니다. 그러나 스트림은 일반적으로 사전 작성되므로 이 방식으로 모수를 설정하는 것이 유용한 예를 제공합니다.

스크립트 예의 첫 번째 부분은 최저 평균 수입을 갖는 영역을 포함하는 스트림 모수를 작성합니다. 이 스크립트는 또한 통합 분기 및 모델 작성 분기에 노드를 작성하고 서로 연결합니다.

```
import modeler.api

stream = modeler.script.stream()

# Initialize a stream parameter
stream.setParameterStorage("LowestRegion", modeler.api.ParameterStorage.INTEGER)

# First create the aggregation branch to compute the average income per region
statisticsimportnode = stream.createAt("statisticsimport", "SPSS File", 114, 142)
statisticsimportnode.setPropertyValue("full_filename", "$CLEO_DEMOS/telco.sav")
statisticsimportnode.setPropertyValue("use_field_format_for_storage", True)

aggregatenode = modeler.script.stream().createAt("aggregate", "Aggregate", 294, 142)
aggregatenode.setPropertyValue("keys", ["region"])
aggregatenode.setKeyedPropertyValue("aggregates", "income", ["Mean"])

tablenode = modeler.script.stream().createAt("table", "Table", 462, 142)

stream.link(statisticsimportnode, aggregatenode)
stream.link(aggregatenode, tablenode)

selectnode = stream.createAt("select", "Select", 210, 232)
selectnode.setPropertyValue("mode", "Discard")
# Reference the stream parameter in the selection
selectnode.setPropertyValue("condition", "'region' = '$P-LowestRegion'")

typenode = stream.createAt("type", "Type", 366, 232)
typenode.setKeyedPropertyValue("direction", "churn", "Target")

c50node = stream.createAt("c50", "C5.0", 534, 232)

stream.link(statisticsimportnode, selectnode)
stream.link(selectnode, typenode)
stream.link(typenode, c50node)
```

스크립트 예는 다음 스트림을 작성합니다.

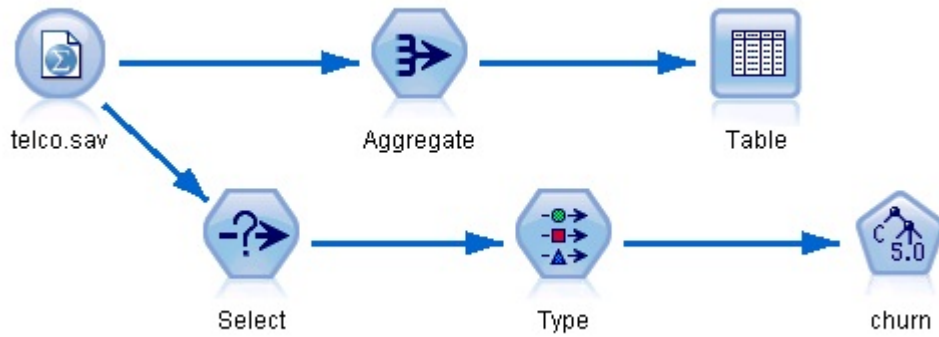


그림 5. 스크립트 예의 결과인 스트림

스크립트 예의 다음 파트는 통합 분기의 끝에서 테이블 노드를 실행합니다.

```
# First execute the table node
results = []
tablenode.run(results)
```

스크립트 예의 다음 파트는 테이블 노드의 실행에 의해 생성된 테이블 출력에 액세스합니다. 스크립트는 그런 다음 테이블의 행을 통해 반복하여 최저 평균 수입을 갖는 영역을 찾습니다.

```
# Running the table node should produce a single table as output
table = results[0]

# table output contains a RowSet so we can access values as rows and columns
rowset = table.getRowSet()
min_income = 1000000.0
min_region = None

# From the way the aggregate node is defined, the first column
# contains the region and the second contains the average income
row = 0
rowcount = rowset.getRowCount()
while row < rowcount:
    if rowset.getValueAt(row, 1) < min_income:
        min_income = rowset.getValueAt(row, 1)
        min_region = rowset.getValueAt(row, 0)
    row += 1
```

스크립트의 다음 파트는 최저 평균 수입을 갖는 영역을 사용하여 이전에 작성된 "LowestRegion" 스트림 모수를 설정합니다. 그런 다음 스크립트는 훈련 데이터에서 제외된 지정된 영역을 갖는 모델 작성을 실행합니다.

```
# Check that a value was assigned
if min_region != None:
    stream.setParameterValue("LowestRegion", min_region)
else:
    stream.setParameterValue("LowestRegion", -1)
```

```
# Finally run the model builder with the selection criteria
c50node.run([])
```

전체 스크립트 예가 아래에 표시됩니다.

```
import modeler.api

stream = modeler.script.stream()

# Create a stream parameter
stream.setParameterStorage("LowestRegion", modeler.api.ParameterStorage.INTEGER)

# First create the aggregation branch to compute the average income per region
statisticsimportnode = stream.createAt("statisticsimport", "SPSS File", 114, 142)
statisticsimportnode.setPropertyValue("full_filename", "$CLEO_DEMOS/telco.sav")
statisticsimportnode.setPropertyValue("use_field_format_for_storage", True)

aggregatenode = modeler.script.stream().createAt("aggregate", "Aggregate", 294, 142)
aggregatenode.setPropertyValue("keys", ["region"])
aggregatenode.setKeyedPropertyValue("aggregates", "income", ["Mean"])

tablenode = modeler.script.stream().createAt("table", "Table", 462, 142)

stream.link(statisticsimportnode, aggregatenode)
stream.link(aggregatenode, tablenode)

selectnode = stream.createAt("select", "Select", 210, 232)
selectnode.setPropertyValue("mode", "Discard")
# Reference the stream parameter in the selection
selectnode.setPropertyValue("condition", "'region' = '$P-LowestRegion'")

typenode = stream.createAt("type", "Type", 366, 232)
typenode.setKeyedPropertyValue("direction", "churn", "Target")

c50node = stream.createAt("c50", "C5.0", 534, 232)

stream.link(statisticsimportnode, selectnode)
stream.link(selectnode, typenode)
stream.link(typenode, c50node)

# First execute the table node
results = []
tablenode.run(results)

# Running the table node should produce a single table as output
table = results[0]

# table output contains a RowSet so we can access values as rows and columns
rowset = table.getRowSet()
min_income = 1000000.0
min_region = None

# From the way the aggregate node is defined, the first column
# contains the region and the second contains the average income
row = 0
rowcount = rowset.getRowCount()
while row < rowcount:
    if rowset.getValueAt(row, 1) < min_income:
        min_income = rowset.getValueAt(row, 1)
        min_region = rowset.getValueAt(row, 0)
    row += 1
```



```

# Check that a value was assigned
if min_region != None:
    stream.setParameterValue("LowestRegion", min_region)
else:
    stream.setParameterValue("LowestRegion", -1)

# Finally run the model builder with the selection criteria
c50node.run([])

```

글로벌 값

글로벌 값은 지정된 필드에 대한 다양한 요약 통계량을 계산하는 데 사용됩니다. 이들 요약 값은 스트림 내의 어디에서나 액세스할 수 있습니다. 글로벌 값은 스트림을 통해 이름에 의해 액세스된다는 점에서 스트림 모수와 비슷합니다. 연관된 값이 스크립팅에 의해 또는 명령행에서 지정되는 대신 전역값 설정 노드가 실행될 때 자동으로 업데이트된다는 점에서 스트림 모수와는 다릅니다. 스트림에 대한 글로벌 값은 스트림의 `getGlobalValues()` 메소드를 호출하여 액세스됩니다.

`GlobalValues` 오브젝트는 다음 테이블에 표시되는 함수를 정의합니다.

표 22. `GlobalValues` 오브젝트에 의해 정의되는 함수

메소드	리턴 유형	설명
<code>g.fieldNameIterator()</code>	Iterator	하나 이상의 글로벌 값을 갖는 각 필드 이름에 대한 반복자를 리턴합니다.
<code>g.getValue(type, fieldName)</code>	Object	지정된 유형 및 필드 이름에 대한 글로벌 값 또는 값을 찾을 수 없는 경우 <code>None</code> 을 리턴합니다. 미래 함수가 다른 값 유형을 리턴할 수 있더라도, 리턴값은 일반적으로 숫자일 것으로 예상됩니다.
<code>g.getValues(fieldName)</code>	Map	지정된 필드 이름에 대한 알려진 항목을 포함하는 맵이나 필드에 대한 기존 항목이 없는 경우 <code>None</code> 을 리턴합니다.

`GlobalValues.Type`은 사용 가능한 요약 통계량의 유형을 정의합니다. 다음 요약 통계량이 사용 가능합니다.

- MAX: 필드의 최대값.
- MEAN: 필드의 평균 값.
- MIN: 필드의 최소값.
- STDDEV: 필드의 표준 편차.
- SUM: 필드에 있는 값의 합계.

예를 들어, 다음 스크립트는 "income" 필드의 평균값을 액세스하며, 이것은 전역값 설정 노드에 의해 계산됩니다.

```
import modeler.api

globals = modeler.script.stream().getGlobalValues()
mean_income = globals.getValue(modeler.api.GlobalValues.Type.MEAN, "income")
```

다중 스트림에 대한 작업: 독립형 스크립트

다중 스트림에 대해 작업하려면 독립형 스크립트를 사용해야 합니다. 독립형 스크립트는 IBM SPSS Modeler UI 내에서 편집 및 실행되거나 일괄처리 모드에서 명령행 모수로서 전달될 수 있습니다.

다음 독립형 스크립트는 두 스트림을 엽니다. 이들 스트림 중 하나는 모델을 작성하는 반면, 두 번째 스트림은 예측값의 분포를 도표로 표시합니다.

```
# Change to the appropriate location for your system
demosDir = "C:/Program Files/IBM/SPSS/Modeler/18.1/DEMOS/streams/"

session = modeler.script.session()
tasks = session.getTaskRunner()

# Open the model build stream, locate the C5.0 node and run it
buildstream = tasks.openStreamFromFile(demosDir + "druglearn.str", True)
c50node = buildstream.findByType("c50", None)
results = []
c50node.run(results)

# Now open the plot stream, find the Na_to_K derive and the histogram
plotstream = tasks.openStreamFromFile(demosDir + "drugplot.str", True)
derivenode = plotstream.findByType("derive", None)
histogramnode = plotstream.findByType("histogram", None)

# Create a model applier node, insert it between the derive and histogram nodes
# then run the histogram
applyc50 = plotstream.createModelApplier(results[0], results[0].getName())
applyc50.setPositionBetween(derivenode, histogramnode)
plotstream.linkBetween(applyc50, derivenode, histogramnode)
histogramnode.setPropertyValue("color_field", "$C-Drug")
histogramnode.run([])

# Finally, tidy up the streams
buildstream.close()
plotstream.close()
```

다음 예에서는 개방형 스트림(모든 스트림이 스트림 탭에 열림)도 반복하는 방법을 보여줍니다. 이 방법은 독립형 스크립트에서만 지원됩니다.

```
for stream in modeler.script.streams():
    print stream.getName()
```

제 5 장 스크립팅 팁

이 절에서는 스트림 실행 수정, 스크립트에서 인코딩된 비밀번호 사용, IBM SPSS Collaboration and Deployment Services Repository에서 오브젝트 액세스를 포함하여 스크립트 사용에 대한 팁과 기법의 개요를 제공합니다.

스트림 실행 수정

스트림이 실행될 때 터미널 노드가 기본 상황에 최적화된 순서로 실행됩니다. 어떤 경우에는 다른 실행 순서를 선호할 수 있습니다. 스트림의 실행 순서를 변경하려면 스트림 특성 대화 상자의 실행 탭에서 다음 단계를 완료하십시오.

1. 비어있는 스크립트로 시작하십시오.
2. 도구 모음의 기본 스크립트 추가 단추를 클릭하여 기본 스트림 스크립트를 추가하십시오.
3. 기본 스트림 스크립트에 있는 명령문의 순서를 명령문이 실행되기 원하는 순서로 변경하십시오.

노드 루핑

for 루프를 사용하여 스트림의 모든 노드를 순환할 수 있습니다. 예를 들어 다음 두 스크립트 예는 모든 노드를 순환하고 모든 필터 노드에 있는 필드 이름을 대문자로 변경합니다.

이 스크립트는 필드가 실제로 필터링되지 않는 경우에도 필터 노드를 갖는 모든 스트림에서 사용할 수 있습니다. 간단히 필드 이름을 보드 사이에 대문자로 변경하기 위해 모든 필드를 전달하는 필터 노드를 추가하십시오.

```
# Alternative 1: using the data model nameIterator() function
stream = modeler.script.stream()
for node in stream.iterator():
    if (node.getTypeName() == "filter"):
        # nameIterator() returns the field names
        for field in node.getInputDataModel().nameIterator():
            newname = field.upper()
            node.setKeyedPropertyValue("new_name", field, newname)

# Alternative 2: using the data model iterator() function
stream = modeler.script.stream()
for node in stream.iterator():
    if (node.getTypeName() == "filter"):
        # iterator() returns the field objects so we need
        # to call getColumnName() to get the name
        for field in node.getInputDataModel().iterator():
            newname = field.getColumnName().upper()
            node.setKeyedPropertyValue("new_name", field.getColumnName(), newname)
```

스크립트는 현재 스트림의 모든 노드를 순환하고 각 노드가 필터인지 여부를 검사합니다. 그 경우, 스크립트는 노드에 있는 각 필드를 루핑하고 `field.upper()` 또는 `field.getColumnName().upper()` 함수를 사용하여 이름을 대문자로 변경합니다.

IBM SPSS Collaboration and Deployment Services Repository의 오브젝트 액세스

IBM SPSS Collaboration and Deployment Services Repository에 대한 사용권이 있으면, 스크립트 명령을 사용하여 리포지토리에서 개체를 저장 및 검색할 수 있습니다. 리포지토리를 사용하여 엔터프라이즈 애플리케이션, 도구 및 솔루션의 컨텍스트에서 데이터 마이닝 모델 및 관련 예측 개체의 라이프 사이클을 관리하십시오.

IBM SPSS Collaboration and Deployment Services Repository에 연결

리포지토리에 액세스하기 위해서는 먼저 SPSS Modeler 사용자 인터페이스의 도구 메뉴나 명령행을 통해 리포지토리에 대한 유효한 연결을 설정해야 합니다. 자세한 정보는 71 페이지의 『IBM SPSS Collaboration and Deployment Services Repository 연결 인수』을 참조하십시오.

리포지토리 액세스

세션에서 리포지토리에 액세스할 수 있습니다. 예를 들어, 다음과 같습니다.

```
repo = modeler.script.session().getRepository()
```

리포지토리에서 개체 검색

스크립트 내에서 `retrieve*` 함수를 사용하여 스트림, 모델, 출력 및 노드를 포함한 다양한 개체에 액세스할 수 있습니다. 검색 함수에 대한 요약이 다음 표에 표시됩니다.

표 23. 검색 스크립트 함수

개체 유형	리포지토리 함수
스트림	<code>repo.retrieveStream(문자열 경로, 문자열 버전, 문자열 레이블, 부울 autoManage)</code>
모델	<code>repo.retrieveModel(문자열 경로, 문자열 버전, 문자열 레이블, 부울 autoManage)</code>
출력	<code>repo.retrieveDocument(문자열 경로, 문자열 버전, 문자열 레이블, 부울 autoManage)</code>
노드	<code>repo.retrieveProcessor(문자열 경로, 문자열 버전, 문자열 레이블, ProcessorDiagram 다이어그램)</code>

예를 들어, 다음 함수를 사용하여 리포지토리에서 스트림을 검색할 수 있습니다.

```
stream = repo.retrieveStream("/projects/retention/risk_score.str", None, "production", True)
```

이 예는 지정된 폴더에서 `risk_score.str` 스트림을 검색합니다. `production` 레이블은 검색할 스트림의 버전을 식별하며 마지막 모수는 SPSS Modeler가 스트림을 관리할 것인지 여부를 지정합니다.(예를 들어, SPSS Modeler 사용자 인터페이스가 표시되면 스트림이 스트림 탭에 표시되도록 설정합니다.) 대안으로 특정한 레이블 없는 버전을 사용할 수도 있습니다.

```
stream = repo.retrieveStream("/projects/retention/risk_score.str", "0:2015-10-12 14:15:41.281", None, True)
```

참고: 버전 및 레이블 매개변수가 모두 None이면 최신 버전이 리턴됩니다.

리포지토리 내의 개체 스코어링

리포지토리에서 개체를 저장하기 위해 스크립트를 사용하려면 store* 함수를 사용하십시오. 저장 함수에 대한 요약이 다음 표에 표시됩니다.

표 24. 저장 스크립트 함수

개체 유형	리포지토리 함수
스트림	repo.storeStream(ProcessorStream 스트림, 문자열 경로, 문자열 레이블)
모델	repo.storeModel(ModelOutput 모델 출력, 문자열 경로, 문자열 레이블)
출력	repo.storeDocument(DocumentOutput 문서 출력, 문자열 경로, 문자열 레이블)
노드	repo.storeProcessor(프로세서 노드, 문자열 경로, 문자열 레이블)

예를 들어, 다음 함수를 사용하여 risk_score.str 스트림의 새 버전을 저장할 수 있습니다.

```
versionId = repo.storeStream(stream, "/projects/retention/risk_score.str", "test")
```

이 예에서는 스트림의 새 버전을 저장하고 "test" 레이블을 이와 연관시키며 새로 작성된 버전에 대한 버전 표식을 리턴합니다.

참고: 레이블을 새 버전과 연관시키지 않으려면 레이블에 대해 None을 전달하십시오.

리포지토리 폴더 관리

리포지토리 내의 폴더를 사용하여 개체를 논리적 그룹으로 구성하면 연관된 개체를 더 쉽게 볼 수 있습니다. 다음 예에서 보듯이 createFolder() 함수를 사용하여 폴더를 작성하십시오.

```
newpath = repo.createFolder("/projects", "cross-sell")
```

이 예에서는 "/projects" 폴더 내에 "cross-sell"이라는 새 폴더를 작성합니다. 함수는 새 폴더의 전체 경로를 리턴합니다.

폴더의 이름을 변경하려면 renameFolder() 함수를 사용하십시오.

```
repo.renameFolder("/projects/cross-sell", "cross-sell-Q1")
```

첫 번째 매개변수는 이름을 변경할 폴더의 전체 경로이며 두 번째는 해당 폴더에 지정할 새 이름입니다.

비어 있는 폴더를 삭제하려면 deleteFolder() 함수를 사용하십시오.

```
repo.deleteFolder("/projects/cross-sell")
```

오브젝트 잠금 및 잠금 해제

스크립트에서 오브젝트를 잠궈서 다른 사용자가 기존 버전 중 하나를 업데이트하거나 새 버전을 작성하지 못하게 막을 수 있습니다. 또한 자신이 잠근 오브젝트를 잠금 해제할 수도 있습니다.

오브젝트를 잠금 및 잠금 해제하는 구문은 다음과 같습니다.

```
repo.lockFile(REPOSITORY_PATH)
repo.lockFile(URI)
```

```
repo.unlockFile(REPOSITORY_PATH)
repo.unlockFile(URI)
```

오브젝트 저장 및 검색에서와 같이, REPOSITORY_PATH는 리포지토리에서 오브젝트의 위치를 제공합니다. 경로는 인용 부호로 묶어야 하며 슬래시를 구분자로 사용하십시오. 대소문자를 구분하지 않습니다.

```
repo.lockFile("/myfolder/Stream1.str")
repo.unlockFile("/myfolder/Stream1.str")
```

또는 오브젝트의 위치를 제공하기 위해 리포지토리 경로 대신 URI(Uniform Resource Identifier)를 사용할 수 있습니다. URI는 접두문자 `spsscr:`을 포함해야 하며 인용 부호로 완전히 묶어야 합니다. 슬래시만 경로 구분자로 허용되고 공백은 인코딩되어야 합니다. 즉, 경로에서 공백 대신 `%20`을 사용하십시오. URI는 대소문자를 구분하지 않습니다. 예를 들면, 다음과 같습니다.

```
repo.lockFile("spsscr:///myfolder/Stream1.str")
repo.unlockFile("spsscr:///myfolder/Stream1.str")
```

오브젝트 잠금은 오브젝트의 모든 버전에 적용됨을 참고하십시오. 개별 버전을 잠금 또는 잠금 해제할 수 없습니다.

인코딩된 비밀번호 생성

어떤 경우에는 스크립트에 비밀번호를 포함해야 할 수도 있습니다. 예를 들어 비밀번호로 보호되는 데이터 소스에 액세스할 수 있습니다. 인코딩된 비밀번호는 다음 경우에 사용할 수 있습니다.

- 데이터베이스 소스 및 출력 노드의 노드 특성
- 서버에 로그인하기 위한 명령행 인수
- `.par` 파일(내보내기 노드의 출판 탭에서 생성되는 모수 파일)에 저장된 데이터베이스 연결 특성

사용자 인터페이스를 통해 Blowfish 알고리즘을 기반으로 인코딩된 비밀번호를 생성하는 도구를 사용할 수 있습니다(자세한 정보는 <http://www.schneier.com/blowfish.html> 참조). 인코딩된 후에는 비밀번호를 스크립트 파일 및 명령행 인수에 복사 및 저장할 수 있습니다. `databasenode` 및 `databaseexportnode`에 사용하는 노드 특성 `epassword`가 인코딩된 비밀번호를 저장합니다.

1. 인코딩된 비밀번호를 생성하려면 도구 메뉴에서 다음을 선택하십시오.

비밀번호 인코딩...

2. 비밀번호 텍스트 상자에 비밀번호를 지정하십시오.

3. 인코드를 클릭하여 비밀번호의 임의 인코딩을 생성하십시오.
4. 복사 단추를 클릭하여 인코딩된 비밀번호를 클립보드에 복사하십시오.
5. 원하는 스크립트나 모수에 비밀번호를 붙여넣으십시오.

스크립트 검사

독립형 스크립트 대화 상자의 도구 모음에 있는 빨간색 확인 단추를 클릭하여 모든 유형의 스크립트 구문을 빨리 검사할 수 있습니다.



그림 6. 스트림 스크립트 도구 모음 아이콘

스크립트 검사는 코드에 있는 모든 오류에 대해 경고하고 개선을 위한 권장사항을 작성합니다. 오류가 있는 행을 보려면 대화 상자의 하단에 있는 피드백을 클릭하십시오. 여기에서 오류는 빨간색으로 강조 표시됩니다.

명령행에서 스크립팅

스크립팅으로 일반적으로 사용자 인터페이스에서 수행되는 작업을 실행할 수 있습니다. IBM SPSS Modeler를 시작할 때 단순히 명령행에서 독립형 스트림을 지정하고 실행하십시오. 예를 들어,

```
client -script scores.txt -execute
```

-script 플래그는 지정된 스크립트를 로드하는 반면, -execute 플래그는 스크립트 파일의 모든 명령을 실행합니다.

이전 릴리스와의 호환성

IBM SPSS Modeler의 이전 릴리스에서 작성된 스크립트는 일반적으로 현재 릴리스에서 변경되지 않은 채로 작동합니다. 그러나 이제는 모델 너깃이 자동으로 스트림에 삽입될 수 있으며(이것이 기본 설정임), 스트림에 있는 해당 유형의 기존 너깃을 바꾸거나 보완할 수 있습니다. 이것이 실제로 발생하는지 여부는 스트림에 모델 추가 및 이전 모델 바꾸기 옵션(도구 > 옵션 > 사용자 옵션 > 알림)의 설정에 따라 다릅니다. 예를 들어 기존 너깃을 삭제하고 새 너깃을 삽입하여 너깃 대체가 처리되는 이전 릴리스의 스크립트를 수정해야 할 수 있습니다.

현재 릴리스에서 작성되는 스크립트는 이전 릴리스에서 작동하지 않을 수 있습니다.

이전 릴리스에서 작성된 스크립트가 그 이후에 대체(또는 더 이상 사용하지 않음)된 명령을 사용하는 경우, 이전 양식이 아직 지원되지만 경고 메시지가 표시됩니다. 예를 들어 이전 generated 키워드가 model로 바뀌었으며, clear generated는 clear generated palette로 바뀌었습니다. 이전 양식을 사용하는 스크립트는 아직 실행되지만 경고가 표시됩니다.

스트림 실행 결과 액세스

많은 IBM SPSS Modeler 노드는 모델 차트, 표 형식 데이터 같은 출력 오브젝트를 생성합니다. 많은 이들 출력은 스크립트가 후속 실행을 안내하기 위해 사용할 수 있는 유용한 값을 포함합니다. 이들 값은 각 컨테이너를 식별하는 태그나 ID를 사용하여 액세스할 수 있는 내용 컨테이너(간단히 컨테이너로 부름)로 그룹화됩니다. 이들 값을 액세스하는 방법은 해당 컨테이너가 사용하는 형식이나 "컨텐츠 모델"에 따라 다릅니다.

예를 들어, 많은 예측 모델 출력은 PMML이라는 XML의 변형을 사용하여 의사결정 트리가 각 분할에서 사용하는 필드나 신경망의 뉴런이 연결되는 방법과 강도 같이 모델에 관한 정보를 나타냅니다. PMML을 사용하는 모델 출력은 해당 정보에 액세스하기 위해 사용할 수 있는 XML 컨텐츠 모델을 제공합니다. 예를 들어,

```
stream = modeler.script.stream()
# Assume the stream contains a single C5.0 model builder node
# and that the datasource, predictors and targets have already been
# set up
modelbuilder = stream.findByType("c50", None)
results = []
modelbuilder.run(results)
modeloutput = results[0]

# Now that we have the C5.0 model output object, access the
# relevant content model
cm = modeloutput.getContentModel("PMML")

# The PMML content model is a generic XML-based content model that
# uses XPath syntax. Use that to find the names of the data fields.
# The call returns a list of strings match the XPath values
dataFieldNames = cm.getStringValues("/PMML/DataDictionary/DataField", "name")
```

IBM SPSS Modeler는 스크립팅에서 다음 컨텐츠 모델을 지원합니다.

- **테이블 컨텐츠 모델**은 행과 열로 표시되는 단순 표 형식 데이터에 대한 액세스를 제공합니다.
- **XML 컨텐츠 모델**은 XML 형식으로 저장된 내용에 대한 액세스를 제공합니다.
- **JSON 컨텐츠 모델**은 JSON 형식으로 저장된 내용에 대한 액세스를 제공합니다.
- **열 통계량 컨텐츠 모델**은 지정된 필드에 관한 요약 통계량에 대한 액세스를 제공합니다.
- **대응별 열 통계량 컨텐츠 모델**은 두 필드 사이의 요약 통계량 또는 두 개별 필드 사이의 값에 대한 액세스를 제공합니다.

테이블 컨텐츠 모델

테이블 컨텐츠 모델은 단순한 행 및 열 데이터 액세스를 위한 단순 모델을 제공합니다. 특정 열의 값은 모두가 동일한 유형의 저장 공간(예: 문자열 또는 정수)을 가져야 합니다.

API

표 25. API

리턴	메소드	설명
int	getRowCount()	이 테이블에 있는 행의 수를 리턴합니다.
int	getColumnCount()	이 테이블에 있는 열의 수를 리턴합니다.
String	getColumnName(int columnIndex)	지정된 열 지수에 있는 열의 이름을 리턴합니다. 열 지수는 0에서 시작합니다.
StorageType	getStorageType(int columnIndex)	지정된 지수에 있는 열의 저장 유형을 리턴합니다. 열 지수는 0에서 시작합니다.
오브젝트	getValueAt(int rowIndex, int columnIndex)	지정된 행 및 열 지수에 있는 값을 리턴합니다. 행 및 열 지수는 0에서 시작합니다.
void	reset()	이 콘텐츠 모델과 연관된 모든 내부 저장 공간을 비웁니다.

노드와 출력

이 테이블은 콘텐츠 모델의 이 유형을 포함하는 출력을 작성하는 노드를 나열합니다.

표 26. 노드와 출력

노드 이름	출력 이름	컨테이너 ID
table	table	"table"

예제 스크립트

```
stream = modeler.script.stream()
from modeler.api import StorageType

# Set up the variable file import node
varfilenode = stream.createAt("variablefile", "DRUG Data", 96, 96)
varfilenode.setPropertyValue("full_filename", "$CLEO_DEMOS/DRUG1n")

# Next create the aggregate node and connect it to the variable file node
aggregatenode = stream.createAt("aggregate", "Aggregate", 192, 96)
stream.link(varfilenode, aggregatenode)

# Configure the aggregate node
aggregatenode.setPropertyValue("keys", ["Drug"])
aggregatenode.setKeyedPropertyValue("aggregates", "Age", ["Min", "Max"])
aggregatenode.setKeyedPropertyValue("aggregates", "Na", ["Mean", "SDev"])

# Then create the table output node and connect it to the aggregate node
tablenode = stream.createAt("table", "Table", 288, 96)
stream.link(aggregatenode, tablenode)

# Execute the table node and capture the resulting table output object
results = []
tablenode.run(results)
tableoutput = results[0]
```

```
# Access the table output's content model
tablecontent = tableoutput.getContentModel("table")

# For each column, print column name, type and the first row
# of values from the table content
col = 0
while col < tablecontent.getColumnCount():
    print tablecontent.getColumnName(col), \
          tablecontent.getStorageType(col), \
          tablecontent.getValueAt(0, col)
    col = col + 1
```

스크립팅 디버그 탭에 있는 출력은 다음과 비슷합니다.

```
Age_Min Integer 15
Age_Max Integer 74
Na_Mean Real 0.730851098901
Na_SDev Real 0.116669731242
Drug String drugY
Record_Count Integer 91
```

XML 콘텐츠 모델

XML 콘텐츠 모델은 XML 기반 콘텐츠에 대한 액세스를 제공합니다.

XML 콘텐츠 모델은 XPath 표현식에 기반한 구성요소에 액세스하는 기능을 지원합니다. XPath 표현식은 호출자에게 필요한 요소 및 속성을 정의하는 문자열입니다. XML 콘텐츠 모델은 다양한 오브젝트 구성 및 일반적으로 XPath 지원에서 필요한 표현식 컴파일의 세부사항을 숨깁니다. 이것은 Python 스크립팅에서 호출하는 것을 더 쉽게 만듭니다.

XML 콘텐츠 모델은 XML 문서를 문자열로서 리턴하는 함수를 포함합니다. 이것은 Python 스크립트 사용자가 선호하는 Python 라이브러리를 사용하여 XML을 구문 분석할 수 있게 합니다.

API

표 27. API

리턴	메소드	설명
String	getXMLAsString()	XML을 문자열로서 리턴합니다.
number	getNumericValue(String xpath)	수치의 리턴 유형을 갖는 경로 평가의 결과를 리턴합니다(예를 들어, 경로 표현식과 매치하는 요소의 수를 셈).
boolean	getBooleanValue(String xpath)	지정된 경로 표현식 평가의 부울 결과를 리턴합니다.
String	getStringValue(String xpath, String attribute)	지정된 경로와 매치하는 속성 값 또는 XML 노드 값을 리턴합니다.
문자열 목록	getStringValues(String xpath, String attribute)	지정된 경로와 매치하는 모든 속성값 또는 XML 노드 값의 목록을 리턴합니다.

표 27. API (계속)

리턴	메소드	설명
문자열 목록의 목록	getValuesList(String xpath, <List of strings> attributes, boolean includeValue)	지정된 경로와 매치하는 모든 속성 값의 목록을 XML 노드 값(필요한 경우)과 함께 리턴합니다.
해시 테이블(key:string, value:list of string)	getValuesMap(String xpath, String keyAttribute, <List of strings> attributes, boolean includeValue)	키 속성이나 XML 노드 값을 키로, 지정된 속성 값의 목록을 테이블 값으로 사용하는 해시 테이블을 리턴합니다.
boolean	isNamespaceAware()	XML 구문 분석기가 네임스페이스를 인식해야 하는지 여부를 리턴합니다. 기본 값은 False입니다.
void	setNamespaceAware(boolean value)	XML 구문 분석기가 네임스페이스를 인식해야 하는지 여부를 설정합니다. 이것은 또한 reset()을 호출하여 후속 호출이 변경을 선택하도록 보장합니다.
void	reset()	이 콘텐츠 모델과 연관된 모든 내부 저장 공간(예: 캐싱된 DOM 오브젝트)을 비웁니다.

노드와 출력

이 테이블은 콘텐츠 모델의 이 유형을 포함하는 출력을 작성하는 노드를 나열합니다.

표 28. 노드와 출력

노드 이름	출력 이름	컨테이너 ID
대부분의 모델 작성기	대부분의 생성된 모델	"PMML"
"autodataprep"	n/a	"PMML"

예제 스크립트

컨텐츠에 액세스하기 위한 Python 스크립팅 코드는 다음과 유사할 수 있습니다.

```
results = []
modelbuilder.run(results)
modeloutput = results[0]
cm = modeloutput.getContentModel("PMML")
```

```
dataFieldNames = cm.getStringValues("/PMML/DataDictionary/DataField", "name")
predictedNames = cm.getStringValues("//MiningSchema/MiningField[@usageType='predicted']", "name")
```

JSON 콘텐츠 모델

JSON 콘텐츠 모델은 JSON 형식 콘텐츠 지원을 제공하는 데 사용됩니다. 이것은 호출자가 어떤 값에 액세스할지를 알고 있다는 가정 하에 값을 추출할 수 있도록 하는 기본 API를 제공합니다.

API

표 29. API

리턴	메소드	설명
String	getJSONAsString()	JSON 콘텐츠를 문자열로 리턴합니다.
오브젝트	getObjectAt(<List of object> path, JSONArtifact artifact) throws Exception	지정된 경로의 오브젝트를 리턴합니다. 제공되는 루트 아티팩트는 널일 수 있으며 그 경우 콘텐츠의 루트를 사용합니다. 리턴된 값은 리터럴 문자열, 정수, 실수나 부울 또는 JSON 아티팩트(JSON 오브젝트 또는 JSON 배열)일 수 있습니다.
해시 테이블(key:object, value:object)	getChildValuesAt(<List of object> path, JSONArtifact artifact) throws Exception	경로가 JSON 오브젝트로 이끄는 경우 지정된 경로의 하위 값, 그렇지 않으면 널을 리턴합니다. 테이블의 키는 문자열인 반면 연관된 값은 리터럴 문자열, 정수, 실수나 부울 또는 JSON 아티팩트(JSON 오브젝트 또는 JSON 배열)일 수 있습니다.
오브젝트 목록	getChildrenAt(<List of object> path path, JSONArtifact artifact) throws Exception	경로가 JSON 배열로 이끄는 경우 지정된 경로에 있는 오브젝트의 목록, 그렇지 않으면 널을 리턴합니다. 리턴된 값은 리터럴 문자열, 정수, 실수나 부울 또는 JSON 아티팩트(JSON 오브젝트 또는 JSON 배열)일 수 있습니다.
void	reset()	이 콘텐츠 모델과 연관된 모든 내부 저장 공간(예: 캐싱된 DOM 오브젝트)을 비웁니다.

예제 스크립트

JSON 형식을 기반으로 출력을 작성하는 출력 작성기 노드가 있는 경우, 서적 세트에 관한 정보에 액세스하는 데 다음을 사용할 수 있습니다.

```

results = []
outputbuilder.run(results)
output = results[0]
cm = output.getContentModel("jsonContent")

bookTitle = cm.getObjectAt(["books", "ISIN123456", "title"], None)

# Alternatively, get the book object and use it as the root
# for subsequent entries
book = cm.getObjectAt(["books", "ISIN123456"], None)
bookTitle = cm.getObjectAt(["title"], book)

# Get all child values for a specific book
bookInfo = cm.getChildValuesAt(["books", "ISIN123456"], None)

# Get the third book entry. Assumes the top-level "books" value
# contains a JSON array which can be indexed

```

```
bookInfo = cm.getObjectAt(["books", 2], None)

# Get a list of all child entries
allBooks = cm.getChildrenAt(["books"], None)
```

열 통계 콘텐츠 모델 및 대응별 통계 콘텐츠 모델

열 통계량 콘텐츠 모델은 각 필드에 대해 계산할 수 있는 통계(일변량 통계량)에 대한 액세스를 제공합니다. 대응별 통계량 콘텐츠 모델은 필드의 쌍 또는 필드의 값 사이에서 계산할 수 있는 통계량에 대한 액세스를 제공합니다.

가능한 통계량 측도는 다음과 같습니다.

- Count
- UniqueCount
- ValidCount
- Mean
- Sum
- Min
- Max
- Range
- Variance
- StandardDeviation
- StandardErrorOfMean
- Skewness
- SkewnessStandardError
- Kurtosis
- KurtosisStandardError
- Median
- Mode
- Pearson
- Covariance
- TTest
- FTest

일부 값은 단일 열 통계량에서만 적합한 반면 다른 값은 대응별 통계량에 대해서만 적합합니다.

이들을 생성하는 노드는 다음과 같습니다.

- **통계량 노드**는 열 통계량을 생성하며 상관관계 필드가 지정될 때 대응별 통계량을 생성할 수 있습니다.

- 데이터 검토 노드는 열을 생성하고 오버레이 필드가 지정될 때 대응별 통계량을 생성할 수 있습니다.
- 평균 노드는 필드 쌍을 비교할 때 또는 필드의 값을 다른 필드 요약값과 비교할 때 대응별 통계량을 생성합니다.

사용 가능한 콘텐츠 모델 및 통계량은 특정 노드의 기능 및 노드 내의 설정 모두에 의존합니다.

ColumnStatsContentModel API

표 30. ColumnStatsContentModel API.

리턴	메소드	설명
List<StatisticType>	getAvailableStatistics()	이 모델의 사용 가능 통계량을 리턴합니다. 모든 필드가 반드시 모든 통계량에 대한 값을 갖지는 않습니다.
List<String>	getAvailableColumns()	통계량이 계산되는 열 이름을 리턴합니다.
Number	getStatistic(String column, StatisticType statistic)	열과 연관된 통계 값을 리턴합니다.
void	reset()	이 콘텐츠 모델과 연관된 모든 내부 저장 공간을 비웁니다.

PairwiseStatsContentModel API

표 31. PairwiseStatsContentModel API.

리턴	메소드	설명
List<StatisticType>	getAvailableStatistics()	이 모델의 사용 가능 통계량을 리턴합니다. 모든 필드가 반드시 모든 통계량에 대한 값을 갖지는 않습니다.
List<String>	getAvailablePrimaryColumns()	통계량이 계산된 1차 열 이름을 리턴합니다.
List<Object>	getAvailablePrimaryValues()	통계량이 계산된 1차 열의 값을 리턴합니다.
List<String>	getAvailableSecondaryColumns()	통계량이 계산된 2차 열 이름을 리턴합니다.
Number	getStatistic(String primaryColumn, String secondaryColumn, StatisticType statistic)	열과 연관된 통계 값을 리턴합니다.
Number	getStatistic(String primaryColumn, Object primaryValue, String secondaryColumn, StatisticType statistic)	1차 열 값 및 2차 열과 연관된 통계 값을 리턴합니다.
void	reset()	이 콘텐츠 모델과 연관된 모든 내부 저장 공간을 비웁니다.

노드와 출력

이 테이블은 콘텐츠 모델의 이 유형을 포함하는 출력을 작성하는 노드를 나열합니다.

표 32. 노드와 출력.

노드 이름	출력 이름	컨테이너 ID	참고
"means" (평균 노드)	"means"	"columnStatistics"	
"means" (평균 노드)	"means"	"pairwiseStatistics"	
"dataaudit" (데이터 검토 노드)	"means"	"columnStatistics"	
"statistics" (통계량 노드)	"statistics"	"columnStatistics"	특정 필드가 검사될 때만 생성됩니다.
"statistics" (통계량 노드)	"statistics"	"pairwiseStatistics"	필드가 상관분석될 때만 생성됩니다.

예제 스크립트

```
from modeler.api import StatisticType
stream = modeler.script.stream()

# Set up the input data
varfile = stream.createAt("variablefile", "File", 96, 96)
varfile.setPropertyValue("full_filename", "$CLEO/DEMOS/DRUG1n")

# Now create the statistics node. This can produce both
# column statistics and pairwise statistics
statisticsnode = stream.createAt("statistics", "Stats", 192, 96)
statisticsnode.setPropertyValue("examine", ["Age", "Na", "K"])
statisticsnode.setPropertyValue("correlate", ["Age", "Na", "K"])
stream.link(varfile, statisticsnode)

results = []
statisticsnode.run(results)
statsoutput = results[0]
statscm = statsoutput.getContentModel("columnStatistics")
if (statscm != None):
    cols = statscm.getAvailableColumns()
    stats = statscm.getAvailableStatistics()
    print "Column stats:", cols[0], str(stats[0]), " = ", statscm.getStatistic(cols[0], stats[0])

statscm = statsoutput.getContentModel("pairwiseStatistics")
if (statscm != None):
    pcols = statscm.getAvailablePrimaryColumns()
    scols = statscm.getAvailableSecondaryColumns()
    stats = statscm.getAvailableStatistics()
    corr = statscm.getStatistic(pcols[0], scols[0], StatisticType.Pearson)
    print "Pairwise stats:", pcols[0], scols[0], " Pearson = ", corr
```

제 6 장 명령행 인수

소프트웨어 호출

운영 체제의 명령행을 사용하여 다음과 같이 IBM SPSS Modeler를 시작할 수 있습니다.

1. IBM SPSS Modeler가 설치된 컴퓨터에서 DOS 또는 명령 프롬프트 창을 여십시오.
2. 대화식 모드로 IBM SPSS Modeler 인터페이스를 시작하려면 modelerclient 명령 뒤에 필수 인수를 입력하십시오. 예를 들어, 다음과 같습니다.

```
modelerclient -stream report.str -execute
```

사용 가능한 인수(플래그)를 사용하면 필요에 따라 서버에 연결하고 스트림을 로드하며, 스크립트를 실행하거나 다른 모수를 지정할 수 있습니다.

명령행 인수 사용

명령행 인수(플래그라고도 부름)를 초기 modelerclient 명령에 추가하여 IBM SPSS Modeler의 호출을 변경할 수 있습니다.

여러 가지 유형의 명령행 인수를 사용할 수 있으며, 이 절의 뒤에서 설명합니다.

표 33. 명령행 인수의 유형.

인수 유형	설명되는 위치
시스템 인수	자세한 정보는 68 페이지의 『시스템 인수』의 내용을 참조하십시오.
모수 인수	자세한 정보는 69 페이지의 『모수 인수』의 내용을 참조하십시오.
서버 연결 인수	자세한 정보는 70 페이지의 『서버 연결 인수』의 내용을 참조하십시오.
IBM SPSS Collaboration and Deployment Services Repository 연결 인수	자세한 정보는 71 페이지의 『IBM SPSS Collaboration and Deployment Services Repository 연결 인수』의 내용을 참조하십시오.
IBM SPSS Analytic Server 연결 인수	자세한 정보는 72 페이지의 『IBM SPSS Analytic Server 연결 인수』의 내용을 참조하십시오.

예를 들어, -server, -stream, -execute 플래그를 사용하여 서버에 연결한 후 다음과 같이 스트림을 로드하고 실행할 수 있습니다.

```
modelerclient -server -hostname myserver -port 80 -username dminer  
-password 1234 -stream mystream.str -execute
```

로컬 클라이언트 연결에 대해 실행할 때는 서버 연결 인수가 필요 없음을 참고하십시오.

공백을 포함하는 모수값은 큰따옴표로 묶을 수 있습니다. 예:

```
modelerclient -stream mystream.str -Pusername="Joe User" -execute
```

-state 및 -script 플래그를 사용하여 IBM SPSS Modeler 상태 및 스크립트를 이 방식으로 각각 실행할 수도 있습니다.

참고: 명령에서 구조화 모수를 사용하는 경우 따옴표 앞에 백슬래시를 선행해야 합니다. 이것은 문자열 해석 중에 따옴표가 제거되지 않게 합니다.

명령행 인수 디버깅

명령행을 디버그하려면 `modelerclient` 명령을 사용하여 IBM SPSS Modeler를 원하는 인수와 함께 시작하십시오. 그러면 명령이 예상대로 실행하는지 확인할 수 있습니다. 또한 세션 모수 대화 상자(도구 메뉴, 세션 모수 설정)에서 명령행에서 전달되는 모든 모수의 값을 확인할 수도 있습니다.

시스템 인수

다음 테이블은 사용자 인터페이스의 명령행 호출에 사용할 수 있는 시스템 인수를 설명합니다.

표 34. 시스템 인수

인수	작동/설명
@ <commandFile>	파일 이름이 뒤에 오는 @ 문자는 명령 목록을 지정합니다. <code>modelerclient</code> 가 @로 시작하는 인수를 만나면 해당 파일의 명령이 명령행에 있는 것처럼 명령을 작동합니다. 자세한 정보는 72 페이지의 『다중 인수 결합』의 내용을 참조하십시오.
-directory <dir>	기본 작업 디렉토리를 설정합니다. 로컬 모드에서는 이 디렉토리를 데이터 및 출력 모두에 사용합니다. 예: <code>-directory c:/</code> 또는 <code>-directory c:\</code>
-server_directory <dir>	데이터를 위한 기본 서버 디렉토리를 설정합니다. <code>-directory</code> 플래그를 사용하여 지정되는 작업 디렉토리는 출력에 사용합니다.
-execute	시작한 후, 시작 시에 로드된 모든 스트림, 상태 또는 스크립트를 실행합니다. 스크립트가 스트림이나 상태에 추가로 로드되는 경우 스크립트만 실행됩니다.
-stream <stream>	시작 시에 지정된 스트림을 로드합니다. 다중 스트림을 지정할 수 있지만, 마지막으로 지정된 스트림이 현재 스트림으로 설정됩니다.
-script <script>	시작 시에, 지정된 독립형 스크립트를 로드합니다. 이것은 아래에서 설명하는 스트림이나 상태에 추가로 지정될 수 있지만 시작 시에 단 하나의 스크립트만 로드할 수 있습니다.
-model <model>	시작 시에, 지정된 생성된 모델(.gm 형식 파일)을 로드합니다.
-state <state>	시작 시에, 지정된 저장 상태를 로드합니다.
-project <project>	지정된 프로젝트를 로드합니다. 시작 시에 단 하나의 프로젝트만 로드할 수 있습니다.
-output <output>	시작 시에, 저장된 출력 오브젝트(.cou 형식 파일)를 로드합니다.
-help	명령행 인수의 목록을 표시합니다. 이 옵션이 지정되면 다른 모든 인수는 무시되고 도움말 화면이 표시됩니다.
-P <name>=<value>	시작 모수를 설정하는 데 사용합니다. 노트 특성(슬롯 모수)을 설정하는 데 사용할 수도 있습니다.

참고: 기본 디렉토리도 사용자 인터페이스에서 설정할 수 있습니다. 옵션에 액세스하려면, 파일 메뉴에서 **작업 디렉토리 설정** 또는 **서버 디렉토리 설정**을 선택하십시오.

다중 파일 로드

명령행에서 로드되는 각 오브젝트에 대한 관련 인수를 반복하여 시작 시에 다중 스트림, 상태, 출력을 로드할 수 있습니다. 예를 들어, report.str 및 train.str이라는 두 스트림을 로드하고 실행하려면 다음 명령을 사용합니다.

```
modelerclient -stream report.str -stream train.str -execute
```

IBM SPSS Collaboration and Deployment Services Repository에서 오브젝트 로드

파일 또는 IBM SPSS Collaboration and Deployment Services Repository(사용권이 있는 경우)로부터 특정 오브젝트를 로드할 수 있기 때문에, 파일 이름 접두문자 spsscr: 및 선택적으로 file:(디스크의 오브젝트 경우)은 IBM SPSS Modeler에 오브젝트를 찾을 위치를 지시합니다. 접두문자는 다음 플래그와 함께 작업합니다.

- -stream
- -script
- -output
- -model
- -project

접두문자를 사용하여 오브젝트의 위치를 지정하는 URI를 작성합니다(예: -stream "spsscr:///folder_1/scoring_stream.str"). spsscr: 접두문자가 있으면 IBM SPSS Collaboration and Deployment Services Repository에 대한 유효한 연결을 동일한 명령에서 지정해야 합니다. 따라서 예를 들어 전체 명령은 다음과 유사합니다.

```
modelerclient -spsscr_hostname myhost -spsscr_port 8080  
-spsscr_username myusername -spsscr_password mypassword  
-stream "spsscr:///folder_1/scoring_stream.str" -execute
```

명령행에서는 반드시 URI를 사용해야 함을 참고하십시오. 더 간단한 REPOSITORY_PATH는 지원되지 않습니다. (스크립트 내에서만 작동합니다.) IBM SPSS Collaboration and Deployment Services Repository에서 오브젝트에 대한 URI의 자세한 내용은 54 페이지의 『IBM SPSS Collaboration and Deployment Services Repository의 오브젝트 액세스』 주제를 참조하십시오.

모수 인수

IBM SPSS Modeler의 명령행 실행 중에 모수를 플래그로 사용할 수 있습니다. 명령행 인수에서 -P 플래그는 -P <name>=<value> 양식의 모수를 표시하는 데 사용합니다.

모수는 다음 중 하나일 수 있습니다.

- 단순 모수(또는 CLEM 표현식에서 직접 사용하는 모수).
- 슬롯 모수(노드 특성이라고도 부름). 이들 모수는 스트림에서 노드의 설정을 수정하는 데 사용됩니다. 자세한 정보는 78 페이지의 『노드 특성 개요』 주제를 참조하십시오.

- IBM SPSS Modeler의 호출을 변경하는 데 사용하는 명령행 모수.

예를 들어, 다음과 같이 데이터 소스 사용자 이름과 비밀번호를 명령행 플래그로서 제공할 수 있습니다.

```
modelerclient -stream response.str -P:databasenode.datasources="{\"ORA 10gR2\", user1, mypsw, true}"
```

형식은 databasenode 노드 특성의 datasource 모수의 형식과 같습니다. 자세한 정보는 90 페이지의 『databasenode 특성』을 참조하십시오.

참고: 노드가 이름 지정되면 노드 이름을 큰따옴표로 묶고 따옴표를 백슬래시로 이스케이프해야 합니다. 예를 들어, 이전 예의 데이터 소스 노드가 *Source_ABC*라는 이름을 갖는 경우 입력은 다음과 같습니다.

```
modelerclient -stream response.str -P:databasenode.\"Source_ABC\".datasources="{\"ORA 10gR2\", user1, mypsw, true}"
```

다음 TM1 데이터 소스 예에서와 같이, 구조화 모수를 식별하는 인용 부호의 앞에 백슬래시도 필요합니다.

```
climb -server -hostname 9.115.21.169 -port 28053 -username administrator
      -execute -stream C:\Share\TM1_Script.str -P:tm1import.pm_host="http://9.115.21.163:9510/pmhub/pm"
      -P:tm1import.tm1_connection="{\"SDData\", \"\", \"admin\", \"apple\"}"
      -P:tm1import.selected_view="{\"SalesPriorCube\", \"salesmargin%\"}"
```

서버 연결 인수

-server 플래그는 IBM SPSS Modeler에게 공용 서버에 연결해야 함을 지시하고, -hostname, -use_ssl, -port, -username, -password, -domain 플래그는 IBM SPSS Modeler에게 공용 서버에 연결하는 방법을 지시하는 데 사용됩니다. -server 인수가 지정되지 않으면 기본 또는 로컬 서버를 사용합니다.

예

공용 서버에 연결하려면 다음을 수행하십시오.

```
modelerclient -server -hostname myserver -port 80 -username dminer
      -password 1234 -stream mystream.str -execute
```

서버 군집에 연결하려면 다음을 수행하십시오.

```
modelerclient -server -cluster "QA Machines" \
      -spsscr_hostname pes_host -spsscr_port 8080 \
      -spsscr_username asmith -spsscr_epassword xyz
```

서버 군집에 연결하려면 IBM SPSS Collaboration and Deployment Services를 통한 프로세스 조정자가 필요하므로, -cluster 인수가 리포지토리 연결 옵션(spsscr_*)과 조합으로 사용해야 합니다. 자세한 정보는 71 페이지의 『IBM SPSS Collaboration and Deployment Services Repository 연결 인수』의 내용을 참조하십시오.

표 35. 서버 연결 인수.

인수	작동/설명
-server	IBM SPSS Modeler를 서버 모드에서 실행하고, -hostname, -port, -username, -password, -domain 플래그를 사용하여 공용 서버에 연결합니다.
-hostname <name>	서버 머신의 호스트 이름입니다. 서버 모드에서만 사용 가능합니다.
-use_ssl	연결이 SSL(Secure Socket Layer)을 사용해야 함을 지정합니다. 이 플래그는 선택사항입니다. 기본 설정은 SSL을 사용하지 않습니다.
-port <number>	지정된 서버의 포트 번호입니다. 서버 모드에서만 사용 가능합니다.
-cluster <name>	이름 지정된 서버가 아니라 서버 군집에 대한 연결을 지정합니다. 이 인수는 hostname, port, use_ssl 인수의 대안입니다. 이름은 군집 이름 또는 IBM SPSS Collaboration and Deployment Services Repository에서 군집을 식별하는 고유 URI입니다. 서버 군집은 IBM SPSS Collaboration and Deployment Services를 통한 프로세스 조정자에 의해 관리됩니다. 자세한 정보는 『IBM SPSS Collaboration and Deployment Services Repository 연결 인수』의 내용을 참조하십시오.
-username <name>	서버에 로그인하는 사용자 이름입니다. 서버 모드에서만 사용 가능합니다.
-password <password>	서버에 로그인하는 비밀번호입니다. 서버 모드에서만 사용 가능합니다. 참고: -password 인수를 사용하지 않으면 사용자에게 비밀번호를 프롬프트합니다.
-epassword <encodedpasswordstring>	서버에 로그인하는 인코딩된 비밀번호입니다. 서버 모드에서만 사용 가능합니다. 참고: 인코딩된 비밀번호는 IBM SPSS Modeler 애플리케이션의 도구 메뉴에서 생성될 수 있습니다.
-domain <name>	서버에 로그인하는 데 사용하는 도메인입니다. 서버 모드에서만 사용 가능합니다.
-P <name>=<value>	시작 모수를 설정하는 데 사용합니다. 노드 특성(슬롯 모수)을 설정하는 데 사용할 수도 있습니다.

IBM SPSS Collaboration and Deployment Services Repository 연결 인수

명령행을 통해 IBM SPSS Collaboration and Deployment Services에서 오브젝트를 저장하거나 검색하려는 경우 IBM SPSS Collaboration and Deployment Services Repository에 대한 유효한 연결을 지정해야 합니다. 예를 들어,

```
modelerclient -spsscr_hostname myhost -spsscr_port 8080
-spsscr_username myusername -spsscr_password mypassword
-stream "spsscr:///folder_1/scoring_stream.str" -execute
```

다음 테이블은 연결을 설정하는 데 사용할 수 있는 인수를 나열합니다.

표 36. IBM SPSS Collaboration and Deployment Services Repository 연결 인수

인수	작동/설명
-spsscr_hostname <호스트 이름 또는 IP 주소>	IBM SPSS Collaboration and Deployment Services Repository가 설치된 서버의 호스트 이름 또는 IP 주소입니다.
-spsscr_port <number>	IBM SPSS Collaboration and Deployment Services Repository가 연결을 승인하는 포트 번호입니다(일반적으로 8080이 기본값임).
-spsscr_use_ssl	연결이 SSL(Secure Socket Layer)을 사용해야 함을 지정합니다. 이 플래그는 선택사항입니다. 기본 설정은 SSL을 사용하지 않습니다.
-spsscr_username <name>	IBM SPSS Collaboration and Deployment Services Repository에 로그인하는 사용자 이름입니다.

표 36. IBM SPSS Collaboration and Deployment Services Repository 연결 인수 (계속)

인수	작동/설명
-spsscr_password <password>	IBM SPSS Collaboration and Deployment Services Repository에 로그인하는 비밀번호입니다.
-spsscr_epassword <encoded password>	IBM SPSS Collaboration and Deployment Services Repository에 로그인하는 인코딩된 비밀번호입니다.
-spsscr_providername <name>	IBM SPSS Collaboration and Deployment Services Repository(Active Directory 또는 LDAP)에 로그인하는 데 인증 제공자가 사용됩니다. 이는 원본(로컬 리포지토리) 제공자를 사용하는 경우 필요하지 않습니다.

IBM SPSS Analytic Server 연결 인수

명령행을 통해 IBM SPSS Analytic Server에서 오브젝트를 저장 또는 검색하려면 IBM SPSS Analytic Server에 대한 유효한 연결을 지정해야 합니다.

참고: Analytic Server의 기본 위치는 SPSS Modeler Server에서 가져옵니다. 사용자가 **도구 > Analytic Server 연결**을 통해 고유 Analytic Server 연결도 정의할 수 있습니다.

다음 테이블은 연결을 설정하는 데 사용할 수 있는 인수를 나열합니다.

표 37. IBM SPSS Analytic Server 연결 인수

인수	작동/설명
-analytic_server_username	IBM SPSS Analytic Server에 로그인하는 사용자 이름입니다.
-analytic_server_password	IBM SPSS Analytic Server에 로그인하는 비밀번호입니다.
-analytic_server_epassword	IBM SPSS Analytic Server에 로그인하기 위한 인코딩된 비밀번호입니다.
-analytic_server_credential	IBM SPSS Analytic Server에 로그인하는 데 사용하는 신임 정보입니다.

다중 인수 결합

@ 기호 및 그 뒤를 따르는 파일 이름을 사용하여 호출 시에 지정되는 단일 명령 파일에 다중 인수를 결합할 수 있습니다. 이것은 명령행 호출을 단축하고 명령 길이에 대한 모든 운영 체제 한계를 극복할 수 있습니다. 예를 들어 다음 시작 명령은 <commandFileName>으로 참조되는 파일에서 지정되는 인수를 사용합니다.

```
modelerclient @<commandFileName>
```

공백이 필수인 경우 파일 이름 및 명령 파일에 대한 경로를 따옴표로 묶으십시오.

```
modelerclient @ "C:\Program Files\IBM\SPSS\Modeler\mn\scripts\my_command_file.txt"
```

명령 파일은 해당 하나의 인수를 갖고 시작 시에 개별적으로 이전에 지정된 모든 인수를 포함할 수 있습니다. 예를 들어,

```
-stream report.str
-Porder.full_filename=APR_orders.dat
-Preport.filename=APR_report.txt
-execute
```

명령 파일을 쓰고 참조할 때 다음 제약조건을 따라야 합니다.

- 행당 하나의 명령만 사용하십시오.
- @CommandFile 인수를 명령 파일 안에 임베드하지 마십시오.

제 7 장 특성 참조

특성 참조 개요

노드, 스트림, 프로젝트 및 슈퍼노드에 대한 수많은 다른 특성을 지정할 수 있습니다. 이름, 주석(Annotation), 도구팁 같은 일부 특성은 모든 노드에 공통적인 반면, 다른 특성은 특정 유형의 노드에만 관계합니다. 기타 특성은 캐싱 또는 슈퍼노드 작동 같은 상위 수준 스트림 작업을 참조합니다. 특성은 표준 사용자 인터페이스를 통해 액세스할 수 있으며(예를 들어, 노드에 대한 옵션을 편집하기 위해 대화 상자를 열 때) 다른 수많은 방법으로 사용할 수도 있습니다.

- 이 절에서 설명하는 대로, 스크립트를 통해 특성을 수정할 수 있습니다. 자세한 정보는 『특성 구문』을 참조하십시오.
- 노드 특성은 슈퍼노드 모수에서 사용할 수 있습니다.
- 노드 특성은 IBM SPSS Modeler를 시작할 때 명령행 옵션의 일부로(-P 플래그를 사용하여) 사용할 수도 있습니다.

IBM SPSS Modeler 내에서 스크립팅의 컨텍스트에서, 노드 및 스트림 특성을 종종 슬롯 모수라고 합니다. 이 안내서에서는 노드 또는 스트림 특성으로 부릅니다.

스크립팅 언어에 대한 자세한 정보는 스크립팅 언어를 참조하십시오.

특성 구문

특성은 다음 구문을 사용하여 설정할 수 있습니다.

```
OBJECT.setPropertyValue(PROPERTY, VALUE)
```

또는:

```
OBJECT.setKeyedPropertyValue(PROPERTY, KEY, VALUE)
```

특성의 값은 다음 구문을 사용하여 검색할 수 있습니다.

```
VARIABLE = OBJECT.getPropertyValue(PROPERTY)
```

또는:

```
VARIABLE = OBJECT.getKeyedPropertyValue(PROPERTY, KEY)
```

여기서 OBJECT는 노드 또는 출력이고, PROPERTY는 표현식이 참조하는 노드 특성의 이름이며, KEY는 키가 있는 특성의 키 값입니다. 예를 들어, 다음 구문은 필터 노드를 찾은 후 모든 필드를 포함하고 다운 스트림 데이터에서 Age 필드를 필터링하도록 기본값을 설정합니다.

```
filternode = modeler.script.stream().findByType("filter", None)
filternode.setPropertyValue("default_include", True)
filternode.setKeyedPropertyValue("include", "Age", False)
```

IBM SPSS Modeler에서 사용하는 모든 노드는 스트림 findByType(TYPE, LABEL) 함수를 사용하여 찾을 수 있습니다. TYPE 또는 LABEL의 하나 이상을 지정해야 합니다.

구조화 특성

스크립팅이 구문 분석할 때 명확성을 향상시키기 위해 구조화 특성을 사용하는 두 가지 방법이 있습니다.

- 유형, 필터 또는 균형 노드 같은 복합 노드에 대한 특성의 이름에 구조를 부여하기 위해.
- 한 번에 다중 특성을 지정하기 위한 형식을 제공하기 위해.

복합 인터페이스에 대한 구조

테이블 및 기타 복합 인터페이스를 갖는 노드(예: 유형, 필터, 균형 노드)에 스크립트는 올바르게 구문 분석하기 위해 특정한 구조를 따라야 합니다. 이들 특성은 단일 식별자에 대한 이름보다 복잡한 이름이 필요한데, 이 이름을 키라고 부릅니다. 예를 들어, 필터 노드 내에서 각 사용 가능 필드(업스트림 쪽에 있는)가 켜지거나 꺼집니다. 이 정보를 참조하기 위해서 필터 노드는 필드당 정보의 한 항목(각 필드가 true 또는 false인지 여부)을 저장합니다. 이 특성은 True 또는 False 값을 가질 수(또는 주어질 수) 있습니다. mynode라는 필터 노드가(업스트림 쪽에) Age라는 필드를 갖는다고 가정하십시오. 이것을 끄려면 다음과 같이 Age 키를 사용하여 include 특성을 False 값으로 설정하십시오.

```
mynode.setKeyedPropertyValue("include", "Age", False)
```

다중 특성을 설정하는 구조

많은 노드의 경우 한 번에 둘 이상의 노드 또는 스트림 특성을 지정할 수 있습니다. 이것을 **다중 세트 명령** 또는 **변수군 블록**이라고 합니다.

일부 경우에는 구조화된 특성이 매우 복잡할 수 있습니다. 예는 다음과 같습니다.

```
sortnode.setPropertyValue("keys", [["K", "Descending"], ["Age", "Ascending"], ["Na", "Descending"]])
```

구조화된 특성이 갖는 또 다른 장점은 노드가 안정되기 전에 노드에 여러 특성을 설정하는 기능입니다. 기본적으로, 다중 세트는 개별 특성 설정을 기반으로 어떤 조치를 취하기 전에 모든 특성을 블록으로 설정합니다. 예를 들어, 고정 파일 노드를 정의할 때 2 단계를 사용하여 필드 특성을 설정하면 오류가 발생합니다. 두 설정이 모두 유효할 때까지 노드가 일관되지 않기 때문입니다. 특성을 다중 세트로서 정의하면 데이터 모델을 업데이트하기 전에 두 특성을 모두 설정하여 이 문제점이 발생하지 않습니다.

약어

노드 특성의 구문 전체에서 표준 약어를 사용합니다. 약어를 학습하면 스크립트를 구성할 때 도움이 됩니다.

표 38. 구문 전체에서 사용하는 표준 약어

약어	의미
abs	절대값

표 38. 구문 전체에서 사용하는 표준 약어 (계속)

약어	의미
len	길이
min	최소
max	최대
correl	상관관계
covar	공분산
num	번호 또는 수치
pct	퍼센트 또는 백분율
transp	투명도
xval	교차 검증
var	분산 또는 변수(소스 노드에서)

노드 및 스트림 특성 예

노드 및 스트림 특성은 IBM SPSS Modeler에서 다양한 방법으로 사용할 수 있습니다. 다중 스트림이나 작업을 자동화하기 위해 사용하는 독립형 스크립트 또는 단일 스트림 내에서 프로세스를 자동화하기 위해 사용하는 스트림 스크립트 중 하나인 스크립트의 일부로서 가장 공통적으로 사용합니다. 또한 수퍼노드 내에서 노드 특성을 사용하여 노드 모수를 지정할 수도 있습니다. 가장 기본적인 수준에서는 특성을 IBM SPSS Modeler 시작을 위한 명령행 옵션으로 사용할 수도 있습니다. -p 인수를 명령행 호출의 일부로 사용하여, 스트림 특성을 사용하여 스트림에서 설정을 변경할 수 있습니다.

표 39. 노드 및 스트림 특성 예

특성	의미
s.max_size	s라는 노드의 max_size 특성을 참조합니다.
s:samplenode.max_size	s라는 노드의 max_size 특성을 참조하는데, 표본 노드여야 합니다.
:samplenode.max_size	현재 스트림에 있는 표본 노드의 max_size 특성을 참조합니다(표본 노드가 하나만 있어야 함).
s:sample.max_size	s라는 노드의 max_size 특성을 참조하는데, 표본 노드여야 합니다.
t.direction.Age	유형 노드 t에서 Age 필드의 역할을 참조합니다.
:.max_size	*** 적합하지 않음 *** 노드 이름이나 노드 유형 중 하나를 지정해야 합니다.

s:sample.max_size 예는 노드 유형을 완전히 입력할 필요가 없음을 설명합니다.

t.direction.Age는 일부 슬롯 이름이 자체적으로 구조화될 수 있음을 설명합니다. 그 경우 노드의 속성은 개별 값을 갖는 단순한 개별 슬롯보다 복잡합니다. 그런 슬롯을 구조화 또는 복합 특성이라고 합니다.

노드 특성 개요

노드의 각 유형은 고유한 합법적 특성 세트를 갖고 있으며, 각 특성은 하나의 유형을 갖습니다. 이 유형은 특성의 케이스 설정이 올바른 유형으로 강제되는 일반 유형(숫자, 플래그 또는 문자열) 일 수 있습니다. 이들을 강제할 수 없는 경우 오류가 발생합니다. 또는 특성 참조가 Discard, PairAndDiscard, IncludeAsText 같은 합법적 값의 범위를 지정할 수 있는데, 이 경우에 다른 값을 사용하면 오류가 발생합니다. 플래그 특성은 true 및 false의 값을 사용하여 읽거나 설정해야 합니다. (Off, OFF, off, No, NO, no, n, N, f, F, false, False, FALSE 또는 0를 포함한 변형도 값을 설정할 때 인식되지만, 어떤 경우에는 특성 값을 읽을 때 오류를 유발할 수 있습니다. 기타 모든 값은 true로 간주됩니다. true와 false를 일관성있게 사용하는 것이 혼동을 피할 수 있습니다.) 이 안내서의 참조 표에서, 구조화된 특성은 특성 설정 열에서와 같이 표시되며, 사용 형식이 제공됩니다.

공통 노드 특성

많은 특성이 IBM SPSS Modeler에 있는 모든 노드(수퍼노드 포함)에 공통적입니다.

표 40. 공통 노드 특성.

특성 이름	데이터 유형	특성 설명
use_custom_name	<i>flag</i>	
name	<i>string</i>	캔버스의 노드에 대한 이름(자동 또는 사용자 정의)을 읽는 읽기 전용 특성입니다.
custom_name	<i>string</i>	노드의 사용자 정의 이름을 지정합니다.
tooltip	<i>string</i>	
annotation	<i>string</i>	
keywords	<i>string</i>	오브젝트와 연관된 키워드의 목록을 지정하는 구조화된 슬롯(예: ["Keyword1" "Keyword2"]).
cache_enabled	<i>flag</i>	
node_type	source_supernode process_supernode terminal_supernode 스크립팅을 위해 지정되는 모든 스크립팅	유형별로 노드를 참조하는 데 사용하는 읽기 전용 특성입니다. 예를 들어 real_income 같이 이름만으로 노드를 참조하는 대신, userinputnode 또는 filternode 같은 유형도 지정할 수 있습니다.

수퍼노드 특정 특성은 다른 모든 노드와 같이 별도로 논의됩니다. 자세한 정보는 359 페이지의 제 20 장 『수퍼노드 특성』의 내용을 참조하십시오.

제 8 장 스트림 특성

스크립팅으로 다양한 스트림 특성을 제어할 수 있습니다. 스트림 특성을 참조하려면 스크립트를 사용할 실행 방법을 설정해야 합니다.

```
stream = modeler.script.stream()
stream.setPropertyValue("execute_method", "Script")
```

예제

노드 특성은 현재 스트림의 노드를 참조하는 데 사용합니다. 다음 스트림 스크립트가 예를 제공합니다.

```
stream = modeler.script.stream()
annotation = stream.getPropertyValue("annotation")

annotation = annotation + "\n\nThis stream is called \"" + stream.getLabel() + "\" and
contains the following nodes:\n"

for node in stream.iterator():
    annotation = annotation + "\n" + node.getTypeName() + " node called \"" + node.getLabel()
    + "\""

stream.setPropertyValue("annotation", annotation)
```

위의 예는 노드 특성을 사용하여 스트림에 있는 모든 노드의 목록을 작성하고 해당 목록을 스트림 주석에 기록합니다. 생성된 주석은 다음과 비슷합니다.

This stream is called "druglearn" and contains the following nodes:

```
type node called "Define Types"
derive node called "Na_to_K"
variablefile node called "DRUG1n"
neuralnetwork node called "Drug"
c50 node called "Drug"
filter node called "Discard Fields"
```

스트림 특성은 다음 표에서 설명됩니다.

표 41. 스트림 특성.

특성 이름	데이터 유형	특성 설명
execute_method	Normal Script	

표 41. 스트림 특성 (계속).

특성 이름	데이터 유형	특성 설명
date_format	"DDMMYY" "MMDDYY" "YYMMDD" "YYYYMMDD" "YYYYDDD" DAY MONTH "DD-MM-YY" "DD-MM-YYYY" "MM-DD-YY" "MM-DD-YYYY" "DD-MON-YY" "DD-MON-YYYY" "YYYY-MM-DD" "DD.MM.YY" "DD.MM.YYYY" "MM.DD.YYYY" "DD.MON.YY" "DD.MON.YYYY" "DD/MM/YY" "DD/MM/YYYY" "MM/DD/YY" "MM/DD/YYYY" "DD/MON/YY" "DD/MON/YYYY" MON YYYY q Q YYYY ww WK YYYY	
date_baseline	<i>number</i>	
date_2digit_baseline	<i>number</i>	
time_format	"HHMMSS" "HHMM" "MMSS" "HH:MM:SS" "HH:MM" "MM:SS" "(H)H:(M)M:(S)S" "(H)H:(M)M" "(M)M:(S)S" "HH.MM.SS" "HH.MM" "MM.SS" "(H)H.(M)M.(S)S" "(H)H.(M)M" "(M)M.(S)S"	
time_rollover	<i>flag</i>	
import_datetime_as_string	<i>flag</i>	
decimal_places	<i>number</i>	
decimal_symbol	Default Period Comma	
angles_in_radians	<i>flag</i>	
use_max_set_size	<i>flag</i>	
max_set_size	<i>number</i>	

표 41. 스트림 특성 (계속).

특성 이름	데이터 유형	특성 설명
ruleset_evaluation	Voting FirstHit	
refresh_source_nodes	flag	스트림 실행 시 소스 노드를 자동으로 새로 고치기 위해 사용됩니다.
script	string	
annotation	string	
name	string	참고: 이 특성은 읽기 전용입니다. 스트림의 이름을 변경하려는 경우 다른 이름으로 저장해야 합니다.
parameters		독립형 스크립트 내에서 스트림 모수를 업데이트하려면 이 특성을 사용하십시오.
nodes		아래의 상세 정보를 참조하십시오.
encoding	SystemDefault "UTF-8"	
stream_rewriting	boolean	
stream_rewriting_maximise_sql	boolean	
stream_rewriting_optimise_clem_실행	boolean	
stream_rewriting_optimise_syntax_실행	boolean	
enable_parallelism	boolean	
sql_generation	boolean	
database_caching	boolean	
sql_logging	boolean	
sql_generation_logging	boolean	
sql_log_native	boolean	
sql_log_prettyprint	boolean	
record_count_suppress_input	boolean	
record_count_feedback_interval	integer	
use_stream_auto_create_node_settings	boolean	True인 경우 스트림 특정 설정을 사용하고, 그렇지 않으면 사용자 기본 설정을 사용합니다.
create_model_applier_for_new_모델	boolean	True인 경우 모델 작성기가 새 모델을 작성하며, 모델이 활성 업데이트 링크를 갖지 않고, 새 모델 적용자가 추가됩니다. 참고: IBM SPSS Modeler Batch 버전 15를 사용 중인 경우 스크립트 안에 모델 적용자를 명시적으로 추가해야 합니다.
create_model_applier_update_links	createEnabled createDisabled doNotCreate	모델 적용자 노드가 자동으로 추가될 때 작성되는 링크의 유형을 정의합니다.
create_source_node_from_builders	boolean	True인 경우 소스 작성기가 새 소스 출력을 작성하며, 활성 업데이트 링크를 갖지 않고, 새 소스 노드가 추가됩니다.

표 41. 스트림 특성 (계속).

특성 이름	데이터 유형	특성 설명
create_source_node_update_links	createEnabled createDisabled doNotCreate	소스 노드가 자동으로 추가될 때 작성되는 링크의 유형을 정의합니다.
has_coordinate_system	boolean	True인 경우 좌표계를 전체 시스템에 적용합니다.
coordinate_system	string	선택된 투영된 좌표계의 이름입니다.
deployment_area	ModelRefresh Scoring None	스트림을 배포할 방법을 선택하십시오. 이 값이 None으로 설정되면 기타 배포 항목이 사용되지 않습니다.
scoring_terminal_node_id	string	스트림에서 스코어링 분기를 선택하십시오. 스트림의 모든 터미널 노드가 될 수 있습니다.
scoring_node_id	string	스코어링 분기에서 너깃을 선택하십시오.
model_build_node_id	string	스트림에서 모델링 노드를 선택하십시오.

제 9 장 소스 노드 특성

소스 노드 공통 특성

모든 소스 노드에 공통적인 특성이 아래에 나열되는데, 뒤따르는 주제에 특정 노드에 대한 정보가 함께 표시됩니다.

예 1

```
varfilenode = modeler.script.stream().create("variablefile", "Var. File")
varfilenode.setPropertyValue("full_filename", "$CLEO_DEMOS/DRUG1n")
varfilenode.setKeyedPropertyValue("check", "Age", "None")
varfilenode.setKeyedPropertyValue("values", "Age", [1, 100])
varfilenode.setKeyedPropertyValue("type", "Age", "Range")
varfilenode.setKeyedPropertyValue("direction", "Age", "Input")
```

예 2

이 스크립트는 지정된 데이터 파일이 다중 행 문자열을 나타내는 Region이라는 필드를 포함한다고 가정합니다.

```
from modeler.api import StorageType
from modeler.api import MeasureType

# Create a Variable File node that reads the data set containing
# the "Region" field
varfilenode = modeler.script.stream().create("variablefile", "My Geo Data")
varfilenode.setPropertyValue("full_filename", "C:/mydata/mygeodata.csv")
varfilenode.setPropertyValue("treat_square_brackets_as_lists", True)

# Override the storage type to be a list...
varfilenode.setKeyedPropertyValue("custom_storage_type", "Region", StorageType.LIST)
# ...and specify the type of values in the list and the list depth
varfilenode.setKeyedPropertyValue("custom_list_storage_type", "Region", StorageType.INTEGER)
varfilenode.setKeyedPropertyValue("custom_list_depth", "Region", 2)

# Now change the measurement to identify the field as a geospatial value...
varfilenode.setKeyedPropertyValue("measure_type", "Region", MeasureType.GEOSPATIAL)
# ...and finally specify the necessary information about the specific
# type of geospatial object
varfilenode.setKeyedPropertyValue("geo_type", "Region", "MultiLineString")
varfilenode.setKeyedPropertyValue("geo_coordinates", "Region", "2D")
varfilenode.setKeyedPropertyValue("has_coordinate_system", "Region", True)
varfilenode.setKeyedPropertyValue("coordinate_system", "Region",
    "ETRS_1989_EPSG_Arctic_zone_5-47")
```

표 42. 소스 노드 공통 특성.

특성 이름	데이터 유형	특성 설명
direction	Input Target Both None Partition Split Frequency RecordID	필드 역할에 대한 키가 있는 특성입니다. 사용 형식: NODE.direction.FIELDNAME 참고: 값 In 및 Out은 이제 더 이상 사용하지 않습니다. 이들에 대한 지원은 차후 릴리스에서 철회될 수 있습니다.
type	Range Flag Set Typeless Discrete Ordered Set Default	필드의 유형입니다. 이 특성을 <i>Default</i> 로 설정하면 모든 values 특성 설정을 지우며, value_mode가 <i>Specify</i> 로 설정되면 <i>Read</i> 로 재설정됩니다. value_mode가 이미 <i>Pass</i> 또는 <i>Read</i> 로 설정된 경우, type 설정에 의해 영향을 받지 않습니다. 사용 형식: NODE.type.FIELDNAME
storage	Unknown String Integer Real Time Date Timestamp	필드 저장 유형에 대한 읽기 전용 키가 있는 특성입니다. 사용 형식: NODE.storage.FIELDNAME
check	None Nullify Coerce Discard Warn Abort	필드 유형 및 범위 검사에 대한 키가 있는 특성입니다. 사용 형식: NODE.check.FIELDNAME
values	[value value]	연속형(범위) 필드의 경우, 첫 번째 값은 최소값이고 마지막 값은 최대값입니다. 명목형(변수군) 필드의 경우 모든 값을 지정하십시오. 플래그 필드의 경우 첫 번째 값은 <i>false</i> 를 나타내고, 마지막 값은 <i>true</i> 를 나타냅니다. 이 특성을 설정하면 자동으로 value_mode 특성을 <i>Specify</i> 로 설정합니다. 저장 공간은 목록의 첫 번째 값을 기반으로 판별됩니다. 예를 들어, 첫 번째 값이 <i>string</i> 인 경우 저장 공간이 문자열로 설정됩니다. 사용 형식: NODE.values.FIELDNAME
value_mode	Read Pass Read+ Current Specify	다음 데이터 전달에서 필드에 대한 값이 설정되는 방법을 판별합니다. 사용 형식: NODE.value_mode.FIELDNAME 이 특성을 <i>Specify</i> 로 직접 설정할 수 없음을 주의하십시오. 특정 값을 사용하려면 values 특성을 설정하십시오.

표 42. 소스 노드 공통 특성 (계속).

특성 이름	데이터 유형	특성 설명
default_value_mode	Read Pass	모든 필드에 대한 값 설정을 위한 기본 방법을 지정합니다. 사용 형식: NODE.default_value_mode 이 설정은 value_mode 특성을 사용하여 특정 필드에 대해 대체할 수 있습니다.
extend_values	flag	value_mode가 Read로 설정될 때 적용됩니다. 필드에 대한 임의의 기존 값에 새로 읽은 값을 추가하려면 T로 설정하십시오. 새로 읽은 값 대신 기존 값을 삭제하려면 F를 설정하십시오. 사용 형식: NODE.extend_values.FIELDNAME
value_labels	string	값 레이블을 지정하는 데 사용됩니다. 값이 먼저 지정되어야 함을 참고하십시오.
enable_missing	flag	T로 설정되면 필드에 대한 결측값 추적을 활성화합니다. 사용 형식: NODE.enable_missing.FIELDNAME
missing_values	[value value ...]	결측값을 표시하는 데이터 값을 지정합니다. 사용 형식: NODE.missing_values.FIELDNAME
range_missing	flag	이 특성이 T로 설정되면 결측값(공백) 범위가 필드에 대해 정의되는지 여부를 지정합니다. 사용 형식: NODE.range_missing.FIELDNAME
missing_lower	string	range_missing이 true일 때, 결측값 범위의 하한을 지정합니다. 사용 형식: NODE.missing_lower.FIELDNAME
missing_upper	string	range_missing이 true일 때, 결측값 범위의 상한을 지정합니다. 사용 형식: NODE.missing_upper.FIELDNAME
null_missing	flag	이 특성이 T로 설정될 때, 널(소프트웨어에서 \$null\$로 표시되는 정의되지 않은 값)은 결측값으로 간주됩니다. 사용 형식: NODE.null_missing.FIELDNAME
whitespace_missing	flag	이 특성이 T로 설정될 때, 공백(공백, 탭 및 줄 바꾸기)만 포함하는 값은 결측값으로 간주됩니다. 사용 형식: NODE.whitespace_missing.FIELDNAME
description	string	필드 레이블 또는 설명을 지정하는 데 사용됩니다.

표 42. 소스 노드 공통 특성 (계속).

특성 이름	데이터 유형	특성 설명
default_include	<i>flag</i>	기본 작동이 필드를 전달 또는 필터링하는 것인지 여부를 지정하는 키가 있는 특성입니다. NODE.default_include 예: set mynode:filternode.default_include = false
include	<i>flag</i>	개별 필드가 포함되는지 아니면 필터링되는지 여부를 판별하는 데 사용하는 키가 있는 특성: NODE.include.FIELDNAME.
new_name	<i>string</i>	
measure_type	Range / MeasureType.RANGE Discrete / MeasureType.DISCRETE Flag / MeasureType.FLAG Set / MeasureType.SET OrderedSet / MeasureType.ORDERED_SET Typeless / MeasureType.TYPELESS Collection / MeasureType.COLLECTION Geospatial / MeasureType.GEOSPATIAL	이 키가 있는 특성은 필드와 연관된 측정을 정의하는 데 사용할 수 있다는 점에서 type과 비슷합니다. 다른 점은 Python 스크립팅에서 setter 함수도 MeasureType 값의 하나로 전달될 수 있는 반면 getter는 항상 MeasureType 값에서 리턴한다는 점입니다.
collection_measure	Range / MeasureType.RANGE Flag / MeasureType.FLAG Set / MeasureType.SET OrderedSet / MeasureType.ORDERED_SET Typeless / MeasureType.TYPELESS	요약도표 필드(0의 깊이를 갖는 목록)의 경우 이 키가 있는 특성은 기본 값과 연관된 측정 유형을 정의합니다.
geo_type	Point MultiPoint LineString MultiLineString Polygon MultiPolygon	지리공간 필드의 경우 이 키가 있는 특성은 이 필드에 의해 표시되는 지리공간 오브젝트의 유형을 정의합니다. 이것은 값의 목록 깊이와 일관성을 가져야 합니다.
has_coordinate_system	<i>boolean</i>	지리공간 필드의 경우 이 특성은 이 필드가 좌표계를 갖는지 여부를 정의합니다.
coordinate_system	<i>string</i>	지리공간 필드의 경우 이 키가 있는 특성이 이 필드의 좌표계를 정의합니다.

표 42. 소스 노드 공통 특성 (계속).

특성 이름	데이터 유형	특성 설명
custom_storage_type	Unknown / MeasureType.UNKNOWN String / MeasureType.STRING Integer / MeasureType.INTEGER Real / MeasureType.REAL Time / MeasureType.TIME Date / MeasureType.DATE Timestamp / MeasureType.TIMESTAMP List / MeasureType.LIST	이 키가 있는 특성은 필드의 대체 저장 공간을 정의하는 데 사용할 수 있다는 점에서 custom_storage와 유사합니다. 다른 점은 Python 스크립팅에서 setter 함수도 StorageType 값의 하나로 전달될 수 있는 반면 getter는 항상 StorageType 값에서 리턴한다는 점입니다.
custom_list_storage_type	String / MeasureType.STRING Integer / MeasureType.INTEGER Real / MeasureType.REAL Time / MeasureType.TIME Date / MeasureType.DATE Timestamp / MeasureType.TIMESTAMP	목록 필드의 경우 이 키가 있는 특성이 기본 값의 저장 유형을 지정합니다.
custom_list_depth	<i>integer</i>	목록 필드의 경우 이 키가 있는 특성은 필드의 깊이를 지정합니다.
max_list_length	<i>integer</i>	측정 수준이 지리 공간 또는 컬렉션인 데이터에만 사용할 수 있습니다. 목록이 포함할 수 있는 요소 수를 지정하여 목록의 최대 길이를 설정하십시오.
max_string_length	<i>integer</i>	유형이 없는 데이터에만 사용할 수 있으며 테이블을 작성하기 위해 SQL을 생성할 때 사용합니다. 데이터에서 가장 큰 문자열 값을 입력하십시오. 그러면 문자열을 포함하기에 충분히 열이 테이블에 생성됩니다.

asimport 특성

Analytic Server 소스를 사용하면 HDFS(Hadoop Distributed File System)에서 스트림을 실행할 수 있습니다.

예제

```
node.setPropertyValue("use_default_as", False)
node.setPropertyValue("connection",
["false","9.119.141.141","9080","analyticserver","ibm","admin","admin","false","","",""])
```

표 43. *asimport* 특성.

asimport 특성	데이터 유형	특성 설명
data_source	string	데이터 소스의 이름
use_default_as	boolean	True로 설정할 경우 서버 options.cfg 파일에 구성된 기본 Analytic Server 연결이 사용됩니다. False로 설정할 경우 해당 노드의 연결이 사용됩니다.
connection	["string","string","string", "string","string","string","string", "string","string","string", "string", ,"string"]	Analytic Server 연결 세부사항이 포함된 목록 특성입니다. 형식은 ["is_secure_connect", "server_url", "server_port", "context_root", "consumer", "user_name", "password", "use-kerberos-auth", "kerberos-krb5-config-file-path", "kerberos-jaas-config-file-path", "kerberos-krb5-service-principal-name", "enable-kerberos-debug"]입니다. 여기서 is_secure_connect:는 보안 연결 사용 여부를 나타내며, true 또는 false입니다. use-kerberos-auth:는 Kerberos 인증 사용 여부를 나타내며, true 또는 false입니다. enable-kerberos-debug:는 Kerberos 인증의 디버그 모드 사용 여부를 나타내며, true 또는 false입니다.

cognosimport 노드 특성



IBM Cognos 소스 노드는 Cognos Analytics 데이터베이스에서 데이터를 가져옵니다.

예제

```
node = stream.create("cognosimport", "My node")
node.setPropertyValue("cognos_connection", ["http://mycogsrv1:9300/p2pd/servlet/dispatch",
True, "", "", ""])
node.setPropertyValue("cognos_package_name", "/Public Folders/GOSALES")
node.setPropertyValue("cognos_items", ["[GreatOutdoors].[BRANCH].[BRANCH_CODE]", "[GreatOutdoors].[BRANCH].[COUNTRY_CODE]"])
```

표 44. *cognosimport* 노드 특성.

cognosimport 노드 특성	데이터 유형	특성 설명
mode	Data Report	Cognos 데이터(기본값) 또는 보고서를 가져올지 여부를 지정합니다.

표 44. cognosimport 노드 특성 (계속).

cognosimport 노드 특성	데이터 유형	특성 설명
cognos_connection	["string", "flag", "string", "string", "string"]	<p>Cognos 서버에 대한 연결 세부사항이 포함된 목록 특성입니다. 형식은 다음과 같습니다. ["Cognos_server_URL", login_mode, "namespace", "username", "password"]</p> <p>여기서, Cognos_server_URL은 소스를 포함하는 Cognos 서버의 URL입니다.</p> <p>login_mode는 익명 로그인을 사용하는지 여부를 표시하며, true 또는 false입니다. true인 경우 다음 필드가 ""로 설정되어야 합니다.</p> <p>namespace는 서버에 로그인하는 데 사용하는 보안 인증 제공자를 지정합니다.</p> <p>username 및 password는 Cognos 서버에 로그인하는 데 사용하는 것입니다.</p> <p>login_mode 대신, 다음 모드도 사용할 수 있습니다.</p> <ul style="list-style-type: none"> • anonymousMode. 예: ['Cognos_server_url', 'anonymousMode', "namespace", "username", "password"] • credentialMode. 예: ['Cognos_server_url', 'credentialMode', "namespace", "username", "password"] • storedCredentialMode. 예: ['Cognos_server_url', 'storedCredentialMode', "stored_credential_name"] <p>여기서 stored_credential_name은 리포지토리에 있는 Cognos 신임 정보의 이름입니다.</p>
cognos_package_name	string	<p>데이터 오브젝트를 가져오고 있는 Cognos 패키지의 경로와 이름입니다. 예: /Public Folders/GOSALES</p> <p>참고: 슬래시만 유효합니다.</p>
cognos_items	["field", "field", ... , "field"]	<p>가져올 하나 이상의 데이터 오브젝트의 이름입니다. field의 형식은 [namespace].[query_subject].[query_item]입니다.</p>
cognos_filters	field	<p>데이터를 가져오기 전에 적용할 하나 이상의 필터의 이름입니다.</p>
cognos_data_parameters	list	<p>데이터에 대한 프롬프트 변수의 값입니다. 이름 및 값 쌍은 대괄호로 묶고, 다중 쌍은 쉼표로 구분되고 전체 문자열은 대괄호로 묶습니다.</p> <p>형식: [{"param1", "value"}, ..., {"paramN", "value"}]</p>

표 44. cognosimport 노드 특성 (계속).

cognosimport 노드 특성	데이터 유형	특성 설명
cognos_report_directory	field	보고서를 가져올 폴더 또는 패키지의 Cognos 경로입니다. 예: /Public Folders/GOSALES 참고: 슬래시만 유효합니다.
cognos_report_name	field	가져올 보고서의 보고서 위치 안에 있는 경로와 이름입니다.
cognos_report_parameters	list	보고서 변수의 값입니다. 이름 및 값 쌍은 대괄호로 묶고, 다중 쌍은 쉼표로 구분되고 전체 문자열은 대괄호로 묶습니다. 형식: [[{"param1", "value"},...,{"paramN", "value"}]]

databasenode 특성



데이터베이스 노드를 사용하면 ODBC(Open Database Connectivity)를 사용하여 Microsoft SQL Server, DB2, Oracle 및 기타를 포함한 다양한 다른 패키지로부터 데이터를 가져올 수 있습니다.

예제

```
import modeler.api
stream = modeler.script.stream()
nnode = stream.create("database", "My node")
node.setPropertyValue("mode", "Table")
node.setPropertyValue("query", "SELECT * FROM drug1n")
node.setPropertyValue("datasource", "Drug1n_db")
node.setPropertyValue("username", "spss")
node.setPropertyValue("password", "spss")
node.setPropertyValue("tablename", ".Drug1n")
```

표 45. databasenode 특성.

databasenode 특성	데이터 유형	특성 설명
mode	Table Query	대화 상자 제어를 사용하여 데이터베이스 테이블에 연결하려면 <i>Table</i> 을 선택하고, SQL을 사용하여 선택된 데이터베이스를 쿼리하려면 <i>Query</i> 를 지정하십시오.
datasource	string	데이터베이스 이름(아래 참고도 참조).
username	string	데이터베이스 연결 세부사항(아래 참고도 참조).
password	string	

표 45. databasenode 특성 (계속).

databasenode 특성	데이터 유형	특성 설명
credential	string	IBM SPSS Collaboration and Deployment Services에 저장된 신임 정보의 이름입니다. 이것을 username 및 password 특성 대신 사용할 수 있습니다. 신임 정보의 사용자 이름 및 비밀번호는 데이터베이스에 액세스하기 위해 필요한 사용자 이름 및 비밀번호와 매치해야 합니다.
use_credential		True 또는 False로 설정하십시오.
epassword	string	비밀번호를 스크립트에 하드코딩하는 것의 대안으로 인코딩된 비밀번호를 지정합니다. 자세한 정보는 56 페이지의 『인코딩된 비밀번호 생성』의 내용을 참조하십시오. 이 특성은 실행 중에 읽기 전용입니다.
tablename	string	액세스하려는 테이블의 이름입니다.
strip_spaces	None Left Right Both	문자열에서 선행 및 후미 공백을 삭제하는 옵션입니다.
use_quotes	AsNeeded Always Never	쿼리가 데이터베이스로 전송될 때(예를 들어 공백이나 구두점을 포함하는 경우) 테이블 및 열 이름을 인용부호로 묶을지 여부를 지정하십시오.
query	string	제출하려는 쿼리에 대한 SQL 코드를 지정합니다.

참고: 데이터베이스 이름(datasource 특성에 있는)이 하나 이상의 공백, 마침표("완전 중단"이라고도 함) 또는 밑줄을 포함하는 경우, 이를 문자열로 처리하기 위해 "백슬래시 큰따옴표" 형식을 사용할 수 있습니다. 예: "{ \"db2v9.7.6_linux\" }" 또는 "{ \"TDATA 131\" }". 또한, 다음 예에서와 같이 항상 datasource 문자열 값을 큰따옴표로 묶으십시오. "{ \"SQL Server\", spssuser, abcd1234, false }".

참고: 데이터베이스 이름(datasource 특성에 있는)이 공백을 포함하는 경우, datasource, username, password에 대한 개별 특성 대신 다음 형식으로 하나의 datasource 특성을 사용할 수도 있습니다.

표 46. databasenode 특성 - 데이터 소스 특정.

databasenode 특성	데이터 유형	특성 설명
datasource	string	형식: [database_name, username, password[, true false]] 마지막 모수는 암호화된 비밀번호와 함께 사용하기 위한 것입니다. 이것이 true로 설정되면 비밀번호는 사용 전에 복호화됩니다.

데이터 소스를 변경하려는 경우에도 이 형식을 사용하십시오. 그러나 사용자 이름이나 비밀번호만 변경하려는 경우 username 또는 password 특성을 사용할 수 있습니다.

datacollectionimportnode 특성



Data Collection 데이터 가져오기 노드는 시장 조사 제품이 사용하는 Data Collection 데이터 모델을 기반으로 설문조사 데이터를 가져옵니다. 이 노드를 사용하려면 Data Collection 데이터 라이브러리가 설치되어야 합니다.

예제

```
node = stream.create("datacollectionimport", "My node")
node.setPropertyValue("metadata_name", "mrQvDsc")
node.setPropertyValue("metadata_file", "C:/Program Files/IBM/SPSS/DataCollection/DDL/Data/
Quanvert/Museum/museum.pkd")
node.setPropertyValue("casedata_name", "mrQvDsc")
node.setPropertyValue("casedata_source_type", "File")
node.setPropertyValue("casedata_file", "C:/Program Files/IBM/SPSS/DataCollection/DDL/Data/
Quanvert/Museum/museum.pkd")
node.setPropertyValue("import_system_variables", "Common")
node.setPropertyValue("import_multi_response", "MultipleFlags")
```

표 47. datacollectionimportnode 특성.

datacollectionimportnode 특성	데이터 유형	특성 설명
metadata_name	string	MDSC의 이름입니다. 특수값 DimensionsMDD는 표준 Data Collection 메타데이터 문서를 사용해야 함을 표시합니다. 기타 가능한 값은 다음과 같습니다. mrAD0Dsc mrI2dDsc mrLogDsc mrQdiDrsDsc mrQvDsc mrSampleReportingMDSC mrSavDsc mrSCDsc mrScriptMDSC 특수값 none은 MDSC가 없음을 표시합니다.
metadata_file	string	메타데이터가 저장되는 파일의 이름입니다.

표 47. *datacollectionimportnode* 특성 (계속).

datacollectionimportnode 특성	데이터 유형	특성 설명
<code>casedata_name</code>	<i>string</i>	CDSC의 이름입니다. 가능한 값은 다음과 같습니다. mrAD0Dsc mrI2dDsc mrLogDsc mrPunchDSC mrQdiDrsDsc mrQvDsc mrRdbDsc2 mrSavDsc mrScDSC mrXm1Dsc 특수값 <code>none</code> 은 CDSC가 없음을 표시합니다.
<code>casedata_source_type</code>	Unknown File Folder UDL DSN	CDSC의 소스 유형을 표시합니다.
<code>casedata_file</code>	<i>string</i>	<code>casedata_source_type</code> 이 <i>File</i> 일 때, 케이스 데이터를 포함하는 파일을 지정합니다.
<code>casedata_folder</code>	<i>string</i>	<code>casedata_source_type</code> 이 <i>Folder</i> 일 때, 케이스 데이터를 포함하는 폴더를 지정합니다.
<code>casedata_udl_string</code>	<i>string</i>	<code>casedata_source_type</code> 이 <i>UDL</i> 일 때, 케이스 데이터를 포함하는 데이터 소스에 대한 OLD-DB 연결 문자열을 지정합니다.
<code>casedata_dsn_string</code>	<i>string</i>	<code>casedata_source_type</code> 이 <i>DSN</i> 일 때, 데이터 소스의 ODBC 연결 문자열을 지정합니다.
<code>casedata_project</code>	<i>string</i>	Data Collection 데이터베이스에서 케이스 데이터를 읽을 때 프로젝트의 이름을 입력할 수 있습니다. 다른 모든 케이스 데이터 유형의 경우 이 설정은 공백이어야 합니다.
<code>version_import_mode</code>	All Latest Specify	버전이 처리되는 방법을 정의합니다.
<code>specific_version</code>	<i>string</i>	<code>version_import_mode</code> 가 <i>Specify</i> 일 때, 가져올 케이스 데이터의 버전을 정의합니다.
<code>use_language</code>	<i>string</i>	특정 언어의 레이블을 사용하는지 여부를 정의합니다.
<code>language</code>	<i>string</i>	<code>use_language</code> 가 <code>true</code> 인 경우, 가져오기 시에 사용할 언어 코드를 정의합니다. 언어 코드는 케이스 데이터에서 사용 가능한 코드 중 하나여야 합니다.
<code>use_context</code>	<i>string</i>	특정 컨텍스트를 가져와야 하는지 여부를 정의합니다. 컨텍스트는 응답과 연관된 설명을 변화시키는데 사용됩니다.

표 47. *datacollectionimportnode* 특성 (계속).

datacollectionimportnode 특성	데이터 유형	특성 설명
context	<i>string</i>	use_context가 true인 경우, 가져올 컨텍스트를 정의합니다. 컨텍스트는 케이스 데이터에서 사용 가능한 컨텍스트 중 하나여야 합니다.
use_label_type	<i>string</i>	특정 유형의 레이블을 가져와야 하는지 여부를 정의합니다.
label_type	<i>string</i>	use_label_type이 true인 경우, 가져올 레이블 유형을 정의합니다. 레이블 유형은 케이스 데이터에서 사용 가능한 유형 중 하나여야 합니다.
user_id	<i>string</i>	명시 로그인에 필요한 데이터베이스의 경우, 해당 데이터 소스에 액세스하기 위한 사용자 ID와 비밀번호를 제공할 수 있습니다.
password	<i>string</i>	
import_system_variables	Common None All	가져오는 시스템 변수를 지정합니다.
import_codes_variables	<i>flag</i>	
import_sourcefile_variables	<i>flag</i>	
import_multi_response	MultipleFlags Single	

excelimportnode 특성



Excel 가져오기 노드는 Microsoft Excel로부터 .xlsx 파일 형식으로 데이터를 가져옵니다. ODBC 데이터 소스는 필요하지 않습니다.

예

```
#To use a named range:
node = stream.create("excelimport", "My node")
node.setPropertyValue("excel_file_type", "Excel2007")
node.setPropertyValue("full_filename", "C:/drug.xlsx")
node.setPropertyValue("use_named_range", True)
node.setPropertyValue("named_range", "DRUG")
node.setPropertyValue("read_field_names", True)
```

```
#To use an explicit range:
node = stream.create("excelimport", "My node")
node.setPropertyValue("excel_file_type", "Excel2007")
node.setPropertyValue("full_filename", "C:/drug.xlsx")
node.setPropertyValue("worksheet_mode", "Name")
node.setPropertyValue("worksheet_name", "Drug")
node.setPropertyValue("explicit_range_start", "A1")
node.setPropertyValue("explicit_range_end", "F300")
```

표 48. excelimportnode 특성.

excelimportnode 특성	데이터 유형	특성 설명
excel_file_type	Excel2007	
full_filename	string	경로를 포함한 완전한 파일 이름입니다.
use_named_range	Boolean	이름 지정된 범위 사용 여부입니다. True인 경우 named_range 특성이 읽을 범위를 지정하는 데 사용하며, 다른 워크시트와 데이터 범위 설정은 무시됩니다.
named_range	string	
worksheet_mode	Index Name	워크시트가 지수 또는 이름으로 정의되는지 여부를 지정합니다.
worksheet_index	integer	읽을 워크시트의 지수로서, 첫 번째 워크시트의 경우 0으로 시작하고 두 번째는 1인 형식입니다.
worksheet_name	string	읽을 워크시트의 이름입니다.
data_range_mode	FirstNonBlank ExplicitRange	범위를 판별하는 방법을 지정합니다.
blank_rows	StopReading ReturnBlankRows	data_range_mode가 FirstNonBlank일 때, 공백 행을 처리하는 방법을 지정합니다.
explicit_range_start	string	data_range_mode가 ExplicitRange일 때, 읽을 범위의 시작점을 지정합니다.
explicit_range_end	string	
read_field_names	Boolean	지정된 범위의 첫 번째 행을 필드(열) 이름으로 사용하는지 여부를 지정합니다.

extensionimportnode 특성



확장 가져오기 노드를 사용하면 R 또는 Python for Spark 스크립트를 실행하여 데이터를 가져올 수 있습니다.

Python for Spark 예제

```
##### Script example for Python for Spark
import modeler.api
stream = modeler.script.stream()
node = stream.create("extension_importer", "extension_importer")
node.setPropertyValue("syntax_type", "Python")

python_script = """
import spss.pyspark
from pyspark.sql.types import *

cxt = spss.pyspark.runtime.getContext()
```

```

_schema = StructType([StructField('id', LongType(), nullable=False), \
StructField('age', LongType(), nullable=True), \
StructField('Sex', StringType(), nullable=True), \
StructField('BP', StringType(), nullable=True), \
StructField('Cholesterol', StringType(), nullable=True), \
StructField('K', DoubleType(), nullable=True), \
StructField('Na', DoubleType(), nullable=True), \
StructField('Drug', StringType(), nullable=True)])

if cxt.isComputeDataModelOnly():
    cxt.setSparkOutputSchema(_schema)
else:
    df = cxt.getSparkInputData()
    if df is None:
        drugList=[(1,23,'F','HIGH','HIGH',0.792535,0.031258,'drugY'), \
(2,47,'M','LOW','HIGH',0.739309,0.056468,'drugC'),\
(3,47,'M','LOW','HIGH',0.697269,0.068944,'drugC'),\
(4,28,'F','NORMAL','HIGH',0.563682,0.072289,'drugX'),\
(5,61,'F','LOW','HIGH',0.559294,0.030998,'drugY'),\
(6,22,'F','NORMAL','HIGH',0.676901,0.078647,'drugX'),\
(7,49,'F','NORMAL','HIGH',0.789637,0.048518,'drugY'),\
(8,41,'M','LOW','HIGH',0.766635,0.069461,'drugC'),\
(9,60,'M','NORMAL','HIGH',0.777205,0.05123,'drugY'),\
(10,43,'M','LOW','NORMAL',0.526102,0.027164,'drugY')]

    sqlcxt = cxt.getSparkSQLContext()
    rdd = cxt.getSparkContext().parallelize(drugList)
    print 'pyspark read data count = '+str(rdd.count())
    df = sqlcxt.createDataFrame(rdd, _schema)

    cxt.setSparkOutputData(df)
    """

node.setPropertyValue("python_syntax", python_script)

```

R 예제

```

#### Script example for R
node.setPropertyValue("syntax_type", "R")

R_script = """# 'JSON Import' Node v1.0 for IBM SPSS Modeler
# 'RJSONIO' package created by Duncan Temple Lang - http://cran.r-project.org/web/packages/RJSONIO
# 'plyr' package created by Hadley Wickham http://cran.r-project.org/web/packages/plyr
# Node developer: Danil Savine - IBM Extreme Blue 2014
# Description: This node allows you to import into SPSS a table data from a JSON.
# Install function for packages
packages <- function(x){
  x <- as.character(match.call())[[2]]
  if (!require(x,character.only=TRUE)){
    install.packages(pkgs=x,repos="http://cran.r-project.org")
    require(x,character.only=TRUE)
  }
}
# packages
packages(RJSONIO)
packages(plyr)
### This function is used to generate automatically the dataModel
getMetaData <- function (data) {
  if (dim(data)[1]<=0) {

```

```

print("Warning : modelerData has no line, all fieldStorage fields set to strings")
getStorage <- function(x){return("string")}

} else {

getStorage <- function(x) {
  res <- NULL
  #if x is a factor, typeof will return an integer so we treat the case on the side
  if(is.factor(x)) {
    res <- "string"
  } else {
    res <- switch(typeof(unlist(x)),
                  integer = "integer",
                  double = "real",
                  character = "string",
                  "string")
  }
  return (res)
}
}

col = vector("list", dim(data)[2])
for (i in 1:dim(data)[2]) {
  col[[i]] <- c(fieldName=names(data[i]),
               fieldLabel="",
               fieldStorage=getStorage(data[i]),
               fieldMeasure="",
               fieldFormat="",
               fieldRole="")
}
mdm<-do.call(cbind,col)
mdm<-data.frame(mdm)
return(mdm)
}

# From JSON to a list
txt <- readLines('C:/test.json')
formattedtxt <- paste(txt, collapse = '')
json.list <- fromJSON(formattedtxt)
# Apply path to json.list
if(strsplit(x='true', split='
',fixed=TRUE)[[1]][1]) {
  path.list <- unlist(strsplit(x='id_array', split=','))
  i = 1
  while(i<length(path.list)+1){
    if(is.null(getElement(json.list, path.list[i]))){
      json.list <- json.list[[1]]
    }else{
      json.list <- getElement(json.list, path.list[i])
      i <- i+1
    }
  }
}

# From list to dataframe via unlisted json
i <-1
filled <- data.frame()
while(i < length(json.list)+ 1){
  unlisted.json <- unlist(json.list[[i]])
  to.fill <- data.frame(t(as.data.frame(unlisted.json, row.names = names(unlisted.json))),
stringsAsFactors=FALSE)
  filled <- rbind.fill(filled,to.fill)
  i <- 1 + i
}

```

```

}
# Export to SPSS Modeler Data
modelerData <- filled
print(modelerData)
modelerDataModel <- getMetaData(modelerData)
print(modelerDataModel)

"""

node.setPropertyValue("r_syntax", R_script)

```

표 49. *extensionimportnode* 특성

extensionimportnode 특성	데이터 유형	특성 설명
syntax_type	R Python	실행할 스크립트, R 또는 Python을 지정하십시오(R이 기본값).
r_syntax	string	실행할 R 스크립팅 구문입니다.
python_syntax	string	실행할 Python 스크립팅 구문입니다.

fixedfilenode 특성



고정 파일 노드는 고정 필드 텍스트 파일, 즉 그의 필드가 구분되지 않고 동일한 위치에서 시작하며 고정된 길이의 파일로부터 데이터를 가져옵니다. 머신 생성 또는 레거시 데이터가 자주 고정 필드 형식으로 저장됩니다.

예제

```

node = stream.create("fixedfile", "My node")
node.setPropertyValue("full_filename", "$CLEO_DEMOS/DRUG1n")
node.setPropertyValue("record_len", 32)
node.setPropertyValue("skip_header", 1)
node.setPropertyValue("fields", [{"Age", 1, 3}, {"Sex", 5, 7}, {"BP", 9, 10}, {"Cholesterol", 12, 22}, {"Na", 24, 25}, {"K", 27, 27}, {"Drug", 29, 32}])
node.setPropertyValue("decimal_symbol", "Period")
node.setPropertyValue("lines_to_scan", 30)

```

표 50. *fixedfilenode* 특성.

fixedfilenode 특성	데이터 유형	특성 설명
record_len	number	각 레코드에 있는 문자 수를 지정합니다.
line_oriented	flag	각 레코드의 끝에서 줄 바꾸기 문자를 건너뛸니다.
decimal_symbol	Default Comma Period	데이터 소스에서 사용하는 소수점 구분자의 유형입니다.
skip_header	number	첫 번째 레코드의 시작에서 무시할 행 수를 지정합니다. 열 헤더 무시를 위해 유용합니다.
auto_recognize_datetime	flag	날짜 또는 시간이 소스 데이터에서 자동으로 식별되는지 여부를 지정합니다.
lines_to_scan	number	
fields	list	구조화 특성입니다.

표 50. *fixedfilenode* 특성 (계속).

fixedfilenode 특성	데이터 유형	특성 설명
<code>full_filename</code>	<i>string</i>	디렉토리를 포함하여, 읽을 파일의 전체 이름입니다.
<code>strip_spaces</code>	None Left Right Both	가져올 때 문자열의 선행 및 후미 공백을 삭제합니다.
<code>invalid_char_mode</code>	Discard Replace	데이터 입력에서 유효하지 않은 문자(널, 0 또는 현재 인코딩에 존재하지 않는 모든 문자)를 제거하거나 유효하지 않은 문자를 지정된 한 문자 기호로 바꿉니다.
<code>invalid_char_replacement</code>	<i>string</i>	
<code>use_custom_values</code>	<i>flag</i>	
<code>custom_storage</code>	Unknown String Integer Real Time Date Timestamp	

표 50. fixedfilenode 특성 (계속).

fixedfilenode 특성	데이터 유형	특성 설명
custom_date_format	"DDMMYY" "MMDDYY" "YYMMDD" "YYYYMMDD" "YYYYDDD" DAY MONTH "DD-MM-YY" "DD-MM-YYYY" "MM-DD-YY" "MM-DD-YYYY" "DD-MON-YY" "DD-MON-YYYY" "YYYY-MM-DD" "DD.MM.YY" "DD.MM.YYYY" "MM.DD.YY" "MM.DD.YYYY" "DD.MON.YY" "DD.MON.YYYY" "DD/MM/YY" "DD/MM/YYYY" "MM/DD/YY" "MM/DD/YYYY" "DD/MON/YY" "DD/MON/YYYY" MON YYYY q Q YYYY ww WK YYYY	이 특성은 사용자 정의 저장 공간이 지정된 경우에만 적용할 수 있습니다.

표 50. *fixedfilenode* 특성 (계속).

fixedfilenode 특성	데이터 유형	특성 설명
custom_time_format	"HHMMSS" "HHMM" "MMSS" "HH:MM:SS" "HH:MM" "MM:SS" "(H)H:(M)M:(S)S" "(H)H:(M)M" "(M)M:(S)S" "HH.MM.SS" "HH.MM" "MM.SS" "(H)H.(M)M.(S)S" "(H)H.(M)M" "(M)M.(S)S"	이 특성은 사용자 정의 저장 공간이 지정된 경우에만 적용할 수 있습니다.
custom_decimal_symbol	<i>field</i>	사용자 정의 저장 공간이 지정된 경우에만 적용 가능합니다.
encoding	StreamDefault SystemDefault "UTF-8"	텍스트 인코딩 방법을 지정합니다.

gsdata_import 노드 특성



맵 또는 공간 데이터를 데이터 마이닝 세션으로 가져오려면 지리 공간적 소스 노드를 사용하십시오.

표 51. *gsdata_import* 노드 특성

gsdata_import 노드 특성	데이터 유형	특성 설명
full_filename	<i>string</i>	로드하려는 .shp 파일의 파일 경로를 입력하십시오.
map_service_URL	<i>string</i>	연결할 맵 서비스 URL을 입력하십시오.
map_name	<i>string</i>	map_service_URL을 사용하는 경우에만 사용됩니다. 이것은 맵 서비스의 최상위 수준 폴더 구조를 포함합니다.

sasimportnode 특성



SAS 가져오기 노드는 SAS 데이터를 IBM SPSS Modeler로 가져옵니다.

예제

```
node = stream.create("sasimport", "My node")
node.setPropertyValue("format", "Windows")
node.setPropertyValue("full_filename", "C:/data/retail.sas7bdat")
node.setPropertyValue("member_name", "Test")
node.setPropertyValue("read_formats", False)
node.setPropertyValue("full_format_filename", "Test")
node.setPropertyValue("import_names", True)
```

표 52. sasimportnode 특성.

sasimportnode 특성	데이터 유형	특성 설명
format	Windows UNIX Transport SAS7 SAS8 SAS9	가져올 파일의 형식.
full_filename	string	경로를 포함하여, 사용자가 입력하는 완전한 파일 이름입니다.
member_name	string	지정된 SAS 전송 파일에서 가져올 멤버를 지정하십시오.
read_formats	flag	지정된 형식 파일에서 데이터 형식(예: 변수 레이블)을 읽습니다.
full_format_filename	string	
import_names	NamesAndLabels LabelsasNames	가져올 때 변수 이름 및 레이블 매핑을 위한 방법을 지정합니다.

simgennode 특성



시뮬레이션 생성 노드는 사용자가 지정한 통계 분포를 사용하는 스크래치로부터 또는 기존 히스토리 데이터에 대해 시뮬레이션 적합 노드를 실행하여 얻은 분포를 자동으로 사용하여 시뮬레이션된 데이터를 생성하는 쉬운 방법을 제공합니다. 이것은 모델 입력에 불확실성이 존재하는 상황에서 예측 모델의 결과를 평가하기 원할 때 유용합니다.

표 53. simgennode 특성.

simgennode 특성	데이터 유형	특성 설명
fields	구조화된 특성	예제 참조

표 53. *simgennode* 특성 (계속).

simgennode 특성	데이터 유형	특성 설명
correlations	구조화된 특성	예제 참조
keep_min_max_setting	<i>boolean</i>	
refit_correlations	<i>boolean</i>	
max_cases	<i>integer</i>	최소값은 1000, 최대값은 2,147,483,647
create_iteration_field	<i>boolean</i>	
iteration_field_name	<i>string</i>	
replicate_results	<i>boolean</i>	
random_seed	<i>integer</i>	
parameter_xml	<i>string</i>	모수 Xml을 문자열로 리턴합니다.

필드 예

이것은 다음 구문을 갖는 구조화된 슬롯 모수입니다.

```
simgennode.setPropertyValue("fields", [
    [field1, storage, locked, [distribution1], min, max],
    [field2, storage, locked, [distribution2], min, max],
    [field3, storage, locked, [distribution3], min, max]
])
```

*distribution*은 식별 이름과 그 뒤에 오는 속성 이름과 값의 쌍을 포함하는 목록의 선언입니다. 각 분포는 다음 방법으로 정의됩니다.

```
[distributionname, [[par1], [par2], [par3]]]
```

```
simgennode = modeler.script.stream().createAt("simgen", u"Sim Gen", 726, 322)
simgennode.setPropertyValue("fields", [{"Age", "integer", False, ["Uniform", [{"min", "1"}, {"max", "2"}]}, "", ""]])
```

예를 들어, 이항 분포를 갖는 단일 필드를 생성하는 노드를 작성하려면 다음 스크립트를 사용할 수 있습니다.

```
simgen_node1 = modeler.script.stream().createAt("simgen", u"Sim Gen", 200, 200)
simgen_node1.setPropertyValue("fields", [{"Education", "Real", False, ["Binomial", [{"n", 32}, {"prob", 0.7}]], "", ""]])
```

이항 분포는 *n* 및 *prob*의 두 모수를 갖습니다. 이항은 최소값과 최대값을 지원하지 않으므로, 이들은 빈 문자열로서 제공됩니다.

참고: *distribution*을 직접 설정할 수 없습니다. *fields* 특성과 결합하여 사용합니다.

다음 예는 모든 가능한 분포 유형을 보여줍니다. 임계값이 *NegativeBinomialFailures* 및 *NegativeBinomialTrial* 모두에서 *thresh*로서 입력됨을 참고하십시오.

```
stream = modeler.script.stream()
simgennode = stream.createAt("simgen", u"Sim Gen", 200, 200)

beta_dist = ["Field1", "Real", False, ["Beta", [{"shape1", "1"}, {"shape2", "2"}]], "", ""]
binomial_dist = ["Field2", "Real", False, ["Binomial", [{"n", "1"}, {"prob", "1"}]], "", ""]
categorical_dist = ["Field3", "String", False, ["Categorical", [{"A", 0.3}, {"B", 0.5}, {"C", 0.2}], "", ""]
dice_dist = ["Field4", "Real", False, ["Dice", [{"1", "0.5"}, {"2", "0.5"}]], "", ""]
exponential_dist = ["Field5", "Real", False, ["Exponential", [{"scale", "1"}]], "", ""]
```

```

fixed_dist = ["Field6", "Real", False, ["Fixed", [{"value","1"}]], "", ""]
gamma_dist = ["Field7", "Real", False, ["Gamma", [{"scale","1"}, {"shape","1"}]], "", ""]
lognormal_dist = ["Field8", "Real", False, ["Lognormal", [{"a","1"}, {"b","1"}]], "", ""]
negbinomialfailures_dist = ["Field9", "Real", False, ["NegativeBinomialFailures", [{"prob","0.5"}, {"thresh","1"}]], "", ""]
negbinomialtrial_dist = ["Field10", "Real", False, ["NegativeBinomialTrials", [{"prob","0.2"}, {"thresh","1"}]], "", ""]
normal_dist = ["Field11", "Real", False, ["Normal", [{"mean","1"}, {"stddev","2"}]], "", ""]
poisson_dist = ["Field12", "Real", False, ["Poisson", [{"mean","1"}]], "", ""]
range_dist = ["Field13", "Real", False, ["Range", [{"BEGIN","[1,3]"}, {"END","[2,4]"}, {"PROB","[[0.5],[0.5]]"}]], "", ""]
triangular_dist = ["Field14", "Real", False, ["Triangular", [{"min","0"}, {"max","1"}, {"mode","1"}]], "", ""]
uniform_dist = ["Field15", "Real", False, ["Uniform", [{"min","1"}, {"max","2"}]], "", ""]
weibull_dist = ["Field16", "Real", False, ["Weibull", [{"a","0"}, {"b","1"}, {"c","1"}]], "", ""]

simgennode.setPropertyValue("fields", [
  beta_dist, \
  binomial_dist, \
  categorical_dist, \
  dice_dist, \
  exponential_dist, \
  fixed_dist, \
  gamma_dist, \
  lognormal_dist, \
  negbinomialfailures_dist, \
  negbinomialtrial_dist, \
  normal_dist, \
  poisson_dist, \
  range_dist, \
  triangular_dist, \
  uniform_dist, \
  weibull_dist
])

```

상관계수 예

이것은 다음 구문을 갖는 구조화된 슬롯 모수입니다.

```

simgennode.setPropertyValue("correlations", [
  [field1, field2, correlation],
  [field1, field3, correlation],
  [field2, field3, correlation]
])

```

상관계수는 +1과 -1 사이의 임의의 숫자일 수 있습니다. 원하는 만큼의 상관계수를 지정할 수 있습니다. 모든 지정되지 않은 상관계수는 0으로 설정됩니다. 어떠한 필드도 알 수 없는 경우 상관계수 값은 상관행렬(또는 테이블)에서 설정되어야 하며 빨간색 텍스트로 표시됩니다. 알 수 없는 필드가 있을 때는 노드를 실행할 수 없습니다.

statisticsimportnode 특성



IBM SPSS Statistics 파일 노드는 IBM SPSS Statistics가 사용하는 .sav 파일 형식뿐 아니라 동일한 형식을 사용하는 IBM SPSS Modeler에 저장된 캐시 파일로부터 데이터를 읽습니다.

이 노드의 특성은 349 페이지의 『statisticsimportnode 특성』에서 설명됩니다.

tm1odataimport 노드 특성



IBM Cognos TM1 소스 노드는 Cognos TM1 데이터베이스에서 데이터를 가져옵니다.

표 54. tm1odataimport 노드 특성

tm1odataimport 노드 특성	데이터 유형	특성 설명
admin_host	string	REST API의 호스트 이름에 대한 URL입니다.
server_name	string	admin_host에서 선택된 TM1 서버의 이름입니다.
credential_type	inputCredential 또는 storedCredential	신임 정보 유형을 표시하는 데 사용합니다.
input_credential	list	credential_type이 inputCredential이면 도메인 이름, 사용자 이름 및 비밀번호를 지정하십시오.
stored_credential_name	string	credential_type이 storedCredential이면 C&DS 서버에서 신임 정보의 이름을 지정하십시오.
selected_view	["field" "field"]	선택된 TM1 큐브의 세부사항과 SPSS로 데이터를 가져올 큐브 보기의 이름이 포함된 목록 특성입니다. 예: TM1_import.setPropertyValue("selected_view", ['plan_BudgetPlan', 'Goal Input'])
is_private_view	flag	selected_view가 개인용 보기인지 지정합니다. 기본값은 false입니다.
selected_columns	["field"]	선택한 열을 지정하십시오. 한 항목만 지정할 수 있습니다. 예: setPropertyValue("selected_columns", ["Measures"])
selected_rows	["field" "field"]	선택한 행을 지정하십시오. 예: setPropertyValue("selected_rows", ["Dimension_1_1", "Dimension_2_1", "Dimension_3_1", "Periods"])

tm1import 노드 특성(더 이상 사용되지 않음)



IBM Cognos TM1 소스 노드는 Cognos TM1 데이터베이스에서 데이터를 가져옵니다.

참고: 이 노드는 Modeler 18.0에서는 더 이상 사용되지 않습니다. 대체 노드 스크립트 이름은 tm1odataimport입니다.

표 55. tm1import 노드 특성.

tm1import 노드 특성	데이터 유형	특성 설명
pm_host	string	참고: 버전 16.0 및 17.0에만 해당 호스트 이름입니다. 예: TM1_import.setPropertyValue("pm_host", 'http://9.191.86.82:9510/pmhub/pm')
tm1_connection	["field";"field", ... ,"field"]	참고: 버전 16.0 및 17.0에만 해당 TM1 서버에 대한 연결 세부사항이 포함된 목록 특성입니다. 형식은 ["TM1_Server_Name", "tm1_username", "tm1_password"]입니다. 예: TM1_import.setPropertyValue("tm1_connection", ['Planning Sample', "admin", "apple"])
selected_view	["field" "field"]	선택된 TM1 큐브의 세부사항과 SPSS로 데이터를 가져올 큐브 보기의 이름이 포함된 목록 특성입니다. 예: TM1_import.setPropertyValue("selected_view", ['plan_BudgetPlan', 'Goal Input'])
selected_column	["field"]	선택한 열을 지정하십시오. 한 항목만 지정할 수 있습니다. 예: setPropertyValue("selected_columns", ["Measures"])
selected_rows	["field" "field"]	선택한 행을 지정하십시오. 예: setPropertyValue("selected_rows", ["Dimension_1_1", "Dimension_2_1", "Dimension_3_1", "Periods"])

twcimport 노드 특성



TWC 소스 노드는 The Weather Company, IBM Business에서 기상 데이터를 가져옵니다. 위치의 히스토리 기상 데이터 또는 예측 기상 데이터를 얻는 데 사용할 수 있습니다. 그러면 사용 가능한 가장 정확하고 정밀한 날짜 데이터를 사용하여 더 나은 의사결정을 내리기 위한 기상 중심 비즈니스 솔루션을 개발할 수 있습니다.

표 56. twcimport 노드 특성

twcimport 노드 특성	데이터 유형	특성 설명
TWCDataImport.latitude	Real	[-90.0~90.0] 형식으로 위도 값을 지정합니다.
TWCDataImport.longitude	Real	[-180.0~180.0] 형식으로 경도 값을 지정합니다.
TWCDataImport.licenseKey	string	The Weather Company에서 얻은 라이선스 키를 지정합니다.
TWCDataImport.measurmentUnit	English Metric Hybrid	측정 단위를 지정합니다. 가능한 값은 English, Metric 또는 Hybrid입니다. Metric이 기본값입니다.
TWCDataImport.dataType	Historical Forecast	입력할 기상 데이터 유형을 지정합니다. 가능한 값은 Historical 또는 Forecast입니다. Historical이 기본값입니다.

표 56. *twcimport* 노드 특성 (계속)

twcimport 노드 특성	데이터 유형	특성 설명
TwCDataImport.startDate	Integer	Historical이 TwCDataImport.dataType에 지정된 경우 yyyyMMdd 형식으로 시작 날짜를 지정하십시오.
TwCDataImport.endDate	Integer	Historical이 TwCDataImport.dataType에 지정된 경우 yyyyMMdd 형식으로 종료 날짜를 지정하십시오.
TwCDataImport.forecastHour	6 12 24 48	TwCDataImport.dataType으로 Forecast가 지정된 경우 시간으로 6, 12, 24 또는 48을 지정하십시오.

userinputnode 특성



사용자 입력 노드는 스크래치로부터 또는 기존 데이터를 변경하여 합성 데이터를 작성하는 쉬운 방법을 제공합니다. 이것은 예를 들어 모델링을 위한 검정 데이터 세트를 작성할 때 유용합니다.

예제

```
node = stream.create("userinput", "My node")
node.setPropertyValue("names", ["test1", "test2"])
node.setKeyedPropertyValue("data", "test1", "2, 4, 8")
node.setKeyedPropertyValue("custom_storage", "test1", "Integer")
node.setPropertyValue("data_mode", "Ordered")
```

표 57. *userinputnode* 특성.

userinputnode 특성	데이터 유형	특성 설명
data		
names		노드가 생성하는 필드 이름 목록을 설정 또는 리턴하는 구조화 슬롯입니다.
custom_storage	Unknown String Integer Real Time Date Timestamp	필드의 저장 공간을 설정 또는 리턴하는 키가 있는 슬롯입니다.

표 57. *userinputnode* 특성 (계속).

userinputnode 특성	데이터 유형	특성 설명
data_mode	Combined Ordered	Combined가 지정되면, 설정된 값과 최소/최대 값의 각 조합에 대한 레코드가 생성됩니다. 생성되는 레코드 수는 각 필드에 있는 값 수의 곱과 같습니다. Ordered가 지정되는 경우 데이터의 행을 생성하기 위해 각 레코드에 대해 각 열에서 하나의 값을 가져옵니다. 생성되는 레코드 수는 필드와 연관된 가장 큰 숫자 값과 같습니다. 작은 데이터 값을 갖는 모든 필드는 널값으로 채워집니다.
values		참고: 이 특성은 <i>userinputnode.data</i> 를 위해 폐기되었으며 더 이상 사용하지 않아야 합니다.

variablefilenode 특성



가변파일 노드는 자유 필드 텍스트 파일, 즉 레코드가 일정한 수의 필드를 포함하지만 변하는 문자를 포함하는 파일로부터 데이터를 읽습니다. 이 노드는 또한 고정 길이 헤더 텍스트와 특정 유형의 주석을 갖는 파일에도 유용합니다.

예제

```
node = stream.create("variablefile", "My node")
node.setPropertyValue("full_filename", "$CLEO_DEMOS/DRUG1n")
node.setPropertyValue("read_field_names", True)
node.setPropertyValue("delimit_other", True)
node.setPropertyValue("other", ",")
node.setPropertyValue("quotes_1", "Discard")
node.setPropertyValue("decimal_symbol", "Comma")
node.setPropertyValue("invalid_char_mode", "Replace")
node.setPropertyValue("invalid_char_replacement", "|")
node.setKeyedPropertyValue("use_custom_values", "Age", True)
node.setKeyedPropertyValue("direction", "Age", "Input")
node.setKeyedPropertyValue("type", "Age", "Range")
node.setKeyedPropertyValue("values", "Age", [1, 100])
```

표 58. *variablefilenode* 특성.

variablefilenode 특성	데이터 유형	특성 설명
skip_header	<i>number</i>	첫 번째 레코드의 시작에서 무시할 문자 수를 지정합니다.
num_fields_auto	<i>flag</i>	각 레코드에 있는 필드 수를 자동으로 판별합니다. 레코드는 줄 바꾸기 문자로 끝나야 합니다.
num_fields	<i>number</i>	각 레코드에 있는 필드 수를 수동으로 지정합니다.
delimit_space	<i>flag</i>	파일에서 필드 경계를 구분하기 위해 사용하는 문자를 지정합니다.
delimit_tab	<i>flag</i>	
delimit_new_line	<i>flag</i>	

표 58. *variablefilenode* 특성 (계속).

variablefilenode 특성	데이터 유형	특성 설명
<code>delimit_non_printing</code>	<i>flag</i>	
<code>delimit_comma</code>	<i>flag</i>	쉼표가 필드 구분자이면서 스트림에 대한 소수점 구분자인 경우 <code>delimit_other</code> 를 <i>true</i> 로 설정하고 <code>other</code> 특성을 사용하여 쉼표를 구분자로 지정하십시오.
<code>delimit_other</code>	<i>flag</i>	항상 <code>other</code> 특성을 사용하여 사용자 정의 구분자를 지정할 수 있습니다.
<code>other</code>	<i>string</i>	<code>delimit_other</code> 가 <i>true</i> 일 때 사용하는 구분자를 지정합니다.
<code>decimal_symbol</code>	Default Comma Period	데이터 소스에서 사용하는 소수점 구분자를 지정합니다.
<code>multi_blank</code>	<i>flag</i>	여러 개의 인접한 공백 구분 문자를 하나의 구분자로 처리합니다.
<code>read_field_names</code>	<i>flag</i>	데이터 파일의 첫 번째 행을 열의 레이블로 사용합니다.
<code>strip_spaces</code>	None Left Right Both	가져올 때 문자열의 선행 및 후미 공백을 삭제합니다.
<code>invalid_char_mode</code>	Discard Replace	데이터 입력에서 유효하지 않은 문자(널, 0 또는 현재 인코딩에 존재하지 않는 모든 문자)를 제거하거나 유효하지 않은 문자를 지정된 한 문자 기호로 바꿉니다.
<code>invalid_char_replacement</code>	<i>string</i>	
<code>break_case_by_newline</code>	<i>flag</i>	행 구분자가 줄 바꾸기 문자이도록 지정합니다.
<code>lines_to_scan</code>	<i>number</i>	지정된 데이터 유형에 대해 스캔할 행 수를 지정합니다.
<code>auto_recognize_datetime</code>	<i>flag</i>	날짜 또는 시간이 소스 데이터에서 자동으로 식별되는지 여부를 지정합니다.
<code>quotes_1</code>	Discard PairAndDiscard IncludeAsText	가져오기 시에 작은따옴표의 처리 방법을 지정합니다.
<code>quotes_2</code>	Discard PairAndDiscard IncludeAsText	가져오기 시에 큰따옴표의 처리 방법을 지정합니다.
<code>full_filename</code>	<i>string</i>	디렉토리를 포함하여, 읽을 파일의 전체 이름입니다.
<code>use_custom_values</code>	<i>flag</i>	

표 58. variablefilenode 특성 (계속).

variablefilenode 특성	데이터 유형	특성 설명
custom_storage	Unknown String Integer Real Time Date Timestamp	
custom_date_format	"DDMMYY" "MMDDYY" "YYMMDD" "YYYYMMDD" "YYYYDDD" DAY MONTH "DD-MM-YY" "DD-MM-YYYY" "MM-DD-YY" "MM-DD-YYYY" "DD-MON-YY" "DD-MON-YYYY" "YYYY-MM-DD" "DD.MM.YY" "DD.MM.YYYY" "MM.DD.YY" "MM.DD.YYYY" "DD.MON.YY" "DD.MON.YYYY" "DD/MM/YY" "DD/MM/YYYY" "MM/DD/YY" "MM/DD/YYYY" "DD/MON/YY" "DD/MON/YYYY" MON YYYY q Q YYYY ww WK YYYY	사용자 정의 저장 공간이 지정된 경우에만 적용 가능합니다.

표 58. *variablefilenode* 특성 (계속).

variablefilenode 특성	데이터 유형	특성 설명
custom_time_format	"HHMMSS" "HHMM" "MMSS" "HH:MM:SS" "HH:MM" "MM:SS" "(H)H:(M)M:(S)S" "(H)H:(M)M" "(M)M:(S)S" "HH.MM.SS" "HH.MM" "MM.SS" "(H)H.(M)M.(S)S" "(H)H.(M)M" "(M)M.(S)S"	사용자 정의 저장 공간이 지정된 경우에만 적용 가능합니다.
custom_decimal_symbol	<i>field</i>	사용자 정의 저장 공간이 지정된 경우에만 적용 가능합니다.
encoding	StreamDefault SystemDefault "UTF-8"	텍스트 인코딩 방법을 지정합니다.

xmlimportnode 특성



XML 소스 노드는 XML 형식의 데이터를 스트림으로 가져옵니다. 단일 파일 또는 디렉토리의 모든 파일을 가져올 수 있습니다. 선택적으로 XML 구조를 읽을 스키마 파일을 지정할 수 있습니다.

예제

```
node = stream.create("xmlimport", "My node")
node.setPropertyValue("full_filename", "c:/import/ebooks.xml")
node.setPropertyValue("records", "/author/name")
```

표 59. *xmlimportnode* 특성.

xmlimportnode 특성	데이터 유형	특성 설명
read	single directory	단일 데이터 파일(기본값) 또는 디렉토리의 모든 XML 파일을 읽습니다.
recurse	<i>flag</i>	지정된 디렉토리의 모든 서브디렉토리로부터 추가로 XML 파일을 읽을지 여부를 지정합니다.
full_filename	<i>string</i>	(필수) 가져올 XML 파일의 전체 경로와 파일 이름(read = single인 경우).

표 59. *xmlimportnode* 특성 (계속).

xmlimportnode 특성	데이터 유형	특성 설명
directory_name	string	(필수) XML 파일을 가져올 디렉토리의 전체 경로 및 이름(read = directory인 경우).
full_schema_filename	string	XML 구조를 읽을 XSD 또는 DTD 파일의 전체 경로 및 파일 이름입니다. 이 모수를 생략하는 경우 XML 소스 파일에서 구조를 읽습니다.
records	string	레코드 경계를 정의할 XPath 표현식(예: /author/name)입니다. 소스 파일에서 이 요소를 만날 때마다 새 레코드가 작성됩니다.
mode	read specify	모든 데이터를 읽거나(기본값), 읽을 항목을 지정하십시오.
fields		가져올 항목(요소 및 속성)의 목록입니다. 목록의 각 항목은 XPath 표현식입니다.

dataviewimport 특성



데이터 보기 노드는 데이터 보기 데이터를 IBM SPSS Modeler로 가져옵니다.

예제

```
stream = modeler.script.stream()

dvnode = stream.createAt("dataviewimport", "Data View", 96, 96)
dvnode.setPropertyValue("analytic_data_source",
["", "/folder/adv", "LATEST"])
dvnode.setPropertyValue("table_name", ["", "com.ibm.spss.Table"])
dvnode.setPropertyValue("data_access_plan",
["", "DataAccessPlan"])
dvnode.setPropertyValue("optional_attributes",
[["", "NewDerivedAttribute"]])
dvnode.setPropertyValue("include_xml", True)
dvnode.setPropertyValue("include_xml_field", "xml_data")
```

표 60. *dataviewimport* 특성

dataviewimport 특성	데이터 유형	특성 설명
analytic_data_source	string	IBM SPSS Collaboration and Deployment Services에 저장된 분석 데이터 보기 오브젝트입니다. 사용할 버전의 경로 이름과 버전 레이블입니다. ["Object ID", "Full path", "Version"]

표 60. *dataviewimport* 특성 (계속)

dataviewimport 특성	데이터 유형	특성 설명
table_name	<i>string</i>	분석 데이터 보기에서 사용하는 데이터 보기 테이블 이름입니다. 테이블 이름은 패키지 규정되어야 합니다. IBM SPSS Collaboration and Deployment Services Deployment Manager 클라이언트에서 BOM을 내보내고 내보낸 zip 아카이브에서 default.bom 파일을 찾아서 패키지를 얻을 수 있습니다. BOM을 IBM Operational Decision Management(iLOG)에서 가져온 것이 아니면 패키지 이름은 항상 같아야 합니다. ["Object ID", "Name"]
data_access_plan	<i>string</i>	분석 데이터 보기를 위한 데이터를 제공하는 데 사용하는 데이터 액세스 계획입니다. ["Object ID","Name"]
optional_attributes	<i>string</i>	포함할 파생 속성의 목록입니다. [["ID1","Name1"], ["ID2", "Name2"]]
include_xml	<i>boolean</i>	XOM 인스턴스 데이터를 갖는 필드가 포함될 경우 True입니다. IBM Analytical Decision Management iLOG 노드를 사용하는 경우가 아니면 권장 설정은 false입니다. 이것을 켜면 많은 추가 처리가 추가될 수 있습니다.
include_xml_field	<i>string</i>	include_xml이 true로 설정될 때 추가할 필드의 이름입니다.

제 10 장 레코드 작업 노드 특성

appendnode 특성



붙여쓰기 노드는 레코드 세트를 연결합니다. 비슷한 구조를 갖지만 상이한 데이터를 갖는 데이터 세트 결합에 유용합니다.

예제

```
node = stream.create("append", "My node")
node.setPropertyValue("match_by", "Name")
node.setPropertyValue("match_case", True)
node.setPropertyValue("include_fields_from", "All")
node.setPropertyValue("create_tag_field", True)
node.setPropertyValue("tag_field_name", "Append_Flag")
```

표 61. *appendnode* 특성.

appendnode 특성	데이터 유형	특성 설명
match_by	Position Name	주 데이터 소스에서 필드의 위치나 입력 데이터 세트에서 필드의 이름을 기반으로 데이터 세트를 추가할 수 있습니다.
match_case	<i>flag</i>	필드 이름을 매치할 때 대소문자 구분을 사용할 수 있습니다.
include_fields_from	Main All	
create_tag_field	<i>flag</i>	
tag_field_name	<i>string</i>	

aggregatenode 특성



통합 노드는 입력 레코드의 시퀀스를 요약되고 통합된 출력 레코드로 대체합니다.

예제

```
node = stream.create("aggregate", "My node")
# dbnode is a configured database import node
stream.link(dbnode, node)
node.setPropertyValue("contiguous", True)
```

```

node.setPropertyValue("keys", ["Drug"])
node.setKeyedPropertyValue("aggregates", "Age", ["Sum", "Mean"])
node.setPropertyValue("inc_record_count", True)
node.setPropertyValue("count_field", "index")
node.setPropertyValue("extension", "Aggregated_")
node.setPropertyValue("add_as", "Prefix")

```

표 62. *aggregatenode* 특성.

aggregatenode 특성	데이터 유형	특성 설명
keys	<i>list</i>	통합을 위한 키로 사용할 수 있는 필드를 나열합니다. 예를 들어 Sex 및 Region이 키 필드인 경우, N과 S를 갖는 M과 F의 각각의 고유한 조합(4가지 고유 조합)이 통합 레코드를 갖습니다.
contiguous	<i>flag</i>	동일한 키 값을 갖는 모든 레코드가 입력에서 함께 그룹화됨(예를 들어, 입력이 키 필드에서 정렬됨)을 아는 경우 이 옵션을 선택하십시오. 그렇게 하면 성능이 개선될 수 있습니다.
aggregates		해당하는 값이 통합되는 숫자 필드뿐 아니라 선택된 통합 모드를 나열하는 구조화된 특성입니다.
aggregate_exprs		집계식의 파생 필드 이름을 계산하는 데 사용하는 키가 포함된 특성입니다. 예를 들어, <code>aggregatenode.setKeyedPropertyValue("aggregate_exprs", "Na_MAX", "MAX('Na')")</code>
extension	<i>string</i>	중복 통합 필드에 대한 접두문자 또는 접미문자를 지정하십시오(아래 표본).
add_as	Suffix Prefix	
inc_record_count	<i>flag</i>	각 통합 레코드를 형성하기 위해 통합되는 입력 레코드 수를 지정하는 여분의 필드를 작성합니다.
count_field	<i>string</i>	레코드 개수 필드의 이름을 지정합니다.
allow_approximation	부울	Analytic Server에서 통합이 수행될 때 순서 통계량의 근사를 허용합니다.
bin_count	<i>integer</i>	근사에서 사용할 구간 수를 지정합니다.

balancenode 특성



균형 노드는 데이터 세트의 불균형을 정정하므로, 데이터 세트가 지정된 조건을 준수합니다. 균형 지시문이 조건이 지정된 요인만큼 참인 레코드 비율을 조정합니다.

예제

```

node = stream.create("balance", "My node")
node.setPropertyValue("training_data_only", True)
node.setPropertyValue("directives", [[1.3, "Age > 60"], [1.5, "Na > 0.5"]])

```

표 63. *balancenode* 특성.

balancenode 특성	데이터 유형	특성 설명
directives		지정된 숫자를 기반으로 필드 값의 비율을 균형 잡기 위한 구조화된 특성입니다(아래 예제 참조).
training_data_only	flag	학습 데이터만 균형이 이루어져야 함을 지정합니다. 스트림에 파티션 필드가 없는 경우 이 옵션은 무시됩니다.

이 노드 특성은 다음 형식을 사용합니다.

`[[number, string] \ [number, string] \ ... [number, string]]`.

참고: 문자열이 (큰따옴표를 사용하여) 표현식에 임베드되는 경우 그 앞에 이스케이프 문자 " \ "가 선행되어야 합니다. " \ " 문자는 행 연속 문자이기도 하며, 명료성을 위해 인수를 정렬하는 데 사용할 수 있습니다.

derive_stbnode 특성



Space-Time-Boxes 노드는 위도, 경도, 시간소인 필드로부터 Space-Time-Boxes를 파생시킵니다. 또한 빈번한 Space-Time-Boxes를 단골집으로 식별할 수도 있습니다.

예제

```
node = modeler.script.stream().createAt("derive_stb", "My node", 96, 96)

# Individual Records mode
node.setPropertyValue("mode", "IndividualRecords")
node.setPropertyValue("latitude_field", "Latitude")
node.setPropertyValue("longitude_field", "Longitude")
node.setPropertyValue("timestamp_field", "OccurredAt")
node.setPropertyValue("densities", ["STB_GH7_1HOUR", "STB_GH7_30MINS"])
node.setPropertyValue("add_extension_as", "Prefix")
node.setPropertyValue("name_extension", "stb_")

# Hangouts mode
node.setPropertyValue("mode", "Hangouts")
node.setPropertyValue("hangout_density", "STB_GH7_30MINS")
node.setPropertyValue("id_field", "Event")
node.setPropertyValue("qualifying_duration", "30MINUTES")
node.setPropertyValue("min_events", 4)
node.setPropertyValue("qualifying_pct", 65)
```

표 64. *Space-Time-Boxes* 노드 특성

derive_stbnode 특성	데이터 유형	특성 설명
mode	IndividualRecords Hangouts	

표 64. Space-Time-Boxes 노드 특성 (계속)

derive_stbnode 특성	데이터 유형	특성 설명
latitude_field	field	
longitude_field	field	
timestamp_field	field	
hangout_density	density	단일 밀도입니다. 유효한 밀도 값에 대해서는 densities를 참조하십시오.
densities	[density,density,..., density]	<p>각 밀도는 문자열입니다(예: STB_GH8_1DAY). 참고: 유효한 밀도에 대한 한계가 있습니다. geohash의 경우 GH1부터 GH15까지의 값을 사용할 수 있습니다. 시간 파트의 경우 다음 값을 사용할 수 있습니다.</p> <p>EVER 1YEAR 1MONTH 1DAY 12HOURS 8HOURS 6HOURS 4HOURS 3HOURS 2HOURS 1HOUR 30MINS 15MINS 10MINS 5MINS 2MINS 1MIN 30SECS 15SECS 10SECS 5SECS 2SECS 1SEC</p>
id_field	field	
qualifying_duration	1DAY 12HOURS 8HOURS 6HOURS 4HOURS 3HOURS 2Hours 1HOUR 30MIN 15MIN 10MIN 5MIN 2MIN 1MIN 30SECS 15SECS 10SECS 5SECS 2SECS 1SECS	문자열이어야 합니다.
min_events	integer	최소 유효 정수 값은 2입니다.

표 64. Space-Time-Boxes 노드 특성 (계속)

derive_stbnode 특성	데이터 유형	특성 설명
qualifying_pct	integer	1과 100의 범위에 있어야 합니다.
add_extension_as	Prefix Suffix	
name_extension	string	

distinctnode 특성



고유 노드는 첫 번째 고유 레코드를 데이터 스트림으로 전달하거나 첫 번째 레코드를 삭제하고 대신 모든 중복을 데이터 스트림으로 전달하여 중복 레코드를 제거합니다.

예제

```
node = stream.create("distinct", "My node")
node.setPropertyValue("mode", "Include")
node.setPropertyValue("fields", ["Age" "Sex"])
node.setPropertyValue("keys_pre_sorted", True)
```

표 65. distinctnode 특성.

distinctnode 특성	데이터 유형	특성 설명
mode	Include Discard	첫 번째 고유 레코드를 데이터 스트림에 포함하거나 첫 번째 고유 레코드를 삭제하고 대신 모든 중복 레코드를 데이터 스트림으로 전달할 수 있습니다.
grouping_fields	list	레코드가 동일한지 여부를 판별하는 데 사용하는 필드를 나열합니다. 참고: 이 특성은 IBM SPSS Modeler 16부터 더 이상 사용하지 않습니다.
composite_value	구조화된 슬롯	아래 예를 참조하십시오.
composite_values	구조화된 슬롯	아래 예를 참조하십시오.
inc_record_count	flag	각 통합 레코드를 형성하기 위해 통합되는 입력 레코드 수를 지정하는 여분의 필드를 작성합니다.
count_field	string	레코드 개수 필드의 이름을 지정합니다.
sort_keys	구조화된 슬롯.	참고: 이 특성은 IBM SPSS Modeler 16부터 더 이상 사용하지 않습니다.
default_ascending	flag	
low_distinct_key_count	flag	작은 수의 레코드 및/또는 키 필드의 작은 수의 고유 값만을 갖고 있음을 지정합니다.
keys_pre_sorted	flag	동일한 키 값을 갖는 모든 레코드가 입력에서 함께 그룹화되도록 지정합니다.
disable_sql_generation	flag	

composite_value 특성의 예

composite_value 특성은 다음과 같은 일반 양식을 갖습니다.

```
node.setKeyedPropertyValue("composite_value", FIELD, FILLOPTION)
```

FILLOPTION의 양식은 다음과 같습니다. [FillType, Option1, Option2, ...].

예:

```
node.setKeyedPropertyValue("composite_value", "Age", ["First"])
node.setKeyedPropertyValue("composite_value", "Age", ["last"])
node.setKeyedPropertyValue("composite_value", "Age", ["Total"])
node.setKeyedPropertyValue("composite_value", "Age", ["Average"])
node.setKeyedPropertyValue("composite_value", "Age", ["Min"])
node.setKeyedPropertyValue("composite_value", "Age", ["Max"])
node.setKeyedPropertyValue("composite_value", "Date", ["Earliest"])
node.setKeyedPropertyValue("composite_value", "Date", ["Latest"])
node.setKeyedPropertyValue("composite_value", "Code", ["FirstAlpha"])
node.setKeyedPropertyValue("composite_value", "Code", ["LastAlpha"])
```

사용자 정의 옵션은 둘 이상의 인수가 필요하며, 이들은 목록으로 추가됩니다. 예:

```
node.setKeyedPropertyValue("composite_value", "Name", ["MostFrequent", "FirstRecord"])
node.setKeyedPropertyValue("composite_value", "Date", ["LeastFrequent", "LastRecord"])
node.setKeyedPropertyValue("composite_value", "Pending", ["IncludesValue", "T", "F"])
node.setKeyedPropertyValue("composite_value", "Marital", ["FirstMatch", "Married", "Divorced", "Separated"])
node.setKeyedPropertyValue("composite_value", "Code", ["Concatenate"])
node.setKeyedPropertyValue("composite_value", "Code", ["Concatenate", "Space"])
node.setKeyedPropertyValue("composite_value", "Code", ["Concatenate", "Comma"])
node.setKeyedPropertyValue("composite_value", "Code", ["Concatenate", "UnderScore"])
```

composite_values 특성의 예

composite_values 특성은 다음과 같은 일반 양식을 갖습니다.

```
node.setPropertyValue("composite_values", [
    [FIELD1, [FILLOPTION1]],
    [FIELD2, [FILLOPTION2]],
    .
    .
])
```

예:

```
node.setPropertyValue("composite_values", [
    ["Age", ["First"]],
    ["Name", ["MostFrequent", "First"]],
    ["Pending", ["IncludesValue", "T"]],
    ["Marital", ["FirstMatch", "Married", "Divorced", "Separated"]],
    ["Code", ["Concatenate", "Comma"]]
])
```

extensionprocessnode 특성



확장 변환 노드를 사용하면 스트림의 데이터를 사용하고 R 스크립팅 또는 Python for Spark 스크립트를 사용하여 데이터에 변환을 적용할 수 있습니다.

Python for Spark 예제

```
#### script example for Python for Spark
import modeler.api
stream = modeler.script.stream()
node = stream.create("extension_process", "extension_process")
node.setPropertyValue("syntax_type", "Python")

process_script = """
import spss.pyspark.runtime
from pyspark.sql.types import *

cxt = spss.pyspark.runtime.getContext()

if cxt.isComputeDataModelOnly():
    _schema = StructType([StructField("Age", LongType(), nullable=True), \
                            StructField("Sex", StringType(), nullable=True), \
                            StructField("BP", StringType(), nullable=True), \
                            StructField("Na", DoubleType(), nullable=True), \
                            StructField("K", DoubleType(), nullable=True), \
                            StructField("Drug", StringType(), nullable=True)])
    cxt.setSparkOutputSchema(_schema)
else:
    df = cxt.getSparkInputData()
    print df.dtypes[:]
    _newDF = df.select("Age","Sex","BP","Na","K","Drug")
    print _newDF.dtypes[:]
    cxt.setSparkOutputData(_newDF)
"""

node.setPropertyValue("python_syntax", process_script)
```

R 예제

```
#### script example for R
node.setPropertyValue("syntax_type", "R")
node.setPropertyValue("r_syntax", """"day<-as.Date(modelerData$dob, format="%Y-%m-%d")
next_day<-day + 1
modelerData<-cbind(modelerData,next_day)
var1<-c(fieldName="Next day",fieldLabel="",fieldStorage="date",fieldMeasure="",fieldFormat="",
fieldRole="")
modelerDataModel<-data.frame(modelerDataModel,var1)""")
```

표 66. *extensionprocessnode* 특성

extensionprocessnode 특성	데이터 유형	특성 설명
<code>syntax_type</code>	<i>R</i> <i>Python</i>	실행할 스크립트, R 또는 Python을 지정하십시오(R이 기본값).
<code>r_syntax</code>	<i>string</i>	실행할 R 스크립팅 구문입니다.
<code>python_syntax</code>	<i>string</i>	실행할 Python 스크립팅 구문입니다.
<code>use_batch_size</code>	<i>flag</i>	배치 처리 사용을 가능하게 합니다.
<code>batch_size</code>	<i>integer</i>	각 배치에 포함할 데이터 레코드의 수를 지정하십시오.
<code>convert_flags</code>	StringsAndDoubles LogicalValues	플래그 필드를 변환하는 옵션.
<code>convert_missing</code>	<i>flag</i>	결측값을 R NA 값으로 변환하는 옵션입니다.
<code>convert_datetime</code>	<i>flag</i>	날짜 또는 날짜/시간 형식을 갖는 변수를 R 날짜/시간 형식으로 변환하는 옵션.
<code>convert_datetime_class</code>	POSIXct POSIXlt	날짜 또는 날짜/시간 형식을 갖는 변수를 변환할 형식을 지정하는 옵션.

mergenode 특성



병합 노드는 다중 입력 레코드를 취하고 입력 필드의 일부 또는 모두를 포함하는 단일 출력 레코드를 작성합니다. 내부 고객 데이터 및 구매한 인구통계학적 데이터 같은 상이한 소스의 데이터 병합에 유용합니다.

예제

```
node = stream.create("merge", "My node")
# assume customerdata and salesdata are configured database import nodes
stream.link(customerdata, node)
stream.link(salesdata, node)
node.setPropertyValue("method", "Keys")
node.setPropertyValue("key_fields", ["id"])
node.setPropertyValue("common_keys", True)
node.setPropertyValue("join", "PartialOuter")
node.setKeyedPropertyValue("outer_join_tag", "2", True)
node.setKeyedPropertyValue("outer_join_tag", "4", True)
node.setPropertyValue("single_large_input", True)
node.setPropertyValue("single_large_input_tag", "2")
node.setPropertyValue("use_existing_sort_keys", True)
node.setPropertyValue("existing_sort_keys", [["id", "Ascending"]])
```


표 67. mergenode 특성.

mergenode 특성	데이터 유형	특성 설명
method	Order Keys Condition Rankedcondition	레코드가 데이터 파일에서 나열되는 순서로 병합되는지 여부, 하나 이상의 키 필드를 키 필드에 동일한 값을 갖는 레코드를 병합하는 데 사용할지 여부, 지정된 조건이 충족되면 레코드가 병합되는지 여부, 1차 및 모든 2차 데이터 세트에서 쌍을 이루는 각 행이 병합되는지 여부를 지정하십시오. 순위화 표현식을 사용하여 모든 다중 매치를 낮은 것부터 높은 것으로 차례로 정렬합니다.
condition	string	method가 Condition으로 설정되면 레코드 포함 또는 제외를 위한 조건을 지정합니다.
key_fields	list	
common_keys	flag	
join	Inner FullOuter PartialOuter Anti	
outer_join_tag.n	flag	이 특성에서 <i>n</i> 은 데이터 세트 선택 대화 상자에 표시되는 태그 이름입니다. 임의의 수의 데이터 세트가 불완전한 레코드를 기여할 수 있으므로 다중 태그 이름이 지정될 수 있습니다.
single_large_input	flag	다른 입력에 비교하여 상대적으로 큰 하나의 입력을 갖는 최적화의 사용 여부를 지정합니다.
single_large_input_tag	string	대형 데이터 세트 선택 대화 상자에 표시되는 태그 이름을 지정합니다. 입력 데이터 세트를 하나만 지정할 수 있기 때문에 이 특성의 사용법은 outer_join_tag 특성(플래그 대 문자열)과 약간 다를 수 있습니다.
use_existing_sort_keys	flag	입력이 이미 하나 이상의 키 필드에 의해 정렬되는지 여부를 지정합니다.
existing_sort_keys	[['string', 'Ascending'] \ \ ['string', 'Descending']]	이미 정렬된 필드 및 필드가 정렬되는 방향을 지정합니다.
primary_dataset	string	method가 Rankedcondition인 경우 병합에서 1차 데이터 세트를 선택하십시오. 이것은 외부 결합 병합의 왼쪽으로 간주될 수 있습니다.
rename_duplicate_fields	부울	method가 Rankedcondition이고 이것이 Y로 설정되는 경우, 결과로 생기는 병합된 데이터 세트가 다른 데이터 소스의 동일한 이름을 갖는 다중 필드를 포함하면 데이터 소스의 해당 태그가 필드 열 헤더의 시작에 추가됩니다.
merge_condition	string	
ranking_expression	string	
Num_matches	integer	merge_condition 및 ranking_expression을 기반으로 리턴될 매치 수입니다. 최소 1, 최대 100.

rfmaggregatenode 특성



RFM(Recency, Frequency, Monetary) 통합 노드를 사용하면 고객의 히스토리 트랜잭션 데이터를 취하고 모든 사용하지 않은 데이터를 제거하고 마지막으로 다룬 시기, 작성한 트랜잭션 수, 해당 트랜잭션의 구매총액을 나열하는 단일 행으로 모든 나머지 트랜잭션 데이터를 결합할 수 있습니다.

예제

```
node = stream.create("rfmaggregate", "My node")
node.setPropertyValue("relative_to", "Fixed")
node.setPropertyValue("reference_date", "2007-10-12")
node.setPropertyValue("id_field", "CardID")
node.setPropertyValue("date_field", "Date")
node.setPropertyValue("value_field", "Amount")
node.setPropertyValue("only_recent_transactions", True)
node.setPropertyValue("transaction_date_after", "2000-10-01")
```

표 68. rfmaggregatenode 특성.

rfmaggregatenode 특성	데이터 유형	특성 설명
relative_to	Fixed Today	트랜잭션의 최근성이 계산되는 날짜를 지정하십시오.
reference_date	date	relative_to에서 Fixed가 선택된 경우에만 사용 가능합니다.
contiguous	flag	동일한 ID를 갖는 모든 레코드가 데이터 스트림에 함께 나타나도록 데이터가 사전 정렬된 경우, 이 옵션을 선택하면 처리가 가속화됩니다.
id_field	field	고객 및 고객의 트랜잭션을 식별하기 위해 사용할 필드를 지정하십시오.
date_field	field	최근성을 계산하기 위해 사용할 날짜 필드를 지정하십시오.
value_field	field	구매총액 값을 계산하기 위해 사용할 필드를 지정하십시오.
extension	string	중복 통합 필드에 대한 접두문자 또는 접미문자를 지정하십시오.
add_as	Suffix Prefix	extension이 접미문자 또는 접두문자로 추가될지를 지정하십시오.
discard_low_value_records	flag	discard_records_below 설정 사용을 가능하게 합니다.
discard_records_below	number	RFM 총계를 계산할 때 그 아래의 모든 트랜잭션 세부사항을 사용하지 않는 최소값을 지정하십시오. 값의 단위는 선택된 value 필드와 관계가 있습니다.
only_recent_transactions	flag	specify_transaction_date 또는 transaction_within_last 설정 중 하나의 사용을 가능하게 합니다.
specify_transaction_date	flag	

표 68. *rfmaggregatenode* 특성 (계속).

rfmaggregatenode 특성	데이터 유형	특성 설명
transaction_date_after	date	specify_transaction_date가 선택된 경우에만 사용 가능합니다. 그 이후의 레코드가 분석에 포함될 트랜잭션 날짜를 지정하십시오.
transaction_within_last	number	transaction_within_last가 선택된 경우에만 사용 가능합니다. 그 이후의 레코드가 분석에 포함될 날짜에 상대적인 최근성 계산으로부터의 기간(일, 주, 월 또는 년)의 유형과 수를 지정하십시오.
transaction_scale	Days Weeks Months Years	transaction_within_last가 선택된 경우에만 사용 가능합니다. 그 이후의 레코드가 분석에 포함될 날짜에 상대적인 최근성 계산으로부터의 기간(일, 주, 월 또는 년)의 유형과 수를 지정하십시오.
save_r2	flag	각 고객에 대한 두 번째 최근 트랜잭션의 날짜를 표시합니다.
save_r3	flag	save_r2가 선택된 경우에만 사용 가능합니다. 각 고객에 대한 세 번째 최근 트랜잭션의 날짜를 표시합니다.

Rprocessnode 특성



R 변환 노드를 사용하면 IBM(r) SPSS(r) Modeler 스트림에서 데이터를 가져와서 데이터를 사용자 고유의 R 스크립트로 수정할 수 있습니다. 데이터가 수정된 후에 스트림으로 리턴됩니다.

예제

```
node = stream.create("rprocess", "My node")
node.setPropertyValue("custom_name", "my_node")
node.setPropertyValue("syntax", """day<-as.Date(modelerData$dob, format="%Y-%m-%d")
next_day<-day + 1
modelerData<-cbind(modelerData,next_day)
var1<-c(fieldName="Next day",fieldLabel="",fieldStorage="date",fieldMeasure="",fieldFormat="",
fieldRole="")
modelerDataModel<-data.frame(modelerDataModel,var1)""")
node.setPropertyValue("convert_datetime", "POSIXct")
```

표 69. *Rprocessnode* 특성.

Rprocessnode 특성	데이터 유형	특성 설명
syntax	string	
convert_flags	StringsAndDoubles LogicalValues	
convert_datetime	flag	
convert_datetime_class	POSIXct POSIXlt	
convert_missing	flag	

표 69. Rprocessnode 특성 (계속).

Rprocessnode 특성	데이터 유형	특성 설명
use_batch_size	flag	배치 처리 사용을 가능하게 합니다.
batch_size	integer	각 배치에 포함할 데이터 레코드의 수를 지정하십시오.

samplenode 특성



표본 노드는 레코드의 서브세트를 선택합니다. 층화, 수평배열, 비임의(구조화) 표본을 포함하여 다양한 표본 유형이 지원됩니다. 표본추출은 성능을 개선하고 분석을 위해 관련 레코드나 트랜잭션 집단을 선택하는 데 유용할 수 있습니다.

예제

```
/* Create two Sample nodes to extract
different samples from the same data */
```

```
node = stream.create("sample", "My node")
node.setPropertyValue("method", "Simple")
node.setPropertyValue("mode", "Include")
node.setPropertyValue("sample_type", "First")
node.setPropertyValue("first_n", 500)
```

```
node = stream.create("sample", "My node")
node.setPropertyValue("method", "Complex")
node.setPropertyValue("stratify_by", ["Sex", "Cholesterol"])
node.setPropertyValue("sample_units", "Proportions")
node.setPropertyValue("sample_size_proportions", "Custom")
node.setPropertyValue("sizes_proportions", [{"M", "High", "Default"}, {"M", "Normal", "Default"}, {"F", "High", 0.3}, {"F", "Normal", 0.3}])
```

표 70. samplenode 특성.

samplenode 특성	데이터 유형	특성 설명
method	Simple Complex	
mode	Include Discard	지정된 조건을 만족하는 레코드를 포함 또는 삭제합니다.
sample_type	First OneInN RandomPct	표본추출 방법을 지정합니다.
first_n	integer	지정된 분리점 위치까지의 레코드가 포함되거나 삭제됩니다.
one_in_n	number	모든 n번째 레코드를 포함 또는 삭제합니다.
rand_pct	number	포함 또는 삭제할 레코드의 퍼센트를 지정하십시오.
use_max_size	flag	maximum_size 설정 사용을 가능하게 합니다.

표 70. *samplenode* 특성 (계속).

samplenode 특성	데이터 유형	특성 설명
maximum_size	<i>integer</i>	데이터 스트림에서 포함 또는 삭제될 가장 큰 표본을 지정하십시오. 이 옵션은 중복이므로 First 및 Include가 지정될 때는 사용하지 않습니다.
set_random_seed	<i>flag</i>	난수 시드 설정을 사용 가능하게 합니다.
random_seed	<i>integer</i>	난수 시드로 사용하는 값을 지정하십시오.
complex_sample_type	Random Systematic	
sample_units	Proportions Counts	
sample_size_proportions	Fixed Custom Variable	
sample_size_counts	Fixed Custom Variable	
fixed_proportions	<i>number</i>	
fixed_counts	<i>integer</i>	
variable_proportions	<i>field</i>	
variable_counts	<i>field</i>	
use_min_stratum_size	<i>flag</i>	
minimum_stratum_size	<i>integer</i>	이 옵션은 Sample units=Proportions를 사용하여 복합 표본을 사용할 때만 적용됩니다.
use_max_stratum_size	<i>flag</i>	
maximum_stratum_size	<i>integer</i>	이 옵션은 Sample units=Proportions를 사용하여 복합 표본을 사용할 때만 적용됩니다.
clusters	<i>field</i>	
stratify_by	[<i>field1 ... fieldN</i>]	
specify_input_weight	<i>flag</i>	
input_weight	<i>field</i>	
new_output_weight	<i>string</i>	
sizes_proportions	[[<i>string string value</i>][<i>string string value</i>]...]	sample_units=proportions 및 sample_size_proportions=Custom인 경우, 층화 필드 값의 가능한 각 조합에 대한 값을 지정합니다.
default_proportion	<i>number</i>	
sizes_counts	[[<i>string string value</i>][<i>string string value</i>]...]	층화 필드 값의 가능한 각 조합에 대한 값을 지정합니다. 사용법은 sizes_proportions와 유사하지만 비율 대신 정수를 지정합니다.
default_count	<i>number</i>	

selectnode 특성



선택 노드는 특정 조건을 기반으로 데이터 스트림에서 레코드의 서브셋을 선택 또는 삭제합니다. 예를 들어, 특정 영업 지역에 관련된 레코드를 선택할 수 있습니다.

예제

```
node = stream.create("select", "My node")
node.setPropertyValue("mode", "Include")
node.setPropertyValue("condition", "Age < 18")
```

표 71. *selectnode* 특성.

selectnode 특성	데이터 유형	특성 설명
mode	Include Discard	선택된 레코드를 포함 또는 삭제할지 여부를 지정합니다.
condition	string	레코드 포함 또는 삭제에 대한 조건입니다.

sortnode 특성



정렬 노드는 하나 이상의 필드의 값을 기반으로 레코드를 내림차순 또는 오름차순으로 정렬합니다.

예제

```
node = stream.create("sort", "My node")
node.setPropertyValue("keys", [["Age", "Ascending"], ["Sex", "Descending"]])
node.setPropertyValue("default_ascending", False)
node.setPropertyValue("use_existing_keys", True)
node.setPropertyValue("existing_keys", [["Age", "Ascending"]])
```

표 72. *sortnode* 특성.

sortnode 특성	데이터 유형	특성 설명
keys	list	정렬하려는 필드를 지정합니다. 방향이 지정되지 않는 경우 기본값을 사용합니다.
default_ascending	flag	기본 정렬 순서를 지정합니다.
use_existing_keys	flag	이미 정렬된 필드에 대한 이전 정렬 순서를 사용하여 정렬이 최적화되는지 여부를 지정합니다.
existing_keys		이미 정렬된 필드 및 필드가 정렬되는 방향을 지정합니다. keys 특성과 동일한 형식을 사용합니다.

spacetimeboxes 특성



STB(Space-Time-Box)는 Geohash 공간 위치의 확장입니다. 보다 상세하게 설명하자면 STB는 공간 및 시간을 정기적으로 모양으로 표시하는 알파뉴메릭 문자열입니다.

표 73. spacetimeboxes 특성

spacetimeboxes 특성	데이터 유형	특성 설명
mode	<i>IndividualRecords</i> <i>Hangouts</i>	
latitude_field	<i>field</i>	
longitude_field	<i>field</i>	
timestamp_field	<i>field</i>	
densities	<i>[density, density, density...]</i>	<p>각 밀도는 문자열입니다. 예: STB_GH8_1DAY</p> <p>유효한 밀도에 대한 한계가 있습니다.</p> <p>geohash의 경우 GH1-GH15의 값을 사용할 수 있습니다.</p> <p>시간 파트의 경우 다음 값을 사용할 수 있습니다.</p> <p>EVER 1YEAR 1MONTH 1DAY 12HOURS 8HOURS 6HOURS 4HOURS 3HOURS 2HOURS 1HOUR 30MINS 15MINS 10MINS 5MINS 2MINS 1MIN 30SECS 15SECS 10SECS 5SECS 2SECS 1SEC</p>
field_name_extension	<i>string</i>	
add_extension_as	<i>Prefix</i> <i>Suffix</i>	
hangout_density	<i>density</i>	단일 밀도(위의 내용 참조)
id_field	<i>field</i>	

표 73. *spacetimeboxes* 특성 (계속)

spacetimeboxes 특성	데이터 유형	특성 설명
qualifying_duration	1DAY 12HOURS 8HOURS 6HOURS 4HOURS 2HOURS 1HOUR 30MIN 15MIN 10MIN 5MIN 2MIN 1MIN 30SECS 15SECS 10SECS 5SECS 2SECS 1SECS	문자열이어야 합니다.
min_events	<i>integer</i>	최소값은 2입니다.
qualifying_pct	<i>integer</i>	1-100의 범위여야 합니다.

streamingtimeseries 특성



스트리밍 시계열 노드는 한 단계로 시계열 모델을 작성하고 스코어링합니다.

참고: 이 스트리밍 시계열 노드는 SPSS Modeler의 버전 18에서 더 이상 사용되지 않는 원래의 스트리밍 시계열 노드를 대체합니다.

표 74. *streamingtimeseries* 특성

streamingtimeseries 특성	값	특성 설명
targets	<i>field</i>	스트리밍 시계열 노드는 선택적으로 하나 이상의 입력 필드를 예측자로 사용하여 하나 이상의 목표를 예측합니다. 빈도 및 가중 필드는 사용하지 않습니다. 자세한 정보는 191 페이지의 『공통 모델링 노드 특성』의 내용을 참조하십시오.
candidate_inputs	[<i>field1 ... fieldN</i>]	모델에서 사용하는 입력 또는 예측변수 필드입니다.
use_period	<i>flag</i>	
date_time_field	<i>field</i>	

표 74. *streamingtimeseries* 특성 (계속)

streamingtimeseries 특성	값	특성 설명
input_interval	None Unknown Year Quarter Month Week Day Hour Hour_nonperiod Minute Minute_nonperiod Second Second_nonperiod	
period_field	<i>field</i>	
period_start_value	<i>integer</i>	
num_days_per_week	<i>integer</i>	
start_day_of_week	Sunday Monday Tuesday Wednesday Thursday Friday Saturday	
num_hours_per_day	<i>integer</i>	
start_hour_of_day	<i>integer</i>	
timestamp_increments	<i>integer</i>	
cyclic_increments	<i>integer</i>	
cyclic_periods	<i>list</i>	
output_interval	None Year Quarter Month Week Day Hour Minute Second	
is_same_interval	<i>flag</i>	
cross_hour	<i>flag</i>	
aggregate_and_distribute	<i>list</i>	

표 74. *streamingtimeseries* 특성 (계속)

streamingtimeseries 특성	값	특성 설명
aggregate_default	Mean Sum Mode Min Max	
distribute_default	Mean Sum	
group_default	Mean Sum Mode Min Max	
missing_imput	Linear_interp Series_mean K_mean K_median Linear_trend	
k_span_points	<i>integer</i>	
use_estimation_period	<i>flag</i>	
estimation_period	Observations Times	
date_estimation	<i>list</i>	date_time_field를 사용하는 경우에만 사용할 수 있습니다.
period_estimation	<i>list</i>	use_period를 사용하는 경우에만 사용할 수 있습니다.
observations_type	Latest Earliest	
observations_num	<i>integer</i>	
observations_exclude	<i>integer</i>	
method	ExpertModeler Exsmooth Arima	
expert_modeler_method	ExpertModeler Exsmooth Arima	
consider_seasonal	<i>flag</i>	
detect_outliers	<i>flag</i>	
expert_outlier_additive	<i>flag</i>	
expert_outlier_level_shift	<i>flag</i>	
expert_outlier_innovational	<i>flag</i>	
expert_outlier_level_shift	<i>flag</i>	
expert_outlier_transient	<i>flag</i>	

표 74. *streamingtimeseries* 특성 (계속)

streamingtimeseries 특성	값	특성 설명
expert_outlier_seasonal_additive	<i>flag</i>	
expert_outlier_local_trend	<i>flag</i>	
expert_outlier_additive_patch	<i>flag</i>	
consider_newesmodels	<i>flag</i>	
exsmooth_model_type	Simple HoltsLinearTrend BrownsLinearTrend DampedTrend SimpleSeasonal WintersAdditive WintersMultiplicative DampedTrendAdditive DampedTrendMultiplicative MultiplicativeTrendAdditive MultiplicativeSeasonal MultiplicativeTrend Multiplicative MultiplicativeTrend	
futureValue_type_method	Compute specify	
exsmooth_transformation_type	None SquareRoot NaturalLog	
arma.p	<i>integer</i>	
arma.d	<i>integer</i>	
arma.q	<i>integer</i>	
arma.sp	<i>integer</i>	
arma.sd	<i>integer</i>	
arma.sq	<i>integer</i>	
arma_transformation_type	None SquareRoot NaturalLog	
arma_include_constant	<i>flag</i>	
tf_arma.p. <i>fieldname</i>	<i>integer</i>	전이 함수용입니다.
tf_arma.d. <i>fieldname</i>	<i>integer</i>	전이 함수용입니다.
tf_arma.q. <i>fieldname</i>	<i>integer</i>	전이 함수용입니다.
tf_arma.sp. <i>fieldname</i>	<i>integer</i>	전이 함수용입니다.
tf_arma.sd. <i>fieldname</i>	<i>integer</i>	전이 함수용입니다.
tf_arma.sq. <i>fieldname</i>	<i>integer</i>	전이 함수용입니다.
tf_arma.delay. <i>fieldname</i>	<i>integer</i>	전이 함수용입니다.

표 74. *streamingtimeseries* 특성 (계속)

streamingtimeseries 특성	값	특성 설명
tf_arima.transformation_type. <i>fieldname</i>	None SquareRoot NaturalLog	전이 함수용입니다.
arima_detect_outliers	<i>flag</i>	
arima_outlier_additive	<i>flag</i>	
arima_outlier_level_shift	<i>flag</i>	
arima_outlier_innovational	<i>flag</i>	
arima_outlier_transient	<i>flag</i>	
arima_outlier_seasonal_additive	<i>flag</i>	
arima_outlier_local_trend	<i>flag</i>	
arima_outlier_additive_patch	<i>flag</i>	
conf_limit_pct	<i>real</i>	
events	<i>fields</i>	
forecastperiods	<i>integer</i>	
extend_records_into_future	<i>flag</i>	
conf_limits	<i>flag</i>	
noise_res	<i>flag</i>	

streamingts 특성(더 이상 사용되지 않음)



참고: 이 원본 스트리밍 시계열 노드는 SPSS Modeler 버전 18에서 더 이상 사용되지 않으며 IBM SPSS Analytic Server의 기능을 이용하여 빅 데이터를 처리하도록 디자인된 새 스트리밍 시계열 노드로 대체되었습니다. 스트리밍 TS 노드는 시간 구간 노드가 필요 없이 한 단계로 시계열 모델을 작성하고 스코어링합니다.

예제

```
node = stream.create("streamingts", "My node")
node.setPropertyValue("deployment_force_rebuild", True)
node.setPropertyValue("deployment_rebuild_mode", "Count")
node.setPropertyValue("deployment_rebuild_count", 3)
node.setPropertyValue("deployment_rebuild_pct", 11)
node.setPropertyValue("deployment_rebuild_field", "Year")
```

표 75. *streamingts* 특성.

streamingts 특성	데이터 유형	특성 설명
custom_fields	<i>flag</i>	custom_fields=false인 경우, 업스트림 유형 노드의 설정을 사용합니다. custom_fields=true인 경우 targets 및 inputs가 지정되어야 합니다.
targets	[<i>field1...fieldN</i>]	
inputs	[<i>field1...fieldN</i>]	

표 75. *streamingts* 특성 (계속).

streamingts 특성	데이터 유형	특성 설명
method	ExpertModeler Exsmooth Arima	
calculate_conf	<i>flag</i>	
conf_limit_pct	<i>real</i>	
use_time_intervals_node	<i>flag</i>	use_time_intervals_node=true인 경우, 업스트림 시간 구간 노드의 설정을 사용합니다. use_time_intervals_node=false인 경우 interval_offset_position, interval_offset, interval_type이 지정되어야 합니다.
interval_offset_position	LastObservation LastRecord	LastObservation은 마지막 유효한 관측값을 의미합니다. LastRecord는 마지막 레코드에서 뒤로 계수를 의미합니다.
interval_offset	<i>number</i>	
interval_type	Periods Years Quarters Months WeeksNonPeriodic DaysNonPeriodic HoursNonPeriodic MinutesNonPeriodic SecondsNonPeriodic	
events	<i>fields</i>	
expert_modeler_method	AllModels Exsmooth Arima	
consider_seasonal	<i>flag</i>	
detect_outliers	<i>flag</i>	
expert_outlier_additive	<i>flag</i>	
expert_outlier_level_shift	<i>flag</i>	
expert_outlier_innovational	<i>flag</i>	
expert_outlier_transient	<i>flag</i>	
expert_outlier_seasonal_additive	<i>flag</i>	
expert_outlier_local_trend	<i>flag</i>	
expert_outlier_additive_patch	<i>flag</i>	
exsmooth_model_type	Simple HoltsLinearTrend BrownsLinearTrend DampedTrend SimpleSeasonal WintersAdditive WintersMultiplicative	
exsmooth_transformation_type	None SquareRoot NaturalLog	
arima_p	<i>integer</i>	시계열 모델링 노드의 경우와 같은 특성
arima_d	<i>integer</i>	시계열 모델링 노드의 경우와 같은 특성

표 75. *streamingts* 특성 (계속).

streamingts 특성	데이터 유형	특성 설명
<i>arma_q</i>	<i>integer</i>	시계열 모델링 노드의 경우와 같은 특성
<i>arma_sp</i>	<i>integer</i>	시계열 모델링 노드의 경우와 같은 특성
<i>arma_sd</i>	<i>integer</i>	시계열 모델링 노드의 경우와 같은 특성
<i>arma_sq</i>	<i>integer</i>	시계열 모델링 노드의 경우와 같은 특성
<i>arma_transformation_type</i>	None SquareRoot NaturalLog	시계열 모델링 노드의 경우와 같은 특성
<i>arma_include_constant</i>	<i>flag</i>	시계열 모델링 노드의 경우와 같은 특성
<i>tf_arma_p.fieldname</i>	<i>integer</i>	시계열 모델링 노드의 경우와 같은 특성입니다. 전이 함수용입니다.
<i>tf_arma_d.fieldname</i>	<i>integer</i>	시계열 모델링 노드의 경우와 같은 특성입니다. 전이 함수용입니다.
<i>tf_arma_q.fieldname</i>	<i>integer</i>	시계열 모델링 노드의 경우와 같은 특성입니다. 전이 함수용입니다.
<i>tf_arma_sp.fieldname</i>	<i>integer</i>	시계열 모델링 노드의 경우와 같은 특성입니다. 전이 함수용입니다.
<i>tf_arma_sd.fieldname</i>	<i>integer</i>	시계열 모델링 노드의 경우와 같은 특성입니다. 전이 함수용입니다.
<i>tf_arma_sq.fieldname</i>	<i>integer</i>	시계열 모델링 노드의 경우와 같은 특성입니다. 전이 함수용입니다.
<i>tf_arma_delay.fieldname</i>	<i>integer</i>	시계열 모델링 노드의 경우와 같은 특성입니다. 전이 함수용입니다.
<i>tf_arma_transformation_type.fieldname</i>	None SquareRoot NaturalLog	
<i>arma_detect_outlier_mode</i>	None Automatic	
<i>arma_outlier_additive</i>	<i>flag</i>	
<i>arma_outlier_level_shift</i>	<i>flag</i>	
<i>arma_outlier_innovational</i>	<i>flag</i>	
<i>arma_outlier_transient</i>	<i>flag</i>	
<i>arma_outlier_seasonal_additive</i>	<i>flag</i>	
<i>arma_outlier_local_trend</i>	<i>flag</i>	
<i>arma_outlier_additive_patch</i>	<i>flag</i>	
<i>deployment_force_rebuild</i>	<i>flag</i>	
<i>deployment_rebuild_mode</i>	Count Percent	
<i>deployment_rebuild_count</i>	<i>number</i>	
<i>deployment_rebuild_pct</i>	<i>number</i>	
<i>deployment_rebuild_field</i>	< <i>fieldname</i> >	

cplexnode 특성



CPLEX 최적화 노드에서는 OPL(Optimization Programming Language) 모델 파일을 통해 복잡한 수학(CPLEX) 기반 최적화를 수행할 수 있습니다. 이 기능은 IBM Analytical Decision Management 제품에서 제공하지만, 이제 IBM Analytical Decision Management 없이도 SPSS Modeler에서 CPLEX 노드를 사용할 수 있습니다.

CPLEX 최적화 및 OPL에 대한 자세한 정보는 IBM Analytical Decision Management 문서를 참조하십시오.

표 76. *cplexnode* 특성

cplexnode 특성	데이터 유형	특성 설명
opl_model_text	<i>string</i>	CPLEX 최적화 노드에서 실행된 다음 최적화 결과를 생성하는 OPL(Optimization Programming Language) 스크립트 프로그램입니다.
opl_tuple_set_name	<i>string</i>	수신 데이터에 해당하는 OPL 모델의 튜플 세트 이름입니다.
opl_data_text	<i>string</i>	OPL에 사용되는 일부 변수 또는 데이터에 대한 정의입니다.
output_value_mode	<i>string</i>	가능한 값은 raw 또는 dvar입니다. dvar을 지정할 경우 출력 탭에서 사용자가 출력을 위해 오브젝트 함수 변수 이름을 OPL로 지정해야 합니다. raw를 지정할 경우 이름에 관계없이 목적 함수가 바로 출력됩니다.
objective_function_value_fieldname	<i>string</i>	출력에 사용할 필드 이름입니다. 기본값은 _OBJECTIVE입니다.

제 11 장 필드 작업 노드 특성

anonymizenode 특성



값 익명화 노드는 필드 이름 및 값이 다운스트림으로 표시되는 방법을 변환하여 원 데이터를 위장합니다. 이것은 다른 사용자가 고객 이름이나 기타 세부사항 같은 민감한 데이터를 사용하여 모델을 작성하도록 허용하려는 경우에 유용할 수 있습니다.

예제

```
stream = modeler.script.stream()
varfilenode = stream.createAt("variablefile", "File", 96, 96)
varfilenode.setPropertyValue("full_filename", "$CLEO/DEMOS/DRUG1n")
node = stream.createAt("anonymize", "My node", 192, 96)
# Anonymize node requires the input fields while setting the values
stream.link(varfilenode, node)
node.setKeyedPropertyValue("enable_anonymize", "Age", True)
node.setKeyedPropertyValue("transformation", "Age", "Random")
node.setKeyedPropertyValue("set_random_seed", "Age", True)
node.setKeyedPropertyValue("random_seed", "Age", 123)
node.setKeyedPropertyValue("enable_anonymize", "Drug", True)
node.setKeyedPropertyValue("use_prefix", "Drug", True)
node.setKeyedPropertyValue("prefix", "Drug", "myprefix")
```

표 77. anonymizenode 특성

anonymizenode 특성	데이터 유형	특성 설명
enable_anonymize	flag	True로 설정되면 필드 값의 값 익명화를 활성화합니다(값 익명화 열에서 해당 필드에 대해 Yes 를 선택하는 것과 동등함).
use_prefix	flag	True로 설정되면 사용자 정의 접두문자가 지정되는 경우 사용합니다. Hash 메소드에 의해 값 익명화될 필드에 적용되며 해당 필드에 대한 값 바꾸기 대화 상자에서 사용자 정의 단일 선택 단추를 선택하는 것과 동등합니다.
prefix	string	값 바꾸기 대화 상자의 텍스트 상자에 접두문자를 입력하는 것과 동등합니다. 다른 것이 지정되지 않은 경우 기본 접두문자는 기본값입니다.
transformation	Random Fixed	Transform 메소드에 의해 익명화되는 필드에 대한 변환 모수가 변량 또는 고정인지 여부를 판별합니다.
set_random_seed	flag	True로 설정되면 지정된 시드 값을 사용합니다(transformation도 Random으로 설정된 경우).
random_seed	integer	set_random_seed가 True로 설정될 때 이것이 난수를 위한 시드입니다.
scale	number	transformation이 Fixed로 설정될 때, 이 값을 "척도 기준"에 사용합니다. 최대 척도 값은 보통 10이지만 오버플로우를 피하기 위해 줄일 수도 있습니다.

표 77. *anonymizenode* 특성 (계속)

anonymizenode 특성	데이터 유형	특성 설명
translate	<i>number</i>	transformation이 Fixed로 설정될 때, 이 값을 "변환"에 사용합니다. 최대 변환 값은 보통 1000이지만 오버플로우를 피하기 위해 줄일 수도 있습니다.

autodatapreprenode 특성



자동 데이터 준비(ADP) 노드는 데이터를 분석하고 수정사항을 식별하고, 문제가 있거나 유용할 것 같지 않은 필드를 제외시키고, 적절한 경우 새 속성을 파생시키고, 지능형 선별 및 표본추출 기법을 통해 성능을 개선할 수 있습니다. 완전 자동화된 방식으로 노드를 사용하여 노드가 수정사항을 선택하고 적용할 수 있게 하거나, 변경사항이 작성 및 승인되기 전에 변경을 미리보거나, 거부 또는 원하는 대로 개정할 수 있습니다.

예제

```
node = stream.create("autodataprep", "My node")
node.setPropertyValue("objective", "Balanced")
node.setPropertyValue("excluded_fields", "Filter")
node.setPropertyValue("prepare_dates_and_times", True)
node.setPropertyValue("compute_time_until_date", True)
node.setPropertyValue("reference_date", "Today")
node.setPropertyValue("units_for_date_durations", "Automatic")
```

표 78. *autodatapreprenode* 특성

autodatapreprenode 특성	데이터 유형	특성 설명
objective	Balanced Speed Accuracy Custom	
custom_fields	<i>flag</i>	True인 경우 현재 노드에 대한 목표, 입력 및 기타 필드를 지정할 수 있습니다. False인 경우 업스트림 유형 노드의 현재 설정을 사용합니다.
target	<i>field</i>	단일 목표 필드를 지정합니다.
inputs	[<i>field1</i> ... <i>fieldN</i>]	모델에서 사용하는 입력 또는 예측변수 필드입니다.
use_frequency	<i>flag</i>	
frequency_field	<i>field</i>	
use_weight	<i>flag</i>	
weight_field	<i>field</i>	
excluded_fields	Filter None	
if_fields_do_not_match	StopExecution ClearAnalysis	
prepare_dates_and_times	<i>flag</i>	모든 날짜 및 시간 필드에 대한 액세스를 제어합니다.

표 78. autodatapreprenode 특성 (계속)

autodatapreprenode 특성	데이터 유형	특성 설명
compute_time_until_date	<i>flag</i>	
reference_date	Today Fixed	
fixed_date	<i>date</i>	
units_for_date_durations	Automatic Fixed	
fixed_date_units	Years Months Days	
compute_time_until_time	<i>flag</i>	
reference_time	CurrentTime Fixed	
fixed_time	<i>time</i>	
units_for_time_durations	Automatic Fixed	
fixed_date_units	Hours Minutes Seconds	
extract_year_from_date	<i>flag</i>	
extract_month_from_date	<i>flag</i>	
extract_day_from_date	<i>flag</i>	
extract_hour_from_time	<i>flag</i>	
extract_minute_from_time	<i>flag</i>	
extract_second_from_time	<i>flag</i>	
exclude_low_quality_inputs	<i>flag</i>	
exclude_too_many_missing	<i>flag</i>	
maximum_percentage_missing	<i>number</i>	
exclude_too_many_categories	<i>flag</i>	
maximum_number_categories	<i>number</i>	
exclude_if_large_category	<i>flag</i>	
maximum_percentage_category	<i>number</i>	
prepare_inputs_and_target	<i>flag</i>	
adjust_type_inputs	<i>flag</i>	
adjust_type_target	<i>flag</i>	
reorder_nominal_inputs	<i>flag</i>	
reorder_nominal_target	<i>flag</i>	
replace_outliers_inputs	<i>flag</i>	
replace_outliers_target	<i>flag</i>	
replace_missing_continuous_inputs	<i>flag</i>	
replace_missing_continuous_target	<i>flag</i>	
replace_missing_nominal_inputs	<i>flag</i>	

표 78. autodatapreptime 특성 (계속)

autodatapreptime 특성	데이터 유형	특성 설명
replace_missing_nominal_target	flag	
replace_missing_ordinal_inputs	flag	
replace_missing_ordinal_target	flag	
maximum_values_for_ordinal	number	
minimum_values_for_continuous	number	
outlier_cutoff_value	number	
outlier_method	Replace Delete	
rescale_continuous_inputs	flag	
rescaling_method	MinMax ZScore	
min_max_minimum	number	
min_max_maximum	number	
z_score_final_mean	number	
z_score_final_sd	number	
rescale_continuous_target	flag	
target_final_mean	number	
target_final_sd	number	
transform_select_input_fields	flag	
maximize_association_with_target	flag	
p_value_for_merging	number	
merge_ordinal_features	flag	
merge_nominal_features	flag	
minimum_cases_in_category	number	
bin_continuous_fields	flag	
p_value_for_binning	number	
perform_feature_selection	flag	
p_value_for_selection	number	
perform_feature_construction	flag	
transformed_target_name_extension	string	
transformed_inputs_name_extension	string	
constructed_features_root_name	string	
years_duration_name_extension	string	
months_duration_name_extension	string	
days_duration_name_extension	string	
hours_duration_name_extension	string	
minutes_duration_name_extension	string	
seconds_duration_name_extension	string	
year_cyclical_name_extension	string	
month_cyclical_name_extension	string	

표 78. *autodatapreprenode* 특성 (계속)

autodatapreprenode 특성	데이터 유형	특성 설명
day_cyclical_name_extension	string	
hour_cyclical_name_extension	string	
minute_cyclical_name_extension	string	
second_cyclical_name_extension	string	

astimeintervalsnode 특성



구간을 지정하고 추정 또는 시계열 분석을 위한 새 시간 필드를 파생하려면 시간 구간 노드를 사용하십시오. 초부터 년까지, 모든 범위의 시간 구간이 지원됩니다.

표 79. *astimeintervalsnode* 특성

astimeintervalsnode 특성	데이터 유형	특성 설명
time_field	field	단일 연속형 필드만 승인할 수 있습니다. 해당 필드는 간격 변환을 위한 통합 키로서 노드가 사용됩니다. 여기에서 정수 필드를 사용하는 경우 시간 지수인 것으로 간주됩니다.
dimensions	[field1 field2 ... fieldn]	이들 필드는 필드 값을 기반으로 개별 시계열을 작성하는 데 사용됩니다.
fields_to_aggregate	[field1 field2 ... fieldn]	이들 필드는 시간 필드의 기간을 변경하는 작업의 일부로 통합됩니다. 이 선택도구에 포함되지 않는 모든 필드는 노드를 나가는 데이터에서 필터링됩니다.

binningnode 특성



구간화 노드는 하나 이상의 기존 연속형(숫자 범위) 필드의 값을 기반으로 새 명목형(세트) 필드를 자동으로 작성합니다. 예를 들어, 연속형 수입 필드를 평균값에서의 편차로서 수입 그룹을 포함하는 새 범주형 필드로 변환할 수 있습니다. 새 필드에 대한 구간을 작성한 후에는 절단점을 기반으로 파생 노드를 생성할 수 있습니다.

예제

```
node = stream.create("binning", "My node")
node.setPropertyValue("fields", ["Na", "K"])
node.setPropertyValue("method", "Rank")
node.setPropertyValue("fixed_width_name_extension", "_binned")
node.setPropertyValue("fixed_width_add_as", "Suffix")
```

```

node.setPropertyValue("fixed_bin_method", "Count")
node.setPropertyValue("fixed_bin_count", 10)
node.setPropertyValue("fixed_bin_width", 3.5)
node.setPropertyValue("tile10", True)

```

표 80. binningnode 특성

binningnode 특성	데이터 유형	특성 설명
fields	[field1 field2 ... fieldn]	변환을 보류 중인 연속형(숫자 범위) 필드입니다. 다중 필드를 동시에 구간화할 수 있습니다.
method	FixedWidth EqualCount Rank SDev Optimal	새 필드 구간(범주)에 대한 절단점 판별에 사용하는 방법입니다.
rcalculate_bins	Always IfNecessary	노드가 실행될 때마다 구간이 재계산되고 데이터가 관련 구간에 위치되는지 여부 또는 데이터가 기존 구간 및 추가된 모든 신규 구간에만 추가됨을 지정합니다.
fixed_width_name_extension	string	기본 확장자는 <i>_BIN</i> 입니다.
fixed_width_add_as	Suffix Prefix	확장자가 필드 이름의 끝에(접미문자) 또는 시작에(접두문자) 추가되는지 여부를 지정합니다. 기본 확장자는 <i>income_BIN</i> 입니다.
fixed_bin_method	Width Count	
fixed_bin_count	integer	새 필드의 고정 너비 구간(범주)의 수를 판별하는데 사용하는 정수를 지정합니다.
fixed_bin_width	real	구간의 너비 계산을 위한 값(정수 또는 실수)입니다.
equal_count_name_extension	string	기본 확장자는 <i>_TILE</i> 입니다.
equal_count_add_as	Suffix Prefix	표준 p-분위수를 사용하여 생성되는 필드 이름에 사용하는 확장자(접미문자 또는 접두문자)를 지정합니다. 기본 확장자는 <i>_TILE</i> 더하기 <i>N</i> 으로, <i>N</i> 은 분위수 번호입니다.
tile4	flag	각각이 케이스 중 25%를 포함하는 4개의 분위수 구간을 생성합니다.
tile5	flag	5개의 5분위수 구간을 생성합니다.
tile10	flag	10개의 십분위수 구간을 생성합니다.
tile20	flag	20개의 20분위수 구간을 생성합니다.
tile100	flag	100개의 백분위수 구간을 생성합니다.
use_custom_tile	flag	
custom_tile_name_extension	string	기본 확장자는 <i>_TILEN</i> 입니다.
custom_tile_add_as	Suffix Prefix	
custom_tile	integer	

표 80. binningnode 특성 (계속)

binningnode 특성	데이터 유형	특성 설명
equal_count_method	RecordCount ValueSum	RecordCount 메소드는 각 구간에 동일한 레코드 수를 지정하지만, ValueSum 메소드는 각 구간의 합계가 동일하도록 레코드를 지정합니다.
tied_values_method	Next Current Random	연결된 값 데이터를 입력할 구간을 지정하십시오.
rank_order	Ascending Descending	이 특성에는 Ascending(최저값이 1로 표시됨) 또는 Descending(최고값이 1로 표시됨)이 포함됩니다.
rank_add_as	Suffix Prefix	이 옵션은 순위, 분수순위, 퍼센트 순위에 적용됩니다.
rank	flag	
rank_name_extension	string	기본 확장자는 <code>_RANK</code> 입니다.
rank_fractional	flag	새 필드의 값이 비결측 케이스의 가중치 합계로 나눈 순위와 동일한 케이스의 순위를 생성합니다. 분수순위는 0 - 1의 범위에 들어갑니다.
rank_fractional_name_extension	string	기본 확장자는 <code>_F_RANK</code> 입니다.
rank_pct	flag	각 순위를 유효한 값을 갖는 레코드 수로 나누고 100을 곱합니다. 퍼센트 분수순위는 1 - 100의 범위에 들어갑니다.
rank_pct_name_extension	string	기본 확장자는 <code>_P_RANK</code> 입니다.
sdev_name_extension	string	
sdev_add_as	Suffix Prefix	
sdev_count	One Two Three	
optimal_name_extension	string	기본 확장자는 <code>_OPTIMAL</code> 입니다.
optimal_add_as	Suffix Prefix	
optimal_supervisor_field	field	구간화를 위해 선택된 필드가 관련된 감독 필드로 선택된 필드입니다.
optimal_merge_bins	flag	작은 케이스 빈도를 갖는 모든 구간이 더 크고 이웃한 구간에 추가되도록 지정합니다.
optimal_small_bin_threshold	integer	
optimal_pre_bin	flag	데이터 세트의 사전 구간화가 발생함을 표시합니다.
optimal_max_bins	integer	지나치게 큰 수의 구간 작성을 피하기 위한 상한을 지정합니다.
optimal_lower_end_point	Inclusive Exclusive	
optimal_first_bin	Unbounded Bounded	

표 80. binningnode 특성 (계속)

binningnode 특성	데이터 유형	특성 설명
optimal_last_bin	Unbounded Bounded	

derivednode 특성



파생 노드는 데이터 값을 수정하거나 하나 이상의 기존 필드로부터 새 필드를 작성합니다. 수식, 플래그, 명목형, 상태, 개수, 조건부 유형의 필드를 작성합니다.

예 1

```
# Create and configure a Flag Derive field node
node = stream.create("derive", "My node")
node.setPropertyValue("new_name", "DrugX_Flag")
node.setPropertyValue("result_type", "Flag")
node.setPropertyValue("flag_true", "1")
node.setPropertyValue("flag_false", "0")
node.setPropertyValue("flag_expr", "'Drug' == \"drugX\"")

# Create and configure a Conditional Derive field node
node = stream.create("derive", "My node")
node.setPropertyValue("result_type", "Conditional")
node.setPropertyValue("cond_if_cond", "@OFFSET(\"Age\", 1) = \"Age\"")
node.setPropertyValue("cond_then_expr", "@OFFSET(\"Age\", 1) = \"Age\" >< @INDEX")
node.setPropertyValue("cond_else_expr", "\"Age\"")
```

예 2

이 스크립트는 한 점(예를 들어, 이벤트가 발생한 장소)의 X 및 Y 좌표를 나타내는 XPos 및 YPos라는 두 개의 숫자 열이 있다고 가정합니다. 스크립트는 특정 좌표계에서 해당 위치를 나타내는 X 및 Y 좌표로부터 지리공간 열을 계산하는 파생 노드를 작성합니다.

```
stream = modeler.script.stream()
# Other stream configuration code
node = stream.createAt("derive", "Location", 192, 96)
node.setPropertyValue("new_name", "Location")
node.setPropertyValue("formula_expr", "['XPos', 'YPos']")
node.setPropertyValue("formula_type", "Geospatial")
# Now we have set the general measurement type, define the
# specifics of the geospatial object
node.setPropertyValue("geo_type", "Point")
node.setPropertyValue("has_coordinate_system", True)
node.setPropertyValue("coordinate_system", "ETRS_1989_EPSG_Arctic_zone_5-47")
```


표 81. *derivednode* 특성

derivednode 특성	데이터 유형	특성 설명
new_name	<i>string</i>	새 필드의 이름.
mode	Single Multiple	단일 또는 다중 필드를 지정합니다.
fields	<i>list</i>	다중 필드를 선택하기 위해서 다중 모드에서만 사용합니다.
name_extension	<i>string</i>	새 필드 이름을 위한 확장자를 지정합니다.
add_as	Suffix Prefix	확장자를 필드 이름의 접두문자(시작 부에) 또는 접미문자(끝에)로 추가합니다.
result_type	Formula Flag Set State Count Conditional	작성할 수 있는 새 필드의 6가지 유형입니다.
formula_expr	<i>string</i>	새 필드 값을 파생 노드에서 계산하기 위한 표현식입니다.
flag_expr	<i>string</i>	
flag_true	<i>string</i>	
flag_false	<i>string</i>	
set_default	<i>string</i>	
set_value_cond	<i>string</i>	주어진 값과 연관된 조건을 제공하기 위해 구조화됩니다.
state_on_val	<i>string</i>	설정 조건이 충족될 때 새 필드의 값을 지정합니다.
state_off_val	<i>string</i>	해제 조건이 충족될 때 새 필드의 값을 지정합니다.
state_on_expression	<i>string</i>	
state_off_expression	<i>string</i>	
state_initial	On Off	새 필드의 각 레코드에 On 또는 Off의 초기 값을 지정합니다. 각 조건이 충족될 때 이 값이 변할 수 있습니다.
count_initial_val	<i>string</i>	
count_inc_condition	<i>string</i>	
count_inc_expression	<i>string</i>	
count_reset_condition	<i>string</i>	
cond_if_cond	<i>string</i>	
cond_then_expr	<i>string</i>	
cond_else_expr	<i>string</i>	

표 81. *derivenode* 특성 (계속)

derivenode 특성	데이터 유형	특성 설명
formula_measure_type	Range / MeasureType.RANGE Discrete / MeasureType.DISCRETE Flag / MeasureType.FLAG Set / MeasureType.SET OrderedSet / MeasureType.ORDERED_SET Typeless / MeasureType.TYPELESS Collection / MeasureType.COLLECTION Geospatial / MeasureType.GEOSPATIAL	이 특성을 사용하여 파생된 필드와 연관된 측정을 정의할 수 있습니다. Setter 함수가 문자열 또는 MeasureType 값의 하나로 전달될 수 있습니다. Getter는 항상 MeasureType 값에 대해 리턴합니다.
collection_measure	Range / MeasureType.RANGE Flag / MeasureType.FLAG Set / MeasureType.SET OrderedSet / MeasureType.ORDERED_SET Typeless / MeasureType.TYPELESS	요약도표 필드(0의 깊이를 갖는 목록)의 경우 이 특성은 기본 값과 연관된 측정 유형을 정의합니다.
geo_type	Point MultiPoint LineString MultiLineString Polygon MultiPolygon	지리공간 필드의 경우 이 특성은 이 필드에 의해 표시되는 지리공간 오브젝트의 유형을 정의합니다. 이것은 값의 목록 깊이와 일관성을 가져야 합니다.
has_coordinate_system	<i>boolean</i>	지리공간 필드의 경우 이 특성은 이 필드가 좌표계를 갖는지 여부를 정의합니다.
coordinate_system	<i>string</i>	지리공간 필드의 경우 이 특성이 이 필드의 좌표계를 정의합니다.

ensemblenode 특성



앙상블 노드는 둘 이상의 모델 너깃을 결합하여 임의의 한 모델에서 얻을 수 있는 것보다 정확한 예측을 얻습니다.

예제

```
# Create and configure an Ensemble node
# Use this node with the models in demos\streams\pm_binaryclassifier.str
node = stream.create("ensemble", "My node")
node.setPropertyValue("ensemble_target_field", "response")
node.setPropertyValue("filter_individual_model_output", False)
node.setPropertyValue("flag_ensemble_method", "ConfidenceWeightedVoting")
node.setPropertyValue("flag_voting_tie_selection", "HighestConfidence")
```

표 82. *ensemblenode* 특성.

ensemblenode 특성	데이터 유형	특성 설명
ensemble_target_field	<i>field</i>	앙상블에서 사용하는 모든 모델에 대한 목표 필드를 지정합니다.
filter_individual_model_output	<i>flag</i>	개별 모델의 스코어링 결과가 억제되어야 하는지 여부를 지정합니다.
flag_ensemble_method	Voting ConfidenceWeightedVoting RawPropensityWeightedVoting AdjustedPropensityWeightedVoting HighestConfidence AverageRawPropensity AverageAdjustedPropensity	앙상블 스코어를 판별하는 데 사용하는 방법을 지정합니다. 이 설정은 선택된 대상이 플래그 필드인 경우에만 적용됩니다.
set_ensemble_method	Voting ConfidenceWeightedVoting HighestConfidence	앙상블 스코어를 판별하는 데 사용하는 방법을 지정합니다. 이 설정은 선택된 대상이 명목 필드인 경우에만 적용됩니다.
flag_voting_tie_selection	Random HighestConfidence RawPropensity AdjustedPropensity	투표 방법이 선택되는 경우 등순위를 해결하는 방법을 지정합니다. 이 설정은 선택된 대상이 플래그 필드인 경우에만 적용됩니다.
set_voting_tie_selection	Random HighestConfidence	투표 방법이 선택되는 경우 등순위를 해결하는 방법을 지정합니다. 이 설정은 선택된 대상이 명목 필드인 경우에만 적용됩니다.
calculate_standard_error	<i>flag</i>	목표 필드가 연속형인 경우, 표준 오차 계산이 기본적으로 실행되어 특정 또는 추정된 값과 실제 값 사이의 차이를 계산하고 해당 추정값이 얼마나 근접하게 매치했는지를 표시합니다.

fillernode 특성



채움 노드는 필드 값을 대체하고 저장 공간을 변경합니다. @BLANK(@FIELD) 같은 CLEM 조건을 기반으로 값을 대체할 수 있습니다. 또는 모든 공백 또는 널값을 특정 값으로 대체할 것을 선택할 수 있습니다. 채움 노드는 종종 유형 노드와 함께 사용하여 결측값을 대체합니다.

예제

```
node = stream.create("filler", "My node")
node.setPropertyValue("fields", ["Age"])
node.setPropertyValue("replace_mode", "Always")
node.setPropertyValue("condition", "(\"Age\" > 60) and (\"Sex\" = \"M\")")
node.setPropertyValue("replace_with", "\"old man\"")
```

표 83. *fillernode* 특성

fillernode 특성	데이터 유형	특성 설명
fields	<i>list</i>	값이 검사되어 대체될 데이터 세트의 필드입니다.
replace_mode	Always Conditional Blank Null BlankAndNull	모든 값, 공백값, 널값을 바꾸거나 지정된 조건을 기반으로 바꿀 수 있습니다.
condition	<i>string</i>	
replace_with	<i>string</i>	

filternode 특성



필터 노드는 필드를 필터링(삭제)하고, 필드 이름을 변경하고 한 소스에서 다른 소스로 필드를 맵핑합니다.

예제

```
node = stream.create("filter", "My node")
node.setPropertyValue("default_include", True)
node.setKeyedPropertyValue("new_name", "Drug", "Chemical")
node.setKeyedPropertyValue("include", "Drug", False)
```

default_include 특성 사용. `default_include` 특성의 값을 설정해도 모든 필드가 자동으로 포함 또는 제외되지는 않습니다. 단순히 현재 선택에 대한 기본값을 판별합니다. 이것은 기능적으로 필터 노드 대화 상자에서 기본적으로 필드 포함 단추를 클릭하는 것과 동등합니다. 예를 들어 다음 스크립트를 실행한다고 가정합니다.

```
node = modeler.script.stream().create("filter", "Filter")
node.setPropertyValue("default_include", False)
# Include these two fields in the list
for f in ["Age", "Sex"]:
    node.setKeyedPropertyValue("include", f, True)
```

그러면 노드는 `Age` 및 `Sex` 필드를 전달하고 다른 모든 필드는 삭제합니다. 이제 동일한 스크립트를 다시 실행하지만 두 개의 다른 필드를 지명한다고 가정합니다.

```
node = modeler.script.stream().create("filter", "Filter")
node.setPropertyValue("default_include", False)
# Include these two fields in the list
for f in ["BP", "Na"]:
    node.setKeyedPropertyValue("include", f, True)
```

이것은 필터에 두 개의 추가 필드를 추가하여 총 4개의 필드(Age, Sex, BP, Na)가 전달되게 합니다. 달리 말하면, default_include의 값을 False로 재설정해도 모든 필드가 자동으로 재설정되지 않습니다.

다른 방법으로는, 스크립트를 사용하여 또는 필터 노드 대화 상자에서 이제 default_include를 True로 변경하는 경우, 작동을 뒤집어서 위에 나열된 4개의 필드가 포함되는 대신 삭제됩니다. 의심이 갈 때는 필터 노드 대화 상자에서 제어를 갖고 실험하는 것이 이 상호작용을 이해하는 데 도움이 될 수 있습니다.

표 84. filternode 특성

filternode 특성	데이터 유형	특성 설명
default_include	flag	기본 작동이 필드를 전달 또는 필터링하는 것인지 여부를 지정하는 키가 있는 특성입니다. 이 특성을 설정해도 모든 필드를 자동으로 포함 또는 제외하지는 않습니다. 단순히 선택된 필드가 기본적으로 포함 또는 제외되는지 여부를 판별합니다. 추가 주석에 대해서는 아래 예를 참조하십시오.
include	flag	필드 포함 및 제거를 위한 키가 있는 특성입니다.
new_name	string	

historynode 특성



히스토리 노드는 이전 레코드의 필드에 있는 데이터를 포함하는 새 필드를 작성합니다. 히스토리 노드는 시계열 데이터 같은 순차 데이터에 가장 자주 사용됩니다. 히스토리 노드를 사용하기 전에 정렬 노드를 사용하여 데이터를 정렬할 수 있습니다.

예제

```
node = stream.create("history", "My node")
node.setPropertyValue("fields", ["Drug"])
node.setPropertyValue("offset", 1)
node.setPropertyValue("span", 3)
node.setPropertyValue("unavailable", "Discard")
node.setPropertyValue("fill_with", "undef")
```

표 85. historynode 특성

historynode 특성	데이터 유형	특성 설명
fields	list	히스토리를 원하는 필드.
offset	number	히스토리 필드 값을 추출하려는 마지막 레코드(현재 레코드 이전의)를 지정합니다.
span	number	값을 추출하려는 이전 레코드의 수를 지정합니다.

표 85. *historynode* 특성 (계속)

historynode 특성	데이터 유형	특성 설명
unavailable	Discard Leave Fill	히스토리 값이 없는 레코드 처리를 위해, 대개 히스토리로 사용할 이전 레코드가 없는 처음 여러 개의 레코드(데이터 세트의 맨 위에 있는)를 말합니다.
fill_with	String Number	히스토리 값을 사용할 수 없는 레코드에 사용할 값이나 문자열을 지정합니다.

partitionnode 특성



파티션 노드는 파티션 필드를 생성하는데, 이 필드는 모델 작성의 학습, 검정, 검증 단계를 위한 별개의 서브세트로 데이터를 분할합니다.

예제

```
node = stream.create("partition", "My node")
node.setPropertyValue("create_validation", True)
node.setPropertyValue("training_size", 33)
node.setPropertyValue("testing_size", 33)
node.setPropertyValue("validation_size", 33)
node.setPropertyValue("set_random_seed", True)
node.setPropertyValue("random_seed", 123)
node.setPropertyValue("value_mode", "System")
```

표 86. *partitionnode* 특성

partitionnode 파티션 노드	데이터 유형	특성 설명
new_name	string	노드가 생성하는 파티션 필드의 이름
create_validation	flag	검증 파티션이 작성되는지 여부를 지정합니다.
training_size	integer	훈련 파티션에 할당될 레코드의 퍼센트(0-100)
testing_size	integer	검정 파티션에 할당될 레코드의 퍼센트(0-100)
validation_size	integer	검증 파티션에 할당될 레코드의 퍼센트(0-100)입니다. 검증 파티션이 작성되지 않으면 무시됩니다.
training_label	string	훈련 파티션의 레이블
testing_label	string	검정 파티션의 레이블.
validation_label	string	검증 파티션의 레이블입니다. 검증 파티션이 작성되지 않으면 무시됩니다.
value_mode	System SystemAndLabel Label	데이터에서 각 파티션을 나타내기 위해 사용하는 값을 지정합니다. 예를 들어, 훈련 표본은 시스템 정수 1, 레이블 Training 또는 둘의 조합인 1_Training으로 나타낼 수 있습니다.
set_random_seed	부울	사용자 지정 난수 시드 사용 여부를 지정합니다.
random_seed	integer	사용자 지정 난수 시드 값입니다. 이 값을 사용하려면 set_random_seed가 True로 설정되어야 합니다.

표 86. *partitionnode* 특성 (계속)

partitionnode 파티션 노드	데이터 유형	특성 설명
enable_sql_generation	부울	SQL 푸시백을 사용하여 레코드를 파티션에 지정하는지 여부를 지정합니다.
unique_field		레코드가 무작위이지만 반복 가능한 방식으로 파티션에 지정되도록 보장하기 위해 사용하는 입력 필드를 지정합니다. 이 값을 사용하려면 enable_sql_generation이 True로 설정되어야 합니다.

reclassifynode 특성



재분류 노드는 한 세트의 범주형 값을 다른 값으로 변환합니다. 재분류는 분석을 위해 범주를 접거나 데이터를 재그룹화하는 데 유용합니다.

예제

```
node = stream.create("reclassify", "My node")
node.setPropertyValue("mode", "Multiple")
node.setPropertyValue("replace_field", True)
node.setPropertyValue("field", "Drug")
node.setPropertyValue("new_name", "Chemical")
node.setPropertyValue("fields", ["Drug", "BP"])
node.setPropertyValue("name_extension", "reclassified")
node.setPropertyValue("add_as", "Prefix")
node.setKeyedPropertyValue("reclassify", "drugA", True)
node.setPropertyValue("use_default", True)
node.setPropertyValue("default", "BrandX")
node.setPropertyValue("pick_list", ["BrandX", "Placebo", "Generic"])
```

표 87. *reclassifynode* 특성

reclassifynode 특성	데이터 유형	특성 설명
mode	Single Multiple	Single은 한 필드의 범주를 재분류합니다. Multiple은 한 번에 둘 이상의 필드의 변환을 가능하게 하는 옵션을 활성화합니다.
replace_field	flag	
field	string	단일 모드에서만 사용합니다.
new_name	string	단일 모드에서만 사용합니다.
fields	[field1 field2 ... fieldn]	다중 모드에서만 사용합니다.
name_extension	string	다중 모드에서만 사용합니다.
add_as	Suffix Prefix	다중 모드에서만 사용합니다.
reclassify	string	필드 값의 구조화된 특성.
use_default	flag	기본값을 사용하십시오.
default	string	기본값을 지정하십시오.

표 87. *reclassifynode* 특성 (계속)

reclassifynode 특성	데이터 유형	특성 설명
pick_list	[string string ... string]	사용자가 알려진 새 값의 목록을 가져와서 테이블의 드롭 다운 목록을 채울 수 있습니다.

reordernode 특성



필드 다시 정렬 노드는 필드를 다운스트림으로 표시하는 데 사용하는 기본 순서를 정의합니다. 이 순서는 테이블, 목록 및 필드 선택기 같은 다양한 장소에서 필드의 표시에 영향을 줍니다. 이 작업은 관심있는 필드를 더 잘 보이게 만들기 위해 넓은 데이터 세트에 대해 작업할 때 유용합니다.

예제

```
node = stream.create("reorder", "My node")
node.setPropertyValue("mode", "Custom")
node.setPropertyValue("sort_by", "Storage")
node.setPropertyValue("ascending", False)
node.setPropertyValue("start_fields", ["Age", "Cholesterol"])
node.setPropertyValue("end_fields", ["Drug"])
```

표 88. *reordernode* 특성

reordernode 특성	데이터 유형	특성 설명
mode	Custom Auto	값을 자동으로 정렬하거나 사용자 정의 순서를 지정할 수 있습니다.
sort_by	Name Type Storage	
ascending	flag	
start_fields	[field1 field2 ... fieldn]	새 필드가 이들 필드 뒤에 삽입됩니다.
end_fields	[field1 field2 ... fieldn]	새 필드가 이들 필드 앞에 삽입됩니다.

reprojectnode 특성



SPSS Modeler 내에서 표현식 작성기 공간 함수, STP(Spatio-Temporal Prediction) 노드, 맵 시각화 노드 같은 항목이 투영된 좌표계를 사용합니다. 지리적 좌표계를 사용하고 사용자가 가져오는 임의의 데이터의 좌표계를 변경하려면 재투영 노드를 사용하십시오.

표 89. *reprojectnode* 특성

reprojectnode 특성	데이터 유형	특성 설명
reproject_fields	[field1 field2 ... fieldn]	재투영될 모든 필드를 나열하십시오.
reproject_type	Streamdefault Specify	필드를 재투영하는 방법을 선택하십시오.

표 89. reprojectnode 특성 (계속)

reprojectnode 특성	데이터 유형	특성 설명
coordinate_system	string	필드에 적용될 좌표계의 이름입니다. 예: set reprojectnode.coordinate_system = "WGS_1984_World_Mercator"

restructurenode 특성



구조변환 노드는 명목 또는 플래그 필드를 아직 또 다른 필드의 값으로 채워질 수 있는 필드 그룹으로 변환합니다. 예를 들어, *payment type*이라는 이름의 필드와 *credit, cash, debit*의 값이 주어진 경우, 각각이 실제 이루어진 지불의 값을 포함할 수 있는 세 개의 새 필드(*credit, cash, debit*)가 작성됩니다.

예제

```
node = stream.create("restructure", "My node")
node.setKeyedPropertyValue("fields_from", "Drug", ["drugA", "drugX"])
node.setPropertyValue("include_field_name", True)
node.setPropertyValue("value_mode", "OtherFields")
node.setPropertyValue("value_fields", ["Age", "BP"])
```

표 90. restructurenode 특성

restructurenode 특성	데이터 유형	특성 설명
fields_from	[<i>category category category</i>] all	
include_field_name	flag	구조변환된 필드 이름에서 필드 이름을 사용할지 여부를 표시합니다.
value_mode	OtherFields Flags	구조변환된 필드의 값을 지정하는 모드를 표시합니다. OtherFields를 사용하면 사용할 필드를 지정해야 합니다(아래 참조). Flags를 사용하면 값은 수치 플래그입니다.
value_fields	list	value_mode가 OtherFields인 경우 필수입니다. 값 필드로 사용할 필드를 지정합니다.

rfmanalysisnode 특성



RFM(Recency, Frequency, Monetary) 분석 노드를 사용하면 얼마나 최근에 사용자로부터 구매했는지(최근성), 얼마나 자주 구매했는지(빈도) 및 모든 트랜잭션에서 얼마나 소비했는지(구매총액)를 조사하여 최고의 고객이 될 수 있는 고객을 정량적으로 판별할 수 있습니다.

예제

```

node = stream.create("rfmanalysis", "My node")
node.setPropertyValue("recency", "Recency")
node.setPropertyValue("frequency", "Frequency")
node.setPropertyValue("monetary", "Monetary")
node.setPropertyValue("tied_values_method", "Next")
node.setPropertyValue("recalculate_bins", "IfNecessary")
node.setPropertyValue("recency_thresholds", [1, 500, 800, 1500, 2000, 2500])

```

표 91. rfmanalysisnode 특성

rfmanalysisnode 특성	데이터 유형	특성 설명
recency	field	최근성 필드를 지정하십시오. 이것은 날짜, 시간소인 또는 단순 숫자일 수 있습니다.
frequency	field	빈도 필드를 지정하십시오.
monetary	field	구매총액 필드를 지정하십시오.
recency_bins	integer	생성될 최근 구간의 수를 지정하십시오.
recency_weight	number	최근 데이터에 적용될 가중치를 지정하십시오. 기본값은 100입니다.
frequency_bins	integer	생성될 빈도 구간의 수를 지정하십시오.
frequency_weight	number	빈도 데이터에 적용될 가중치를 지정하십시오. 기본값은 10입니다.
monetary_bins	integer	생성될 구매총액 구간의 수를 지정하십시오.
monetary_weight	number	구매총액 데이터에 적용될 가중치를 지정하십시오. 기본값은 1입니다.
tied_values_method	Next Current	연결된 값 데이터를 입력할 구간을 지정하십시오.
recalculate_bins	Always IfNecessary	
add_outliers	flag	recalculate_bins가 IfNecessary로 설정된 경우에만 사용할 수 있습니다. 설정되면 하한 구간 아래에 위치한 레코드는 하한 구간에 추가되고 최고 구간 위의 레코드는 최고 구간에 추가됩니다.
binned_field	Recency Frequency Monetary	
recency_thresholds	value value	recalculate_bins가 Always로 설정된 경우에만 사용 가능합니다. 최근 구간의 상한 및 하한 임계값을 지정하십시오. 한 구간의 상한 임계값을 다음 구간의 하한 임계값으로 사용합니다. 예를 들어, [10 30 60]은 두 개의 구간을 정의하는데, 첫 번째 구간은 10과 30의 상한 및 하한 임계값을 갖고 두 번째 구간 임계값은 30과 60입니다.
frequency_thresholds	value value	recalculate_bins가 Always로 설정된 경우에만 사용 가능합니다.
monetary_thresholds	value value	recalculate_bins가 Always로 설정된 경우에만 사용 가능합니다.

settoflagnode 특성



플래그로 설정 노드는 하나 이상의 명목 필드에 대해 정의된 범주형 값을 바탕으로 다중 플래그 필드를 파생시킵니다.

예제

```
node = stream.create("settoflag", "My node")
node.setKeyedPropertyValue("fields_from", "Drug", ["drugA", "drugX"])
node.setPropertyValue("true_value", "1")
node.setPropertyValue("false_value", "0")
node.setPropertyValue("use_extension", True)
node.setPropertyValue("extension", "Drug_Flag")
node.setPropertyValue("add_as", "Suffix")
node.setPropertyValue("aggregate", True)
node.setPropertyValue("keys", ["Cholesterol"])
```

표 92. settoflagnode 특성

settoflagnode 특성	데이터 유형	특성 설명
fields_from	[<i>category category</i> <i>category</i>] all	
true_value	<i>string</i>	플래그를 설정할 때 노드가 사용하는 true 값을 지정합니다. 기본값은 T입니다.
false_value	<i>string</i>	플래그를 설정할 때 노드가 사용하는 false 값을 지정합니다. 기본값은 F입니다.
use_extension	<i>flag</i>	확장자를 새 플래그 필드에 대한 접미문자 또는 접두문자로 사용하십시오.
extension	<i>string</i>	
add_as	Suffix Prefix	확장자가 접미문자 또는 접두문자로 추가되는지 여부를 지정합니다.
aggregate	<i>flag</i>	키 필드를 기준으로 레코드를 함께 그룹화합니다. 그룹의 모든 플래그 필드는 임의의 레코드가 true로 설정될 때 사용 가능합니다.
keys	<i>list</i>	키 필드

statistictransformnode 특성



통계량 변환 노드는 IBM SPSS Modeler의 데이터 소스에 대해 IBM SPSS Statistics 구문 명령문의 선택을 실행합니다. 이 노드는 IBM SPSS Statistics의 사용권 사본이 필요합니다.

이 노드의 특성은 349 페이지의 『statisticstransformnode 특성』에서 설명됩니다.

timeintervalsnode 특성(더 이상 사용되지 않음)



참고: 이 노드는 SPSS Modeler 버전 18에서 더 이상 사용되지 않으며 새 시계열 노드로 대체되었습니다. 시간 구간 노드는 시계열 데이터 모델링을 위한 구간을 지정하고 (필요한 경우) 레이블을 작성합니다. 조건이 균등하게 분포되지 않는 경우, 노드는 필요에 따라 값을 채우거나 통합하여 레코드 사이에 균일한 구간을 생성할 수 있습니다.

예제

```
node = stream.create("timeintervals", "My node")
node.setPropertyValue("interval_type", "SecondsPerDay")
node.setPropertyValue("days_per_week", 4)
node.setPropertyValue("week_begins_on", "Tuesday")
node.setPropertyValue("hours_per_day", 10)
node.setPropertyValue("day_begins_hour", 7)
node.setPropertyValue("day_begins_minute", 5)
node.setPropertyValue("day_begins_second", 17)
node.setPropertyValue("mode", "Label")
node.setPropertyValue("year_start", 2005)
node.setPropertyValue("month_start", "January")
node.setPropertyValue("day_start", 4)
node.setKeyedPropertyValue("pad", "AGE", "MeanOfRecentPoints")
node.setPropertyValue("agg_mode", "Specify")
node.setPropertyValue("agg_set_default", "Last")
```

표 93. timeintervalsnode 특성.

timeintervalsnode 특성	데이터 유형	특성 설명
interval_type	None Periods CyclicPeriods Years Quarters Months DaysPerWeek DaysNonPeriodic HoursPerDay HoursNonPeriodic MinutesPerDay MinutesNonPeriodic SecondsPerDay SecondsNonPeriodic	
mode	Label Create	레코드를 연속적으로 레이블할지 아니면 지정된 날짜, 시간소인 또는 시간 필드를 기반으로 계열을 작성할지 여부를 지정합니다.

표 93. *timeintervalsnode* 특성 (계속).

timeintervalsnode 특성	데이터 유형	특성 설명
field	<i>field</i>	데이터로부터 계열을 작성할 때, 각 레코드의 날짜나 시간을 표시하는 필드를 지정합니다.
period_start	<i>integer</i>	주기 또는 순환 주기의 시작 구간을 지정합니다.
cycle_start	<i>integer</i>	순환 주기의 시작 주기
year_start	<i>integer</i>	적용 가능한 구간 유형에 대해, 첫 번째 구간에 해당하는 년도입니다.
quarter_start	<i>integer</i>	적용 가능한 구간 유형에 대해, 첫 번째 구간에 해당하는 분기입니다.
month_start	January February March April May June July August September October November December	
day_start	<i>integer</i>	
hour_start	<i>integer</i>	
minute_start	<i>integer</i>	
second_start	<i>integer</i>	
periods_per_cycle	<i>integer</i>	순환 주기의 경우, 각 순환 내의 숫자입니다.
fiscal_year_begins	January February March April May June July August September October November December	분기별 구간의 경우, 회계연도가 시작하는 월을 지정합니다.
week_begins_on	Sunday Monday Tuesday Wednesday Thursday Friday Saturday Sunday	주기적 구간(주당 일, 일당 시간, 일당 분, 일당 초)의 경우, 주가 시작하는 요일을 지정합니다.

표 93. *timeintervalsnode* 특성 (계속).

timeintervalsnode 특성	데이터 유형	특성 설명
day_begins_hour	<i>integer</i>	주기적 구간(일당 시간, 일당 분, 일당 초)의 경우, 하루를 시작하는 시간을 지정합니다. day_begins_minute 및 day_begins_second와 함께 사용하여 8:05:01 같은 정확한 시간을 지정할 수 있습니다. 아래 사용법 예를 참조하십시오.
day_begins_minute	<i>integer</i>	주기적 구간(일당 시간, 일당 분, 일당 초)의 경우, 하루를 시작하는 분을 지정합니다(예: 8:05의 5).
day_begins_second	<i>integer</i>	주기적 구간(일당 시간, 일당 분, 일당 초)의 경우, 하루를 시작하는 초를 지정합니다(예: 8:05:17의 17).
days_per_week	<i>integer</i>	주기적 구간(주당 일, 일당 시간, 일당 분, 일당 초)의 경우, 주당 요일 수를 지정합니다.
hours_per_day	<i>integer</i>	주기적 구간(일당 시간, 일당 분, 일당 초)의 경우, 하루의 시간을 지정합니다.
interval_increment	1 2 3 4 5 6 10 15 20 30	일당 분 및 일당 초의 경우, 각 레코드에 대해 증분할 분 또는 초를 지정합니다.
field_name_extension	<i>string</i>	
field_name_extension_as_prefix	<i>flag</i>	

표 93. *timeintervalsnode* 특성 (계속).

timeintervalsnode 특성	데이터 유형	특성 설명
date_format	"DDMMYY" "MMDDYY" "YYMMDD" "YYYYMMDD" "YYYYDDD" DAY MONTH "DD-MM-YY" "DD-MM-YYYY" "MM-DD-YY" "MM-DD-YYYY" "DD-MON-YY" "DD-MON-YYYY" "YYYY-MM-DD" "DD.MM.YY" "DD.MM.YYYY" "MM.DD.YYYY" "DD.MON.YY" "DD.MON.YYYY" "DD/MM/YY" "DD/MM/YYYY" "MM/DD/YY" "MM/DD/YYYY" "DD/MON/YY" "DD/MON/YYYY" MON YYYY q Q YYYY ww WK YYYY	
time_format	"HHMMSS" "HHMM" "MMSS" "HH:MM:SS" "HH:MM" "MM:SS" "(H)H:(M)M:(S)S" "(H)H:(M)M" "(M)M:(S)S" "HH.MM.SS" "HH.MM" "MM.SS" "(H)H.(M)M.(S)S" "(H)H.(M)M" "(M)M.(S)S"	
aggregate	Mean Sum Mode Min Max First Last TrueIfAnyTrue	필드의 통합 방법을 지정합니다.
pad	Blank MeanOfRecentPoints True False	필드의 채우기 방법을 지정합니다.

표 93. *timeintervalsnode* 특성 (계속).

timeintervalsnode 특성	데이터 유형	특성 설명
agg_mode	All Specify	모든 필드를 통합하거나 필요에 따라 기본 함수로 채울지 아니면 사용할 필드 및 함수를 지정할지 여부를 지정합니다.
agg_range_default	Mean Sum Mode Min Max	연속형 필드를 통합할 때 사용할 기본 함수를 지정합니다.
agg_set_default	Mode First Last	명목 필드를 통합할 때 사용할 기본 함수를 지정합니다.
agg_flag_default	TrueIfAnyTrue Mode First Last	
pad_range_default	Blank MeanOfRecentPoints	연속형 필드를 채울 때 사용할 기본 함수를 지정합니다.
pad_set_default	Blank MostRecentValue	
pad_flag_default	Blank True False	
max_records_to_create	<i>integer</i>	계열을 채울 때 작성할 최대 레코드 수를 지정합니다.
estimation_from_beginning	<i>flag</i>	
estimation_to_end	<i>flag</i>	
estimation_start_offset	<i>integer</i>	
estimation_num_holdouts	<i>integer</i>	
create_future_records	<i>flag</i>	
num_future_records	<i>integer</i>	
create_future_field	<i>flag</i>	
future_field_name	<i>string</i>	

transposenode 특성



전치 노드는 행 및 열의 데이터를 바꿔서 레코드가 필드가 되고 필드가 레코드가 되게 합니다.

예제


```

node = stream.create("transpose", "My node")
node.setPropertyValue("transposed_names", "Read")
node.setPropertyValue("read_from_field", "TimeLabel")
node.setPropertyValue("max_num_fields", "1000")
node.setPropertyValue("id_field_name", "ID")

```

표 94. *transposenode* 특성

transposenode 특성	데이터 유형	특성 설명
transpose_method	<i>enum</i>	정상(normal), CASE에서 VAR로(casetovar) 또는 VAR에서 CASE로(vartocase)와 같은 전치 방법을 지정합니다.
transposed_names	Prefix Read	정상 전치 방법의 특성입니다. 새 필드 이름이 지정된 접두문자를 기반으로 자동으로 생성될 수 있거나, 데이터의 기존 필드에서 읽을 수 있습니다.
prefix	<i>string</i>	정상 전치 방법의 특성입니다.
num_new_fields	<i>integer</i>	정상 전치 방법의 특성입니다. 접두문자를 사용할 때, 작성할 새 필드의 최대 수를 지정합니다.
read_from_field	<i>field</i>	정상 전치 방법의 특성입니다. 이름을 읽는 필드입니다. 이것은 인스턴스화된 필드여야 하며, 그렇지 않으면 노드가 실행될 때 오류가 발생합니다.
max_num_fields	<i>integer</i>	정상 전치 방법의 특성입니다. 필드에서 이름을 읽을 때, 지나치게 많은 필드 수의 작성을 피하기 위한 상한을 지정합니다.
transpose_type	Numeric String Custom	정상 전치 방법의 특성입니다. 기본적으로 연속형(수치 범위) 필드가 전치되지만, 수치 필드의 사용자 정의 서브세트를 선택하거나 대신 모든 문자열 필드를 전치할 수 있습니다.
transpose_fields	<i>list</i>	정상 전치 방법의 특성입니다. Custom 옵션을 사용할 때 전치할 필드를 지정합니다.
id_field_name	<i>field</i>	정상 전치 방법의 특성입니다.
index	<i>field</i>	CASE에서 VAR로(casetovar) 전치 방법의 특성입니다. 색인 필드로 사용할 여러 필드를 승인합니다. field1 ... fieldN
column	<i>field</i>	CASE에서 VAR로(casetovar) 전치 방법의 특성입니다. 열 필드로 사용할 여러 필드를 승인합니다. field1 ... fieldN
value	<i>field</i>	CASE에서 VAR로(casetovar) 전치 방법의 특성입니다. 값 필드로 사용할 여러 필드를 승인합니다. field1 ... fieldN
id_variables	<i>field</i>	VAR에서 CASE로(vartocase) 전치 방법의 특성입니다. ID 변수 필드로 사용할 여러 필드를 승인합니다. field1 ... fieldN
value_variables	<i>field</i>	VAR에서 CASE로(vartocase) 전치 방법의 특성입니다. 값 변수 필드로 사용할 여러 필드를 승인합니다. field1 ... fieldN

typenode 특성



유형 노드는 필드 메타데이터 및 특성을 지정합니다. 예를 들어 각 필드에 대한 측정 수준(연속형, 명목형, 순서형 또는 플래그)을 지정하고, 결측값 및 시스템 널 처리를 위한 옵션을 설정하고, 모델링 목적으로 필드의 역할을 설정하고, 필드와 값 레이블을 지정하고, 필드의 값을 지정할 수 있습니다.

예제

```
node = stream.createAt("type", "My node", 50, 50)
node.setKeyedPropertyValue("check", "Cholesterol", "Coerce")
node.setKeyedPropertyValue("direction", "Drug", "Input")
node.setKeyedPropertyValue("type", "K", "Range")
node.setKeyedPropertyValue("values", "Drug", ["drugA", "drugB", "drugC", "drugD", "drugX",
"drugY", "drugZ"])
node.setKeyedPropertyValue("null_missing", "BP", False)
node.setKeyedPropertyValue("whitespace_missing", "BP", False)
node.setKeyedPropertyValue("description", "BP", "Blood Pressure")
node.setKeyedPropertyValue("value_labels", "BP", [["HIGH", "High Blood Pressure"],
["NORMAL", "normal blood pressure"]])
```

일부 경우에는 다른 노드가 올바르게 작업하기 위해 플래그로 설정 노드의 `fields from` 특성 같은 유형 노드를 완전히 인스턴스화할 필요가 있을 수 있습니다. 단순히 테이블 노드를 연결하고 실행하여 필드를 인스턴스화할 수 있습니다.

```
tablenode = stream.createAt("table", "Table node", 150, 50)
stream.link(node, tablenode)
tablenode.run(None)
stream.delete(tablenode)
```

표 95. *typenode* 특성.

typenode 특성	데이터 유형	특성 설명
direction	Input Target Both None Partition Split Frequency RecordID	필드 역할에 대한 키가 있는 특성입니다. 참고: 값 In 및 Out은 이제 더 이상 사용하지 않습니다. 이들에 대한 지원은 차후 릴리스에서 철회될 수 있습니다.

표 95. *typenode* 특성 (계속).

typenode 특성	데이터 유형	특성 설명
type	Range Flag Set Typeless Discrete OrderedSet Default	필드의 측정 수준(이전에는 필드의 "유형"으로 불렀음). type을 Default로 설정하면 모든 values 모수 설정을 지우며, value_mode가 Specify 값을 갖는 경우 Read로 재설정됩니다. value_mode가 Pass 또는 Read로 설정되는 경우, type을 설정해도 value_mode에 영향을 주지 않습니다. 참고: 내부적으로 사용하는 데이터 유형은 유형 노드에서 볼 수 있는 것과 다릅니다. 대응은 다음과 같습니다. 범위 -> 연속형 변수군 -> 명목형 OrderedSet -> 순서 이산형 -> 범주형
storage	Unknown String Integer Real Time Date Timestamp	필드 저장 유형에 대한 읽기 전용 키가 있는 특성입니다.
check	None Nullify Coerce Discard Warn Abort	필드 유형 및 범위 검사에 대한 키가 있는 특성입니다.
values	[<i>value value</i>]	연속형 필드의 경우 첫 번째 값은 최소값, 마지막 값은 최대값입니다. 명목 필드의 경우 모든 값을 지정하십시오. 플래그 필드의 경우 첫 번째 값은 <i>false</i> 를 나타내고, 마지막 값은 <i>true</i> 를 나타냅니다. 이 특성을 설정하면 자동으로 value_mode 특성을 Specify로 설정합니다.
value_mode	Read Pass Read+ Current Specify	값을 설정하는 방법을 판별합니다. 이 특성을 Specify로 직접 설정할 수 없음을 주의하십시오. 특정 값을 사용하려면 values 특성을 설정하십시오.
extend_values	<i>flag</i>	value_mode가 Read로 설정될 때 적용됩니다. 필드에 대한 임의의 기존 값에 새로 읽은 값을 추가하려면 T로 설정하십시오. 새로 읽은 값 대신 기존 값을 삭제하려면 F를 설정하십시오.
enable_missing	<i>flag</i>	T로 설정되면 필드에 대한 결측값 추적을 활성화합니다.
missing_values	[<i>value value ...</i>]	결측값을 표시하는 데이터 값을 지정합니다.
range_missing	<i>flag</i>	결측값(공백) 범위가 필드에 대해 정의되는지 여부를 지정합니다.

표 95. *typenode* 특성 (계속).

typenode 특성	데이터 유형	특성 설명
missing_lower	string	range_missing이 true일 때, 결측값 범위의 하한을 지정합니다.
missing_upper	string	range_missing이 true일 때, 결측값 범위의 상한을 지정합니다.
null_missing	flag	T로 설정될 때, nulls(소프트웨어에서 \$null\$로 표시되는 정의되지 않은 값)는 결측값으로 간주됩니다.
whitespace_missing	flag	T로 설정될 때, 공백(공백, 탭 및 줄 바꾸기)만 포함하는 값은 결측값으로 간주됩니다.
description	string	필드의 설명을 지정합니다.
value_labels	[[Value LabelString] [Value LabelString] ...]	값 쌍에 대한 레이블을 지정하는 데 사용합니다.
display_places	integer	표시될 때 필드에 대한 소수점 이하 자리수를 설정합니다(REAL 저장 공간을 갖는 필드에만 적용됨). 값 -1은 스트림 기본값을 사용합니다.
export_places	integer	내보낼 때 필드에 대한 소수점 이하 자리수를 설정합니다(REAL 저장 공간을 갖는 필드에만 적용됨). 값 -1은 스트림 기본값을 사용합니다.
decimal_separator	DEFAULT PERIOD COMMA	필드의 소수점 구분자를 설정합니다(REAL 저장 공간을 갖는 필드에만 적용됨).
date_format	"DDMMYY" "MMDDYY" "YYMMDD" "YYYYMMDD" "YYYYDDD" DAY MONTH "DD-MM-YY" "DD-MM-YYYY" "MM-DD-YY" "MM-DD-YYYY" "DD-MON-YY" "DD-MON-YYYY" "YYYY-MM-DD" "DD.MM.YY" "DD.MM.YYYY" "MM.DD.YYYY" "DD.MON.YY" "DD.MON.YYYY" "DD/MM/YY" "DD/MM/YYYY" "MM/DD/YY" "MM/DD/YYYY" "DD/MON/YY" "DD/MON/YYYY" MON YYYY q Q YYYY ww WK YYYY	필드의 날짜 형식을 설정합니다(DATE 또는 TIMESTAMP 저장 공간을 갖는 필드에만 적용됨).

표 95. *typenode* 특성 (계속).

typenode 특성	데이터 유형	특성 설명
time_format	"HHMMSS" "HHMM" "MMSS" "HH:MM:SS" "HH:MM" "MM:SS" "(H)H:(M)M:(S)S" "(H)H:(M)M" "(M)M:(S)S" "HH.MM.SS" "HH.MM" "MM.SS" "(H)H.(M)M.(S)S" "(H)H.(M)M" "(M)M.(S)S"	필드의 시간 형식을 설정합니다(TIME 또는 TIMESTAMP 저장 공간을 갖는 필드에만 적용됨).
number_format	DEFAULT STANDARD SCIENTIFIC CURRENCY	필드의 숫자 표시 형식을 설정합니다.
standard_places	<i>integer</i>	표준 형식으로 표시될 때 필드의 소수점 이하 자리수를 설정합니다. 값 -1은 스트림 기본값을 사용합니다. 기존 display_places 슬롯도 이것을 변경하지만 더 이상 사용하지 않음을 참고하십시오.
scientific_places	<i>integer</i>	지수표기 형식으로 표시될 때 필드의 소수점 이하 자리수를 설정합니다. 값 -1은 스트림 기본값을 사용합니다.
currency_places	<i>integer</i>	통화 형식으로 표시될 때 필드의 소수점 이하 자리수를 설정합니다. 값 -1은 스트림 기본값을 사용합니다.
grouping_symbol	DEFAULT NONE LOCALE PERIOD COMMA SPACE	필드의 집단 기호를 설정합니다.
column_width	<i>integer</i>	필드의 열 너비를 설정합니다. 값 -1은 열 너비를 Auto로 설정합니다.
justify	AUTO CENTER LEFT RIGHT	필드의 열 맞춤을 설정합니다.

표 95. *typenode* 특성 (계속).

typenode 특성	데이터 유형	특성 설명
measure_type	Range / MeasureType.RANGE Discrete / MeasureType.DISCRETE Flag / MeasureType.FLAG Set / MeasureType.SET OrderedSet / MeasureType.ORDERED_SET Typeless / MeasureType.TYPELESS Collection / MeasureType.COLLECTION Geospatial / MeasureType.GEOSPATIAL	이 키가 있는 특성은 필드와 연관된 측정을 정의하는 데 사용할 수 있다는 점에서 <i>type</i> 과 비슷합니다. 다른 점은 Python 스크립팅에서 setter 함수도 MeasureType 값의 하나로 전달될 수 있는 반면 getter는 항상 MeasureType 값에서 리턴한다는 점입니다.
collection_measure	Range / MeasureType.RANGE Flag / MeasureType.FLAG Set / MeasureType.SET OrderedSet / MeasureType.ORDERED_SET Typeless / MeasureType.TYPELESS	요약도표 필드(0의 깊이를 갖는 목록)의 경우 이 키가 있는 특성은 기본 값과 연관된 측정 유형을 정의합니다.
geo_type	Point MultiPoint LineString MultiLineString Polygon MultiPolygon	지리공간 필드의 경우 이 키가 있는 특성은 이 필드에 의해 표시되는 지리공간 오브젝트의 유형을 정의합니다. 이것은 값의 목록 깊이와 일관성을 가져야 합니다.
has_coordinate_system	<i>boolean</i>	지리공간 필드의 경우 이 특성은 이 필드가 좌표계를 갖는지 여부를 정의합니다.
coordinate_system	<i>string</i>	지리공간 필드의 경우 이 키가 있는 특성이 이 필드의 좌표계를 정의합니다.
custom_storage_type	Unknown / MeasureType.UNKNOWN String / MeasureType.STRING Integer / MeasureType.INTEGER Real / MeasureType.REAL Time / MeasureType.TIME Date / MeasureType.DATE Timestamp / MeasureType.TIMESTAMP List / MeasureType.LIST	이 키가 있는 특성은 필드의 대체 저장 공간을 정의하는 데 사용할 수 있다는 점에서 <i>custom_storage</i> 와 유사합니다. 다른 점은 Python 스크립팅에서 setter 함수도 StorageType 값의 하나로 전달될 수 있는 반면 getter는 항상 StorageType 값에서 리턴한다는 점입니다.
custom_list_storage_type	String / MeasureType.STRING Integer / MeasureType.INTEGER Real / MeasureType.REAL Time / MeasureType.TIME Date / MeasureType.DATE Timestamp / MeasureType.TIMESTAMP	목록 필드의 경우 이 키가 있는 특성이 기본 값의 저장 유형을 지정합니다.
custom_list_depth	<i>integer</i>	목록 필드의 경우 이 키가 있는 특성은 필드의 깊이를 지정합니다.
max_list_length	<i>integer</i>	측정 수준이 지리 공간 또는 컬렉션인 데이터에만 사용할 수 있습니다. 목록이 포함할 수 있는 요소 수를 지정하여 목록의 최대 길이를 설정하십시오.

표 95. *typenode* 특성 (계속).

typenode 특성	데이터 유형	특성 설명
max_string_length	<i>integer</i>	유형이 없는 데이터에만 사용할 수 있으며 테이블을 작성하기 위해 SQL을 생성할 때 사용됩니다. 데이터에서 가장 큰 문자열 값을 입력하십시오. 그러면 문자열을 포함하기에 충분히 열이 테이블에 생성됩니다.

제 12 장 그래프 노드 특성

그래프 노드 공통 특성

이 절에서는 공통 특성 및 각 노드 유형에 특정한 특성을 포함하여 그래프 노드에 사용 가능한 특성을 설명합니다.

표 96. 공통 그래프 노드 특성

공통 그래프 노드 특성	데이터 유형	특성 설명
title	<i>string</i>	제목을 지정합니다. 예: "This is a title."
caption	<i>string</i>	캡션을 지정합니다. 예: "This is a caption."
output_mode	Screen File	그래프 노드의 출력이 표시되는지 아니면 파일에 기록되는지 여부를 지정합니다.
output_format	BMP JPEG PNG HTML output (.cou)	출력의 유형을 지정합니다. 각 노드에 대해 허용되는 출력의 정확한 유형은 다릅니다.
full_filename	<i>string</i>	그래프 노드로부터 생성되는 출력에 대한 대상 경로 및 파일 이름을 지정합니다.
use_graph_size	<i>flag</i>	아래의 너비 및 높이 특성을 사용하여 그래프가 명시적으로 크기 지정되는지 여부를 제어합니다. 화면으로 출력되는 그래프에만 영향을 줍니다. 분포 노드에는 사용할 수 없습니다.
graph_width	<i>number</i>	use_graph_size가 True일 때, 그래프 너비를 픽셀 단위로 설정합니다.
graph_height	<i>number</i>	use_graph_size가 True일 때, 그래프 높이를 픽셀 단위로 설정합니다.

선택적 필드 끄기

도표에 대한 오버레이 필드와 같은 선택적 필드는 다음 예에서 표시되는 것처럼 특성 값을 ""(빈 문자열)로 설정하여 끌 수 있습니다.

```
plotnode.setPropertyValue("color_field", "")
```

색상 지정

제목, 캡션, 배경, 레이블에 대한 색상은 해시(#) 기호로 시작하는 16진 문자열을 사용하여 지정할 수 있습니다. 예를 들어, 그래프 배경을 파란색으로 설정하려면 다음 명령문을 사용합니다.

```
mygraphnode.setPropertyValue("graph_background", "#87CEEB")
```

여기에서 처음 두 숫자 87은 빨간색 내용을 지정하고, 중간 두 숫자 CE는 녹색 내용을 지정하고, 마지막 두 숫자 EB는 파란색 내용을 지정합니다. 각 숫자는 0 - 9 또는 A - F 범위의 값을 가질 수 있습니다. 이들 값은 함께 빨강-녹색-파랑 또는 RGB 색상을 지정할 수 있습니다.

참고: RGB로 색상을 지정할 때, 사용자 인터페이스의 필드 선택기를 사용하여 올바른 색상 코드를 판별할 수 있습니다. 원하는 정보를 갖는 도구팁을 활성화하려면 단순히 색상 위로 마우스를 이동하십시오.

collectionnode 특성



요약도표 노드는 다른 필드의 값에 상대적으로 하나의 숫자 필드의 값의 분포를 표시합니다. (히스토그램과 유사한 그래프를 작성합니다.) 값이 시간에 따라 변하는 변수 또는 필드를 설명하는 데 유용합니다. 3-D 그래프를 사용하여 범주별 분포를 표시하는 기호 축을 포함할 수도 있습니다.

예제

```
node = stream.create("collection", "My node")
# "Plot" tab
node.setPropertyValue("three_D", True)
node.setPropertyValue("collect_field", "Drug")
node.setPropertyValue("over_field", "Age")
node.setPropertyValue("by_field", "BP")
node.setPropertyValue("operation", "Sum")
# "Overlay" section
node.setPropertyValue("color_field", "Drug")
node.setPropertyValue("panel_field", "Sex")
node.setPropertyValue("animation_field", "")
# "Options" tab
node.setPropertyValue("range_mode", "Automatic")
node.setPropertyValue("range_min", 1)
node.setPropertyValue("range_max", 100)
node.setPropertyValue("bins", "ByNumber")
node.setPropertyValue("num_bins", 10)
node.setPropertyValue("bin_width", 5)
```

표 97. collectionnode 특성

collectionnode 특성	데이터 유형	특성 설명
over_field	field	
over_label_auto	flag	
over_label	string	
collect_field	field	
collect_label_auto	flag	
collect_label	string	
three_D	flag	
by_field	field	

표 97. *collectionmode* 특성 (계속)

collectionmode 특성	데이터 유형	특성 설명
by_label_auto	<i>flag</i>	
by_label	<i>string</i>	
operation	Sum Mean Min Max SDev	
color_field	<i>string</i>	
panel_field	<i>string</i>	
animation_field	<i>string</i>	
range_mode	Automatic UserDefined	
range_min	<i>number</i>	
range_max	<i>number</i>	
bins	ByNumber ByWidth	
num_bins	<i>number</i>	
bin_width	<i>number</i>	
use_grid	<i>flag</i>	
graph_background	<i>color</i>	표준 그래프 색상은 이 절의 시작부에서 설명됩니다.
page_background	<i>color</i>	표준 그래프 색상은 이 절의 시작부에서 설명됩니다.

distributionnode 특성



분포 노드는 대출 유형이나 성별 같은 기호적(범주형) 값의 발생을 보여줍니다. 일반적으로, 데이터의 불균형을 표시하기 위해 분포 노드를 사용하는 경우 모델을 작성하기 전에 균형 노드를 사용하여 교정할 수 있습니다.

예제

```
node = stream.create("distribution", "My node")
# "Plot" tab
node.setPropertyValue("plot", "Flags")
node.setPropertyValue("x_field", "Age")
node.setPropertyValue("color_field", "Drug")
node.setPropertyValue("normalize", True)
node.setPropertyValue("sort_mode", "ByOccurence")
node.setPropertyValue("use_proportional_scale", True)
```

표 98. *distributionnode* 특성

distributionnode 특성	데이터 유형	특성 설명
plot	SelectedFields Flags	
x_field	field	
color_field	field	오버레이 필드.
normalize	flag	
sort_mode	ByOccurence Alphabetic	
use_proportional_scale	flag	

evaluationnode 특성



평가 노드는 예측 모델을 평가하고 비교하는 데 도움이 됩니다. 평가 차트는 모델이 특정 결과를 얼마나 잘 예측하는지를 보여줍니다. 예측값과 예측의 신뢰도를 바탕으로 레코드를 정렬합니다. 레코드를 동일한 크기의 그룹(분위수)으로 분할한 후 각 분위수에 대한 비즈니스 기준의 값을 가장 높은 값부터 가장 낮은 값으로 도표를 그립니다. 다중 모델이 도표에 선구분 변수로 표시됩니다.

예제

```
node = stream.create("evaluation", "My node")
# "Plot" tab
node.setPropertyValue("chart_type", "Gains")
node.setPropertyValue("cumulative", False)
node.setPropertyValue("field_detection_method", "Name")
node.setPropertyValue("inc_baseline", True)
node.setPropertyValue("n_tile", "Deciles")
node.setPropertyValue("style", "Point")
node.setPropertyValue("point_type", "Dot")
node.setPropertyValue("use_fixed_cost", True)
node.setPropertyValue("cost_value", 5.0)
node.setPropertyValue("cost_field", "Na")
node.setPropertyValue("use_fixed_revenue", True)
node.setPropertyValue("revenue_value", 30.0)
node.setPropertyValue("revenue_field", "Age")
node.setPropertyValue("use_fixed_weight", True)
node.setPropertyValue("weight_value", 2.0)
node.setPropertyValue("weight_field", "K")
```

표 99. *evaluationnode* 특성.

evaluationnode 특성	데이터 유형	특성 설명
chart_type	Gains Response Lift Profit ROI ROC	
inc_baseline	flag	

표 99. *evaluationnode* 특성 (계속).

evaluationnode 특성	데이터 유형	특성 설명
field_detection_method	Metadata Name	
use_fixed_cost	flag	
cost_value	number	
cost_field	string	
use_fixed_revenue	flag	
revenue_value	number	
revenue_field	string	
use_fixed_weight	flag	
weight_value	number	
weight_field	field	
n_tile	Quartiles Quintles Deciles Vingtiles Percentiles 1000-tiles	
cumulative	flag	
style	Line Point	
point_type	Rectangle Dot Triangle Hexagon Plus Pentagon Star BowTie HorizontalDash VerticalDash IronCross Factory House Cathedral OnionDome ConcaveTriangle OblateGlobe CatEye FourSidedPillow RoundRectangle Fan	
export_data	flag	
data_filename	string	
delimiter	string	
new_line	flag	
inc_field_names	flag	
inc_best_line	flag	

표 99. *evaluationnode* 특성 (계속).

evaluationnode 특성	데이터 유형	특성 설명
inc_business_rule	<i>flag</i>	
business_rule_condition	<i>string</i>	
plot_score_fields	<i>flag</i>	
score_fields	<i>[field1 ... fieldN]</i>	
target_field	<i>field</i>	
use_hit_condition	<i>flag</i>	
hit_condition	<i>string</i>	
use_score_expression	<i>flag</i>	
score_expression	<i>string</i>	
caption_auto	<i>flag</i>	

graphboardnode 특성



그래프보드 노드는 하나의 단일 노드에 있는 여러 가지 유형의 많은 그래프를 제공합니다. 이 노드를 사용하여 탐색하려는 데이터 필드를 선택하고 선택된 데이터에 대해 사용 가능한 것 중에서 그래프를 선택할 수 있습니다. 이 노드는 필드 선택사항에 대해 작업하지 않는 모든 그래프 유형을 자동으로 필터링합니다.

참고: 그래프 유형에 올바르지 않은 특성을 설정하는 경우(예를 들어, 히스토그램에 대해 *y_field* 지정), 해당 특성은 무시됩니다.

참고: UI에서 여러 가지 많은 그래프 유형의 세부사항 탭에 **요약** 필드가 있습니다. 이 필드는 현재 스크립팅에 의해 지원되지 않습니다.

예제

```
node = stream.create("graphboard", "My node")
node.setPropertyValue("graph_type", "Line")
node.setPropertyValue("x_field", "K")
node.setPropertyValue("y_field", "Na")
```

표 100. graphboardnode 특성

graphboard 특성	데이터 유형	특성 설명
graph_type	2DDotplot 3DArea 3DBar 3DDensity 3DHistogram 3DPie 3DScatterplot Area ArrowMap Bar BarCounts BarCountsMap BarMap BinnedScatter Boxplot Bubble ChoroplethMeans ChoroplethMedians ChoroplethSums ChoroplethValues ChoroplethCounts CoordinateMap CoordinateChoroplethMeans CoordinateChoroplethMedians CoordinateChoroplethSums CoordinateChoroplethValues CoordinateChoroplethCounts Dotplot Heatmap HexBinScatter Histogram Line LineChartMap LineOverlayMap Parallel Path Pie PieCountMap PieCounts PieMap PointOverlayMap PolygonOverlayMap Ribbon Scatterplot SPLOM Surface	그래프 유형을 식별합니다.

표 100. graphboardnode 특성 (계속)

graphboard 특성	데이터 유형	특성 설명
x_field	field	x축의 사용자 정의 레이블을 지정합니다. 레이블에만 사용 가능합니다.
y_field	field	y축의 사용자 정의 레이블을 지정합니다. 레이블에만 사용 가능합니다.
z_field	field	일부 3-D 그래프에서 사용합니다.
color_field	field	히트 맵에서 사용합니다.
size_field	field	버블 도표에서 사용합니다.
categories_field	field	
values_field	field	
rows_field	field	
columns_field	field	
fields	field	
start_longitude_field	field	참조 맵에서 화살표와 함께 사용합니다.
end_longitude_field	field	
start_latitude_field	field	
end_latitude_field	field	
data_key_field	field	다양한 맵에서 사용합니다.
panelrow_field	string	
panelcol_field	string	
animation_field	string	
longitude_field	field	맵에서 좌표와 함께 사용합니다.
latitude_field	field	
map_color_field	field	

histogramnode 특성



히스토그램 노드는 수치 필드에 대한 값의 발생을 표시합니다. 보통 조작 및 모델 작성 전에 데이터를 탐색하는 데 사용합니다. 분포 노드와 비슷하게, 히스토그램 노드는 자주 데이터의 불균형을 드러내 보입니다.

예제

```
node = stream.create("histogram", "My node")
# "Plot" tab
node.setPropertyValue("field", "Drug")
node.setPropertyValue("color_field", "Drug")
node.setPropertyValue("panel_field", "Sex")
node.setPropertyValue("animation_field", "")
# "Options" tab
node.setPropertyValue("range_mode", "Automatic")
node.setPropertyValue("range_min", 1.0)
node.setPropertyValue("range_max", 100.0)
node.setPropertyValue("num_bins", 10)
```



```
node.setPropertyValue("bin_width", 10)
node.setPropertyValue("normalize", True)
node.setPropertyValue("separate_bands", False)
```

표 101. *histogramnode* 특성

histogramnode 특성	데이터 유형	특성 설명
field	<i>field</i>	
color_field	<i>field</i>	
panel_field	<i>field</i>	
animation_field	<i>field</i>	
range_mode	Automatic UserDefined	
range_min	<i>number</i>	
range_max	<i>number</i>	
bins	ByNumber ByWidth	
num_bins	<i>number</i>	
bin_width	<i>number</i>	
normalize	<i>flag</i>	
separate_bands	<i>flag</i>	
x_label_auto	<i>flag</i>	
x_label	<i>string</i>	
y_label_auto	<i>flag</i>	
y_label	<i>string</i>	
use_grid	<i>flag</i>	
graph_background	<i>color</i>	표준 그래프 색상은 이 절의 시작부에서 설명됩니다.
page_background	<i>color</i>	표준 그래프 색상은 이 절의 시작부에서 설명됩니다.
normal_curve	<i>flag</i>	정규 분포 곡선이 출력에 표시되어야 하는지 여부를 표시합니다.

mapvisualization 특성



맵 시각화 노드는 다중 입력 연결을 승인하고 지리 공간적 데이터를 맵에 일련의 레이어로 표시할 수 있습니다. 각각의 레이어는 하나의 지리 공간적 필드입니다. 예를 들어, 기준 레이어가 한 국가의 맵이고 그 위에 도로에 대한 레이어 하나, 강에 대한 레이어 하나, 도시에 대한 레이어 하나가 있을 수 있습니다.

표 102. mapvisualization 특성

mapvisualization 특성	데이터 유형	특성 설명
tag	string	입력의 태그 이름을 설정합니다. 기본 태그는 입력이 노드에 연결된 순서를 기반으로 하는 번호입니다(첫 번째 연결 태그는 1이고 두 번째 연결 태그는 2 등임).
layer_field	field	<p>데이터 세트에서 맵의 레이어로 표시되는 geo-field를 선택합니다. 기본 선택사항은 다음 정렬 순서를 기반으로 합니다.</p> <ul style="list-style-type: none"> • 첫 번째 - 점 • 선 스트링 • 다각형 • 다중 점 • 다중 선 스트링 • 마지막 - 다중 다각형 <p>측정 유형이 동일한 필드가 두 개 있으면 기본적으로 알파벳순(이름별)으로 첫 번째 필드가 선택됩니다.</p>
color_type	boolean	표준 색상을 geo-field의 모든 기능에 적용하는지 아니면 데이터 세트의 다른 필드 값을 기반으로 기능의 색상을 지정하는 오버레이 필드를 적용하는지 지정합니다. 가능한 값은 standard 또는 overlay입니다. 기본값은 standard입니다.
color	string	<p>color_type으로 standard가 선택되면 드롭 다운에 사용자 옵션 표시 탭의 차트 카테고리 색상 순서와 동일한 색상표를 포함합니다.</p> <p>기본값은 차트 카테고리 색상 1입니다.</p>
color_field	field	color_type으로 overlay가 선택되면 드롭 다운에 레이어로 선택된 geo-field와 동일한 데이터 세트의 모든 필드가 포함됩니다.
symbol_type	boolean	표준 기호를 geo-field의 모든 레코드에 적용하는지 아니면 데이터 세트의 다른 필드 값을 기반으로 점의 기호 아이콘을 변경하는 오버레이 기호를 적용하는지 지정합니다. 가능한 값은 standard 또는 overlay입니다. 기본값은 standard입니다.
symbol	string	symbol_type으로 standard를 선택한 경우 드롭 다운에 맵에 점을 표시하는 데 사용할 수 있는 기호의 선택이 포함됩니다.
symbol_field	field	symbol_type으로 overlay가 선택되면 드롭 다운에 레이어로 선택된 geo-field와 동일한 데이터 세트의 모든 명목, 순서 또는 범주형 필드가 포함됩니다.
size_type	boolean	표준 크기를 geo-field의 모든 레코드에 적용하는지 아니면 데이터 세트의 다른 필드 값을 기반으로 기호 아이콘의 크기 또는 선의 굵기를 변경하는 오버레이 크기를 적용하는지 지정합니다. 가능한 값은 standard 또는 overlay입니다. 기본값은 standard입니다.

표 102. mapvisualization 특성 (계속)

mapvisualization 특성	데이터 유형	특성 설명
size	string	size_type으로 standard가 선택되면 point 또는 multipoint의 드롭 다운에 선택한 기호의 크기 선택 사항이 포함됩니다. linestring 또는 multilinestring의 경우 드롭 다운에 선 굵기 선택 사항이 포함됩니다.
size_field	field	size_type으로 overlay가 선택되면 드롭 다운에 레이어로 선택된 geo-field와 동일한 데이터 세트의 모든 필드가 포함됩니다.
transp_type	boolean	표준 투명도를 geo-field의 모든 레코드에 적용하는지 아니면 데이터 세트의 다른 필드 값을 기반으로 기호, 선 또는 다각형의 투명도 레벨을 변경하는 오버레이 투명도를 적용하는지 지정합니다. 가능한 값은 standard 또는 overlay입니다. 기본값은 standard입니다.
transp	integer	transp_type으로 standard가 선택되면 드롭 다운에 0%(불투명)부터 시작하여 10%씩 증분되어 100%(투명)까지 증가되는 투명도 레벨의 선택사항이 포함됩니다. 맵에서 점, 선 또는 다각형의 투명도를 설명합니다. size_type으로 overlay가 선택되면 드롭 다운에 레이어로 선택된 geo-field와 동일한 데이터 세트의 모든 필드가 포함됩니다. points, multipoints, linestrings 및 multilinestrings, polygons 및 multipolygons(맨 아래 레이어)의 경우 기본값은 0%입니다. 맨 아래 레이어가 아닌 polygons 및 multipolygons의 기본값은 50%입니다. 이 경우 이러한 다각형 아래의 명확하지 않은 레이어를 방지할 수 있습니다.
transp_field	field	transp_type으로 overlay가 선택되면 드롭 다운에 레이어로 선택된 geo-field와 동일한 데이터 세트의 모든 필드가 포함됩니다.
data_label_field	field	맵에서 데이터 레이블로 사용할 필드를 지정합니다. 예를 들어, 이 설정이 적용된 레이어가 다각형 레이어이면 데이터 레이블이 각 다각형의 이름을 포함하는 name 필드여야 합니다. 따라서 여기서 name 필드를 선택하면 해당 이름이 맵에 표시됩니다.
use_hex_binning	boolean	16진 구간화를 사용하고 모든 통합 드롭 다운을 사용합니다. 이 설정은 기본적으로 꺼집니다.

표 102. *mapvisualization* 특성 (계속)

mapvisualization 특성	데이터 유형	특성 설명
color_aggregation 및 transp_aggregation	string	<p>16진 구간화를 사용하여 점 레이어로 오버레이 필드를 선택하면 6각형에 있는 모든 점을 위해 해당 필드의 모든 값을 통합해야 합니다. 그러므로 맵에 적용할 오버레이 필드의 통합 함수를 지정해야 합니다.</p> <p>사용 가능한 통합 함수는 다음과 같습니다.</p> <p>연속(실수 또는 정수 저장 공간):</p> <ul style="list-style-type: none"> • 합계 • 평균 • 최소값 • 최대값 • 중앙값 • 제1사분위수 • 제3사분위수 <p>연속(시간, 날짜 또는 시간소인 저장 공간):</p> <ul style="list-style-type: none"> • 평균 • 최소값 • 최대값 <p>명목/범주:</p> <ul style="list-style-type: none"> • 모드 • 최소값 • 최대값 <p>플래그:</p> <ul style="list-style-type: none"> • 참일 경우 True • 거짓일 경우 False
custom_storage	string	<p>필드의 전체 저장 유형을 설정합니다. 기본값은 List입니다. List가 지정된 경우 다음 custom_value_storage 및 list_depth 제어를 사용하지 않습니다.</p>
custom_value_storage	string	<p>필드에 있는 요소의 저장 유형을 일률적으로 설정하지 않고, 목록에 있는 요소의 저장 유형을 설정합니다. 기본값은 Real입니다.</p>

표 102. *mapvisualization* 특성 (계속)

mapvisualization 특성	데이터 유형	특성 설명
list_depth	integer	<p>목록 필드의 깊이를 설정합니다. 필수 깊이는 다음 기준에 따른 geofield의 유형에 따라 다릅니다.</p> <ul style="list-style-type: none"> • Point - 0 • LineString - 1 • Polygon - 2 • Multipoint - 1 • MultiLineString - 2 • Multipolygon - 3 <p>목록으로 다시 변환하는 지리 공간 필드의 유형과 해당 필드 유형에 필요한 깊이를 알아야 합니다. 올바르게 설정한 경우 필드를 사용할 수 없습니다.</p> <p>기본값은 0이고 최소값은 0이며 최대값은 10입니다.</p>

multiplotnode 특성



다중 도표 노드는 단일 X 필드 위에 다중 Y 필드를 표시하는 도표를 작성합니다. Y 필드는 색상이 지정된 선으로 도표됩니다. 각각은 스타일이 **Line**으로 설정되고 X 모드가 **Sort**로 설정된 구성 노드와 동등합니다. 다중 도표는 시간에 따라서 여러 변수의 변동을 탐색하기 원할 때 유용합니다.

예제

```
node = stream.create("multiplot", "My node")
# "Plot" tab
node.setPropertyValue("x_field", "Age")
node.setPropertyValue("y_fields", ["Drug", "BP"])
node.setPropertyValue("panel_field", "Sex")
# "Overlay" section
node.setPropertyValue("animation_field", "")
node.setPropertyValue("tooltip", "test")
node.setPropertyValue("normalize", True)
node.setPropertyValue("use_overlay_expr", False)
node.setPropertyValue("overlay_expression", "test")
node.setPropertyValue("records_limit", 500)
node.setPropertyValue("if_over_limit", "PlotSample")
```

표 103. *multiplotnode* 특성

multiplotnode 특성	데이터 유형	특성 설명
x_field	field	
y_fields	list	
panel_field	field	

표 103. *multiplotnode* 특성 (계속)

multiplotnode 특성	데이터 유형	특성 설명
animation_field	<i>field</i>	
normalize	<i>flag</i>	
use_overlay_expr	<i>flag</i>	
overlay_expression	<i>string</i>	
records_limit	<i>number</i>	
if_over_limit	PlotBins PlotSample PlotAll	
x_label_auto	<i>flag</i>	
x_label	<i>string</i>	
y_label_auto	<i>flag</i>	
y_label	<i>string</i>	
use_grid	<i>flag</i>	
graph_background	<i>color</i>	표준 그래프 색상은 이 절의 시작부에서 설명됩니다.
page_background	<i>color</i>	표준 그래프 색상은 이 절의 시작부에서 설명됩니다.

plotnode 특성



구성 노드는 수치 필드 사이의 관계를 보여줍니다. 포인트(산점도) 또는 선을 사용하여 도표를 작성할 수 있습니다.

예제

```
node = stream.create("plot", "My node")
# "Plot" tab
node.setPropertyValue("three_D", True)
node.setPropertyValue("x_field", "BP")
node.setPropertyValue("y_field", "Cholesterol")
node.setPropertyValue("z_field", "Drug")
# "Overlay" section
node.setPropertyValue("color_field", "Drug")
node.setPropertyValue("size_field", "Age")
node.setPropertyValue("shape_field", "")
node.setPropertyValue("panel_field", "Sex")
node.setPropertyValue("animation_field", "BP")
node.setPropertyValue("transp_field", "")
node.setPropertyValue("style", "Point")
# "Output" tab
node.setPropertyValue("output_mode", "File")
node.setPropertyValue("output_format", "JPEG")
node.setPropertyValue("full_filename", "C:/temp/graph_output/plot_output.jpeg")
```

표 104. plotnode 특성.

plotnode 특성	데이터 유형	특성 설명
x_field	field	x축의 사용자 정의 레이블을 지정합니다. 레이블에만 사용 가능합니다.
y_field	field	y축의 사용자 정의 레이블을 지정합니다. 레이블에만 사용 가능합니다.
three_D	flag	y축의 사용자 정의 레이블을 지정합니다. 3-D 그래프의 레이블에만 사용 가능합니다.
z_field	field	
color_field	field	오버레이 필드.
size_field	field	
shape_field	field	
panel_field	field	각 범주에 대한 별도의 차트를 만들기 위해 사용할 명목형 또는 플래그 필드를 지정합니다. 도표는 하나의 출력 창에 함께 작성됩니다.
animation_field	field	범주형을 사용하여 순차적으로 표시되는 일련의 차트를 작성하여 데이터 값 범주를 설명하기 위한 명목형 또는 플래그 필드를 지정합니다.
transp_field	field	각 범주에 대해 상이한 수준의 투명도를 사용하여 데이터 값 범주를 설명하는 필드를 지정합니다. 선 도표에는 사용할 수 없습니다.
overlay_type	None Smoother Function	오버레이 함수나 LOESS 다듬기가 표시되는지 여부를 지정합니다.
overlay_expression	string	overlay_type이 Function으로 설정될 때 사용하는 표현식을 지정합니다.
style	Point Line	
point_type	Rectangle Dot Triangle Hexagon Plus Pentagon Star BowTie HorizontalDash VerticalDash IronCross Factory House Cathedral OnionDome ConcaveTriangle OblateGlobe CatEye FourSidedPillow RoundRectangle Fan	

표 104. *plotnode* 특성 (계속).

plotnode 특성	데이터 유형	특성 설명
x_mode	Sort Overlay AsRead	
x_range_mode	Automatic UserDefined	
x_range_min	<i>number</i>	
x_range_max	<i>number</i>	
y_range_mode	Automatic UserDefined	
y_range_min	<i>number</i>	
y_range_max	<i>number</i>	
z_range_mode	Automatic UserDefined	
z_range_min	<i>number</i>	
z_range_max	<i>number</i>	
jitter	<i>flag</i>	
records_limit	<i>number</i>	
if_over_limit	PlotBins PlotSample PlotAll	
x_label_auto	<i>flag</i>	
x_label	<i>string</i>	
y_label_auto	<i>flag</i>	
y_label	<i>string</i>	
z_label_auto	<i>flag</i>	
z_label	<i>string</i>	
use_grid	<i>flag</i>	
graph_background	<i>color</i>	표준 그래프 색상은 이 절의 시작부에서 설명됩니다.
page_background	<i>color</i>	표준 그래프 색상은 이 절의 시작부에서 설명됩니다.
use_overlay_expr	<i>flag</i>	overlay_type을 위해 더 이상 사용하지 않습니다.

timeplotnode 특성



시간 구성 노드는 하나 이상의 시계열 데이터 세트를 표시합니다. 일반적으로, 먼저 시간 구간 노드를 사용하여 *TimeLabel* 필드를 작성하는데, 이것이 *x*축을 레이블하는 데 사용됩니다.

예제


```

node = stream.create("timeplot", "My node")
node.setPropertyValue("y_fields", ["sales", "men", "women"])
node.setPropertyValue("panel", True)
node.setPropertyValue("normalize", True)
node.setPropertyValue("line", True)
node.setPropertyValue("smoother", True)
node.setPropertyValue("use_records_limit", True)
node.setPropertyValue("records_limit", 2000)
# Appearance settings
node.setPropertyValue("symbol_size", 2.0)

```

표 105. *timeplotnode* 특성.

timeplotnode 특성	데이터 유형	특성 설명
plot_series	Series Models	
use_custom_x_field	<i>flag</i>	
x_field	<i>field</i>	
y_fields	<i>list</i>	
panel	<i>flag</i>	
normalize	<i>flag</i>	
line	<i>flag</i>	
points	<i>flag</i>	
point_type	Rectangle Dot Triangle Hexagon Plus Pentagon Star BowTie HorizontalDash VerticalDash IronCross Factory House Cathedral OnionDome ConcaveTriangle OblateGlobe CatEye FourSidedPillow RoundRectangle Fan	
smoother	<i>flag</i>	panel을 True로 설정한 경우에만 도표에 다듬기를 추가할 수 있습니다.
use_records_limit	<i>flag</i>	
records_limit	<i>integer</i>	
symbol_size	<i>number</i>	기호 크기를 지정합니다.
panel_layout	Horizontal Vertical	

webnode 특성



웹 노드는 둘 이상의 기호(범주형) 필드의 값 사이의 관계의 강도를 설명합니다. 그래프는 다양한 너비의 선을 사용하여 연결 강도를 표시합니다. 예를 들어 웹 노드를 사용하여 전자상거래 사이트에 있는 항목 세트의 구매 사이의 관계를 탐색할 수 있습니다.

예제

```
node = stream.create("web", "My node")
# "Plot" tab
node.setPropertyValue("use_directed_web", True)
node.setPropertyValue("to_field", "Drug")
node.setPropertyValue("fields", ["BP", "Cholesterol", "Sex", "Drug"])
node.setPropertyValue("from_fields", ["BP", "Cholesterol", "Sex"])
node.setPropertyValue("true_flags_only", False)
node.setPropertyValue("line_values", "Absolute")
node.setPropertyValue("strong_links_heavier", True)
# "Options" tab
node.setPropertyValue("max_num_links", 300)
node.setPropertyValue("links_above", 10)
node.setPropertyValue("num_links", "ShowAll")
node.setPropertyValue("discard_links_min", True)
node.setPropertyValue("links_min_records", 5)
node.setPropertyValue("discard_links_max", True)
node.setPropertyValue("weak_below", 10)
node.setPropertyValue("strong_above", 19)
node.setPropertyValue("link_size_continuous", True)
node.setPropertyValue("web_display", "Circular")
```

표 106. webnode 특성

webnode 특성	데이터 유형	특성 설명
use_directed_web	<i>flag</i>	
fields	<i>list</i>	
to_field	<i>field</i>	
from_fields	<i>list</i>	
true_flags_only	<i>flag</i>	
line_values	Absolute OverallPct PctLarger PctSmaller	
strong_links_heavier	<i>flag</i>	
num_links	ShowMaximum ShowLinksAbove ShowAll	
max_num_links	<i>number</i>	
links_above	<i>number</i>	

표 106. *webnode* 특성 (계속)

webnode 특성	데이터 유형	특성 설명
discard_links_min	<i>flag</i>	
links_min_records	<i>number</i>	
discard_links_max	<i>flag</i>	
links_max_records	<i>number</i>	
weak_below	<i>number</i>	
strong_above	<i>number</i>	
link_size_continuous	<i>flag</i>	
web_display	Circular Network Directed Grid	
graph_background	<i>color</i>	표준 그래프 색상은 이 절의 시작부에서 설명됩니다.
symbol_size	<i>number</i>	기호 크기를 지정합니다.

제 13 장 모델링 노드 특성

공통 모델링 노드 특성

다음 특성은 일부 또는 모든 모델링 노드에 공통적입니다. 모든 예외는 개별 모델링 노드에 대한 문서에 적절하게 언급됩니다.

표 107. 공통 모델링 노드 특성

특성	값	특성 설명
custom_fields	<i>flag</i>	True인 경우 현재 노드에 대한 목표, 입력 및 기타 필드를 지정할 수 있습니다. False인 경우 업스트림 유형 노드의 현재 설정을 사용합니다.
target 또는 targets	<i>field</i> 또는 [<i>field1</i> ... <i>fieldN</i>]	모델 유형에 따라서 단일 목표 필드 또는 다중 목표 필드를 지정합니다.
inputs	[<i>field1</i> ... <i>fieldN</i>]	모델에서 사용하는 입력 또는 예측변수 필드입니다.
partition	<i>field</i>	
use_partitioned_data	<i>flag</i>	파티션 필드가 정의된 경우, 이 옵션은 훈련 파티션의 데이터만 모델을 작성하는 데 사용하도록 보장합니다.
use_split_data	<i>flag</i>	
splits	[<i>field1</i> ... <i>fieldN</i>]	분할 모델링에 사용할 필드를 지정합니다. use_split_data가 True로 설정된 경우에만 유효합니다.
use_frequency	<i>flag</i>	가중치 및 빈도 필드는 각 모델 유형에 대해 언급되는 대로 특정 모델이 사용됩니다.
frequency_field	<i>field</i>	
use_weight	<i>flag</i>	
weight_field	<i>field</i>	
use_model_name	<i>flag</i>	
model_name	<i>string</i>	새 모델의 사용자 정의 이름입니다.
mode	Simple Expert	

anomalydetectionnode 특성



이상 항목 발견 노드는 "정상" 데이터 패턴을 따르지 않는 특이 케이스 또는 이상값을 식별합니다. 이 노드를 사용하면 이전에 알려진 패턴에 적합하지 않고, 찾고 있는 패턴을 정확하게 모르더라도 이상값을 식별할 수 있습니다.

예제

```
node = stream.create("anomalydetection", "My node")
node.setPropertyValue("anomaly_method", "PerRecords")
node.setPropertyValue("percent_records", 95)
node.setPropertyValue("mode", "Expert")
node.setPropertyValue("peer_group_num_auto", True)
node.setPropertyValue("min_num_peer_groups", 3)
node.setPropertyValue("max_num_peer_groups", 10)
```

표 108. anomalydetectionnode 특성

anomalydetectionnode 특성	값	특성 설명
inputs	[field1 ... fieldN]	이상 항목 발견은 지정된 입력 필드를 기반으로 화면 레코드를 모델링합니다. 목표 필드를 사용하지 않습니다. 가중치 및 빈도 필드도 사용하지 않습니다. 자세한 정보는 191 페이지의 『공통 모델링 노드 특성』 주제를 참조하십시오.
mode	Expert Simple	
anomaly_method	IndexLevel PerRecords NumRecords	레코드를 이상 항목으로 플래그 지정하기 위한 절사 값을 판별하는 데 사용하는 방법을 지정합니다.
index_level	number	플래그 지정 이상 항목의 최소 절사 값을 지정합니다.
percent_records	number	학습 데이터에 있는 레코드의 퍼센트를 기반으로 플래그 지정 레코드에 대한 임계값을 설정합니다.
num_records	number	학습 데이터에 있는 레코드의 수를 기반으로 플래그 지정 레코드에 대한 임계값을 설정합니다.
num_fields	integer	각 이상 항목 레코드에 대해 보고할 필드 수입니다.
impute_missing_values	flag	
adjustment_coeff	number	거리 계산에서 연속형 및 범주형 필드에 주어진 상대값 가중치의 균형을 잡는 데 사용하는 값입니다.
peer_group_num_auto	flag	동등 집단 수를 자동으로 계산합니다.
min_num_peer_groups	integer	peer_group_num_auto가 True로 설정될 때 최소 동등 집단 수를 지정합니다.
max_num_per_groups	integer	최대 동등 집단 수를 지정합니다.
num_peer_groups	integer	peer_group_num_auto가 False로 설정될 때 사용하는 동등 집단 수를 지정합니다.
noise_level	number	이상값이 군집 중에 처리되는 방법을 판별합니다. 0과 0.5 사이의 값을 지정하십시오.
noise_ratio	number	잡음 버퍼링에 사용해야 하는 구성요소에 할당되는 메모리 부분을 지정합니다. 0과 0.5 사이의 값을 지정하십시오.

apriorinode 특성



Apriori 노드는 데이터에서 규칙 세트를 추출하고 정보 내용이 가장 많은 규칙을 꺼냅니다. Apriori는 규칙을 선택하는 5개의 서로 다른 방법을 제공하며 정교한 색인화 스킴을 사용하여 대형 데이터 세트를 효율적으로 처리합니다. 큰 문제점의 경우, Apriori는 일반적으로 훈련 속도가 빠릅니다. 보유할 수 있는 규칙 수에 임의의 제한이 없으며 최대 32개의 전제조건을 가진 규칙을 처리할 수 있습니다. Apriori에서는 입력 및 출력 필드가 모두 범주형이어야 하지만 이런 유형의 데이터에 최적화되어 있기 때문에 우수한 성능을 제공합니다.

예제

```
node = stream.create("apriori", "My node")
# "Fields" tab
node.setPropertyValue("custom_fields", True)
node.setPropertyValue("partition", "Test")
# For non-transactional
node.setPropertyValue("use_transactional_data", False)
node.setPropertyValue("consequents", ["Age"])
node.setPropertyValue("antecedents", ["BP", "Cholesterol", "Drug"])
# For transactional
node.setPropertyValue("use_transactional_data", True)
node.setPropertyValue("id_field", "Age")
node.setPropertyValue("contiguous", True)
node.setPropertyValue("content_field", "Drug")
# "Model" tab
node.setPropertyValue("use_model_name", False)
node.setPropertyValue("model_name", "Apriori_bp_choles_drug")
node.setPropertyValue("min_supp", 7.0)
node.setPropertyValue("min_conf", 30.0)
node.setPropertyValue("max_antecedents", 7)
node.setPropertyValue("true_flags", False)
node.setPropertyValue("optimize", "Memory")
# "Expert" tab
node.setPropertyValue("mode", "Expert")
node.setPropertyValue("evaluation", "ConfidenceRatio")
node.setPropertyValue("lower_bound", 7)
```

표 109. apriorinode 특성

apriorinode 특성	값	특성 설명
consequents	<i>field</i>	Apriori 모델은 표준 대상 및 입력 필드 대신 후향 값(Consequents) 및 전향(Antecedents)을 사용합니다. 가중치 및 빈도 필드는 사용하지 않습니다. 자세한 정보는 191 페이지의 『공통 모델링 노드 특성』 주제를 참조하십시오.
antecedents	<i>[field1 ... fieldN]</i>	
min_supp	<i>number</i>	
min_conf	<i>number</i>	
max_antecedents	<i>number</i>	
true_flags	<i>flag</i>	

표 109. *apriorinode* 특성 (계속)

apriorinode 특성	값	특성 설명
optimize	Speed Memory	
use_transactional_data	<i>flag</i>	
contiguous	<i>flag</i>	
id_field	<i>string</i>	
content_field	<i>string</i>	
mode	Simple Expert	
evaluation	RuleConfidence DifferenceToPrior ConfidenceRatio InformationDifference NormalizedChiSquare	
lower_bound	<i>number</i>	
optimize	Speed Memory	모델 작성이 속도 또는 메모리에 대해 최적화되어야 하는지 여부를 지정하는 데 사용됩니다.

associationrulesnode 특성



연관 규칙 노드는 Apriori 노드와 유사합니다. 그러나 Apriori와는 달리, 연관 규칙 노드는 목록 데이터를 처리할 수 있습니다. 또한, 연관 규칙 노드는 빅 데이터를 처리하고 더 빠른 병렬 처리를 사용하기 위해 IBM SPSS Analytic Server와 함께 사용할 수 있습니다.

표 110. *associationrulesnode* 특성

associationrulesnode 특성	데이터 유형	특성 설명
predictions	<i>field</i>	이 목록의 필드는 규칙의 예측변수로서만 나타날 수 있습니다.
conditions	<i>[field1...fieldN]</i>	이 목록의 필드는 규칙의 조건으로서만 나타날 수 있습니다.
max_rule_conditions	<i>integer</i>	단일 규칙에 포함될 수 있는 조건의 최대 수입니다. 최소 1, 최대 9.
max_rule_predictions	<i>integer</i>	단일 규칙에 포함될 수 있는 예측의 최대 수입니다. 최소 1, 최대 5.
max_num_rules	<i>integer</i>	규칙 작성의 일부로 고려될 수 있는 규칙의 최대 수입니다. 최소 1, 최대 10,000.
rule_criterion_top_n	Confidence Rulesupport Lift Conditionsupport Deployability	모델의 최상위 "N" 규칙이 선택되는 값을 판별하는 규칙 기준입니다.

표 110. associationrulesnode 특성 (계속)

associationrulesnode 특성	데이터 유형	특성 설명
true_flags	부울	Y로 설정하면 플래그 필드에 대한 참 값만 규칙 작성 중에 고려됨을 판별합니다.
rule_criterion	부울	Y로 설정하면 규칙 기준 값이 모델 작성 중에 규칙 제외를 위해 사용됨을 판별합니다.
min_confidence	number	0.1부터 100까지 - 모델에 의해 생성되는 규칙에 대한 최소 필수 신뢰수준에 대한 퍼센트 값입니다. 모델이 여기에 지정된 값보다 작은 신뢰수준을 갖는 규칙을 생성하는 경우 해당 규칙은 삭제됩니다.
min_rule_support	number	0.1부터 100까지 - 모델에 의해 생성되는 규칙에 대한 최소 필수 규칙 지원에 대한 퍼센트 값입니다. 모델이 지정된 값보다 작은 규칙 지원 수준을 갖는 규칙을 생성하는 경우 해당 규칙은 삭제됩니다.
min_condition_support	number	0.1부터 100까지 - 모델에 의해 생성되는 규칙에 대한 최소 필수 조건 지원에 대한 퍼센트 값입니다. 모델이 지정된 값보다 작은 조건 지원 수준을 갖는 규칙을 생성하는 경우 해당 규칙은 삭제됩니다.
min_lift	integer	1부터 10까지 - 모델에 의해 생성되는 규칙에 대한 최소 필수 리프트를 나타냅니다. 모델이 지정된 값보다 작은 리프트 수준을 갖는 규칙을 생성하는 경우 해당 규칙은 삭제됩니다.
exclude_rules	부울	모델이 규칙을 작성하기 원하지 않는 관련 필드의 목록을 선택하는 데 사용됩니다. 예: set :gsarsnode.exclude_rules = [[[field1,field2, field3]],[[field4, field5]]] - []에 의해 구분되는 각 필드 목록이 테이블의 한 행입니다.
num_bins	integer	연속형 필드가 구간화되는 자동 구간 수를 설정합니다. 최소 2, 최대 10.
max_list_length	integer	최대 길이를 알 수 없는 모든 목록 필드에 적용됩니다. 여기에 지정되는 숫자까지의 목록에 있는 요소가 모델 작성에 포함됩니다. 모든 추가 요소는 삭제됩니다. 최소 1, 최대 100.
output_confidence	부울	
output_rule_support	부울	
output_lift	부울	
output_condition_support	부울	
output_deployability	부울	
rules_to_display	upto all	출력 테이블에 표시할 규칙의 최대 수입니다.
display_upto	integer	upto가 rules_to_display에서 설정되는 경우, 출력 테이블에 표시할 규칙 수를 설정하십시오. 최소 1.
field_transformations	부울	
records_summary	부울	
rule_statistics	부울	
most_frequent_values	부울	

표 110. associationrulesnode 특성 (계속)

associationrulesnode 특성	데이터 유형	특성 설명
most_frequent_fields	부울	
word_cloud	부울	
word_cloud_sort	Confidence Rulesupport Lift Conditionsupport Deployability	
word_cloud_display	integer	최소 1, 최대 20
max_predictions	integer	스코어에 대한 각 입력에 적용될 수 있는 최대 규칙 수입니다.
criterion	Confidence Rulesupport Lift Conditionsupport Deployability	규칙의 강도를 판별하는 데 사용하는 측도를 선택하십시오.
allow_repeats	부울	동일한 예측을 갖는 규칙이 스코어에 포함되는지 여부를 판별합니다.
check_input	NoPredictions Predictions NoCheck	

autoclassifiernode 특성



자동 분류자 노드는 이분형 결과(예 또는 아니오, 이탈 또는 이탈 안함 등)에 대해 다수의 여러 모델을 작성하고 비교하여 주어진 분석을 위한 최상의 접근 방식을 선택할 수 있게 합니다. 많은 모델링 알고리즘이 지원되어 사용할 방법, 각각에 대한 특정 옵션, 결과 비교 기준을 선택할 수 있습니다. 이 노드는 지정된 옵션을 기반으로 모델 세트를 생성하고 사용자가 지정하는 기준에 따라 최상의 후보를 순위화합니다.

예제

```
node = stream.create("autoclassifier", "My node")
node.setPropertyValue("ranking_measure", "Accuracy")
node.setPropertyValue("ranking_dataset", "Training")
node.setPropertyValue("enable_accuracy_limit", True)
node.setPropertyValue("accuracy_limit", 0.9)
node.setPropertyValue("calculate_variable_importance", True)
node.setPropertyValue("use_costs", True)
node.setPropertyValue("svm", False)
```

표 111. *autoclassifiernode* 특성.

autoclassifiernode 특성	값	특성 설명
target	<i>field</i>	플래그 대상의 경우, 자동 분류자 노드는 하나의 대상과 하나 이상의 입력 필드가 필요합니다. 가중치 및 빈도 필드도 지정할 수 있습니다. 자세한 정보는 191 페이지의 『공통 모델링 노드 특성』 주제를 참조하십시오.
ranking_measure	Accuracy Area_under_curve Profit Lift Num_variables	
ranking_dataset	Training Test	
number_of_models	<i>integer</i>	모델 너깃에 포함할 모델 수입니다. 1과 100 사이의 정수를 지정하십시오.
calculate_variable_importance	<i>flag</i>	
enable_accuracy_limit	<i>flag</i>	
accuracy_limit	<i>integer</i>	0과 100 사이의 정수입니다.
enable_area_under_curve_limit	<i>flag</i>	
area_under_curve_limit	<i>number</i>	0.0과 1.0 사이의 실수입니다.
enable_profit_limit	<i>flag</i>	
profit_limit	<i>number</i>	0보다 큰 정수입니다.
enable_lift_limit	<i>flag</i>	
lift_limit	<i>number</i>	1.0보다 큰 실수입니다.
enable_number_of_variables_limit	<i>flag</i>	
number_of_variables_limit	<i>number</i>	0보다 큰 정수입니다.
use_fixed_cost	<i>flag</i>	
fixed_cost	<i>number</i>	0.0보다 큰 실수입니다.
variable_cost	<i>field</i>	
use_fixed_revenue	<i>flag</i>	
fixed_revenue	<i>number</i>	0.0보다 큰 실수입니다.
variable_revenue	<i>field</i>	
use_fixed_weight	<i>flag</i>	
fixed_weight	<i>number</i>	0.0보다 큰 실수입니다.
variable_weight	<i>field</i>	
lift_percentile	<i>number</i>	0과 100 사이의 정수입니다.
enable_model_build_time_limit	<i>flag</i>	
model_build_time_limit	<i>number</i>	각 개별 모델을 작성하는 데 걸리는 시간을 제한하기 위해 분 단위로 설정하는 정수입니다.
enable_stop_after_time_limit	<i>flag</i>	

표 111. *autoclassifiernode* 특성 (계속).

autoclassifiernode 특성	값	특성 설명
stop_after_time_limit	<i>number</i>	자동 분류자 실행에 대한 전체 경과 시간을 제한하기 위해 시간 수로 설정되는 실수입니다.
enable_stop_after_valid_model_produced	<i>flag</i>	
use_costs	<i>flag</i>	
<algorithm>	<i>flag</i>	특정 알고리즘의 사용을 사용 또는 사용 안합니다.
<algorithm>.<property>	<i>string</i>	특정 알고리즘의 특성 값을 설정합니다. 자세한 정보는 『알고리즘 특성 설정』의 내용을 참조하십시오.

알고리즘 특성 설정

자동 분류자, 자동 수치, 자동 군집 노드의 경우 노드가 사용하는 특정 알고리즘에 대한 특성은 다음 일반 양식을 사용하여 설정할 수 있습니다.

```
autonode.setKeyedPropertyValue(<algorithm>, <property>, <value>)
```

예를 들어,

```
node.setKeyedPropertyValue("neuralnetwork", "method", "MultilayerPerceptron")
```

자동 분류자 노드의 알고리즘 이름은 cart, chaid, quest, c50, logreg, decisionlist, bayesnet, discriminant, svm 및 knn입니다.

자동 수치 노드의 알고리즘 이름은 cart, chaid, neuralnetwork, genlin, svm, regression, linear 및 knn입니다.

자동 군집 노드의 알고리즘 이름은 twostep, k-means, kohonen입니다.

특성 이름은 각 알고리즘 노드에 대해 문서화된 대로 표준입니다.

마침표나 기타 구두점을 포함하는 알고리즘 특성은 작은따옴표로 묶어야 합니다. 예:

```
node.setKeyedPropertyValue("logreg", "tolerance", "1.0E-5")
```

특성에 대해 다중 값을 지정할 수도 있습니다. 예:

```
node.setKeyedPropertyValue("decisionlist", "search_direction", ["Up", "Down"])
```

특정 알고리즘의 사용을 사용 또는 사용 안하려면 다음을 수행하십시오.

```
node.setPropertyValue("chaid", True)
```

참고: 특정 알고리즘 옵션이 자동 분류자 노드에서 사용할 수 없는 경우 또는 값의 범위가 아니라 단 일 값만 지정할 수 있을 때, 표준 방식으로 노드에 액세스할 때와 동일한 한계가 스크립팅에 적용됩니다.

autoclusternode 특성



자동 군집 노드는 유사한 특성을 가진 레코드 그룹을 식별하는 군집 모델을 추정하고 비교합니다. 이 노드는 다른 자동 모델링 노드와 동일한 방법으로 작동하여 단일 모델링 패스에서 다중 옵션 조합을 실험할 수 있습니다. 군집 모델의 유용성을 필터링하고 순위화하며 특정 필드의 중요성을 기반으로 측도를 제공하려고 시도하는 기본 측도를 사용하여 모델을 비교할 수 있습니다.

예제

```
node = stream.create("autocluster", "My node")
node.setPropertyValue("ranking_measure", "Silhouette")
node.setPropertyValue("ranking_dataset", "Training")
node.setPropertyValue("enable_silhouette_limit", True)
node.setPropertyValue("silhouette_limit", 5)
```

표 112. *autoclusternode* 특성

autoclusternode 특성	값	특성 설명
evaluation	<i>field</i>	참고: 자동 군집 노드만 해당됩니다. 중요도 값이 계산되는 필드를 식별합니다. 또는 군집이 이 필드의 값을 차별화하여 모델이 이 필드를 예측하는 정도를 식별하는 데 사용할 수 있습니다.
ranking_measure	Silhouette Num_clusters Size_smallest_cluster Size_largest_cluster Smallest_to_largest Importance	
ranking_dataset	Training Test	
summary_limit	<i>integer</i>	보고서에 나열할 모델 수입니다. 1과 100 사이의 정수를 지정하십시오.
enable_silhouette_limit	<i>flag</i>	
silhouette_limit	<i>integer</i>	0과 100 사이의 정수입니다.
enable_number_less_limit	<i>flag</i>	
number_less_limit	<i>number</i>	0.0과 1.0 사이의 실수입니다.
enable_number_greater_limit	<i>flag</i>	
number_greater_limit	<i>number</i>	0보다 큰 정수입니다.
enable_smallest_cluster_limit	<i>flag</i>	
smallest_cluster_units	Percentage Counts	
smallest_cluster_limit_percentage	<i>number</i>	
smallest_cluster_limit_count	<i>integer</i>	0보다 큰 정수입니다.
enable_largest_cluster_limit	<i>flag</i>	
largest_cluster_units	Percentage Counts	

표 112. *autoclusternode* 특성 (계속)

autoclusternode 특성	값	특성 설명
largest_cluster_limit_percentage	<i>number</i>	
largest_cluster_limit_count	<i>integer</i>	
enable_smallest_largest_limit	<i>flag</i>	
smallest_largest_limit	<i>number</i>	
enable_importance_limit	<i>flag</i>	
importance_limit_condition	Greater_than Less_than	
importance_limit_greater_than	<i>number</i>	0과 100 사이의 정수입니다.
importance_limit_less_than	<i>number</i>	0과 100 사이의 정수입니다.
<algorithm>	<i>flag</i>	특정 알고리즘의 사용을 사용 또는 사용 안합니다.
<algorithm>.<property>	<i>string</i>	특정 알고리즘의 특성 값을 설정합니다. 자세한 정보는 198 페이지의 『알고리즘 특성 설정』의 내용을 참조하십시오.

autonumericnode 특성



자동 수치 노드는 수많은 방법을 사용하여 연속적 수치 범위 결과의 모델을 추정하고 비교합니다. 이 노드는 자동 분류자 노드에서와 같은 방식으로 작동하므로 사용할 알고리즘을 선택하고 단일 모델링 전달에서 여러 옵션의 조합을 실험할 수 있습니다. 지원되는 알고리즘에는 신경망, C&R 트리, CHAID, 선형 회귀, 일반화 선형 회귀 및 지원 벡터 머신(SVM)이 있습니다. 모델은 상관관계, 상대 오차 또는 사용된 변수의 수를 기반으로 비교할 수 있습니다.

예제

```
node = stream.create("autonumeric", "My node")
node.setPropertyValue("ranking_measure", "Correlation")
node.setPropertyValue("ranking_dataset", "Training")
node.setPropertyValue("enable_correlation_limit", True)
node.setPropertyValue("correlation_limit", 0.8)
node.setPropertyValue("calculate_variable_importance", True)
node.setPropertyValue("neuralnetwork", True)
node.setPropertyValue("chaid", False)
```

표 113. *autonumericnode* 특성

autonumericnode 특성	값	특성 설명
custom_fields	<i>flag</i>	True인 경우 사용자 정의 필드 설정을 유형 노드 설정 대신 사용합니다.
target	<i>field</i>	자동 수치 노드는 하나의 목표 및 하나 이상의 입력 필드가 필요합니다. 가중치 및 빈도 필드도 지정할 수 있습니다. 자세한 정보는 191 페이지의 『공통 모델링 노드 특성』 주제를 참조하십시오.
inputs	[<i>field1</i> ... <i>field2</i>]	

표 113. *autonumericnode* 특성 (계속)

autonumericnode 특성	값	특성 설명
partition	<i>field</i>	
use_frequency	<i>flag</i>	
frequency_field	<i>field</i>	
use_weight	<i>flag</i>	
weight_field	<i>field</i>	
use_partitioned_data	<i>flag</i>	파티션 필드가 정의되는 경우, 훈련 데이터만 모델 작성에 사용합니다.
ranking_measure	Correlation NumberOfFields	
ranking_dataset	Test Training	
number_of_models	<i>integer</i>	모델 너깃에 포함할 모델 수입니다. 1과 100 사이의 정수를 지정하십시오.
calculate_variable_importance	<i>flag</i>	
enable_correlation_limit	<i>flag</i>	
correlation_limit	<i>integer</i>	
enable_number_of_fields_limit	<i>flag</i>	
number_of_fields_limit	<i>integer</i>	
enable_relative_error_limit	<i>flag</i>	
relative_error_limit	<i>integer</i>	
enable_model_build_time_limit	<i>flag</i>	
model_build_time_limit	<i>integer</i>	
enable_stop_after_time_limit	<i>flag</i>	
stop_after_time_limit	<i>integer</i>	
stop_if_valid_model	<i>flag</i>	
<algorithm>	<i>flag</i>	특정 알고리즘의 사용을 사용 또는 사용 안합니다.
<algorithm>.<property>	<i>string</i>	특정 알고리즘의 특성 값을 설정합니다. 자세한 정보는 198 페이지의 『알고리즘 특성 설정』의 내용을 참조하십시오.

bayesnetnode 특성



베이지안 네트워크 노드를 통해 관측 및 레코딩된 증거를 실세계 지식과 조합하여 발생 우도를 확립함으로써 확률 모델을 작성할 수 있습니다. 이 노드는 주로 분류에 사용하는 TAN(Tree Augmented Naïve Bayes) 및 Markov Blanket 네트워크에 초점을 맞춥니다.

예제

```

node = stream.create("bayesnet", "My node")
node.setPropertyValue("continue_training_existing_model", True)
node.setPropertyValue("structure_type", "MarkovBlanket")
node.setPropertyValue("use_feature_selection", True)
# Expert tab
node.setPropertyValue("mode", "Expert")
node.setPropertyValue("all_probabilities", True)
node.setPropertyValue("independence", "Pearson")

```

표 114. bayesnetnode 특성

bayesnetnode 특성	값	특성 설명
inputs	[field1 ... fieldN]	베이지안 신경망 모형은 단일 목표 필드와 하나 이상의 입력 필드를 사용합니다. 연속형 필드는 자동으로 구간화됩니다. 자세한 정보는 191 페이지의 『공통 모델링 노드 특성』 주제를 참조하십시오.
continue_training_existing_model	flag	
structure_type	TAN MarkovBlanket	베이지안 네트워크를 작성할 때 사용할 구조를 선택하십시오.
use_feature_selection	flag	
parameter_learning_method	Likelihood Bayes	상위의 값이 알려진 노드 사이에 조건부 확률 테이블을 추정하는 데 사용하는 방법을 지정합니다.
mode	Expert Simple	
missing_values	flag	
all_probabilities	flag	
independence	Likelihood Pearson	두 변수에 대해 쌍을 이루는 관측값이 서로 독립적인지 여부를 판별하는 데 사용하는 방법을 지정합니다.
significance_level	number	독립성 판별을 위한 절사 값을 지정합니다.
maximal_conditioning_set	number	독립성 검정에 사용할 조건부 변수의 최대 수를 설정합니다.
inputs_always_selected	[field1 ... fieldN]	베이지안 네트워크를 작성할 때 항상 사용할 데이터 세트의 필드를 지정합니다. 참고: 목표 필드는 항상 선택됩니다.
maximum_number_inputs	number	베이지안 네트워크 작성에 사용할 입력 필드의 최대 수를 지정합니다.
calculate_variable_importance	flag	
calculate_raw_propensities	flag	
calculate_adjusted_propensities	flag	
adjusted_propensity_partition	Test Validation	

buildr 특성



R 작성 노드를 사용하면 IBM SPSS Modeler에서 배포되는 모델 작성 및 모델 스코어링을 수행하기 위한 사용자 정의 R 스크립트를 입력할 수 있습니다.

예제

```
node = stream.create("buildr", "My node")
node.setPropertyValue("score_syntax", "")
result<-predict(modelerModel,newdata=modelerData)
modelerData<-cbind(modelerData,result)
var1<-c(fieldName="NaPrediction",fieldLabel="",fieldStorage="real",fieldMeasure="",
fieldFormat="",fieldRole="")
modelerDataModel<-data.frame(modelerDataModel,var1)""")
```

표 115. buildr 특성.

buildr 특성	값	특성 설명
build_syntax	string	모델 작성을 위한 R 스크립트 구문.
score_syntax	string	모델 스코어링의 R 스크립팅 구문
convert_flags	StringsAndDoubles LogicalValues	플래그 필드를 변환하는 옵션.
convert_datetime	flag	날짜 또는 날짜/시간 형식을 갖는 변수를 R 날짜/시간 형식으로 변환하는 옵션.
convert_datetime_class	POSIXct POSIX1t	날짜 또는 날짜/시간 형식을 갖는 변수를 변환할 형식을 지정하는 옵션.
convert_missing	flag	결측값을 R NA 값으로 변환하는 옵션.
output_html	flag	R 모델 너깃에서 탭에 그래프를 표시하는 옵션.
output_text	flag	R 모델 너깃에서 R 콘솔 텍스트 출력을 탭에 쓰는 옵션.

c50node 특성



C5.0 노드는 의사결정 트리 또는 규칙 세트를 작성합니다. 모델은 각 수준에서 최대 정보 이익을 제공하는 필드를 기반으로 샘플을 분할하여 작동합니다. 목표 필드는 범주형이어야 합니다. 세 개 이상의 부집단으로의 다중 분할이 허용됩니다.

예제

```
node = stream.create("c50", "My node")
# "Model" tab
node.setPropertyValue("use_model_name", False)
node.setPropertyValue("model_name", "C5_Drug")
node.setPropertyValue("use_partitioned_data", True)
```

```

node.setPropertyValue("output_type", "DecisionTree")
node.setPropertyValue("use_xval", True)
node.setPropertyValue("xval_num_folds", 3)
node.setPropertyValue("mode", "Expert")
node.setPropertyValue("favor", "Generality")
node.setPropertyValue("min_child_records", 3)
# "Costs" tab
node.setPropertyValue("use_costs", True)
node.setPropertyValue("costs", [["drugA", "drugX", 2]])

```

표 116. c50node 특성

c50node 특성	값	특성 설명
target	<i>field</i>	C50 모델은 단일 목표 필드와 하나 이상의 입력 필드를 사용합니다. 가중 필드도 지정할 수 있습니다. 자세한 정보는 191 페이지의 『공통 모델링 노드 특성』 주제를 참조하십시오.
output_type	DecisionTree RuleSet	
group_symbolics	<i>flag</i>	
use_boost	<i>flag</i>	
boost_num_trials	<i>number</i>	
use_xval	<i>flag</i>	
xval_num_folds	<i>number</i>	
mode	Simple Expert	
favor	Accuracy Generality	정확도 또는 범용성을 선호합니다.
expected_noise	<i>number</i>	
min_child_records	<i>number</i>	
pruning_severity	<i>number</i>	
use_costs	<i>flag</i>	
costs	<i>structured</i>	이것은 구조화 특성입니다.
use_winning	<i>flag</i>	
use_global_pruning	<i>flag</i>	기본적으로 On(True)입니다.
calculate_variable_importance	<i>flag</i>	
calculate_raw_propensities	<i>flag</i>	
calculate_adjusted_propensities	<i>flag</i>	
adjusted_propensity_partition	Test Validation	

carmanode 특성



CARMA 모델은 입력 또는 목표 필드를 지정하지 않아도 데이터에서 규칙 세트를 추출합니다. Apriori와 대조적으로 CARMA 노드는 단지 전향 지원이 아니라 규칙 지원(전향 및 후향 둘 다에 대한 지원)을 위한 작성 설정을 제공합니다. 이는 생성된 규칙을 보다 다양한 애플리케이션에 사용하여, 예를 들어 후향이 이번 휴가철에 홍보할 항목인 제품 또는 서비스 목록을 찾을 수 있음을 의미합니다.

예제

```
node = stream.create("carma", "My node")
# "Fields" tab
node.setPropertyValue("custom_fields", True)
node.setPropertyValue("use_transactional_data", True)
node.setPropertyValue("inputs", ["BP", "Cholesterol", "Drug"])
node.setPropertyValue("partition", "Test")
# "Model" tab
node.setPropertyValue("use_model_name", False)
node.setPropertyValue("model_name", "age_bp_drug")
node.setPropertyValue("use_partitioned_data", False)
node.setPropertyValue("min_supp", 10.0)
node.setPropertyValue("min_conf", 30.0)
node.setPropertyValue("max_size", 5)
# Expert Options
node.setPropertyValue("mode", "Expert")
node.setPropertyValue("use_pruning", True)
node.setPropertyValue("pruning_value", 300)
node.setPropertyValue("vary_support", True)
node.setPropertyValue("estimated_transactions", 30)
node.setPropertyValue("rules_without_antecedents", True)
```

표 117. carmanode 특성

carmanode 특성	값	특성 설명
inputs	<i>[field1 ... fieldn]</i>	CARMA 모델은 입력 필드의 목록을 사용하지만 대상은 사용하지 않습니다. 가중치 및 빈도 필드는 사용하지 않습니다. 자세한 정보는 191 페이지의 『공통 모델링 노드 특성』 주제를 참조하십시오.
id_field	<i>field</i>	모델 작성을 위한 ID 필드로 사용하는 필드입니다.
contiguous	<i>flag</i>	ID 필드의 ID가 연속적인지 여부를 지정하는데 사용합니다.
use_transactional_data	<i>flag</i>	
content_field	<i>field</i>	
min_supp	<i>number</i> (퍼센트)	전향 지원이 아니라 규칙 지원과 관련됩니다. 기본값은 20%입니다.
min_conf	<i>number</i> (퍼센트)	기본값은 20%입니다.
max_size	<i>number</i>	기본값은 10입니다.

표 117. carmanode 특성 (계속)

carmanode 특성	값	특성 설명
mode	Simple Expert	기본값은 Simple입니다.
exclude_multiple	flag	다중 후향값을 갖는 규칙을 제외합니다. 기본값은 False입니다.
use_pruning	flag	기본값은 False입니다.
pruning_value	number	기본값은 500입니다.
vary_support	flag	
estimated_transactions	integer	
rules_without_antecedents	flag	

cartnode 특성



분류 및 회귀(C&R) 트리 노드는 추가 관측값을 예측하거나 분류할 수 있게 하는 의사결정 트리를 생성합니다. 이 방법은 재귀적 파티셔닝을 사용하여 각 단계마다 불순도를 최소화하여 훈련 레코드를 세그먼트로 분할합니다. 여기서 트리의 노드는 노드의 케이스의 100%가 목표 필드의 특정 범주에 속하면 "순수"로 간주됩니다. 목표 및 입력 필드는 숫자 범위 또는 범주형(명목형, 순서형 또는 플래그)입니다. 모든 분할은 이분형입니다(오직 두 개의 부집단).

예제

```
node = stream.createAt("cart", "My node", 200, 100)
# "Fields" tab
node.setPropertyValue("custom_fields", True)
node.setPropertyValue("target", "Drug")
node.setPropertyValue("inputs", ["Age", "BP", "Cholesterol"])
# "Build Options" tab, "Objective" panel
node.setPropertyValue("model_output_type", "InteractiveBuilder")
node.setPropertyValue("use_tree_directives", True)
node.setPropertyValue("tree_directives", """Grow Node Index 0 Children 1 2
Grow Node Index 2 Children 3 4""")
# "Build Options" tab, "Basics" panel
node.setPropertyValue("prune_tree", False)
node.setPropertyValue("use_std_err_rule", True)
node.setPropertyValue("std_err_multiplier", 3.0)
node.setPropertyValue("max_surrogates", 7)
# "Build Options" tab, "Stopping Rules" panel
node.setPropertyValue("use_percentage", True)
node.setPropertyValue("min_parent_records_pc", 5)
node.setPropertyValue("min_child_records_pc", 3)
# "Build Options" tab, "Advanced" panel
node.setPropertyValue("min_impurity", 0.0003)
node.setPropertyValue("impurity_measure", "Twoing")
# "Model Options" tab
node.setPropertyValue("use_model_name", True)
node.setPropertyValue("model_name", "Cart_Drug")
```

표 118. cartnode 특성

cartnode 특성	값	특성 설명
target	<i>field</i>	C&R 트리 모델은 하나의 대상과 하나 이상의 입력 필드가 필요합니다. 빈도 필드도 지정할 수 있습니다. 자세한 정보는 191 페이지의 『공통 모델링 노드 특성』 주제를 참조하십시오.
continue_training_existing_model	<i>flag</i>	
objective	Standard Boosting Bagging psm	psm은 매우 큰 데이터 세트에 사용하며 서버 연결이 필요합니다.
model_output_type	Single InteractiveBuilder	
use_tree_directives	<i>flag</i>	
tree_directives	<i>string</i>	트리 성장을 위한 지시문을 지정하십시오. 줄 바꾸기 또는 인용부호 이스케이프를 피하기 위해 지시문을 삼중 인용부호로 묶을 수 있습니다. 지시문은 데이터 또는 모델링 옵션의 사소한 변화에 매우 민감할 수 있으며 다른 데이터 세트에 일반화하지 않을 수 있음을 참고하십시오.
use_max_depth	Default Custom	
max_depth	<i>integer</i>	0부터 1000까지의 최대 트리 깊이입니다. use_max_depth = Custom인 경우에만 사용됩니다.
prune_tree	<i>flag</i>	과적합을 방지하기 위해 트리를 가지치기합니다.
use_std_err	<i>flag</i>	(표준 오차에서) 위험의 최대 차이를 사용합니다.
std_err_multiplier	<i>number</i>	최대 차이입니다.
max_surrogates	<i>number</i>	최대 서로게이트입니다.
use_percentage	<i>flag</i>	
min_parent_records_pc	<i>number</i>	
min_child_records_pc	<i>number</i>	
min_parent_records_abs	<i>number</i>	
min_child_records_abs	<i>number</i>	
use_costs	<i>flag</i>	
costs	<i>structured</i>	구조화 특성입니다.
priors	Data Equal Custom	
custom_priors	<i>structured</i>	구조화 특성입니다.
adjust_priors	<i>flag</i>	

표 118. cartnode 특성 (계속)

cartnode 특성	값	특성 설명
trails	<i>number</i>	부스팅 또는 배깅을 위한 구성요소 모델 수입니다.
set_ensemble_method	Voting HighestProbability HighestMeanProbability	범주형 대상에 대한 기본 결합 규칙입니다.
range_ensemble_method	Mean Median	연속형 대상에 대한 기본 결합 규칙입니다.
large_boost	<i>flag</i>	매우 큰 데이터 세트에 부스팅을 적용합니다.
min_impurity	<i>number</i>	
impurity_measure	Gini Twoing Ordered	
train_pct	<i>number</i>	과적합 방지 세트입니다.
set_random_seed	<i>flag</i>	결과 복제 옵션입니다.
seed	<i>number</i>	
calculate_variable_importance	<i>flag</i>	
calculate_raw_propensities	<i>flag</i>	
calculate_adjusted_propensities	<i>flag</i>	
adjusted_propensity_partition	Test Validation	

chaidnode 특성



CHAID 노드는 최적 분할을 식별하기 위해 카이제곱 통계량을 사용하여 의사결정 트리를 생성합니다. C&R 트리 및 QUEST 노드와 달리, CHAID는 비이분형 트리를 생성할 수 있으며 이는 일부 분할에 둘 이상의 분기가 있음을 의미합니다. 목표 및 입력 필드는 숫자 범위(연속형) 또는 범주형입니다. Exhaustive CHAID는 가능한 모든 분할을 탐색하는 보다 철저한 작업을 수행하지만 계산하는 데 시간이 더 걸리는 변형 CHAID입니다.

예제

```

filenode = stream.createAt("variablefile", "My node", 100, 100)
filenode.setPropertyValue("full_filename", "$CLEO_DEMOS/DRUG1n")
node = stream.createAt("chaid", "My node", 200, 100)
stream.link(filenode, node)

node.setPropertyValue("custom_fields", True)
node.setPropertyValue("target", "Drug")
node.setPropertyValue("inputs", ["Age", "Na", "K", "Cholesterol", "BP"])
node.setPropertyValue("use_model_name", True)
node.setPropertyValue("model_name", "CHAID")
node.setPropertyValue("method", "Chaid")
node.setPropertyValue("model_output_type", "InteractiveBuilder")
node.setPropertyValue("use_tree_directives", True)
    
```

```

node.setPropertyValue("tree_directives", "Test")
node.setPropertyValue("split_alpha", 0.03)
node.setPropertyValue("merge_alpha", 0.04)
node.setPropertyValue("chi_square", "Pearson")
node.setPropertyValue("use_percentage", False)
node.setPropertyValue("min_parent_records_abs", 40)
node.setPropertyValue("min_child_records_abs", 30)
node.setPropertyValue("epsilon", 0.003)
node.setPropertyValue("max_iterations", 75)
node.setPropertyValue("split_merged_categories", True)
node.setPropertyValue("bonferroni_adjustment", True)

```

표 119. chaidnode 특성

chaidnode 특성	값	특성 설명
target	field	CHAID 모델은 하나의 대상과 하나 이상의 입력 필드가 필요합니다. 빈도 필드도 지정할 수 있습니다. 자세한 정보는 191 페이지의 『공통 모델링 노드 특성』 주제를 참조하십시오.
continue_training_existing_model	flag	
objective	Standard Boosting Bagging psm	psm은 매우 큰 데이터 세트에 사용하며 서버 연결이 필요합니다.
model_output_type	Single InteractiveBuilder	
use_tree_directives	flag	
tree_directives	string	
method	Chaid ExhaustiveChaid	
use_max_depth	Default Custom	
max_depth	integer	0부터 1000까지의 최대 트리 깊이입니다. use_max_depth = Custom인 경우에만 사용됩니다.
use_percentage	flag	
min_parent_records_pc	number	
min_child_records_pc	number	
min_parent_records_abs	number	
min_child_records_abs	number	
use_costs	flag	
costs	structured	구조화 특성입니다.
trails	number	부스팅 또는 배깅을 위한 구성요소 모델 수입니다.
set_ensemble_method	Voting HighestProbability HighestMeanProbability	범주형 대상에 대한 기본 결합 규칙입니다.

표 119. *chaidnode* 특성 (계속)

chaidnode 특성	값	특성 설명
range_ensemble_method	Mean Median	연속형 대상에 대한 기본 결합 규칙입니다.
large_boost	flag	매우 큰 데이터 세트에 부스팅을 적용합니다.
split_alpha	number	분할 유의수준입니다.
merge_alpha	number	병합 유의수준입니다.
bonferroni_adjustment	flag	Bonferroni 방법을 사용하여 유의성 값을 조정합니다.
split_merged_categories	flag	병합된 범주의 재분할을 허용합니다.
chi_square	Pearson LR	카이제곱 통계량을 계산하는 데 사용하는 방법 (Pearson 또는 우도비(Likelihood Ratio))입니다.
epsilon	number	셀 기대빈도의 최소 변화입니다.
max_iterations	number	수렴을 위한 최대 반복입니다.
set_random_seed	integer	
seed	number	
calculate_variable_importance	flag	
calculate_raw_propensities	flag	
calculate_adjusted_propensities	flag	
adjusted_propensity_partition	Test Validation	
maximum_number_of_models	integer	

coxregnode 특성



Cox 회귀 노드를 통해 중도절단된 레코드가 있는 데서 시간 대 이벤트 데이터에 대한 생존 모델을 작성할 수 있습니다. 이 모델은 주어진 입력 변수 값에 대해 주어진 시간(t)에 흥미있는 이벤트가 발생한 확률을 예측하는 생존함수를 생성합니다.

예제

```
node = stream.create("coxreg", "My node")
node.setPropertyValue("survival_time", "tenure")
node.setPropertyValue("method", "BackwardsStepwise")
# Expert tab
node.setPropertyValue("mode", "Expert")
node.setPropertyValue("removal_criterion", "Conditional")
node.setPropertyValue("survival", True)
```

표 120. *coxregnode* 특성

coxregnode 특성	값	특성 설명
survival_time	field	Cox 회귀 모형은 생존 시간을 포함하는 단일 필드가 필요합니다.

표 120. *coxregnode* 특성 (계속)

coxregnode 특성	값	특성 설명
target	<i>field</i>	Cox 회귀 모형은 단일 목표 필드와 하나 이상의 입력 필드가 필요합니다. 자세한 정보는 191 페이지의 『공통 모델링 노드 특성』 주제를 참조하십시오.
method	Enter Stepwise BackwardsStepwise	
groups	<i>field</i>	
model_type	MainEffects Custom	
custom_terms	["BP*Sex" "BP*Age"]	
mode	Expert Simple	
max_iterations	<i>number</i>	
p_converge	1.0E-4 1.0E-5 1.0E-6 1.0E-7 1.0E-8 0	
p_converge	1.0E-4 1.0E-5 1.0E-6 1.0E-7 1.0E-8 0	
l_converge	1.0E-1 1.0E-2 1.0E-3 1.0E-4 1.0E-5 0	
removal_criterion	LR Wald Conditional	
probability_entry	<i>number</i>	
probability_removal	<i>number</i>	
output_display	EachStep LastStep	
ci_enable	<i>flag</i>	
ci_value	90 95 99	

표 120. *coxregnode* 특성 (계속)

coxregnode 특성	값	특성 설명
correlation	<i>flag</i>	
display_baseline	<i>flag</i>	
survival	<i>flag</i>	
hazard	<i>flag</i>	
log_minus_log	<i>flag</i>	
one_minus_survival	<i>flag</i>	
separate_line	<i>field</i>	
value	<i>number</i> 또는 <i>string</i>	필드에 대한 값이 지정되지 않은 경우, 해당 필드에 대해 기본 옵션인 "Mean"을 사용합니다.

decisionlistnode 특성



의사결정 목록 노드는 전체 채우기에 상대적인 주어진 이분형 결과의 상위 또는 하위 우도를 표시하는 부집단 또는 세그먼트를 식별합니다. 예를 들어, 캠페인을 이탈할 가능성이 없거나 우호적으로 응답할 가능성이 가장 많은 고객을 찾고 있습니다. 자체 사용자 정의 세그먼트를 추가하고 대체 모델을 나란히 미리보기하여 결과를 비교함으로써 비즈니스 지식을 모델에 통합할 수 있습니다. 의사결정 목록 모델은 각 규칙에 조건과 결과가 있는 규칙 목록으로 구성됩니다. 규칙은 순서대로 적용되며 매치하는 첫 번째 규칙이 결과를 결정합니다.

예제

```
node = stream.create("decisionlist", "My node")
node.setPropertyValue("search_direction", "Down")
node.setPropertyValue("target_value", 1)
node.setPropertyValue("max_rules", 4)
node.setPropertyValue("min_group_size_pct", 15)
```

표 121. *decisionlistnode* 특성

decisionlistnode 특성	값	특성 설명
target	<i>field</i>	의사결정 목록 모델은 단일 대상과 하나 이상의 입력 필드를 사용합니다. 빈도 필드도 지정할 수 있습니다. 자세한 정보는 191 페이지의 『공동 모델링 노드 특성』 주제를 참조하십시오.
model_output_type	Model InteractiveBuilder	
search_direction	Up Down	세그먼트 찾기와 관련됩니다. Up은 높은 확률과 증가이고 Down은 낮은 확률과 증가입니다.
target_value	<i>string</i>	지정되지 않으면 플래그에 대해 true 값을 가정합니다.
max_rules	<i>integer</i>	나머지를 제외한 최대 세그먼트 수입니다.
min_group_size	<i>integer</i>	최소 세그먼트 크기입니다.
min_group_size_pct	<i>number</i>	퍼센트로서의 최소 세그먼트 크기입니다.

표 121. *decisionlistnode* 특성 (계속)

decisionlistnode 특성	값	특성 설명
confidence_level	<i>number</i>	세그먼트 정의에 추가하는 것을 의미있게 만들기 위해 응답의 우도를 개선(리프트 제공)해야 하는 최소 임계값입니다.
max_segments_per_rule	<i>integer</i>	
mode	Simple Expert	
bin_method	EqualWidth EqualCount	
bin_count	<i>number</i>	
max_models_per_cycle	<i>integer</i>	목록의 검색 너무입니다.
max_rules_per_cycle	<i>integer</i>	세그먼트 규칙의 검색 너무입니다.
segment_growth	<i>number</i>	
include_missing	<i>flag</i>	
final_results_only	<i>flag</i>	
reuse_fields	<i>flag</i>	속성(규칙에 나타나는 입력 필드)을 다시 사용하도록 허용합니다.
max_alternatives	<i>integer</i>	
calculate_raw_propensities	<i>flag</i>	
calculate_adjusted_propensities	<i>flag</i>	
adjusted_propensity_partition	Test Validation	

discriminantnode 특성



판별 분석은 로지스틱 회귀분석보다 엄격한 가정을 하지만 해당 가정이 충족되면 로지스틱 회귀 분석의 귀중한 대안 또는 보조물이 될 수 있습니다.

예제

```
node = stream.create("discriminant", "My node")
node.setPropertyValue("target", "custcat")
node.setPropertyValue("use_partitioned_data", False)
node.setPropertyValue("method", "Stepwise")
```

표 122. *discriminantnode* 특성

discriminantnode 특성	값	특성 설명
target	<i>field</i>	판별 모델은 단일 목표 필드와 하나 이상의 입력 필드가 필요합니다. 가중치 및 빈도 필드는 사용하지 않습니다. 자세한 정보는 191 페이지의 『공통 모델링 노드 특성』 주제를 참조하십시오.

표 122. discriminantnode 특성 (계속)

discriminantnode 특성	값	특성 설명
method	Enter Stepwise	
mode	Simple Expert	
prior_probabilities	AllEqual ComputeFromSizes	
covariance_matrix	WithinGroups SeparateGroups	
means	flag	고급 옵션 대화 상자의 통계량 옵션입니다.
univariate_anovas	flag	
box_m	flag	
within_group_covariance	flag	
within_groups_correlation	flag	
separate_groups_covariance	flag	
total_covariance	flag	
fishers	flag	
unstandardized	flag	
casewise_results	flag	고급 옵션 대화 상자의 분류 옵션입니다.
limit_to_first	number	기본값은 10입니다.
summary_table	flag	
leave_one_classification	flag	
combined_groups	flag	
separate_groups_covariance	flag	행렬 옵션 개별-집단 공분산 행렬.
territorial_map	flag	
combined_groups	flag	도표 옵션 결합-집단.
separate_groups	flag	도표 옵션 개별-집단.
summary_of_steps	flag	
F_pairwise	flag	
stepwise_method	WilksLambda UnexplainedVariance MahalanobisDistance SmallestF RaosV	
V_to_enter	number	
criteria	UseValue UseProbability	
F_value_entry	number	기본값은 3.84입니다.
F_value_removal	number	기본값은 2.71입니다.
probability_entry	number	기본값은 0.05입니다.
probability_removal	number	기본값은 0.10입니다.
calculate_variable_importance	flag	

표 122. *discriminantnode* 특성 (계속)

discriminantnode 특성	값	특성 설명
calculate_raw_propensities	<i>flag</i>	
calculate_adjusted_propensities	<i>flag</i>	
adjusted_propensity_partition	Test Validation	

extensionmodelnode 특성



확장 모델 노드를 사용하면 R 또는 Python for spark 스크립트를 실행하여 결과를 작성하고 스코어링할 수 있습니다.

Python for Spark 예제

```

#### script example for Python for Spark
import modeler.api
stream = modeler.script.stream()
node = stream.create("extension_build", "extension_build")
node.setPropertyValue("syntax_type", "Python")

build_script = """
import json
import spss.pyspark.runtime
from pyspark.mllib.regression import LabeledPoint
from pyspark.mllib.linalg import DenseVector
from pyspark.mllib.tree import DecisionTree

cxt = spss.pyspark.runtime.getContext()
df = cxt.getSparkInputData()
schema = df.dtypes[:]

target = "Drug"
predictors = ["Age","BP","Sex","Cholesterol","Na","K"]

def metaMap(row,schema):
    col = 0
    meta = []
    for (cname, ctype) in schema:
        if ctype == 'string':
            meta.append(set([row[col]]))
        else:
            meta.append((row[col],row[col]))
        col += 1
    return meta

def metaReduce(meta1,meta2,schema):
    col = 0
    meta = []

```

```

for (cname, ctype) in schema:
    if ctype == 'string':
        meta.append(meta1[col].union(meta2[col]))
    else:
        meta.append((min(meta1[col][0],meta2[col][0]),max(meta1[col][1],meta2[col][1])))
    col += 1
return meta

metadata = df.rdd.map(lambda row: metaMap(row,schema)).reduce(lambda x,y:metaReduce(x,y,schema))

def setToList(v):
    if isinstance(v,set):
        return list(v)
    return v

metadata = map(lambda x: setToList(x), metadata)
print metadata

lookup = {}
for i in range(0,len(schema)):
    lookup[schema[i][0]] = i

def row2LabeledPoint(dm,lookup,target,predictors,row):
    target_index = lookup[target]
    tval = dm[target_index].index(row[target_index])
    pvals = []
    for predictor in predictors:
        predictor_index = lookup[predictor]
        if isinstance(dm[predictor_index],list):
            pval = dm[predictor_index].index(row[predictor_index])
        else:
            pval = row[predictor_index]
        pvals.append(pval)
    return LabeledPoint(tval,DenseVector(pvals))

# count number of target classes
predictorClassCount = len(metadata[lookup[target]])

# define function to extract categorical predictor information from datamodel
def getCategoricalFeatureInfo(dm,lookup,predictors):
    info = {}
    for i in range(0,len(predictors)):
        predictor = predictors[i]
        predictor_index = lookup[predictor]
        if isinstance(dm[predictor_index],list):
            info[i] = len(dm[predictor_index])
    return info

# convert dataframe to an RDD containing LabeledPoint
lps = df.rdd.map(lambda row: row2LabeledPoint(metadata,lookup,target,predictors,row))

treeModel = DecisionTree.trainClassifier(
    lps,
    numClasses=predictorClassCount,
    categoricalFeaturesInfo=getCategoricalFeatureInfo(metadata, lookup, predictors),
    impurity='gini',
    maxDepth=5,
    maxBins=100)

```

```

_outputPath = cxt.createTemporaryFolder()
treeModel.save(cxt.getSparkContext(), _outputPath)
cxt.setModelContentFromPath("TreeModel", _outputPath)
cxt.setModelContentFromString("model.dm", json.dumps(metadata), mimeType="application/json")\
    .setModelContentFromString("model.structure", treeModel.toDebugString())

"""

node.setPropertyValue("python_build_syntax", build_script)

```

R 예제

```

#### script example for R
node.setPropertyValue("syntax_type", "R")
node.setPropertyValue("r_build_syntax", """modelerModel
<- lm(modelerData$Na~modelerData$K,modelerData)
modelerDataModel
modelerModel
""")

```

표 123. *extensionmodelnode* 특성

extensionmodelnode 특성	값	특성 설명
syntax_type	R Python	실행할 스크립트, R 또는 Python을 지정하십시오(R이 기본값).
r_build_syntax	string	모델 작성을 위한 R 스크립팅 구문입니다.
r_score_syntax	string	모델 스코어링을 위한 R 스크립팅 구문입니다.
python_build_syntax	string	모델 작성을 위한 Python 스크립팅 구문입니다.
python_score_syntax	string	모델 스코어링을 위한 Python 스크립팅 구문입니다.
convert_flags	StringsAndDoubles LogicalValues	플래그 필드를 변환하는 옵션.
convert_missing	flag	결측값을 R NA 값으로 변환하는 옵션.
convert_datetime	flag	날짜 또는 날짜/시간 형식을 갖는 변수를 R 날짜/시간 형식으로 변환하는 옵션.
convert_datetime_class	POSIXct POSIX1t	날짜 또는 날짜/시간 형식을 갖는 변수를 변환할 형식을 지정하는 옵션.
output_html	flag	R 모델 너깃에서 탭에 그래프를 표시하는 옵션.
output_text	flag	R 모델 너깃에서 R 콘솔 텍스트 출력을 탭에 쓰는 옵션.

factornode 특성



PCA/요인 노드에서는 강력한 데이터 축소 기법을 제공하여 데이터의 복잡도를 줄입니다. 비선형 주성분분석(PCA)은 구성요소가 서로 직각(수직)인 전체 필드 세트에서 변동을 캡처하는 입력 필드의 선형 조합을 찾습니다. 요인 분석은 관측된 필드 세트 내에서 상관관계 패턴을 설명하는 기본 요인을 식별하려고 시도합니다. 두 접근 방식 모두 목표는 원래 필드 세트의 정보를 효과적으로 요약하는 적은 수의 파생 필드를 찾는 것입니다.

예제

```
node = stream.create("factor", "My node")
# "Fields" tab
node.setPropertyValue("custom_fields", True)
node.setPropertyValue("inputs", ["BP", "Na", "K"])
node.setPropertyValue("partition", "Test")
# "Model" tab
node.setPropertyValue("use_model_name", True)
node.setPropertyValue("model_name", "Factor_Age")
node.setPropertyValue("use_partitioned_data", False)
node.setPropertyValue("method", "GLS")
# Expert options
node.setPropertyValue("mode", "Expert")
node.setPropertyValue("complete_records", True)
node.setPropertyValue("matrix", "Covariance")
node.setPropertyValue("max_iterations", 30)
node.setPropertyValue("extract_factors", "ByFactors")
node.setPropertyValue("min_eigenvalue", 3.0)
node.setPropertyValue("max_factor", 7)
node.setPropertyValue("sort_values", True)
node.setPropertyValue("hide_values", True)
node.setPropertyValue("hide_below", 0.7)
# "Rotation" section
node.setPropertyValue("rotation", "DirectOblimin")
node.setPropertyValue("delta", 0.3)
node.setPropertyValue("kappa", 7.0)
```

표 124. *factornode* 특성

factornode 특성	값	특성 설명
inputs	<i>[field1 ... fieldN]</i>	PCA/요인 모델은 입력 필드의 목록을 사용하지만 대상은 없습니다. 가중치 및 빈도 필드는 사용하지 않습니다. 자세한 정보는 191 페이지의 『공통 모델링 노드 특성』 주제를 참조하십시오.
method	PC ULS GLS ML PAF Alpha Image	
mode	Simple Expert	
max_iterations	<i>number</i>	
complete_records	<i>flag</i>	
matrix	Correlation Covariance	
extract_factors	ByEigenvalues ByFactors	

표 124. *factornode* 특성 (계속)

factornode 특성	값	특성 설명
min_eigenvalue	<i>number</i>	
max_factor	<i>number</i>	
rotation	None Varimax DirectOblimin Equamax Quartimax Promax	
delta	<i>number</i>	DirectOblimin을 회전 데이터 유형으로 선택하는 경우 delta의 값을 지정할 수 있습니다. 값을 지정하지 않으면 delta의 기본값을 사용합니다.
kappa	<i>number</i>	Promax를 회전 데이터 유형으로 선택하는 경우 kappa의 값을 지정할 수 있습니다. 값을 지정하지 않으면 kappa의 기본값을 사용합니다.
sort_values	<i>flag</i>	
hide_values	<i>flag</i>	
hide_below	<i>number</i>	

featureselectionnode 특성



필드선택 노드는 기준(예: 결측값의 퍼센트) 세트를 기반으로 제거용 입력 필드를 차단합니다. 그런 다음 지정된 대상에 상대적인 남아 있는 입력의 중요도에 대해 순위를 매깁니다. 예를 들어, 수백 개의 잠재 입력이 있는 데이터 세트가 있다면 환자 결과 모델링 시 어느 것이 가장 유용합니까?

예제

```
node = stream.create("featureselection", "My node")
node.setPropertyValue("screen_single_category", True)
node.setPropertyValue("max_single_category", 95)
node.setPropertyValue("screen_missing_values", True)
node.setPropertyValue("max_missing_values", 80)
node.setPropertyValue("criteria", "Likelihood")
node.setPropertyValue("unimportant_below", 0.8)
node.setPropertyValue("important_above", 0.9)
node.setPropertyValue("important_label", "Check Me Out!")
node.setPropertyValue("selection_mode", "TopN")
node.setPropertyValue("top_n", 15)
```

필드선택 모델을 작성 및 적용하는 보다 상세한 예에 대해서는 [in](#)을 참조하십시오.

표 125. *featureselectionmode* 특성

featureselectionmode 특성	값	특성 설명
target	<i>field</i>	필드선택 모델은 지정된 대상에 상대적으로 예측자 순위를 매깁니다. 가중치 및 빈도 필드는 사용하지 않습니다. 자세한 정보는 191 페이지의 『공통 모델링 노드 특성』 주제를 참조하십시오.
screen_single_category	<i>flag</i>	True인 경우, 레코드의 총 수에 상대적으로 동일한 범주에 들어가는 레코드가 너무 많은 필드를 차단합니다.
max_single_category	<i>number</i>	screen_single_category가 True일 때 사용하는 임계값을 지정합니다.
screen_missing_values	<i>flag</i>	True인 경우, 레코드의 총 수의 퍼센트로 표현되는 결측값이 너무 많은 필드를 차단합니다.
max_missing_values	<i>number</i>	
screen_num_categories	<i>flag</i>	True인 경우, 레코드의 총 수에 상대적으로 너무 많은 범주를 갖는 필드를 차단합니다.
max_num_categories	<i>number</i>	
screen_std_dev	<i>flag</i>	True인 경우, 지정된 최소보다 작거나 같은 표준 편차를 갖는 필드를 차단합니다.
min_std_dev	<i>number</i>	
screen_coeff_of_var	<i>flag</i>	True인 경우, 지정된 최소보다 작거나 같은 분산 계수를 갖는 필드를 차단합니다.
min_coeff_of_var	<i>number</i>	
criteria	Pearson Likelihood CramersV Lambda	범주형 예측변수를 범주형 대상에 대해 순위화할 때, 중요도 값이 기반이 되는 측도를 지정합니다.
unimportant_below	<i>number</i>	중요, 보통 또는 중요하지 않음으로 변수 순위를 지정하는 데 사용하는 임계값 p 값을 지정합니다. 0.0부터 1.0까지의 값을 승인합니다.
important_above	<i>number</i>	0.0부터 1.0까지의 값을 승인합니다.
unimportant_label	<i>string</i>	중요하지 않은 순위에 대한 레이블을 지정합니다.
marginal_label	<i>string</i>	
important_label	<i>string</i>	
selection_mode	ImportanceLevel ImportanceValue TopN	
select_important	<i>flag</i>	selection_mode가 ImportanceLevel로 설정될 때, 중요 필드 선택 여부를 지정합니다.
select_marginal	<i>flag</i>	selection_mode가 ImportanceLevel로 설정될 때, 보통 필드 선택 여부를 지정합니다.

표 125. *featureselectionnode* 특성 (계속)

featureselectionnode 특성	값	특성 설명
select_unimportant	<i>flag</i>	selection_mode가 ImportanceLevel로 설정될 때, 중요하지 않은 필드 선택 여부를 지정합니다.
importance_value	<i>number</i>	selection_mode가 ImportanceValue로 설정될 때, 사용할 절사 값을 지정합니다. 0부터 100까지의 값을 승인합니다.
top_n	<i>integer</i>	selection_mode가 TopN으로 설정될 때, 사용할 절사 값을 지정합니다. 0에서부터 1000까지의 값을 승인합니다.

genlinnode 특성



일반화 선형 모델은 종속변수가 요인과 선형으로 관련되고 지정된 연결함수를 통해 공변되도록 일반 선형 모델을 확장합니다. 더욱이 모델을 사용하면 종속변수가 비정규 분포를 가질 수 있습니다. 선형 회귀, 로지스틱 회귀분석, 카운트 데이터에 대한 로그선형 모델, 간격 중도절단 생존 모델을 포함하여 상당수 통계 모델의 기능을 포함합니다.

예제

```
node = stream.create("genlin", "My node")
node.setPropertyValue("model_type", "MainAndAllTwoWayEffects")
node.setPropertyValue("offset_type", "Variable")
node.setPropertyValue("offset_field", "Claimant")
```

표 126. *genlinnode* 특성

genlinnode 특성	값	특성 설명
target	<i>field</i>	일반화 선형 모델은 명목 또는 플래그 필드여야 하는 단일 목표 필드와 하나 이상의 입력 필드가 필요합니다. 가장 필드도 지정할 수 있습니다. 자세한 정보는 191 페이지의 『공통 모델링 노드 특성』 주제를 참조하십시오.
use_weight	<i>flag</i>	
weight_field	<i>field</i>	필드 유형은 단지 연속형입니다.
target_represents_trials	<i>flag</i>	
trials_type	Variable FixedValue	
trials_field	<i>field</i>	필드 유형은 연속형, 플래그 또는 순서형입니다.
trials_number	<i>number</i>	기본값은 10입니다.
model_type	MainEffects MainAndAllTwoWayEffects	
offset_type	Variable FixedValue	
offset_field	<i>field</i>	필드 유형은 단지 연속형입니다.

표 126. *genlinnode* 특성 (계속)

genlinnode 특성	값	특성 설명
offset_value	<i>number</i>	실수여야 합니다.
base_category	Last First	
include_intercept	<i>flag</i>	
mode	Simple Expert	
distribution	BINOMIAL GAMMA IGAUSS NEGBIN NORMAL POISSON TWEEDIE MULTINOMIAL	IGAUSS: 역가우스. NEGBIN: 음이항.
negbin_para_type	Specify Estimate	
negbin_parameter	<i>number</i>	기본값은 1입니다. 음이 아닌 실수를 포함해야 합니다.
tweedie_parameter	<i>number</i>	
link_function	IDENTITY CLOGLOG LOG LOGC LOGIT NEGBIN NLOGLOG ODDSPower PROBIT POWER CUMCAUCHIT CUMCLOGLOG CUMLOGIT CUMNLOGLOG CUMPROBIT	CLOGLOG: 보 로그-로그. LOGC: 로그 보. NEGBIN: 음이항. NLOGLOG: 음 로그-로그. CUMCAUCHIT: 누적 Cauchit. CUMCLOGLOG: 누적 보 로그-로그. CUMLOGIT: 누적 로짓. CUMNLOGLOG: 누적 음 로그-로그. CUMPROBIT: 누적 프로빗.
power	<i>number</i>	값은 실수로 0이 아닌 숫자여야 합니다.
method	Hybrid Fisher NewtonRaphson	
max_fisher_iterations	<i>number</i>	기본값은 1입니다. 양의 정수만 허용됩니다.

표 126. *genlnode* 특성 (계속)

genlnode 특성	값	특성 설명
scale_method	MaxLikelihoodEstimate Deviance PearsonChiSquare FixedValue	
scale_value	number	기본값은 1입니다. 0보다 커야 합니다.
covariance_matrix	ModelEstimator RobustEstimator	
max_iterations	number	기본값은 100입니다. 음이 아닌 정수만 가능합니다.
max_step_halving	number	기본값은 5입니다. 양의 정수만 가능합니다.
check_separation	flag	
start_iteration	number	기본값은 20입니다. 양의 정수만 허용됩니다.
estimates_change	flag	
estimates_change_min	number	기본값은 1E-006입니다. 양수만 허용됩니다.
estimates_change_type	Absolute Relative	
loglikelihood_change	flag	
loglikelihood_change_min	number	양수만 허용됩니다.
loglikelihood_change_type	Absolute Relative	
hessian_convergence	flag	
hessian_convergence_min	number	양수만 허용됩니다.
hessian_convergence_type	Absolute Relative	
case_summary	flag	
contrast_matrices	flag	
descriptive_statistics	flag	
estimable_functions	flag	
model_info	flag	
iteration_history	flag	
goodness_of_fit	flag	
print_interval	number	기본값은 1입니다. 양의 정수여야 합니다.
model_summary	flag	
lagrange_multiplier	flag	
parameter_estimates	flag	
include_exponential	flag	
covariance_estimates	flag	
correlation_estimates	flag	
analysis_type	TypeI TypeIII TypeIAndTypeIII	

표 126. *genlmmnode* 특성 (계속)

genlmmnode 특성	값	특성 설명
statistics	Wald LR	
citype	Wald Profile	
tolerancelevel	<i>number</i>	기본값은 0.0001입니다.
confidence_interval	<i>number</i>	기본값은 95입니다.
loglikelihood_function	Full Kernel	
singularity_tolerance	1E-007 1E-008 1E-009 1E-010 1E-011 1E-012	
value_order	Ascending Descending DataOrder	
calculate_variable_importance	<i>flag</i>	
calculate_raw_propensities	<i>flag</i>	
calculate_adjusted_propensities	<i>flag</i>	
adjusted_propensity_partition	Test Validation	

glmmnode 특성



일반화 선형 혼합 모델(GLMM)은 목표가 비정규 분포를 갖고 지정된 연결함수를 통해 요인 및 공변량과 선형으로 관련되어 관측값을 상관시킬 수 있도록 일반 선형 모델을 확장합니다. 일반화 선형 혼합 모델은 단순 선형 회귀에서 비정규 장기적인 데이터에 대한 복합 다중 수준 모델에 이르기까지 다양한 모델을 포함합니다.

표 127. *glmmnode* 특성.

glmmnode 특성	값	특성 설명
residual_subject_spec	<i>structured</i>	데이터 세트 내에서 개체를 고유하게 정의하는 지정된 범주형 필드의 값의 조합입니다.
repeated_measures	<i>structured</i>	반복되는 관측값을 식별하는 데 사용하는 필드입니다.
residual_group_spec	[<i>field1 ... fieldN</i>]	반복 효과 공분산 모수의 독립적 세트를 정의하는 필드입니다.

표 127. *glmmnode* 특성 (계속).

glmmnode 특성	값	특성 설명
residual_covariance_type	Diagonal AR1 ARMA11 COMPOUND_SYMMETRY IDENTITY TOEPLITZ UNSTRUCTURED VARIANCE_COMPONENTS	잔차에 대한 공분산 구조를 지정합니다.
custom_target	<i>flag</i>	업스트림 노드에서 정의된 대상(false) 또는 target_field에 의해 지정되는 사용자 정의 대상(true)을 사용할지 여부를 표시합니다.
target_field	<i>field</i>	custom_target이 true인 경우 대상으로 사용할 필드입니다.
use_trials	<i>flag</i>	시행 수를 지정하는 추가 필드 또는 값을 대상 반응이 시행 세트에서 발생하는 이벤트 수일 때 사용할지 여부를 표시합니다. 기본값은 false입니다.
use_field_or_value	Field Value	필드(기본값) 또는 값이 시행 수를 지정하는 데 사용하는지 여부를 표시합니다.
trials_field	<i>field</i>	시행 수를 지정하는 데 사용할 필드입니다.
trials_value	<i>integer</i>	시행 수를 지정하는 데 사용할 값입니다. 지정되는 경우 최소값은 1입니다.
use_custom_target_reference	<i>flag</i>	사용자 정의 참조 범주가 범주형 대상에 사용할지 여부를 표시합니다. 기본값은 false입니다.
target_reference_value	<i>string</i>	use_custom_target_reference가 true인 경우 사용할 참조 범주입니다.
dist_link_combination	Nominal Logit GammaLog BinomialLogit PoissonLog BinomialProbit NegbinLog BinomialLogC Custom	대상에 대한 값의 분포를 위한 공통 모델입니다. target_distribution에 의해 제공되는 목록에서 분포를 지정하려면 Custom을 선택하십시오.
target_distribution	Normal Binomial Multinomial Gamma Inverse NegativeBinomial Poisson	dist_link_combination은 Custom일 때 대상에 대한 값의 분포입니다.

표 127. *glmmnnode* 특성 (계속).

glmmnnode 특성	값	특성 설명
link_function_type	Identity LogC Log CLOGLOG Logit NLOGLOG PROBIT POWER CAUCHIT	대상 값을 예측자와 관련시키는 연결함수입니다. target_distribution이 Binomial인 경우 나열된 연결함수 중 하나를 사용할 수 있습니다. target_distribution이 Multinomial인 경우 CLOGLOG, CAUCHIT, LOGIT, NLOGLOG 또는 PROBIT를 사용할 수 있습니다. target_distribution이 Binomial 또는 Multinomial인 경우 IDENTITY, LOG 또는 POWER를 사용할 수 있습니다.
link_function_param	number	사용할 연결함수 모수값입니다. normal_link_function 또는 link_function_type이 POWER인 경우에만 적용 가능합니다.
use_predefined_inputs	flag	고정 효과 필드가 입력 필드로서 정의된 업스트림(true) 또는 fixed_effects_list의 필드(false)인지 여부를 표시합니다. 기본값은 false입니다.
fixed_effects_list	structured	use_predefined_inputs가 false인 경우, 고정 효과 필드로 사용할 입력 필드를 지정합니다.
use_intercept	flag	true(기본값)인 경우 모델의 절편을 포함합니다.
random_effects_list	structured	변량효과로 지정할 필드의 목록입니다.
regression_weight_field	field	분석 가중값으로 사용할 필드입니다.
use_offset	None offset_value offset_field	오프셋이 지정되는 방법을 표시합니다. 값 None은 오프셋을 사용하지 않음을 의미합니다.
offset_value	number	use_offset이 offset_value로 설정되는 경우 오프셋에 사용할 값입니다.
offset_field	field	use_offset이 offset_field로 설정된 경우 오프셋 값에 사용할 필드입니다.
target_category_order	Ascending Descending Data	범주형 대상에 대한 정렬 순서입니다. 값 Data는 데이터에서 발견되는 정렬 순서를 사용 중임을 지정합니다. 기본값은 Ascending입니다.
inputs_category_order	Ascending Descending Data	범주형 예측변수에 대한 정렬 순서입니다. 값 Data는 데이터에서 발견되는 정렬 순서를 사용 중임을 지정합니다. 기본값은 Ascending입니다.
max_iterations	integer	알고리즘이 수행할 최대 반복 수입니다. 음이 아닌 정수이며, 기본값은 100입니다.
confidence_level	integer	모델 계수의 구간 추정값을 계산하는 데 사용하는 신뢰수준입니다. 음이 아닌 정수이며, 최대값은 100입니다. 기본값은 95입니다.

표 127. *glmmnode* 특성 (계속).

glmmnode 특성	값	특성 설명
degrees_of_freedom_method	Fixed Varied	유의성 검정을 위해 자유도가 계산되는 방법을 지정합니다.
test_fixed_effects_coefficients	Model Robust	모수 추정값 공분산 교차표 계산 방법입니다.
use_p_converge	<i>flag</i>	모수 수렴에 대한 옵션입니다.
p_converge	<i>number</i>	공란 또는 임의의 양수값입니다.
p_converge_type	Absolute Relative	
use_l_converge	<i>flag</i>	로그-우도 수렴에 대한 옵션입니다.
l_converge	<i>number</i>	공란 또는 임의의 양수값입니다.
l_converge_type	Absolute Relative	
use_h_converge	<i>flag</i>	Hessian 수렴에 대한 옵션입니다.
h_converge	<i>number</i>	공란 또는 임의의 양수값입니다.
h_converge_type	Absolute Relative	
max_fisher_steps	<i>integer</i>	
singularity_tolerance	<i>number</i>	
use_model_name	<i>flag</i>	모델의 사용자 정의 이름을 지정(true)하거나 시스템 생성 이름을 사용(false)하는지 여부를 표시합니다. 기본값은 false입니다.
model_name	<i>string</i>	use_model_name이 true인 경우 사용할 모델 이름을 지정합니다.
confidence	onProbability onIncrease	스코어링 신뢰도 계산을 위한 기초입니다. 최고 예측 확률 또는 최고 및 두 번째 최고 예측 확률 사이의 차이입니다.
score_category_probabilities	<i>flag</i>	true인 경우, 범주형 대상에 대한 예측 확률을 생성합니다. 기본값은 false입니다.
max_categories	<i>integer</i>	score_category_probabilities가 true인 경우, 저장할 최대 범주 수를 지정합니다.
score_propensity	<i>flag</i>	true인 경우, 필드에 대한 "true" 결과의 우도를 표시하는 플래그 목표 필드의 성향 스코어를 생성합니다.
emeans	<i>structure</i>	고정 효과 목록의 각 범주형 필드의 경우, 추정 주변 평균을 생성할지 여부를 지정합니다.
covariance_list	<i>structure</i>	고정 효과 목록의 각 연속형 필드의 경우, 추정 주변 평균을 계산할 때 평균 또는 사용자 정의 값을 사용할지 여부를 지정합니다.
mean_scale	Original Transformed	대상의 원래 척도(기본값) 또는 연결함수 변환을 기반으로 추정 주변 평균을 계산할지 여부를 지정합니다.
comparison_adjustment_method	LSD SEQBONFERRONI SEQSIDAK	다중 대비를 갖고 가설검정을 수행할 때 사용할 조정 방법입니다.

gle 특성



GLE는 목표가 비정규 분포를 갖고 지정된 연결함수를 통해 요인 및 공변량과 선형으로 관련되어 관측값을 상관시킬 수 있도록 일반 선형 모델을 확장합니다. 일반화 선형 혼합 모델은 단순 선형 회귀에서 비정규 장기적인 데이터에 대한 복합 다중 수준 모델에 이르기까지 다양한 모델을 포함합니다.

표 128. gle 특성

gle 특성	값	특성 설명
custom_target	<i>flag</i>	업스트림 노드에서 정의된 대상(<i>false</i>) 또는 <i>target_field</i> 에 의해 지정되는 사용자 정의 대상(<i>true</i>)을 사용할지 여부를 표시합니다.
target_field	<i>field</i>	<i>custom_target</i> 이 <i>true</i> 인 경우 대상으로 사용할 필드입니다.
use_trials	<i>flag</i>	시행 수를 지정하는 추가 필드 또는 값을 대상 반응이 시행 세트에서 발생하는 이벤트 수일 때 사용할지 여부를 표시합니다. 기본값은 <i>false</i> 입니다.
use_trials_field_or_value	Field Value	필드(기본값) 또는 값이 시행 수를 지정하는 데 사용하는지 여부를 표시합니다.
trials_field	<i>field</i>	시행 수를 지정하는 데 사용할 필드입니다.
trials_value	<i>integer</i>	시행 수를 지정하는 데 사용할 값입니다. 지정되는 경우 최소값은 1입니다.
use_custom_target_reference	<i>flag</i>	사용자 정의 참조 범주가 범주형 대상에 사용할지 여부를 표시합니다. 기본값은 <i>false</i> 입니다.
target_reference_value	<i>string</i>	<i>use_custom_target_reference</i> 가 <i>true</i> 인 경우 사용할 참조 범주입니다.
dist_link_combination	NormalIdentity GammaLog PoissonLog NegbinLog TweedieIdentity NominalLogit BinomialLogit BinomialProbit BinomialLogC CUSTOM	대상에 대한 값의 분포를 위한 공통 모델입니다. <i>target_distribution</i> 에 의해 제공되는 목록에서 분포를 지정하려면 CUSTOM을 선택하십시오.
target_distribution	Normal Binomial Multinomial Gamma INVERSE_GAUSS NEG_BINOMIAL Poisson TWEEDIE UNKNOWN	<i>dist_link_combination</i> 은 Custom일 때 대상에 대한 값의 분포입니다.

표 128. *gle* 특성 (계속)

gle 특성	값	특성 설명
link_function_type	UNKNOWN IDENTITY LOG LOGIT PROBIT COMPL_LOG_LOG POWER LOG_COMPL NEG_LOG_LOG ODDS_POWER NEG_BINOMIAL GEN_LOGIT CUMUL_LOGIT CUMUL_PROBIT CUMUL_COMPL_LOG_LOG CUMUL_NEG_LOG_LOG CUMUL_CAUCHIT	대상 값을 예측자와 관련시키는 연결함수입니다. target_distribution이 Binomial이면 다음을 사용할 수 있습니다. UNKNOWN IDENTITY LOG LOGIT PROBIT COMPL_LOG_LOG POWER LOG_COMPL NEG_LOG_LOG ODDS_POWER target_distribution이 NEG_BINOMIAL이면 다음을 사용할 수 있습니다. NEG_BINOMIAL. target_distribution이 UNKNOWN이면 다음을 사용할 수 있습니다. GEN_LOGIT CUMUL_LOGIT CUMUL_PROBIT CUMUL_COMPL_LOG_LOG CUMUL_NEG_LOG_LOG CUMUL_CAUCHIT
link_function_param	<i>number</i>	사용할 Tweedie 모수값입니다. normal_link_function 또는 link_function_type이 POWER인 경우에만 적용 가능합니다.
tweedie_param	<i>number</i>	사용할 연결함수 모수값입니다. dist_link_combination이 TweedieIdentity로 설정되거나 link_function_type이 TWEEDIE로 설정된 경우에만 적용됩니다.
use_predefined_inputs	<i>flag</i>	모델 효과 필드가 입력 필드로서 정의된 업스트림(true) 또는 fixed_effects_list의 필드(false)인지 여부를 표시합니다.
model_effects_list	<i>structured</i>	use_predefined_inputs가 false인 경우 모델 효과 필드로 사용할 입력 필드를 지정합니다.
use_intercept	<i>flag</i>	true(기본값)인 경우 모델의 절편을 포함합니다.
regression_weight_field	<i>field</i>	분석 가중값으로 사용할 필드입니다.
use_offset	None Value Variable	오프셋이 지정되는 방법을 표시합니다. 값 None은 오프셋을 사용하지 않음을 의미합니다.
offset_value	<i>number</i>	use_offset이 offset_value로 설정되는 경우 오프셋에 사용할 값입니다.

표 128. gle 특성 (계속)

gle 특성	값	특성 설명
offset_field	field	use_offset이 offset_field로 설정된 경우 오프셋 값에 사용할 필드입니다.
target_category_order	Ascending Descending	범주형 대상에 대한 정렬 순서입니다. 기본값은 Ascending입니다.
inputs_category_order	Ascending Descending	범주형 예측변수에 대한 정렬 순서입니다. 기본값은 Ascending입니다.
max_iterations	integer	알고리즘이 수행할 최대 반복 수입니다. 음이 아닌 정수이며, 기본값은 100입니다.
confidence_level	number	모델 계수의 구간 추정값을 계산하는 데 사용하는 신뢰수준입니다. 음이 아닌 정수이며, 최대값은 100입니다. 기본값은 95입니다.
test_fixed_effects_coefficients	Model Robust	모수 추정값 공분산 교차표 계산 방법입니다.
detect_outliers	flag	참인 경우 알고리즘이 다항 분포를 제외한 모든 분포의 영향력 있는 이상값을 찾습니다.
conduct_trend_analysis	flag	참인 경우 알고리즘이 산점도 도표의 추세 분석을 수행합니다.
estimation_method	FISHER_SCORING NEWTON_RAPHSON HYBRID	최대우도 추정 알고리즘을 지정하십시오.
max_fisher_iterations	integer	FISHER_SCORING estimation_method를 사용 중인 경우, 최대 반복수입니다. 최소 0, 최대 20입니다.
scale_parameter_method	MLE FIXED DEVIANCE PEARSON_CHISQUARE	척도 모수의 추정에 사용할 방법을 지정하십시오.
scale_value	number	scale_parameter_method가 Fixed로 설정된 경우에만 사용 가능합니다.
negative_binomial_method	MLE FIXED	음이항 보조 모수의 추정에 사용할 방법을 지정하십시오.
negative_binomial_value	number	negative_binomial_method가 Fixed로 설정된 경우에만 사용 가능합니다.
use_p_converge	flag	모수 수렴에 대한 옵션입니다.
p_converge	number	공란 또는 임의의 양수값입니다.
p_converge_type	flag	참 = 절대값, 거짓 = 상대값
use_l_converge	flag	로그-우도 수렴에 대한 옵션입니다.
l_converge	number	공란 또는 임의의 양수값입니다.
l_converge_type	flag	참 = 절대값, 거짓 = 상대값
use_h_converge	flag	Hessian 수렴에 대한 옵션입니다.
h_converge	number	공란 또는 임의의 양수값입니다.
h_converge_type	flag	참 = 절대값, 거짓 = 상대값
max_iterations	integer	알고리즘이 수행할 최대 반복 수입니다. 음이 아닌 정수이며, 기본값은 100입니다.
sing_tolerance	integer	

표 128. *gle* 특성 (계속)

gle 특성	값	특성 설명
use_model_selection	<i>flag</i>	모두 임계값 및 모델 선택 방법 제어를 사용으로 설정합니다..
method	LASSO ELASTIC_NET FORWARD_STEPWISE RIDGE	모델 선택 방법을 판별하거나 Ridge를 사용하는 경우 정규화 방법이 사용됩니다.
detect_two_way_interactions	<i>flag</i>	True인 경우 모델은 입력 필드 간의 이원 상호작용을 자동으로 발견합니다. 이 제어는 모델이 주효과이며(즉, 사용자가 더 높은 순서의 기타 효과를 작성하지 않은 경우) 선택된 method가 단계별 전진, Lasso 또는 Elastic Net인 경우에만 사용되어야 합니다.
automatic_penalty_params	<i>flag</i>	모델 선택 method가 Lasso 또는 Elastic net인 경우에만 사용 가능합니다. 이 함수를 사용하여 Lasso 또는 Elastic net 변수 선택 방법과 연관된 페널티 모수를 입력하십시오. True인 경우 기본값이 사용됩니다. False인 경우 페널티 모수가 사용되며 사용자 정의 값을 입력할 수 있습니다.
lasso_penalty_param	<i>number</i>	모델 선택 method가 Lasso 또는 Elastic net이며 automatic_penalty_params가 False인 경우에만 사용 가능합니다. Lasso의 페널티 모수값을 지정하십시오.
elastic_net_penalty_param1	<i>number</i>	모델 선택 method가 Lasso 또는 Elastic net이며 automatic_penalty_params가 False인 경우에만 사용 가능합니다. Elastic Net 모수 1의 페널티 모수값을 지정하십시오.
elastic_net_penalty_param2	<i>number</i>	모델 선택 method가 Lasso 또는 Elastic net이며 automatic_penalty_params가 False인 경우에만 사용 가능합니다. Elastic Net 모수 2의 페널티 모수값을 지정하십시오.
probability_entry	<i>number</i>	선택된 method가 단계별 전진인 경우에만 사용 가능합니다. 효과 포함에 대해 F 통계량 기준의 유의 수준 레벨을 지정하십시오.
probability_removal	<i>number</i>	선택된 method가 단계별 전진인 경우에만 사용 가능합니다. 제거 포함에 대해 F 통계량 기준의 유의 수준 레벨을 지정하십시오.
use_max_effects	<i>flag</i>	선택된 method가 단계별 전진인 경우에만 사용 가능합니다. max_effects 제어를 사용으로 설정합니다. False인 경우 포함된 효과의 기본값은 모델에 제공된 효과의 총 수에서 절편을 뺀 값과 같아야 합니다.
max_effects	<i>integer</i>	단계별 전진 작성 방법을 사용할 때의 최대 효과 수를 지정합니다.

표 128. *gle* 특성 (계속)

gle 특성	값	특성 설명
use_max_steps	<i>flag</i>	max_steps 제어를 사용으로 설정합니다. False인 경우 단계의 기본값은 모델에 제공된 효과의 수의 3배에서 절편을 뺀 값과 같아야 합니다.
max_steps	<i>integer</i>	단계별 전진 작성 method를 사용할 때 적용할 최대 단계 수를 지정합니다.
use_model_name	<i>flag</i>	모델의 사용자 정의 이름을 지정(true)하거나 시스템 생성 이름을 사용(false)하는지 여부를 표시합니다. 기본값은 false입니다.
model_name	<i>string</i>	use_model_name이 true인 경우 사용할 모델 이름을 지정합니다.
usePI	<i>flag</i>	true이면 예측자 중요도가 계산됩니다.

kmeansnode 특성



K-평균 노드는 데이터 세트를 고유 그룹(또는 군집)으로 군집화합니다. 이 방법은 고정된 수의 군집을 정의하고 반복적으로 레코드를 군집에 지정하며, 추가 세분화가 더 이상 모델을 향상시킬 수 없을 때까지 군집중심을 조정합니다. 결과를 예상하는 대신 k-평균은 자율 학습으로 알려진 프로세스를 사용하여 입력 필드 세트의 패턴을 찾아냅니다.

예제

```
node = stream.create("kmeans", "My node")
# "Fields" tab
node.setPropertyValue("custom_fields", True)
node.setPropertyValue("inputs", ["Cholesterol", "BP", "Drug", "Na", "K", "Age"])
# "Model" tab
node.setPropertyValue("use_model_name", True)
node.setPropertyValue("model_name", "Kmeans_allinputs")
node.setPropertyValue("num_clusters", 9)
node.setPropertyValue("gen_distance", True)
node.setPropertyValue("cluster_label", "Number")
node.setPropertyValue("label_prefix", "Kmeans_")
node.setPropertyValue("optimize", "Speed")
# "Expert" tab
node.setPropertyValue("mode", "Expert")
node.setPropertyValue("stop_on", "Custom")
node.setPropertyValue("max_iterations", 10)
node.setPropertyValue("tolerance", 3.0)
node.setPropertyValue("encoding_value", 0.3)
```

표 129. *kmeansnode* 특성

kmeansnode 특성	값	특성 설명
inputs	[<i>field1</i> ... <i>fieldN</i>]	K-평균 모델은 입력 필드 세트에 대한 군집분석을 수행하지만 목표 필드를 사용하지 않습니다. 가중치 및 빈도 필드는 사용하지 않습니다. 자세한 정보는 191 페이지의 『공통 모델링 노드 특성』 주제를 참조하십시오.
num_clusters	<i>number</i>	
gen_distance	<i>flag</i>	
cluster_label	String Number	
label_prefix	<i>string</i>	
mode	Simple Expert	
stop_on	Default Custom	
max_iterations	<i>number</i>	
tolerance	<i>number</i>	
encoding_value	<i>number</i>	
optimize	Speed Memory	모델 작성이 속도 또는 메모리에 대해 최적화되어야 하는지 여부를 지정하는 데 사용됩니다.

knnnode 특성



KNN(*k*-Nearest Neighbor) 노드는 새 케이스를 *k*가 정수인 예측자 공간에서 가장 가까이에 있는 *k* 오브젝트의 범주 또는 값과 연관시킵니다. 유사한 케이스는 서로 가까이에 있고 유사하지 않은 케이스는 서로 멀리 떨어져 있습니다.

예제

```
node = stream.create("knn", "My node")
# Objectives tab
node.setPropertyValue("objective", "Custom")
# Settings tab - Neighbors panel
node.setPropertyValue("automatic_k_selection", False)
node.setPropertyValue("fixed_k", 2)
node.setPropertyValue("weight_by_importance", True)
# Settings tab - Analyze panel
node.setPropertyValue("save_distances", True)
```

표 130. *knnnode* 특성

knnnode 특성	값	특성 설명
analysis	PredictTarget IdentifyNeighbors	

표 130. knnnode 특성 (계속)

knnnode 특성	값	특성 설명
objective	Balance Speed Accuracy Custom	
normalize_ranges	<i>flag</i>	
use_case_labels	<i>flag</i>	다음 옵션을 가능하게 하는 확인 상자.
case_labels_field	<i>field</i>	
identify_focal_cases	<i>flag</i>	다음 옵션을 가능하게 하는 확인 상자.
focal_cases_field	<i>field</i>	
automatic_k_selection	<i>flag</i>	
fixed_k	<i>integer</i>	automatic_k_selectio가 False인 경우에만 사용.
minimum_k	<i>integer</i>	automatic_k_selectio가 True인 경우에만 사용.
maximum_k	<i>integer</i>	
distance_computation	Euclidean CityBlock	
weight_by_importance	<i>flag</i>	
range_predictions	Mean Median	
perform_feature_selection	<i>flag</i>	
forced_entry_inputs	[<i>field1 ... fieldN</i>]	
stop_on_error_ratio	<i>flag</i>	
number_to_select	<i>integer</i>	
minimum_change	<i>number</i>	
validation_fold_assign_by_field	<i>flag</i>	
number_of_folds	<i>integer</i>	validation_fold_assign_by_field가 False인 경우에만 사용
set_random_seed	<i>flag</i>	
random_seed	<i>number</i>	
folds_field	<i>field</i>	validation_fold_assign_by_field가 True인 경우에만 사용
all_probabilities	<i>flag</i>	
save_distances	<i>flag</i>	
calculate_raw_propensities	<i>flag</i>	
calculate_adjusted_propensities	<i>flag</i>	
adjusted_propensity_partition	Test Validation	

kohonennode 특성



코호넨 노드는 데이터 세트를 고유 그룹으로 군집화하는 데 사용할 수 있는 신경망 유형을 생성합니다. 네트워크가 완전히 숙달되면, 유사 레코드는 출력 맵 가까이 있지만, 다른 레코드는 멀리 떨어져 있을 것 입니다. 모델 너깃에서 각 단위별로 캡처된 관측값을 살펴 강한 단위를 식별할 수 있습니다. 이것은 적당한 군집 수에 대한 감각을 제공할 것 입니다.

예제

```
node = stream.create("kohonen", "My node")
# "Model" tab
node.setPropertyValue("use_model_name", False)
node.setPropertyValue("model_name", "Symbolic Cluster")
node.setPropertyValue("stop_on", "Time")
node.setPropertyValue("time", 1)
node.setPropertyValue("set_random_seed", True)
node.setPropertyValue("random_seed", 12345)
node.setPropertyValue("optimize", "Speed")
# "Expert" tab
node.setPropertyValue("mode", "Expert")
node.setPropertyValue("width", 3)
node.setPropertyValue("length", 3)
node.setPropertyValue("decay_style", "Exponential")
node.setPropertyValue("phase1_neighborhood", 3)
node.setPropertyValue("phase1_eta", 0.5)
node.setPropertyValue("phase1_cycles", 10)
node.setPropertyValue("phase2_neighborhood", 1)
node.setPropertyValue("phase2_eta", 0.2)
node.setPropertyValue("phase2_cycles", 75)
```

표 131. kohonennode 특성

kohonennode 특성	값	특성 설명
inputs	[<i>field1 ... fieldN</i>]	코호넨 모델은 입력 필드 목록을 사용하지만 목표는 사용하지 않습니다. 빈도 및 가중 필드는 사용하지 않습니다. 자세한 정보는 191 페이지의 『공통 모델링 노드 특성』 주제를 참조하십시오.
continue	<i>flag</i>	
show_feedback	<i>flag</i>	
stop_on	Default Time	
time	<i>number</i>	
optimize	Speed Memory	모델 작성이 속도 또는 메모리에 대해 최적화되어야 하는지 여부를 지정하는 데 사용됩니다.
cluster_label	<i>flag</i>	
mode	Simple Expert	
width	<i>number</i>	

표 131. *kohonenmode* 특성 (계속)

kohonenmode 특성	값	특성 설명
length	<i>number</i>	
decay_style	Linear Exponential	
phase1_neighborhood	<i>number</i>	
phase1_eta	<i>number</i>	
phase1_cycles	<i>number</i>	
phase2_neighborhood	<i>number</i>	
phase2_eta	<i>number</i>	
phase2_cycles	<i>number</i>	

linearnode 특성



선형 회귀 모형은 목표와 하나 이상의 예측변수 간의 선형 관계를 기반으로 연속형 목표를 예측합니다.

예제

```
node = stream.create("linear", "My node")
# Build Options tab - Objectives panel
node.setPropertyValue("objective", "Standard")
# Build Options tab - Model Selection panel
node.setPropertyValue("model_selection", "BestSubsets")
node.setPropertyValue("criteria_best_subsets", "ASE")
# Build Options tab - Ensembles panel
node.setPropertyValue("combining_rule_categorical", "HighestMeanProbability")
```

표 132. *linearnode* 특성.

linearnode 특성	값	특성 설명
target	<i>field</i>	단일 목표 필드를 지정합니다.
inputs	[<i>field1 ... fieldN</i>]	모델이 사용하는 예측변수 필드
continue_training_existing_model	<i>flag</i>	
objective	Standard Bagging Boosting psm	psm은 매우 큰 데이터 세트에 사용하며 서버 연결이 필요합니다.
use_auto_data_preparation	<i>flag</i>	
confidence_level	<i>number</i>	

표 132. *linearnode* 특성 (계속).

linearnode 특성	값	특성 설명
model_selection	ForwardStepwise BestSubsets None	
criteria_forward_stepwise	AICC Fstatistics AdjustedRSquare ASE	
probability_entry	<i>number</i>	
probability_removal	<i>number</i>	
use_max_effects	<i>flag</i>	
max_effects	<i>number</i>	
use_max_steps	<i>flag</i>	
max_steps	<i>number</i>	
criteria_best_subsets	AICC AdjustedRSquare ASE	
combining_rule_continuous	Mean Median	
component_models_n	<i>number</i>	
use_random_seed	<i>flag</i>	
random_seed	<i>number</i>	
use_custom_model_name	<i>flag</i>	
custom_model_name	<i>string</i>	
use_custom_name	<i>flag</i>	
custom_name	<i>string</i>	
tooltip	<i>string</i>	
keywords	<i>string</i>	
annotation	<i>string</i>	

linearnode 특성



선형 회귀 모형은 목표와 하나 이상의 예측변수 간의 선형 관계를 기반으로 연속형 목표를 예측합니다.

표 133. *linearnode* 특성

linearnode 특성	값	특성 설명
target	<i>field</i>	단일 목표 필드를 지정합니다.

표 133. *linearasnode* 특성 (계속)

linearasnode 특성	값	특성 설명
inputs	[<i>field1</i> ... <i>fieldN</i>]	모델이 사용하는 예측변수 필드
weight_field	<i>field</i>	모델이 사용하는 분석 필드
custom_fields	<i>flag</i>	기본값은 TRUE입니다.
intercept	<i>flag</i>	기본값은 TRUE입니다.
detect_2way_interaction	<i>flag</i>	양방향 상호작용을 고려할지 여부입니다. 기본값은 TRUE입니다.
cin	<i>number</i>	모델 계수의 추정값을 계산하기 위해 사용하는 신뢰도 구간입니다. 0보다 크고 100보다 작은 값을 지정하십시오. 기본값은 95입니다.
factor_order	ascending descending	범주형 예측변수의 정렬 순서입니다. 기본값은 ascending입니다.
var_select_method	ForwardStepwise BestSubsets none	사용할 모델 선택 방법입니다. 기본값은 ForwardStepwise입니다.
criteria_for_forward_stepwise	AICC Fstatistics AdjustedRSquare ASE	모델에서 효과를 추가 또는 제거해야 하는지 여부를 판별하기 위해 사용하는 통계입니다. 기본값은 AdjustedRSquare입니다.
pin	<i>number</i>	이 지정된 pin 임계값 미만의 가장 작은 P-값을 갖는 효과가 모델에 추가됩니다. 기본값은 0.05입니다.
pout	<i>number</i>	모델에서 이 지정된 pout 임계값보다 큰 p-값을 갖는 모든 효과가 제거됩니다. 기본값은 0.10입니다.
use_custom_max_effects	<i>flag</i>	최종 모델에서 최대 효과 수를 사용할지 여부입니다. 기본값은 FALSE입니다.
max_effects	<i>number</i>	최종 모델에서 사용할 최대 효과 수입입니다. 기본값은 1입니다.
use_custom_max_steps	<i>flag</i>	최대 단계 수 사용 여부입니다. 기본값은 FALSE입니다.
max_steps	<i>number</i>	단계 선택 알고리즘이 중지하기 전의 최대 단계 수입입니다. 기본값은 1입니다.
criteria_for_best_subsets	AICC AdjustedRSquare ASE	사용할 기준 모드입니다. 기본값은 AdjustedRSquare입니다.

logregnode 특성



로지스틱 회귀분석은 입력 필드 값을 기반으로 레코드를 분류하는 통계 기법입니다. 선형 회귀와 유사하지만 숫자 범위 대신 범주형 목표 필드를 사용합니다.

다항 예제

```
node = stream.create("logreg", "My node")
# "Fields" tab
node.setPropertyValue("custom_fields", True)
node.setPropertyValue("target", "Drug")
node.setPropertyValue("inputs", ["BP", "Cholesterol", "Age"])
node.setPropertyValue("partition", "Test")
# "Model" tab
node.setPropertyValue("use_model_name", True)
node.setPropertyValue("model_name", "Log_reg Drug")
node.setPropertyValue("use_partitioned_data", True)
node.setPropertyValue("method", "Stepwise")
node.setPropertyValue("logistic_procedure", "Multinomial")
node.setPropertyValue("multinomial_base_category", "BP")
node.setPropertyValue("model_type", "FullFactorial")
node.setPropertyValue("custom_terms", [["BP", "Sex"], ["Age"], ["Na", "K"]])
node.setPropertyValue("include_constant", False)
# "Expert" tab
node.setPropertyValue("mode", "Expert")
node.setPropertyValue("scale", "Pearson")
node.setPropertyValue("scale_value", 3.0)
node.setPropertyValue("all_probabilities", True)
node.setPropertyValue("tolerance", "1.0E-7")
# "Convergence..." section
node.setPropertyValue("max_iterations", 50)
node.setPropertyValue("max_steps", 3)
node.setPropertyValue("l_converge", "1.0E-3")
node.setPropertyValue("p_converge", "1.0E-7")
node.setPropertyValue("delta", 0.03)
# "Output..." section
node.setPropertyValue("summary", True)
node.setPropertyValue("likelihood_ratio", True)
node.setPropertyValue("asymptotic_correlation", True)
node.setPropertyValue("goodness_fit", True)
node.setPropertyValue("iteration_history", True)
node.setPropertyValue("history_steps", 3)
node.setPropertyValue("parameters", True)
node.setPropertyValue("confidence_interval", 90)
node.setPropertyValue("asymptotic_covariance", True)
node.setPropertyValue("classification_table", True)
# "Stepping" options
node.setPropertyValue("min_terms", 7)
node.setPropertyValue("use_max_terms", True)
node.setPropertyValue("max_terms", 10)
node.setPropertyValue("probability_entry", 3)
node.setPropertyValue("probability_removal", 5)
node.setPropertyValue("requirements", "Containment")
```

이항 예제

```
node = stream.create("logreg", "My node")
# "Fields" tab
node.setPropertyValue("custom_fields", True)
```

```

node.setPropertyValue("target", "Cholesterol")
node.setPropertyValue("inputs", ["BP", "Drug", "Age"])
node.setPropertyValue("partition", "Test")
# "Model" tab
node.setPropertyValue("use_model_name", False)
node.setPropertyValue("model_name", "Log_reg Cholesterol")
node.setPropertyValue("multinomial_base_category", "BP")
node.setPropertyValue("use_partitioned_data", True)
node.setPropertyValue("binomial_method", "Forwards")
node.setPropertyValue("logistic_procedure", "Binomial")
node.setPropertyValue("binomial_categorical_input", "Sex")
node.setKeyedPropertyValue("binomial_input_contrast", "Sex", "Simple")
node.setKeyedPropertyValue("binomial_input_category", "Sex", "Last")
node.setPropertyValue("include_constant", False)
# "Expert" tab
node.setPropertyValue("mode", "Expert")
node.setPropertyValue("scale", "Pearson")
node.setPropertyValue("scale_value", 3.0)
node.setPropertyValue("all_probabilities", True)
node.setPropertyValue("tolerance", "1.0E-7")
# "Convergence..." section
node.setPropertyValue("max_iterations", 50)
node.setPropertyValue("l_converge", "1.0E-3")
node.setPropertyValue("p_converge", "1.0E-7")
# "Output..." section
node.setPropertyValue("binomial_output_display", "at_each_step")
node.setPropertyValue("binomial_goodness_of_fit", True)
node.setPropertyValue("binomial_iteration_history", True)
node.setPropertyValue("binomial_parameters", True)
node.setPropertyValue("binomial_ci_enable", True)
node.setPropertyValue("binomial_ci", 85)
# "Stepping" options
node.setPropertyValue("binomial_removal_criterion", "LR")
node.setPropertyValue("binomial_probability_removal", 0.2)

```

표 134. logregnode 특성.

logregnode 특성	값	특성 설명
target	<i>field</i>	로지스틱 회귀분석 모델은 단일 목표 필드와 하나 이상의 입력 필드가 필요합니다. 빈도 및 가중 필드는 사용하지 않습니다. 자세한 정보는 191 페이지의 『공통 모델링 노트 특성』 주제를 참조하십시오.
logistic_procedure	Binomial Multinomial	
include_constant	<i>flag</i>	
mode	Simple Expert	

표 134. logregnode 특성 (계속).

logregnode 특성	값	특성 설명
method	Enter Stepwise Forwards Backwards BackwardsStepwise	
binomial_method	Enter Forwards Backwards	
model_type	MainEffects FullFactorial Custom	FullFactorial이 모델 유형으로 지정될 때 단계화 방법은 지정된 경우에도 실행되지 않습니다. 대신, Enter가 사용하는 메소드입니다. 모델 유형이 Custom으로 설정되지만 사용자 정의 필드가 지정되지 않는 경우, 주효과 모델이 작성됩니다.
custom_terms	[[BP Sex][BP][Age]]	
multinomial_base_category	string	참조 범주가 판별되는 방식을 지정합니다.
binomial_categorical_input	string	
binomial_input_contrast	Indicator Simple Difference Helmert Repeated Polynomial Deviation	대비가 판별되는 방법을 지정하는 범주형 입력에 대한 키가 있는 특성입니다.
binomial_input_category	First Last	참조범주가 판별되는 방법을 지정하는 범주형 입력에 대한 키가 있는 특성입니다.
scale	None UserDefined Pearson Deviance	
scale_value	number	
all_probabilities	flag	
tolerance	1.0E-5 1.0E-6 1.0E-7 1.0E-8 1.0E-9 1.0E-10	
min_terms	number	
use_max_terms	flag	
max_terms	number	

표 134. logregnode 특성 (계속).

logregnode 특성	값	특성 설명
entry_criterion	Score LR	
removal_criterion	LR Wald	
probability_entry	<i>number</i>	
probability_removal	<i>number</i>	
binomial_probability_entry	<i>number</i>	
binomial_probability_removal	<i>number</i>	
requirements	HierarchyDiscrete HierarchyAll Containment None	
max_iterations	<i>number</i>	
max_steps	<i>number</i>	
p_converge	1.0E-4 1.0E-5 1.0E-6 1.0E-7 1.0E-8 0	
l_converge	1.0E-1 1.0E-2 1.0E-3 1.0E-4 1.0E-5 0	
delta	<i>number</i>	
iteration_history	<i>flag</i>	
history_steps	<i>number</i>	
summary	<i>flag</i>	
likelihood_ratio	<i>flag</i>	
asymptotic_correlation	<i>flag</i>	
goodness_fit	<i>flag</i>	
parameters	<i>flag</i>	
confidence_interval	<i>number</i>	
asymptotic_covariance	<i>flag</i>	
classification_table	<i>flag</i>	
stepwise_summary	<i>flag</i>	
info_criteria	<i>flag</i>	
monotonicity_measures	<i>flag</i>	
binomial_output_display	at_each_step at_last_step	

표 134. logregnode 특성 (계속).

logregnode 특성	값	특성 설명
binomial_goodness_of_fit	flag	
binomial_parameters	flag	
binomial_iteration_history	flag	
binomial_classification_plots	flag	
binomial_ci_enable	flag	
binomial_ci	number	
binomial_residual	outliers all	
binomial_residual_enable	flag	
binomial_outlier_threshold	number	
binomial_classification_cutoff	number	
binomial_removal_criterion	LR Wald Conditional	
calculate_variable_importance	flag	
calculate_raw_propensities	flag	

lsvmnode 특성



선형 지원 벡터 머신(LSVM) 노드를 사용하면 과적합 없이 두 개의 그룹 중 하나로 데이터를 분류할 수 있습니다. LSVM은 선형이며, 다수의 레코드가 있는 데이터 세트와 같은 광범위한 데이터 세트와 함께 잘 작동합니다.

표 135. lsvmnode 특성

lsvmnode 특성	값	특성 설명
intercept	flag	모델에 절편을 포함합니다. 기본값은 참입니다.
target_order	Ascending Descending	범주형 대상에 대한 정렬 순서를 지정합니다. 연속형 대상에서는 무시됩니다. 기본값은 Ascending입니다.
precision	number	목표 필드의 측정 수준이 Continuous인 경우에만 사용합니다. 회귀분석의 손실에 대한 민감도와 관련된 모수를 지정합니다. 최소값은 0이고 최대값은 없습니다. 기본값은 0.1입니다.
exclude_missing_values	flag	참인 경우 단일 값이 결측되면 레코드가 제외됩니다. 기본값은 False입니다.
penalty_function	L1 L2	사용되는 페널티 함수 유형을 지정합니다. 기본값은 L2입니다.
lambda	number	페널티(정규화) 모수입니다.

표 135. *lsvmnode* 특성 (계속)

lsvmnode 특성	값	특성 설명
calculate_variable_importance	flag	중요도의 적절한 측도를 생성하는 모델에 대해 이 옵션은 모델을 추정하는 각 예측 변수의 상대 중요도를 표시하는 차트를 표시합니다. 일부 모델의 경우, 특히 대형 데이터베이스와 작업하는 경우에 변수 중요도를 계산하는 데 시간이 오래 걸릴 수 있으며 그 결과로 일부 모델의 경우 기본으로 줄어들 수 있습니다. 변수 중요도는 의사결정 목록 모델에 사용할 수 없습니다.

neuralnetnode 특성

중요사항: 신경망 모델링 노드의 기능이 향상된 새 버전이 이 릴리스에서 사용 가능하며 다음 절 (*neuralnetwork*)에서 설명됩니다. 아직 이전 버전을 사용하여 모델을 작성하고 스코어링 할 수 있지만, 새 버전으로 사용하도록 스크립트를 업데이트할 것을 권장합니다. 이전 버전의 세부사항이 참조를 위해 여기에 보존됩니다.

예제

```
node = stream.create("neuralnet", "My node")
# "Fields" tab
node.setPropertyValue("custom_fields", True)
node.setPropertyValue("targets", ["Drug"])
node.setPropertyValue("inputs", ["Age", "Na", "K", "Cholesterol", "BP"])
# "Model" tab
node.setPropertyValue("use_partitioned_data", True)
node.setPropertyValue("method", "Dynamic")
node.setPropertyValue("train_pct", 30)
node.setPropertyValue("set_random_seed", True)
node.setPropertyValue("random_seed", 12345)
node.setPropertyValue("stop_on", "Time")
node.setPropertyValue("accuracy", 95)
node.setPropertyValue("cycles", 200)
node.setPropertyValue("time", 3)
node.setPropertyValue("optimize", "Speed")
# "Multiple Method Expert Options" section
node.setPropertyValue("m_topologies", "5 30 5; 2 20 3, 1 10 1")
node.setPropertyValue("m_non_pyramids", False)
node.setPropertyValue("m_persistence", 100)
```

표 136. *neuralnetnode* 특성

neuralnetnode 특성	값	특성 설명
targets	[field1 ... fieldN]	신경망 노드는 하나 이상의 목표 필드와 하나 이상의 입력 필드가 필요합니다. 빈도 및 가중 필드는 무시됩니다. 자세한 정보는 191 페이지의 『공통 모델링 노드 특성』 주제를 참조하십시오.

표 136. *neuralnetnode* 특성 (계속)

neuralnetnode 특성	값	특성 설명
method	Quick Dynamic Multiple Prune ExhaustivePrune RBFN	
prevent_overtrain	<i>flag</i>	
train_pct	<i>number</i>	
set_random_seed	<i>flag</i>	
random_seed	<i>number</i>	
mode	Simple Expert	
stop_on	Default Accuracy Cycles Time	중지 모드.
accuracy	<i>number</i>	정확도 중지.
cycles	<i>number</i>	훈련 주기.
time	<i>number</i>	훈련 시간(분)
continue	<i>flag</i>	
show_feedback	<i>flag</i>	
binary_encode	<i>flag</i>	
use_last_model	<i>flag</i>	
gen_logfile	<i>flag</i>	
logfile_name	<i>string</i>	
alpha	<i>number</i>	
initial_eta	<i>number</i>	
high_eta	<i>number</i>	
low_eta	<i>number</i>	
eta_decay_cycles	<i>number</i>	
hid_layers	One Two Three	
h1_units_one	<i>number</i>	
h1_units_two	<i>number</i>	
h1_units_three	<i>number</i>	
persistence	<i>number</i>	
m_topologies	<i>string</i>	
m_non_pyramids	<i>flag</i>	
m_persistence	<i>number</i>	

표 136. *neuralnetnode* 특성 (계속)

neuralnetnode 특성	값	특성 설명
p_hid_layers	One Two Three	
p_hl_units_one	number	
p_hl_units_two	number	
p_hl_units_three	number	
p_persistence	number	
p_hid_rate	number	
p_hid_pers	number	
p_inp_rate	number	
p_inp_pers	number	
p_overall_pers	number	
r_persistence	number	
r_num_clusters	number	
r_eta_auto	flag	
r_alpha	number	
r_eta	number	
optimize	Speed Memory	모델 작성이 속도 또는 메모리에 대해 최적화되어야 하는지 여부를 지정하는 데 사용합니다.
calculate_variable_importance	flag	참고: 이전 릴리스에서 사용한 <i>sensitivity_analysis</i> 특성은 이 특성을 위해 더 이상 사용하지 않습니다. 이전 특성이 아직 지원되지만 <i>calculate_variable_importance</i> 가 권장됩니다.
calculate_raw_propensities	flag	
calculate_adjusted_propensities	flag	
adjusted_propensity_partition	Test Validation	

neuralnetworknode 특성



신경망 노드는 인간 두뇌가 정보를 처리하는 방법의 단순화된 모델을 사용합니다. 뉴런의 추상 버전을 닮은 상호연결된 많은 수의 단순 처리 장치를 시뮬레이션하여 작업합니다. 신경망은 강력한 범용 함수 추정량이며 학습하거나 적용하기 위해 약간의 통계 또는 수학적 지식이 필요합니다.

예제

```
node = stream.create("neuralnetwork", "My node")
# Build Options tab - Objectives panel
node.setPropertyValue("objective", "Standard")
# Build Options tab - Ensembles panel
node.setPropertyValue("combining_rule_categorical", "HighestMeanProbability")
```

표 137. neuralnetworknode 특성

neuralnetworknode 특성	값	특성 설명
targets	[field1 ... fieldN]	목표 필드를 지정합니다.
inputs	[field1 ... fieldN]	모델이 사용하는 예측변수 필드
splits	[field1 ... fieldN]	분할 모델링에 사용할 필드를 지정합니다.
use_partition	flag	파티션 필드가 정의된 경우, 이 옵션은 훈련 파티션의 데이터만 모델을 작성하는 데 사용하도록 보장합니다.
continue	flag	기존 모델 학습 계속.
objective	Standard Bagging Boosting psm	psm은 매우 큰 데이터 세트에 사용하며 서버 연결이 필요합니다.
method	MultilayerPerceptron RadialBasisFunction	
use_custom_layers	flag	
first_layer_units	number	
second_layer_units	number	
use_max_time	flag	
max_time	number	
use_max_cycles	flag	
max_cycles	number	
use_min_accuracy	flag	
min_accuracy	number	
combining_rule_categorical	Voting HighestProbability HighestMeanProbability	
combining_rule_continuous	Mean Median	
component_models_n	number	
overfit_prevention_pct	number	
use_random_seed	flag	
random_seed	number	
missing_values	listwiseDeletion missingValueImputation	
use_model_name	boolean	
model_name	string	
confidence	onProbability onIncrease	
score_category_probabilities	flag	
max_categories	number	
score_propensity	flag	

표 137. *neuralnetworknode* 특성 (계속)

neuralnetworknode 특성	값	특성 설명
use_custom_name	<i>flag</i>	
custom_name	<i>string</i>	
tooltip	<i>string</i>	
keywords	<i>string</i>	
annotation	<i>string</i>	

questnode 특성



QUEST 노드는 의사결정 트리를 작성하기 위한 이분형 분류 방법을 제공하며, 대형 C&R 트리 분석에 필요한 처리 시간을 줄이는 동시에 분류 트리 방법에서 찾은 경향을 줄여 더 많은 분할을 허용하는 입력을 선호하도록 설계되었습니다. 입력 필드는 숫자 범위(연속)일 수 있지만 목표 필드는 범주형이어야 합니다. 모든 분할은 이분형입니다.

예제

```
node = stream.create("quest", "My node")
node.setPropertyValue("custom_fields", True)
node.setPropertyValue("target", "Drug")
node.setPropertyValue("inputs", ["Age", "Na", "K", "Cholesterol", "BP"])
node.setPropertyValue("model_output_type", "InteractiveBuilder")
node.setPropertyValue("use_tree_directives", True)
node.setPropertyValue("max_surrogates", 5)
node.setPropertyValue("split_alpha", 0.03)
node.setPropertyValue("use_percentage", False)
node.setPropertyValue("min_parent_records_abs", 40)
node.setPropertyValue("min_child_records_abs", 30)
node.setPropertyValue("prune_tree", True)
node.setPropertyValue("use_std_err", True)
node.setPropertyValue("std_err_multiplier", 3)
```

표 138. *questnode* 특성

questnode 특성	값	특성 설명
target	<i>field</i>	QUEST 모델은 하나의 목표 및 하나 이상의 입력 필드가 필요합니다. 빈도 필드도 지정할 수 있습니다. 자세한 정보는 191 페이지의 『공통 모델링 노드 특성』 주제를 참조하십시오.
continue_training_existing_model	<i>flag</i>	
objective	Standard Boosting Bagging psm	psm은 매우 큰 데이터 세트에 사용하며 서버 연결이 필요합니다.
model_output_type	Single InteractiveBuilder	
use_tree_directives	<i>flag</i>	

표 138. *questnode* 특성 (계속)

questnode 특성	값	특성 설명
tree_directives	string	
use_max_depth	Default Custom	
max_depth	integer	0부터 1000까지의 최대 트리 깊이입니다. use_max_depth = Custom인 경우에만 사용됩니다.
prune_tree	flag	과적합을 방지하기 위해 트리를 가지치기합니다.
use_std_err	flag	(표준 오차에서) 위험의 최대 차이를 사용합니다.
std_err_multiplier	number	최대 차이입니다.
max_surrogates	number	최대 서로게이트입니다.
use_percentage	flag	
min_parent_records_pc	number	
min_child_records_pc	number	
min_parent_records_abs	number	
min_child_records_abs	number	
use_costs	flag	
costs	structured	구조화 특성입니다.
priors	Data Equal Custom	
custom_priors	structured	구조화 특성입니다.
adjust_priors	flag	
trails	number	부스팅 또는 배경을 위한 구성요소 모델 수입니다.
set_ensemble_method	Voting HighestProbability HighestMeanProbability	범주형 대상에 대한 기본 결합 규칙입니다.
range_ensemble_method	Mean Median	연속형 대상에 대한 기본 결합 규칙입니다.
large_boost	flag	매우 큰 데이터 세트에 부스팅을 적용합니다.
split_alpha	number	분할 유의수준입니다.
train_pct	number	과적합 방지 세트입니다.
set_random_seed	flag	결과 복제 옵션입니다.
seed	number	
calculate_variable_importance	flag	
calculate_raw_propensities	flag	
calculate_adjusted_propensities	flag	
adjusted_propensity_partition	Test Validation	

randomtrees 특성



이 랜덤 트리 노드는 기존 C&RT 노드와 유사하지만, 빅 데이터를 처리하여 단일 트리를 작성하도록 설계되었으며 SPSS Modeler 버전 17에 추가된 출력 뷰어에 결과 모델을 표시합니다. 랜덤 트리 노드는 추가 관측값을 예측하거나 분류하는 데 사용하는 의사결정 트리를 생성합니다. 이 방법은 재귀적 파티셔닝을 사용하여 각 단계마다 불순도를 최소화하여 훈련 레코드를 세그먼트로 분할합니다. 여기서 트리의 노드는 노드의 케이스의 100%가 대상 필드의 특정 범주에 속하면 순수로 간주됩니다. 목표 및 입력 필드는 숫자 범위 또는 범주형(명목형, 순서형 또는 플래그)입니다. 모든 분할은 이분형입니다(오직 두 개의 부집단).

표 139. randomtrees 특성

randomtrees 특성	값	특성 설명
target	<i>field</i>	랜덤 트리 노드에서 모델은 단일 목표와 하나 이상의 입력 필드가 필요합니다. 빈도 필드도 지정할 수 있습니다. 자세한 정보는 191 페이지의 『공통 모델링 노드 특성』의 내용을 참조하십시오.
number_of_models	<i>integer</i>	양상불 모델링의 일부로 작성할 모델 수를 판별합니다.
use_number_of_predictors	<i>flag</i>	number_of_predictors의 사용 여부를 판별합니다.
number_of_predictors	<i>integer</i>	분할 모델을 작성할 때 사용할 예측자 수를 지정합니다.
use_stop_rule_for_accuracy	<i>flag</i>	정확도를 향상시킬 수 없을 때 모델 작성 중단 여부를 판별합니다.
sample_size	<i>number</i>	매우 큰 데이터 세트를 처리할 때 성능을 향상시키려면 이 값을 줄이십시오.
handle_imbalanced_data	<i>flag</i>	모델 대상이 특정 플래그 결과이고 원하는 결과에 원하지 않는 결과의 비율이 매우 작으면 데이터의 균형이 맞지 않고 모델이 수행한 붓스트랩 표본추출이 모델의 정확도에 영향을 줄 수 있습니다. 모델이 원하는 결과의 더 많은 부분을 캡처하고 더 강력한 모델을 생성할 수 있도록 불균형한 데이터 처리를 사용하십시오.
use_weighted_sampling	<i>flag</i>	거짓인 경우 각 노드의 변수가 동일한 확률로 임의의 선택됩니다. 참인 경우 변수가 적절하게 가중되고 선택됩니다.
max_node_number	<i>integer</i>	개별 트리에 허용되는 노드의 최대 수입니다. 다음 분할에서 이 수가 초과되면 트리 성장이 정지합니다.
max_depth	<i>integer</i>	성장이 정지될 때까지의 최대 트리 깊이입니다.
min_child_node_size	<i>integer</i>	상위 노드가 분할된 후 하위 노드에 허용되는 레코드의 최소 수를 판별합니다. 하위 노드의 레코드 수가 여기에 지정된 수보다 적으면 상위 노드가 분할되지 않습니다.
use_costs	<i>flag</i>	

표 139. *randomtrees* 특성 (계속)

randomtrees 특성	값	특성 설명
costs	<i>structured</i>	구조화 특성입니다. 형식은 실제 값, 예측값 및 예측이 틀린 경우의 비용인 3 값의 목록입니다. 예를 들어, <code>tree.setPropertyValue("costs", [{"drugA", "drugB", 3.0}, {"drugX", "drugY", 4.0}])</code>
default_cost_increase	none linear square custom	참고: 순서형 목표에만 사용할 수 있습니다. 비용 교차표에서 기본값을 설정하십시오.
max_pct_missing	<i>integer</i>	입력의 결측값에 대한 퍼센트가 여기에 지정된 값보다 크면 입력이 제외됩니다. 최소값 0, 최대값 100.
exclude_single_cat_pct	<i>integer</i>	하나의 범주 값이 여기에 지정된 퍼센트보다 높은 레코드 퍼센트를 나타내면 모델 작성에서 전체 필드가 제외됩니다. 최소값은 1이고 최대값은 99입니다.
max_category_number	<i>integer</i>	필드의 범주 수가 이 값을 초과하면 모델 작성에서 필드가 제외됩니다. 최소값은 2입니다.
min_field_variation	<i>number</i>	연속형 필드의 변동계수가 이 값보다 작으면 모델 작성에서 필드가 제외됩니다.
num_bins	<i>integer</i>	데이터가 연속형 입력으로 구성되는 경우에만 사용합니다. 입력에 사용할 동일한 빈도 구간 수를 설정하십시오. 옵션은 2, 4, 5, 10, 20, 25, 50 또는 100입니다.
topN	<i>integer</i>	보고할 규칙 수를 지정합니다. 기본값은 50이고 최소값은 1이며 최대값은 1000입니다.

regressionnode 특성



선형 회귀는 데이터를 요약통계하고 예측 및 실제 출력 값 사이의 불일치를 최소화하는 직선이나 표면에 적합하게 하여 예측하기 위한 일반적인 통계 기법입니다.

참고: 회귀분석 노드는 차후 릴리스에서 선형 노드로 바뀔 예정입니다. 지금부터 선형 회귀에 대해 선형 모델을 사용할 것을 권장합니다.

예제

```
node = stream.create("regression", "My node")
# "Fields" tab
node.setPropertyValue("custom_fields", True)
node.setPropertyValue("target", "Age")
node.setPropertyValue("inputs", ["Na", "K"])
```

```

node.setPropertyValue("partition", "Test")
node.setPropertyValue("use_weight", True)
node.setPropertyValue("weight_field", "Drug")
# "Model" tab
node.setPropertyValue("use_model_name", True)
node.setPropertyValue("model_name", "Regression Age")
node.setPropertyValue("use_partitioned_data", True)
node.setPropertyValue("method", "Stepwise")
node.setPropertyValue("include_constant", False)
# "Expert" tab
node.setPropertyValue("mode", "Expert")
node.setPropertyValue("complete_records", False)
node.setPropertyValue("tolerance", "1.0E-3")
# "Stepping..." section
node.setPropertyValue("stepping_method", "Probability")
node.setPropertyValue("probability_entry", 0.77)
node.setPropertyValue("probability_removal", 0.88)
node.setPropertyValue("F_value_entry", 7.0)
node.setPropertyValue("F_value_removal", 8.0)
# "Output..." section
node.setPropertyValue("model_fit", True)
node.setPropertyValue("r_squared_change", True)
node.setPropertyValue("selection_criteria", True)
node.setPropertyValue("descriptives", True)
node.setPropertyValue("p_correlations", True)
node.setPropertyValue("collinearity_diagnostics", True)
node.setPropertyValue("confidence_interval", True)
node.setPropertyValue("covariance_matrix", True)
node.setPropertyValue("durbin_watson", True)

```

표 140. regressionnode 특성

regressionnode 특성	값	특성 설명
target	field	회귀 모형은 단일 목표 필드와 하나 이상의 입력 필드가 필요합니다. 가중 필드도 지정할 수 있습니다. 자세한 정보는 191 페이지의 『공통 모델링 노드 특성』 주제를 참조하십시오.
method	Enter Stepwise Backwards Forwards	
include_constant	flag	
use_weight	flag	
weight_field	field	
mode	Simple Expert	
complete_records	flag	

표 140. regressionmode 특성 (계속)

regressionmode 특성	값	특성 설명
tolerance	1.0E-1 1.0E-2 1.0E-3 1.0E-4 1.0E-5 1.0E-6 1.0E-7 1.0E-8 1.0E-9 1.0E-10 1.0E-11 1.0E-12	인수에 큰따옴표를 사용하십시오.
stepping_method	useP useF	useP : F-확률 사용 useF: F-값 사용
probability_entry	number	
probability_removal	number	
F_value_entry	number	
F_value_removal	number	
selection_criteria	flag	
confidence_interval	flag	
covariance_matrix	flag	
collinearity_diagnostics	flag	
regression_coefficients	flag	
exclude_fields	flag	
durbin_watson	flag	
model_fit	flag	
r_squared_change	flag	
p_correlations	flag	
descriptives	flag	
calculate_variable_importance	flag	

sequencenode 특성



순차규칙 노드는 순차 또는 시간 지향 데이터에서 연관 규칙을 발견합니다. 순차규칙은 예측 가능한 순서로 발생하는 경향이 있는 항목 세트 목록입니다. 예를 들어, 면도기와 애프터셰이브 로션을 구매하는 고객은 다음 번 구매 시에 면도용 크림을 구매할 수 있습니다. 순차규칙 노드는 순차규칙을 찾는 데 효율적인 2패스 방법을 사용하는 CARMA 연관 규칙 알고리즘을 기반으로 합니다.

예제

```

node = stream.create("sequence", "My node")
# "Fields" tab
node.setPropertyValue("id_field", "Age")
node.setPropertyValue("contiguous", True)
node.setPropertyValue("use_time_field", True)
node.setPropertyValue("time_field", "Date1")
node.setPropertyValue("content_fields", ["Drug", "BP"])
node.setPropertyValue("partition", "Test")
# "Model" tab
node.setPropertyValue("use_model_name", True)
node.setPropertyValue("model_name", "Sequence_test")
node.setPropertyValue("use_partitioned_data", False)
node.setPropertyValue("min_supp", 15.0)
node.setPropertyValue("min_conf", 14.0)
node.setPropertyValue("max_size", 7)
node.setPropertyValue("max_predictions", 5)
# "Expert" tab
node.setPropertyValue("mode", "Expert")
node.setPropertyValue("use_max_duration", True)
node.setPropertyValue("max_duration", 3.0)
node.setPropertyValue("use_pruning", True)
node.setPropertyValue("pruning_value", 4.0)
node.setPropertyValue("set_mem_sequences", True)
node.setPropertyValue("mem_sequences", 5.0)
node.setPropertyValue("use_gaps", True)
node.setPropertyValue("min_item_gap", 20.0)
node.setPropertyValue("max_item_gap", 30.0)

```

표 141. sequencenode 특성

sequencenode 특성	값	특성 설명
id_field	field	시퀀스 모델을 작성하려면 ID 필드, 선택적 시간 필드 및 하나 이상의 내용 필드를 지정해야 합니다. 가중치 및 빈도 필드는 사용하지 않습니다. 자세한 정보는 191 페이지의 『공통 모델링 노드 특성』 주제를 참조하십시오.
time_field	field	
use_time_field	flag	
content_fields	[field1 ... fieldn]	
contiguous	flag	
min_supp	number	
min_conf	number	
max_size	number	
max_predictions	number	
mode	Simple Expert	
use_max_duration	flag	
max_duration	number	
use_gaps	flag	
min_item_gap	number	

표 141. sequencenode 특성 (계속)

sequencenode 특성	값	특성 설명
max_item_gap	number	
use_pruning	flag	
pruning_value	number	
set_mem_sequences	flag	
mem_sequences	integer	

slrmnode 특성



SLRM(Self-Learning Response Model) 노드를 사용하면 하나의 새 케이스 또는 소수의 새 케이스를 사용하여 모든 데이터를 사용하는 모델을 재교육할 필요 없이 모델을 재평가할 수 있는 모델을 작성할 수 있습니다.

예제

```
node = stream.create("slrm", "My node")
node.setPropertyValue("target", "Offer")
node.setPropertyValue("target_response", "Response")
node.setPropertyValue("inputs", ["Cust_ID", "Age", "Ave_Bal"])
```

표 142. slrmnode 특성

slrmnode 특성	값	특성 설명
target	field	목표 필드는 명목형 또는 플래그 필드여야 합니다. 빈도 필드도 지정할 수 있습니다. 자세한 정보는 191 페이지의 『공통 모델링 노드 특성』 주제를 참조하십시오.
target_response	field	유형이 플래그여야 합니다.
continue_training_existing_model	flag	
target_field_values	flag	Use all: 소스의 모든 값을 사용합니다. Specify: 필요한 값을 선택하십시오.
target_field_values_specify	[field1 ... fieldN]	
include_model_assessment	flag	
model_assessment_random_seed	number	실수여야 합니다.
model_assessment_sample_size	number	실수여야 합니다.
model_assessment_iterations	number	반복 횟수
display_model_evaluation	flag	
max_predictions	number	
randomization	number	
scoring_random_seed	number	
sort	Ascending Descending	최고 또는 최저 스코어를 갖는 제안이 처음 표시되는지 여부를 지정합니다.
model_reliability	flag	

표 142. *slrmnode* 특성 (계속)

<i>slrmnode</i> 특성	값	특성 설명
calculate_variable_importance	<i>flag</i>	

statisticsmodelnode 특성



통계량 모델 노드를 사용하면 PMML을 생성하는 IBM SPSS Statistics 프로시저를 실행하여 데이터를 분석하고 작업할 수 있습니다. 이 노드는 IBM SPSS Statistics의 사용권 사본이 필요합니다.

이 노드의 특성은 350 페이지의 『statisticsmodelnode 특성』에서 설명됩니다.

stpnode 특성



STP(Spatio-Temporal Prediction) 노드는 위치 데이터, 예측(예측자)을 위한 입력 필드, 시간 필드 및 목표 필드를 포함하는 데이터를 사용합니다. 각 위치에는 각 측정 시간에 각 예측변수의 값을 나타내는 데이터에 여러 행이 있습니다. 데이터가 분석된 후에는 분석에 사용된 모양 데이터 내에서 어떤 위치에서든 목표 값을 예측하는 데 사용할 수 있습니다.

표 143. *stpnode* 특성

<i>stpnode</i> 특성	데이터 유형	특성 설명
필드 탭		
target	<i>field</i>	이것은 목표 필드입니다.
location	<i>field</i>	모델의 위치 필드입니다. 지리공간적 필드만 허용됩니다.
location_label	<i>field</i>	location에서 선택된 위치를 레이블하기 위해 출력에서 사용하는 범주형 필드
time_field	<i>field</i>	모델의 시간 필드입니다. 연속형 측정을 갖는 필드만 허용되며, 저장 유형은 시간, 날짜, 시간소인 또는 정수여야 합니다.
inputs	[<i>field1 ... fieldN</i>]	입력 필드의 목록
시간 구간 탭		
interval_type_timestamp	Years Quarters Months Weeks Days Hours Minutes Seconds	

표 143. *stpnode* 특성 (계속)

stpnode 특성	데이터 유형	특성 설명
interval_type_date	Years Quarters Months Weeks Days	
interval_type_time	Hours Minutes Seconds	STP가 계산에 사용하는 시간 지수를 작성할 때 고려되는 주당 일을 제한합니다.
interval_type_integer	Periods (시간 지수 필드만, 정수 저장 공간)	데이터 세트가 변환될 정수입니다. 사용 가능한 선택은 모델에 대해 <code>time_field</code> 로 선택되는 필드의 저장 유형에 따라 다릅니다.
period_start	<i>integer</i>	
start_month	January February March April May June July August September October November December	모델이 지수화하기 시작하는 월입니다(예를 들어, March로 설정되지만 데이터 세트의 첫 번째 레코드가 January로 설정되는 경우, 모델은 처음 두 레코드를 건너뛰고 3월에 지수화를 시작함).
week_begins_on	Sunday Monday Tuesday Wednesday Thursday Friday Saturday	데이터로부터 STP에 의해 작성되는 시간 지수의 시작점입니다.
days_per_week	<i>integer</i>	1의 증분으로 최소 1, 최대 7
hours_per_day	<i>integer</i>	모델이 하루에 대해 고려하는 시간입니다. 10으로 설정되는 경우 모델은 <code>day_begins_at</code> 시간에 지수화를 시작하고 10시간 동안 지수화를 계속한 후, <code>day_begins_at</code> 값과 매치하는 다음 값으로 건너웁니다.

표 143. *stpnode* 특성 (계속)

stpnode 특성	데이터 유형	특성 설명
day_begins_at	00:00 01:00 02:00 03:00 ... 23:00	모델이 지수화를 시작하는 시간 값을 설정합니다.
interval_increment	1 2 3 4 5 6 10 12 15 20 30	이 증분 설정은 분 또는 초를 위한 것입니다. 이것은 모델이 데이터로부터 지수를 작성하는지 여부를 판별합니다. 그러므로 증분이 30이고 구간 유형이 seconds일 때 모델은 30초마다 데이터로부터 지수를 작성합니다.
data_matches_interval	부울	N으로 설정되는 경우, 일반 interval_type으로의 데이터 변환이 모델이 작성되기 전에 발생합니다. 데이터가 이미 올바른 형식이고 interval_type 및 모든 연관된 속성이 사용자 데이터와 매치하는 경우, 이것을 Y로 설정하여 데이터의 변환이나 통합을 막으십시오. 이것을 Y로 설정하면 모든 통합 제어를 사용하지 않습니다.
agg_range_default	Sum Mean Min Max Median 1stQuartile 3rdQuartile	이것은 연속형 필드에 사용하는 기본 통합 방법을 판별합니다. 사용자 정의 통합에서 특별히 포함되지 않는 모든 연속형 필드는 여기에서 지정되는 방법을 사용하여 통합됩니다.
custom_agg	[[field, aggregation method],[..] 데모: [['x5' 'FirstQuartile']]['x4' 'Sum']]	구조화된 특성: 스크립트 모수: custom_agg 예를 들어, set :stpnode.custom_agg = [[field1 function] [field2 function]] 여기서 function은 해당 필드에 사용할 통합 함수입니다.

표 143. *stpnode* 특성 (계속)

stpnode 특성	데이터 유형	특성 설명
기본 탭		
include_intercept	<i>flag</i>	
max_autoregressive_lag	<i>integer</i>	1의 증분으로 최소 1, 최대 5입니다. 이것은 예측을 위해 필요한 이전 레코드 수입니다. 따라서 예를 들어 5로 설정되면 이전 5개 레코드를 사용하여 새 예측을 작성합니다. 작성 데이터로부터 여기에서 지정되는 레코드 수가 모델에 통합되므로 사용자는 모델을 스코어링할 때 데이터를 다시 제공할 필요가 없습니다.
estimation_method	Parametric Nonparametric	지리공간적 공분산 교차표 모델링 방법입니다.
parametric_model	Gaussian Exponential PoweredExponential	Parametric 지리공간적 공분산 모델의 이전 변수
exponential_power	<i>number</i>	PoweredExponential 모델의 거듭제곱 수준입니다. 최소 1, 최대 2.
고급 탭		
max_missing_values	<i>integer</i>	모델에서 허용되는 결측값을 갖는 레코드의 최대 퍼센트입니다.
significance	<i>number</i>	모델 작성에서 가설 검정을 위한 유의 수준입니다. 두 개의 적합도 검정, 유효 F-검정 및 계수 t-검정을 포함하여 STP 모델 추정의 모든 검정을 위한 유의성 값을 지정합니다.
출력 탭		
model_specifications	<i>flag</i>	
temporal_summary	<i>flag</i>	
location_summary	<i>flag</i>	위치 요약 테이블이 모델 출력에 포함되는지 여부를 판별합니다.
model_quality	<i>flag</i>	
test_mean_structure	<i>flag</i>	
mean_structure_coefficients	<i>flag</i>	
autoregressive_coefficients	<i>flag</i>	
test_decay_space	<i>flag</i>	
parametric_spatial_covariance	<i>flag</i>	
correlations_heat_map	<i>flag</i>	
correlations_map	<i>flag</i>	
location_clusters	<i>flag</i>	
similarity_threshold	<i>number</i>	출력 군집이 단일 군집으로 병합되기에 충분히 유사하다고 간주되는 임계값입니다.
max_number_clusters	<i>integer</i>	모델 출력에 포함될 수 있는 군집 수에 대한 상한입니다.

표 143. *stpnode* 특성 (계속)

stpnode 특성	데이터 유형	특성 설명
모델 옵션 탭		
use_model_name	flag	
model_name	string	
uncertainty_factor	number	최소 0, 최대 100. 미래의 예측에 적용되는 불확실성(오류)의 증가를 판별합니다. 예측에 대한 상한 및 하한입니다.

svmnode 특성



지원 벡터 머신(SVM) 노드를 사용하면 데이터를 과적합 없이 두 개의 그룹 중 하나로 분류할 수 있습니다. SVM은 다수의 입력 필드가 있는 데이터 세트 등과 같은 광범위한 데이터 세트와 잘 작동합니다.

예제

```
node = stream.create("svm", "My node")
# Expert tab
node.setPropertyValue("mode", "Expert")
node.setPropertyValue("all_probabilities", True)
node.setPropertyValue("kernel", "Polynomial")
node.setPropertyValue("gamma", 1.5)
```

표 144. *svmnode* 특성.

svmnode 특성	값	특성 설명
all_probabilities	flag	
stopping_criteria	1.0E-1 1.0E-2 1.0E-3(기본값) 1.0E-4 1.0E-5 1.0E-6	최적화 알고리즘의 중지 시점을 판별합니다.
regularization	number	C 모수라고도 합니다.
precision	number	목표 필드의 측정 수준이 Continuous인 경우에만 사용합니다.
kernel	RBF(기본값) Polynomial Sigmoid Linear	변환에 사용하는 커널 함수의 유형입니다.
rbf_gamma	number	kernel이 RBF인 경우에만 사용합니다.
gamma	number	kernel이 Polynomial 또는 Sigmoid인 경우에만 사용합니다.
bias	number	

표 144. *svmnnode* 특성 (계속).

svmnnode 특성	값	특성 설명
degree	<i>number</i>	kernel이 Polynomial인 경우에만 사용됩니다.
calculate_variable_importance	<i>flag</i>	
calculate_raw_propensities	<i>flag</i>	
calculate_adjusted_propensities	<i>flag</i>	
adjusted_propensity_partition	Test Validation	

tcmnode 특성



시간 인과 모델링은 시계열 데이터에서 핵심 인과 관계를 검색하려고 시도합니다. 시간 인과 모델링에서 목표 계열 세트 및 해당 목표에 대한 후보 입력 세트를 지정합니다. 그런 다음 프로시저가 각 목표에 대한 자기회귀 시계열 모델을 작성하고 목표와 가장 중요한 인과 관계를 갖는 입력만 포함합니다.

표 145. *tcmnode* 특성

tcmnode 특성	값	특성 설명
custom_fields	부울	
dimensionlist	[<i>dimension1 ... dimensionN</i>]	
data_struct	Multiple Single	
metric_fields	<i>fields</i>	
both_target_and_input	[<i>f1 ... fN</i>]	
targets	[<i>f1 ... fN</i>]	
candidate_inputs	[<i>f1 ... fN</i>]	
forced_inputs	[<i>f1 ... fN</i>]	
use_timestamp	Timestamp Period	

표 145. tcmnode 특성 (계속)

tcmnode 특성	값	특성 설명
input_interval	None Unknown Year Quarter Month Week Day Hour Hour_nonperiod Minute Minute_nonperiod Second Second_nonperiod	
period_field	<i>string</i>	
period_start_value	<i>integer</i>	
num_days_per_week	<i>integer</i>	
start_day_of_week	Sunday Monday Tuesday Wednesday Thursday Friday Saturday	
num_hours_per_day	<i>integer</i>	
start_hour_of_day	<i>integer</i>	
timestamp_increments	<i>integer</i>	
cyclic_increments	<i>integer</i>	
cyclic_periods	<i>list</i>	
output_interval	None Year Quarter Month Week Day Hour Minute Second	
is_same_interval	Same Notsame	
cross_hour	부울	
aggregate_and_distribute	<i>list</i>	

표 145. *tcmnode* 특성 (계속)

tcmnode 특성	값	특성 설명
aggregate_default	Mean Sum Mode Min Max	
distribute_default	Mean Sum	
group_default	Mean Sum Mode Min Max	
missing_imput	Linear_interp Series_mean K_mean K_meridian Linear_trend None	
k_mean_param	<i>integer</i>	
k_median_param	<i>integer</i>	
missing_value_threshold	<i>integer</i>	
conf_level	<i>integer</i>	
max_num_predictor	<i>integer</i>	
max_lag	<i>integer</i>	
epsilon	<i>number</i>	
threshold	<i>integer</i>	
is_re_est	부울	
num_targets	<i>integer</i>	
percent_targets	<i>integer</i>	
fields_display	<i>list</i>	
series_display	<i>list</i>	
network_graph_for_target	부울	
sign_level_for_target	<i>number</i>	
fit_and_outlier_for_target	부울	
sum_and_para_for_target	부울	
impact_diag_for_target	부울	
impact_diag_type_for_target	Effect Cause Both	
impact_diag_level_for_target	<i>integer</i>	
series_plot_for_target	부울	

표 145. tcmnode 특성 (계속)

tcmnode 특성	값	특성 설명
res_plot_for_target	부울	
top_input_for_target	부울	
forecast_table_for_target	부울	
same_as_for_target	부울	
network_graph_for_series	부울	
sign_level_for_series	<i>number</i>	
fit_and_outlier_for_series	부울	
sum_and_para_for_series	부울	
impact_diagram_for_series	부울	
impact_diagram_type_for_series	Effect Cause Both	
impact_diagram_level_for_series	<i>integer</i>	
series_plot_for_series	부울	
residual_plot_for_series	부울	
forecast_table_for_series	부울	
outlier_root_cause_analysis	부울	
causal_levels	<i>integer</i>	
outlier_table	Interactive Pivot Both	
rmsp_error	부울	
bic	부울	
r_square	부울	
outliers_over_time	부울	
series_transormation	부울	
use_estimation_period	부울	
estimation_period	Times Observation	
observations	<i>list</i>	
observations_type	Latest Earliest	
observations_num	<i>integer</i>	
observations_exclude	<i>integer</i>	
extend_records_into_future	부울	
forecastperiods	<i>integer</i>	
max_num_distinct_values	<i>integer</i>	
display_targets	FIXEDNUMBER PERCENTAGE	

표 145. tcmnode 특성 (계속)

tcmnode 특성	값	특성 설명
goodness_fit_measure	ROOTMEAN BIC RSQUARE	
top_input_for_series	부울	
aic	부울	
rmse	부울	

ts 특성



시계열 노드는 시계열 데이터에 대한 지수평활, 일변량 자기회귀 통합 이동 평균(ARIMA), 다변량 ARIMA(또는 전이 함수) 모델을 추정하고 미래 성능을 위한 예측값을 생성합니다. 이 시계열 노드는 SPSS Modeler 버전 18에서 더 이상 사용되지 않는 이전의 시계열 노드와 유사합니다. 그러나 이 새 시계열 노드는 IBM SPSS Analytic Server의 기능을 이용하여 빅 데이터를 처리해서 SPSS Modeler 버전 17에 추가된 출력 뷰어에 결과 모델을 표시하도록 설계되었습니다.

표 146. ts 특성

ts 특성	값	특성 설명
targets	<i>field</i>	시계열 노드는 선택적으로 하나 이상의 입력 필드를 예측자로 사용하여 하나 이상의 목표를 예측합니다. 빈도 및 가중 필드는 사용하지 않습니다. 자세한 정보는 191 페이지의 『공통 모델링 노드 특성』의 내용을 참조하십시오.
candidate_inputs	[<i>field1 ... fieldN</i>]	모델에서 사용하는 입력 또는 예측변수 필드입니다.
use_period	<i>flag</i>	
date_time_field	<i>field</i>	
input_interval	None Unknown Year Quarter Month Week Day Hour Hour_nonperiod Minute Minute_nonperiod Second Second_nonperiod	

표 146. ts 특성 (계속)

ts 특성	값	특성 설명
period_field	<i>field</i>	
period_start_value	<i>integer</i>	
num_days_per_week	<i>integer</i>	
start_day_of_week	Sunday Monday Tuesday Wednesday Thursday Friday Saturday	
num_hours_per_day	<i>integer</i>	
start_hour_of_day	<i>integer</i>	
timestamp_increments	<i>integer</i>	
cyclic_increments	<i>integer</i>	
cyclic_periods	<i>list</i>	
output_interval	None Year Quarter Month Week Day Hour Minute Second	
is_same_interval	<i>flag</i>	
cross_hour	<i>flag</i>	
aggregate_and_distribute	<i>list</i>	
aggregate_default	Mean Sum Mode Min Max	
distribute_default	Mean Sum	
group_default	Mean Sum Mode Min Max	

표 146. ts 특성 (계속)

ts 특성	값	특성 설명
missing_imput	Linear_interp Series_mean K_mean K_median Linear_trend	
k_span_points	<i>integer</i>	
use_estimation_period	<i>flag</i>	
estimation_period	Observations Times	
date_estimation	<i>list</i>	date_time_field를 사용하는 경우에만 사용할 수 있습니다.
period_estimation	<i>list</i>	use_period를 사용하는 경우에만 사용할 수 있습니다.
observations_type	Latest Earliest	
observations_num	<i>integer</i>	
observations_exclude	<i>integer</i>	
method	ExpertModeler Exsmooth Arima	
expert_modeler_method	ExpertModeler Exsmooth Arima	
consider_seasonal	<i>flag</i>	
detect_outliers	<i>flag</i>	
expert_outlier_additive	<i>flag</i>	
expert_outlier_level_shift	<i>flag</i>	
expert_outlier_innovational	<i>flag</i>	
expert_outlier_level_shift	<i>flag</i>	
expert_outlier_transient	<i>flag</i>	
expert_outlier_seasonal_additive	<i>flag</i>	
expert_outlier_local_trend	<i>flag</i>	
expert_outlier_additive_patch	<i>flag</i>	
consider_newesmodels	<i>flag</i>	

표 146. ts 특성 (계속)

ts 특성	값	특성 설명
exsmooth_model_type	Simple HoltLinearTrend BrownsLinearTrend DampedTrend SimpleSeasonal WintersAdditive WintersMultiplicative DampedTrendAdditive DampedTrendMultiplicative MultiplicativeTrendAdditive MultiplicativeSeasonal MultiplicativeTrend Multiplicative MultiplicativeTrend	지수평활 방법을 지정합니다. 기본값은 Simple입니다.
futureValue_type_method	Compute specify	
exsmooth_transformation_type	None SquareRoot NaturalLog	
arma.p	<i>integer</i>	
arma.d	<i>integer</i>	
arma.q	<i>integer</i>	
arma.sp	<i>integer</i>	
arma.sd	<i>integer</i>	
arma.sq	<i>integer</i>	
arma_transformation_type	None SquareRoot NaturalLog	
arma_include_constant	<i>flag</i>	
tf_arma.p. <i>fieldname</i>	<i>integer</i>	전이 함수용입니다.
tf_arma.d. <i>fieldname</i>	<i>integer</i>	전이 함수용입니다.
tf_arma.q. <i>fieldname</i>	<i>integer</i>	전이 함수용입니다.
tf_arma.sp. <i>fieldname</i>	<i>integer</i>	전이 함수용입니다.
tf_arma.sd. <i>fieldname</i>	<i>integer</i>	전이 함수용입니다.
tf_arma.sq. <i>fieldname</i>	<i>integer</i>	전이 함수용입니다.
tf_arma.delay. <i>fieldname</i>	<i>integer</i>	전이 함수용입니다.
tf_arma.transformation_type. <i>fieldname</i>	None SquareRoot NaturalLog	전이 함수용입니다.
arma_detect_outliers	<i>flag</i>	
arma_outlier_additive	<i>flag</i>	
arma_outlier_level_shift	<i>flag</i>	

표 146. ts 특성 (계속)

ts 특성	값	특성 설명
arima_outlier_innovational	flag	
arima_outlier_transient	flag	
arima_outlier_seasonal_additive	flag	
arima_outlier_local_trend	flag	
arima_outlier_additive_patch	flag	
max_lags	integer	
cal_PI	flag	
conf_limit_pct	real	
events	fields	
continue	flag	
scoring_model_only	flag	매우 많은 시계열 수(수만 개)가 있는 모델에 사용됩니다.
forecastperiods	integer	
extend_records_into_future	flag	
extend_metric_values	fields	예측변수의 미래 값을 제공할 수 있습니다.
conf_limits	flag	
noise_res	flag	
max_models_output	integer	출력에 표시할 모델 수를 제어합니다. 기본값은 10입니다. 작성된 총 모델 수가 이 값을 초과할 경우 모델이 출력에 표시되지 않습니다. 여전히 모델을 스코어링에 사용할 수 있습니다.

timeseriesnode 특성(더 이상 사용되지 않음)



참고: 이 원본 시계열 노드는 SPSS Modeler 버전 18에서 더 이상 사용되지 않으며 IBM SPSS Analytic Server의 기능을 이용하여 빅 데이터를 처리하도록 디자인된 새 시계열 노드로 대체되었습니다. 시계열 노드는 시계열 데이터에 대한 지수평활, 일변량 자기회귀 통합 이동 평균(ARIMA), 다변량 ARIMA(또는 전이 함수) 모델을 추정하고 미래 성능을 위한 예측값을 생성합니다. 시계열 노드 앞에는 항상 시간 구간 노드가 와야 합니다.

예제

```
node = stream.create("timeseries", "My node")
node.setPropertyValue("method", "Exsmooth")
node.setPropertyValue("exsmooth_model_type", "HoltsLinearTrend")
node.setPropertyValue("exsmooth_transformation_type", "None")
```

표 147. timeseriesnode 특성

timeseriesnode 특성	값	특성 설명
targets	field	시계열 노드는 선택적으로 하나 이상의 입력 필드를 예측자로 사용하여 하나 이상의 목표를 예측합니다. 빈도 및 가중 필드는 사용하지 않습니다. 자세한 정보는 191 페이지의 『공통 모델링 노드 특성』 주제를 참조하십시오.
continue	flag	
method	ExpertModeler Exsmooth Arima Reuse	
expert_modeler_method	flag	
consider_seasonal	flag	
detect_outliers	flag	
expert_outlier_additive	flag	
expert_outlier_level_shift	flag	
expert_outlier_innovational	flag	
expert_outlier_level_shift	flag	
expert_outlier_transient	flag	
expert_outlier_seasonal_additive	flag	
expert_outlier_local_trend	flag	
expert_outlier_additive_patch	flag	
exsmooth_model_type	Simple HoltsLinearTrend BrownsLinearTrend DampedTrend SimpleSeasonal WintersAdditive WintersMultiplicative	
exsmooth_transformation_type	None SquareRoot NaturalLog	
arima_p	integer	
arima_d	integer	
arima_q	integer	
arima_sp	integer	
arima_sd	integer	
arima_sq	integer	

표 147. timeseriesnode 특성 (계속)

timeseriesnode 특성	값	특성 설명
arima_transformation_type	None SquareRoot NaturalLog	
arima_include_constant	flag	
tf_arima_p. fieldname	integer	전이 함수용입니다.
tf_arima_d. fieldname	integer	전이 함수용입니다.
tf_arima_q. fieldname	integer	전이 함수용입니다.
tf_arima_sp. fieldname	integer	전이 함수용입니다.
tf_arima_sd. fieldname	integer	전이 함수용입니다.
tf_arima_sq. fieldname	integer	전이 함수용입니다.
tf_arima_delay. fieldname	integer	전이 함수용입니다.
tf_arima_transformation_type. fieldname	None SquareRoot NaturalLog	전이 함수용입니다.
arima_detect_outlier_mode	None Automatic	
arima_outlier_additive	flag	
arima_outlier_level_shift	flag	
arima_outlier_innovational	flag	
arima_outlier_transient	flag	
arima_outlier_seasonal_additive	flag	
arima_outlier_local_trend	flag	
arima_outlier_additive_patch	flag	
conf_limit_pct	real	
max_lags	integer	
events	fields	
scoring_model_only	flag	매우 많은 시계열 수(수만 개)가 있는 모델에 사용됩니다.

treesas 특성



Tree-AS 노드는 기존 CHAID 노드와 유사하지만, 빅 데이터를 처리하여 단일 트리를 작성하도록 설계되었으며 SPSS Modeler 버전 17에 추가된 출력 뷰어에 결과 모델을 표시합니다. 이 노드는 최적 분할을 식별하기 위해 카이제곱 통계량(CHAID)을 사용하여 의사결정 트리를 생성합니다. 이 CHAID의 사용은 일부 분할이 셋 이상의 분기를 가짐을 의미하는 비2진 트리를 생성할 수 있습니다. 목표 및 입력 필드는 숫자 범위(연속형) 또는 범주형입니다. Exhaustive CHAID는 가능한 모든 분할을 탐색하는 보다 철저한 작업을 수행하지만 계산하는 데 시간이 더 걸리는 변형 CHAID입니다.

표 148. *treeas* 특성

treeas 특성	값	특성 설명
target	<i>field</i>	트리-AS 노드에서 CHAID 모델은 단일 목표와 하나 이상의 입력 필드가 필요합니다. 빈도 필드도 지정할 수 있습니다. 자세한 정보는 191 페이지의 『공통 모델링 노드 특성』의 내용을 참조하십시오.
method	chaid exhaustive_chaid	
max_depth	<i>integer</i>	0에서부터 20까지의 최대 트리 깊이입니다. 기본값은 5입니다.
num_bins	<i>integer</i>	데이터가 연속형 입력으로 구성되는 경우에만 사용합니다. 입력에 사용할 동일한 빈도 구간 수를 설정하십시오. 옵션은 2, 4, 5, 10, 20, 25, 50 또는 100입니다.
record_threshold	<i>integer</i>	모델이 트리 작성 중에 p-값 사용에서 유효 크기로 전환하는 레코드 수입니다. 기본값은 1,000,000이며, 10,000의 증분으로 이 값을 늘리거나 줄이십시오.
split_alpha	<i>number</i>	분할 유의수준입니다. 값은 0.01과 0.99 사이에 있어야 합니다.
merge_alpha	<i>number</i>	병합 유의수준입니다. 값은 0.01과 0.99 사이에 있어야 합니다.
bonferroni_adjustment	<i>flag</i>	Bonferroni 방법을 사용하여 유의성 값을 조정합니다.
effect_size_threshold_cont	<i>number</i>	노드를 분할할 때 유효 크기 임계값 및 연속형 목표를 사용할 때 병합 범주를 설정하십시오. 값은 0.01과 0.99 사이에 있어야 합니다.
effect_size_threshold_cat	<i>number</i>	노드를 분할할 때 유효 크기 임계값 및 범주형 목표를 사용할 때 병합 범주를 설정하십시오. 값은 0.01과 0.99 사이에 있어야 합니다.
split_merged_categories	<i>flag</i>	병합된 범주의 재분할을 허용합니다.
grouping_sig_level	<i>number</i>	노드 집단이 형성되는 방법 또는 비정상적인 노드를 식별하는 방법을 판별하는 데 사용합니다.
chi_square	pearson likelihood_ratio	카이제곱 통계량을 계산하는 데 사용하는 방법 (Pearson 또는 우도비(Likelihood Ratio))입니다.
minimum_record_use	use_percentage use_absolute	
min_parent_records_pc	<i>number</i>	기본값은 2입니다. 1의 증분으로 최소 1, 최대 100입니다. 상위 분기 값이 하위 분기보다 높아야 합니다.
min_child_records_pc	<i>number</i>	기본값은 1입니다. 1의 증분으로 최소 1, 최대 100입니다.
min_parent_records_abs	<i>number</i>	기본값은 100입니다. 1의 증분으로 최소 1, 최대 100입니다. 상위 분기 값이 하위 분기보다 높아야 합니다.

표 148. *treeas* 특성 (계속)

treeas 특성	값	특성 설명
min_child_records_abs	<i>number</i>	기본값은 50입니다. 1의 증분으로 최소 1, 최대 100입니다.
epsilon	<i>number</i>	셀 기대빈도의 최소 변화입니다.
max_iterations	<i>number</i>	수렴을 위한 최대 반복입니다.
use_costs	<i>flag</i>	
costs	<i>structured</i>	구조화 특성입니다. 형식은 실제 값, 예측값 및 예측이 틀린 경우의 비용인 3 값의 목록입니다. 예를 들어, <code>tree.setPropertyValue("costs", [{"drugA", "drugB", 3.0}, {"drugX", "drugY", 4.0}])</code>
default_cost_increase	none linear square custom	참고: 순서형 목표에만 사용할 수 있습니다. 비용 교차표에서 기본값을 설정하십시오.
calculate_conf	<i>flag</i>	
display_rule_id	<i>flag</i>	각 레코드가 지정되는 터미널 노드에 대한 ID를 표시하는 스코어링 출력에 필드를 추가합니다.

twostepnode 특성



이단계 노드는 2단계 군집방법을 사용합니다. 첫 번째 단계는 원시 입력 데이터를 관리 가능한 하위 군집 세트로 압축하기 위해 데이터를 통한 단일 전달을 수행합니다. 두 번째 단계는 계층적 군집 방법을 사용하여 하위 군집을 점점 더 큰 군집으로 계속해서 병합하는 것입니다. 이단계는 훈련 데이터에 대한 최적 군집 수를 자동으로 평가하는 장점이 있습니다. 혼합 필드 유형과 대형 데이터 세트를 효율적으로 처리할 수 있습니다.

예제

```
node = stream.create("twostep", "My node")
node.setPropertyValue("custom_fields", True)
node.setPropertyValue("inputs", ["Age", "K", "Na", "BP"])
node.setPropertyValue("partition", "Test")
node.setPropertyValue("use_model_name", False)
node.setPropertyValue("model_name", "TwoStep_Drug")
node.setPropertyValue("use_partitioned_data", True)
node.setPropertyValue("exclude_outliers", True)
node.setPropertyValue("cluster_label", "String")
node.setPropertyValue("label_prefix", "TwoStep_")
node.setPropertyValue("cluster_num_auto", False)
node.setPropertyValue("max_num_clusters", 9)
node.setPropertyValue("min_num_clusters", 3)
node.setPropertyValue("num_clusters", 7)
```

표 149. *twostepnode* 특성

twostepnode 특성	값	특성 설명
inputs	[<i>field1</i> ... <i>fieldN</i>]	이단계 모델은 입력 필드의 목록을 사용하지만 목표는 없습니다. 가중치 및 빈도 필드는 인식되지 않습니다. 자세한 정보는 191 페이지의 『공통 모델링 노드 특성』 주제를 참조하십시오.
standardize	<i>flag</i>	
exclude_outliers	<i>flag</i>	
percentage	<i>number</i>	
cluster_num_auto	<i>flag</i>	
min_num_clusters	<i>number</i>	
max_num_clusters	<i>number</i>	
num_clusters	<i>number</i>	
cluster_label	String Number	
label_prefix	<i>string</i>	
distance_measure	Euclidean Loglikelihood	
clustering_criterion	AIC BIC	

twostepAS 특성



이단계 군집은 명확하지 않은 데이터 세트 안에서 자연적 집단(또는 군집)을 드러내도록 디자인된 탐색 도구입니다. 이 프로시저가 채택하는 알고리즘은 범주형 및 연속형 변수의 처리, 군집 수의 자동 선택 및 확장성 같은 전통적인 군집 기법과 차별화하는 여러 가지 바람직한 기능을 갖고 있습니다.

표 150. *twostepAS* 특성

twostepAS 특성	값	특성 설명
inputs	[<i>f1</i> ... <i>fN</i>]	TwoStepAS 모델은 입력 필드의 목록을 사용하지만 목표는 없습니다. 가중치 및 빈도 필드는 인식되지 않습니다.
use_predefined_roles	Boolean	기본값=True
use_custom_field_assignments	Boolean	기본값=False
cluster_num_auto	Boolean	기본값=True
min_num_clusters	integer	기본값=2
max_num_clusters	integer	기본값=15
num_clusters	integer	기본값=5

표 150. twostepAS 특성 (계속)

twostepAS 특성	값	특성 설명
clustering_criterion	AIC BIC	
automatic_clustering_method	use_clustering_criterion_setting Distance_jump Minimum Maximum	
feature_importance_method	use_clustering_criterion_setting effect_size	
use_random_seed	Boolean	
random_seed	integer	
distance_measure	Euclidean Loglikelihood	
include_outlier_clusters	Boolean	기본값=True
num_cases_in_feature_tree_leaf_is_less_than	integer	기본값=10
top_perc_outliers	integer	기본값=5
initial_dist_change_threshold	integer	기본값=0
leaf_node_maximum_branches	integer	기본값=8
non_leaf_node_maximum_branches	integer	기본값=8
max_tree_depth	integer	기본값=3
adjustment_weight_on_measurement_level	integer	기본값=6
memory_allocation_mb	number	기본값=512
delayed_split	Boolean	기본값=True
fields_to_standardize	[f1 ... fN]	
adaptive_feature_selection	Boolean	기본값=True
featureMisPercent	integer	기본값=70
coefRange	number	기본값=0.05
percCasesSingleCategory	integer	기본값=95
numCases	integer	기본값=24
include_model_specifications	Boolean	기본값=True
include_record_summary	Boolean	기본값=True
include_field_transformations	Boolean	기본값=True
excluded_inputs	Boolean	기본값=True
evaluate_model_quality	Boolean	기본값=True
show_feature_importance_bar_chart	Boolean	기본값=True
show_feature_importance_word_cloud	Boolean	기본값=True
show_outlier_clusters_interactive_table_and_chart	Boolean	기본값=True
show_outlier_clusters_pivot_table	Boolean	기본값=True
across_cluster_feature_importance	Boolean	기본값=True
across_cluster_profiles_pivot_table	Boolean	기본값=True
withinprofiles	Boolean	기본값=True

표 150. *twostepAS* 특성 (계속)

twostepAS 특성	값	특성 설명
cluster_distances	Boolean	기본값=True
cluster_label	String Number	
label_prefix	String	

제 14 장 모델 너깃 노드 특성

모델 너깃 노드는 다른 노드와 동일한 공통 특성을 공유합니다. 자세한 정보는 78 페이지의 『공통 노드 특성』의 내용을 참조하십시오.

applyanomalydetectionnode 특성

이상 항목 발견 모델링 노드를 사용하면 이상 항목 발견 모델 너깃을 생성할 수 있습니다. 이 모델 너깃의 스크립팅 이름은 *applyanomalydetectionnode*입니다. 모델링 노드 자체의 스크립팅에 대한 자세한 정보는 191 페이지의 『anomalydetectionnode 특성』을 참조하십시오.

표 151. *applyanomalydetectionnode* 특성.

applyanomalydetectionnode 특성	값	특성 설명
anomaly_score_method	FlagAndScore FlagOnly ScoreOnly	스코어링을 위해 작성되는 출력을 판별합니다.
num_fields	<i>integer</i>	보고할 필드입니다.
discard_records	<i>flag</i>	레코드가 출력에서 삭제되는지 여부를 표시합니다.
discard_anomalous_records	<i>flag</i>	이상 항목 또는 비이상 항목 레코드를 삭제할지 여부의 표시기입니다. 기본값은 off로, 비이상 항목 레코드가 삭제됨을 의미합니다. 그렇지 않고 on인 경우 이상 항목 레코드가 삭제됩니다. 이 특성은 discard_records 특성이 사용 가능한 경우에만 사용 가능합니다.

applyapriorinode 특성

Apriori 모델링 노드를 사용하여 Apriori 모델 너깃을 생성할 수 있습니다. 이 모델 너깃의 스크립팅 이름은 *applyapriorinode*입니다. 모델링 노드 자체의 스크립팅에 대한 자세한 정보는 193 페이지의 『apriorinode 특성』을 참조하십시오.

표 152. *applyapriorinode* 특성.

applyapriorinode 특성	값	특성 설명
max_predictions	<i>number</i> (정수)	
ignore_unmattached	<i>flag</i>	
allow_repeats	<i>flag</i>	
check_basket	NoPredictions Predictions NoCheck	

표 152. *applypriorinode* 특성 (계속).

applypriorinode 특성	값	특성 설명
criterion	Confidence Support RuleSupport Lift Deployability	

applyassociationrulesnode 특성

연관 규칙 모델링 노드를 사용하여 연관 규칙 모델 너깃을 생성할 수 있습니다. 이 모델 너깃의 스크립팅 이름은 *applyassociationrulesnode*입니다. 모델링 노드 자체의 스크립팅에 대한 자세한 정보는 194 페이지의 『associationrulesnode 특성』을 참조하십시오.

표 153. *applyassociationrulesnode* 특성

applyassociationrulesnode 특성	데이터 유형	특성 설명
max_predictions	integer	스코어에 대한 각 입력에 적용될 수 있는 최대 규칙 수입니다.
criterion	Confidence Rulesupport Lift Conditionsupport Deployability	규칙의 강도를 판별하는 데 사용하는 측도를 선택하십시오.
allow_repeats	부울	동일한 예측을 갖는 규칙이 스코어에 포함되는지 여부를 판별합니다.
check_input	NoPredictions Predictions NoCheck	

applyautoclassifiernode 특성

자동 분류자 모델링 노드를 사용하여 자동 분류자 모델 너깃을 생성할 수 있습니다. 이 모델 너깃의 스크립팅 이름은 *applyautoclassifiernode*입니다. 모델링 노드 자체의 스크립팅에 대한 자세한 정보는 196 페이지의 『autoclassifiernode 특성』을 참조하십시오.

표 154. *applyautoclassifiernode* 특성.

applyautoclassifiernode 특성	값	특성 설명
flag_ensemble_method	Voting ConfidenceWeightedVoting RawPropensityWeightedVoting HighestConfidence AverageRawPropensity	앙상블 스코어를 판별하는 데 사용하는 방법을 지정합니다. 이 설정은 선택된 대상이 플래그 필드인 경우에만 적용됩니다.

표 154. *applyautoclassifiernode* 특성 (계속).

applyautoclassifiernode 특성	값	특성 설명
<code>flag_voting_tie_selection</code>	Random HighestConfidence RawPropensity	투표 방법이 선택되는 경우 등순위를 해결하는 방법을 지정합니다. 이 설정은 선택된 대상이 플래그 필드인 경우에만 적용됩니다.
<code>set_ensemble_method</code>	Voting ConfidenceWeightedVoting HighestConfidence	앙상블 스코어를 판별하는 데 사용하는 방법을 지정합니다. 이 설정은 선택된 대상이 세트 필드인 경우에만 적용됩니다.
<code>set_voting_tie_selection</code>	Random HighestConfidence	투표 방법이 선택되는 경우 등순위를 해결하는 방법을 지정합니다. 이 설정은 선택된 대상이 명목 필드인 경우에만 적용됩니다.

applyautoclusternode 특성

자동 군집 모델링 노드를 사용하여 자동 군집 모델 너트를 생성할 수 있습니다. 이 모델 너트의 스크립팅 이름은 *applyautoclusternode*입니다. 이 모델 너트의 다른 특성은 존재하지 않습니다. 모델링 노드 자체의 스크립팅에 대한 자세한 정보는 199 페이지의 『autoclusternode 특성』을 참조하십시오.

applyautonumericnode 특성

자동 수치 모델링 노드를 사용하여 자동 수치 모델 너트를 생성할 수 있습니다. 이 모델 너트의 스크립팅 이름은 *applyautonumericnode*입니다. 모델링 노드 자체의 스크립팅에 대한 자세한 정보는 200 페이지의 『autonumericnode 특성』을 참조하십시오.

표 155. *applyautonumericnode* 특성.

applyautonumericnode 특성	값	특성 설명
<code>calculate_standard_error</code>	<i>flag</i>	

applybayesnetnode 특성

베이지안 신경망 모형 노드를 사용하여 베이지안 신경망 모형 너트를 생성할 수 있습니다. 이 모델 너트의 스크립팅 이름은 *applybayesnetnode*입니다. 모델링 노드 자체의 스크립팅에 대한 자세한 정보는 201 페이지의 『bayesnetnode 특성』을 참조하십시오.

표 156. *applybayesnetnode* 특성.

applybayesnetnode 특성	값	특성 설명
<code>all_probabilities</code>	<i>flag</i>	
<code>raw_propensity</code>	<i>flag</i>	
<code>adjusted_propensity</code>	<i>flag</i>	
<code>calculate_raw_propensities</code>	<i>flag</i>	
<code>calculate_adjusted_propensities</code>	<i>flag</i>	

applyc50node 특성

C5.0 모델링 노드를 사용하면 C5.0 모델 너깃을 생성할 수 있습니다. 이 모델 너깃의 스크립팅 이름은 *applyc50node*입니다. 모델링 노드 자체의 스크립팅에 대한 자세한 정보는 203 페이지의 『c50node 특성』을 참조하십시오.

표 157. *applyc50node* 특성.

applyc50node 특성	값	특성 설명
sql_generate	Never NoMissingValues	규칙 세트 실행 중에 SQL 생성 옵션을 설정하는 데 사용됩니다.
calculate_conf	flag	SQL 생성이 사용 가능할 때 사용할 수 있습니다. 이 특성에는 생성된 트리에서의 신뢰도 계산이 포함됩니다.
calculate_raw_propensities	flag	
calculate_adjusted_propensities	flag	

applycarmanode 특성

CARMA 모델링 노드를 사용하면 CARMA 모델 너깃을 생성할 수 있습니다. 이 모델 너깃의 스크립팅 이름은 *applycarmanode*입니다. 이 모델 너깃의 다른 특성은 존재하지 않습니다. 모델링 노드 자체의 스크립팅에 대한 자세한 정보는 205 페이지의 『carmanode 특성』을 참조하십시오.

applycartnode 특성

C&R 트리 모델링 노드를 사용하면 C&R 트리 모델 너깃을 생성할 수 있습니다. 이 모델 너깃의 스크립팅 이름은 *applycartnode*입니다. 모델링 노드 자체의 스크립팅에 대한 자세한 정보는 206 페이지의 『cartnode 특성』을 참조하십시오.

표 158. *applycartnode* 특성.

applycartnode 특성	값	특성 설명
enable_sql_generation	Never MissingValues NoMissingValues	규칙 세트 실행 중에 SQL 생성 옵션을 설정하는 데 사용됩니다.
calculate_conf	flag	SQL 생성이 사용 가능할 때 사용할 수 있습니다. 이 특성에는 생성된 트리에서의 신뢰도 계산이 포함됩니다.
display_rule_id	flag	각 레코드가 지정되는 터미널 노드에 대한 ID를 표시하는 스코어링 출력에 필드를 추가합니다.
calculate_raw_propensities	flag	
calculate_adjusted_propensities	flag	

applychaidnode 특성

CHAID 모델링 노드를 사용하여 CHAID 모델 너깃을 생성할 수 있습니다. 이 모델 너깃의 스크립팅 이름은 *applychaidnode*입니다. 모델링 노드 자체의 스크립팅에 대한 자세한 정보는 208 페이지의 『chaidnode 특성』을 참조하십시오.

표 159. *applychaidnode* 특성.

applychaidnode 특성	값	특성 설명
enable_sql_generation	Never MissingValues	규칙 세트 실행 중에 SQL 생성 옵션을 설정하는 데 사용됩니다.
calculate_conf	flag	
display_rule_id	flag	각 레코드가 지정되는 터미널 노드에 대한 ID를 표시하는 스코어링 출력에 필드를 추가합니다.
calculate_raw_propensities	flag	
calculate_adjusted_propensities	flag	

applycoxregnode 특성

Cox 모델링 노드를 사용하면 Cox 모델 너깃을 생성할 수 있습니다. 이 모델 너깃의 스크립팅 이름은 *applycoxregnode*입니다. 모델링 노드 자체의 스크립팅에 대한 자세한 정보는 210 페이지의 『coxregnode 특성』을 참조하십시오.

표 160. *applycoxregnode* 특성.

applycoxregnode 특성	값	특성 설명
future_time_as	Intervals Fields	
time_interval	number	
num_future_times	integer	
time_field	field	
past_survival_time	field	
all_probabilities	flag	
cumulative_hazard	flag	

applydecisionlistnode 특성

의사결정 목록 모델링 노드를 사용하면 의사결정 목록 모델 너깃을 생성할 수 있습니다. 이 모델 너깃의 스크립팅 이름은 *applydecisionlistnode*입니다. 모델링 노드 자체의 스크립팅에 대한 자세한 정보는 212 페이지의 『decisionlistnode 특성』을 참조하십시오.

표 161. *applydecisionlistnode* 특성.

applydecisionlistnode 특성	값	특성 설명
enable_sql_generation	flag	true일 때 IBM SPSS Modeler는 의사결정 목록 모델을 SQL에 다시 넣으려고 시도합니다.

표 161. *applydecisionlistnode* 특성 (계속).

applydecisionlistnode 특성	값	특성 설명
calculate_raw_propensities	<i>flag</i>	
calculate_adjusted_propensities	<i>flag</i>	

applydiscriminantnode 특성

판별 모델링 노드를 사용하여 판별 모델 너깃을 생성할 수 있습니다. 이 모델 너깃의 스크립팅 이름은 *applydiscriminantnode*입니다. 모델링 노드 자체의 스크립팅에 대한 자세한 정보는 213 페이지의 『discriminantnode 특성』을 참조하십시오.

표 162. *applydiscriminantnode* 특성.

applydiscriminantnode 특성	값	특성 설명
calculate_raw_propensities	<i>flag</i>	
calculate_adjusted_propensities	<i>flag</i>	

applyextension 특성



확장 모델 노드는 확장 모델 너깃을 생성하는 데 사용할 수 있습니다. 이 모델 너깃의 스크립팅 이름은 *applyextension*입니다. 모델링 노드 자체의 스크립팅에 대한 자세한 정보는 215 페이지의 『extensionmodelnode 특성』을 참조하십시오.

Python for Spark 예제

```
#### script example for Python for Spark
applyModel = stream.findByType("extension_apply", None)

score_script = """
import json
import spss.pyspark.runtime
from pyspark.mllib.regression import LabeledPoint
from pyspark.mllib.linalg import DenseVector
from pyspark.mllib.tree import DecisionTreeModel
from pyspark.sql.types import StringType, StructField

cxt = spss.pyspark.runtime.getContext()

if cxt.isComputeDataModelOnly():
    _schema = cxt.getSparkInputSchema()
    _schema.fields.append(StructField("Prediction", StringType(), nullable=True))
    cxt.setSparkOutputSchema(_schema)
else:
    df = cxt.getSparkInputData()

    _modelPath = cxt.getModelContentToPath("TreeModel")
    metadata = json.loads(cxt.getModelContentToString("model.dm"))
```



```

schema = df.dtypes[:]
target = "Drug"
predictors = ["Age","BP","Sex","Cholesterol","Na","K"]

lookup = {}
for i in range(0,len(schema)):
    lookup[schema[i][0]] = i

def row2LabeledPoint(dm,lookup,target,predictors,row):
    target_index = lookup[target]
    tval = dm[target_index].index(row[target_index])
    pvals = []
    for predictor in predictors:
        predictor_index = lookup[predictor]
        if isinstance(dm[predictor_index],list):
            pval = row[predictor_index] in dm[predictor_index].index
(row[predictor_index]) or -1
        else:
            pval = row[predictor_index]
        pvals.append(pval)
    return LabeledPoint(tval, DenseVector(pvals))

# convert dataframe to an RDD containing LabeledPoint
lps = df.rdd.map(lambda row: row2LabeledPoint(metadata,lookup,target,predictors,row))
treeModel = DecisionTreeModel.load(cxt.getSparkContext(), _modelPath);
# score the model, produces an RDD containing just double values
predictions = treeModel.predict(lps.map(lambda lp: lp.features))

def addPrediction(x,dm,lookup,target):
    result = []
    for _idx in range(0, len(x[0])):
        result.append(x[0][_idx])
    result.append(dm[lookup[target]][int(x[1])])
    return result

_schema = cxt.getSparkInputSchema()
_schema.fields.append(StructField("Prediction", StringType(), nullable=True))
rdd2 = df.rdd.zip(predictions).map(lambda x:addPrediction(x, metadata, lookup, target))
outDF = cxt.getSparkSQLContext().createDataFrame(rdd2, _schema)

cxt.setSparkOutputData(outDF)
"""
applyModel.setPropertyValue("python_syntax", score_script)

```

R 예제

```

#### script example for R
applyModel.setPropertyValue("r_syntax", """
result<-predict(modelerModel,newdata=modelerData)
modelerData<-cbind(modelerData,result)
var1<-c(fieldName="NaPrediction",fieldLabel="",fieldStorage="real",fieldMeasure="",
fieldFormat="",fieldRole="")
modelerDataModel<-data.frame(modelerDataModel,var1)""")

```

표 163. *applyextension* 특성

applyextension 특성	값	특성 설명
r_syntax	string	모델 스코어링의 R 스크립팅 구문

표 163. *applyextension* 특성 (계속)

applyextension 특성	값	특성 설명
python_syntax	string	모델 스코어링을 위한 Python 스크립팅 구문입니다.
use_batch_size	flag	배치 처리 사용을 가능하게 합니다.
batch_size	integer	각 배치에 포함할 데이터 레코드의 수를 지정하십시오.
convert_flags	StringsAndDoubles LogicalValues	플래그 필드를 변환하는 옵션.
convert_missing	flag	결측값을 R NA 값으로 변환하는 옵션입니다.
convert_datetime	flag	날짜 또는 날짜/시간 형식을 갖는 변수를 R 날짜/시간 형식으로 변환하는 옵션.
convert_datetime_class	POSIXct POSIXlt	날짜 또는 날짜/시간 형식을 갖는 변수를 변환할 형식을 지정하는 옵션.

applyfactornode 특성

PCA/요인 모델링 노드를 사용하여 PCA/요인 모델 너깃을 생성할 수 있습니다. 이 모델 너깃의 스크립팅 이름은 *applyfactornode*입니다. 이 모델 너깃의 다른 특성은 존재하지 않습니다. 모델링 노드 자체의 스크립팅에 대한 자세한 정보는 217 페이지의 『factornode 특성』을 참조하십시오.

applyfeatureselectionnode 특성

필드선택 모델링 노드를 사용하면 필드선택 모델 너깃을 생성할 수 있습니다. 이 모델 너깃의 스크립팅 이름은 *applyfeatureselectionnode*입니다. 모델링 노드 자체의 스크립팅에 대한 자세한 정보는 219 페이지의 『featureselectionnode 특성』을 참조하십시오.

표 164. *applyfeatureselectionnode* 특성.

applyfeatureselectionnode 특성	값	특성 설명
selected_ranked_fields		모델 브라우저에서 선택된 순위 필드를 지정합니다.
selected_screened_fields		모델 브라우저에서 선택된 차단 필드를 지정합니다.

applygeneralizedlinearnode 특성

일반화 선형(genlin) 모델링 노드를 사용하여 일반화 선형 모델 너깃을 생성할 수 있습니다. 이 모델 너깃의 스크립팅 이름은 *applygeneralizedlinearnode*입니다. 모델링 노드 자체의 스크립팅에 대한 자세한 정보는 221 페이지의 『genlinnode 특성』을 참조하십시오.

표 165. *applygeneralizedlinearnode* 특성.

applygeneralizedlinearnode 특성	값	특성 설명
calculate_raw_propensities	flag	
calculate_adjusted_propensities	flag	

applyglmnode 특성

GLMM 모델링 노드를 사용하여 GLMM 모델 너깃을 생성할 수 있습니다. 이 모델 너깃의 스크립팅 이름은 *applyglmnode*입니다. 모델링 노드 자체의 스크립팅에 대한 자세한 정보는 224 페이지의 『glmnode 특성』을 참조하십시오.

표 166. *applyglmnode* 특성.

applyglmnode 특성	값	특성 설명
confidence	onProbability onIncrease	스코어링 신뢰도 계산을 위한 기초입니다. 최고 예측 확률 또는 최고 및 두 번째 최고 예측 확률 사이의 차이입니다.
score_category_probabilities	flag	True로 설정되면 범주형 대상에 대한 예측 확률을 생성합니다. 범주마다 하나의 필드가 작성됩니다. 기본값은 False입니다.
max_categories	integer	확률을 예측할 최대 범주 수입니다. score_category_probabilities가 True인 경우에만 사용합니다.
score_propensity	flag	True로 설정되면 플래그 목표를 갖는 모델에 대한 원시 성향 스코어("True" 결과의 우도)를 생성합니다. 파티션이 유효한 경우, 검정 파티션을 기반으로 수정된 성향 스코어도 생성합니다. 기본값은 False입니다.
enable_sql_generation	udf native	스트림 실행 중에 SQL 생성 옵션을 설정하는데 사용합니다. 데이터베이스에 푸시백하고 SPSS® Modeler Server 스코어링 어댑터를 사용하여 스코어링하거나(스코어링 어댑터가 설치된 데이터베이스에 연결된 경우) SPSS Modeler에서 스코어링하는 옵션이 있습니다. 기본값은 udf입니다.

applygle 특성

GLE 모델링 노드는 GLE 모델 너깃을 생성하는 데에만 사용할 수 있습니다. 이 모델 너깃의 스크립팅 이름은 *applygle*입니다. 모델링 노드 자체의 스크립팅에 대한 자세한 정보는 228 페이지의 『gle 특성』을 참조하십시오.

표 167. *applygle* 특성

applygle 특성	값	특성 설명
enable_sql_generation	udf native	스트림 실행 중에 SQL 생성 옵션을 설정하는데 사용합니다. 데이터베이스로 푸시백하고 SPSS Modeler Server 스코어링 어댑터를 사용하여 스코어링하거나(스코어링 어댑터가 설치된 데이터베이스에 연결된 경우) SPSS Modeler 안에서 스코어링할 것을 선택하십시오.

applykmeansnode 특성

K-평균 모델링 노드를 사용하면 K-평균 모델 너깃을 생성할 수 있습니다. 이 모델 너깃의 스크립팅 이름은 *applykmeansnode*입니다. 이 모델 너깃의 다른 특성은 존재하지 않습니다. 모델링 노드 자체의 스크립팅에 대한 자세한 정보는 232 페이지의 『kmeansnode 특성』을 참조하십시오.

applyknnnode 특성

KNN 모델링 노드를 사용하면 KNN 모델 너깃을 생성할 수 있습니다. 이 모델 너깃의 스크립팅 이름은 *applyknnnode*입니다. 모델링 노드 자체의 스크립팅에 대한 자세한 정보는 233 페이지의 『knnnode 특성』을 참조하십시오.

표 168. *applyknnnode* 특성.

applyknnnode 특성	값	특성 설명
all_probabilities	flag	
save_distances	flag	

applykohonenode 특성

코호넨 모델링 노드를 사용하면 코호넨 모델 너깃을 생성할 수 있습니다. 이 모델 너깃의 스크립팅 이름은 *applykohonenode*입니다. 이 모델 너깃의 다른 특성은 존재하지 않습니다. 모델링 노드 자체의 스크립팅에 대한 자세한 정보는 203 페이지의 『c50node 특성』을 참조하십시오.

applylinearnode 특성

선형 모델링 노드를 사용하면 선형 모델 너깃을 생성할 수 있습니다. 이 모델 너깃의 스크립팅 이름은 *applylinearnode*입니다. 모델링 노드 자체의 스크립팅에 대한 자세한 정보는 236 페이지의 『linearnode 특성』을 참조하십시오.

표 169. *applylinearnode* 특성.

linear 특성	값	특성 설명
use_custom_name	flag	
custom_name	string	
enable_sql_generation	udf native puresql	스트림 실행 중에 SQL 생성 옵션을 설정하는데 사용합니다. 데이터베이스에 푸시백하고 SPSS® Modeler Server 스코어링 어댑터를 사용하여 스코어링하거나(스코어링 어댑터가 설치된 데이터베이스에 연결된 경우) SPSS Modeler에서 스코어링하거나 데이터베이스에 푸시백하고 SQL을 사용하여 스코어링하는 옵션이 있습니다. 기본값은 udf입니다.

applylinearasnode 특성

Linear-AS 모델링 노드를 사용하여 Linear-AS 모델 너깃을 생성할 수 있습니다. 이 모델 너깃의 스크립팅 이름은 *applylinearasnode*입니다. 모델링 노드 자체의 스크립팅에 대한 자세한 정보는 237 페이지의 『linearasnode 특성』을 참조하십시오.

표 170. *applylinearasnode* 특성

applylinearasnode 특성	값	특성 설명
enable_sql_generation	udf native	기본값은 udf입니다.

applylogregnode 특성

로지스틱 회귀분석 모델링 노드를 사용하면 로지스틱 회귀분석 모델 너깃을 생성할 수 있습니다. 이 모델 너깃의 스크립팅 이름은 *applylogregnode*입니다. 모델링 노드 자체의 스크립팅에 대한 자세한 정보는 238 페이지의 『logregnode 특성』을 참조하십시오.

표 171. *applylogregnode* 특성.

applylogregnode 특성	값	특성 설명
calculate_raw_propensities	flag	
calculate_conf	flag	
enable_sql_generation	flag	

applylsvmnode 특성

LSVM 모델링 노드를 사용하여 LSVM 모델 너깃을 생성할 수 있습니다. 이 모델 너깃의 스크립팅 이름은 *applylsvmnode*입니다. 모델링 노드 자체의 스크립팅에 대한 자세한 정보는 243 페이지의 『lsvmnode 특성』을 참조하십시오.

표 172. *applylsvmnode* 특성

applylsvmnode 특성	값	특성 설명
calculate_raw_propensities	flag	원시 성향 스코어 계산 여부를 지정합니다.
enable_sql_generation	udf native	스코어링 어댑터(설치된 경우)를 사용하거나 프로세스에서 스코어를 계산할지 아니면 데이터베이스 외부에서 스코어를 계산할지를 지정합니다.

applyneuralnetnode 특성

신경망 모델링 노드를 사용하여 신경망 모델 너깃을 생성할 수 있습니다. 이 모델 너깃의 스크립팅 이름은 *applyneuralnetnode*입니다. 모델링 노드 자체의 스크립팅에 대한 자세한 정보는 244 페이지의 『neuralnetnode 특성』을 참조하십시오.

주의: 신경망 너깃의 기능이 향상된 새 버전이 이 릴리스에서 사용 가능하며 다음 절(*applyneuralnetwork*)에서 설명됩니다. 이전 버전을 아직 사용할 수 있지만 새 버전을 사용하도록 스크립트를 업데이트할 것을 권장합니다. 이전 버전의 세부사항은 참조를 위해 여기에 보유되지만, 해당 지원은 차후 릴리스에서 제거됩니다.

표 173. *applyneuralnetnode* 특성.

applyneuralnetnode 특성	값	특성 설명
calculate_conf	flag	SQL 생성이 사용 가능할 때 사용할 수 있습니다. 이 특성에는 생성된 트리에서의 신뢰도 계산이 포함됩니다.
enable_sql_generation	flag	
nn_score_method	Difference SoftMax	
calculate_raw_propensities	flag	
calculate_adjusted_propensities	flag	

applyneuralnetworknode 특성

신경망 모델링 노드를 사용하여 신경망 모델 너깃을 생성할 수 있습니다. 이 모델 너깃의 스크립팅 이름은 *applyneuralnetworknode*입니다. 모델 노드 자체의 스크립팅에 대한 자세한 정보는 *neuralnetworknode* 특성을 참조하십시오.

표 174. *applyneuralnetworknode* 특성

applyneuralnetworknode 특성	값	특성 설명
use_custom_name	<i>flag</i>	
custom_name	<i>string</i>	
confidence	onProbability onIncrease	
score_category_probabilities	<i>flag</i>	
max_categories	<i>number</i>	
score_propensity	<i>flag</i>	
enable_sql_generation	udf native puresql	스트림 실행 중에 SQL 생성 옵션을 설정하는 데 사용합니다. 데이터베이스에 푸시백하고 SPSS® Modeler Server 스코어링 어댑터를 사용하여 스코어링하거나(스코어링 어댑터가 설치된 데이터베이스에 연결된 경우) SPSS Modeler에서 스코어링하거나 데이터베이스에 푸시백하고 SQL을 사용하여 스코어링하는 옵션이 있습니다. 기본값은 udf입니다.

applyocsvmnode 특성

One-Class SVM 노드를 사용하여 One-Class SVM 모델 너깃을 생성할 수 있습니다. 이 모델 너깃의 스크립팅 이름은 *applyocsvmnode*입니다. 이 모델 너깃의 다른 특성은 존재하지 않습니다. 모델링 노드 자체의 스크립팅에 대한 자세한 정보는 355 페이지의 『ocsvmnode 특성』을 참조하십시오.

applyquestnode 특성

QUEST 모델링 노드를 사용하여 QUEST 모델 너깃을 생성할 수 있습니다. 이 모델 너깃의 스크립팅 이름은 *applyquestnode*입니다. 모델링 노드 자체의 스크립팅에 대한 자세한 정보는 248 페이지의 『questnode 특성』을 참조하십시오.

표 175. *applyquestnode* 특성.

applyquestnode 특성	값	특성 설명
enable_sql_generation	Never MissingValues NoMissingValues	규칙 세트 실행 중에 SQL 생성 옵션을 설정하는 데 사용합니다.
calculate_conf	<i>flag</i>	

표 175. *applyquestnode* 특성 (계속).

applyquestnode 특성	값	특성 설명
display_rule_id	<i>flag</i>	각 레코드가 지정되는 터미널 노드에 대한 ID를 표시하는 스코어링 출력에 필드를 추가합니다.
calculate_raw_propensities	<i>flag</i>	
calculate_adjusted_propensities	<i>flag</i>	

applyr 특성

R 작성 노드를 사용하여 R 모델 너깃을 생성할 수 있습니다. 이 모델 너깃의 스크립팅 이름은 *applyr* 입니다. 모델링 노드 자체의 스크립팅에 대한 자세한 정보는 203 페이지의 『*buildr* 특성』을 참조하십시오.

표 176. *applyr* 특성

applyr 특성	값	특성 설명
score_syntax	<i>string</i>	모델 스코어링의 R 스크립팅 구문
convert_flags	StringsAndDoubles LogicalValues	플래그 필드를 변환하는 옵션.
convert_datetime	<i>flag</i>	날짜 또는 날짜/시간 형식을 갖는 변수를 R 날짜/시간 형식으로 변환하는 옵션.
convert_datetime_class	POSIXct POSIXlt	날짜 또는 날짜/시간 형식을 갖는 변수를 변환할 형식을 지정하는 옵션.
convert_missing	<i>flag</i>	결측값을 R NA 값으로 변환하는 옵션.
use_batch_size	<i>flag</i>	배치 처리 사용을 가능하게 합니다.
batch_size	<i>integer</i>	각 배치에 포함할 데이터 레코드의 수를 지정하십시오.

applyrandomtrees 특성

랜덤 트리 모델링 노드를 사용하여 랜덤 트리 모델 너깃을 생성할 수 있습니다. 이 모델 너깃의 스크립트 이름은 *applyrandomtrees*입니다. 모델링 노드 자체의 스크립팅에 대한 자세한 정보는 250 페이지의 『*randomtrees* 특성』을 참조하십시오.

표 177. *applyrandomtrees* 특성

applyrandomtrees 특성	값	특성 설명
calculate_conf	<i>flag</i>	이 특성에는 생성된 트리에서의 신뢰도 계산이 포함됩니다.
enable_sql_generation	udf native	스트림 실행 중에 SQL 생성 옵션을 설정하는데 사용합니다. 데이터베이스로 푸시백하고 SPSS Modeler Server 스코어링 어댑터를 사용하여 스코어링하거나(스코어링 어댑터가 설치된 데이터베이스에 연결된 경우) SPSS Modeler 안에서 스코어링할 것을 선택하십시오.

applyregressionnode 특성

선형 회귀 모델링 노드를 사용하여 선형 회귀 모형 너깃을 생성할 수 있습니다. 이 모델 너깃의 스크립팅 이름은 *applyregressionnode*입니다. 이 모델 너깃의 다른 특성은 존재하지 않습니다. 모델링 노드 자체의 스크립팅에 대한 자세한 정보는 251 페이지의 『*regressionnode* 특성』을 참조하십시오.

applyselflearningnode 특성

자체 학습 반응 모델(SLRM) 모델링 노드를 사용하여 SLRM 모델 너깃을 생성할 수 있습니다. 이 모델 너깃의 스크립팅 이름은 *applyselflearningnode*입니다. 모델링 노드 자체의 스크립팅에 대한 자세한 정보는 255 페이지의 『*slrmnode* 특성』을 참조하십시오.

표 178. *applyselflearningnode* 특성.

applyselflearningnode 특성	값	특성 설명
max_predictions	<i>number</i>	
randomization	<i>number</i>	
scoring_random_seed	<i>number</i>	
sort	ascending descending	최고 또는 최저 스코어를 갖는 제안이 처음 표시되는지 여부를 지정합니다.
model_reliability	<i>flag</i>	설정 탭에서 모델 신뢰도 옵션을 고려합니다.

applysequencenode 특성

시퀀스 모델링 노드를 사용하여 시퀀스 모델 너깃을 생성할 수 있습니다. 이 모델 너깃의 스크립팅 이름은 *applysequencenode*입니다. 이 모델 너깃의 다른 특성은 존재하지 않습니다. 모델링 노드 자체의 스크립팅에 대한 자세한 정보는 253 페이지의 『*sequencenode* 특성』을 참조하십시오.

applysvmnode 특성

SVM 모델링 노드를 사용하면 SVM 모델 너깃을 생성할 수 있습니다. 이 모델 너깃의 스크립팅 이름은 *applysvmnode*입니다. 모델링 노드 자체의 스크립팅에 대한 자세한 정보는 260 페이지의 『*svmnode* 특성』을 참조하십시오.

표 179. *applysvmnode* 특성.

applysvmnode 특성	값	특성 설명
all_probabilities	<i>flag</i>	
calculate_raw_propensities	<i>flag</i>	
calculate_adjusted_propensities	<i>flag</i>	

applystpnode 특성

STP 모델링 노드를 사용하여 연관된 모델 너깃을 생성할 수 있는데, 이 너깃은 출력 뷰어에서 모델 출력을 표시합니다. 이 모델 너깃의 스크립팅 이름은 *applystpnode*입니다. 모델링 노드 자체의 스크립팅에 대한 자세한 정보는 256 페이지의 『stpnode 특성』을 참조하십시오.

표 180. *applystpnode* 특성

applystpnode 특성	데이터 유형	특성 설명
uncertainty_factor	부울	최소값 0, 최대값 100.

applytcmnode 특성

시간 인과 모델화(TCM) 모델링 노드를 사용하면 TCM 모델 너깃을 생성할 수 있습니다. 이 모델 너깃의 스크립팅 이름은 *applytcmnode*입니다. 모델링 노드 자체의 스크립팅에 대한 자세한 정보는 261 페이지의 『tcmnode 특성』을 참조하십시오.

표 181. *applytcmnode* 특성

applytcmnode 특성	값	특성 설명
ext_future	<i>boolean</i>	
ext_future_num	<i>integer</i>	
noise_res	<i>boolean</i>	
conf_limits	<i>boolean</i>	
target_fields	<i>list</i>	
target_series	<i>list</i>	

applyts 특성

시계열 모델링 노드를 사용하여 시계열 모델 너깃을 생성할 수 있습니다. 이 모델 너깃의 스크립트 이름은 *applyts*입니다. 모델링 노드 자체의 스크립팅에 대한 자세한 정보는 265 페이지의 『ts 특성』을 참조하십시오.

표 182. *applyts* 특성

applyts 특성	값	특성 설명
extend_records_into_future	부울	
ext_future_num	<i>integer</i>	
compute_future_values_input	부울	
forecastperiods	<i>integer</i>	
noise_res	<i>boolean</i>	
conf_limits	<i>boolean</i>	
target_fields	<i>list</i>	
target_series	<i>list</i>	
includeTargets	<i>field</i>	

applytimeseriesnode 특성(더 이상 사용되지 않음)

시계열 모델링 노드를 사용하여 시계열 모델 너깃을 생성할 수 있습니다. 이 모델 너깃의 스크립팅 이름은 *applytimeseriesnode*입니다. 모델링 노드 자체의 스크립팅에 대한 자세한 정보는 269 페이지의 『timeseriesnode 특성(더 이상 사용되지 않음)』을 참조하십시오.

표 183. *applytimeseriesnode* 특성.

applytimeseriesnode 특성	값	특성 설명
calculate_conf	flag	
calculate_residuals	flag	

applytreeas 특성

Tree-AS 모델링 노드를 사용하여 Tree-AS 모델 너깃을 생성할 수 있습니다. 이 모델 너깃의 스크립팅 이름은 *applytreeas*입니다. 모델링 노드 자체의 스크립팅에 대한 자세한 정보는 271 페이지의 『treeas 특성』을 참조하십시오.

표 184. *applytreeas* 특성

applytreeas 특성	값	특성 설명
calculate_conf	flag	이 특성에는 생성된 트리에서의 신뢰도 계산이 포함됩니다.
display_rule_id	flag	각 레코드가 지정되는 터미널 노드에 대한 ID를 표시하는 스코어링 출력에 필드를 추가합니다.
enable_sql_generation	udf native	스트림 실행 중에 SQL 생성 옵션을 설정하는데 사용합니다. 데이터베이스로 푸시백하고 SPSS Modeler Server 스코어링 어댑터를 사용하여 스코어링하거나(스코어링 어댑터가 설치된 데이터베이스에 연결된 경우) SPSS Modeler 안에서 스코어링할 것을 선택하십시오.

applytwostepnode 특성

이단계 모델링 노드를 사용하여 이단계 모델 너깃을 생성할 수 있습니다. 이 모델 너깃의 스크립팅 이름은 *applytwostepnode*입니다. 이 모델 너깃의 다른 특성은 존재하지 않습니다. 모델링 노드 자체의 스크립팅에 대한 자세한 정보는 273 페이지의 『twostepnode 특성』을 참조하십시오.

applytwestepAS 특성

이단계 AS 모델링 노드를 사용하여 이단계 AS 모델 너깃을 생성할 수 있습니다. 이 모델 너깃의 스크립팅 이름은 *applytwestepAS*입니다. 모델링 노드 자체의 스크립팅에 대한 자세한 정보는 274 페이지의 『twestepAS 특성』을 참조하십시오.

표 185. *applytwestepAS* 특성

applytwestepAS 특성	값	특성 설명
enable_sql_generation	udf native	스트림 실행 중에 SQL 생성 옵션을 설정하는데 사용합니다. 데이터베이스에 푸시백하고 SPSS® Modeler Server 스코어링 어댑터를 사용하여 스코어링하거나(스코어링 어댑터가 설치된 데이터베이스에 연결된 경우) SPSS Modeler에서 스코어링하는 옵션이 있습니다. 기본값은 udf입니다.

applyxgboosttreenode 특성

XGBoost Tree 노드를 사용하여 XGBoost Tree 모델 너깃을 생성할 수 있습니다. 이 모델 너깃의 스크립팅 이름은 *applyxgboosttreenode*입니다. 이 모델 너깃의 다른 특성은 존재하지 않습니다. 모델링 노드 자체의 스크립팅에 대한 자세한 정보는 353 페이지의 『xgboosttreenode 특성』을 참조하십시오.

applyxgboostlinearnode 특성

XGBoost Linear 노드를 사용하여 XGBoost Linear 모델 너깃을 생성할 수 있습니다. 이 모델 너깃의 스크립팅 이름은 *applyxgboostlinearnode*입니다. 이 모델 너깃의 다른 특성은 존재하지 않습니다. 모델링 노드 자체의 스크립팅에 대한 자세한 정보는 355 페이지의 『xboostlinearnode 특성』을 참조하십시오.

제 15 장 데이터베이스 모델링 노드 특성

IBM SPSS Modeler에서는 Microsoft SQL Server Analysis Services, Oracle Data Mining, 및 IBM Netezza® Analytics 등의 데이터베이스 벤더에서 사용 가능한 데이터 마이닝 및 모델링 도구와 통합을 지원합니다. 모두가 IBM SPSS Modeler 애플리케이션 안에서 제공되는 기본 데이터베이스 알고리즘을 사용하여 모델을 작성 및 스코어링할 수 있습니다. 또한 이 절에서 설명하는 특성을 사용하여 스크립팅을 통해 데이터베이스 모델을 작성 및 조작할 수도 있습니다.

예를 들어, 다음 스크립트 발췌본은 IBM SPSS Modeler 스크립팅 인터페이스를 사용하여 Microsoft 의사결정 트리 모형의 작성을 설명합니다.

```
stream = modeler.script.stream()
msbuilder = stream.createAt("mstreenode", "MSBuilder", 200, 200)

msbuilder.setPropertyValue("analysis_server_name", 'localhost')
msbuilder.setPropertyValue("analysis_database_name", 'TESTDB')
msbuilder.setPropertyValue("mode", 'Expert')
msbuilder.setPropertyValue("datasource", 'LocalServer')
msbuilder.setPropertyValue("target", 'Drug')
msbuilder.setPropertyValue("inputs", ['Age', 'Sex'])
msbuilder.setPropertyValue("unique_field", 'IDX')
msbuilder.setPropertyValue("custom_fields", True)
msbuilder.setPropertyValue("model_name", 'MSDRUG')

typenode = stream.findByType("type", None)
stream.link(typenode, msbuilder)
results = []
msbuilder.run(results)
msapplier = stream.createModelApplierAt(results[0], "Drug", 200, 300)
tablenode = stream.createAt("table", "Results", 300, 300)
stream.linkBetween(msapplier, typenode, tablenode)
msapplier.setPropertyValue("sql_generate", True)
tablenode.run([])
```

Microsoft 모델링에 대한 노드 특성

Microsoft 모델링 노드 특성

공통 특성

다음 특성은 Microsoft 데이터베이스 모델링 노드에 공통적입니다.

표 186. 공통 Microsoft 노드 특성

공통 Microsoft 노드 특성	값	특성 설명
analysis_database_name	string	분석 서비스 데이터베이스의 이름.

표 186. 공통 Microsoft 노드 특성 (계속)

공통 Microsoft 노드 특성	값	특성 설명
analysis_server_name	string	분석 서비스 호스트의 이름.
use_transactional_data	flag	입력 데이터가 표 형식 또는 트랜잭션 형식인지 여부를 지정합니다.
inputs	list	표 형식 데이터에 대한 입력 필드.
target	field	예측 필드(MS 군집 또는 시퀀스 군집 노드에 적용되지 않음).
unique_field	field	키 필드.
msas_parameters	structured	알고리즘 모수. 자세한 정보는 297 페이지의 『알고리즘 모수』의 내용을 참조하십시오.
with_drillthrough	flag	드릴스루 사용 옵션.

MS 의사결정 트리

mstreenode 유형의 노드에 대해 정의되는 특정 특성이 없습니다. 이 절의 시작에 있는 공통 Microsoft 특성을 참조하십시오.

MS 군집화

msclusternode 유형의 노드에 대해 정의되는 특정 특성이 없습니다. 이 절의 시작에 있는 공통 Microsoft 특성을 참조하십시오.

MS 연관 규칙

다음 특정 특성이 msassocnode 유형의 노드에 사용 가능합니다.

표 187. msassocnode 특성

msassocnode 특성	값	특성 설명
id_field	field	데이터의 각 트랜잭션을 식별합니다.
trans_inputs	list	트랜잭션 데이터의 입력 필드입니다.
transactional_target	field	예측 필드(트랜잭션 데이터).

MS Naive Bayes

msbayesnode 유형의 노드에 대해 정의되는 특정 특성이 없습니다. 이 절의 시작에 있는 공통 Microsoft 특성을 참조하십시오.

MS 선형 회귀

msregressionnode 유형의 노드에 대해 정의되는 특정 특성이 없습니다. 이 절의 시작에 있는 공통 Microsoft 특성을 참조하십시오.

MS 신경망

msneuralnetworknode 유형의 노드에 대해 정의되는 특정 특성이 없습니다. 이 절의 시작에 있는 공통 Microsoft 특성을 참조하십시오.

MS 로지스틱 회귀분석

mslogisticnode 유형의 노드에 대해 정의되는 특정 특성이 없습니다. 이 절의 시작에 있는 공통 Microsoft 특성을 참조하십시오.

MS 시계열

mstimeseriesnode 유형의 노드에 대해 정의되는 특정 특성이 없습니다. 이 절의 시작에 있는 공통 Microsoft 특성을 참조하십시오.

MS 시퀀스 군집

다음 특정 특성이 mssequenceclusternode 유형의 노드에 사용 가능합니다.

표 188. mssequenceclusternode 특성

mssequenceclusternode 특성	값	특성 설명
id_field	field	데이터의 각 트랜잭션을 식별합니다.
input_fields	list	트랜잭션 데이터의 입력 필드입니다.
sequence_field	field	순차규칙 식별자입니다.
target_field	field	예측 필드(표 형식 데이터).

알고리즘 모수

각 Microsoft 데이터베이스 모델 유형은 msas_parameters 특성을 사용하여 설정할 수 있는 특정 모수를 갖고 있습니다. 예:

```
stream = modeler.script.stream()
msregressionnode = stream.findByType("msregression", None)
msregressionnode.setPropertyValue("msas_parameters", [{"MAXIMUM_INPUT_ATTRIBUTES", 255}, {"MAXIMUM_OUTPUT_ATTRIBUTES", 255}])
```

이들 모수는 SQL Server에서 파생됩니다. 각 노드에 대한 관련 모수를 보려면 다음을 수행하십시오.

1. 데이터베이스 소스 노드를 캔버스에 배치하십시오.
2. 데이터베이스 소스 노드를 여십시오.
3. 데이터 소스 드롭 다운 목록에서 유효한 소스를 선택하십시오.
4. 테이블 이름 목록에서 유효한 테이블을 선택하십시오.
5. 확인을 클릭하여 데이터베이스 소스 노드를 닫으십시오.
6. 해당 특성을 나열하려는 Microsoft 데이터베이스 모델링 노드를 연결하십시오.
7. 데이터베이스 모델링 노드를 여십시오.
8. 전문가 탭을 선택하십시오.

이 노드에 대한 사용 가능한 `msas_parameters` 특성이 표시됩니다.

Microsoft 모델 너깃 특성

다음 특성은 Microsoft 데이터베이스 모델링 노드를 사용하여 작성되는 모델 너깃을 위한 것입니다.

MS 의사결정 트리

표 189. MS 의사결정 트리 특성

aplymstreenode 특성	값	설명
analysis_database_name	string	이 노드는 스트림에서 직접 스코어 기록될 수 있습니다. 이 특성은 분석 서비스 데이터베이스의 이름을 식별하는 데 사용됩니다.
analysis_server_name	string	분석 서버 호스트의 이름.
datasource	string	SQL Server ODBC 데이터 소스 이름(DSN)의 이름.
sql_generate	flag	SQL 생성을 가능하게 합니다.

MS 선형 회귀

표 190. MS 선형 회귀 특성

aplymsregressionnode 특성	값	설명
analysis_database_name	string	이 노드는 스트림에서 직접 스코어 기록될 수 있습니다. 이 특성은 분석 서비스 데이터베이스의 이름을 식별하는 데 사용됩니다.
analysis_server_name	string	분석 서버 호스트의 이름.

MS 신경망

표 191. MS 신경망 특성

aplymsneuralnetworknode 특성	값	설명
analysis_database_name	string	이 노드는 스트림에서 직접 스코어 기록될 수 있습니다. 이 특성은 분석 서비스 데이터베이스의 이름을 식별하는 데 사용됩니다.
analysis_server_name	string	분석 서버 호스트의 이름.

MS 로지스틱 회귀분석

표 192. MS 로지스틱 회귀분석 특성

aplymslogisticnode 특성	값	설명
analysis_database_name	string	이 노드는 스트림에서 직접 스코어 기록될 수 있습니다. 이 특성은 분석 서비스 데이터베이스의 이름을 식별하는 데 사용됩니다.
analysis_server_name	string	분석 서버 호스트의 이름.

MS 시계열

표 193. MS 시계열 특성

aplymstimeseriesnode 특성	값	설명
analysis_database_name	string	이 노드는 스트림에서 직접 스코어 기록될 수 있습니다. 이 특성은 분석 서비스 데이터베이스의 이름을 식별하는 데 사용합니다.
analysis_server_name	string	분석 서버 호스트의 이름.
start_from	new_prediction historical_prediction	미래 예측 또는 히스토리 예측을 할지 여부를 지정합니다.
new_step	number	미래 예측을 위한 시작 시간 기간을 정의합니다.
historical_step	number	히스토리 예측에 대한 시작 시간 기간을 정의합니다.
end_step	number	예측에 대한 종료 시간 기간을 정의합니다.

MS 시퀀스 군집

표 194. MS 시퀀스 군집 특성

aplymssequenceclusternode 특성	값	설명
analysis_database_name	string	이 노드는 스트림에서 직접 스코어 기록될 수 있습니다. 이 특성은 분석 서비스 데이터베이스의 이름을 식별하는 데 사용합니다.
analysis_server_name	string	분석 서버 호스트의 이름.

Oracle 모델링의 노드 특성

Oracle 모델링 노드 특성

다음 특성은 Oracle 데이터베이스 모델링 노드에 공통적입니다.

표 195. 공통 Oracle 노드 특성.

공통 Oracle 노드 특성	값	특성 설명
target	field	
inputs	필드 목록	
partition	field	모델링 작성의 훈련, 검정 및 다양한 단계를 위한 별도의 표본으로 데이터를 분할하는 데 사용하는 필드입니다.
datasource		
username		
password		
epassword		
use_model_name	flag	
model_name	string	새 모델의 사용자 정의 이름입니다.
use_partitioned_data	flag	파티션 필드가 정의된 경우, 이 옵션은 훈련 파티션의 데이터만 모델을 작성하는 데 사용하도록 보장합니다.

표 195. 공통 Oracle 노드 특성 (계속).

공통 Oracle 노드 특성	값	특성 설명
unique_field	<i>field</i>	
auto_data_prep	<i>flag</i>	Oracle 자동 데이터 준비 기능을 사용 또는 사용 안함으로 설정합니다(11g 데이터베이스만 해당).
costs	<i>structured</i>	다음 양식의 구조화 특성: [[drugA drugB 1.5] [drugA drugC 2.1]], 여기서 []의 인수는 실제 예측 비용입니다.
mode	Simple Expert	개별 노드 특성에서 언급되는 것처럼, Simple로 설정되는 경우 특정 속성이 무시되도록 합니다.
use_prediction_probability	<i>flag</i>	
prediction_probability	<i>string</i>	
use_prediction_set	<i>flag</i>	

Oracle Naive Bayes

다음 특성은 oranbnode 유형의 노드에 사용 가능합니다.

표 196. oranbnode 특성

oranbnode 특성	값	특성 설명
singleton_threshold	<i>number</i>	0.0-1.0*
pairwise_threshold	<i>number</i>	0.0-1.0*
priors	Data Equal Custom	
custom_priors	<i>structured</i>	다음 양식의 구조화 특성: set :oranbnode.custom_priors = [[drugA 1][drugB 2][drugC 3][drugX 4][drugY 5]]

* mode가 Simple로 설정되면 특성이 무시됩니다.

Oracle 적응형 베이스

다음 특성은 oraabnnode 유형의 노드에 사용 가능합니다.

표 197. oraabnnode 특성

oraabnnode 특성	값	특성 설명
model_type	SingleFeature MultiFeature NaiveBayes	
use_execution_time_limit	<i>flag</i>	*
execution_time_limit	<i>integer</i>	값은 0보다 커야 합니다.*
max_naive_bayes_predictors	<i>integer</i>	값은 0보다 커야 합니다.*
max_predictors	<i>integer</i>	값은 0보다 커야 합니다.*

표 197. oraabnnode 특성 (계속)

oraabnnode 특성	값	특성 설명
priors	Data Equal Custom	
custom_priors	<i>structured</i>	다음 양식의 구조화 특성: set :oraabnnode.custom_priors = [[drugA 1][drugB 2][drugC 3][drugX 4][drugY 5]]

* mode가 Simple로 설정되면 특성이 무시됩니다.

Oracle 지원 벡터 머신

다음 특성은 orasvmnode 유형의 노드에 사용 가능합니다.

표 198. orasvmnode 특성

orasvmnode 특성	값	특성 설명
active_learning	Enable Disable	
kernel_function	Linear Gaussian System	
normalization_method	zscore minmax none	
kernel_cache_size	<i>integer</i>	Gaussian kernel만 해당합니다. 값은 0보다 커야 합니다.*
convergence_tolerance	<i>number</i>	값은 0보다 커야 합니다.*
use_standard_deviation	<i>flag</i>	Gaussian kernel만 해당합니다.*
standard_deviation	<i>number</i>	값은 0보다 커야 합니다.*
use_epsilon	<i>flag</i>	회귀 모형만 해당합니다.*
epsilon	<i>number</i>	값은 0보다 커야 합니다.*
use_complexity_factor	<i>flag</i>	*
complexity_factor	<i>number</i>	*
use_outlier_rate	<i>flag</i>	1클래스 변량만 해당합니다.*
outlier_rate	<i>number</i>	1클래스 변량만 해당합니다. 0.0-1.0.*
weights	Data Equal Custom	
custom_weights	<i>structured</i>	다음 양식의 구조화 특성: set :orasvmnode.custom_weights = [[drugA 1][drugB 2][drugC 3][drugX 4][drugY 5]]

* mode가 Simple로 설정되면 특성이 무시됩니다.

Oracle 일반화 선형 모델

다음 특성은 oraglmnode 유형의 노드에 사용 가능합니다.

표 199. oraglmnode 특성

oraglmnode 특성	값	특성 설명
normalization_method	zscore minmax none	
missing_value_handling	ReplaceWithMean UseCompleteRecords	
use_row_weights	<i>flag</i>	*
row_weights_field	<i>field</i>	*
save_row_diagnostics	<i>flag</i>	*
row_diagnostics_table	<i>string</i>	*
coefficient_confidence	<i>number</i>	*
use_reference_category	<i>flag</i>	*
reference_category	<i>string</i>	*
ridge_regression	Auto Off On	*
parameter_value	<i>number</i>	*
vif_for_ridge	<i>flag</i>	*

* mode가 Simple로 설정되면 특성이 무시됩니다.

Oracle 의사결정 트리

다음 특성은 oradecisiontreenode 유형의 노드에 사용 가능합니다.

표 200. oradecisiontreenode 특성

oradecisiontreenode 특성	값	특성 설명
use_costs	<i>flag</i>	
impurity_metric	Entropy Gini	
term_max_depth	<i>integer</i>	2-20*
term_minpct_node	<i>number</i>	0.0-10.0*
term_minpct_split	<i>number</i>	0.0-20.0*
term_minrec_node	<i>integer</i>	값은 0보다 커야 합니다.*
term_minrec_split	<i>integer</i>	값은 0보다 커야 합니다.*
display_rule_ids	<i>flag</i>	*

* mode가 Simple로 설정되면 특성이 무시됩니다.

Oracle O-Cluster

다음 특성은 oraclusternode 유형의 노드에 사용 가능합니다.

표 201. oraclusternode 특성

oraclusternode 특성	값	특성 설명
max_num_clusters	<i>integer</i>	값은 0보다 커야 합니다.
max_buffer	<i>integer</i>	값은 0보다 커야 합니다.*
sensitivity	<i>number</i>	0.0-1.0*

* mode가 Simple로 설정되면 특성이 무시됩니다.

Oracle K-평균

다음 특성은 orakmeansnode 유형의 노드에 사용 가능합니다.

표 202. orakmeansnode 특성

orakmeansnode 특성	값	특성 설명
num_clusters	<i>integer</i>	값은 0보다 커야 합니다.
normalization_method	zscore minmax none	
distance_function	Euclidean Cosine	
iterations	<i>integer</i>	0-20*
conv_tolerance	<i>number</i>	0.0-0.5*
split_criterion	Variance Size	기본값은 Variance입니다.*
num_bins	<i>integer</i>	값은 0보다 커야 합니다.*
block_growth	<i>integer</i>	1-5*
min_pct_attr_support	<i>number</i>	0.0-1.0*

* mode가 Simple로 설정되면 특성이 무시됩니다.

Oracle NMF

다음 특성은 oranmfnode 유형의 노드에 사용 가능합니다.

표 203. oranmfnode 특성

oranmfnode 특성	값	특성 설명
normalization_method	minmax none	
use_num_features	<i>flag</i>	*

표 203. oranmfnode 특성 (계속)

oranmfnode 특성	값	특성 설명
num_features	integer	0-1. 기본값은 알고리즘에 의해 데이터로부터 추정됩니다. *
random_seed	number	*
num_iterations	integer	0-500*
conv_tolerance	number	0.0-0.5*
display_all_features	flag	*

* mode가 Simple로 설정되면 특성이 무시됩니다.

Oracle Apriori

다음 특성은 oraapriorinode 유형의 노드에 사용 가능합니다.

표 204. oraapriorinode 특성

oraapriorinode 특성	값	특성 설명
content_field	field	
id_field	field	
max_rule_length	integer	2-20.
min_confidence	number	0.0-1.0.
min_support	number	0.0-1.0.
use_transactional_data	flag	

Oracle 최소 설명 길이(MDL)

oramdlnode 유형의 노드에 대해 정의되는 특정 특성이 없습니다. 이 절의 시작에 있는 공통 Oracle 특성을 참조하십시오.

Oracle 속성 중요도(AI)

다음 특성은 oraainode 유형의 노드에 사용 가능합니다.

표 205. oraainode 특성

oraainode 특성	값	특성 설명
custom_fields	flag	True인 경우 현재 노드에 대한 목표, 입력 및 기타 필드를 지정할 수 있습니다. False인 경우 업스트림 유형 노드의 현재 설정을 사용합니다.
selection_mode	ImportanceLevel ImportanceValue TopN	
select_important	flag	selection_mode가 ImportanceLevel로 설정될 때, 중요 필드 선택 여부를 지정합니다.
important_label	string	"중요한" 순위에 대한 레이블을 지정합니다.

표 205. *oraainode* 특성 (계속)

oraainode 특성	값	특성 설명
select_marginal	<i>flag</i>	selection_mode가 ImportanceLevel로 설정될 때, 보통 필드 선택 여부를 지정합니다.
marginal_label	<i>string</i>	"보통" 순위에 대한 레이블을 지정합니다.
important_above	<i>number</i>	0.0-1.0.
select_unimportant	<i>flag</i>	selection_mode가 ImportanceLevel로 설정될 때, 중요하지 않은 필드 선택 여부를 지정합니다.
unimportant_label	<i>string</i>	"중요하지 않은" 순위에 대한 레이블을 지정합니다.
unimportant_below	<i>number</i>	0.0-1.0.
importance_value	<i>number</i>	selection_mode가 ImportanceValue로 설정될 때, 사용할 절사 값을 지정합니다. 0부터 100까지의 값을 승인합니다.
top_n	<i>number</i>	selection_mode가 TopN으로 설정될 때, 사용할 절사 값을 지정합니다. 0에서부터 1000까지의 값을 승인합니다.

Oracle 모델 너깃 특성

다음 특성은 Oracle 모델을 사용하여 작성되는 모델 너깃을 위한 것입니다.

Oracle Naive Bayes

applyoranbnode 유형의 노드에 대해 정의되는 특정 특성이 없습니다.

Oracle 적응형 베이스

applyoraabnnode 유형의 노드에 대해 정의되는 특정 특성이 없습니다.

Oracle 지원 벡터 머신

applyorasvmnode 유형의 노드에 대해 정의되는 특정 특성이 없습니다.

Oracle 의사결정 트리

다음 특성은 applyoradecisiontreenode 유형의 노드에 사용 가능합니다.

표 206. *applyoradecisiontreenode* 특성

applyoradecisiontreenode 특성	값	특성 설명
use_costs	<i>flag</i>	
display_rule_ids	<i>flag</i>	

Oracle O-Cluster

applyoraoclusternode 유형의 노드에 대해 정의되는 특정 특성이 없습니다.

Oracle K-평균

applyorakmeansnode 유형의 노드에 대해 정의되는 특정 특성이 없습니다.

Oracle NMF

다음 특정 특성이 applyoranmfnode 유형의 노드에 사용 가능합니다.

표 207. applyoranmfnode 특성

applyoranmfnode 특성	값	특성 설명
display_all_features	flag	

Oracle Apriori

이 모델 너깃은 스크립팅에서 적용될 수 없습니다.

Oracle MDL

이 모델 너깃은 스크립팅에서 적용될 수 없습니다.

IBM Netezza Analytics 모델링의 노드 특성

Netezza 모델링 노드 특성

다음 특성은 IBM Netezza 데이터베이스 모델링 노드에 공통적입니다.

표 208. 공통 Netezza 노드 특성.

공통 Netezza 노드 특성	값	특성 설명
custom_fields	flag	True인 경우 현재 노드에 대한 목표, 입력 및 기타 필드를 지정할 수 있습니다. False인 경우 업스트림 유형 노드의 현재 설정을 사용합니다.
inputs	[field1 ... fieldN]	모델에서 사용하는 입력 또는 예측변수 필드입니다.
target	field	목표 필드(연속형 또는 범주형).
record_id	field	고유 레코드 식별자로 사용할 필드입니다.
use_upstream_connection	flag	True(기본값)인 경우, 업스트림 노드에 지정된 연결 세부사항입니다. move_data_to_connection이 지정되면 사용하지 않습니다.
move_data_connection	flag	True인 경우 connection에 의해 지정된 데이터베이스로 데이터를 이동합니다. use_upstream_connection이 지정되면 사용하지 않습니다.

표 208. 공통 Netezza 노드 특성 (계속).

공통 Netezza 노드 특성	값	특성 설명
connection	structured	모델이 저장되는 Netezza 데이터베이스에 대한 연결 문자열입니다. 다음 양식의 구조화 특성: ['odbc' '<dsn>' '<username>' '<psw>' '<catname>' '<conn_attribs>' [true false]] 여기서, <dsn>은 데이터 소스 이름 <username> 및 <psw>는 데이터베이스에 대한 사용자 이름 및 비밀번호 <catname>은 카탈로그 이름 <conn_attribs>는 연결 속성 true false는 비밀번호가 필요한지 여부를 표시합니다.
table_name	string	모델이 저장될 데이터베이스 테이블의 이름입니다.
use_model_name	flag	True인 경우 모델의 이름으로 model_name에 의해 지정된 이름, 그렇지 않으면 모델 이름이 시스템에 의해 작성됩니다.
model_name	string	새 모델의 사용자 정의 이름입니다.
include_input_fields	flag	True인 경우 모든 입력 필드를 다운스트림 전달하고, 그렇지 않으면 record_id 및 모델에 의해 생성된 필드만 전달합니다.

Netezza 의사결정 트리

다음 특성은 netezzadectreenode 유형의 노드에 사용 가능합니다.

표 209. netezzadectreenode 특성

netezzadectreenode 특성	값	특성 설명
impurity_measure	Entropy Gini	트리를 분할할 최상의 장소를 평가하는데 사용하는 불순도의 측정입니다.
max_tree_depth	integer	트리가 성장할 수 있는 최대 수준 수입니다. 기본값은 62(최대 가능값)입니다.
min_improvement_splits	number	분할이 발생할 불순도의 최소 개선도입니다. 기본값은 0.01입니다.
min_instances_split	integer	분할이 발생할 수 있기 전에 남은 분할되지 않은 최소 레코드 수입니다. 기본값은 2(최소 가능값)입니다.
weights	structured	클래스에 대한 상대값 가중치입니다. 다음 양식의 구조화 특성: set :netezza_dectree.weights = [[drugA 0.3][drugB 0.6]] 기본값은 모든 클래스에 대해 1의 가중치입니다.
pruning_measure	Acc wAcc	기본값은 Acc(정확도)입니다. 선택적 wAcc(가중 정확도)는 가지치기를 적용하는 중에 클래스 가중값을 고려합니다.

표 209. netezzadectreenode 특성 (계속)

netezzadectreenode 특성	값	특성 설명
prune_tree_options	allTrainingData partitionTrainingData useOtherTable	기본값은 allTrainingData를 사용하여 모델 정확도를 평가하는 것입니다. partitionTrainingData를 사용하여 사용할 훈련 데이터의 퍼센트를 지정하거나, useOtherTable을 사용하여 지정된 데이터베이스 테이블의 훈련 데이터 세트를 사용하십시오.
perc_training_data	number	prune_tree_options가 partitionTrainingData로 설정되는 경우, 훈련에 사용할 데이터의 퍼센트를 지정합니다.
prune_seed	integer	prune_tree_options가 partitionTrainingData로 설정될 때 분석 결과 복제에 사용할 난수 시드입니다. 기본값은 1입니다.
pruning_table	string	모델 정확도 평가를 위한 별도의 가지치기 데이터 세트의 테이블 이름입니다.
compute_probabilities	flag	True인 경우 신뢰수준(확률)뿐 아니라 예측 필드를 생성합니다.

Netezza K-평균

다음 특성은 netezzakmeansnode 유형의 노드에 사용 가능합니다.

표 210. netezzakmeansnode 특성

netezzakmeansnode 특성	값	특성 설명
distance_measure	Euclidean Manhattan Canberra maximum	데이터 점 사이의 거리 측정에 사용할 방법입니다.
num_clusters	integer	작성될 군집 수이며, 기본값은 3입니다.
max_iterations	integer	그 뒤에 모델 훈련을 중지할 알고리즘 반복의 수이며, 기본값은 5입니다.
rand_seed	integer	분석 결과 복제에 사용할 난수 시드이며, 기본값은 12345입니다.

Netezza Bayes Net

다음 특성은 netezزابayesnode 유형의 노드에 사용 가능합니다.

표 211. netezزابayesnode 특성

netezزابayesnode 특성	값	특성 설명
base_index	integer	내부 관리를 위한 첫 번째 입력 필드에 지정된 숫자 식별자이며, 기본값은 777입니다.

표 211. netezabayesnode 특성 (계속)

netezabayesnode 특성	값	특성 설명
sample_size	integer	속성 수가 매우 큰 경우 취할 표본의 크기입니다. 기본값은 10,000입니다.
display_additional_information	flag	True인 경우 메시지 대화 상자에 추가 진행률 정보를 표시합니다.
type_of_prediction	best neighbors nn-neighbors	사용할 예측 알고리즘의 유형으로, best(가장 상관된 이웃 항목), neighbors(이웃 항목의 가중 예측) 또는 nn-neighbors(널 이웃 항목이 아님)입니다.

Netezza Naive Bayes

다음 특성은 netezanaivebayesnode 유형의 노드에 사용 가능합니다.

표 212. netezanaivebayesnode 특성

netezanaivebayesnode 특성	값	특성 설명
compute_probabilities	flag	True인 경우 신뢰수준(확률)뿐 아니라 예측 필드를 생성합니다.
use_m_estimation	flag	True인 경우 추정 중에 0값 확률을 피하기 위한 m-estimation 기법을 사용합니다.

Netezza KNN

다음 특성은 netezaknnnode 유형의 노드에 사용 가능합니다.

표 213. netezaknnnode 특성

netezaknnnode 특성	값	특성 설명
weights	structured	개별 클래스에 가중값을 지정하는 데 사용하는 구조화 특성입니다. 예: set :netezaknnnode.weights = [[drugA 0.3][drugB 0.6]]
distance_measure	Euclidean Manhattan Canberra Maximum	데이터 점 사이의 거리 측정에 사용할 방법입니다.
num_nearest_neighbors	integer	특정 케이스에 대한 최근접 이웃 수로서, 기본값은 3입니다.
standardize_measurements	flag	True인 경우 거리 값을 계산하기 전에 연속형 입력 필드에 대한 측정을 표준화합니다.
use_coresets	flag	True인 경우 코어 세트 표본추출을 사용하여 큰 데이터 세트에 대한 계산을 가속화합니다.

Netezza 분열 군집

다음 특성은 netezzadivclusternode 유형의 노드에 사용 가능합니다.

표 214. netezzadivclusternode 특성

netezzadivclusternode 특성	값	특성 설명
distance_measure	Euclidean Manhattan Canberra Maximum	데이터 점 사이의 거리 측정에 사용할 방법입니다.
max_iterations	integer	모델 훈련이 중지하기 전에 수행할 알고리즘 반복의 최대 수이며, 기본값은 5입니다.
max_tree_depth	integer	데이터 세트가 소분류될 수 있는 최대 수준 수이며, 기본값은 3입니다.
rand_seed	integer	분석을 복제하는 데 사용하는 난수 시드로, 기본값은 12345입니다.
min_instances_split	integer	분할될 수 있는 최소 레코드 수로, 기본값은 5입니다.
level	integer	레코드가 스코어 지정될 계층 수준으로, 기본값은 -1입니다.

Netezza PCA

다음 특성은 netezzapcanode 유형의 노드에 사용 가능합니다.

표 215. netezzapcanode 특성

netezzapcanode 특성	값	특성 설명
center_data	flag	True(기본값)인 경우 분석 전에 데이터 센터링("평균 뺄셈"이라고도 함)을 수행합니다.
perform_data_scaling	flag	True인 경우 분석 전에 데이터 배율 조정을 수행합니다. 그렇게 하면 다른 변수가 다른 장치에서 측정될 때 분석을 덜 임의적으로 만들 수 있습니다.
force_eigensolve	flag	True인 경우 주성분을 찾는 정확성은 떨어지지만 더 빠른 방법을 사용합니다.
pc_number	integer	데이터 세트가 축소될 주성분 수이며, 기본값은 1입니다.

Netezza 회귀분석 트리

다음 특성은 netezzaregtreenode 유형의 노드에 사용 가능합니다.

표 216. netezzaregtreenode 특성

netezzaregtreenode 특성	값	특성 설명
max_tree_depth	integer	트리가 루트 노드 아래에서 성장할 수 있는 최대 수준 수이며, 기본값은 10입니다.
split_evaluation_measure	Variance	트리를 분할할 최상의 장소를 평가하는 데 사용하는 클래스 불순도 측정도이며, 기본값(및 현재 유일한 옵션)은 Variance입니다.

표 216. *netezzaregtreenode* 특성 (계속)

netezzaregtreenode 특성	값	특성 설명
min_improvement_splits	<i>number</i>	새 분할이 트리에서 작성되기 전에 불순도를 줄일 최소량입니다.
min_instances_split	<i>integer</i>	분할될 수 있는 최소 레코드 수입니다.
pruning_measure	mse r2 pearson spearman	가지치기에 사용할 방법입니다.
prune_tree_options	allTrainingData partitionTrainingData useOtherTable	기본값은 allTrainingData를 사용하여 모델 정확도를 평가하는 것입니다. partitionTrainingData를 사용하여 사용할 훈련 데이터의 퍼센트를 지정하거나, useOtherTable을 사용하여 지정된 데이터베이스 테이블의 훈련 데이터 세트를 사용하십시오.
perc_training_data	<i>number</i>	prune_tree_options가 PercTrainingData로 설정되는 경우, 훈련에 사용할 데이터의 퍼센트를 지정합니다.
prune_seed	<i>integer</i>	prune_tree_options가 PercTrainingData로 설정될 때 분석 결과 복제에 사용할 난수 시드입니다. 기본값은 1입니다.
pruning_table	<i>string</i>	모델 정확도 평가를 위한 별도의 가지치기 데이터 세트의 테이블 이름입니다.
compute_probabilities	<i>flag</i>	True인 경우 지정된 클래스의 분산이 출력에 포함되어야 함을 지정합니다.

Netezza 선형 회귀

다음 특성은 netezzalineressionnode 유형의 노드에 사용 가능합니다.

표 217. *netezzalineressionnode* 특성

netezzalineressionnode 특성	값	특성 설명
use_svd	<i>flag</i>	True인 경우 향상된 속도 및 수치 정확도를 위해 원래 교차표 대신 비정칙값 분해 교차표를 사용합니다.
include_intercept	<i>flag</i>	True(기본값)인 경우 솔루션의 전체 정확도를 늘립니다.
calculate_model_diagnostics	<i>flag</i>	True인 경우 모델에 대한 진단을 계산합니다.

Netezza 시계열

다음 특성은 netezzatimeseriesnode 유형의 노드에 사용 가능합니다.

표 218. *netezzatimeseriesnode* 특성

netezzatimeseriesnode 특성	값	특성 설명
time_points	<i>field</i>	시계열의 날짜 또는 시간 값을 포함하는 입력 필드입니다.

표 218. *netezzatimeseriesnode* 특성 (계속)

netezzatimeseriesnode 특성	값	특성 설명
time_series_ids	<i>field</i>	시계열 ID를 포함하는 입력 필드로서, 입력이 둘 이상의 시계열을 포함하는 경우에 사용됩니다.
model_table	<i>field</i>	Netezza 시계열 모델이 저장되는 데이터베이스 테이블의 이름입니다.
description_table	<i>field</i>	시계열 이름 및 설명을 포함하는 입력 테이블의 이름입니다.
seasonal_adjustment_table	<i>field</i>	지수평활 또는 계절별 추세 분해 알고리즘에 의해 계산되는 계절적으로 조정된 값이 저장되는 출력 테이블의 이름입니다.
algorithm_name	SpectralAnalysis 또는 spectral ExponentialSmoothing 또는 esmoothing ARIMA SeasonalTrendDecomposition 또는 std	시계열 모델링에 사용하는 알고리즘입니다.
trend_name	N A DA M DM	지수평활의 추세 유형: N - 없음 A - 가법 DA - 진폭감소 가법 M - 승법 DM - 진폭감소 승법
seasonality_type	N A M	지수평활에 대한 계절성 유형: N - 없음 A - 가법 M - 승법
interpolation_method	linear cubicspline exponentialspline	사용할 보간법 방법.
timerange_setting	SD SP	사용할 시간 범위 설정: SD - 시스템 판별(시계열 데이터의 전체 범위 사용) SP - earliest_time 및 latest_time을 통한 사용자 지정
earliest_time	<i>integer</i>	timerange_setting이 SP인 경우 시작 및 종료 값입니다. 형식은 time_points 값을 따라야 합니다. 예를 들어 time_points 필드가 날짜를 포함하면 이것도 날짜여야 합니다. 예: set NZ_DT1.timerange_setting = 'SP' set NZ_DT1.earliest_time = '1921-01-01' set NZ_DT1.latest_time = '2121-01-01'
latest_time	<i>date</i> <i>time</i> <i>timestamp</i>	

표 218. netezatimeseriesnode 특성 (계속)

netezatimeseriesnode 특성	값	특성 설명
arima_setting	SD SP	ARIMA 알고리즘의 설정 (algorithm_name이 ARIMA로 설정된 경우 에만 사용): SD - 시스템 판별 SP - 사용자 지정 arima_setting = SP인 경우 다음 모수 를 사용하여 계절 및 비계절 값을 설정하 십시오. 예(비계절만 해당): set NZ_DT1.algorithm_name = 'arima' set NZ_DT1.arima_setting = 'SP' set NZ_DT1.p_symbol = 'lesseq' set NZ_DT1.p = '4' set NZ_DT1.d_symbol = 'lesseq' set NZ_DT1.d = '2' set NZ_DT1.q_symbol = 'lesseq' set NZ_DT1.q = '4'
p_symbol	less	ARIMA - p, d, q, sp, sd, sq 모수에 대 한 연산자: less - 미만 eq - 같음 lesseq - 이하
d_symbol	eq	
q_symbol	lesseq	
sp_symbol		
sd_symbol		
sq_symbol		
p	integer	ARIMA - 자기상관의 비계절 차수.
q	integer	ARIMA - 비계절 파생 값.
d	integer	ARIMA - 모델에서 이동 평균 순서의 비 계절 번호.
sp	integer	ARIMA - 자기상관의 계절 차수.
sq	integer	ARIMA - 계절 파생 값.
sd	integer	ARIMA - 모델에서 이동 평균 순서의 계 절 번호.
advanced_setting	SD SP	고급 설정 처리 방법을 판별합니다. SD - 시스템 판별 SP - period, units_period 및 forecast_setting을 통한 사용자 지정. 예: set NZ_DT1.advanced_setting = 'SP' set NZ_DT1.period = 5 set NZ_DT1.units_period = 'd'
period	integer	units_period와 함께 지정되는 계절 순 환의 길이입니다. 스펙트럼 분석에는 적 용되지 않습니다.

표 218. netezzatimeseriesnode 특성 (계속)

netezzatimeseriesnode 특성	값	특성 설명
units_period	ms s min h d wk q y	period가 표현되는 단위입니다. ms - 밀리초 s - 초 min - 분 h - 시간 d - 일 wk - 주 q - 분기 y - 년 예를 들어, 주별 시계열의 경우 period에 대해 1, units_period에 대해 wk를 사용하십시오.
forecast_setting	forecasthorizon forecasttimes	예측값이 작성되는 방법을 지정합니다.
forecast_horizon	integer date time timestamp	forecast_setting = forecasthorizon인 경우 시계열 분석의 끝점을 지정합니다. 형식은 time_points 값을 따라야 합니다. 예를 들어 time_points 필드가 날짜를 포함하면 이것도 날짜여야 합니다.
forecast_times	integer date time timestamp	forecast_setting = forecasttimes인 경우 예측값 작성에 사용할 값을 지정합니다. 형식은 time_points 값을 따라야 합니다. 예를 들어 time_points 필드가 날짜를 포함하면 이것도 날짜여야 합니다.
include_history	flag	히스토리 값이 출력에 포함되는지 여부를 표시합니다.
include_interpolated_values	flag	보간된 값이 출력에 포함되는지 여부를 표시합니다. include_history가 false인 경우에는 적용되지 않습니다.

Netezza 일반화 선형

다음 특성은 netezzaglmnode 유형의 노드에 사용 가능합니다.

표 219. netezzaglmnode 특성

netezzaglmnode 특성	값	특성 설명
dist_family	bernoulli gaussian poisson negativebinomial wald gamma	분포 유형이며, 기본값은 bernoulli입니다.

표 219. netezzaglmnode 특성 (계속)

netezzaglmnode 특성	값	특성 설명
dist_params	number	사용할 분포모수 값입니다. distribution이 Negativebinomial인 경우에만 적용 가능합니다.
trials	integer	distribution이 Binomial인 경우에만 적용 가능합니다. 목표 반응이 시행 수에서 발생하는 이벤트 수일 때, target 필드에는 이벤트 수가 들어 있고 trials 필드에는 시행 수가 들어 있습니다.
model_table	field	Netezza 일반화 선형 모델이 저장되는 데이터베이스 테이블의 이름입니다.
maxit	integer	알고리즘이 수행할 최대 반복 수입니다. 기본값은 20입니다.
eps	number	알고리즘이 최적 맞춤 모델 찾기를 중지해야 하는 최대 오류 값(지수 표기법으로 표현)입니다. 기본값은 -3으로, 1E-3 또는 0.001을 의미합니다.
tol	number	그 이하에서 오류가 0의 값을 갖는 것으로 처리되는 값(지수 표기법으로 표현)입니다. 기본값은 -7로, 1E-7(또는 0.0000001) 아래의 오류 값은 무의미한 것으로 계수됨을 의미합니다.
link_func	identity inverse invnegative invsquare sqrt power oddspower log clog loglog cloglog logit probit gaussit cauchit canbinom cangeom cannegbinom	사용할 연결함수이며, 기본값은 logit입니다.
link_params	number	사용할 연결함수 모수값입니다. link_function이 power 또는 oddspower인 경우에만 적용할 수 있습니다.

표 219. *netezzaglmnode* 특성 (계속)

netezzaglmnode 특성	값	특성 설명
interaction	[[[<i>colnames1</i>],[<i>levels1</i>]], [[<i>colnames2</i>],[<i>levels2</i>]], ...,[[<i>colnamesN</i>],[<i>levelsN</i>]],]	필드 사이의 상호작용을 지정합니다. <i>colnames</i> 는 입력 필드의 목록이고, <i>level</i> 은 각 필드에 대해 항상 0입니다. 예: [[["K","BP","Sex","K"],[0,0,0,0]], [["Age","Na"],[0,0]]]
intercept	<i>flag</i>	true인 경우 절편을 모델에 포함합니다.

Netezza 모델 너깃 특성

다음 특성은 Netezza 데이터베이스 모델 너깃에 공통적입니다.

표 220. 공통 Netezza 모델 너깃 특성

공통 Netezza 모델 너깃 특성	값	특성 설명
connection	<i>string</i>	모델이 저장되는 Netezza 데이터베이스에 대한 연결 문자열입니다.
table_name	<i>string</i>	모델이 저장될 데이터베이스 테이블의 이름입니다.

기타 모델 너깃 특성은 대응하는 모델링 노드에 대한 특성과 동일합니다.

모델 너깃의 스크립트 이름은 다음과 같습니다.

표 221. Netezza 모델 너깃의 스크립트 이름

모델 너깃	스크립트 이름
의사결정 트리	applynetez zadectreenode
K-평균	applynetez zakmeansnode
Bayes Net	applynetez zabayesnode
Naive Bayes	applynetez zanaivebayesnode
KNN	applynetez zaknnnode
분열 군집	applynetez zadivclusternode
PCA	applynetez zapcanode
회귀분석 트리	applynetez zaregtreenode
선형 회귀	applynetez zalineregressionnode
시계열	applynetez zatimeseriesnode
일반화 선형	applynetez zaglmnode

제 16 장 출력 노드 특성

출력 노드 특성은 다른 노드 유형의 특성과 약간 다릅니다. 출력 노드 특성은 특정 노드 옵션을 참조하는 대신 출력 오브젝트에 참조를 저장합니다. 이것은 테이블에서 값을 가져와서 스트림 모수로 설정하기에 유용합니다.

이 절에서는 출력 노드에 사용할 수 있는 스크립팅 특성을 설명합니다.

analysisnode 특성



분석 노드는 정확한 예측을 생성하기 위한 예측 모델의 능력을 평가합니다. 분석 노드는 하나 이상의 모델 너깃에 대해 예측값과 실제 값 사이의 다양한 비교를 수행합니다. 또한 예측 모델을 서로 비교할 수도 있습니다.

예제

```
node = stream.create("analysis", "My node")
# "Analysis" tab
node.setPropertyValue("coincidence", True)
node.setPropertyValue("performance", True)
node.setPropertyValue("confidence", True)
node.setPropertyValue("threshold", 75)
node.setPropertyValue("improve_accuracy", 3)
node.setPropertyValue("inc_user_measure", True)
# "Define User Measure..."
node.setPropertyValue("user_if", "@TARGET = @PREDICTED")
node.setPropertyValue("user_then", "101")
node.setPropertyValue("user_else", "1")
node.setPropertyValue("user_compute", ["Mean", "Sum"])
node.setPropertyValue("by_fields", ["Drug"])
# "Output" tab
node.setPropertyValue("output_format", "HTML")
node.setPropertyValue("full_filename", "C:/output/analysis_out.html")
```

표 222. *analysisnode* 특성.

analysisnode 특성	데이터 유형	특성 설명
output_mode	Screen File	출력 노드로부터 생성되는 출력의 대상 위치를 지정하는 데 사용됩니다.
use_output_name	<i>flag</i>	사용자 정의 출력결과 이름의 사용 여부를 지정합니다.
output_name	<i>string</i>	use_output_name이 true인 경우 사용할 이름을 지정합니다.

표 222. *analysisnode* 특성 (계속).

analysisnode 특성	데이터 유형	특성 설명
output_format	Text(.txt) HTML(.html) Output(.cou)	출력의 유형을 지정하는 데 사용됩니다.
by_fields	list	
full_filename	string	디스크, 데이터 또는 HTML 출력인 경우, 출력 파일의 이름입니다.
coincidence	flag	
performance	flag	
evaluation_binary	flag	
confidence	flag	
threshold	number	
improve_accuracy	number	
inc_user_measure	flag	
user_if	expr	
user_then	expr	
user_else	expr	
user_compute	[Mean Sum Min Max SDev]	

dataauditnode 특성



데이터 검토 노드는 요약 통계량, 각 필드에 대한 히스토그램과 분포뿐 아니라 이상값, 결측값, 극단값에 대한 정보를 포함하여 데이터에 대한 포괄적인 정보를 간략하게 제공합니다. 결과는 전체 크기 그래프 및 데이터 준비 노드를 생성하기 위해 정렬하고 사용할 수 있는 읽기 쉬운 행렬로 표시됩니다.

예제

```

filenode = stream.createAt("variablefile", "File", 100, 100)
filenode.setPropertyValue("full_filename", "$CLEO_DEMOS/DRUG1n")
node = stream.createAt("dataaudit", "My node", 196, 100)
stream.link(filenode, node)
node.setPropertyValue("custom_fields", True)
node.setPropertyValue("fields", ["Age", "Na", "K"])
node.setPropertyValue("display_graphs", True)
node.setPropertyValue("basic_stats", True)
node.setPropertyValue("advanced_stats", True)
node.setPropertyValue("median_stats", False)
node.setPropertyValue("calculate", ["Count", "Breakdown"])
node.setPropertyValue("outlier_detection_method", "std")
node.setPropertyValue("outlier_detection_std_outlier", 1.0)
node.setPropertyValue("outlier_detection_std_extreme", 3.0)
node.setPropertyValue("output_mode", "Screen")
    
```

표 223. dataauditnode 특성.

dataauditnode 특성	데이터 유형	특성 설명
custom_fields	flag	
fields	[field1 ... fieldN]	
overlay	field	
display_graphs	flag	출력 행렬에서 그래프의 표시를 켜거나 끄는 데 사용합니다.
basic_stats	flag	
advanced_stats	flag	
median_stats	flag	
calculate	Count Breakdown	결측값을 계산하는 데 사용합니다. either, both 또는 neither 계산 방법을 선택하십시오.
outlier_detection_method	std iqr	이상값 및 극단값에 대한 발견 방법을 지정하는 데 사용합니다.
outlier_detection_std_outlier	number	outlier_detection_method가 std인 경우, 이상값을 정의하는 데 사용하는 숫자를 지정합니다.
outlier_detection_std_extreme	number	outlier_detection_method가 std인 경우, 극단값을 정의하는 데 사용하는 숫자를 지정합니다.
outlier_detection_iqr_outlier	number	outlier_detection_method가 iqr인 경우, 이상값을 정의하는 데 사용하는 숫자를 지정합니다.
outlier_detection_iqr_extreme	number	outlier_detection_method가 iqr인 경우, 극단값을 정의하는 데 사용하는 숫자를 지정합니다.
use_output_name	flag	사용자 정의 출력결과 이름의 사용 여부를 지정합니다.
output_name	string	use_output_name이 true인 경우 사용할 이름을 지정합니다.
output_mode	Screen File	출력 노드로부터 생성되는 출력의 대상 위치를 지정하는 데 사용합니다.
output_format	Formatted(.tab) Delimited(.csv) HTML(.html) Output(.cou)	출력의 유형을 지정하는 데 사용합니다.
paginate_output	flag	output_format이 HTML일 때 출력이 페이지로 구분되게 만듭니다.
lines_per_page	number	paginate_output과 함께 사용할 때 출력의 페이지당 행 수를 지정합니다.
full_filename	string	

extensionoutputnode 특성



확장 출력 노드를 사용하면 데이터 및 사용자 고유의 사용자 정의 R 또는 Python for Spark 스크립트를 사용한 모델 스코어링의 결과를 분석할 수 있습니다. 분석 결과는 텍스트나 그래픽일 수 있습니다. 결과는 관리자 분할창의 출력 탭에 추가되거나, 출력이 파일로 경로 재지정될 수 있습니다.

Python for Spark 예제

```
#### script example for Python for Spark
import modeler.api
stream = modeler.script.stream()
node = stream.create("extension_output", "extension_output")
node.setPropertyValue("syntax_type", "Python")

python_script = """
import json
import spss.pyspark.runtime

cxt = spss.pyspark.runtime.getContext()
df = cxt.getSparkInputData()
schema = df.dtypes[:]
print df
"""

node.setPropertyValue("python_syntax", python_script)
```

R 예제

```
#### script example for R
node.setPropertyValue("syntax_type", "R")
node.setPropertyValue("r_syntax", "print(modelerData$Age)")
```

표 224. extensionoutputnode 특성

extensionoutputnode 특성	데이터 유형	특성 설명
syntax_type	R Python	실행할 스크립트, R 또는 Python을 지정하십시오(R이 기본값).
r_syntax	string	모델 스코어링의 R 스크립팅 구문
python_syntax	string	모델 스코어링을 위한 Python 스크립팅 구문입니다.
convert_flags	StringsAndDoubles LogicalValues	플래그 필드를 변환하는 옵션.
convert_missing	flag	결측값을 R NA 값으로 변환하는 옵션입니다.
convert_datetime	flag	날짜 또는 날짜/시간 형식을 갖는 변수를 R 날짜/시간 형식으로 변환하는 옵션.
convert_datetime_class	POSIXct POSIX1t	날짜 또는 날짜/시간 형식을 갖는 변수를 변환할 형식을 지정하는 옵션.

표 224. *extensionoutputnode* 특성 (계속)

extensionoutputnode 특성	데이터 유형	특성 설명
output_to	Screen File	출력 유형(Screen 또는 File)을 지정하십시오.
output_type	Graph Text	그래픽 또는 텍스트 출력 생성 여부를 지정합니다.
full_filename	<i>string</i>	생성된 출력에 사용할 파일 이름입니다.
graph_file_type	HTML COU	출력 파일의 파일 유형입니다(.html 또는 .cou).
text_file_type	HTML TEXT COU	텍스트 출력의 파일 유형을 지정하십시오(.html, .txt 또는 .cou).

matrixnode 특성



행렬 노드는 필드 사이의 관계를 표시하는 테이블을 작성합니다. 두 기호 필드 사이의 관계를 표시하기 위해 가장 일반적으로 사용하지만, 플래그 필드나 수치 필드 사이의 관계도 표시할 수 있습니다.

예제

```
node = stream.create("matrix", "My node")
# "Settings" tab
node.setPropertyValue("fields", "Numerics")
node.setPropertyValue("row", "K")
node.setPropertyValue("column", "Na")
node.setPropertyValue("cell_contents", "Function")
node.setPropertyValue("function_field", "Age")
node.setPropertyValue("function", "Sum")
# "Appearance" tab
node.setPropertyValue("sort_mode", "Ascending")
node.setPropertyValue("highlight_top", 1)
node.setPropertyValue("highlight_bottom", 5)
node.setPropertyValue("display", ["Counts", "Expected", "Residuals"])
node.setPropertyValue("include_totals", True)
# "Output" tab
node.setPropertyValue("full_filename", "C:/output/matrix_output.html")
node.setPropertyValue("output_format", "HTML")
node.setPropertyValue("paginate_output", True)
node.setPropertyValue("lines_per_page", 50)
```

표 225. *matrixnode* 특성.

matrixnode 특성	데이터 유형	특성 설명
fields	Selected Flags Numerics	

표 225. *matrixnode* 특성 (계속).

matrixnode 특성	데이터 유형	특성 설명
row	<i>field</i>	
column	<i>field</i>	
include_missing_values	<i>flag</i>	사용자 결측(공백) 및 시스템 결측(널) 값이 행 및 열 출력에 포함되는지 여부를 지정합니다.
cell_contents	CrossTabs Function	
function_field	<i>string</i>	
function	Sum Mean Min Max SDev	
sort_mode	Unsorted Ascending Descending	
highlight_top	<i>number</i>	0이 아니면 true입니다.
highlight_bottom	<i>number</i>	0이 아니면 true입니다.
display	[Counts Expected Residuals RowPct ColumnPct TotalPct]	
include_totals	<i>flag</i>	
use_output_name	<i>flag</i>	사용자 정의 출력결과 이름의 사용 여부를 지정합니다.
output_name	<i>string</i>	use_output_name이 true인 경우 사용할 이름을 지정합니다.
output_mode	Screen File	출력 노드로부터 생성되는 출력의 대상 위치를 지정하는 데 사용합니다.
output_format	Formatted(<i>.tab</i>) Delimited(<i>.csv</i>) HTML(<i>.html</i>) Output(<i>.cou</i>)	출력의 유형을 지정하는 데 사용됩니다. Formatted 및 Delimited 형식 둘 다 수정자 transposed를 취할 수 있는데, 이것은 테이블의 행과 열을 전치시킵니다.
paginate_output	<i>flag</i>	output_format이 HTML일 때 출력이 페이지로 구분되게 만듭니다.
lines_per_page	<i>number</i>	paginate_output과 함께 사용할 때 출력의 페이지당 행 수를 지정합니다.
full_filename	<i>string</i>	

meansnode 특성



평균 노드는 독립 집단 사이 또는 관련된 필드의 쌍 사이의 평균을 비교하여 상당한 차이가 존재하는지 여부를 검정합니다. 예를 들어, 프로모션을 실행하기 전후의 평균 수익을 비교하거나 프로모션을 받지 않은 고객과 받은 고객으로부터의 수익을 비교할 수 있습니다.

예제

```
node = stream.create("means", "My node")
node.setPropertyValue("means_mode", "BetweenFields")
node.setPropertyValue("paired_fields", [["OPEN_BAL", "CURR_BAL"]])
node.setPropertyValue("label_correlations", True)
node.setPropertyValue("output_view", "Advanced")
node.setPropertyValue("output_mode", "File")
node.setPropertyValue("output_format", "HTML")
node.setPropertyValue("full_filename", "C:/output/means_output.html")
```

표 226. meansnode 특성.

meansnode 특성	데이터 유형	특성 설명
means_mode	BetweenGroups BetweenFields	데이터에 대해 실행될 평균 통계의 유형을 지정합니다.
test_fields	[field1 ... fieldn]	means_mode가 BetweenGroups로 설정될 때 검정 필드를 지정합니다.
grouping_field	field	집단 필드를 지정합니다.
paired_fields	[[field1 field2] [field3 field4] ...]	means_mode가 BetweenFields로 설정될 때 사용할 필드 쌍을 지정합니다.
label_correlations	flag	상관관계 레이블이 출력에 표시되는지 여부를 지정합니다. 이 설정은 means_mode가 BetweenFields로 설정될 때만 적용됩니다.
correlation_mode	Probability Absolute	상관계수를 확률 또는 절대값으로 레이블할지 여부를 지정합니다.
weak_label	string	
medium_label	string	
strong_label	string	
weak_below_probability	number	correlation_mode가 Probability로 설정될 때, 약한 상관관계에 대한 절사 값을 지정합니다. 이것은 0과 1 사이의 값이어야 합니다(예: 0.90).
strong_above_probability	number	강한 상관관계에 대한 절사 값입니다.
weak_below_absolute	number	correlation_mode가 Absolute로 설정될 때, 약한 상관관계에 대한 절사 값을 지정합니다. 이것은 0과 1 사이의 값이어야 합니다(예: 0.90).
strong_above_absolute	number	강한 상관관계에 대한 절사 값입니다.

표 226. *meansnode* 특성 (계속).

meansnode 특성	데이터 유형	특성 설명
unimportant_label	string	
marginal_label	string	
important_label	string	
unimportant_below	number	낮은 필드 중요도에 대한 절사 값입니다. 이것은 0과 1 사이의 값이어야 합니다(예: 0.90).
important_above	number	
use_output_name	flag	사용자 정의 출력결과 이름의 사용 여부를 지정합니다.
output_name	string	사용할 이름.
output_mode	Screen File	출력 노드로부터 생성되는 출력의 목표 위치를 지정합니다.
output_format	Formatted(.tab) Delimited(.csv) HTML(.html) Output(.cou)	출력의 유형을 지정합니다.
full_filename	string	
output_view	Simple Advanced	출력에 단순 또는 고급 보기가 표시되는지 여부를 지정합니다.

reportnode 특성



보고서 노드는 고정 텍스트뿐 아니라 데이터 및 데이터로부터 파생된 기타 표현식을 포함한 형식화된 보고서를 작성합니다. 텍스트 템플릿을 사용하여 보고서의 형식을 지정하여 고정 텍스트 및 데이터 출력 생성을 정의합니다. 템플릿에서 HTML 태그를 사용하고 출력 탭에서 옵션을 설정하여 사용자 정의 텍스트 형식화를 제공할 수 있습니다. 템플릿에서 CLEM 표현식을 사용하여 데이터 값과 기타 조건부 출력을 포함할 수 있습니다.

예제

```
node = stream.create("report", "My node")
node.setPropertyValue("output_format", "HTML")
node.setPropertyValue("full_filename", "C:/report_output.html")
node.setPropertyValue("lines_per_page", 50)
node.setPropertyValue("title", "Report node created by a script")
node.setPropertyValue("highlights", False)
```

표 227. *reportnode* 특성.

reportnode 특성	데이터 유형	특성 설명
output_mode	Screen File	출력 노드로부터 생성되는 출력의 대상 위치를 지정하는 데 사용합니다.
output_format	HTML(.html) Text(.txt) Output(.cou)	파일 출력의 유형을 지정하는 데 사용합니다.

표 227. reportnode 특성 (계속).

reportnode 특성	데이터 유형	특성 설명
format	Auto Custom	출력이 자동으로 형식화되거나 템플릿에 포함된 HTML을 사용하여 형식화되도록 선택하는 데 사용됩니다. 템플릿에서 HTML 형식화를 사용하려면 Custom을 지정하십시오.
use_output_name	flag	사용자 정의 출력결과 이름의 사용 여부를 지정합니다.
output_name	string	use_output_name이 true인 경우 사용할 이름을 지정합니다.
text	string	
full_filename	string	
highlights	flag	
title	string	
lines_per_page	number	

rouputnode 특성



R 출력 노드를 사용하면 데이터 및 사용자 고유의 사용자 정의 R 스크립트를 사용한 모델 스코어링의 결과를 분석할 수 있습니다. 분석 결과는 텍스트나 그래픽일 수 있습니다. 결과는 관리자 분할창의 출력 탭에 추가되거나, 출력이 파일로 경로 재 지정될 수 있습니다.

표 228. rouputnode 특성

rouputnode 특성	데이터 유형	특성 설명
syntax	string	
convert_flags	StringsAndDoubles LogicalValues	
convert_datetime	flag	
convert_datetime_class	POSIXct POSIXlt	
convert_missing	flag	
output_name	Auto Custom	
custom_name	string	
output_to	Screen File	
output_type	Graph Text	
full_filename	string	
graph_file_type	HTML COU	

표 228. *rouputnode* 특성 (계속)

rouputnode 특성	데이터 유형	특성 설명
text_file_type	HTML TEXT COU	

setglobalsnode 특성



전역값 설정 노드는 데이터를 스캔하고 CLEM 표현식에서 사용할 수 있는 요약 값을 계산합니다. 예를 들어, 이 노드를 사용하여 *age*라는 필드에 대한 통계량을 계산한 후 @GLOBAL_MEAN(*age*) 함수를 삽입하여 CLEM 표현식에서 *age*의 전체 평균을 사용할 수 있습니다.

예제

```
node = stream.create("setglobals", "My node")
node.setKeyedPropertyValue("globals", "Na", ["Max", "Sum", "Mean"])
node.setKeyedPropertyValue("globals", "K", ["Max", "Sum", "Mean"])
node.setKeyedPropertyValue("globals", "Age", ["Max", "Sum", "Mean", "SDev"])
node.setPropertyValue("clear_first", False)
node.setPropertyValue("show_preview", True)
```

표 229. *setglobalsnode* 특성.

setglobalsnode 특성	데이터 유형	특성 설명
globals	[Sum Mean Min Max SDev]	설정될 필드가 다음 구문으로 참조되어야 하는 구조화된 특성: node.setKeyedPropertyValue("globals", "Age", ["Max", "Sum", "Mean", "SDev"])
clear_first	flag	
show_preview	flag	

simevalnode 특성



시뮬레이션 평가 노드는 지정된 예측 목표 필드를 평가하고 목표 필드에 관한 분포 및 상관관계 정보를 제공합니다.

표 230. *simevalnode* 특성.

simevalnode 특성	데이터 유형	특성 설명
target	field	
iteration	field	
presorted_by_iteration	boolean	

표 230. *simevalnode* 특성 (계속).

simevalnode 특성	데이터 유형	특성 설명
max_iterations	number	
tornado_fields	[field1...fieldN]	
plot_pdf	boolean	
plot_cdf	boolean	
show_ref_mean	boolean	
show_ref_median	boolean	
show_ref_sigma	boolean	
num_ref_sigma	number	
show_ref_pct	boolean	
ref_pct_bottom	number	
ref_pct_top	number	
show_ref_custom	boolean	
ref_custom_values	[number1...numberN]	
category_values	Category Probabilities Both	
category_groups	Categories Iterations	
create_pct_table	boolean	
pct_table	Quartiles Intervals Custom	
pct_intervals_num	number	
pct_custom_values	[number1...numberN]	

simfitnode 특성



시뮬레이션 적합 노드는 각 필드에 있는 데이터의 통계 분포를 분석하고 각 필드에 최상의 적합 분포가 지정된 시뮬레이션 생성 노드를 생성(또는 업데이트)합니다. 시뮬레이션 생성 노드를 사용하여 시뮬레이션된 데이터를 생성할 수 있습니다.

표 231. *simfitnode* 특성.

simfitnode 특성	데이터 유형	특성 설명
build	Node XMLExport Both	
use_source_node_name	boolean	
source_node_name	string	생성 또는 업데이트되고 있는 소스 노드의 사용자 정의 이름입니다.
use_cases	All LimitFirstN	

표 231. *simfitnode* 특성 (계속).

simfitnode 특성	데이터 유형	특성 설명
use_case_limit	<i>integer</i>	
fit_criterion	AndersonDarling KolmogorovSmirnov	
num_bins	<i>integer</i>	
parameter_xml_filename	<i>string</i>	
generate_parameter_import	<i>boolean</i>	

statisticsnode 특성



통계량 노드는 수치 필드에 관한 기본 요약 정보를 제공합니다. 개별 필드에 대한 요약 통계량 및 필드 사이의 상관계수를 계산합니다.

예제

```
node = stream.create("statistics", "My node")
# "Settings" tab
node.setPropertyValue("examine", ["Age", "BP", "Drug"])
node.setPropertyValue("statistics", ["mean", "sum", "sdev"])
node.setPropertyValue("correlate", ["BP", "Drug"])
# "Correlation Labels..." section
node.setPropertyValue("label_correlations", True)
node.setPropertyValue("weak_below_absolute", 0.25)
node.setPropertyValue("weak_label", "lower quartile")
node.setPropertyValue("strong_above_absolute", 0.75)
node.setPropertyValue("medium_label", "middle quartiles")
node.setPropertyValue("strong_label", "upper quartile")
# "Output" tab
node.setPropertyValue("full_filename", "c:/output/statistics_output.html")
node.setPropertyValue("output_format", "HTML")
```

표 232. *statisticsnode* 특성.

statisticsnode 특성	데이터 유형	특성 설명
use_output_name	<i>flag</i>	사용자 정의 출력결과 이름의 사용 여부를 지정합니다.
output_name	<i>string</i>	use_output_name이 true인 경우 사용할 이름을 지정합니다.
output_mode	Screen File	출력 노드로부터 생성되는 출력의 대상 위치를 지정하는 데 사용됩니다.
output_format	Text(<i>.txt</i>) HTML(<i>.html</i>) Output(<i>.cou</i>)	출력의 유형을 지정하는 데 사용됩니다.
full_filename	<i>string</i>	

표 232. *statisticsnode* 특성 (계속).

statisticsnode 특성	데이터 유형	특성 설명
examine	<i>list</i>	
correlate	<i>list</i>	
statistics	[count mean sum min max range variance sdev semean median mode]	
correlation_mode	Probability Absolute	상관계수를 확률 또는 절대값으로 레이블할지 여부를 지정합니다.
label_correlations	<i>flag</i>	
weak_label	<i>string</i>	
medium_label	<i>string</i>	
strong_label	<i>string</i>	
weak_below_probability	<i>number</i>	correlation_mode가 Probability로 설정될 때, 약한 상관관계에 대한 절사 값을 지정합니다. 이것은 0과 1 사이의 값이어야 합니다(예: 0.90).
strong_above_probability	<i>number</i>	강한 상관관계에 대한 절사 값입니다.
weak_below_absolute	<i>number</i>	correlation_mode가 Absolute로 설정될 때, 약한 상관관계에 대한 절사 값을 지정합니다. 이것은 0과 1 사이의 값이어야 합니다(예: 0.90).
strong_above_absolute	<i>number</i>	강한 상관관계에 대한 절사 값입니다.

statisticsoutputnode 특성



통계량 출력 노드를 사용하면 IBM SPSS Statistics 프로시저를 호출하여 IBM SPSS Modeler 데이터를 분석할 수 있습니다. 광범위한 IBM SPSS Statistics 분석 프로시저를 사용할 수 있습니다. 이 노드는 IBM SPSS Statistics의 사용권 사본이 필요합니다.

이 노드의 특성은 351 페이지의 『statisticsoutputnode 특성』에서 설명됩니다.

tablenode 특성



테이블 노드는 데이터를 표 형식으로 표시하는데, 이것을 파일에 쓸 수도 있습니다. 이것은 쉽게 읽을 수 있는 양식으로 데이터 값을 조사하거나 내보내야 할 때 유용합니다.

예제

```

node = stream.create("table", "My node")
node.setPropertyValue("highlight_expr", "Age > 30")
node.setPropertyValue("output_format", "HTML")
node.setPropertyValue("transpose_data", True)
node.setPropertyValue("full_filename", "C:/output/table_output.htm")
node.setPropertyValue("paginate_output", True)
node.setPropertyValue("lines_per_page", 50)

```

표 233. *tablenode* 특성.

tablenode 특성	데이터 유형	특성 설명
full_filename	<i>string</i>	디스크, 데이터 또는 HTML 출력인 경우, 출력 파일의 이름입니다.
use_output_name	<i>flag</i>	사용자 정의 출력결과 이름의 사용 여부를 지정합니다.
output_name	<i>string</i>	use_output_name이 true인 경우 사용할 이름을 지정합니다.
output_mode	Screen File	출력 노드로부터 생성되는 출력의 대상 위치를 지정하는 데 사용됩니다.
output_format	Formatted(.tab) Delimited(.csv) HTML(.html) Output(.cou)	출력의 유형을 지정하는 데 사용됩니다.
transpose_data	<i>flag</i>	행이 필드를 나타내고 열이 레코드를 나타내도록 내보내기 전에 데이터를 전치합니다.
paginate_output	<i>flag</i>	output_format이 HTML일 때 출력이 페이지로 구분되게 만듭니다.
lines_per_page	<i>number</i>	paginate_output과 함께 사용할 때 출력의 페이지당 행 수를 지정합니다.
highlight_expr	<i>string</i>	
output	<i>string</i>	노드가 작성한 마지막 테이블에 대한 참조를 보유하는 읽기 전용 특성입니다.
value_labels	[[Value LabelString] [Value LabelString] ...]	값 쌍에 대한 레이블을 지정하는 데 사용됩니다.
display_places	<i>integer</i>	표시될 때 필드에 대한 소수점 이하 자리수를 설정합니다(REAL 저장 공간을 갖는 필드에만 적용됨). 값 -1은 스트림 기본값을 사용합니다.
export_places	<i>integer</i>	내보낼 때 필드에 대한 소수점 이하 자리수를 설정합니다(REAL 저장 공간을 갖는 필드에만 적용됨). 값 -1은 스트림 기본값을 사용합니다.
decimal_separator	DEFAULT PERIOD COMMA	필드의 소수점 구분자를 설정합니다(REAL 저장 공간을 갖는 필드에만 적용됨).

표 233. *tablenode* 특성 (계속).

tablenode 특성	데이터 유형	특성 설명
date_format	"DDMMYY" "MMDDYY" "YYMMDD" "YYYYMMDD" "YYYYDDD" DAY MONTH "DD-MM-YY" "DD-MM-YYYY" "MM-DD-YY" "MM-DD-YYYY" "DD-MON-YY" "DD-MON-YYYY" "YYYY-MM-DD" "DD.MM.YY" "DD.MM.YYYY" "MM.DD.YYYY" "DD.MON.YY" "DD.MON.YYYY" "DD/MM/YY" "DD/MM/YYYY" "MM/DD/YY" "MM/DD/YYYY" "DD/MON/YY" "DD/MON/YYYY" MON YYYY q Q YYYY ww WK YYYY	필드의 날짜 형식을 설정합니다(DATE 또는 TIMESTAMP 저장 공간을 갖는 필드에만 적용 됨).
time_format	"HHMMSS" "HHMM" "MMSS" "HH:MM:SS" "HH:MM" "MM:SS" "(H)H:(M)M:(S)S" "(H)H:(M)M" "(M)M:(S)S" "HH.MM.SS" "HH.MM" "MM.SS" "(H)H.(M)M.(S)S" "(H)H.(M)M" "(M)M.(S)S"	필드의 시간 형식을 설정합니다(TIME 또는 TIMESTAMP 저장 공간을 갖는 필드에만 적용 됨).
column_width	<i>integer</i>	필드의 열 너비를 설정합니다. 값 -1은 열 너 비를 Auto로 설정합니다.
justify	AUTO CENTER LEFT RIGHT	필드의 열 맞춤을 설정합니다.

transformnode 특성



변환 노드를 사용하면 선택된 필드에 적용하기 전에 변환 결과를 선택하고 시각적으로 미리볼 수 있습니다.

예제

```
node = stream.create("transform", "My node")
node.setPropertyValue("fields", ["AGE", "INCOME"])
node.setPropertyValue("formula", "Select")
node.setPropertyValue("formula_log_n", True)
node.setPropertyValue("formula_log_n_offset", 1)
```

표 234. transformnode 특성.

transformnode 특성	데이터 유형	특성 설명
fields	[field1... fieldn]	변환에 사용할 필드입니다.
formula	All Select	모두 또는 선택된 변환이 계산되어야 하는지 여부를 표시합니다.
formula_inverse	flag	역변환을 사용할지 여부를 표시합니다.
formula_inverse_offset	number	공식에 사용할 데이터 오프셋을 표시합니다. 사용자가 지정하지 않는 한, 기본적으로 0으로 설정합니다.
formula_log_n	flag	\log_n 변환 사용 여부를 표시합니다.
formula_log_n_offset	number	
formula_log_10	flag	\log_{10} 변환 사용 여부를 표시합니다.
formula_log_10_offset	number	
formula_exponential	flag	지수 변환(e^x) 사용 여부를 표시합니다.
formula_square_root	flag	제곱근 변환 사용 여부를 표시합니다.
use_output_name	flag	사용자 정의 출력결과 이름의 사용 여부를 지정합니다.
output_name	string	use_output_name이 true인 경우 사용할 이름을 지정합니다.
output_mode	Screen File	출력 노드로부터 생성되는 출력의 대상 위치를 지정하는 데 사용됩니다.
output_format	HTML(.html) Output(.cou)	출력의 유형을 지정하는 데 사용됩니다.
paginate_output	flag	output_format이 HTML일 때 출력이 페이지로 구분되게 만듭니다.
lines_per_page	number	paginate_output과 함께 사용할 때 출력의 페이지당 행 수를 지정합니다.
full_filename	string	파일 출력에 사용할 파일 이름을 표시합니다.

제 17 장 내보내기 노드 특성

공통 내보내기 노드 특성

다음 특성은 모든 내보내기 노드에 공통적입니다.

표 235. 공통 내보내기 노드 특성

특성	값	특성 설명
publish_path	string	출판된 이미지 및 모수 파일에 사용할 루트 이름을 입력하십시오.
publish_metadata	flag	이미지 및 해당 데이터 모델의 입력 및 출력을 설명하는 메타데이터 파일이 생성되는지를 지정합니다.
publish_use_parameters	flag	스트림 모수가 *.par 파일에 포함되는지를 지정합니다.
publish_parameters	문자열 목록	포함될 모수를 지정하십시오.
execute_mode	export_data publish	노드가 스트림을 출판하지 않고 실행하는지 여부 또는 노드가 실행될 때 스트림이 자동으로 출판되는지를 지정합니다.

asexport 특성

Analytic Server 내보내기를 사용하면 HDFS(Hadoop Distributed File System)에서 스트림을 실행할 수 있습니다.

예제

```
node.setPropertyValue("use_default_as", False)
node.setPropertyValue("connection",
["false","9.119.141.141","9080","analyticserver","ibm","admin","admin","false","","",""])
```

표 236. asexport 특성.

asexport 특성	데이터 유형	특성 설명
data_source	string	데이터 소스의 이름
export_mode	string	내보낸 데이터를 기존 데이터 소스에 붙여쓰기하는지 아니면 기존 데이터 소스를 덮어쓰기하는지 여부를 지정합니다.
use_default_as	boolean	True로 설정할 경우 서버 options.cfg 파일에 구성된 기본 Analytic Server 연결이 사용됩니다. False로 설정할 경우 해당 노드의 연결이 사용됩니다.

표 236. *asexport* 특성 (계속).

asexport 특성	데이터 유형	특성 설명
connection	["string","string","string", "string","string","string","string", "string","string","string", "string", ,"string"]	Analytic Server 연결 세부사항이 포함된 목록 특성입니다. 형식은 ["is_secure_connect", "server_url", "server_port", "context_root", "consumer", "user_name", "password", "use-kerberos-auth", "kerberos-krb5-config-file-path", "kerberos-jaas-config-file-path", "kerberos-krb5-service-principal-name", "enable-kerberos-debug"]입니다. 여기서 is_secure_connect:는 보안 연결 사용 여부를 나타내며, true 또는 false입니다. use-kerberos-auth:는 Kerberos 인증 사용 여부를 나타내며, true 또는 false입니다. enable-kerberos-debug:는 Kerberos 인증의 디버그 모드 사용 여부를 나타내며, true 또는 false입니다.

cognosexportnode 특성



IBM Cognos 내보내기 노드는 Cognos 데이터베이스가 읽을 수 있는 형식으로 데이터를 내보냅니다.

이 노드의 경우 Cognos 연결 및 ODBC 연결을 정의해야 합니다.

Cognos 연결

Cognos 연결을 위한 특성은 다음과 같습니다.

표 237. *cognosexportnode* 특성

cognosexportnode 특성	데이터 유형	특성 설명
cognos_connection	<code>["string","flag","string","string","string"]</code>	<p>Cognos 서버에 대한 연결 세부사항이 포함된 목록 특성입니다. 형식은 다음과 같습니다. <code>["Cognos_server_URL", login_mode, "namespace", "username", "password"]</code> 여기서, <code>Cognos_server_URL</code>은 소스를 포함하는 Cognos 서버의 URL입니다. <code>login_mode</code>는 익명 로그인을 사용하는지 여부를 표시하며, <code>true</code> 또는 <code>false</code>입니다. <code>true</code>인 경우 다음 필드가 ""로 설정되어야 합니다.</p> <p><code>namespace</code>는 서버에 로그인하는 데 사용하는 보안 인증 제공자를 지정합니다. <code>username</code> 및 <code>password</code>는 Cognos 서버에 로그인하는 데 사용하는 것입니다. <code>login_mode</code> 대신, 다음 모드도 사용할 수 있습니다.</p> <ul style="list-style-type: none"> <code>anonymousMode</code>. 예: <code>['Cognos_server_url', 'anonymousMode', "namespace", "username", "password"]</code> <code>credentialMode</code>. 예: <code>['Cognos_server_url', 'credentialMode', "namespace", "username", "password"]</code> <code>storedCredentialMode</code>. 예: <code>['Cognos_server_url', 'storedCredentialMode', "stored_credential_name"]</code> 여기서 <code>stored_credential_name</code>은 리포지토리에 있는 Cognos 신임 정보의 이름입니다.
cognos_package_name	<i>string</i>	데이터를 내보내고 있는 Cognos 패키지의 경로와 이름입니다. 예: /Public Folders/MyPackage
cognos_datasource	<i>string</i>	
cognos_export_mode	Publish ExportFile	
cognos_filename	<i>string</i>	

ODBC 연결

ODBC 연결의 특성은 다음 절에서 databaseexportnode에 대해 나열되는 것과 동일하지만, datasource 특성이 유효하지 않은 점은 예외입니다.

databaseexportnode 특성



데이터베이스 내보내기 노드는 데이터를 ODBC 준수 관계형 데이터 소스에 기록합니다. ODBC 데이터 소스에 쓰기 위해 데이터 소스가 존재하고 사용자에게 쓰기 권한이 있어야 합니다.

예제

```
...
Assumes a datasource named "MyDatasource" has been configured
...
stream = modeler.script.stream()
db_exportnode = stream.createAt("databaseexport", "DB Export", 200, 200)
applynn = stream.findByType("applyneuralnetwork", None)
stream.link(applynn, db_exportnode)

# Export tab
db_exportnode.setPropertyValue("username", "user")
db_exportnode.setPropertyValue("datasource", "MyDatasource")
db_exportnode.setPropertyValue("password", "password")
db_exportnode.setPropertyValue("table_name", "predictions")
db_exportnode.setPropertyValue("write_mode", "Create")
db_exportnode.setPropertyValue("generate_import", True)
db_exportnode.setPropertyValue("drop_existing_table", True)
db_exportnode.setPropertyValue("delete_existing_rows", True)
db_exportnode.setPropertyValue("default_string_size", 32)

# Schema dialog
db_exportnode.setKeyedPropertyValue("type", "region", "VARCHAR(10)")
db_exportnode.setKeyedPropertyValue("export_db_primarykey", "id", True)
db_exportnode.setPropertyValue("use_custom_create_table_command", True)
db_exportnode.setPropertyValue("custom_create_table_command", "My SQL Code")

# Indexes dialog
db_exportnode.setPropertyValue("use_custom_create_index_command", True)
db_exportnode.setPropertyValue("custom_create_index_command", "CREATE BITMAP INDEX <index-name>
ON <table-name> <(index-columns)>")
db_exportnode.setKeyedPropertyValue("indexes", "MYINDEX", ["fields", ["id", "region"]])
```

표 238. databaseexportnode 특성.

databaseexportnode 특성	데이터 유형	특성 설명
datasource	string	
username	string	
password	string	

표 238. *databaseexportnode* 특성 (계속).

databaseexportnode 특성	데이터 유형	특성 설명
epassword	<i>string</i>	이 슬롯은 실행 중에 읽기 전용입니다. 인코딩된 비밀번호를 생성하려면 도구 메뉴에서 사용 가능한 비밀번호 도구를 사용하십시오. 자세한 정보는 56 페이지의 『인코딩된 비밀번호 생성』의 내용을 참조하십시오.
table_name	<i>string</i>	
write_mode	Create Append Merge	
map	<i>string</i>	스트림 필드 이름을 데이터베이스 열 이름에 맵핑합니다(write_mode가 Merge인 경우에만 유효함). 병합의 경우, 모든 필드가 내보내지기 위해서는 맵핑되어야 합니다. 데이터베이스에 존재하지 않는 필드 이름은 새 열로 추가됩니다.
key_fields	<i>list</i>	키에 사용하는 스트림 필드를 지정합니다. map 특성은 이것이 데이터베이스에서 대응하는 것을 표시합니다.
join	Database Add	
drop_existing_table	<i>flag</i>	
delete_existing_rows	<i>flag</i>	
default_string_size	<i>integer</i>	
type		스키마 유형을 설정하는 데 사용하는 구조화 특성입니다.
generate_import	<i>flag</i>	
use_custom_create_table_command	<i>flag</i>	표준 CREATE TABLE SQL 명령을 수정하려면 <i>custom_create_table</i> 슬롯을 사용하십시오.
custom_create_table_command	<i>string</i>	표준 CREATE TABLE SQL 명령 자리에 사용할 문자열 명령을 지정합니다.
use_batch	<i>flag</i>	다음 특성은 데이터베이스 대량 로드를 위한 고급 옵션입니다. Use_batch의 True 값은 데이터베이스에 대한 행별 커밋을 끕니다.
batch_size	<i>number</i>	메모리로 커밋하기 전에 데이터베이스로 보낼 레코드 수를 지정합니다.
bulk_loading	Off ODBC External	대량 로드의 유형을 지정합니다. ODBC 및 External에 대한 추가 옵션이 아래에 나열됩니다.
not_logged	<i>flag</i>	

표 238. databaseexportnode 특성 (계속).

databaseexportnode 특성	데이터 유형	특성 설명
odbc_binding	Row Column	ODBC를 통한 대량 로드에 대한 행 방향 또는 열 방향 바인딩을 지정하십시오.
loader_delimit_mode	Tab Space Other	외부 프로그램을 통한 대량 로드의 경우 구분자의 유형을 지정하십시오. 쉼표(,) 같은 구분자를 지정하려면 Other를 loader_other_delimiter 특성과 함께 선택하십시오.
loader_other_delimiter	string	
specify_data_file	flag	True 플래그는 아래의 data_file 특성을 활성화하는데, 여기에서 데이터베이스에 대량 로드할 때 작성할 파일 이름과 경로를 지정할 수 있습니다.
data_file	string	
specify_loader_program	flag	True 플래그는 아래의 loader_program 특성을 활성화하며, 여기에서 외부 로더 스크립트나 프로그램의 이름과 위치를 지정할 수 있습니다.
loader_program	string	
gen_logfile	flag	True 플래그는 아래 logfile_name을 활성화하는데, 여기에서 오류 로그를 생성할 서버에 있는 파일의 이름을 지정할 수 있습니다.
logfile_name	string	
check_table_size	flag	True 플래그는 테이블 검사가 데이터베이스 테이블 크기의 증가량이 IBM SPSS Modeler에서 내보내진 행 수에 대응하도록 보장할 수 있습니다.
loader_options	string	-comment 및 -specialdir 같은 추가 인수를 로더 프로그램에 지정하십시오.
export_db_primarykey	flag	주어진 필드가 기본 키인지 여부를 지정합니다.
use_custom_create_index_command	flag	true인 경우 모든 지수에 대한 사용자 정의 SQL을 가능하게 합니다.
custom_create_index_command	string	사용자 정의 SQL이 사용 가능할 때 지수를 작성하는 데 사용하는 SQL 명령을 지정합니다. (이 값은 아래에 표시된 대로 특정 지수에 대해 대체될 수 있습니다.)
indexes.INDEXNAME.fields		필요한 경우 지정된 지수를 작성하고 해당 지수에 포함될 필드 이름을 나열합니다.
INDEXNAME "use_custom_create_index_command"	flag	특정 지수에 대해 사용자 정의 SQL을 사용 또는 사용 안하는 데 사용합니다. 다음 표 뒤의 예를 참조하십시오.

표 238. *databaseexportnode* 특성 (계속).

databaseexportnode 특성	데이터 유형	특성 설명
INDEXNAME "custom_create_index_command"	string	지정된 지수에 사용하는 사용자 정의 SQL을 지정합니다. 다음 표 뒤의 예를 참조하십시오.
indexes.INDEXNAME.remove	flag	True인 경우 지수 세트에서 지정된 지수를 제거합니다.
table_space	string	작성될 테이블스페이스를 지정합니다.
use_partition	flag	분배 해시 필드를 사용하도록 지정합니다.
partition_field	string	분배 해시 필드의 내용을 지정합니다.

참고: 일부 데이터베이스의 경우, 데이터베이스 테이블이 압축(예를 들어, SQL에서 CREATE TABLE MYTABLE (...) COMPRESS YES;의 등가)과 함께 내보내기를 위해 작성되도록 지정할 수 있습니다. 다음과 같이 use_compression 및 compression_mode 특성이 이 기능을 지원하기 위해 제공됩니다.

표 239. 압축 기능을 사용하는 *databaseexportnode* 특성.

databaseexportnode 특성	데이터 유형	특성 설명
use_compression	Boolean	True로 설정되면 압축하여 내보낼 테이블을 작성합니다.
compression_mode	Row Page	SQL Server 데이터베이스의 압축 수준을 설정합니다.
	Default Direct_Load_Operations All_Operations Basic OLTP Query_High Query_Low Archive_High Archive_Low	Oracle 데이터베이스의 압축 수준을 설정합니다. 값 OLTP, Query_High, Query_Low, Archive_High, Archive_Low는 최소 Oracle 11gR2가 필요합니다.

특정 지수에 대해 CREATE INDEX 명령을 변경하는 방법을 보여주는 예:

```
db_exportnode.setKeyedPropertyValue("indexes", "MYINDEX", ["use_custom_create_index_command", True])db_exportnode.setKeyedPropertyValue("indexes", "MYINDEX", ["custom_create_index_command", "CREATE BITMAP INDEX <index-name> ON <table-name> <(index-columns)>"])
```

다른 방법으로는, 해시 테이블을 통해 이를 수행할 수 있습니다.

```
db_exportnode.setKeyedPropertyValue("indexes", "MYINDEX", ["fields":["id", "region"], "use_custom_create_index_command":True, "custom_create_index_command":"CREATE INDEX <index-name> ON <table-name> <(index-columns)>"])
```

datacollectionexportnode 특성



Data Collection 내보내기 노드는 Data Collection 시장 조사 소프트웨어에서 사용하는 형식으로 데이터를 출력합니다. 이 노드를 사용하려면 Data Collection 데이터 라이브러리가 설치되어야 합니다.

예제

```
stream = modeler.script.stream()
datacollectionexportnode = stream.createAt("datacollectionexport", "Data Collection", 200, 200)
datacollectionexportnode.setPropertyValue("metadata_file", "c:\\museums.mdd")
datacollectionexportnode.setPropertyValue("merge_metadata", "Overwrite")
datacollectionexportnode.setPropertyValue("casedata_file", "c:\\museumdata.sav")
datacollectionexportnode.setPropertyValue("generate_import", True)
datacollectionexportnode.setPropertyValue("enable_system_variables", True)
```

표 240. datacollectionexportnode 특성

datacollectionexportnode 특성	데이터 유형	특성 설명
metadata_file	string	내보낼 메타데이터 파일의 이름입니다.
merge_metadata	Overwrite MergeCurrent	
enable_system_variables	flag	내보내진 .mdd 파일이 Data Collection 시스템 변수를 포함해야 하는지 여부를 지정합니다.
casedata_file	string	케이스 데이터가 내보내진 .sav 파일의 이름입니다.
generate_import	flag	

excelexportnode 특성



Excel 내보내기 노드는 데이터를 Microsoft Excel .xlsx 파일 형식으로 출력합니다. (선택사항)노드가 실행될 때 Excel을 자동으로 시작하고 내보내진 파일을 열도록 선택할 수 있습니다.

예제

```
stream = modeler.script.stream()
excelexportnode = stream.createAt("excelexport", "Excel", 200, 200)
excelexportnode.setPropertyValue("full_filename", "C:/output/myexport.xlsx")
excelexportnode.setPropertyValue("excel_file_type", "Excel2007")
excelexportnode.setPropertyValue("inc_field_names", True)
excelexportnode.setPropertyValue("inc_labels_as_cell_notes", False)
excelexportnode.setPropertyValue("launch_application", True)
excelexportnode.setPropertyValue("generate_import", True)
```

표 241. *excelexportnode* 특성

excelexportnode 특성	데이터 유형	특성 설명
full_filename	<i>string</i>	
excel_file_type	Excel2007	
export_mode	Create Append	
inc_field_names	<i>flag</i>	필드 이름이 워크시트의 첫 번째 행에 포함되는지 여부를 지정합니다.
start_cell	<i>string</i>	내보내기 시작 셀을 지정합니다.
worksheet_name	<i>string</i>	작성될 워크시트의 이름입니다.
launch_application	<i>flag</i>	Excel이 결과 파일에서 호출되는지 여부를 지정합니다. Excel을 시작하기 위한 경로가 헬퍼 애플리케이션 대화 상자(도구 메뉴, 헬퍼 애플리케이션)에서 지정되어야 합니다.
generate_import	<i>flag</i>	내보낸 데이터 파일을 읽을 Excel 가져오기 노드가 생성되어야 하는지 여부를 지정합니다.

extensionexportnode 특성



확장 내보내기 노드를 사용하면 R 또는 Python for Spark 스크립트를 실행하여 데이터를 내보낼 수 있습니다.

Python for Spark 예제

```
#### script example for Python for Spark
import modeler.api
stream = modeler.script.stream()
node = stream.create("extension_export", "extension_export")
node.setPropertyValue("syntax_type", "Python")

python_script = """import spss.pyspark.runtime
from pyspark.sql import SQLContext
from pyspark.sql.types import *

cxt = spss.pyspark.runtime.getContext()
df = cxt.getSparkInputData()
print df.dtypes[:]
_newDF = df.select("Age","Drug")
print _newDF.dtypes[:]

df.select("Age", "Drug").write.save("c:/data/ageAndDrug.json", format="json")
```

```
"""
```

```
node.setPropertyValue("python_syntax", python_script)
```

R 예제

```
#### script example for R
node.setPropertyValue("syntax_type", "R")
node.setPropertyValue("r_syntax", ""write.csv(modelerData, "C:/export.csv)""")
```

표 242. *extensionexportnode* 특성

extensionexportnode 특성	데이터 유형	특성 설명
syntax_type	R Python	실행할 스크립트, R 또는 Python을 지정하십시오(R이 기본값).
r_syntax	string	실행할 R 스크립팅 구문입니다.
python_syntax	string	실행할 Python 스크립팅 구문입니다.
convert_flags	StringsAndDoubles LogicalValues	플래그 필드를 변환하는 옵션.
convert_missing	flag	결측값을 R NA 값으로 변환하는 옵션입니다.
convert_datetime	flag	날짜 또는 날짜/시간 형식을 갖는 변수를 R 날짜/시간 형식으로 변환하는 옵션.
convert_datetime_class	POSIXct POSIXlt	날짜 또는 날짜/시간 형식을 갖는 변수를 변환할 형식을 지정하는 옵션.

outputfilenode 특성



플랫 파일 내보내기 노드는 데이터를 구분된 텍스트 파일로 출력합니다. 다른 분석 또는 스프레드시트 소프트웨어가 읽을 수 있는 데이터 내보내기에 유용합니다.

예제

```
stream = modeler.script.stream()
outputfile = stream.createAt("outputfile", "File Output", 200, 200)
outputfile.setPropertyValue("full_filename", "c:/output/flatfile_output.txt")
outputfile.setPropertyValue("write_mode", "Append")
outputfile.setPropertyValue("inc_field_names", False)
outputfile.setPropertyValue("use_newline_after_records", False)
outputfile.setPropertyValue("delimit_mode", "Tab")
outputfile.setPropertyValue("other_delimiter", ",")
outputfile.setPropertyValue("quote_mode", "Double")
outputfile.setPropertyValue("other_quote", "*")
outputfile.setPropertyValue("decimal_symbol", "Period")
outputfile.setPropertyValue("generate_import", True)
```

표 243. *outputfilenode* 특성

outputfilenode 특성	데이터 유형	특성 설명
full_filename	string	출력 파일의 이름.
write_mode	Overwrite Append	
inc_field_names	flag	
use_newline_after_records	flag	
delimit_mode	Comma Tab Space Other	
other_delimiter	char	
quote_mode	None Single Double Other	
other_quote	flag	
generate_import	flag	
encoding	StreamDefault SystemDefault "UTF-8"	

sasexportnode 특성



SAS 내보내기 노드는 SAS 또는 SAS 호환 가능한 소프트웨어 패키지로 읽어들이기 위해 데이터를 SAS 형식으로 출력합니다. SAS for Windows/OS2, SAS for UNIX 또는 SAS 버전 7/8의 세 가지 SAS 파일 형식이 사용 가능합니다.

예제

```
stream = modeler.script.stream()
sasexportnode = stream.createAt("sasexport", "SAS Export", 200, 200)
sasexportnode.setPropertyValue("full_filename", "c:/output/SAS_output.sas7bdat")
sasexportnode.setPropertyValue("format", "SAS8")
sasexportnode.setPropertyValue("export_names", "NamesAndLabels")
sasexportnode.setPropertyValue("generate_import", True)
```

표 244. *sasexportnode* 특성

sasexportnode 특성	데이터 유형	특성 설명
format	Windows UNIX SAS7 SAS8	변량 특성 레이블 필드

표 244. sasexportnode 특성 (계속)

sasexportnode 특성	데이터 유형	특성 설명
full_filename	string	
export_names	NamesAndLabels NamesAsLabels	내보내기 시 IBM SPSS Modeler의 필드 이름을 IBM SPSS Statistics 또는 SAS 변수 이름에 맵핑하는 데 사용됩니다.
generate_import	flag	

statisticsexportnode 특성



통계량 내보내기 노드는 IBM SPSS Statistics .sav 또는 .zsav 형식으로 데이터를 출력합니다. .sav 또는 .zsav 파일은 IBM SPSS Statistics Base 및 기타 제품에서 읽을 수 있습니다. 이것은 또한 IBM SPSS Modeler의 캐시 파일에 사용하는 형식입니다.

이 노드의 특성은 351 페이지의 『statisticsexportnode 특성』에서 설명됩니다.

tm1odataexport 노드 특성



IBM Cognos TM1 내보내기 노드는 Cognos TM1 데이터베이스가 읽을 수 있는 형식으로 데이터를 내보냅니다.

표 245. tm1odataexport 노드 특성

tm1odataexport 노드 특성	데이터 유형	특성 설명
admin_host	string	REST API의 호스트 이름에 대한 URL입니다.
server_name	string	admin_host에서 선택된 TM1 서버의 이름입니다.
credential_type	inputCredential 또는 storedCredential	신임 정보 유형을 표시하는 데 사용됩니다.
input_credential	list	credential_type이 inputCredential이면 도메인 이름, 사용자 이름 및 비밀번호를 지정하십시오.
stored_credential_name	string	credential_type이 storedCredential이면 C&DS 서버에서 신임 정보의 이름을 지정하십시오.
selected_cube	field	데이터를 내보내고 있는 큐브의 이름입니다. 예: TM1_export.setPropertyValue("selected_cube", "plan_BudgetPlan")

표 245. tm1odataexport 노드 특성 (계속)

tm1odataexport 노드 특성	데이터 유형	특성 설명
spss_field_to_tm1_element_mapping	list	<p>맵핑될 tm1 요소가 선택된 큐브 보기에 대한 열 차원의 일부여야 합니다. 형식은 다음과 같습니다. <code>[[[Field_1, Dimension_1, False], [Element_1, Dimension_2, True], ...], [[Field_2, ExistMeasureElement, False], [Field_3, NewMeasureElement, True], ...]]</code></p> <p>두 개의 목록을 통해 맵핑 정보를 설명합니다. 아래 예제 2에서는 리프 요소를 차원에 맵핑하는 것을 설명합니다.</p> <p>예제 1: 첫 번째 목록, (<code>[[Field_1, Dimension_1, False], [Element_1, Dimension_2, True], ...]</code>)는 TM1 차원 맵 정보에 사용됩니다.</p> <p>각 3개의 값 목록은 차원 맵핑 정보를 표시합니다. 세 번째 부울 값은 차원의 요소를 선택하는지 나타냅니다.</p> <p>예제: "<code>[Field_1, Dimension_1, False]</code>"는 Field_1이 Dimension_1에 맵핑되어 있으며, "<code>[Element_1, Dimension_2, True]</code>"는 Dimension_2에 Element_1이 선택되었음을 나타냅니다.</p> <p>예제 2: 두 번째 목록, (<code>[[Field_2, ExistMeasureElement, False], [Field_3, NewMeasureElement, True], ...]</code>)는 TM1 측정 차원 요소 맵 정보에 사용됩니다.</p> <p>각 3개의 값 목록은 측정 요소 맵핑 정보를 표시합니다. 세 번째 부울 값은 새 요소를 작성해야 함을 나타내는데 사용됩니다. "<code>[Field_2, ExistMeasureElement, False]</code>"는 Field_2가 ExistMeasureElement에 맵핑되었음을 나타내고 "<code>[Field_3, NewMeasureElement, True]</code>"는 NewMeasureElement가 selected_measure에서 선택된 측정 차원이어야 하고 Field_3이 맵핑되었음을 나타냅니다.</p>
selected_measure	string	<p>측도 차원을 지정하십시오. 예: <code>setProperty("selected_measure", "Measures")</code></p>

tm1export 노드 특성(더 이상 사용되지 않음)



IBM Cognos TM1 내보내기 노드는 Cognos TM1 데이터베이스가 읽을 수 있는 형식으로 데이터를 내보냅니다.

참고: 이 노드는 Modeler 18.0에서는 더 이상 사용되지 않습니다. 대체 노드 스크립트 이름은 *tm1odataexport*입니다.

표 246. *tm1export* 노드 특성.

tm1export 노드 특성	데이터 유형	특성 설명
pm_host	<i>string</i>	참고: 버전 16.0 및 17.0에만 해당 호스트 이름입니다. 예: <code>TM1_export.setPropertyValue("pm_host", 'http://9.191.86.82:9510/pmhub/pm')</code>
tm1_connection	<i>["field","field", ... ,"field"]</i>	참고: 버전 16.0 및 17.0에만 해당 TM1 서버에 대한 연결 세부사항이 포함된 목록 특성입니다. 형식은 ["TM1_Server_Name", "tm1_username", "tm1_password"]입니다. 예: <code>TM1_export.setPropertyValue("tm1_connection", ['Planning Sample', "admin" "apple"])</code>
selected_cube	<i>field</i>	데이터를 내보내고 있는 큐브의 이름입니다. 예: <code>TM1_export.setPropertyValue("selected_cube", "plan_BudgetPlan")</code>

표 246. tmlexport 노드 특성 (계속).

tmlexport 노드 특성	데이터 유형	특성 설명
spssfield_tmlelement_mapping	list	<p>맵핑될 tm1 요소가 선택된 큐브 보기에 대한 열 차원의 일부여야 합니다. 형식은 다음과 같습니다.</p> <pre>[[[Field_1, Dimension_1, False], [Element_1, Dimension_2, True], ...], [[Field_2, ExistMeasureElement, False], [Field_3, NewMeasureElement, True], ...]]</pre> <p>두 개의 목록을 통해 맵핑 정보를 설명합니다. 아래 예제 2에서는 리프 요소를 차원에 맵핑하는 것을 설명합니다.</p> <p>예제 1: 첫 번째 목록, ([[Field_1, Dimension_1, False], [Element_1, Dimension_2, True], ...])는 TM1 차원 맵 정보에 사용됩니다.</p> <p>각 3개의 값 목록은 차원 맵핑 정보를 표시합니다. 세 번째 부울 값은 차원의 요소를 선택하는지 나타냅니다.</p> <p>예제: "[Field_1, Dimension_1, False]"는 Field_1이 Dimension_1에 맵핑되어 있으며, "[Element_1, Dimension_2, True]"는 Dimension_2에 Element_1이 선택되었음을 나타냅니다.</p> <p>예제 2: 두 번째 목록, ([[Field_2, ExistMeasureElement, False], [Field_3, NewMeasureElement, True], ...]) 는 TM1 측정 차원 요소 맵 정보에 사용됩니다.</p> <p>각 3개의 값 목록은 측정 요소 맵핑 정보를 표시합니다. 세 번째 부울 값은 새 요소를 작성해야 함을 나타내는데 사용됩니다. "[Field_2, ExistMeasureElement, False]"는 Field_2가 ExistMeasureElement에 맵핑되었음을 나타내고 "[Field_3, NewMeasureElement, True]"는 NewMeasureElement가 selected_measure에서 선택된 측정 차원이어야 하고 Field_3이 맵핑되었음을 나타냅니다.</p>
selected_measure	string	<p>측도 차원을 지정하십시오.</p> <p>예: <code>setPropertyValue("selected_measure", "Measures")</code></p>

xmlexportnode 특성



XML 내보내기 노드는 데이터를 XML 형식의 파일로 출력합니다. 선택적으로 XML 소스 노드를 작성하여 내보내진 데이터를 다시 스트림으로 읽을 수 있습니다.

예제

```

stream = modeler.script.stream()
xmlexportnode = stream.createAt("xmlexport", "XML Export", 200, 200)
xmlexportnode.setPropertyValue("full_filename", "c:/export/data.xml")
xmlexportnode.setPropertyValue("map", [{"/catalog/book/genre", "genre"}, {"/catalog/book/title", "title"}])

```

표 247. *xmlexportnode* 특성

xmlexportnode 특성	데이터 유형	특성 설명
full_filename	string	(필수) XML 내보내기 파일의 전체 경로와 파일 이름입니다.
use_xml_schema	flag	내보낸 데이터의 구조를 제어하기 위해 XML 스키마(XSD 또는 DTD 파일) 사용 여부를 지정합니다.
full_schema_filename	string	사용할 XSD 또는 DTD 파일의 전체 경로 및 파일 이름입니다. use_xml_schema가 true로 설정되는 경우 필수입니다.
generate_import	flag	내보낸 데이터 파일을 다시 스트림으로 읽는 XML 소스 노드를 생성합니다.
records	string	레코드 경계를 표시하는 XPath 표현식입니다.
map	string	필드 이름을 XML 구조에 매핑합니다.

제 18 장 IBM SPSS Statistics 노드 특성

statisticsimportnode 특성



통계량 파일 노드는 IBM SPSS Statistics에서 사용하는 *.sav* 또는 *.zsav* 파일 형식뿐 아니라 동일한 형식을 사용하는 IBM SPSS Modeler에 저장된 캐시 파일로부터 데이터를 읽습니다.

예제

```
stream = modeler.script.stream()
statisticsimportnode = stream.createAt("statisticsimport", "SAV Import", 200, 200)
statisticsimportnode.setPropertyValue("full_filename", "C:/data/drugIn.sav")
statisticsimportnode.setPropertyValue("import_names", True)
statisticsimportnode.setPropertyValue("import_data", True)
```

표 248. *statisticsimportnode* 특성.

statisticsimportnode 특성	데이터 유형	특성 설명
full_filename	<i>string</i>	경로를 포함한 완전한 파일 이름입니다.
password	<i>string</i>	비밀번호입니다. password 모수는 file_encrypted 모수 전에 설정되어야 합니다.
file_encrypted	<i>flag</i>	파일이 비밀번호로 보호되는지 여부입니다.
import_names	NamesAndLabels LabelsAsNames	변수 이름 및 레이블 처리 방법
import_data	DataAndLabels LabelsAsData	값과 레이블 처리 방법
use_field_format_for_storage	<i>Boolean</i>	가져올 때 IBM SPSS Statistics 필드 형식 정보를 사용할지 여부를 지정합니다.

statistictransformnode 특성



통계량 변환 노드는 IBM SPSS Modeler의 데이터 소스에 대해 IBM SPSS Statistics 구문 명령문의 선택을 실행합니다. 이 노드는 IBM SPSS Statistics의 사용권 사본이 필요합니다.

예제

```

stream = modeler.script.stream()
statisticstransformnode = stream.createAt("statisticstransform", "Transform", 200, 200)
statisticstransformnode.setPropertyValue("syntax", "COMPUTE NewVar = Na + K.")
statisticstransformnode.setKeyedPropertyValue("new_name", "NewVar", "Mixed Drugs")
statisticstransformnode.setPropertyValue("check_before_saving", True)

```

표 249. *statisticstransformnode* 특성

statisticstransformnode 특성	데이터 유형	특성 설명
syntax	<i>string</i>	
check_before_saving	<i>flag</i>	항목을 저장하기 전에 입력된 구문을 검증합니다. 구문이 유효하지 않은 경우 오류 메시지를 표시합니다.
default_include	<i>flag</i>	자세한 정보는 150 페이지의 『 <i>filternode</i> 특성』의 내용을 참조하십시오.
include	<i>flag</i>	자세한 정보는 150 페이지의 『 <i>filternode</i> 특성』의 내용을 참조하십시오.
new_name	<i>string</i>	자세한 정보는 150 페이지의 『 <i>filternode</i> 특성』의 내용을 참조하십시오.

statisticsmodelnode 특성



통계량 모델 노드를 사용하면 PMML을 생성하는 IBM SPSS Statistics 프로시저를 실행하여 데이터를 분석하고 작업할 수 있습니다. 이 노드는 IBM SPSS Statistics의 사용권 사본이 필요합니다.

예제

```

stream = modeler.script.stream()
statisticsmodelnode = stream.createAt("statisticsmodel", "Model", 200, 200)
statisticsmodelnode.setPropertyValue("syntax", "COMPUTE NewVar = Na + K.")
statisticsmodelnode.setKeyedPropertyValue("new_name", "NewVar", "Mixed Drugs")

```

statisticsmodelnode 특성	데이터 유형	특성 설명
syntax	<i>string</i>	
default_include	<i>flag</i>	자세한 정보는 150 페이지의 『 <i>filternode</i> 특성』의 내용을 참조하십시오.
include	<i>flag</i>	자세한 정보는 150 페이지의 『 <i>filternode</i> 특성』의 내용을 참조하십시오.

statisticsmodelnode 특성	데이터 유형	특성 설명
new_name	string	자세한 정보는 150 페이지의 『filternode 특성』의 내용을 참조하십시오.

statisticsoutputnode 특성



통계량 출력 노드를 사용하면 IBM SPSS Statistics 프로시저를 호출하여 IBM SPSS Modeler 데이터를 분석할 수 있습니다. 광범위한 IBM SPSS Statistics 분석 프로시저를 사용할 수 있습니다. 이 노드는 IBM SPSS Statistics의 사용권 사본이 필요합니다.

예제

```
stream = modeler.script.stream()
statisticsoutputnode = stream.createAt("statisticsoutput", "Output", 200, 200)
statisticsoutputnode.setPropertyValue("syntax", "SORT CASES BY Age(A) Sex(A) BP(A) Cholesterol(A)")
statisticsoutputnode.setPropertyValue("use_output_name", False)
statisticsoutputnode.setPropertyValue("output_mode", "File")
statisticsoutputnode.setPropertyValue("full_filename", "Cases by Age, Sex and Medical History")
statisticsoutputnode.setPropertyValue("file_type", "HTML")
```

표 250. statisticsoutputnode 특성

statisticsoutputnode 특성	데이터 유형	특성 설명
mode	Dialog Syntax	"IBM SPSS Statistics 대화 상자" 옵션이나 명령문 편집기를 선택하십시오.
syntax	string	
use_output_name	flag	
output_name	string	
output_mode	Screen File	
full_filename	string	
file_type	HTML SPV SPW	

statisticsexportnode 특성



통계량 내보내기 노드는 IBM SPSS Statistics .sav 또는 .zsav 형식으로 데이터를 출력합니다. .sav 또는 .zsav 파일은 IBM SPSS Statistics Base 및 기타 제품에서 읽을 수 있습니다. 이것은 또한 IBM SPSS Modeler의 캐시 파일에 사용하는 형식입니다.

예제

```

stream = modeler.script.stream()
statisticsexportnode = stream.createAt("statisticsexport", "Export", 200, 200)
statisticsexportnode.setPropertyValue("full_filename", "c:/output/SPSS_Statistics_out.sav")
statisticsexportnode.setPropertyValue("field_names", "Names")
statisticsexportnode.setPropertyValue("launch_application", True)
statisticsexportnode.setPropertyValue("generate_import", True)

```

표 251. *statisticsexportnode* 특성.

statisticsexportnode 특성	데이터 유형	특성 설명
full_filename	<i>string</i>	
file_type	sav zsav	<i>sav</i> 또는 <i>zsav</i> 형식으로 파일을 저장하십시오. 예를 들어, <code>statisticsexportnode.setPropertyValue("file_type","sav")</code>
encrypt_file	<i>flag</i>	파일이 비밀번호로 보호되는지 여부입니다.
password	<i>string</i>	비밀번호입니다.
launch_application	<i>flag</i>	
export_names	NamesAndLabels NamesAsLabels	내보내기 시 IBM SPSS Modeler의 필드 이름을 IBM SPSS Statistics 또는 SAS 변수 이름에 매핑하는 데 사용합니다.
generate_import	<i>flag</i>	

제 19 장 Python 노드 특성

smotenode 특성



SMOTE(Synthetic Minority Over-sampling Technique) 노드는 불균형 데이터 세트를 처리하기 위해 초과 표본추출 알고리즘을 제공합니다. 또한 데이터 균형을 조정하기 위한 고급 방법을 제공합니다. SPSS Modeler의 SMOTE 프로세스 노드는 Python으로 구현되며, imbalanced-learn© Python 라이브러리가 필요합니다.

표 252. smotenode 특성

smotenode 특성	데이터 유형	특성 설명
target_field	field	목표 필드입니다.
sample_ratio	string	사용자 정의 비율 값을 사용으로 설정합니다. 두 개의 옵션은 자동(sample_ratio_auto) 또는 비율통계량 설정(sample_ratio_manual)입니다.
sample_ratio_value	float	비율은 다수 클래스의 표본 수 대비 소수 클래스의 표본 수입니다. 0보다 크고 1보다 작거나 같아야 합니다. 기본값은 auto입니다.
random_seed	integer	난수 생성기에 사용되는 시드입니다.
k_neighbours	integer	합성 표본을 생성하는 데 사용할 초근접 이웃 수입니다. 기본값은 5입니다.
m_neighbours	integer	소수 표본이 위험한지 판별하는 데 사용할 초근접 이웃 수입니다. 이 옵션은 SMOTE 알고리즘 유형 borderline1 및 borderline2으로만 사용으로 설정됩니다. 기본값은 10입니다.
algorithm_kind	string	SMOTE 알고리즘의 유형입니다(regular, borderline1 또는 borderline2).
use_partition	부울	true로 설정할 경우 모델 작성에 훈련 데이터만 사용합니다. 기본값은 true입니다.

xgboosttreenode 특성



XGBoost Tree©는 트리 모델을 기본 모델로 사용하는 기울기 부스팅 알고리즘의 고급 구현입니다. 부스팅 알고리즘은 약한 분류자를 반복적으로 학습한 다음 이를 강한 최종 분류자에 추가합니다. XGBoost Tree는 유연성이 매우 뛰어나 대부분의 사용자에게 유용한 여러 매개변수를 제공하므로, SPSS Modeler의 XGBoost Tree 노드에는 핵심 기능과 일반적으로 사용되는 매개변수가 표시됩니다. 이 노드는 Python으로 구현됩니다.

표 253. xgboosttreenode 특성

xgboosttreenode 특성	데이터 유형	특성 설명
TargetField	field	목표 필드입니다.

표 253. *xgboosttreenode* 특성 (계속)

xgboosttreenode 특성	데이터 유형	특성 설명
InputFields	<i>field</i>	입력 필드입니다.
treeMethod	<i>string</i>	모델 작성을 위한 트리 방법입니다. 가능한 값은 auto, exact 또는 approx이고, 기본값은 auto입니다.
numBoostRound	<i>integer</i>	모델 작성을 위한 숫자 부스트 반복 횟수입니다. 1 - 1000 범위의 값을 지정하십시오. 기본값은 10입니다.
maxDepth	<i>integer</i>	트리 성장을 위한 최대 깊이입니다. 1 이상의 값을 지정하십시오. 기본값은 6입니다.
minChildWeight	<i>Double</i>	트리 성장을 위한 최소 하위 가중치입니다. 0 이상의 값을 지정하십시오. 기본값은 1입니다.
maxDeltaStep	<i>Double</i>	트리 성장을 위한 최대 델타 단계입니다. 0 이상의 값을 지정하십시오. 기본값은 0입니다.
objectiveType	<i>string</i>	학습 작업의 목적 유형입니다. 가능한 값은 reg:linear, reg:logistic, reg:gamma, reg:tweedie, count:poisson, rank:pairwise, binary:logistic, 또는 multi입니다. 플래그 대상의 경우 binary:logistic 또는 multi만 사용할 수 있습니다. multi를 사용할 경우 multi:softmax 및 multi:softprob XGBoost 목적 유형이 점수 결과에 표시됩니다.
random_seed	<i>integer</i>	난수 시드입니다. 0 - 9999999 범위의 모든 숫자입니다. 기본값은 0입니다.
sampleSize	<i>Double</i>	과적합 제어를 위한 하위 표본입니다. 0.1 - 1.0 범위의 값을 지정하십시오. 기본값은 0.1입니다.
eta	<i>Double</i>	과적합 제어를 위한 에타입니다. 0 - 1 범위의 값을 지정하십시오. 기본값은 0.3입니다.
gamma	<i>Double</i>	과적합 제어를 위한 감마입니다. 0 이상의 숫자를 지정하십시오. 기본값은 6입니다.
colsSampleRatio	<i>Double</i>	과적합 제어를 위한 트리별 colsample입니다. 0.01 - 1 범위의 값을 지정하십시오. 기본값은 1입니다.
colsSampleLevel	<i>Double</i>	과적합 제어를 위한 레벨별 colsample입니다. 0.01 - 1 범위의 값을 지정하십시오. 기본값은 1입니다.
lambda	<i>Double</i>	과적합 제어를 위한 람다입니다. 0 이상의 숫자를 지정하십시오. 기본값은 1입니다.
alpha	<i>Double</i>	과적합 제어를 위한 알파입니다. 0 이상의 숫자를 지정하십시오. 기본값은 0입니다.
scalePosWeight	<i>Double</i>	불균형 데이터 세트를 처리하기 위한 척도 양수 가중치입니다. 기본값은 1입니다.

xgboostlinearnode 특성



XGBoost Linear[®]는 선형 모델을 기본 모델로 사용하는 기울기 부스팅 알고리즘의 고급 구현입니다. 부스팅 알고리즘은 약한 분류자를 반복적으로 학습한 다음 이를 강한 최종 분류자에 추가합니다. SPSS Modeler의 XGBoost Linear 노드는 Python으로 구현됩니다.

표 254. xgboostlinearnode 특성

xgboostlinearnode 특성	데이터 유형	특성 설명
TargetField	field	
InputFields	field	
alpha	Double	알파 선형 부스터 매개변수입니다. 0 이상의 숫자를 지정하십시오. 기본값은 0입니다.
lambda	Double	람다 선형 부스터 매개변수입니다. 0 이상의 숫자를 지정하십시오. 기본값은 1입니다.
lambdaBias	Double	람다 편향 선형 부스터 매개변수입니다. 원하는 숫자를 지정하십시오. 기본값은 0입니다.
numBoostRound	integer	모델 작성을 위한 숫자 부스트 반올림 값입니다. 1 - 1000 범위의 값을 지정하십시오. 기본값은 10입니다.
objectiveType	string	학습 작업의 목적 유형입니다. 가능한 값은 reg:linear, reg:logistic, reg:gamma, reg:tweedie, count:poisson, rank:pairwise, binary:logistic, 또는 multi입니다. 플래그 대상의 경우 binary:logistic 또는 multi만 사용할 수 있습니다. multi를 사용할 경우 multi:softmax 및 multi:softprob XGBoost 목적 유형이 점수 결과에 표시됩니다.
random_seed	integer	난수 시드입니다. 0 - 9999999 범위의 모든 숫자입니다. 기본값은 0입니다.

ocsvmnode 특성



One-Class SVM 노드에는 자율 학습 알고리즘이 사용됩니다. 이 노드는 이상 탐지에 사용할 수 있습니다. 주어진 표본 세트의 소프트 경계를 탐지하여 새 포인트를 해당 세트에 속하거나 속하지 않는 것으로 분류합니다. SPSS Modeler의 이 One-Class SVM 모델링 노드는 Python으로 구현되며, scikit-learn[®] Python 라이브러리가 필요합니다.

표 255. ocsvmnode 특성

ocsvmnode 특성	데이터 유형	특성 설명
role_use	string	사전 정의된 역할을 사용하려면 predefined를 지정하고, 사용자 정의 필드 할당을 사용하려면 custom을 지정하십시오. 기본값은 predefined입니다.

표 255. *ocsvmnode* 특성 (계속)

ocsvmnode 특성	데이터 유형	특성 설명
inputs	<i>field</i>	입력용 필드 이름 목록입니다.
splits	<i>field</i>	분할용 필드 이름 목록입니다.
use_partition	부울	true 또는 false를 지정하십시오. 기본값은 true입니다. true로 설정할 경우 모델 작성에 훈련 데이터만 사용합니다.
mode_type	<i>string</i>	모드입니다. 가능한 값은 simple 또는 expert입니다. simple을 지정할 경우 고급 탭의 모든 매개변수가 사용 안함으로 설정됩니다.
stopping_criteria	<i>string</i>	지수 표기법의 문자열입니다. 가능한 값은 1.0E-1, 1.0E-2, 1.0E-3, 1.0E-4, 1.0E-5 또는 1.0E-6입니다. 기본값은 1.0E-3입니다.
precision	<i>float</i>	회귀분석 정밀도(nu)입니다. 훈련 오차 및 지원 벡터의 분수 부분에 대한 한도입니다. 0보다 크고 1.0보다 작거나 같은 숫자를 지정하십시오. 기본값은 0.1입니다.
kernel	<i>string</i>	알고리즘에 사용할 커널 유형입니다. 가능한 값은 linear, poly, rbf, sigmoid 또는 precomputed입니다. 기본값은 rbf입니다.
enable_gamma	부울	gamma 매개변수를 사용으로 설정합니다. true 또는 false를 지정하십시오. 기본값은 true입니다.
gamma	<i>float</i>	이 매개변수는 rbf, poly 및 sigmoid 커널에 대해서만 사용으로 설정합니다. enable_gamma 매개변수가 false로 설정할 경우 이 매개변수는 auto로 설정됩니다. true로 설정할 경우 기본값은 0.1입니다.
coef0	<i>float</i>	커널 함수의 독립 항입니다. 이 매개변수는 poly 커널 및 sigmoid 커널에 대해서만 사용으로 설정됩니다. 기본값은 0.0입니다.
degree	<i>integer</i>	다항 커널 함수의 차수입니다. 이 매개변수는 poly 커널에 대해서만 사용으로 설정됩니다. 정수를 지정하십시오. 기본값은 3입니다.
shrinking	부울	축소 휴리스틱 옵션을 사용할지 여부를 지정합니다. true 또는 false를 지정하십시오. 기본값은 false입니다.
enable_cache_size	부울	cache_size 매개변수를 사용으로 설정합니다. true 또는 false를 지정하십시오. 기본값은 false입니다.
cache_size	<i>float</i>	커널 캐시의 크기(MB)입니다. 기본값은 200입니다.
enable_random_seed	부울	random_seed 매개변수를 사용으로 설정합니다. true 또는 false를 지정하십시오. 기본값은 false입니다.
random_seed	<i>integer</i>	확률 추정 데이터를 셔플링할 때 사용할 난수 시드입니다. 정수를 지정하십시오.
pc_type	<i>string</i>	평행 좌표 그래픽의 유형입니다. 가능한 옵션은 independent 또는 general입니다.
lines_amount	<i>integer</i>	그래픽에 포함될 최대 행 수입니다. 1 - 1000 범위의 정수를 지정하십시오.

표 255. *ocsvmnode* 특성 (계속)

ocsvmnode 특성	데이터 유형	특성 설명
lines_fields_custom	부울	그래프 출력에 표시할 사용자 정의 필드를 지정할 수 있는 lines_fields 매개변수를 사용하여 설정합니다. false로 설정할 경우 모든 필드가 표시됩니다. true로 설정할 경우 lines_fields 매개변수로 지정된 필드만 표시됩니다. 성능상의 이유로 최대 20개의 필드가 표시됩니다.
lines_fields	<i>field</i>	그래픽에 세로 축으로 포함될 필드 이름 목록입니다

제 20 장 슈퍼노드 특성

슈퍼노드에 특정한 특성이 다음 표에서 설명됩니다. 공통 노드 특성도 슈퍼노드에 적용됨을 참고하십시오.

표 256. 터미널 슈퍼노드 특성

특성 이름	특성 유형/값 목록	특성 설명
execute_method	Script Normal	
script	<i>string</i>	

슈퍼노드 모수

다음 일반 형식을 사용하여 슈퍼노드 모수를 작성 또는 설정하기 위해 스크립트를 사용할 수 있습니다.

```
mySuperNode.setParameterValue("minvalue", 30)
```

다음을 사용하여 모수값을 검색할 수 있습니다.

```
value mySuperNode.getParameterValue("minvalue")
```

기존 슈퍼노드 찾기

findByType() 함수를 사용하여 스트림에서 슈퍼노드를 찾을 수 있습니다.

```
source_supernode = modeler.script.stream().findByType("source_super", None)
process_supernode = modeler.script.stream().findByType("process_super", None)
terminal_supernode = modeler.script.stream().findByType("terminal_super", None)
```

캡슐화 노드의 특성 설정

슈퍼노드 내의 하위 다이어그램에 액세스하여 슈퍼노드에 캡슐화된 특정 노드에 대한 특성을 설정할 수 있습니다. 예를 들어, 데이터에서 읽을 캡슐화된 가변파일 노드를 갖는 소스 슈퍼노드가 있다고 가정합니다. 다음과 같이 하위 다이어그램에 액세스하고 관련 노드를 찾아서 읽을 파일의 이름 (full_filename 특성을 사용하여 지정됨)을 전달할 수 있습니다.

```
childDiagram = source_supernode.getChildDiagram()
varfilenode = childDiagram.findByType("variablefile", None)
varfilenode.setPropertyValue("full_filename", "c:/mydata.txt")
```

수퍼노드 작성

스크래치로부터 수퍼노드 및 그의 내용을 작성하려는 경우, 수퍼노드를 작성하고 하위 다이어그램에 액세스하고 원하는 노드를 작성하여 비슷한 방법으로 수행할 수 있습니다. 또한 수퍼노드 다이어그램 내의 노드도 입력 및/또는 출력 연결자 노드에 링크되도록 보장해야 합니다. 예를 들어, 프로세스 수퍼노드를 작성하려는 경우 다음을 수행하십시오.

```
process_supernode = modeler.script.stream().createAt("process_super", "My SuperNode", 200, 200)
childDiagram = process_supernode.getChildDiagram()
filternode = childDiagram.createAt("filter", "My Filter", 100, 100)
childDiagram.linkFromInputConnector(filternode)
childDiagram.linkToOutputConnector(filternode)
```

부록 A. 노드 이름 참조

이 절에서는 IBM SPSS Modeler에 있는 노드의 스크립팅 이름에 대한 참조를 제공합니다.

모델 너깃 이름

모델 너깃(생성된 모델이라고도 함)은 노드 및 출력 오브젝트 같이 유형에 의해 참조될 수 있습니다. 다음 목록은 모델 오브젝트 참조 이름을 나열합니다.

이들 이름은 특히 모델 팔레트(IBM SPSS Modeler 창의 우측 상단 모서리에 있는)에서 모델 너깃을 참조하는 데 사용됩니다. 스코어링 목적으로 스트림에 추가된 모델 노드를 참조하려면 apply...가 접두어로 붙는 여러 가지 이름 세트를 사용합니다. 자세한 정보는 모델 너깃 노드 특성의 내용을 참조하십시오.

참고: 정상 환경에서는 혼동을 피하기 위해 이름 및 유형 모두에 의해 모델을 참조하는 것이 권장됩니다.

표 257. 모델 너깃 이름(모델링 팔레트).

모델 이름	모델
anomalydetection	Anomaly
apriori	Apriori
autoclassifier	자동 분류자
autocluster	자동 군집
autonumeric	자동 숫자
bayesnet	베이지안 네트워크
c50	C5.0
carma	Carma
cart	C&R 트리
chaid	CHAID
coxreg	Cox 회귀
decisionlist	의사결정 목록
discriminant	판별
factor	PCA/요인
featureselection	필드선택
genlin	일반화 선형 회귀분석
glmm	GLMM
kmeans	K-평균
knn	k-최근접 이웃
kohonen	코호넨
linear	선형

표 257. 모델 너짓 이름(모델링 팔레트) (계속).

모델 이름	모델
logreg	로지스틱 회귀분석
neuralnetwork	신경망
quest	QUEST
regression	선형 회귀
sequence	순차규칙
slrm	자체 학습 응답 모델
statisticsmodel	IBM SPSS Statistics 모델
svm	지원 벡터 머신
timeseries	시계열
twostep	이단계

표 258. 모델 너짓 이름(데이터베이스 모델링 팔레트).

모델 이름	모델
db2imcluster	IBM ISW 군집
db2imlog	IBM ISW 로지스틱 회귀분석
db2imnb	IBM ISW Naive Bayes
db2imreg	IBM ISW 회귀분석
db2imtree	IBM ISW 의사결정 트리
msassoc	MS 연관 규칙
msbayes	MS Naive Bayes
mscluster	MS 군집화
mslogistic	MS 로지스틱 회귀분석
msneuralnetwork	MS 신경망
msregression	MS 선형 회귀
mssequencecluster	MS 시퀀스 군집
mstimeseries	MS 시계열
mstree	MS 의사결정 트리
netezzabayes	Netezza Bayes Net
netezzadectree	Netezza 의사결정 트리
netezzadivcluster	Netezza 분열 군집
netezzaglm	Netezza 일반화 선형
netezzakmeans	Netezza K-평균
netezzaknn	Netezza KNN
netezalineregression	Netezza 선형 회귀
netezanaivebayes	Netezza Naive Bayes
netezzapca	Netezza PCA
netezzaregtree	Netezza 회귀분석 트리
netezzatimeseries	Netezza 시계열
oraabn	Oracle 적응형 베이스
oraai	Oracle AI

표 258. 모델 너짓 이름(데이터베이스 모델링 팔레트) (계속).

모델 이름	모델
oradecisiontree	Oracle 의사결정 트리
oraglm	Oracle GLM
orakmeans	Oracle <i>k</i> -평균
oranb	Oracle Naive Bayes
oranmf	Oracle NMF
oraocluster	Oracle O-Cluster
orasvm	Oracle SVM

중복 모델 이름 피하기

스크립트를 사용하여 생성된 모델을 조작할 때 중복 모델 이름을 허용하면 불확실한 참조가 발생할 수 있음을 유의하십시오. 이를 피하기 위해 스크립팅할 때 생성된 모델에 대해 고유한 이름을 요구하는 것이 좋습니다.

중복 모델 이름에 대한 옵션을 설정하려면 다음을 수행하십시오.

1. 메뉴에서 다음을 선택하십시오.

도구 > 사용자 옵션

2. 알림 탭을 클릭하십시오.
3. 이전 모델 바꾸기를 선택하여 생성된 모델에 대한 중복 이름 지정을 제한하십시오.

스크립트 실행의 작동은 불확실한 모델 참조가 있을 때 SPSS Modeler 및 IBM SPSS Collaboration and Deployment Services 사이에 변할 수 있습니다. SPSS Modeler 클라이언트는 "이전 모델 바꾸기" 옵션을 포함하는데, 이것은 동일한 이름을 갖는 모델을 자동으로 바꿉니다(예를 들어, 스크립트가 루프를 반복하여 매번 상이한 모델을 생성하는 경우). 그러나 이 옵션은 동일한 스크립트가 IBM SPSS Collaboration and Deployment Services에서 실행될 때는 사용할 수 없습니다. 모델에 대한 불확실한 참조를 피하기 위해 각 반복에서 생성되는 모델의 이름을 바꾸거나 루프의 종료 전에 현재 모델을 지워서(예를 들어, clear generated palette 명령문 추가) 이 상황을 피할 수 있습니다.

출력 유형 이름

다음 테이블은 모든 출력 오브젝트 유형 및 이를 작성하는 노드를 나열합니다. 각 유형의 출력 오브젝트에 대해 사용 가능한 내보내기 형식의 전체 목록에 대해서는 출력 유형을 작성하는 노드에 대한 특성 설명을 참조하십시오(그래프 노드 공통 특성 및 출력 노드 특성에서 사용 가능).

표 259. 출력 오브젝트 유형 및 이를 작성하는 노드.

출력 오브젝트 유형	노드
analysisoutput	분석
collectionoutput	요약도표
dataauditoutput	데이터 검토

표 259. 출력 오브젝트 유형 및 이를 작성하는 노드 (계속).

출력 오브젝트 유형	노드
distributionoutput	분포
evaluationoutput	평가
histogramoutput	히스토그램
matrixoutput	교차표
meansoutput	평균
multiplotoutput	다중 도표
plotoutput	도표
qualityoutput	품질
reportdocumentoutput	이 오브젝트 유형은 노드에서 오지 않습니다. 프로젝트 보고서에 의해 작성되는 출력입니다.
reportoutput	보고서
statisticsprocedureoutput	Statistics 출력
statisticsoutput	통계량
tableoutput	테이블
timeplotoutput	시간 도표
weboutput	웹

부록 B. 레거시 스크립팅에서 Python 스크립팅으로 마이그레이션

레거시 스크립팅 마이그레이션 개요

이 절에서는 IBM SPSS Modeler에서 Python 및 레거시 스크립팅 사이의 차이의 요약을 제공하며, 레거시 스크립트를 Python 스크립트로 마이그레이션하는 방법에 관한 정보를 제공합니다. 이 절에서는 표준 SPSS Modeler 레거시 명령과 동등한 Python 명령의 목록을 찾습니다.

일반 차이점

레거시 스크립팅은 대부분의 계획을 OS 명령 스크립트에 의존합니다. 레거시 스크립팅은 행 지향적이며, 일부 블록 구조가 있지만(예: `if...then...else...endif` 및 `for...endfor`) 들여쓰기가 일반적으로 중요하지 않습니다.

Python 스크립팅에서는 들여쓰기가 중요하며 동일한 논리 블록에 속하는 행은 같은 수준으로 들여쓰기 되어야 합니다.

참고: Python 코드를 복사 및 붙여넣을 때 주의해야 합니다. 탭을 사용하여 들여쓰지는 행은 편집기에서 공백을 사용하여 들여쓰지는 행과 동일하게 보일 수 있습니다. 그러나 행이 동일하게 들여쓰진 것으로 간주되지 않기 때문에 Python 스크립트는 오류를 생성합니다.

스크립팅 컨텍스트

스크립팅 컨텍스트는 스크립트가 실행될 환경(예를 들어, 스크립트를 실행하는 스트림 또는 슈퍼노드)을 정의합니다. 레거시 스크립팅에서 컨텍스트는 내재적이며, 이것은 예를 들어 스트림 스크립트의 임의의 노드 참조가 해당 스크립트를 실행하는 스트림 내에 있다고 가정됨을 의미합니다.

Python 스크립팅에서 스크립팅 컨텍스트는 `modeler.script` 모듈을 통해 명시적으로 제공됩니다. 예를 들어, Python 스트림 스크립트는 다음 코드로 해당 스크립트를 실행하는 스트림에 액세스할 수 있습니다.

```
s = modeler.script.stream()
```

그런 다음 리턴된 오브젝트를 통해 관련 함수를 호출할 수 있습니다.

명령 대 함수

레거시 스크립팅은 명령 지향적입니다. 이것은 스크립트의 각 행이 일반적으로 실행될 명령으로 시작하고 모수가 그 뒤에 옵니다. 예:

```
connect 'Type':typenode to :filternode  
rename :derivenode as "Compute Total"
```

Python은 대개 해당 함수를 정의하는 오브젝트(모듈, 클래스 또는 오브젝트)를 통해 호출되는 함수를 사용합니다. 예:

```
stream = modeler.script.stream()
typenode = stream.findByType("type", "Type")
filternode = stream.findByType("filter", None)
stream.link(typenode, filternode)
derive.setLabel("Compute Total")
```

리터럴 및 주석

IBM SPSS Modeler에서 공통적으로 사용하는 일부 리터럴 및 주석 명령은 Python 스크립팅에서 동등한 명령을 갖습니다. 이것이 기존 SPSS Modeler 레거시 스크립트를 IBM SPSS Modeler 17에서 사용할 Python 스크립트로 변환하는 데 도움이 될 수 있습니다.

표 260. 리터럴 및 주석에 대한 레거시 스크립팅 대 Python 스크립팅 맵핑.

레거시 스크립팅	Python 스크립팅
정수(예: 4)	동일
Float(예: 0.003)	동일
작은따옴표 문자열(예: 'Hello')	동일 참고: ASCII가 아닌 문자를 포함하는 문자열 리터럴은 유니코드로 표시되도록 보장하기 위해 u가 접두문자로 붙어야 합니다.
큰따옴표 문자열(예: "Hello again")	동일 참고: ASCII가 아닌 문자를 포함하는 문자열 리터럴은 유니코드로 표시되도록 보장하기 위해 u가 접두문자로 붙어야 합니다.
긴 문자열(예: """This is a string that spans multiple lines""")	동일
목록(예: [1 2 3])	[1, 2, 3]
변수 참조(예: set x = 3)	x = 3
행 연속(\), 예: set x = [1 2 \ 3 4]	x = [1, 2,\n3, 4]
블록 주석(예: /* This is a long comment over a line. */	""" This is a long comment over a line. """
행 주석(예: set x = 3 # make x 3)	x = 3 # make x 3
undef	None
true	True
false	False

연산자

IBM SPSS Modeler에서 공통적으로 사용하는 일부 연산자 명령은 Python 스크립팅에서 동등한 명령을 갖습니다. 이것이 기존 SPSS Modeler 레거시 스크립트를 IBM SPSS Modeler 17에서 사용할 Python 스크립트로 변환하는 데 도움이 될 수 있습니다.

표 261. 연산자에 대한 레거시 스크립팅 대 Python 스크립팅 매핑.

레거시 스크립팅	Python 스크립팅
NUM1 + NUM2 LIST + ITEM LIST1 + LIST2	NUM1 + NUM2 LIST.append(ITEM) LIST1.extend(LIST2)
NUM1 - NUM2 LIST - ITEM	NUM1 - NUM2 LIST.remove(ITEM)
NUM1 * NUM2	NUM1 * NUM2
NUM1 / NUM2	NUM1 / NUM2
= ==	==
/= /==	!=
X ** Y	X ** Y
X < Y X <= Y X > Y X >= Y	X < Y X <= Y X > Y X >= Y
X div Y X rem Y X mod Y	X // Y X % Y X % Y
and or not(EXPR)	and or not EXPR

조건부 및 루핑

IBM SPSS Modeler에서 공통적으로 사용하는 일부 조건부 및 루핑 명령은 Python 스크립팅에서 동등한 명령을 갖습니다. 이것이 기존 SPSS Modeler 레거시 스크립트를 IBM SPSS Modeler 17에서 사용할 Python 스크립트로 변환하는 데 도움이 될 수 있습니다.

표 262. 조건부 및 루핑에 대한 레거시 스크립팅 대 Python 스크립팅 매핑.

레거시 스크립팅	Python 스크립팅
for VAR from INT1 to INT2 ... endfor	for VAR in range(INT1, INT2): ... 또는 VAR = INT1 while VAR <= INT2: ... VAR += 1

표 262. 조건부 및 루핑에 대한 레거시 스크립팅 대 Python 스크립팅 맵핑 (계속).

레거시 스크립팅	Python 스크립팅
for VAR in LIST ... endfor	for VAR in LIST: ...
for VAR in_fields_to NODE ... endfor	for VAR in NODE.getInputDataModel(): ...
for VAR in_fields_at NODE ... endfor	for VAR in NODE.getOutputDataModel(): ...
if...then ... elseif...then ... else ... endif	if ...: ... elif ...: ... else: ... endif
with TYPE OBJECT ... endwith	해당 값이 없음
var VAR1	변수 선언이 필요하지 않음

변수

레거시 스크립팅에서 변수는 참조되기 전에 선언됩니다. 예:

```
var mynode
set mynode = create typenode at 96 96
```

Python 스크립팅에서는 변수가 처음 참조될 때 작성됩니다. 예:

```
mynode = stream.createAt("type", "Type", 96, 96)
```

레거시 스크립팅에서, 변수에 대한 참조는 ^ 연산자를 사용하여 명시적으로 제거되어야 합니다. 예:

```
var mynode
set mynode = create typenode at 96 96
set ^mynode.direction."Age" = Input
```

대부분의 스크립팅 언어처럼, 이것은 Python 스크립팅에서 필요하지 않습니다. 예:

```
mynode = stream.createAt("type", "Type", 96, 96)
mynode.setKeyedPropertyValue("direction","Age","Input")
```

노드, 출력 및 모델 유형

레거시 스크립팅에서, 상이한 오브젝트 유형(노드, 출력 및 모델)은 일반적으로 오브젝트 유형에 추가된 유형을 갖습니다. 예를 들어, 파생 노드는 derivenode 유형을 갖습니다.

```
set feature_name_node = create derivenode at 96 96
```

Python의 IBM SPSS Modeler API는 node 접미문자를 포함하지 않으므로, 파생 노드는 derive 유형을 갖습니다. 예:

```
feature_name_node = stream.createAt("derive", "Feature", 96, 96)
```

레거시 및 Python 스크립팅에서 유형 이름의 유일한 차이는 유형 접미문자가 없는 것입니다.

특성 이름

특성 이름은 레거시 및 Python 스크립팅 모두에서 동일합니다. 예를 들어, 변수 파일 노드에서 파일 위치를 정의하는 특성은 두 스크립팅 환경 모두에서 full_filename입니다.

노드 참조

많은 레거시 스크립트는 내재적 검색을 사용하여 수정될 노드를 찾고 액세스합니다. 예를 들어, 다음 명령은 "Type" 레이블을 갖는 유형 노드에 대한 현재 스트림을 검색한 후, "Age" 필드의 방향(또는 모델링 역할)을 입력으로, "Drug" 필드는 목표(예측될 값)가 되도록 설정합니다.

```
set 'Type':typenode.direction."Age" = Input
set 'Type':typenode.direction."Drug" = Target
```

Python 스크립팅에서는 특성 값을 설정하기 위해 함수를 호출하기 전에 노드 오브젝트를 명시적으로 찾아야 합니다. 예:

```
typenode = stream.findByType("type", "Type")
typenode.setKeyedPropertyValue("direction", "Age", "Input")
typenode.setKeyedPropertyValue("direction", "Drug", "Target")
```

참고: 이 경우에 "Target"을 문자열 인용부호로 묶어야 합니다.

Python 스크립팅은 modeler.api 패키지의 ModelingRole 열거를 사용할 수도 있습니다.

Python 스크립팅 버전이 더 상세할 수 있지만, 노드 검색이 대개 한 번만 수행되기 때문에 런타임 성능이 더 좋습니다. 레거시 스크립팅 예에서는 노드에 대한 검색이 각 명령에 대해 수행됩니다.

ID로 노드 찾기도 지원됩니다(노드 ID는 노드 대화 상자의 주석 탭에서 볼 수 있음). 예를 들어, 레거시 스크립팅에서는

```
# id65EMPB9VL87 is the ID of a Type node
set @id65EMPB9VL87.direction."Age" = Input
```

다음 스크립트는 Python 스크립팅에서의 동일한 예를 보여줍니다.

```
typenode = stream.findById("id65EMPB9VL87")
typenode.setKeyedPropertyValue("direction", "Age", "Input")
```

특성 가져오기 및 설정

레거시 스크립팅은 set 명령을 사용하여 값을 지정합니다. set 명령 뒤에 오는 항목은 특성 정의일 수 있습니다. 다음 스크립트는 특성 설정을 위한 두 가지 가능한 스크립트 형식을 보여줍니다.

```
set <node reference>.<property> = <value>
set <node reference>.<keyed-property>.<key> = <value>
```

Python 스크립팅에서는 함수 `setProperty()` 및 `setKeyedPropertyValue()`를 사용하여 동일한 결과를 얻습니다. 예:

```
object.setPropertyValue(property, value)
object.setKeyedPropertyValue(keyed-property, key, value)
```

레거시 스크립팅에서 특성 값 액세스는 `get` 명령을 사용하여 달성할 수 있습니다. 예:

```
var n v
set n = get node :filternode
set v = ^n.name
```

Python 스크립팅에서는 `getPropertyValue()` 함수를 사용하여 동일한 결과를 달성합니다. 예:

```
n = stream.findByType("filter", None)
v = n.getPropertyValue("name")
```

스트림 편집

레거시 스크립팅에서 `create` 명령이 새 노드를 작성하는 데 사용됩니다. 예:

```
var agg select
set agg = create aggregatenode at 96 96
set select = create selectnode at 164 96
```

Python 스크립팅에서는 스트림이 노드 작성을 위한 다양한 방법을 갖고 있습니다. 예:

```
stream = modeler.script.stream()
agg = stream.createAt("aggregate", "Aggregate", 96, 96)
select = stream.createAt("select", "Select", 164, 96)
```

레거시 스크립팅에서 `connect` 명령이 노드 사이의 연결을 작성하는 데 사용됩니다. 예:

```
connect ^agg to ^select
```

Python 스크립팅에서는 `link` 메소드가 노드 사이의 연결을 작성하는 데 사용됩니다. 예:

```
stream.link(agg, select)
```

레거시 스크립팅에서 `disconnect` 명령은 노드 사이의 연결을 제거하는 데 사용됩니다. 예:

```
disconnect ^agg from ^select
```

Python 스크립팅에서는 `unlink` 메소드가 노드 사이의 연결을 제거하는 데 사용됩니다. 예:

```
stream.unlink(agg, select)
```

레거시 스크립팅에서 `position` 명령은 노드를 스트림 캔버스에 또는 다른 노드 사이에 위치시키는 데 사용됩니다. 예:

```
position ^agg at 256 256
position ^agg between ^myselect and ^mydistinct
```


Python 스크립팅에서는 `setXYPosition` 및 `setPositionBetween`의 두 메소드를 사용하여 동일한 결과를 얻습니다. 예를 들어,

```
agg.setXYPosition(256, 256)
agg.setPositionBetween(myselect, mydistinct)
```

노드 작업

IBM SPSS Modeler에서 공통적으로 사용하는 일부 노드 작업은 Python 스크립팅에서 동등한 명령을 갖습니다. 이것이 기존 SPSS Modeler 레거시 스크립트를 IBM SPSS Modeler 17에서 사용할 Python 스크립트로 변환하는 데 도움이 될 수 있습니다.

표 263. 노드 작업을 위한 레거시 스크립팅 대 Python 스크립팅 맵핑.

레거시 스크립팅	Python 스크립팅
create <i>nodespec</i> at <i>x y</i>	<code>stream.create(type, name)</code> <code>stream.createAt(type, name, x, y)</code> <code>stream.createBetween(type, name, preNode, postNode)</code> <code>stream.createModelApplier(model, name)</code>
connect <i>fromNode</i> to <i>toNode</i>	<code>stream.link(fromNode, toNode)</code>
delete <i>node</i>	<code>stream.delete(node)</code>
disable <i>node</i>	<code>stream.setEnabled(node, False)</code>
enable <i>node</i>	<code>stream.setEnabled(node, True)</code>
disconnect <i>fromNode</i> from <i>toNode</i>	<code>stream.unlink(fromNode, toNode)</code> <code>stream.disconnect(node)</code>
duplicate <i>node</i>	<code>node.duplicate()</code>
execute <i>node</i>	<code>stream.runSelected(nodes, results)</code> <code>stream.runAll(results)</code>
flush <i>node</i>	<code>node.flushCache()</code>
position <i>node</i> at <i>x y</i>	<code>node.setXYPosition(x, y)</code>
position <i>node</i> between <i>node1</i> and <i>node2</i>	<code>node.setPositionBetween(node1, node2)</code>
rename <i>node</i> as <i>name</i>	<code>node.setLabel(name)</code>

루핑

레거시 스크립팅에서 지원되는 두 가지 기본 루핑 옵션이 있습니다.

- 개수 루프에서는 지수 변수가 두 정수 경계 사이에서 이동합니다.
- 시퀀스 루프는 값의 순서를 통해 순환하여 현재 값을 루프 변수에 바인드합니다.

다음 스크립트는 레거시 스크립팅에 있는 빈도 루프의 예입니다.

```
for i from 1 to 10
  println ^i
endfor
```

다음 스크립트는 레거시 스크립팅에 있는 시퀀스 루프의 예입니다.

```

var items
set items = [a b c d]

for i in items
  println ^i
endfor

```

사용할 수 있는 다른 유형의 루프도 있습니다.

- 모델 팔레트의 모델을 통해서 또는 출력 팔레트의 출력을 통한 반복계산.
- 노드로 들어오거나 나가는 필드를 통한 반복계산.

Python 스크립팅도 여러 가지 유형의 루프를 지원합니다. 다음 스크립트는 Python 스크립팅에서 빈도 루프의 예입니다.

```

i = 1
while i <= 10:
  print i
  i += 1

```

다음 스크립트는 Python 스크립팅에서 시퀀스 루프의 예입니다.

```

items = ["a", "b", "c", "d"]
for i in items:
  print i

```

시퀀스 루프는 매우 탄력적이며, IBM SPSS Modeler API 메소드와 결합될 때 대부분의 레거시 스크립팅 유스 케이스를 지원할 수 있습니다. 다음 예는 Python 스크립팅에서 시퀀스 루프를 사용하여 노드를 나가는 필드를 통해 반복하는 방법을 보여줍니다.

```

node = modeler.script.stream().findByType("filter", None)
for column in node.getOutputDataModel().columnIterator():
  print column.getColumnname()

```

스트림 실행

스트림 실행 중에, 생성되는 모델 또는 출력 오브젝트가 오브젝트 관리자 중 하나에 추가됩니다. 레거시 스크립팅에서 스크립트는 오브젝트 관리자로부터 작성된 오브젝트를 찾거나 출력을 생성한 노드로부터 가장 최근에 생성된 출력에 액세스해야 합니다.

Python에서의 스트림 실행은 다릅니다. 실행으로부터 생성되는 모든 모델 또는 출력 오브젝트가 실행 함수로 전달되는 목록에서 리턴됩니다. 이것은 스트림 실행의 결과에 액세스하는 것을 더 쉽게 만듭니다.

레거시 스크립팅은 다음 3가지 스트림 실행 명령을 지원합니다.

- `execute_all`은 스트림의 모든 실행 가능 터미널 노드를 실행합니다.
- `execute_script`는 스크립트 실행의 설정과 상관 없이 스트림 스크립트를 실행합니다.
- `execute node`는 지정된 노드를 실행합니다.

Python 스크립팅은 비슷한 함수 세트를 지원합니다.

- `stream.runAll(results-list)`는 스트림의 모든 실행 가능 터미널 노드를 실행합니다.
- `stream.runScript(results-list)`는 스크립트 실행의 설정과 상관 없이 스트림 스크립트를 실행합니다.
- `stream.runSelected(node-array, results-list)`는 지정된 노드 세트를 적용되는 순서대로 실행합니다.
- `node.run(results-list)`는 지정된 노드를 실행합니다.

레거시 스크립트에서, 스트림 실행은 `exit` 명령을 선택적 정수 코드와 함께 사용하여 종료할 수 있습니다. 예:

```
exit 1
```

Python 스크립팅에서는 다음 스크립트로 동일한 결과를 달성할 수 있습니다.

```
modeler.script.exit(1)
```

파일 시스템 및 리포지토리를 통한 오브젝트 액세스

레거시 스크립팅에서 `open` 명령을 사용하여 기존 스트림, 모델 또는 출력 오브젝트를 열 수 있습니다. 예:

```
var s
set s = open stream "c:/my streams/modeling.str"
```

Python 스크립팅에서는 세션에서 액세스할 수 있고 비슷한 작업을 수행하는 데 사용할 수 있는 `TaskRunner` 클래스가 있습니다. 예:

```
taskrunner = modeler.script.session().getTaskRunner()
s = taskrunner.openStreamFromFile("c:/my streams/modeling.str", True)
```

레거시 스크립팅에서 오브젝트를 저장하려면 `save` 명령을 사용할 수 있습니다. 예:

```
save stream s as "c:/my streams/new_modeling.str"
```

동등한 Python 스크립트 접근 방식은 `TaskRunner` 클래스를 사용합니다. 예:

```
taskrunner.saveStreamToFile(s, "c:/my streams/new_modeling.str")
```

IBM SPSS Collaboration and Deployment Services Repository 기반 작업은 레거시 스크립팅에서 `retrieve` 및 `store` 명령을 통해 지원됩니다. 예:

```
var s
set s = retrieve stream "/my repository folder/my_stream.str"
store stream ^s as "/my repository folder/my_stream_copy.str"
```

Python 스크립팅에서는 동등한 기능이 세션과 연관된 리포지토리 오브젝트를 통해 액세스됩니다. 예:

```
session = modeler.script.session()
repo = session.getRepository()
s = repo.retrieveStream("/my repository folder/my_stream.str", None, None, True)
repo.storeStream(s, "/my repository folder/my_stream_copy.str", None)
```

참고: 리포지토리 액세스는 세션이 올바른 리포지토리 연결을 사용하여 구성되었어야 합니다.

스트림 작업

IBM SPSS Modeler에서 공통적으로 사용하는 일부 스트림 작업은 Python 스크립팅에서 동등한 명령을 갖습니다. 이것이 기존 SPSS Modeler 레거시 스크립트를 IBM SPSS Modeler 17에서 사용할 Python 스크립트로 변환하는 데 도움이 될 수 있습니다.

표 264. 스트림 작업을 위한 레거시 스크립팅 대 Python 스크립팅 맵핑.

레거시 스크립팅	Python 스크립팅
create stream <i>DEFAULT_FILENAME</i>	<i>taskrunner.createStream(name, autoConnect, autoManage)</i>
close stream	<i>stream.close()</i>
clear stream	<i>stream.clear()</i>
get stream <i>stream</i>	해당 값이 없음
load stream <i>path</i>	해당 값이 없음
open stream <i>path</i>	<i>taskrunner.openStreamFromFile(path, autoManage)</i>
save <i>stream</i> as <i>path</i>	<i>taskrunner.saveStreamToFile(stream, path)</i>
retrieve stream <i>path</i>	<i>repository.retrieveStream(path, version, label, autoManage)</i>
store <i>stream</i> as <i>path</i>	<i>repository.storeStream(stream, path, label)</i>

모델 작업

IBM SPSS Modeler에서 공통적으로 사용하는 일부 모델 작업은 Python 스크립팅에서 동등한 명령을 갖습니다. 이것이 기존 SPSS Modeler 레거시 스크립트를 IBM SPSS Modeler 17에서 사용할 Python 스크립트로 변환하는 데 도움이 될 수 있습니다.

표 265. 모델 작업을 위한 레거시 스크립팅 대 Python 스크립팅 맵핑.

레거시 스크립팅	Python 스크립팅
open model <i>path</i>	<i>taskrunner.openModelFromFile(path, autoManage)</i>
save <i>model</i> as <i>path</i>	<i>taskrunner.saveModelToFile(model, path)</i>
retrieve model <i>path</i>	<i>repository.retrieveModel(path, version, label, autoManage)</i>
store <i>model</i> as <i>path</i>	<i>repository.storeModel(model, path, label)</i>

문서 출력 작업

IBM SPSS Modeler에서 공통적으로 사용하는 일부 문서 출력 작업은 Python 스크립팅에서 동등한 명령을 갖습니다. 이것이 기존 SPSS Modeler 레거시 스크립트를 IBM SPSS Modeler 17에서 사용할 Python 스크립트로 변환하는 데 도움이 될 수 있습니다.

표 266. 문서 출력 작업을 위한 레거시 스크립팅 대 Python 스크립팅 맵핑.

레거시 스크립팅	Python 스크립팅
open output <i>path</i>	<i>taskrunner.openDocumentFromFile(path, autoManage)</i>
save <i>output</i> as <i>path</i>	<i>taskrunner.saveDocumentToFile(output, path)</i>

표 266. 문서 출력 작업을 위한 레거시 스크립팅 대 Python 스크립팅 매핑 (계속).

레거시 스크립팅	Python 스크립팅
retrieve output <i>path</i>	<code>repository.retrieveDocument(path, version, label, autoManage)</code>
store output as <i>path</i>	<code>repository.storeDocument(output, path, label)</code>

레거시 스크립팅과 Python 스크립트 사이의 기타 차이

레거시 스크립트는 IBM SPSS Modeler 프로젝트 조작을 위한 지원을 제공합니다. Python 스크립팅은 현재 이를 지원하지 않습니다.

레거시 스크립팅은 상태 오브젝트(스트림 및 모델의 조합) 로드를 위한 일부 지원을 제공합니다. 상태 오브젝트는 IBM SPSS Modeler 8.0 이후 더 이상 사용하지 않습니다. Python 스크립팅은 상태 오브젝트를 지원하지 않습니다.

Python 스크립팅은 레거시 스크립팅에서 사용할 수 없는 다음의 추가 기능을 제공합니다.

- 클래스 및 함수 정의
- 오류 처리
- 더 정교한 입/출력(I/O) 지원
- 외부 및 써드파티 모듈

주의사항

이 정보는 미국에서 제공되는 제품 및 서비스용으로 작성된 것입니다. 이 자료는 IBM에서 다른 언어로 사용 가능합니다. 그러나 자료에 접근하기 위해서는 해당 언어로 된 제품 또는 제품 버전의 사본이 필요할 수 있습니다.

IBM은 다른 국가에서 이 책에 기술된 제품, 서비스 또는 기능을 제공하지 않을 수도 있습니다. 현재 사용할 수 있는 제품 및 서비스에 대한 정보는 한국 IBM 담당자에게 문의하십시오. 이 책에서 IBM 제품, 프로그램 또는 서비스를 언급했다고 해서 해당 IBM 제품, 프로그램 또는 서비스만을 사용할 수 있다는 것을 의미하지는 않습니다. IBM의 지적 재산을 침해하지 않는 한, 기능상으로 동등한 제품, 프로그램 또는 서비스를 대신 사용할 수도 있습니다. 그러나 비IBM 제품, 프로그램 또는 서비스의 운영에 대한 평가 및 검증은 사용자의 책임입니다.

IBM은 이 책에서 다루고 있는 특정 내용에 대해 특허를 보유하고 있거나 현재 특허 출원 중일 수 있습니다. 이 책을 제공한다고 해서 특허에 대한 라이선스까지 부여하는 것은 아닙니다. 라이선스에 대한 의문사항은 다음으로 문의하십시오.

07326

서울특별시 영등포구

국제금융로 10, 31FC

한국 아이.비.엠 주식회사

대표전화서비스: 02-3781-7114

2바이트(DBCS) 정보에 관한 라이선스 문의는 한국 IBM에 문의하거나 다음 주소로 서면 문의하시기 바랍니다.

Intellectual Property Licensing

Legal and Intellectual Property Law

IBM Japan Ltd.

19-21, Nihonbashi-Hakozakicho, Chuo-ku

Tokyo 103-8510, Japan

IBM은 타인의 권리 비침해, 상품성 및 특정 목적에의 적합성에 대한 묵시적 보증을 포함하여(단, 이에 한하지 않음) 묵시적이든 명시적이든 어떠한 종류의 보증 없이 이 책을 "현상태대로" 제공합니다. 일부 국가에서는 특정 거래에서 명시적 또는 묵시적 보증의 면책사항을 허용하지 않으므로, 이 사항이 적용되지 않을 수도 있습니다.

이 정보에는 기술적으로 부정확한 내용이나 인쇄상의 오류가 있을 수 있습니다. 이 정보는 주기적으로 변경되며, 변경된 사항은 최신판에 통합됩니다. IBM은 이 책에서 설명한 제품 및/또는 프로그램을 사전 통지 없이 언제든지 개선 및/또는 변경할 수 있습니다.

이 정보에서 언급되는 비IBM 웹 사이트는 단지 편의상 제공된 것으로, 어떤 방식으로든 이들 웹 사이트를 옹호하고자 하는 것은 아닙니다. 해당 웹 사이트의 자료는 본 IBM 제품 자료의 일부가 아니므로 해당 웹 사이트 사용으로 인한 위험은 사용자 본인이 감수해야 합니다.

IBM은 귀하의 권리를 침해하지 않는 범위 내에서 적절하다고 생각하는 방식으로 귀하가 제공한 정보를 사용하거나 배포할 수 있습니다.

(i) 독립적으로 작성된 프로그램과 기타 프로그램(본 프로그램 포함) 간의 정보 교환 및 (ii) 교환된 정보의 상호 이용을 목적으로 본 프로그램에 관한 정보를 얻고자 하는 라이선스 사용자는 다음 주소로 문의하십시오.

07326

서울특별시 영등포구

국제금융로 10, 31FC

한국 아이.비.엠 주식회사

대표전화서비스: 02-3781-7114

이러한 정보는 해당 조건(예를 들면, 사용료 지불 등)하에서 사용될 수 있습니다.

이 정보에 기술된 라이선스가 부여된 프로그램 및 프로그램에 대해 사용 가능한 모든 라이선스가 부여된 자료는 IBM이 IBM 기본 계약, IBM 프로그램 라이선스 계약(IPLA) 또는 이와 동등한 계약에 따라 제공한 것입니다.

인용된 성능 데이터와 고객 예제는 예시 용도로만 제공됩니다. 실제 성능 결과는 특정 구성과 운영 조건에 따라 다를 수 있습니다.

비IBM 제품에 관한 정보는 해당 제품의 공급업체, 공개 자료 또는 기타 범용 소스로부터 얻은 것입니다. IBM에서는 이러한 제품들을 테스트하지 않았으므로, 비IBM 제품과 관련된 성능의 정확성, 호환성 또는 기타 청구에 대해서는 확신할 수 없습니다. 비IBM 제품의 성능에 대한 의문사항은 해당 제품의 공급업체에 문의하십시오.

IBM의 향후 방향 또는 의도에 관한 언급은 별도의 통지없이 변경될 수 있습니다.

이 정보에는 일상의 비즈니스 운영에서 사용되는 자료 및 보고서에 대한 예제가 들어 있습니다. 이들 예제에는 개념을 가능한 완벽하게 설명하기 위하여 개인, 회사, 상표 및 제품의 이름이 사용될 수 있습니다. 이들 이름은 모두 가공의 것이며 실제 인물 또는 기업의 이름과 유사하더라도 이는 전적으로 우연입니다.

상표

IBM, IBM 로고 및 ibm.com은 전세계 여러 국가에 등록된 International Business Machines Corp.의 상표 또는 등록상표입니다. 기타 제품 및 서비스 이름은 IBM 또는 타사의 상표입니다. IBM 상표의 최신 목록은 웹 사이트(www.ibm.com/legal/copytrade.shtml)에서 "Copyright and trademark information"을 참조하십시오.

Adobe, Adobe 로고, PostScript 및 PostScript 로고는 미국 및/또는 기타 국가에서 사용되는 Adobe Systems Incorporated의 등록상표 또는 상표입니다.

Intel, Intel 로고, Intel Inside, Intel Inside 로고, Intel Centrino, Intel Centrino 로고, Celeron, Intel Xeon, Intel SpeedStep, Itanium 및 Pentium은 미국 또는 기타 국가에서 사용되는 Intel Corporation 또는 그 계열사의 상표 또는 등록상표입니다.

Linux는 미국 또는 기타 국가에서 사용되는 Linus Torvalds의 등록상표입니다.

Microsoft, Windows, Windows NT 및 Windows 로고는 미국 또는 기타 국가에서 사용되는 Microsoft Corporation의 상표입니다.

UNIX는 미국 및 기타 국가에서 사용되는 The Open Group의 등록상표입니다.

Java 및 모든 Java 기반 상표와 로고는 Oracle 및/또는 그 계열사의 상표 또는 등록상표입니다.

제품 문서의 이용 약관

다음 이용 약관에 따라 이 책을 사용할 수 있습니다.

적용성

본 이용 약관은 IBM 웹 사이트의 모든 이용 약관에 추가됩니다.

개인적 사용

모든 소유권 사항을 표시하는 경우에 한하여 귀하는 이 책을 개인적, 비상업적 용도로 복제할 수 있습니다. 귀하는 IBM의 명시적 동의 없이 본 발행물 또는 그 일부를 배포 또는 전시하거나 2차적 저작물을 만들 수 없습니다.

상업적 사용

모든 소유권 사항을 표시하는 경우에 한하여 귀하는 이 책을 귀하 기업집단 내에서만 복제, 배포 및 전시할 수 있습니다. 귀하는 귀하의 기업집단 외에서는 IBM의 명시적 동의 없이 이 책의 2차적 저작물을 만들거나 이 책 또는 그 일부를 복제, 배포 또는 전시할 수 없습니다.

권한

본 허가에서 명시적으로 부여된 경우를 제외하고, 이 책이나 이 책에 포함된 정보, 데이터, 소프트웨어 또는 기타 지적 재산권에 대한 어떠한 허가나 라이선스 또는 권한도 명시적 또는 묵시적으로 부여되지 않습니다.

IBM은 이 책의 사용이 IBM의 이익을 해친다고 판단되거나 위에서 언급된 지시사항이 준수되지 않는다고 판단하는 경우 언제든지 부여한 허가를 철회할 수 있습니다.

귀하는 미국 수출법 및 관련 규정을 포함하여 모든 적용 가능한 법률 및 규정을 철저히 준수하는 경우에만 본 정보를 다운로드, 송신 또는 재송신할 수 있습니다.

IBM은 이 책의 내용과 관련하여 아무런 보장을 하지 않습니다. 타인의 권리 비침해, 상품성 및 특정 목적에의 적합성에 대한 묵시적 보증을 포함하여 (단 이에 한하지 않음) 묵시적이든 명시적이든 어떠한 종류의 보증 없이 현 상태대로 제공합니다.

색인

[가]

가변파일 노드
특성 108
값 익명화 노드
특성 139
고유 노드
특성 119
고정 파일 노드
특성 98
교차표 노드
특성 321
구간화 노드
특성 143
구성 노드
특성 184
구조변환 노드
특성 155
구조화 특성 76
균형 노드
특성 116
그래프 노드
스크립팅 특성 171
그래프보드 노드
특성 176

[나]

내보내기 노드
노드 스크립팅 특성 333
너깃
노드 스크립팅 특성 277
노드
가져오기 37
노드 링크 35
노드 링크 해제 35
바꾸기 37
삭제 37
스크립트에서 루핑 53
이름 참조 361
정보 39
노드 스크립팅 특성 295
내보내기 노드 333
모델 너깃 277

노드 스크립팅 특성 (계속)
모델링 노드 191
노드 작성 35, 37
노드 참조 33
노드 찾기 33
특성 설정 34
노드 찾기 33
노드 횡단 37

[다]

다시 정렬 노드
특성 154
다이어그램 31
다중 도표 노드
특성 183
다중 세트 명령 76
데이터 검토 노드
특성 318
데이터 보기 소스 노드
특성 112
데이터베이스 내보내기 노드
특성 336
데이터베이스 노드
특성 90
데이터베이스 모델링 295
독립형 스크립트 1, 4, 31

[라]

랜덤 트리 모델
노드 스크립팅 특성 250, 290
로지스틱 회귀분석 모델
노드 스크립팅 특성 238, 287
루프
스크립트에서 사용 53

[마]

마이그레이션
개요 365
기타 375
노드 유형 368
노드 참조 369

마이그레이션 (계속)
루핑 371
리포지토리 373
명령 365
모델 유형 368
변수 368
스크립팅 컨텍스트 365
스트림 실행 372
스트림 편집 370
스트림, 출력, 모델 관리자 지우기 38
오브젝트 액세스 373
일반 차이점 365
출력 유형 368
특성 가져오기 369
특성 설정 369
특성 이름 369
파일 시스템 373
함수 365
맵 시각화 노드
특성 179
메소드 정의 28
명령문 21
명령행
다중 인수 72
모수 69
스크립팅 57
인수 목록 68, 70, 71, 72
IBM SPSS Modeler 실행 67
모델
스크립팅 이름 361, 363
모델 너깃
노드 스크립팅 특성 277
스크립팅 이름 361, 363
모델 오브젝트
스크립팅 이름 361, 363
모델링 노드
노드 스크립팅 특성 191
모수 6, 75, 76, 77, 79
수퍼노드 359
스크립팅 18
목록 19
문자열 19
케이스 변경 53

[바]

반복 변수
스크립트에서 루핑 11

반복 키
스크립트에서 루핑 9

베이지안 신경망 모형
노드 스크립팅 특성 201, 279

변수
스크립팅 18

변환 노드
특성 332

보고서 노드
특성 324

보안
인코딩된 비밀번호 56, 70

분석 노드
특성 317

분석 서버 소스 노드
특성 87

분포 노드
특성 173

붙여쓰기 노드
특성 115

비밀번호
스크립트에 추가 56
인코딩됨 70

[사]

사용자 입력 노드
특성 107

상속 29

생성된 모델
스크립팅 이름 361, 363

생성된 키워드 57

생성된 팔레트 명령 지우기 57

서버
명령행 인수 70

선택 노드
특성 128

선형 모델
노드 스크립팅 특성 236, 287

선형 지원 벡터 머신 모델
노드 스크립팅 특성 243, 288

선형 특성 236

선형 회귀 모형
노드 스크립팅 특성 251, 290, 291

소스 노드
특성 83

속성 정의 28

속성 추가 27

수퍼노드 75
내부에 특성 설정 359

모수 359

스크립트 1, 6, 7, 31

스크립팅 359

스트림 31

특성 359

수학적 메소드 24

숨겨진 변수 28

스크립트
루핑 7, 8

반복 변수 11

반복 키 9

저장 2

조건부 실행 7, 12

텍스트 파일에서 가져오기 2

필드 선택 12

스크립트 실행 13

스크립트 중단 13

스크립팅
개요 1, 17

공통 특성 78

구문 18, 19, 21, 22, 23, 24, 25, 26, 27, 28, 29

그래프 노드 171

다이아그램 31

독립형 스크립트 1, 31

레거시 스크립팅 366, 367, 371, 374

명령행에서 57

반복 변수 11

반복 키 9

사용자 인터페이스 2, 4, 6

사용하는 약어 76

수퍼노드 스크립트 1, 31

수퍼노드 스트림 31

수퍼노드에서 6

스트림 1, 31

스트림 실행 순서 53

시각적 루핑 7, 8

실행 13

오류 검사 57

이전 버전과의 호환성 57

조건부 실행 7, 12

중단 13

스크립팅 (계속)
출력 노드 317

컨텍스트 32

필드 선택 12

필드선택 모델 5

Python 스크립팅 366, 367, 371, 374

스크립팅 API
검색 41

글로벌 값 51

다중 스트림 52

독립형 스크립트 52

디렉토리 가져오기 41

메타데이터 42

생성된 오브젝트 액세스 45

세션 모수 47

소개 41

수퍼노드 모수 47

스트림 모수 47

예 41

오류 처리 46

스트리밍 시계열 노드
특성 134

스트리밍 시계열 모델
노드 스크립팅 특성 130

스트림
다중 세트 명령 75

루핑 7, 8

수정 35

스크립팅 1, 2, 31

실행 32

조건부 실행 7, 12

특성 79

스트림 수정 35, 37

스트림 실행 32

스트림 실행 결과 액세스 58, 63

테이블 콘텐츠 모델 58

JSON 콘텐츠 모델 61

XML 콘텐츠 모델 60

스트림 실행 순서
스크립트를 사용한 변경 53

스트림에서 루핑 7, 8

스트림의 조건부 실행 7, 12

슬롯 모수 6, 75, 78

시간 구간 노드
특성 158

시간 구성 노드
특성 186

시간 인과 모델
 노드 스크립팅 특성 261

시계열 모델
 노드 스크립팅 특성 265, 269, 292, 293

시뮬레이션 생성 노드
 특성 102

시뮬레이션 적합 노드
 특성 327

시뮬레이션 평가 노드
 특성 326

시스템
 명령행 인수 68

시퀀스 모델
 노드 스크립팅 특성 253, 291

식별자 21

신경망
 노드 스크립팅 특성 246, 289

신경망 모델
 노드 스크립팅 특성 244, 288

실행 순서
 스크립트를 사용한 변경 53

[아]

양상블 노드
 특성 148

연관 규칙 노드
 특성 194

연관 규칙 노드 너깃
 특성 278

예 23

오류 검사
 스크립팅 57

오브젝트 지향 26

요약도표 노드
 특성 172

웹 노드
 특성 188

유형 노드
 특성 164

의사결정 목록 모델
 노드 스크립팅 특성 212, 281

이단계 모델
 노드 스크립팅 특성 273, 293

이단계 AS 모델
 노드 스크립팅 특성 274, 294

이상 항목 발견 모델
 노드 스크립팅 특성 191, 277

인수
 명령 파일 72
 서버 연결 70
 시스템 68
 IBM SPSS Analytic Server
 Repository 연결 72
 IBM SPSS Collaboration and
 Deployment Services Repository 연
 결 71

인수 전달 22

인코딩된 비밀번호
 스크립트에 추가 56

일반화 선형 모델
 노드 스크립팅 특성 221, 285

[자]

자동 군집 노드
 노드 스크립팅 특성 199

자동 군집 모델
 노드 스크립팅 특성 279

자동 데이터 준비
 특성 140

자동 분류자 노드
 노드 스크립팅 특성 196

자동 분류자 모델
 노드 스크립팅 특성 278

자동 수치 모델
 노드 스크립팅 특성 200, 279

자체 학습 반응 모델
 노드 스크립팅 특성 255, 291

작업 18

재분류 노드
 특성 153

재투영 노드
 특성 154

전역값 설정 노드
 특성 326

전치 노드
 특성 162

정렬 노드
 특성 128

좌표계 재투영
 특성 154

주석 21

지리 공간적 소스 노드
 특성 101

지시 웹 노드
 특성 188

지원 벡터 머신 모델
 노드 스크립팅 특성 260, 291

[차]

채움 노드
 특성 149

최근접 이웃 모델
 노드 스크립팅 특성 233

출력 노드
 스크립팅 특성 317

출력 오브젝트
 스크립팅 이름 363

[카]

코드 블록 22

코호넨 모델
 노드 스크립팅 특성 235, 286

클래스 생성 27

클래스 정의 27

[타]

테이블 노드
 특성 329

테이블 콘텐츠 모델 58

통계량 노드
 특성 328

통합 노드
 특성 115

특성
 공통 스크립팅 78
 데이터베이스 모델링 노드 295
 슈퍼노드 359
 스크립팅 75, 76, 77, 78, 191, 277, 333
 스트림 79
 필터 노드 76
 특성 설정 34

[파]

- 파생 노드
 - 특성 146
- 파티션 노드
 - 특성 152
- 판별 모델
 - 노드 스크립팅 특성 213, 282
- 평가 노드
 - 특성 174
- 평균 노드
 - 특성 323
- 표본 노드
 - 특성 126
- 플래그
 - 다중 플래그 결합 72
 - 명령행 인수 67
- 플래그로 설정 노드
 - 특성 157
- 플랫 파일 노드
 - 특성 342
- 필드
 - 스크립팅에서 끄기 171
- 필드 다시 정렬 노드
 - 특성 154
- 필드 이름
 - 케이스 변경 53
- 필드선택 모델
 - 노드 스크립팅 특성 219, 284
 - 스크립팅 5
 - 적용 5
- 필터 노드
 - 특성 150

[하]

- 함수
 - 노드 작업 371
 - 루핑 367
 - 리터럴 366
 - 모델 작업 374
 - 문서 출력 작업 374
 - 스트림 작업 374
 - 연산자 367
 - 오브젝트 참조 366
 - 조건부 367
 - 주석 366

- 합치기 노드
 - 특성 122
- 확장 가져오기 노드
 - 특성 95
- 확장 내보내기 노드
 - 특성 341
- 확장 모델 노드
 - 노드 스크립팅 특성 215
- 확장 변환 노드
 - 특성 121
- 확장 출력 노드
 - 특성 320
- 히스토그램 노드
 - 특성 178
- 히스토리 노드
 - 특성 151

A

- aggregatenode 특성 115
- analysisnode 특성 317
- anomalydetectionnode 특성 191
- anonymizenode 특성 139
- appendnode 특성 115
- applyanomalydetectionnode 특성 277
- applyapriorinode 특성 277
- applyassociationrulesnode 특성 278
- applyautoclassifiernode 특성 278
- applyautoclusternode 특성 279
- applyautonumericnode 특성 279
- applybayesnetnode 특성 279
- applyc50node 특성 280
- applycarmanode 특성 280
- applycartnode 특성 280
- applychaidnode 특성 281
- applycoxregnode 특성 281
- applydecisionlistnode 특성 281
- applydiscriminantnode 특성 282
- applyextension 특성 282
- applyfactornode 특성 284
- applyfeatureselectionnode 특성 284
- applygeneralizedlinearnode 특성 285
- applygle 특성 286
- applyglmmnode 특성 285
- applykmeansnode 특성 286
- applyknnnode 특성 286
- applykohonenode 특성 286
- applylinearnode 특성 287
- applylinearnode 특성 287
- applylogregnode 특성 287
- applylsvmnode 특성 288
- applymslogisticnode 특성 298
- applymsneuralnetworknode 특성 298
- applymsregressionnode 특성 298
- applymssequenceclusternode 특성 298
- applymstimeseriesnode 특성 298
- applymstreenode 특성 298
- applynetzezbayesnode 특성 316
- applynetzezzadectreenode 특성 316
- applynetzezzadivclusternode 특성 316
- applynetzezzakmeansnode 특성 316
- applynetzezzaknnnode 특성 316
- applynetzezzalineregressionnode 특성 316
- applynetzezzanaivebayesnode 특성 316
- applynetzezzapcanode 특성 316
- applynetzezzaregtreenode 특성 316
- applyneuralnetnode 특성 288
- applyneuralnetworknode 특성 289
- applyocsvm 특성 289
- applyoraabnnode 특성 305
- applyoradecisiontreenode 특성 305
- applyorakmeansnode 특성 305
- applyoranbnode 특성 305
- applyoranmfnode 특성 305
- applyoraoclusternode 특성 305
- applyorasvmnode 특성 305
- applyquestnode 특성 289
- applyr 특성 290
- applyrandomtrees 특성 290
- applyregressionnode 특성 291
- applyselflearningnode 특성 291
- applysequencenode 특성 291
- applystpnode 특성 292
- applysvmnode 특성 291
- applytcmnode 특성 292
- applytimeseriesnode 특성 293
- applytreeas 특성 293
- applyts 특성 292
- applytwostepAS 특성 294
- applytwostepnode 특성 293
- applyxgboostlinearnode 특성 294
- applyxgboosttreenode 특성 294
- Apriori 모델
 - 노드 스크립팅 특성 193, 277
- apriorinode 특성 193

AS 시간 간격 노드
특성 143
ASCII가 아닌 문자 25
asexport 특성 333
asimport 특성 87
associationrulesnode 특성 194
astimeintervalsnode 특성 143
autoclassifiernode 특성 196
autoclusternode 특성 199
autodataprepnnode 특성 140
autonumericnode 특성 200

B

balancenode 특성 116
bayesnet 특성 201
binningnode 특성 143
buildr 특성 203

C

c50node 특성 203
C5.0 모델
노드 스크립팅 특성 203, 280
CARMA 모델
노드 스크립팅 특성 205, 280
carmanode 특성 205
cartnode 특성 206
CHAID 모델
노드 스크립팅 특성 208, 281
chaidnode 특성 208
CLEM
스크립팅 1
cognosimport 노드 특성 88
collectionnode 특성 172
Cox 회귀 모형
노드 스크립팅 특성 210, 281
coxregnode 특성 210
CPLEX 최적화 노드
특성 137
cplexnode 특성 137
C&R 트리 모델
노드 스크립팅 특성 206, 280

D

Data Collection 내보내기 노드
특성 340

Data Collection 소스 노드
특성 92
dataauditnode 특성 318
databaseexportnode 특성 336
databasenode 특성 90
datacollectionexportnode 특성 340
datacollectionimportnode 특성 92
dataviewimport 특성 112
decisionlist 특성 212
derivernode 특성 146
derive_stbnode
특성 117
directedwebnode 특성 188
discriminantnode 특성 213
distinctnode 특성 119
distributionnode 특성 173

E

ensemblenode 특성 148
evaluationnode 특성 174
Excel 내보내기 노드
특성 340
Excel 소스 노드
특성 94
excelexportnode 특성 340
excelimportnode 특성 94
exportModelToFile 45
extensionexportnode 특성 341
extensionimportnode 특성 95
extensionmodelnode 특성 215
extensionoutputnode 특성 320
extensionprocessnode 특성 121

F

factornode 특성 217
featureselectionnode 특성 5, 219
fillernode 특성 149
filternode 특성 150
fixedfilenode 특성 98
flatfilenode 특성 342
for 명령 53

G

genlinnode 특성 221

GLE 모델
노드 스크립팅 특성 228, 286
gle 특성 228
GLMM 모델
노드 스크립팅 특성 224, 285
glmnode 특성 224
graphboardnode 특성 176
gsdata_import 노드 특성 101

H

histogramnode 특성 178
historynode 특성 151

I

IBM Cognos TM1 소스 노드
특성 105
IBM Cognos 소스 노드
특성 88
IBM SPSS Analytic Server Repository
명령행 인수 72
IBM SPSS Collaboration and
Deployment Services Repository
명령행 인수 71
스크립팅 54
IBM SPSS Modeler
명령행에서 실행 67
IBM SPSS Statistics 내보내기 노드
특성 351
IBM SPSS Statistics 모델
노드 스크립팅 특성 350
IBM SPSS Statistics 변환 노드
특성 349
IBM SPSS Statistics 소스 노드
특성 349
IBM SPSS Statistics 출력 노드
특성 351

J

JSON 콘텐츠 모델 61
Jython 17

K

kmeansnode 특성 232

KNN 모델
 노드 스크리핑 특성 286
 knnnode 특성 233
 kohonennode 특성 235
 K-평균 모델
 노드 스크리핑 특성 232, 286

L

Linear-AS 모델
 노드 스크리핑 특성 237, 287
 linear-AS 특성 237
 logregnode 특성 238
 lowertoupper 함수 53
 LSVM 모델
 노드 스크리핑 특성 243
 lsvmnode 특성 243

M

mapvisualization 특성 179
 matrixnode 특성 321
 meansnode 특성 323
 mergenode 특성 122
 Microsoft 모델
 노드 스크리핑 특성 295, 298
 MS 로지스틱 회귀분석
 노드 스크리핑 특성 295, 298
 MS 선형 회귀
 노드 스크리핑 특성 295, 298
 MS 시계열
 노드 스크리핑 특성 298
 MS 시퀀스 군집
 노드 스크리핑 특성 298
 MS 신경망
 노드 스크리핑 특성 295, 298
 MS 의사결정 트리
 노드 스크리핑 특성 295, 298
 msassocnode 특성 295
 msbayesnode 특성 295
 msclusternode 특성 295
 mslogisticnode 특성 295
 msneuralnetworknode 특성 295
 msregressionnode 특성 295
 mssequenceclusternode 특성 295
 mstimeseriesnode 특성 295
 mstreenode 특성 295
 multiplotnode 특성 183

N

Netezza Bayes Net 모델
 노드 스크리핑 특성 306, 316
 Netezza KNN 모델
 노드 스크리핑 특성 306, 316
 Netezza K-평균 모델
 노드 스크리핑 특성 306, 316
 Netezza Naive Bayes 모델
 노드 스크리핑 특성 306, 316
 Netezza PCA 모델
 노드 스크리핑 특성 306, 316
 Netezza 모델
 노드 스크리핑 특성 306
 Netezza 분열 군집 모델
 노드 스크리핑 특성 306, 316
 Netezza 선형 회귀 모형
 노드 스크리핑 특성 306, 316
 Netezza 시계열 모델
 노드 스크리핑 특성 306
 Netezza 의사결정 트리 모형
 노드 스크리핑 특성 306, 316
 Netezza 일반화 선형 모델
 노드 스크리핑 특성 306
 Netezza 회귀분석 트리 모델
 노드 스크리핑 특성 306, 316
 netezabayesnode 특성 306
 netezadectreenode 특성 306
 netezadivclusternode 특성 306
 netezzaglmnode 특성 306
 netezzakmeansnode 특성 306
 netezzaknnode 특성 306
 netezzalineressionnode 특성 306
 netezzanaivebayesnode 특성 306
 netezzapcanode 특성 306
 netezzaregtreenode 특성 306
 netezzatimeseriesnode 특성 306
 neuralnetnode 특성 244
 neuralnetworknode 특성 246
 numericpredictornode 특성 200

O

ocsvmnode 특성 355
 One-Class SVM 노드
 특성 355
 oraabnode 특성 299
 oraainode 특성 299

oraapriorinode 특성 299
 Oracle AI 모델
 노드 스크리핑 특성 299
 Oracle Apriori 모델
 노드 스크리핑 특성 299, 305
 Oracle K-평균 모델
 노드 스크리핑 특성 299, 305
 Oracle MDL 모델
 노드 스크리핑 특성 299, 305
 Oracle Naive Bayes 모델
 노드 스크리핑 특성 299, 305
 Oracle NMF 모델
 노드 스크리핑 특성 299, 305
 Oracle O-Cluster
 노드 스크리핑 특성 299, 305
 Oracle 모델
 노드 스크리핑 특성 299
 Oracle 의사결정 트리 모형
 노드 스크리핑 특성 299, 305
 Oracle 일반화 선형 모델
 노드 스크리핑 특성 299
 Oracle 적응형 베이스 모델
 노드 스크리핑 특성 299, 305
 Oracle 지원 벡터 머신 모델
 노드 스크리핑 특성 299, 305
 oradecisiontreenode 특성 299
 oraglmnode 특성 299
 orakmeansnode 특성 299
 oramdlnode 특성 299
 oranbnode 특성 299
 oranmfnode 특성 299
 oraoclusternode 특성 299
 orasvmnode 특성 299
 outputfilenode 특성 342

P

partitionnode 특성 152
 PCA 모델
 노드 스크리핑 특성 217, 284
 PCA/요인 모델
 노드 스크리핑 특성 217, 284
 plotnode 특성 184
 Python 17
 스크리핑 18
 Python 모델
 노드 스크리핑 특성 289, 294

Q

QUEST 모델

- 노드 스크립팅 특성 248, 289
- questnode 특성 248

R

R 변환 노드

- 특성 125

R 작성 노드

- 노드 스크립팅 특성 203

R 출력 노드

- 특성 325

randomtrees 특성 250

reclassifynode 특성 153

regressionnode 특성 251

reordernode 특성 154

reportnode 특성 324

reprojectnode 특성 154

restructurenode 특성 155

retrieve 명령 54

RFM 분석 노드

- 특성 155

RFM 통합 노드

- 특성 124

rfmaggregatenode 특성 124

rfmanalysisnode 특성 155

rouputnode 특성 325

Rprocessnode 특성 125

S

samplenode 특성 126

SAS 내보내기 노드

- 특성 343

SAS 소스 노드

- 특성 102

sasexportnode 특성 343

sasimportnode 특성 102

selectnode 특성 128

sequencenode 특성 253

setglobalsnode 특성 326

settoflagnode 특성 157

Sim Eval 노드

- 특성 326

Sim Fit 노드

- 특성 327

Sim Gen 노드

- 특성 102

simevalnode 특성 326

simfitnode 특성 327

simgennode 특성 102

SLRM 모델

- 노드 스크립팅 특성 255, 291

slrmnode 특성 255

SMOTE 노드

- 특성 353

smotenode 특성 353

sortnode 특성 128

spacetimeboxes 특성 129

Space-Time-Boxes 노드

- 특성 117, 129

Space-Time-Boxes 노드 특성 117

statisticsexportnode 특성 351

statisticsimportnode 특성 5, 349

statisticsmodelnode 특성 350

statisticsnode 특성 328

statisticsoutputnode 특성 351

statistictransformnode 특성 349

store 명령 54

STP 노드

- 특성 256

STP 노드 너깃

- 특성 292

stpnode 특성 256

STP(Spatio-Temporal Prediction) 노드

- 특성 256

streamingtimeseries 특성 130

streamingts 특성 134

stream.nodes 특성 53

string 함수 53

SVM 모델

- 노드 스크립팅 특성 260

svmnode 특성 260

T

tablenode 특성 329

tcm 모델

- 노드 스크립팅 특성 292

tcmnode 특성 261

timeintervalsnode 특성 158

timeplotnode 특성 186

timeseriesnode 특성 269

tmlimport 노드 특성 105

tmlodataimport 노드 특성 105

transformnode 특성 332

transposenode 특성 162

treeas 특성 271

Tree-AS 모델

- 노드 스크립팅 특성 271, 293

ts 특성 265

TWC 가져오기 소스 노드

- 특성 106

twcimport 노드 특성 106

twostepAS 특성 274

twostepnode 특성 273

typenode 특성 5, 164

U

userinputnode 특성 107

V

variablefilenode 특성 108

W

webnode 특성 188

X

XGBoost Linear 노드

- 특성 355

XGBoost Tree 노드

- 특성 353

xgboostlinearnode 특성 355

xgboosttreenode 특성 353

XML 내보내기 노드

- 특성 347

XML 소스 노드

- 특성 111

XML 컨텐츠 모델 60

xmlexportnode 특성 347

xmlimportnode 특성 111

