

*IBM SPSS Modeler 18.1 —
podręcznik tworzenia skryptów w
języku Python i automatyzacji*

IBM

Uwaga

Przed skorzystaniem z niniejszych informacji oraz produktu, którego one dotyczą, należy zapoznać się z informacjami zamieszczonymi w sekcji “Uwagi” na stronie 355.

Informacje o produkcie

Niniejsze wydanie publikacji dotyczy wersji 18, wydania 0, modyfikacji 0 produktu IBM SPSS Modeler oraz wszystkich następnych wersji i modyfikacji do czasu, aż w kolejnym wydaniu publikacji zostanie zawarta informacja o stosownej zmianie.

Spis treści

Rozdział 1. Skrypty i język skryptowy . . . 1

Ogólne informacje o skryptach	1
Typy skryptów	1
Skrypty strumieni	1
Przykład skryptu strumienia: uczenie sieci neuronowej	3
Limity wielkości kodu Jython	4
Skrypty samodzielne	4
Przykład skryptu samodzielnego: zapisywanie i ładowanie modelu	4
Przykład skryptu samodzielnego: generowanie modelu wyboru predyktora	5
Skrypty superwęzłów	5
Przykład skryptu superwęzła	6
Pętle i wykonanie warunkowe w strumieniach	6
Pętle w strumieniach	7
Wykonanie warunkowe w strumieniach	10
Wykonywanie i przerywanie wykonania skryptów	11
Znajdowanie i zastępowanie	12

Rozdział 2. Język skryptowy 15

Przegląd języka skryptowego	15
Python i Jython	15
Język skryptowy Python	16
Operacje	16
Listy	16
Łańcuchy	17
Uwagi	18
Składnia instrukcji	19
Identyfikatory	19
Blok kodu	19
Przekazywanie argumentów do skryptu	20
Przykłady	20
Metody matematyczne	21
Korzystanie ze znaków spoza zestawu ASCII	22
Programowanie zorientowane obiektowo	23
Definiowanie klasy	23
Tworzenie instancji klasy	24
Dodawanie atrybutów do instancji klasy	24
Definiowanie atrybutów i metod klas	24
Zmienne ukryte	25
Dziedziczenie	25

Rozdział 3. Skrypty w programie IBM SPSS Modeler. 27

Typy skryptów	27
Strumienie, strumienie superwęzłów i diagramy	27
Strumienie	27
Strumienie superwęzłów	27
Diagramy	27
Wykonywanie strumienia	27
Kontekst skryptu	28
Odwołania do istniejących węzłów	29
Znajdowanie węzłów	29
Ustawianie właściwości	30
Tworzenie węzłów i modyfikowanie strumieni	30

Tworzenie węzłów	31
Tworzenie i usuwanie połączeń między węzłami	31
Importowanie, zastępowanie i usuwanie węzłów	32
Przechodzenie przez węzły w strumieniu	33
Kasowanie (lub usuwanie) elementów	34
Uzyskiwanie informacji o węzłach	34

Rozdział 4. Skryptowy interfejs API. 37

Wprowadzenie do skryptowego interfejsu API	37
Przykład 1: wyszukiwanie węzłów za pomocą filtra niestandardowego	37
Przykład 2: umożliwianie użytkownikom uzyskanie informacji o katalogu lub pliku na podstawie swoich uprawnień	37
Metadane: informacje o danych	38
Dostęp do wygenerowanych obiektów	40
Obsługa błędów	42
Parametry strumienia, sesji i superwęzła	42
Wartości globalne	46
Praca z wieloma strumieniami: skrypty samodzielne.	47

Rozdział 5. Wskazówki dotyczące skryptów 49

Wpływanie na wykonywanie strumienia	49
Przechodzenie przez węzły w pętli	49
Uzyskiwanie dostępu do obiektów w repozytorium IBM SPSS Collaboration and Deployment Services Repository	49
Generowanie hasła kodowanego	52
Sprawdzanie skryptów	52
Wywoływanie skryptów z wiersza komend	52
Zgodność z wcześniejszymi wersjami	52
Uzyskiwanie dostępu do wyników wykonania strumienia	53
Model zawartości tabeli	53
Model zawartości XML	55
Model zawartości JSON	56
Model zawartości statystyki kolumn i model zawartości statystyki kolumn wyznaczonych parami	57

Rozdział 6. Argumenty w wierszu komend 61

Wywoływanie oprogramowania	61
Używanie argumentów wiersza komend	61
Argumenty systemowe	62
Argumenty określające parametry	63
Argumenty połączenia z serwerem	64
Argumenty połączenia z repozytorium IBM SPSS Collaboration and Deployment Services Repository	65
Argumenty połączenia z repozytorium IBM SPSS Analytic Server	65
Łączne stosowanie wielu argumentów	66

Rozdział 7. Skorowidz właściwości. 67

Przegląd skorowidza właściwości	67
Składnia właściwości	67

Przykłady właściwości węzłów i strumieni	68
Przegląd właściwości węzłów	69
Wspólne właściwości węzłów	69

Rozdział 8. Właściwości strumienia . . . 71

Rozdział 9. Właściwości węzłów źródłowych 75

Właściwości wspólne węzłów źródłowych	75
Właściwości węzła asimport	79
Właściwości węzła cognosimport	80
Właściwości węzła databasenode	82
Właściwości węzła datacollectionimportnode	84
Właściwości węzła excelimportnode	86
Właściwości węzła extensionimportnode	87
Właściwości węzła fixedfilenode	89
Właściwości węzła gsdata_import	92
Właściwości węzła sasimportnode	92
Właściwości węzła simgenode	93
Właściwości węzła statisticsimportnode	94
Właściwości węzła tm1odataimport	95
Właściwości węzła tm1import (zdeprecjonowanego)	95
Właściwości węzła twcimport	96
Właściwości węzła userinputnode	97
Właściwości węzła variablefilenode	98
Właściwości węzła xmlimportnode	101
Właściwości węzła dataviewimport	101

Rozdział 10. Właściwości węzłów związanych z operacjami na rekordach 103

Właściwości węzła appendnode	103
Właściwości węzła aggregatenode	103
Właściwości węzła balancenode	104
Właściwości węzła derive_stbnode	105
Właściwości węzła distinctnode	107
Właściwości węzła extensionprocessnode	108
Właściwości węzła mergenode	110
Właściwości węzła rfmaggregatenode	111
Właściwości węzła samplenode	113
Właściwości węzła selectnode	115
Właściwości węzła sortnode	115
Właściwości węzła spacetimeboxes	116
Właściwości węzła streamingtimeseries	118
Właściwości cplexnode	124

Rozdział 11. Właściwości węzłów związanych z operacjami na zmiennych 125

Właściwości węzła anonymizenode	125
Właściwości węzła autodataprepnode	126
Właściwości węzła astimeintervalsnode	129
Właściwości węzła binningnode	129
Właściwości węzła derivenode	132
Właściwości węzła ensemblenode	134
Właściwości węzła fillernode	135
Właściwości węzła filternode	136
Właściwości węzła historynode	137
Właściwości węzła partitionnode	137
Właściwości węzła reclassifynode	138

Właściwości węzła reordernode	139
Właściwości węzła reprojectnode	140
Właściwości węzła restructurenode	140
Właściwości węzła rfmanalysisnode	141
Właściwości węzła settoflagnode	142
Właściwości węzła statisticstransformnode	143
Właściwości węzła timeintervalsnode (zdeprecjonowanego)	143
Właściwości węzła transposenode	147
Właściwości węzła typenode	149

Rozdział 12. Właściwości węzłów wykresów 155

Właściwości wspólne węzła wykresu	155
Właściwości węzła collectionnode	156
Właściwości węzła distributionnode	157
Właściwości węzła evaluationnode	158
Właściwości węzła graphboardnode	160
Właściwości węzła histogramnode	162
Właściwości węzła mapvisualization	163
Właściwości węzła multiplotnode	167
Właściwości węzła plotnode	168
Właściwości węzła timeplotnode	170
Właściwości węzła webnode	171

Rozdział 13. Właściwości węzłów modelowania 173

Wspólne właściwości węzłów modelowania	173
właściwości anomalydetectionnode węzła	173
Właściwości węzła apriorinode	175
Właściwości węzła associationrulesnode	176
Właściwości węzła autotoclassifiernode	178
Ustawianie właściwości algorytmu	180
Właściwości węzła autoclusternode	180
Właściwości węzła autonumericnode	182
Właściwości węzła bayesnetnode	183
Właściwości węzła c50node	185
Właściwości węzła carmanode	186
Właściwości węzła cartnode	187
Właściwości węzła chaidnode	190
Właściwości węzła coxregnode	192
Właściwości węzła decisionlistnode	193
Właściwości węzła discriminantnode	195
Właściwości węzła extensionmodelnode	196
Właściwości węzła factornode	199
Właściwości węzła featureselectionnode	200
Właściwości węzła genlinnode	202
Właściwości węzła glmnode	205
Właściwości węzła gle	209
Właściwości węzła kmeansnode	213
Właściwości węzła knnnode	214
Właściwości węzła kohonenode	216
Właściwości węzła linearnode	217
Właściwości węzła linearasnode	218
Właściwości węzła logregnode	219
Właściwości węzła lsvmnode	223
Właściwości węzła neuralnetnode	224
Właściwości węzła neuralnetworknode	226
Właściwości węzła questnode	228
Właściwości węzła randomtrees	230

Właściwości węzła regressionnode	231
Właściwości węzła sequencenode	233
Właściwości węzła slrmnode	235
właściwości węzła statisticsmodelnode	235
Właściwości węzła stpnode	236
Właściwości węzła svmnode	239
Właściwości węzła tcmtree	240
Właściwości węzła ts	244
właściwości węzła treeas	250
Właściwości węzła twostepnode	252
Właściwości węzła twostepAS	253

Rozdział 14. Właściwości węzła modelu użytkowego 255

Właściwości węzła applyanomalydetectionnode	255
Właściwości węzła applyapriorinode	255
Właściwości węzła applyassociationrulesnode	256
Właściwości węzła applyautoclassifiernode	256
Właściwości węzła applyautoclusternode	257
Właściwości węzła applyautonumericnode	257
Właściwości węzła applybayesnetnode	257
Właściwości węzła applyc50node	257
Właściwości węzła applycarmanode	258
Właściwości węzła applycartnode	258
Właściwości węzła applychaidnode	258
Właściwości węzła applycoxregnode	259
Właściwości węzła applydecisionlistnode	259
Właściwości węzła applydiscriminantnode	259
Właściwości węzła applyextension	260
Właściwości węzła applyfactornode	261
Właściwości węzła applyfeatureselectionnode	262
Właściwości węzła applygeneralizedlinearnode	262
Właściwości węzła applyglmnode	262
Właściwości węzła applyggle	263
Właściwości węzła applykmeansnode	263
Właściwości węzła applyknnnode	263
Właściwości węzła applykohonenode	264
Właściwości węzła applylinearode	264
Właściwości węzła applylinearasnode	264
Właściwości węzła applylogregnode	264
Właściwości węzła applysvmnode	265
Właściwości węzła applyneuralnetnode	265
Właściwości węzła applyneuralnetworknode	265
Właściwości applyocsvmnode	266
Właściwości węzła applyquestnode	266
Właściwości węzła applyrandomtrees	267
Właściwości węzła applyregressionnode	267
Właściwości węzła applyselflearningnode	268
Właściwości węzła applysequencenode	268
Właściwości węzła applysvmnode	268
Właściwości węzła applystpnode	268
Właściwości węzła applytcmtree	269
Właściwości węzła applytsts	269
Właściwości węzła applytimeseriesnode (zdeprecjonowanego)	269
Właściwości węzła applytreeas	270
Właściwości węzła applytwostepnode	270
Właściwości węzła applytwostepAS	270
Właściwości applyxgboostretnode	270
Właściwości applyxgboostlinearnode	271

Rozdział 15. Właściwości węzłów związanych z modelowaniem w bazie danych 273

Właściwości węzłów związanych z modelowaniem w bazie danych Microsoft	273
Właściwości węzłów związanych z modelowaniem w bazie danych Microsoft	273
Właściwości modelu użytkowego Microsoft	275
Właściwości węzłów związanych z modelowaniem w bazie danych Oracle	277
Właściwości węzłów związanych z modelowaniem w bazie danych Oracle	277
Właściwości modeli użytkowych Oracle	283
Właściwości węzłów związanych z modelowaniem w bazie danych IBM Netezza Analytics	284
Właściwości węzłów związanych z modelowaniem w bazie danych Netezza	284
Właściwości modelu użytkowego Netezza	294

Rozdział 16. Właściwości węzła wyjściowego. 297

Właściwości węzła analysisnode	297
Właściwości węzła dataauditnode	298
Właściwości węzła extensionoutputnode	300
Właściwości węzła matrixnode	301
właściwości węzła meansnode	302
Właściwości węzła reportnode	304
Właściwości węzła setglobalsnode	305
Właściwości węzła simevalnode	306
Właściwości węzła simfitnode	307
Właściwości węzła statisticsnode	307
Właściwości węzła statisticsoutputnode	309
Właściwości węzła tablenode	309
Właściwości węzła transformnode	311

Rozdział 17. Właściwości węzłów eksportu 313

Wspólne właściwości węzłów eksportu	313
Właściwości węzła asexport	313
Właściwości węzła cognosexportnode	314
Właściwości węzła databaseexportnode	316
Właściwości węzła datacollectionexportnode	320
Właściwości węzła excelexportnode	320
Właściwości węzła extensionexportnode	321
Właściwości węzła outputfilenode	322
Właściwości węzła sasexportnode	323
Właściwości węzła statisticsexportnode	324
Właściwości węzła tm1odataexport	324
Właściwości węzła tm1export (zdeprecjonowanego)	325
Właściwości węzła xmlexportnode	327

Rozdział 18. Właściwości węzła IBM SPSS Statistics 329

Właściwości węzła statisticsimportnode	329
Właściwości węzła statisticstransformnode	329
właściwości węzła statisticsmodelnode	330
Właściwości węzła statisticsoutputnode	331
Właściwości węzła statisticsexportnode	331

Rozdział 19. Właściwości węzła Python	333	Zmienne	348
Właściwości smotemode	333	Typy węzłów, wyników i modeli	348
Właściwość xgboosttreemode	333	Nazwy właściwości	348
Właściwości xboostlinearmode	335	Odwołania do węzłów	348
Właściwości ocsvmnode	335	Odczytywanie i ustawianie właściwości	349
Rozdział 20. Właściwości superwęzłów	339	Edytowanie strumieni	349
Dodatek A. Skorowidz nazw węzłów	341	Operacje na węzłach	350
Nazwy modeli użytkowych	341	W pętli	351
Unikanie duplikowania nazw modeli	343	Wykonywanie strumieni	351
Nazwy typów wynikowych	343	Dostęp do obiektów za pośrednictwem systemu plików i repozytorium	352
Dodatek B. Migracja z wcześniejszego języka skryptowego na język Python	345	Operacje na strumieniach	352
Przegląd migracji z wcześniejszego języka skryptowego	345	Operacje na modelach	353
Różnice ogólne	345	Operacje zapisywania dokumentów	353
Kontekst skryptu	345	Pozostałe różnice między wcześniejszym językiem skryptowym a językiem Python	353
Komendy a funkcje	345	Uwagi	355
Literały i komentarze	346	Znaki towarowe	356
Operatory	346	Warunki dotyczące dokumentacji produktu	357
Komendy warunkowe i pętle	347	Indeks	359

Rozdział 1. Skrypty i język skryptowy

Ogólne informacje o skryptach

Skrypty w IBM® SPSS Modeler są potężnym narzędziem automatyzacji procesów interfejsu użytkownika. Skrypty mogą wykonywać takie same działania, które wykonujesz ty, używając myszy lub klawiatury; możesz je wykorzystać do zautomatyzowania wysoce powtarzalnych zadań lub takich zadań, których wykonanie ręczne wymaga dużo czasu.

Skryptów można używać do:

- Wymuszenia konkretnej kolejności wykonania węzłów w strumieniu.
- Nadawania wartości właściwościom węzła oraz przeprowadzania wyliczeń przy użyciu podzbioru języka CLEM (Control Language for Expression Manipulation).
- Zdefiniowania sekwencji czynności wykonywanej automatycznie, bez udziału użytkownika — można na przykład zbudować model, a potem go przetestować.
- Budowania skomplikowanych procesów, które wymagałyby intensywnych interakcji z użytkownikiem, np. procedur walidacji krzyżowej, które wymagają wielokrotnego generowania i testowania modeli.
- Budowania procesów, które manipulują strumieniami — można na przykład uruchomić strumień uczący model i wygenerować automatycznie odpowiedni strumień testujący model.

Ten rozdział zawiera ogólne informacje i przykłady dotyczące skryptów wykonywanych na poziomie strumienia, skryptów samodzielnych i skryptów w superwęzłach w ramach interfejsu programu IBM SPSS Modeler. Więcej informacji o języku skryptowym, jego składni i komendach zawierają kolejne rozdziały.

Uwaga:

W programie IBM SPSS Modeler nie można importować i uruchamiać skryptów utworzonych w programie IBM SPSS Statistics.

Typy skryptów

W programie IBM SPSS Modeler stosowane są trzy typy skryptów:

- **Skrypty strumieni** są przechowywane jako właściwości strumieni, a zatem są zapisywane i ładowane razem z konkretnymi strumieniami. Można na przykład napisać skrypt strumienia, który będzie automatyzował proces uczenia i stosowania modelu użytkowego. Można także nakazać, by przy każdym uruchomieniu konkretnego strumienia uruchamiany był skrypt, a nie zawartość obszaru roboczego.
- **Skrypty samodzielne** nie są powiązane z żadnymi konkretnymi strumieniami i są zapisywane w zewnętrznych plikach tekstowych. Skryptów samodzielnych można używać na przykład do manipulowania kilkoma skryptami jednocześnie.
- **Skrypty superwęzłów** są przechowywane jako właściwości strumieni superwęzłów. Skrypty superwęzłów są dostępne tylko w końcowych superwęzłach. Skryptu superwęzła można użyć do sterowania kolejnością wykonania zawartości superwęzła. W przypadku superwęzłów innych niż końcowe (tj. źródłowych lub procesowych) właściwości superwęzła i zawartych w nim węzłów można definiować bezpośrednio w skrypcie strumienia.

Skrypty strumieni

Skrypty można stosować do realizowania niestandardowych operacji w konkretnym strumieniu. Skrypty takie są zapisywane razem z tym strumieniem. Skrypt strumienia umożliwia określenie kolejności wykonywania węzłów końcowych w strumieniu. Skrypt zapisywany z bieżącym strumieniem edytuje się w oknie dialogowym skryptu strumienia.

Aby uzyskać dostęp do karty skryptu strumienia w oknie dialogowym Właściwości strumienia:

1. Z menu **Narzędzia** wybierz:
Właściwości strumienia > Wykonywanie
2. Kliknij kartę **Wykonywanie**, aby pracować ze skryptami bieżącego strumienia.

Ikony na pasku narzędzi u góry okna dialogowego strumienia umożliwiają następujące operacje:

- Zaimportowanie zawartości istniejącego skryptu samodzielnie do okna.
- Zapisanie skryptu jako pliku tekstowego.
- Wydrukowanie skryptu.
- Dołączenie skryptu domyślnego.
- Edytowanie skryptu (dostępne są funkcje cofania, wycinania, kopiowania, wklejania i inne typowe funkcje edycyjne).
- Wykonanie całego bieżącego skryptu.
- Wykonanie wybranych wierszy skryptu.
- Zatrzymanie wykonywania skryptu. (Ta ikona jest aktywna tylko w trakcie wykonywania skryptu).
- Sprawdzenie składni skryptu i wyświetlenie informacji o ewentualnych błędach w dolnym panelu okna dialogowego.

Uwaga: Począwszy od wersji 16.0 w programie SPSS Modeler używany jest język skryptowy Python. We wszystkich wersjach niższych niż 16.0 używany był własny język skryptowy produktu SPSS Modeler, obecnie nazywany „wcześniejszym językiem skryptowym”. W zależności od rodzaju skryptu, z jakim pracujesz, na karcie **Wykonywanie** wybierz tryb wykonania **Domyślny (skrypt opcjonalny)**, a następnie wybierz **Python** albo **Wcześniejszy**.

Można określić, czy podczas wykonywania strumienia skrypt ma być uruchamiany, czy nie. Wybranie opcji **Wykonaj ten skrypt** spowoduje, że skrypt będzie uruchamiany przy każdym wykonaniu strumienia, z zachowaniem kolejności wykonania określonej w skrypcie. To ustawienie umożliwi automatyzację na poziomie strumienia, a w efekcie szybsze budowanie modeli. Jednak zgodnie z ustawieniem domyślnym skrypt jest ignorowany w trakcie wykonywania strumienia. Nawet jeśli wybierzesz opcję **Ignoruj ten skrypt**, możesz zawsze uruchomić skrypt bezpośrednio z okna dialogowego.

Edytor skryptów oferuje następujące funkcje ułatwiające pisanie skryptów:

- Wyróżnianie składni: słowa kluczowe, literały (takie jak łańcuchy i liczby) oraz komentarze są wyróżniane.
- Numerowanie wierszy.
- Wyszukiwanie bloków: umieszczenie kursora na początku bloku programu powoduje wyróżnienie także końca tego bloku.
- Automatyczne uzupełnianie tekstu.

Kolory i style używane do wyróżniania składni można zmodyfikować za pośrednictwem preferencji wyświetlania w programie IBM SPSS Modeler. Aby uzyskać dostęp do preferencji wyświetlania, należy wybrać kolejno **Narzędzia > Opcje > Opcje użytkownika** i kliknąć kartę **Komendy**.

Listę podpowiedzi sugerowanych przez funkcję automatycznego uzupełniania można wyświetlić, wybierając opcję **Automatyczne sugerowanie** z menu kontekstowego lub naciskając kombinację klawiszy Ctrl + Spacja. Za pomocą klawiszy kursora wybierz pozycję na liście i naciśnij klawisz Enter, aby wstawić wybrany tekst. Naciśnięcie klawisza Esc powoduje wyjście z trybu automatycznego sugerowania bez zmiany tekstu w edytorze.

Na karcie **Debugowanie** wyświetlane są komunikaty debugowania, które umożliwiają ocenę stanu skryptu po jego wykonaniu. karta **Debugowanie** zawiera obszar tekstowy przeznaczony tylko do odczytu oraz jednowierszowe pole wprowadzania tekstu. W obszarze tekstowym wyświetlany jest tekst wysyłany przez skrypty do standardowego wyjścia lub standardowego błędu — na przykład komunikaty o błędach. W polu tekstowym użytkownik może wprowadzać dane. Te dane są analizowane w kontekście skryptu ostatnio wykonywanego w oknie dialogowym (jest to tzw. *kontekst*

skryptu). Obszar tekstowy zawiera komendę i uzyskane wyniki, zatem użytkownik może prześledzić przebieg wykonania komend. Pole wprowadzania tekstu zawsze zawiera zachętę wiersza komend (--> w przypadku wcześniejszego języka skryptowego).

Nowy kontekst skryptu tworzony jest w następujących okolicznościach:

- Skrypt jest wykonywany poprzez użycie opcji **Wykonaj ten skrypt** lub **Wykonaj zaznaczone wiersze**.
- Gdy wybrany zostanie inny niż dotąd język skryptowy.

Utworzenie nowego kontekst skryptu powoduje wyczyszczenie obszaru tekstowego.

Uwaga: Wykonanie strumienia poza panelem skryptu nie powoduje zmodyfikowania kontekstu skryptu w tym panelu. Wartości zmiennych wygenerowane w wyniku wykonania nie są widoczne w oknie dialogowym skryptu.

Przykład skryptu strumienia: uczenie sieci neuronowej

Strumienia można użyć do uczenia modelu sieci neuronowej. Zwykle aby przetestować model, należałoby uruchomić węzeł modelowania w celu dodania modelu do strumienia, utworzyć odpowiednie połączenia, a następnie uruchomić węzeł Analiza.

Korzystając ze skryptu programu IBM SPSS Modeler, można zautomatyzować proces testowania modelu użytkowego po jego utworzeniu. Poniższy skrypt strumienia testuje strumień demonstracyjny `druglearn.str` (dostępny w folderze `/Demos/streams/` wewnątrz folderu instalacyjnego programu IBM SPSS Modeler). Skrypt można uruchomić z okna dialogowego Właściwości strumienia (**Narzędzia > Właściwości strumienia > Skrypt**):

```
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([])
```

W następujących punktach opisano poszczególne wiersze przykładowego skryptu.

- Pierwszy wiersz definiuje zmienną wskazującą na bieżący strumień.
- W 2. wierszu skrypt znajduje węzeł tworzący sieć neuronową.
- W 3. wierszu skrypt tworzy listę, w której mogą być zapisywane wyniki wykonania.
- W 4. wierszu tworzony jest model użytkowy Sieć neuronowa. Jest on zapisywany w liście zdefiniowanej w wierszu 3.
- W 5. wierszu tworzony jest węzeł modelu do zastosowania dla modelu użytkowego. Węzeł ten jest umieszczany w obszarze roboczym strumienia.
- W 6. wierszu tworzony jest węzeł analizy o nazwie Drug.
- W 7. wierszu skrypt znajduje węzeł Typ.
- W 8. wierszu skrypt łączy węzeł modelu do zastosowania utworzony w 5. wierszu, umieszczając go między węzłem Typ a węzłem Analiza.
- Na koniec wykonywany jest węzeł Analiza w celu wygenerowania raportu analitycznego.

Za pomocą skryptu można utworzyć i uruchomić strumień od podstaw, począwszy od pustego obszaru roboczego. Więcej ogólnych informacji o języku skryptowym zawiera sekcja Przegląd języka skryptowego.

Limity wielkości kodu Jython

Język Jython kompiluje każdy skrypt do kodu bajtowego Java który jest następnie wykonywany wirtualną maszyną języka Java (JVM). Jednak język Java nakłada limity dotyczące wielkości pojedynczego pliku kodu bajtowego. Dlatego próba załadowania kodu bajtowego przez Jython może doprowadzić do awarii JVM. IBM SPSS Modeler nie może temu zapobiec.

Należy upewnić się, że skrypty w języku Python są pisane zgodnie z dobrymi praktykami programowania (polegającymi m.in. na minimalizacji powielającego się kodu poprzez wykorzystanie zmiennych lub funkcji do obliczania wartości wspólnego języka pośredniego). Jeśli to konieczne, należy podzielić kod dla plików źródłowych lub zdefiniować go przy użyciu modułów, ponieważ są one kompilowane do oddzielnych plików kodu bajtowego.

Skrypty samodzielne

Okno dialogowe Skrypt samodzielny służy do tworzenia i edytowania skryptów zapisywanych w postaci plików tekstowych. W oknie wyświetlana jest nazwa pliku i dostępne są funkcje ładowania, importowania, zapisywania i wykonywania skryptów.

Aby uzyskać dostęp do okna dialogowego Skrypt samodzielny:

Z menu głównego wybierz opcje:

Narzędzia > Skrypt samodzielny

W przypadku skryptów samodzielnych dostępny jest ten sam pasek narzędzi i te same opcje sprawdzania składni, co w przypadku skryptów strumieni. Więcej informacji można znaleźć w temacie “Skrypty strumieni” na stronie 1.

Przykład skryptu samodzielnego: zapisywanie i ładowanie modelu

Skrypty samodzielne są przydatne do manipulowania strumieniami. Załóżmy, że mamy dwa strumienie: strumień, który tworzy model, i strumień, który służy do eksploracji zestawu reguł wygenerowanego przez pierwszy strumień po zastosowaniu tego zestawu do istniejących zmiennych z danymi. Skrypt samodzielny realizujący taki scenariusz mógłby wyglądać na przykład tak:

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

# Zmień na właściwą ścieżkę folderu Demos w konkretnej
# instalacji. Uwaga: ukośniki zwykły i ukośnik na końcu.
installation = "C:/Program Files/IBM/SPSS/Modeler/18.1/Demos/"

# Najpierw załaduj z pliku strumień budowy modelu i zbuduj model
druglearn_stream = taskrunner.openStreamFromFile(installation + "streams/druglearn.str", True)
results = []
druglearn_stream.findByType("c50", None).run(results)

# Zapisz model w pliku
taskrunner.saveModelToFile(results[0], "rule.gm")

# Teraz załaduj strumień do pracy z wykresami, wczytaj model z pliku i wstaw do strumienia
drugplot_stream = taskrunner.openStreamFromFile(installation + "streams/drugplot.str", True)
model = taskrunner.openModelFromFile("rule.gm", True)
modelapplier = drugplot_stream.createModelApplier(model, "Drug")

# Teraz znajdź węzeł wykresu, odłącz go i podłącz węzeł
# modelu do zastosowania między węzłem wyliczeń a węzłem wykresu
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([])
```

Uwaga: Więcej ogólnych informacji o języku skryptowym zawiera sekcja Przegląd języka skryptowego.

Przykład skryptu samodzielnego: generowanie modelu wyboru predyktora

Wychodząc od pustego obszaru roboczego, ten przykład buduje strumień, który generuje model wyboru predyktora, stosuje model i tworzy tabelę z 15 zmiennymi najważniejszymi względem określonej zmiennej przewidywanej.

```
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)

# Zakładamy, że strumień automatycznie umieszcza węzły modelu do zastosowania
applynode = stream.findByType("applyfeatureselection", None)
tablenode = stream.createAt("table", "Table", applynode.getXPosition() + 96,
applynode.getYPosition())
stream.link(applynode, tablenode)
tablenode.run([])
```

Skrypt tworzy węzeł źródłowy odczytujący dane, za pomocą węzła Typ określa dla zmiennej `response_01` rolę `Target` (zmiennej docelowej), a następnie tworzy i wykonuje węzeł wyboru predyktora. Ponadto skrypt łączy węzły i umieszcza je w czytelnym układzie w obszarze roboczym. Wynikowy model użytkowy jest następnie łączony z węzłem Tabela, który zawiera listę najważniejszych pól określonych przez właściwości `selection_mode` i `top_n`. Więcej informacji można znaleźć w temacie “Właściwości węzła `featureselectionnode`” na stronie 200.

Skrypty superwęzłów

Korzystając z języka skryptowego programu IBM SPSS Modeler, można tworzyć i zapisywać skrypty w dowolnych superwęzłach końcowych. Skrypty te są dostępne tylko w końcowych superwęzłach i często używane do tworzenia szablonów strumieni bądź wymuszania określonej kolejności wykonywania zawartości superwęzła. Skrypty superwęzłów umożliwiają również jednoczesne wykonywanie więcej niż jednego skryptu wewnątrz strumienia.

Załóżmy na przykład, że chcemy uzyskać konkretną kolejność wykonywania złożonego strumienia, a superwęzeł zawiera kilka węzłów, w tym węzeł `Globalne`, który musi być wykonany przed wyliczeniem nowej zmiennej używanej w węźle `Wykres`. W takim przypadku możemy utworzyć skrypt superwęzła, który w pierwszej kolejności wykona węzeł `Globalne`. Wartości obliczone przez ten węzeł, takich jak średnia lub odchylenie standardowe, będzie można następnie użyć podczas wykonywania węzła `Wykres`.

W skrypcie superwęzła można nadawać wartości właściwościom węzła, tak samo jak w innych skryptach. Można też zmienić i definiować właściwości dowolnego superwęzła i zawartych w nim węzłów bezpośrednio ze skryptu strumienia. Więcej informacji można znaleźć w rozdziale Rozdział 20, “Właściwości superwęzłów”, na stronie 339. Ta metoda działa zarówno z superwęzłami źródłowymi i procesowymi, jak i z superwęzłami końcowymi.

Uwaga: Ponieważ tylko superwęzły końcowe mogą wykonywać własne skrypty, karta Skrypty okna dialogowego Superwęzeł jest dostępna tylko w końcowych superwęzłach.

Aby otworzyć okno dialogowe skryptu superwęzła z głównego obszaru roboczego:

Zaznacz końcowy superwęzeł w obszarze roboczym strumienia i z menu superwęzła wybierz:

Skrypt Superwęzła...

Aby otworzyć okno dialogowe skryptu superwęzła z obszaru roboczego powiększonego superwęzła:

Prawym przyciskiem myszy kliknij obszar roboczy superwęzła i z menu kontekstowego wybierz:

Skrypt Superwęzła...

Przykład skryptu superwęzła

Następujący skrypt superwęzła deklaruje kolejność wykonywania węzłów końcowych wewnątrz superwęzła. Kolejność ta gwarantuje, że węzeł wartości globalnych zostanie wykonany jako pierwszy, tak aby wartości obliczone przez ten węzeł dało się wykorzystać przy wykonywaniu następnych węzłów.

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

Blokowanie i odblokowywanie superwęzłów

W poniższym przykładzie przedstawiono, w jaki sposób można zablokować i odblokować Superwęzeł:

```
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')
```

Pętle i wykonanie warunkowe w strumieniach

Począwszy od wersji 16.0 SPSS Modeler umożliwia tworzenie prostych skryptów wewnątrz strumienia poprzez wybór wartości w różnych oknach dialogowych — bez konieczności pisania instrukcji w języku skryptowym. Dwa główne typy skryptów, jakie można utworzyć w ten sposób, to pętle i skrypty wykonujące węzły, jeśli spełniony jest określony warunek.

W strumieniu można stosować jednocześnie pętle i reguły wykonywania warunkowego. Załóżmy, że mamy dane opisujące sprzedaż samochodów różnych producentów z całego świata. Możemy skonfigurować pętlę przetwarzającą dane w strumieniu, identyfikującą informacje na podstawie kraju produkcji i generującą różne wykresy przedstawiające np. sprzedaż poszczególnych modeli, poziomy emisji w zależności od producenta i pojemności silnika itd. Gdyby interesowały nas tylko informacje z Europy, możemy dodać do pętli warunki wykluczające tworzenie wykresów dotyczących producentów amerykańskich i azjatyckich.

Uwaga: Ponieważ zarówno pętle, jak i wykonanie warunkowe bazuje na skryptach działających w tle, techniki te są zawsze stosowane do całego strumienia.

- **Pętle** Pętle umożliwiają automatyzację powtarzalnych zadań, na przykład dodanie określonej liczby węzłów do strumienia, każdorazowo ze zmianą jednego parametru węzła. Można też nakazać wykonanie strumienia lub gałęzi określoną liczbę razy, tak jak ilustrują to poniższe przykłady:
 - Uruchomienie strumienia zadaną liczbę razy, za każdym razem z innym źródłem.
 - Uruchomienie strumienia zadaną liczbę razy, za każdym razem z inną wartością jednej ze zmiennych.
 - Uruchomienie strumienia zadaną liczbę razy, za każdym razem z jedną dodatkową zmienną.
 - Zbudowanie modelu zadaną liczbę razy, za każdym razem z innym ustawieniem.
- **Wykonanie warunkowe** Ta technika umożliwia sterowanie wykonaniem węzłów końcowych w zależności od wstępnie zdefiniowanych warunków, na przykład:
 - Decydowanie o uruchomieniu lub nieuruchomieniu strumienia na podstawie tego, czy dana wartość jest równa true, czy false.
 - Definiowanie równoległego albo sekwencyjnego wykonania węzłów pętli.

Zarówno pętle, jak i wykonanie warunkowe można skonfigurować na karcie Wykonywanie w oknie dialogowym Właściwości strumienia. Wszelkie węzły używane w warunkach lub pętli są opatrzone dodatkowym symbolem w obszarze roboczym strumienia. Symbol ten oznacza, że uczestniczą w pętli lub wykonaniu warunkowym.

Dostęp do karty Wykonywanie można uzyskać na 3 różne sposoby:

- Za pomocą menu w górnej części głównego okna dialogowego:
 1. Z menu Narzędzia wybierz:
 - Właściwości strumienia > Wykonywanie**
 2. Kliknij kartę Wykonywanie, aby pracować ze skryptami bieżącego strumienia.
- Z wnętrza strumienia:
 1. Kliknij węzeł prawym przyciskiem myszy i wybierz opcję **Wykonanie w pętli / warunkowe**.
 2. Wybierz odpowiednią opcję z podmenu.
- Na graficznym pasku narzędzi u góry głównego okna dialogowego kliknij ikonę właściwości strumienia.

Jeśli po raz pierwszy konfigurujesz pętlę lub wykonanie warunkowe, na karcie Wykonywanie wybierz tryb wykonania **Wykonanie w pętli / warunkowe**, a następnie wybierz kartę podrzędną **Warunkowe** albo **W pętli**.

Pętle w strumieniach

Pętle umożliwiają automatyzowanie powtarzalnych zadań w strumieniach, na przykład:

- Uruchomienie strumienia zadaną liczbę razy, za każdym razem z innym źródłem.
- Uruchomienie strumienia zadaną liczbę razy, za każdym razem z inną wartością jednej ze zmiennych.
- Uruchomienie strumienia zadaną liczbę razy, za każdym razem z jedną dodatkową zmienną.
- Zbudowanie modelu zadaną liczbę razy, za każdym razem z innym ustawieniem.

Warunki, które muszą być spełnione, można skonfigurować na karcie podrzędnej **W pętli** karty Wykonywanie. Aby wyświetlić tę podkartę, wybierz tryb wykonania **Wykonanie w pętli / warunkowe**.

Wszelkie zdefiniowane ustawienia pętli odniosą skutek po uruchomieniu strumienia, o ile wybrany jest tryb wykonania **Wykonanie w pętli / warunkowe**. Opcjonalnie można wygenerować kod skryptu dla pętli i wkleić go do edytora skryptów, klikając opcję **Wklej...** w prawym dolnym rogu karty podrzędnej **W pętli**; na karcie Wykonywanie będzie teraz wskazywany tryb wykonania **Domyślny (skrypt opcjonalny)**, a w górnej części karty wyświetlany będzie kod skryptu. Oznacza to, że można zdefiniować strukturę pętli za pomocą opcji w oknie dialogowym, a potem wygenerować skrypt i rozbudować go lub zmodyfikować w edytorze. Należy zwrócić uwagę, że kliknięcie opcji **Wklej...** spowoduje także uwzględnienie w wygenerowanym skrypcie wszelkich ustawień wykonania warunkowego.

Ważne: Zmienne pętli określone w strumieniu programu SPSS Modeler mogą zostać przesłonięte, jeśli strumień będzie wykonywany w ramach zadania IBM SPSS Collaboration and Deployment Services. Wynika to z faktu, że wpisy w edytorze zadań IBM SPSS Collaboration and Deployment Services przesłaniają wpisy wprowadzone w

programie SPSS Modeler . Na przykład, jeśli skonfigurujesz w strumieniu zmienną pętli w celu utworzenia w każdym przebiegu pętli pliku wynikowego o innej nazwie, pliki będą prawidłowo nazywane w programie SPSS Modeler, ale zostaną przesłonięte stałą nazwą wprowadzoną na karcie Wynik w programie IBM SPSS Collaboration and Deployment Services Deployment Manager.

Aby skonfigurować pętlę

1. Utwórz klucz iteracji, aby zdefiniować główną strukturę pętli, która ma być wykonywana w strumieniu. Więcej informacji można znaleźć w sekcji Tworzenie klucza iteracji.
2. W razie potrzeby zdefiniuj jedną lub więcej zmiennych iteracji. Więcej informacji można znaleźć w sekcji Tworzenie zmiennej iteracji.
3. Utworzone iteracje i zmienne będą widoczne w głównej części karty podrzędnej. Domyślnie iteracje wykonywane są w kolejności, w jakiej są wymienione; aby przenieść iterację wyżej lub niżej na liście, zaznacz ją kliknięciem, a potem użyj przycisków ze strzałkami w górę i w dół w kolumnie po prawej stronie karty podrzędnej, aby zmienić kolejność.

Tworzenie klucza iteracji pętli w strumieniach

Klucz iteracji służy do definiowania głównej struktury pętli, która ma być wykonywana w strumieniu. Na przykład, jeśli analizujesz sprzedaż samochodów, możesz utworzyć parametr strumienia *Country of manufacture* (Kraj produkcji) i wykorzystać go jako klucz iteracji; podczas wykonywania strumienia klucz przyjmował będzie wartości kolejnych krajów określone w danych. Do konfigurowania klucza służy okno dialogowe Zdefiniuj klucz iteracji.

Aby otworzyć to okno dialogowe, wybierz przycisk **Klucz iteracji...** w lewym dolnym rogu karty podrzędnej W pętli albo prawym przyciskiem myszy kliknij dowolny węzeł w strumieniu i wybierz kolejno opcje **Wykonanie w pętli / warunkowe > Zdefiniuj klucz iteracji (zmienne)** albo **Wykonanie w pętli / warunkowe > Zdefiniuj klucz iteracji (wartości)**. Jeśli otworzysz okno dialogowe ze strumienia, niektóre pola, takie jak nazwa węzła, mogą zostać wypełnione automatycznie.

Aby skonfigurować klucz iteracji, wypełnij następujące pola:

Wykonaj iterację. Można wybrać jedną z następujących opcji:

- **Parametr strumienia - zmienne.** Ta opcja służy do tworzenia pętli, która w każdym przebiegu przypisuje istniejącemu parametrowi strumienia kolejną z określonych zmiennych.
- **Parametr strumienia - wartości.** Ta opcja służy do tworzenia pętli, która w każdym przebiegu przypisuje istniejącemu parametrowi strumienia kolejną z określonych wartości.
- **Właściwość węzła - zmienne.** Ta opcja służy do tworzenia pętli, która w każdym przebiegu przypisuje właściwości węzła kolejną z określonych zmiennych.
- **Właściwość węzła - wartości.** Ta opcja służy do tworzenia pętli, która w każdym przebiegu przypisuje właściwości węzła kolejną z określonych wartości.

Co ustawić. Wybierz element, któremu będzie przypisywana wartość w każdym przebiegu pętli. Można wybrać jedną z następujących opcji:

- **Parametr.** Opcja dostępna tylko wtedy, gdy wybrano **Parametr strumienia - zmienne** albo **Parametr strumienia - wartości**. Wybierz wymagany parametr z listy.
- **Węzeł.** Opcja dostępna tylko wtedy, gdy wybrano **Właściwość węzła - zmienne** albo **Właściwość węzła - wartości**. Wybierz węzeł, dla którego chcesz skonfigurować pętlę. Kliknij przycisk przeglądania, aby otworzyć okno dialogowe wyboru węzła, i wybierz żądany węzeł; jeśli wyświetlonych zostanie zbyt dużo węzłów, można odfiltrować węzły tak, by widoczne były tylko węzły określonego typu: źródłowe, procesowe, wykresów, modelowania, wynikowe, eksportu lub modelu do zastosowania.
- **Właściwość.** Opcja dostępna tylko wtedy, gdy wybrano **Właściwość węzła - zmienne** albo **Właściwość węzła - wartości**. Wybierz właściwość węzła z listy.

Zmienne do użycia. Opcja dostępna tylko wtedy, gdy wybrano **Parametr strumienia - zmienne** albo **Właściwość węzła - zmienne**. Wybierz w ramach węzła zmienną lub zmienną, które będą dostarczały wartości w kolejnej iteracji. Można wybrać jedną z następujących opcji:

- **Węzeł.** Opcja dostępna tylko wtedy, gdy wybrano **Parametr strumienia - zmienne**. Wybierz węzeł zawierający szczegółowe informacje, dla których chcesz skonfigurować pętlę. Kliknij przycisk przeglądania, aby otworzyć okno dialogowe wyboru węzła, i wybierz żądany węzeł; jeśli wyświetlonych zostanie zbyt dużo węzłów, można odfiltrować węzły tak, by widoczne były tylko węzły określonego typu: źródłowe, procesowe, wykresów, modelowania, wynikowe, eksportu lub modelu do zastosowania.
- **Lista zmiennych.** Kliknij przycisk listy w prawej kolumnie, aby wyświetlić okno dialogowe wyboru zmiennych, w którym można będzie wybrać z węzła zmienne dostarczające wartości w kolejnych interakcjach. Więcej informacji można znaleźć w “Wybieranie zmiennych do iteracji” na stronie 10.

Wartości do użycia. Opcja dostępna tylko wtedy, gdy wybrano **Parametr strumienia - wartości** albo **Właściwość węzła - wartości**. Wybierz w ramach wybranej zmiennej wartość lub wartości, które będą używane w kolejnych iteracjach. Można wybrać jedną z następujących opcji:

- **Węzeł.** Opcja dostępna tylko wtedy, gdy wybrano **Parametr strumienia - wartości**. Wybierz węzeł zawierający szczegółowe informacje, dla których chcesz skonfigurować pętlę. Kliknij przycisk przeglądania, aby otworzyć okno dialogowe wyboru węzła, i wybierz żądany węzeł; jeśli wyświetlonych zostanie zbyt dużo węzłów, można odfiltrować węzły tak, by widoczne były tylko węzły określonego typu: źródłowe, procesowe, wykresów, modelowania, wynikowe, eksportu lub modelu do zastosowania.
- **Lista zmiennych.** Wybierz w węźle zmienną, która będzie dostarczać danych w kolejnych iteracjach.
- **Lista wartości.** Kliknij przycisk listy w prawej kolumnie, aby wyświetlić okno dialogowe wyboru wartości, w którym można będzie wybrać z węzła wartości będące danymi w kolejnych interakcjach.

Tworzenie zmiennej iteracji pętli w strumieniach

Zmienne iteracji umożliwiają modyfikowanie wartości parametrów strumienia lub właściwości wybranych węzłów w strumieniu w każdym przebiegu pętli. Na przykład, jeśli pętla w strumieniu analizuje sprzedaż samochodów i kluczem iteracji jest kraj produkcji (*Country of manufacture*), można wygenerować jeden wykres wynikowy przedstawiający sprzedaż poszczególnych modeli oraz drugi wykres wynikowy z informacjami o emisji. W takim przypadku można utworzyć zmienne iteracji w celu utworzenia nowych tytułów dla wynikowych wykresów, na przykład *Swedish vehicle emissions* (Emisja pojazdów szwedzkich) i *Japanese car sales by model* (Sprzedaż samochodów japońskich wg modelu). Do konfigurowania wymaganych zmiennych służy okno dialogowe Zdefiniuj zmienną iteracji.

Aby otworzyć to okno dialogowe, wybierz przycisk **Dodaj zmienną...** w lewym dolnym rogu karty podrzędnej W pętli albo prawym przyciskiem myszy kliknij dowolny węzeł w strumieniu i wybierz kolejno opcje **Wykonanie w pętli / warunkowe > Zdefiniuj zmienną iteracji**.

Aby skonfigurować zmienną iteracji, wypełnij następujące pola:

Zmień. Wybierz typ atrybutu, który chcesz modyfikować. Można wybrać **Parametr strumienia** albo **Właściwość węzła**.

- W przypadku wybrania opcji **Parametr strumienia** wybierz wymagany parametr, a następnie za pośrednictwem jednej z poniższych opcji (o ile są dostępne w danym strumieniu) określ, jaka wartość ma być przypisywana temu parametrowi w każdej iteracji pętli:
 - **Zmienna globalna.** Wybierz zmienną globalną, która ma być przypisywana do parametru strumienia
 - **Komórka wynikowa tabeli.** Aby przypisać parametrowi strumienia wartość komórki tabeli wynikowej, wybierz tabelę z listy i wprowadź **Wiersz i Kolumnę**.
 - **Wprowadź ręcznie.** Wybierz tę opcję, aby ręcznie wprowadzać wartość parametru w każdej iteracji. Po powrocie do karty podrzędnej W pętli zostanie utworzona nowa kolumna, w której można będzie wprowadzić wymagany tekst.
- W przypadku wybrania opcji **Właściwość węzła** wybierz węzeł i jedną z jego właściwości, a następnie określ wartość, którą ta właściwość ma przyjmować. Nadaj właściwości nową wartość, używając jednej z następujących opcji:
 - **Sama wartość.** Wartość właściwości będzie równa wartości klucza iteracji. Więcej informacji można znaleźć w sekcji “Tworzenie klucza iteracji pętli w strumieniach” na stronie 8.

- **Jako przedrostek do rdzenia.** Wartość klucza iteracji będzie przedrostkiem poprzedzającym wartość wpisaną w polu **Rdzeń**.
- **Jako przyrostek do rdzenia.** Wartość klucza iteracji będzie przyrostkiem następującym po wartości wpisanej w polu **Rdzeń**.

W razie wybrania opcji przedrostka lub przyrostka pojawi się monit o wprowadzenie dodatkowego tekstu w polu **Rdzeń**. Na przykład, jeśli klucz iteracji ma wartość *Country of manufacture* i wybierzesz opcję **Jako przyrostek do rdzenia**, możesz wprowadzić w tym polu tekst - *sales by model*.

Wybieranie zmiennych do iteracji

Tworząc iterację, można wybrać jedną lub więcej zmiennych za pomocą okna dialogowego Wybierz zmienne.

Sortuj według Można sortować widoczne zmienne, wybierając jedną z następujących opcji:

- **Naturalnie** Zmienne będą widoczne w kolejności, w jakiej zostały przekazane do bieżącego węzła w strumieniu danych.
- **Nazwa** Zmienne będą posortowane alfabetycznie.
- **Typ** Zmienne będą posortowane według poziomu pomiaru. Ta opcja jest przydatna do wybierania zmiennych o określonym poziomie pomiaru.

Wybieraj zmienne z listy po jednej albo zaznaczaj więcej zmiennych, klikając je z naciśniętym klawiszem Shift lub Ctrl. Można także użyć przycisków pod listą do wybierania grup zmiennych na podstawie ich poziomu pomiaru lub do wybrania lub anulowania wyboru wszystkich zmiennych w tabeli.

Należy zwrócić uwagę, że zmienne dostępne do wyboru są odfiltrowane, tak że lista zawiera tylko zmienne odpowiednie dla wybranego parametru strumienia lub wybranej właściwości węzła. Na przykład, jeśli używany jest parametr strumienia o typie składowania Łańcuch, to wyświetlane są tylko zmienne o tym typie składowania.

Wykonanie warunkowe w strumieniach


Wykonanie warunkowe umożliwia sterowanie wykonaniem węzłów końcowych w zależności od tego, czy zawartość strumienia spełnia warunki określone przez użytkownika, na przykład:

- Decydowanie o uruchomieniu lub nieuruchomieniu strumienia na podstawie tego, czy dana wartość jest równa true, czy false.
- Definiowanie równoległego albo sekwencyjnego wykonania węzłów pętli.

Warunki, które muszą być spełnione, można skonfigurować na karcie podrzędnej **Warunkowe** karty Wykonywanie. Aby wyświetlić tę podkartę, wybierz tryb wykonania **Wykonanie w pętli / warunkowe**.

Wszelkie zdefiniowane ustawienia wykonania warunkowego odniosą skutek po uruchomieniu strumienia, o ile wybrany jest tryb wykonania **Wykonanie warunkowe / warunkowe**. Opcjonalnie można wygenerować kod skryptu dla wykonania warunkowego i wkleić go do edytora skryptów, klikając opcję **Wklej...** w prawym dolnym rogu karty podrzędnej Warunkowe; na karcie Wykonywanie będzie teraz wskazywany tryb wykonania **Domyslny (skrypt opcjonalny)**, a w górnej części karty wyświetlany będzie kod skryptu. Oznacza to, że można zdefiniować warunki za pomocą opcji w oknie dialogowym, a potem wygenerować skrypt i rozbudować warunki w edytorze. Należy zwrócić uwagę, że kliknięcie opcji **Wklej...** spowoduje także uwzględnienie w wygenerowanym skrypcie wszelkich ustawień pętli.

Aby skonfigurować warunek:

1. W prawej kolumnie karty podrzędnej Warunkowe kliknij przycisk Dodaj nowy warunek , aby otworzyć okno dialogowe Dodaj warunkową instrukcję wykonania. To okno dialogowe służy do określania warunku, który musi być spełniony, aby węzeł został wykonany.
2. W oknie dialogowym Dodaj warunkową instrukcję wykonania określ następujące dane:

- a. **Węzeł.** Wybierz węzeł, dla którego chcesz skonfigurować wykonanie warunkowe. Kliknij przycisk przeglądania, aby otworzyć okno dialogowe wyboru węzła, i wybierz żądany węzeł; jeśli wyświetlonych zostanie zbyt dużo węzłów, można odfiltrować węzły tak, by widoczne były tylko węzły określonego typu: eksportu, wykresów, modelowania lub wynikowe.
- b. **Warunek w oparciu o.** Określ warunek, który musi być spełniony, aby węzeł został wykonany. Można wybrać jedną z czterech opcji: **Parametr strumienia**, **Zmienna globalna**, **Komórka wynikowa tabeli** lub **Zawsze prawda**. To, jakie informacje szczegółowe należy wprowadzić w dolnej połowie okna dialogowego, zależy od wybranego warunku.
 - **Parametr strumienia.** Wybierz parametr z listy, a następnie wybierz **Operator** dla tego parametru. Operatorem może być na przykład: Więcej niż, Równa się, Mniej niż, Między itd. Następnie należy wprowadzić **Wartość** albo wartość minimalną i maksymalną, w zależności od operatora.
 - **Zmienna globalna.** Wybierz zmienną z listy; może to być na przykład: Średnia, Suma, Wartość minimalna, Wartość maksymalna lub Odchylenie standardowe. Następnie wybiera się **Operator** i wymagane wartości.
 - **Komórka wynikowa tabeli.** Wybierz węzeł tabeli z listy, a następnie wybierz **Wiersz** i **Kolumnę** w tabeli. Następnie wybiera się **Operator** i wymagane wartości.
 - **Zawsze prawda.** Wybierz tę opcję, jeśli węzeł ma być zawsze wykonywany. Po wybraniu tej opcji nie ma dostępnych do wyboru żadnych innych parametrów.
3. Powtórz kroki 1 i 2 tyle razy, ile trzeba, by zdefiniować wszystkie wymagane warunki. Wybrany węzeł i warunek, który musi być spełniony, aby węzeł został wykonany, są widoczne w głównej części karty podrzędnej, odpowiednio w kolumnach **Węzły wykonywalne** i **Warunek do spełnienia**.
4. Domyślnie węzły i warunki wykonywane są w kolejności, w jakiej są wymienione; aby przenieść węzeł i warunek wyżej lub niżej na liście, zaznacz ją kliknięciem, a potem użyj przycisków ze strzałkami w górę i w dół w kolumnie po prawej stronie karty podrzędnej, aby zmienić kolejność.

Ponadto można skonfigurować następujące opcje u dołu karty podrzędnej Warunkowe:

- **Przeprowadź ewaluację według ustalonej kolejności.** Wybierz tę opcję, aby badać warunki w kolejności, w jakiej są widoczne na karcie podrzędnej. Wszystkie węzły, dla których warunki są spełnione, zostaną wykonane po zbadaniu wszystkich warunków.
- **Wykonuj warunki po kolei.** Opcja dostępna tylko wtedy, gdy wybrana jest opcja **Przeprowadź ewaluację według ustalonej kolejności**. Wybór tej opcji powoduje, że jeśli warunek jest spełniony, to węzeł powiązany z tym warunkiem zostanie wykonany przed zbadaniem następnego warunku.
- **Oceniaj do pierwszego trafienia.** Wybranie tej opcji powoduje, że uruchomiony zostanie tylko pierwszy węzeł, którego warunek będzie spełniony.

Wykonywanie i przerywanie wykonania skryptów

Wykonywanie skryptu można rozpocząć na różne sposoby. Na przykład w oknie dialogowym skryptu strumienia lub skryptu samodzielnego dostępny jest przycisk „Wykonaj ten skrypt”, który inicjuje wykonanie całego skryptu:



Rysunek 1. Przycisk Wykonaj ten skrypt

Przycisk „Wykonaj tylko zaznaczone wiersze” umożliwi wykonanie jednego wiersza lub bloku sąsiadujących ze sobą wierszy zaznaczonych w skrypcie:



Rysunek 2. Przycisk Wykonaj tylko zaznaczone wiersze

Wykonywanie skryptu można rozpocząć przy użyciu dowolnej z następujących metod:

- Kliknij przycisk „Wykonaj ten skrypt” lub „Wykonaj tylko zaznaczone wiersze” w oknie dialogowym skryptu strumienia lub skryptu samodzielnego.
- Uruchom strumień, w którym jako domyślną metodę wykonania wybrano **Wykonaj ten skrypt**.
- Użyj flagi `-execute` przy uruchamianiu programu w trybie wsadowym. Więcej informacji można znaleźć w temacie “Używanie argumentów wiersza komend” na stronie 61.

Uwaga: Skrypt superwęzła jest wykonywany, gdy wykonywany jest superwęzeł, o ile zaznaczono pole wyboru **Wykonaj ten skrypt** w oknie dialogowym skryptu superwęzła.

Przerywanie wykonywania skryptów

Podczas wykonywania skryptu w oknie dialogowym skryptu strumienia aktywny jest czerwony przycisk zatrzymania. Za jego pomocą można przerwać wykonywanie skryptu i bieżącego strumienia.

Znajdowanie i zastępowanie

Okno dialogowe Znajdź i zamień jest dostępne tylko w miejscach, w których można edytować tekst skryptu lub wyrażenia, takich jak edytor skryptów czy Konstruktor wyrażeń CLEM, lub podczas definiowania szablonu w węźle Raport. Podczas edytowania tekstu w dowolnym z tych obszarów naciśnij kombinację klawiszy Ctrl+F, aby uzyskać dostęp do okna dialogowego, upewniając się wcześniej, że kursor znajduje się w obszarze tekstowym i ma fokus. W przypadku pracy w węźle wypełniania można uzyskać dostęp do okna dialogowego z dowolnego z obszarów tekstowych na karcie Ustawienia lub ze zmiennej tekstowej w Konstruktorze wyrażeń.

1. Po umieszczeniu kursora w obszarze tekstowym naciśnij kombinację klawiszy Ctrl+F, aby uzyskać dostęp do okna dialogowego Znajdź i zamień.
2. Wprowadź tekst, który chcesz wyszukać, lub wybierz z listy rozwijanej ostatnio wyszukiwany element.
3. Wprowadź nowy tekst odpowiednio do potrzeb.
4. Kliknij opcję **Znajdź następne**, aby rozpocząć wyszukiwanie.
5. Kliknij opcję **Zamień**, aby zastąpić bieżący wybór, lub opcję **Zamień wszystko**, aby zmienić wszystkie lub tylko wybrane wystąpienia.
6. Okno dialogowe zamyka się po każdej operacji. Naciśnij klawisz F3 z poziomu dowolnego obszaru tekstowego, aby powtórzyć ostatnią operację wyszukiwania, lub naciśnij kombinację klawiszy Ctrl+F, aby ponownie uzyskać dostęp do tego okna dialogowego.

Opcje wyszukiwania

Uwzględnij wielkość liter. Określa, czy operacja wyszukiwania rozróżnia wielkość liter; na przykład czy *myvar* stanowi dopasowanie dla *myVar*. Tekst zastępujący jest wstawiany zawsze dokładnie tak, jak został wprowadzony, niezależnie od tego ustawienia.

Tylko całe słowa. Określa, czy operacja wyszukiwania dopasowuje tylko tekst odzwierciedlający całe słowa. W przypadku zaznaczenia tej opcji wyszukiwanie tekstu *spider* nie zwróci takich wyników, jak *spiderman* czy *spider-man*.

Wyrażenia regularne. Określa, czy używana jest składnia wyrażenia regularnego (zob. następna sekcja). Zaznaczenie tej opcji powoduje, że opcja **Tylko całe słowa** jest nieaktywna, a jej wartość jest ignorowana.

Tylko wybrany tekst. Wpływa na zasięg wyszukiwania w przypadku korzystania z opcji **Zamień wszystko**.

Składnia wyrażenia regularnego

Wyrażenia regularne umożliwiają wyszukiwanie znaków specjalnych, takich jak znaki tabulacji czy znaki podziału wiersza, klasy lub zakresy znaków, np. od *a* do *d*, dowolne cyfry lub znaki alfanumeryczne, oraz granice, takie jak początek czy koniec wiersza. Obsługiwane są następujące typy wyrażeń.

Tabela 1. Dopasowania znaków.

Znaki	Dopasowania
x	Znak x
\\	Ukośnik lewy
\0n	Znak o wartości ósemkowej 0n (0 <= n <= 7)
\0nn	Znak o wartości ósemkowej 0nn (0 <= n <= 7)
\0mnn	Znak o wartości ósemkowej 0mnn (0 <= m <= 3, 0 <= n <= 7)
\xhh	Znak o wartości szesnastkowej 0xhh
\uhhhh	Znak o wartości szesnastkowej 0xhhhh
\t	Znak tabulacji ('\u0009')
\n	Znak nowego wiersza (podziału wiersza) ('\u000A')
\r	Znak powrotu karetki ('\u000D')
\f	Znak nowej strony ('\u000C')
\a	Znak alertu (sygnału dźwiękowego) ('\u0007')
\e	Znak zmiany znaczenia ('\u001B')
\cx	Znak sterujący odpowiadający x

Tabela 2. Dopasowywanie klas znaków.

Klasy znaków	Dopasowania
[abc]	a, b lub c (klasa prosta)
[^abc]	Dowolny znak z wyjątkiem a, b lub c (odejmowanie)
[a-zA-Z]	Od a do z lub od A do Z włącznie (przedział)
[a-d[m-p]]	Od a do d lub od m do p (suma przedziałów). Można też przedstawić ten zapis w postaci [a-dm-p]
[a-z&&[def]]	Od a do z i d, e lub f (część wspólna)
[a-z&&[^bc]]	Od a do z, z wyjątkiem b i c (odejmowanie). Można też przedstawić ten zapis w postaci [ad-z]
[a-z&&[^m-p]]	Od a do z i nie m do p (odejmowanie). Można też przedstawić ten zapis w postaci [a-lq-z]

Tabela 3. Predefiniowane klasy znaków.

Predefiniowane klasy znaków	Dopasowania
.	Dowolny znak (może odpowiadać lub może nie odpowiadać znakom końca wiersza)
\d	Dowolna cyfra: [0-9]
\D	Dowolny znak inny niż cyfra: [^0-9]
\s	Biały znak: [\t\n\x0B\f\r]
\S	Znak inny niż biały znak: [^\s]
\w	Znak należący do słowa: [a-zA-Z_0-9]
\W	Znak nienależący do słowa: [^\w]

Tabela 4. Dopasowania granic.

Dopasowania granic	Dopasowania
^	Początek wiersza

Tabela 4. Dopasowania granic (kontynuacja).

Dopasowania granic	Dopasowania
\$	Koniec wiersza
\b	Granica słowa
\B	Granica inna niż słowa
\A	Początek danych wejściowych
\Z	Koniec danych wejściowych, z wyjątkiem finalnego znaku kończącego, o ile występuje
\z	Koniec danych wejściowych

Rozdział 2. Język skryptowy

Przegląd języka skryptowego

Środowisko skryptowe produktu IBM SPSS Modeler umożliwia tworzenie skryptów operujących na interfejsie użytkownika programu SPSS Modeler, manipulujących obiektami wynikowymi i uruchamiających komendy. Skrypty uruchamia się bezpośrednio z programu SPSS Modeler.

Skrypty dla programu IBM SPSS Modeler pisze się w języku Python. IBM SPSS Modeler używa implementacji języka Python opartej na środowisku Java, która nosi nazwę Jython. Oto najważniejsze cechy języka skryptowego:

- Format odwołań do węzłów, strumieni, projektów, wyników i innych obiektów programu IBM SPSS Modeler.
- Zestaw instrukcji lub komend skryptowych, które umożliwiają manipulowanie tymi obiektami.
- Język wyrażeń umożliwiający nadawanie wartości zmiennym, parametrom i innym obiektom.
- Obsługa komentarzy, kontynuacji i blokowych literałów tekstowych.

W kolejnych sekcjach opisano język skryptowy Python, jego implementację Jython oraz podstawy składni. Informacje te umożliwiają rozpoczęcie pisania skryptów dla programu IBM SPSS Modeler. W dalszej części podręcznika zamieszczono informacje o konkretnych właściwościach i komendach.

Python i Jython

Jython to implementacja języka skryptowego Python napisana w języku Java i zintegrowana z platformą Java. Python jest zaawansowanym językiem skryptowym zorientowanym obiektowo. Użyteczność implementacji Jython wynika z faktu, że oferuje ona zalety dojrzałego języka skryptowego, ale w odróżnieniu od języka Python działa w każdym środowisku obsługującym maszynę wirtualną Java (JVM). Oznacza to, że programista ma do dyspozycji biblioteki Java dostępne w maszynie JVM. Jednocześnie Jython umożliwia korzystanie ze składni i większości cech charakterystycznych języka Python.

Jako język skryptowy Python (i jego implementacja Jython) jest łatwy do opanowania i daje się efektywnie wykorzystać w pracy programisty, a ponadto wymaga tylko minimalnej struktury do stworzenia działającego programu. Kod można wprowadzać interaktywnie, tj. po jednym wierszu. Python jest interpretowanym językiem skryptowym; nie ma etapu prekompilacji znanego z języka Java. Programy w języku Python są po prostu plikami tekstowymi interpretowanymi w miarę wprowadzania (po zweryfikowaniu poprawności składni). Proste wyrażenia, takie jak wartości zdefiniowane, oraz bardziej złożone działania, takie jak definicje funkcji, są natychmiast wykonywane i dostępne do wykorzystania. Wszelkie zmiany wprowadzone w kodzie można szybko przetestować. Interpretacja skryptu ma jednak pewne wady. Na przykład użycie niezdefiniowanej zmiennej nie zostanie wykryte przez kompilator, a dopiero w momencie (ewentualnego) wykonania instrukcji, w której ta zmienna jest używana. W takim przypadku program można poddać edycji i uruchomić w celu zdebugowania błędu.

W języku Python wszystko, w tym wszelkie dane i kod, postrzegane jest jako obiekty. Obiektami tymi można manipulować programowo. Wybrane typy, takie jak liczby i łańcuchy, wygodnie jest traktować jako wartości, a nie obiekty; Python oferuje taką możliwość. Obsługiwana jest jedna wartość null. Ta wartość null ma nazwę zastrzeżoną `None`.

Aby zapoznać się z bardziej szczegółowym wprowadzeniem do języków Python i Jython, a także ze skryptami przykładowymi, patrz <http://www.ibm.com/developerworks/java/tutorials/j-jython1/j-jython1.html> i <http://www.ibm.com/developerworks/java/tutorials/j-jython2/j-jython2.html> .

Język skryptowy Python

Niniejszy przewodnik po języku skryptowym Python zawiera wstępne omówienie elementów, które są najczęściej stosowane w skryptach dla środowiska IBM SPSS Modeler. Przedstawiono tutaj najważniejsze pojęcia i podstawy programowania. Wiedza ta wystarczy, by rozpocząć pisanie skryptów w języku Python przeznaczonych do uruchamiania w programie IBM SPSS Modeler.

Operacje

Operatorem przypisania jest znak równości (=). Na przykład, aby przypisać wartość "3" zmiennej o nazwie "x", należałoby użyć instrukcji:

```
x = 3
```

Znak równości służy także do przypisywania danych typu łańcuchowego do zmiennych. Na przykład, aby przypisać wartość "a string value" zmiennej o nazwie "y", należałoby użyć instrukcji:

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

W następującej tabeli wymieniono niektóre często stosowane operatory porównania i operatory liczbowe wraz z opisami.

Tabela 5. Często stosowane operatory porównania i operatory liczbowe

Operacja	Opis
<code>x < y</code>	Czy x jest mniejsze od y?
<code>x > y</code>	Czy x jest większe od y?
<code>x <= y</code>	Czy x jest mniejsze lub równe y?
<code>x >= y</code>	Czy x jest większe lub równe y?
<code>x == y</code>	Czy x jest równe y?
<code>x != y</code>	Czy x jest różne od y?
<code>x <> y</code>	Czy x jest różne od y?
<code>x + y</code>	Dodaj y do x
<code>x - y</code>	Odejmij y od x
<code>x * y</code>	Pomnóż x przez y
<code>x / y</code>	Podziel x przez y
<code>x ** y</code>	Podnieś x do potęgi y

Listy

Listy są szeregami elementów. Lista może zawierać dowolną liczbę elementów, a elementy listy mogą być obiektami dowolnego typu. Listy można też koncepcyjnie przedstawić jako tablice. Liczba elementów listy może zwiększać się i zmniejszać w miarę dodawania, usuwania lub zastępowania elementów.

Przykłady

```
[]
```

Lista pusta.

```
[1]
```

Lista z jednym elementem będącym liczbą całkowitą.

```
["Mike", 10, "Don", 20]
```

Lista z czterema elementami: dwoma łańcuchami i dwiema liczbami całkowitymi.

```
[[], [7], [8, 9]]
```

Lista list. Każda podlista jest albo listą pustą, albo listą liczb całkowitych.

```
x = 7; y = 2; z = 3;
[1, x, y, x + y]
```

Lista liczb całkowitych. Ten przykład ilustruje zastosowanie zmiennych i wyrażeń.

Listę można przypisać do zmiennej, na przykład:

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

Można uzyskiwać dostęp do konkretnych elementów listy, na przykład:

```
mylist[0]
```

Wynik działania skryptu będzie następujący:

```
one
```

Liczba w nawiasach kwadratowych ([]) nazywana jest *indeksem* i wskazuje na konkretny element listy. Elementy na liście są numerowane od zera.

Można także wybrać z listy zakres elementów; operacja ta nazywana jest *wydzielaniem*. Na przykład `x[1:3]` wybiera drugi i trzeci element listy `x`.

Łańcuchy

Łańcuch to ciąg znaków o niezmiennej kolejności, który traktowany jest jak jedna wartość. Względem łańcuchów można stosować wszystkie funkcje i operatory działające na ciągach o niezmiennej kolejności, których zastosowanie skutkuje powstaniem nowego łańcucha. Na przykład `"abcdef"[1:4]` daje w wyniku `"bcd"`.

W języku Python znaki przedstawiane są jako łańcuchy o jednostkowej długości.

Literały łańcuchowe definiuje się za pomocą pojedynczych lub potrójnych cudzysłowów. Łańcuchy zdefiniowane za pomocą pojedynczych cudzysłowów nie mogą być podzielone między wiersze, natomiast łańcuchy zdefiniowane za pomocą potrójnych cudzysłowów mogą być podzielone między wiersze. Łańcuch może być ujęty w znaki apostrofu (') lub cudzysłowu ("). Cudzysłów może obejmować zawierać inny znak cudzysłowu (tj. cudzysłów w apostrofie albo apostrof w cudzysłowie) bez znaku zmiany znaczenia albo ten sam znak cudzysłowu poprzedzony znakiem zmiany znaczenia (\).

Przykłady

```
"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"
```

Analizator składni języka Python automatycznie łączy łańcuchy oddzielone białym znakiem. Ułatwia to wprowadzanie długich łańcuchów i stosowanie różnych cudzysłowów w jednym łańcuchu, na przykład:

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

Wynikowy łańcuch ma postać:

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

Obiekty łańcuchowe oferują kilka przydatnych metod. Niektóre z nich opisano w poniższej tabeli.

Tabela 6. Metody udostępniane przez łańcuchy

Metoda	Zastosowanie
<code>s.capitalize()</code>	Zmień na wielkie wszystkie pierwsze litery wyrazów łańcucha <code>s</code>
<code>s.count(ss {,start {,end}})</code>	Zlicz wystąpienia łańcucha <code>ss</code> w łańcuchu <code>s[start:end]</code>

Tabela 6. Metody udostępniane przez łańcuchy (kontynuacja)

Metoda	Zastosowanie
s.startswith(str {, start {, end}}) s.endswith(str {, start {, end}})	Sprawdź, czy s zaczyna się od str Sprawdź, czy s kończy się na str
s.expandtabs({size})	Zastąp znaki tabulacji spacjami, domyślna szerokość tabulacji (size) wynosi 8
s.find(str {, start {, end}}) s.rfind(str {, start {, end}})	Znajduje indeks początku łańcucha str w łańcuchu s; jeśli nie zostanie znaleziony, zwraca -1. rfind szuka od prawej do lewej.
s.index(str {, start {, end}}) s.rindex(str {, start {, end}})	Znajduje indeks początku łańcucha str w łańcuchu s; jeśli nie zostanie znaleziony, generuje wyjątek ValueError. rindex szuka od prawej do lewej
s.isalnum	Sprawdź, czy łańcuch zawiera tylko znaki alfanumeryczne.
s.isalpha	Sprawdź, czy łańcuch zawiera tylko znaki alfabetu.
s.isnum	Sprawdź, czy łańcuch zawiera liczbę.
s.isupper	Sprawdź, czy wszystkie litery w łańcuchu są wielkie.
s.islower	Sprawdź, czy wszystkie litery w łańcuchu są małe.
s.isspace	Sprawdź, czy wszystkie litery w łańcuchu są białymi znakami.
s.istitle	Sprawdź, czy łańcuch jest szeregiem podłańcuchów alfanumerycznych, z których każdy zaczyna się wielką literą.
s.lower() s.upper() s.swapcase() s.title()	Zamień wszystkie litery na małe. Zamień wszystkie litery na wielkie. Zamień wszystkie małe litery na wielkie, a wielkie na małe. Zamień wszystkie pierwsze litery wyrazów na wielkie.
s.join(seq)	Połącz łańcuchy tworzące listę seq, stosując łańcuch s jako separator.
s.splitlines({keep})	Podziel s na wiersze; jeśli keep =true, zachowaj znaki nowego wiersza.
s.split({sep {, max}})	Podziel s na "wyrazy", używając separatora sep (domyślnie sep jest białym znakiem) maksymalnie max razy
s.ljust(width) s.rjust(width) s.center(width) s.zfill(width)	Wyjustuj łańcuch do lewej w polu o szerokości width Wyjustuj łańcuch do prawej w polu o szerokości width Wycentrum łańcuch w polu o szerokości width Wypełnij znakami 0.
s.lstrip() s.rstrip() s.strip()	Usuń początkowy biały znak Usuń końcowy biały znak Usuń początkowy i końcowy biały znak
s.translate(str {, delc})	Przekształć s na podstawie tabeli, usuwając uprzednio wszelkie znaki występujące w delc. str powinien być łańcuchem o długości == 256.
s.replace(old, new {, max})	Zastępuje wszystkie lub max wystąpienia łańcucha old łańcuchem new

Uwagi

Uwagi są komentarzami poprzedzonymi znakiem krzyżyka (#). Cały tekst od znaku krzyżyka do końca wiersza jest uznawany za część uwagi i ignorowany. Uwaga może zaczynać się od dowolnej kolumny. Poniższy przykład ilustruje użycie uwag:

```
#HelloWorld to jedna z najprostszych aplikacji
print 'Hello World' # drukuj tekst Hello World
```


Składnia instrukcji

Składnia instrukcji języka Python jest bardzo prosta. Co do zasady każdy wiersz kodu źródłowego jest jedną instrukcją. Każda instrukcja, z wyjątkiem **wyrażeń** i **przypisań**, zaczyna się od słowa kluczowego, takiego jak **if** lub **for**. W dowolnym miejscu między instrukcjami w kodzie można wstawiać puste wiersze i uwagi. Jeśli jeden wiersz ma zawierać więcej niż jedną instrukcję, to należy je rozdzielić średnikami (;).

Bardzo długie instrukcje mogą zajmować więcej niż jeden wiersz. W tym przypadku wiersz instrukcji kontynuowanej w następnym wierszu musi kończyć się ukośnikiem odwrotnym (\):

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

Gdy struktura programowa jest ujęta w nawiasy okrągłe (()), kwadratowe ([]) lub klamrowe ({}), instrukcja może być kontynuowana w następnym wierszu po dowolnym przecinku, bez konieczności dodawania ukośnika odwrotnego, na przykład:

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

Identyfikatory

Identyfikatory służą do nazywania zmiennych, funkcji, klas i słów kluczowych. Identyfikator może mieć dowolną długość, ale musi zaczynać się od wielkiej lub małej litery albo od znaku podkreślenia (_). Nazwy rozpoczynające się od znaku podkreślenia są zasadniczo zarezerwowane jako nazwy wewnętrzne lub prywatne. Po pierwszym znaku identyfikatora może nastąpić dowolna liczba i kombinacja znaków alfabetycznych, cyfr od 0 do 9 i znaków podkreślenia.

W implementacji Python istnieją pewne słowa zastrzeżone, których nie można używać jako nazw zmiennych, funkcji i klas. Słowa te są podzielone na następujące kategorie:

- **Słowa wprowadzające instrukcje:** `assert`, `break`, `class`, `continue`, `def`, `del`, `elif`, `else`, `except`, `exec`, `finally`, `for`, `from`, `global`, `if`, `import`, `pass`, `print`, `raise`, `return`, `try` i `while`
- **Słowa wprowadzające parametry:** `as`, `import` i `in`
- **Operatory:** `and`, `in`, `is`, `lambda`, `not` i `or`

Niewłaściwe użycie słowa kluczowego z reguły powoduje zgłoszenie wyjątku `SyntaxError`.

Bloki kodu

Bloki kodu są grupami instrukcji używanymi w miejsce pojedynczych instrukcji. Blok kodu może nastąpić po dowolnej z następujących instrukcji: `if`, `elif`, `else`, `for`, `while`, `try`, `except`, `def` i `class`. Znak dwukropka (:) po jednej z tych instrukcji oznacza początek bloku kodu, na przykład:

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

Granice bloków kodu wyznaczone są przez wcięcia (a nie nawiasy klamrowe, jak w języku Java). Wszystkie wiersze w bloku muszą być wcięte na tę samą głębokość. Wynika to z faktu, że jakkolwiek zmiana wcięcia sygnalizuje koniec bloku kodu. Zwyczajowo stosuje się wcięcie o szerokości czterech spacji na jeden poziom. Zaleca się tworzenie wcięć za pomocą spacji, a nie znaków tabulacji. Nie wolno stosować jednocześnie i spacji, i znaków tabulacji. Wiersze w najbardziej zewnętrznym bloku modułu muszą rozpoczynać się w pierwszej kolumnie, w przeciwnym razie zostanie zgłoszony wyjątek `SyntaxError`.

Instrukcje składające się na blok kodu (po dwukropku) mogą także znajdować się w jednym wierszu i być rozdzielone średnikami, na przykład:

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

Przekazywanie argumentów do skryptu

Przekazywanie argumentów do skryptu umożliwia użycie tego samego skryptu w różnych okolicznościach. Argumenty przekazane z wiersza komend są dostępne jako wartości na liście `sys.argv`. Liczbę przekazanych wartości można ustalić za pomocą komendy `len(sys.argv)`. Na przykład:

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

W tym przykładzie komenda `import` importuje całą klasę `sys`, aby można było używać metod zdefiniowanych w tej klasie, takich jak `argv`.

Przykładowy skrypt można wywołać za pomocą komendy:

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

Wynik wykonania będzie następujący:

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

Przykłady

Słowo kluczowe `print` powoduje wydrukowanie (wyprowadzenie na wyjście) argumentów, które bezpośrednio po nim następują. Jeśli po instrukcji znajduje się przecinek, w wynikach nie jest wprowadzany znak nowego wiersza. Na przykład:

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

Wynik działania skryptu będzie następujący:

Ten przykład ilustruje użycie przecinka na końcu instrukcji `print`.

Instrukcja `for` służy do iteracyjnego wykonywania bloku kodu. Na przykład:

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

W tym przykładzie do listy `mylist1` przypisywane są trzy łańcuchy. Następnie elementy listy są drukowane, tak by każdy element znalazł się w osobnym wierszu. Wynik działania skryptu będzie następujący:

```
one
two
three
```

W tym przykładzie iterator `lv` przyjmuje kolejno wartości poszczególnych elementów listy `mylist1`, a dla każdego elementu wykonywany jest blok kodu. Iterator może być dowolnym poprawnym identyfikatorem o dowolnej długości.

Instrukcja `if` jest instrukcją warunkową. Bada ona spełnienie warunku i zwraca `true` albo `false`, w zależności od tego, czy jest spełniony. Na przykład:

```
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
```

W tym przykładzie badana jest wartość iteratora `lv`. Jeśli `lv` jest równy `two`, zwracany jest inny łańcuch niż w przypadku, gdy `lv` nie jest równy `two`. Wynikowy łańcuch ma postać:

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

Metody matematyczne

Moduł `math` zawiera użyteczne metody matematyczne. Niektóre z nich opisano w poniższej tabeli. O ile nie zaznaczono inaczej, wszystkie zwracane wartości są liczbami zmiennopozycyjnymi.

Tabela 7. Metody matematyczne

Metoda	Zastosowanie
<code>math.ceil(x)</code>	Zwraca, jako wartość zmiennopozycyjną, najmniejszą liczbę całkowitą nie mniejszą niż x .
<code>math.copysign(x, y)</code>	Zwraca x ze znakiem argumentu y . <code>copysign(1, -0.0)</code> zwraca <code>-1</code>
<code>math.fabs(x)</code>	Zwraca wartość bezwzględną x
<code>math.factorial(x)</code>	Zwraca silnię x . Jeśli x jest liczbą ujemną lub niecałkowitą, generowany jest wyjątek <code>ValueError</code> .
<code>math.floor(x)</code>	Zwraca, jako wartość zmiennopozycyjną, największą liczbę całkowitą nie większą niż x .
<code>math.frexp(x)</code>	Zwraca mantysę (m) i wykładnik (e) argumentu x jako parę (m , e). m jest taką liczbą zmiennopozycyjną, a e jest taką liczbą całkowitą, że dokładnie spełniona jest zależność $x == m * 2**e$. Jeśli x jest równe zero, zwraca <code>(0.0, 0)</code> , w przeciwnym razie $0.5 <= \text{abs}(m) < 1$.
<code>math.fsum(iterable)</code>	Zwraca dokładną zmiennopozycyjną sumę wartości zawartych w argumentcie <code>iterable</code>
<code>math.isinf(x)</code>	Sprawdza, czy zmiennopozycyjny argument x jest nieskończonością dodatnią lub ujemną
<code>math.isnan(x)</code>	Sprawdza, czy zmiennopozycyjny argument x jest wartością nieliczbową <code>NaN</code>
<code>math.ldexp(x, i)</code>	Zwraca $x * (2**i)$. Jest to zasadniczo odwrócenie działania funkcji <code>frexp</code> .
<code>math.modf(x)</code>	Zwraca część ułamkową i całkowitą argumentu x . Oba wyniki mają znak argumentu x i są liczbami zmiennopozycyjnymi.
<code>math.trunc(x)</code>	Zwraca wartość rzeczywistą argumentu x , która została obcięta do części całkowitej.
<code>math.exp(x)</code>	Zwraca $e**x$
<code>math.log(x[, base])</code>	Zwraca logarytm x o podstawie <code>base</code> . Jeśli wartość <code>base</code> nie jest określona, zwraca logarytm naturalny x .
<code>math.log1p(x)</code>	Zwraca logarytm naturalny $1+x$ (o podstawie e)
<code>math.log10(x)</code>	Zwraca logarytm x o podstawie 10
<code>math.pow(x, y)</code>	Zwraca x do potęgi y . <code>pow(1.0, x)</code> i <code>pow(x, 0.0)</code> zawsze zwracają <code>1</code> , nawet gdy x jest równe zero lub <code>NaN</code> .
<code>math.sqrt(x)</code>	Zwraca pierwiastek kwadratowy x

Oprócz funkcji matematycznych dostępne są także użyteczne funkcje trygonometryczne. Zestawienie odpowiednich metod przedstawiono w poniższej tabeli.

Tabela 8. Metody trygonometryczne

Metoda	Zastosowanie
math.acos(x)	Zwraca arcus cosinus x w radianach
math.asin(x)	Zwraca arcus sinus x w radianach
math.atan(x)	Zwraca arcus tangens x w radianach
math.atan2(y, x)	Zwraca atan(y / x) w radianach
math.cos(x)	Zwraca cosinus x w radianach.
math.hypot(x, y)	Zwraca normalną euklidesową $\sqrt{x^2 + y^2}$. Jest to długość wektora od początku do punktu (x, y).
math.sin(x)	Zwraca sinus x w radianach
math.tan(x)	Zwraca tangens x w radianach
math.degrees(x)	Przelicza kąt x z radianów na stopnie
math.radians(x)	Przelicza kąt x ze stopni na radiany
math.acosh(x)	Zwraca odwrotny cosinus hiperboliczny x
math.asinh(x)	Zwraca odwrotny sinus hiperboliczny x
math.atanh(x)	Zwraca odwrotny tangens hiperboliczny x
math.cosh(x)	Zwraca cosinus hiperboliczny x
math.sinh(x)	Zwraca sinus hiperboliczny x
math.tanh(x)	Zwraca tangens hiperboliczny x

Dostępne są także dwie stałe matematyczne. Wartość `math.pi` równa jest matematycznej stałej pi. Wartość `math.e` równa jest matematycznej stałej e.

Korzystanie ze znaków spoza zestawu ASCII

Korzystanie ze znaków spoza zestawu ASCII w języku Python jest możliwe pod warunkiem zastosowania jawnego kodowania i dekodowania łańcuchów zgodnie ze standardem Unicode. W programie IBM SPSS Modeler przyjmuje się, że skrypty w języku Python są zakodowane w standardzie Unicode UTF-8, który obsługuje znaki spoza zestawu ASCII. Następujący skrypt zostanie prawidłowo skompilowany, ponieważ SPSS Modeler skonfigurował kompilator Python do pracy z kodowaniem UTF-8.

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

Jednak wynikowy węzeł będzie miał nieprawidłową etykietę.



ãfã, 'ãf^ãf ãf%ãf%

Rysunek 3. Etykieta węzła zawierająca znaki spoza zestawu ASCII, wyświetlana nieprawidłowo

Etykieta jest nieprawidłowa, ponieważ literał łańcuchowy został przekształcony w łańcuch ASCII przez środowisko Python.

Python dopuszcza podawanie literałów łańcuchowych Unicode, pod warunkiem poprzedzenia takiego literału znakiem u:

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

Spowoduje to utworzenie łańcucha Unicode i prawidłowe wyświetlanie etykiety.



Rysunek 4. Etykieta węzła zawierająca znaki spoza zestawu ASCII, wyświetlana prawidłowo

Korzystanie z łańcuchów Unicode w środowisku Python jest obszernym zagadnieniem, które wykracza poza tematykę niniejszej publikacji. Dostępne są liczne książki i materiały elektroniczne, w których temat ten został omówiony bardziej szczegółowo.

Programowanie zorientowane obiektowo

Istotą programowania zorientowanego obiektowo jest stworzenie w programie modelu rozwiązywanego problemu. Programowanie zorientowane obiektowo zmniejsza ryzyko popełnienia błędu przy pisaniu programu i sprzyja wielokrotnemu wykorzystaniu raz napisanego kodu. Python jest językiem programowania zorientowanym obiektowo. Obiekty zdefiniowane w języku Python mają następujące cechy:

- **Tożsamość.** Każdy obiekt musi być odrębny i musi być możliwość sprawdzenia, czy faktycznie jest odrębny. Służą do tego operatory `is` i `is not`.
- **Stan.** Musi istnieć możliwość zapisania stanu w każdym obiekcie. Służą temu atrybuty, takie jak zmienne, w tym zmienne lokalne instancji.
- **Zachowanie.** Każdy obiekt musi mieć zdolność manipulowania swoim stanem. Służą do tego metody.

Następujące cechy języka Python pomagają w programowaniu zorientowanym obiektowo:

- **Tworzenie obiektów na podstawie klas.** Klasy są szablonami do tworzenia obiektów. Obiekty i struktury danych, z którymi są skojarzone zachowania.
- **Dziedziczenie i polimorfizm.** Język Python oferuje mechanizm dziedziczenia pojedynczego i wielokrotnego. W języku Python wszystkie metody instancji są polimorficzne i mogą być przesłaniane przez podklasy.
- **Hermetyzacja z ukrywaniem danych.** Python umożliwia ukrywanie atrybutów. Ukryte atrybuty są dostępne z zewnątrz klasy wyłącznie za pośrednictwem metod tej klasy. Klasy implementują metody służące do modyfikacji danych.

Definiowanie klasy

W klasie języka Python mogą być zdefiniowane zarówno zmienne, jak i metody. Inaczej niż w języku Java, w języku Python można zdefiniować dowolną liczbę klas publicznych w jednym pliku źródłowym (lub *module*). Dlatego moduł w języku Python jest koncepcyjnie zbliżony do pakietu w języku Java.

W języku Python klasy definiuje się przy użyciu instrukcji `class`. Instrukcja `class` ma następującą postać:

```
class nazwa (nadklasy): instrukcja
```

lub

```
class nazwa (nadklasy):
    assignment
```

```
.  
function  
.  
.
```

Definiując klasę, można użyć instrukcji *assignment* (jednej lub wielu) bądź ich nie używać. Instrukcje te tworzą atrybuty wspólne dla wszystkich instancji klasy. Można też podać definicje funkcji *function* (jednej lub wielu) bądź ich nie podawać. Definicje funkcji tworzą metody. Lista nadklas jest opcjonalna.

Nazwa klasy powinna być unikalna w swoim zasięgu, tj. w module, funkcji lub klasie. Można zdefiniować wiele zmiennych odwołujących się do tej samej klasy.

Tworzenie instancji klasy

Klas używa się do przechowywania atrybutów klasy (wspólnych) lub do tworzenia instancji klasy. Aby utworzyć instancję klasy, wywołuje się klasę tak, jak gdyby była funkcją. Rozważmy na przykład następującą klasę:

```
class MyClass:  
    pass
```

Użyto tutaj instrukcji `pass`, ponieważ instrukcja jest wymagana dla kompletności klasy, ale nie jest wymagane wykonywanie żadnych działań.

Następująca instrukcja tworzy instancję klasy `MyClass`:

```
x = MyClass()
```

Dodawanie atrybutów do instancji klasy

W języku Python, inaczej niż w języku Java, klient może dodawać atrybuty do instancji klasy. Zmieniana jest wówczas tylko jedna instancja. Na przykład, aby dodać atrybuty do instancji `x`, należy ustawić nowe wartości w tej instancji:

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

Definiowanie atrybutów i metod klas

Dowolna zmienna powiązana z klasą jest *atrybutem klasy*. Dowolna funkcja zdefiniowana wewnątrz klasy jest *metodą*. Pierwszym argumentem każdej metody jest instancja klasy, zwyczajowo nazywana `self`. Na przykład, aby zdefiniować kilka atrybutów i metod klasy, można wprowadzić następujący kod:

```
class MyClass  
    attr1 = 10          #atrybuty klasy  
    attr2 = "hello"  
  
    def method1(self):  
        print MyClass.attr1    #odwołanie do atrybutu klasy  
  
    def method2(self):  
        print MyClass.attr2    #odwołanie do atrybutu klasy  
  
    def method3(self, text):  
        self.text = text        #atrybut instancji  
        print text, self.text    #wydrukuj argument i atrybut  
  
    method4 = method3    #utwórz alias dla method3
```

Wewnątrz klasy wszystkie odwołania do jej atrybutów muszą być kwalifikowane nazwą klasy, na przykład `MyClass.attr1`. Wszystkie odwołania do atrybutów instancji powinny być kwalifikowane zmienną `self`, na przykład `self.text`. Na zewnątrz klasy wszystkie odwołania do atrybutów klasy powinny być kwalifikowane nazwą klasy (na przykład `MyClass.attr1`) lub instancją klasy (na przykład `x.attr1`, gdzie `x` jest instancją klasy). Na zewnątrz klasy wszystkie odwołania do zmiennych instancji powinny być kwalifikowane instancją klasy, na przykład `x.text`.

Zmienne ukryte

Dane można ukrywać, tworząc zmienne *prywatne*. Do zmiennych prywatnych może uzyskiwać dostęp tylko sama klasa. Jeśli nazwy zostaną zadeklarowane w postaci `__xxx` or `__xxx_yyy`, tj. z dwoma początkowymi znakami podkreślenia, to analizator składni języka Python automatycznie doda nazwę klasy do nazwy zadeklarowanej, tworząc zmienne ukryte, na przykład:

```
class MyClass:
    __attr = 10    #atrybut klasy prywatnej

    def method1(self):
        pass

    def method2(self, p1, p2):
        pass

    def __privateMethod(self, text):
        self.__text = text    #atrybut prywatny
```

W języku Python, inaczej niż w języku Java, wszystkie odwołania do zmiennych instancji muszą być kwalifikowane nazwą `self`; nie stosuje się niejawnego kwalifikatora `this`.

Dziedziczenie

Możliwość dziedziczenia z klas jest fundamentem programowania zorientowanego obiektowo. Język Python oferuje mechanizm dziedziczenia pojedynczego i wielokrotnego. W przypadku *dziedziczenia pojedynczego* może istnieć tylko jedna nadklasa. W przypadku *dziedziczenia wielokrotnego* może istnieć więcej niż jedna nadklasa.

Dziedziczenie implementuje się, tworząc podklasy innych klas. Dowolna liczba klas języka Python może być nadklasami. Jednak w implementacji Python możliwe jest bezpośrednio lub pośrednio dziedziczenie tylko z jednej klasy języka Java. Nie jest wymagane określenie nadklasy.

Podklasa lub klient może używać dowolnych nieukrytych atrybutów i metod z nadklasy. Wszędzie tam, gdzie można użyć instancji nadklasy, można także użyć instancji podklasy; jest to przykład *polimorfizmu*. Mechanizmy te umożliwiają wielokrotne wykorzystanie tego samego kodu i ułatwiają rozbudowę oprogramowania.

Przykład

```
class Class1: pass    #bez dziedziczenia

class Class2: pass

class Class3(Class1): pass    #jednokrotne dziedziczenie

class Class4(Class3, Class2): pass    #wielokrotne dziedziczenie
```

Rozdział 3. Skrypty w programie IBM SPSS Modeler

Typy skryptów

W programie IBM SPSS Modeler wyróżnia się trzy typy skryptów:

- *Skrypty strumieni* służą do sterowania wykonaniem pojedynczych strumieni i są w nich zapisane.
- *Skrypty superwęzłów* służą do sterowania zachowaniem superwęzłów.
- *Skrypty samodzielne lub sesyjne* można stosować do koordynacji wykonania wielu różnych strumieni.

W programie IBM SPSS Modeler można używać różnych metod zapewniających dostęp do szeregu funkcji programu SPSS Modeler. Metody te są także używane w rozdziale Rozdział 4, “Skryptowy interfejs API”, na stronie 37 do tworzenia bardziej zaawansowanych funkcji.

Strumienie, strumienie superwęzłów i diagramy

Znaczenie terminu *strumień* jest zwykle takie samo, niezależnie od tego, czy mówimy o strumieniu załadowanym z pliku, czy używanym wewnątrz superwęzła. Ogólnie rzecz ujmując, strumień jest zbiorem węzłów połączonych ze sobą nawzajem, które można wykonać. Jednak w skryptach nie wszystkie operacje są dostępne we wszystkich miejscach/kontekstach, zatem autor skryptu musi zwracać uwagę na to, na którym wariantcie strumienia operuje kod.

Strumienie

Strumień jest głównym typem dokumentów programu IBM SPSS Modeler. Strumienie można zapisywać, ładować, edytować i wykonywać. Ze strumieniami mogą być skojarzone parametry, wartości globalne, skrypt i inne informacje.

Strumienie superwęzłów

Strumień superwęzła jest szczególnym rodzajem strumienia używanym wewnątrz superwęzła. Podobnie jak zwykły strumień, zawiera węzły połączone ze sobą nawzajem. Jednak strumienie superwęzłów pod wieloma względami różnią się od zwykłych strumieni:

- Parametry i ewentualne skrypty są skojarzone superwęzłem, do którego należy strumień superwęzła, a nie z samym strumieniem.
- Strumienie niektórych typów superwęzłów mają dodatkowe wejściowe i wyjściowe węzły łącznikowe. Te węzły łącznikowe służą do wprowadzania i wyprowadzania informacji do/z strumienia superwęzła i są tworzone automatycznie, gdy tworzony jest superwęzeł.

Diagramy

Termin *diagram* obejmuje funkcje, które są obsługiwane zarówno przez zwykłe strumienie, jak i strumienie klasy SuperNode, takie jak dodawanie i usuwanie węzłów oraz modyfikowanie połączeń między węzłami.

Wykonywanie strumienia

Poniższy przykład uruchamia wszystkie wykonywalne węzły w strumieniu i jest najprostszym możliwym skryptem strumienia:

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

Poniższy przykład również uruchamia wszystkie wykonywalne węzły w strumieniu:

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

W tym przykładzie strumień jest zapisany w zmiennej o nazwie **stream**. Zapisanie strumienia w zmiennej jest użyteczne, ponieważ skrypt zwykle służy do modyfikowania strumienia albo zawartych w nim węzłów. Utworzenie zmiennej, w której zapisany będzie strumień, skraca zapis kodu w skrypcie.

Kontekst skryptu

Moduł `modeler.script` udostępnia kontekst, w którym wykonywany jest skrypt. W czasie wykonywania moduł ten jest automatycznie importowany do każdego skryptu w programie SPSS Modeler. Moduł definiuje cztery funkcje zapewniające skrypcie dostęp do środowiska wykonawczego:

- Funkcja `session()` zwraca sesję wykonania skryptu. Sesja określa takie informacje, jak ustawienia regionalne i zaplecze programu SPSS Modeler (proces lokalny albo sieciowy serwer SPSS Modeler Server) używane do wykonywania strumieni.
- Funkcja `stream()` może być używana w skryptach strumieni i superwęzłów. Funkcja ta zwraca strumień, do którego należy albo skrypt strumienia, albo skrypt superwęzła, który jest obecnie wykonywany.
- Funkcja `diagram()` może być używana w skryptach superwęzłów. Funkcja ta zwraca diagram w superwęźle. W skryptach innego typu funkcja działa tak samo, jak `stream()`.
- Funkcja `supernode()` może być używana w skryptach superwęzłów. Funkcja ta zwraca superwęzeł, do którego należy właśnie wykonywany skrypt.

Poniższa tabela zawiera podsumowanie tych czterech funkcji i ich wyników.

Tabela 9. Podsumowanie funkcji modułu `modeler.script`

Typ skryptu	<code>session()</code>	<code>stream()</code>	<code>diagram()</code>	<code>supernode()</code>
Standalone	Zwraca sesję	Zwraca strumień zarządzany w momencie wywołania skryptu (na przykład strumień przekazany za pomocą opcji <code>-stream</code> trybu wsadowego) albo <code>None</code> .	Działa tak samo, jak <code>stream()</code>	Nie dotyczy
Strumień	Zwraca sesję	Zwraca strumień	Działa tak samo, jak <code>stream()</code>	Nie dotyczy
Superwęzeł	Zwraca sesję	Zwraca strumień	Zwraca strumień superwęzła	Zwraca superwęzeł

W module `modeler.script` zdefiniowano również sposób zakończenia działania skryptu i zwrócenia kodu wyjścia. Funkcja `exit(exit-code)` przerywa wykonywanie skryptu i zwraca całkowitoliczbowy kod wyjścia przekazany jako argument.

Jedną z metod strumienia jest `runAll(List)`. Ta metoda uruchamia wszystkie wykonywalne węzły. Wszelkie modele lub wyniki wygenerowane w trakcie wykonywania węzłów są dodawane do listy przekazanej jako argument.

Wykonanie strumienia prowadzi często do wygenerowania wyników, takich jak modele, wykresy i inne obiekty wynikowe. Aby odebrać te wyniki, skrypt może przekazać zmienną, która zostanie zainicjowana do listy, na przykład:

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

Po zakończeniu wykonywania wszelkie wygenerowane obiekty są dostępne na liście `results`.

Odwołania do istniejących węzłów

Często skrypt operuje na strumieniu zbudowanym wcześniej i zawierającym parametry, które muszą zostać zmodyfikowane przed uruchomieniem strumienia. Aby zmodyfikować takie parametry, należy:

1. Odszukać węzły w odpowiednim strumieniu.
2. Zmienić ustawienia węzłów i/lub strumienia.

Znajdowanie węzłów

Istniejący węzeł można odszukać w strumieniu na wiele sposobów. Zestawienie tych metod przedstawiono w poniższej tabeli.

Tabela 10. Metody wyszukiwania istniejącego węzła

Metoda	Typ zwracanego wyniku	Opis
s.findAll(type, label)	Zbiór	Zwraca listę wszystkich węzłów określonego typu mających określoną etykietę. Jako typ albo jako etykietę można podać None , a wówczas używany będzie drugi parametr.
s.findAll(filter, recursive)	Zbiór	Zwraca zbiór wszystkich węzłów akceptowanych przez określony filtr. Jeśli flaga recursive ma wartość True , przeszukiwane są także wszystkie superwęzły w określonym strumieniu.
s.findById(id)	Węzeł	Zwraca węzeł o podanym identyfikatorze lub None , jeśli taki węzeł nie istnieje. Wyszukiwanie jest ograniczone do bieżącego strumienia.
s.findByType(type, label)	Węzeł	Zwraca węzeł o podanym typie i/lub podanej etykiecie. Jako typ albo jako nazwę można podać None , a wówczas używany będzie drugi parametr. Jeśli kryterium jest spełnione przez więcej niż jeden węzeł, to arbitralnie wybierany i zwracany jest jeden z nich. Jeśli żaden węzeł nie spełnia kryterium, to zwracana jest wartość None .
s.findDownstream(fromNodes)	Zbiór	Wyszukuje węzły znajdujące się w strumieniu wcześniej niż węzły przekazane jako argument. Zwrócona lista zawiera także węzły przekazane jako argument.
s.findUpstream(fromNodes)	Zbiór	Wyszukuje węzły znajdujące się w strumieniu dalej niż węzły przekazane jako argument. Zwrócona lista zawiera także węzły przekazane jako argument.

Na przykład, jeśli strumień zawiera jeden węzeł filtrowania, do którego skrypt musi uzyskać dostęp, to ten węzeł można znaleźć za pomocą następującego kodu:

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

Jeśli znany jest identyfikator węzła (widoczny na karcie Adnotacje w oknie dialogowym węzła), to można go wykorzystać do znalezienia węzła, na przykład:

```

stream = modeler.script.stream()
node = stream.findByID("id32FJT71G2") # identyfikator węzła filtrowania
...

```

Ustawianie właściwości

Węzły, strumienie, modele i wyniki mają właściwości, które można odczytywać i w większości przypadków także ustawiać (tj. można nadawać im wartości). Właściwości służą zwykle do modyfikowania zachowania lub wyglądu obiektu. Poniższa tabela zawiera zestawienie metod służących do odczytywania i ustawiania właściwości obiektów.

Tabela 11. Metody służące do odczytywania i ustawiania właściwości obiektów

Metoda	Typ zwracanego wyniku	Opis
<code>p.getPropertyValue(propertyName)</code>	Obiekt	Zwraca wartość właściwości o określonej nazwie lub <code>None</code> , jeśli taka właściwość nie istnieje.
<code>p.setPropertyValue(propertyName, value)</code>	Nie dotyczy	Ustawia wartość właściwości o podanej nazwie.
<code>p.setPropertyValues(properties)</code>	Nie dotyczy	Ustawia wartości właściwości o podanych nazwach. Każdy wpis w mapie właściwości składa się z klucza reprezentującego nazwę właściwości oraz wartości, która ma być przypisana do tej właściwości.
<code>p.getKeyedPropertyValue(propertyName, keyName)</code>	Obiekt	Zwraca wartość właściwości o określonej nazwie i kluczu lub <code>None</code> , jeśli taka właściwość nie istnieje.
<code>p.setKeyedPropertyValue(propertyName, keyName, value)</code>	Nie dotyczy	Ustawia wartość właściwości o podanej nazwie i kluczu.

Na przykład, jeśli chcemy ustawić wartość węzła Plik zmienny na początku strumienia, można użyć następującego kodu:

```

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

```

Można też odfiltrować zmienną z węzła filtrowania. W tym przypadku wartość jest kluczowana nazwą zmiennej, na przykład:

```

stream = modeler.script.stream()
# Odszukaj węzeł filtrowania...
node = stream.findByType("filter", None)
# ... i odfiltruj zmienną "Na"
node.setKeyedPropertyValue("include", "Na", False)

```

Tworzenie węzłów i modyfikowanie strumieni

W niektórych sytuacjach konieczne jest dodawanie nowych węzłów do istniejących strumieni. Zwykle proces dodawania węzłów do istniejących strumieni obejmuje następujące zadania:

1. Utworzenie węzłów.
2. Utworzenie połączeń między nowymi węzłami a istniejącą strukturą strumienia.

Tworzenie węzłów

Istnieje wiele sposobów tworzenia węzłów w strumieniu. Zestawienie tych metod przedstawiono w poniższej tabeli.

Tabela 12. Metody służące do tworzenia węzłów

Metoda	Typ zwracanego wyniku	Opis
s.create(nodeType, name)	Węzeł	Tworzy węzeł określonego typu i dodaje go do określonego strumienia.
s.createAt(nodeType, name, x, y)	Węzeł	Tworzy węzeł określonego typu i dodaje go do określonego strumienia w określonym miejscu. Jeśli $x < 0$ lub $y < 0$, miejsce nie jest określone.
s.createModelApplier(modelOutput, name)	Węzeł	Tworzy węzeł modelu do zastosowania wyliczany z podanego obiektu wynikowego modelu.

Na przykład, aby utworzyć nowy węzeł Typ w strumieniu, można użyć następującego kodu:

```
stream = modeler.script.stream()
# Utwórz nowy węzeł Typ
node = stream.create("type", "My Type")
```

Tworzenie i usuwanie połączeń między węzłami

Nowy węzeł utworzony w strumieniu trzeba włączyć do sekwencji węzłów, aby można było z niego w praktyce korzystać. Istnieje wiele sposobów tworzenia i usuwania połączeń między węzłami w strumieniu. Zestawienie tych metod przedstawiono w poniższej tabeli.

Tabela 13. Metody służące do tworzenia i usuwania połączeń między węzłami

Metoda	Typ zwracanego wyniku	Opis
s.link(source, target)	Nie dotyczy	Tworzy nowe połączenie między węzłem źródłowym a docelowym.
s.link(source, targets)	Nie dotyczy	Tworzy nowe połączenia między węzłem źródłowym a wszystkimi węzłami docelowymi z przekazanej listy.
s.linkBetween(inserted, source, target)	Nie dotyczy	Umieszcza węzeł między dwiema innymi instancjami węzłów (źródłowym i docelowym), tworząc połączenia z tymi węzłami. Wcześniej usuwa ewentualne połączenia między węzłem źródłowym a docelowym.
s.linkPath(path)	Nie dotyczy	Tworzy nową ścieżkę między instancjami węzłów. Pierwszy węzeł jest łączony z drugim, drugi z trzecim i tak dalej.
s.unlink(source, target)	Nie dotyczy	Usuwa bezpośrednie połączenie między węzłem źródłowym a docelowym.
s.unlink(source, targets)	Nie dotyczy	Usuwa bezpośrednie połączenia między węzłem źródłowym a wszystkimi węzłami docelowymi na przekazanej liście.
s.unlinkPath(path)	Nie dotyczy	Usuwa ścieżkę istniejącą między instancjami węzłów.
s.disconnect(node)	Nie dotyczy	Usuwa połączenia między przekazanym węzłem a innymi węzłami w określonym strumieniu.

Tabela 13. Metody służące do tworzenia i usuwania połączeń między węzłami (kontynuacja)

Metoda	Typ zwracanego wyniku	Opis
s.isValidLink(source, target)	<i>boolean</i>	Zwraca True, jeśli dozwolone byłoby utworzenie połączenia między określonym węzłem źródłowym a docelowym. Ta metoda sprawdza, czy oba obiekty należą do określonego strumienia, czy z węzła źródłowego może wychodzić połączenie, czy do węzła docelowego może dochodzić połączenie, i czy utworzenie takiego połączenia nie spowoduje utworzenie zamkniętej pętli w strumieniu.

Poniższy przykładowy strumień wykonuje pięć następujących zadań:

1. Tworzy węzeł wejściowy Plik zmienny, węzeł filtrowania i węzeł wynikowy Tabela.
2. Łączy te węzły ze sobą nawzajem.
3. Ustawia nazwę pliku w węźle wejściowym Plik zmienny.
4. Odfiltrowuje zmienną "Drug" z wyników.
5. Wykonuje węzeł Tabela.

```
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)
```

Importowanie, zastępowanie i usuwanie węzłów

Oprócz tworzenia i łączenia węzłów często konieczne jest zastępowanie i usuwanie węzłów ze strumienia. Poniższa tabela zawiera zestawienie metod służących do importowania, zastępowania i usuwania węzłów.

Tabela 14. Metody służące do importowania, zastępowania i usuwania węzłów

Metoda	Typ zwracanego wyniku	Opis
s.replace(originalNode, replacementNode, discardOriginal)	Nie dotyczy	Zastępuje określony węzeł w określonym strumieniu. Zarówno pierwotny, jak i nowy węzeł muszą należeć do określonego strumienia.

Tabela 14. Metody służące do importowania, zastępowania i usuwania węzłów (kontynuacja)

Metoda	Typ zwracanego wyniku	Opis
s.insert(source, nodes, newIDs)	Lista	Wstawia kopie węzłów z przekazanej listy. Zakłada się, że wszystkie węzły na przekazanej liście są zawarte w określonym strumieniu. Flaga newIDs wskazuje, czy należy dla każdego węzła generować nowe identyfikatory czy istniejące identyfikatory powinny być skopiowane i ponownie wykorzystane. Zakłada się, że każdy węzeł w strumieniu ma unikalny identyfikator, dlatego tę flagę należy ustawić na True , jeśli strumień źródłowy jest taki sam, jak określony. Ta metoda zwraca listę nowo wstawionych węzłów, przy czym ich kolejność jest nieokreślona (tj. nie musi być taka sama, jak kolejność, w jakiej węzły występują na liście wejściowej).
s.delete(node)	Nie dotyczy	Usuwa określony węzeł z określonego strumienia. Węzeł musi należeć do określonego strumienia.
s.deleteAll(nodes)	Nie dotyczy	Usuwa wszystkie określone węzły z określonego strumienia. Wszystkie węzły w zbiorze muszą należeć do określonego strumienia.
s.clear()	Nie dotyczy	Usuwa wszystkie węzły z określonego strumienia.

Przechodzenie przez węzły w strumieniu

Jednym z typowych zadań jest ustalanie, które węzły w strumieniu znajdują się przed, a które za określonym węzłem. Strumień oferuje szereg metod umożliwiających uzyskanie takich informacji. Zestawienie tych metod przedstawiono w poniższej tabeli.

Tabela 15. Metody do ustalania, które węzły znajdują się przed, a które za określonym węzłem

Metoda	Typ zwracanego wyniku	Opis
s.iterator()	Iterator	Zwraca iterator obiektów węzła zawartych w określonym strumieniu. Jeśli strumień zostanie zmodyfikowany pomiędzy wywołaniami funkcji next() , to zachowanie iteratora będzie nieokreślone.
s.predecessorAt(node, index)	Węzeł	Zwraca określony bezpośredni poprzednik przekazanego węzła lub None , jeśli numer jest poza dopuszczalnym zakresem.
s.predecessorCount(node)	<i>int</i>	Zwraca liczbę bezpośrednich poprzedników przekazanego węzła.
s.predecessors(node)	Lista	Zwraca bezpośrednie poprzedniki przekazanego węzła.
s.successorAt(node, index)	Węzeł	Zwraca określony bezpośredni następnik przekazanego węzła lub None , jeśli numer jest poza dopuszczalnym zakresem.
s.successorCount(node)	<i>int</i>	Zwraca liczbę bezpośrednich następników przekazanego węzła.

Tabela 15. Metody do ustalania, które węzły znajdują się przed, a które za określonym węzłem (kontynuacja)

Metoda	Typ zwracanego wyniku	Opis
s.successors(node)	Lista	Zwraca bezpośrednie następniki przekazanego węzła.

Kasowanie (lub usuwanie) elementów

We wcześniejszym języku skryptowym dostępne są różne warianty komendy clear, na przykład:

- clear outputs Usuwa wszystkie elementy wynikowe z palety menedżera.
- clear generated palette Usuwa wszystkie modele użytkowe z palety Modele.
- clear stream Usuwa zawartość strumienia.

W środowisku skryptowym Python dostępne są podobne funkcje; komenda removeAll() służy do usuwania zawartości menedżerów strumieni, wyników i modeli, na przykład:

- Aby usunąć zawartość menedżera strumieni:


```
session = modeler.script.session()
session.getStreamManager.removeAll()
```
- Aby usunąć zawartość menedżera wyników:


```
session = modeler.script.session()
session.getDocumentOutputManager().removeAll()
```
- Aby usunąć zawartość menedżera modeli:


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

Uzyskiwanie informacji o węzłach

Węzły dzielą się na różne kategorie, na przykład węzły importu i eksportu danych, węzły budowania modelu i inne typy węzłów. Każdy węzeł udostępnia szereg metod umożliwiających uzyskanie informacji o węzle.

Poniższa tabela zawiera zestawienie węzłów umożliwiających odczytywanie identyfikatorów, nazw i etykiet węzłów.

Tabela 16. Metody umożliwiające odczytywanie identyfikatorów, nazw i etykiet węzłów

Metoda	Typ zwracanego wyniku	Opis
n.getLabel()	string	Zwraca wyświetlaną etykietę określonego węzła. Etykieta jest wartością właściwości custom_name, o ile właściwość ta jest łańcuchem niepustym, a właściwość use_custom_name nie jest ustawiona; w przeciwnym razie etykieta jest wartością funkcji getName().
n.setLabel(label)	Nie dotyczy	Ustawia wyświetlaną etykietę określonego węzła. Jeśli nowa etykieta jest łańcuchem niepustym, to jest wpisywana do właściwości custom_name, a właściwości use_custom_name przypisywana jest wartość False, tak aby określona etykieta miała pierwszeństwo; w przeciwnym razie do właściwości custom_name jest wpisywany pusty łańcuch, a właściwości use_custom_name jest przypisywana wartość True.
n.getName()	string	Zwraca nazwę określonego węzła.

Tabela 16. Metody umożliwiające odczytywanie identyfikatorów, nazw i etykiet węzłów (kontynuacja)

Metoda	Typ zwracanego wyniku	Opis
n.getID()	string	Zwraca identyfikator określonego węzła. Każdorazowo przy tworzeniu nowego węzła tworzony jest nowy identyfikator. Identyfikator jest zachowywany w węźle, gdy ten zapisywany jest jako część strumienia, zatem po otwarciu strumienia węzły mają te same identyfikatory, co dotychczas. Jeśli jednak zapisany węzeł zostanie wstawiony do strumienia, to będzie traktowany jak nowy obiekt i otrzyma nowy identyfikator.

Poniższa tabela zawiera zestawienie metod umożliwiających uzyskiwanie innych informacji o węzłach.

Tabela 17. Metody umożliwiające uzyskiwanie informacji o węzłach

Metoda	Typ zwracanego wyniku	Opis
n.getTypeName()	string	Zwraca nazwę tego węzła używaną w skryptach. Jest to ta sama nazwa, której można byłoby używać do tworzenia nowych instancji tego węzła.
n.isInitial()	Boolean	Zwraca True, jeśli jest to węzeł początkowy, tj. węzeł znajdujący się na początku strumienia.
n.isInline()	Boolean	Zwraca True, jeśli jest to węzeł wstawiony, tj. węzeł znajdujący się na pośrodku strumienia.
n.isTerminal()	Boolean	Zwraca True, jeśli jest to węzeł końcowy, tj. węzeł znajdujący się na końcu strumienia.
n.getXPosition()	int	Zwraca przesunięcie x węzła w strumieniu.
n.getYPosition()	int	Zwraca przesunięcie y węzła w strumieniu.
n.setXYPosition(x, y)	Nie dotyczy	Ustawia położenie węzła w strumieniu.
n.setPositionBetween(source, target)	Nie dotyczy	Ustawia takie położenie węzła w strumieniu, by znalazł się on między dwoma przekazanymi węzłami.
n.isCacheEnabled()	Boolean	Zwraca True, jeśli buforowanie w pamięci podręcznej jest włączone; w przeciwnym wypadku zwraca False.
n.setCacheEnabled(val)	Nie dotyczy	Włącza lub wyłącza buforowanie tego obiektu w pamięci podręcznej. Jeśli pamięć podręczna jest zapełniona i buforowanie zostało wyłączone, to pamięć podręczna jest opróżniana.
n.isCacheFull()	Boolean	Zwraca True, jeśli pamięć podręczna jest zapełniona; w przeciwnym wypadku zwraca False.
n.flushCache()	Nie dotyczy	Opróżnia pamięć podręczną tego węzła. Nie odnosi skutku, jeśli pamięć podręczna nie jest włączona lub jest zapełniona.

Rozdział 4. Skryptowy interfejs API

Wprowadzenie do skryptowego interfejsu API

Skryptowy interfejs API zapewnia dostęp do szerokiej gamy funkcji programu SPSS Modeler. Wszystkie opisane dotąd metody są częścią interfejsu API i można do nich uzyskiwać dostęp z poziomu skryptu bez konieczności importowania dodatkowych modułów. Jednak aby móc odwoływać się do klas interfejsu API, należy jawnie zaimportować interfejs API za pomocą następującej instrukcji:

```
import modeler.api
```

Ta instrukcja importu jest wymagana do działania wielu przykładów wykorzystania skryptowego interfejsu API.

Kompletny przewodnik po klasach, metodach i parametrach dostępnych za pośrednictwem skryptowego interfejsu API można znaleźć w dokumencie *IBM SPSS Modeler Python Scripting API Reference Guide*.

Przykład 1: wyszukiwanie węzłów za pomocą filtra niestandardowego

Sekcja “Znajdowanie węzłów” na stronie 29 zawiera przykład wyszukiwania węzła w strumieniu na podstawie nazwy węzła. W niektórych sytuacjach wymagane jest wyszukiwanie bardziej ogólne, które można zrealizować za pomocą klasy `NodeFilter` i metody `findAll()` obiektu `stream`. Taka metoda wyszukiwania obejmuje dwa kroki:

1. Utworzenie nowej klasy, która rozszerza klasę `NodeFilter`, a potem implementuje własną wersję metody `accept()`.
2. Wywołanie metody `findAll()` strumienia z instancją tej nowej klasy. Metoda zwróci wszystkie węzły spełniające kryteria zdefiniowane w metodzie `accept()`.

Poniższy przykład ilustruje sposób wyszukiwania węzłów w strumieniu, dla którego włączona jest pamięć podręczna węzłów. Zwróconej liście węzłów można użyć do usunięcia tych węzłów z pamięci podręcznej lub wyłączenia ich buforowania.

```
import modeler.api

class CacheFilter(modeler.api.NodeFilter):
    """Filtr węzłów z włączonym buforowaniem"""
    def accept(this, node):
        return node.isCacheEnabled()

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

Przykład 2: umożliwianie użytkownikom uzyskanie informacji o katalogu lub pliku na podstawie swoich uprawnień

Aby uniknąć otwierania PSAPI przez użytkowników, można się posłużyć metodą zwaną `session.getServerFileSystem()`. Wystarczy wywołać funkcję PSAPI służącą do utworzenia obiektu systemu plików.

W poniższym przykładzie przedstawiono, w jaki sposób można zezwolić użytkownikom na uzyskanie informacji o katalogu lub pliku na podstawie uprawnień użytkownika, który łączy się z serwerem IBM SPSS Modeler Server.

```
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()

```

Metadane: informacje o danych

Ponieważ węzły są połączone w strumień, dostępne są informacje o kolumnach lub zmiennych dostępnych w każdym węźle. Na przykład w interfejsie użytkownika programu Modeler użytkownik może wybrać zmienne, według których ma odbywać się sortowanie lub agregacja. Te informacje nazywane są modelem danych.

Skrypty mają dostęp do modelu danych, ponieważ mogą odczytywać zmienne wchodzące do węzła i wychodzące z węzła. W przypadku niektórych węzłów modele danych wejściowych i wyjściowych są takie same: na przykład węzeł Sortowanie po prostu zmienia kolejność rekordów, ale nie zmienia modelu danych. Inne węzły, na przykład Wyliczanie, mogą dodawać nowe zmienne. Jeszcze inne, takie jak węzeł filtrowania, mogą zmieniać nazwy zmiennych lub usuwać zmienne.

Poniższy przykładowy skrypt operuje na standardowym strumieniu programu IBM SPSS Modeler o nazwie `druglearn.str` i dla każdej zmiennej buduje model z pominięciem jednej ze zmiennych wejściowych. Skrypt:

1. Uzyskuje dostęp do modelu danych wynikowych węzła Typ.
2. W pętli przegląda zmienne modelu danych wynikowych.
3. Modyfikuje węzeł filtrowania dla każdej zmiennej wejściowej.
4. Zmienia nazwę generowanego modelu.
5. Uruchamia węzeł budowy modelu.

Uwaga: Przed uruchomieniem skryptu w strumieniu `druglearn.str` należy koniecznie wybrać Python jako język skryptowy (strumień został utworzony w poprzedniej wersji programu IBM SPSS Modeler, zatem jest w nim wybrany wcześniejszy język skryptowy).

```

import modeler.api

stream = modeler.script.stream()
filternode = stream.findByType("filter", None)
typenode = stream.findByType("type", None)
c50node = stream.findByType("c50", None)
# Zawsze używaj niestandardowej nazwy modelu
c50node.setPropertyValue("use_model_name", True)

lastRemoved = None
fields = typenode.getOutputDataModel()
for field in fields:
    # Jeśli jest to zmienna przewidywana, to ją zignoruj
    if field.getModelingRole() == modeler.api.ModelingRole.OUT:
        continue

    # Ponownie włącz zmienną, która ostatnio została usunięta
    if lastRemoved != None:
        filternode.setKeyedPropertyValue("include", lastRemoved, True)

    # Usuń zmienną
    lastRemoved = field.getColumnName()
    filternode.setKeyedPropertyValue("include", lastRemoved, False)

    # Określ nazwę nowego modelu, a następnie uruchom budowanie
    c50node.setPropertyValue("model_name", "Exclude " + lastRemoved)
    c50node.run([])

```

Obiekt `DataModel` udostępnia szereg metod służących do uzyskiwania dostępu do informacji o zmiennych lub kolumnach w modelu danych. Zestawienie tych metod przedstawiono w poniższej tabeli.

Tabela 18. Metody obiektu `DataModel` służące do uzyskiwania dostępu do informacji o zmiennych lub kolumnach

Metoda	Typ zwracanego wyniku	Opis
<code>d.getColumnCount()</code>	<i>int</i>	Zwraca liczbę kolumn w modelu danych
<code>d.columnIterator()</code>	Iterator	Zwraca iterator zwracający kolumny w kolejności, w jakiej były wstawiane. Iterator zwraca instancje klasy <code>Column</code> .
<code>d.nameIterator()</code>	Iterator	Zwraca iterator zwracający nazwy kolumn w kolejności, w jakiej były wstawiane.
<code>d.contains(name)</code>	<i>Boolean</i>	Zwraca <code>True</code> , jeśli kolumna o podanej nazwie istnieje w tym obiekcie <code>DataModel</code> , a w przeciwnym wypadku zwraca <code>False</code> .
<code>d.getColumn(name)</code>	Kolumna	Zwraca kolumnę o określonej nazwie.
<code>d.getColumnGroup(name)</code>	<code>ColumnGroup</code>	Zwraca grupę kolumn o podanej nazwie lub <code>None</code> , jeśli taka grupa kolumn nie istnieje.
<code>d.getColumnGroupCount()</code>	<i>int</i>	Zwraca liczbę grup kolumn w tym modelu danych.
<code>d.columnGroupIterator()</code>	Iterator	Zwraca iterator zwracający kolejne grupy kolumn.
<code>d.toArray()</code>	<code>Column[]</code>	Zwraca model danych jako tablicę kolumn. Kolumny są zwracane w kolejności, w jakiej były wstawiane.

Każda zmienna (obiekt `Column`) zawiera szereg metod umożliwiający uzyskanie dostępu do informacji o kolumnie. W poniższej tabeli przedstawiono niektóre z nich.

Tabela 19. Metody obiektu `Column` służące do uzyskiwania dostępu do informacji o w kolumnie

Metoda	Typ zwracanego wyniku	Opis
<code>c.getColumnName()</code>	<i>string</i>	Zwraca nazwę kolumny.
<code>c.getColumnLabel()</code>	<i>string</i>	Zwraca etykietę kolumny lub łańcuch pusty, jeśli kolumna nie ma etykiety.
<code>c.getMeasureType()</code>	<code>MeasureType</code>	Zwraca typ pomiaru kolumny.
<code>c.getStorageType()</code>	<code>StorageType</code>	Zwraca typ składowania kolumny.
<code>c.isMeasureDiscrete()</code>	<i>Boolean</i>	Zwraca <code>True</code> , jeśli kolumna jest dyskretna. Za dyskretne uznawane są kolumny typu zbiór i typu flaga.
<code>c.isModelOutputColumn()</code>	<i>Boolean</i>	Zwraca <code>True</code> , jeśli kolumna jest wynikiem modelu.
<code>c.isStorageDatetime()</code>	<i>Boolean</i>	Zwraca <code>True</code> , jeśli typem składowania kolumny jest czas, data lub znacznik czasu.
<code>c.isStorageNumeric()</code>	<i>Boolean</i>	Zwraca <code>True</code> , jeśli typem składowania kolumny jest liczba całkowita lub rzeczywista.

Tabela 19. Metody obiektu *Column* służące do uzyskiwania dostępu do informacji o w kolumnie (kontynuacja)

Metoda	Typ zwracanego wyniku	Opis
<code>c.isValidValue(value)</code>	<i>Boolean</i>	Zwraca <code>True</code> , jeśli określona wartość jest poprawna dla tego typu składowania, i <code>valid</code> , gdy znane są poprawne wartości kolumny.
<code>c.getModelingRole()</code>	<code>ModelingRole</code>	Zwraca rolę kolumny w modelowaniu.
<code>c.getSetValues()</code>	<code>Object[]</code>	Zwraca tablicę poprawnych wartości kolumny lub <code>None</code> , jeśli wartości są nieznane lub jeśli kolumna nie jest zbiorem.
<code>c.getValueLabel(value)</code>	<i>string</i>	Zwraca etykietę wartości kolumny lub łańcuch pusty, jeśli wartość nie ma etykiety.
<code>c.getFalseFlag()</code>	Obiekt	Zwraca wartość kolumny oznaczającą „fałsz” albo zwraca <code>None</code> , jeśli wartość jest nieznana lub jeśli kolumna nie jest flagą.
<code>c.getTrueFlag()</code>	Obiekt	Zwraca wartość kolumny oznaczającą „prawda” albo zwraca <code>None</code> , jeśli wartość jest nieznana lub jeśli kolumna nie jest flagą.
<code>c.getLowerBound()</code>	Obiekt	Zwraca dolną wartość graniczną kolumny lub <code>None</code> , jeśli wartość jest nieznana lub kolumna nie jest ciągła.
<code>c.getUpperBound()</code>	Obiekt	Zwraca górną wartość graniczną kolumny lub <code>None</code> , jeśli wartość jest nieznana lub kolumna nie jest ciągła.

Należy zwrócić uwagę, że większość metod służących do dostępu do informacji o kolumnie ma swoje odpowiedniki zdefiniowane w samym obiekcie `DataModel`. Na przykład poniższe dwie instrukcje są równoważne:

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

Dostęp do wygenerowanych obiektów

Podczas wykonywania strumienia zwykle generowane są dodatkowe obiekty wynikowe. Takim obiektem może być nowy model lub element wynikowy zawierający informacje do wykorzystania przy następnych wykonaniach.

W poniższym przykładzie znów zaczynamy od strumienia `druglearn.str`. W przykładzie wszystkie węzły w strumieniu są wykonywane, a wyniki są zapisywane na liście. Następnie skrypt w pętli przechodzi przez wyniki, a modele będące wynikami wykonania są zapisywane jako pliki modelu IBM SPSS Modeler (`.gm`), po czym modele są eksportowane w formacie PMML.

```
import modeler.api

stream = modeler.script.stream()

# Należy wpisać ścieżkę do folderu istniejącego w systemie użytkownika.
# Wymagany jest końcowy separator katalogów
modelFolder = "C:/temp/models/"

# Wykonaj strumień
models = []
stream.runAll(models)
```

```

# Zapisz wszystkie utworzone modele
taskrunner = modeler.script.session().getTaskRunner()
for model in models:
    # Jeśli w wyniku wykonania strumienia zostały wygenerowane inne wyniki, zignoruj je
    if not(isinstance(model, modeler.api.ModelOutput)):
        continue

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

    # zapisz każdy z modeli...
    modelFile = modelFolder + label + algorithm + ".gm"
    taskrunner.saveModelToFile(model, modelFile)

    # ...i wyeksportuj każdy model w formacie PMML...
    modelFile = modelFolder + label + algorithm + ".xml"
    taskrunner.exportModelToFile(model, modelFile, modeler.api.FileFormat.XML)

```

Klasa taskrunner oferuje wygodny mechanizm uruchamiania różnych typowych zadań. Poniższa tabela zawiera podsumowanie metod dostępnych w tej klasie.

Tabela 20. Metody klasy taskrunner do wykonywania typowych zadań

Metoda	Typ zwracanego wyniku	Opis
t.createStream(name, autoConnect, autoManage)	Strumień	Tworzy i zwraca nowy strumień. Kod, który musi tworzyć strumień prywatnie, tak aby były niewidoczne dla użytkownika, powinien ustawiać flagę autoManage na False.
t.exportDocumentToFile (documentOutput, filename, fileFormat)	Nie dotyczy	Eksportuje opis strumienia do pliku w określonym formacie.
t.exportModelToFile(modelOutput, filename, fileFormat)	Nie dotyczy	Eksportuje model do pliku w określonym formacie.
t.exportStreamToFile(stream, filename, fileFormat)	Nie dotyczy	Eksportuje strumień do pliku w określonym formacie.
t.insertNodeFromFile(filename, diagram)	Węzeł	Odczytuje i zwraca węzeł z określonego pliku, wstawiając go do przekazanego diagramu. Tej metody można używać zarówno do odczytywania węzłów Node, jak i SuperNode.
t.openDocumentFromFile(filename, autoManage)	DocumentOutput	Odczytuje i zwraca dokument z określonego pliku.
t.openModelFromFile(filename, autoManage)	ModelOutput	Odczytuje i zwraca model z określonego pliku.
t.openStreamFromFile(filename, autoManage)	Strumień	Odczytuje i zwraca strumień z określonego pliku.
t.saveDocumentToFile (documentOutput, filename)	Nie dotyczy	Zapisuje dokument w pliku w określonej lokalizacji.
t.saveModelToFile(modelOutput, filename)	Nie dotyczy	Zapisuje model w pliku w określonej lokalizacji.
t.saveStreamToFile(stream, filename)	Nie dotyczy	Zapisuje strumień w pliku w określonej lokalizacji.

Obsługa błędów

W języku Python obsługę błędów realizuje się przy użyciu bloku kodu `try...except`. Można go używać w skryptach do przechwytywania wyjątków i obsługi problemów, które — gdyby nie zostały obsłużone — spowodowałyby przerwanie wykonania skryptu.

W poniższym przykładowym skrypcie podejmowana jest próba pobrania modelu z repozytorium IBM SPSS Collaboration and Deployment Services Repository. Ta operacja może spowodować wygenerowanie wyjątku, np. gdy dane uwierzytelniające repozytorium nie zostaną prawidłowo skonfigurowane lub ścieżka repozytorium będzie nieprawidłowa. W skrypcie może to spowodować wygenerowanie wyjątku `ModelerException` (wszystkie wyjątki generowane przez IBM SPSS Modelers są pochodną klasy `modeler.api.ModelerException`).

```
import modeler.api

session = modeler.script.session()
try:
    repo = session.getRepository()
    m = repo.retrieveModel("/some-non-existent-path", None, None, True)
    # wydruki generowane są na karcie Debugowanie w panelu skryptów interfejsu
    programu Modeler
    print "Wszystko OK"
except modeler.api.ModelerException, e:
    print "Wystąpił błąd:", e.getMessage()
```

Uwaga: Niektóre operacje skryptowe mogą powodować generowanie standardowych wyjątków środowiska Java; takie wyjątki nie są pochodną klasy `ModelerException`. Aby wychwytywać takie wyjątki, można zastosować dodatkowy blok `except` przechwytyjący wszystkie wyjątki Java, na przykład:

```
import modeler.api

session = modeler.script.session()
try:
    repo = session.getRepository()
    m = repo.retrieveModel("/some-non-existent-path", None, None, True)
    # wydruki generowane są na karcie Debugowanie w panelu skryptów interfejsu
    programu Modeler
    print "Wszystko OK"
except modeler.api.ModelerException, e:
    print "Wystąpił błąd:", e.getMessage()
except java.lang.Exception, e:
    print "Wystąpił wyjątek Java:", e.getMessage()
```

Parametry strumienia, sesji i superwęzła

Parametry są użytecznym mechanizmem przekazywania wartości w czasie wykonywania i stanowią alternatywę dla wpisywania wartości na stałe do skryptu. Parametry i ich wartości definiuje się tak samo dla strumieni i superwęzłów, tj. jak wpisy w tabeli parametrów strumienia lub superwęzła, bądź jako parametry w wierszu komend. Klasy `Stream` i `SuperNode` implementują zestaw funkcji zdefiniowanych przez obiekt `ParameterProvider`. Funkcje te przedstawiono w poniższej tabeli. Sesja udostępnia wywołanie `getParameters()`, które zwraca obiekt definiujący te funkcje.

Tabela 21. Funkcje zdefiniowane przez obiekt `ParameterProvider`

Metoda	Typ zwracanego wyniku	Opis
<code>p.parameterIterator()</code>	Iterator	Zwraca iterator nazw parametrów dla tego obiektu.

Tabela 21. Funkcje zdefiniowane przez obiekt *ParameterProvider* (kontynuacja)

Metoda	Typ zwracanego wyniku	Opis
<code>p.getParameterDefinition (parameterName)</code>	<code>ParameterDefinition</code>	Zwraca definicję parametru o określonej nazwie lub <code>None</code> , jeśli w tym dostawcy nie istnieje podany parametr. Wynik może być obrazem stanu definicji z chwili wywołania metody i może nie odzwierciedlać późniejszych modyfikacji wprowadzonych w parametrze przez dostawcę.
<code>p.getParameterLabel(parameterName)</code>	<code>string</code>	Zwraca etykietę parametru o określonej nazwie lub <code>None</code> , jeśli taki parametr nie istnieje.
<code>p.setParameterLabel(parameterName, label)</code>	Nie dotyczy	Ustawia etykietę parametru o podanej nazwie.
<code>p.getParameterStorage (parameterName)</code>	<code>ParameterStorage</code>	Zwraca typ składowania parametru o określonej nazwie lub <code>None</code> , jeśli taki parametr nie istnieje.
<code>p.setParameterStorage (parameterName, storage)</code>	Nie dotyczy	Ustawia typ składowania parametru o podanej nazwie.
<code>p.getParameterType(parameterName)</code>	<code>ParameterType</code>	Zwraca typ parametru o określonej nazwie lub <code>None</code> , jeśli taki parametr nie istnieje.
<code>p.setParameterType(parameterName, type)</code>	Nie dotyczy	Ustawia typ parametru o podanej nazwie.
<code>p.getParameterValue(parameterName)</code>	Obiekt	Zwraca wartość parametru o określonej nazwie lub <code>None</code> , jeśli taki parametr nie istnieje.
<code>p.setParameterValue(parameterName, value)</code>	Nie dotyczy	Ustawia wartość parametru o podanej nazwie.

W poniższym przykładzie skrypt agreguje pewne dane telekomunikacyjne, by ustalić, w którym regionie średni przychód jest najniższy. Następnie region ten jest przypisywany parametrowi strumienia. Następnie ten parametr strumienia jest używany w węźle selekcji do wykluczenia odpowiedniego regionu z danych, zanim na podstawie pozostałych danych utworzony zostanie model odchodzenia klientów.

Przykład jest sztucznie zawikłany, ponieważ to sam skrypt generuje węzeł selekcji i mógłby wygenerować poprawną wartość bezpośrednio w wyrażeniu tego węzła. Jednak strumienie są zwykle tworzone wcześniej, zatem ten sposób ustawiania parametrów jest w praktyce użyteczny.

Pierwsza część przykładowego skryptu tworzy parametr strumienia, który będzie zawierał region o najniższym średnim przychodzie. Skrypt tworzy także węzły w gałęzi agregacji i gałęzi budowania modelu i łączy je ze sobą.

```
import modeler.api

stream = modeler.script.stream()

# Zainicjuj parametr strumienia
stream.setParameterStorage("LowestRegion", modeler.api.ParameterStorage.INTEGER)

# Najpierw utwórz gałąź agregacji do obliczania średniego przychodu na 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")
# Odwołanie do parametru strumienia w ramach wyboru
selectnode.setPropertyValue("condition", "'region' = '$P-LowestRegion'")

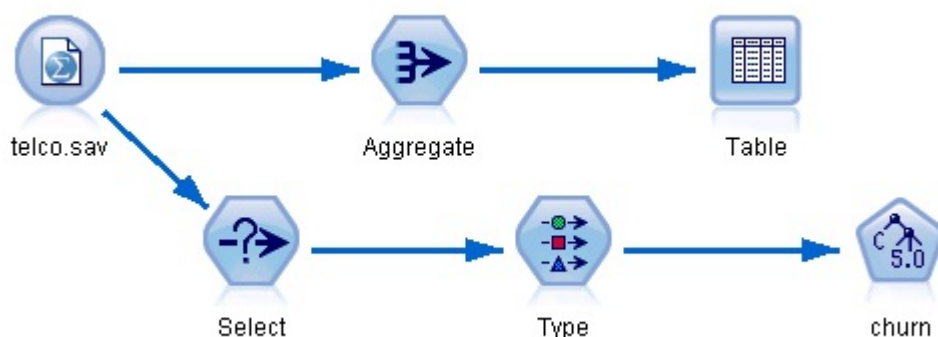
typenode = stream.createAt("type", "Type", 366, 232)
typenode.setKeyedPropertyValue("direction", "churn", "Target")

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

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

```

Przykładowy skrypt tworzy następujący strumień/



Rysunek 5. Strumień utworzony przez przykładowy skrypt

Poniższa część przykładowego skryptu wykonuje węzeł Tabela na końcu gałęzi agregacji.

```

# Najpierw wykonaj węzeł Tabela
results = []
tablenode.run(results)

```

Poniższa część przykładowego skryptu uzyskuje dostęp do wynikowej tabeli wygenerowanej przez węzeł Tabela. Następnie skrypt iteracyjnie przeszukuje wiersze tabeli, wybierając region o najniższym średnim przychodzie.

```

# W wyniku wykonania węzła Tabela powinna zostać wygenerowana jedna tabela
table = results[0]

```

```

# Wynikowa tabela zawiera obiekt RowSet, więc możemy uzyskiwać dostęp do wartości,
odwołując się do wierszy i kolumn
rowset = table.getRowSet()
min_income = 1000000.0
min_region = None

```

```

# Zgodnie z definicją węzła agregacji pierwsza kolumna zawiera
# region, a druga średni przychód
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

```

Poniższa część przykładowego skryptu przypisuje parametrowi strumienia "LowestRegion" znaleziony wcześniej region o najniższym średnim przychodzie. Następnie skrypt uruchamia budowanie modelu, przy czym określony region jest wykluczony z danych uczących.

```

# Sprawdź, czy została przypisana wartość
if min_region != None:
    stream.setParameterValue("LowestRegion", min_region)
else:
    stream.setParameterValue("LowestRegion", -1)

# Na koniec uruchom budowanie modelu z kryterium wyboru
c50node.run([])

```

Kompletny skrypt przykładowy przedstawiono poniżej.

```

import modeler.api

stream = modeler.script.stream()

# Utwórz parametr strumienia
stream.setParameterStorage("LowestRegion", modeler.api.ParameterStorage.INTEGER)

# Najpierw utwórz gałąź agregacji do obliczania średniego przychodu na 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")
# Odwołanie do parametru strumienia w ramach wyboru
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)

# Najpierw wykonaj węzeł Tabela
results = []
tablenode.run(results)

# W wyniku wykonania węzła Tabela powinna zostać wygenerowana jedna tabela
table = results[0]

# Wynikowa tabela zawiera obiekt RowSet, więc możemy uzyskiwać dostęp do wartości,
odwołując się do wierszy i kolumn
rowset = table.getRowSet()
min_income = 1000000.0

```

```

min_region = None

# Zgodnie z definicją węzła agregacji pierwsza kolumna zawiera
# region, a druga średni przychód
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

# Sprawdź, czy została przypisana wartość
if min_region != None:
    stream.setParameterValue("LowestRegion", min_region)
else:
    stream.setParameterValue("LowestRegion", -1)

# Na koniec uruchom budowanie modelu z kryterium wyboru
c50node.run([])

```

Wartości globalne

Wartości globalne służą do obliczania różnych statystyk podsumowujących dla określonych zmiennych. Dostęp do tych wartości podsumowujących można uzyskać z dowolnego punktu strumienia. Dostęp do wartości globalnych w strumieniu uzyskuje się, podobnie jak do parametrów strumienia, za pośrednictwem nazwy. Jednak, w odróżnieniu od parametrów strumienia, ich wartości są aktualizowane automatycznie podczas wykonywania węzła Globalne, a nie są przypisywane przez skrypt lub z wiersza komend. Dostęp do wartości globalnych strumienia uzyskuje się za pomocą metody `getGlobalValues()`.

Obiekt `GlobalValues` definiuje funkcje widoczne w poniższej tabeli.

Tabela 22. Funkcje zdefiniowane przez obiekt `GlobalValues`

Metoda	Typ zwracanego wyniku	Opis
<code>g.fieldNameIterator()</code>	Iterator	Zwraca iterator z nazwami zmiennych zawierających co najmniej jedną wartość globalną.
<code>g.getValue(type, fieldName)</code>	Obiekt	Zwraca wartość globalną na podstawie określonego typu i nazwy zmiennej albo <code>None</code> , jeśli nie uda się znaleźć wartości. Zasadniczo oczekuje się, że zwrócona wartość będzie liczbą, jednak w przyszłości może pojawić się możliwość zwracania wartości innego typu.
<code>g.getValues(fieldName)</code>	Mapa	Zwraca mapę zawierającą znane wpisy dla określonej nazwy zmiennej albo <code>None</code> , jeśli brak jest wpisów dla zmiennej.

`GlobalValues.Type` definiuje typ dostępnych statystyk podsumowujących. Dostępne są następujące statystyki podsumowujące:

- **MAX**: maksymalna wartość zmiennej.
- **MEAN**: średnia wartość zmiennej.
- **MIN**: minimalna wartość zmiennej.
- **STDDEV**: odchylenie standardowe zmiennej.
- **SUM**: suma wartości zmiennej.

Poniższy skrypt przykładowy odczytuje wartość średnią zmiennej "income", która jest obliczana przez węzeł Globalne.

```
import modeler.api

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

Praca z wieloma strumieniami: skrypty samodzielne

Do pracy z wieloma strumieniami należy użyć skryptu samodzielnego. Skrypt samodzielny można edytować i uruchamiać w interfejsie użytkownika programu IBM SPSS Modeler lub przekazać jako parametr wywołania programu w trybie wsadowym, w wierszu komend.

Następujący skrypt samodzielny otwiera dwa strumienie. Jeden z tych strumieni tworzy model, a drugi strumień wykreśla rozkład wartości przewidywanych.

```
# Należy zmienić na lokalizację właściwą dla konkretnego systemu
demosDir = "C:/Program Files/IBM/SPSS/Modeler/18.1/DEMOS/streams/"

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

# Otwórz strumień tworzący model, odszukaj węzeł C5.0 i uruchom go
buildstream = tasks.openStreamFromFile(demosDir + "druglearn.str", True)
c50node = buildstream.findByType("c50", None)
results = []
c50node.run(results)

# Teraz otwórz strumień wykresu, znajdź wyliczenie Na_to_K i histogram
plotstream = tasks.openStreamFromFile(demosDir + "drugplot.str", True)
derivenode = plotstream.findByType("derive", None)
histogramnode = plotstream.findByType("histogram", None)

# Utwórz węzeł modelu do zastosowania, wstaw go między węzeł wyliczany a węzeł histogramu,
# następnie uruchom 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([])

# Na koniec wyczyść strumienie
buildstream.close()
plotstream.close()
```

W poniższym przykładzie przedstawiono, w jaki sposób można również przeprowadzić iterowania otwartych strumieni (wszystkie strumienie są otwierane na karcie Strumienie). Należy zwrócić uwagę, że ta opcja jest obsługiwana tylko w skryptach autonomicznych.

```
dla strumienia w modeler.script.streams ():
    print stream.getName()
```

Rozdział 5. Wskazówki dotyczące skryptów

Niniejsza sekcja zawiera ogólne wskazówki i omówienie technik związanych z korzystaniem ze skryptów, dotyczące m.in. wpływania na wykonywanie strumienia, użycia zakodowanego hasła w skrypcie i uzyskiwania dostępu do obiektów w repozytorium IBM SPSS Collaboration and Deployment Services Repository.

Wpływanie na wykonywanie strumienia

Węzły końcowe strumienia są wykonywane w kolejności zoptymalizowanej dla sytuacji domyślnej. W pewnych okolicznościach preferowana może być inna kolejność wykonywania. Aby zmienić kolejność wykonywania strumienia, należy wykonać następujące kroki na karcie Wykonywanie w oknie dialogowym właściwości strumienia:

1. Rozpocznij od pustego skryptu.
2. Kliknij przycisk **Dołącz domyślny skrypt** na pasku narzędzi, aby dodać domyślny skrypt strumienia.
3. Zmień kolejność instrukcji w domyślnym skrypcie strumienia na taki, w jakim mają być wykonywane.

Przechodzenie przez węzły w pętli

Pętla for umożliwia przejście przez wszystkie węzły w strumieniu. Na przykład poniższe dwa skrypty przechodzą przez wszystkie węzły i zmieniają w węzłach filtrowania nazwy zmiennych na zapisane wielkimi literami.

Tych skryptów można używać w każdym strumieniu zawierającym węzeł filtrowania, nawet jeśli żadne zmienne nie są faktycznie filtrowane. Wystarczy dodać węzeł filtrowania, który będzie przechodził przez wszystkie zmienne, aby zmienić wszystkie nazwy zmiennych na zapisane wielkimi literami.

```
# Sposób 1: użycie funkcji nameIterator() modelu danych
stream = modeler.script.stream()
for node in stream.iterator():
    if (node.getTypeName() == "filter"):
        # nameIterator() zwraca nazwy zmiennych
        for field in node.getInputDataModel().nameIterator():
            newname = field.upper()
            node.setKeyedPropertyValue("new_name", field, newname)

# Sposób 2: użycie funkcji iterator() modelu danych
stream = modeler.script.stream()
for node in stream.iterator():
    if (node.getTypeName() == "filter"):
        # iterator() zwraca zmienne jako obiekty, zatem
        # musimy wywołać getColumnName(), by odczytać nazwę
        for field in node.getInputDataModel().iterator():
            newname = field.getColumnName().upper()
            node.setKeyedPropertyValue("new_name", field.getColumnName(), newname)
```

Skrypt przechodzi w pętli przez wszystkie węzły w bieżącym strumieniu i przy każdym węźle sprawdza, czy jest to węzeł filtrowania. Jeśli tak, skrypt w pętli przechodzi przez wszystkie zmienne węzła i za pomocą funkcji `field.upper()` lub `field.getColumnName().upper()` zmienia litery w nazwie na wielkie.

Uzyskiwanie dostępu do obiektów w repozytorium IBM SPSS Collaboration and Deployment Services Repository

Jeśli masz licencję na IBM SPSS Collaboration and Deployment Services Repository, możesz zapisywać i pobierać obiekty w repozytorium za pomocą komend skryptowych. Repozytorium służy do zarządzania cyklem życia modeli eksploracji danych i powiązanych z nimi obiektów predykcyjnych w kontekście aplikacji, narzędzi i rozwiązań korporacyjnych.

Nawiązywanie połączenia z repozytorium IBM SPSS Collaboration and Deployment Services Repository

Aby uzyskać dostęp do repozytorium, należy najpierw skonfigurować poprawne połączenie z repozytorium. Czynność tę wykonuje się za pośrednictwem menu **Narzędzia** w interfejsu użytkownika programu SPSS Modeler, albo z wiersza komend. Aby uzyskać więcej informacji, patrz “Argumenty połączenia z repozytorium IBM SPSS Collaboration and Deployment Services Repository” na stronie 65.

Uzyskiwanie dostępu do repozytorium

Dostęp do repozytorium można uzyskać za pośrednictwem obiektu sesji, na przykład:

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

Pobieranie obiektów z repozytorium

W skrypcie można używać funkcji `retrieve*`, by uzyskiwać dostęp do różnych obiektów, w tym strumieni, modeli, wyników i węzłów. W poniższej tabeli przedstawiono zestawienie funkcji służących do pobierania obiektów.

Tabela 23. Funkcje skryptowe do pobierania obiektów

Typ obiektu	Funkcja repozytorium
Strumień	<code>repo.retrieveStream(String path, String version, String label, Boolean autoManage)</code>
Model	<code>repo.retrieveModel(String path, String version, String label, Boolean autoManage)</code>
Output	<code>repo.retrieveDocument(String path, String version, String label, Boolean autoManage)</code>
Węzeł	<code>repo.retrieveProcessor(String path, String version, String label, ProcessorDiagram diagram)</code>

Można na przykład pobrać strumień z repozytorium, korzystając z następującej funkcji:

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

Ten przykład pobiera strumień `risk_score.str` z określonego folderu. Etykieta `production` wskazuje wersję strumienia, która ma być pobrana, a ostatni parametr określa, że SPSS Modeler ma zarządzać strumieniem (powoduje to na przykład, że strumień będzie widoczny na karcie **Strumienie**, o ile widoczny jest interfejs użytkownika programu SPSS Modeler). Alternatywą jest użycie konkretnej wersji bez etykiety:

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

Uwaga: Jeśli i parametr `version`, i `label` będą równe `None`, to zwrócona zostanie ostatnia wersja.

Zapisywanie obiektów w repozytorium

Do zapisywania obiektów w repozytorium można w skryptach używać funkcji `store*`. W poniższej tabeli przedstawiono zestawienie funkcji służących do zapisywania obiektów.

Tabela 24. Funkcje skryptowe do zapisywania obiektów

Typ obiektu	Funkcja repozytorium
Strumień	<code>repo.storeStream(ProcessorStream stream, String path, String label)</code>
Model	<code>repo.storeModel(ModelOutput modelOutput, String path, String label)</code>
Output	<code>repo.storeDocument(DocumentOutput documentOutput, String path, String label)</code>
Węzeł	<code>repo.storeProcessor(Processor node, String path, String label)</code>

Na przykład za pomocą poniższego wywołania można zapisać nową wersję strumienia `risk_score.str`:

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


Ten przykład zapisuje nową wersję strumienia, kojarzy z nią etykietę "test" i zwraca znacznik nowo utworzonej wersji.

Uwaga: Jeśli nie chcesz kojarzyć etykiety z nową wersją, przełącz None jako etykietę.

Zarządzanie folderami w repozytorium

Korzystając z folderów w repozytorium, można organizować obiekty w logiczne grupy i lepiej uwidocznić relacje między obiektami. Do tworzenia folderów służy funkcja `createFolder()`, co ilustruje poniższy przykład:

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

Ten przykład tworzy nowy folder o nazwie "cross-sell" w folderze "/projects". Funkcja zwraca pełną ścieżkę do nowego folderu.

Do zmieniania nazwy folderu służy funkcja `renameFolder()`:

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

Pierwszy parametr jest pełną ścieżką do folderu, którego nazwa ma być zmieniona, a drugi jest nową nazwą folderu.

Do usuwania pustego folderu służy funkcja `deleteFolder()`:

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

Blokowanie i odblokowywanie obiektów

Za pomocą skryptu można zablokować obiekt, aby uniemożliwić innym użytkownikom aktualizowanie jego istniejących wersji lub tworzenie nowych. Istnieje również możliwość odblokowania obiektu, który został zablokowany.

Składnia do blokowania i odblokowywania obiektu:

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

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

Podobnie jak w przypadku zapisywania i pobierania obiektów, `REPOSITORY_PATH` wskazuje lokalizację obiektu w repozytorium. Ścieżka musi być ujęta w cudzysłów i zawierać ukośniki jako separatory. Nie jest rozróżniana wielkość liter.

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

Aby wskazać lokalizację obiektu, można także użyć identyfikatora URI (Uniform Resource Identifier) zamiast ścieżki repozytorium. Identyfikator URI musi zawierać przedrostek `spsscr:` i być całkowicie ujęty w cudzysłów. Dozwolone są tylko ukośniki jako separatory ścieżki; spacje muszą być zakodowane. Oznacza to, że należy użyć komendy `%20` zamiast spacji w ścieżce. Identyfikator URI nie rozróżnia wielkości liter. Kilka przykładów:

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

Należy pamiętać, że blokowanie obiektów ma zastosowanie do wszystkich wersji obiektu — nie można blokować lub odblokowywać poszczególnych wersji.

Generowanie hasła kodowanego

W niektórych przypadkach (np. przy dostępie do źródła danych chronionego hasłem) konieczne jest umieszczenie hasła w treści skryptu. Hasła kodowanych można używać:

- we właściwościach węzłów źródłowych i wynikowych baz danych;
- w argumentach wiersza komend podczas logowania się do serwera;
- we właściwościach połączenia z bazą danych zapisanych w pliku *.par* (plik parametrów generowany z karty publikowania węzła eksportu).

W interfejsie użytkownika dostępne jest narzędzie do generowania kodowanych haseł zgodnie z algorytmem Blowfish (więcej informacji na stronie <http://www.schneier.com/blowfish.html>). Po zakodowaniu hasło można kopiować do plików ze skryptami i argumentów wiersza komend. Hasło kodowane jest zapisane we właściwości *epassword* węzłów *databasenode* i *databaseexportnode*.

1. Aby wygenerować hasło kodowane, z menu Narzędzia wybierz:

Kodowanie hasła...

2. Wpisz hasło w polu tekstowym Hasło.
3. Kliknij przycisk **Koduj**, aby wygenerować losowy kod na podstawie wprowadzonego hasła.
4. Kliknij przycisk Kopiuj, aby skopiować zakodowane hasło do schowka.
5. Wklej hasło do skryptu lub jako parametr.

Sprawdzanie skryptów

Można szybko sprawdzić składnię każdego rodzaju skryptu, klikając czerwony przycisk kontroli na pasku narzędzi okna dialogowego Skrypt samodzielny.



Rysunek 6. Ikony na pasku narzędzi skryptu strumienia

W wyniku sprawdzenia skryptu zgłaszane są wszelkie błędy w kodzie oraz przedstawiane rekomendacje poprawek. Aby wyświetlić wiersz z błędami, kliknij informację o błędzie w dolnej części okna dialogowego. Spowoduje to wyróżnienie błędu na czerwono.

Wywoływanie skryptów z wiersza komend

Skrypty umożliwiają realizowanie operacji, które zwykle wykonuje się w interfejsie użytkownika. Wystarczy wskazać i uruchomić samodzielny skrypt w wywołaniu programu IBM SPSS Modeler z wiersza komend. Na przykład:

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

Flaga *-script* powoduje załadowanie określonego skryptu, a flaga *-execute* powoduje wykonanie wszystkich komend z pliku skryptu.

Zgodność z wcześniejszymi wersjami

Skrypty utworzone we wcześniejszych wersjach programu IBM SPSS Modeler powinny zasadniczo działać bez zmian w bieżącej wersji. Jednak obecnie modele użytkowe mogą być wstawiane do strumienia automatycznie (jest to ustawienie domyślne) i mogą albo zastępować, albo uzupełniać model użytkowy tego typu istniejący już w strumieniu. To, czy faktycznie się tak stanie, zależy od ustawienia opcji **Dodaj model do strumienia i Zastąp poprzedni model (Narzędzia > Opcje > Opcje użytkownika > Powiadomienia)**. Może, na przykład, zająć konieczność zmodyfikowania skryptu z wcześniejszej wersji, w którym zastępowanie modelu użytkowego odbywa się poprzez usunięcie istniejącego modelu i wstawienie nowego.

Skrypty utworzone w bieżącej wersji mogą nie działać we wcześniejszych wersjach.

Jeśli skrypt utworzony we wcześniejszej wersji używa komendy, która od tamtego czasu została zastąpiona (lub jest już nieaktualna), to stara postać będzie nadal obsługiwana, ale pojawi się Komunikat o błędzie. Na przykład stare słowo kluczowe `generated` zastąpiono słowem `model`, a komendę `clear generated` zastąpiono komendą `clear generated palette`. Skrypty korzystające ze starszej postaci nadal będą działać, ale wyświetlane będzie ostrzeżenie.

Uzyskiwanie dostępu do wyników wykonania strumienia

Wiele węzłów programu IBM SPSS Modeler generuje obiekty wynikowe, takie jak modele, wykresy i dane tabelaryczne. Wiele z tych obiektów wynikowych zawiera użyteczne wartości, które można wykorzystać w skryptach do sterowania dalszym przebiegiem wykonania kodu. Wartości pogrupowane są w kontenery zawartości (nazywane po prostu kontenerami), do których można uzyskiwać dostęp za pośrednictwem znaczników lub identyfikatorów kontenerów. Sposób dostępu do wartości zależy od formatu lub „modelu zawartości” danego kontenera.

Na przykład wiele modeli predykcyjnych generuje wyniki w formacie PMML (wariancie XML), przedstawiając w ten sposób informacje o modelu, np. których zmiennych używa drzewo decyzyjne w poszczególnych podziałach, lub jak połączone są neurony w sieci neuronowej i jakie są siły tych połączeń. Dostęp do informacji zapisanych w formacie PMML można uzyskiwać za pośrednictwem modelu zawartości XML. Na przykład:

```
stream = modeler.script.stream()
# Zakładamy, że strumień zawiera jeden węzeł tworzący model C5,
# i że już skonfigurowano źródło danych, predyktory i zmienne
# przewidywane
modelbuilder = stream.findByType("c50", None)
results = []
modelbuilder.run(results)
modeloutput = results[0]

# Mamy teraz obiekt wynikowy modelu C5.0. Uzyskujemy
# dostęp do odpowiedniego modelu zawartości
cm = modeloutput.getContentModel("PMML")

# Model zawartości PMML to ogólny model oparty na XML, w którym
# stosowana jest składnia XPath. Użyjemy go do odczytania nazw zmiennych z danymi.
# Wywołanie zwraca listę łańcuchów zgodnych z wartościami XPath
dataFieldNames = cm.getStringValues("/PMML/DataDictionary/DataField", "name")
```

IBM SPSS Modeler obsługuje w skryptach następujące modele zawartości:

- **Model zawartości tabeli** zapewnia dostęp do prostych danych tabelarycznych w formie wierszy i kolumn
- **Model zawartości XML** zapewnia dostęp do zawartości zapisanej w formacie XML
- **Model zawartości JSON** zapewnia dostęp do zawartości zapisanej w formacie JSON
- **Model zawartości statystyk kolumn** zapewnia dostęp do statystyk podsumowujących określoną zmienną
- **Model zawartości statystyk kolumn wyznaczonych parami** zapewnia dostęp do statystyk podsumowujących wyznaczonych między dwiema zmiennymi lub wartości między dwiema odrębnymi zmiennymi.

Model zawartości tabeli

Model zawartości tabeli jest prostym modelem dostępu do danych w formie wierszy i kolumn. Wszystkie wartości w jednej kolumnie muszą mieć ten sam typ składowania (np. być łańcuchami lub liczbami całkowitymi).

API

Tabela 25. API

Wynik	Metoda	Opis
int	getRowCount()	Zwraca liczbę wierszy w tej tabeli.

Tabela 25. API (kontynuacja)

Wynik	Metoda	Opis
int	getColumnCount()	Zwraca liczbę kolumn w tej tabeli.
String	getColumnName(int columnIndex)	Zwraca nazwę kolumny o określonym indeksie. Indeksy kolumn liczone są od 0.
StorageType	getStorageType(int columnIndex)	Zwraca typ składowania kolumny o określonym indeksie. Indeksy kolumn liczone są od 0.
Object	getValueAt(int rowIndex, int columnIndex)	Zwraca wartość z przecięcia wiersza i kolumny o podanych indeksach. Indeksy wierszy i kolumn są liczone od 0.
void	reset()	Opróżnia wewnętrzną pamięć związaną z tym modelem zawartości.

Węzły i wyniki

Ta tabela zawiera listę węzłów generujących wyniki, które zawierają ten typ zawartości.

Tabela 26. Węzły i wyniki

Nazwa węzła	Nazwa wyniku	Identyfikator kontenera
table	table	"table"

Przykładowy skrypt

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

# Skonfiguruj węzeł importu pliku zmiennych
varfilenode = stream.createAt("variablefile", "DRUG Data", 96, 96)
varfilenode.setPropertyValue("full_filename", "$CLEO_DEMOS/DRUG1n")

# Następnie utwórz węzeł agregacji i połącz go z węzłem plików zmiennych
aggregatenode = stream.createAt("aggregate", "Aggregate", 192, 96)
stream.link(varfilenode, aggregatenode)

# Skonfiguruj węzeł agregacji
aggregatenode.setPropertyValue("keys", ["Drug"])
aggregatenode.setKeyedPropertyValue("aggregates", "Age", ["Min", "Max"])
aggregatenode.setKeyedPropertyValue("aggregates", "Na", ["Mean", "SDev"])

# Następnie utwórz węzeł wynikowy Tabela i połącz go z węzłem agregacji
tablenode = stream.createAt("table", "Table", 288, 96)
stream.link(aggregatenode, tablenode)

# Wykona węzeł Tabela i przechwyc wynikową tabelę
results = []
tablenode.run(results)
tableoutput = results[0]

# Uzyskaj dostęp do modelu zawartości wynikowej tabeli
tablecontent = tableoutput.getContentModel("table")

# Dla każdej kolumny wydrukuj jej nazwę, typ i pierwszy wiersz
# wartości
col = 0
```

```

while col < tablecontent.getColumnCount():
    print tablecontent.getColumnName(col), \
          tablecontent.getStorageType(col), \
          tablecontent.getValueAt(0, col)
    col = col + 1

```

Wynik na karcie Debugowanie będzie wyglądał podobnie do przedstawionego poniżej:

```

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

```

Model zawartości XML

Model zawartości XML zapewnia dostęp do zawartości w formacie XML.

Model zawartości XML umożliwia uzyskiwanie dostępu do komponentów na podstawie wyrażeń XPath. Wyrażenia XPath są łańcuchami, które określają elementy lub atrybuty żądane przez stronę wywołującą. Model zawartości XML ukrywa szczegóły konstrukcji różnych obiektów i kompilowania wyrażeń, które zwykle są wymagane do realizacji obsługi formatu XPath. Dzięki temu wywołania ze skryptów Python mogą być prostsze.

Model zawartości XML zawiera funkcję zwracającą dokument XML jako łańcuch. Dzięki temu użytkownicy języka skryptowego Python mogą analizować dane XML przy użyciu wybranej przez siebie biblioteki Python.

API

Tabela 27. API

Wynik	Metoda	Opis
String	getXMLAsString()	Zwraca dane XML jako łańcuch.
number	getNumericValue(String xpath)	Zwraca liczbowy wynik analizy ścieżki (np. liczbę elementów pasujących do wyrażenia ścieżki).
boolean	getBooleanValue(String xpath)	Zwraca boolowski wynik analizy określonego wyrażenia ścieżki.
String	getStringValue(String xpath, String attribute)	Zwraca wartość atrybutu lub wartość węzła XML pasującego do przekazanej ścieżki.
Lista obiektów string	getStringValues(String xpath, String attribute)	Zwraca listę wszystkich wartości atrybutów lub wartości węzłów XML pasujących do przekazanej ścieżki.
Lista list obiektów string	getValuesList(String xpath, <Lista obiektów string> attributes, boolean includeValue)	Zwraca listę wszystkich wartości atrybutów pasujących do przekazanej ścieżki oraz, w razie potrzeby, wartości węzłów XML.
Tabela mieszająca (key:string, value:list of string)	getValuesMap(String xpath, String keyAttribute, <Lista obiektów string> attributes, boolean includeValue)	Zwraca tabelę mieszającą, której kluczem jest atrybut klucza lub wartość XML, zawierającą listę wartości określonych atrybutów.
boolean	isNamespaceAware()	Zwraca informację o tym, czy analizatory XML powinny uwzględniać przestrzenie nazw. Domyślną wartością jest False.

Tabela 27. API (kontynuacja)

Wynik	Metoda	Opis
void	setNamespaceAware(boolean value)	Określa, czy analizatory XML powinny uwzględniać przestrzenie nazw. Powoduje także wywołanie funkcji <code>reset()</code> , aby zmiany zostały uwzględnione w kolejnych wywołaniach.
void	reset()	Opróżnia wewnętrzną pamięć związaną z tym modelem zawartości (na przykład zbuforowany obiekt DOM).

Węzły i wyniki

Ta tabela zawiera listę węzłów generujących wyniki, które zawierają ten typ zawartości.

Tabela 28. Węzły i wyniki

Nazwa węzła	Nazwa wyniku	Identyfikator kontenera
Większość węzłów budujących modele	Większość wygenerowanych modeli	"PMML"
"autodataprep"	n/a	"PMML"

Przykładowy skrypt

Kod w języku Python uzyskujący dostęp do treści może mieć postać:

```
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")
```

Model zawartości JSON

Model zawartości JSON służy do obsługi danych w formacie JSON. Udostępnia prosty interfejs API umożliwiający skryptom wywołującym wyodrębnianie wartości przy założeniu, że wiedzą, do których wartości chcą uzyskać dostęp.

API

Tabela 29. API

Wynik	Metoda	Opis
String	getJSONAsString()	Zwraca zawartość JSON jako łańcuch.
Object	getObjectAt(<Lista obiektów> path, JSONArtifact artifact) throws Exception	Zwraca obiekt wskazany przez określoną ścieżkę. Podany artefakt początku ścieżki może być równy null; wówczas użyty będzie poziom początkowy zawartości. Zwrócona wartość może być literałem łańcuchowym, liczbą całkowitą, liczbą rzeczywistą, wartością boolowską lub artefaktem JSON (obiektom JSON albo tablicą JSON).

Tabela 29. API (kontynuacja)

Wynik	Metoda	Opis
Tabela mieszająca (key:object, value:object>	getChildValuesAt(<Lista obiektów> path, JSONArtifact artifact) throws Exception	Zwraca wartości podrzędne określonej ścieżki, jeśli ścieżka prowadzi do obiektu JSON, albo null w przeciwnym wypadku. Klucze w tabeli są łańcuchami, a powiązana wartość może być literałem łańcuchowym, liczbą całkowitą, liczbą rzeczywistą, wartością boolowską lub artefaktem JSON (obiektem JSON albo tablicą JSON).
Lista obiektów	getChildrenAt(<Lista obiektów> path path, JSONArtifact artifact) throws Exception	Zwraca listę obiektów z określonej ścieżki, jeśli ścieżka prowadzi do tablicy JSON, albo null w przeciwnym wypadku. Zwrócone wartości mogą być literałami łańcuchowymi, liczbami całkowitymi, liczbami rzeczywistymi, wartościami boolowskimi lub artefaktami JSON (obiektami JSON albo tablicami JSON).
void	reset()	Opróżnia wewnętrzną pamięć związaną z tym modelem zawartości (na przykład zbuforowany obiekt DOM).

Przykładowy skrypt

Jeśli istnieje węzeł budujący wyniki, który tworzy wyniki w formacie JSON, to można użyć następującego kodu, aby uzyskać dostęp do informacji o zestawie obiektów book:

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

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

# Inny sposób: pobierz obiekt book i użyj go jako punktu
# początkowego dla następnych wpisów
book = cm.getObjectAt(["books", "ISIN123456"], None)
bookTitle = cm.getObjectAt(["title"], book)

# Pobierz wszystkie wartości podrzędne dla konkretnego obiektu book
bookInfo = cm.getChildValuesAt(["books", "ISIN123456"], None)

# Pobierz trzeci wpis. Zakłada najwyższą wartość "books"
# zawiera tablicę JSON, którą można indeksować
bookInfo = cm.getObjectAt(["books", 2], None)

# Pobierz listę wszystkich wpisów podrzędnych
allBooks = cm.getChildrenAt(["books"], None)
```

Model zawartości statystyki kolumn i model zawartości statystyki kolumn wyznaczonych parami

Model zawartości statystyki kolumn zapewnia dostęp do statystyk, które mogą być obliczane dla każdej zmiennej (statystyk jednej zmiennej). Model zawartości statystyki kolumn wyznaczonych parami zapewnia dostęp do statystyk, które mogą być obliczane między parami zmiennych lub wartości zmiennej.

Możliwe miary statystyk to:

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

Niektóre wartości są odpowiednie tylko w przypadku statystyk jednokolumnowych, a inne tylko w przypadku statystyk wyznaczonych parami.

Węzły, które generują takie wyniki, to:

- **Węzeł Statistics:** generuje statystyki kolumnowe i może generować statystyki parami, gdy określone są zmienne korelacji
- **Węzeł Audyt danych** generuje statystyki kolumnowe i może generować statystyki parami, gdy określona jest zmienna nakładkowa.
- **Węzeł Średnie** generuje statystyki parami przy porównywaniu par zmiennych lub porównywaniu wartości zmiennej z innymi podsumowaniami zmiennych.

To, które modele zawartości i statystyki są dostępne, zależy od możliwości konkretnego węzła i ustawień wewnątrz węzła.

Interfejs API modelu zawartości statystyki kolumn

Tabela 30. Interfejs API modelu zawartości statystyki kolumn.

Wynik	Metoda	Opis
List<StatisticType>	getAvailableStatistics()	Zwraca wszystkie statystyki dostępne w tym modelu. Nie zawsze wszystkie zmienne mają wartości wszystkich statystyk.
List<String>	getAvailableColumns()	Zwraca nazwy kolumn, dla których obliczono statystyki.
Number	getStatistic(String column, StatisticType statistic)	Zwraca wartości statystyk powiązane z kolumną.

Tabela 30. Interfejs API modelu zawartości statystyki kolumn (kontynuacja).

Wynik	Metoda	Opis
void	reset()	Opróżnia wewnętrzną pamięć związaną z tym modelem zawartości.

Interfejs API modelu zawartości statystyk kolumn wyznaczonych parami

Tabela 31. Interfejs API modelu zawartości statystyk kolumn wyznaczonych parami.

Wynik	Metoda	Opis
List<StatisticType>	getAvailableStatistics()	Zwraca wszystkie statystyki dostępne w tym modelu. Nie zawsze wszystkie zmienne mają wartości wszystkich statystyk.
List<String>	getAvailablePrimaryColumns()	Zwraca nazwy głównych kolumn, dla których obliczono statystyki.
List<Object>	getAvailablePrimaryValues()	Zwraca wartości głównej kolumny, dla której obliczono statystyki.
List<String>	getAvailableSecondaryColumns()	Zwraca nazwy dodatkowych kolumn, dla których obliczono statystyki.
Number	getStatistic(String primaryColumn, String secondaryColumn, StatisticType statistic)	Zwraca wartości statystyk powiązane z kolumnami.
Number	getStatistic(String primaryColumn, Object primaryValue, String secondaryColumn, StatisticType statistic)	Zwraca wartości statystyk powiązane z wartością głównej kolumny i kolumną dodatkową.
void	reset()	Opróżnia wewnętrzną pamięć związaną z tym modelem zawartości.

Węzły i wyniki

Ta tabela zawiera listę węzłów generujących wyniki, które zawierają ten typ zawartości.

Tabela 32. Węzły i wyniki.

Nazwa węzła	Nazwa wyniku	Identyfikator kontenera	Uwagi
"means" (węzeł Średnie)	"means"	"columnStatistics"	
"means" (węzeł Średnie)	"means"	"pairwiseStatistics"	
"dataaudit" (węzeł Audyt danych)	"means"	"columnStatistics"	
"statistics" (węzeł Statistics)	"statistics"	"columnStatistics"	Generowany tylko wtedy, gdy badane są określone zmienne.
"statistics" (węzeł Statistics)	"statistics"	"pairwiseStatistics"	Generowany tylko wtedy, gdy zmienne są skorelowane.

Przykładowy skrypt

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

```

# Skonfiguruj dane wejściowe
varfile = stream.createAt("variablefile", "File", 96, 96)
varfile.setPropertyValue("full_filename", "$CLEO/DEMOS/DRUG1n")

# Teraz utwórz węzeł Statistics. Może on generować statystyki
# kolumnowe i statystyki parami
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

```

Rozdział 6. Argumenty w wierszu komend

Wywoływanie oprogramowania

Można użyć wiersza komend systemu operacyjnego w celu uruchomienia programu IBM SPSS Modeler w następujący sposób:

1. Na komputerze, na którym zainstalowano program IBM SPSS Modeler, otwórz okno DOS lub okno wiersza komend.
2. Aby uruchomić interfejs IBM SPSS Modeler w trybie interaktywnym, wpisz komendę `modelerclient` wraz z wymaganymi argumentami, na przykład:

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

Dostępne argumenty (flagi) umożliwiają nawiązanie połączenia z serwerem, załadowanie strumieni, uruchomienie skryptów lub określenie innych parametrów, stosownie do potrzeb.

Używanie argumentów wiersza komend

Do komendy `modelerclient` można dopisywać argumenty (nazywane także flagami), aby wpływać na wywołanie programu IBM SPSS Modeler.

Wyróżnia się kilka typów argumentów wiersza komend. Zostaną one omówione w dalszej części tej sekcji.

Tabela 33. Typy argumentów wiersza komend.

Typ argumentu	Gdzie jest opisany
Argumenty systemowe	Więcej informacji można znaleźć w temacie “Argumenty systemowe” na stronie 62.
Argumenty określające parametry	Więcej informacji można znaleźć w temacie “Argumenty określające parametry” na stronie 63.
Argumenty połączenia z serwerem	Więcej informacji można znaleźć w temacie “Argumenty połączenia z serwerem” na stronie 64.
Argumenty połączenia z repozytorium IBM SPSS Collaboration and Deployment Services Repository	Więcej informacji można znaleźć w temacie “Argumenty połączenia z repozytorium IBM SPSS Collaboration and Deployment Services Repository” na stronie 65.
Argumenty połączenia z repozytorium IBM SPSS Analytic Server	Więcej informacji można znaleźć w temacie “Argumenty połączenia z repozytorium IBM SPSS Analytic Server” na stronie 65.

Na przykład można użyć flag `-server`, `-stream` i `-execute`, aby nawiązać połączenie z serwerem, a następnie załadować i uruchomić strumień, w następujący sposób:

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

Należy zwrócić uwagę, że w przypadku uruchamiania lokalnej instalacji klienckiej nie są wymagane argumenty połączenia z serwerem.

Wartości parametrów zawierające spacje można ująć w cudzysłów, na przykład:

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

W ten sam sposób można także wywoływać stany i skrypty programu IBM SPSS Modeler, korzystając odpowiednio z flag `-state` i `-script`.

Uwaga: Jeśli w komendzie używany jest parametr ustrukturyzowany, należy poprzedzać cudzysłowy ukośnikami odwrotnymi. Zapobiega to usuwaniu cudzysłówów podczas interpretacji łańcucha.

Debugowanie argumentów wiersza komend

Aby debugować wiersz komend, należy użyć komendy `modelerclient` w celu uruchomienia programu IBM SPSS Modeler z żądanymi parametrami. Pozwoli to sprawdzić, czy komendy zostaną wykonane zgodnie z oczekiwaniami. W oknie dialogowym Parametry sesji można także sprawdzić wartości parametrów przekazywanych z wiersza komend (menu Narzędzia, Ustaw parametry sesji).

Argumenty systemowe

W poniższej tabeli opisano argumenty systemowe, których można używać przy wywoływaniu interfejsu użytkownika z wiersza komend.

Tabela 34. Argumenty systemowe

Argument	Działanie/opis
@ <plikKomend>	Znak @ wraz z następującą po nim nazwą pliku określa listę komend. Gdy <code>modelerclient</code> napotka argument rozpoczynający się od znaku @, wykonuje komendy z pliku tak, jak gdyby były wywoływane z wiersza komend. Więcej informacji można znaleźć w temacie “Łączne stosowanie wielu argumentów” na stronie 66.
-directory <katalog>	Ustawia domyślny katalog roboczy. W trybie lokalnym w tym katalogu są zapisywane zarówno dane, jak i wyniki. Przykład: <code>-directory c:/</code> lub <code>-directory c:\</code>
-server_directory <dir>	Ustawia domyślny katalog dla danych. Wyniki zapisywane są w katalogu roboczym określonym za pomocą flagi <code>-directory</code> .
-execute	Po uruchomieniu wykonuje dowolny strumień, stan lub skrypt załadowany po uruchomieniu programu. Jeśli załadowany jest zarówno skrypt, jak i strumień lub stan, to wykonany zostanie tylko skrypt.
-stream <strumień>	Powoduje załadowanie określonego strumienia bezpośrednio po uruchomieniu programu. Można określić więcej niż jeden strumień, ale ostatni z wymienionych stanie się strumieniem bieżącym.
-script <skrypt>	Powoduje załadowanie określonego skryptu autonomicznego bezpośrednio po uruchomieniu programu. Można określić zarówno skrypt, jak i strumień lub stan (patrz niżej), ale wówczas przy uruchamianiu zostanie załadowany tylko skrypt.
-model <model>	Powoduje załadowanie wygenerowanego modelu (pliku w formacie <code>.gm</code>) bezpośrednio po uruchomieniu programu.
-state <stan>	Powoduje załadowanie określonego zapisanego stanu bezpośrednio po uruchomieniu programu.
-project <projekt>	Ładuje określony projekt. Bezpośrednio przy uruchamianiu programu można załadować tylko jeden projekt.
-output <wynik>	Powoduje załadowanie zapisanego obiektu wynikowego (pliku w formacie <code>.cou</code>) bezpośrednio po uruchomieniu programu.
-help	Wyświetla listę argumentów dostępnych w wierszu komend. Gdy ta opcja jest określona, wszystkie pozostałe argumenty są ignorowane i wyświetlany jest ekran pomocy.
-P <nazwa>=<wartość>	Służy do określania wartości parametru startowego. Umożliwia także ustawianie właściwości węzłów (parametrów zagnieżdżonych).

Uwaga: Katalogi domyślne można też wybrać w interfejsie użytkownika. Aby uzyskać dostęp do odpowiednich opcji, w menu Plik wybierz opcję **Katalog klienta** lub **Katalog serwera**.

Ładowanie więcej niż jednego pliku

Z wiersza komend można nakazać załadowanie wielu strumieni, stanów lub wyników zaraz po uruchomieniu programu, powtarzając odpowiednie argumenty dla każdego ładowanego obiektu. Na przykład, aby załadować i uruchomić dwa skrypty o nazwach `report.str` i `train.str`, należałoby użyć komendy:

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

Ładowanie obiektów z repozytorium IBM SPSS Collaboration and Deployment Services Repository

Ponieważ określone obiekty można ładować albo z pliku, albo z repozytorium IBM SPSS Collaboration and Deployment Services Repository (jeśli jest aktywna licencja na ten komponent), przedrostek nazwy pliku `spsscr:` lub opcjonalnie `file:` (w przypadku obiektów na dysku) informuje program IBM SPSS Modeler, gdzie poszukiwać obiektu. Przedrostek można stosować z następującymi flagami:

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

Przedrostka używa się do utworzenia identyfikatora URI określającego położenie obiektu, na przykład `-stream "spsscr:///folder_1/scoring_stream.str"`. Obecność przedrostka `spsscr:` wymusza konieczność określenia w tej samej komendzie poprawnego połączenia z repozytorium IBM SPSS Collaboration and Deployment Services Repository. Zatem, na przykład, cała komenda miałaby postać:

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

Należy zauważyć, że w wierszu komend *wymagane jest* użycie identyfikatora URI. Prostsza forma `REPOSITORY_PATH` nie jest obsługiwana. (Działa tylko w skryptach). Więcej informacji o identyfikatorach URI obiektów w repozytorium IBM SPSS Collaboration and Deployment Services Repository zawiera temat “Uzyskiwanie dostępu do obiektów w repozytorium IBM SPSS Collaboration and Deployment Services Repository” na stronie 49.

Argumenty określające parametry

Parametry można przekazywać jako flagi w wywołaniu programu z wiersza komend IBM SPSS Modeler. W argumentach w wierszu komend flaga `-P` służy do definiowania parametrów w postaci `-P <nazwa>=<wartość>`.

Wyróżnia się następujące rodzaje parametrów:

- **Parametry proste** (używane bezpośrednio w wyrażeniach programu CLEM).
- **Parametry zagnieżdżone**, nazywane także **właściwościami węzłów**. Te parametry służą do modyfikowania ustawień węzłów w strumieniu. Więcej informacji można znaleźć w temacie “Przegląd właściwości węzłów” na stronie 69.
- **Parametry wiersza komend** używane do modyfikowania wywołania programu IBM SPSS Modeler.

Można na przykład przekazywać nazwy użytkowników i hasła do źródeł danych jako flagi w wierszu komend, w następujący sposób:

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

Obowiązuje ten sam format, co dla parametru `datasource` właściwości węzła `databasenode`. Aby uzyskać więcej informacji, patrz: “Właściwości węzła `databasenode`” na stronie 82.

Uwaga: Jeśli węzeł jest nazwany, należy ująć nazwę węzła w cudzysłów i poprzedzać cudzysłowy ukośnikami odwrotnymi. Na przykład, gdyby węzeł źródła danych w poprzednim przykładzie miał nazwę `Source_ABC`, to wpis miałby postać:

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

Ukośniki odwrotne są też wymagane przed cudzysłowami oznaczającymi parametry ustrukturyzowane, co ilustruje poniższy przykład dotyczący źródła danych TMI:

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

Argumenty połączenia z serwerem

Flaga `-server` informuje IBM SPSS Modeler, że powinien łączyć się z serwerem publicznym, a flagi `-hostname`, `-use_ssl`, `-port`, `-username`, `-password` oraz `-domain` informują IBM SPSS Modeler, jak połączyć się z serwerem publicznym. Jeśli argument `-server` nie zostanie określony, używany jest serwer domyślny lub lokalny.

Przykłady

Aby połączyć się z serwerem publicznym:

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

Aby połączyć się z klastrem serwerów:

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

Uwaga: do nawiązania połączenia z klastrem serwerów wymagany jest koordynator procesów działający za pośrednictwem IBM SPSS Collaboration and Deployment Services, zatem argument `-cluster` musi być używany razem z opcjami połączenia z repozytorium (`spsscr_*`). Więcej informacji można znaleźć w temacie “Argumenty połączenia z repozytorium IBM SPSS Collaboration and Deployment Services Repository” na stronie 65.

Tabela 35. Argumenty połączenia z serwerem.

Argument	Działanie/opis
<code>-server</code>	Uruchamia IBM SPSS Modeler w trybie serwerowym i nawiązuje połączenie z serwerem publicznym na podstawie flag <code>-hostname</code> , <code>-port</code> , <code>-username</code> , <code>-password</code> i <code>-domain</code> .
<code>-hostname <nazwa></code>	Nazwa hosta serwera. Argument dostępny tylko w trybie serwerowym.
<code>-use_ssl</code>	Określa, że w połączeniu ma być używany protokół SSL (secure socket layer). Ta flaga jest opcjonalna; domyślnie protokół SSL <i>nie</i> jest używany.
<code>-port <numer></code>	Numer portu na określonym serwerze. Argument dostępny tylko w trybie serwerowym.
<code>-cluster <nazwa></code>	Określa połączenie z serwerem klastrów, a nie z nazwanym serwerem; ten argument jest alternatywą dla argumentów <code>hostname</code> , <code>port</code> i <code>use_ssl</code> . Nazwa jest nazwą klastra lub unikalnym identyfikatorem URI klastra w repozytorium IBM SPSS Collaboration and Deployment Services Repository. Klastrem serwerów zarządza koordynator procesów za pośrednictwem komponentu IBM SPSS Collaboration and Deployment Services. Więcej informacji można znaleźć w temacie “Argumenty połączenia z repozytorium IBM SPSS Collaboration and Deployment Services Repository” na stronie 65.
<code>-username <nazwa></code>	Nazwa użytkownika, z użyciem której klient ma się logować do serwera. Argument dostępny tylko w trybie serwerowym.
<code>-password <hasło></code>	Hasło, z użyciem którego klient ma się logować do serwera. Argument dostępny tylko w trybie serwerowym. Uwaga: Jeśli argument <code>-password</code> nie zostanie użyty, pojawi się monit o hasło.
<code>-epassword <łańcuchzakodowanegohasła></code>	Hasło zakodowane, z użyciem którego klient ma się logować do serwera. Argument dostępny tylko w trybie serwerowym. Uwaga: Zakodowane hasło można wygenerować z menu Narzędzia w aplikacji IBM SPSS Modeler.

Tabela 35. Argumenty połączenia z serwerem (kontynuacja).

Argument	Działanie/opis
-domain <nazwa>	Domena, która ma być używana do zalogowania się na serwerze. Argument dostępny tylko w trybie serwerowym.
-P <nazwa>=<wartość>	Służy do określania wartości parametru startowego. Umożliwia także ustawianie właściwości węzłów (parametrów zagnieżdżonych).

Argumenty połączenia z repozytorium IBM SPSS Collaboration and Deployment Services Repository

Jeśli chcesz zapisywać lub odczytywać obiekty z komponentu IBM SPSS Collaboration and Deployment Services za pośrednictwem wiersza komend, musisz określić poprawne połączenie z repozytorium IBM SPSS Collaboration and Deployment Services Repository. Na przykład:

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

W poniższej tabeli wymieniono argumenty, których można używać do konfigurowania połączenia.

Tabela 36. Argumenty połączenia z repozytorium IBM SPSS Collaboration and Deployment Services Repository

Argument	Działanie/opis
-spsscr_hostname <nazwa hosta lub adres IP>	Nazwa hosta lub adres IP serwera, na którym jest zainstalowane repozytorium IBM SPSS Collaboration and Deployment Services Repository.
-spsscr_port <numer>	Numer portu, pod którym IBM SPSS Collaboration and Deployment Services Repository przyjmuje połączenia (zwykle obowiązuje numer domyślny 8080).
-spsscr_use_ssl	Określa, że w połączeniu ma być używany protokół SSL (secure socket layer). Ta flaga jest opcjonalna; domyślnie protokół SSL <i>nie</i> jest używany.
-spsscr_username <nazwa>	Nazwa użytkownika, którego konto ma być używane do logowania się do repozytorium IBM SPSS Collaboration and Deployment Services Repository.
-spsscr_password <hasło>	Hasło, które ma być używane do logowania się do repozytorium IBM SPSS Collaboration and Deployment Services Repository.
-spsscr_epassword <hasło kodowane>	Hasło kodowane, które ma być używane do logowania się do repozytorium IBM SPSS Collaboration and Deployment Services Repository.
-spsscr_providername <nazwa>	Dostawca uwierzytelniania używany przy logowaniu do IBM SPSS Collaboration and Deployment Services Repository (Active Directory lub LDAP). Nie jest wymagany w przypadku korzystania z rodzimego dostawcy (repozytorium lokalnego).

Argumenty połączenia z repozytorium IBM SPSS Analytic Server

Jeśli chcesz zapisywać lub odczytywać obiekty w/z repozytorium IBM SPSS Analytic Server za pośrednictwem wiersza komend, musisz określić poprawne połączenie z repozytorium IBM SPSS Analytic Server.

Uwaga: Domyślna lokalizacja produktu Analytic Server jest pobierana z produktu SPSS Modeler Server. Użytkownicy mogą również definiować własne połączenia produktu Analytic Server za pomocą opcji **Narzędzia > Połączenia programu Analytic Server**.

W poniższej tabeli wymieniono argumenty, których można używać do konfigurowania połączenia.

Tabela 37. Argumenty połączenia z repozytorium IBM SPSS Analytic Server

Argument	Działanie/opis
-analytic_server_username	Nazwa użytkownika, którego konto ma być używane do logowania się do repozytorium IBM SPSS Analytic Server.
-analytic_server_password	Hasło, które ma być używane do logowania się do repozytorium IBM SPSS Analytic Server.
-analytic_server_epassword	Zakodowane hasło, które ma być używane do logowania się do repozytorium IBM SPSS Analytic Server.
-analytic_server_credential	Dane uwierzytelniające, które mają być używane do logowania się do repozytorium IBM SPSS Analytic Server.

Łączne stosowanie wielu argumentów

Wiele argumentów można połączyć w jeden plik komend określany przy wywołaniu za pomocą symbolu @, po którym powinna nastąpić nazwa pliku. W ten sposób można skrócić wywołanie w wierszu komend i ominąć ewentualne ograniczenie w długości komendy narzucone przez system operacyjny. Na przykład następująca komenda uruchomienia używa argumentów określonych w pliku o nazwie <commandFileName>.

```
modelerclient @<commandFileName>
```

Jeśli nazwa pliku komend i ścieżka do niego zawiera spację, należy ująć te elementy w cudzysłów, w następujący sposób:

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

Plik komend może zawierać wszystkie argumenty, które miałyby zawierać komenda, przy czym w jednym wierszu pliku powinien znajdować się jeden argument. Na przykład:

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

Zapisując pliki komend i odwołując się do nich, należy przestrzegać następujących reguł:

- W jednym wierszu należy zapisywać tylko jeden argument.
- Nie należy umieszczać argumentu @PlikKomend wewnątrz pliku komend.

Rozdział 7. Skorowidz właściwości

Przegląd skorowidza właściwości

Można nadawać wartości różnym właściwościom węzłów, strumieni, projektów i superwęzłów. Niektóre właściwości, takie jak nazwa, adnotacja i tekst podpowiedzi, są wspólne dla wszystkich węzłów, natomiast inne są charakterystyczne dla określonych typów węzłów. Jeszcze inne właściwości dotyczą wysokopoziomowych operacji na strumieniu, takich jak buforowanie w pamięci podręcznej lub zachowanie superwęzłów. Dostęp do właściwości można uzyskiwać za pośrednictwem standardowego interfejsu użytkownika (np. po otwarciu okna dialogowego w celu edycji opcji węzła), a także na różne inne sposoby.

- Właściwości można modyfikować za pomocą skryptów, tak jak opisano to w niniejszej sekcji. Aby uzyskać więcej informacji, patrz “Składnia właściwości”.
- Właściwości węzłów mogą być używane w parametrach superwęzłów.
- Właściwości węzłów mogą być także używane w opcjach wiersza komend (przy użyciu flagi `-P`) w wywołaniu programu IBM SPSS Modeler.

W kontekście środowiska skryptowego programu IBM SPSS Modeler właściwości węzłów i strumieni są często nazywane **parametrami zagnieżdżonymi**. W tym podręczniku nazywamy je właściwościami węzłów lub strumieni.

Aby uzyskać więcej informacji o języku skryptowym, patrz Język skryptowy.

Składnia właściwości

Właściwościom można nadawać wartości przy użyciu następującej składni

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

lub:

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

Wartości właściwości można odczytywać przy użyciu następującej składni:

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

lub:

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

gdzie `OBJECT` jest węzłem lub obiektem wynikowym, `PROPERTY` jest nazwą właściwości węzła, do której odwołuje się wyrażenie, a `KEY` jest kluczem w przypadku właściwości kluczowanych. Na przykład poniższy kod znajduje węzeł filtrowania, ustawia domyślną opcję uwzględniania wszystkich zmiennych i odfiltrowuje zmienną `Age` z danych przekazywanych do następnym węzłów:

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

Wszystkie węzły używane w programie IBM SPSS Modeler można znajdować za pomocą funkcji `findByType(TYPE, LABEL)` obiektu `stream`. Należy określić co najmniej jeden z elementów `TYPE` lub `LABEL`.

Właściwości ustrukturyzowane

Właściwości ustrukturyzowane, które zwiększają przejrzystość kodu, używa się w dwóch celach:

- Aby ustrukturyzować nazwy właściwości skomplikowanych węzłów, takich jak `Typ`, `Filtruj` lub `Zrównoważenie`.
- Aby nadawać wartości wielu właściwościom jednocześnie w spójnym formacie.

Struktura skomplikowanych interfejsów

W skryptach operujących na węzłach z tabelami i innymi skomplikowanymi interfejsami (np. węzłach Typ, Filtruj i Zrównoważenie) musi być stosowana określona struktura, by możliwe było prawidłowe przeanalizowanie ich składni. Właściwości takich węzłów muszą mieć nazwy bardziej złożone niż prosty identyfikator. Taka złożona nazwa to tzw. klucz. Na przykład w węźle Filtruj każda dostępna zmienna (po stronie wejściowej) może być włączona albo wyłączona. Aby możliwe było odwoływanie się do tych informacji, węzeł Filtruj zawiera po jednej dodatkowej informacji o każdej zmiennej (true albo false). Właściwość ta może przyjmować (lub można jej przypisać) wartość True albo False. Załóżmy, że węzeł filtrowania o nazwie mynode zawiera (po stronie wejściowej) zmienną o nazwie Age. Aby ją wyłączyć, należy nadać właściwości include z kluczem Age wartość False w następujący sposób:

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

Nadawanie wartości wielu właściwościom jednocześnie

W przypadku wielu węzłów możliwe jest jednoczesne nadawanie wartości więcej niż jednej właściwości węzła lub strumienia. Operację taką nazywamy **ustawianiem wielokrotnym** lub **ustawianiem blokowym**.

W niektórych przypadkach właściwość ustrukturyzowana może być stosunkowo skomplikowana. Przykład:

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

Kolejną zaletą właściwości ustrukturyzowanych jest możliwość nadania za ich pośrednictwem wartości kilku właściwościom zanim węzeł osiągnie stan stabilny. Domyślnie operacja ustawiania wielokrotnego nadaje wartości wszystkim właściwościom w bloku zanim podjęte zostanie jakiekolwiek działanie w oparciu o wartość pojedynczej właściwości. Na przykład podczas definiowania węzła Plik kolumnowy ustawienie właściwości w dwóch krokach spowodowałoby błędy, ponieważ dopóki obie właściwości nie będą poprawne, węzeł pozostaje w stanie niestabilnym. Blokowe nadanie wartości właściwościom pozwala uniknąć tego problemu, ponieważ zanim model danych zostanie zmieniony, obie właściwości będą już miały poprawne wartości.

Skróty

W odwołaniach do właściwości węzłów używane są pewne standardowe skróty. Znajomość tych skrótów będzie przydatna przy tworzeniu własnych skryptów.

Tabela 38. Standardowe skróty używane w składni skryptów

Skrót	Znaczenie
abs	Wartość bezwzględna
len	Długość
min.	Minimum
maks.	Maksimum
correl	Korelacja
covar	Kowariancja
num	Liczba lub numeryczny
pct	Procent lub odsetek
transp	Przezroczystość
xval	Walidacja krzyżowa
var	Wariancja lub zmienna (w węzłach źródłowych)

Przykłady właściwości węzłów i strumieni

Właściwości węzłów i strumieni można w programie IBM SPSS Modeler wykorzystać na różne sposoby. Najczęściej używane są w skryptach — w **skryptach samodzielnych** służących do automatyzacji wielu operacji lub strumieni, albo **skryptach strumieni** służących do automatyzacji procesów w jednym strumieniu. Można też nadawać wartości parametrom węzła, korzystając z właściwości węzłów w superwęźle. Na najbardziej podstawowym poziomie

właściwości mogą być także używane w opcjach uruchamiania programu IBM SPSS Modeler z wiersza komend. Korzystając z argumentu `-p` w wywołaniu programu z wiersza komend, można za pośrednictwem właściwości strumienia zmienić jakieś ustawienie w tym strumieniu.

Tabela 39. Przykłady właściwości węzłów i strumieni

Właściwość	Znaczenie
<code>s.max_size</code>	Odwołanie do właściwości <code>max_size</code> węzła o nazwie <code>s</code> .
<code>s:samplenode.max_size</code>	Odwołanie do właściwości <code>max_size</code> węzła o nazwie <code>s</code> , który musi być węzłem próby.
<code>:samplenode.max_size</code>	Odwołanie do właściwości <code>max_size</code> węzła próby w bieżącym strumieniu (musi istnieć tylko jeden węzeł próby).
<code>s:sample.max_size</code>	Odwołanie do właściwości <code>max_size</code> węzła o nazwie <code>s</code> , który musi być węzłem próby.
<code>t.direction.Age</code>	Odwołanie do roli zmiennej <code>Age</code> w węźle Typ o nazwie <code>t</code> .
<code>:.max_size</code>	*** NIEDOZWOLONE *** Należy określić albo nazwę, albo typ węzła.

Jak pokazuje przykład `s:sample.max_size`, nie trzeba wpisywać całych nazw typów węzłów.

Jak pokazuje przykład `t.direction.Age`, niektóre nazwy parametrów mogą być ustrukturyzowane — gdy atrybuty węzła są czymś więcej niż miejscami na pojedyncze wartości. Takie parametry nazywamy właściwościami **ustrukturyzowanymi** lub **złożonymi**.

Przegląd właściwości węzłów

Każdy typ węzła ma własny zestaw dozwolonych właściwości, a każda właściwość ma określony typ. Może to być typ ogólny — liczba, flaga lub łańcuch. W takim przypadku ustawienia właściwości są interpretowane tak, jak gdyby należały do określonego typu. Jeśli ustawienia (wartości) właściwości nie da się tak zinterpretować, to zgłaszany jest błąd. Dla właściwości może być także określony zbiór dozwolonych wartości, na przykład `Discard`, `PairAndDiscard` i `IncludeAsText`. W takim przypadku każda wartość spoza zbioru powoduje zgłoszenie błędu. Właściwości typu flaga należy odczytywać i zapisywać, posługując się wartościami `true` (prawda) albo `false` (fałsz). (Przy ustawianiu wartości dozwolone są także różne warianty, w tym `Off`, `OFF`, `off`, `No`, `NO`, `no`, `n`, `N`, `f`, `F`, `false`, `False`, `FALSE` i `0`, ale przy odczytywaniu wartości mogą one w niektórych przypadkach powodować błędy. Wszystkie pozostałe wartości są traktowane jak `true`. Spójne posługiwanie się wartościami `true` i `false` pozwoli uniknąć wszelkich niejednoznaczności). W tabelach zamieszczonych w niniejszym podręczniku właściwości ustrukturyzowane są oznaczone w kolumnie **Opis właściwości** i podane są formaty ich zastosowania.

Wspólne właściwości węzłów

Istnieje szereg właściwości wspólnych dla wszystkich węzłów (w tym także dla superwęzłów) w programie IBM SPSS Modeler.

Tabela 40. Wspólne właściwości węzłów.

Nazwa właściwości	Typ danych	Opis właściwości
<code>use_custom_name</code>	<i>flag</i>	
<code>name</code>	<i>string</i>	Właściwość przeznaczona tylko do odczytu, która odczytuje nazwę (automatyczną lub niestandardową) węzła w obszarze roboczym.
<code>custom_name</code>	<i>string</i>	Określa niestandardową nazwę węzła.
<code>tooltip</code>	<i>string</i>	
<code>annotation</code>	<i>string</i>	

Tabela 40. Wspólne właściwości węzłów (kontynuacja).

Nazwa właściwości	Typ danych	Opis właściwości
keywords	<i>string</i>	Parametr ustrukturyzowany, który określa listę słów kluczowych związanych z obiektem (na przykład ["Keyword1" "Keyword2"]).
cache_enabled	<i>flag</i>	
node_type	source_supernode process_supernode terminal_supernode wszystkie nazwy węzłów określone dla skrypty	Właściwość przeznaczona tylko do odczytu, która służy do odwoływania się do węzła po jego typie. Na przykład zamiast odwoływać się do węzła tylko przy użyciu nazwy, takiej jak <code>real_income</code> , można określić typ węzła, taki jak <code>userinputnode</code> lub <code>filternode</code> .

Właściwości charakterystyczne dla superwęzłów zostały omówione osobno, tak jak właściwości poszczególnych typów węzłów. Więcej informacji można znaleźć w rozdziale Rozdział 20, “Właściwości superwęzłów”, na stronie 339.

Rozdział 8. Właściwości strumienia

Za pomocą skryptów można sterować różnymi właściwościami strumienia. Aby odwoływać się do właściwości strumienia, należy wybrać skrypt jako metodę wykonania:

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

Przykład

Właściwość `node` służy do odwoływania się do węzłów w bieżącym strumieniu. Oto przykładowy skrypt strumienia.

```
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)
```

W powyższym przykładzie zastosowano właściwość `node` do utworzenia listy wszystkich węzłów w strumieniu i zapisano tę listę w adnotacjach do strumienia. Wygenerowana adnotacja wygląda tak

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"
```

Właściwości strumienia są opisane w poniższej tabeli.

Tabela 41. Właściwości strumienia.

Nazwa właściwości	Typ danych	Opis właściwości
execute_method	Normal Script	

Tabela 41. Właściwości strumienia (kontynuacja).

Nazwa właściwości	Typ danych	Opis właściwości
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	number	
date_2digit_baseline	number	
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	flag	
import_datetime_as_string	flag	
decimal_places	number	
decimal_symbol	Default Period Comma	
angles_in_radians	flag	
use_max_set_size	flag	
max_set_size	number	
ruleset_evaluation	Voting FirstHit	

Tabela 41. Właściwości strumienia (kontynuacja).

Nazwa właściwości	Typ danych	Opis właściwości
refresh_source_nodes	flag	Służy do automatycznego odświeżania węzłów źródłowych przy wykonywaniu strumienia.
script	string	
annotation	string	
name	string	Uwaga: Ta właściwość jest dostępna tylko do odczytu. Aby zmienić nazwę strumienia, należy zapisać go pod inną nazwą.
parameters		Ta właściwość służy do aktualizowania parametrów strumienia ze skryptu autonomicznego.
nodes		Szczegółowe informacje znajdują się poniżej.
encoding	SystemDefault "UTF-8"	
stream_rewriting	boolean	
stream_rewriting_maximise_sql	boolean	
stream_rewriting_optimise_clem_execution	boolean	
stream_rewriting_optimise_syntax_execution	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	Wartość true powoduje użycie ustawień charakterystycznych dla strumienia, w przeciwnym razie obowiązują preferencje użytkownika.
create_model_applier_for_new_models	boolean	Wartość true powoduje, że gdy tworzony jest nowy model bez aktywnych odsyłaczy aktualizacji, dodawany jest nowy model do zastosowania. Uwaga: W wersji 15 programu IBM SPSS Modeler Batch należy jawnie dodać model do zastosowania w skrypcie.
create_model_applier_update_links	createEnabled createDisabled doNotCreate	Definiuje typ odsyłacza tworzony przy automatycznym dodawaniu węzła modelu do zastosowania.

Tabela 41. Właściwości strumienia (kontynuacja).

Nazwa właściwości	Typ danych	Opis właściwości
create_source_node_from_builders	<i>boolean</i>	Wartość true powoduje, że gdy tworzony jest nowy wynik źródłowy bez aktywnych odsyłaczy aktualizacji, dodawany jest nowy węzeł źródłowy.
create_source_node_update_links	createEnabled createDisabled doNotCreate	Definiuje typ odsyłacza tworzony przy automatycznym dodawaniu węzła źródłowego.
has_coordinate_system	<i>boolean</i>	Wartość true powoduje, że układ współrzędnych jest stosowany względem całego strumienia.
coordinate_system	<i>string</i>	Nazwa wybranego rzutowanego układu współrzędnych.
deployment_area	ModelRefresh Określanie wyniku None	Zdecyduj, jak chcesz wdrożyć strumień. Jeśli ta wartość jest ustawiona jako None, nie są używane inne pozycje wdrożenia.
scoring_terminal_node_id	<i>string</i>	Wybierz gałąź oceniania w strumieniu. Może to być dowolny węzeł końcowy w strumieniu.
scoring_node_id	<i>string</i>	Wybierz model użytkowy w gałęzi oceniania.
model_build_node_id	<i>string</i>	Wybierz węzeł modelowania w strumieniu.

Rozdział 9. Właściwości węzłów źródłowych

Właściwości wspólne węzłów źródłowych

Poniżej wymieniono właściwości wspólne dla wszystkich węzłów źródłowych, a w następnych tematach opisano właściwości charakterystyczne dla poszczególnych węzłów.

Przykład 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")
```

Przykład 2

W tym skrypcie założono, że określony plik danych zawiera zmienną Region reprezentującą łańcuch wielowierszowy.

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

# Utwórz węzeł Plik zmiennych, który odczytuje zbiór danych
# zawierający zmienną "Region"
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)

# Wymuś typ składowania lista...
varfilenode.setKeyedPropertyValue("custom_storage_type", "Region", StorageType.LIST)
# ...i określ typ, jeśli wartości są na liście, a głębokość listy
varfilenode.setKeyedPropertyValue("custom_list_storage_type", "Region", StorageType.INTEGER)
varfilenode.setKeyedPropertyValue("custom_list_depth", "Region", 2)

# Teraz zmień pomiar, aby określić zmienną jako wartość geoprzestrzenną...
varfilenode.setKeyedPropertyValue("measure_type", "Region", MeasureType.GEOSPATIAL)
# ...
i na koniec określ wszystkie niezbędne informacje
# o typie obiektu geoprzestrzennego
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")
```

Tabela 42. Właściwości wspólne węzłów źródłowych.

Nazwa właściwości	Typ danych	Opis właściwości
direction	Input Target Both None Partition Split Frequency RecordID	Właściwość wprowadzana określająca role zmiennych. Format stosowania: NODE.direction.FIELDNAME Uwaga: Wartości In i Out są obecnie nieaktualne. Ich obsługa może zostać wycofana w przyszłej wersji produktu.

Tabela 42. Właściwości wspólne węzłów źródłowych (kontynuacja).

Nazwa właściwości	Typ danych	Opis właściwości
type	Range Flag Set Typeless Discrete Ordered Set Default	Typ zmiennej. Nadanie tej właściwości wartości <i>Default</i> spowoduje skasowanie ustawienia właściwości <i>values</i> , a jeśli <i>value_mode</i> ma wartość <i>Specify</i> , to otrzyma wartość <i>Read</i> . Jeśli właściwość <i>value_mode</i> już ma wartość <i>Pass</i> lub <i>Read</i> , to ustawienie <i>type</i> nie będzie miało na nią wpływu. Format stosowania: NODE.type.FIELDNAME
storage	Unknown String Integer Real Time Date Timestamp	Właściwość wprowadzana przeznaczona tylko do odczytu; określa typ składowania zmiennej. Format stosowania: NODE.storage.FIELDNAME
check	None Nullify Coerce Discard Warn Abort	Właściwość wprowadzana określająca typ zmiennej i zasady sprawdzania zakresu. Format stosowania: NODE.check.FIELDNAME
values	[value value]	W przypadku zmiennej ciągłej (przedziałowej) pierwsza wartość to minimum, a ostatnia wartość to maksimum. W przypadku zmiennych nominalnych (zbiorów) należy określić wszystkie wartości. W przypadku zmiennych typu flaga pierwsza wartość reprezentuje <i>false</i> , a ostatnia <i>true</i> . Nadanie wartości tej właściwości powoduje, że właściwość <i>value_mode</i> jest automatycznie ustawiana na <i>Specify</i> . Typ składowania jest określany na podstawie pierwszej wartości na liście. Na przykład, jeśli pierwsza wartość jest łańcuchem (<i>string</i>), to za typ składowania przyjmuje się łańcuch. Format stosowania: NODE.values.FIELDNAME
value_mode	Read Pass Read+ Current Specify	Określa sposób nadawania wartości zmiennej w następnym przebiegu przez dane. Format stosowania: NODE.value_mode.FIELDNAME Należy zwrócić uwagę, że nie można nadać tej właściwości bezpośrednio wartości <i>Specify</i> ; aby użyć konkretnych wartości, należy przypisać je do właściwości <i>values</i> .
default_value_mode	Read Pass	Określa domyślną metodę nadawania wartości wszystkim zmiennym. Format stosowania: NODE.default_value_mode To ustawienie można przesłonić dla konkretnych zmiennych, używając właściwości <i>value_mode</i> .

Tabela 42. Właściwości wspólne węzłów źródłowych (kontynuacja).

Nazwa właściwości	Typ danych	Opis właściwości
extend_values	<i>flag</i>	Ma zastosowanie wtedy, gdy <i>value_mode</i> ma wartość <i>Read</i> . Ustawienie <i>T</i> powoduje dodawanie nowych wczytanych wartości do istniejących wartości zmiennej. Ustawienie <i>F</i> powoduje odrzucenie istniejących wartości i zastąpienie ich nowo wczytanymi. Format stosowania: NODE.extend_values.FIELDNAME
value_labels	<i>string</i>	Służy do określania etykiety wartości. Najpierw muszą zostać określone wartości.
enable_missing	<i>flag</i>	Ustawienie <i>T</i> włącza rejestrowanie braków danych dla zmiennej. Format stosowania: NODE.enable_missing.FIELDNAME
missing_values	[<i>value value ...</i>]	Określa wartości danych oznaczające braki danych. Format stosowania: NODE.missing_values.FIELDNAME
range_missing	<i>flag</i>	Gdy ta właściwość jest ustawiona na <i>T</i> , określa, czy dla zmiennej zdefiniowany jest przedział braków danych (wartości pustych). Format stosowania: NODE.range_missing.FIELDNAME
missing_lower	<i>string</i>	Gdy <i>range_missing</i> ma wartość <i>true</i> , określa dolną granicę przedziału braków danych. Format stosowania: NODE.missing_lower.FIELDNAME
missing_upper	<i>string</i>	Gdy <i>range_missing</i> ma wartość <i>true</i> , określa górną granicę przedziału braków danych. Format stosowania: NODE.missing_upper.FIELDNAME
null_missing	<i>flag</i>	Gdy ta właściwość jest ustawiona na <i>T</i> , wartości <i>null</i> (niezdefiniowane wartości są w programie wyświetlane jako <i>\$null\$</i>) są uznawane za braki danych. Format stosowania: NODE.null_missing.FIELDNAME
whitespace_missing	<i>flag</i>	Gdy ta właściwość jest ustawiona na <i>T</i> , wartości zawierające tylko białe znaki (spacje, znaki tabulacji i znaki nowego wiersza) są uznawane za braki danych. Format stosowania: NODE.whitespace_missing.FIELDNAME
description	<i>string</i>	Służy do określania etykiety lub opisu zmiennej.

Tabela 42. Właściwości wspólne węzłów źródłowych (kontynuacja).

Nazwa właściwości	Typ danych	Opis właściwości
default_include	<i>flag</i>	Właściwość wprowadzana, która określa, czy zmienne mają być domyślnie przekazywane, czy filtrowane: NODE.default_include Przykład: set mynode:filternode.default_include = false
include	<i>flag</i>	Właściwość wprowadzana służąca do określania, czy poszczególne pola mają być uwzględniane, czy filtrowane: 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	Ta właściwość wprowadzana jest podobna do właściwości <i>type</i> w tym sensie, że może służyć do definiowania poziomu pomiaru zmiennej. Jednak w skryptach w języku Python do funkcji ustawiającej można też przekazać jedną z wartości <i>MeasureType</i> natomiast funkcja pobierająca zawsze zwraca wartości <i>MeasureType</i> .
collection_measure	Range / MeasureType.RANGE Flag / MeasureType.FLAG Set / MeasureType.SET OrderedSet / MeasureType.ORDERED_SET Typeless / MeasureType.TYPELESS	W przypadku zmiennych <i>collection</i> (listy o głębokości 0) ta właściwość definiuje typ pomiaru związany z bazowymi wartościami.
geo_type	Point MultiPoint LineString MultiLineString Polygon MultiPolygon	W przypadku zmiennych geoprzestrzennych ta właściwość definiuje typ obiektu reprezentowanego przez zmienną. Powinien być on zgodny z głębokością listy wartości.
has_coordinate_system	<i>boolean</i>	W przypadku zmiennych geoprzestrzennych ta właściwość określa, czy zmienna ma układ współrzędnych.
coordinate_system	<i>string</i>	W przypadku zmiennych geoprzestrzennych ta właściwość definiuje układ współrzędnych dla zmiennej.
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	Ta właściwość wprowadzana jest podobna do właściwości <i>custom_storage</i> w tym sensie, że może służyć do definiowania przesłoniętego typu składowania zmiennej. Jednak w skryptach w języku Python do funkcji ustawiającej można też przekazać jedną z wartości <i>StorageType</i> natomiast funkcja pobierająca zawsze zwraca wartości <i>StorageType</i> .

Tabela 42. Właściwości wspólne węzłów źródłowych (kontynuacja).

Nazwa właściwości	Typ danych	Opis właściwości
custom_list_storage_type	String / MeasureType.STRING Integer / MeasureType.INTEGER Real / MeasureType.REAL Time / MeasureType.TIME Date / MeasureType.DATE Timestamp / MeasureType.TIMESTAMP	W przypadku zmiennych typu lista ta właściwość wprowadzana określa typ składowania wartości.
custom_list_depth	<i>integer</i>	W przypadku zmiennych typu lista ta właściwość wprowadzana określa głębokość zmiennej
max_list_length	<i>integer</i>	Tylko dla danych z <i>Geoprzestrzennym</i> lub <i>Przedziałowym</i> poziomem pomiaru. Należy ustawić maksymalną długość listy, ustawiając dozwoloną dla listy liczbę elementów.
max_string_length	<i>integer</i>	Tylko dla danych <i>beztypowych</i> podczas generowania kodu SQL w celu utworzenia tabeli. Należy wprowadzić największy łańcuch danych; umożliwi to wygenerowanie w tabeli kolumny wystarczająco dużej do pomieszczenia łańcucha.

Właściwości węzła asimport

Źródło Analytic Server umożliwia uruchamianie strumienia w systemie Hadoop Distributed File System (HDFS).

Przykład

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

Tabela 43. Właściwości węzła asimport.

Właściwości węzła asimport	Typ danych	Opis właściwości
data_source	<i>string</i>	Nazwa źródła danych.
use_default_as	<i>boolean</i>	Ustawienie wartości True powoduje użycie domyślnego połączenia Analytic Server skonfigurowanego w pliku serwera options.cfg. Ustawienie wartości False powoduje, że używane jest połączenie tego węzła.

Tabela 43. Właściwości węzła *asimport* (kontynuacja).

Właściwości węzła <i>asimport</i>	Typ danych	Opis właściwości
connection	["string", "string", "string", "string", "string", "string", "string", "string", "string"]	Właściwość zawierająca listę informacji o szczegółach połączenia Analytic Server. Format jest następujący: ["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"] Gdzie: is_secure_connect: oznacza, czy użyto połączenia bezpiecznego, i może mieć wartość true lub false. use-kerberos-auth: wskazuje, czy zastosowanie ma uwierzytelnianie kerberos, i może mieć wartość true lub false. enable-kerberos-debug: wskazuje, czy dla uwierzytelniania kerberos użyto trybu debugowania, i może mieć wartość true lub false.

Właściwości węzła *cognosimport*



Węzeł źródłowy IBM Cognos importuje dane z baz danych Cognos Analytics.

Przykład

```
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]"])
```

Tabela 44. Właściwości węzła *cognosimport*.

Właściwości węzła <i>cognosimport</i>	Typ danych	Opis właściwości
mode	Data Raport	Określa, czy importować dane (domyślnie), czy raporty programu Cognos.

Tabela 44. Właściwości węzła *cognosimport* (kontynuacja).

Właściwości węzła cognosimport	Typ danych	Opis właściwości
cognos_connection	["string", "flag", "string", "string", "string"]	<p>Właściwość zawierająca listę informacji o połączeniu z serwerem Cognos. Obowiązuje format: ["namespace", "tm1_username", "password"]</p> <p>gdzie: Cognos_server_URL to adres URL serwera Cognos zawierającego źródło. login_mode określa, czy jest używane logowanie anonimowe, i ma wartość true lub false; jeśli ta właściwość jest ustawiona na true, to wymienione poniżej pola należy ustawić na "". namespace określa dostawcę uwierzytelniania używanego do logowania się na serwerze. username i password to nazwa użytkownika i hasło używane do logowania na serwerze Cognos.</p> <p>Zamiast login_mode dostępne są także następujące tryby:</p> <ul style="list-style-type: none"> • anonymousMode. Na przykład: ['Cognos_server_url', 'anonymousMode', 'namespace', 'username', 'password'] • credentialMode. Na przykład: ['Cognos_server_url', 'credentialMode', 'namespace', 'username', 'password'] • storedCredentialMode. Na przykład: ['Cognos_server_url', 'storedCredentialMode', 'stored_credential_name'] <p>Gdzie stored_credential_name jest nazwą danych uwierzytelniających Cognos w repozytorium.</p>
cognos_package_name	string	<p>Ścieżka i nazwa pakietu Cognos, z którego importowane są obiekty danych, na przykład: /Public Folders/GOSALES</p> <p>Uwaga: Dozwolone są tylko ukośniki zwykłe (nie odwrotne).</p>
cognos_items	["field", "field", ..., "field"]	<p>Nazwa jednego lub większej liczby obiektów danych do zaimportowania. Format wartości <i>field</i> to [namespace].[query_subject].[query_item]</p>
cognos_filters	field	<p>Nazwa jednego lub większej liczby filtrów, które należy zastosować przed zaimportowaniem danych.</p>
cognos_data_parameters	list	<p>Wartości parametrów pytań o dane. Pary nazwa-wartość są ujęte w nawiasy kwadratowe, poszczególne pary oddzielone są od siebie przecinkami, natomiast cały łańcuch jest ujęty w nawiasy kwadratowe.</p> <p>Format: [{"param1", "value"}, ..., {"paramN", "value"}]</p>

Tabela 44. Właściwości węzła *cognosimport* (kontynuacja).

Właściwości węzła <i>cognosimport</i>	Typ danych	Opis właściwości
<i>cognos_report_directory</i>	<i>field</i>	Ścieżka (w środowisku) Cognos folderu lub pakietu, z którego mają być importowane raporty, na przykład: /Public Folders/GOSALES Uwaga: Dozwolone są tylko ukośniki zwykłe (nie odwrotne).
<i>cognos_report_name</i>	<i>field</i>	Ścieżka i nazwa raportu do zaimportowania w lokalizacji raportów.
<i>cognos_report_parameters</i>	<i>list</i>	Wartości parametrów raportu. Pary nazwa-wartość są ujęte w nawiasy kwadratowe, poszczególne pary oddzielone są od siebie przecinkami, natomiast cały łańcuch jest ujęty w nawiasy kwadratowe. Format: [[" <i>param1</i> ", " <i>value</i> "],...,[" <i>paramN</i> ", " <i>value</i> "]]

Właściwości węzła *databasenode*



Węzeł Baza danych może służyć do importowania danych z różnych innych programów (m.in. z Microsoft SQL Server, DB2, Oracle) za pośrednictwem interfejsu ODBC (Open Database Connectivity),

Przykład

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

Tabela 45. Właściwości węzła *databasenode*.

Właściwości węzła <i>databasenode</i>	Typ danych	Opis właściwości
<i>mode</i>	Table Query	Określ wartość <i>Table</i> , aby łączyć się z tabelą bazy danych w oparciu o ustawienia wybierane w oknie dialogowym, albo określ wartość <i>Query</i> , aby skierować zapytanie SQL do wybranej bazy danych.
<i>datasource</i>	<i>string</i>	Nazwa bazy danych (patrz także uwaga poniżej).
<i>username</i>	<i>string</i>	Szczegóły połączenia z bazą danych (patrz także uwaga poniżej).
<i>password</i>	<i>string</i>	

Tabela 45. Właściwości węzła *databasenode* (kontynuacja).

Właściwości węzła <i>databasenode</i>	Typ danych	Opis właściwości
credential	<i>string</i>	Nazwa danych uwierzytelniających zapisana w komponencie IBM SPSS Collaboration and Deployment Services. Można jej użyć zamiast właściwości <code>username</code> i <code>password</code> . Nazwa użytkownika i hasło w danych uwierzytelniających muszą zgadzać się z nazwą użytkownika i hasłem wymaganymi do uzyskania dostępu do bazy danych.
use_credential		Ustaw na <code>True</code> albo <code>False</code> .
epassword	<i>string</i>	Określa hasło kodowane, będące alternatywą dla hasła wpisanego na stałe w skrypcie. Więcej informacji można znaleźć w temacie “Generowanie hasła kodowanego” na stronie 52. Ta właściwość w czasie wykonywania jest dostępna tylko do odczytu.
tablename	<i>string</i>	Nazwa tabeli, do której chcesz uzyskać dostęp.
strip_spaces	None Left Right Both	Opcje odrzucania spacji wiodących i końcowych w łańcuchach.
use_quotes	AsNeeded Always Never	Określ, czy w zapytaniach wysyłanych do bazy danych nazwy tabel i kolumn mają być ujmowane w cudzysłów (na przykład, jeśli zawierają spacje lub znaki interpunkcyjne).
query	<i>string</i>	Określa kod SQL zapytania, które ma zostać wysłane.

Uwaga: Jeśli nazwa bazy danych (we właściwości `datasource`) zawiera jedną lub więcej spacji, kropek lub znaków podkreślenia, można użyć formatu z ukośnikiem odwrotnym i cudzysłowem, aby potraktować ją jako łańcuch. Na przykład: `"{\"db2v9.7.6_linux\"}"` lub `"{\"TDATA 131\"}"`. Ponadto wartości łańcuchów `datasource` zawsze ujmuj w cudzysłowy i nawiasy sześciennne, tak jak w poniższym przykładzie: `"{\"SQL Server\",spssuser,abcd1234,false}"`.

Uwaga: Jeśli nazwa bazy danych (we właściwości `datasource`) zawiera spację, to zamiast odrębnych właściwości `datasource`, `username` i `password` można użyć jednej właściwości `datasource` w następującym formacie:

Tabela 46. Właściwości węzła *databasenode* — charakterystyczne dla źródła danych.

Właściwości węzła <i>databasenode</i>	Typ danych	Opis właściwości
datasource	<i>string</i>	Format: [database_name,username,password[,true false]] Ostatni parametr jest używany z hasłami zaszyfrowanymi. Jeśli ma wartość <code>true</code> , hasło zostanie zdeszyfrowane przed użyciem.

Tego formatu należy używać także w przypadku zmiany źródła danych; jeśli jednak chcesz zmienić tylko nazwę użytkownika lub hasło, możesz użyć właściwości `username` lub `password`.

Właściwości węzła `datacollectionimportnode`



Węzeł Import danych z Data Collection importuje dane ankietowe oparte na modelu danych Data Collection używanym w oprogramowaniu do badań rynku. Aby możliwe było korzystanie z tego węzła, konieczne jest zainstalowanie programu Data Collection Data Library.

Przykład

```
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")
```

Tabela 47. Właściwości węzła `datacollectionimportnode`.

Właściwości węzła <code>datacollectionimportnode</code>	Typ danych	Opis właściwości
<code>metadata_name</code>	<i>string</i>	Nazwa MDSC. Wartość specjalna <code>DimensionsMDD</code> oznacza, że ma być używany standardowy dokument metadanych Data Collection. Do innych możliwych wartości należą: <code>mrADODsc</code> <code>mrI2dDsc</code> <code>mrLogDsc</code> <code>mrQdiDrsDsc</code> <code>mrQvDsc</code> <code>mrSampleReportingMDSC</code> <code>mrSavDsc</code> <code>mrSCDsc</code> <code>mrScriptMDSC</code> Wartość specjalna <code>none</code> oznacza brak MDSC.
<code>metadata_file</code>	<i>string</i>	Nazwa pliku, w którym są przechowywane metadane.
<code>casedata_name</code>	<i>string</i>	Nazwa CDSC. Do możliwych wartości należą: <code>mrADODsc</code> <code>mrI2dDsc</code> <code>mrLogDsc</code> <code>mrPunchDSC</code> <code>mrQdiDrsDsc</code> <code>mrQvDsc</code> <code>mrRdbDsc2</code> <code>mrSavDsc</code> <code>mrScDSC</code> <code>mrXmlDsc</code> Wartość specjalna <code>none</code> oznacza brak CDSC.

Tabela 47. Właściwości węzła `datacollectionimportnode` (kontynuacja).

Właściwości węzła <code>datacollectionimportnode</code>	Typ danych	Opis właściwości
<code>casedata_source_type</code>	Unknown File Folder UDL DSN	Określa typ źródła CDSC.
<code>casedata_file</code>	<i>string</i>	Gdy właściwość <code>casedata_source_type</code> ma wartość <i>File</i> , określa plik zawierający dane obserwacji.
<code>casedata_folder</code>	<i>string</i>	Gdy właściwość <code>casedata_source_type</code> ma wartość <i>Folder</i> , określa folder zawierający dane obserwacji.
<code>casedata_udl_string</code>	<i>string</i>	Gdy <code>casedata_source_type</code> ma wartość <i>UDL</i> , określa łańcuch połączenia OLD-DB dla źródła danych zawierającego dane obserwacji.
<code>casedata_dsn_string</code>	<i>string</i>	Gdy <code>casedata_source_type</code> ma wartość <i>DSN</i> , określa łańcuch połączenia ODBC dla źródła danych.
<code>casedata_project</code>	<i>string</i>	W przypadku odczytu z bazy danych Data Collection można wprowadzić nazwę projektu. W przypadku wszystkich innych typów danych obserwacji to ustawienie powinno pozostać puste.
<code>version_import_mode</code>	All Latest Specify	Definiuje sposób obsługi wersji.
<code>specific_version</code>	<i>string</i>	Gdy <code>version_import_mode</code> ma wartość <i>Specify</i> , określa wersję danych obserwacji do zaimportowania.
<code>use_language</code>	<i>string</i>	Określa, czy mają być używane etykiety we wskazanym języku.
<code>language</code>	<i>string</i>	Jeśli <code>use_language</code> ma wartość <i>true</i> , określa kod języka, który ma być używany przy importowaniu. Kod języka powinien być jednym z dostępnych w danych obserwacji.
<code>use_context</code>	<i>string</i>	Określa, czy należy zaimportować konkretny kontekst. Konteksty służą do różnicowania opisów związanych z odpowiedziami.
<code>context</code>	<i>string</i>	Jeśli <code>use_context</code> ma wartość <i>true</i> , definiuje kontekst do zaimportowania. Kontekst powinien być jednym z dostępnych w danych obserwacji.
<code>use_label_type</code>	<i>string</i>	Określa, czy należy zaimportować konkretny typ etykiet.
<code>label_type</code>	<i>string</i>	Jeśli <code>use_label_type</code> ma wartość <i>true</i> , definiuje typ etykiet do zaimportowania. Typ etykiet powinien być jednym z dostępnych w danych obserwacji.
<code>user_id</code>	<i>string</i>	W przypadku baz danych wymagających jawnego logowania można podać identyfikator użytkownika i hasło dostępu do źródła danych.
<code>password</code>	<i>string</i>	

Tabela 47. Właściwości węzła `datacollectionimportnode` (kontynuacja).

Właściwości węzła <code>datacollectionimportnode</code>	Typ danych	Opis właściwości
<code>import_system_variables</code>	Common None All	Określa, które zmienne systemowe będą importowane.
<code>import_codes_variables</code>	<i>flag</i>	
<code>import_sourcefile_variables</code>	<i>flag</i>	
<code>import_multi_response</code>	MultipleFlags Single	

Właściwości węzła `excelimportnode`



Węzeł Excel importuje dane z programu Microsoft Excel zapisane w pliku w formacie `.xlsx`. Źródło danych ODBC nie jest wymagane.

Przykłady

```
#Aby użyć zakresu nazwanego:
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)
```

```
#Aby jawnie określić zakres:
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")
```

Tabela 48. Właściwości węzła `excelimportnode`.

Właściwości węzła <code>excelimportnode</code>	Typ danych	Opis właściwości
<code>excel_file_type</code>	Excel2007	
<code>full_filename</code>	<i>string</i>	Pełna nazwa pliku wraz ze ścieżką.
<code>use_named_range</code>	<i>Boolean</i>	Określa, czy ma być używany węzeł nazwany. Wartość <code>true</code> powoduje, że zakres do odczytu określany jest na podstawie właściwości <code>named_range</code> , a pozostałe ustawienia dotyczące arkusza i zakresu danych są ignorowane.
<code>named_range</code>	<i>string</i>	
<code>worksheet_mode</code>	Index Name	Określa, czy arkusz jest wskazany przez numer, czy nazwę.
<code>worksheet_index</code>	<i>integer</i>	Numer arkusza do odczytania, liczony od 0 (0 to pierwszy arkusz, 1 to drugi arkusz itd.).

Tabela 48. Właściwości węzła `excelimportnode` (kontynuacja).

Właściwości węzła <code>excelimportnode</code>	Typ danych	Opis właściwości
<code>worksheet_name</code>	<i>string</i>	Nazwa arkusza do odczytania.
<code>data_range_mode</code>	FirstNonBlank ExplicitRange	Określa sposób ustalania zakresu.
<code>blank_rows</code>	StopReading ReturnBlankRows	Gdy <code>data_range_mode</code> ma wartość <i>FirstNonBlank</i> , określa sposób traktowania pustych wierszy.
<code>explicit_range_start</code>	<i>string</i>	Gdy <code>data_range_mode</code> ma wartość <i>ExplicitRange</i> , określa punkt początkowy odczytywanego zakresu.
<code>explicit_range_end</code>	<i>string</i>	
<code>read_field_names</code>	<i>Boolean</i>	Określa, czy pierwszy wiersz określonego zakresu zawiera nazwy zmiennych (kolumn).

Właściwości węzła `extensionimportnode`



Dzięki węzłowi importowania przez rozszerzenie można uruchomić skrypty R lub Python for Spark w celu importu danych.

Python for Spark — przykład

```
#### przykład skryptu w języku 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)

```

Przykład w języku R

```

#### przykład skryptu w języku R
node.setPropertyValue("syntax_type", "R")

R_script = """# 'JSON Import' Node v1.0 for IBM SPSS Modeler
# Pakiet 'RJSONIO' utworzony przez Duncan Temple Lang - http://cran.r-project.org/web/packages/RJSONIO
# Pakiet 'plyr' utworzony przez Hadley Wickham http://cran.r-project.org/web/packages/plyr
# Programista węzła: Danil Savine - IBM Extreme Blue 2014
# Opis: Węzeł ten umożliwia zaimportowanie do oprogramowania SPSS tabeli danych z oprogramowania JSON.
# Instalacja funkcji dla pakietów
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)
  }
}
# pakiety
packages(RJSONIO)
packages(plyr)
### Niniejsza funkcja służy do automatycznego generowania modelu danych
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
      #jeśli x jest czynnikiem, typeof zwróci liczbę całkowitą, co pozwala dodatkowo zająć się tym przypadkiem
      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)
}
# Z JSON do listy
txt <- readLines('C:/test.json')
formattedtxt <- paste(txt, collapse = '')
json.list <- fromJSON(formattedtxt)
# Stosowanie ścieżki do 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
    }
  }
}
# Z listy do dataframe za pośrednictwem unlisted json spoza listy
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
}
# Eksport to SPSS Modeler Data
modelerData <- filled
print(modelerData)
modelerDataModel <- getMetaData(modelerData)
print(modelerDataModel)

"""

node.setPropertyValue("r_syntax", R_script)

```

Tabela 49. Właściwości węzła *extensionimportnode*

Właściwości węzła <i>extensionimportnode</i>	Typ danych	Opis właściwości
<i>syntax_type</i>	<i>R</i> <i>Python</i>	Określa, który skrypt ma być wykonany – R lub Python (domyślnie R).
<i>r_syntax</i>	<i>string</i>	Komendy skryptu R do uruchomienia.
<i>python_syntax</i>	<i>string</i>	Komendy skryptu Python do uruchomienia.

Właściwości węzła *fixedfilenode*



Węzeł Plik kolumnowy importuje dane z plików tekstowych ze stałymi polami — czyli z plików, w których pola nie są separowane, ale rozpoczynają się w tym samym miejscu i mają stałą długość. Dane wygenerowane maszynowo lub pochodzące ze starszych wersji często są zapisywane w formacie ze stałymi polami.

Przykład

```

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)

```

Tabela 50. Właściwości węzła fixedfilenode.

Właściwości węzła fixedfilenode	Typ danych	Opis właściwości
record_len	<i>number</i>	Określa liczbę znaków w każdym rekordzie.
line_oriented	<i>flag</i>	Powoduje pomijanie znaku nowego wiersza na końcu każdego rekordu.
decimal_symbol	Default Comma Period	Typ separatora dziesiętnego, jaki jest używany w źródle danych.
skip_header	<i>number</i>	Określa liczbę wierszy do zignorowania na początku pierwszego rekordu. Opcja przydatna do pomijania nagłówek kolumn.
auto_recognize_datetime	<i>flag</i>	Określa, czy daty lub godziny mają być automatycznie rozpoznawane w danych źródłowych.
lines_to_scan	<i>number</i>	
fields	<i>list</i>	Właściwość ustrukturyzowana.
full_filename	<i>string</i>	Pełna nazwa pliku do odczytania, wraz ze ścieżką do katalogu.
strip_spaces	None Left Right Both	Powoduje odrzucanie spacji wiodących i końcowych z importowanych łańcuchów.
invalid_char_mode	Discard Replace	Powoduje usuwanie niepoprawnych znaków (null, 0 i dowolnych znaków nieistniejących w bieżącym systemie kodowania) z danych wejściowych lub zastępowanie niepoprawnych znaków określonym symbolem jednoznakowym.
invalid_char_replacement	<i>string</i>	
use_custom_values	<i>flag</i>	
custom_storage	Unknown String Integer Real Time Date Timestamp	

Tabela 50. Właściwości węzła *fixedfilenode* (kontynuacja).

Właściwości węzła <i>fixedfilenode</i>	Typ danych	Opis właściwości
<i>custom_date_format</i>	"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" MMM RRRR q Q RRRR ww WK YYYY	Ta właściwość ma zastosowanie tylko wtedy, gdy określono niestandardowy typ składowania.
<i>custom_time_format</i>	"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"	Ta właściwość ma zastosowanie tylko wtedy, gdy określono niestandardowy typ składowania.
<i>custom_decimal_symbol</i>	<i>field</i>	Ta właściwość ma zastosowanie tylko wtedy, gdy określono niestandardowy typ składowania.
<i>encoding</i>	StreamDefault SystemDefault "UTF-8"	Określa metodę kodowania tekstu.

Właściwości węzła gsdata_import



Węzeł źródłowy danych geoprzestrzennych umożliwia przeniesienie danych z mapy lub danych przestrzennych do sesji eksploracji danych.

Tabela 51. Właściwości węzła gsdata_import

Właściwości węzła gsdata_import	Typ danych	Opis właściwości
full_filename	string	Ścieżka do pliku .shp, który ma zostać załadowany.
map_service_URL	string	Adres URL usługi mapowej, z którą należy się połączyć.
map_name	string	Tylko jeśli używana jest wartość map_service_URL; zawiera strukturę folderów usługi mapowej na najwyższym poziomie.

Właściwości węzła sasimportnode



Węzeł importu SAS importuje dane SAS do programu IBM SPSS Modeler.

Przykład

```
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)
```

Tabela 52. Właściwości węzła sasimportnode.

Właściwości węzła sasimportnode	Typ danych	Opis właściwości
format	Windows UNIX Transport SAS7 SAS8 SAS9	Format pliku do zaimportowania.
full_filename	string	Pełna nazwa pliku wraz ze ścieżką.
member_name	string	Określa element, który ma być importowany z określonego pliku transportu SAS.
read_formats	flag	Odczytuje formaty danych (np. etykiety zmiennych) z określonego pliku formatu.
full_format_filename	string	
import_names	NamesAndLabels LabelsasNames	Określa metodę odwzorowywania nazw zmiennych i etykiet przy importowaniu.

Właściwości węzła simgennode



Węzeł Symulacje Generowanie zapewnia łatwy sposób na wygenerowanie danych objętych symulacją — od podstaw, korzystając z rozkładów statystycznych określonych przez użytkownika lub automatycznie, korzystając z rozkładów uzyskanych po uruchomieniu węzła Symulacje Dopasowanie dla istniejących danych historycznych. Jest to przydatne, kiedy ma zostać przeprowadzona ocena wyniku modelu predykcyjnego przy braku pewności dla danych wejściowych modelu.

Tabela 53. Właściwości węzła simgennode.

Właściwości węzła simgennode	Typ danych	Opis właściwości
fields	Właściwość ustrukturyzowana	Patrz przykład
korelacje	Właściwość ustrukturyzowana	Patrz przykład
keep_min_max_setting	<i>boolean</i>	
refit_correlations	<i>boolean</i>	
max_cases	<i>integer</i>	Wartość minimalna to 1000, wartość maksymalna to 2147483647
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>	Zwraca dane Xml parametru jako łańcuch.

Przykłady zmiennych

Jest to ustrukturyzowany parametr węzła o następującej składni:

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

distribution jest deklaracją nazwy rozkładu, po której następuje lista zawierająca pary nazw i wartości atrybutów. Każdy rozkład jest zdefiniowany w następujący sposób:

```
[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"}]}, "", ""]])
```

Na przykład, aby utworzyć węzeł generujący jedną zmienną o rozkładzie dwumianowym, można użyć następującego kodu:

```
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}]}], "", ""]])
```

Rozkład dwumianowy ma 2 parametry: *n* i *prob*. Ponieważ w rozkładzie dwumianowym nie stosuje się wartości minimalnych ani maksymalnych, są one przekazywane jako puste łańcuchy.

Uwaga: Właściwości *distribution* nie można nadać wartości bezpośrednio; używa się jej łącznie z właściwością *fields*.

Następujące przykłady ilustrują wszystkie możliwe typy rozkładu: Należy zwrócić uwagę, że próg jest wprowadzany jako *thresh* zarówno w *NegativeBinomialFailures*, jak i w *NegativeBinomialTrial*.

```

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
])

```

Przykład korelacji

Jest to ustrukturyzowany parametr węzła o następującej składni:

```

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

```

Correlation może być dowolną wartością spomiędzy +1 a -1. Można określić dowolną liczbę korelacji. Wszelkim nieokreślonym korelacjom jest przypisywane zero. Jeśli jakiegokolwiek zmienne są nieznanne, wartość korelacji powinna być określona w macierzy (tabeli) korelacji i jest wyświetlana na czerwono. Gdy istnieją nieznanne zmienne, nie jest możliwe wykonanie węzła.

Właściwości węzła statisticsimportnode



Węzeł Plik IBM SPSS Statistics odczytuje dane z pliku w formacie *.sav* używanym przez produkt IBM SPSS Statistics, a także z plików pamięci podręcznej zapisanych w programie IBM SPSS Modeler, których format jest identyczny.

Właściwości tego węzła są opisane w sekcji “Właściwości węzła statisticsimportnode” na stronie 329.

Właściwości węzła tm1odataimport



Węzeł źródłowy IBM Cognos TM1 importuje dane z baz danych Cognos TM1.

Tabela 54. Właściwości węzła tm1odataimport

Właściwości węzła tm1odataimport	Typ danych	Opis właściwości
admin_host	<i>string</i>	Adres URL dla nazwy hosta interfejsu API REST.
server_name	<i>string</i>	Nazwa serwera TM1 wybranego z admin_host.
credential_type	<i>inputCredential</i> lub <i>storedCredential</i>	Służy do wskazywania typu informacji autoryzacyjnych.
input_credential	<i>list</i>	Jeśli parametr credential_type ma wartość inputCredential, należy podać nazwę domeny, nazwę użytkownika i hasło.
stored_credential_name	<i>string</i>	Jeśli parametr credential_type ma wartość storedCredential, należy podać nazwę referencji na serwerze C&DS.
selected_view	<i>["field" "field"]</i>	Właściwość zawierająca listę informacji o wybranej kostce TM1 i nazwę widoku kostki, z którego dane mają być importowane do programu SPSS. Na przykład: <code>TM1_import.setPropertyValue("selected_view", ["plan_BudgetPlan", 'Goal Input'])</code>
is_private_view	<i>flag</i>	Określa, czy parametr selected_view to widok prywatny. Wartość domyślna to false.
selected_columns	<i>["field"]</i>	Określ wybraną kolumnę; tylko jeden element może być określony. Na przykład: <code>setPropertyValue("selected_columns", ["Measures"])</code>
selected_rows	<i>["field" "field"]</i>	Określ wybrane wiersze. Na przykład: <code>setPropertyValue("selected_rows", ["Dimension_1_1", "Dimension_2_1", "Dimension_3_1", "Periods"])</code>

Właściwości węzła tm1import (zdeprecjonowanego)



Węzeł źródłowy IBM Cognos TM1 importuje dane z baz danych Cognos TM1.

Uwaga: Ten węzeł został zdeprecjonowany w produkcie Modeler 18.0. Nazwa skryptu węzła zamiany to *tm1odataimport*.

Tabela 55. Właściwości węzła tm1import.

Właściwości węzła tm1import	Typ danych	Opis właściwości
pm_host	string	Uwaga: Tylko dla wersji 16.0 i 17.0 Nazwa hosta. Na przykład: TM1_import.setPropertyValue("pm_host", 'http://9.191.86.82:9510/pmhub/pm')
tm1_connection	["field", "field", ... , "field"]	Uwaga: Tylko dla wersji 16.0 i 17.0 Właściwość zawierająca listę informacji o połączeniu z serwerem TM1. Obowiązuje format: ["TM1_Server_Name", "tm1_username", "tm1_password"] Na przykład: TM1_import.setPropertyValue("tm1_connection", ['Planning Sample', "admin", "apple"])
selected_view	["field" "field"]	Właściwość zawierająca listę informacji o wybranej kostce TM1 i nazwę widoku kostki, z którego dane mają być importowane do programu SPSS. Na przykład: TM1_import.setPropertyValue("selected_view", ['plan_BudgetPlan', 'Goal Input'])
selected_column	["field"]	Określ wybraną kolumnę; tylko jeden element może być określony. Na przykład: setPropertyValue("selected_columns", ["Measures"])
selected_rows	["field" "field"]	Określ wybrane wiersze. Na przykład: setPropertyValue("selected_rows", ["Dimension_1_1", "Dimension_2_1", "Dimension_3_1", "Periods"])

Właściwości węzła twcimport



Węzeł źródłowy TWC importuje dane pogodowe z przedsiębiorstwa The Weather Company należącego do IBM. Można je wykorzystać w celu uzyskania historycznych lub prognozowanych danych pogodowych dla swojej lokalizacji. Umożliwia to tworzenie rozwiązań biznesowych opartych na danych meteorologicznych, które pozwalają podejmować lepsze decyzje dzięki najdokładniejszym i najtrafniejszym dostępnym informacjom pogodowym.

Tabela 56. Właściwości węzła twcimport

Właściwości węzła twcimport	Typ danych	Opis właściwości
TWCDatImport.latitude	Real	Określa szerokość geograficzną w formacie [-90.0Δ90.0]
TWCDatImport.longitude	Real	Określa długość geograficzną w formacie [-180.0Δ180.0].
TWCDatImport.licenseKey	string	Określa klucz licencyjny uzyskany od przedsiębiorstwa The Weather Company.
TWCDatImport.measurementUnit	English Metric Hybrid	Określa jednostkę miary. Możliwe wartości to: English, Metric lub Hybrid. Wartość Metric jest wartością domyślną.

Tabela 56. Właściwości węzła `twcimport` (kontynuacja)

Właściwości węzła <code>twcimport</code>	Typ danych	Opis właściwości
<code>TWCDatImport.dataType</code>	Historical Forecast	Określa typ wejściowych danych pogodowych. Możliwe wartości to <code>Historical</code> lub <code>Forecast</code> . Wartość <code>Historical</code> jest wartością domyślną.
<code>TWCDatImport.startDate</code>	<i>Integer</i>	Jeśli wartość <code>Historical</code> zostanie określona dla właściwości <code>TWCDatImport.dataType</code> , należy określić datę rozpoczęcia w formacie <code>rrrrMMdd</code> .
<code>TWCDatImport.endDate</code>	<i>Integer</i>	Jeśli wartość <code>Historical</code> zostanie określona dla właściwości <code>TWCDatImport.dataType</code> , należy określić datę końcową w formacie <code>rrrrMMdd</code> .
<code>TWCDatImport.forecastHour</code>	6 12 24 48	Jeśli wartość <code>Forecast</code> zostanie określona dla właściwości <code>TWCDatImport.dataType</code> , należy podać wartość 6, 12, 24 lub 48 dla godziny.

Właściwości węzła `userinputnode`



Węzeł Dane niestandardowe umożliwia łatwe wygenerowanie danych sztucznych — albo od początku, albo poprzez modyfikację istniejących danych. Jest to przydatne na przykład do tworzenia testowego zbioru danych do modelowania.

Przykład

```
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")
```

Tabela 57. Właściwości węzła `userinputnode`.

Właściwości węzła <code>userinputnode</code>	Typ danych	Opis właściwości
<code>data</code>		
<code>names</code>		Parametr ustrukturyzowany, który ustawia lub zwraca listę nazw zmiennych wygenerowanych przez węzeł.
<code>custom_storage</code>	Unknown String Integer Real Time Date Timestamp	Parametr kluczowany, który ustawia lub zwraca typ składowania zmiennej.

Tabela 57. Właściwości węzła `userinputnode` (kontynuacja).

Właściwości węzła <code>userinputnode</code>	Typ danych	Opis właściwości
<code>data_mode</code>	Combined Ordered	W przypadku określenia ustawienia <code>Combined</code> generowane są rekordy dla każdej kombinacji wartości ze zbioru i wartości minimalnej/ maksymalnej. Liczba wygenerowanych rekordów jest równa iloczynowi liczby wartości w każdej zmiennej. W przypadku określenia ustawienia <code>Ordered</code> z każdej kolumny dla każdego rekordu wybierana jest jedna wartość w celu wygenerowania wiersza danych. Liczba wygenerowanych rekordów jest równa największej liczbie wartości powiązanych ze zmienną. Wszelkie zmienne z mniejszą liczbą wartości danych muszą być uzupełnione wartościami <code>null</code> .
<code>values</code>		Uwaga: Ta właściwość jest nieaktualna i została zastąpiona przez właściwość <code>userinputnode.data</code> ; nie należy jej już używać.

Właściwości węzła `variablefilenode`



Węzeł Plik zmienny odczytuje dane z plików tekstowych o formacie swobodnym, tj. plików, których rekordy zawierają stałą liczbę zmiennych ale o różnej liczbie znaków. Węzeł tego typu jest również przydatny w przypadku zmiennych z tekstem nagłówka o ustalonej długości i niektórych typów adnotacji.

Przykład

```
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])
```

Tabela 58. Właściwości węzła `variablefilenode`.

Właściwości węzła <code>variablefilenode</code>	Typ danych	Opis właściwości
<code>skip_header</code>	<i>number</i>	Określa liczbę znaków do zignorowania na początku pierwszego rekordu.
<code>num_fields_auto</code>	<i>flag</i>	Powoduje automatyczne wyznaczenie liczby zmiennych w każdym rekordzie. Rekordy muszą być zakończone znakami nowego wiersza.
<code>num_fields</code>	<i>number</i>	Oznacza, że liczba zmiennych w każdym rekordzie będzie określona ręcznie.
<code>delimit_space</code>	<i>flag</i>	Określa znak separujący zmienne w pliku.
<code>delimit_tab</code>	<i>flag</i>	

Tabela 58. Właściwości węzła *variablefilenode* (kontynuacja).

Właściwości węzła <i>variablefilenode</i>	Typ danych	Opis właściwości
<i>delimit_new_line</i>	<i>flag</i>	
<i>delimit_non_printing</i>	<i>flag</i>	
<i>delimit_comma</i>	<i>flag</i>	Jeśli przecinek jest zarówno separatorem zmiennych i separatorem dziesiętnym w strumieniach, należy ustawić <i>delimit_other</i> na <i>true</i> i określić przecinek jako separator za pośrednictwem właściwości <i>other</i> .
<i>delimit_other</i>	<i>flag</i>	Umożliwia określenie niestandardowego separatora za pośrednictwem właściwości <i>other</i> .
<i>other</i>	<i>string</i>	Określa separator używany, gdy <i>delimit_other</i> ma wartość <i>true</i> .
<i>decimal_symbol</i>	Default Comma Period	Określa separator dziesiętny używany w źródle danych.
<i>multi_blank</i>	<i>flag</i>	Powoduje, że wiele sąsiednich separatorów traktowanych jest jako jeden separator.
<i>read_field_names</i>	<i>flag</i>	Powoduje, że pierwszy wiersz pliku danych traktowany jest jako zawierający etykiety kolumn.
<i>strip_spaces</i>	None Left Right Both	Powoduje odrzucanie spacji wiodących i końcowych z importowanych łańcuchów.
<i>invalid_char_mode</i>	Discard Replace	Powoduje usuwanie niepoprawnych znaków (null, 0 i dowolnych znaków nieistniejących w bieżącym systemie kodowania) z danych wejściowych lub zastępowanie niepoprawnych znaków określonym symbolem jednoznakowym.
<i>invalid_char_replacement</i>	<i>string</i>	
<i>break_case_by_newline</i>	<i>flag</i>	Powoduje, że separatorem wierszy jest znak nowego wiersza.
<i>lines_to_scan</i>	<i>number</i>	Określa, ile wierszy przeszukiwać w poszukiwaniu określonych typów danych.
<i>auto_recognize_datetime</i>	<i>flag</i>	Określa, czy daty lub godziny mają być automatycznie rozpoznawane w danych źródłowych.
<i>quotes_1</i>	Discard PairAndDiscard IncludeAsText	Określa, w jaki sposób pojedyncze cudzysłowy mają być traktowane przy imporcie.
<i>quotes_2</i>	Discard PairAndDiscard IncludeAsText	Określa, w jaki sposób podwójne cudzysłowy mają być traktowane przy imporcie.
<i>full_filename</i>	<i>string</i>	Pełna nazwa pliku do odczytania, wraz ze ścieżką do katalogu.
<i>use_custom_values</i>	<i>flag</i>	
<i>custom_storage</i>	Unknown String Integer Real Time Date Timestamp	

Tabela 58. Właściwości węzła *variablefilenode* (kontynuacja).

Właściwości węzła <i>variablefilenode</i>	Typ danych	Opis właściwości
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" MMM RRRR q Q RRRR ww WK YYYY	Ta właściwość ma zastosowanie tylko wtedy, gdy określono niestandardowy typ składowania.
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"	Ta właściwość ma zastosowanie tylko wtedy, gdy określono niestandardowy typ składowania.
custom_decimal_symbol	<i>field</i>	Ta właściwość ma zastosowanie tylko wtedy, gdy określono niestandardowy typ składowania.
encoding	StreamDefault SystemDefault "UTF-8"	Określa metodę kodowania tekstu.

Właściwości węzła xmlimportnode



Węzeł źródłowy XML importuje do strumienia dane w formacie XML. Można zaimportować jeden plik lub wszystkie pliki w katalogu. Opcjonalnie można określić plik schematu, z którego będzie odczytywana struktura danych XML.

Przykład

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

Tabela 59. Właściwości węzła xmlimportnode.

Właściwości węzła xmlimportnode	Typ danych	Opis właściwości
read	single directory	Odczytuje jeden plik danych (domyślnie) lub wszystkie pliki XML z katalogu.
recurse	flag	Określa, czy dodatkowo mają być odczytywane pliki XML ze wszystkich podkatalogów określonego katalogu.
full_filename	string	(wymagana) Pełna ścieżka i nazwa pliku XML, który ma zostać zaimportowany (jeśli read = single).
directory_name	string	(wymagana) Pełna ścieżka i nazwa katalogu, z którego mają być importowane pliki XML (jeśli read = directory).
full_schema_filename	string	Pełna ścieżka i nazwa pliku XSD lub DTD, z którego ma być odczytywana struktura XML. Pominięcie tego parametru spowoduje, że struktura będzie odczytywana z pliku źródłowego XML.
records	string	Wyrażenie XPath (np. /author/name) definiujące granicę rekordu. Za każdym razem, gdy w pliku źródłowym zostanie napotkany ten element, tworzony jest nowy rekord.
mode	read specify	Powoduje odczytanie wszystkich danych (domyślnie) lub umożliwia określenie danych do odczytania.
fields		Lista pozycji (elementów i atrybutów) do zaimportowania. Każda pozycja na liście jest wyrażeniem XPath.

Właściwości węzła dataviewimport



Węzeł Dane importuje dane z obiektu Data View do programu IBM SPSS Modeler.

Przykład

```
stream = modeler.script.stream()
dvnnode = stream.createAt("dataviewimport", "Data View", 96, 96)
```

```

dvnnode.setPropertyValue("analytic_data_source",
["", "/folder/adv", "LATEST"])
dvnnode.setPropertyValue("table_name", ["", "com.ibm.spss.Table"])
dvnnode.setPropertyValue("data_access_plan",
["", "DataAccessPlan"])
dvnnode.setPropertyValue("optional_attributes",
[["", "NewDerivedAttribute"]])
dvnnode.setPropertyValue("include_xml", True)
dvnnode.setPropertyValue("include_xml_field", "xml_data")

```

Tabela 60. właściwości węzła `dataviewimport`

Właściwości węzła <code>dataviewimport</code>	Typ danych	Opis właściwości
<code>analytic_data_source</code>	<i>string</i>	Obiekt Analytic Data View przechowywany w programie IBM SPSS Collaboration and Deployment Services. Nazwa ścieżki i etykieta wersji, która ma być używana. ["Object ID", "Full path", "Version"]
<code>table_name</code>	<i>string</i>	Tabela widoku danych używana w obiekcie Analytic Data View. Nazwa tabeli musi być kwalifikowana nazwą pakietu. Pakiet można uzyskać, eksportując dane BOM z klienta IBM SPSS Collaboration and Deployment Services Deployment Manager i analizując zawartość pliku <code>default.bom</code> w wyeksportowanym archiwum ZIP. Nazwa pakietu powinna być zawsze taka sama, chyba że dane BOM wyeksportowano z pliku IBM Operational Decision Management (iLOG). ["Object ID", "Name"]
<code>data_access_plan</code>	<i>string</i>	Plan dostępu do danych dostarczający dane dla obiektu Analytic Data View. ["Object ID", "Name"]
<code>optional_attributes</code>	<i>string</i>	Lista atrybutów wywiedzionych, które mają być uwzględnione. [["ID1", "Name1"], ["ID2", "Name2"]]
<code>include_xml</code>	<i>boolean</i>	True, jeśli ma być uwzględnione pole z danymi instancji XOM. Zalecane jest ustawienie <code>false</code> , chyba że używane są węzły programu IBM Analytical Decision Management iLOG. Włączenie tej opcji może spowodować istotne zwiększenie narzutu przetwarzania.
<code>include_xml_field</code>	<i>string</i>	Nazwa zmiennej, która ma zostać dodana, gdy właściwość <code>include_xml</code> jest ustawiona na <code>true</code> .

Rozdział 10. Właściwości węzłów związanych z operacjami na rekordach

Właściwości węzła appendnode



Węzeł Dołączanie łączy zestawy rekordów. Jest to przydatne do łączenia zbiorów danych z podobnymi strukturami zawierającymi inne dane.

Przykład

```
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")
```

Tabela 61. Właściwości węzła appendnode.

Właściwości węzła appendnode	Typ danych	Opis właściwości
match_by	Position Name	Można dołączać zbiory danych na podstawie pozycji zmiennych w głównym źródle danych lub nazw zmiennych w wejściowych zbiorach danych.
match_case	<i>flag</i>	Powoduje rozróżnianie wielkości liter przy dopasowywaniu nazw zmiennych.
include_fields_from	Main All	
create_tag_field	<i>flag</i>	
tag_field_name	<i>string</i>	

Właściwości węzła aggregatenode



Węzeł agregacji zastępuje sekwencję rekordów wejściowych zsumowanymi, zagregowanymi rekordami wyjściowymi.

Przykład

```
node = stream.create("aggregate", "My node")
# dbnode to skonfigurowany węzeł importu z bazy danych
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")
```

Tabela 62. Właściwości węzła *aggregatenode*.

Właściwości węzła <i>aggregatenode</i>	Typ danych	Opis właściwości
keys	<i>list</i>	Lista zmiennych, które mogą być używane jako klucze dla agregacji. Na przykład, jeśli Sex (Płeć) i Region są zmiennymi kluczowymi, dla każdej unikatowej kombinacji zmiennych M (Mężczyzna) i F (Kobieta) z regionami N (Północny) i S (Południowy) — cztery unikatowe kombinacje — utworzony zostanie zagregowany rekord.
contiguous	<i>flag</i>	Należy wybrać tę opcję, jeśli wiadomo, że wszystkie rekordy zawierające takie same wartości kluczowe są w wartościach wejściowych pogrupowane (na przykład, jeśli wartości wejściowe są posortowane według zmiennych kluczowych). Dzięki temu można zwiększyć wydajność.
aggregates		Właściwość ustrukturyzowana zawierająca listę zmiennych liczbowych, których wartości zostaną poddane agregacji, oraz wybrane tryby agregacji.
aggregate_exprs		Właściwość wprowadzana, która wiąże wywiedzioną nazwę zmiennej z wyrażeniem agregującym, które tę zmienną oblicza. Na przykład: <code>aggregatenode.setKeyedPropertyValue("aggregate_exprs", "Na_MAX", "MAX('Na')")</code>
extension	<i>string</i>	Określa przedrostek lub przyrostek zduplikowanych zmiennych zagregowanych (przykład poniżej).
add_as	Suffix Prefix	
inc_record_count	<i>flag</i>	Tworzy dodatkową zmienną, która określa, ile rekordów wejściowych zostało zagregowanych w każdym z rekordów zagregowanych.
count_field	<i>string</i>	Określa nazwę zmiennej liczebności rekordów.
allow_approximation	<i>Boolean</i>	Dopuszcza przybliżenie statystyk porządkowych, gdy Analytic Server przeprowadza agregację
bin_count	<i>integer</i>	Określa liczbę przedziałów, jaka ma być używana w aproksymacji.

Właściwości węzła *balancenode*



Węzeł Zrównoważenie poprawia dysproporcje w zbiorze danych, tak aby spełniał określone warunki. Dyrektywa równoważenia koryguje proporcje rekordów, w których warunek został spełniony, na podstawie określonego czynnika.

Przykład

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

Tabela 63. Właściwości węzła `balancenode`.

Właściwości węzła <code>balancenode</code>	Typ danych	Opis właściwości
<code>directives</code>		Właściwość ustrukturyzowana służąca do równoważenia proporcji wartości zmiennej na podstawie określonej liczby (przykład poniżej)
<code>training_data_only</code>	<i>flag</i>	Określa, że równoważone mają być wyłącznie dane uczące. Jeśli w strumieniu nie określono zmiennej dzielącej na podzbiory, wówczas ta opcja jest ignorowana.

Ta właściwość węzła ma format:

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

Uwaga: Jeśli wyrażenie zawiera łańcuchy (ujęte w cudzysłowy), to cudzysłowy muszą być poprzedzone znakiem zmiany znaczenia `\"`. Znak `\"` oznacza także kontynuację wiersza i można go zastosować, aby rozmieścić argumenty w czytelny sposób.

Właściwości węzła `derive_stbnode`



Węzeł Siatka czasoprzestrzeni wylicza siatkę czasoprzestrzeni na podstawie zmiennych długości i szerokości geograficznej oraz znacznika czasu. Można też zidentyfikować często występujące elementy siatki czasoprzestrzeni jako `hangouts`.

Przykład

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

```
# Tryb poszczególnych rekordów
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_")
```

```
# Tryb hangouts
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)
```

Tabela 64. Właściwości węzła Siatka czasoprzestrzeni

Właściwości węzła <code>derive_stbnode</code>	Typ danych	Opis właściwości
<code>mode</code>	IndividualRecords Hangouts	
<code>latitude_field</code>	<i>field</i>	
<code>longitude_field</code>	<i>field</i>	
<code>timestamp_field</code>	<i>field</i>	

Tabela 64. Właściwości węzła Siatka czasoprzestrzeni (kontynuacja)

Właściwości węzła derive_stbnode	Typ danych	Opis właściwości
hangout_density	<i>density</i>	Jedna gęstość. Poprawne wartości gęstości omówiono w opisie właściwości <i>densities</i> .
densities	[<i>density,density,...., density</i>]	Każda gęstość jest łańcuchem, na przykład STB_GH8_1DAY. Uwaga: Obowiązują limity, do których gęstości są poprawne. Jako łańcucha mieszającego geoprzestrzennego można używać wartości od GH1 do GH15. Jako części czasowej można używać wartości: EVER 1YEAR 1MONTH 1DAY 12HOURS 8HOURS 6HOURS 4HOURS 3HOURS 2HOURS 1HOUR 30MINS 15MINS 10MINS 5MINS 2MINS 1MIN 30SECS 15SECS 10SECS 5SECS 2SECS 1SEC
id_field	<i>field</i>	
qualifying_duration	1DAY 12HOURS 8HOURS 6HOURS 4HOURS 3HOURS 2Hours 1HOUR 30MIN 15MIN 10MIN 5MIN 2MIN 1MIN 30SECS 15SECS 10SECS 5SECS 2SECS 1SECS	Wartość musi być łańcuchem.
min_events	<i>integer</i>	Minimalna dozwolona wartość całkowita to 2.
qualifying_pct	<i>integer</i>	Musi być liczbą całkowitą z przedziału od 1 do 100.
add_extension_as	Prefix Suffix	
name_extension	<i>string</i>	

Właściwości węzła distinctnode



Węzeł Powtórzenia usuwa zduplikowane rekordy, przekazując pierwszy odmienny rekord do strumienia danych lub odrzucając pierwszy rekord i przekazując do strumienia danych wszystkie duplikaty.

Przykład

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

Tabela 65. Właściwości węzła distinctnode.

Właściwości węzła distinctnode	Typ danych	Opis właściwości
mode	Include Discard	Można uwzględnić w strumieniu danych pierwszy rekord unikalny lub odrzucić pierwszy rekord unikalny i przekazywać do strumienia danych wszelkie rekordy zduplikowane.
grouping_fields	<i>list</i>	Lista zmiennych, na podstawie których określana jest identyczność rekordów. Uwaga: Począwszy od wersji 16 programu IBM SPSS Modeler ta właściwość jest nieaktualna.
composite_value	Właściwość ustrukturyzowana.	Patrz przykład poniżej.
composite_values	Właściwość ustrukturyzowana.	Patrz przykład poniżej.
inc_record_count	<i>flag</i>	Tworzy dodatkową zmienną, która określa, ile rekordów wejściowych zostało zagregowanych w każdym z rekordów zagregowanych.
count_field	<i>string</i>	Określa nazwę zmiennej liczebności rekordów.
sort_keys	Właściwość ustrukturyzowana.	Uwaga: Począwszy od wersji 16 programu IBM SPSS Modeler ta właściwość jest nieaktualna.
default_ascending	<i>flag</i>	
low_distinct_key_count	<i>flag</i>	Określa, że istnieje tylko mała liczba rekordów i/lub mała liczba unikalnych wartości zmiennych kluczowych.
keys_pre_sorted	<i>flag</i>	Określa, że wszystkie rekordy o tych samych wartościach kluczowych mają być grupowane w wynikach.
disable_sql_generation	<i>flag</i>	

Przykład dla właściwości composite_value

Właściwość composite_value ma następującą postać ogólną:

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

FILLOPTION ma postać [FillType, Option1, Option2, ...].

Przykłady:

```

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"])

```

Opcje niestandardowe wymagają więcej niż jednego argumentu. Są dodawane w postaci listy, na przykład:

```

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"])

```

Przykład dla właściwości composite_values

Właściwość composite_values ma następującą postać ogólną:

```

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

```

Przykład:

```

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

```

Właściwości węzła extensionprocessnode



Dzięki węzłowi transformacji przez rozszerzenie można pobrać dane ze strumienia, a następnie zastosować do przekształcenia danych za pomocą skryptów R lub Python for Spark.

Python for Spark — przykład

```

#### przykład skryptu w języku 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)

```

Przykład w języku R

```

#### przykład skryptu w języku 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)""")

```

Tabela 66. Właściwości węzła *extensionprocessnode*

Właściwości węzła extensionprocessnode	Typ danych	Opis właściwości
syntax_type	<i>R</i> <i>Python</i>	Określa, który skrypt ma być wykonany – R lub Python (domyślnie R).
r_syntax	<i>string</i>	Komendy skryptu R do uruchomienia.
python_syntax	<i>string</i>	Komendy skryptu Python do uruchomienia.
use_batch_size	<i>flag</i>	Włącza przetwarzanie wsadowe.
batch_size	<i>integer</i>	Określa liczbę rekordów danych, które mają być uwzględnione w każdej partii.
convert_flags	StringsAndDoubles LogicalValues	Opcja przekształcania zmiennych typu flaga.
convert_missing	<i>flag</i>	Opcja przekształcania braków danych na wartości R NA.
convert_datetime	<i>flag</i>	Opcja przekształcania zmiennych o formacie daty lub daty/czasu na formaty daty/czasu języka R.
convert_datetime_class	POSIXct POSIXlt	Opcje określające format docelowy, na który przekształcane są zmienne w formacie daty i daty/czasu.

Właściwości węzła mergenode



Węzeł Łączenie na podstawie wielu rekordów wejściowych tworzy pojedynczy rekord wyjściowy zawierający niektóre lub wszystkie zmienne wejściowe. Jest przydatny podczas scalania danych z różnych źródeł, takich jak dane wewnętrzne klienta oraz dane demograficzne osób, które dokonały zakupu.

Przykład

```
node = stream.create("merge", "My node")
# zakładamy, że customerdata i salesdata są skonfigurowanymi węzłami importu z bazy danych
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"]])
```

Tabela 67. Właściwości węzła mergenode.

Właściwości węzła mergenode	Typ danych	Opis właściwości
method	Order Keys Condition Rankedcondition	Określa, czy rekordy mają być łączone w kolejności, w jakiej są wymienione w plikach danych, czy do łączenia rekordów o tych samych wartościach zmiennych kluczowych będzie używana jedna, czy więcej zmiennych źródłowych, czy rekordy będą łączone, jeśli spełniony będzie określony warunek, czy też łączona ma być każda para wierszy w podstawowym i wszystkich dodatkowych zbiorach danych; przy czym wyrażenie rangujące używane jest do sortowania wielokrotnych dopasowań w kolejności rosnącej.
condition	<i>string</i>	Jeśli method ma wartość Condition, określa warunek uwzględniania lub odrzucania rekordów.
key_fields	<i>list</i>	
common_keys	<i>flag</i>	
join	Inner FullOuter PartialOuter Anti	
outer_join_tag.n	<i>flag</i>	W tej właściwości <i>n</i> jest nazwą znacznika widoczną w oknie dialogowym Wybierz zbiór danych. Można określić więcej niż jedną nazwę znacznika, ponieważ niekompletne rekordy mogą być składane z wielu zbiorów danych.
single_large_input	<i>flag</i>	Określa, czy stosowana będzie optymalizacja prowadząca do tego, aby jeden wejściowy zestaw danych był stosunkowo duży w porównaniu z pozostałymi.

Tabela 67. Właściwości węzła *mergenode* (kontynuacja).

Właściwości węzła <i>mergenode</i>	Typ danych	Opis właściwości
<code>single_large_input_tag</code>	<i>string</i>	Określa nazwę znacznika widoczną w oknie dialogowym Wybierz zbiór danych. Należy zauważyć, że sposób korzystania z tej właściwości różni się nieco od sposobu korzystania z właściwości <code>outer_join_tag</code> (flaga wobec łańcucha), ponieważ można określić tylko jeden wejściowy zestaw danych.
<code>use_existing_sort_keys</code>	<i>flag</i>	Określa, czy dane wejściowe są już posortowane według co najmniej jednej zmiennej kluczowej.
<code>existing_sort_keys</code>	[[<i>'string'</i> , 'Ascending'] \ [<i>'string'</i> , 'Descending']]	Określa zmienne, które są już posortowane i kierunek ich sortowania.
<code>primary_dataset</code>	<i>string</i>	Jeśli <code>method</code> ma wartość <code>Rankedcondition</code> , wybiera główny zestaw danych do łączenia. Można to uznać za lewą stronę złączenia zewnętrznego.
<code>rename_duplicate_fields</code>	<i>Boolean</i>	Jeśli <code>method</code> ma wartość <code>Rankedcondition</code> , a ta właściwość ma wartość <code>Y</code> , to jeśli wynikowy połączony zestaw danych zawiera wiele zmiennych o tej samej nazwie pochodzących z różnych źródeł danych, to na początku nagłówków kolumn zmiennych dodane zostaną odpowiednie znaczniki ze źródeł danych.
<code>merge_condition</code>	<i>string</i>	
<code>ranking_expression</code>	<i>string</i>	
<code>Num_matches</code>	<i>integer</i>	Liczba dopasowań, jakie są zwracane, na podstawie właściwości <code>merge_condition</code> i <code>ranking_expression</code> . Minimum 1, maksimum 100.

Właściwości węzła *rfmaggregatenode*



Węzeł Agregacja RFM umożliwia analizowanie danych historycznych o transakcjach klientów, usunięcie z nich danych nieużywanych i połączenie pozostałych danych transakcyjnych w jeden wiersz opisujący ostatnią transakcję z danym klientem, liczbę transakcji z tym klientem oraz łączną kwotę tych transakcji.

Przykład

```
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")
```

Tabela 68. Właściwości węzła *rfmaggregatenode*.

Właściwości węzła <i>rfmaggregatenode</i>	Typ danych	Opis właściwości
<code>relative_to</code>	Fixed Today	Należy określić datę, od której obliczana będzie aktualność transakcji.

Tabela 68. Właściwości węzła *rfmaggregatenode* (kontynuacja).

Właściwości węzła rfmaggregatenode	Typ danych	Opis właściwości
reference_date	<i>date</i>	Dostępna tylko wtedy, gdy właściwość <i>relative_to</i> ma wartość <i>Fixed</i> .
contiguous	<i>flag</i>	Jeśli dane zostały wstępnie posortowane tak, że wszystkie rekordy o tym samym identyfikatorze są zgrupowane w strumieniu danych, należy wybrać tę opcję w celu przyspieszenia przetwarzania.
id_field	<i>field</i>	Określa zmienną, która ma identyfikować klientów i ich transakcje.
date_field	<i>field</i>	Określa zmienną daty, na podstawie której ma być obliczana aktualność.
value_field	<i>field</i>	Określa zmienną, która ma być używana do obliczania kwoty.
extension	<i>string</i>	Określa przedrostek lub przyrostek zduplikowanych zmiennych zagregowanych.
add_as	Suffix Prefix	Określa, czy rozszerzenie (<i>extension</i>) ma być dodawane jako przyrostek, czy jako przedrostek.
discard_low_value_records	<i>flag</i>	Umożliwia korzystanie z ustawienia <i>discard_records_below</i> .
discard_records_below	<i>number</i>	Określa minimalną wartość, poniżej której dane transakcji nie będą uwzględniane podczas obliczania sum RFM. Jednostki tej wartości odnoszą się do wybranej zmiennej <i>value</i> .
only_recent_transactions	<i>flag</i>	Umożliwia korzystanie z ustawienia <i>specify_transaction_date</i> lub <i>transaction_within_last</i> .
specify_transaction_date	<i>flag</i>	
transaction_date_after	<i>date</i>	Dostępna tylko wtedy, gdy wybrane jest ustawienie <i>specify_transaction_date</i> . Należy określić datę transakcji, po której rekordy będą uwzględniane w analizie.
transaction_within_last	<i>number</i>	Dostępna tylko wtedy, gdy wybrane jest ustawienie <i>transaction_within_last</i> . Określa liczbę i typ okresów (dni, tygodnie, miesiące lub lata) wstecz od daty Oblicz okres od, dla których rekordy będą uwzględniane w analizie.
transaction_scale	Days Weeks Months Years	Dostępna tylko wtedy, gdy wybrane jest ustawienie <i>transaction_within_last</i> . Określa liczbę i typ okresów (dni, tygodnie, miesiące lub lata) wstecz od daty Oblicz okres od, dla których rekordy będą uwzględniane w analizie.
save_r2	<i>flag</i>	Powoduje uwidocznienie daty drugiej co do aktualności transakcji dla każdego klienta.
save_r3	<i>flag</i>	Dostępna tylko wtedy, gdy wybrane jest ustawienie <i>save_r2</i> . Powoduje uwidocznienie daty trzeciej co do aktualności transakcji dla każdego klienta.

Właściwości węzła Rprocessnode



Węzeł przekształceń R umożliwia pobieranie danych ze strumienia programu IBM(r) SPSS(r) Modeler i modyfikowanie ich przy użyciu skryptu w języku R udostępnionego przez użytkownika. Po zmodyfikowaniu dane są zwracane do strumienia.

Przykład

```
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")
```

Tabela 69. Właściwości węzła Rprocessnode.

Właściwości węzła Rprocessnode	Typ danych	Opis właściwości
syntax	<i>string</i>	
convert_flags	StringsAndDoubles LogicalValues	
convert_datetime	<i>flag</i>	
convert_datetime_class	POSIXct POSIXlt	
convert_missing	<i>flag</i>	
use_batch_size	<i>flag</i>	Włącza przetwarzanie wsadowe.
batch_size	<i>integer</i>	Określa liczbę rekordów danych w każdej partii do przetworzenia wsadowego

Właściwości węzła samplenode



Węzeł Próba wybiera podzbiór rekordów. Obsługiwanych jest wiele typów próby, w tym próby: warstwowa, zgrupowana i nielosowa (strukturalna). Próbkowanie może być przydatne do zwiększenia wydajności oraz podczas wyboru grup powiązanych rekordów lub transakcji do analizy.

Przykład

```
/* Utwórz dwa węzły próby w celu wyodrębnienia
różnych prób z tych samych danych */

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]])

```

Tabela 70. Właściwości węzła *samplenode*.

Właściwości węzła <i>samplenode</i>	Typ danych	Opis właściwości
method	Proste Złożone	
mode	Include Discard	Powoduje uwzględnianie lub odrzucanie rekordów spełniających określony warunek.
sample_type	First OneInN RandomPct	Określa metodę doboru próby.
first_n	<i>integer</i>	Rekordy do określonego punktu odcięcia będą uwzględniane lub odrzucane.
one_in_n	<i>number</i>	Powoduje uwzględnianie lub odrzucanie co <i>n</i> -tego rekordu.
rand_pct	<i>number</i>	Określa odsetek rekordów do uwzględnienia lub odrzucenia.
use_max_size	<i>flag</i>	Umożliwia korzystanie z ustawienia <i>maximum_size</i> .
maximum_size	<i>integer</i>	Określa największą próbę, jaka ma być uwzględniona lub odrzucona ze strumienia danych. Ta opcja jest nadmiarowa, a tym samym wyłączona, gdy określone są wartości <i>First</i> i <i>Include</i> .
set_random_seed	<i>flag</i>	Umożliwia korzystanie z wartości początkowej generatora liczb losowych.
random_seed	<i>integer</i>	Określa wartość używaną jako wartość początkowa generatora liczb losowych.
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>	Ta opcja ma zastosowanie tylko wtedy, gdy pobierana jest próba złożona (Complex) z ustawieniem <i>Sample units=Proportions</i> .
use_max_stratum_size	<i>flag</i>	

Tabela 70. Właściwości węzła *samplenode* (kontynuacja).

Właściwości węzła <i>samplenode</i>	Typ danych	Opis właściwości
maximum_stratum_size	<i>integer</i>	Ta opcja ma zastosowanie tylko wtedy, gdy pobierana jest próba złożona (Complex) z ustawieniem <code>Sample units=Proportions</code> .
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][string string value]...]</i>	Jeśli <code>sample_units=proportions</code> i <code>sample_size_proportions=Custom</code> , określa wartość każdej możliwej kombinacji wartości zmiennych specyfikacji.
default_proportion	<i>number</i>	
sizes_counts	<i>[[string string value][string string value]...]</i>	Określa wartość każdej możliwej kombinacji wartości zmiennych specyfikacji. Właściwości tej używa się podobnie, jak <code>sizes_proportions</code> , ale podaje się w niej liczbę całkowitą, a nie proporcję.
default_count	<i>number</i>	

Właściwości węzła *selectnode*



Węzeł Selekcja wybiera lub odrzuca podzbiór rekordów ze strumienia danych na podstawie określonego warunku. Na przykład, można wybrać rekordy należące do konkretnego regionu sprzedaży.

Przykład

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

Tabela 71. Właściwości węzła *selectnode*.

Właściwości węzła <i>selectnode</i>	Typ danych	Opis właściwości
mode	Include Discard	Określa, czy wybrane rekordy mają być uwzględniane, czy odrzucane.
condition	<i>string</i>	Warunek uwzględniania lub odrzucania rekordów.

Właściwości węzła *sortnode*



Węzeł Sortowanie sortuje rekordy w kolejności rosnącej lub malejącej na podstawie wartości jednej lub większej liczby zmiennych.

Przykład

```

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"]])

```

Tabela 72. Właściwości węzła *sortnode*.

Właściwości węzła <i>sortnode</i>	Typ danych	Opis właściwości
keys	<i>list</i>	Określa zmienne, według których ma się odbywać sortowanie. Jeśli nie zostanie określony kierunek, to obowiązuje kierunek domyślny.
default_ascending	<i>flag</i>	Określa domyślny kierunek sortowania.
use_existing_keys	<i>flag</i>	Określa, czy sortowanie ma być optymalizowane przy użyciu poprzedniej kolejności sortowania w przypadku zmiennych, które są już posortowane.
existing_keys		Określa zmienne, które są już posortowane i kierunek ich sortowania. Obowiązuje ten sam format, co we właściwości <i>keys</i> .

Właściwości węzła *spacetimeboxes*



Siatka czasoprzestrzeni stanowi rozszerzenie geokodowanych lokalizacji przestrzennych. Mówiąc dokładniej, siatka czasoprzestrzeni to łańcuch alfanumeryczny reprezentujący obszar czasu i przestrzeni o regularnych kształtach.

Tabela 73. Właściwości węzła *spacetimeboxes*

Właściwości węzła <i>spacetimeboxes</i>	Typ danych	Opis właściwości
mode	<i>IndividualRecords</i> <i>Hangouts</i>	
latitude_field	<i>field</i>	
longitude_field	<i>field</i>	
timestamp_field	<i>field</i>	

Tabela 73. Właściwości węzła *spacetimeboxes* (kontynuacja)

Właściwości węzła <i>spacetimeboxes</i>	Typ danych	Opis właściwości
densities	<i>[density, density, density...]</i>	<p>Każda gęstość jest łańcuchem. Na przykład: STB_GH8_1DAY</p> <p>Należy pamiętać, że obowiązują limity, do których gęstości są poprawne.</p> <p>Jako łańcucha mieszającego geoprzestrzennego można używać wartości od GH1 do GH15.</p> <p>Jako części czasowej można używać wartości:</p> <p>EVER 1YEAR 1MONTH 1DAY 12HOURS 8HOURS 6HOURS 4HOURS 3HOURS 2HOURS 1HOUR 30MINS 15MINS 10MINS 5MINS 2 MINS 1 MIN 30SECS 15SECS 10SECS 5 SECS 2 SECS 1SEC</p>
field_name_extension	<i>string</i>	
add_extension_as	<i>Prefix</i> <i>Suffix</i>	
hangout_density	<i>density</i>	Pojedyncza gęstość (patrz wyżej)
id_field	<i>field</i>	
qualifying_duration	1DAY 12HOURS 8HOURS 6HOURS 4HOURS 2HOURS 1HOUR 30MIN 15MIN 10MIN 5MIN 2MIN 1MIN 30SECS 15SECS 10SECS 5SECS 2SECS 1SECS	Ta wartość musi być łańcuchem.

Tabela 73. Właściwości węzła *spacetimeboxes* (kontynuacja)

Właściwości węzła <i>spacetimeboxes</i>	Typ danych	Opis właściwości
min_events	<i>integer</i>	Wartość minimalna to 2
qualifying_pct	<i>integer</i>	Musi być liczbą całkowitą z przedziału od 1 do 100.

Właściwości węzła *streamingtimeseries*



Węzeł Szeregi czasowe tworzy i ocenia modele szeregów czasowych w jednym kroku.

Uwaga: Ten węzeł zastępuje oryginalny węzeł Szeregi czasowe, który od wersji 18 programu SPSS Modeler jest już nieaktualny.

Tabela 74. Właściwości węzła *streamingtimeseries*

Właściwości węzła <i>streamingtimeseries</i>	Wartości	Opis właściwości
targets	<i>field</i>	Węzeł Szeregi czasowe przewiduje co najmniej jedną zmienną przewidywaną i opcjonalnie korzysta z jednej lub wielu zmiennych wejściowych jako predyktorów. Zmienne częstości i ważące nie są używane. Więcej informacji można znaleźć w temacie “Wspólne właściwości węzłów modelowania” na stronie 173.
candidate_inputs	[<i>field1 ... fieldN</i>]	Zmienne wejściowe lub predyktory używane przez model.
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	
period_field	<i>field</i>	
period_start_value	<i>integer</i>	
num_days_per_week	<i>integer</i>	

Tabela 74. Właściwości węzła *streamingtimeseries* (kontynuacja)

Właściwości węzła <i>streamingtimeseries</i>	Wartości	Opis właściwości
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	
missing_input	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>	Dostępna tylko, gdy wybrano ustawienie <i>date_time_field</i>
period_estimation	<i>list</i>	Dostępna tylko gdy wybrano ustawienie <i>use_period</i>

Tabela 74. Właściwości węzła *streamingtimeseries* (kontynuacja)

Właściwości węzła <i>streamingtimeseries</i>	Wartości	Opis właściwości
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>	
exsmooth_model_type	Simple HoltsLinearTrend BrownsLinearTrend DampedTrend SimpleSeasonal WintersAdditive WintersMultiplicative DampedTrendAdditive DampedTrendMultiplicative MultiplicativeTrendAdditive MultiplicativeSeasonal MultiplicativeTrendMultiplicative MultiplicativeTrend	
futureValue_type_method	Compute specify	
exsmooth_transformation_type	None SquareRoot NaturalLog	
arima.p	<i>integer</i>	
arima.d	<i>integer</i>	
arima.q	<i>integer</i>	
arima.sp	<i>integer</i>	
arima.sd	<i>integer</i>	
arima.sq	<i>integer</i>	

Tabela 74. Właściwości węzła *streamingtimeseries* (kontynuacja)

Właściwości węzła <i>streamingtimeseries</i>	Wartości	Opis właściwości
<i>arima_transformation_type</i>	None SquareRoot NaturalLog	
<i>arima_include_constant</i>	<i>flag</i>	
<i>tf_arima.p.fieldname</i>	<i>integer</i>	Dla funkcji przenoszenia.
<i>tf_arima.d.fieldname</i>	<i>integer</i>	Dla funkcji przenoszenia.
<i>tf_arima.q.fieldname</i>	<i>integer</i>	Dla funkcji przenoszenia.
<i>tf_arima.sp.fieldname</i>	<i>integer</i>	Dla funkcji przenoszenia.
<i>tf_arima.sd.fieldname</i>	<i>integer</i>	Dla funkcji przenoszenia.
<i>tf_arima.sq.fieldname</i>	<i>integer</i>	Dla funkcji przenoszenia.
<i>tf_arima.delay.fieldname</i>	<i>integer</i>	Dla funkcji przenoszenia.
<i>tf_arima.transformation_type.fieldname</i>	None SquareRoot NaturalLog	Dla funkcji przenoszenia.
<i>arima_detect_outliers</i>	<i>flag</i>	
<i>arima_outlier_additive</i>	<i>flag</i>	
<i>arima_outlier_level_shift</i>	<i>flag</i>	
<i>arima_outlier_innovational</i>	<i>flag</i>	
<i>arima_outlier_transient</i>	<i>flag</i>	
<i>arima_outlier_seasonal_additive</i>	<i>flag</i>	
<i>arima_outlier_local_trend</i>	<i>flag</i>	
<i>arima_outlier_additive_patch</i>	<i>flag</i>	
<i>conf_limit_pct</i>	<i>real</i>	
<i>events</i>	<i>fields</i>	
<i>forecastperiods</i>	<i>integer</i>	
<i>extend_records_into_future</i>	<i>flag</i>	
<i>conf_limits</i>	<i>flag</i>	
<i>noise_res</i>	<i>flag</i>	

Właściwości węzła *streamingts* (zdeprecjonowanego)



Uwaga: Pierwotny węzeł Szeregi czasowe jest nieaktualny w wersji 18 produktu SPSS Modeler i został zastąpiony przez nowy węzeł Szeregi czasowe, który zaprojektowano w taki sposób, aby wykorzystywał potencjał serwera IBM SPSS Analytic Server i przetwarzał wielkie zbiory danych (big data). Węzeł Szeregi czasowe tworzy i ocenia modele szeregów czasowych w jednym kroku, bez konieczności użycia węzła Przedziały czasowe.

Przykład

```
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")
```

Tabela 75. Właściwości węzła *streamingts*.

Właściwości węzła <i>streamingts</i>	Typ danych	Opis właściwości
custom_fields	<i>flag</i>	Jeśli custom_fields=false, używane są ustawienia z wcześniejszego węzła Typ. Jeśli custom_fields=true, należy określić właściwości targets i inputs.
targets	[field1...fieldN]	
inputs	[field1...fieldN]	
method	ExpertModeler Exsmooth Arima	
calculate_conf	<i>flag</i>	
conf_limit_pct	<i>real</i>	
use_time_intervals_node	<i>flag</i>	Jeśli use_time_intervals_node=true, to używane są ustawienia z wcześniejszego węzła Przedziały czasowe. Jeśli use_time_intervals_node=false, to należy określić właściwości interval_offset_position, interval_offset i interval_type.
interval_offset_position	LastObservation LastRecord	LastObservation to odpowiednik opcji Ostatnia prawidłowa obserwacja . LastRecord to odpowiednik Odliczaj wstecz od ostatniego rekordu .
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	Prosty HoltsLinearTrend BrownsLinearTrend DampedTrend SimpleSeasonal WintersAdditive WintersMultiplicative	

Tabela 75. Właściwości węzła *streamingts* (kontynuacja).

Właściwości węzła <i>streamingts</i>	Typ danych	Opis właściwości
<i>exsmooth_transformation_type</i>	None SquareRoot NaturalLog	
<i>arima_p</i>	<i>integer</i>	Ta sama właściwość co w węźle modelowania Szereg czasowy
<i>arima_d</i>	<i>integer</i>	Ta sama właściwość co w węźle modelowania Szereg czasowy
<i>arima_q</i>	<i>integer</i>	Ta sama właściwość co w węźle modelowania Szereg czasowy
<i>arima_sp</i>	<i>integer</i>	Ta sama właściwość co w węźle modelowania Szereg czasowy
<i>arima_sd</i>	<i>integer</i>	Ta sama właściwość co w węźle modelowania Szereg czasowy
<i>arima_sq</i>	<i>integer</i>	Ta sama właściwość co w węźle modelowania Szereg czasowy
<i>arima_transformation_type</i>	None SquareRoot NaturalLog	Ta sama właściwość co w węźle modelowania Szereg czasowy
<i>arima_include_constant</i>	<i>flag</i>	Ta sama właściwość co w węźle modelowania Szereg czasowy
<i>tf_arima_p.fieldname</i>	<i>integer</i>	Ta sama właściwość co w węźle modelowania Szereg czasowy. Dla funkcji przenoszenia.
<i>tf_arima_d.fieldname</i>	<i>integer</i>	Ta sama właściwość co w węźle modelowania Szereg czasowy. Dla funkcji przenoszenia.
<i>tf_arima_q.fieldname</i>	<i>integer</i>	Ta sama właściwość co w węźle modelowania Szereg czasowy. Dla funkcji przenoszenia.
<i>tf_arima_sp.fieldname</i>	<i>integer</i>	Ta sama właściwość co w węźle modelowania Szereg czasowy. Dla funkcji przenoszenia.
<i>tf_arima_sd.fieldname</i>	<i>integer</i>	Ta sama właściwość co w węźle modelowania Szereg czasowy. Dla funkcji przenoszenia.
<i>tf_arima_sq.fieldname</i>	<i>integer</i>	Ta sama właściwość co w węźle modelowania Szereg czasowy. Dla funkcji przenoszenia.
<i>tf_arima_delay.fieldname</i>	<i>integer</i>	Ta sama właściwość co w węźle modelowania Szereg czasowy. Dla funkcji przenoszenia.
<i>tf_arima_transformation_type.fieldname</i>	None SquareRoot NaturalLog	
<i>arima_detect_outlier_mode</i>	None Automatic	
<i>arima_outlier_additive</i>	<i>flag</i>	
<i>arima_outlier_level_shift</i>	<i>flag</i>	
<i>arima_outlier_innovational</i>	<i>flag</i>	
<i>arima_outlier_transient</i>	<i>flag</i>	
<i>arima_outlier_seasonal_additive</i>	<i>flag</i>	
<i>arima_outlier_local_trend</i>	<i>flag</i>	
<i>arima_outlier_additive_patch</i>	<i>flag</i>	
<i>deployment_force_rebuild</i>	<i>flag</i>	

Tabela 75. Właściwości węzła *streamingts* (kontynuacja).

Właściwości węzła <i>streamingts</i>	Typ danych	Opis właściwości
deployment_rebuild_mode	Count Percent	
deployment_rebuild_count	<i>number</i>	
deployment_rebuild_pct	<i>number</i>	
deployment_rebuild_field	<field>	

Właściwości *cplexnode*



Węzeł optymalizacji CPLEX zapewnia możliwość korzystania z zaawansowanej optymalizacji matematycznej (CPLEX) za pośrednictwem pliku modelu OPL (Optimization Programming Language). Ta funkcja jest dostępna w produkcie IBM Analytical Decision Management, jednak teraz węzła CPLEX można również używać w programie SPSS Modeler bez IBM Analytical Decision Management.

Więcej informacji na temat optymalizacji CPLEX i OPL zawiera dokumentacja IBM Analytical Decision Management https://www.ibm.com/support/knowledgecenter/SS6A3P_18.0.0/configurableapps/knowledge_center/product_landing.html.

Tabela 76. Właściwości *cplexnode*

Właściwości <i>cplexnode</i>	Typ danych	Opis właściwości
opl_model_text	<i>string</i>	Program skryptowy OPL (Optimization Programming Language), który będzie wykonywany przez węzeł optymalizacji CPLEX, a następnie generował wyniki optymalizacji.
opl_tuple_set_name	<i>string</i>	Nazwa zbioru krotek w modelu OPL odpowiadającym danym wejściowym.
opl_data_text	<i>string</i>	Definicja niektórych zmiennych lub danych używanych w OPL.
output_value_mode	<i>string</i>	Możliwe wartości to <i>raw</i> lub <i>dvar</i> . Jeśli wprowadzona zostanie wartość <i>dvar</i> , na karcie Wynik użytkownik musi określić nazwę zmiennej funkcji obiektu w OPL dla wyników. Jeśli wybrana zostanie wartość <i>raw</i> , funkcja celu zwróci wynik bezpośrednio, niezależnie od nazwy.
objective_function_value_fieldname	<i>string</i>	Nazwa zmiennej używana w danych wynikowych. Wartością domyślną jest <i>_OBJECTIVE</i> .

Rozdział 11. Właściwości węzłów związanych z operacjami na zmiennych

Właściwości węzła `anonymizenode`



Węzeł Anonimizacja przekształca sposób, w jaki nazwy i wartości zmiennych są reprezentowane w dalszej części strumienia, maskując oryginalne dane. Może to być przydatne, jeśli inni użytkownicy mają mieć możliwość budowania modeli z wykorzystaniem danych poufnych, takich jak nazwiska klientów lub inne szczegóły.

Przykład

```
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)
# Węzeł Anonimizacja wymaga zmiennych wejściowych przy ustawianiu wartości
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")
```

Tabela 77. Właściwości węzła `anonymizenode`

Właściwości węzła <code>anonymizenode</code>	Typ danych	Opis właściwości
<code>enable_anonymize</code>	<i>flag</i>	Wartość <code>True</code> włącza anonimizację wartości zmiennych (jest to równoważne wybraniu opcji Tak dla zmiennej w kolumnie Anonimizacja wartości).
<code>use_prefix</code>	<i>flag</i>	Wartość <code>True</code> powoduje, że używany będzie niestandardowy przedrostek, o ile został określony. Ma zastosowanie do zmiennych, które będą anonimizowane metodą mieszania (Hash) i jest równoważne z wybraniem dla tej zmiennej opcji Użytkownika w oknie dialogowym Zastąp wartości.
<code>prefix</code>	<i>string</i>	Równoważne wpisaniu przedrostka w polu tekstowym w oknie dialogowym Zastąp wartości. Wartością domyślną jest przedrostek domyślny, o ile nie określono innego.
<code>transformation</code>	Random Fixed	Określa, czy parametry przekształcenia zmiennej anonimizowanej metodą Przekształcenia będą losowe, czy stałe.
<code>set_random_seed</code>	<i>flag</i>	Wartość <code>True</code> powoduje użycie określonej wartości startowej (jeśli właściwość <code>transformation</code> ma wartość <code>Random</code>).
<code>random_seed</code>	<i>integer</i>	Gdy właściwość <code>set_random_seed</code> jest ustawiona na <code>True</code> , jest to wartość startowa generatora liczb losowych.
<code>scale</code>	<i>number</i>	Gdy właściwość <code>transformation</code> jest ustawiona na <code>Fixed</code> , ta wartość używana jest jako współczynnik skalowania. Maksymalna wartość skali wynosi 10, ale można ją zmniejszyć, aby uniknąć przepełnienia.

Tabela 77. Właściwości węzła *anonymizenode* (kontynuacja)

Właściwości węzła <i>anonymizenode</i>	Typ danych	Opis właściwości
translate	<i>number</i>	Gdy właściwość <i>transformation</i> jest ustawiona na <i>Fixed</i> , ta wartość używana jest jako współczynnik translacji. Maksymalna wartość translacji wynosi 1000, ale można ją zmniejszyć, aby uniknąć przepełnienia.

Właściwości węzła *autodatapreptime*



Węzeł Automatyczne przygotowywanie danych (Automated Data Preparation — ADP) może analizować dane użytkownika i wskazać poprawki, odfiltrować pola problematyczne lub prawdopodobnie bezużyteczne, w razie potrzeby obliczać nowe atrybuty i przyczyniać się do poprawy wydajności poprzez inteligentne filtrowanie i próbkowanie danych. Węzła tego można używać w sposób całkowicie zautomatyzowany, pozwalając mu wybierać i stosować poprawki, albo można przejrzeć zmiany przed ich dokonaniem i zaakceptować je lub odrzucić i wprowadzić poprawki.

Przykład

```
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")
```

Tabela 78. Właściwości węzła *autodatapreptime*

Właściwości węzła <i>autodatapreptime</i>	Typ danych	Opis właściwości
objective	Balanced Speed Accuracy Custom	
custom_fields	<i>flag</i>	Wartość <i>true</i> umożliwia określenie zmiennej przewidywanej, wejściowej i innych zmiennych dla bieżącego węzła. Wartość <i>false</i> powoduje użycie bieżących ustawień z wcześniejszego węzła Typ.
target	<i>field</i>	Określa jedną zmienną przewidywaną.
inputs	[<i>field1</i> ... <i>fieldN</i>]	Zmienne wejściowe lub predyktory używane przez model.
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>	Steruje dostępem do wszystkich zmiennych daty i czasu
compute_time_until_date	<i>flag</i>	

Tabela 78. Właściwości węzła autodataprepnode (kontynuacja)

Właściwości węzła autodataprepnode	Typ danych	Opis właściwości
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>	
replace_missing_nominal_target	<i>flag</i>	
replace_missing_ordinal_inputs	<i>flag</i>	

Tabela 78. Właściwości węzła autodataprepnode (kontynuacja)

Właściwości węzła autodataprepnode	Typ danych	Opis właściwości
replace_missing_ordinal_target	<i>flag</i>	
maximum_values_for_ordinal	<i>number</i>	
minimum_values_for_continuous	<i>number</i>	
outlier_cutoff_value	<i>number</i>	
outlier_method	Replace Delete	
rescale_continuous_inputs	<i>flag</i>	
rescaling_method	MinMax ZScore	
min_max_minimum	<i>number</i>	
min_max_maximum	<i>number</i>	
z_score_final_mean	<i>number</i>	
z_score_final_sd	<i>number</i>	
rescale_continuous_target	<i>flag</i>	
target_final_mean	<i>number</i>	
target_final_sd	<i>number</i>	
transform_select_input_fields	<i>flag</i>	
maximize_association_with_target	<i>flag</i>	
p_value_for_merging	<i>number</i>	
merge_ordinal_features	<i>flag</i>	
merge_nominal_features	<i>flag</i>	
minimum_cases_in_category	<i>number</i>	
bin_continuous_fields	<i>flag</i>	
p_value_for_binning	<i>number</i>	
perform_feature_selection	<i>flag</i>	
p_value_for_selection	<i>number</i>	
perform_feature_construction	<i>flag</i>	
transformed_target_name_extension	<i>string</i>	
transformed_inputs_name_extension	<i>string</i>	
constructed_features_root_name	<i>string</i>	
years_duration_name_extension	<i>string</i>	
months_duration_name_extension	<i>string</i>	
days_duration_name_extension	<i>string</i>	
hours_duration_name_extension	<i>string</i>	
minutes_duration_name_extension	<i>string</i>	
seconds_duration_name_extension	<i>string</i>	
year_cyclical_name_extension	<i>string</i>	
month_cyclical_name_extension	<i>string</i>	
day_cyclical_name_extension	<i>string</i>	
hour_cyclical_name_extension	<i>string</i>	

Tabela 78. Właściwości węzła `autodatapreptime` (kontynuacja)

Właściwości węzła <code>autodatapreptime</code>	Typ danych	Opis właściwości
<code>minute_cyclical_name_extension</code>	<i>string</i>	
<code>second_cyclical_name_extension</code>	<i>string</i>	

Właściwości węzła `astimeintervalsnode`



Węzeł Przedziały czasowe może być używany do określenia przedziałów i wyliczenia nowej zmiennej czasu na potrzeby oszacowania lub prognozowania. Obsługiwany jest cały zakres przedziałów czasowych, od sekund po lata.

Tabela 79. Właściwości węzła `astimeintervalsnode`

Właściwości węzła <code>astimeintervalsnode</code>	Typ danych	Opis właściwości
<code>time_field</code>	<i>field</i>	Przyjmuje tylko jedną zmienną ciągłą. Ta zmienna jest używana w węźle jako wartość kluczowa agregacji podczas przekształcania przedziału. Zmienna całkowita jest traktowana jak indeks czasu.
<code>dimensions</code>	<i>[field1 field2 ... fieldn]</i>	Te zmienne służą do tworzenia poszczególnych szeregów czasowych na podstawie wartości zmiennych.
<code>fields_to_aggregate</code>	<i>[field1 field2 ... fieldn]</i>	Te zmienne są agregowane w ramach modyfikacji przedziału zmiennej czasu. Wszystkie zmienne, które nie zostaną uwzględnione w tym selektorze, zostaną odfiltrowane z danych opuszczających węzeł.

Właściwości węzła `binningnode`



Węzeł Kategoryzacja automatycznie tworzy nowe zmienne nominalne (zbioru) na podstawie wartości z jednej lub większej liczby istniejących zmiennych ilościowych (zakres liczbowy). Można na przykład przekształcić ilościową zmienną przychodu na nową zmienną jakościową zawierającą grupy przychodu stanowiące odchylenia od średniej. Po utworzeniu kategorii dla nowej zmiennej na podstawie punktu podziału można wygenerować węzeł Wyliczanie.

Przykład

```
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)
```

Tabela 80. Właściwości węzła binningnode

Właściwości węzła binningnode	Typ danych	Opis właściwości
fields	[field1 field2 ... fieldn]	Zmienne ilościowe (zakres liczbowy) oczekujące na przekształcenie. Można kategoryzować wiele zmiennych jednocześnie.
method	FixedWidth EqualCount Rank SDev Optimal	Metoda używana do ustalania punktów podziału dla nowych przedziałów zmiennych (kategorii).
rcalculate_bins	Always IfNecessary	Określa, czy kategorie są ponownie obliczane, a dane umieszczane w odpowiednich kategoriach przy każdym wykonaniu węzła, czy też dane są tylko dodawane do istniejących kategorii i ewentualnie dodanych nowych kategorii.
fixed_width_name_extension	string	Rozszerzenie domyślne to <i>_BIN</i> .
fixed_width_add_as	Suffix Prefix	Określa, czy rozszerzenie jest dodawane na końcu (suffix, przyrostek), czy na początku (prefix, przedrostek) nazwy zmiennej. Rozszerzenie domyślne to <i>income_BIN</i> .
fixed_bin_method	Width Count	
fixed_bin_count	integer	Liczba całkowita używana do określenia liczby przedziałów o ustalonej szerokości (kategorii) dla nowych zmiennych.
fixed_bin_width	real	Wartość (liczba całkowita lub rzeczywista), jaka będzie używana do obliczenia „szerokości” przedziału.
equal_count_name_extension	string	Domyślne rozszerzenie to <i>_TILE</i> .
equal_count_add_as	Suffix Prefix	Określa rozszerzenie (przyrostek albo przedrostek) nazw zmiennych generowanych z zastosowaniem standardowych p-tyli. Domyślnym rozszerzeniem jest <i>_TILE</i> plus <i>N</i> , gdzie <i>N</i> oznacza liczbę N-tyli.
tile4	flag	Generuje cztery kwantyle, każdy zawierający po 25% obserwacji.
tile5	flag	Generuje pięć kwintyli.
tile10	flag	Generuje dziesięć decyli.
tile20	flag	Generuje 20 równych przedziałów.
tile100	flag	Generuje 100 centyli.
use_custom_tile	flag	
custom_tile_name_extension	string	Rozszerzenie domyślne to <i>_TILEN</i> .
custom_tile_add_as	Suffix Prefix	
custom_tile	integer	
equal_count_method	RecordCount ValueSum	Metoda RecordCount próbuje przypisać do każdego przedziału równą liczbę rekordów, natomiast ValueSum przypisuje rekordy tak, by sumy wartości w każdym przedziale były równe.

Tabela 80. Właściwości węzła *binningnode* (kontynuacja)

Właściwości węzła <i>binningnode</i>	Typ danych	Opis właściwości
<code>tied_values_method</code>	Next Current Random	Określa, w którym przedziale ma być umieszczona powiązana wartość danych.
<code>rank_order</code>	Ascending Descending	Właściwość przyjmuje wartość Ascending (rosnąco, najniższa wartość oznaczona 1) albo Descending (malejąco, najwyższa wartość oznaczona 1).
<code>rank_add_as</code>	Suffix Prefix	Ta opcja ma zastosowanie do rangi, rangi ułamkowej i rangi procentowej.
<code>rank</code>	<i>flag</i>	
<code>rank_name_extension</code>	<i>string</i>	Rozszerzenie domyślne to <i>_RANK</i> .
<code>rank_fractional</code>	<i>flag</i>	Przeprowadza rangowanie obserwacji, w którym nowa zmienna jest równa randze podzielonej przez sumę wag obserwacji bez braków danych. Rangi ułamkowe mieszczą się w przedziale od 0 do 1.
<code>rank_fractional_name_extension</code>	<i>string</i>	Rozszerzenie domyślne to <i>_F_RANK</i> .
<code>rank_pct</code>	<i>flag</i>	Każda ranga jest podzielona przez liczbę rekordów o poprawnych wartościach i pomnożona przez 100. Ułamkowe rangi procentowe mieszczą się w przedziale od 1 do 100.
<code>rank_pct_name_extension</code>	<i>string</i>	Rozszerzenie domyślne to <i>_P_RANK</i> .
<code>sdev_name_extension</code>	<i>string</i>	
<code>sdev_add_as</code>	Suffix Prefix	
<code>sdev_count</code>	One Two Three	
<code>optimal_name_extension</code>	<i>string</i>	Rozszerzenie domyślne to <i>_OPTIMAL</i> .
<code>optimal_add_as</code>	Suffix Prefix	
<code>optimal_supervisor_field</code>	<i>field</i>	Zmienna używana jako nadzorująca; są z nią powiązane zmienne wybrane do kategoryzacji.
<code>optimal_merge_bins</code>	<i>flag</i>	Określa, że przedziały (kategorie) z małą liczbą obserwacji będą dodawane do większych sąsiednich przedziałów.
<code>optimal_small_bin_threshold</code>	<i>integer</i>	
<code>optimal_pre_bin</code>	<i>flag</i>	Określa, że ma być wykonywana wstępna kategoryzacja.
<code>optimal_max_bins</code>	<i>integer</i>	Określa górną granicę, by uniknąć tworzenia zbyt dużej liczby przedziałów.
<code>optimal_lower_end_point</code>	Inclusive Exclusive	
<code>optimal_first_bin</code>	Unbounded Bounded	
<code>optimal_last_bin</code>	Unbounded Bounded	

Właściwości węzła *derivenode*



Węzeł wyliczeń modyfikuje wartości danych lub tworzy nowe zmienne z co najmniej jednej istniejącej zmiennej. Tworzy pola typu formuła, flaga, nominalne, stan, liczebność i warunkowe.

Przykład 1

```
# Utwórz i skonfiguruj węzeł Wyliczenie zmiennej typu flaga
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\"")

# Utwórz i skonfiguruj węzeł Wyliczenie zmiennej typu Condition
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\"")
```

Przykład 2

W tym skrypcie przyjęto założenie, że istnieją dwie kolumny liczbowe o nazwach XPos i YPos, które reprezentują współrzędne X i Y punktu (na przykład miejsca, w którym zaszło zdarzenie). Skrypt tworzy węzeł Wyliczenie, który oblicza kolumnę geoprzestrzenną na podstawie współrzędnych X i Y odzwierciedlających dany punkt w konkretnym układzie współrzędnych:

```
stream = modeler.script.stream()
# Pozostały kod konfiguracji strumienia
node = stream.createAt("derive", "Location", 192, 96)
node.setPropertyValue("new_name", "Location")
node.setPropertyValue("formula_expr", "['XPos', 'YPos']")
node.setPropertyValue("formula_type", "Geospatial")
# Teraz mamy ogólny typ miary. Definiujemy szczegóły
# obiektu geoprzestrzennego
node.setPropertyValue("geo_type", "Point")
node.setPropertyValue("has_coordinate_system", True)
node.setPropertyValue("coordinate_system", "ETRS_1989_EPSG_Arctic_zone_5-47")
```

Tabela 81. Właściwości węzła *derivenode*

Właściwości węzła <i>derivenode</i>	Typ danych	Opis właściwości
<code>new_name</code>	<i>string</i>	Nazwa nowej zmiennej.
<code>mode</code>	Single Multiple	Określa jedną lub wiele zmiennych.
<code>fields</code>	<i>list</i>	Wartość używana tylko w węźle Multiple do wybierania wielu zmiennych.
<code>name_extension</code>	<i>string</i>	Określa rozszerzenie nazw(y) nowej zmiennej/nowych zmiennych.
<code>add_as</code>	Suffix Prefix	Rozszerzenie jest dodawane jako przedrostek (na początku) albo jako przyrostek (na końcu) nazwy zmiennej.

Tabela 81. Właściwości węzła *derivenode* (kontynuacja)

Właściwości węzła <i>derivenode</i>	Typ danych	Opis właściwości
result_type	Formuła Flag Set State Count Conditional	Sześć typów nowych zmiennych, jakie można tworzyć.
formula_expr	<i>string</i>	Wyrażenie obliczające wartość nowej zmiennej w węźle Wyliczanie.
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>	Zmienna ustrukturyzowana udostępniająca warunek związany z daną wartością.
state_on_val	<i>string</i>	Określa wartość nowej zmiennej, gdy spełniony jest warunek włączenia.
state_off_val	<i>string</i>	Określa wartość nowej zmiennej, gdy spełniony jest warunek wyłączenia.
state_on_expression	<i>string</i>	
state_off_expression	<i>string</i>	
state_initial	On Off	Przypisuje każdemu rekordowi nowej zmiennej wartość początkową On (włączenie) albo Off (wyłączenie). Ta wartość może się zmieniać w zależności od tego, czy poszczególne warunki są spełnione.
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>	
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	Ta właściwość umożliwia zdefiniowanie pomiaru związanego ze zmienną wyliczaną. Do funkcji ustawiającej można przekazać łańcuch albo jedną z wartości MeasureType. Funkcja pobierająca zawsze zwraca wartości MeasureType.
collection_measure	Range / MeasureType.RANGE Flag / MeasureType.FLAG Set / MeasureType.SET OrderedSet / MeasureType.ORDERED_SET Typeless / MeasureType.TYPELESS	W przypadku zmiennych collection (listy o głębokości 0) ta właściwość definiuje typ pomiaru związany z bazowymi wartościami.

Tabela 81. Właściwości węzła *derivenode* (kontynuacja)

Właściwości węzła <i>derivenode</i>	Typ danych	Opis właściwości
geo_type	Point MultiPoint LineString MultiLineString Polygon MultiPolygon	W przypadku zmiennych geoprzestrzennych ta właściwość definiuje typ obiektu reprezentowanego przez zmienną. Powinien być on zgodny z głębokością listy wartości.
has_coordinate_system	<i>boolean</i>	W przypadku zmiennych geoprzestrzennych ta właściwość określa, czy zmienna ma układ współrzędnych.
coordinate_system	<i>string</i>	W przypadku zmiennych geoprzestrzennych ta właściwość definiuje układ współrzędnych dla zmiennej.

Właściwości węzła *ensemblenode*



Węzeł Zespoleń łączy co najmniej dwa modele użytkowe w celu uzyskania bardziej dokładnych predykcji, jakie można uzyskać z dowolnego modelu.

Przykład

```
# Utwórz i skonfiguruj węzeł Zespoleń
# Użyj tego węzła z modelami z pliku 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")
```

Tabela 82. Właściwości węzła *ensemblenode*.

Właściwości węzła <i>ensemblenode</i>	Typ danych	Opis właściwości
ensemble_target_field	<i>field</i>	Określa zmienną docelową dla wszystkich modeli używanych w zespoleń.
filter_individual_model_output	<i>flag</i>	Określa, czy wyniki oceny z poszczególnych modeli powinny być pomijane.
flag_ensemble_method	Voting ConfidenceWeightedVoting RawPropensityWeightedVoting AdjustedPropensityWeightedVoting HighestConfidence AverageRawPropensity AverageAdjustedPropensity	Określa metodę wyznaczania oceny zespoleń. To ustawienie obowiązuje tylko wtedy, gdy wybrana zmienna docelowa jest zmienną typu flaga.
set_ensemble_method	Voting ConfidenceWeightedVoting HighestConfidence	Określa metodę wyznaczania oceny zespoleń. To ustawienie obowiązuje tylko wtedy, gdy wybrana zmienna docelowa jest zmienną nominalną.

Tabela 82. Właściwości węzła *ensemblenode* (kontynuacja).

Właściwości węzła <i>ensemblenode</i>	Typ danych	Opis właściwości
<code>flag_voting_tie_selection</code>	Random HighestConfidence RawPropensity AdjustedPropensity	Jeśli wybrana jest metoda głosowania, określa sposób rozstrzygnięcia powiązań. To ustawienie obowiązuje tylko wtedy, gdy wybrana zmienna docelowa jest zmienną typu flaga.
<code>set_voting_tie_selection</code>	Random HighestConfidence	Jeśli wybrana jest metoda głosowania, określa sposób rozstrzygnięcia powiązań. To ustawienie obowiązuje tylko wtedy, gdy wybrana zmienna docelowa jest zmienną nominalną.
<code>calculate_standard_error</code>	<i>flag</i>	W przypadku ciągłej zmiennej przewidywanej obliczenia błędu standardowego są uruchamiane domyślnie w celu obliczenia różnicy między wartością zmierzoną a rzeczywistą, a także do prezentacji stopnia dopasowania tych oszacowań.

Właściwości węzła *fillernode*



Węzeł wypełniania zastępuje wartości zmiennych i zmienia typ składowania. Wartości mogą być zastępowane na podstawie warunku CLEM, np. `@BLANK(@FIELD)`. Alternatywnie, można wybrać, aby wszystkie wartości puste lub null zastępowane były konkretną wartością. Węzeł wypełniania często jest używany z węzłem Typ do zastępowania braków danych.

Przykład

```
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\"")
```

Tabela 83. Właściwości węzła *fillernode*

Właściwości węzła <i>fillernode</i>	Typ danych	Opis właściwości
<code>fields</code>	<i>list</i>	Zmienne ze zbioru danych, których wartości będą badane i zastępowane.
<code>replace_mode</code>	Always Conditional Blank Null BlankAndNull	Można zastępować wszystkie wartości, wartości puste lub wartości null albo zastępować na podstawie określonego warunku.
<code>condition</code>	<i>string</i>	
<code>replace_with</code>	<i>string</i>	

Właściwości węzła filternode



Węzeł Filtrowanie filtruje (odrzuca) zmienne, zmienia nazwy zmiennych i mapuje zmienne z jednego węzła źródłowego do drugiego.

Przykład

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

Użycie właściwości default_include. Należy zwrócić uwagę, że ustawienie wartości właściwości default_include nie powoduje automatycznie uwzględnienia lub wykluczenia wszystkich zmiennych; określa jedynie stan domyślny dla bieżącego wyboru. Funkcjonalnie jest to odpowiednik zaznaczenia przycisku **Domyślne uwzględnianie zmiennych** w oknie dialogowym węzła filtrowania. Załóżmy na przykład, że uruchamiamy następujący skrypt:

```
node = modeler.script.stream().create("filter", "Filter")
node.setPropertyValue("default_include", False)
# Uwzględnij te dwie zmienne na liście
for f in ["Age", "Sex"]:
    node.setKeyedPropertyValue("include", f, True)
```

Spowoduje to, że węzeł przekaże zmienne *Age* i *Sex*, a odrzuci wszystkie pozostałe. Załóżmy teraz, że ponownie uruchamiamy ten sam skrypt, ale wskazujemy dwie inne zmienne:

```
node = modeler.script.stream().create("filter", "Filter")
node.setPropertyValue("default_include", False)
# Uwzględnij te dwie zmienne na liście
for f in ["BP", "Na"]:
    node.setKeyedPropertyValue("include", f, True)
```

Spowoduje to dodanie dwóch zmiennych do filtru, zatem łącznie przekazane zostaną cztery zmienne (*Age*, *Sex*, *BP*, *Na*). Innymi słowy, zmiana wartości default_include na False nie kasuje wyboru zmiennych.

Jeśli teraz zmienimy default_include na True — czy to za pomocą skryptu, czy okna dialogowego węzła filtrowania — działanie węzła zostanie odwrócone i cztery wymienione zmienne będą odrzucane, a nie uwzględniane. W razie wątpliwości można poeksperymentować z elementami sterującymi w oknie dialogowym węzła filtrowania, aby zorientować się w działaniu tego mechanizmu.

Tabela 84. Właściwości węzła filternode

Właściwości węzła filternode	Typ danych	Opis właściwości
default_include	flag	Właściwość wprowadzana, która określa, czy zmienne mają być domyślnie przekazywane, czy filtrowane: Należy zwrócić uwagę, że ustawienie wartości tej właściwości nie powoduje automatycznie uwzględnienia lub wykluczenia wszystkich zmiennych; określa jedynie, czy zmienne są domyślnie uwzględniane, czy wykluczane. Dodatkowe komentarze zawiera poniższy przykład.
include	flag	Właściwość wprowadzana decydująca o uwzględnianiu lub usuwaniu zmiennych.
new_name	string	

Właściwości węzła historynode



Węzeł Historia tworzy nowe zmienne zawierające dane ze zmiennych z wcześniejszych rekordów. Węzły historii są najczęściej używane w przypadku danych sekwencyjnych, takich jak dane szeregu czasowego. Przed użyciem węzła historii można posortować dane za pomocą węzła Sortowanie.

Przykład

```
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")
```

Tabela 85. Właściwości węzła historynode

Właściwości węzła historynode	Typ danych	Opis właściwości
fields	<i>list</i>	Zmienne, dla których ma być wygenerowana historia.
offset	<i>number</i>	Określa ostatni rekord przed rekordem bieżącym, z którego mają zostać wyodrębnione historyczne wartości zmiennych.
span	<i>number</i>	Określa liczbę wcześniejszych rekordów, z których wartości mają zostać wyodrębnione.
unavailable	Discard Leave Fill	Dotyczy rekordów bez wartości historycznych, zwykle kilku pierwszych rekordów znajdujących się w górnej części zbioru danych, dla których nie istnieją wcześniejsze rekordy mogące stanowić źródło danych historycznych.
fill_with	String Number	Określa wartość lub łańcuch, który ma być używany w przypadku rekordów, w których wartości historyczne są niedostępne.

Właściwości węzła partitionnode



Węzeł Podział na podzbiory generuje zmienną dzielącą na podzbiory, która dzieli dane na osobne podzbiory dla etapów do uczenia, testowania i walidacji podczas budowania modelu.

Przykład

```
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")
```

Tabela 86. Właściwości węzła *partitionnode*

Właściwości węzła <i>partitionnode</i>	Typ danych	Opis właściwości
<code>new_name</code>	<i>string</i>	Nazwa zmiennej dzielącej na podzbiory wygenerowanej przez węzeł.
<code>create_validation</code>	<i>flag</i>	Określa, czy ma być tworzony podzbiór walidacyjny.
<code>training_size</code>	<i>integer</i>	Odsetek rekordów (0–100), jaki ma być przydzielony do podzbioru uczącego.
<code>testing_size</code>	<i>integer</i>	Odsetek rekordów (0–100), jaki ma być przydzielony do podzbioru testowego.
<code>validation_size</code>	<i>integer</i>	Odsetek rekordów (0–100), jaki ma być przydzielony do podzbioru walidacyjnego. Ignorowana, jeśli podzbiór walidacyjny nie jest tworzony.
<code>training_label</code>	<i>string</i>	Etykieta podzbioru uczącego.
<code>testing_label</code>	<i>string</i>	Etykieta podzbioru testowego.
<code>validation_label</code>	<i>string</i>	Etykieta podzbioru walidacyjnego. Ignorowana, jeśli podzbiór walidacyjny nie jest tworzony.
<code>value_mode</code>	System SystemAndLabel Label	Określa wartości użyte do reprezentowania każdego podzbioru w danych. Na przykład próba ucząca może być reprezentowana przez systemową wartość całkowitą 1, etykietę <code>Training</code> albo kombinację <code>1_Training</code> .
<code>set_random_seed</code>	<i>Boolean</i>	Określa, czy ma być używana określona przez użytkownika wartość początkowa generatora liczb losowych.
<code>random_seed</code>	<i>integer</i>	Określona przez użytkownika wartość początkowa generatora liczb losowych. Aby ta wartość była używana, <code>set_random_seed</code> musi mieć wartość <code>True</code> .
<code>enable_sql_generation</code>	<i>Boolean</i>	Określa, czy przypisywanie rekordów do podzbiorów ma być zlecane serwerowi SQL.
<code>unique_field</code>		Określa zmienne wejściowe zapewniające losowe, ale powtarzalne przypisywanie rekordów do podzbiorów. Aby ta wartość była używana, <code>enable_sql_generation</code> musi mieć wartość <code>True</code> .

Właściwości węzła *reclassifynode*



Węzeł Rekodowanie przekształca jeden zestaw wartości jakościowych w inny. Rekodowanie jest przydatne do związania kategorii lub ponownego pogrupowania danych do analizy.

Przykład

```
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"])
```

Tabela 87. Właściwości węzła reclassifynode

Właściwości węzła reclassifynode	Typ danych	Opis właściwości
mode	Single Multiple	Single reklasyfikuje kategorie dla jednej zmiennej. Multiple aktywuje opcje umożliwiające transformację więcej niż jednej zmiennej na raz.
replace_field	flag	
field	string	Używana tylko w trybie Single.
new_name	string	Używana tylko w trybie Single.
fields	[field1 field2 ... fieldn]	Używana tylko w trybie Multiple.
name_extension	string	Używana tylko w trybie Multiple.
add_as	Suffix Prefix	Używana tylko w trybie Multiple.
reclassify	string	Właściwość ustrukturyzowana przeznaczona na wartości zmiennych.
use_default	flag	Użyj wartości domyślnej.
domyślny	string	Określ wartość domyślną.
pick_list	[string string ... string]	Umożliwia użytkownikowi zaimportowanie listy znanych nowych wartości, które wypełnią listę rozwijaną w tabeli.

Właściwości węzła reordernode



Węzeł Reorganizacja definiuje rzeczywistą kolejność, w jakiej wyświetlane są zmienne w dalszej części strumienia. Ta kolejność wpływa na wyświetlanie zmiennych w różnych obszarach, takich jak tabele, listy i selektor zmiennych. Ta operacja jest przydatna podczas pracy z obszernymi bazami danych w celu zapewnienia lepszej widoczności zmiennych, które interesują użytkownika.

Przykład

```
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"])
```

Tabela 88. właściwości węzła reordernode

Właściwości węzła reordernode	Typ danych	Opis właściwości
mode	Custom Auto	Można posortować wartości automatycznie lub określić kolejność niestandardową.
sort_by	Name Type Storage	
ascending	flag	
start_fields	[field1 field2 ... fieldn]	Nowe zmienne są wstawiane za tymi zmiennymi.

Tabela 88. właściwości węzła *reordernode* (kontynuacja)

Właściwości węzła <i>reordernode</i>	Typ danych	Opis właściwości
end_fields	[<i>field1 field2 ... fieldn</i>]	Nowe zmienne są wstawiane przed tymi zmiennymi.

Właściwości węzła *reprojectnode*



W programie SPSS Modeler składniki, takie jak funkcje przestrzenne w Konstruktorze wyrażań, węzeł STP i węzeł Wizualizacja na mapie używają rzutowanego układu współrzędnych. Węzeł Odwzorowanie umożliwia zmianę układu współrzędnych każdego rodzaju importowanych danych, które korzystają z układu współrzędnych geograficznych.

Tabela 89. Właściwości węzła *reprojectnode*

Właściwości węzła <i>reprojectnode</i>	Typ danych	Opis właściwości
reproject_fields	[<i>field1 field2 ... fieldn</i>]	Lista wszystkich zmiennych, których rzutowanie ma być zmienione.
reproject_type	Streamdefault Specify	Wybierz sposób zmiany rzutowania zmiennych.
coordinate_system	<i>string</i>	Nazwa układu współrzędnych, który ma być zastosowany względem zmiennych. Przykład: set <i>reprojectnode.coordinate_system</i> = "WGS_1984_World_Mercator"

Właściwości węzła *restructurenode*



Węzeł Restrukturyzacja przekształca zmienną nominalną lub typu flaga na grupę zmiennych, które mogą być wypełnione wartościami jeszcze innej zmiennej. Na przykład, dana jest zmienna o nazwie *payment type* (rodzaj płatności), której wartości to *credit* (kredyt), *cash* (gotówka) i *debit* (debet) i utworzone zostaną trzy nowe zmienne (*credit*, *cash*, *debit*), a każda z nich może zawierać wartość dla rzeczywistej dokonanej płatności.

Przykład

```
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"])
```

Tabela 90. Właściwości węzła *restructurenode*

Właściwości węzła <i>restructurenode</i>	Typ danych	Opis właściwości
fields_from	[<i>category category category</i>] all	
include_field_name	<i>flag</i>	Określa, czy w nazwie zreorganizowanej zmiennej ma być używana nazwa zmiennej.
value_mode	OtherFields Flags	Wskazuje tryb określania wartości dla zmiennych zreorganizowanych. W przypadku wartości OtherFields należy określić zmienne, które mają być używane. W przypadku wartości Flags flagi są liczbowe.

Tabela 90. Właściwości węzła `restructurenode` (kontynuacja)

Właściwości węzła <code>restructurenode</code>	Typ danych	Opis właściwości
<code>value_fields</code>	<i>list</i>	Wymagana, jeśli <code>value_mode</code> ma wartość <code>OtherFields</code> . Określa, których zmiennych użyć jako zmiennych wartości.

Właściwości węzła `rfanalysisnode`



Węzeł analizy RFM (Recency — Aktualność, Frequency — Częstość, Monetary — Kwota) umożliwia określenie ilościowo, którzy klienci najprawdopodobniej będą najlepszymi, poprzez dokonanie oceny, kiedy ostatnio dokonali zakupu (aktualność), jak często dokonują zakupu (częstość) i jak dużo wydali na wszystkie transakcje (kwota).

Przykład

```
node = stream.create("rfanalysis", "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])
```

Tabela 91. Właściwości węzła `rfanalysisnode`

Właściwości węzła <code>rfanalysisnode</code>	Typ danych	Opis właściwości
<code>recency</code>	<i>field</i>	Określa zmienną aktualności. Może to być data, znacznik czasu lub zwykła liczba.
<code>frequency</code>	<i>field</i>	Określa zmienną częstości.
<code>monetary</code>	<i>field</i>	Określa zmienną kwoty.
<code>recency_bins</code>	<i>integer</i>	Określa liczbę przedziałów aktualności do wygenerowania.
<code>recency_weight</code>	<i>number</i>	Określa wagę, jaka ma być zastosowana względem danych o aktualności. Domyślną wartością jest 100.
<code>frequency_bins</code>	<i>integer</i>	Określa liczbę przedziałów częstości do wygenerowania.
<code>frequency_weight</code>	<i>number</i>	Określa wagę, jaka ma być zastosowana względem danych o częstości. Domyślną wartością jest 10.
<code>monetary_bins</code>	<i>integer</i>	Określa liczbę przedziałów kwot do wygenerowania.
<code>monetary_weight</code>	<i>number</i>	Określa wagę, jaka ma być zastosowana względem danych o kwotach. Domyślną wartością jest 1.
<code>tied_values_method</code>	Next Current	Określa, w którym przedziale ma być umieszczona powiązana wartość.
<code>recalculate_bins</code>	Always IfNecessary	

Tabela 91. Właściwości węzła *rfanalysisnode* (kontynuacja)

Właściwości węzła <i>rfanalysisnode</i>	Typ danych	Opis właściwości
add_outliers	<i>flag</i>	Dostępna tylko wtedy, gdy <i>recalculate_bins</i> ma wartość <i>IfNecessary</i> . Jeśli jest ustawiona, rekordy leżące poniżej dolnego przedziału zostaną dodane do dolnego przedziału, a rekordy powyżej najwyższego przedziału zostaną dodane do najwyższego przedziału.
binned_field	Recency Frequency Monetary	
recency_thresholds	<i>value value</i>	Dostępna tylko wtedy, gdy <i>recalculate_bins</i> ma wartość <i>Always</i> . Określ górne i dolne granice przedziałów aktualności. Górna granica jednego przedziału jest dolną granicą następnego. Na przykład [10 30 60] definiuje dwa przedziały: pierwszy z górną i dolną granicą równą odpowiednio 10 i 30, a drugi z granicami 30 i 60.
frequency_thresholds	<i>value value</i>	Dostępna tylko wtedy, gdy <i>recalculate_bins</i> ma wartość <i>Always</i> .
monetary_thresholds	<i>value value</i>	Dostępna tylko wtedy, gdy <i>recalculate_bins</i> ma wartość <i>Always</i> .

Właściwości węzła *settoflagnode*



Węzeł Ustaw jako flaga jest używany do wyliczania zmiennych flag na podstawie zmiennych wartości jakościowych zdefiniowanych dla co najmniej jednej zmiennej nominalnej.

Przykład

```
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"])
```

Tabela 92. Właściwości węzła *settoflagnode*

Właściwości węzła <i>settoflagnode</i>	Typ danych	Opis właściwości
fields_from	[<i>category category</i> <i>category</i>] all	
true_value	<i>string</i>	Określa wartość oznaczającą prawdę, jaka będzie używana przez węzeł podczas ustawiania flagi. Domyślną wartością jest T.
false_value	<i>string</i>	Określa wartość oznaczającą fałsz, jaka będzie używana przez węzeł podczas ustawiania flagi. Domyślną wartością jest F.

Tabela 92. Właściwości węzła *settoflagnode* (kontynuacja)

Właściwości węzła <i>settoflagnode</i>	Typ danych	Opis właściwości
use_extension	<i>flag</i>	Powoduje użycie rozszerzenia jako przedrostka lub przyrostka do nowej zmiennej typu flaga.
extension	<i>string</i>	
add_as	Suffix Prefix	Określa, czy rozszerzenie jest dodawane jako przyrostek, czy przedrostek.
aggregate	<i>flag</i>	Grupuje rekordy na podstawie zmiennych grupujących. Wszystkie zmienne typu flaga w grupie są włączone, jeśli którykolwiek rekord ma wartość true.
keys	<i>list</i>	Zmienne grupujące.

Właściwości węzła *statisticstransformnode*



Węzeł przekształceń Statistics uruchamia wybór komend składni IBM SPSS Statistics dla źródeł danych w programie IBM SPSS Modeler. Ten węzeł wymaga licencjonowanej kopii programu IBM SPSS Statistics.

Właściwości tego węzła są opisane w sekcji “Właściwości węzła *statisticstransformnode*” na stronie 329.

Właściwości węzła *timeintervalsnode* (zdeprecjonowanego)



Uwaga: Ten węzeł został uznany za nieaktualny w wersji 18 produktu SPSS Modeler i zastąpiony przez nowy węzeł Szereg czasowy. Węzeł Przedział czasowy określa przedziały i tworzy etykiety (w razie potrzeby) do modelowania danych szeregów czasowych. Jeśli odstępy między wartościami nie są równomierne, węzeł może dopełniać lub agregować wartości, by uzyskać równe odstępy między rekordami.

Przykład

```
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")
```

Tabela 93. Właściwości węzła *timeintervalsnode*.

Właściwości węzła <i>timeintervalsnode</i>	Typ danych	Opis właściwości
interval_type	None Periods CyclicPeriods Years Quarters Months DaysPerWeek DaysNonPeriodic HoursPerDay HoursNonPeriodic MinutesPerDay MinutesNonPeriodic SecondsPerDay SecondsNonPeriodic	
mode	Etykieta Create	Określa, czy etykiety mają być dodawane kolejno do rekordów, czy też szereg ma być budowany w oparciu o zmienną daty, znacznika czasu lub czasu.
field	<i>field</i>	W przypadku budowania szeregu z danych określa zmienną zawierającą datę lub godzinę dla każdego rekordu.
period_start	<i>integer</i>	Określa przedział początkowy dla okresów lub okresów cyklicznych
cycle_start	<i>integer</i>	Cykl początkowy dla okresów cyklicznych.
year_start	<i>integer</i>	W przypadku typów przedziałów, w których ma to zastosowanie: rok, w którym przypada pierwszy przedział.
quarter_start	<i>integer</i>	W przypadku typów przedziałów, w których ma to zastosowanie: kwartał, w którym przypada pierwszy przedział.
month_start	Styczeń Luty Marzec Kwiecień Maj Czerwiec Lipiec Sierpień Wrzesień Październik Listopad Grudzień	
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>	W przypadku okresów cyklicznych: numer w każdym cyklu.

Tabela 93. Właściwości węzła *timeintervalsnode* (kontynuacja).

Właściwości węzła <i>timeintervalsnode</i>	Typ danych	Opis właściwości
<i>fiscal_year_begins</i>	Styczeń Luty Marzec Kwiecień Maj Czerwiec Lipiec Sierpień Wrzesień Październik Listopad Grudzień	W przypadku okresów kwartalnych określa miesiąc, w którym rozpoczyna się rok obrotowy (fiskalny).
<i>week_begins_on</i>	Sunday Monday Tuesday Wednesday Thursday Friday Saturday Sunday	W przypadku przedziałów okresowych (dni na tydzień, godziny na dzień, minuty na dzień i sekundy na dzień) określa dzień, w którym rozpoczyna się tydzień.
<i>day_begins_hour</i>	<i>integer</i>	W przypadku przedziałów okresowych (godziny na dzień, minuty na dzień, sekundy na dzień) określa godzinę, o której rozpoczyna się dzień. Może być używana w kombinacji z właściwościami <i>day_begins_minute</i> i <i>day_begins_second</i> do określenia dokładnej godziny, na przykład 8:05:01. Przykład zastosowania przedstawiono poniżej.
<i>day_begins_minute</i>	<i>integer</i>	W przypadku przedziałów okresowych (godziny na dzień, minuty na dzień, sekundy na dzień) określa minutę, w której rozpoczyna się dzień (na przykład 5 w 8:05).
<i>day_begins_second</i>	<i>integer</i>	W przypadku przedziałów okresowych (godziny na dzień, minuty na dzień, sekundy na dzień) określa sekundę, w której rozpoczyna się dzień (na przykład 17 w 8:05:17).
<i>days_per_week</i>	<i>integer</i>	W przypadku przedziałów okresowych (dni na tydzień, godziny na dzień, minuty na dzień i sekundy na dzień) określa liczbę dni w tygodniu.
<i>hours_per_day</i>	<i>integer</i>	W przypadku przedziałów okresowych (godziny na dzień, minuty na dzień i sekundy na dzień) określa liczbę godzin w dniu.
<i>interval_increment</i>	1 2 3 4 5 6 10 15 20 30	W przypadku minut na dzień i sekund na dzień określa przyrost liczby minut lub sekund w każdym rekordzie.
<i>field_name_extension</i>	<i>string</i>	
<i>field_name_extension_as_prefix</i>	<i>flag</i>	

Tabela 93. Właściwości węzła *timeintervalsnode* (kontynuacja).

Właściwości węzła <i>timeintervalsnode</i>	Typ danych	Opis właściwości
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	Określa metodę agregacji zmiennej.
pad	Blank MeanOfRecentPoints True False	Określa metodę wypełniania (uzupełniania) zmiennej.
agg_mode	All Specify	Określa, czy agregować lub uzupełniać wszystkie zmienne przy użyciu funkcji domyślnych, czy też mają być określone konkretne zmienne i funkcje.

Tabela 93. Właściwości węzła `timeintervalsnode` (kontynuacja).

Właściwości węzła <code>timeintervalsnode</code>	Typ danych	Opis właściwości
<code>agg_range_default</code>	Mean Sum Mode Min Max	Określa domyślną funkcję agregacji zmiennych ciągłych.
<code>agg_set_default</code>	Mode First Last	Określa domyślną funkcję agregacji zmiennych nominalnych.
<code>agg_flag_default</code>	TrueIfAnyTrue Mode First Last	
<code>pad_range_default</code>	Blank MeanOfRecentPoints	Określa domyślną funkcję wypełniania (uzupełniania) zmiennych ciągłych.
<code>pad_set_default</code>	Blank MostRecentValue	
<code>pad_flag_default</code>	Blank True False	
<code>max_records_to_create</code>	<i>integer</i>	Określa maksymalną liczbę rekordów, jaka ma być utworzona przy uzupełnianiu szeregu.
<code>estimation_from_beginning</code>	<i>flag</i>	
<code>estimation_to_end</code>	<i>flag</i>	
<code>estimation_start_offset</code>	<i>integer</i>	
<code>estimation_num_holdouts</code>	<i>integer</i>	
<code>create_future_records</code>	<i>flag</i>	
<code>num_future_records</code>	<i>integer</i>	
<code>create_future_field</code>	<i>flag</i>	
<code>future_field_name</code>	<i>string</i>	

Właściwości węzła `transposenode`



Węzeł Transpozycja zamienia miejscami wiersze i kolumny danych, tak aby rekordy stały się zmiennymi, a zmienne rekordami.

Przykład

```
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")
```

Tabela 94. Właściwości węzła *transposenode*

Właściwości węzła <i>transposenode</i>	Typ danych	Opis właściwości
<i>transpose_method</i>	<i>enum</i>	Określa metodę transpozycji: Normalną (<i>normal</i>), CASE to VAR (<i>casetovar</i>) lub VAR to CASE (<i>varnocase</i>).
<i>transposed_names</i>	Prefix Read	Właściwości normalnej metody transpozycji. Nowe nazwy zmiennych mogą być generowane automatycznie na podstawie określonego przedrostka lub odczytywane z istniejących zmiennej w danych.
<i>prefix</i>	<i>string</i>	Właściwości normalnej metody transpozycji.
<i>num_new_fields</i>	<i>integer</i>	Właściwości normalnej metody transpozycji. Gdy stosowany jest przedrostek, określa maksymalną liczbę nowych zmiennych do utworzenia.
<i>read_from_field</i>	<i>field</i>	Właściwości normalnej metody transpozycji. Zmienna, z której odczytywane są nazwy. Instancja zmiennych musi istnieć; w przeciwnym razie przy wykonywaniu węzła zostanie zgłoszony błąd.
<i>max_num_fields</i>	<i>integer</i>	Właściwości normalnej metody transpozycji. Określa górny limit liczby nazw odczytywanych ze zmiennej, by uniknąć tworzenia zbyt dużej liczby zmiennych.
<i>transpose_type</i>	Numeric String Custom	Właściwości normalnej metody transpozycji. Domyślnie transponowane są tylko zmienne ciągłe (przedział liczbowy), ale można wybrać niestandardowy podzbiór zmiennych liczbowych lub transponować wszystkie zmienne łańcuchowe.
<i>transpose_fields</i>	<i>list</i>	Właściwości normalnej metody transpozycji. Określa zmienne, które mają być transponowane, gdy używana jest opcja <i>Custom</i> .
<i>id_field_name</i>	<i>field</i>	Właściwości normalnej metody transpozycji.
<i>index</i>	<i>field</i>	Właściwości metody transpozycji CASE to VAR (<i>casetovar</i>). Dopuszcza wiele zmiennych, które mają być używane jako zmienne indeksowe. <i>field1 ... fieldN</i>
<i>column</i>	<i>field</i>	Właściwości metody transpozycji CASE to VAR (<i>casetovar</i>). Dopuszcza wiele pól, które mają być używane jako zmienne kolumny. <i>field1 ... fieldN</i>
<i>value</i>	<i>field</i>	Właściwości metody transpozycji CASE to VAR (<i>casetovar</i>). Dopuszcza wiele pól, które mają być używane jako zmienne przewidywane. <i>field1 ... fieldN</i>
<i>id_variables</i>	<i>field</i>	Właściwości metody transpozycji VAR to CASE (<i>varnocase</i>). Dopuszcza wiele pól, które mają być używane jako zmienne identyfikacji. <i>field1 ... fieldN</i>
<i>value_variables</i>	<i>field</i>	Właściwości metody transpozycji VAR to CASE (<i>varnocase</i>). Dopuszcza wiele pól, które mają być używane jako zmienne wartości. <i>field1 ... fieldN</i>

Właściwości węzła typenode



Węzeł Typ określa metadane i właściwości zmiennej. Można na przykład określić poziom pomiaru (ciągły, nominalny, porządkowy, flaga) dla każdej zmiennej, określić opcje obsługi braków danych i systemowych braków danych, ustawić rolę zmiennej w modelowaniu, określić etykiety zmiennej i wartości oraz określić wartości zmiennej.

Przykład

```
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"]])
```

Należy zwrócić uwagę, że w niektórych przypadkach może być konieczne pełne utworzenie instancji węzła Typ, aby pozostałe węzły działały prawidłowo (np. właściwość fields from węzła Flagowanie). Wystarczy podłączyć węzeł Tabela i wykonać go, aby utworzyć instancje zmiennych:

```
tablenode = stream.createAt("table", "Table node", 150, 50)
stream.link(node, tablenode)
tablenode.run(None)
stream.delete(tablenode)
```

Tabela 95. Właściwości węzła typenode.

Właściwości węzła typenode	Typ danych	Opis właściwości
direction	Input Target Both None Partition Split Frequency RecordID	Właściwość wprowadzana określająca role zmiennych. Uwaga: Wartości In i Out są obecnie nieaktualne. Ich obsługa może zostać wycofana w przyszłej wersji produktu.
type	Range Flag Set Typeless Discrete OrderedSet Default	Poziom pomiaru zmiennej (dawniej nazywany „typem” zmiennej). Ustawienie właściwości type na Default spowoduje wyczyszczenie ustawienia parametru values, a jeśli value_mode ma wartość Specify, to otrzyma wartość Read. Jeśli value_mode ma wartość Pass lub Read, ustawienie type nie wpłynie na value_mode. Uwaga: Typy danych używane wewnętrznie różnią się od widocznych w węźle Typ. Odpowiedniki są następujące: Przedział -> Ciągły zbiór -> Nominalny zbiór uporządkowany -> Porządkowy dyskretny -> Jakościowy

Tabela 95. Właściwości węzła typenode (kontynuacja).

Właściwości węzła typenode	Typ danych	Opis właściwości
storage	Unknown String Integer Real Time Date Timestamp	Właściwość wprowadzana przeznaczona tylko do odczytu; określa typ składowania zmiennej.
check	None Nullify Coerce Discard Warn Abort	Właściwość wprowadzana określająca typ zmiennej i zasady sprawdzania zakresu.
values	[value value]	W przypadku zmiennej ciągłej pierwsza wartość to minimum, a ostatnia wartość to maksimum. W przypadku zmiennych nominalnych należy określić wszystkie wartości. W przypadku zmiennych typu flaga pierwsza wartość reprezentuje <i>false</i> , a ostatnia <i>true</i> . Nadanie wartości tej właściwości powoduje, że właściwość <i>value_mode</i> jest automatycznie ustawiana na <i>Specify</i> .
value_mode	Read Pass Read+ Current Specify	Określa sposób nadawania wartości. Należy zwrócić uwagę, że nie można nadać tej właściwości bezpośrednio wartości <i>Specify</i> ; aby użyć konkretnych wartości, należy przypisać je do właściwości <i>values</i> .
extend_values	flag	Ma zastosowanie wtedy, gdy <i>value_mode</i> ma wartość <i>Read</i> . Ustawienie <i>T</i> powoduje dodawanie nowych wczytanych wartości do istniejących wartości zmiennej. Ustawienie <i>F</i> powoduje odrzucenie istniejących wartości i zastąpienie ich nowo wczytanymi.
enable_missing	flag	Ustawienie <i>T</i> włącza rejestrowanie braków danych dla zmiennej.
missing_values	[value value ...]	Określa wartości danych oznaczające braki danych.
range_missing	flag	Określa, czy dla zmiennej zdefiniowany jest przedział braków danych (wartości pustych).
missing_lower	string	Gdy <i>range_missing</i> ma wartość <i>true</i> , określa dolną granicę przedziału braków danych.
missing_upper	string	Gdy <i>range_missing</i> ma wartość <i>true</i> , określa górną granicę przedziału braków danych.
null_missing	flag	Gdy ta właściwość jest ustawiona na <i>T</i> , wartości <i>null</i> (niezdefiniowane wartości są w programie wyświetlane jako <i>\$null\$</i>) są uznawane za braki danych.
whitespace_missing	flag	Gdy ta właściwość jest ustawiona na <i>T</i> , wartości zawierające tylko białe znaki (spacje, znaki tabulacji i znaki nowego wiersza) są uznawane za braki danych.
description	string	Określa opis zmiennej.
value_labels	[[Value LabelString] [Value LabelString] ...]	Służy do określania etykiet par wartości.

Tabela 95. Właściwości węzła typenode (kontynuacja).

Właściwości węzła typenode	Typ danych	Opis właściwości
display_places	integer	Określa liczbę miejsc dziesiętnych obowiązującą przy wyświetlaniu zmiennej (dotyczy tylko zmiennych o typie składowania REAL). Wartość -1 powoduje użycie ustawienia domyślnego dla strumienia.
export_places	integer	Określa liczbę miejsc dziesiętnych obowiązującą przy eksportowaniu zmiennej (dotyczy tylko zmiennych o typie składowania REAL). Wartość -1 powoduje użycie ustawienia domyślnego dla strumienia.
decimal_separator	DEFAULT PERIOD COMMA	Określa separator dziesiętny dla zmiennej (dotyczy tylko zmiennych o typie składowania REAL).
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	Określa format daty dla zmiennej (dotyczy tylko zmiennych o typie składowania DATE lub TIMESTAMP).
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"	Określa format czasu dla zmiennej (dotyczy tylko zmiennych o typie składowania TIME lub TIMESTAMP).

Tabela 95. Właściwości węzła typenode (kontynuacja).

Właściwości węzła typenode	Typ danych	Opis właściwości
number_format	DEFAULT STANDARD SCIENTIFIC CURRENCY	Określa format wyświetlania wartości liczbowych zmiennej.
standard_places	<i>integer</i>	Określa liczbę miejsc dziesiętnych obowiązującą przy wyświetlaniu zmiennej w formacie standardowym. Wartość -1 powoduje użycie ustawienia domyślnego dla strumienia. Należy zwrócić uwagę, że istniejący parametr <code>display_places</code> także zostanie zmieniony, ale w bieżącej wersji produktu jest nieaktualny.
scientific_places	<i>integer</i>	Określa liczbę miejsc dziesiętnych obowiązującą przy wyświetlaniu zmiennej w formacie naukowym. Wartość -1 powoduje użycie ustawienia domyślnego dla strumienia.
currency_places	<i>integer</i>	Określa liczbę miejsc dziesiętnych obowiązującą przy wyświetlaniu zmiennej w formacie walutowym. Wartość -1 powoduje użycie ustawienia domyślnego dla strumienia.
grouping_symbol	DEFAULT BRAK LOCALE PERIOD COMMA SPACE	Określa symbol grupowania cyfr dla zmiennej.
column_width	<i>integer</i>	Określa szerokość kolumny dla zmiennej. Wartość -1 powoduje ustawienie szerokości kolumny na Auto.
justify	AUTO CENTER LEFT RIGHT	Określa justowanie kolumny dla zmiennej.
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	Ta właściwość wprowadzana jest podobna do właściwości <code>type</code> w tym sensie, że może służyć do definiowania poziomu pomiaru zmiennej. Jednak w skryptach w języku Python do funkcji ustawiającej można też przekazać jedną z wartości <code>MeasureType</code> natomiast funkcja pobierająca zawsze zwraca wartości <code>MeasureType</code> .
collection_measure	Range / MeasureType.RANGE Flag / MeasureType.FLAG Set / MeasureType.SET OrderedSet / MeasureType.ORDERED_SET Typeless / MeasureType.TYPELESS	W przypadku zmiennych <code>collection</code> (listy o głębokości 0) ta właściwość definiuje typ pomiaru związany z bazowymi wartościami.
geo_type	Point MultiPoint LineString MultiLineString Polygon MultiPolygon	W przypadku zmiennych geoprzestrzennych ta właściwość definiuje typ obiektu reprezentowanego przez zmienną. Powinien być on zgodny z głębokością listy wartości.

Tabela 95. Właściwości węzła `typenode` (kontynuacja).

Właściwości węzła <code>typenode</code>	Typ danych	Opis właściwości
<code>has_coordinate_system</code>	<i>boolean</i>	W przypadku zmiennych geoprzestrzennych ta właściwość określa, czy zmienna ma układ współrzędnych.
<code>coordinate_system</code>	<i>string</i>	W przypadku zmiennych geoprzestrzennych ta właściwość definiuje układ współrzędnych dla zmiennej.
<code>custom_storage_type</code>	Unknown / MeasureType.UNKNOWN String / MeasureType.STRING Integer / MeasureType.INTEGER Real / MeasureType.REAL Time / MeasureType.TIME Date / MeasureType.DATE Timestamp / MeasureType.TIMESTAMP List / MeasureType.LIST	Ta właściwość wprowadzana jest podobna do właściwości <code>custom_storage</code> w tym sensie, że może służyć do definiowania przesłoniętego typu składowania zmiennej. Jednak w skryptach w języku Python do funkcji ustawiającej można też przekazać jedną z wartości <code>StorageType</code> natomiast funkcja pobierająca zawsze zwraca wartości <code>StorageType</code> .
<code>custom_list_storage_type</code>	String / MeasureType.STRING Integer / MeasureType.INTEGER Real / MeasureType.REAL Time / MeasureType.TIME Date / MeasureType.DATE Timestamp / MeasureType.TIMESTAMP	W przypadku zmiennych typu lista ta właściwość wprowadzana określa typ składowania wartości.
<code>custom_list_depth</code>	<i>integer</i>	W przypadku zmiennych typu lista ta właściwość wprowadzana określa głębokość zmiennej
<code>max_list_length</code>	<i>integer</i>	Tylko dla danych z <i>Geoprzestrzennym</i> lub <i>Przedziałowym</i> poziomem pomiaru. Należy ustawić maksymalną długość listy, ustawiając dozwoloną dla listy liczbę elementów.
<code>max_string_length</code>	<i>integer</i>	Tylko dla danych <i>beztypowych</i> podczas generowania kodu SQL w celu utworzenia tabeli. Należy wprowadzić największy łańcuch danych; umożliwi to wygenerowanie w tabeli kolumny wystarczająco dużej do pomieszczenia łańcucha.

Rozdział 12. Właściwości węzłów wykresów

Właściwości wspólne węzła wykresu

W tej sekcji opisano właściwości węzłów wykresów, w tym właściwości wspólne i właściwości charakterystyczne dla poszczególnych typów wykresów.

Tabela 96. Wspólne właściwości węzłów wykresów

Wspólne właściwości węzłów wykresów	Typ danych	Opis właściwości
title	<i>string</i>	Określa tytuł. Przykład: "To jest tytuł."
caption	<i>string</i>	Określa podpis. Przykład: "To jest podpis."
output_mode	Screen File	Określa, czy wynik wykonania węzła wykresu ma być wyświetlany, czy zapisywany w pliku.
output_format	BMP JPEG PNG HTML output (.cou)	Określa typ wyników. Typy wyników dozwolone dla poszczególnych typów węzłów są różne.
full_filename	<i>string</i>	Określa ścieżkę i nazwę pliku, w którym ma być zapisany wynik wykonania węzła.
use_graph_size	<i>flag</i>	Określa, czy wykres ma być skalowany jawnie, na podstawie poniższych właściwości width i height. Wpływa tylko na wykresy wyświetlane na ekranie. Nieodstępna w węźle rozkładu.
graph_width	<i>number</i>	Gdy use_graph_size ma wartość True, sets określa szerokość wykresu w pikselach.
graph_height	<i>number</i>	Gdy use_graph_size ma wartość True, sets określa wysokość wykresu w pikselach.

Wyłączanie pól opcjonalnych

Pola opcjonalne, takie jak nakładki na wykresy, można wyłączyć, nadając właściwości wartość "" (łańcuch pusty), co ilustruje poniższy przykład:

```
plotnode.setPropertyValue("color_field", "")
```

Określanie kolorów

Kolory tytułów, podpisów, tła i etykiet można określać w postaci łańcuchów szesnastkowych rozpoczynających się od symbolu krzyżyka (#). Na przykład, aby tło wykresu miało kolor błękitu nieba, należy użyć instrukcji:

```
mygraphnode.setPropertyValue("graph_background", "#87CEEB")
```

Dwie pierwsze cyfry, 87, to składowa czerwona; dwie środkowe litery, CE, to składowa zielona, a dwie ostatnie litery, EB, to składowa niebieska. Każdy element może przyjmować wartość z zakresu 0–9 lub A–F. Łącznie wartości opisują kolor RGB (składowa czerwona, zielona i niebieska).

Uwaga: Określając kolory w formacie RGB, można skorzystać z selektora zmiennych w interfejsie użytkownika, aby wybrać właściwy kod koloru. Wystarczy zatrzymać wskaźnik nad kolorem, aby wywołać podpowiedź z właściwą informacją.

Właściwości węzła collectionnode



Węzeł Zbiór przedstawia rozkład wartości dla jednej zmiennej numerycznej względem wartości innej zmiennej. (Tworzy wykresy podobne do histogramów). Jest przydatny do prezentacji zmiennej, której wartości zmieniają się w czasie. Na wykresie 3-W można dodać oś symboliczną odzwierciedlającą rozkład według kategorii.

Przykład

```
node = stream.create("collection", "My node")
# Karta "Wykres"
node.setPropertyValue("three_D", True)
node.setPropertyValue("collect_field", "Drug")
node.setPropertyValue("over_field", "Age")
node.setPropertyValue("by_field", "BP")
node.setPropertyValue("operation", "Sum")
# "Nałożenie" – sekcja
node.setPropertyValue("color_field", "Drug")
node.setPropertyValue("panel_field", "Sex")
node.setPropertyValue("animation_field", "")
# Karta "Opcje"
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)
```

Tabela 97. Właściwości węzła collectionnode

Właściwości węzła collectionnode	Typ danych	Opis właściwości
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	
by_label_auto	flag	
by_label	string	
operation	Sum Mean Min Max SDev	
color_field	string	
panel_field	string	
animation_field	string	
range_mode	Automatic UserDefined	
range_min	number	

Tabela 97. Właściwości węzła *collectionnode* (kontynuacja)

Właściwości węzła <i>collectionnode</i>	Typ danych	Opis właściwości
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>	Standardowe kolory wykresu są opisane na początku tej sekcji.
page_background	<i>color</i>	Standardowe kolory wykresu są opisane na początku tej sekcji.

Właściwości węzła *distributionnode*



Węzeł Rozkład przedstawia wystąpienia wartości symbolicznych (jakościowych), takich jak typ kredytu lub płeć. Zwykle węzeł rozkładu jest używany do przedstawienia dysproporcji danych, które można później naprawić za pomocą węzła zrównoważenia przed utworzeniem modelu.

Przykład

```
node = stream.create("distribution", "My node")
# Karta "Wykres"
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)
```

Tabela 98. Właściwości węzła *distributionnode*

Właściwości węzła <i>distributionnode</i>	Typ danych	Opis właściwości
plot	SelectedFields Flags	
x_field	<i>field</i>	
color_field	<i>field</i>	Zmienna nakładkowa.
normalize	<i>flag</i>	
sort_mode	ByOccurence Alphabetic	
use_proportional_scale	<i>flag</i>	

Właściwości węzła evaluationnode



Węzeł Ewaluacja pomaga w dokonaniu oceny i porównaniu modeli predykcyjnych. Na wykresie ewaluacyjnym przedstawiane jest, w jakim stopniu modele przewidują określone wyniki. Rekordy sortowane są na podstawie wartości przewidywanej i poziomu ufności predykcji. Rekordy są dzielone na grupy o jednakowej wielkości (**kwantyle**), a następnie tworzone są wykresy wartości wg kryterium biznesowego dla każdego kwantyla, od najwyższego do najniższego. Modele wielokrotnie prezentowane są jako osobne linie na wykresie.

Przykład

```
node = stream.create("evaluation", "My node")
# Karta "Wykres"
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")
```

Tabela 99. Właściwości węzła evaluationnode.

Właściwości węzła evaluationnode	Typ danych	Opis właściwości
chart_type	Gains Response Lift Profit ROI ROC	
inc_baseline	flag	
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	

Tabela 99. Właściwości węzła *evaluationnode* (kontynuacja).

Właściwości węzła <i>evaluationnode</i>	Typ danych	Opis właściwości
<i>n_tile</i>	Quartiles Quintiles Deciles Vingtiles Percentiles 1000-tiles	
<i>cumulative</i>	<i>flag</i>	
<i>style</i>	Line Point	
<i>point_type</i>	Rectangle Dot Triangle Hexagon Plus Pentagon Star BowTie HorizontalDash VerticalDash IronCross Factory House Cathedral OnionDome ConcaveTriangle OblateGlobe CatEye FourSidedPillow RoundRectangle Fan	
<i>export_data</i>	<i>flag</i>	
<i>data_filename</i>	<i>string</i>	
<i>delimiter</i>	<i>string</i>	
<i>new_line</i>	<i>flag</i>	
<i>inc_field_names</i>	<i>flag</i>	
<i>inc_best_line</i>	<i>flag</i>	
<i>inc_business_rule</i>	<i>flag</i>	
<i>business_rule_condition</i>	<i>string</i>	
<i>plot_score_fields</i>	<i>flag</i>	
<i>score_fields</i>	<i>[field1 ... fieldN]</i>	
<i>target_field</i>	<i>field</i>	
<i>use_hit_condition</i>	<i>flag</i>	
<i>hit_condition</i>	<i>string</i>	
<i>use_score_expression</i>	<i>flag</i>	
<i>score_expression</i>	<i>string</i>	
<i>caption_auto</i>	<i>flag</i>	

Właściwości węzła graphboardnode



Węzeł Wizualizacja oferuje wiele różnych typów wykresów w pojedynczym węźle. Korzystając z tego węzła można wybrać zmienne zawierające dane, dla których ma zostać przeprowadzona eksploracja, a następnie wybrać wykres spośród tych, które zostały udostępnione dla wybranych danych. Węzeł automatycznie filtruje wszelkie typy wykresów, które nie współpracowałyby z wybranymi polami.

Uwaga: W przypadku określenia wartości właściwości niedozwolonej dla danego typu wykresu (na przykład właściwości `y_field` dla histogramu) właściwość zostanie zignorowana.

Uwaga: W interfejsie użytkownika na karcie Szczegółowe różnych typów wykresów znajduje się pole

Podsumowanie; obecnie nie jest ono obsługiwane w skryptach.

Przykład

```
node = stream.create("graphboard", "My node")
node.setPropertyValue("graph_type", "Line")
node.setPropertyValue("x_field", "K")
node.setPropertyValue("y_field", "Na")
```

Tabela 100. Właściwości węzła graphboardnode

Właściwości węzła graphboard	Typ danych	Opis właściwości
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	Określa typ wykresu.
x_field	<i>field</i>	Określa niestandardową etykietę osi x. Dostępna tylko dla etykiet.
y_field	<i>field</i>	Określa niestandardową etykietę osi y. Dostępna tylko dla etykiet.
z_field	<i>field</i>	Używana w niektórych wykresach trójwymiarowych.
color_field	<i>field</i>	Używana w mapach natężeń.
size_field	<i>field</i>	Używana na wykresach bąbelkowych.

Tabela 100. Właściwości węzła graphboardnode (kontynuacja)

Właściwości węzła graphboard	Typ danych	Opis właściwości
categories_field	field	
values_field	field	
rows_field	field	
columns_field	field	
fields	field	
start_longitude_field	field	Używana ze strzałkami na mapie referencyjnej.
end_longitude_field	field	
start_latitude_field	field	
end_latitude_field	field	
data_key_field	field	Używana na różnych mapach.
panelrow_field	string	
panelcol_field	string	
animation_field	string	
longitude_field	field	Używana ze współrzędnymi na mapach.
latitude_field	field	
map_color_field	field	

Właściwości węzła histogramnode



Węzeł Histogram pokazuje wystąpienia wartości zmiennych numerycznych. Często używany jest do eksploracji danych przed przystąpieniem do manipulowania i budowy modelu. Podobnie jak w przypadku węzła rozkładu, węzeł histogramu często ujawnia dysproporcje danych.

Przykład

```
node = stream.create("histogram", "My node")
# Karta "Wykres"
node.setPropertyValue("field", "Drug")
node.setPropertyValue("color_field", "Drug")
node.setPropertyValue("panel_field", "Sex")
node.setPropertyValue("animation_field", "")
# Karta "Opcje"
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)
```

Tabela 101. Właściwości węzła histogramnode

Właściwości węzła histogramnode	Typ danych	Opis właściwości
field	field	
color_field	field	

Tabela 101. Właściwości węzła histogramnode (kontynuacja)

Właściwości węzła histogramnode	Typ danych	Opis właściwości
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>	Standardowe kolory wykresu są opisane na początku tej sekcji.
page_background	<i>color</i>	Standardowe kolory wykresu są opisane na początku tej sekcji.
normal_curve	<i>flag</i>	Wskazuje, czy w wynikach powinna być uwidoczniona krzywa rozkładu normalnego.

Właściwości węzła mapvisualization



Węzeł Wizualizacja na mapie może akceptować wiele połączeń wejściowych i wyświetlać dane geoprzestrzenne na mapie w formie szeregu warstw. Każda warstwa stanowi pojedynczą zmienną geoprzestrzenną; na przykład warstwa podstawowa może być mapą kraju, a nad nią może znajdować się jedna warstwa dróg, jedna warstwa rzek i jedna warstwa miejscowości.

Tabela 102. Właściwości węzła mapvisualization

Właściwości węzła mapvisualization	Typ danych	Opis właściwości
tag	<i>string</i>	Ustawia nazwę znacznika dla wejścia. Domyślny znacznik jest liczbą opartą na kolejności, według której dane wejściowe zostały podłączone do węzła (pierwszy znacznik połączenia to 1, drugi znacznik połączenia to 2 itd).

Tabela 102. Właściwości węzła *mapvisualization* (kontynuacja)

Właściwości węzła <i>mapvisualization</i>	Typ danych	Opis właściwości
<i>layer_field</i>	<i>field</i>	<p>Wybiera, która zmienna geoprzestrzenna z zestawu danych jest wyświetlana jako warstwa na mapie. Wybór domyślny bazuje na następującym porządku sortowania:</p> <ul style="list-style-type: none"> • Pierwsza — punkt • Łańcuch • Wielokąt • Zbiór punktów • Multiłańcuch • Ostatnia — Multiwielokąt <p>Jeśli istnieją dwie zmienne z takim samym typem pomiaru, pierwsza zmienna w kolejności alfabetycznej (według nazwy) zostanie wybrana jako domyślna.</p>
<i>color_type</i>	<i>boolean</i>	Określa kolor standardowy stosowany do wszystkich funkcji zmiennej geoprzestrzennej lub zmienną nałożenia, która zmienia kolor funkcji w zależności od wartości z innej zmiennej w zbiorze danych. Możliwe wartości to <i>standard</i> lub <i>overlay</i> . Domyślnie wybrana to <i>standard</i> .
<i>color</i>	<i>string</i>	<p>Jeśli została wybrana opcja <i>standard</i> dla <i>color_type</i>, rozwijana lista zawiera taką samą paletę kolorów jak porządek kolorów kategorii na wykresie w opcjach użytkownika na karcie Pokaż.</p> <p>Domyślną wartością jest kolor kategorii wykresu 1.</p>
<i>color_field</i>	<i>field</i>	Jeśli została wybrana opcja <i>overlay</i> dla <i>color_type</i> , lista rozwijana zawiera wszystkie zmienne z tego samego zestawu danych co zmienna geoprzestrzenna wybrana jako warstwa.
<i>symbol_type</i>	<i>boolean</i>	Określa symbol standardowy stosowany do wszystkich rekordów zmiennej geoprzestrzennej lub symbol nałożenia, który zmienia ikonę symboli w zależności od wartości z innej zmiennej w zbiorze danych. Możliwe wartości to <i>standard</i> lub <i>overlay</i> . Domyślnie wybrana to <i>standard</i> .
<i>symbol</i>	<i>string</i>	Jeśli została wybrana opcja <i>standard</i> dla <i>symbol_type</i> , lista rozwijana zawiera wybór symboli, który może służyć do wyświetlania punktów na mapie.
<i>symbol_field</i>	<i>field</i>	Jeśli została wybrana opcja <i>overlay</i> dla <i>symbol_type</i> , lista rozwijana zawiera wszystkie zmienne nominalne, porządkowe lub jakościowe z tego samego zestawu danych co zmienna geoprzestrzenna wybrana jako warstwa.
<i>size_type</i>	<i>boolean</i>	Określa rozmiar standardowy stosowany do wszystkich rekordów zmiennej geoprzestrzennej lub rozmiar nałożenia, który zmienia rozmiar symbolu ikony bądź grubość linii w zależności od wartości z innej zmiennej w zbiorze danych. Możliwe wartości to <i>standard</i> lub <i>overlay</i> . Domyślnie wybrana to <i>standard</i> .

Tabela 102. Właściwości węzła *mapvisualization* (kontynuacja)

Właściwości węzła <i>mapvisualization</i>	Typ danych	Opis właściwości
<i>size</i>	<i>string</i>	Jeśli została wybrana opcja <i>standard</i> dla <i>size_type</i> , dla <i>point</i> lub <i>multipoint</i> , rozwijana lista umożliwia wybór wielkości dla wybranego symbolu. Dla komendy <i>linestring</i> lub <i>multilinestring</i> lista rozwijana umożliwia wybór grubości linii.
<i>size_field</i>	<i>field</i>	Jeśli została wybrana opcja <i>overlay</i> dla <i>size_type</i> , lista rozwijana zawiera wszystkie zmienne z tego samego zestawu danych co zmienna geoprzestrzenna wybrana jako warstwa.
<i>transp_type</i>	<i>boolean</i>	Określa standardową przezroczystość stosowaną do wszystkich rekordów zmiennej geoprzestrzennej lub przezroczystość nałożenia, która zmienia poziom przezroczystości symbolu, linii lub wielokąta w zależności od wartości z innej zmiennej w zbiorze danych. Możliwe wartości to <i>standard</i> lub <i>overlay</i> . Domyślnie wybrana to <i>standard</i> .
<i>transp</i>	<i>integer</i>	<p>Jeśli została wybrana opcja <i>standard</i> dla <i>transp_type</i>, lista rozwijana umożliwia wybór poziomu przezroczystości, począwszy od 0% (nieprzezroczystość) po wartość 100% (przezroczysty). Poszczególne poziomy zwiększają się o 10%. Ustawia przezroczystość punktów, linii i wielokątów na mapie.</p> <p>Jeśli została wybrana opcja <i>overlay</i> dla <i>size_type</i>, lista rozwijana zawiera wszystkie zmienne z tego samego zestawu danych co zmienna geoprzestrzenna wybrana jako warstwa.</p> <p>Wartość domyślna dla <i>points</i>, <i>multipoints</i>, <i>linestrings</i> oraz <i>multilinestrings</i>, <i>polygons</i> i <i>multipolygons</i> (stanowiących dolną warstwę) wynosi 0%. Wartość domyślna dla <i>polygons</i> i <i>multipolygons</i> nie stanowiących dolnej warstwy wynosi 50% (w celu uniknięcia przesłaniania warstw poniżej tych wielokątów).</p>
<i>transp_field</i>	<i>field</i>	Jeśli została wybrana opcja <i>overlay</i> dla <i>transp_type</i> , lista rozwijana zawiera wszystkie zmienne z tego samego zestawu danych co zmienna geoprzestrzenna wybrana jako warstwa.
<i>data_label_field</i>	<i>field</i>	Określa zmienną, która ma być używana jako etykieta danych na mapie. Na przykład, jeśli warstwą, do której zastosowano to ustawienie, jest warstwa wielokąta, to etykietą danych może być zmienna <i>name</i> — zawierająca nazwę każdego wielokąta. A zatem wybór zmiennej <i>name</i> w tym miejscu spowoduje, że te nazwy zostaną wyświetlone na mapie.
<i>use_hex_binning</i>	<i>boolean</i>	Włącza grupowanie sześciokątne i wszystkie rozwijane listy agregacji. To ustawienie jest domyślnie wyłączone.

Tabela 102. Właściwości węzła *mapvisualization* (kontynuacja)

Właściwości węzła mapvisualization	Typ danych	Opis właściwości
color_aggregation i transp_aggregation	<i>string</i>	<p>Po wybraniu zmiennej nałożenia dla warstwy punktów za pomocą grupowania sześciokątnej wszystkie wartości dla tej zmiennej muszą być zagregowane dla wszystkich punktów w obrębie sześciokąta. Z tego powodu należy określić funkcję agregującą dla dowolnej zmiennej nałożenia, którą chcesz zastosować dla mapy.</p> <p>Dostępne funkcje agregacji to:</p> <p>Ciągła (składowanie liczb rzeczywistych lub całkowitych):</p> <ul style="list-style-type: none"> • Suma • Średnia • Minimum • Maksimum • Mediana • 1 Kwartyl • 3 Kwartyl <p>Ciągły (składowanie godziny, daty, znacznika czasu):</p> <ul style="list-style-type: none"> • Średnia • Minimum • Maksimum <p>Nominalna/Jakościowa:</p> <ul style="list-style-type: none"> • Dominanta • Minimum • Maksimum <p>Flaga:</p> <ul style="list-style-type: none"> • Prawda, gdy jakaś jest prawdziwa • Fałsz, jeśli choć jeden fałsz
custom_storage	<i>string</i>	<p>Ustawia ogólny typ pamięci masowej dla zmiennej. Wartością domyślną jest List. Jeśli zostanie podana wartość List, opcje custom_value_storage i list_depth są wyłączone.</p>
custom_value_storage	<i>string</i>	<p>Ustawia typy składowania dla elementów na liście, a nie dla zmiennej jako do całości. Wartością domyślną jest Real.</p>

Tabela 102. Właściwości węzła *mapvisualization* (kontynuacja)

Właściwości węzła <i>mapvisualization</i>	Typ danych	Opis właściwości
<code>list_depth</code>	<i>integer</i>	<p>Ustawia głębokość zmiennej typu lista. Wymagana głębokość zależy od typu zmiennej geoprzestrzennej, obowiązują następujące kryteria:</p> <ul style="list-style-type: none"> • Punkt - 0 • Łańcuch - 1 • Wielokąt - 2 • Multipunkt - 1 • Multiłańcuch - 2 • Multiwielokąt - 3 <p>Użytkownik musi znać typ zmiennej geoprzestrzennej przekształcanej z powrotem do listy oraz wymaganą głębokość dla tego typu zmiennej. Niepoprawne ustawienia uniemożliwiają używanie zmiennej.</p> <p>Wartość domyślna to 0, minimalna to 0, a maksymalna 10.</p>

Właściwości węzła *multiplotnode*



Węzeł Wykres wielokrotny tworzy wykres zawierający wiele zmiennych *Y* dla jednej zmiennej *X*. Zmienne *Y* są wykresane jako kolorowe linie, a każda z nich jest równoważna węzłowi wykresu ze stylem ustawionym na wartość **Liniiowy** i trybem osi *X* ustawionym na **Sortuj**. Wykresy wielokrotne są przydatne do zbadania wahań kilku zmiennych w czasie.

Przykład

```
node = stream.create("multiplot", "My node")
# Karta "Wykres"
node.setPropertyValue("x_field", "Age")
node.setPropertyValue("y_fields", ["Drug", "BP"])
node.setPropertyValue("panel_field", "Sex")
# "Nałożenie" – sekcja
node.setPropertyValue("animation_field", "")
node.setPropertyValue("tooltip", "BP")
node.setPropertyValue("normalize", True)
node.setPropertyValue("use_overlay_expr", False)
node.setPropertyValue("overlay_expression", "BP")
node.setPropertyValue("records_limit", 500)
node.setPropertyValue("if_over_limit", "PlotSample")
```

Tabela 103. Właściwości węzła *multiplotnode*

Właściwości węzła <i>multiplotnode</i>	Typ danych	Opis właściwości
<code>x_field</code>	<i>field</i>	
<code>y_fields</code>	<i>list</i>	
<code>panel_field</code>	<i>field</i>	
<code>animation_field</code>	<i>field</i>	
<code>normalize</code>	<i>flag</i>	
<code>use_overlay_expr</code>	<i>flag</i>	

Tabela 103. Właściwości węzła *multiplotnode* (kontynuacja)

Właściwości węzła <i>multiplotnode</i>	Typ danych	Opis właściwości
<code>overlay_expression</code>	<i>string</i>	
<code>records_limit</code>	<i>number</i>	
<code>if_over_limit</code>	PlotBins PlotSample PlotAll	
<code>x_label_auto</code>	<i>flag</i>	
<code>x_label</code>	<i>string</i>	
<code>y_label_auto</code>	<i>flag</i>	
<code>y_label</code>	<i>string</i>	
<code>use_grid</code>	<i>flag</i>	
<code>graph_background</code>	<i>color</i>	Standardowe kolory wykresu są opisane na początku tej sekcji.
<code>page_background</code>	<i>color</i>	Standardowe kolory wykresu są opisane na początku tej sekcji.

Właściwości węzła *plotnode*



Węzeł Wykres przedstawia relacje pomiędzy zmiennymi numerycznymi. Wykres można utworzyć na podstawie dwóch punktów (wykres rozrzutu) lub linii.

Przykład

```
node = stream.create("plot", "My node")
# Karta "Wykres"
node.setPropertyValue("three_D", True)
node.setPropertyValue("x_field", "BP")
node.setPropertyValue("y_field", "Cholesterol")
node.setPropertyValue("z_field", "Drug")
# "Nałożenie" – sekcja
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")
# Karta "Wynik"
node.setPropertyValue("output_mode", "File")
node.setPropertyValue("output_format", "JPEG")
node.setPropertyValue("full_filename", "C:/temp/graph_output/plot_output.jpeg")
```

Tabela 104. Właściwości węzła *plotnode*.

Właściwości węzła <i>plotnode</i>	Typ danych	Opis właściwości
<code>x_field</code>	<i>field</i>	Określa niestandardową etykietę osi <i>x</i> . Dostępna tylko dla etykiet.
<code>y_field</code>	<i>field</i>	Określa niestandardową etykietę osi <i>y</i> . Dostępna tylko dla etykiet.

Tabela 104. Właściwości węzła plotnode (kontynuacja).

Właściwości węzła plotnode	Typ danych	Opis właściwości
three_D	<i>flag</i>	Określa niestandardową etykietę osi y. Dostępna tylko dla etykiet na wykresach trójwymiarowych.
z_field	<i>field</i>	
color_field	<i>field</i>	Zmienna nakładkowa.
size_field	<i>field</i>	
shape_field	<i>field</i>	
panel_field	<i>field</i>	Określa zmienną nominalną lub typu flaga używaną przy tworzeniu osobnych wykresów poszczególnych kategorii. Wykresy są zbierane w jednym oknie wyników.
animation_field	<i>field</i>	Określa zmienną nominalną lub typu flaga używaną do zobrazowania kategorii wartości danych poprzez utworzenie serii wykresów wyświetlanych kolejno w ramach animacji.
transp_field	<i>field</i>	Określa zmienną służącą do zobrazowania kategorii wartości danych przy użyciu różnych poziomów przezroczystości. Niedostępna w przypadku wykresów liniowych.
overlay_type	None Smoother Function	Określa, czy wyświetlana jest funkcja nakładkowa, czy też ma być stosowane wygładzanie LOESS.
overlay_expression	<i>string</i>	Określa wyrażenie używane, gdy overlay_type ma wartość Function.
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	
x_mode	Sort Overlay AsRead	
x_range_mode	Automatic UserDefined	
x_range_min	<i>number</i>	
x_range_max	<i>number</i>	

Tabela 104. Właściwości węzła plotnode (kontynuacja).

Właściwości węzła plotnode	Typ danych	Opis właściwości
y_range_mode	Automatic UserDefined	
y_range_min	number	
y_range_max	number	
z_range_mode	Automatic UserDefined	
z_range_min	number	
z_range_max	number	
jitter	flag	
records_limit	number	
if_over_limit	PlotBins PlotSample PlotAll	
x_label_auto	flag	
x_label	string	
y_label_auto	flag	
y_label	string	
z_label_auto	flag	
z_label	string	
use_grid	flag	
graph_background	color	Standardowe kolory wykresu są opisane na początku tej sekcji.
page_background	color	Standardowe kolory wykresu są opisane na początku tej sekcji.
use_overlay_expr	flag	Nieaktualna, zastąpiona przez overlay_type.

Właściwości węzła timeplotnode



Węzeł Sekwencyjny wyświetla jeden lub więcej zbiorów danych szeregów czasowych. Zwykle najpierw używa się węzła Przedziały czasowe do utworzenia zmiennej *TimeLabel*, która posłuży do generowania etykiet osi *x*.

Przykład

```
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)
# Ustawienia wyglądu
node.setPropertyValue("symbol_size", 2.0)
```


Tabela 105. Właściwości węzła `timeplotnode`.

Właściwości węzła <code>timeplotnode</code>	Typ danych	Opis właściwości
<code>plot_series</code>	Series Models	
<code>use_custom_x_field</code>	<i>flag</i>	
<code>x_field</code>	<i>field</i>	
<code>y_fields</code>	<i>list</i>	
<code>panel</code>	<i>flag</i>	
<code>normalize</code>	<i>flag</i>	
<code>line</code>	<i>flag</i>	
<code>points</code>	<i>flag</i>	
<code>point_type</code>	Rectangle Dot Triangle Hexagon Plus Pentagon Star BowTie HorizontalDash VerticalDash IronCross Factory House Cathedral OnionDome ConcaveTriangle OblateGlobe CatEye FourSidedPillow RoundRectangle Fan	
<code>smoother</code>	<i>flag</i>	Funkcje wygładzające można dodawać do wykresu tylko wtedy, gdy <code>panel</code> ma wartość <code>True</code> .
<code>use_records_limit</code>	<i>flag</i>	
<code>records_limit</code>	<i>integer</i>	
<code>symbol_size</code>	<i>number</i>	Określa wielkość symbolu.
<code>panel_layout</code>	Horizontal Vertical	

Właściwości węzła `webnode`



Węzeł sieciowy obrazuje siłę relacji między wartościami dwóch lub większej liczby zmiennych symbolicznych (jakościowych). Na wykresie linie o różnej szerokości odzwierciedlają siłę połączenia. Węzła sieciowego można użyć na przykład do eksploracji relacji między zakupami różnych towarów w sklepie internetowym.

Przykład

```
node = stream.create("web", "My node")
# Karta "Wykres"
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)
# Karta "Opcje"
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")

```

Tabela 106. Właściwości węzła webnode

Właściwości węzła webnode	Typ danych	Opis właściwości
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>	
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>	Standardowe kolory wykresu są opisane na początku tej sekcji.
symbol_size	<i>number</i>	Określa wielkość symbolu.

Rozdział 13. Właściwości węzłów modelowania

Wspólne właściwości węzłów modelowania

Poniższe właściwości są wspólne dla niektórych lub wszystkich węzłów modelowania. Wszelkie wyjątki odnotowano w opisach poszczególnych węzłów.

Tabela 107. Wspólne właściwości węzłów modelowania

Właściwość	Wartości	Opis właściwości
custom_fields	<i>flag</i>	Wartość true umożliwia określenie zmiennej przewidywanej, wejściowej i innych zmiennych dla bieżącego węzła. Wartość false powoduje użycie bieżących ustawień z wcześniejszego węzła Typ.
target lub targets	<i>field</i> lub [<i>field1 ... fieldN</i>]	Określa jedną zmienną przewidywaną lub wiele zmiennych przewidywanych, w zależności od typu modelu.
inputs	[<i>field1 ... fieldN</i>]	Zmienne wejściowe lub predyktory używane przez model.
partition	<i>field</i>	
use_partitioned_data	<i>flag</i>	Jeśli zdefiniowano zmienną dzielącą na podzbiory, ta opcja umożliwia użycie podczas budowania modelu wyłącznie danych z podzbioru uczącego.
use_split_data	<i>flag</i>	
splits	[<i>field1 ... fieldN</i>]	Określa zmienną lub zmienne używane do modelowania podziałów. Odnosi skutek tylko wtedy, gdy <i>use_split_data</i> ma wartość True.
use_frequency	<i>flag</i>	W opisach poszczególnych typów modeli wskazano, czy modele danego typu korzystają ze zmiennych wagi i częstości.
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>	Niestandardowa nazwa nowego modelu.
mode	Simple Expert	

właściwości anomalydetectionnode węzła



Węzeł Wykrywanie anomalii umożliwia identyfikację nietypowych obserwacji lub wartości odstających, które są niezgodne z wzorcami dla „normalnych” danych. Korzystając z tego węzła, można zidentyfikować wartości odstające nawet, jeśli nie pasują one do żadnego z wcześniej znanych wzorców oraz jeśli brak pewności co do charakteru poszukiwanych danych.

Przykład

```
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)
```

Tabela 108. właściwości anomalydetectionnode węzła

Właściwości węzła anomalydetectionnode	Wartości	Opis właściwości
inputs	<i>[field1 ... fieldN]</i>	Modele Wykrywanie anomalii filtrują rekordy na podstawie określonych zmiennych wejściowych. Nie używają zmiennej przewidywanej. Nie używają także zmiennych wagi i częstości. Więcej informacji można znaleźć w temacie “Wspólne właściwości węzłów modelowania” na stronie 173.
mode	Expert Simple	
anomaly_method	IndexLevel PerRecords NumRecords	Określa metodę używaną do określania wartości odcięcia przy oznaczaniu rekordów jako anomalii.
index_level	<i>number</i>	Określa minimalną wartość odcięcia używaną do oznaczania anomalii.
percent_records	<i>number</i>	Ustawia próg oznaczania rekordów jako odsetek rekordów w danych uczących.
num_records	<i>number</i>	Ustawia próg oznaczania rekordów jako liczbę rekordów w danych uczących.
num_fields	<i>integer</i>	Liczba zmiennych, jaką należy zgłosić dla każdego rekordu oznaczonego jako anomalia.
impute_missing_values	<i>flag</i>	
adjustment_coeff	<i>number</i>	Wartość używana do równoważenia względnej wagi nadanej zmiennym ciągłym i jakościowym przy obliczaniu odległości.
peer_group_num_auto	<i>flag</i>	Automatycznie oblicza liczbę grup elementów równorzędnych.
min_num_peer_groups	<i>integer</i>	Określa minimalną liczbę grup elementów równorzędnych, gdy peer_group_num_auto ma wartość True.
max_num_per_groups	<i>integer</i>	Określa maksymalną liczbę grup elementów równorzędnych.
num_peer_groups	<i>integer</i>	Określa minimalną liczbę grup elementów równorzędnych, gdy peer_group_num_auto ma wartość False.
noise_level	<i>number</i>	Określa sposób traktowania wartości odstających podczas grupowania. Podaj wartość w zakresie od 0 do 0,5.
noise_ratio	<i>number</i>	Określa, jaka część pamięci przydzielonej dla komponentu ma być używana do buforowania szumów. Podaj wartość w zakresie od 0 do 0,5.

Właściwości węzła apriorinode



Węzeł Apriori pozwala wyodrębnić zestaw reguł na podstawie danych, pobierając reguły o najwyższej możliwej zawartości informacji. Apriori oferuje pięć różnych metod wybierania reguł i korzysta ze złożonego schematu indeksowania do efektywnego przetwarzania dużych zbiorów danych. W przypadku dużych problemów czas uczenia Apriori jest zwykle krótszy. Brak jest arbitralnego limitu co do liczby reguł do utrzymania, możliwa jest obsługa reguł z maksymalnie 32 predykcjami. Apriori wymaga, aby wszystkie zmienne wejściowe i wyjściowe były zmiennymi jakościowymi, lecz oferuje wyższą wydajność z uwagi na optymalizację pod kątem tego typu danych.

Przykład

```
node = stream.create("apriori", "My node")
# Karta "Zmienne"
node.setPropertyValue("custom_fields", True)
node.setPropertyValue("partition", "Test")
# Dla danych nietransakcyjnych
node.setPropertyValue("use_transactional_data", False)
node.setPropertyValue("consequents", ["Age"])
node.setPropertyValue("antecedents", ["BP", "Cholesterol", "Drug"])
# Dla danych transakcyjnych
node.setPropertyValue("use_transactional_data", True)
node.setPropertyValue("id_field", "Age")
node.setPropertyValue("contiguous", True)
node.setPropertyValue("content_field", "Drug")
# Karta "Model"
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")
# Karta "Zaawansowane"
node.setPropertyValue("mode", "Expert")
node.setPropertyValue("evaluation", "ConfidenceRatio")
node.setPropertyValue("lower_bound", 7)
```

Tabela 109. Właściwości węzła apriorinode

Właściwości węzła apriorinode	Wartości	Opis właściwości
consequents	<i>field</i>	W modelach Apriori zamiast standardowych zmiennych przewidywanych i wejściowych używa się następników i poprzedników. Zmienne wagi i częstości nie są używane. Więcej informacji można znaleźć w temacie "Wspólne właściwości węzłów modelowania" na stronie 173.
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>	
optimize	Speed Memory	
use_transactional_data	<i>flag</i>	
contiguous	<i>flag</i>	

Tabela 109. Właściwości węzła apriorinode (kontynuacja)

Właściwości węzła apriorinode	Wartości	Opis właściwości
id_field	string	
content_field	string	
mode	Simple Expert	
evaluation	RuleConfidence DifferenceToPrior ConfidenceRatio InformationDifference NormalizedChiSquare	
lower_bound	number	
optimize	Speed Memory	Służy do określania, czy budowanie modelu ma być optymalizowane pod kątem szybkości, czy zużycia pamięci.

Właściwości węzła associationrulesnode



Węzeł Reguły asocjacyjne jest podobny do węzła Apriori; jednak inaczej niż w przypadku Apriori, węzeł Reguły asocjacyjne umożliwia przetwarzanie danych w postaci listy. Ponadto węzeł Reguły asocjacyjne może być używany wraz z IBM SPSS Analytic Server do przetwarzania dużych zbiorów danych i korzystania z szybszego przetwarzania równoległego.

Tabela 110. Właściwości węzła associationrulesnode

Właściwości węzła associationrulesnode	Typ danych	Opis właściwości
predictions	field	Zmienne na tej liście mogą być tylko predyktorami reguły.
conditions	[field1...fieldN]	Zmienne na tej liście mogą być tylko warunkami reguły.
max_rule_conditions	integer	Maksymalna liczba warunków, które mogą być uwzględnione w jednej regule. Minimum 1, maksimum 9.
max_rule_predictions	integer	Maksymalna liczba predykcji, które mogą być uwzględnione w jednej regule. Minimum 1, maksimum 5.
max_num_rules	integer	Maksymalna liczba reguł, które mogą być uwzględnione przy budowaniu reguł. Minimum 1, maksimum 10 000.
rule_criterion_top_n	Confidence Rulesupport Lift Conditionsupport Deployability	Kryterium reguł, które określa wartość wg której wybieranych jest N najlepszych reguł w modelu.
true_flags	Boolean	Wartość Y powoduje, że przy budowaniu reguł uwzględniane będą tylko wartości prawdziwe zmiennych typu flaga.
rule_criterion	Boolean	Wartość Y powoduje, że wartości kryterium reguły są używane do wykluczania reguł przy budowaniu modelu.

Tabela 110. Właściwości węzła *associationrulesnode* (kontynuacja)

Właściwości węzła associationrulesnode	Typ danych	Opis właściwości
min_confidence	<i>number</i>	Od 0,1 do 100 — określony procentowo minimalny poziom ufności reguły generowanej przez model. Jeśli model generuje regułę o poziomie ufności niższym niż ta wartość, reguła jest odrzucana.
min_rule_support	<i>number</i>	Od 0,1 do 100 — określone procentowo minimalne pokrycie dla reguły generowanej przez model. Jeśli model generuje regułę o pokryciu niższym niż ta wartość, reguła jest odrzucana.
min_condition_support	<i>number</i>	Od 0,1 do 100 — określone procentowo minimalne pokrycie dla warunku generowanego przez model. Jeśli model generuje warunek o pokryciu niższym niż ta wartość, reguła jest odrzucana.
min_lift	<i>integer</i>	Od 1 do 10 — minimalny wymagany przyrost dla reguły generowanej przez model. Jeśli model generuje regułę o przyroście niższym niż ta wartość, reguła jest odrzucana.
exclude_rules	<i>Boolean</i>	Służy do wybierania listy powiązanych zmiennych, z których model nie powinien tworzyć reguł. Przykład: set :gsarsnode.exclude_rules = [[field1,field2, field3],[field4, field5]] - gdzie każda lista zmiennych oddzielona znakami [] jest osobnym wierszem w tabeli.
num_bins	<i>integer</i>	Określa liczbę automatycznie wyznaczanych kategorii, do których przydzielane są zmienne ciągłe. Minimum 2, maksimum 10.
max_list_length	<i>integer</i>	Ma zastosowanie do dowolnych zmiennych typu lista o nieznannej długości maksymalnej. Elementy na liście do określonego numeru włącznie są uwzględniane przy budowaniu modelu; wszystkie dalsze elementy są odrzucane. Minimum 1, maksimum 100.
output_confidence	<i>Boolean</i>	
output_rule_support	<i>Boolean</i>	
output_lift	<i>Boolean</i>	
output_condition_support	<i>Boolean</i>	
output_deployability	<i>Boolean</i>	
rules_to_display	upto all	Maksymalna liczba reguł, która może być ujęta w tabelach wynikowych.
display_upto	<i>integer</i>	Jeśli określono wartość <i>upto</i> dla właściwości <i>rules_to_display</i> , określa liczbę reguł, która ma być ujęta w tabelach wynikowych. Minimum 1.
field_transformations	<i>Boolean</i>	
records_summary	<i>Boolean</i>	
rule_statistics	<i>Boolean</i>	
most_frequent_values	<i>Boolean</i>	
most_frequent_fields	<i>Boolean</i>	
word_cloud	<i>Boolean</i>	

Tabela 110. Właściwości węzła *associationrulesnode* (kontynuacja)

Właściwości węzła associationrulesnode	Typ danych	Opis właściwości
word_cloud_sort	Confidence Rulesupport Lift Conditionsupport Deployability	
word_cloud_display	<i>integer</i>	Minimum 1, maksimum 20
max_predictions	<i>integer</i>	Maksymalna liczba reguł, która może być stosowana do każdej zmiennej wejściowej przy ocenie.
criterion	Confidence Rulesupport Lift Conditionsupport Deployability	Wybiera miarę, która będzie używana w celu określenia siły reguł.
allow_repeats	<i>Boolean</i>	Określa, czy reguły o tej samej predykcji będą uwzględniane w ocenie.
check_input	NoPredictions Predictions NoCheck	

Właściwości węzła *autoclassifiernode*



Węzeł Auto Klasyfikacja tworzy i porównuje różne modele pod kątem wyników binarnych (tak lub nie, odejścia lub brak odejścia itd.), umożliwiając użytkownikowi wybór optymalnego podejścia do danej analizy. Obsługiwana jest pewna liczba algorytmów modelowania, co umożliwia wybór metod, które mają zostać użyte, konkretnych opcji dla każdej z nich oraz kryteriów porównywania wyników. Węzeł generuje zestaw modeli w oparciu o określone opcje i nadaje rangi najlepszym kandydatom wybranym według wskazanych kryteriów.

Przykład

```
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)
```

Tabela 111. Właściwości węzła *autoclassifiernode*.

Właściwości węzła autoclassifiernode	Wartości	Opis właściwości
target	<i>field</i>	W przypadku zmiennych przewidywanych typu flaga węzeł Auto Klasyfikacja wymaga jednej zmiennej przewidywanej i co najmniej jednej zmiennej wejściowej. Można też określić zmienne wagi i częstości. Więcej informacji można znaleźć w temacie "Wspólne właściwości węzłów modelowania" na stronie 173.

Tabela 111. Właściwości węzła *autoclassifiernode* (kontynuacja).

Właściwości węzła <i>autoclassifiernode</i>	Wartości	Opis właściwości
ranking_measure	Accuracy Area_under_curve Profit Lift Num_variables	
ranking_dataset	Training Test	
number_of_models	<i>integer</i>	Liczba modeli do uwzględnienia w modelu użytkowym. Liczba całkowita od 1 do 100.
calculate_variable_importance	<i>flag</i>	
enable_accuracy_limit	<i>flag</i>	
accuracy_limit	<i>integer</i>	Liczba całkowita od 0 do 100.
enable_area_under_curve_limit	<i>flag</i>	
area_under_curve_limit	<i>number</i>	Liczba rzeczywista od 0,0 do 1,0.
enable_profit_limit	<i>flag</i>	
profit_limit	<i>number</i>	Liczba całkowita większa od 0.
enable_lift_limit	<i>flag</i>	
lift_limit	<i>number</i>	Liczba rzeczywista większa od 1,0.
enable_number_of_variables_limit	<i>flag</i>	
number_of_variables_limit	<i>number</i>	Liczba całkowita większa od 0.
use_fixed_cost	<i>flag</i>	
fixed_cost	<i>number</i>	Liczba rzeczywista większa od 0,0.
variable_cost	<i>field</i>	
use_fixed_revenue	<i>flag</i>	
fixed_revenue	<i>number</i>	Liczba rzeczywista większa od 0,0.
variable_revenue	<i>field</i>	
use_fixed_weight	<i>flag</i>	
fixed_weight	<i>number</i>	Liczba rzeczywista większa od 0,0
variable_weight	<i>field</i>	
lift_percentile	<i>number</i>	Liczba całkowita od 0 do 100.
enable_model_build_time_limit	<i>flag</i>	
model_build_time_limit	<i>number</i>	Liczba (całkowita) minut ograniczająca czas budowania jednego modelu.
enable_stop_after_time_limit	<i>flag</i>	
stop_after_time_limit	<i>number</i>	Liczba (rzeczywista) godzin ograniczająca łączny czas działania automatycznego klasyfikatora.
enable_stop_after_valid_model_produced	<i>flag</i>	
use_costs	<i>flag</i>	
<algorithm>	<i>flag</i>	Włącza lub wyłącza stosowanie konkretnego algorytmu.

Tabela 111. Właściwości węzła `autoclassifiernode` (kontynuacja).

Właściwości węzła <code>autoclassifiernode</code>	Wartości	Opis właściwości
<code><algorithm>.<property></code>	<i>string</i>	Ustawia wartość właściwości konkretnego algorytmu. Więcej informacji można znaleźć w temacie "Ustawianie właściwości algorytmu".

Ustawianie właściwości algorytmu

W przypadku węzłów Auto Klasyfikacja, Auto Predykcja i Auto Grupowanie właściwości konkretnych algorytmów używanych przez węzeł można ustawiać, korzystając z instrukcji w postaci:

```
autonode.setKeyedPropertyValue(<algorithm>, <property>, <value>)
```

Na przykład:

```
node.setKeyedPropertyValue("neuralnetwork", "method", "MultilayerPerceptron")
```

Nazwy algorytmów dla węzła Auto Klasyfikacja to: `cart`, `chaid`, `quest`, `c50`, `logreg`, `decisionlist`, `bayesnet`, `discriminant`, `svm` i `knn`.

Nazwy algorytmów dla węzła Auto Predykcja to: `cart`, `chaid`, `neuralnetwork`, `genlin`, `svm`, `regression`, `linear` i `knn`.

Nazwy algorytmów dla węzła Auto Grupowanie to: `twostep`, `k-means` i `kohonen`.

Nazwy właściwości są standardowe, zgodnie z dokumentacją poszczególnych węzłów i algorytmów.

Właściwości algorytmów zawierające kropki i inne znaki interpunkcyjne muszą być ujęte w pojedyncze cudzysłowy, na przykład:

```
node.setKeyedPropertyValue("logreg", "tolerance", "1.0E-5")
```

Właściwości można też nadać więcej niż jedną wartość, na przykład:

```
node.setKeyedPropertyValue("decisionlist", "search_direction", ["Up", "Down"])
```

Aby włączyć lub wyłączyć stosowanie konkretnego algorytmu:

```
node.setPropertyValue("chaid", True)
```

Uwaga: W przypadkach, gdy pewne opcje algorytmu nie są dostępne w węźle Auto Klasyfikacja, lub gdy można określić tylko jedną wartość, a nie zakres wartości, w skryptach obowiązują te same ograniczenia, co przy standardowym dostępie do węzła.

Właściwości węzła `autoclusternode`



Węzeł Autogrupowanie szacuje i porównuje modele skupień identyfikujące grupy rekordów o podobnej charakterystyce. Węzeł działa tak samo, jak pozostałe zautomatyzowane węzły modelowania, umożliwiając eksperymentowanie z wieloma kombinacjami opcji w pojedynczym przebiegu modelowania. Modele można porównywać, korzystając z miar bazowych, które pozwalają podejmować próby filtrowania i oceny przydatności modelu skupień oraz udostępniają miary bazujące na istotności poszczególnych zmiennych.

Przykład

```

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)

```

Tabela 112. Właściwości węzła *autoclusternode*

Właściwości węzła <i>autoclusternode</i>	Wartości	Opis właściwości
evaluation	<i>field</i>	Uwaga: Tylko węzeł Auto Grupowanie. Określa zmienną, dla której będzie obliczana wartość ważności. Może też służyć do określania, na ile dobrze grupa różnicuje wartość tej zmiennej, a tym samym, na ile dobrze model będzie przewidywał tę zmienną.
ranking_measure	Silhouette Num_clusters Size_smallest_cluster Size_largest_cluster Smallest_to_largest Importance	
ranking_dataset	Training Test	
summary_limit	<i>integer</i>	Liczba modeli do uwzględnienia w raporcie. Liczba całkowita od 1 do 100.
enable_silhouette_limit	<i>flag</i>	
silhouette_limit	<i>integer</i>	Liczba całkowita od 0 do 100.
enable_number_less_limit	<i>flag</i>	
number_less_limit	<i>number</i>	Liczba rzeczywista od 0,0 do 1,0.
enable_number_greater_limit	<i>flag</i>	
number_greater_limit	<i>number</i>	Liczba całkowita większa od 0.
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>	Liczba całkowita większa od 0.
enable_largest_cluster_limit	<i>flag</i>	
largest_cluster_units	Percentage Counts	
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>	Liczba całkowita od 0 do 100.
importance_limit_less_than	<i>number</i>	Liczba całkowita od 0 do 100.
<algorithm>	<i>flag</i>	Włącza lub wyłącza stosowanie konkretnego algorytmu.

Tabela 112. Właściwości węzła `autoclusternode` (kontynuacja)

Właściwości węzła <code>autoclusternode</code>	Wartości	Opis właściwości
<code><algorithm>.<property></code>	<i>string</i>	Ustawia wartość właściwości konkretnego algorytmu. Więcej informacji można znaleźć w temacie “Ustawianie właściwości algorytmu” na stronie 180.

Właściwości węzła `autonumericnode`



Węzeł Auto Predykcja estymuje i porównuje modele zwracające wyniki w formie ciągłego przedziału liczbowego, korzystając z szeregu różnych metod. Węzeł działa tak samo, jak węzeł Auto Klasyfikacja, umożliwiając użytkownikowi wybór używanych algorytmów oraz eksperymentowanie z wieloma kombinacjami opcji w pojedynczym przebiegu modelowania. Obsługiwane algorytmy obejmują sieci neuronowe, drzewo C&R, CHAID, regresję liniową, uogólnioną regresję liniową oraz algorytmy SVM. Modele można porównywać na podstawie korelacji, błędu względnego lub liczby używanych zmiennych.

Przykład

```
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)
```

Tabela 113. Właściwości węzła `autonumericnode`

Właściwości węzła <code>autonumericnode</code>	Wartości	Opis właściwości
<code>custom_fields</code>	<i>flag</i>	Wartość True powoduje, że zamiast ustawień węzła Typ używane będą niestandardowe ustawienia zmiennych.
<code>target</code>	<i>field</i>	Węzeł Auto Predykcja wymaga jednej zmiennej przewidywanej i co najmniej jednej zmiennej wejściowej. Można też określić zmienne wagi i częstości. Więcej informacji można znaleźć w temacie “Wspólne właściwości węzłów modelowania” na stronie 173.
<code>inputs</code>	<i>[field1 ... field2]</i>	
<code>partition</code>	<i>field</i>	
<code>use_frequency</code>	<i>flag</i>	
<code>frequency_field</code>	<i>field</i>	
<code>use_weight</code>	<i>flag</i>	
<code>weight_field</code>	<i>field</i>	
<code>use_partitioned_data</code>	<i>flag</i>	Jeśli zdefiniowana jest zmienna dzieląca na podzbiory, do budowania modelu używane są wyłącznie dane uczące.
<code>ranking_measure</code>	Correlation NumberOfFields	
<code>ranking_dataset</code>	Test Training	

Tabela 113. Właściwości węzła `autonumericnode` (kontynuacja)

Właściwości węzła <code>autonumericnode</code>	Wartości	Opis właściwości
<code>number_of_models</code>	<i>integer</i>	Liczba modeli do uwzględnienia w modelu użytkowym. Liczba całkowita od 1 do 100.
<code>calculate_variable_importance</code>	<i>flag</i>	
<code>enable_correlation_limit</code>	<i>flag</i>	
<code>correlation_limit</code>	<i>integer</i>	
<code>enable_number_of_fields_limit</code>	<i>flag</i>	
<code>number_of_fields_limit</code>	<i>integer</i>	
<code>enable_relative_error_limit</code>	<i>flag</i>	
<code>relative_error_limit</code>	<i>integer</i>	
<code>enable_model_build_time_limit</code>	<i>flag</i>	
<code>model_build_time_limit</code>	<i>integer</i>	
<code>enable_stop_after_time_limit</code>	<i>flag</i>	
<code>stop_after_time_limit</code>	<i>integer</i>	
<code>stop_if_valid_model</code>	<i>flag</i>	
<code><algorithm></code>	<i>flag</i>	Włącza lub wyłącza stosowanie konkretnego algorytmu.
<code><algorithm>.<property></code>	<i>string</i>	Ustawia wartość właściwości konkretnego algorytmu. Więcej informacji można znaleźć w temacie "Ustawianie właściwości algorytmu" na stronie 180.

Właściwości węzła `bayesnetnode`



Węzeł Sieć Bayesowska umożliwia utworzenie modelu prawdopodobieństwa przez połączenie zaobserwowanych i zarejestrowanych dowodów z wiedzą rzeczywistą w celu ustanowienia prawdopodobieństwa występowania. Węzeł koncentruje się na sieciach Tree Augmented Naïve Bayes (TAN) i Markov Blanket, używanych głównie podczas klasyfikacji.

Przykład

```
node = stream.create("bayesnet", "My node")
node.setPropertyValue("continue_training_existing_model", True)
node.setPropertyValue("structure_type", "MarkovBlanket")
node.setPropertyValue("use_feature_selection", True)
# Karta Zaawansowane
node.setPropertyValue("mode", "Expert")
node.setPropertyValue("all_probabilities", True)
node.setPropertyValue("independence", "Pearson")
```

Tabela 114. Właściwości węzła bayesnetnode

Właściwości węzła bayesnetnode	Wartości	Opis właściwości
inputs	[field1 ... fieldN]	Modele sieci bayesowskiej używają jednej zmiennej przewidywanej i co najmniej jednej zmiennej wejściowej. Zmienne ciągłe są automatycznie kategoryzowane. Więcej informacji można znaleźć w temacie “Wspólne właściwości węzłów modelowania” na stronie 173.
continue_training_existing_model	flag	
structure_type	TAN MarkovBlanket	Wybierz strukturę, która ma być używana przy budowaniu sieci bayesowskiej.
use_feature_selection	flag	
parameter_learning_method	Likelihood Bayes	Określa metodą używaną do szacowania tabel prawdopodobieństwa warunkowego między węzłami, gdy wartości węzłów nadrzędnych nie są znane.
mode	Expert Simple	
missing_values	flag	
all_probabilities	flag	
independence	Likelihood Pearson	Określa metodę ustalania, czy pary obserwacji dwóch zmiennych są od siebie niezależne.
significance_level	number	Określa wartość odcięcia używaną do określania niezależności.
maximal_conditioning_set	number	Określa maksymalną liczbę zmiennych warunkujących, która ma być używana do testowania niezależności.
inputs_always_selected	[field1 ... fieldN]	Określa zmienne ze zbioru danych, które mają być zawsze używane przy budowaniu sieci bayesowskiej. Uwaga: Zmienna przewidywana jest zawsze wybrana.
maximum_number_inputs	number	Określa maksymalną liczbę zmiennych wejściowych, które mają być używane przy budowaniu sieci bayesowskiej.
calculate_variable_importance	flag	
calculate_raw_propensities	flag	
calculate_adjusted_propensities	flag	
adjusted_propensity_partition	Test Validation	

Właściwości węzła buildr



Węzeł budowy R umożliwia wprowadzenie niestandardowego skryptu w języku R, który będzie przeprowadzał budowanie i ocenę modelu wdrożonego w programie IBM SPSS Modeler.

Przykład

```
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)""
```

Tabela 115. Właściwości węzła buildr.

Właściwości węzła buildr	Wartości	Opis właściwości
build_syntax	string	Komendy skryptu R realizujące budowę modelu.
score_syntax	string	Komendy skryptu R realizujące ocenę modelu.
convert_flags	StringsAndDoubles LogicalValues	Opcja przekształcania zmiennych typu flaga.
convert_datetime	flag	Opcja przekształcania zmiennych o formacie daty lub daty/czasu na formaty daty/czasu języka R.
convert_datetime_class	POSIXct POSIXlt	Opcje określające format docelowy, na który przekształcane są zmienne w formacie daty i daty/czasu.
convert_missing	flag	Opcja przekształcania braków danych na wartości R NA.
output_html	flag	Opcja wyświetlania wykresów na karcie w modelu użytkowym R.
output_text	flag	Opcja zapisywania wyników tekstowych kierowanych przez skrypt na konsolę na karcie modelu użytkowego R.

Właściwości węzła c50node



Węzeł C5.0 tworzy drzewo decyzyjne lub zestaw reguł. Model działa w oparciu o podział próby na podstawie zmiennej oferującej maksimum korzyści z informacji na każdym z poziomów. Zmienna przewidywana musi być jakościowa. Dozwolonych jest wiele podziałów na więcej niż dwie podgrupy.

Przykład

```
node = stream.create("c50", "My node")
# Karta "Model"
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)
# Karta "Koszty"
node.setPropertyValue("use_costs", True)
node.setPropertyValue("costs", [["drugA", "drugX", 2]])
```

Tabela 116. Właściwości węzła c50node

Właściwości węzła c50node	Wartości	Opis właściwości
target	<i>field</i>	Modele c50 używają jednej zmiennej przewidywanej i co najmniej jednej zmiennej wejściowej. Można również określić zmienną ważącą. Więcej informacji można znaleźć w temacie “Wspólne właściwości węzłów modelowania” na stronie 173.
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	Preferowanie dokładności albo ogólności.
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>	Jest to właściwość ustrukturyzowana.
use_winning	<i>flag</i>	
use_global_pruning	<i>flag</i>	Domyślnie włączona (True).
calculate_variable_importance	<i>flag</i>	
calculate_raw_propensities	<i>flag</i>	
calculate_adjusted_propensities	<i>flag</i>	
adjusted_propensity_partition	Test Validation	

Właściwości węzła carmanode



Model CARMA pozwala wyodrębnić zestaw reguł na podstawie danych bez konieczności określania zmiennych wejściowych lub przewidywanych. Inaczej niż Apriori, węzeł CARMA oferuje ustawienia tworzenia dla obsługi reguł (pokrycie poprzedników i następników) zamiast pokrycia tylko poprzedników. Oznacza to, że wygenerowane reguły mogą być używane w szerszym spektrum zastosowań — na przykład w celu znalezienia listy produktów lub usług (poprzedników), z których wynikać będzie decyzja o promowaniu konkretnego produktu (następnika) w tegorocznym sezonie świątecznym.

Przykład

```
node = stream.create("carma", "My node")
# Karta "Zmienne"
node.setPropertyValue("custom_fields", True)
node.setPropertyValue("use_transactional_data", True)
node.setPropertyValue("inputs", ["BP", "Cholesterol", "Drug"])
node.setPropertyValue("partition", "Test")
```



```

# Karta "Model"
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)
# Opcje zaawansowane
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)

```

Tabela 117. Właściwości węzła carmanode

Właściwości węzła carmanode	Wartości	Opis właściwości
inputs	[field1 ... fieldn]	W modelach CARMA używana jest lista zmiennych wejściowych, ale nie są używane zmienne przewidywane. Zmienne wagi i częstości nie są używane. Więcej informacji można znaleźć w temacie “Wspólne właściwości węzłów modelowania” na stronie 173.
id_field	field	Zmienna identyfikacyjna używana przy budowaniu modelu.
contiguous	flag	Służy do określania, czy identyfikatory w zmiennej identyfikacyjnej są ciągle.
use_transactional_data	flag	
content_field	field	
min_supp	number(percent)	Odnosi się do pokrycia reguł, a nie pokrycia poprzedników. Domyślną wartością jest 20%.
min_conf	number(percent)	Domyślną wartością jest 20%.
max_size	number	Domyślną wartością jest 10.
mode	Simple Expert	Domyślną wartością jest Simple.
exclude_multiple	flag	Wyklucza reguły z wieloma następnikami. Domyślną wartością jest Fałsz.
use_pruning	flag	Domyślną wartością jest Fałsz.
pruning_value	number	Domyślną wartością jest 500.
vary_support	flag	
estimated_transactions	integer	
rules_without_antecedents	flag	

Właściwości węzła cartnode



Węzeł Klasyfikacja i regresja (K&R) generuje drzewo decyzyjne umożliwiające predykcję lub klasyfikację przyszłych obserwacji. W metodzie tej stosowany jest rekursywny podział rekordów na segmenty przez minimalizację zanieczyszczeń w każdym kroku, przy czym węzeł w drzewie jest uważany za „czysty”, jeśli 100% obserwacji w węźle przypada na konkretną kategorię zmiennej przewidywanej. Zmienne przewidywana i wejściowa mogą być zakresami liczbowymi lub jakościowymi (nominalnymi, porządkowymi lub flagami); wszystkie podziały są binarne (tylko dwie podgrupy).

Przykład

```
node = stream.createAt("cart", "My node", 200, 100)
# Karta "Zmienne"
node.setPropertyValue("custom_fields", True)
node.setPropertyValue("target", "Drug")
node.setPropertyValue("inputs", ["Age", "BP", "Cholesterol"])
# Karta "Opcje budowania", panel "cel"
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""")
# Karta "Opcje budowania", panel "Podstawowe"
node.setPropertyValue("prune_tree", False)
node.setPropertyValue("use_std_err_rule", True)
node.setPropertyValue("std_err_multiplier", 3.0)
node.setPropertyValue("max_surrogates", 7)
# Karta "Opcje budowania", panel "Reguły zatrzymujące"
node.setPropertyValue("use_percentage", True)
node.setPropertyValue("min_parent_records_pc", 5)
node.setPropertyValue("min_child_records_pc", 3)
# Karta "Opcje budowania", panel "Zaawansowane"
node.setPropertyValue("min_impurity", 0.0003)
node.setPropertyValue("impurity_measure", "Twoing")
# Karta "Opcje modelu"
node.setPropertyValue("use_model_name", True)
node.setPropertyValue("model_name", "Cart_Drug")
```

Tabela 118. Właściwości węzła cartnode

Właściwości węzła cartnode	Wartości	Opis właściwości
target	<i>field</i>	Modele C&R Tree używają jednej zmiennej przewidywanej i co najmniej jednej zmiennej wejściowej. Można też określić zmienną częstości. Więcej informacji można znaleźć w temacie “Wspólne właściwości węzłów modelowania” na stronie 173.
continue_training_existing_model	<i>flag</i>	
objective	Standard Boosting Bagging psm	Wartość psm jest używana w przypadku bardzo dużych zbiorów danych i wymaga połączenia z serwerem.
model_output_type	Single InteractiveBuilder	
use_tree_directives	<i>flag</i>	
tree_directives	<i>string</i>	Określ dyrektywy rozrostu drzewa. Dyrektywy można ujmować w potrójne cudzysłowy, aby uniknąć poprzedzania znaków nowego wiersza lub cudzysłowów znakami zmiany znaczenia. Uwaga: dyrektywy mogą być silnie wrażliwe na drobne zmiany w danych lub opcjach modelowania i mogą nie nadawać się do stosowania z innymi zbiorami danych.
use_max_depth	Default Custom	

Tabela 118. Właściwości węzła `cartnode` (kontynuacja)

Właściwości węzła <code>cartnode</code>	Wartości	Opis właściwości
<code>max_depth</code>	<i>integer</i>	Maksymalna głębokość drzewa, od 0 do 1000. Używana tylko wtedy, gdy <code>use_max_depth = Custom</code> .
<code>prune_tree</code>	<i>flag</i>	Przytnij drzewo, aby uniknąć przeuczenia.
<code>use_std_err</code>	<i>flag</i>	Użyj maksymalnej różnicy w ryzyku (w błędach standardowych).
<code>std_err_multiplier</code>	<i>number</i>	Maksymalna różnica.
<code>max_surrogates</code>	<i>number</i>	Maksymalna liczba substytutów.
<code>use_percentage</code>	<i>flag</i>	
<code>min_parent_records_pc</code>	<i>number</i>	
<code>min_child_records_pc</code>	<i>number</i>	
<code>min_parent_records_abs</code>	<i>number</i>	
<code>min_child_records_abs</code>	<i>number</i>	
<code>use_costs</code>	<i>flag</i>	
<code>costs</code>	<i>structured</i>	Właściwość ustrukturyzowana.
<code>priors</code>	Data Equal Custom	
<code>custom_priors</code>	<i>structured</i>	Właściwość ustrukturyzowana.
<code>adjust_priors</code>	<i>flag</i>	
<code>trails</code>	<i>number</i>	Liczba modeli zespolonych dla boostingu i/lub agregacji bootstrapowej.
<code>set_ensemble_method</code>	Voting HighestProbability HighestMeanProbability	Domyślna reguła zespolenia dla przewidywanych zmiennych jakościowych.
<code>range_ensemble_method</code>	Mean Median	Domyślna reguła zespolenia dla docelowych wartości ilościowych.
<code>large_boost</code>	<i>flag</i>	Zastosuj wzmocnienie do bardzo dużych zbiorów danych.
<code>min_impurity</code>	<i>number</i>	
<code>impurity_measure</code>	Gini Twoing Ordered	
<code>train_pct</code>	<i>number</i>	Zbiór zabezpieczający przed przeuczeniem.
<code>set_random_seed</code>	<i>flag</i>	Opcja replikacji wyników.
<code>seed</code>	<i>number</i>	
<code>calculate_variable_importance</code>	<i>flag</i>	
<code>calculate_raw_propensities</code>	<i>flag</i>	
<code>calculate_adjusted_propensities</code>	<i>flag</i>	
<code>adjusted_propensity_partition</code>	Test Validation	

Właściwości węzła chaidnode



Węzeł CHAID generuje drzewa decyzyjne, korzystając ze statystyk chi-kwadrat w celu identyfikacji optymalnych podziałów. W odróżnieniu od węzłów drzewa C&R i węzłów QUEST, CHAID może generować drzewa niebinarne, co oznacza, że niektóre podziały mają więcej niż dwie gałęzie. Zmienne przewidywana i wejściowa mogą być zakresami liczbowymi (ilościowymi) lub jakościowymi.

Wyczerpujący CHAID stanowi modyfikację CHAID umożliwiającą dokładniejsze badanie wszystkich możliwych podziałów, lecz obliczenia w jego przypadku zajmują więcej czasu.

Przykład

```

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)

```

Tabela 119. Właściwości węzła chaidnode

Właściwości węzła chaidnode	Wartości	Opis właściwości
target	<i>field</i>	Modele CHAID używają jednej zmiennej przewidywanej i co najmniej jednej zmiennej wejściowej. Można też określić zmienną częstości. Więcej informacji można znaleźć w temacie “Wspólne właściwości węzłów modelowania” na stronie 173.
continue_training_existing_model	<i>flag</i>	
objective	Standard Boosting Bagging psm	Wartość psm jest używana w przypadku bardzo dużych zbiorów danych i wymaga połączenia z serwerem.
model_output_type	Single InteractiveBuilder	
use_tree_directives	<i>flag</i>	
tree_directives	<i>string</i>	
method	Chaid ExhaustiveChaid	

Tabela 119. Właściwości węzła *chaidnode* (kontynuacja)

Właściwości węzła <i>chaidnode</i>	Wartości	Opis właściwości
<i>use_max_depth</i>	Default Custom	
<i>max_depth</i>	<i>integer</i>	Maksymalna głębokość drzewa, od 0 do 1000. Używana tylko wtedy, gdy <i>use_max_depth</i> = Custom.
<i>use_percentage</i>	<i>flag</i>	
<i>min_parent_records_pc</i>	<i>number</i>	
<i>min_child_records_pc</i>	<i>number</i>	
<i>min_parent_records_abs</i>	<i>number</i>	
<i>min_child_records_abs</i>	<i>number</i>	
<i>use_costs</i>	<i>flag</i>	
<i>costs</i>	<i>structured</i>	Właściwość ustrukturyzowana.
<i>trails</i>	<i>number</i>	Liczba modeli zespolonych dla boostingu i/lub agregacji bootstrapowej.
<i>set_ensemble_method</i>	Voting HighestProbability HighestMeanProbability	Domyślna reguła zespolenia dla przewidywanych zmiennych jakościowych.
<i>range_ensemble_method</i>	Mean Median	Domyślna reguła zespolenia dla docelowych wartości ilościowych.
<i>large_boost</i>	<i>flag</i>	Zastosuj wzmocnienie do bardzo dużych zbiorów danych.
<i>split_alpha</i>	<i>number</i>	Poziom istotności dla podziałów.
<i>merge_alpha</i>	<i>number</i>	Poziom istotności dla łączenia.
<i>bonferroni_adjustment</i>	<i>flag</i>	Koryguj wartości istotności metodą Bonferroni.
<i>split_merged_categories</i>	<i>flag</i>	Zezwalaj na ponowny podział połączonych kategorii.
<i>chi_square</i>	Pearson LR	Metoda obliczania statystyki chi-kwadrat: Pearsona albo iloraz wiarygodności
<i>epsilon</i>	<i>number</i>	Minimalna zmiana oczekiwanych częstości komórek.
<i>max_iterations</i>	<i>number</i>	Maksimum iteracji dla uzyskania zbieżności.
<i>set_random_seed</i>	<i>integer</i>	
<i>seed</i>	<i>number</i>	
<i>calculate_variable_importance</i>	<i>flag</i>	
<i>calculate_raw_propensities</i>	<i>flag</i>	
<i>calculate_adjusted_propensities</i>	<i>flag</i>	
<i>adjusted_propensity_partition</i>	Test Validation	
<i>maximum_number_of_models</i>	<i>integer</i>	

Właściwości węzła coxregnode



Węzeł regresji Coxa umożliwia utworzenie modelu przeżycia dla danych określających czasy do wystąpienia zdarzeń i zawierających ocenzone rekordy. Model generuje funkcję przeżycia przewidującą prawdopodobieństwo, że zdarzenie będące przedmiotem zainteresowania wystąpiło w określonym czasie (t) dla danych wartości zmiennych wejściowych.

Przykład

```
node = stream.create("coxreg", "My node")
node.setPropertyValue("survival_time", "tenure")
node.setPropertyValue("method", "BackwardsStepwise")
# Karta Zaawansowane
node.setPropertyValue("mode", "Expert")
node.setPropertyValue("removal_criterion", "Conditional")
node.setPropertyValue("survival", True)
```

Tabela 120. Właściwości węzła coxregnode

Właściwości węzła coxregnode	Wartości	Opis właściwości
survival_time	<i>field</i>	Modele regresji Coxa wymagają jednej zmiennej zawierającej czasy przeżycia.
target	<i>field</i>	Modele regresji Coxa wymagają jednej zmiennej przewidywanej i co najmniej jednej zmiennej wejściowej. Więcej informacji można znaleźć w temacie “Wspólne właściwości węzłów modelowania” na stronie 173.
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	

Tabela 120. Właściwości węzła *coxregnode* (kontynuacja)

Właściwości węzła <i>coxregnode</i>	Wartości	Opis właściwości
<i>l_converge</i>	1.0E-1 1.0E-2 1.0E-3 1.0E-4 1.0E-5 0	
<i>removal_criterion</i>	LR Wald Conditional	
<i>probability_entry</i>	<i>number</i>	
<i>probability_removal</i>	<i>number</i>	
<i>output_display</i>	EachStep LastStep	
<i>ci_enable</i>	<i>flag</i>	
<i>ci_value</i>	90 95 99	
Korelacje	<i>flag</i>	
<i>display_baseline</i>	<i>flag</i>	
<i>survival</i>	<i>flag</i>	
<i>hazard</i>	<i>flag</i>	
<i>log_minus_log</i>	<i>flag</i>	
<i>one_minus_survival</i>	<i>flag</i>	
<i>separate_line</i>	<i>field</i>	
<i>value</i>	<i>number</i> lub <i>string</i>	Jeśli nie zostanie określona wartość dla zmiennej, to będzie dla niej obowiązywać domyślna opcja "Mean".

Właściwości węzła *decisionlistnode*



Węzeł Lista decyzyjna identyfikuje podgrupy lub segmenty wskazujące wyższe lub niższe prawdopodobieństwo danego wyniku binarnego względem całej populacji. Można na przykład wyszukać klientów, których prawdopodobieństwo odejścia jest niewielkie, lub którzy z dużym prawdopodobieństwem pozytywnie zareagują na kampanię. Istnieje możliwość zastosowania posiadanej wiedzy biznesowej w modelu przez dodanie własnych, niestandardowych segmentów i przejrzanie modeli alternatywnych jeden obok drugiego w celu porównania wyników. Modele Lista decyzyjna składają się z list reguł, w których każda reguła ma warunek i wynik. Reguły są stosowane w kolejności wprowadzania, a pierwsza reguła spełniona określa wynik.

Przykład

```
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)
```

Tabela 121. Właściwości węzła decisionlistnode

Właściwości węzła decisionlistnode	Wartości	Opis właściwości
target	<i>field</i>	Modele Lista decyzyjna używają jednej zmiennej przewidywanej i co najmniej jednej zmiennej wejściowej. Można też określić zmienną częstości. Więcej informacji można znaleźć w temacie “Wspólne właściwości węzłów modelowania” na stronie 173.
model_output_type	Model InteractiveBuilder	
search_direction	Up Down	Dotyczy wyszukiwania segmentów; wartość Up odpowiada wysokiemu prawdopodobieństwu, a wartość Down odpowiada niskiemu prawdopodobieństwu.
target_value	<i>string</i>	Jeśli wartość nie zostanie określona, to dla flag przyjmowana będzie wartość true.
max_rules	<i>integer</i>	Maksymalna liczba segmentów, bez reszty.
min_group_size	<i>integer</i>	Minimalna wielkość segmentu.
min_group_size_pct	<i>number</i>	Minimalna wielkość segmentu (jako procent).
confidence_level	<i>number</i>	Minimalny wzrost prawdopodobieństwa odpowiedzi, przy którym warto uzupełnić definicję segmentu.
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>	Szerokość przeszukiwania list.
max_rules_per_cycle	<i>integer</i>	Szerokość przeszukiwania reguł segmentacji.
segment_growth	<i>number</i>	
include_missing	<i>flag</i>	
final_results_only	<i>flag</i>	
reuse_fields	<i>flag</i>	Dopuszcza ponowne użycie atrybutów (zmiennych wejściowych występujących w regułach).
max_alternatives	<i>integer</i>	
calculate_raw_propensities	<i>flag</i>	
calculate_adjusted_propensities	<i>flag</i>	
adjusted_propensity_partition	Test Validation	

Właściwości węzła discriminantnode



Analiza dyskryminacyjna opiera się na ściślejszych założeniach niż regresja logistyczna, lecz może stanowić wartościową alternatywę lub uzupełnienie analizy metodą regresji logistycznej w przypadku spełnienia tych założeń.

Przykład

```
node = stream.create("discriminant", "My node")
node.setPropertyValue("target", "custcat")
node.setPropertyValue("use_partitioned_data", False)
node.setPropertyValue("method", "Stepwise")
```

Tabela 122. Właściwości węzła discriminantnode

Właściwości węzła discriminantnode	Wartości	Opis właściwości
target	field	Modele Analiza dyskryminacyjna wymagają jednej zmiennej przewidywanej i co najmniej jednej zmiennej wejściowej. Zmienne wagi i częstości nie są używane. Więcej informacji można znaleźć w temacie “Wspólne właściwości węzłów modelowania” na stronie 173.
method	Enter Stepwise	
mode	Simple Expert	
prior_probabilities	AllEqual ComputeFromSizes	
covariance_matrix	WithinGroups SeparateGroups	
means	flag	Opcje statystyk w oknie dialogowym zaawansowanych opcji wyników.
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	Opcje klasyfikacji w oknie dialogowym zaawansowanych opcji wyników.
limit_to_first	number	Wartość domyślna to 10.
summary_table	flag	
leave_one_classification	flag	
combined_groups	flag	
separate_groups_covariance	flag	Opcja macierzy Kowariancja odrębnych grup .
territorial_map	flag	

Tabela 122. Właściwości węzła *discriminantnode* (kontynuacja)

Właściwości węzła <i>discriminantnode</i>	Wartości	Opis właściwości
combined_groups	<i>flag</i>	Opcja wykresu Połączone grupy .
separate_groups	<i>flag</i>	Opcja wykresu Osobne grupy .
summary_of_steps	<i>flag</i>	
F_pairwise	<i>flag</i>	
stepwise_method	WilksLambda UnexplainedVariance MahalanobisDistance SmallestF RaosV	
V_to_enter	<i>number</i>	
criteria	UseValue UseProbability	
F_value_entry	<i>number</i>	Wartość domyślna to 3,84.
F_value_removal	<i>number</i>	Wartość domyślna to 2,71.
probability_entry	<i>number</i>	Wartość domyślna to 0,05.
probability_removal	<i>number</i>	Wartość domyślna to 0,10.
calculate_variable_importance	<i>flag</i>	
calculate_raw_propensities	<i>flag</i>	
calculate_adjusted_propensities	<i>flag</i>	
adjusted_propensity_partition	Test Validation	

Właściwości węzła *extensionmodelnode*



Z węzłem wynikowym *Extension* można uruchomić skrypty R lub Python for Spark w celu utworzenia i oceny wyników.

Python for Spark — przykład

```
#### przykład skryptu w języku 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))

# liczba klas docelowych
predictorClassCount = len(metadata[lookup[target]])

# definicja funkcji do wyodrębniania informacji o predyktorach jakościowych z modelu danych
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

# konwersja Dataframe do RDD z 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)
```

Przykład w języku R

```
#### przykład skryptu w języku R
node.setPropertyValue("syntax_type", "R")
node.setPropertyValue("r_build_syntax", """modelerModel <- lm(modelerData$Na~modelerData$K,modelerData)
modelerDataModel
modelerModel
""")
```

Tabela 123. Właściwości węzła *extensionmodelnode*

Właściwości węzła <i>extensionmodelnode</i>	Wartości	Opis właściwości
<code>syntax_type</code>	<i>R</i> <i>Python</i>	Określa, który skrypt ma być wykonany – R lub Python (domyślnie R).
<code>r_build_syntax</code>	<i>string</i>	Komendy skryptu R do budowy modelu.
<code>r_score_syntax</code>	<i>string</i>	Komendy skryptu R realizujące ocenę modelu.
<code>python_build_syntax</code>	<i>string</i>	Komendy skryptu Python realizujące ocenę modelu.
<code>python_score_syntax</code>	<i>string</i>	Komendy skryptu Python realizujące ocenę modelu.
<code>convert_flags</code>	StringsAndDoubles LogicalValues	Opcja przekształcania zmiennych typu flaga.
<code>convert_missing</code>	<i>flag</i>	Opcja przekształcania braków danych na wartości R NA.
<code>convert_datetime</code>	<i>flag</i>	Opcja przekształcania zmiennych o formacie daty lub daty/czasu na formaty daty/czasu języka R.
<code>convert_datetime_class</code>	POSIXct POSIXlt	Opcje określające format docelowy, na który przekształcane są zmienne w formacie daty i daty/czasu.
<code>output_html</code>	<i>flag</i>	Opcja wyświetlania wykresów na karcie w modelu użytkowym R.
<code>output_text</code>	<i>flag</i>	Opcja zapisywania wyników tekstowych kierowanych przez skrypt na konsolę na karcie modelu użytkowego R.

Właściwości węzła factornode



Węzeł Redukcja wymiarów udostępnia wydajne techniki redukcji danych pozwalające obniżyć stopień złożoności danych. Analiza głównych składowych (ang. Principal Components Analysis, PCA) znajduje kombinacje liniowe zmiennych wejściowych, które umożliwiają przechwytywanie wariacji w całym zestawie zmiennych, pod warunkiem że składowe są zlokalizowane ortogonalnie (prostopadle) do siebie. Analiza czynnikowa próbuje zidentyfikować współczynniki objaśniające wzory korelacji występujące w ramach zbiorów obserwowanych zmiennych. W przypadku obu podejść celem jest znalezienie niewielkiej liczby zmiennych wyliczanych w efektywny sposób podsumowującej informację w oryginalnym zestawie zmiennych.

Przykład

```
node = stream.create("factor", "My node")
# Karta "Zmienne"
node.setPropertyValue("custom_fields", True)
node.setPropertyValue("inputs", ["BP", "Na", "K"])
node.setPropertyValue("partition", "Test")
# Karta "Model"
node.setPropertyValue("use_model_name", True)
node.setPropertyValue("model_name", "Factor_Age")
node.setPropertyValue("use_partitioned_data", False)
node.setPropertyValue("method", "GLS")
# Karta Zaawansowane
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)
# "Nałożenie" – sekcja
node.setPropertyValue("rotation", "DirectOblimin")
node.setPropertyValue("delta", 0.3)
node.setPropertyValue("kappa", 7.0)
```

Tabela 124. Właściwości węzła factornode

Właściwości węzła factornode	Wartości	Opis właściwości
inputs	[field1 ... fieldN]	W modelach Redukcja wymiarów używana jest lista zmiennych wejściowych, ale nie są używane zmienne przewidywane. Zmienne wagi i częstości nie są używane. Więcej informacji można znaleźć w temacie “Wspólne właściwości węzłów modelowania” na stronie 173.
method	PC ULS GLS ML PAF Alfa Image	
mode	Simple Expert	
max_iterations	number	

Tabela 124. Właściwości węzła factornode (kontynuacja)

Właściwości węzła factornode	Wartości	Opis właściwości
complete_records	<i>flag</i>	
matrix	Correlation Covariance	
extract_factors	ByEigenvalues ByFactors	
min_eigenvalue	<i>number</i>	
max_factor	<i>number</i>	
rotation	None Varimax DirectOblimin Equamax Quartimax Promax	
delta	<i>number</i>	W przypadku wybrania typu DirectOblimin jako typu danych, można określić wartość delta. Jeśli wartość nie zostanie określona, przyjęta zostanie domyślna wartość delta.
kappa	<i>number</i>	W przypadku wybrania typu Promax jako typu danych, można określić wartość kappa. Jeśli wartość nie zostanie określona, przyjęta zostanie domyślna wartość kappa.
sort_values	<i>flag</i>	
hide_values	<i>flag</i>	
hide_below	<i>number</i>	

Właściwości węzła featureselectionnode



Węzeł Dobór predyktorów przegląda zmienne wejściowe do usunięcia w oparciu o zbiór kryteriów (takich jak procent braków danych); następnie nadaje rangę istotności pozostałych danych wejściowych względem określonej zmiennej przewidywanej. Na przykład, jeśli mamy zbiór danych z setkami potencjalnych danych wejściowych, to które z nich z dużym prawdopodobieństwem okażą się użyteczne w modelowaniu wyników leczenia pacjenta?

Przykład

```
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)
```

Aby zapoznać się z szczegółowym przykładem, który tworzy model wyboru predyktora, patrz “Przykład skryptu samodzielnego: generowanie modelu wyboru predyktora” na stronie 5.

Tabela 125. Właściwości węzła *featureselectionnode*

Właściwości węzła <i>featureselectionnode</i>	Wartości	Opis właściwości
target	<i>field</i>	Modele Wybór predyktora określają rangi predyktorów względem określonej zmiennej przewidywanej. Zmienne wagi i częstości nie są używane. Więcej informacji można znaleźć w temacie "Wspólne właściwości węzłów modelowania" na stronie 173.
screen_single_category	<i>flag</i>	Wartość True powoduje, że monitorowane są zmienne ze zbyt dużą liczbą rekordów należących do tej samej kategorii w odniesieniu do łącznej liczby rekordów.
max_single_category	<i>number</i>	Określa próg używany, gdy <i>screen_single_category</i> ma wartość True.
screen_missing_values	<i>flag</i>	Wartość True powoduje, że monitorowane są zmienne ze zbyt dużą liczbą braków danych, wyrażoną jako procent łącznej liczby rekordów.
max_missing_values	<i>number</i>	
screen_num_categories	<i>flag</i>	Wartość True powoduje, że monitorowane są zmienne ze zbyt dużą liczbą kategorii w odniesieniu do łącznej liczby rekordów.
max_num_categories	<i>number</i>	
screen_std_dev	<i>flag</i>	Wartość True powoduje, że monitorowane są zmienne z odchyleniem standardowym nie większym od określonego minimum.
min_std_dev	<i>number</i>	
screen_coeff_of_var	<i>flag</i>	Wartość True powoduje, że monitorowane są zmienne ze współczynnikiem zmienności nie większym od określonego minimum.
min_coeff_of_var	<i>number</i>	
criteria	Pearson Likelihood CramersV Lambda	Przy określaniu rankingu predyktorów jakościowych względem przewidywanej zmiennej jakościowej określa pomiar będący podstawą do wyznaczania ważności.
unimportant_below	<i>number</i>	Określa progowe wartości <i>p</i> używane do nadawania zmiennym rangi ważnych, brzegowych i nieważnych. Przyjmuje wartości z zakresu od 0,0 do 1,0.
important_above	<i>number</i>	Przyjmuje wartości z zakresu od 0,0 do 1,0.
unimportant_label	<i>string</i>	Określa etykietę rangi "nieważne".
marginal_label	<i>string</i>	
important_label	<i>string</i>	
selection_mode	ImportanceLevel ImportanceValue TopN	
select_important	<i>flag</i>	Gdy <i>selection_mode</i> ma wartość <i>ImportanceLevel</i> , określa, czy wybierać zmienne ważne

Tabela 125. Właściwości węzła *featureselectionnode* (kontynuacja)

Właściwości węzła <i>featureselectionnode</i>	Wartości	Opis właściwości
<code>select_marginal</code>	<i>flag</i>	Gdy <code>selection_mode</code> ma wartość <code>ImportanceLevel</code> , określa, czy wybierać zmienne brzegowe.
<code>select_unimportant</code>	<i>flag</i>	Gdy <code>selection_mode</code> ma wartość <code>ImportanceLevel</code> , określa, czy wybierać zmienne nieważne.
<code>importance_value</code>	<i>number</i>	Gdy <code>selection_mode</code> ma wartość <code>ImportanceValue</code> , określa wartość odcięcia, która ma być używana. Przyjmuje wartości z zakresu od 0 do 100.
<code>top_n</code>	<i>integer</i>	Gdy <code>selection_mode</code> ma wartość <code>TopN</code> , określa wartość odcięcia, która ma być używana. Przyjmuje wartości z zakresu od 0 do 1000.

Właściwości węzła *genlinnode*



Procedura ogólnych modeli liniowych rozszerza ogólny model liniowy w taki sposób, że zmienna zależna jest liniowo powiązana z czynnikami i współzmiennymi za pośrednictwem określonej funkcji łączenia. Model pozwala ponadto, aby zmienna zależna nie miała rozkładu normalnego. Obejmuje ona funkcjonalność dużej liczby modeli statystycznych, m.in. regresji liniowej, regresji logistycznej, modeli logarytmiczno-liniowych dla danych o liczebności.

Przykład

```
node = stream.create("genlin", "My node")
node.setPropertyValue("model_type", "MainAndAllTwoWayEffects")
node.setPropertyValue("offset_type", "Variable")
node.setPropertyValue("offset_field", "Claimant")
```

Tabela 126. Właściwości węzła *genlinnode*

Właściwości węzła <i>genlinnode</i>	Wartości	Opis właściwości
<code>target</code>	<i>field</i>	Uogólnione modele liniowe wymagają jednej zmiennej przewidywanej, która musi być zmienną nominalną lub flagą, i co najmniej jednej zmiennej wejściowej. Można również określić zmienną ważącą. Więcej informacji można znaleźć w temacie “Wspólne właściwości węzłów modelowania” na stronie 173.
<code>use_weight</code>	<i>flag</i>	
<code>weight_field</code>	<i>field</i>	Dopuszczalne są tylko zmienne typu ciągłego.
<code>target_represents_trials</code>	<i>flag</i>	
<code>trials_type</code>	Variable FixedValue	
<code>trials_field</code>	<i>field</i>	Dopuszczalne są zmienne typu ciągłego, flagi lub porządkowe.
<code>trials_number</code>	<i>number</i>	Wartość domyślna to 10.
<code>model_type</code>	MainEffects MainAndAllTwoWayEffects	

Tabela 126. Właściwości węzła *genlinnode* (kontynuacja)

Właściwości węzła <i>genlinnode</i>	Wartości	Opis właściwości
offset_type	Variable FixedValue	
offset_field	<i>field</i>	Dopuszczalne są tylko zmienne typu ciągłego.
offset_value	<i>number</i>	Musi być liczbą rzeczywistą.
base_category	Last First	
include_intercept	<i>flag</i>	
mode	Simple Expert	
distribution	BINOMIAL GAMMA IGAUSS NEGBIN NORMAL POISSON TWEEDIE MULTINOMIAL	IGAUSS: odwrócony Gaussa. NEGBIN: ujemny dwumianowy.
negbin_para_type	Specify Estimate	
negbin_parameter	<i>number</i>	Wartość domyślna to 1. Wartość musi być nieujemną liczbą rzeczywistą.
tweedie_parameter	<i>number</i>	
link_function	IDENTITY CLOGLOG LOG LOGC LOGIT NEGBIN NLOGLOG ODDSPOWER PROBIT POWER CUMCAUCHIT CUMCLOGLOG CUMLOGIT CUMNLOGLOG CUMPROBIT	CLOGLOG: komplementarny log-log. LOGC: log-komplementarny. NEGBIN: ujemny dwumianowy. NLOGLOG: ujemny log-log. CUMCAUCHIT: Cauchit skumulowane. CUMCLOGLOG: skumulowany komplementarny log-log CUMLOGIT: skumulowany logit. CUMNLOGLOG: skumulowany ujemny log-log. CUMPROBIT: skumulowany probit.
power	<i>number</i>	Wartość musi być niezerową liczbą rzeczywistą.
method	Hybrid Fisher NewtonRaphson	
max_fisher_iterations	<i>number</i>	Wartość domyślna to 1; dozwolone są tylko dodatnie liczby całkowite.
scale_method	MaxLikelihoodEstimate Deviance PearsonChiSquare FixedValue	
scale_value	<i>number</i>	Wartość domyślna to 1; wartość musi być większa od 0.

Tabela 126. Właściwości węzła *genlinnode* (kontynuacja)

Właściwości węzła <i>genlinnode</i>	Wartości	Opis właściwości
covariance_matrix	ModelEstimator RobustEstimator	
max_iterations	<i>number</i>	Wartość domyślna to 100; dopuszczalne są tylko nieujemne liczby całkowite.
max_step_halving	<i>number</i>	Wartość domyślna to 5; dopuszczalne są tylko dodatnie liczby całkowite.
check_separation	<i>flag</i>	
start_iteration	<i>number</i>	Wartość domyślna to 20; dozwolone są tylko dodatnie liczby całkowite.
estimates_change	<i>flag</i>	
estimates_change_min	<i>number</i>	Wartość domyślna to 1E-006; dozwolone są tylko dodatnie liczby całkowite.
estimates_change_type	Absolute Relative	
loglikelihood_change	<i>flag</i>	
loglikelihood_change_min	<i>number</i>	Dopuszczalne są tylko dodatnie liczby całkowite.
loglikelihood_change_type	Absolute Relative	
hessian_convergence	<i>flag</i>	
hessian_convergence_min	<i>number</i>	Dopuszczalne są tylko dodatnie liczby całkowite.
hessian_convergence_type	Absolute Relative	
case_summary	<i>flag</i>	
contrast_matrices	<i>flag</i>	
descriptive_statistics	<i>flag</i>	
estimable_functions	<i>flag</i>	
model_info	<i>flag</i>	
iteration_history	<i>flag</i>	
goodness_of_fit	<i>flag</i>	
print_interval	<i>number</i>	Wartość domyślna to 1; wartość musi być dodatnią liczbą całkowitą.
model_summary	<i>flag</i>	
lagrange_multiplier	<i>flag</i>	
parameter_estimates	<i>flag</i>	
include_exponential	<i>flag</i>	
covariance_estimates	<i>flag</i>	
correlation_estimates	<i>flag</i>	
analysis_type	Typel Typelll TypelAndTypelll	
statistics	Wald LR	

Tabela 126. Właściwości węzła *genlnode* (kontynuacja)

Właściwości węzła <i>genlnode</i>	Wartości	Opis właściwości
citype	Wald Profile	
tolerancelevel	<i>number</i>	Wartość domyślna to 0,0001.
confidence_interval	<i>number</i>	Wartość domyślna to 95.
loglikelihood_function	Pełne 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	

Właściwości węzła *glmnode*



Uogólniony liniowy model mieszany (GLMM) stanowi rozszerzenie modelu liniowego, w taki sposób, że przewidywany model może nie mieć standardowego rozłożenia w sposób umożliwiający liniowe powiązanie z czynnikami i współzmiennymi za pośrednictwem określonej funkcji łączenia, a także skorelowanie obserwacji. Uogólnione liniowe modele mieszane obejmują szeroki wachlarz modeli, począwszy od prostych modeli regresji liniowej, aż po złożone wielopoziomowe modele dla danych z obserwacji długofalowych nieposiadających rozkładu normalnego.

Tabela 127. Właściwości węzła *glmnode*.

Właściwości węzła <i>glmnode</i>	Wartości	Opis właściwości
residual_subject_spec	<i>structured</i>	Kombinacja wartości określonych zmiennych jakościowych, która jednoznacznie definiuje podmioty w zbiorze danych.
repeated_measures	<i>structured</i>	Zmienne używane do identyfikacji obserwacji powtórzonych.
residual_group_spec	[<i>field1 ... fieldN</i>]	Zmienne definiujące parametry kowariancji niezależnych zbiorów powtórzonych efektów.
residual_covariance_type	Diagonal AR1 ARMA11 COMPOUND_SYMMETRY IDENTITY TOEPLITZ UNSTRUCTURED VARIANCE_COMPONENTS	Określa strukturę kowariancji dla reszt.

Tabela 127. Właściwości węzła glmnode (kontynuacja).

Właściwości węzła glmnode	Wartości	Opis właściwości
custom_target	flag	Określa, czy ma być używana zmienna przewidywana zdefiniowana we wcześniejszym węźle (false), czy niestandardowa zmienna przewidywana określona przez właściwość target_field (true).
target_field	field	Zmienna docelowa, gdy custom_target ma wartość true.
use_trials	flag	Określa, czy gdy odpowiedź przewidywana jest liczbą zdarzeń występujących w zbiorze prób, ma być używana dodatkowa zmienna lub wartość określająca liczbę prób. Domyślną wartością jest false.
use_field_or_value	Field Value	Określa, czy do określania liczby prób używana jest zmienna (domyślnie), czy wartość.
trials_field	field	Zmienna określająca liczbę prób.
trials_value	integer	Wartość określająca liczbę prób. Jeśli wartość jest określona, to minimum wynosi 1.
use_custom_target_reference	flag	Określa, czy dla jakościowej zmiennej przewidywanej ma być używana niestandardowa kategoria odwołania. Domyślną wartością jest false.
target_reference_value	string	Kategoria odwołania, która ma być używana, gdy use_custom_target_reference ma wartość true.
dist_link_combination	Nominal Logit GammaLog BinomialLogit PoissonLog BinomialProbit NegbinLog BinomialLogC Custom	Wspólne modele rozkładu wartości zmiennej przewidywanej. Wybierz Użytkownika, aby określić rozkład z listy przekazanej we właściwości target_distribution.
target_distribution	Normal Binomial Multinomial Gamma Inverse NegativeBinomial Poisson	Rozkład wartości zmiennej przewidywanej, gdy dist_link_combination ma wartość Custom.
link_function_type	Identity LogC Log CLOGLOG Logit NLOGLOG PROBIT POWER CAUCHIT	Funkcja łączenia, która kojarzy wartości zmiennych przewidywanych z predyktorami. Jeśli target_distribution dla rozkładu Binomial można użyć dowolnej z wymienionych funkcji łączenia. Jeśli target_distribution Multinomial, można użyć funkcji CLOGLOG, CAUCHIT, LOGIT, NLOGLOG lub PROBIT. Jeśli target_distribution ma wartość inną niż Binomial lub Multinomial, można użyć funkcji IDENTITY, LOG lub POWER.

Tabela 127. Właściwości węzła *glmnode* (kontynuacja).

Właściwości węzła <i>glmnode</i>	Wartości	Opis właściwości
<code>link_function_param</code>	<i>number</i>	Wartość parametru funkcji łączenia. Ma zastosowanie tylko wtedy, gdy <code>normal_link_function</code> lub <code>link_function_type</code> ma wartość <code>POWER</code> .
<code>use_predefined_inputs</code>	<i>flag</i>	Określa, czy zmiennymi efektów stałych mają być zmienne zdefiniowane we wcześniejszych węzłach jako zmienne wejściowe (<code>true</code>), czy zmienne z listy <code>fixed_effects_list</code> (<code>false</code>). Domyślną wartością jest <code>false</code> .
<code>fixed_effects_list</code>	<i>structured</i>	Jeśli <code>use_predefined_inputs</code> ma wartość <code>false</code> , określa zmienne wejściowe, które mają być używane jako zmienne efektów stałych.
<code>use_intercept</code>	<i>flag</i>	Wartość <code>true</code> (domyślna) powoduje, że w modelu jest uwzględniany wyraz wolny.
<code>random_effects_list</code>	<i>structured</i>	Lista zmiennych, które mają być określone jako efekty losowe.
<code>regression_weight_field</code>	<i>field</i>	Zmienna, która ma być używana jako zmienna wagi analizy.
<code>use_offset</code>	None <code>offset_value</code> <code>offset_field</code>	Wskazuje sposób określenia przesunięcia. Wartość <code>None</code> oznacza, że nie jest używane przesunięcie.
<code>offset_value</code>	<i>number</i>	Wartość przesunięcia, gdy <code>use_offset</code> ma wartość <code>offset_value</code> .
<code>offset_field</code>	<i>field</i>	Zmienna używana jako wartość przesunięcia, gdy <code>use_offset</code> ma wartość <code>offset_field</code> .
<code>target_category_order</code>	Ascending Descending Data	Porządek sortowania jakościowych zmiennych przewidywanych. Wartość <code>Data</code> nakazuje zastosowanie porządku sortowania występującego w danych. Wartość domyślna to <code>Ascending</code> (rosnąco).
<code>inputs_category_order</code>	Ascending Descending Data	Porządek sortowania predyktorów jakościowych. Wartość <code>Data</code> nakazuje zastosowanie porządku sortowania występującego w danych. Wartość domyślna to <code>Ascending</code> (rosnąco).
<code>max_iterations</code>	<i>integer</i>	Maksymalna liczba iteracji, jaką wykona algorytm. Nieujemna liczba całkowita; wartość domyślna wynosi 100.
<code>confidence_level</code>	<i>integer</i>	Poziom ufności używany do obliczania oszacowań przedziałów współczynników modelu. Nieujemna liczba całkowita; maksimum wynosi 100; wartość domyślna wynosi 95.
<code>degrees_of_freedom_method</code>	Fixed Varied	Określa sposób obliczania stopni swobody dla testów istotności.
<code>test_fixed_effects_coefficients</code>	Model Robust	Metoda obliczania macierzy kowariancji oszacowań parametrów.
<code>use_p_converge</code>	<i>flag</i>	Opcja zbieżności parametru.
<code>p_converge</code>	<i>number</i>	Wartość pusta lub dowolna dodatnia.
<code>p_converge_type</code>	Absolute Relative	

Tabela 127. Właściwości węzła *glmnode* (kontynuacja).

Właściwości węzła <i>glmnode</i>	Wartości	Opis właściwości
<code>use_l_converge</code>	<i>flag</i>	Opcja zbieżności dla logarytmu wiarygodności.
<code>l_converge</code>	<i>number</i>	Wartość pusta lub dowolna dodatnia.
<code>l_converge_type</code>	Absolute Relative	
<code>use_h_converge</code>	<i>flag</i>	Opcja zbieżności Hessego.
<code>h_converge</code>	<i>number</i>	Wartość pusta lub dowolna dodatnia.
<code>h_converge_type</code>	Absolute Relative	
<code>max_fisher_steps</code>	<i>integer</i>	
<code>singularity_tolerance</code>	<i>number</i>	
<code>use_model_name</code>	<i>flag</i>	Wskazuje, czy ma być określana niestandardowa nazwa modelu (<i>true</i>), czy używana nazwa wygenerowana przez system (<i>false</i>). Domyślną wartością jest <i>false</i> .
<code>model_name</code>	<i>string</i>	Jeśli <code>use_model_name</code> ma wartość <i>true</i> , określa nazwę modelu.
<code>confidence</code>	<i>onProbability</i> <i>onIncrease</i>	Podstawa obliczania współczynnika ufności oceny: najwyższe przewidywane prawdopodobieństwo albo różnica między najwyższym a drugim co do wielkości prawdopodobieństwem.
<code>score_category_probabilities</code>	<i>flag</i>	Wartość <i>true</i> powoduje generowanie przewidywanych prawdopodobieństw dla jakościowych zmiennych przewidywanych. Domyślną wartością jest <i>false</i> .
<code>max_categories</code>	<i>integer</i>	Jeśli <code>score_category_probabilities</code> ma wartość <i>true</i> , określa maksymalną liczbę kategorii do zapisania.
<code>score_propensity</code>	<i>flag</i>	Wartość <i>true</i> powoduje, że generowane są oceny skłonności dla zmiennych przewidywanych typu flaga; oceny te określają prawdopodobieństwo wyniku „prawdziwego” dla danej zmiennej.
<code>emeans</code>	<i>structure</i>	Dla każdej zmiennej jakościowej z listy efektów stałych określa, czy mają być generowane szacowane średnie brzegowe.
<code>covariance_list</code>	<i>structure</i>	Dla każdej zmiennej ciągłej z listy efektów stałych określa, czy do obliczania szacowanych średnich brzegowych ma być używana średnia, czy wartość niestandardowa.
<code>mean_scale</code>	Original Transformed	Określa, czy szacowane średnie brzegowe są obliczane na podstawie pierwotnej skali zmiennej przewidywanej (ustawienie domyślne), czy na podstawie transformacji funkcji łączenia.
<code>comparison_adjustment_method</code>	LSD SEQBONFERRONI SEQSIDAK	Metoda korekty, która ma być używana przy testowaniu hipotez z użyciem wielu kontrastów.

Właściwości węzła gle



GLE stanowi wersję modelu liniowego rozszerzoną w taki sposób, że zmienna przewidywana może mieć rozkład inny niż normalny, jest liniowo powiązana z czynnikami i współzmiennymi za pośrednictwem określonej funkcji łączenia, a obserwacje mogą być skorelowane. Uogólnione liniowe modele mieszane obejmują szeroki wachlarz modeli, począwszy od prostych modeli regresji liniowej, aż po złożone wielopoziomowe modele dla danych z obserwacji długofalowych nieposiadających rozkładu normalnego.

Tabela 128. Właściwości węzła gle

Właściwości węzła gle	Wartości	Opis właściwości
custom_target	<i>flag</i>	Określa, czy ma być używana zmienna przewidywana zdefiniowana we wcześniejszym węźle (<i>false</i>), czy niestandardowa zmienna przewidywana określona przez właściwość <i>target_field</i> (<i>true</i>).
target_field	<i>field</i>	Zmienna docelowa, gdy <i>custom_target</i> ma wartość <i>true</i> .
use_trials	<i>flag</i>	Określa, czy gdy odpowiedź przewidywana jest liczbą zdarzeń występujących w zbiorze prób, ma być używana dodatkowa zmienna lub wartość określająca liczbę prób. Domyślną wartością jest <i>false</i> .
use_trials_field_or_value	Field Value	Określa, czy do określania liczby prób używana jest zmienna (domyślnie), czy wartość.
trials_field	<i>field</i>	Zmienna określająca liczbę prób.
trials_value	<i>integer</i>	Wartość określająca liczbę prób. Jeśli wartość jest określona, to minimum wynosi 1.
use_custom_target_reference	<i>flag</i>	Określa, czy dla jakościowej zmiennej przewidywanej ma być używana niestandardowa kategoria odwołania. Domyślną wartością jest <i>false</i> .
target_reference_value	<i>string</i>	Kategoria odwołania, która ma być używana, gdy <i>use_custom_target_reference</i> ma wartość <i>true</i> .
dist_link_combination	NormalIdentity GammaLog PoissonLog NegbinLog TweedieIdentity NominalLogit BinomialLogit BinomialProbit BinomialLogC CUSTOM	Wspólne modele rozkładu wartości zmiennej przewidywanej. Wybierz CUSTOM, aby określić rozkład z listy przekazanej we właściwości <i>target_distribution</i> .
target_distribution	Normal Binomial Multinomial Gamma INVERSE_GAUSS NEG_BINOMIAL Poisson TWEEDIE UNKNOWN	Rozkład wartości zmiennej przewidywanej, gdy <i>dist_link_combination</i> ma wartość Custom.

Tabela 128. Właściwości węzła gle (kontynuacja)

Właściwości węzła gle	Wartości	Opis właściwości
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	Funkcja łączenia, która kojarzy wartości przewidywane z predyktorami. Jeśli target_distribution ma wartość Binomial, można użyć wartości: UNKNOWN IDENTITY LOG LOGIT PROBIT COMPL_LOG_LOG POWER LOG_COMPL NEG_LOG_LOG ODDS_POWER Jeśli target_distribution ma wartość NEG_BINOMIAL, można użyć wartości: NEG_BINOMIAL. Jeśli target_distribution ma wartość UNKNOWN, można użyć wartości: GEN_LOGIT CUMUL_LOGIT CUMUL_PROBIT CUMUL_COMPL_LOG_LOG CUMUL_NEG_LOG_LOG CUMUL_CAUCHIT
link_function_param	number	Wartość parametru Tweediego. Ma zastosowanie tylko wtedy, gdy normal_link_function lub link_function_type ma wartość POWER.
tweedie_param	number	Wartość parametru funkcji łączenia. Ma zastosowanie tylko wtedy, gdy dist_link_combination ma wartość Tweedieidentity lub link_function_type ma wartość TWEEDIE.
use_predefined_inputs	flag	Określa, czy zmiennymi efektów modelu mają być zmienne zdefiniowane we wcześniejszych węzłach jako zmienne wejściowe (true), czy zmienne z listy fixed_effects_list (false).
model_effects_list	structured	Jeśli use_predefined_inputs ma wartość false, określa zmienne wejściowe, które mają być używane jako zmienne efektów modelu.
use_intercept	flag	Wartość true (domyślna) powoduje, że w modelu jest uwzględniany wyraz wolny.
regression_weight_field	field	Zmienna, która ma być używana jako zmienna wagi analizy.
use_offset	None Value Variable	Wskazuje sposób określenia przesunięcia. Wartość None oznacza, że nie jest używane przesunięcie.
offset_value	number	Wartość przesunięcia, gdy use_offset ma wartość offset_value.
offset_field	field	Zmienna używana jako wartość przesunięcia, gdy use_offset ma wartość offset_field.

Tabela 128. Właściwości węzła gle (kontynuacja)

Właściwości węzła gle	Wartości	Opis właściwości
target_category_order	Ascending Descending	Porządek sortowania jakościowych zmiennych przewidywanych. Wartość domyślna to Ascending (rosnąco).
inputs_category_order	Ascending Descending	Porządek sortowania predyktorów jakościowych. Wartość domyślna to Ascending (rosnąco).
max_iterations	<i>integer</i>	Maksymalna liczba iteracji, jaką wykona algorytm. Nieujemna liczba całkowita; wartość domyślna wynosi 100.
confidence_level	<i>number</i>	Poziom ufności używany do obliczania oszacowań przedziałów współczynników modelu. Nieujemna liczba całkowita; maksimum wynosi 100; wartość domyślna wynosi 95.
test_fixed_effects_coefficients	Model Robust	Metoda obliczania macierzy kowariancji oszacowań parametrów.
detect_outliers	<i>flag</i>	Wartość true powoduje, że algorytm znajduje wpływowe wartości odstające dla wszystkich rozkładów z wyjątkiem wielomianowego.
conduct_trend_analysis	<i>flag</i>	Wartość true powoduje, że algorytm przeprowadza analizę trendów na wykresie rozrzutu.
estimation_method	FISHER_SCORING NEWTON_RAPHSON HYBRID	Określa algorytm estymacji maksymalnej wiarygodności.
max_fisher_iterations	<i>integer</i>	Gdy używane jest FISHER_SCORING estimation_method, jest to maksymalna liczba iteracji. Minimum 0, maksimum 20.
scale_parameter_method	MLE FIXED DEVIANCE PEARSON_CHISQUARE	Określa metodę estymacji parametru skali.
scale_value	<i>number</i>	Dostępna tylko wtedy, gdy scale_parameter_method ma wartość Fixed.
negative_binomial_method	MLE FIXED	Określa metodę estymacji parametru pomocniczego rozkładu ujemnego dwumianowego.
negative_binomial_value	<i>number</i>	Dostępna tylko wtedy, gdy negative_binomial_method ma wartość Fixed.
use_p_converge	<i>flag</i>	Opcja zbieżności parametru.
p_converge	<i>number</i>	Wartość pusta lub dowolna dodatnia.
p_converge_type	<i>flag</i>	True = bezwzględna, False = względna
use_l_converge	<i>flag</i>	Opcja zbieżności dla logarytmu wiarygodności.
l_converge	<i>number</i>	Wartość pusta lub dowolna dodatnia.
l_converge_type	<i>flag</i>	True = bezwzględna, False = względna
use_h_converge	<i>flag</i>	Opcja zbieżności Hessego.
h_converge	<i>number</i>	Wartość pusta lub dowolna dodatnia.
h_converge_type	<i>flag</i>	True = bezwzględna, False = względna
max_iterations	<i>integer</i>	Maksymalna liczba iteracji, jaką wykona algorytm. Nieujemna liczba całkowita; wartość domyślna wynosi 100.

Tabela 128. Właściwości węzła gle (kontynuacja)

Właściwości węzła gle	Wartości	Opis właściwości
sing_tolerance	<i>integer</i>	
use_model_selection	<i>flag</i>	Umożliwia określanie progu parametru i metody wyboru modelu.
method	LASSO ELASTIC_NET FORWARD_STEPWISE RIDGE	Określa metodę wyboru modelu lub, w przypadku opcji Ridge, metodę regularyzacji.
detect_two_way_interactions	<i>flag</i>	<p>Wartość True powoduje, że model będzie automatycznie wykrywał dwukierunkowe interakcje między zmiennymi wejściowymi.</p> <p>Ten element sterujący powinien być włączony tylko wtedy, gdy model ma tylko główne efekty (tj. użytkownik nie utworzył efektów wyższego rzędu), a właściwość method ma wartość Forward Stepwise, Lasso lub Elastic Net.</p>
automatic_penalty_params	<i>flag</i>	<p>Dostępna, gdy metoda wyboru modelu (method) to Lasso lub Elastic Net.</p> <p>Ta funkcja służy do wprowadzania parametrów kary związanych z metodą wyboru Lasso albo Elastic Net.</p> <p>Wartość True powoduje, że używane są wartości domyślne. Wartość False powoduje, że włączone są parametry kary i można wprowadzać wartości niestandardowe.</p>
lasso_penalty_param	<i>number</i>	Dostępna tylko wtedy, gdy metoda wyboru modelu (method) to Lasso lub Elastic Net, a automatic_penalty_params ma wartość False. Określa parametr kary dla metody Lasso.
elastic_net_penalty_param1	<i>number</i>	Dostępna tylko wtedy, gdy metoda wyboru modelu (method) to Lasso lub Elastic Net, a automatic_penalty_params ma wartość False. Określa parametr kary dla parametru nr 1 metody Elastic Net.
elastic_net_penalty_param2	<i>number</i>	Dostępna tylko wtedy, gdy metoda wyboru modelu (method) to Lasso lub Elastic Net, a automatic_penalty_params ma wartość False. Określa parametr kary dla parametru nr 2 metody Elastic Net.
probability_entry	<i>number</i>	Dostępna tylko wtedy, gdy method ma wartość Forward Stepwise. Określa poziom istotności kryterium statystyki f przy uwzględnianiu efektów.
probability_removal	<i>number</i>	Dostępna tylko wtedy, gdy method ma wartość Forward Stepwise. Określa poziom istotności kryterium statystyki f przy usuwaniu efektów.

Tabela 128. Właściwości węzła *gle* (kontynuacja)

Właściwości węzła <i>gle</i>	Wartości	Opis właściwości
<code>use_max_effects</code>	<i>flag</i>	Dostępna tylko wtedy, gdy <code>method</code> ma wartość <code>Forward Stepwise</code> . Włącza możliwość stosowania właściwości <code>max_effects</code> . Gdy ma wartość <code>False</code> , domyślna liczba efektów powinna być równa łącznej liczbie efektów przekazanych do modelu pomniejszonej o wyraz wolny.
<code>max_effects</code>	<i>integer</i>	Określa maksymalną liczbę efektów w metodzie krokowej postępującej.
<code>use_max_steps</code>	<i>flag</i>	Włącza możliwość stosowania właściwości <code>max_steps</code> . Gdy ma wartość <code>False</code> , domyślna liczba efektów powinna być równa trzykrotności liczby efektów przekazanych do modelu pomniejszonej o wyraz wolny.
<code>max_steps</code>	<i>integer</i>	Określa maksymalną liczbę kroków, gdy jako metodę budowania (<code>method</code>) wybrano <code>Forward Stepwise</code> .
<code>use_model_name</code>	<i>flag</i>	Wskazuje, czy ma być określana niestandardowa nazwa modelu (<code>true</code>), czy używana nazwa wygenerowana przez system (<code>false</code>). Domyślną wartością jest <code>false</code> .
<code>model_name</code>	<i>string</i>	Jeśli <code>use_model_name</code> ma wartość <code>true</code> , określa nazwę modelu.
<code>usePI</code>	<i>flag</i>	Gdy ma wartość <code>true</code> , obliczana jest ważność predyktorów.

Właściwości węzła *kmeansnode*



Węzeł *K-średnie* grupuje zbiór danych w osobne grupy (lub skupienia). Metoda ta definiuje stałą liczbę skupień, w sposób iteracyjny przypisuje rekordy do skupień i dopasowuje centra skupień do chwili, gdy dalsze pokrycie nie będzie miało wpływu na ulepszenie modelu. Zamiast prób predykcji danych wynikowych *k-średnia* korzysta z procesu znanego jako nienadzorowane uczenie w celu ujawnienia wzorców w zbiorze zmiennych wejściowych.

Przykład

```
node = stream.create("kmeans", "My node")
# Karta "Zmienne"
node.setPropertyValue("custom_fields", True)
node.setPropertyValue("inputs", ["Cholesterol", "BP", "Drug", "Na", "K", "Age"])
# Karta "Model"
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")
# Karta "Zaawansowane"
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)

```

Tabela 129. Właściwości węzła kmeansnode

Właściwości węzła kmeansnode	Wartości	Opis właściwości
inputs	[field1 ... fieldN]	Modele k-średnich przeprowadzają analizę skupień na zbiorze zmiennych wejściowych, ale nie korzystają ze zmiennej przewidywanej. Zmienne wagi i częstości nie są używane. Więcej informacji można znaleźć w temacie “Wspólne właściwości węzłów modelowania” na stronie 173.
num_clusters	number	
gen_distance	flag	
cluster_label	String Number	
label_prefix	string	
mode	Simple Expert	
stop_on	Default Custom	
max_iterations	number	
tolerance	number	
encoding_value	number	
optimize	Speed Memory	Służy do określania, czy budowanie modelu ma być optymalizowane pod kątem szybkości, czy zużycia pamięci.

Właściwości węzła knnnode



Węzeł k -najbliższego sąsiedztwa (KNN) wiąże nową obserwację z kategorią lub wartością k (gdzie k jest liczbą całkowitą) najbliższych obiektów w przestrzeni predyktora. Podobne obserwacje znajdują się blisko siebie, a niepodobne – daleko.

Przykład

```

node = stream.create("knn", "My node")
# Karta Cele
node.setPropertyValue("objective", "Custom")
# Karta Ustawienia - panel Sąsiedzi
node.setPropertyValue("automatic_k_selection", False)
node.setPropertyValue("fixed_k", 2)
node.setPropertyValue("weight_by_importance", True)
# Karta Ustawienia - panel Analiza
node.setPropertyValue("save_distances", True)

```

Tabela 130. Właściwości węzła knnnode

Właściwości węzła knnnode	Wartości	Opis właściwości
analysis	PredictTarget IdentifyNeighbors	
objective	Balance Speed Accuracy Custom	
normalize_ranges	<i>flag</i>	
use_case_labels	<i>flag</i>	Zaznacz pole, aby aktywować następną opcję.
case_labels_field	<i>field</i>	
identify_focal_cases	<i>flag</i>	Zaznacz pole, aby aktywować następną opcję.
focal_cases_field	<i>field</i>	
automatic_k_selection	<i>flag</i>	
fixed_k	<i>integer</i>	Aktywna tylko wtedy, gdy automatic_k_selectio ma wartość False.
minimum_k	<i>integer</i>	Aktywna tylko wtedy, gdy automatic_k_selectio ma wartość True.
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</i> ... <i>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>	Aktywna tylko wtedy, gdy validation_fold_assign_by_field ma wartość False
set_random_seed	<i>flag</i>	
random_seed	<i>number</i>	
folds_field	<i>field</i>	Aktywna tylko wtedy, gdy validation_fold_assign_by_field ma wartość True
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	

Właściwości węzła kohonennode



Węzeł Kohonen generuje typ sieci neuronowej, którą można wykorzystać do grupowania zbioru danych w osobne grupy. Po pełnym przeszkoleniu sieci rekordy podobne do siebie powinny znajdować się blisko siebie na mapie wyników, podczas gdy rekordy różne od siebie powinny znajdować się daleko od siebie. Na podstawie liczby obserwacji przechwyconych przez każdą jednostkę w modelu użytkowym można rozpoznać silne jednostki. Może to dać pojęcie o odpowiedniej liczbie skupień.

Przykład

```
node = stream.create("kohonen", "My node")
# Karta "Model"
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")
# Karta "Zaawansowane"
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)
```

Tabela 131. Właściwości węzła kohonennode

Właściwości węzła kohonennode	Wartości	Opis właściwości
inputs	[field1 ... fieldN]	W modelach Sieć Kohonena używana jest lista zmiennych wejściowych, ale nie są używane zmienne przewidywane. Zmienne częstości i wazące nie są używane. Więcej informacji można znaleźć w temacie “Wspólne właściwości węzłów modelowania” na stronie 173.
continue	flag	
show_feedback	flag	
stop_on	Default Time	
time	number	
optimize	Speed Memory	Służy do określania, czy budowanie modelu ma być optymalizowane pod kątem szybkości, czy zużycia pamięci.
cluster_label	flag	
mode	Simple Expert	
width	number	
length	number	
decay_style	Linear Exponential	

Tabela 131. Właściwości węzła kohonenode (kontynuacja)

Właściwości węzła kohonenode	Wartości	Opis właściwości
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>	

Właściwości węzła linearnode



Modele regresji liniowej przewidują docelową wartość ilościową na podstawie liniowych relacji między docelową wartością ilościową a jednym lub większą liczbą predyktorów.

Przykład

```
node = stream.create("linear", "My node")
# Karta Opcje budowania - panel Cele
node.setPropertyValue("objective", "Standard")
# Karta Opcje budowania - panel Wybór modelu
node.setPropertyValue("model_selection", "BestSubsets")
node.setPropertyValue("criteria_best_subsets", "ASE")
# Karta Opcje budowania - panel Zespół
node.setPropertyValue("combining_rule_categorical", "HighestMeanProbability")
```

Tabela 132. Właściwości węzła linearnode.

Właściwości węzła linearnode	Wartości	Opis właściwości
target	<i>field</i>	Określa jedną zmienną przewidywaną.
inputs	[<i>field1 ... fieldN</i>]	Predyktory używane przez model.
continue_training_existing_model	<i>flag</i>	
objective	Standard Bagging Boosting psm	Wartość psm jest używana w przypadku bardzo dużych zbiorów danych i wymaga połączenia z serwerem.
use_auto_data_preparation	<i>flag</i>	
confidence_level	<i>number</i>	
model_selection	ForwardStepwise BestSubsets None	
criteria_forward_stepwise	AICC Fstatistics AdjustedRSquare ASE	
probability_entry	<i>number</i>	
probability_removal	<i>number</i>	

Tabela 132. Właściwości węzła linearnode (kontynuacja).

Właściwości węzła linearnode	Wartości	Opis właściwości
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>	

Właściwości węzła linearasnode



Modele regresji liniowej przewidują docelową wartość ilościową na podstawie liniowych relacji między docelową wartością ilościową a jednym lub większą liczbą predyktorów.

Tabela 133. Właściwości węzła linearasnode

Właściwości węzła linearasnode	Wartości	Opis właściwości
target	<i>field</i>	Określa jedną zmienną przewidywaną.
inputs	[<i>field1</i> ... <i>fieldN</i>]	Predyktory używane przez model.
weight_field	<i>field</i>	Zmienna analizy używana przez model.
custom_fields	<i>flag</i>	Wartość domyślna to TRUE.
intercept	<i>flag</i>	Wartość domyślna to TRUE.
detect_2way_interaction	<i>flag</i>	Określa, czy mają być brane pod uwagę interakcje dwukierunkowe. Wartość domyślna to TRUE.
cin	<i>number</i>	Przedział ufności używany do obliczania oszacowań współczynników modelu. Należy podać wartość większą od 0 i mniejszą od 100. Domyślna wartość to 95.

Tabela 133. Właściwości węzła *linearasnode* (kontynuacja)

Właściwości węzła <i>linearasnode</i>	Wartości	Opis właściwości
factor_order	ascending descending	Porządek sortowania predyktorów jakościowych. Wartość domyślna to <i>ascending</i> , czyli rosnąco.
var_select_method	ForwardStepwise BestSubsets none	Metoda wyboru modelu. Wartość domyślna to <i>ForwardStepwise</i> .
criteria_for_forward_stepwise	AICC Fstatistics AdjustedRSquare ASE	Statystyka używana do określenia, czy należy dodać lub usunąć efekt z modelu. Wartość domyślna to <i>AdjustedRSquare</i> .
pin	<i>number</i>	Efekt, którego najmniejsza wartość <i>p</i> jest mniejsza od określonego progu <i>pin</i> , zostanie dodany do modelu. Domyślna wartość to 0,05.
pout	<i>number</i>	Wszystkie efekty w modelu, których wartości <i>p</i> są większe od określonego progu <i>pout</i> , zostaną usunięte. Domyślna wartość to 0,10.
use_custom_max_effects	<i>flag</i>	Określa, czy w ostatecznym modelu ma obowiązywać maksymalna liczba efektów. Wartość domyślna to <i>FALSE</i> .
max_effects	<i>number</i>	Maksymalna liczba efektów, jaka ma być używana w ostatecznym modelu. Domyślna wartość to 1.
use_custom_max_steps	<i>flag</i>	Określa, czy ma obowiązywać maksymalna liczba kroków. Wartość domyślna to <i>FALSE</i> .
max_steps	<i>number</i>	Maksymalna liczba kroków, po której algorytm krokowy przerwie działanie. Domyślna wartość to 1.
criteria_for_best_subsets	AICC AdjustedRSquare ASE	Tryb kryteriów, który ma być używany. Wartość domyślna to <i>AdjustedRSquare</i> .

Właściwości węzła *logregnode*



Regresja logistyczna to technika statystyczna umożliwiająca klasyfikację rekordów na podstawie wartości zmiennych wejściowych. Jest ona analogiczna do regresji liniowej, lecz bazuje na przewidywanej zmiennej jakościowej zamiast na rozstępie liczbowym.

Przykład dla funkcji wielomianowej

```
node = stream.create("logreg", "My node")
# Karta "Zmienne"
node.setPropertyValue("custom_fields", True)
node.setPropertyValue("target", "Drug")
node.setPropertyValue("inputs", ["BP", "Cholesterol", "Age"])
node.setPropertyValue("partition", "Test")
# Karta "Model"
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)
# Karta "Zaawansowane"
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")
# Sekcja "Zbieżność..." 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)
# Sekcja "Wyniki..." 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)
# Opcje "Krokowa"
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")

```

Przykład dla funkcji dwumianowej

```

node = stream.create("logreg", "My node")
# Karta "Zmienne"
node.setPropertyValue("custom_fields", True)
node.setPropertyValue("target", "Cholesterol")
node.setPropertyValue("inputs", [{"BP", "Drug", "Age"}])
node.setPropertyValue("partition", "Test")
# Karta "Model"
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)
# Karta "Zaawansowane"
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")
# Sekcja "Zbieżność..." node.setPropertyValue("max_iterations", 50)
node.setPropertyValue("l_converge", "1.0E-3")
node.setPropertyValue("p_converge", "1.0E-7")

```

```
# Sekcja "Wyniki..." 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)
# Opcje "Krokowa"
node.setPropertyValue("binomial_removal_criterion", "LR")
node.setPropertyValue("binomial_probability_removal", 0.2)
```

Tabela 134. Właściwości węzła logregnode.

Właściwości węzła logregnode	Wartości	Opis właściwości
target	<i>field</i>	Modele regresji logistycznej wymagają jednej zmiennej przewidywanej i co najmniej jednej zmiennej wejściowej. Zmienne częstości i wazące nie są używane. Więcej informacji można znaleźć w temacie "Wspólne właściwości węzłów modelowania" na stronie 173.
logistic_procedure	Binomial Multinomial	
include_constant	<i>flag</i>	
mode	Simple Expert	
method	Enter Stepwise Forwards Backwards BackwardsStepwise	
binomial_method	Enter Forwards Backwards	
model_type	MainEffects FullFactorial Custom	Gdy jako typ modelu wybrany jest FullFactorial, metody krokowe nie będą używane, nawet jeśli zostały wybrane. Zamiast nich stosowana będzie metoda Enter. Jeśli jako typ modelu wybrano Custom, ale nie określono zmiennych niestandardowych, zbudowany zostanie model efektów głównych.
custom_terms	<i>[BP Sex][BP][Age]</i>	
multinomial_base_category	<i>string</i>	Określa sposób ustalania kategorii odniesienia.
binomial_categorical_input	<i>string</i>	
binomial_input_contrast	Indicator Simple Difference Helmert Repeated Polynomial Deviation	Właściwość wprowadzana dla zmiennych jakościowych, która określa sposób wyznaczania kontrastu.
binomial_input_category	First Last	Właściwość wprowadzana dla zmiennych jakościowych, która określa sposób wyznaczania kategorii odniesienia.

Tabela 134. Właściwości węzła logregnode (kontynuacja).

Właściwości węzła logregnode	Wartości	Opis właściwości
scale	None UserDefined Pearson Deviance	
scale_value	<i>number</i>	
all_probabilities	<i>flag</i>	
tolerance	1.0E-5 1.0E-6 1.0E-7 1.0E-8 1.0E-9 1.0E-10	
min_terms	<i>number</i>	
use_max_terms	<i>flag</i>	
max_terms	<i>number</i>	
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>	

Tabela 134. Właściwości węzła logregnode (kontynuacja).

Właściwości węzła logregnode	Wartości	Opis właściwości
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	
binomial_goodness_of_fit	<i>flag</i>	
binomial_parameters	<i>flag</i>	
binomial_iteration_history	<i>flag</i>	
binomial_classification_plots	<i>flag</i>	
binomial_ci_enable	<i>flag</i>	
binomial_ci	<i>number</i>	
binomial_residual	outliers all	
binomial_residual_enable	<i>flag</i>	
binomial_outlier_threshold	<i>number</i>	
binomial_classification_cutoff	<i>number</i>	
binomial_removal_criterion	LR Wald Conditional	
calculate_variable_importance	<i>flag</i>	
calculate_raw_propensities	<i>flag</i>	

Właściwości węzła lsvmnode



Węzeł LSVM umożliwia klasyfikację danych do jednej lub dwu grup bez przeuczenia. Algorytm LSVM jest liniowy i działa prawidłowo z szerokimi zbiorami danych, na przykład zbiorami o bardzo dużej liczbie rekordów.

Tabela 135. Właściwości węzła lsvmnode

Właściwości węzła lsvmnode	Wartości	Opis właściwości
intercept	<i>flag</i>	Powoduje uwzględnienie wyrazu wolnego w modelu. Wartość domyślna to True.
target_order	Ascending Descending	Określa porządek sortowania jakościowych zmiennych przewidywanych. Ignorowana w przypadku ciągłych zmiennych przewidywanych. Wartość domyślna to Ascending (rosnąco).

Tabela 135. Właściwości węzła *lsvmnode* (kontynuacja)

Właściwości węzła <i>lsvmnode</i>	Wartości	Opis właściwości
precision	<i>number</i>	Używana tylko wtedy, gdy zmienna przewidywana jest ciągła. Określa parametr związany z czułością reakcji na utratę regresji. Wartość minimalna to 0, a wartość maksymalna nie jest określona. Wartość domyślna to 0,1.
exclude_missing_values	<i>flag</i>	Wartość True powoduje, że rekord jest wykluczany, jeśli brakuje choć jednej wartości. Wartość domyślna to False.
penalty_function	E1 E2	Określa typ używanej funkcji kary. Domyślna wartość to L2.
lambda	<i>number</i>	Parametr kary (regularyzacji).
calculate_variable_importance	<i>flag</i>	W przypadku modeli generujących odpowiednią miarę ważności powoduje wygenerowanie tabeli wskazującej wagę względną każdego predyktora w procesie estymacji modelu. Należy zauważyć, że obliczenie ważności predyktora może potrwać dłużej dla niektórych modeli, szczególnie w przypadku pracy z dużymi zbiorami danych, i domyślnie ta opcja dla niektórych modeli jest wyłączona. Waga zmiennych jest niedostępna dla modeli listy decyzyjnej.

Właściwości węzła *neuralnetnode*

Ważne: W tej wersji produktu dostępna jest nowa wersja węzła modelowania sieci neuronowej, która oferuje udoskonalone funkcje. Została ona opisana w następnej sekcji (*neuralnetwork*). Mimo że nadal możliwe jest budowanie i ocenianie modelu w poprzedniej wersji, zalecamy zaktualizowanie skryptów tak, by korzystały z nowej wersji. Szczegółowe informacje o poprzedniej wersji zamieszczono w celach referencyjnych.

Przykład

```
node = stream.create("neuralnet", "My node")
# Karta "Zmienne"
node.setPropertyValue("custom_fields", True)
node.setPropertyValue("targets", ["Drug"])
node.setPropertyValue("inputs", ["Age", "Na", "K", "Cholesterol", "BP"])
# Karta "Model"
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")
```

```
# "Zaawansowane opcje metody Wieloraka" – sekcja
node.setPropertyValue("m_topologies", "5 30 5; 2 20 3, 1 10 1")
node.setPropertyValue("m_non_pyramids", False)
node.setPropertyValue("m_persistence", 100)
```

Tabela 136. Właściwości węzła *neuralnetnode*

Właściwości węzła <i>neuralnetnode</i>	Wartości	Opis właściwości
targets	[<i>field1 ... fieldN</i>]	Węzeł Sieci neuronowe oczekuje co najmniej jednej zmiennej przewidywanej i co najmniej jednej zmiennej wejściowej. Zmienne częstości i wazące są ignorowane. Więcej informacji można znaleźć w temacie "Wspólne właściwości węzłów modelowania" na stronie 173.
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	Tryb zatrzymywania.
accuracy	<i>number</i>	Dokładność zatrzymywania.
cycles	<i>number</i>	Liczba cykli uczenia.
time	<i>number</i>	Czas uczenia (w minutach).
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	
hl_units_one	<i>number</i>	
hl_units_two	<i>number</i>	
hl_units_three	<i>number</i>	

Tabela 136. Właściwości węzła neuralnetnode (kontynuacja)

Właściwości węzła neuralnetnode	Wartości	Opis właściwości
persistence	number	
m_topologies	string	
m_non_pyramids	flag	
m_persistence	number	
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	Służy do określania, czy budowanie modelu ma być optymalizowane pod kątem szybkości, czy zużycia pamięci.
calculate_variable_importance	flag	Uwaga: Właściwość sensitivity_analysis używana w poprzednich wersjach jest nieaktualna i została zastąpiona przez tę właściwość. Stara właściwość nadal jest używana, ale zaleca się używanie właściwości calculate_variable_importance.
calculate_raw_propensities	flag	
calculate_adjusted_propensities	flag	
adjusted_propensity_partition	Test Validation	

Właściwości węzła neuralnetworknode



Węzeł sieci neuronowej korzysta z uproszczonego modelu przetwarzania informacji przez ludzki umysł. Polega on na symulowaniu dużej liczby połączonych wzajemnie jednostek prostego przetwarzania, które przypominają abstrakcyjne wersje neuronów. Sieci neuronowe są estymatorami funkcji ogólnych o dużej wydajności, a do uczenia i stosowania ich wymagane jest tylko minimum wiedzy w zakresie statystyki lub matematyki.

Przykład


```

node = stream.create("neuralnetwork", "My node")
# Karta Opcje budowania - panel Cele
node.setPropertyValue("objective", "Standard")
# Karta Opcje budowania - panel Zespół
node.setPropertyValue("combining_rule_categorical", "HighestMeanProbability")

```

Tabela 137. Właściwości węzła *neuralnetworknode*

Właściwości węzła neuralnetworknode	Wartości	Opis właściwości
targets	[<i>field1</i> ... <i>fieldN</i>]	Określa zmienne przewidywane.
inputs	[<i>field1</i> ... <i>fieldN</i>]	Predyktory używane przez model.
splits	[<i>field1</i> ... <i>fieldN</i>]	Określa zmienną lub zmienne używane do modelowania podziałów.
use_partition	<i>flag</i>	Jeśli zdefiniowano zmienną dzielącą na podzbiory, ta opcja umożliwia użycie podczas budowania modelu wyłącznie danych z podzbioru uczącego.
continue	<i>flag</i>	Kontynuuj uczenie istniejącego modelu.
objective	Standard Bagging Boosting psm	Wartość psm jest używana w przypadku bardzo dużych zbiorów danych i wymaga połączenia z serwerem.
method	MultilayerPerceptron RadialBasisFunction	
use_custom_layers	<i>flag</i>	
first_layer_units	<i>number</i>	
second_layer_units	<i>number</i>	
use_max_time	<i>flag</i>	
max_time	<i>number</i>	
use_max_cycles	<i>flag</i>	
max_cycles	<i>number</i>	
use_min_accuracy	<i>flag</i>	
min_accuracy	<i>number</i>	
combining_rule_categorical	Voting HighestProbability HighestMeanProbability	
combining_rule_continuous	Mean Median	
component_models_n	<i>number</i>	
overfit_prevention_pct	<i>number</i>	
use_random_seed	<i>flag</i>	
random_seed	<i>number</i>	
missing_values	listwiseDeletion missingValueImputation	
use_model_name	<i>boolean</i>	
model_name	<i>string</i>	

Tabela 137. Właściwości węzła neuralnetworknode (kontynuacja)

Właściwości węzła neuralnetworknode	Wartości	Opis właściwości
confidence	onProbability onIncrease	
score_category_probabilities	flag	
max_categories	number	
score_propensity	flag	
use_custom_name	flag	
custom_name	string	
tooltip	string	
keywords	string	
annotation	string	

Właściwości węzła questnode



Węzeł QUEST oferuje metodę klasyfikacji binarnej służącą do budowania drzew decyzyjnych, zaprojektowaną w celu redukcji czasu przetwarzania analiz dużych drzew decyzyjnych C&R, a jednocześnie w celu redukcji tendencji obecnej w metodach drzew klasyfikacji do preferowania danych wejściowych dopuszczających więcej podziałów. Zmienne wejściowe mogą być zakresami liczbowymi (ciągłymi), lecz zmienna przewidywana musi być jakościowa. Wszystkie podziały są binarne.

Przykład

```
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)
```

Tabela 138. Właściwości węzła questnode

Właściwości węzła questnode	Wartości	Opis właściwości
target	field	Modele QUEST wymagają jednej zmiennej przewidywanej i co najmniej jednej zmiennej wejściowej. Można też określić zmienną częstości. Więcej informacji można znaleźć w temacie “Wspólne właściwości węzłów modelowania” na stronie 173.
continue_training_existing_model	flag	
objective	Standard Boosting Bagging psm	Wartość psm jest używana w przypadku bardzo dużych zbiorów danych i wymaga połączenia z serwerem.

Tabela 138. Właściwości węzła `questnode` (kontynuacja)

Właściwości węzła <code>questnode</code>	Wartości	Opis właściwości
<code>model_output_type</code>	Single InteractiveBuilder	
<code>use_tree_directives</code>	<i>flag</i>	
<code>tree_directives</code>	<i>string</i>	
<code>use_max_depth</code>	Default Custom	
<code>max_depth</code>	<i>integer</i>	Maksymalna głębokość drzewa, od 0 do 1000. Używana tylko wtedy, gdy <code>use_max_depth = Custom</code> .
<code>prune_tree</code>	<i>flag</i>	Przytnij drzewo, aby uniknąć przeuczenia.
<code>use_std_err</code>	<i>flag</i>	Użyj maksymalnej różnicy w ryzyku (w błędach standardowych).
<code>std_err_multiplier</code>	<i>number</i>	Maksymalna różnica.
<code>max_surrogates</code>	<i>number</i>	Maksymalna liczba substytutów.
<code>use_percentage</code>	<i>flag</i>	
<code>min_parent_records_pc</code>	<i>number</i>	
<code>min_child_records_pc</code>	<i>number</i>	
<code>min_parent_records_abs</code>	<i>number</i>	
<code>min_child_records_abs</code>	<i>number</i>	
<code>use_costs</code>	<i>flag</i>	
<code>costs</code>	<i>structured</i>	Właściwość ustrukturyzowana.
<code>priors</code>	Data Equal Custom	
<code>custom_priors</code>	<i>structured</i>	Właściwość ustrukturyzowana.
<code>adjust_priors</code>	<i>flag</i>	
<code>trails</code>	<i>number</i>	Liczba modeli zespolonych dla boostingu i/lub agregacji bootstrapowej.
<code>set_ensemble_method</code>	Voting HighestProbability HighestMeanProbability	Domyślna reguła zespolenia dla przewidywanych zmiennych jakościowych.
<code>range_ensemble_method</code>	Mean Median	Domyślna reguła zespolenia dla docelowych wartości ilościowych.
<code>large_boost</code>	<i>flag</i>	Zastosuj wzmocnienie do bardzo dużych zbiorów danych.
<code>split_alpha</code>	<i>number</i>	Poziom istotności dla podziałów.
<code>train_pct</code>	<i>number</i>	Zbiór zabezpieczający przed przeuczeniem.
<code>set_random_seed</code>	<i>flag</i>	Opcja replikacji wyników.
<code>seed</code>	<i>number</i>	
<code>calculate_variable_importance</code>	<i>flag</i>	
<code>calculate_raw_propensities</code>	<i>flag</i>	
<code>calculate_adjusted_propensities</code>	<i>flag</i>	
<code>adjusted_propensity_partition</code>	Test Validation	

Właściwości węzła randomtrees



Węzeł Drzewa losowe jest podobny do istniejącego węzła C&RT; jednak węzeł Drzewa losowe jest przeznaczony do przetwarzania dużych zbiorów danych w celu utworzenia pojedynczego drzewa i wyświetla model wynikowy w przeglądarce wyników, która została dodana w wersji 17 programu SPSS Modeler. Węzeł Drzewa losowe generuje drzewo decyzyjne umożliwiające predykcję lub klasyfikację przyszłych obserwacji. W metodzie tej stosowany jest rekursywny podział rekordów na segmenty przez minimalizację zanieczyszczeń w każdym kroku, przy czym węzeł w drzewie jest uważany za *czysty*, jeśli 100% obserwacji w węźle przypada na konkretną kategorię zmiennej przewidywanej. Zmienne przewidywana i wejściowa mogą być zakresami liczbowymi lub jakościowymi (nominalnymi, porządkowymi lub flagami); wszystkie podziały są binarne (tylko dwie podgrupy).

Tabela 139. Właściwości węzła randomtrees

Właściwości węzła randomtrees	Wartości	Opis właściwości
target	<i>field</i>	W węźle Drzewa losowe modele wymagają jednej zmiennej przewidywanej i co najmniej jednej zmiennej wejściowej. Można też określić zmienną częstości. Więcej informacji można znaleźć w temacie “Wspólne właściwości węzłów modelowania” na stronie 173.
number_of_models	<i>integer</i>	Określa liczbę modeli, jakie mają być utworzone w ramach modelowania zespoleń.
use_number_of_predictors	<i>flag</i>	Określa, czy używana jest wartość <i>number_of_predictors</i> .
number_of_predictors	<i>integer</i>	Określa liczbę predyktorów, które mają być używane przy budowaniu modeli rozdzielonych.
use_stop_rule_for_accuracy	<i>flag</i>	Określa, czy budowanie modelu jest przerywane, gdy nie da się już poprawić dokładności.
sample_size	<i>number</i>	Zmniejszenie tej wartości przynosi wzrost wydajności przetwarzania bardzo obszernych zbiorów danych.
handle_imbalanced_data	<i>flag</i>	Jeśli zmienna przewidywana modelu jest wynikiem typu flaga, a stosunek pożądanego wyniku do niepożądanego wyniku jest bardzo mały, to dane są nierównoważone i dobór próby metodą bootstrap wykonywany przez model może niekorzystnie wpłynąć na dokładność modelu. Ta właściwość włącza obsługę danych nierównoważonych, tak by model uwzględniał większy udział wyników poświadczonych i był przez to silniejszy.
use_weighted_sampling	<i>flag</i>	Wartość <i>False</i> powoduje, że zmienne dla każdego węzła są wybierane losowo z tym samym prawdopodobieństwem. Wartość <i>True</i> powoduje ważne i odpowiedni dobór zmiennych.
max_node_number	<i>integer</i>	Maksymalna liczba węzłów dozwolona w jednym drzewie. Jeśli następny podział spowodowałby przekroczenie tej liczby, rozrost drzewa jest zatrzymywany.
max_depth	<i>integer</i>	Maksymalna głębokość drzewa przed zatrzymaniem rozrostu.

Tabela 139. Właściwości węzła *randomtrees* (kontynuacja)

Właściwości węzła <i>randomtrees</i>	Wartości	Opis właściwości
<code>min_child_node_size</code>	<i>integer</i>	Określa minimalną liczbę rekordów dozwoloną w węźle podrzędnym po podziale węzła nadrzędnego. Gdyby węzeł podrzędny miał zawierać mniej rekordów, węzeł nadrzędny nie jest dzielony
<code>use_costs</code>	<i>flag</i>	
<code>costs</code>	<i>structured</i>	Właściwość ustrukturyzowana. Obowiązujący format to lista złożona z 3 wartości: wartości rzeczywistej, wartości przewidywanej i kosztu, gdyby ta predykcja była błędna. Na przykład: <code>tree.setPropertyValue("costs", [{"drugA", "drugB", 3.0}, {"drugX", "drugY", 4.0}])</code>
<code>default_cost_increase</code>	<code>none</code> <code>linear</code> <code>square</code> <code>custom</code>	Uwaga: włączona tylko w przypadku porządkowych zmiennych przewidywanych. Określa wartości domyślne w macierzy kosztów.
<code>max_pct_missing</code>	<i>integer</i>	Jeśli odsetek braków danych w jakiegokolwiek zmiennej wejściowej byłby większy od określonej tutaj wartości, zmienna wejściowa jest wykluczana. Minimum 0, maksimum 100.
<code>exclude_single_cat_pct</code>	<i>integer</i>	Jeśli jedna kategoria reprezentuje większy odsetek rekordów niż określony tutaj, cała zmienna jest wykluczana z budowania modelu. Minimum 1, maksimum 99.
<code>max_category_number</code>	<i>integer</i>	Jeśli liczba kategorii w zmiennej przekracza tę wartość, zmienna jest wykluczana z budowania modelu. Minimum 2.
<code>min_field_variation</code>	<i>number</i>	Jeśli współczynnik zmienności zmiennej ilościowej jest mniejszy od tej wartości, zmienna jest wykluczana z budowania modelu.
<code>num_bins</code>	<i>integer</i>	Używana tylko wtedy, gdy dane wejściowe są złożone ze zmiennych ciągłych. Określa liczbę przedziałów o równej częstości, na którą mają być podzielone dane wejściowe; dostępne opcje: 2, 4, 5, 10, 20, 25, 50 i 100.
<code>topN</code>	<i>integer</i>	Określa liczbę reguł do ujęcia w raporcie. Domyślna wartość wynosi 50, minimalna 1, a maksymalna 1000.

Właściwości węzła *regressionnode*



Regresja liniowa to typowa technika statystyczna umożliwiająca podsumowanie danych i przewidywanie poprzez dopasowanie do linii prostej lub powierzchni, co powoduje zminimalizowane rozbieżności pomiędzy przewidywanymi a rzeczywistymi wartościami zmiennych wyjściowych.

Uwaga: W kolejnych wersjach produktu węzeł Regresja zostanie zastąpiony węzłem Liniowa. Zalecamy, by od tej pory używać modeli Liniowa do realizacji regresji liniowej.

Przykład

```

node = stream.create("regression", "My node")
# Karta "Zmienne"
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")
# Karta "Model"
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)
# Karta "Zaawansowane"
node.setPropertyValue("mode", "Expert")
node.setPropertyValue("complete_records", False)
node.setPropertyValue("tolerance", "1.0E-3")
# Sekcja "Krokowa..." 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)
# Sekcja "Wyniki..." 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)

```

Tabela 140. Właściwości węzła regressionnode

Właściwości węzła regressionnode	Wartości	Opis właściwości
target	field	Modele Regresja wymagają jednej zmiennej przewidywanej i co najmniej jednej zmiennej wejściowej. Można również określić zmienną ważącą. Więcej informacji można znaleźć w temacie "Wspólne właściwości węzłów modelowania" na stronie 173.
method	Enter Stepwise Backwards Forwards	
include_constant	flag	
use_weight	flag	
weight_field	field	
mode	Simple Expert	
complete_records	flag	

Tabela 140. Właściwości węzła *regressionnode* (kontynuacja)

Właściwości węzła <i>regressionnode</i>	Wartości	Opis właściwości
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	Argumenty należy używać w podwójne cudzysłowy.
stepping_method	useP useF	useP: użyj prawdopodobieństwa F useF: użyj wartości F
probability_entry	<i>number</i>	
probability_removal	<i>number</i>	
F_value_entry	<i>number</i>	
F_value_removal	<i>number</i>	
selection_criteria	<i>flag</i>	
confidence_interval	<i>flag</i>	
covariance_matrix	<i>flag</i>	
collinearity_diagnostics	<i>flag</i>	
regression_coefficients	<i>flag</i>	
exclude_fields	<i>flag</i>	
durbin_watson	<i>flag</i>	
model_fit	<i>flag</i>	
r_squared_change	<i>flag</i>	
p_correlations	<i>flag</i>	
descriptives	<i>flag</i>	
calculate_variable_importance	<i>flag</i>	

Właściwości węzła *sequencenode*



Węzeł Sekwencje wykrywa reguły asocjacyjne w danych sekwencyjnych lub zorientowanych czasowo. Sekwencja to lista zbiorów elementów z tendencją do występowania w przewidywalnej kolejności. Na przykład klient dokonujący zakupu brzytwy i balsamu po goleniu przy następnej wizycie w sklepie może dokonać zakupu kremu po goleniu. Węzeł Sekwencje bazuje na algorytmie reguł asocjacyjnych CARMA, który efektywnie znajduje sekwencje w dwóch przejściach przez dane.

Przykład

```
node = stream.create("sequence", "My node")
# Karta "Zmienne"
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")
# Karta "Model"
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)
# Karta "Zaawansowane"
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)

```

Tabela 141. Właściwości węzła sequencenode

Właściwości węzła sequencenode	Wartości	Opis właściwości
id_field	<i>field</i>	W celu utworzenia modelu Numer konieczne jest określenie zmiennej identyfikacyjnej, opcjonalnej zmiennej czasowej i jednej lub więcej zmiennych zawartości. Zmienne wagi i częstości nie są używane. Więcej informacji można znaleźć w temacie “Wspólne właściwości węzłów modelowania” na stronie 173.
time_field	<i>field</i>	
use_time_field	<i>flag</i>	
content_fields	<i>[field1 ... fieldn]</i>	
contiguous	<i>flag</i>	
min_supp	<i>number</i>	
min_conf	<i>number</i>	
max_size	<i>number</i>	
max_predictions	<i>number</i>	
mode	Simple Expert	
use_max_duration	<i>flag</i>	
max_duration	<i>number</i>	
use_gaps	<i>flag</i>	
min_item_gap	<i>number</i>	
max_item_gap	<i>number</i>	
use_pruning	<i>flag</i>	
pruning_value	<i>number</i>	
set_mem_sequences	<i>flag</i>	
mem_sequences	<i>integer</i>	

Właściwości węzła slrmnode



Węzeł Model odpowiedzi samonauczenia (SLRM) umożliwia utworzenie modelu, w którym pojedyncza nowa obserwacja lub niewielka liczba nowych obserwacji może zostać użyta do ponownej oceny modelu bez konieczności ponownego uczenia modelu z wykorzystaniem wszystkich danych.

Przykład

```
node = stream.create("slrm", "My node")
node.setPropertyValue("target", "Offer")
node.setPropertyValue("target_response", "Response")
node.setPropertyValue("inputs", ["Cust_ID", "Age", "Ave_Bal"])
```

Tabela 142. Właściwości węzła slrmnode

Właściwości węzła slrmnode	Wartości	Opis właściwości
target	<i>field</i>	Zmienna docelowa musi być zmienną nominalną lub typu flaga. Można też określić zmienną częstości. Więcej informacji można znaleźć w temacie “Wspólne właściwości węzłów modelowania” na stronie 173.
target_response	<i>field</i>	Musi być typu flaga.
continue_training_existing_model	<i>flag</i>	
target_field_values	<i>flag</i>	Use all: użyj wszystkich wartości ze źródła. Specify: wybierz wymagane wartości.
target_field_values_specify	<i>[field1 ... fieldN]</i>	
include_model_assessment	<i>flag</i>	
model_assessment_random_seed	<i>number</i>	Musi być liczbą rzeczywistą.
model_assessment_sample_size	<i>number</i>	Musi być liczbą rzeczywistą.
model_assessment_iterations	<i>number</i>	Liczba iteracji.
display_model_evaluation	<i>flag</i>	
max_predictions	<i>number</i>	
randomization	<i>number</i>	
scoring_random_seed	<i>number</i>	
sort	Ascending Descending	Określa, czy jako pierwsze będą wyświetlane oferty z najwyższym, czy z najniższym wynikiem.
model_reliability	<i>flag</i>	
calculate_variable_importance	<i>flag</i>	

właściwości węzła statisticsmodelnode



Węzeł Model Statistics umożliwia analizowanie danych i pracę z nimi poprzez uruchomienie procedur IBM SPSS Statistics tworzących PMML. Ten węzeł wymaga licencjonowanej kopii programu IBM SPSS Statistics.

Właściwości tego węzła są opisane w sekcji “właściwości węzła statisticsmodelnode” na stronie 330.

Właściwości węzła stpnode



Węzeł predykcji przestrzenno-czasowej używa danych zawierających informacje o lokalizacji, zmiennych wejściowych predykcji (predyktorów), zmiennej czasu i zmiennej przewidywanej. W danych z każdą lokalizacją powiązany jest szereg wierszy, które odzwierciedlają wartości predyktorów w różnych punktach w czasie. Po przeanalizowaniu danych mogą być one używane do przewidywania wartości w dowolnej lokalizacji w danych kształtu używanych w analizie.

Tabela 143. Właściwości węzła stpnode

Właściwości węzła stpnode	Typ danych	Opis właściwości
Karta Zmienne		
target	<i>field</i>	Jest to zmienna przewidywana.
location	<i>field</i>	Zmienna lokalizacji dla modelu. Dozwolone są tylko zmienne geoprzestrzenne.
location_label	<i>field</i>	Zmienna kategoryczna, która ma być używana w wynikach w charakterze etykiet lokalizacji wybranych we właściwości location
time_field	<i>field</i>	Zmienna czasu dla modelu. Dozwolone są tylko zmienne ciągłe o typie składowania: czas, data, znacznik czasu lub liczba całkowita.
inputs	<i>[field1 ... fieldN]</i>	Lista zmiennych wejściowych.
Karta Przedziały czasowe		
interval_type_timestamp	Years Quarters Months Weeks Days Hours Minutes Seconds	
interval_type_date	Years Quarters Months Weeks Days	
interval_type_time	Hours Minutes Seconds	Ogranicza liczbę dni w tygodniu brana pod uwagę przy tworzeniu indeksu czasu używanego do obliczeń przez algorytm STP.
interval_type_integer	Periods (Tylko dla indeksowych zmiennych czasowych składowanych jako liczba całkowita)	Przedział, w który zostanie przekształcony zbiór danych. Opcje dostępne do wyboru zależą od typu składowania zmiennej wybranej jako <i>time_field</i> dla modelu.
period_start	<i>integer</i>	

Tabela 143. Właściwości węzła *stptime* (kontynuacja)

Właściwości węzła <i>stptime</i>	Typ danych	Opis właściwości
<i>start_month</i>	January February March April May June July August September October November December	Miesiąc, od którego model zacznie indeksowanie (na przykład, jeśli będzie to March, ale pierwszy rekord zbioru danych do January, model pominiemy dwa pierwsze rekordy i rozpocznie od marca).
<i>week_begins_on</i>	Sunday Monday Tuesday Wednesday Thursday Friday Saturday	Punkt początkowy indeksu czasu utworzonego przez algorytm STP z danych.
<i>days_per_week</i>	<i>integer</i>	Minimum 1, maksimum 7, z przyrostem co 1
<i>hours_per_day</i>	<i>integer</i>	Liczba godzin w dniu uwzględniana przez model. Na przykład wartość 10 spowoduje, że model rozpocznie indeksowanie od czasu <i>day_begins_at</i> , będzie kontynuował przez 10 godzin, następnie przeskoczy do następnej wartości <i>day_begins_at</i> itd.
<i>day_begins_at</i>	00:00 01:00 02:00 03:00 ... 23:00	Godzina, o której model rozpoczyna indeksowanie.
<i>interval_increment</i>	1 2 3 4 5 6 10 12 15 20 30	Przyrost jest określony w minutach lub sekundach. Określa, czy model tworzy indeksy z danych. Zatem przy przyroście 30 i przedziale typu <i>seconds</i> model utworzy indeks z danych w odstępach 30-sekundowych.

Tabela 143. Właściwości węzła *stnode* (kontynuacja)

Właściwości węzła <i>stnode</i>	Typ danych	Opis właściwości
<code>data_matches_interval</code>	<i>Boolean</i>	<p>Wartość N powoduje, że przekształcenie danych w zwykły typ <code>interval_type</code> odbywa się przed zbudowaniem modelu.</p> <p>Jeśli dane są już zapisane we właściwym formacie, a <code>interval_type</code> oraz wszelkie powiązane ustawienia są zgodne z charakterystyką danych, należy ustawić tę właściwość na Y, aby dane nie były w żaden sposób przekształcane ani agregowane.</p> <p>Ustawienie Y dezaktywuje wszystkie właściwości agregacji.</p>
<code>agg_range_default</code>	Sum Mean Min Max Median 1stQuartile 3rdQuartile	Określa domyślną metodę agregacji zmiennych ciągłych. Wszelkie zmienne ciągle nieujęte wprost w agregacji niestandardowej będą agregowane określoną tutaj metodą.
<code>custom_agg</code>	<pre>[[field, aggregation method],[..]]</pre> <p>Demo: <pre>[['x5' 'FirstQuartile']['x4' 'Sum']]</pre></p>	<p>Właściwość ustrukturyzowana:</p> <p>Parametr skryptu: <code>custom_agg</code></p> <p>Na przykład: <pre>set :stnode.custom_agg = [[field1 function] [field2 function]]</pre></p> <p>Gdzie <code>function</code> jest funkcją agregacji, która ma by używana z tą zmienną.</p>
Karta Podstawowe		
<code>include_intercept</code>	<i>flag</i>	
<code>max_autoregressive_lag</code>	<i>integer</i>	Minimum 1, maksimum 5, z przyrostem 1. Jest to liczba poprzednich rekordów potrzebna do predykcji. Na przykład wartość 5 spowoduje, że do utworzenia nowej prognozy użytych zostanie 5 poprzednich rekordów. Określona tutaj liczba rekordów z danych budowania zostanie włączona do modelu, a użytkownik nie musi ponownie podawać danych podczas oceny modelu.
<code>estimation_method</code>	Parametric Nonparametric	Metoda modelowania macierzy kowariancji przestrzennej
<code>parametric_model</code>	Gaussian Exponential PoweredExponential	Parametr określający rząd dla modelu kowariancji przestrzennej Parametric
<code>exponential_power</code>	<i>number</i>	Poziom siły dla modelu PoweredExponential . Minimum 1, maksimum 2.

Tabela 143. Właściwości węzła *stpnode* (kontynuacja)

Właściwości węzła <i>stpnode</i>	Typ danych	Opis właściwości
Karta Zaawansowane		
max_missing_values	<i>integer</i>	Maksymalny odsetek rekordów z brakami danych dozwolony w modelu.
significance	<i>number</i>	Poziom istotności dla testowania hipotez w procesie budowania modelu. Określa istotność dla wszystkich testów w estymacji modelu STP, w tym dwóch testów dobroci dopasowania, testów F efektów i testów t współczynnika.
Karta Wynik		
model_specifications	<i>flag</i>	
temporal_summary	<i>flag</i>	
location_summary	<i>flag</i>	Określa, czy w wynikach modelu ma być uwzględniona tabela Podsumowanie lokalizacji.
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>	Wartość progowa, przy której grupy wyników będą uznawane za wystarczająco podobne, aby mogły zostać scalone w jedną grupę.
max_number_clusters	<i>integer</i>	Górny limit liczby skupień, które mogą zostać uwzględnione w wyniku modelu.
Karta Opcje modelu		
use_model_name	<i>flag</i>	
model_name	<i>string</i>	
uncertainty_factor	<i>number</i>	Minimum 0, maksimum 100. Określa przyrost niepewności (błędu) przypisywany do predykcji w przyszłości. Jest to górna i dolna granica predykcji.

Właściwości węzła *svmnode*



Węzeł Algorytm SVM umożliwia szybką klasyfikację danych do jednej lub dwu grup bez przeuczenia. Algorytm SVM działa prawidłowo dla szerokiego zbioru danych, na przykład takiego o bardzo dużej liczbie zmiennych wejściowych.

Przykład

```
node = stream.create("svm", "My node")
# Karta Zaawansowane
node.setPropertyValue("mode", "Expert")
node.setPropertyValue("all_probabilities", True)
node.setPropertyValue("kernel", "Polynomial")
node.setPropertyValue("gamma", 1.5)
```

Tabela 144. Właściwości węzła svmnode.

Właściwości węzła svmnode	Wartości	Opis właściwości
all_probabilities	<i>flag</i>	
stopping_criteria	1.0E-1 1.0E-2 1.0E-3 (domyślnie) 1.0E-4 1.0E-5 1.0E-6	Ta opcja określa moment zatrzymania algorytmu optymalizacji.
regularization	<i>number</i>	Znana także jako parametr C.
precision	<i>number</i>	Używana tylko wtedy, gdy zmienna przewidywana jest ciągła.
kernel	RBF(domyślnie) Polynomial Sigmoid Linear	Typ funkcji algorytmu używanego do przekształceń.
rbf_gamma	<i>number</i>	Używana tylko wtedy, gdy kernel ma wartość RBF.
gamma	<i>number</i>	Używana tylko wtedy, gdy kernel ma wartość Polynomial lub Sigmoid.
bias	<i>number</i>	
degree	<i>number</i>	Używana tylko wtedy, gdy kernel ma wartość Polynomial.
calculate_variable_importance	<i>flag</i>	
calculate_raw_propensities	<i>flag</i>	
calculate_adjusted_propensities	<i>flag</i>	
adjusted_propensity_partition	Test Validation	

Właściwości węzła tcmnode



Modelowanie przyczynowe szeregów czasowych jest próbą wykrycia kluczowych zależności przyczynowych w danych o szeregach czasowych. W procesie takiego modelowania użytkownik określa zbiór szeregów przewidywanych i zbiór potencjalnych zmiennych wejściowych dla tych szeregów przewidywanych. Następnie procedura buduje autoregresyjny model każdego szeregu przewidywanego i uwzględnia tylko te zmienne wejściowe, które z przewidywanym szeregiem łączą najbardziej istotne relacje przyczynowe.

Tabela 145. Właściwości węzła tcmnode

Właściwości węzła tcmnode	Wartości	Opis właściwości
custom_fields	<i>Boolean</i>	

Tabela 145. Właściwości węzła tcmnode (kontynuacja)

Właściwości węzła tcmnode	Wartości	Opis właściwości
dimensionlist	[<i>dimension1</i> ... <i>dimensionN</i>]	
data_struct	Multiple Single	
metric_fields	<i>fields</i>	
both_target_and_input	[<i>f1</i> ... <i>fN</i>]	
targets	[<i>f1</i> ... <i>fN</i>]	
candidate_inputs	[<i>f1</i> ... <i>fN</i>]	
forced_inputs	[<i>f1</i> ... <i>fN</i>]	
use_timestamp	Timestamp Period	
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	

Tabela 145. Właściwości węzła tcmnode (kontynuacja)

Właściwości węzła tcmnode	Wartości	Opis właściwości
cross_hour	<i>Boolean</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	
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	<i>Boolean</i>	
num_targets	<i>integer</i>	
percent_targets	<i>integer</i>	
fields_display	<i>list</i>	
series_display	<i>list</i>	
network_graph_for_target	<i>Boolean</i>	
sign_level_for_target	<i>number</i>	
fit_and_outlier_for_target	<i>Boolean</i>	
sum_and_para_for_target	<i>Boolean</i>	
impact_diag_for_target	<i>Boolean</i>	
impact_diag_type_for_target	Effect Cause Both	
impact_diag_level_for_target	<i>integer</i>	
series_plot_for_target	<i>Boolean</i>	
res_plot_for_target	<i>Boolean</i>	
top_input_for_target	<i>Boolean</i>	

Tabela 145. Właściwości węzła tcmnode (kontynuacja)

Właściwości węzła tcmnode	Wartości	Opis właściwości
forecast_table_for_target	Boolean	
same_as_for_target	Boolean	
network_graph_for_series	Boolean	
sign_level_for_series	number	
fit_and_outlier_for_series	Boolean	
sum_and_para_for_series	Boolean	
impact_diagram_for_series	Boolean	
impact_diagram_type_for_series	Effect Cause Both	
impact_diagram_level_for_series	integer	
series_plot_for_series	Boolean	
residual_plot_for_series	Boolean	
forecast_table_for_series	Boolean	
outlier_root_cause_analysis	Boolean	
causal_levels	integer	
outlier_table	Interactive Pivot Both	
rmsp_error	Boolean	
bic	Boolean	
r_square	Boolean	
outliers_over_time	Boolean	
series_transormation	Boolean	
use_estimation_period	Boolean	
estimation_period	Times Observation	
observations	list	
observations_type	Latest Earliest	
observations_num	integer	
observations_exclude	integer	
extend_records_into_future	Boolean	
forecastperiods	integer	
max_num_distinct_values	integer	
display_targets	FIXEDNUMBER PERCENTAGE	
goodness_fit_measure	ROOTMEAN BIC RSQUARE	
top_input_for_series	Boolean	
aic	Boolean	
rmse	Boolean	

Właściwości węzła ts



Węzeł Szereg czasowy umożliwia estymację modelu wykładniczego, modelu autoregresyjnej zintegrowanej średniej ruchomej (ARIMA) jednej zmiennej oraz modelu ARIMA wielu zmiennych (lub funkcji przenoszenia) dla danych szeregów czasowych i generuje prognozy przyszłej wydajności. Ten węzeł Szeregi czasowe jest podobny do starego węzła Szereg czasowy, który od wersji 18 programu SPSS Modeler jest już nieaktualny. Nowszy węzeł Szeregi czasowe umożliwia wykorzystanie potencjału produktu IBM SPSS Analytic Server przy przetwarzaniu wielkich zbiorów danych i prezentuje wynikowy model w przeglądarce wyników dodanej w wersji 17 programu SPSS Modeler.

Tabela 146. Właściwości węzła ts

Właściwości węzła ts	Wartości	Opis właściwości
targets	<i>field</i>	Węzeł Szereg czasowy przewiduje co najmniej jedną zmienną przewidywaną i opcjonalnie korzysta z jednej lub wielu zmiennych wejściowych jako predyktorów. Zmienne częstości i ważące nie są używane. Więcej informacji można znaleźć w temacie “Wspólne właściwości węzłów modelowania” na stronie 173.
candidate_inputs	[<i>field1 ... fieldN</i>]	Zmienne wejściowe lub predyktory używane przez model.
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	
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	

Tabela 146. Właściwości węzła ts (kontynuacja)

Właściwości węzła ts	Wartości	Opis właściwości
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	
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>	Dostępna tylko gdy wybrano ustawienie date_time_field
period_estimation	<i>list</i>	Dostępna tylko gdy wybrano ustawienie use_period
observations_type	Latest Earliest	
observations_num	<i>integer</i>	
observations_exclude	<i>integer</i>	

Tabela 146. Właściwości węzła ts (kontynuacja)

Właściwości węzła ts	Wartości	Opis właściwości
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>	
exsmooth_model_type	Simple HoltsLinearTrend BrownsLinearTrend DampedTrend SimpleSeasonal WintersAdditive WintersMultiplicative DampedTrendAdditive DampedTrendMultiplicative MultiplicativeTrendAdditive MultiplicativeSeasonal MultiplicativeTrendMultiplicative MultiplicativeTrend	Określa metodę wygładzania wykładniczego. Domyślnie Simple.
futureValue_type_method	Compute specify	
exsmooth_transformation_type	None SquareRoot NaturalLog	
arima.p	<i>integer</i>	
arima.d	<i>integer</i>	
arima.q	<i>integer</i>	
arima.sp	<i>integer</i>	
arima.sd	<i>integer</i>	
arima.sq	<i>integer</i>	
arima_transformation_type	None SquareRoot NaturalLog	
arima_include_constant	<i>flag</i>	
tf_arima.p. <i>fieldname</i>	<i>integer</i>	Dla funkcji przenoszenia.

Tabela 146. Właściwości węzła ts (kontynuacja)

Właściwości węzła ts	Wartości	Opis właściwości
tf_arima.d. <i>fieldname</i>	<i>integer</i>	Dla funkcji przenoszenia.
tf_arima.q. <i>fieldname</i>	<i>integer</i>	Dla funkcji przenoszenia.
tf_arima.sp. <i>fieldname</i>	<i>integer</i>	Dla funkcji przenoszenia.
tf_arima.sd. <i>fieldname</i>	<i>integer</i>	Dla funkcji przenoszenia.
tf_arima.sq. <i>fieldname</i>	<i>integer</i>	Dla funkcji przenoszenia.
tf_arima.delay. <i>fieldname</i>	<i>integer</i>	Dla funkcji przenoszenia.
tf_arima.transformation_type. <i>fieldname</i>	None SquareRoot NaturalLog	Dla funkcji przenoszenia.
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>	
max_lags	<i>integer</i>	
cal_PI	<i>flag</i>	
conf_limit_pct	<i>real</i>	
events	<i>fields</i>	
continue	<i>flag</i>	
scoring_model_only	<i>flag</i>	Używany do modeli z bardzo dużą liczbą (rzędu dziesiątek tysięcy) szeregów czasowych.
forecastperiods	<i>integer</i>	
extend_records_into_future	<i>flag</i>	
extend_metric_values	<i>fields</i>	Umożliwia wprowadzenie przyszłych wartości do predyktorów.
conf_limits	<i>flag</i>	
noise_res	<i>flag</i>	
max_models_output	<i>integer</i>	Kontroluje, ile modeli jest przedstawianych w wynikach. Wartością domyślną jest 10. Modele nie są przedstawiane w wynikach, jeśli łączna liczba modeli przekracza tę wartość. Modele są nadal dostępne do oceny.

Właściwości węzła timeseriesnode (zdeprecjonowanego)



Uwaga: Pierwotny węzeł Szereg czasowy jest nieaktualny w wersji 18 produktu SPSS Modeler i został zastąpiony przez nowy węzeł Szereg czasowy, który zaprojektowano w taki sposób, aby wykorzystywał potencjał serwera IBM SPSS Analytic Server i przetwarzał wielkie zbiory danych (big data). Węzeł Szereg czasowy umożliwia estymację modelu wykładniczego, modelu autoregresyjnej zintegrowanej średniej ruchomej (ARIMA) jednej zmiennej oraz modelu ARIMA wielu zmiennych (lub funkcji przenoszenia) dla danych szeregów czasowych i generuje prognozy przyszłej wydajności. Węzeł Szereg czasowy musi zawsze być poprzedzony węzłem Przedziały czasowe.

Przykład

```
node = stream.create("timeseries", "My node")
node.setPropertyValue("method", "Exsmooth")
node.setPropertyValue("exsmooth_model_type", "HoltsLinearTrend")
node.setPropertyValue("exsmooth_transformation_type", "None")
```

Tabela 147. Właściwości węzła timeseriesnode

Właściwości węzła timeseriesnode	Wartości	Opis właściwości
targets	<i>field</i>	Węzeł Szereg czasowy przewiduje co najmniej jedną zmienną przewidywaną i opcjonalnie korzysta z jednej lub wielu zmiennych wejściowych jako predyktorów. Zmienne częstości i ważące nie są używane. Więcej informacji można znaleźć w temacie “Wspólne właściwości węzłów modelowania” na stronie 173.
continue	<i>flag</i>	
method	ExpertModeler Exsmooth Arima Reuse	
expert_modeler_method	<i>flag</i>	
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>	

Tabela 147. Właściwości węzła *timeseriesnode* (kontynuacja)

Właściwości węzła <i>timeseriesnode</i>	Wartości	Opis właściwości
<i>exsmooth_model_type</i>	Simple HoltsLinearTrend BrownsLinearTrend DampedTrend SimpleSeasonal WintersAdditive WintersMultiplicative	
<i>exsmooth_transformation_type</i>	None SquareRoot NaturalLog	
<i>arima_p</i>	<i>integer</i>	
<i>arima_d</i>	<i>integer</i>	
<i>arima_q</i>	<i>integer</i>	
<i>arima_sp</i>	<i>integer</i>	
<i>arima_sd</i>	<i>integer</i>	
<i>arima_sq</i>	<i>integer</i>	
<i>arima_transformation_type</i>	None SquareRoot NaturalLog	
<i>arima_include_constant</i>	<i>flag</i>	
<i>tf_arima_p.fieldname</i>	<i>integer</i>	Dla funkcji przenoszenia.
<i>tf_arima_d.fieldname</i>	<i>integer</i>	Dla funkcji przenoszenia.
<i>tf_arima_q.fieldname</i>	<i>integer</i>	Dla funkcji przenoszenia.
<i>tf_arima_sp.fieldname</i>	<i>integer</i>	Dla funkcji przenoszenia.
<i>tf_arima_sd.fieldname</i>	<i>integer</i>	Dla funkcji przenoszenia.
<i>tf_arima_sq.fieldname</i>	<i>integer</i>	Dla funkcji przenoszenia.
<i>tf_arima_delay.fieldname</i>	<i>integer</i>	Dla funkcji przenoszenia.
<i>tf_arima_transformation_type.fieldname</i>	None SquareRoot NaturalLog	Dla funkcji przenoszenia.
<i>arima_detect_outlier_mode</i>	None Automatic	
<i>arima_outlier_additive</i>	<i>flag</i>	
<i>arima_outlier_level_shift</i>	<i>flag</i>	
<i>arima_outlier_innovational</i>	<i>flag</i>	
<i>arima_outlier_transient</i>	<i>flag</i>	
<i>arima_outlier_seasonal_additive</i>	<i>flag</i>	
<i>arima_outlier_local_trend</i>	<i>flag</i>	
<i>arima_outlier_additive_patch</i>	<i>flag</i>	
<i>conf_limit_pct</i>	<i>real</i>	
<i>max_lags</i>	<i>integer</i>	
<i>events</i>	<i>fields</i>	

Tabela 147. Właściwości węzła *timeseriesnode* (kontynuacja)

Właściwości węzła <i>timeseriesnode</i>	Wartości	Opis właściwości
<code>scoring_model_only</code>	<i>flag</i>	Używany do modeli z bardzo dużą liczbą (rzędu dziesiątek tysięcy) szeregów czasowych.

właściwości węzła *treeas*



Węzeł Drzewo-AS jest podobny do istniejącego węzła CHAID; jednak węzeł Drzewo-AS jest przeznaczony do przetwarzania dużych zbiorów w celu utworzenia pojedynczego drzewa i wyświetlenia modelu wyników w przeglądarce wyników, która została dodana w programie SPSS Modeler, wersja 17. Węzeł generuje drzewo decyzyjne używając statystyki chi-kwadrat (CHAID), aby określić optymalne podziały. CHAID może generować drzewa niebinarne, co oznacza, że niektóre podziały mają więcej niż dwie gałęzie. Zmienne przewidywana i wejściowa mogą być zakresami liczbowymi (ilościowymi) lub jakościowymi. Wyczerpujący CHAID stanowi modyfikację CHAID umożliwiającą dokładniejsze badanie wszystkich możliwych podziałów, lecz obliczenia w jego przypadku zajmują więcej czasu.

Tabela 148. Właściwości węzła *treeas*

Właściwości węzła <i>treeas</i>	Wartości	Opis właściwości
<code>target</code>	<i>field</i>	W węzle Drzewo-AS modele CHAID wymagają jednej zmiennej przewidywanej i co najmniej jednej zmiennej wejściowej. Można też określić zmienną częstości. Więcej informacji można znaleźć w temacie "Wspólne właściwości węzłów modelowania" na stronie 173.
<code>method</code>	<code>chaid</code> <code>exhaustive_chaid</code>	
<code>max_depth</code>	<i>integer</i>	Maksymalna głębokość drzewa, od 0 do 20. Wartość domyślna to 5.
<code>num_bins</code>	<i>integer</i>	Używana tylko wtedy, gdy dane wejściowe są złożone ze zmiennych ciągłych. Określa liczbę przedziałów o równej częstości, na którą mają być podzielone dane wejściowe; dostępne opcje: 2, 4, 5, 10, 20, 25, 50 i 100.
<code>record_threshold</code>	<i>integer</i>	Liczba rekordów, przy której podczas budowania drzewa model przełączy się ze stosowania wartości <i>p</i> na wielkości efektu. Wartość domyślna to 1 000 000; można ją zwiększać lub zmniejszać z krokiem 10 000.
<code>split_alpha</code>	<i>number</i>	Poziom istotności dla podziałów. Wartość musi mieścić się w przedziale od 0,01 do 0,99.
<code>merge_alpha</code>	<i>number</i>	Poziom istotności dla łączenia. Wartość musi mieścić się w przedziale od 0,01 do 0,99.
<code>bonferroni_adjustment</code>	<i>flag</i>	Koryguj wartości istotności metodą Bonferroni.
<code>effect_size_threshold_cont</code>	<i>number</i>	Określa próg wielkości efektów obowiązujący podczas podziału węzłów i łączenia kategorii w przypadku używania ilościowej zmiennej przewidywanej. Wartość musi mieścić się w przedziale od 0,01 do 0,99.

Tabela 148. Właściwości węzła *treeas* (kontynuacja)

Właściwości węzła <i>treeas</i>	Wartości	Opis właściwości
effect_size_threshold_cat	<i>number</i>	Określa próg wielkości efektów obowiązujący podczas podziału węzłów i łączenia kategorii w przypadku używania jakościowej zmiennej przewidywanej. Wartość musi mieścić się w przedziale od 0,01 do 0,99.
split_merged_categories	<i>flag</i>	Zezwalaj na ponowny podział połączonych kategorii.
grouping_sig_level	<i>number</i>	Służy do określania sposobu tworzenia grup węzłów lub rozpoznawania nietypowych węzłów.
chi_square	pearson likelihood_ratio	Metoda obliczania statystyki chi-kwadrat: Pearsona albo iloraz wiarygodności
minimum_record_use	use_percentage use_absolute	
min_parent_records_pc	<i>number</i>	Wartość domyślna to 2. Minimum to 1, maksimum 100, przyrost co 1. Wartość gałęzi nadrzędnej musi być wyższa niż gałęzi podrzędnej.
min_child_records_pc	<i>number</i>	Wartość domyślna to 1. Minimum to 1, maksimum to 100, przyrost co 1.
min_parent_records_abs	<i>number</i>	Wartość domyślna to 100. Minimum to 1, maksimum 100, przyrost co 1. Wartość gałęzi nadrzędnej musi być wyższa niż gałęzi podrzędnej.
min_child_records_abs	<i>number</i>	Wartość domyślna to 50. Minimum 1, maksimum 100, z przyrostem co 1.
epsilon	<i>number</i>	Minimalna zmiana oczekiwanych częstości komórek.
max_iterations	<i>number</i>	Maksimum iteracji dla uzyskania zbieżności.
use_costs	<i>flag</i>	
costs	<i>structured</i>	Właściwość ustrukturyzowana. Obowiązujący format to lista złożona z 3 wartości: wartości rzeczywistej, wartości przewidywanej i kosztu, gdyby ta predykcja była błędna. Na przykład: <code>tree.setPropertyValue("costs", [{"drugA", "drugB", 3.0}, {"drugX", "drugY", 4.0}])</code>
default_cost_increase	none linear square custom	Uwaga: włączona tylko w przypadku porządkowych zmiennych przewidywanych. Określa wartości domyślne w macierzy kosztów.
calculate_conf	<i>flag</i>	
display_rule_id	<i>flag</i>	Dodaje do wyników oceniania zmienną określającą identyfikator węzła końcowego, do którego przypisany jest każdy rekord.

Właściwości węzła twostepnode



Węzeł Dwustopniowa korzysta z dwustopniowej metody grupowania. Pierwszy krok stanowi pojedynczy przebieg danych z myślą o kompresji surowych danych wejściowych w łatwy w zarządzaniu zestaw podgrup. Drugi krok korzysta z hierarchicznej metody grupowania w celu progresywnego scalania podgrup w coraz większe grupy. Metoda Dwustopniowa oferuje korzyści wynikające z automatycznego szacowania optymalnej liczby grup na potrzeby danych szkoleniowych. Pozwala ona skutecznie obsługiwać mieszane typy zmiennych i duże zbiory danych.

Przykład

```
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)
```

Tabela 149. właściwości węzła twostepnode

Właściwości węzła twostepnode	Wartości	Opis właściwości
inputs	[<i>field1</i> ... <i>fieldN</i>]	Modele Dwustopniowa korzystają z listy zmiennych wejściowych, ale nie mają zmiennych przewidywanych. Zmienne wagi i częstości nie są rozpoznawane. Więcej informacji można znaleźć w temacie “Wspólne właściwości węzłów modelowania” na stronie 173.
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	

Właściwości węzła twostepAS



Procedura Dwustopniowa analiza skupień jest narzędziem eksploracyjnym służącym do ujawniania występowania w zbiorze danych naturalnych zgrupowań (lub skupień), które nie są widoczne w inny sposób. Algorytm stosowany przez tę procedurę ma kilka cech, które odróżniają go od tradycyjnych metod grupowania. Do tych cech należy obsługa zmiennych jakościowych i ciągłych, automatyczny wybór liczby skupień oraz skalowalność.

Tabela 150. Właściwości węzła twostepAS

Właściwości węzła twostepAS	Wartości	Opis właściwości
inputs	[f1 ... fN]	Modele Dwustopniowa analiza skupień korzystają z listy zmiennych wejściowych, ale nie mają zmiennych przewidywanych. Zmienne wagi i częstości nie są rozpoznawane.
use_predefined_roles	Boolean	Domyślnie=True
use_custom_field_assignments	Boolean	Domyślnie=False
cluster_num_auto	Boolean	Domyślnie=True
min_num_clusters	integer	Domyślnie=2
max_num_clusters	integer	Domyślnie=15
num_clusters	integer	Domyślnie=5
clustering_criterion	AIC BIC	
automatic_clustering_method	use_clustering_criterion_setting Distance_jump Minimum Maksimum	
feature_importance_method	use_clustering_criterion_setting effect_size	
use_random_seed	Boolean	
random_seed	integer	
distance_measure	Euclidean Loglikelihood	
include_outlier_clusters	Boolean	Domyślnie=True
num_cases_in_feature_tree_leaf_is_less_than	integer	Domyślnie=10
top_perc_outliers	integer	Domyślnie=5
initial_dist_change_threshold	integer	Domyślnie=0
leaf_node_maximum_branches	integer	Domyślnie=8
non_leaf_node_maximum_branches	integer	Domyślnie=8
max_tree_depth	integer	Domyślnie=3
adjustment_weight_on_measurement_level	integer	Domyślnie=6
memory_allocation_mb	number	Domyślnie=512
delayed_split	Boolean	Domyślnie=True
fields_to_standardize	[f1 ... fN]	

Tabela 150. Właściwości węzła twostepAS (kontynuacja)

Właściwości węzła twostepAS	Wartości	Opis właściwości
adaptive_feature_selection	Boolean	Domyślnie=True
featureMisPercent	integer	Domyślnie=70
coefRange	number	Domyślnie=0,05
percCasesSingleCategory	integer	Domyślnie=95
numCases	integer	Domyślnie=24
include_model_specifications	Boolean	Domyślnie=True
include_record_summary	Boolean	Domyślnie=True
include_field_transformations	Boolean	Domyślnie=True
excluded_inputs	Boolean	Domyślnie=True
evaluate_model_quality	Boolean	Domyślnie=True
show_feature_importance_bar_chart	Boolean	Domyślnie=True
show_feature_importance_word_cloud	Boolean	Domyślnie=True
show_outlier_clusters_interactive_table_and_chart	Boolean	Domyślnie=True
show_outlier_clusters_pivot_table	Boolean	Domyślnie=True
across_cluster_feature_importance	Boolean	Domyślnie=True
across_cluster_profiles_pivot_table	Boolean	Domyślnie=True
withinprofiles	Boolean	Domyślnie=True
cluster_distances	Boolean	Domyślnie=True
cluster_label	String Number	
label_prefix	String	

Rozdział 14. Właściwości węzła modelu użytkowego

Węzły modeli użytkowych mają te same właściwości wspólne, co pozostałe węzły. Więcej informacji można znaleźć w temacie “Wspólne właściwości węzłów” na stronie 69.

Właściwości węzła `applyanomalydetectionnode`

Węzły modelowania Wykrywanie anomalii mogą być używane do generowania modeli użytkowych Wykrywanie anomalii. Ten model użytkowy w skryptach nosi nazwę `applyanomalydetectionnode`. Aby uzyskać więcej informacji o skryptowej obsłudze samego węzła modelowania, patrz “właściwości `anomalydetectionnode` węzła” na stronie 173

Tabela 151. Właściwości węzła `applyanomalydetectionnode`.

Właściwości węzła <code>applyanomalydetectionnode</code>	Wartości	Opis właściwości
<code>anomaly_score_method</code>	FlagAndScore FlagOnly ScoreOnly	Określa, jakie wyniki będą generowane przy ocenianiu.
<code>num_fields</code>	<i>integer</i>	Zmienne do ujęcia w raporcie.
<code>discard_records</code>	<i>flag</i>	Określa, czy rekordy mają być usuwane z danych wejściowych, czy nie.
<code>discard_anomalous_records</code>	<i>flag</i>	Wskazuje, czy odrzucać rekordy będące anomaliami, czy <i>niebędące</i> anomaliami. Wartość domyślna <code>off</code> oznacza, że odrzucane są rekordy <i>niebędące</i> anomaliami. Wartość <code>on</code> powoduje, że odrzucane będą rekordy będące anomaliami. Ta właściwość jest włączona tylko wtedy, gdy włączona jest właściwość <code>discard_records</code> .

Właściwości węzła `applyapriorinode`

Węzły modelowania Apriori mogą być używane do generowania modeli użytkowych Apriori. Ten model użytkowy w skryptach nosi nazwę `applyapriorinode`. Aby uzyskać więcej informacji o skryptowej obsłudze samego węzła modelowania, patrz “Właściwości węzła `apriorinode`” na stronie 175

Tabela 152. Właściwości węzła `applyapriorinode`.

Właściwości węzła <code>applyapriorinode</code>	Wartości	Opis właściwości
<code>max_predictions</code>	<i>number (integer)</i>	
<code>ignore_unmatched</code>	<i>flag</i>	
<code>allow_repeats</code>	<i>flag</i>	
<code>check_basket</code>	NoPredictions Predictions NoCheck	
<code>criterion</code>	Confidence Support RuleSupport Lift Deployability	

Właściwości węzła applyassociationrulesnode

Węzły modelowania Reguły asocjacyjne mogą być używane do generowania modeli użytkowych reguł asocjacyjnych. Ten model użytkowy w skryptach nosi nazwę *applyassociationrulesnode*. Aby uzyskać więcej informacji o skryptowej obsłudze samego węzła modelowania, patrz “Właściwości węzła associationrulesnode” na stronie 176.

Tabela 153. Właściwości węzła applyassociationrulesnode

Właściwości węzła applyassociationrulesnode	Typ danych	Opis właściwości
max_predictions	integer	Maksymalna liczba reguł, która może być stosowana do każdej zmiennej wejściowej przy ocenie.
criterion	Confidence Rulesupport Lift Conditionsupport Deployability	Wybiera miarę, która będzie używana w celu określenia siły reguł.
allow_repeats	Boolean	Określa, czy reguły o tej samej predykcji będą uwzględniane w ocenie.
check_input	NoPredictions Predictions NoCheck	

Właściwości węzła applyautoclassifiernode

Węzły modelowania Auto Klasyfikacja mogą być używane do generowania modeli użytkowych Auto Klasyfikacja. Ten model użytkowy w skryptach nosi nazwę *applyautoclassifiernode*. Aby uzyskać więcej informacji o skryptowej obsłudze samego węzła modelowania, patrz “Właściwości węzła autoclassifiernode” na stronie 178

Tabela 154. Właściwości węzła applyautoclassifiernode.

Właściwości węzła applyautoclassifiernode	Wartości	Opis właściwości
flag_ensemble_method	Voting ConfidenceWeightedVoting RawPropensityWeightedVoting HighestConfidence AverageRawPropensity	Określa metodę wyznaczania oceny zespołenia. To ustawienie obowiązuje tylko wtedy, gdy wybrana zmienna docelowa jest zmienną typu flaga.
flag_voting_tie_selection	Random HighestConfidence RawPropensity	Jeśli wybrana jest metoda głosowania, określa sposób rozstrzygania powiązań. To ustawienie obowiązuje tylko wtedy, gdy wybrana zmienna docelowa jest zmienną typu flaga.
set_ensemble_method	Voting ConfidenceWeightedVoting HighestConfidence	Określa metodę wyznaczania oceny zespołenia. To ustawienie obowiązuje tylko wtedy, gdy wybrana zmienna docelowa jest zmienną typu zbiór.
set_voting_tie_selection	Random HighestConfidence	Jeśli wybrana jest metoda głosowania, określa sposób rozstrzygania powiązań. To ustawienie obowiązuje tylko wtedy, gdy wybrana zmienna docelowa jest zmienną nominalną.

Właściwości węzła applyautoclusternode

Węzły modelowania Auto Klasyfikacja mogą być używane do generowania modeli użytkowych Auto Klasyfikacja. Ten model użytkowy w skryptach nosi nazwę *applyautoclusternode*. Ten model użytkowy nie ma innych właściwości. Aby uzyskać więcej informacji o skryptowej obsłudze samego węzła modelowania, patrz “Właściwości węzła autoclusternode” na stronie 180

Właściwości węzła applyautonumericnode

Węzły modelowania Auto Predykcja mogą być używane do generowania modeli użytkowych Auto Predykcja. Ten model użytkowy w skryptach nosi nazwę *applyautonumericnode*. Aby uzyskać więcej informacji o skryptowej obsłudze samego węzła modelowania, patrz “Właściwości węzła autonumericnode” na stronie 182

Tabela 155. Właściwości węzła applyautonumericnode.

Właściwości węzła applyautonumericnode	Wartości	Opis właściwości
calculate_standard_error	<i>flag</i>	

Właściwości węzła applybayesnetnode

Węzły modelowania sieci bayesowskiej mogą być używane do generowania modeli użytkowych Sieć bayesowska. Ten model użytkowy w skryptach nosi nazwę *applybayesnetnode*. Aby uzyskać więcej informacji o skryptowej obsłudze samego węzła modelowania, patrz “Właściwości węzła bayesnetnode” na stronie 183.

Tabela 156. Właściwości węzła applybayesnetnode.

Właściwości węzła applybayesnetnode	Wartości	Opis właściwości
all_probabilities	<i>flag</i>	
raw_propensity	<i>flag</i>	
adjusted_propensity	<i>flag</i>	
calculate_raw_propensities	<i>flag</i>	
calculate_adjusted_propensities	<i>flag</i>	

Właściwości węzła applyc50node

Węzły modelowania C5.0 mogą być używane do generowania modeli użytkowych C5.0. Ten model użytkowy w skryptach nosi nazwę *applyc50node*. Aby uzyskać więcej informacji o skryptowej obsłudze samego węzła modelowania, patrz “Właściwości węzła c50node” na stronie 185.

Tabela 157. Właściwości węzła applyc50node.

Właściwości węzła applyc50node	Wartości	Opis właściwości
sql_generate	Never NoMissingValues	Służy do ustawiania opcji generowania kodu SQL podczas wykonywania zestawu reguł.
calculate_conf	<i>flag</i>	Dostępna, gdy włączone jest generowanie kodu SQL; ta właściwość dotyczy obliczeń ufności w wygenerowanym drzewie.
calculate_raw_propensities	<i>flag</i>	
calculate_adjusted_propensities	<i>flag</i>	

Właściwości węzła applycarmanode

Węzły modelowania CARMA mogą być używane do generowania modeli użytkowych CARMA. Ten model użytkowy w skryptach nosi nazwę *applycarmanode*. Ten model użytkowy nie ma innych właściwości. Aby uzyskać więcej informacji o skryptowej obsłudze samego węzła modelowania, patrz “Właściwości węzła carmanode” na stronie 186.

Właściwości węzła applycartnode

Węzły modelowania C&R Tree mogą być używane do generowania modeli użytkowych C&R Tree. Ten model użytkowy w skryptach nosi nazwę *applycartnode*. Aby uzyskać więcej informacji o skryptowej obsłudze samego węzła modelowania, patrz “Właściwości węzła cartnode” na stronie 187.

Tabela 158. Właściwości węzła applycartnode.

Właściwości węzła applycartnode	Wartości	Opis właściwości
enable_sql_generation	Never MissingValues NoMissingValues	Służy do ustawiania opcji generowania kodu SQL podczas wykonywania zestawu reguł.
calculate_conf	<i>flag</i>	Dostępna, gdy włączone jest generowanie kodu SQL; ta właściwość dotyczy obliczeń ufności w wygenerowanym drzewie.
display_rule_id	<i>flag</i>	Dodaje do wyników oceniania zmienną określającą identyfikator węzła końcowego, do którego przypisany jest każdy rekord.
calculate_raw_propensities	<i>flag</i>	
calculate_adjusted_propensities	<i>flag</i>	

Właściwości węzła applychaidnode

Węzły modelowania CHAID mogą być używane do generowania modeli użytkowych CHAID. Ten model użytkowy w skryptach nosi nazwę *applychaidnode*. Aby uzyskać więcej informacji o skryptowej obsłudze samego węzła modelowania, patrz “Właściwości węzła chaidnode” na stronie 190.

Tabela 159. Właściwości węzła applychaidnode.

Właściwości węzła applychaidnode	Wartości	Opis właściwości
enable_sql_generation	Never MissingValues	Służy do ustawiania opcji generowania kodu SQL podczas wykonywania zestawu reguł.
calculate_conf	<i>flag</i>	
display_rule_id	<i>flag</i>	Dodaje do wyników oceniania zmienną określającą identyfikator węzła końcowego, do którego przypisany jest każdy rekord.
calculate_raw_propensities	<i>flag</i>	
calculate_adjusted_propensities	<i>flag</i>	

Właściwości węzła applycoxregnode

Węzły modelowania Coxa mogą być używane do generowania modeli użytkowych Coxa. Ten model użytkowy w skryptach nosi nazwę *applycoxregnode*. Aby uzyskać więcej informacji o skryptowej obsłudze samego węzła modelowania, patrz “Właściwości węzła coxregnode” na stronie 192.

Tabela 160. Właściwości węzła applycoxregnode.

Właściwości węzła applycoxregnode	Wartości	Opis właściwości
future_time_as	Intervals Fields	
time_interval	<i>number</i>	
num_future_times	<i>integer</i>	
time_field	<i>field</i>	
past_survival_time	<i>field</i>	
all_probabilities	<i>flag</i>	
cumulative_hazard	<i>flag</i>	

Właściwości węzła applydecisionlistnode

Węzły modelowania Lista decyzyjna mogą być używane do generowania modeli użytkowych Lista decyzyjna. Ten model użytkowy w skryptach nosi nazwę *applydecisionlistnode*. Aby uzyskać więcej informacji o skryptowej obsłudze samego węzła modelowania, patrz “Właściwości węzła decisionlistnode” na stronie 193.

Tabela 161. Właściwości węzła applydecisionlistnode.

Właściwości węzła applydecisionlistnode	Wartości	Opis właściwości
enable_sql_generation	<i>flag</i>	Wartość true powoduje, że IBM SPSS Modeler będzie zlecał przetwarzanie modelu Lista decyzyjna serwerowi SQL.
calculate_raw_propensities	<i>flag</i>	
calculate_adjusted_propensities	<i>flag</i>	

Właściwości węzła applydiscriminantnode

Węzły modelowania Analiza dyskryminacyjna mogą być używane do generowania modeli użytkowych Analiza dyskryminacyjna. Ten model użytkowy w skryptach nosi nazwę *applydiscriminantnode*. Aby uzyskać więcej informacji o skryptowej obsłudze samego węzła modelowania, patrz “Właściwości węzła discriminantnode” na stronie 195.

Tabela 162. Właściwości węzła applydiscriminantnode.

Właściwości węzła applydiscriminantnode	Wartości	Opis właściwości
calculate_raw_propensities	<i>flag</i>	
calculate_adjusted_propensities	<i>flag</i>	

Właściwości węzła applyextension



Węzły budowania Extension mogą być używane do generowania modeli użytkowych Extension. Ten model użytkowy w skryptach nosi nazwę *applyextension*. Aby uzyskać więcej informacji o skryptowej obsłudze samego węzła modelowania, patrz “Właściwości węzła extensionmodelnode” na stronie 196.

Python for Spark — przykład

```
#### przykład skryptu w języku 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] and
dm[predictor_index].index(row[predictor_index]) or -1
            else:
                pval = row[predictor_index]
            pvals.append(pval)
        return LabeledPoint(tval, DenseVector(pvals))

    # konwersja Dataframe do RDD z LabeledPoint
    lps = df.rdd.map(lambda row: row2LabeledPoint(metadata, lookup, target, predictors, row))
    treeModel = DecisionTreeModel.load(cxt.getSparkContext(), _modelPath);
    # ocena modelu, utworzenie RDD z wyłącznie wartościami typu double
    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)

```

Przykład w języku R

```

#### przykład skryptu w języku 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)""")

```

Tabela 163. Właściwości węzła *applyextension*

Właściwości węzła <i>applyextension</i>	Wartości	Opis właściwości
<i>r_syntax</i>	<i>string</i>	Komendy skryptu R realizujące ocenę modelu.
<i>python_syntax</i>	<i>string</i>	Komendy skryptu Python realizujące ocenę modelu.
<i>use_batch_size</i>	<i>flag</i>	Włącza przetwarzanie wsadowe.
<i>batch_size</i>	<i>integer</i>	Określ liczbę rekordów danych, które mają być uwzględnione w każdej partii.
<i>convert_flags</i>	StringsAndDoubles LogicalValues	Opcja przekształcania zmiennych typu flaga.
<i>convert_missing</i>	<i>flag</i>	Opcja przekształcania braków danych na wartości R NA.
<i>convert_datetime</i>	<i>flag</i>	Opcja przekształcania zmiennych o formacie daty lub daty/czasu na formaty daty/czasu języka R.
<i>convert_datetime_class</i>	POSIXct POSIXlt	Opcje określające format docelowy, na który przekształcane są zmienne w formacie daty i daty/czasu.

Właściwości węzła *applyfactornode*

Węzły modelowania Redukcja wymiarów mogą być używane do generowania modeli użytkowych Redukcja wymiarów. Ten model użytkowy w skryptach nosi nazwę *applyfactornode*. Ten model użytkowy nie ma innych właściwości. Aby uzyskać więcej informacji o skryptowej obsłudze samego węzła modelowania, patrz “Właściwości węzła *factornode*” na stronie 199.

Właściwości węzła applyfeatureselectionnode

Węzły modelowania Dobór predyktorów mogą być używane do generowania modeli użytkowych Dobór predyktorów. Ten model użytkowy w skryptach nosi nazwę *applyfeatureselectionnode*. Aby uzyskać więcej informacji o skryptowej obsłudze samego węzła modelowania, patrz “Właściwości węzła featureselectionnode” na stronie 200.

Tabela 164. Właściwości węzła *applyfeatureselectionnode*.

Właściwości węzła applyfeatureselectionnode	Wartości	Opis właściwości
selected_ranked_fields		Określa, które pola rangowane są zaznaczone w przeglądarce modelu.
selected_screened_fields		Określa, które pola monitorowane są zaznaczone w przeglądarce modelu.

Właściwości węzła applygeneralizedlinearnode

Węzły modelowania Uogólnione modele liniowe (GENLIN) mogą być używane do generowania modeli użytkowych Uogólnione modele liniowe (GENLIN). Ten model użytkowy w skryptach nosi nazwę *applygeneralizedlinearnode*. Aby uzyskać więcej informacji o skryptowej obsłudze samego węzła modelowania, patrz “Właściwości węzła genlinnode” na stronie 202.

Tabela 165. Właściwości węzła *applygeneralizedlinearnode*.

Właściwości węzła applygeneralizedlinearnode	Wartości	Opis właściwości
calculate_raw_propensities	<i>flag</i>	
calculate_adjusted_propensities	<i>flag</i>	

Właściwości węzła applyglmnode

Węzły modelowania GLMM mogą być używane do generowania modeli użytkowych GLMM. Ten model użytkowy w skryptach nosi nazwę *applyglmnode*. Aby uzyskać więcej informacji o skryptowej obsłudze samego węzła modelowania, patrz “Właściwości węzła glmnode” na stronie 205.

Tabela 166. Właściwości węzła *applyglmnode*.

Właściwości węzła applyglmnode	Wartości	Opis właściwości
confidence	onProbability onIncrease	Podstawa obliczania współczynnika ufności oceny: najwyższe przewidywane prawdopodobieństwo albo różnica między najwyższym a drugim co do wielkości prawdopodobieństwem.
score_category_probabilities	<i>flag</i>	Wartość True powoduje generowanie przewidywanych prawdopodobieństw dla jakościowych zmiennych przewidywanych. Dla każdej kategorii tworzona jest jedna zmienna. Domyślną wartością jest False.
max_categories	<i>integer</i>	Maksymalna liczba kategorii, dla których mają być przewidywane prawdopodobieństwa. Używana tylko wtedy, gdy <i>score_category_probabilities</i> ma wartość True.

Tabela 166. Właściwości węzła *applyglmnode* (kontynuacja).

Właściwości węzła <i>applyglmnode</i>	Wartości	Opis właściwości
score_propensity	<i>flag</i>	Wartość True powoduje generowanie surowych ocen skłonności (prawdopodobieństwo wyniku „prawdziwego”) dla modeli ze zmiennymi przewidywanymi typu flaga. Jeśli obowiązują podzbiory, generowane są także skorygowane oceny skłonności na podstawie podzbioru testowego. Domyślną wartością jest False.
enable_sql_generation	udf native	Służy do ustawiania opcji generowania kodu SQL podczas wykonywania strumienia. Można zlecić ocenę bazy danych i przeprowadzić ją za pomocą adaptera SPSS® Modeler Server scoring adapter (jeśli dostępne jest połączenie z bazą danych mającą zainstalowany adapter) lub przeprowadzić ocenę w programie SPSS Modeler. Wartość domyślna to udf.

Właściwości węzła *applygle*

Węzły modelowania GLE mogą być używane do generowania modeli użytkowych GLE. Ten model użytkowy w skryptach nosi nazwę *applygle*. Aby uzyskać więcej informacji o skryptowej obsłudze samego węzła modelowania, patrz “Właściwości węzła *gle*” na stronie 209.

Tabela 167. Właściwości węzła *applygle*

Właściwości węzła <i>applygle</i>	Wartości	Opis właściwości
enable_sql_generation	udf native	Służy do ustawiania opcji generowania kodu SQL podczas wykonywania strumienia. Można zlecić ocenę bazy danych i przeprowadzić ją za pomocą adaptera SPSS Modeler Server (jeśli dostępne jest połączenie z bazą danych mającą zainstalowany adapter) albo przeprowadzić ocenę w programie SPSS Modeler.

Właściwości węzła *applykmeansnode*

Węzły modelowania K-średnie mogą być używane do generowania modeli użytkowych K-średnie. Ten model użytkowy w skryptach nosi nazwę *applykmeansnode*. Ten model użytkowy nie ma innych właściwości. Aby uzyskać więcej informacji o skryptowej obsłudze samego węzła modelowania, patrz “Właściwości węzła *kmeansnode*” na stronie 213.

Właściwości węzła *applyknnnode*

Węzły modelowania KNN mogą być używane do generowania modeli użytkowych KNN. Ten model użytkowy w skryptach nosi nazwę *applyknnnode*. Aby uzyskać więcej informacji o skryptowej obsłudze samego węzła modelowania, patrz “Właściwości węzła *knnnode*” na stronie 214.

Tabela 168. Właściwości węzła *applyknnnode*.

Właściwości węzła <i>applyknnnode</i>	Wartości	Opis właściwości
all_probabilities	<i>flag</i>	
save_distances	<i>flag</i>	

Właściwości węzła applykohonennode

Węzły modelowania Sieć Kohonena mogą być używane do generowania modeli użytkowych Sieć Kohonena. Ten model użytkowy w skryptach nosi nazwę *applykohonennode*. Ten model użytkowy nie ma innych właściwości. Aby uzyskać więcej informacji o skryptowej obsłudze samego węzła modelowania, patrz “Właściwości węzła c50node” na stronie 185.

Właściwości węzła applylinearnode

Węzły modelowania Liniowy mogą być używane do generowania modeli użytkowych Liniowy. Ten model użytkowy w skryptach nosi nazwę *applylinearnode*. Aby uzyskać więcej informacji o skryptowej obsłudze samego węzła modelowania, patrz “Właściwości węzła linearnode” na stronie 217.

Tabela 169. Właściwości węzła applylinearnode.

Właściwości węzła linear	Wartości	Opis właściwości
use_custom_name	<i>flag</i>	
custom_name	<i>string</i>	
enable_sql_generation	udf native puresql	Służy do ustawiania opcji generowania kodu SQL podczas wykonywania strumienia. Można zlecić ocenę bazy danych i przeprowadzić ją za pomocą adaptera SPSS® Modeler Server scoring adapter (jeśli dostępne jest połączenie z bazą danych mającą zainstalowany adapter), przeprowadzić ocenę w programie SPSS Modeler lub zlecić ocenę bazy danych i przeprowadzić ją za pomocą SQL. Wartość domyślna to udf.

Właściwości węzła applylinearasnode

Węzły modelowania Liniowy-AS mogą być używane do generowania modeli użytkowych Liniowy-AS. Ten model użytkowy w skryptach nosi nazwę *applylinearasnode*. Aby uzyskać więcej informacji o skryptowej obsłudze samego węzła modelowania, patrz “Właściwości węzła linearasnode” na stronie 218.

Tabela 170. Właściwości węzła applylinearasnode

Właściwości węzła applylinearasnode	Wartości	Opis właściwości
enable_sql_generation	udf native	Wartość domyślna to udf.

Właściwości węzła applylogregnode

Węzły modelowania Regresja logistyczna mogą być używane do generowania modeli użytkowych Regresja logistyczna. Ten model użytkowy w skryptach nosi nazwę *applylogregnode*. Aby uzyskać więcej informacji o skryptowej obsłudze samego węzła modelowania, patrz “Właściwości węzła logregnode” na stronie 219.

Tabela 171. Właściwości węzła applylogregnode.

Właściwości węzła applylogregnode	Wartości	Opis właściwości
calculate_raw_propensities	<i>flag</i>	
calculate_conf	<i>flag</i>	
enable_sql_generation	<i>flag</i>	

Właściwości węzła *applysvmnode*

Węzły modelowania LSVM mogą być używane do generowania modeli użytkowych LSVM. Ten model użytkowy w skryptach nosi nazwę *applysvmnode*. Aby uzyskać więcej informacji o skryptowej obsłudze samego węzła modelowania, patrz “Właściwości węzła *lsvmnode*” na stronie 223.

Tabela 172. Właściwości węzła *applysvmnode*

Właściwości węzła <i>applysvmnode</i>	Wartości	Opis właściwości
<code>calculate_raw_propensities</code>	<i>flag</i>	Określa, czy mają być obliczane surowe oceny skłonności.
<code>enable_sql_generation</code>	<code>udf</code> <code>native</code>	Określa, czy ocenianie ma być wykonywane przy użyciu komponentu Scoring Adapter (jeśli jest zainstalowany) lub w ramach procesu, czy też poza bazą danych.

Właściwości węzła *applyneuralnetnode*

Węzły modelowania Sieć neuronowa mogą być używane do generowania modeli użytkowych Sieć neuronowa. Ten model użytkowy w skryptach nosi nazwę *applyneuralnetnode*. Aby uzyskać więcej informacji o skryptowej obsłudze samego węzła modelowania, patrz “Właściwości węzła *neuralnetnode*” na stronie 224.

Przestroga: W tej wersji produktu dostępna jest nowa wersja modelu użytkowego sieci neuronowej, która oferuje udoskonalone funkcje. Została ona opisana w następnej sekcji (*applyneuralnetwork*). Mimo że poprzednia wersja jest nadal dostępna, zalecamy zaktualizowanie skryptów tak, by korzystały z nowej wersji. Opis poprzedniej wersji został tutaj zamieszczony w celach referencyjnych, ale wersja ta przestanie być obsługiwana w przyszłej wersji produktu.

Tabela 173. Właściwości węzła *applyneuralnetnode*.

Właściwości węzła <i>applyneuralnetnode</i>	Wartości	Opis właściwości
<code>calculate_conf</code>	<i>flag</i>	Dostępna, gdy włączone jest generowanie kodu SQL; ta właściwość dotyczy obliczeń ufności w wygenerowanym drzewie.
<code>enable_sql_generation</code>	<i>flag</i>	
<code>nn_score_method</code>	Difference SoftMax	
<code>calculate_raw_propensities</code>	<i>flag</i>	
<code>calculate_adjusted_propensities</code>	<i>flag</i>	

Właściwości węzła *applyneuralnetworknode*

Węzły modelowania Sieć neuronowa mogą być używane do generowania modeli użytkowych Sieć neuronowa. Ten model użytkowy w skryptach nosi nazwę *applyneuralnetworknode*. Aby uzyskać więcej informacji o skryptowej obsłudze samego węzła modelowania, patrz Właściwości węzła *neuralnetworknode*.

Tabela 174. Właściwości węzła *applyneuralnetworknode*

Właściwości węzła <i>applyneuralnetworknode</i>	Wartości	Opis właściwości
<code>use_custom_name</code>	<i>flag</i>	
<code>custom_name</code>	<i>string</i>	
<code>confidence</code>	<code>onProbability</code> <code>onIncrease</code>	

Tabela 174. Właściwości węzła *applyneuralnetworknode* (kontynuacja)

Właściwości węzła <i>applyneuralnetworknode</i>	Wartości	Opis właściwości
score_category_probabilities	<i>flag</i>	
max_categories	<i>number</i>	
score_propensity	<i>flag</i>	
enable_sql_generation	udf native puresql	Służy do ustawiania opcji generowania kodu SQL podczas wykonywania strumienia. Można zlecić ocenę bazy danych i przeprowadzić ją za pomocą adaptera SPSS® Modeler Server scoring adapter (jeśli dostępne jest połączenie z bazą danych mającą zainstalowany adapter), przeprowadzić ocenę w programie SPSS Modeler lub zlecić ocenę bazy danych i przeprowadzić ją za pomocą SQL. Wartość domyślna to udf.

Właściwości *applyocsvmnode*

Węzłów SVM z jedną klasą można używać do generowania modeli użytkowych SVM z jedną klasą. Ten model użytkowy w skryptach nosi nazwę *applyocsvmnode*. Ten model użytkowy nie ma innych właściwości. Aby uzyskać więcej informacji o skryptowej obsłudze samego węzła modelowania, patrz “Właściwości *ocsvmnode*” na stronie 335.

Właściwości węzła *applyquestnode*

Węzły modelowania QUEST mogą być używane do generowania modeli użytkowych QUEST. Ten model użytkowy w skryptach nosi nazwę *applyquestnode*. Aby uzyskać więcej informacji o skryptowej obsłudze samego węzła modelowania, patrz “Właściwości węzła *questnode*” na stronie 228.

Tabela 175. Właściwości węzła *applyquestnode*.

Właściwości węzła <i>applyquestnode</i>	Wartości	Opis właściwości
enable_sql_generation	Never MissingValues NoMissingValues	Służy do ustawiania opcji generowania kodu SQL podczas wykonywania zestawu reguł.
calculate_conf	<i>flag</i>	
display_rule_id	<i>flag</i>	Dodaje do wyników oceniania zmienną określającą identyfikator węzła końcowego, do którego przypisany jest każdy rekord.
calculate_raw_propensities	<i>flag</i>	
calculate_adjusted_propensities	<i>flag</i>	

Właściwości węzła `applyr`

Węzły budowania R mogą być używane do generowania modeli użytkowych R. Ten model użytkowy w skryptach nosi nazwę `applyr`. Aby uzyskać więcej informacji o skryptowej obsłudze samego węzła modelowania, patrz “Właściwości węzła `buildr`” na stronie 184.

Tabela 176. Właściwości węzła `applyr`

Właściwości węzła <code>applyr</code>	Wartości	Opis właściwości
<code>score_syntax</code>	<i>string</i>	Komendy skryptu R realizujące ocenę modelu.
<code>convert_flags</code>	StringsAndDoubles LogicalValues	Opcja przekształcania zmiennych typu flaga.
<code>convert_datetime</code>	<i>flag</i>	Opcja przekształcania zmiennych o formacie daty lub daty/czasu na formaty daty/czasu języka R.
<code>convert_datetime_class</code>	POSIXct POSIXlt	Opcje określające format docelowy, na który przekształcane są zmienne w formacie daty i daty/czasu.
<code>convert_missing</code>	<i>flag</i>	Opcja przekształcania braków danych na wartości R NA.
<code>use_batch_size</code>	<i>flag</i>	Włącza przetwarzanie wsadowe.
<code>batch_size</code>	<i>integer</i>	Określa liczbę rekordów danych w każdej partii do przetworzenia wsadowego

Właściwości węzła `applyrandomtrees`

Węzły modelowania Drzewa losowe mogą być używane do generowania modeli użytkowych Drzewa losowe. Ten model użytkowy w skryptach nosi nazwę `applyrandomtrees`. Aby uzyskać więcej informacji o skryptowej obsłudze samego węzła modelowania, patrz “Właściwości węzła `randomtrees`” na stronie 230.

Tabela 177. Właściwości węzła `applyrandomtrees`

Właściwości węzła <code>applyrandomtrees</code>	Wartości	Opis właściwości
<code>calculate_conf</code>	<i>flag</i>	Ta właściwość dotyczy obliczeń ufności w wygenerowanym drzewie.
<code>enable_sql_generation</code>	udf native	Służy do ustawiania opcji generowania kodu SQL podczas wykonywania strumienia. Można zlecić ocenę bazy danych i przeprowadzić ją za pomocą adaptera SPSS Modeler Server (jeśli dostępne jest połączenie z bazą danych mającą zainstalowany adapter) albo przeprowadzić ocenę w programie SPSS Modeler.

Właściwości węzła `applyregressionnode`

Węzły modelowania Regresja liniowa mogą być używane do generowania modeli użytkowych Regresja liniowa. Ten model użytkowy w skryptach nosi nazwę `applyregressionnode`. Ten model użytkowy nie ma innych właściwości. Aby uzyskać więcej informacji o skryptowej obsłudze samego węzła modelowania, patrz “Właściwości węzła `regressionnode`” na stronie 231.

Właściwości węzła *applyselflearningnode*

Węzły modelowania odpowiedzi samonauzania (SLRM) mogą być używane do generowania modeli użytkowych SLRM. Ten model użytkowy w skryptach nosi nazwę *applyselflearningnode*. Aby uzyskać więcej informacji o skryptowej obsłudze samego węzła modelowania, patrz “Właściwości węzła *slrmnode*” na stronie 235.

Tabela 178. Właściwości węzła *applyselflearningnode*.

Właściwości węzła <i>applyselflearningnode</i>	Wartości	Opis właściwości
<code>max_predictions</code>	<i>number</i>	
<code>randomization</code>	<i>number</i>	
<code>scoring_random_seed</code>	<i>number</i>	
<code>sort</code>	<i>ascending</i> <i>descending</i>	Określa, czy jako pierwsze będą wyświetlane oferty z najwyższym, czy z najniższym wynikiem.
<code>model_reliability</code>	<i>flag</i>	Powoduje uwzględnienie opcji niezawodności modelu na karcie Ustawienia.

Właściwości węzła *applysequencenode*

Węzły modelowania Numer mogą być używane do generowania modeli użytkowych Numer. Ten model użytkowy w skryptach nosi nazwę *applysequencenode*. Ten model użytkowy nie ma innych właściwości. Aby uzyskać więcej informacji o skryptowej obsłudze samego węzła modelowania, patrz “Właściwości węzła *sequencenode*” na stronie 233.

Właściwości węzła *applysvmnnode*

Węzły modelowania SVM mogą być używane do generowania modeli użytkowych SVM. Ten model użytkowy w skryptach nosi nazwę *applysvmnnode*. Aby uzyskać więcej informacji o skryptowej obsłudze samego węzła modelowania, patrz “Właściwości węzła *svmnnode*” na stronie 239.

Tabela 179. Właściwości węzła *applysvmnnode*.

Właściwości węzła <i>applysvmnnode</i>	Wartości	Opis właściwości
<code>all_probabilities</code>	<i>flag</i>	
<code>calculate_raw_propensities</code>	<i>flag</i>	
<code>calculate_adjusted_propensities</code>	<i>flag</i>	

Właściwości węzła *applystpnnode*

Węzła modelowania STP można użyć do wygenerowania powiązanego modelu użytkowego, który prezentuje wyniki modelu w oknie wynikowym. Ten model użytkowy w skryptach nosi nazwę *applystpnnode*. Aby uzyskać więcej informacji o skryptowej obsłudze samego węzła modelowania, patrz “Właściwości węzła *stpnnode*” na stronie 236.

Tabela 180. Właściwości węzła *applystpnnode*

Właściwości węzła <i>applystpnnode</i>	Typ danych	Opis właściwości
<code>uncertainty_factor</code>	<i>Boolean</i>	Minimum 0, maksimum 100.

Właściwości węzła `applytcmnode`

Węzły modelowania Modelowanie przyczynowe szeregów czasowych (TCM) mogą być używane do generowania modeli użytkowych Modelowanie przyczynowe szeregów czasowych (TCM). Ten model użytkowy w skryptach nosi nazwę `applytcmnode`. Aby uzyskać więcej informacji o skryptowej obsłudze samego węzła modelowania, patrz “Właściwości węzła `tcmnode`” na stronie 240.

Tabela 181. Właściwości węzła `applytcmnode`

Właściwości węzła <code>applytcmnode</code>	Wartości	Opis właściwości
<code>ext_future</code>	<i>boolean</i>	
<code>ext_future_num</code>	<i>integer</i>	
<code>noise_res</code>	<i>boolean</i>	
<code>conf_limits</code>	<i>boolean</i>	
<code>target_fields</code>	<i>list</i>	
<code>target_series</code>	<i>list</i>	

Właściwości węzła `applyts`

Węzeł modelowania Szereg czasowy mogą być używany do generowania modelu użytkowego Szereg czasowy. Ten model użytkowy w skryptach nosi nazwę `applyts`. Aby uzyskać więcej informacji o skryptowej obsłudze samego węzła modelowania, patrz “Właściwości węzła `ts`” na stronie 244.

Tabela 182. Właściwości węzła `applyts`

Właściwości węzła <code>applyts</code>	Wartości	Opis właściwości
<code>extend_records_into_future</code>	<i>Boolean</i>	
<code>ext_future_num</code>	<i>integer</i>	
<code>compute_future_values_input</code>	<i>Boolean</i>	
<code>forecastperiods</code>	<i>integer</i>	
<code>noise_res</code>	<i>boolean</i>	
<code>conf_limits</code>	<i>boolean</i>	
<code>target_fields</code>	<i>list</i>	
<code>target_series</code>	<i>list</i>	
<code>includeTargets</code>	<i>field</i>	

Właściwości węzła `applytimeseriesnode` (zdeprecjonowanego)

Węzeł modelowania Szereg czasowy mogą być używany do generowania modelu użytkowego Szereg czasowy. Ten model użytkowy w skryptach nosi nazwę `applytimeseriesnode`. Aby uzyskać więcej informacji o skryptowej obsłudze samego węzła modelowania, patrz “Właściwości węzła `timeseriesnode` (zdeprecjonowanego)” na stronie 248.

Tabela 183. Właściwości węzła `applytimeseriesnode`.

Właściwości węzła <code>applytimeseriesnode</code>	Wartości	Opis właściwości
<code>calculate_conf</code>	<i>flag</i>	
<code>calculate_residuals</code>	<i>flag</i>	

Właściwości węzła applytreeas

Węzły modelowania Drzewo-AS mogą być używane do generowania modeli użytkowych Drzewo-AS. Ten model użytkowy w skryptach nosi nazwę *applytreeas*. Aby uzyskać więcej informacji o skryptowej obsłudze samego węzła modelowania, patrz “właściwości węzła treeas” na stronie 250.

Tabela 184. Właściwości węzła applytreeas

Właściwości węzła applytreeas	Wartości	Opis właściwości
calculate_conf	flag	Ta właściwość dotyczy obliczeń ufności w wygenerowanym drzewie.
display_rule_id	flag	Dodaje do wyników oceniania zmienną określającą identyfikator węzła końcowego, do którego przypisany jest każdy rekord.
enable_sql_generation	udf native	Służy do ustawiania opcji generowania kodu SQL podczas wykonywania strumienia. Można zlecić ocenę bazie danych i przeprowadzić ją za pomocą adaptera SPSS Modeler Server (jeśli dostępne jest połączenie z bazą danych mającą zainstalowany adapter) albo przeprowadzić ocenę w programie SPSS Modeler.

Właściwości węzła applytwostepnode

Węzły modelowania Dwustopniowa mogą być używane do generowania modeli użytkowych Dwustopniowa. Ten model użytkowy w skryptach nosi nazwę *applytwostepnode*. Ten model użytkowy nie ma innych właściwości. Aby uzyskać więcej informacji o skryptowej obsłudze samego węzła modelowania, patrz “Właściwości węzła twostepnode” na stronie 252.

Właściwości węzła applytwostepAS

Węzły modelowania TwoStep-AS mogą być używane do generowania modeli użytkowych TwoStep-AS. Ten model użytkowy w skryptach nosi nazwę *applytwostepAS*. Aby uzyskać więcej informacji o skryptowej obsłudze samego węzła modelowania, patrz “Właściwości węzła twostepAS” na stronie 253.

Tabela 185. Właściwości węzła applytwostepAS

Właściwości węzła applytwostepAS	Wartości	Opis właściwości
enable_sql_generation	udf native	Służy do ustawiania opcji generowania kodu SQL podczas wykonywania strumienia. Można zlecić ocenę bazie danych i przeprowadzić ją za pomocą adaptera SPSS® Modeler Server scoring adapter (jeśli dostępne jest połączenie z bazą danych mającą zainstalowany adapter) lub przeprowadzić ocenę w programie SPSS Modeler. Wartość domyślna to udf.

Właściwości applyxgboosttreenode

Węzeł XGBoost Tree może być używany do generowania modeli użytkowych XGBoost Tree. Ten model użytkowy w skryptach nosi nazwę *applyxgboosttreenode*. Ten model użytkowy nie ma innych właściwości. Aby uzyskać więcej informacji o skryptowej obsłudze samego węzła modelowania, patrz “Właściwość xgboosttreenode” na stronie 333.

Właściwości `applyxgboostlinearnode`

Węzły XGBoost Linear mogą być używane do generowania modeli użytkowych XGBoost Linear. Ten model użytkowy w skryptach nosi nazwę `applyxgboostlinearnode`. Ten model użytkowy nie ma innych właściwości. Aby uzyskać więcej informacji o skryptowej obsłudze samego węzła modelowania, patrz “Właściwości `xgboostlinearnode`” na stronie 335.

Rozdział 15. Właściwości węzłów związanych z modelowaniem w bazie danych

IBM SPSS Modeler może współpracować z narzędziami do eksploracji i modelowania danych oferowanymi przez różnych dostawców baz danych, w tym z oprogramowaniem Microsoft SQL Server Analysis Services, Oracle Data Mining, a także IBM Netezza Analytics. W aplikacji IBM SPSS Modeler można budować i oceniać modele przy użyciu rodzimych algorytmów udostępnianych przez te bazy dane. Do tworzenia modeli w bazach danych i operowania na nich można też wykorzystać język skryptowy i właściwości opisane w niniejszej sekcji.

Na przykład poniższy fragment skryptu ilustruje tworzenie modelu Microsoft Decision Trees za pomocą interfejsu skryptowego programu IBM SPSS Modeler:

```
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([])
```

Właściwości węzłów związanych z modelowaniem w bazie danych Microsoft

Właściwości węzłów związanych z modelowaniem w bazie danych Microsoft

Właściwości wspólne

Poniższe właściwości są wspólne dla niektórych lub wszystkich węzłów modelowania w bazie danych firmy Microsoft.

Tabela 186. Wspólne właściwości węzłów Microsoft

Wspólne właściwości węzłów Microsoft	Wartości	Opis właściwości
analysis_database_name	string	Nazwa bazy danych usług Analysis Services.
analysis_server_name	string	Nazwa hosta usług Analysis Services.
use_transactional_data	flag	Określa, czy dane wejściowe mają format tabelaryczny, czy transakcyjny.
inputs	list	Zmienne wejściowe dla danych tabelarycznych.

Tabela 186. Wspólne właściwości węzłów Microsoft (kontynuacja)

Wspólne właściwości węzłów Microsoft	Wartości	Opis właściwości
target	<i>field</i>	Zmienna przewidywana (nie ma zastosowania w przypadku węzłów MS Clustering i Sequence Clustering).
unique_field	<i>field</i>	Zmienna kluczowa.
msas_parameters	<i>structured</i>	Parametry algorytmu. Więcej informacji można znaleźć w temacie "Parametry algorytmu" na stronie 275.
with_drillthrough	<i>flag</i>	Z opcją drążenia wszcz.

MS Decision Tree

Węzły typu *mstreenode* nie mają charakterystycznych właściwości. Właściwości wspólne dla węzłów bazy danych Microsoft opisano na początku niniejszej sekcji.

MS Clustering

Węzły typu *msclusternode* nie mają charakterystycznych właściwości. Właściwości wspólne dla węzłów bazy danych Microsoft opisano na początku niniejszej sekcji.

MS Association Rules

Węzły typu *msassocnode* mają następujące właściwości charakterystyczne:

Tabela 187. właściwości węzła *msassocnode*

Właściwości węzła <i>msassocnode</i>	Wartości	Opis właściwości
id_field	<i>field</i>	Identyfikuje każdą transakcję w danych.
trans_inputs	<i>list</i>	Zmienne wejściowe dla danych transakcyjnych.
transactional_target	<i>field</i>	Zmienna przewidywana (dane transakcyjne).

MS Naive Bayes

Węzły typu *msbayesnode* nie mają charakterystycznych właściwości. Właściwości wspólne dla węzłów bazy danych Microsoft opisano na początku niniejszej sekcji.

MS Linear Regression

Węzły typu *msregressionnode* nie mają charakterystycznych właściwości. Właściwości wspólne dla węzłów bazy danych Microsoft opisano na początku niniejszej sekcji.

MS Neural Network

Węzły typu *msneuralnetworknode* nie mają charakterystycznych właściwości. Właściwości wspólne dla węzłów bazy danych Microsoft opisano na początku niniejszej sekcji.

MS Logistic Regression

Węzły typu *mslogisticnode* nie mają charakterystycznych właściwości. Właściwości wspólne dla węzłów bazy danych Microsoft opisano na początku niniejszej sekcji.

MS Time Series

Węzły typu `mstimeseriesnode` nie mają charakterystycznych właściwości. Właściwości wspólne dla węzłów bazy danych Microsoft opisano na początku niniejszej sekcji.

MS Sequence Clustering

Węzły typu `mssequenceclusternode` mają następujące właściwości charakterystyczne:

Tabela 188. właściwości węzła `mssequenceclusternode`

Właściwości węzła <code>mssequenceclusternode</code>	Wartości	Opis właściwości
<code>id_field</code>	<i>field</i>	Identyfikuje każdą transakcję w danych.
<code>input_fields</code>	<i>list</i>	Zmienne wejściowe dla danych transakcyjnych.
<code>sequence_field</code>	<i>field</i>	Identyfikator sekwencji.
<code>target_field</code>	<i>field</i>	Zmienna przewidywana (dane tabelaryczne).

Parametry algorytmu

Każdy typ modelu współpracującego z bazą danych Microsoft ma charakterystyczne parametry, które można ustawiać za pośrednictwem właściwości `msas_parameters`, na przykład:

```
stream = modeler.script.stream()
msregressionnode = stream.findByType("msregression", None)
msregressionnode.setPropertyValue("msas_parameters", [{"MAXIMUM_INPUT_ATTRIBUTES", 255}, {"MAXIMUM_OUTPUT_ATTRIBUTES", 255}])
```

Parametry te pochodzą z serwera SQL Server. Aby wyświetlić parametry właściwe dla każdego z węzłów:

1. Umieść węzeł źródłowy bazy danych w obszarze roboczym.
2. Otwórz węzeł źródłowy bazy danych.
3. Wybierz poprawne źródło z listy rozwijanej **Źródło danych**.
4. Wybierz poprawną tabelę z listy **Nazwa tabeli**.
5. Kliknij przycisk **OK**, aby zamknąć węzeł źródłowy bazy.
6. Dołącz węzeł modelowania w bazie danych Microsoft, którego właściwości chcesz wyświetlić.
7. Otwórz węzeł modelowania w bazie danych.
8. Wybierz kartę **Zaawansowane**.

Zostaną wyświetlone dostępne właściwości `msas_parameters` otwartego węzła.

Właściwości modelu użytkowego Microsoft

Modele użytkowe utworzone przy użyciu węzłów modelowania w bazie danych firmy Microsoft mają następujące właściwości.

MS Decision Tree

Tabela 189. Właściwości węzła `MS Decision Tree`

Właściwości węzła <code>appliedstreenode</code>	Wartości	Opis
<code>analysis_database_name</code>	<i>string</i>	Ten węzeł może być oceniany bezpośrednio w strumieniu. Ta właściwość służy do określania nazwy bazy danych Analysis Services.
<code>analysis_server_name</code>	<i>string</i>	Nazwa hosta serwera Analysis Services.

Tabela 189. Właściwości węzła MS Decision Tree (kontynuacja)

Właściwości węzła appliedstreenode	Wartości	Opis
datasource	string	Nazwa źródła danych (DSN) w interfejsie ODBC serwera SQL Server.
sql_generate	flag	Umożliwia generowanie kodu SQL.

MS Linear Regression

Tabela 190. Właściwości węzła MS Linear Regression

Właściwości węzła appliedregressionnode	Wartości	Opis
analysis_database_name	string	Ten węzeł może być oceniany bezpośrednio w strumieniu. Ta właściwość służy do określania nazwy bazy danych Analysis Services.
analysis_server_name	string	Nazwa hosta serwera Analysis Services.

MS Neural Network

Tabela 191. Właściwości węzła MS Neural Network

Właściwości węzła appliedneuralnetworknode	Wartości	Opis
analysis_database_name	string	Ten węzeł może być oceniany bezpośrednio w strumieniu. Ta właściwość służy do określania nazwy bazy danych Analysis Services.
analysis_server_name	string	Nazwa hosta serwera Analysis Services.

MS Logistic Regression

Tabela 192. Właściwości węzła MS Logistic Regression

Właściwości węzła appliedlogisticnode	Wartości	Opis
analysis_database_name	string	Ten węzeł może być oceniany bezpośrednio w strumieniu. Ta właściwość służy do określania nazwy bazy danych Analysis Services.
analysis_server_name	string	Nazwa hosta serwera Analysis Services.

MS Time Series

Tabela 193. Właściwości węzła MS Time Series

Właściwości węzła appliedtimeseriesnode	Wartości	Opis
analysis_database_name	string	Ten węzeł może być oceniany bezpośrednio w strumieniu. Ta właściwość służy do określania nazwy bazy danych Analysis Services.
analysis_server_name	string	Nazwa hosta serwera Analysis Services.

Tabela 193. Właściwości węzła MS Time Series (kontynuacja)

Właściwości węzła appliedtimeseriesnode	Wartości	Opis
start_from	new_prediction historical_prediction	Określa, czy mają być dokonywane predykcje przyszłe, czy historyczne.
new_step	number	Określa początkowy przedział czasu dla przyszłych predykcji.
historical_step	number	Określa początkowy przedział czasu dla predykcji historycznych.
end_step	number	Określa końcowy przedział czasu dla predykcji.

MS Sequence Clustering

Tabela 194. Właściwości węzła MS Sequence Clustering

Właściwości węzła appliedsequenceclusternode	Wartości	Opis
analysis_database_name	string	Ten węzeł może być oceniany bezpośrednio w strumieniu. Ta właściwość służy do określania nazwy bazy danych Analysis Services.
analysis_server_name	string	Nazwa hosta serwera Analysis Services.

Właściwości węzłów związanych z modelowaniem w bazie danych Oracle

Właściwości węzłów związanych z modelowaniem w bazie danych Oracle

Poniższe właściwości są wspólne dla niektórych lub wszystkich węzłów modelowania w bazie danych Oracle.

Tabela 195. Wspólne właściwości węzłów Oracle.

Wspólne właściwości węzłów Oracle	Wartości	Opis właściwości
target	field	
inputs	Lista zmiennych	
partition	field	Umożliwia podział danych na osobne próby do uczenia, testowania i walidacji podczas budowania modelu.
datasource		
username		
password		
epassword		
use_model_name	flag	
model_name	string	Niestandardowa nazwa nowego modelu.
use_partitioned_data	flag	Jeśli zdefiniowano zmienną dzielącą na podzbiory, ta opcja umożliwia użycie podczas budowania modelu wyłącznie danych z podzbioru uczącego.
unique_field	field	

Tabela 195. Wspólne właściwości węzłów Oracle (kontynuacja).

Wspólne właściwości węzłów Oracle	Wartości	Opis właściwości
auto_data_prep	flag	Włącza lub wyłącza funkcję automatycznego przygotowywania danych na serwerze Oracle (tylko w bazach danych 11g).
costs	structured	Właściwość ustrukturyzowana o postaci: [[drugA drugB 1.5] [drugA drugC 2.1]], gdzie argumenty [] są rzeczywistymi przewidywanymi kosztami.
mode	Simple Expert	Wartość Simple powoduje ignorowanie niektórych właściwości, zgodnie z opisem poszczególnych węzłów.
use_prediction_probability	flag	
prediction_probability	string	
use_prediction_set	flag	

Oracle Naive Bayes

Dostępne są następujące właściwości węzłów typu oranbnode.

Tabela 196. Właściwości węzła oranbnode

Właściwości węzła oranbnode	Wartości	Opis właściwości
singleton_threshold	number	0,0–1,0.*
pairwise_threshold	number	0,0–1,0.*
priors	Data Equal Custom	
custom_priors	structured	Właściwość ustrukturyzowana o postaci: set :oranbnode.custom_priors = [[drugA 1][drugB 2][drugC 3][drugX 4][drugY 5]]

* Właściwość ignorowana, jeśli mode ma wartość Simple.

Oracle Adaptive Bayes

Dostępne są następujące właściwości węzłów typu oraabnnode.

Tabela 197. Właściwości węzła oraabnnode

Właściwości węzła oraabnnode	Wartości	Opis właściwości
model_type	SingleFeature MultiFeature NaiveBayes	
use_execution_time_limit	flag	*
execution_time_limit	integer	Wartość musi być większa niż 0.*
max_naive_bayes_predictors	integer	Wartość musi być większa niż 0.*
max_predictors	integer	Wartość musi być większa niż 0.*
priors	Data Equal Custom	

Tabela 197. Właściwości węzła oraabnnode (kontynuacja)

Właściwości węzła oraabnnode	Wartości	Opis właściwości
custom_priors	<i>structured</i>	Właściwość ustrukturyzowana o postaci: set :oraabnnode.custom_priors = [[drugA 1][drugB 2][drugC 3][drugX 4][drugY 5]]

* Właściwość ignorowana, jeśli mode ma wartość Simple.

Oracle Support Vector Machines

Dostępne są następujące właściwości węzłów typu orasvmnode.

Tabela 198. Właściwości węzła orasvmnode

Właściwości węzła orasvmnode	Wartości	Opis właściwości
active_learning	Enable Disable	
kernel_function	Linear Gaussian System	
normalization_method	zscore minmax none	
kernel_cache_size	<i>integer</i>	Tylko algorytm gaussowski. Wartość musi być większa niż 0.*
convergence_tolerance	<i>number</i>	Wartość musi być większa niż 0.*
use_standard_deviation	<i>flag</i>	Tylko algorytm gaussowski.*
standard_deviation	<i>number</i>	Wartość musi być większa niż 0.*
use_epsilon	<i>flag</i>	Tylko modele regresji.*
epsilon	<i>number</i>	Wartość musi być większa niż 0.*
use_complexity_factor	<i>flag</i>	*
complexity_factor	<i>number</i>	*
use_outlier_rate	<i>flag</i>	Tylko wariant z jedną klasą.*
outlier_rate	<i>number</i>	Tylko wariant z jedną klasą. 0,0–1,0.*
weights	Data Equal Custom	
custom_weights	<i>structured</i>	Właściwość ustrukturyzowana o postaci: set :orasvmnode.custom_weights = [[drugA 1][drugB 2][drugC 3][drugX 4][drugY 5]]

* Właściwość ignorowana, jeśli mode ma wartość Simple.

Modele Oracle Generalized Linear

Dostępne są następujące właściwości węzłów typu `oraglmnode`.

Tabela 199. Właściwości węzła `oraglmnode`

Właściwości węzła <code>oraglmnode</code>	Wartości	Opis właściwości
<code>normalization_method</code>	zscore minmax none	
<code>missing_value_handling</code>	ReplaceWithMean UseCompleteRecords	
<code>use_row_weights</code>	<i>flag</i>	*
<code>row_weights_field</code>	<i>field</i>	*
<code>save_row_diagnostics</code>	<i>flag</i>	*
<code>row_diagnostics_table</code>	<i>string</i>	*
<code>coefficient_confidence</code>	<i>number</i>	*
<code>use_reference_category</code>	<i>flag</i>	*
<code>reference_category</code>	<i>string</i>	*
<code>ridge_regression</code>	Auto Off On	*
<code>parameter_value</code>	<i>number</i>	*
<code>vif_for_ridge</code>	<i>flag</i>	*

* Właściwość ignorowana, jeśli `mode` ma wartość `Simple`.

Oracle Decision Tree

Dostępne są następujące właściwości węzłów typu `oradecisiontreenode`.

Tabela 200. Właściwości węzła `oradecisiontreenode`

Właściwości węzła <code>oradecisiontreenode</code>	Wartości	Opis właściwości
<code>use_costs</code>	<i>flag</i>	
<code>impurity_metric</code>	Entropy Gini	
<code>term_max_depth</code>	<i>integer</i>	2–20.*
<code>term_minpct_node</code>	<i>number</i>	0,0–10,0.*
<code>term_minpct_split</code>	<i>number</i>	0,0–20,0.*
<code>term_minrec_node</code>	<i>integer</i>	Wartość musi być większa niż 0.*
<code>term_minrec_split</code>	<i>integer</i>	Wartość musi być większa niż 0.*
<code>display_rule_ids</code>	<i>flag</i>	*

* Właściwość ignorowana, jeśli `mode` ma wartość `Simple`.

Oracle O-Cluster

Dostępne są następujące właściwości węzłów typu oraoclusternode.

Tabela 201. Właściwości węzła oraoclusternode

Właściwości węzła oraoclusternode	Wartości	Opis właściwości
max_num_clusters	<i>integer</i>	Wartość musi być większa niż 0.
max_buffer	<i>integer</i>	Wartość musi być większa niż 0.*
sensitivity	<i>number</i>	0,0–1,0.*

* Właściwość ignorowana, jeśli mode ma wartość Simple.

Oracle KMeans

Dostępne są następujące właściwości węzłów typu orakmeansnode.

Tabela 202. Właściwości węzła orakmeansnode

Właściwości węzła orakmeansnode	Wartości	Opis właściwości
num_clusters	<i>integer</i>	Wartość musi być większa niż 0.
normalization_method	zscore minmax none	
distance_function	Euclidean Cosinus	
iterations	<i>integer</i>	0–20.*
conv_tolerance	<i>number</i>	0,0–0,5.*
split_criterion	Variance Size	Domyślnie Variance.*
num_bins	<i>integer</i>	Wartość musi być większa niż 0.*
block_growth	<i>integer</i>	1–5.*
min_pct_attr_support	<i>number</i>	0,0–1,0.*

* Właściwość ignorowana, jeśli mode ma wartość Simple.

Oracle NMF

Dostępne są następujące właściwości węzłów typu oranmfnode.

Tabela 203. Właściwości węzła oranmfnode

Właściwości węzła oranmfnode	Wartości	Opis właściwości
normalization_method	minmax none	
use_num_features	<i>flag</i>	*
num_features	<i>integer</i>	0–1. Wartość domyślna jest szacowana przez algorytm na podstawie danych.*
random_seed	<i>number</i>	*
num_iterations	<i>integer</i>	0–500.*

Tabela 203. Właściwości węzła oranmfnode (kontynuacja)

Właściwości węzła oranmfnode	Wartości	Opis właściwości
conv_tolerance	number	0,0–0,5.*
display_all_features	flag	*

* Właściwość ignorowana, jeśli mode ma wartość Simple.

Oracle Apriori

Dostępne są następujące właściwości węzłów typu oraapriorinode.

Tabela 204. Właściwości węzła oraapriorinode

Właściwości węzła oraapriorinode	Wartości	Opis właściwości
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 Minimum Description Length (MDL)

Węzły typu oramdlnode nie mają charakterystycznych właściwości. Właściwości wspólne dla węzłów bazy danych Oracle opisano na początku niniejszej sekcji.

Oracle Attribute Importance (AI)

Dostępne są następujące właściwości węzłów typu oraainode.

Tabela 205. Właściwości węzła oraainode

Właściwości węzła oraainode	Wartości	Opis właściwości
custom_fields	flag	Wartość true umożliwia określenie zmiennej przewidywanej, wejściowej i innych zmiennych dla bieżącego węzła. Wartość false powoduje użycie bieżących ustawień z wcześniejszego węzła Typ.
selection_mode	ImportanceLevel ImportanceValue TopN	
select_important	flag	Gdy selection_mode ma wartość ImportanceLevel, określa, czy wybierać zmienne ważne
important_label	string	Określa etykietę rangi "ważne".
select_marginal	flag	Gdy selection_mode ma wartość ImportanceLevel, określa, czy wybierać zmienne brzegowe.
marginal_label	string	Określa etykietę rangi "brzegowe".
important_above	number	0,0–1,0.
select_unimportant	flag	Gdy selection_mode ma wartość ImportanceLevel, określa, czy wybierać zmienne nieważne.
unimportant_label	string	Określa etykietę rangi "nieważne".

Tabela 205. Właściwości węzła oraainode (kontynuacja)

Właściwości węzła oraainode	Wartości	Opis właściwości
unimportant_below	number	0,0–1,0.
importance_value	number	Gdy selection_mode ma wartość ImportanceValue, określa wartość odcięcia, która ma być używana. Przyjmuje wartości z zakresu od 0 do 100.
top_n	number	Gdy selection_mode ma wartość TopN, określa wartość odcięcia, która ma być używana. Przyjmuje wartości z zakresu od 0 do 1000.

Właściwości modeli użytkowych Oracle

Modele użytkowe utworzone przy użyciu modelu Oracle mają następujące właściwości.

Oracle Naive Bayes

Węzły typu applyoranbnode nie mają charakterystycznych właściwości.

Oracle Adaptive Bayes

Węzły typu applyoraabnode nie mają charakterystycznych właściwości.

Oracle Support Vector Machines

Węzły typu applyorasvmnode nie mają charakterystycznych właściwości.

Oracle Decision Tree

Dostępne są następujące właściwości węzłów typu applyoradecisiontreenode.

Tabela 206. Właściwości węzła applyoradecisiontreenode

Właściwości węzła applyoradecisiontreenode	Wartości	Opis właściwości
use_costs	flag	
display_rule_ids	flag	

Oracle O-Cluster

Węzły typu applyoraoclusternode nie mają charakterystycznych właściwości.

Oracle KMeans

Węzły typu applyorakmeansnode nie mają charakterystycznych właściwości.

Oracle NMF

Węzeł typu applyoranmfnode ma następującą właściwość:

Tabela 207. Właściwości węzła applyoranmfnode

Właściwości węzła applyoranmfnode	Wartości	Opis właściwości
display_all_features	flag	

Oracle Apriori

Tego modelu użytkowego nie można stosować w skryptach.

Oracle MDL

Tego modelu użytkowego nie można stosować w skryptach.

Właściwości węzłów związanych z modelowaniem w bazie danych IBM Netezza Analytics

Właściwości węzłów związanych z modelowaniem w bazie danych Netezza

Poniższe właściwości są wspólne dla niektórych lub wszystkich węzłów modelowania w bazie danych firmy IBM Netezza.

Tabela 208. Wspólne właściwości węzłów Netezza.

Wspólne właściwości węzłów Netezza	Wartości	Opis właściwości
custom_fields	<i>flag</i>	Wartość true umożliwia określenie zmiennej przewidywanej, wejściowej i innych zmiennych dla bieżącego węzła. Wartość false powoduje użycie bieżących ustawień z wcześniejszego węzła Typ.
inputs	<i>[field1 ... fieldN]</i>	Zmienne wejściowe lub predyktory używane przez model.
target	<i>field</i>	Zmienna docelowa (ciągła lub jakościowa).
record_id	<i>field</i>	Zmienna, która będzie używana jako unikatowy identyfikator rekordu.
use_upstream_connection	<i>flag</i>	Wartość (true) powoduje, że używane są dane połączenia określone we wcześniejszym węźle. Nie używana, jeśli określono właściwość <i>move_data_to_connection</i> .
move_data_connection	<i>flag</i>	Wartość true powoduje, że dane są przenoszone do bazy danych określonej przez właściwość <i>connection</i> . Nie używana, jeśli określono właściwość <i>use_upstream_connection</i> .
connection	<i>structured</i>	Łańcuch połączenia z bazą danych Netezza, w której przechowywany jest model. Właściwość ustrukturyzowana o postaci: ['odbc' '<dsn>' '<username>' '<psw>' '<catname>' '<conn_attribs>' [true false]] gdzie: <dsn> jest nazwą źródła danych <username> i <psw> to nazwa użytkownika i hasło dla bazy danych <catname> to nazwa katalogu <conn_attribs> to atrybuty połączenia true false wskazuje, czy potrzebne jest hasło.
table_name	<i>string</i>	Nazwa tabeli bazy danych, w której przechowywany jest model.
use_model_name	<i>flag</i>	Wartość true powoduje, że używana jest nazwa modelu określona we właściwości <i>model_name</i> . W przeciwnym razie nazwa modelu jest tworzona przez system.
model_name	<i>string</i>	Niestandardowa nazwa nowego modelu.

Tabela 208. Wspólne właściwości węzłów Netezza (kontynuacja).

Wspólne właściwości węzłów Netezza	Wartości	Opis właściwości
include_input_fields	<i>flag</i>	Wartość true powoduje, że wszystkie zmienne wejściowe są przekazywane do następnego węzła. W przeciwnym razie przekazywana jest tylko wartość record_id i zmienne wygenerowane przez model.

Netezza Decision Tree

Dostępne są następujące właściwości węzłów typu netezzadectreenode.

Tabela 209. właściwości węzła netezzadectreenode

Właściwości węzła netezzadectreenode	Wartości	Opis właściwości
impurity_measure	Entropy Gini	Miara zanieczyszczenia, używana do określania najlepszego miejsca podziału drzewa.
max_tree_depth	<i>integer</i>	Maksymalna liczba poziomów, do której może rozrosnąć się drzewo. Wartość domyślna to 62 (dopuszczalne maksimum).
min_improvement_splits	<i>number</i>	Minimalna poprawa zanieczyszczenia wymagana, by nastąpił podział. Wartość domyślna to 0.01.
min_instances_split	<i>integer</i>	Minimalna liczba niepodzielonych rekordów, jaka musi pozostać, aby możliwy był podział. Wartość domyślna to 2 (dopuszczalne minimum).
weights	<i>structured</i>	Względne wagi klas. Właściwość ustrukturyzowana o postaci: set :netezza_dectree.weights = [[drugA 0.3][drugB 0.6][drugC 3][drugX 4][drugY 5]] Domyślnie wszystkie klasy mają wagę 1.
pruning_measure	Acc wAcc	Domyślnie Acc (dokładność). Alternatywne ustawienie wAcc (dokładność ważona) powoduje, że przy przycinaniu uwzględniane są wagi klas.
prune_tree_options	allTrainingData partitionTrainingData useOtherTable	Domyślne ustawienie allTrainingData powoduje, że oceniana jest dokładność modelu. Należy użyć wartości partitionTrainingData w celu określania odsetka danych uczących, jaki ma być używany, albo wartości useOtherTable, aby użyć zbioru danych uczących z określonej tabeli bazy danych.
perc_training_data	<i>number</i>	Jeśli właściwość prune_tree_options ma wartość partitionTrainingData, określa odsetek danych, jaki ma być używany do uczenia.

Tabela 209. właściwości węzła *netezzadectreenode* (kontynuacja)

Właściwości węzła netezzadectreenode	Wartości	Opis właściwości
prune_seed	<i>integer</i>	Wartość początkowa generatora liczb losowych używana do replikacji wyników analizy, gdy właściwość prune_tree_options jest ustawiona na partitionTrainingData ; domyślnie 1.
pruning_table	<i>string</i>	Nazwa tabeli osobnego zbioru danych do przycinania służącego do szacowania dokładności modelu.
compute_probabilities	<i>flag</i>	Wartość true powoduje generowanie zarówno zmiennej poziomu ufności (prawdopodobieństwa), jak i zmiennej predykcyjnej.

Netezza K-Means

Dostępne są następujące właściwości węzłów typu *netezzakmeansnode*.

Tabela 210. Właściwości węzła *netezzakmeansnode*

Właściwości węzła netezzakmeansnode	Wartości	Opis właściwości
distance_measure	Euclidean Manhattan Canberra maximum	Metoda, która ma być stosowana do pomiaru odległości między punktami danych.
num_clusters	<i>integer</i>	Liczba grup do utworzenia; domyślnie 3.
max_iterations	<i>integer</i>	Liczba iteracji algorytmu, po której uczenie modelu ma zostać przerwane; domyślnie 5.
rand_seed	<i>integer</i>	Wartość początkowa generatora liczb losowych używana do replikacji wyników analizy; domyślnie 12345.

Netezza Bayes Net

Dostępne są następujące właściwości węzłów typu *netezabayesnode*.

Tabela 211. Właściwości węzła *netezabayesnode*

Właściwości węzła netezabayesnode	Wartości	Opis właściwości
base_index	<i>integer</i>	Identyfikator liczbowy przypisany do pierwszej zmiennej wejściowej na potrzeby wewnętrznego zarządzania; domyślnie 777.
sample_size	<i>integer</i>	Wielkość próby, jaka ma być stosowana, gdy liczba atrybutów jest bardzo duża; domyślnie 10 000.
display_additional_information	<i>flag</i>	Wartość true powoduje, że w oknie dialogowym z komunikatem wyświetlane są dodatkowe informacje o postępie.
type_of_prediction	best neighbors nn-neighbors	Typ algorytmu predykcyjnego, który ma być stosowany: best (najlepiej skorelowany sąsiad), neighbors (ważona predykcja sąsiadów) albo nn-neighbors (sąsiedzi inni niż null).

Netezza Naive Bayes

Dostępne są następujące właściwości węzłów typu `netezzanaivebayesnode`.

Tabela 212. Właściwości węzła `netezzanaivebayesnode`

Właściwości węzła <code>netezzanaivebayesnode</code>	Wartości	Opis właściwości
<code>compute_probabilities</code>	<i>flag</i>	Wartość <code>true</code> powoduje generowanie zarówno zmiennej poziomu ufności (prawdopodobieństwa), jak i zmiennej predykcyjnej.
<code>use_m_estimation</code>	<i>flag</i>	Wartość <code>true</code> powoduje stosowanie techniki m-estymacji, która unika prawdopodobieństw zerowych.

Netezza KNN

Dostępne są następujące właściwości węzłów typu `netezzaknnnode`.

Tabela 213. Właściwości węzła `netezzaknnnode`

Właściwości węzła <code>netezzaknnnode</code>	Wartości	Opis właściwości
<code>weights</code>	<i>structured</i>	Właściwość ustrukturyzowana służąca do przypisywania wag do poszczególnych klas. Przykład: <code>set :netezzaknnnode.weights = [[drugA 0.3][drugB 0.6]]</code>
<code>distance_measure</code>	Euclidean Manhattan Canberra Maksimum	Metoda, która ma być stosowana do pomiaru odległości między punktami danych.
<code>num_nearest_neighbors</code>	<i>integer</i>	Liczba najbliższych sąsiadów dla konkretnej obserwacji; domyślnie 3.
<code>standardize_measurements</code>	<i>flag</i>	Wartość <code>true</code> powoduje standaryzację pomiarów dla ciągłych danych wejściowych przed obliczeniem wartości odległości.
<code>use_coresets</code>	<i>flag</i>	Wartość <code>true</code> powoduje używanie próbkowania zestawu podstawowego w celu przyspieszenia obliczeń, gdy używane są duże zbiory danych.

Netezza Divisive Clustering

Dostępne są następujące właściwości węzłów typu `netez zadivclusternode`.

Tabela 214. Właściwości węzła `netez zadivclusternode`

Właściwości węzła <code>netez zadivclusternode</code>	Wartości	Opis właściwości
<code>distance_measure</code>	Euclidean Manhattan Canberra Maksimum	Metoda, która ma być stosowana do pomiaru odległości między punktami danych.
<code>max_iterations</code>	<i>integer</i>	Maksymalna liczba iteracji algorytmu, po której uczenie modelu ma zostać przerwane; domyślnie 5.
<code>max_tree_depth</code>	<i>integer</i>	Maksymalna liczba poziomów, na które zostanie podzielony zbiór danych; domyślnie 3.
<code>rand_seed</code>	<i>integer</i>	Wartość początkowa generatora liczb losowych, używana do replikacji analiz; domyślnie 12345.

Tabela 214. Właściwości węzła *netezzadivclusternode* (kontynuacja)

Właściwości węzła netezzadivclusternode	Wartości	Opis właściwości
min_instances_split	<i>integer</i>	Minimalna liczba rekordów, jaka może zostać podzielona; domyślnie 5.
level	<i>integer</i>	Poziom hierarchii, do którego mają być oceniane rekordy; domyślnie -1.

Netezza PCA

Dostępne są następujące właściwości węzłów typu *netezzapcanode*.

Tabela 215. Właściwości węzła *netezzapcanode*

Właściwości węzła netezzapcanode	Wartości	Opis właściwości
center_data	<i>flag</i>	Wartość true (domyślna) powoduje wyśrodkowanie danych (znane również jako „odejmowanie średniej”) przed analizą.
perform_data_scaling	<i>flag</i>	Wartość true powoduje skalowanie danych przed analizą. Taki sposób postępowania sprawia, że analiza jest mniej dowolna, gdy różne zmienne są mierzone w różnych jednostkach.
force_eigensolve	<i>flag</i>	Wartość true powoduje, że stosowana jest mniej dokładna, ale szybsza metoda znajdowania głównych składowych.
pc_number	<i>integer</i>	Liczba głównych składowych, do której ma być zredukowany zbiór danych; domyślnie 1.

Netezza Regression Tree

Dostępne są następujące właściwości węzłów typu *netezzaregtreenode*.

Tabela 216. Właściwości węzła *netezzaregtreenode*

Właściwości węzła netezzaregtreenode	Wartości	Opis właściwości
max_tree_depth	<i>integer</i>	Maksymalna liczba poziomów, do której może rosnąć drzewo poniżej węzła głównego; domyślnie 10.
split_evaluation_measure	Variance	Miara zanieczyszczenia klasy, używana do określania najlepszego miejsca podziału drzewa; domyślną (i obecnie jedynie dostępną) opcją jest Variance.
min_improvement_splits	<i>number</i>	Minimalna redukcja zanieczyszczenia pozwalająca na utworzenie nowego podziału w drzewie.
min_instances_split	<i>integer</i>	Minimalna liczba rekordów, jaka może zostać podzielona.
pruning_measure	mse R2 pearson spearman	Metoda, która ma być stosowana do przycinania.

Tabela 216. Właściwości węzła *netezzaregtreenode* (kontynuacja)

Właściwości węzła netezzaregtreenode	Wartości	Opis właściwości
prune_tree_options	allTrainingData partitionTrainingData useOtherTable	Domyślne ustawienie allTrainingData powoduje, że oceniana jest dokładność modelu. Należy użyć wartości partitionTrainingData w celu określania odsetka danych uczących, jaki ma być używany, albo wartości useOtherTable, aby użyć zbioru danych uczących z określonej tabeli bazy danych.
perc_training_data	number	Jeśli właściwość prune_tree_options ma wartość PercTrainingData, określa odsetek danych, jaki ma być używany do uczenia.
prune_seed	integer	Wartość początkowa generatora liczb losowych używana do replikacji wyników analizy, gdy właściwość prune_tree_options jest ustawiona na PercTrainingData; domyślnie 1.
pruning_table	string	Nazwa tabeli osobnego zbioru danych do przycinania służącego do szacowania dokładności modelu.
compute_probabilities	flag	Wartość true oznacza, że wariancje przypisanych klas powinny być uwzględnione w wynikach.

Netezza Linear Regression

Dostępne są następujące właściwości węzłów typu *netezzalineressionnode*.

Tabela 217. właściwości węzła *netezzalineressionnode*

Właściwości węzła netezzalineressionnode	Wartości	Opis właściwości
use_svd	flag	Wartość true powoduje użycie macierzy dekompozycji wartości osobliwych zamiast pierwotnej macierzy w celu przyspieszenia obliczeń i uzyskania większej dokładności numerycznej.
include_intercept	flag	Wartość true (domyślna) zwiększa ogólną dokładność rozwiązania.
calculate_model_diagnostics	flag	Wartość true powoduje wykonanie obliczeń diagnostycznych na modelu.

Netezza Time Series

Dostępne są następujące właściwości węzłów typu *netezzatimeseriesnode*.

Tabela 218. Właściwości węzła *netezzatimeseriesnode*

Właściwości węzła netezzatimeseriesnode	Wartości	Opis właściwości
time_points	field	Zmienna wejściowa zawierająca wartości daty lub czasu dla szeregu czasowego.

Tabela 218. Właściwości węzła *netezzatimeseriesnode* (kontynuacja)

Właściwości węzła <i>netezzatimeseriesnode</i>	Wartości	Opis właściwości
time_series_ids	<i>field</i>	Zmienna wejściowa zawierająca identyfikatory szeregów czasowych; używana, jeśli dane wejściowe zawierają więcej niż jeden szereg czasowy.
model_table	<i>field</i>	Nazwa tabeli bazy danych, w której przechowywany jest model szeregów czasowych Netezza.
description_table	<i>field</i>	Nazwa tabeli wejściowej zawierającej nazwy i opisy szeregów czasowych.
seasonal_adjustment_table	<i>field</i>	Nazwa tabeli wynikowej, w której będą zapisywane skorygowane sezonowo wartości obliczone przez algorytm wygładzania wykładniczego lub dekompozycji trendu sezonowego.
algorithm_name	SpectralAnalysis lub spectral ExponentialSmoothing lub esmoothing ARIMA SeasonalTrendDecomposition lub std	Algorytm, który ma być używany do modelowania szeregów czasowych.
trend_name	N A DA M DM	Typ trendu dla wygładzania wykładniczego: N — brak A — addytywny DA — wygasający addytywny M — multiplikatywny DM — wygasający multiplikatywny
seasonality_type	N A P	Typ sezonowości dla wygładzania wykładniczego: N — brak A — addytywny M — multiplikatywny
interpolation_method	linear cubicspline exponentialspline	Metoda interpolacja, która ma być stosowana.
timerange_setting	SD SP	Przedział czasu, jaki ma być używany: SD — określony przez system (cały przedział ujęty w danych szeregów czasowych) SP — określony przez użytkownika we właściwościach <i>earliest_time</i> i <i>latest_time</i>

Tabela 218. Właściwości węzła *netezzatimeseriesnode* (kontynuacja)

Właściwości węzła <i>netezzatimeseriesnode</i>	Wartości	Opis właściwości
earliest_time	<i>integer</i>	Wartość początkowa i końcowa, jeśli <i>timerange_setting</i> ma wartość SP. Obowiązuje format wartości <i>time_points</i> . Na przykład, jeśli zmienna <i>time_points</i> zawiera datę, to także ta właściwość powinna być datą. Przykład: 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>	
arima_setting	SD SP	Ustawienie dla algorytmu ARIMA (używane tylko wtedy, gdy <i>algorithm_name</i> ma wartość ARIMA): SD — określone przez system SP — określone przez użytkownika Jeśli <i>arima_setting</i> = SP, użyj następujących parametrów, by ustawić wartości sezonowe i niesezonowe. Przykład (tylko wartości niesezonowe): 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 eq lesseq	ARIMA — operator dla parametrów p, d, q, sp, sd i sq: less — mniejsze niż eq — równe lesseq — mniejsze lub równe
d_symbol		
q_symbol		
sp_symbol		
sd_symbol		
sq_symbol		
p	<i>integer</i>	ARIMA — niesezonowe stopnie autokorelacji.
q	<i>integer</i>	ARIMA — niesezonowa wartość wyliczania.
d	<i>integer</i>	ARIMA — niesezonowa liczba rzędów średniej ruchomej w modelu.
sp	<i>integer</i>	ARIMA — sezonowe stopnie autokorelacji.
sq	<i>integer</i>	ARIMA — sezonowa wartość wyliczania.
sd	<i>integer</i>	ARIMA — sezonowa liczba rzędów średniej ruchomej w modelu.

Tabela 218. Właściwości węzła *netezatimeseriesnode* (kontynuacja)

Właściwości węzła netezatimeseriesnode	Wartości	Opis właściwości
advanced_setting	SD SP	Określa sposób traktowania ustawień zaawansowanych: SD — określone przez system SP — określone przez użytkownika we właściwościach <code>period</code> , <code>units_period</code> i <code>forecast_setting</code> . Przykład: set NZ_DT1.advanced_setting = 'SP' set NZ_DT1.period = 5 set NZ_DT1.units_period = 'd'
period	<i>integer</i>	Długość cyklu sezonowego określona razem z właściwością <code>units_period</code> . Nie dotyczy analizy spektralnej.
units_period	ms s min h d wk q y	Jednostki, w jakich wyrażona jest wartość <code>period</code> : ms — milisekundy s — sekundy min — minuty h — godziny d — dni wk — tygodnie q — kwartały y — lata Na przykład dla tygodniowego szeregu czasowego należy nadać wartość 1 właściwości <code>period</code> i wartość <code>wk</code> właściwości <code>units_period</code> .
forecast_setting	forecasthorizon forecasttimes	Określa sposób prognozowania.
forecast_horizon	<i>integer</i> <i>date</i> <i>time</i> <i>timestamp</i>	Jeśli <code>forecast_setting</code> = <code>forecasthorizon</code> , określa wartość punktu końcowego prognozowania. Obowiązuje format wartości <code>time_points</code> . Na przykład, jeśli zmienna <code>time_points</code> zawiera datę, to także ta właściwość powinna być datą.
forecast_times	<i>integer</i> <i>date</i> <i>time</i> <i>timestamp</i>	Jeśli <code>forecast_setting</code> = <code>forecasttimes</code> , określa wartości, które mają być używane do prognozowania. Obowiązuje format wartości <code>time_points</code> . Na przykład, jeśli zmienna <code>time_points</code> zawiera datę, to także ta właściwość powinna być datą.
include_history	<i>flag</i>	Określa, czy w wynikach mają być uwzględniane wartości historyczne.

Tabela 218. Właściwości węzła *netezzatimeseriesnode* (kontynuacja)

Właściwości węzła <i>netezzatimeseriesnode</i>	Wartości	Opis właściwości
include_interpolated_values	<i>flag</i>	Określa, czy w wynikach mają być uwzględniane wartości interpolowane. Nie ma zastosowania, jeśli include_history ma wartość false.

Netezza Generalized Linear

Dostępne są następujące właściwości węzłów typu *netezzaglmnode*.

Tabela 219. Właściwości węzła *netezzaglmnode*

Właściwości węzła <i>netezzaglmnode</i>	Wartości	Opis właściwości
dist_family	bernoulli gaussian poisson negativebinomial wald gamma	Typ rozkładu; domyślnie bernoulli.
dist_params	<i>number</i>	Wartość parametru rozkładu, która ma być używana. Ma zastosowanie tylko wtedy, gdy distribution ma wartość Negativebinomial.
trials	<i>integer</i>	Ma zastosowanie tylko wtedy, gdy distribution ma wartość Binomial. Gdy przewidywana odpowiedź jest liczbą zdarzeń występujących w zbiorze prób, target zawiera liczbę zdarzeń, a trials zawiera liczbę prób.
model_table	<i>field</i>	Nazwa tabeli bazy danych, w której przechowywany jest uogólniony model liniowy Netezza.
maxit	<i>integer</i>	Maksymalna liczba iteracji, jaką ma wykonać algorytm; domyślnie 20.
eps	<i>number</i>	Wartość błędu maksymalnego (w notacji naukowej), przy której algorytm powinien zatrzymać wyszukiwanie modelu najlepiej dopasowanego. Domyślnie -3, czyli 1E-3 lub 0,001.
tol	<i>number</i>	Wartość (w notacji naukowej), poniżej której błędy są traktowane jako posiadające wartość zero. Domyślnie -7, co oznacza, że wartości błędów poniżej 1E-7 (lub 0,0000001) są traktowane jako nieznaczące.

Tabela 219. Właściwości węzła *netezzaglmnode* (kontynuacja)

Właściwości węzła <i>netezzaglmnode</i>	Wartości	Opis właściwości
link_func	identity inverse invnegative invsquare sqrt power oddspower log clog loglog cloglog logit probit gaussit cauchit canbinom cangeom cannegbinom	Funkcja łączenia, która ma być używana; domyślnie logit.
link_params	<i>number</i>	Wartość parametru funkcji łączenia. Ma zastosowanie tylko wtedy, gdy link_function ma wartość power lub oddspower.
interaction	[[[<i>colnames1</i>],[<i>levels1</i>]], [[<i>colnames2</i>],[<i>levels2</i>]], ...,[[<i>colnamesN</i>],[<i>levelsN</i>]],]	Określa interakcje między zmiennymi. <i>colnames</i> jest listą zmiennych wejściowych, a <i>level</i> dla każdej zmiennej wynosi 0. Przykład: [[["K", "BP", "Sex", "K"], [0, 0, 0, 0]], [["Age", "Na"], [0, 0]]]
intercept	<i>flag</i>	Wartość true powoduje, że w modelu jest uwzględniany wyraz wolny.

Właściwości modelu użytkowego Netezza

Poniższe właściwości są wspólne dla modeli użytkowych w bazie danych Netezza.

Tabela 220. Wspólne właściwości modeli użytkowych Netezza

Wspólne właściwości modeli użytkowych Netezza	Wartości	Opis właściwości
connection	<i>string</i>	Łańcuch połączenia z bazą danych Netezza, w której przechowywany jest model.
table_name	<i>string</i>	Nazwa tabeli bazy danych, w której przechowywany jest model.

Pozostałe właściwości modeli użytkowych są takie same, jak właściwości odpowiednich węzłów modelowania.

Nazwy modeli użytkowych stosowane w skryptach są następujące.

Tabela 221. Nazwy modeli użytkowych Netezza stosowane w skryptach

Model użytkowy	Nazwa skryptu
Drzewo decyzyjne	applynetez zadectreenode
K-średnie	applynetez zakmeansnode

Tabela 221. Nazwy modeli użytkowych Netezza stosowane w skryptach (kontynuacja)

Model użytkowy	Nazwa skryptu
Sieć Bayesa	applynetezزابayesnode
Naive Bayes	applynetezzanaivebayesnode
KNN	applynetezzaknnnode
Grupowanie dzielące	applynetezzaDIVclusternode
PCA	applynetezzapcanode
Drzewo regresji	applynetezzaregtreenode
Regresja liniowa	applynetezzaLINEREGRESSIONnode
Szereg czasowy	applynetezzaTIMESERIESnode
Uogólnione liniowe	applynetezzaglmnode

Rozdział 16. Właściwości węzła wyjściowego

Właściwości węzłów wynikowych nieznacznie różnią się od właściwości pozostałych typów węzłów. Właściwości węzła wynikowego nie odwołują się do konkretnych opcji węzłów, lecz przechowują odwołanie do obiektu wynikowego. Jest to przydatne, gdy pobieramy wartość z tabeli, a następnie ustawiamy ją jako parametr strumienia.

W tej sekcji opisano właściwości węzłów wynikowych dostępne w skryptach.

Właściwości węzła `analysisnode`



Węzeł Analiza ocenia zdolność modeli predykcyjnych do wygenerowania dokładnych predykcji. Węzły analizy przeprowadzają różne porównania pomiędzy wartościami przewidywanymi a rzeczywistymi dla co najmniej jednego modelu użytkowego. Mogą również porównywać modele predykcyjne pomiędzy sobą.

Przykład

```
node = stream.create("analysis", "My node")
# Karta "Analiza"
node.setPropertyValue("coincidence", True)
node.setPropertyValue("performance", True)
node.setPropertyValue("confidence", True)
ode.setPropertyValue("threshold", 75)
node.setPropertyValue("improve_accuracy", 3)
node.setPropertyValue("inc_user_measure", True)
# "Definiuj miarę..."
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"])
# Karta "Wynik"
node.setPropertyValue("output_format", "HTML")
node.setPropertyValue("full_filename", "C:/output/analysis_out.html")
```

Tabela 222. Właściwości węzła `analysisnode`.

Właściwości węzła <code>analysisnode</code>	Typ danych	Opis właściwości
<code>output_mode</code>	Screen File	Służy do określania docelowej lokalizacji wyników generowanych przez węzeł.
<code>use_output_name</code>	<i>flag</i>	Określa, czy używana jest niestandardowa nazwa wyniku.
<code>output_name</code>	<i>string</i>	Jeśli <code>use_output_name</code> ma wartość <code>true</code> , określa nazwę.
<code>output_format</code>	Text (<i>.txt</i>) HTML (<i>.html</i>) Output (<i>.cou</i>)	Służy do określania typu wyniku.
<code>by_fields</code>	<i>list</i>	

Tabela 222. Właściwości węzła *analysisnode* (kontynuacja).

Właściwości węzła <i>analysisnode</i>	Typ danych	Opis właściwości
full_filename	<i>string</i>	W przypadku zapisywania wyników na dysk, w pliku danych lub w formacie HTML, jest to nazwa pliku wyjściowego.
coincidence	<i>flag</i>	
performance	<i>flag</i>	
evaluation_binary	<i>flag</i>	
confidence	<i>flag</i>	
threshold	<i>number</i>	
improve_accuracy	<i>number</i>	
inc_user_measure	<i>flag</i>	
user_if	<i>expr</i>	
user_then	<i>expr</i>	
user_else	<i>expr</i>	
user_compute	[Mean Sum Min Max SDev]	

Właściwości węzła *dataauditnode*



Węzeł Audyt danych umożliwia kompleksowe spojrzenie na dane, udostępniając statystyki podsumowujące, histogramy i rozkład dla każdej zmiennej, jak również informacje o wartościach odstających, brakujących wartościach i wartościach skrajnych. Wyniki są wyświetlane w postaci czytelnej macierzy, która może zostać posortowana i użyta do wygenerowania wykresów w pełnym rozmiarze oraz węzłów przygotowania danych.

Przykład

```

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")
    
```

Tabela 223. Właściwości węzła *dataauditnode*.

Właściwości węzła <i>dataauditnode</i>	Typ danych	Opis właściwości
custom_fields	<i>flag</i>	
fields	[<i>field1 ... fieldN</i>]	
overlay	<i>field</i>	

Tabela 223. Właściwości węzła `dataauditnode` (kontynuacja).

Właściwości węzła <code>dataauditnode</code>	Typ danych	Opis właściwości
<code>display_graphs</code>	<i>flag</i>	Służy do włączania lub wyłączenia wyświetlania wykresów w macierzy wynikowej.
<code>basic_stats</code>	<i>flag</i>	
<code>advanced_stats</code>	<i>flag</i>	
<code>median_stats</code>	<i>flag</i>	
<code>calculate</code>	Count Breakdown	Służy do obliczania braków danych. Można wybrać jedną metodę, obie metody albo nie wybierać żadnej.
<code>outlier_detection_method</code>	std iqr	Służy do określania metody wykrywania wartości odstających i skrajnych.
<code>outlier_detection_std_outlier</code>	<i>number</i>	Jeśli <code>outlier_detection_method</code> ma wartość <code>std</code> , określa liczbę używaną do definiowania wartości odstających.
<code>outlier_detection_std_extreme</code>	<i>number</i>	Jeśli <code>outlier_detection_method</code> ma wartość <code>std</code> , określa liczbę używaną do definiowania wartości skrajnych.
<code>outlier_detection_iqr_outlier</code>	<i>number</i>	Jeśli <code>outlier_detection_method</code> ma wartość <code>iqr</code> , określa liczbę używaną do definiowania wartości odstających.
<code>outlier_detection_iqr_extreme</code>	<i>number</i>	Jeśli <code>outlier_detection_method</code> ma wartość <code>iqr</code> , określa liczbę używaną do definiowania wartości skrajnych.
<code>use_output_name</code>	<i>flag</i>	Określa, czy używana jest niestandardowa nazwa wyniku.
<code>output_name</code>	<i>string</i>	Jeśli <code>use_output_name</code> ma wartość <code>true</code> , określa nazwę.
<code>output_mode</code>	Screen File	Służy do określania docelowej lokalizacji wyników generowanych przez węzeł.
<code>output_format</code>	Formatted (<i>.tab</i>) Delimited (<i>.csv</i>) HTML (<i>.html</i>) Output (<i>.cou</i>)	Służy do określania typu wyniku.
<code>paginate_output</code>	<i>flag</i>	Gdy <code>output_format</code> ma wartość <code>HTML</code> , powoduje dzielenie wyników na strony.
<code>lines_per_page</code>	<i>number</i>	Gdy jest używana razem z właściwością <code>paginate_output</code> , określa liczbę wierszy na stronę wyników.
<code>full_filename</code>	<i>string</i>	

Właściwości węzła extensionoutputnode



Węzeł wynikowy Extension umożliwia analizowanie danych i wyników oceny modelu przy użyciu skryptu R lub Python for Spark udostępnionego przez użytkownika. Wyniki analizy mogą być tekstowe lub graficzne. Wyniki są dodawane do karty **Wynik** na panelu menedżera albo mogą być przekierowane do pliku.

Python for Spark — przykład

```
#### przykład skryptu w języku 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)
```

Przykład w języku R

```
#### przykład skryptu w języku R
node.setPropertyValue("syntax_type", "R")
node.setPropertyValue("r_syntax", "print(modelerData$Age)")
```

Tabela 224. Właściwości węzła extensionoutputnode

Właściwości węzła extensionoutputnode	Typ danych	Opis właściwości
syntax_type	R Python	Określa, który skrypt ma być wykonany – R lub Python (domyślnie R).
r_syntax	string	Komendy skryptu R realizujące ocenę modelu.
python_syntax	string	Komendy skryptu Python realizujące ocenę modelu.
convert_flags	StringsAndDoubles LogicalValues	Opcja przekształcania zmiennych typu flaga.
convert_missing	flag	Opcja przekształcania braków danych na wartości R NA.
convert_datetime	flag	Opcja przekształcania zmiennych o formacie daty lub daty/czasu na formaty daty/czasu języka R.
convert_datetime_class	POSIXct POSIXlt	Opcje określające format docelowy, na który przekształcane są zmienne w formacie daty i daty/czasu.
output_to	Screen File	Określ typ wyjściowy (Screen lub File).

Tabela 224. Właściwości węzła *extensionoutputnode* (kontynuacja)

Właściwości węzła <i>extensionoutputnode</i>	Typ danych	Opis właściwości
output_type	Graph Text	Określ, czy należy generować wyniki graficzne lub tekstowe.
full_filename	<i>string</i>	Nazwa pliku, która ma zostać użyta dla wygenerowanego wyniku.
graph_file_type	HTML COU	Typ pliku dla pliku wyjściowego (.html lub .cou).
text_file_type	HTML TEXT COU	Podaj typ pliku dla wyniku tekstowego (.html, .txt lub .cou).

Właściwości węzła *matrixnode*



Węzeł Macierz tworzy tabelę przedstawiającą relacje pomiędzy zmiennymi. Najczęściej służy do przedstawienia relacji pomiędzy dwoma zmiennymi symbolicznymi, ale również do zaprezentowania relacji pomiędzy zmiennymi flagi lub zmiennymi numerycznymi.

Przykład

```
node = stream.create("matrix", "My node")
# Karta "Ustawienia"
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")
# Karta "Wygląd"
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)
# Karta "Wynik"
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)
```

Tabela 225. Właściwości węzła *matrixnode*.

Właściwości węzła <i>matrixnode</i>	Typ danych	Opis właściwości
fields	Selected Flags Numerics	
row	<i>field</i>	
column	<i>field</i>	
include_missing_values	<i>flag</i>	Określa, czy braki danych użytkownika (wartości puste) oraz systemowe braki danych (null) będą uwzględniane w wynikach kolumnowych i wierszowych.

Tabela 225. Właściwości węzła *matrixnode* (kontynuacja).

Właściwości węzła <i>matrixnode</i>	Typ danych	Opis właściwości
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>	Wartość niezerowa oznacza prawdę.
highlight_bottom	<i>number</i>	Wartość niezerowa oznacza prawdę.
display	[Counts Expected Residuals RowPct ColumnPct TotalPct]	
include_totals	<i>flag</i>	
use_output_name	<i>flag</i>	Określa, czy używana jest niestandardowa nazwa wyniku.
output_name	<i>string</i>	Jeśli <i>use_output_name</i> ma wartość <i>true</i> , określa nazwę.
output_mode	Screen File	Służy do określania docelowej lokalizacji wyników generowanych przez węzeł.
output_format	Formatted (<i>.tab</i>) Delimited (<i>.csv</i>) HTML (<i>.html</i>) Output (<i>.cou</i>)	Służy do określania typu wyniku. Zarówno z formatem Formatted, jak i Delimited można stosować modyfikator <i>transposed</i> , który powoduje transpozycję wierszy i kolumn w tabeli.
paginate_output	<i>flag</i>	Gdy <i>output_format</i> ma wartość HTML, powoduje dzielenie wyników na strony.
lines_per_page	<i>number</i>	Gdy jest używana razem z właściwością <i>paginate_output</i> , określa liczbę wierszy na stronę wyników.
full_filename	<i>string</i>	

właściwości węzła *meansnode*



Węzeł Średnie porównuje średnie między niezależnymi grupami lub między parami powiązanych zmiennych w celu przetestowania, czy istnieje dla nich znaczna różnica. Na przykład, można porównać średnie przychody przed uruchomieniem promocji i po jej zakończeniu lub porównać przychody od klientów, którzy nie otrzymali oferty promocyjnej z tymi, którzy z niej skorzystali.

Przykład

```
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")
```

Tabela 226. Właściwości węzła *meansnode*.

Właściwości węzła <i>meansnode</i>	Typ danych	Opis właściwości
<code>means_mode</code>	BetweenGroups BetweenFields	Określa typ statystyki średnich, która ma być obliczona na danych.
<code>test_fields</code>	[<i>field1</i> ... <i>fieldn</i>]	Określa zmienną testową, gdy <code>means_mode</code> ma wartość <code>BetweenGroups</code> .
<code>grouping_field</code>	<i>field</i>	Określa zmienną grupującą.
<code>paired_fields</code>	[[<i>field1 field2</i>] [<i>field3 field4</i>] ...]	Określa pary zmiennych, które mają być używane, gdy <code>means_mode</code> ma wartość <code>BetweenFields</code> .
<code>label_correlations</code>	<i>flag</i>	Określa, czy wyniki zawierają etykiety korelacji. To ustawienie ma zastosowanie tylko wtedy, gdy <code>means_mode</code> ma wartość <code>BetweenFields</code> .
<code>correlation_mode</code>	Probability Absolute	Określa, czy etykiety korelacji mają zawierać informacje o prawdopodobieństwie, czy wartości bezwzględne.
<code>weak_label</code>	<i>string</i>	
<code>medium_label</code>	<i>string</i>	
<code>strong_label</code>	<i>string</i>	
<code>weak_below_probability</code>	<i>number</i>	Gdy <code>correlation_mode</code> ma wartość <code>Probability</code> , określa wartość odcięcia dla słabych korelacji. Musi to być wartość z przedziału od 0 do 1 — na przykład 0,90.
<code>strong_above_probability</code>	<i>number</i>	Wartość odcięcia dla silnych korelacji.
<code>weak_below_absolute</code>	<i>number</i>	Gdy <code>correlation_mode</code> ma wartość <code>Absolute</code> , określa wartość odcięcia dla słabych korelacji. Musi to być wartość z przedziału od 0 do 1 — na przykład 0,90.
<code>strong_above_absolute</code>	<i>number</i>	Wartość odcięcia dla silnych korelacji.
<code>unimportant_label</code>	<i>string</i>	
<code>marginal_label</code>	<i>string</i>	
<code>important_label</code>	<i>string</i>	
<code>unimportant_below</code>	<i>number</i>	Wartość odcięcia dla niskiej ważności zmiennej. Musi to być wartość z przedziału od 0 do 1 — na przykład 0,90.

Tabela 226. Właściwości węzła *meansnode* (kontynuacja).

Właściwości węzła <i>meansnode</i>	Typ danych	Opis właściwości
<code>important_above</code>	<i>number</i>	
<code>use_output_name</code>	<i>flag</i>	Określa, czy używana jest niestandardowa nazwa wyniku.
<code>output_name</code>	<i>string</i>	Nazwa, która ma być używana.
<code>output_mode</code>	Screen File	Określa docelową lokalizację wyników generowanych przez węzeł.
<code>output_format</code>	Formatted (. <i>tab</i>) Delimited (. <i>csv</i>) HTML (. <i>html</i>) Output (. <i>cou</i>)	Określa typ wyników.
<code>full_filename</code>	<i>string</i>	
<code>output_view</code>	Simple Advanced	Określa, czy wyniki mają być prezentowane w widoku prostym, czy zaawansowanym.

Właściwości węzła *reportnode*



Węzeł Raport tworzy sformatowane raporty zawierające stały tekst oraz dane i inne wyrażenia wydzielone z danych. Można określić format raportu, korzystając z szablonów tekstowych do zdefiniowania konstrukcji tekstu stałego i danych wyjściowych. Możliwe jest zastosowanie niestandardowego formatowania tekstu poprzez wprowadzenie do szablonu znaczników HTML i ustawienie opcji na karcie Wynik. Wartości danych i inne warunkowe wartości wyjściowe można dołączyć do szablonu za pośrednictwem wyrażań CLEM.

Przykład

```
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)
```

Tabela 227. Właściwości węzła *reportnode*.

Właściwości węzła <i>reportnode</i>	Typ danych	Opis właściwości
<code>output_mode</code>	Screen File	Służy do określania docelowej lokalizacji wyników generowanych przez węzeł.
<code>output_format</code>	HTML (. <i>html</i>) Text (. <i>txt</i>) Output (. <i>cou</i>)	Służy do określania typu wyniku.
<code>format</code>	Auto Custom	Pozwala wybrać, czy wynik będzie formatowany automatycznie, czy też za pomocą kodu HTML uwzględnionego w szablonie. Aby użyć formatowania HTML z szablonu, określ wartość Custom.
<code>use_output_name</code>	<i>flag</i>	Określa, czy używana jest niestandardowa nazwa wyniku.

Tabela 227. Właściwości węzła reportnode (kontynuacja).

Właściwości węzła reportnode	Typ danych	Opis właściwości
output_name	string	Jeśli use_output_name ma wartość true, określa nazwę.
text	string	
full_filename	string	
highlights	flag	
title	string	
lines_per_page	number	

Właściwości węzła routputnode



Węzeł wynikowy R umożliwia analizowanie danych i wyników oceny modelu przy użyciu skryptu R udostępnionego przez użytkownika. Wyniki analizy mogą być tekstowe lub graficzne. Wyniki są dodawane do karty **Wynik** na panelu menedżera albo mogą być przekierowane do pliku.

Tabela 228. Właściwości węzła routputnode

Właściwości węzła routputnode	Typ danych	Opis właściwości
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	
text_file_type	HTML TEXT COU	

Właściwości węzła setglobalsnode



Węzeł wartości globalnych skanuje dane i oblicza wartości sumaryczne, które mogą zostać użyte w wyrażeniach CLEM. Na przykład, można użyć tego węzła do obliczenia statystyk dla zmiennej o nazwie *age* (wiek), a następnie użyć ogólnej średniej dla wartości *age* w wyrażeniach CLEM, wstawiając funkcję @GLOBAL_MEAN(*age*).

Przykład

```
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"])
node.setPropertyValue("clear_first", False)
node.setPropertyValue("show_preview", True)
```

Tabela 229. Właściwości węzła *setglobalsnode*.

Właściwości węzła <i>setglobalsnode</i>	Typ danych	Opis właściwości
globals	[Sum Mean Min Max SDev]	Właściwość ustrukturyzowana; odwołania do zmiennych do ustawienia muszą mieć postać: node.setKeyedPropertyValue("globals", "Age", ["Max", "Sum", "Mean", "SDev"])
clear_first	<i>flag</i>	
show_preview	<i>flag</i>	

Właściwości węzła *simevalnode*



Węzeł Ocena symulacji przeprowadza ewaluację określonej predykcyjnej zmiennej przewidywanej i prezentuje informacje na temat rozkładu i korelacji zmiennej przewidywanej.

Tabela 230. Właściwości węzła *simevalnode*.

Właściwości węzła <i>simevalnode</i>	Typ danych	Opis właściwości
target	<i>field</i>	
iteration	<i>field</i>	
presorted_by_iteration	<i>boolean</i>	
max_iterations	<i>number</i>	
tornado_fields	[<i>field1...fieldN</i>]	
plot_pdf	<i>boolean</i>	
plot_cdf	<i>boolean</i>	
show_ref_mean	<i>boolean</i>	
show_ref_median	<i>boolean</i>	
show_ref_sigma	<i>boolean</i>	
num_ref_sigma	<i>number</i>	
show_ref_pct	<i>boolean</i>	
ref_pct_bottom	<i>number</i>	
ref_pct_top	<i>number</i>	
show_ref_custom	<i>boolean</i>	
ref_custom_values	[<i>number1...numberN</i>]	

Tabela 230. Właściwości węzła *simevalnode* (kontynuacja).

Właściwości węzła <i>simevalnode</i>	Typ danych	Opis właściwości
category_values	Category Probabilities Both	
category_groups	Categories Iterations	
create_pct_table	<i>boolean</i>	
pct_table	Quartiles Intervals Custom	
pct_intervals_num	<i>number</i>	
pct_custom_values	[<i>number</i> 1... <i>number</i> N]	

Właściwości węzła *simfitnode*



Węzeł Symulacje Dopasowanie sprawdza rozkład statystyczny danych w każdej zmiennej i generuje (lub aktualizuje) węzeł Symulacje Generowanie, stosując rozkład o najlepszym dopasowaniu przypisany do każdej zmiennej. Węzeł Symulacje Generowanie może być wówczas użyty do wygenerowania danych objętych symulacją.

Tabela 231. Właściwości węzła *simfitnode*.

Właściwości węzła <i>simfitnode</i>	Typ danych	Opis właściwości
build	Node XMLExport Both	
use_source_node_name	<i>boolean</i>	
source_node_name	<i>string</i>	Niestandardowa nazwa węzła źródłowego, który jest generowany lub aktualizowany.
use_cases	All LimitFirstN	
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>	

Właściwości węzła *statisticsnode*



Węzeł Statistics udostępnia informacje podsumowujące na temat zmiennych numerycznych. Oblicza statystyki podsumowujące dla poszczególnych zmiennych oraz korelacje pomiędzy zmiennymi.

Przykład

```
node = stream.create("statistics", "My node")
# Karta "Ustawienia"
node.setPropertyValue("examine", ["Age", "BP", "Drug"])
node.setPropertyValue("statistics", ["mean", "sum", "sdev"])
node.setPropertyValue("correlate", ["BP", "Drug"])
# Sekcja "Etykiety korelacji..."
node.setPropertyValue("label_correlations", True)
node.setPropertyValue("weak_below_absolute", 0.25)
node.setPropertyValue("weak_label", "lower")
node.setPropertyValue("strong_above_absolute", 0.75)
node.setPropertyValue("medium_label", "middle")
node.setPropertyValue("strong_label", "upper")
# Karta "Wynik"
node.setPropertyValue("full_filename", "c:/output/statistics_output.html")
node.setPropertyValue("output_format", "HTML")
```

Tabela 232. Właściwości węzła *statisticsnode*.

Właściwości węzła <i>statisticsnode</i>	Typ danych	Opis właściwości
use_output_name	<i>flag</i>	Określa, czy używana jest niestandardowa nazwa wyniku.
output_name	<i>string</i>	Jeśli use_output_name ma wartość true, określa nazwę.
output_mode	Screen File	Służy do określania docelowej lokalizacji wyników generowanych przez węzeł.
output_format	Text (.txt) HTML (.html) Output (.cou)	Służy do określania typu wyniku.
full_filename	<i>string</i>	
examine	<i>list</i>	
correlate	<i>list</i>	
statistics	[count mean sum min max range variance sdev semean median mode]	
correlation_mode	Probability Absolute	Określa, czy etykiety korelacji mają zawierać informacje o prawdopodobieństwie, czy wartości bezwzględne.
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>	Gdy correlation_mode ma wartość Probability, określa wartość odcięcia dla słabych korelacji. Musi to być wartość z przedziału od 0 do 1 — na przykład 0,90.
strong_above_probability	<i>number</i>	Wartość odcięcia dla silnych korelacji.
weak_below_absolute	<i>number</i>	Gdy correlation_mode ma wartość Absolute, określa wartość odcięcia dla słabych korelacji. Musi to być wartość z przedziału od 0 do 1 — na przykład 0,90.

Tabela 232. Właściwości węzła *statisticsnode* (kontynuacja).

Właściwości węzła <i>statisticsnode</i>	Typ danych	Opis właściwości
<code>strong_above_absolute</code>	<i>number</i>	Wartość odcięcia dla silnych korelacji.

Właściwości węzła *statisticsoutputnode*



Węzeł wynikowy Statistics umożliwia wywołanie procedury IBM SPSS Statistics w celu przeprowadzenia analizy danych IBM SPSS Modeler. Dostępne są różnorodne procedury analityczne programu IBM SPSS Statistics. Ten węzeł wymaga licencjonowanej kopii programu IBM SPSS Statistics.

Właściwości tego węzła są opisane w sekcji “Właściwości węzła *statisticsoutputnode*” na stronie 331.

Właściwości węzła *tablenode*



Węzeł Tabela wyświetla dane w formacie tabeli, którą można również zapisać jako plik. Jest to pomocne, kiedy konieczne jest sprawdzenie wartości danych lub wyeksportowanie ich w czytelnej postaci.

Przykład

```
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)
```

Tabela 233. Właściwości węzła *tablenode*.

Właściwości węzła <i>tablenode</i>	Typ danych	Opis właściwości
<code>full_filename</code>	<i>string</i>	W przypadku zapisywania wyników na dysk, w pliku danych lub w formacie HTML, jest to nazwa pliku wyjściowego.
<code>use_output_name</code>	<i>flag</i>	Określa, czy używana jest niestandardowa nazwa wyniku.
<code>output_name</code>	<i>string</i>	Jeśli <code>use_output_name</code> ma wartość <code>true</code> , określa nazwę.
<code>output_mode</code>	Screen File	Służy do określania docelowej lokalizacji wyników generowanych przez węzeł.
<code>output_format</code>	Formatted (.tab) Delimited (.csv) HTML (.html) Output (.cou)	Służy do określania typu wyniku.
<code>transpose_data</code>	<i>flag</i>	Powoduje transponowanie danych przed ich wyeksportowaniem w taki sposób, że wiersze reprezentują zmienne, a kolumny reprezentują rekordy.

Tabela 233. Właściwości węzła `tablenode` (kontynuacja).

Właściwości węzła <code>tablenode</code>	Typ danych	Opis właściwości
<code>paginate_output</code>	<i>flag</i>	Gdy <code>output_format</code> ma wartość HTML, powoduje dzielenie wyników na strony.
<code>lines_per_page</code>	<i>number</i>	Gdy jest używana razem z właściwością <code>paginate_output</code> , określa liczbę wierszy na stronę wyników.
<code>highlight_expr</code>	<i>string</i>	
<code>output</code>	<i>string</i>	Właściwość przeznaczona tylko do odczytu, która zawiera odwołanie do ostatniej tabeli zbudowanej przez węzeł.
<code>value_labels</code>	<code>[[Value LabelString]</code> <code>[Value LabelString] ...]</code>	Służy do określania etykiet par wartości.
<code>display_places</code>	<i>integer</i>	Określa liczbę miejsc dziesiętnych obowiązującą przy wyświetlaniu zmiennej (dotyczy tylko zmiennych o typie składowania REAL). Wartość <code>-1</code> powoduje użycie ustawienia domyślnego dla strumienia.
<code>export_places</code>	<i>integer</i>	Określa liczbę miejsc dziesiętnych obowiązującą przy eksportowaniu zmiennej (dotyczy tylko zmiennych o typie składowania REAL). Wartość <code>-1</code> powoduje użycie ustawienia domyślnego dla strumienia.
<code>decimal_separator</code>	DEFAULT PERIOD COMMA	Określa separator dziesiętny dla zmiennej (dotyczy tylko zmiennych o typie składowania REAL).
<code>date_format</code>	"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	Określa format daty dla zmiennej (dotyczy tylko zmiennych o typie składowania DATE lub TIMESTAMP).

Tabela 233. Właściwości węzła *tablenode* (kontynuacja).

Właściwości węzła <i>tablenode</i>	Typ danych	Opis właściwości
<code>time_format</code>	"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"	Określa format czasu dla zmiennej (dotyczy tylko zmiennych o typie składowania TIME lub TIMESTAMP).
<code>column_width</code>	<i>integer</i>	Określa szerokość kolumny dla zmiennej. Wartość -1 powoduje ustawienie szerokości kolumny na <i>Auto</i> .
<code>justify</code>	AUTO CENTER LEFT RIGHT	Określa justowanie kolumny dla zmiennej.

Właściwości węzła *transformnode*



Węzeł Transformacja umożliwia wybór i podgląd wyników transformacji przed zastosowaniem ich w wybranych zmiennych.

Przykład

```
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)
```

Tabela 234. Właściwości węzła *transformnode*.

Właściwości węzła <i>transformnode</i>	Typ danych	Opis właściwości
<code>fields</code>	[<i>field1</i> ... <i>fieldn</i>]	Zmienne, które mają być używane w przekształceniu.
<code>formula</code>	All Select	Wskazuje, czy należy obliczyć wszystkie, czy wybrane przekształcenia.
<code>formula_inverse</code>	<i>flag</i>	Wskazuje, czy ma być stosowane przekształcenie odwrotne.
<code>formula_inverse_offset</code>	<i>number</i>	Określa przesunięcie danych używane w formule. Domyślnie 0, chyba że użytkownik określi inną wartość.

Tabela 234. Właściwości węzła transformnode (kontynuacja).

Właściwości węzła transformnode	Typ danych	Opis właściwości
formula_log_n	flag	Wskazuje, czy ma być stosowane przekształcenie \log_n .
formula_log_n_offset	number	
formula_log_10	flag	Wskazuje, czy ma być stosowane przekształcenie \log_{10} .
formula_log_10_offset	number	
formula_exponential	flag	Wskazuje, czy ma być stosowane przekształcenie wykładnicze (e^x).
formula_square_root	flag	Wskazuje, czy ma być stosowane przekształcenie pierwiastkiem kwadratowym.
use_output_name	flag	Określa, czy używana jest niestandardowa nazwa wyniku.
output_name	string	Jeśli use_output_name ma wartość true, określa nazwę.
output_mode	Screen File	Służy do określania docelowej lokalizacji wyników generowanych przez węzeł.
output_format	HTML (.html) Output (.cou)	Służy do określania typu wyniku.
paginate_output	flag	Gdy output_format ma wartość HTML, powoduje dzielenie wyników na strony.
lines_per_page	number	Gdy jest używana razem z właściwością paginate_output, określa liczbę wierszy na stronę wyników.
full_filename	string	Określa nazwę pliku wynikowego.

Rozdział 17. Właściwości węzłów eksportu

Wspólne właściwości węzłów eksportu

Poniższe właściwości są wspólne dla wszystkich węzłów eksportu.

Tabela 235. Wspólne właściwości węzłów eksportu

Właściwość	Wartości	Opis właściwości
publish_path	<i>string</i>	Rdzeń nazwy publikowanych plików obrazów i parametrów.
publish_metadata	<i>flag</i>	Określa, czy generowany będzie plik metadanych opisujący dane wejściowe i wyjściowe obrazu oraz ich modele danych.
publish_use_parameters	<i>flag</i>	Określa, czy plik *.par ma zawierać parametry strumienia.
publish_parameters	<i>string list</i>	Określa parametry do uwzględnienia.
execute_mode	export_data publish	Określa, czy węzeł jest wykonywany bez publikowania strumienia, czy też strumień jest automatycznie publikowany po wykonaniu strumienia.

Właściwości węzła asexport

Funkcja eksportu do programu Analytic Server umożliwia uruchomienie strumienia w rozproszonym systemie plików HDFS (Hadoop Distributed File System).

Przykład

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

Tabela 236. Właściwości węzła asexport.

Właściwości węzła asexport	Typ danych	Opis właściwości
data_source	<i>string</i>	Nazwa źródła danych.
export_mode	<i>string</i>	Określa, czy dołączać (append) eksportowane dane do istniejącego źródła danych, czy nadpisywać (overwrite) dane w istniejącym źródle.
use_default_as	<i>boolean</i>	Ustawienie wartości True powoduje użycie domyślnego połączenia Analytic Server skonfigurowanego w pliku serwera options.cfg. Ustawienie wartości False powoduje, że używane jest połączenie tego węzła.

Tabela 236. Właściwości węzła *asexport* (kontynuacja).

Właściwości węzła asexport	Typ danych	Opis właściwości
connection	["string", "string", "string", "string", "string", "string", "string", "string", "string"]	<p>Właściwość zawierająca listę informacji o szczegółach połączenia Analytic Server. Format jest następujący:</p> <pre>["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"]</pre> <p>Gdzie: <code>is_secure_connect</code>: oznacza, czy użyto połączenia bezpiecznego, i może mieć wartość <code>true</code> lub <code>false</code>. <code>use-kerberos-auth</code>: wskazuje, czy zastosowanie ma uwierzytelnianie kerberos, i może mieć wartość <code>true</code> lub <code>false</code>. <code>enable-kerberos-debug</code>: wskazuje, czy dla uwierzytelniania kerberos użyto trybu debugowania, i może mieć wartość <code>true</code> lub <code>false</code>.</p>

Właściwości węzła **cognosexportnode**



Węzeł eksportu IBM Cognos eksportuje dane w formacie możliwym do odczytania przez bazy danych Cognos.

Dla tego węzła należy zdefiniować połączenie z serwerem Cognos i połączenie ODBC.

Połączenie Cognos

Właściwości połączenia z serwerem Cognos opisano poniżej.

Tabela 237. Właściwości węzła *cognosexportnode*

Właściwości węzła <i>cognosexportnode</i>	Typ danych	Opis właściwości
<i>cognos_connection</i>	<i>["string", "flag", "string", "string", "string"]</i>	<p>Właściwość zawierająca listę informacji o połączeniu z serwerem Cognos. Obowiązuje format: ["namespace", "tm1_username", "password"]</p> <p>gdzie: <i>Cognos_server_URL</i> to adres URL serwera Cognos zawierającego źródło. <i>login_mode</i> określa, czy jest używane logowanie anonimowe, i ma wartość <i>true</i> lub <i>false</i>; jeśli ta właściwość jest ustawiona na <i>true</i>, to wymienione poniżej pola należy ustawić na "". <i>namespace</i> określa dostawcę uwierzytelniania używanego do logowania się na serwerze. <i>username</i> i <i>password</i> to nazwa użytkownika i hasło używane do logowania na serwerze Cognos.</p> <p>Zamiast <i>login_mode</i> dostępne są także następujące tryby:</p> <ul style="list-style-type: none"> <i>anonymousMode</i>. Na przykład: ['Cognos_server_url', 'anonymousMode', "namespace", "username", "password"] <i>credentialMode</i>. Na przykład: ['Cognos_server_url', 'credentialMode', "namespace", "username", "password"] <i>storedCredentialMode</i>. Na przykład: ['Cognos_server_url', 'storedCredentialMode', "stored_credential_name"] <p>Gdzie <i>stored_credential_name</i> jest nazwą danych uwierzytelniających Cognos w repozytorium.</p>
<i>cognos_package_name</i>	<i>string</i>	Ścieżka i nazwa pakietu Cognos, do którego mają być eksportowane dane, na przykład: /Public Folders/MyPackage
<i>cognos_datasource</i>	<i>string</i>	
<i>cognos_export_mode</i>	Publish ExportFile	
<i>cognos_filename</i>	<i>string</i>	

Połączenie ODBC

Właściwości połączenia ODBC są identyczne z wymienionymi w opisie węzła *databaseexportnode* w następnej sesji, z tym że właściwość *datasource* nie jest dozwolona.

Właściwości węzła databaseexportnode



Węzeł eksportu do bazy danych zapisuje dane w źródle danych zgodnym z interfejsem ODBC. Aby możliwy był zapis w źródle danych ODBC, to źródło danych musi istnieć, a użytkownik musi mieć w nim uprawnienia do zapisu.

Przykład

```
...
Przyjęto, że skonfigurowane jest źródło danych o nazwie "MyDatasource"
...
stream = modeler.script.stream()
db_exportnode = stream.createAt("databaseexport", "DB Export", 200, 200)
applynn = stream.findByType("applyneuralnetwork", None)
stream.link(applynn, db_exportnode)

# Karta Eksport
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)

# Okno dialogowe Schemat
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")

# Okno dialogowe Indeksy
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"]}]])
```

Tabela 238. Właściwości węzła databaseexportnode.

Właściwości węzła databaseexportnode	Typ danych	Opis właściwości
datasource	string	
username	string	
password	string	
epassword	string	Ten parametr zagnieżdżony w czasie wykonywania jest dostępny tylko do odczytu. Do generowania haseł zakodowanych służy narzędzie Kodowanie hasła dostępne w menu Narzędzia. Więcej informacji można znaleźć w temacie "Generowanie hasła kodowanego" na stronie 52.
table_name	string	

Tabela 238. Właściwości węzła `databaseexportnode` (kontynuacja).

Właściwości węzła <code>databaseexportnode</code>	Typ danych	Opis właściwości
<code>write_mode</code>	Create Append Merge	
<code>map</code>	<i>string</i>	<p>Odwzorowuje nazwę zmiennej w strumieniu na nazwę kolumny w bazie danych (tylko jeśli <code>write_mode</code> to Merge).</p> <p>W przypadku opcji Merge wszystkie zmienne muszą być odwzorowane, aby zostały wyeksportowane. Nazwy zmiennych nieistniejących w bazie danych są dodawane jako nowe kolumny.</p>
<code>key_fields</code>	<i>list</i>	Określa zmienną strumienia używaną jako klucz; właściwość <code>map</code> określa, któremu elementowi w bazie danych ona odpowiada.
<code>join</code>	Database Add	
<code>drop_existing_table</code>	<i>flag</i>	
<code>delete_existing_rows</code>	<i>flag</i>	
<code>default_string_size</code>	<i>integer</i>	
<code>type</code>		Właściwość ustrukturyzowana używana do określania typu schematu.
<code>generate_import</code>	<i>flag</i>	
<code>use_custom_create_table_command</code>	<i>flag</i>	Parametr <code>custom_create_table</code> służy do modyfikowania standardowej komendy SQL CREATE TABLE.
<code>custom_create_table_command</code>	<i>string</i>	Określa komendę (w formie łańcucha), która ma być używana zamiast standardowej komendy SQL CREATE TABLE.
<code>use_batch</code>	<i>flag</i>	Następujące właściwości są opcjami zaawansowanymi masowego ładowania z bazy danych. Wartość True właściwości <code>Use_batch</code> powoduje wyłączenie zatwierdzania wiersz po wierszu przy zapisie w bazie danych.
<code>batch_size</code>	<i>number</i>	Określa liczbę rekordów, która ma zostać wysłana do bazy danych przed zatwierdzeniem w pamięci.
<code>bulk_loading</code>	Off ODBC External	Określa typ ładowania masowego. Dodatkowe opcje dla wartości ODBC i External wymieniono poniżej.
<code>not_logged</code>	<i>flag</i>	
<code>odbc_binding</code>	Row Kolumna	Określa wiązanie wierszowe lub kolumnowe przy ładowaniu masowym przez ODBC.

Tabela 238. Właściwości węzła `databaseexportnode` (kontynuacja).

Właściwości węzła <code>databaseexportnode</code>	Typ danych	Opis właściwości
<code>loader_delimit_mode</code>	Tab Space Other	W przypadku ładowania masowego przez program zewnętrzny należy określić separator. Wybierz <code>Other</code> razem z właściwością <code>loader_other_delimiter</code> , aby określić separator, na przykład przecinek (,).
<code>loader_other_delimiter</code>	<i>string</i>	
<code>specify_data_file</code>	<i>flag</i>	Flaga <code>True</code> aktywuje opisaną niżej właściwość <code>data_file</code> , która umożliwia określenie nazwy pliku i ścieżki zapisu przy ładowaniu masowym do bazy danych.
<code>data_file</code>	<i>string</i>	
<code>specify_loader_program</code>	<i>flag</i>	Flaga <code>True</code> aktywuje opisaną niżej właściwość <code>loader_program</code> , która umożliwia określenie nazwy i lokalizacji zewnętrznego skryptu lub programu ładującego.
<code>loader_program</code>	<i>string</i>	
<code>gen_logfile</code>	<i>flag</i>	Flaga <code>True</code> aktywuje opisaną poniżej właściwość <code>logfile_name</code> , która umożliwia określenie nazwy pliku na serwerze, w którym ma być generowany dziennik błędów.
<code>logfile_name</code>	<i>string</i>	
<code>check_table_size</code>	<i>flag</i>	Flaga <code>True</code> umożliwia sprawdzanie, czy wzrost wielkości tabeli w bazie danych odpowiada liczbie wierszy wyeksportowanych z programu IBM SPSS Modeler.
<code>loader_options</code>	<i>string</i>	Określa dodatkowe argumenty wywołania programu ładującego, takie jak <code>-comment</code> i <code>-specialdir</code> .
<code>export_db_primarykey</code>	<i>flag</i>	Określa, czy dana zmienna jest kluczem podstawowym.
<code>use_custom_create_index_command</code>	<i>flag</i>	Jeśli <code>true</code> , dla wszystkich indeksów włączona jest obsługa niestandardowego kodu SQL.
<code>custom_create_index_command</code>	<i>string</i>	Określa komendę SQL używaną do tworzenia indeksów, gdy włączona jest obsługa niestandardowego kodu SQL. (Tę wartość można przesłonić dla konkretnych indeksów, tak jak opisano to poniżej).
<code>indexes.INDEXNAME.fields</code>		W razie potrzeby tworzy określony indeks i generuje listę nazw zmiennych, które mają być w nim uwzględnione.

Tabela 238. Właściwości węzła *databaseexportnode* (kontynuacja).

Właściwości węzła <i>databaseexportnode</i>	Typ danych	Opis właściwości
INDEXNAME "use_custom_create_index_command"	<i>flag</i>	Służy do włączania lub wyłączania obsługi niestandardowego kodu SQL dla konkretnego indeksu. Przykłady zamieszczono poniżej następującej tabeli.
INDEXNAME "custom_create_index_command"	<i>string</i>	Określa niestandardowy kod SQL dla określonego indeksu. Przykłady zamieszczono poniżej następującej tabeli.
indexes.INDEXNAME.remove	<i>flag</i>	Wartość True powoduje usunięcie określonego indeksu z zestawu indeksów.
table_space	<i>string</i>	Określa obszar tabel, który zostanie utworzony.
use_partition	<i>flag</i>	Powoduje użycie rozproszonej zmiennej mieszającej.
partition_field	<i>string</i>	Określa zawartość rozproszonej zmiennej mieszającej.

Uwaga: W niektórych bazach danych można nakazać tworzenie tabel do eksportu z kompresją (np. równoważnik komendy CREATE TABLE MYTABLE (...) COMPRESS YES; w SQL). Właściwości *use_compression* i *compression_mode* umożliwiają korzystanie z tej funkcji w sposób opisany poniżej.

Tabela 239. Właściwości węzła *databaseexportnode* związane z funkcją kompresji.

Właściwości węzła <i>databaseexportnode</i>	Typ danych	Opis właściwości
use_compression	<i>Boolean</i>	Wartość True powoduje tworzenie tabel do eksportu z kompresją.
compression_mode	Row Page	Określa poziom kompresji dla baz danych na serwerze SQL Server.
	Default Direct_Load_Operations All_Operations Basic OLTP Query_High Query_Low Archive_High Archive_Low	Określa poziom kompresji dla baz danych na serwerze Oracle. Wartości OLTP, Query_High, Query_Low, Archive_High i Archive_Low można używać na serwerach Oracle w wersji nie niższej niż 11gR2.

Przykład modyfikacji komendy CREATE INDEX dla konkretnego indeksu:

```
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)>"])
```

Operację tę można też wykonać za pomocą tabeli mieszającej:

```
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)>"}])
```

Właściwości węzła datacollectionexportnode



Dane wyjściowe węzła eksportu Data Collection są w formacie używanym przez oprogramowanie do badań rynku Data Collection. Aby można było korzystać z tego węzła, musi być zainstalowana biblioteka Data Collection Data Library.

Przykład

```
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)
```

Tabela 240. Właściwości węzła datacollectionexportnode

Właściwości węzła datacollectionexportnode	Typ danych	Opis właściwości
metadata_file	string	Nazwa pliku metadanych do wyeksportowania.
merge_metadata	Overwrite MergeCurrent	
enable_system_variables	flag	Określa, czy wyeksportowany plik .mdd powinien zawierać zmienne systemowe Data Collection.
casedata_file	string	Nazwa pliku .sav, do którego mają być eksportowane dane obserwacji.
generate_import	flag	

Właściwości węzła excelexportnode



Węzeł eksportu Plik Excel powoduje zapisanie danych wyjściowych w pliku Microsoft Excel w formacie .xlsx. Opcjonalnie można zdecydować, aby program Excel był uruchamiany automatycznie i otwierał wyeksportowany plik po wykonaniu węzła.

Przykład

```
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)
```

Tabela 241. Właściwości węzła excelexportnode

Właściwości węzła excelexportnode	Typ danych	Opis właściwości
full_filename	string	
excel_file_type	Excel2007	

Tabela 241. Właściwości węzła *excelexportnode* (kontynuacja)

Właściwości węzła <i>excelexportnode</i>	Typ danych	Opis właściwości
<code>export_mode</code>	Create Append	
<code>inc_field_names</code>	<i>flag</i>	Określa, czy pierwszy wiersz arkusza ma zawierać nazwy zmiennych.
<code>start_cell</code>	<i>string</i>	Określa początkową komórkę do wyeksportowania.
<code>worksheet_name</code>	<i>string</i>	Nazwa arkusza do zapisania.
<code>launch_application</code>	<i>flag</i>	Określa, czy od razu należy otworzyć wynikowy plik w programie Excel. Należy zwrócić uwagę, że ścieżka do programu Excel musi być określona w oknie dialogowym Aplikacje pomocnicze (menu Narzędzia, Aplikacje pomocnicze).
<code>generate_import</code>	<i>flag</i>	Określa, czy ma zostać wygenerowany węzeł importu z programu Excel, który odczyta wyeksportowany plik danych.

Właściwości węzła *extensionexportnode*



Z węzłem wynikowym eksportu Extension można uruchomić skrypty R lub Python for Spark w celu eksportu danych.

Python for Spark — przykład

```
#### przykład skryptu w języku 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)
```

Przykład w języku R

```
#### przykład skryptu w języku R
node.setPropertyValue("syntax_type", "R")
node.setPropertyValue("r_syntax", ""write.csv(modelerData, "C:/export.csv")"")
```

Tabela 242. Właściwości węzła *extensionexportnode*

Właściwości węzła <i>extensionexportnode</i>	Typ danych	Opis właściwości
<code>syntax_type</code>	<i>R</i> <i>Python</i>	Określa, który skrypt ma być wykonany – R lub Python (domyślnie R).
<code>r_syntax</code>	<i>string</i>	Komendy skryptu R do uruchomienia.
<code>python_syntax</code>	<i>string</i>	Komendy skryptu Python do uruchomienia.
<code>convert_flags</code>	StringsAndDoubles LogicalValues	Opcja przekształcania zmiennych typu flaga.
<code>convert_missing</code>	<i>flag</i>	Opcja przekształcania braków danych na wartości R NA.
<code>convert_datetime</code>	<i>flag</i>	Opcja przekształcania zmiennych o formacie daty lub daty/czasu na formaty daty/czasu języka R.
<code>convert_datetime_class</code>	POSIXct POSIXlt	Opcje określające format docelowy, na który przekształcane są zmienne w formacie daty i daty/czasu.

Właściwości węzła *outputfilenode*



Dane wyjściowe wykresu eksportu Plik płaski zapisywane są w pliku tekstowym z danymi rozgraniczonymi. Jest to przydatne podczas eksportowania danych, które mogą być odczytywane przez inne oprogramowanie do przeprowadzania analizy lub obsługujące arkusz kalkulacyjny.

Przykład

```
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("delimiter_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)
```

Tabela 243. Właściwości węzła *outputfilenode*

Właściwości węzła <i>outputfilenode</i>	Typ danych	Opis właściwości
<code>full_filename</code>	<i>string</i>	Nazwa pliku wyjściowego.
<code>write_mode</code>	Overwrite Append	
<code>inc_field_names</code>	<i>flag</i>	

Tabela 243. Właściwości węzła *outputfilenode* (kontynuacja)

Właściwości węzła <i>outputfilenode</i>	Typ danych	Opis właściwości
<code>use_newline_after_records</code>	<i>flag</i>	
<code>delimit_mode</code>	Comma Tab Space Other	
<code>other_delimiter</code>	<i>char</i>	
<code>quote_mode</code>	None Single Double Other	
<code>other_quote</code>	<i>flag</i>	
<code>generate_import</code>	<i>flag</i>	
<code>encoding</code>	StreamDefault SystemDefault "UTF-8"	

Właściwości węzła *sasexportnode*



Dane wyjściowe węzła eksportu SAS są zapisywane w formacie SAS, aby mogły zostać odczytane w systemie SAS lub za pomocą pakietu oprogramowania kompatybilnego z systemem SAS. Dostępne są trzy pliki w formacie SAS: SAS dla Windows/OS2, SAS dla UNIX lub SAS wersja 7/8.

Przykład

```
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)
```

Tabela 244. Właściwości węzła *sasexportnode*

Właściwości węzła <i>sasexportnode</i>	Typ danych	Opis właściwości
<code>format</code>	Windows UNIX SAS7 SAS8	Zmienne etykiet właściwości wariantu.
<code>full_filename</code>	<i>string</i>	
<code>export_names</code>	NamesAndLabels NamesAsLabels	Służą do odwzorowywania nazw zmiennych z programu IBM SPSS Modeler po eksporcie na zmienne programu IBM SPSS Statistics lub SAS.
<code>generate_import</code>	<i>flag</i>	

Właściwości węzła statisticsexportnode



Dane wynikowe węzła eksportu Statistics zapisywane są w formacie IBM SPSS Statistics: *.sav* lub *.zsav*. Pliki *.sav* lub *.zsav* mogą być odczytywane przez produkty IBM SPSS Statistics Base i inne. Jest to również format używany przez pliki pamięci podręcznej w programie IBM SPSS Modeler.

Właściwości tego węzła są opisane w sekcji “Właściwości węzła statisticsexportnode” na stronie 331.

Właściwości węzła tm1odataexport



Węzeł eksportu IBM Cognos TM1 eksportuje dane w formacie możliwym do odczytania przez bazy danych Cognos TM1.

Tabela 245. Właściwości węzła tm1odataexport

Właściwości węzła tm1odataexport	Typ danych	Opis właściwości
admin_host	<i>string</i>	Adres URL dla nazwy hosta interfejsu API REST.
server_name	<i>string</i>	Nazwa serwera TM1 wybranego z <code>admin_host</code> .
credential_type	<code>inputCredential</code> lub <code>storedCredential</code>	Służy do wskazywania typu informacji autoryzacyjnych.
input_credential	<i>list</i>	Jeśli parametr <code>credential_type</code> ma wartość <code>inputCredential</code> , należy podać nazwę domeny, nazwę użytkownika i hasło.
stored_credential_name	<i>string</i>	Jeśli parametr <code>credential_type</code> ma wartość <code>storedCredential</code> , należy podać nazwę referencji na serwerze C&DS.
selected_cube	<i>field</i>	Nazwa kostki, do której mają być eksportowane dane. Na przykład: <code>TM1_export.setPropertyValue("selected_cube", "plan_BudgetPlan")</code>

Tabela 245. Właściwości węzła *tm1odataexport* (kontynuacja)

Właściwości węzła <i>tm1odataexport</i>	Typ danych	Opis właściwości
<i>spss_field_to_tm1_element_mapping</i>	<i>list</i>	<p>Element <i>tm1</i>, do którego ma nastąpić odwzorowanie, musi być częścią wymiaru kolumny dla wybranego widoku kostki. Format wygląda następująco: [[[Field_1, Dimension_1, False], [Element_1, Dimension_2, True], ...], [[Field_2, ExistMeasureElement, False], [Field_3, NewMeasureElement, True], ...]]</p> <p>Istnieją dwie listy do opisywania informacji o mapowaniu. Mapowanie elementu liścia do wymiaru odpowiada drugiemu z poniższych przykładów: Przykład 1: Pierwsza lista: ([[Field_1, Dimension_1, False], [Element_1, Dimension_2, True], ...]) służy do wskazywania informacji o mapowaniu wymiarów TM1.</p> <p>Każda lista z 3 wartościami wskazuje informacje o mapowaniu wymiarów. Trzecia wartość boolowska służy do wskazywania, czy wybiera element wymiaru. Na przykład: "[Field_1, Dimension_1, False]" oznacza, że <i>Field_1</i> zmapowano na <i>Dimension_1</i>; "[Element_1, Dimension_2, True]" oznacza, że <i>Element_1</i> wybrano dla <i>Dimension_2</i>.</p> <p>Przykład 2: Druga lista: ([[Field_2, ExistMeasureElement, False], [Field_3, NewMeasureElement, True], ...]) służy do wskazywania informacji o mapowaniu elementów wymiarów miar TM1.</p> <p>Każda lista z 3 wartościami wskazuje informacje o mapowaniu miar. Trzecia wartość boolowska służy do wskazywania konieczności utworzenia nowego elementu. "[Field_2, ExistMeasureElement, False]" oznacza, że <i>Field_2</i> zmapowano na <i>ExistMeasureElement</i>; "[Field_3, NewMeasureElement, True]" oznacza, że <i>NewMeasureElement</i> musi być wymiarem miary wybranym w <i>selected_measure</i>, a <i>Field_3</i> na niego zmapowano.</p>
<i>selected_measure</i>	<i>string</i>	<p>Określ wymiar miary.</p> <p>Przykład: <code>setProperty("selected_measure", "Measures")</code></p>

Właściwości węzła *tm1export* (zdeprecjonowanego)



Węzeł eksportu IBM Cognos TM1 eksportuje dane w formacie możliwym do odczytania przez bazy danych Cognos TM1.

Uwaga: Ten węzeł został zdeprecjonowany w produkcie Modeler 18.0. Nazwa skryptu węzła zamiany to *tm1odataexport*.

Tabela 246. Właściwości węzła tm1export.

Właściwości węzła tm1export	Typ danych	Opis właściwości
pm_host	string	<p>Uwaga: Tylko dla wersji 16.0 i 17.0</p> <p>Nazwa hosta. Na przykład: <code>TM1_export.setPropertyValue("pm_host", 'http://9.191.86.82:9510/pmhub/pm')</code></p>
tm1_connection	[<i>"field"</i> , <i>"field"</i> , ... <i>"field"</i>]	<p>Uwaga: Tylko dla wersji 16.0 i 17.0</p> <p>Właściwość zawierająca listę informacji o połączeniu z serwerem TM1. Obowiązuje format: [<code>"TM1_Server_Name", "tm1_username", "tm1_password"</code>]</p> <p>Na przykład: <code>TM1_export.setPropertyValue("tm1_connection", ['Planning Sample', "admin" "apple"])</code></p>
selected_cube	field	<p>Nazwa kostki, do której mają być eksportowane dane. Na przykład: <code>TM1_export.setPropertyValue("selected_cube", "plan_BudgetPlan")</code></p>
spssfield_tm1element_mapping	list	<p>Element tm1, do którego ma nastąpić odwzorowanie, musi być częścią wymiaru kolumny dla wybranego widoku kostki. Format wygląda następująco: [[[Field_1, Dimension_1, False], [Element_1, Dimension_2, True], ...], [[Field_2, ExistMeasureElement, False], [Field_3, NewMeasureElement, True], ...]]</p> <p>Istnieją dwie listy do opisywania informacji o mapowaniu. Mapowanie elementu liścia do wymiaru odpowiada drugiemu z poniższych przykładów: Przykład 1: Pierwsza lista: ([[Field_1, Dimension_1, False], [Element_1, Dimension_2, True], ...]) służy do wskazywania informacji o mapowaniu wymiarów TM1.</p> <p>Każda lista z 3 wartościami wskazuje informacje o mapowaniu wymiarów. Trzecia wartość boolowska służy do wskazywania, czy wybiera element wymiaru. Na przykład: "[Field_1, Dimension_1, False]" oznacza, że Field_1 zmapowano na Dimension_1; "[Element_1, Dimension_2, True]" oznacza, że Element_1 wybrano dla Dimension_2.</p> <p>Przykład 2: Druga lista: ([[Field_2, ExistMeasureElement, False], [Field_3, NewMeasureElement, True], ...]) służy do wskazywania informacji o mapowaniu elementów wymiarów miar TM1.</p> <p>Każda lista z 3 wartościami wskazuje informacje o mapowaniu miar. Trzecia wartość boolowska służy do wskazywania konieczności utworzenia nowego elementu. "[Field_2, ExistMeasureElement, False]" oznacza, że Field_2 zmapowano na ExistMeasureElement; "[Field_3, NewMeasureElement, True]" oznacza, że NewMeasureElement musi być wymiarem miary wybranym w selected_measure, a Field_3 na niego zmapowano.</p>

Tabela 246. Właściwości węzła `tm1export` (kontynuacja).

Właściwości węzła <code>tm1export</code>	Typ danych	Opis właściwości
<code>selected_measure</code>	<i>string</i>	Określ wymiar miary. Przykład: <code>setProperty("selected_measure", "Measures")</code>

Właściwości węzła `xmlexportnode`



Węzeł eksportu XML zapisuje dane w pliku w formacie XML. Opcjonalnie można utworzyć węzeł źródłowy XML, który odczyta wyeksportowane dane z powrotem do strumienia.

Przykład

```
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"]])
```

Tabela 247. Właściwości węzła `xmlexportnode`

Właściwości węzła <code>xmlexportnode</code>	Typ danych	Opis właściwości
<code>full_filename</code>	<i>string</i>	(Wymagana) Pełna ścieżka i nazwa docelowego pliku XML.
<code>use_xml_schema</code>	<i>flag</i>	Określa, czy do sterowania strukturą eksportowanych danych ma być używany schemat XML (plik XSD lub DTD).
<code>full_schema_filename</code>	<i>string</i>	Pełna ścieżka i nazwa pliku XSD lub DTD, który ma być używany. Wymagana, jeśli <code>use_xml_schema</code> ma wartość <code>true</code> .
<code>generate_import</code>	<i>flag</i>	Powoduje wygenerowanie źródłowego węzła XML, który odczyta wyeksportowane dane z powrotem do strumienia.
<code>records</code>	<i>string</i>	Wyrażenie XPath opisujące granicę rekordu.
<code>map</code>	<i>string</i>	Odwzorowuje nazwę zmiennej na strukturę XML.

Rozdział 18. Właściwości węzła IBM SPSS Statistics

Właściwości węzła statisticsimportnode



Węzeł Plik Statistics odczytuje dane z pliku w formacie *.sav* lub *.zsav* używanym przez program IBM SPSS Statistics, jak również pliki pamięci podręcznej zapisane w programie IBM SPSS Modeler, które również używają tego samego formatu.

Przykład

```
stream = modeler.script.stream()
statisticsimportnode = stream.createAt("statisticsimport", "SAV Import", 200, 200)
statisticsimportnode.setPropertyValue("full_filename", "C:/data/drug1n.sav")
statisticsimportnode.setPropertyValue("import_names", True)
statisticsimportnode.setPropertyValue("import_data", True)
```

Tabela 248. Właściwości węzła statisticsimportnode.

Właściwości węzła statisticsimportnode	Typ danych	Opis właściwości
full_filename	string	Pełna nazwa pliku wraz ze ścieżką.
password	string	Hasło. Parametrowi password należy nadać wartość przed nadaniem wartości parametrowi file_encrypted.
file_encrypted	flag	Określa, czy plik jest zabezpieczony hasłem.
import_names	NamesAndLabels LabelsAsNames	Metoda postępowania z nazwami i etykietami zmiennych.
import_data	DataAndLabels LabelsAsData	Metoda postępowania z wartościami i etykietami.
use_field_format_for_storage	Boolean	Określa, czy przy imporcie używać informacji o formacie zmiennych z programu IBM SPSS Statistics.

Właściwości węzła statisticstransformnode



Węzeł przekształceń Statistics uruchamia wybór komend składni IBM SPSS Statistics dla źródeł danych w programie IBM SPSS Modeler. Ten węzeł wymaga licencjonowanej kopii programu IBM SPSS Statistics.

Przykład

```
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)
```

Tabela 249. Właściwości węzła *statisticstransformnode*

Właściwości węzła <i>statisticstransformnode</i>	Typ danych	Opis właściwości
<code>syntax</code>	<i>string</i>	
<code>check_before_saving</code>	<i>flag</i>	Powoduje sprawdzanie poprawności składni wprowadzonych komend przed zapisaniem wpisów. Jeśli składnia jest niepoprawna, wyświetlany jest komunikat o błędzie.
<code>default_include</code>	<i>flag</i>	Więcej informacji można znaleźć w temacie “Właściwości węzła <i>filternode</i> ” na stronie 136.
<code>include</code>	<i>flag</i>	Więcej informacji można znaleźć w temacie “Właściwości węzła <i>filternode</i> ” na stronie 136.
<code>new_name</code>	<i>string</i>	Więcej informacji można znaleźć w temacie “Właściwości węzła <i>filternode</i> ” na stronie 136.

właściwości węzła *statisticsmodelnode*



Węzeł Model Statistics umożliwia analizowanie danych i pracę z nimi poprzez uruchomienie procedur IBM SPSS Statistics tworzących PMML. Ten węzeł wymaga licencjonowanej kopii programu IBM SPSS Statistics.

Przykład

```
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")
```

Właściwości węzła <i>statisticsmodelnode</i>	Typ danych	Opis właściwości
<code>syntax</code>	<i>string</i>	
<code>default_include</code>	<i>flag</i>	Więcej informacji można znaleźć w temacie “Właściwości węzła <i>filternode</i> ” na stronie 136.
<code>include</code>	<i>flag</i>	Więcej informacji można znaleźć w temacie “Właściwości węzła <i>filternode</i> ” na stronie 136.
<code>new_name</code>	<i>string</i>	Więcej informacji można znaleźć w temacie “Właściwości węzła <i>filternode</i> ” na stronie 136.

Właściwości węzła statisticsoutputnode



Węzeł wynikowy Statistics umożliwia wywołanie procedury IBM SPSS Statistics w celu przeprowadzenia analizy danych IBM SPSS Modeler. Dostępne są różnorodne procedury analityczne programu IBM SPSS Statistics. Ten węzeł wymaga licencjonowanej kopii programu IBM SPSS Statistics.

Przykład

```
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")
```

Tabela 250. Właściwości węzła statisticsoutputnode

Właściwości węzła statisticsoutputnode	Typ danych	Opis właściwości
mode	Dialog Syntax	Wybiera opcję "Okno dialogowe IBM SPSS Statistics" albo Edytor komend
syntax	string	
use_output_name	flag	
output_name	string	
output_mode	Screen File	
full_filename	string	
file_type	HTML SPV SPW	

Właściwości węzła statisticsexportnode



Dane wynikowe węzła eksportu Statistics zapisywane są w formacie IBM SPSS Statistics: *.sav* lub *.zsav*. Pliki *.sav* lub *.zsav* mogą być odczytywane przez produkty IBM SPSS Statistics Base i inne. Jest to również format używany przez pliki pamięci podręcznej w programie IBM SPSS Modeler.

Przykład

```
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)
```

Tabela 251. Właściwości węzła *statisticsexportnode*.

Właściwości węzła statisticsexportnode	Typ danych	Opis właściwości
full_filename	<i>string</i>	
file_type	sav zsav	Zapisywanie pliku w formacie <i>sav</i> albo <i>zsav</i> . Na przykład: statisticsexportnode.setPropertyValue("file_type","sav")
encrypt_file	<i>flag</i>	Określa, czy plik jest zabezpieczony hasłem.
password	<i>string</i>	Hasło.
launch_application	<i>flag</i>	
export_names	NamesAndLabels NamesAsLabels	Służą do odwzorowywania nazw zmiennych z programu IBM SPSS Modeler po eksporcie na zmienne programu IBM SPSS Statistics lub SAS.
generate_import	<i>flag</i>	

Rozdział 19. Właściwości węzła Python

Właściwości smotenode



Węzeł SMOTE (Synthetic Minority Over-sampling Technique — generowanie próbek syntetycznych z klasy mniejszościowej) realizuje algorytm nadpróbkowania przydatny w pracy z nierównoważonymi zbiorami danych. Udostępnia on zaawansowaną metodę równoważenia danych. Węzeł procesowy SMOTE w programie SPSS Modeler jest zaimplementowany w języku Python i wymaga biblioteki Python `imbalanced-learn`©.

Tabela 252. Właściwości smotenode

Właściwości smotenode	Typ danych	Opis właściwości
<code>target_field</code>	<i>field</i>	Zmienna przewidywana.
<code>sample_ratio</code>	<i>string</i>	Aktywuje niestandardową wartość współczynnika. Dostępne są dwie opcje: Automatyczny (<code>sample_ratio_auto</code>) lub Ustaw współczynnik (<code>sample_ratio_manual</code>).
<code>sample_ratio_value</code>	<i>float</i>	Współczynnik oznacza liczbę próbek w klasie mniejszościowej ponad liczbę próbek w klasie większościowej. Musi to być wartość większa niż 0 i mniejsza lub równa 1. Wartością domyślną jest <code>auto</code> .
<code>random_seed</code>	<i>integer</i>	Wartość początkowa używana przez generator liczb losowych.
<code>k_neighbours</code>	<i>integer</i>	Liczba najbliższych sąsiadów używanych do tworzenia próbek syntetycznych. Wartością domyślną jest 5.
<code>m_neighbours</code>	<i>integer</i>	Liczba najbliższych sąsiadów, na podstawie której określa się, czy próba mniejszościowa jest zagrożona. Ta opcja jest aktywowana tylko z zastosowaniem algorytmów SMOTE typu <code>borderline1</code> i <code>borderline2</code> . Wartością domyślną jest 10.
<code>algorithm_kind</code>	<i>string</i>	Typ algorytmu SMOTE: <code>regular</code> , <code>borderline1</code> lub <code>borderline2</code> .
<code>use_partition</code>	<i>Boolean</i>	Ustawienie wartości <code>true</code> spowoduje, że do budowania modelu użyte zostaną tylko dane uczące. Wartością domyślną jest <code>true</code> .

Właściwość xgboostreencode



XGBoost Tree© to zaawansowana implementacja algorytmu wzmocnienia gradientowego, który jako model bazowy wykorzystuje model drzewa. Algorytm wzmocnienia iteracyjnie uczy się, wyznaczając słabe klasyfikatory i dodając je do ostatecznego silnego klasyfikatora. XGBoost Tree jest algorytmem bardzo elastycznym i oferuje liczne parametry, które mogą być trudne do praktycznego wykorzystania przez użytkowników. Dlatego węzeł Drzewo XGBoost w programie SPSS Modeler eksponuje tylko funkcje podstawowe i najczęściej używane parametry. Węzeł jest zaimplementowany w języku Python.

Tabela 253. Właściwości *xgboosttreenode*

Właściwości <i>xgboosttreenode</i>	Typ danych	Opis właściwości
TargetField	<i>field</i>	Zmienne przewidywane.
InputFields	<i>field</i>	Zmienne wejściowe.
treeMethod	<i>string</i>	Metoda drzewa dla budowania modelu. Możliwe wartości to <i>auto</i> , <i>exact</i> lub <i>approx</i> . Domyślną wartością jest <i>auto</i> .
numBoostRound	<i>integer</i>	Wartość zaokrąglania num boost dla budowania modelu. Należy określić wartość z zakresu od 1 do 1000. Wartością domyślną jest 10.
maxDepth	<i>integer</i>	Maksymalna głębokość dla wzrostu drzewa. Należy określić wartość 1 lub wyższą. Wartością domyślną jest 6.
minChildWeight	<i>Double</i>	Minimalna waga elementu podrzędnego dla wzrostu drzewa. Należy określić wartość 0 lub wyższą. Wartością domyślną jest 1.
maxDeltaStep	<i>Double</i>	Maksymalny krok delta dla wzrostu drzewa. Należy określić wartość 0 lub wyższą. Wartością domyślną jest 0.
objectiveType	<i>string</i>	Typ celu dla zadania uczenia. Możliwe wartości to Δ reg:linear, Δ reg:logistic, Δ reg:gamma, Δ reg:tweedie, count:poisson, Δ rank:pairwise, binary:logistic lub multi. Należy pamiętać, że w przypadku zmiennych przewidywanych typu flaga można używać tylko wartości binary:logistic lub multi. W razie użycia wartości multi wynik oceny będzie przedstawiał typy celów XGBoost multi:softmax i multi:softprob.
random_seed	<i>integer</i>	Wartość początkowa liczb losowych. Jest to dowolna liczba z zakresu od 0 do 9999999. Wartością domyślną jest 0.
sampleSize	<i>Double</i>	Podpróby do sterowania przeuczeniem. Należy określić wartość z zakresu od 0,1 do 1,0. Wartość domyślna to 0,1.
eta	<i>Double</i>	Wartość eta do sterowania przeuczeniem. Należy określić wartość z zakresu od 0 do 1. Wartością domyślną jest 0.3.
gamma	<i>Double</i>	Wartość gamma do sterowania przeuczeniem. Należy określić wartość 0 lub większą. Wartością domyślną jest 6.
colsSampleRatio	<i>Double</i>	Wartość colsample by tree do sterowania przeuczeniem. Należy określić wartość z zakresu od 0,01 do 1. Wartością domyślną jest 1.
colsSampleLevel	<i>Double</i>	Wartość colsample by level do sterowania przeuczeniem. Należy określić wartość z zakresu od 0,01 do 1. Wartością domyślną jest 1.
lambda	<i>Double</i>	Wartość lambda do sterowania przeuczeniem. Należy określić wartość 0 lub większą. Wartością domyślną jest 1.
alpha	<i>Double</i>	Wartość alpha do sterowania przeuczeniem. Należy określić wartość 0 lub większą. Wartością domyślną jest 0.

Tabela 253. Właściwości *xgboosttreenode* (kontynuacja)

Właściwości <i>xgboosttreenode</i>	Typ danych	Opis właściwości
scalePosWeight	<i>Double</i>	Wartość wagi scale pos do obsługi niezrównoważonych zbiorów danych. Wartością domyślną jest 1.

Właściwości *xboostlinearnode*



XGBoost Linear© to zaawansowana implementacja algorytmu wzmacniania gradientowego, który jako model bazowy wykorzystuje model liniowy. Algorytmy wzmacniania iteracyjnie uczą się, wyznaczają słabe klasyfikatory i dodają je do ostatecznego silnego klasyfikatora. Węzeł Liniowy XGBoost w programie SPSS Modeler jest zaimplementowany w języku Python.

Tabela 254. Właściwości *xgboostlinearnode*

Właściwości <i>xgboostlinearnode</i>	Typ danych	Opis właściwości
TargetField	<i>field</i>	
InputFields	<i>field</i>	
alpha	<i>Double</i>	Parametr alpha linear booster. Należy określić wartość 0 lub większą. Wartością domyślną jest 0.
lambda	<i>Double</i>	Parametr lambda linear booster. Należy określić wartość 0 lub większą. Wartością domyślną jest 1.
lambdaBias	<i>Double</i>	Parametr lambda bias linear booster. Należy określić dowolną liczbę. Wartością domyślną jest 0.
numBoostRound	<i>integer</i>	Wartość zaokrąglania num boost dla budowania modelu. Należy określić wartość z zakresu od 1 do 1000. Wartością domyślną jest 10.
objectiveType	<i>string</i>	Typ celu dla zadania uczenia. Możliwe wartości to Δ reg:linear, Δ reg:logistic, Δ reg:gamma, Δ reg:tweedie, count:poisson, Δ rank:pairwise, binary:logistic lub multi. Należy pamiętać, że w przypadku zmiennych przewidywanych typu flaga można używać tylko wartości binary:logistic lub multi. W razie użycia wartości multi wynik oceny będzie przedstawiał typy celów XGBoost multi:softmax i multi:softprob.
random_seed	<i>integer</i>	Wartość początkowa liczb losowych. Jest to dowolna liczba z zakresu od 0 do 9999999. Wartością domyślną jest 0.

Właściwości *ocsvmnode*



Węzeł SVM z jedną klasą korzysta z algorytmu uczenia nienadzorowanego. Węzeł ten można wykorzystać do wykrywania nowości. Wykryje on miękką granicę danego zbioru próbek, a następnie sklasyfikuje nowe punkty jako należące do tego zbioru albo do niego nienależące. Węzeł modelowania SVM z jedną klasą w programie SPSS Modeler został zaimplementowany w języku Python i wymaga biblioteki Python scikit-learn©.

Tabela 255. Właściwości *ocsvmnode*

Właściwości <i>ocsvmnode</i>	Typ danych	Opis właściwości
<code>role_use</code>	<i>string</i>	Określenie wartości <code>predefined</code> pozwala na użycie ról predefiniowanych, a wartości <code>custom</code> na użycie niestandardowych przypisań zmiennych. Wartość domyślna to <code>predefined</code> .
<code>inputs</code>	<i>field</i>	Lista nazw zmiennych dla wartości wejściowych.
<code>splits</code>	<i>field</i>	Lista nazw zmiennych do podziału.
<code>use_partition</code>	<i>Boolean</i>	Można określić wartość <code>true</code> lub <code>false</code> . Wartością domyślną jest <code>true</code> . Ustawienie wartości <code>true</code> spowoduje, że podczas budowania modelu użyte zostaną tylko dane uczące.
<code>mode_type</code>	<i>string</i>	Tryb. Możliwe wartości to <code>simple</code> lub <code>expert</code> . Wszystkie parametry na karcie Zaawansowane zostaną wyłączone, jeśli wybrana zostanie wartość <code>simple</code> .
<code>stopping_criteria</code>	<i>string</i>	Łańcuch notacji naukowych. Możliwe wartości to <code>1.0E-1</code> , <code>1.0E-2</code> , <code>1.0E-3</code> , <code>1.0E-4</code> , <code>1.0E-5</code> lub <code>1.0E-6</code> . Wartością domyślną jest <code>1.0E-3</code> .
<code>precision</code>	<i>float</i>	Precyzja regresji (<code>nu</code>). Granica frakcji dla błędów uczenia i wektorów wspierających. Należy określić wartość większą niż 0 i mniejszą niż lub równą 1,0. Wartość domyślna to 0,1.
<code>kernel</code>	<i>string</i>	Typ jądra do użycia w algorytmie. Możliwe wartości to <code>linear</code> , <code>poly</code> , <code>rbf</code> , <code>sigmoid</code> lub <code>precomputed</code> . Wartość domyślna to <code>rbf</code> .
<code>enable_gamma</code>	<i>Boolean</i>	Włącza parametr <code>gamma</code> . Można określić wartość <code>true</code> lub <code>false</code> . Wartością domyślną jest <code>true</code> .
<code>gamma</code>	<i>float</i>	Ten parametr jest aktywowany tylko dla jąder typu <code>rbf</code> , <code>poly</code> i <code>sigmoid</code> . Jeśli parametr <code>enable_gamma</code> jest ustawiony na wartość <code>false</code> , wówczas ten parametr będzie ustawiony na wartość <code>auto</code> . Jeśli wartość to <code>true</code> , ustawieniem domyślnym jest 0,1.
<code>coef0</code>	<i>float</i>	Warunek niezależny w funkcji algorytmu domyślnego. Ten parametr jest aktywowany tylko dla jądra <code>poly</code> oraz jądra <code>sigmoid</code> . Wartość domyślna to 0,0.
<code>degree</code>	<i>integer</i>	Stopień funkcji algorytmu wielomianowego. Ten parametr jest aktywowany tylko dla jądra <code>poly</code> . Należy określić dowolną liczbę całkowitą. Wartością domyślną jest 3.
<code>shrinking</code>	<i>Boolean</i>	Określa, że ma być używana opcja heurystyki z redukcją. Można określić wartość <code>true</code> lub <code>false</code> . Domyślną wartością jest <code>false</code> .
<code>enable_cache_size</code>	<i>Boolean</i>	Włącza parametr <code>cache_size</code> . Można określić wartość <code>true</code> lub <code>false</code> . Domyślną wartością jest <code>false</code> .
<code>cache_size</code>	<i>float</i>	Wielkość pamięci podręcznej jądra w MB. Wartością domyślną jest 200.
<code>enable_random_seed</code>	<i>Boolean</i>	Włącza parametr <code>random_seed</code> . Można określić wartość <code>true</code> lub <code>false</code> . Domyślną wartością jest <code>false</code> .

Tabela 255. Właściwości `ocsvmnode` (kontynuacja)

Właściwości <code>ocsvmnode</code>	Typ danych	Opis właściwości
<code>random_seed</code>	<i>integer</i>	Wartość początkowa generatora liczb losowych używana podczas mieszania danych na potrzeby oszacowania prawdopodobieństwa. Należy określić dowolną liczbę całkowitą.
<code>pc_type</code>	<i>string</i>	Typ grafiki równoległych współrzędnych. Możliwe opcje to <code>independent</code> lub <code>general</code> .
<code>lines_amount</code>	<i>integer</i>	Maksymalna liczba wierszy, jaką można dołączyć do grafiki. Należy określić liczbę całkowitą z zakresu od 1 do 1000.
<code>lines_fields_custom</code>	<i>Boolean</i>	Aktywuje parametr <code>lines_fields</code> , który umożliwia określenie niestandardowych zmiennych, jakie będą wyświetlane w wyniku graficznym. Ustawienie wartości <code>false</code> spowoduje wyświetlenie wszystkich zmiennych. Po ustawieniu wartości <code>true</code> wyświetlane są tylko zmienne określone przy użyciu parametru <code>lines_fields</code> . Ze względu na wydajność wyświetlanych jest maksymalnie 20 zmiennych.
<code>lines_fields</code>	<i>field</i>	Lista nazw zmiennych do uwzględnienia na grafice w postaci pionowych osi.

Rozdział 20. Właściwości superwęzłów

W poniższej tabeli opisano właściwości charakterystyczne dla superwęzłów. Superwęzły mają też właściwości wspólne dla wszystkich węzłów.

Tabela 256. Właściwości superwęzła końcowego

Nazwa właściwości	Typ właściwości/lista wartości	Opis właściwości
execute_method	Script Normal	
script	string	

Parametry superwęzłów

Za pomocą skryptów można tworzyć lub ustawiać parametry superwęzłów, korzystając z formatu ogólnego:

```
mySuperNode.setParameterValue("minvalue", 30)
```

Wartość parametru można odczytać za pomocą instrukcji:

```
value mySuperNode.getParameterValue("minvalue")
```

Znajdowanie istniejących superwęzłów

Za pomocą funkcji `findByType()` można znajdować superwęzły w strumieniach:

```
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)
```

Ustawianie właściwości węzłów opakowanych

Można ustawiać właściwości konkretnych węzłów opakowanych w superwęzle, uzyskując dostęp do diagramu podrzędnego w superwęzle. Załóżmy na przykład, że mamy źródłowy superwęzeł z opakowanym węzłem Variable File służącym do wczytywania danych. Możemy przekazać nazwę pliku do odczytania (za pomocą właściwości `full_filename`), uzyskując dostęp do diagramu podrzędnego i znajdując odpowiedni węzeł w następujący sposób:

```
childDiagram = source_supernode.getChildDiagram()
varfilenode = childDiagram.findByType("variablefile", None)
varfilenode.setPropertyValue("full_filename", "c:/mydata.txt")
```

Tworzenie superwęzłów

Aby utworzyć superwęzeł i jego zawartość od podstaw, można utworzyć superwęzeł, uzyskać dostęp do diagramu podrzędnego, a następnie utworzyć żądane węzły. Należy zadbać o to, by węzły wewnątrz diagramu superwęzła były powiązane z węzłami-łącznikami na wejściu i/lub wyjściu. Na przykład, aby utworzyć superwęzeł procesowy:

```
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)
```

Dodatek A. Skorowidz nazw węzłów

Ta sekcja zawiera skorowidz nazw węzłów programu IBM SPSS Modeler używanych w skryptach.

Nazwy modeli użytkowych

Do modeli użytkowych (nazywanych także modelami wygenerowanymi) można odwoływać się wg typu, tak jak do węzłów i obiektów wynikowych. W poniższej tabeli wymieniono nazwy służące do odwoływania się do obiektów modeli.

Nazwy te służą do odwoływania się do modeli użytkowych na palecie Modele (w prawym górnym rogu okna programu IBM SPSS Modeler). Na potrzeby odwołań do węzłów dodanych do strumienia w celu oceniania stosuje się inny zestaw nazw — z przedrostkiem `apply...` Więcej informacji można znaleźć w temacie Właściwości węzłów modeli użytkowych.

Uwaga: W normalnych okolicznościach zalecane jest odwoływanie się do modeli przez nazwę i typ, co pozwoli uniknąć niejednoznaczności.

Tabela 257. Nazwy modeli użytkowych (paleta Modelowanie).

Nazwa modelu	Model
anomalydetection	Anomalie
apriori	Apriori
autoclassifier	Auto Klasyfikacja
autocluster	Auto Grupowanie
autonumeric	Auto Predykcja
bayesnet	Sieć bayesowska
c50	C5.0
carma	Carma
cart	Drzewo C&R
chaid	CHAID
coxreg	Regresja Coxa
decisionlist	Lista decyzyjna
discriminant	Dyskryminacyjne
factor	Redukcja wymiarów
featureselection	Dobór predyktorów
genlin	Uogólniona regresja liniowa
glmm	GLMM
kmeans	K-średnie
knn	<i>k</i> -najbliższy sąsiad
kohonen	Sieć Kohonena
linear	Liniowy
logreg	Regresja logistyczna
neuralnetwork	Sieci neuronowe
quest	QUEST

Tabela 257. Nazwy modeli użytkowych (paleta Modelowanie) (kontynuacja).

Nazwa modelu	Model
regression	Regresja liniowa
sequence	Sekwencje
slrm	Model odpowiedzi samonauczenia
statisticsmodel	Model IBM SPSS Statistics
svm	Algorytm SVM
timeseries	Szereg czasowy
twostep	Dwustopniowa

Tabela 258. Nazwy modeli użytkowych (paleta Modelowanie w bazie danych).

Nazwa modelu	Model
db2imcluster	IBM ISW Clustering
db2imlog	IBM ISW Logistic Regression
db2imnb	IBM ISW Naive Bayes
db2imreg	IBM ISW Regression
db2imtree	IBM ISW Decision Tree
msassoc	MS Association Rules
msbayes	MS Naive Bayes
mscluster	MS Clustering
mslogistic	MS Logistic Regression
msneuralnetwork	MS Neural Network
msregression	MS Linear Regression
mssequencecluster	MS Sequence Clustering
mstimeseries	MS Time Series
mstree	MS Decision Tree
netezزابayes	Netezza Bayes Net
netez zadectree	Netezza Decision Tree
netez zadivcluster	Netezza Divisive Clustering
netez zaglm	Netezza Generalized Linear
netez zakmeans	Netezza K-Means
netez zaknn	Netezza KNN
netez zalineregression	Netezza Linear Regression
netez zanaivebayes	Netezza Naive Bayes
netez zapca	Netezza PCA
netez zaregtree	Netezza Regression Tree
netez zatimeseries	Netezza Time Series
oraabn	Oracle Adaptive Bayes
oraai	Oracle AI
oradecisiontree	Oracle Decision Tree
oraglm	Oracle GLM
orakmeans	Oracle <i>k</i> -Means

Tabela 258. Nazwy modeli użytkowych (paleta Modelowanie w bazie danych) (kontynuacja).

Nazwa modelu	Model
oranb	Oracle Naive Bayes
oranmf	Oracle NMF
oraocluster	Oracle O-Cluster
orasvm	Oracle SVM

Unikanie duplikowania nazw modeli

Używając skryptów do manipulowania wygenerowanymi modelami, należy pamiętać, że dopuszczenie do zduplikowania nazw modeli może powodować niejednoznaczność odwołań. Aby tego uniknąć, wskazane jest wymaganie nazw unikalnych dla modeli wygenerowanych w skryptach.

Aby ustawić opcje dotyczące zduplikowanych nazw modeli:

1. Z menu wybierz:
Narzędzia > Opcje użytkownika
2. Kliknij kartę **Powiadomienia**.
3. Wybierz opcję **Zastąp poprzedni model**, aby uniemożliwić duplikowanie nazw wygenerowanych modeli.

Programy SPSS Modeler i IBM SPSS Collaboration and Deployment Services w różny sposób traktują niejednoznaczne odwołania do modelu w skrypcie. Klient SPSS Modeler udostępnia opcję „Zastąp poprzedni model”, która powoduje automatyczne zastępowanie modeli o tej samej nazwie (np. gdy skrypt w każdej iteracji pętli generuje inny model). Jednak opcja ta nie jest dostępna, gdy skrypt jest wykonywany w programie IBM SPSS Collaboration and Deployment Services. Można uniknąć tej sytuacji, w każdej interakcji zmieniając nazwę generowanego modelu albo kasując bieżący model (np. dodając instrukcję `clear generated palette`) przed końcem pętli.

Nazwy typów wynikowych

W poniższej tabeli wymieniono wszystkie typy obiektów wynikowych i węzły, które te obiekty tworzą. Pełną listę formatów eksportu dostępnych dla każdego typu obiektu wynikowego zawiera opis właściwości węzła, który tworzy taki obiekt (patrz Właściwości wspólne węzłów wykresu i Właściwości węzłów wynikowych).

Tabela 259. Typy obiektów wynikowych i węzły, które te obiekty tworzą.

Typ obiektu wynikowego	Węzeł
analysisoutput	Analiza
collectionoutput	Zbiór
dataauditoutput	Audyt danych
distributionoutput	Rozkład
evaluationoutput	Ocena
histogramoutput	Histogram
matrixoutput	Macierz
meansoutput	Średnie
multiplotoutput	Wykres wielokrotny
plotoutput	Wykresy
qualityoutput	Jakość
reportdocumentoutput	Obiekty tego typu nie są tworzone przez węzeł, lecz przez raport z projektu
reportoutput	Raport

Tabela 259. Typy obiektów wynikowych i węzły, które te obiekty tworzą (kontynuacja).

Typ obiektu wynikowego	Węzeł
statisticsprocedureoutput	Wynik Statistics
statisticsoutput	Statystyki
tableoutput	Tabela
timeplotoutput	Wykres sekwencyjny
weboutput	Sieciowy

Dodatek B. Migracja z wcześniejszego języka skryptowego na język Python.

Przegląd migracji z wcześniejszego języka skryptowego

Niniejsza sekcja zawiera podsumowanie różnic między językiem Python a wcześniejszym językiem skryptowym używanym w programie IBM SPSS Modeler. Przedstawiono również informacje o sposobie migracji wcześniejszych skryptów do języka Python. Sekcja ta zawiera listę standardowych używanych wcześniej komend programu SPSS Modeler oraz ich odpowiedników w języku Python.

Różnice ogólne

Wcześniejszy język skryptowy ma wiele wspólnego ze skryptowymi językami komend systemów operacyjnych. Zasadniczo ma on strukturę wierszową i choć występuje w nim kilka struktur blokowych, na przykład `if...then...else...endif` i `for...endfor`, wcięcia w kodzie nie mają znaczenia.

W języku Python wcięcia mają znaczenie i wiersze należące do tego samego bloku logicznego muszą być wcięte do tego samego poziomu.

Uwaga: Przy kopiowaniu i wklejaniu kodu w języku Python należy zachować ostrożność. Wiersz wcięty za pomocą znaków tabulacji może w edytorze wyglądać tak samo, jak wiersz wcięty za pomocą spacji. Jednak interpreter skryptu w języku Python zgłosi błąd, ponieważ wiersze nie zostaną uznane za wcięte do tego samego poziomu.

Kontekst skryptu

Kontekst skryptu definiuje środowisko, w którym jest wykonywany skrypt — na przykład strumień lub superwęzeł, w którym działa skrypt. We wcześniejszym języku skryptowym kontekst był domniemany, co oznaczało, na przykład, że wszelkie odwołania do węzłów w skrypcie strumienia traktowane były z założenia jako odwołania do strumienia wykonującego skrypt.

W języku Python kontekst skryptu jest określony jawnie za pośrednictwem modułu `modeler.script`. Na przykład skrypt strumienia Python może uzyskiwać dostęp do strumienia wykonującego skrypt za pośrednictwem następującego kodu:

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

Następnie, za pośrednictwem zwróconego obiektu, może wywoływać funkcje związane ze strumieniem.

Komendy a funkcje

Wcześniejszy język skryptowy jest zorientowany na komendy. Oznacza to, że każdy wiersz skryptu zwykle zaczyna się od komendy do wykonania, po której następują parametry, na przykład:

```
connect 'Type':typenode to :filternode  
rename :derivenode as "Compute Total"
```

W języku Python używane są funkcje, które zwykle wywołuje się za pośrednictwem obiektu (modułu, klasy lub obiektu), który definiuje funkcję, na przykład:

```
stream = modeler.script.stream()  
typenode = stream.findByType("type", "Type")  
filternode = stream.findByType("filter", None)  
stream.link(typenode, filternode)  
derive.setLabel("Compute Total")
```

Literały i komentarze

Niektóre literały i sposoby komentowania powszechnie używane w języku skryptowym programu IBM SPSS Modeler mają swoje odpowiedniki w języku Python. Może to ułatwić przekształcenie istniejących skryptów SPSS Modeler w skrypty w języku Python przeznaczone do użycia w programie IBM SPSS Modeler 17.

Tabela 260. Odzworowanie literałów i komentarzy między wcześniejszym językiem skryptowym a językiem Python.

Wcześniejszy język skryptowy	Język skryptowy Python
Liczba całkowita, na przykład 4	Tak samo
Liczba zmiennopozycyjna, na przykład 0,003	Tak samo
Łańcuch ujęty w pojedynczy cudzysłów, na przykład 'Hello'	Tak samo Uwaga: Literały łańcuchowe zawierające znaki spoza zestawu ASCII muszą być poprzedzone symbolem u, aby były interpretowane jako zapisane w kodowaniu Unicode.
Łańcuch ujęty w podwójny cudzysłów, na przykład "Hello"	Tak samo Uwaga: Literały łańcuchowe zawierające znaki spoza zestawu ASCII muszą być poprzedzone symbolem u, aby były interpretowane jako zapisane w kodowaniu Unicode.
Długie łańcuchy, na przykład """This is a string that spans multiple lines"""	Tak samo
Listy, na przykład [1 2 3]	[1, 2, 3]
Odwołania do zmiennych, na przykład set x = 3	x = 3
Kontynuacja wiersza (\), na przykład set x = [1 2 \ 3 4]	x = [1, 2,\n3, 4]
Bloki komentarzy, na przykład /* This is a long comment over a line. */	""" This is a long comment over a line. """
Komentarze do wierszy, na przykład set x = 3 # razy 3	x = 3 # razy 3
undef	None
true	True
false	False

Operatory

Niektóre operatory powszechnie używane w języku skryptowym programu IBM SPSS Modeler mają swoje odpowiedniki w języku Python. Może to ułatwić przekształcenie istniejących skryptów SPSS Modeler w skrypty w języku Python przeznaczone do użycia w programie IBM SPSS Modeler 17.

Tabela 261. Odzworowanie operatorów między wcześniejszym językiem skryptowym a językiem Python.

Wcześniejszy język skryptowy	Język skryptowy Python
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

Tabela 261. Odzworowanie operatorów między wcześniejszym językiem skryptowym a językiem Python (kontynuacja).

Wcześniejszy język skryptowy	Język skryptowy Python
= ==	==
/= /==	!=
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
i lub not (EXPR)	i lub not EXPR

Komendy warunkowe i pętle

Niektóre komendy warunkowe i pętle powszechnie używane w języku skryptowym programu IBM SPSS Modeler mają swoje odpowiedniki w języku Python. Może to ułatwić przekształcenie istniejących skryptów SPSS Modeler w skrypty w języku Python przeznaczone do użycia w programie IBM SPSS Modeler 17.

Tabela 262. Odzworowanie komend warunkowych i pętli między wcześniejszym językiem skryptowym a językiem Python.

Wcześniejszy język skryptowy	Język skryptowy Python
for VAR from INT1 to INT2 ... endfor	for VAR in range(INT1, INT2): ... lub VAR = INT1 while VAR <= INT2: ... VAR += 1
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:
with TYPE OBJECT ... endwith	Brak odpowiednika

Tabela 262. Odzworowanie komend warunkowych i pętli między wcześniejszym językiem skryptowym a językiem Python (kontynuacja).

Wcześniejszy język skryptowy	Język skryptowy Python
var VAR1	Deklarowanie zmiennych nie jest wymagane

Zmienne

We wcześniejszym języku skryptowym zmienne były deklarowane przed pierwszym odwołaniem do nich, na przykład:

```
var mynode
set mynode = create typenode at 96 96
```

W języku skryptowym Python zmienne są tworzone przy pierwszym odwołaniu, na przykład:

```
mynode = stream.createAt("type", "Type", 96, 96)
```

We wcześniejszym języku skryptowym odwołania do zmiennych muszą być jawnie usuwane przy użyciu operatora \wedge , na przykład:

```
var mynode
set mynode = create typenode at 96 96
set  $\wedge$ mynode.direction."Age" = Input
```

W języku Python, podobnie jak w większości języków skryptowych, nie jest to wymagane, na przykład:

```
mynode = stream.createAt("type", "Type", 96, 96)
mynode.setKeyedPropertyValue("direction", "Age", "Input")
```

Typy węzłów, wyników i modeli

We wcześniejszym języku skryptowym do nazw typów obiektów zwykle dołączone są przyrostki (node, output i model) określające szerszy rodzaj obiektów. Na przykład węzeł Derive ma typ `derivenode`:

```
set feature_name_node = create derivenode at 96 96
```

W interfejsie API IBM SPSS Modeler w języku Python nie jest stosowany przyrostek `node`, zatem węzeł Derive ma typ `derive`, na przykład:

```
feature_name_node = stream.createAt("derive", "Feature", 96, 96)
```

Jedyna różnica w nazwach typów między wcześniejszym językiem skryptowym a językiem Python polega na braku przyrostka typu.

Nazwy właściwości

Nazwy właściwości są takie same we wcześniejszym języku skryptowym i w języku Python. Na przykład w węźle Plik zmienny właściwość definiująca lokalizację pliku nosi nazwę `full_filename` w obu środowiskach skryptowych.

Odwołania do węzłów

W wielu starszych skryptach stosuje się wyszukiwanie niejawnie węzła, który ma być modyfikowany. Na przykład następujące komendy wyszukiwają w bieżącym strumieniu węzeł Typ z etykietą „Type”, a następnie nadają zmiennej „Age” rolę wejściowej, a zmiennej „Drug” rolę przewidywanej:

```
set 'Type':typenode.direction."Age" = Input
set 'Type':typenode.direction."Drug" = Target
```

W środowisku skryptowym Python obiekty węzłów muszą być odszukane jawnie przed wywołaniem funkcji ustawiającej wartość właściwości, na przykład:

```
typenode = stream.findByType("type", "Type")
typenode.setKeyedPropertyValue("direction", "Age", "Input")
typenode.setKeyedPropertyValue("direction", "Drug", "Target")
```

Uwaga: W tym przypadku słowo "Target" musi być ujęte w cudzysłowy właściwe dla łańcucha.

W skryptach w języku Python można zamiast tej metody użyć także wyliczenia `ModelingRole` z pakietu `modeler.api`.

Mimo że alternatywna wersja w języku Python może być dłuższa, będzie działała wydajniej, ponieważ zwykle wymaga tylko jednokrotnego wyszukania węzła. We wcześniejszym języku skryptowym węzeł jest wyszukiwany przy każdej komendzie.

Możliwe jest także znajdowanie węzłów według identyfikatorów (identyfikator węzła jest widoczny na karcie Adnotacje w oknie dialogowym węzła). Na przykład we wcześniejszym języku skryptowym:

```
# id65EMPB9VL87 jest identyfikatorem węzła Typ
set @id65EMPB9VL87.direction."Age" = Input
```

Ten sam przykład w języku Python:

```
typenode = stream.findById("id65EMPB9VL87")
typenode.setKeyedPropertyValue("direction", "Age", "Input")
```

Odczytywanie i ustawianie właściwości

We wcześniejszym języku skryptowym do przypisywania wartości używana była komenda `set`. Po komendzie `set` mogła nastąpić definicja właściwości. Następujący skrypt ilustruje dwa możliwe formaty komendy ustawiającej wartość właściwości:

```
set <node reference>.<property> = <value>
set <node reference>.<keyed-property>.<key> = <value>
```

W języku skryptowym Python ten sam wynik uzyskuje się, korzystając z funkcji `setProperty()` i `setKeyedPropertyValue()`, na przykład:

```
object.setProperty(property, value)
object.setKeyedPropertyValue(keyed-property, key, value)
```

We wcześniejszym języku skryptowym wartości właściwości można odczytywać za pomocą komendy `get`, na przykład:

```
var n v
set n = get node :filternode
set v = ^n.name
```

W języku skryptowym Python ten sam wynik uzyskuje się za pomocą funkcji `getPropertyValue()`, na przykład:

```
n = stream.findByType("filter", None)
v = n.getPropertyValue("name")
```

Edytowanie strumieni

We wcześniejszym języku skryptowym do tworzenia nowego węzła służy komenda `create`, na przykład:

```
var agg select
set agg = create aggregatenode at 96 96
set select = create selectnode at 164 96
```

W języku Python strumienie oferują różne metody tworzenia węzłów, na przykład:

```
stream = modeler.script.stream()
agg = stream.createAt("aggregate", "Aggregate", 96, 96)
select = stream.createAt("select", "Select", 164, 96)
```

We wcześniejszym języku skryptowym do tworzenia połączeń między węzłami służy komenda `connect`, na przykład:

```
connect ^agg to ^select
```

We języku Python do tworzenia połączeń między węzłami służy metoda `link`, na przykład:

```
stream.link(agg, select)
```

We wcześniejszym języku skryptowym do usuwania połączeń między węzłami służy komenda `disconnect`, na przykład:

```
disconnect ^agg from ^select
```

We języku Python do usuwania połączeń między węzłami służy metoda `unlink`, na przykład:

```
stream.unlink(agg, select)
```

We wcześniejszym języku skryptowym do określania położenia węzłów w obszarze roboczym strumienia lub między innymi węzłami służy komenda `position`, na przykład:

```
position ^agg at 256 256  
position ^agg between ^myselect and ^mydistinct
```

W języku skryptowym Python ten sam wynik uzyskuje się za pomocą dwóch różnych metod: `setXYPosition` i `setPositionBetween`. Na przykład:

```
agg.setXYPosition(256, 256)  
agg.setPositionBetween(myselect, mydistinct)
```

Operacje na węzłach

Niektóre komendy operujące na węzłach powszechnie używane w języku skryptowym programu IBM SPSS Modeler mają swoje odpowiedniki w języku Python. Może to ułatwić przekształcanie istniejących skryptów SPSS Modeler w skrypty w języku Python przeznaczone do użycia w programie IBM SPSS Modeler 17.

Tabela 263. Odzworowanie operacji na węzłach między wcześniejszym językiem skryptowym a językiem Python.

Wcześniejszy język skryptowy	Język skryptowy Python
create <i>nodespec</i> at x y	<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 x y	<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>

W pętli

We wcześniejszym języku skryptowym obsługiwane są dwa główne rodzaje pętli:

- Pętle *odliczane*, w których zmienna sterująca przyjmuje wartości spomiędzy dwóch liczb całkowitych.
- Pętle *sekwencyjne*, w których zmienna sterująca przyjmuje kolejne wartości z określonego szeregu.

Poniższy przykład ilustruje pętlę odliczaną we wcześniejszym języku skryptowym:

```
for i from 1 to 10
  println ^i
endfor
```

Poniższy przykład ilustruje pętlę sekwencyjną we wcześniejszym języku skryptowym:

```
var items
set items = [a b c d]

for i in items
  println ^i
endfor
```

Można stosować także dwa inne typy pętli:

- iteracje przez modele na palecie modeli lub przez wyniki na palecie wyników;
- iteracje przez zmienne wchodzące do węzła lub wychodzące z węzła.

W języku Python również obsługiwane są różne typy pętli. Poniższy przykład ilustruje pętlę odliczaną w języku Python:

```
i = 1
while i <= 10:
  print i
  i += 1
```

Poniższy przykład ilustruje pętlę sekwencyjną w języku Python:

```
items = ["a", "b", "c", "d"]
for i in items:
  print i
```

Pętla sekwencyjna jest bardzo elastyczna, a w połączeniu z metodami interfejsu API IBM SPSS Modeler umożliwia realizację większości konstrukcji stosowanych we wcześniejszym języku skryptowym. Poniższy przykład ilustruje zastosowanie pętli sekwencyjnej w języku Python do iteracyjnego przechodzenia przez zmienne wychodzące z węzła:

```
node = modeler.script.stream().findByType("filter", None)
for column in node.getOutputDataModel().columnIterator():
  print column.getColumnname()
```

Wykonywanie strumieni

W trakcie wykonywania strumienia generowane modele lub obiekty wyjściowe są dodawane do jednego z menedżerów obiektów. We wcześniejszym języku skryptowym skrypt musi wyszukiwać wygenerowane obiekty w menedżerze obiektów albo uzyskiwać dostęp do wyniku ostatnio wygenerowanego z węzła.

W środowisku Python strumienie wykonywane są inaczej: wszystkie modele lub obiekty wynikowe wygenerowane w toku wykonania są zwracane w liście przekazywanej do funkcji wykonawczej. Upraszcza to dostęp do wyników wykonania strumienia.

We wcześniejszym języku skryptowym obsługiwane są trzy komendy wykonywania strumienia:

- `execute_all` wykonuje wszystkie wykonywalne węzły końcowe w strumieniu.
- `execute_script` wykonuje skrypt strumienia niezależnie od ustawienia wykonywania skryptów.
- `execute node` wykonuje określony węzeł.

W środowisku skryptowym Python obsługiwany jest podobny zestaw funkcji:

- `stream.runAll(results-list)` wykonuje wszystkie wykonywalne węzły końcowe w strumieniu.
- `stream.runScript(results-list)` wykonuje skrypt strumienia niezależnie od ustawienia wykonywania skryptów.
- `stream.runSelected(node-array, results-list)` wykonuje określony zbiór węzłów w kolejności, w jakiej zostaną wymienione.
- `node.run(results-list)` wykonuje określony węzeł.

We wcześniejszym języku skryptowym wykonywanie strumienia można zakończyć komendą `exit` z opcjonalnym kodem całkowitoliczbowym, na przykład:

```
exit 1
```

W języku Python ten sam efekt uzyskuje się przy użyciu następującego kodu:

```
modeler.script.exit(1)
```

Dostęp do obiektów za pośrednictwem systemu plików i repozytorium

We wcześniejszym języku skryptowym można otworzyć istniejący strumień, model lub obiekt wynikowy za pomocą komendy `open`, na przykład:

```
var s
set s = open stream "c:/my streams/modeling.str"
```

W środowisku skryptowym Python istnieje klasa `TaskRunner` dostępna z sesji, której można używać do wykonywania podobnych zadań, na przykład:

```
taskrunner = modeler.script.session().getTaskRunner()
s = taskrunner.openStreamFromFile("c:/my streams/modeling.str", True)
```

Aby zapisać obiekt, we wcześniejszym języku skryptowym można użyć komendy `save`, na przykład:

```
save stream s as "c:/my streams/new_modeling.str"
```

Równoważnym rozwiązaniem w środowisku skryptowym Python byłoby zastosowanie klasy `TaskRunner`, na przykład:

```
taskrunner.saveStreamToFile(s, "c:/my streams/new_modeling.str")
```

Operacje oparte na IBM SPSS Collaboration and Deployment Services Repository są we wcześniejszym języku skryptowym obsługiwane za pośrednictwem komend `retrieve` i `store`, na przykład:

```
var s
set s = retrieve stream "/my repository folder/my_stream.str"
store stream ^s as "/my repository folder/my_stream_copy.str"
```

W języku skryptowym Python równoważny efekt uzyskuje się za pośrednictwem obiektu `Repository` skojarzonego z sesją, na przykład:

```
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)
```

Uwaga: Dostęp do repozytorium jest możliwy pod warunkiem, że w sesji skonfigurowane jest poprawne połączenie z repozytorium.

Operacje na strumieniach

Niektóre komendy operujące na strumieniach powszechnie używane w języku skryptowym programu IBM SPSS Modeler mają swoje odpowiedniki w języku Python. Może to ułatwić przekształcenie istniejących skryptów SPSS Modeler w skrypty w języku Python przeznaczone do użycia w programie IBM SPSS Modeler 17.

Tabela 264. Odzworowanie operacji na strumieniach między wcześniejszym językiem skryptowym a językiem Python.

Wcześniejszy język skryptowy	Język skryptowy Python
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>	Brak odpowiednika
load stream <i>path</i>	Brak odpowiednika
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>
retrive stream <i>path</i>	<i>repository.retriveStream(path, version, label, autoManage)</i>
store <i>stream</i> as <i>path</i>	<i>repository.storeStream(stream, path, label)</i>

Operacje na modelach

Niektóre komendy operujące na modelach powszechnie używane w języku skryptowym programu IBM SPSS Modeler mają swoje odpowiedniki w języku Python. Może to ułatwić przekształcenie istniejących skryptów SPSS Modeler w skrypty w języku Python przeznaczone do użycia w programie IBM SPSS Modeler 17.

Tabela 265. Odzworowanie operacji na modelach między wcześniejszym językiem skryptowym a językiem Python.

Wcześniejszy język skryptowy	Język skryptowy Python
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>

Operacje zapisywania dokumentów

Niektóre operacje zapisu dokumentów powszechnie używane w języku skryptowym programu IBM SPSS Modeler mają swoje odpowiedniki w języku Python. Może to ułatwić przekształcenie istniejących skryptów SPSS Modeler w skrypty w języku Python przeznaczone do użycia w programie IBM SPSS Modeler 17.

Tabela 266. Odzworowanie operacji zapisywania dokumentów między wcześniejszym językiem skryptowym a językiem Python.

Wcześniejszy język skryptowy	Język skryptowy Python
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>
retrieve output <i>path</i>	<i>repository.retrieveDocument(path, version, label, autoManage)</i>
store <i>output</i> as <i>path</i>	<i>repository.storeDocument(output, path, label)</i>

Pozostałe różnice między wcześniejszym językiem skryptowym a językiem Python.

Wcześniejszy język skryptowy oferuje mechanizmy manipulowania projektami IBM SPSS Modeler. Środowisko skryptowe Python obecnie nie oferuje takiej możliwości.

Wcześniejszy język skryptowy oferował pewne mechanizmy służące do ładowania obiektów *state* (stanów, tj. kombinacji strumieni i modeli). Stany są nieaktualne od wersji 8.0 programu IBM SPSS Modeler. Środowisko skryptowe Python nie obsługuje stanów.

Język skryptowy Python oferuje następujące dodatkowe mechanizmy niedostępne we wcześniejszym języku skryptowym:

- Definicje funkcji klas
- Obsługa błędów
- Bardziej zaawansowane metody obsługi wejścia/wyjścia
- Moduły zewnętrzne i moduły innych firm

Uwagi

Niniejsza publikacja została przygotowana z myślą o produktach i usługach oferowanych w Stanach Zjednoczonych. Materiał ten jest również dostępny w IBM w innych językach. Jednakże w celu uzyskania dostępu do takiego materiału istnieje konieczność posiadania egzemplarza produktu w takim języku.

Produktów, usług lub opcji opisywanych w tym dokumencie IBM nie musi oferować we wszystkich krajach. Informacje o produktach i usługach dostępnych w danym kraju można uzyskać od lokalnego przedstawiciela IBM. Odwołanie do produktu, programu lub usługi IBM nie oznacza, że można użyć wyłącznie tego produktu, programu lub usługi IBM. Zamiast nich można zastosować ich odpowiednik funkcjonalny pod warunkiem, że nie narusza to praw własności intelektualnej IBM. Jednakże cała odpowiedzialność za ocenę przydatności i sprawdzenie działania produktu, programu lub usługi pochodzących od producenta innego niż IBM spoczywa na użytkowniku.

IBM może posiadać patenty lub złożone wnioski patentowe na towary i usługi, o których mowa w niniejszej publikacji. Przedstawienie niniejszej publikacji nie daje żadnych uprawnień licencyjnych do tychże patentów. Pisemne zapytania w sprawie licencji można przysyłać na adres:

*IBM Director of Licensing
IBM Corporation
North Castle Drive, MD-NC119
Armonk, NY 10504-1785
U.S.A.*

Zapytania dotyczące zestawów znaków dwubajtowych (DBCS) należy kierować do lokalnych działów własności intelektualnej IBM (IBM Intellectual Property Department) lub wysłać je na piśmie na adres:

*Intellectual Property Licensing
Legal and Intellectual Property Law
IBM Japan, Ltd.
19-21, Nihonbashi-Hakozakicho, Chuo-ku
Tokio 103-8510, Japonia*

INTERNATIONAL BUSINESS MACHINES CORPORATION DOSTARCZA TĘ PUBLIKACJĘ W STANIE, W JAKIM SIĘ ZNAJDUJE ("AS IS") BEZ UDZIELANIA JAKICHKOLWIEK GWARANCJI (RĘKOJMIĘ RÓWNIEŻ WYŁĄCZA SIĘ), WYRAŹNYCH LUB DOMNIEMANYCH, A W SZCZEGÓLNOŚCI DOMNIEMANYCH GWARANCJI PRZYDATNOŚCI HANDLOWEJ, PRZYDATNOŚCI DO OKREŚLONEGO CELU ORAZ GWARANCJI, ŻE PUBLIKACJA TA NIE NARUSZA PRAW OSÓB TRZECICH. Ustawodawstwa niektórych krajów nie dopuszczają zastrzeżeń dotyczących gwarancji wyraźnych lub domniemanych w odniesieniu do pewnych transakcji; w takiej sytuacji powyższe zdanie nie ma zastosowania.

Informacje zawarte w niniejszej publikacji mogą zawierać nieścisłości techniczne lub błędy drukarskie. Informacje te są okresowo aktualizowane, a zmiany te zostaną uwzględnione w kolejnych wydaniach tej publikacji. IBM zastrzega sobie prawo do wprowadzania ulepszeń i/lub zmian w produktach i/lub programach opisanych w tej publikacji w dowolnym czasie, bez wcześniejszego powiadomienia.

Wszelkie wzmianki w tej publikacji na temat stron internetowych innych podmiotów zostały wprowadzone wyłącznie dla wygody użytkownika i w żadnym wypadku nie stanowią zachęty do ich odwiedzania. Materiały dostępne na tych stronach nie są częścią materiałów opracowanych dla tego produktu IBM, a użytkownik korzysta z nich na własną odpowiedzialność.

IBM ma prawo do używania i rozpowszechniania informacji przysłanych przez użytkownika w dowolny sposób, jaki uzna za właściwy, bez żadnych zobowiązań wobec ich autora.

Licencjobiorcy tego programu, którzy chcieliby uzyskać informacje na temat programu w celu: (i) wdrożenia wymiany informacji między niezależnie utworzonymi programami i innymi programami (łącznie z tym opisywanym) oraz (ii) wspólnego wykorzystywania wymienianych informacji, powinni skontaktować się z:

*IBM Director of Licensing
IBM Corporation
North Castle Drive, MD-NC119
Armonk, NY 10504-1785
U.S.A.*

Informacje takie mogą być udostępnione, o ile spełnione zostaną odpowiednie warunki, w tym, w niektórych przypadkach, zostanie uiszczona stosowna opłata.

Licencjonowany program opisany w niniejszej publikacji oraz wszystkie inne licencjonowane materiały dostępne dla tego programu są dostarczane przez IBM na warunkach określonych w Umowie IBM z Klientem, Międzynarodowej Umowie Licencyjnej IBM na Program lub w innych podobnych umowach zawartych między IBM i użytkownikami.

Dane dotyczące wydajności i cytowane przykłady zostały przedstawione jedynie w celu zobrazowania sytuacji. Faktyczne wyniki dotyczące wydajności mogą się różnić w zależności do konkretnych warunków konfiguracyjnych i operacyjnych.

Informacje dotyczące produktów innych podmiotów niż IBM zostały uzyskane od dostawców tych produktów, z ich publicznych ogłoszeń lub innych dostępnych publicznie źródeł. IBM nie testował tych produktów i nie może potwierdzić dokładności pomiarów wydajności, kompatybilności ani żadnych innych danych związanych z tymi produktami. Pytania dotyczące możliwości produktów innych podmiotów należy kierować do dostawców tych produktów.

Wszelkie stwierdzenia dotyczące przyszłych kierunków rozwoju i zamierzeń IBM mogą zostać zmienione lub wycofane bez powiadomienia.

Publikacja ta zawiera przykładowe dane i raporty używane w codziennej pracy. W celu kompleksowego ich zilustrowania, podane przykłady zawierają nazwiska osób prywatnych, nazwy przedsiębiorstw oraz nazwy produktów. Wszystkie te nazwy/nazwiska są fikcyjne i jakiegokolwiek podobieństwo do istniejących nazw/nazwisk jest całkowicie przypadkowe.

Znaki towarowe

IBM, logo IBM i ibm.com są znakami towarowymi lub zastrzeżonymi znakami towarowymi International Business Machines Corp. zarejestrowanymi w wielu systemach prawnych na całym świecie. Pozostałe nazwy produktów i usług mogą być znakami towarowymi IBM lub innych przedsiębiorstw. Aktualna lista znaków towarowych IBM dostępna jest w serwisie WWW IBM, w sekcji "Copyright and trademark information" (Informacje o prawach autorskich i znakach towarowych), pod adresem www.ibm.com/legal/copytrade.shtml.

Adobe, logo Adobe, PostScript oraz logo PostScript są znakami towarowymi lub zastrzeżonymi znakami towarowymi Adobe Systems Incorporated w Stanach Zjednoczonych i/lub w innych krajach.

Intel, logo Intel, Intel Inside, logo Intel Inside, Intel Centrino, logo Intel Centrino, Celeron, Intel Xeon, Intel SpeedStep, Itanium i Pentium są znakami towarowymi lub zastrzeżonymi znakami towarowymi Intel Corporation lub przedsiębiorstw podporządkowanych Intel Corporation w Stanach Zjednoczonych i w innych krajach.

Linux jest zastrzeżonym znakiem towarowym Linusa Torvaldsa w Stanach Zjednoczonych i/lub w innych krajach.

Microsoft, Windows, Windows NT oraz logo Windows są znakami towarowymi Microsoft Corporation w Stanach Zjednoczonych i/lub w innych krajach.

UNIX jest zastrzeżonym znakiem towarowym The Open Group w Stanach Zjednoczonych i/lub w innych krajach.

Java oraz wszystkie znaki towarowe i logo dotyczące języka Java są znakami towarowymi lub zastrzeżonymi znakami towarowymi Oracle i/lub przedsiębiorstw afiliowanych.

Warunki dotyczące dokumentacji produktu

Zezwolenie na korzystanie z tych publikacji jest przyznawane na poniższych warunkach.

Zakres stosowania

Niniejsze warunki stanowią uzupełnienie warunków używania serwisu WWW IBM.

Użytek osobisty

Użytkownik ma prawo kopiować te publikacje do własnego, niekomercyjnego użytku pod warunkiem zachowania wszelkich uwag dotyczących praw własności. Użytkownik nie ma prawa dystrybuować ani wyświetlać tych publikacji czy ich części, ani też wykonywać na ich podstawie prac pochodnych bez wyraźnej zgody IBM.

Użytek służbowy

Użytkownik ma prawo kopiować te publikacje, dystrybuować je i wyświetlać wyłącznie w ramach przedsiębiorstwa Użytkownika pod warunkiem zachowania wszelkich uwag dotyczących praw własności. Użytkownik nie ma prawa wykonywać na podstawie tych publikacji ani ich fragmentów prac pochodnych, kopiować ich, dystrybuować ani wyświetlać poza przedsiębiorstwem Użytkownika bez wyraźnej zgody IBM.

Prawa

Z wyjątkiem zezwoleń wyraźnie udzielonych w niniejszym dokumencie, nie udziela się jakichkolwiek innych zezwoleń, licencji ani praw, wyraźnych czy domniemanych, odnoszących się do tych publikacji czy jakichkolwiek informacji, danych, oprogramowania lub innej własności intelektualnej, o których mowa w niniejszym dokumencie.

IBM zastrzega sobie prawo do anulowania zezwolenia przyznanego w niniejszym dokumencie w każdej sytuacji, gdy, według uznania IBM, korzystanie z tych publikacji jest szkodliwe dla IBM lub jeśli IBM uzna, że warunki niniejszego dokumentu nie są przestrzegane.

Użytkownik ma prawo pobierać, eksportować lub reeksportować niniejsze informacje pod warunkiem zachowania bezwzględnej i pełnej zgodności z obowiązującym prawem i przepisami, w tym ze wszelkimi prawami i przepisami eksportowymi Stanów Zjednoczonych.

IBM NIE UDZIELA JAKICHKOLWIEK GWARANCJI, W TYM TAKŻE RĘKOJMI, DOTYCZĄCYCH TREŚCI TYCH PUBLIKACJI. PUBLIKACJE TE SĄ DOSTARCZANE W STANIE, W JAKIM SIĘ ZNAJDUJĄ ("AS-IS") BEZ UDZIELANIA JAKICHKOLWIEK GWARANCJI (RĘKOJMIĘ RÓWNIEŻ WYŁĄCZA SIĘ), WYRAŹNYCH CZY DOMNIEMANYCH, A W SZCZEGÓLNOŚCI DOMNIEMANYCH GWARANCJI PRZYDATNOŚCI HANDLOWEJ, PRZYDATNOŚCI DO OKREŚLONEGO CELU CZY NIENARUSZANIA PRAW OSÓB TRZECICH.

Indeks

A

- argumenty
 - IBM SPSS Analytic Server Repository,
 - połączenie 65
 - plik komend 66
 - połączenie z repozytorium IBM SPSS Collaboration and Deployment Services Repository 65
 - połączenie z serwerem 64
 - system 62
- automatyczne przygotowanie danych
 - właściwości 126

B

- bezpieczeństwo
 - kodowane hasła 52, 64
- bloki kodu 19

C

- CLEM
 - skrypty 1

D

- definiowanie atrybutów 24
- definiowanie klasy 23
- definiowanie metod 24
- derive_stbnode
 - właściwości 105
- diagramy 27
- dodawanie atrybutów 24
- dostęp do wyników wykonania strumienia 53, 57
 - model zawartości JSON 56
 - model zawartości tabeli 53
 - model zawartości XML 55
- dziedziczenie 25

E

- exportModelToFile 40

F

- flagi
 - argumenty wiersza komend 61
 - łączone stosowanie wielu flag 66
- funkcja lowertoupper 49
- funkcje
 - komendy warunkowe 347
 - komentarze 346
 - literały 346
 - odwołania do obiektów 346
 - operacje na modelach 353
 - operacje na strumieniach 353
 - operacje na węzłach 350

- funkcje (*kontynuacja*)
 - operacje zapisywania dokumentów 353
 - operatory 346
 - wykonanie w pętli 347
- funkcje łańcuchowe 49

H

- hasła
 - dodawanie do skryptów 52
 - kodowane 64

I

- IBM SPSS Analytic Server Repository
 - argumenty wiersza komend 65
- IBM SPSS Collaboration and Deployment Services Repository
 - argumenty wiersza komend 65
 - skrypty 49
- IBM SPSS Modeler
 - uruchamianie z wiersza komend 61
- identyfikatory 19
- instrukcje 19

J

- Jython 15

K

- klucz iteracji
 - pętle w skryptach 8
- kodowane hasła
 - dodawanie do skryptów 52
- kolejność wykonywania
 - zmiana za pomocą skryptów 49
- kolejność wykonywania strumienia
 - zmiana za pomocą skryptów 49
- komenda clear generated palette 52
- komenda for 49
- komenda store 49
- komendy retrieve 49

L

- listy 16
- looping in streams 6, 7

Ł

- łańcuchy 17
 - zmiana wielkości liter 49

M

- metody matematyczne 21

migrowanie

- Czyszczenie zawartości menedżerów strumieni, wyników i modeli 34
- dostęp do obiektów 352
- edytowanie strumieni 349
- funkcje 345
- komendy 345
- kontekst skryptu 345
- nazwy właściwości 348
- odczytywanie właściwości 349
- odwołania do węzłów 348
- przegląd 345
- repozytorium 352
- różne 353
- różnice ogólne 345
- system plików 352
- typy modeli 348
- typy węzłów 348
- typy wyników 348
- ustawianie właściwości 349
- wykonanie w pętli 351
- wykonywanie strumieni 351
- zmienne 348
- Model użytkowy Reguły asocjacyjne
 - właściwości 256
- Model użytkowy STP
 - właściwości 268
- model zawartości JSON 56
- model zawartości tabeli 53
- model zawartości XML 55
- modele
 - nazwy używane w skryptach 341, 343
- modele apriori
 - właściwości węzła w skryptach 175, 255
- modele Auto Grupowanie
 - właściwości węzła w skryptach 257
- Modele Auto Klasyfikacja
 - właściwości węzła w skryptach 256
- modele Auto Predykcja
 - właściwości węzła w skryptach 257
- modele autopredykcji
 - właściwości węzła w skryptach 182
- Modele C&R Tree
 - właściwości węzła w skryptach 187, 258
- modele C5.0
 - właściwości węzła w skryptach 185, 257
- modele CARMA
 - właściwości węzła w skryptach 186, 258
- modele CHAID
 - właściwości węzła w skryptach 190, 258
- modele Drzewa losowe
 - właściwości węzła w skryptach 230, 267
- modele Drzewo-AS
 - właściwości węzła w skryptach 250, 270
- Modele Dwustopniowa
 - właściwości węzła w skryptach 252, 270
- Modele Dwustopniowa-AS
 - właściwości węzła w skryptach 253, 270
- modele dyskryminacyjne
 - właściwości węzła w skryptach 195, 259

modele GLE			
właściwości węzła w skryptach	209, 263		
Modele GLMM			
właściwości węzła w skryptach	205, 262		
Modele IBM SPSS Statistics			
właściwości węzła w skryptach	330		
modele K-średnich			
właściwości węzła w skryptach	213, 263		
modele KNN			
właściwości węzła w skryptach	263		
modele Kohonena			
właściwości węzła w skryptach	216, 264		
modele liniowe			
właściwości węzła w skryptach	217, 264		
modele liniowe maszyny wektorów nośnych			
właściwości węzła w skryptach	223, 265		
modele Liniowy-AS			
właściwości węzła w skryptach	218, 264		
modele list decyzyjnych			
właściwości węzła w skryptach	193, 259		
modele LSVM			
właściwości węzła w skryptach	223		
Modele Microsoft			
właściwości węzła w skryptach	273, 275		
modele najbliższego sąsiedztwa			
właściwości węzła w skryptach	214		
Modele Netezza			
właściwości węzła w skryptach	284		
Modele Netezza Bayes Net			
właściwości węzła w skryptach	284, 294		
Modele Netezza Decision Tree			
właściwości węzła w skryptach	284, 294		
Modele Netezza Divisive Clustering			
właściwości węzła w skryptach	284, 294		
Modele Netezza Generalized Linear			
właściwości węzła w skryptach	284		
Modele Netezza K-Means			
właściwości węzła w skryptach	284, 294		
Modele Netezza KNN			
właściwości węzła w skryptach	284, 294		
Modele Netezza Linear Regression			
właściwości węzła w skryptach	284, 294		
Modele Netezza Naive Bayes			
właściwości węzła w skryptach	284, 294		
Modele Netezza PCA			
właściwości węzła w skryptach	284, 294		
Modele Netezza Regression Tree			
właściwości węzła w skryptach	284, 294		
Modele Netezza Time Series			
właściwości węzła w skryptach	284		
Modele odpowiedzi samonauczania			
właściwości węzła w skryptach	235, 268		
Modele Oracle			
właściwości węzła w skryptach	277		
Modele Oracle Adaptive Bayes			
właściwości węzła w skryptach	277, 283		
Modele Oracle AI			
właściwości węzła w skryptach	277		
Modele Oracle Apriori			
właściwości węzła w skryptach	277, 283		
Modele Oracle Decision Tree			
właściwości węzła w skryptach	277, 283		
Modele Oracle Generalized Linear			
właściwości węzła w skryptach	277		
Modele Oracle KMeans			
właściwości węzła w skryptach	277, 283		
Modele Oracle MDL			
właściwości węzła w skryptach	277, 283		
Modele Oracle Naive Bayes			
właściwości węzła w skryptach	277, 283		
Modele Oracle NMF			
właściwości węzła w skryptach	277, 283		
Modele Oracle Support Vector Machines			
właściwości węzła w skryptach	277, 283		
modele PCA			
właściwości węzła w skryptach	199, 261		
Modele przyczynowe szeregów czasowych			
właściwości węzła w skryptach	240		
modele Python			
właściwości węzła w skryptach	266, 270, 271		
modele QUEST			
właściwości węzła w skryptach	228, 266		
modele Redukcja wymiarów			
właściwości węzła w skryptach	199, 261		
modele regresji Coxa			
właściwości węzła w skryptach	192, 259		
modele regresji liniowej			
właściwości węzła w skryptach	231, 267		
modele regresji logistycznej			
właściwości węzła w skryptach	219, 264		
modele sekwencji			
właściwości węzła w skryptach	233, 268		
modele sieci bayesowskiej			
właściwości węzła w skryptach	183		
Modele sieci bayesowskiej			
właściwości węzła w skryptach	257		
modele sieci neuronowych			
właściwości węzła w skryptach	224, 265		
modele SLRM			
właściwości węzła w skryptach	235, 268		
modele SVM			
właściwości węzła w skryptach	239, 268		
Modele SVM			
właściwości węzła w skryptach	239		
Modele Szeregi czasowe			
właściwości węzła w skryptach	118		
modele szeregów czasowych			
właściwości węzła w skryptach	244, 248, 269		
modele TCM			
właściwości węzła w skryptach	269		
modele użytkowe			
nazwy używane w skryptach	341, 343		
właściwości węzła w skryptach	255		
modele wyboru predyktora			
właściwości węzła w skryptach	200, 262		
Modele wyboru predyktora			
skrypty	5		
stosowanie	5		
modele wykrywania anomalii			
właściwości węzła w skryptach	173, 255		
modelowanie w bazie danych	273		
modyfikowanie strumieni	30, 33		
MS Decision Tree			
właściwości węzła w skryptach	273, 275		
MS Linear Regression			
właściwości węzła w skryptach	273, 275		
MS Logistic Regression			
właściwości węzła w skryptach	273, 275		
MS Neural Network			
właściwości węzła w skryptach	273, 275		
MS Sequence Clustering			
właściwości węzła w skryptach	275		
MS Time Series			
właściwości węzła w skryptach	275		
N			
nazwy zmiennych			
zmiana wielkości liter	49		
O			
obiekty modeli			
nazwy używane w skryptach	341, 343		
obiekty wynikowe			
nazwy używane w skryptach	343		
odwołania do węzłów	29		
ustawianie właściwości	30		
znajdowanie węzłów	29		
operacje	16		
Oracle O-Cluster			
właściwości węzła w skryptach	277, 283		
P			
parametry	5, 67, 68, 71		
skrypty	16		
Superwęzły	339		
parametry zagnieżdżone	5, 67, 69		
pętle			
używanie w skryptach	49		
przechodzenie przez węzły	33		
przekazywanie argumentów	20		
przerywanie wykonania skryptów	11		
przykłady	20		
Python	15		
skrypty	16		
R			
reguły asocjacyjne, węzeł			
właściwości	176		
S			
serwer			
argumenty wiersza komend	64		
sieci neuronowe			
właściwości węzła w skryptach	226, 265		
Skryptowy interfejs API			
dostęp do wygenerowanych obiektów	40		
metadane	38		
obsługa błędów	42		
parametry sesji	42		
parametry strumienia	42		
Parametry superwęzła	42		
pobieranie katalogu	37		
przykład	37		
skrypty samodzielne	47		
wartości globalne	46		
wiele strumieni	47		
wstęp	37		
wyszukiwanie	37		
skrypty			
diagramy	27		

skrypty (*kontynuacja*)
 importowanie z plików tekstowych 1
 interfejs użytkownika 1, 4, 5
 język skryptowy Python 346, 347, 350, 353
 klucz iteracji 8
 kontekst 28
 Modele wyboru predyktora 5
 przegląd 1, 15
 przerywanie 11
 składnia 16, 17, 18, 19, 20, 21, 22, 23, 24, 25
 skróty używane skryptach 68
 skrypty samodzielne 1, 27
 skrypty superwęzłów 1, 27
 strumienie 1, 27
 Strumienie superwęzłów 27
 w superwęzłach 5
 wcześniejszy język skryptowy 346, 347, 350, 353
 węzły wykresu 155
 węzły wyników 297
 wizualne konfigurowanie pętli 6, 7
 Właściwości wspólne 69
 wybieranie zmiennych 10
 wykonanie w pętli 6, 7
 wykonanie warunkowe 6, 10
 wykonywanie 11
 z wiersza komend 52
 zapisywanie 1
 zgodność z wcześniejszymi wersjami 52
 zmienna iteracji 9
 skrypty samodzielne 1, 4, 27
 słowo kluczowe generated 52
 sprawdzanie w poszukiwaniu błędów skrypty 52
 strumienie
 modyfikowanie 30
 skrypty 1, 27
 ustawianie wielokrotne 67
 właściwości 71
 wykonanie w pętli 6, 7
 wykonanie warunkowe 6, 10
 wykonywanie 27
 superwęzeł 67
 Superwęzeł
 strumień 27
 Superwęzły
 parametry 339
 skrypty 1, 5, 6, 27, 339
 strumienie 27
 ustawianie właściwości wewnątrz 339
 właściwości 339
 system
 argumenty wiersza komend 62

T
 tworzenie klasy 24
 tworzenie skryptów
 kolejność wykonywania strumienia 49
 sprawdzanie w poszukiwaniu błędów 52
 tworzenie węzłów 30, 31, 32

U
 uogólnione modele liniowe
 właściwości węzła w skryptach 202, 262
 ustawianie wielokrotne 67
 ustawianie właściwości 30
 uwagi 18
 uzyskiwanie dostępu do wyników wykonania strumienia 53, 57
 model zawartości JSON 56
 model zawartości tabeli 53
 model zawartości XML 55

W
 warunkowe wykonanie strumieni 6, 10
 węzeł Agregacja
 właściwości 103
 węzeł Agregacja RFM
 właściwości 111
 węzeł Analiza
 właściwości 297
 węzeł Analiza RFM
 właściwości 141
 węzeł Anonimizacja
 właściwości 125
 węzeł Audyt danych
 właściwości 298
 Węzeł Auto Grupowanie
 właściwości węzła w skryptach 180
 Węzeł Auto Klasyfikacja
 właściwości węzła w skryptach 178
 Węzeł Baza danych
 właściwości 82
 Węzeł budowy R
 właściwości węzła w skryptach 184
 Węzeł Dane użytkownika
 właściwości 97
 węzeł dołączania
 właściwości 103
 Węzeł dopasowania symulacji
 właściwości 307
 Węzeł eksportu Data Collection
 właściwości 320
 Węzeł eksportu do bazy danych
 właściwości 316
 Węzeł eksportu do pliku IBM SPSS Statistics
 właściwości 331
 Węzeł eksportu do programu Excel
 właściwości 320
 Węzeł eksportu SAS
 właściwości 323
 Węzeł eksportu XML
 właściwości 327
 węzeł ewaluacji
 właściwości 158
 węzeł Filtruj
 właściwości 136
 Węzeł generowania symulacji
 właściwości 93
 Węzeł Globalne
 właściwości 305
 Węzeł Graphboard
 właściwości 160
 Węzeł histogramu
 właściwości 162

węzeł Historia
 właściwości 137
 węzeł importowania przez rozszerzenie
 właściwości 87
 węzeł kategoryzacji
 właściwości 129
 węzeł Liniowy XGBoost
 właściwości 335
 węzeł Łączenie
 właściwości 110
 węzeł Macierz
 właściwości 301
 Węzeł oceny symulacji
 właściwości 306
 węzeł optymalizacji CPLEX
 właściwości 124
 Węzeł Plik kolumnowy
 właściwości 89
 Węzeł Plik płaski
 właściwości 322
 węzeł Plik zmienny
 właściwości 98
 węzeł Podział na podzbiory
 właściwości 137
 węzeł Powtórzenia
 właściwości 107
 Węzeł Predykcja przestrzenno-czasowa (STP)
 właściwości 236
 Węzeł próby
 właściwości 113
 węzeł Przedziały czasowe
 właściwości 143
 węzeł Przedziały czasowe AS
 właściwości 129
 Węzeł przekształceń IBM SPSS Statistics
 właściwości 329
 Węzeł przekształceń R
 właściwości 113
 węzeł Raport
 właściwości 304
 węzeł Rekodowanie
 właściwości 138
 węzeł Reorganizacja
 właściwości 139
 Węzeł Reorganizacja
 właściwości 139
 węzeł Restrukturyzacja
 właściwości 140
 Węzeł rozkładu
 właściwości 157
 węzeł Selekcja
 właściwości 115
 węzeł Siatka czasoprzestrzeni
 właściwości 105, 116
 Węzeł sieciowy
 właściwości 171
 Węzeł Sieć kierunkowa
 właściwości 171
 węzeł SMOTE
 właściwości 333
 Węzeł Sortowanie
 właściwości 115
 węzeł Statistics
 właściwości 307
 węzeł STP
 właściwości 236

węzeł SVM z jedną klasą
właściwości 335

Węzeł Szeregi czasowe
właściwości 121

węzeł Średnie
właściwości 302

węzeł Tabela
właściwości 309

węzeł Transformacja
właściwości 311

węzeł transformacji przez rozszerzenie
właściwości 108

węzeł Transpozycja
właściwości 147

węzeł typu
właściwości 149

węzeł Ustaw jako flaga
właściwości 142

Węzeł ważenia
właściwości 104

Węzeł wizualizacja na mapie
właściwości 163

Węzeł wykresu
właściwości 168

Węzeł wykresu sekwencyjnego
właściwości 170

Węzeł wykresu wielokrotnego
właściwości 167

Węzeł wycięzeń
właściwości 132

węzeł wynikowy Extension
właściwości 321

Węzeł wynikowy Extension
właściwości 300
właściwości węzła w skryptach 196

Węzeł wynikowy IBM SPSS Statistics
właściwości 331

Węzeł wynikowy R
właściwości 305

węzeł Wypełnianie
właściwości 135

węzeł XGBoost Tree
właściwości 333

Węzeł zbioru
właściwości 156

węzeł Zespolenie
właściwości 134

węzeł Zmiana rzutowania
właściwości 140

Węzeł źródła IBM Cognos TM1
właściwości 95

Węzeł źródłowy Analytic Server
właściwości 79

Węzeł źródłowy Dane
właściwości 101

Węzeł źródłowy danych geoprzestrzennych
właściwości 92

Węzeł źródłowy Data Collection
właściwości 84

Węzeł źródłowy IBM Cognos
właściwości 80

węzeł źródłowy IBM SPSS Statistics
właściwości 329

Węzeł źródłowy programu Excel
właściwości 86

Węzeł źródłowy SAS
właściwości 92

węzeł źródłowy TWC Import
właściwości 96

Węzeł źródłowy XML
właściwości 101

węzły
importowanie 32
informacyjne 34
przechodzenie w pętli w skryptach 49
skorowidz nazw 341
tworzenie połączeń między węzłami 31
usuwanie 32
usuwanie połączeń między węzłami 31
zastępowanie 32

węzły eksportu
właściwości węzła w skryptach 313

węzły modelowania
właściwości węzła w skryptach 173

węzły wykresu
właściwości używane w skryptach 155

węzły wyników
właściwości używane w skryptach 297

węzły źródłowe
właściwości 75

wiersz komend
lista argumentów 62, 64, 65
parametry 63
skrypty 52
uruchamianie programu IBM SPSS Modeler 61
wiele argumentów 66

właściwości
skrypty 67, 68, 69, 173, 255, 313
strumień 71
Superwęzły 339
węzły filtrowania 67
węzły modelowania w bazie danych 273
wspólne w skryptach 69

właściwości anomalydetectionnode
węzła 173

właściwości applyocsvm 266

właściwości applyxgboostlinearnode 271

właściwości applyxgboosttreenode 270

właściwości cplexnode 124

właściwości ocsvmnode 335

właściwości sieci bayesowskiej 183

właściwości smotenode 333

właściwości ustrukturyzowane 67

właściwości węzła aggregatednode 103

właściwości węzła analysisnode 297

właściwości węzła anonymizenode 125

właściwości węzła appendnode 103

właściwości węzła
applyanomalydetectionnode 255

właściwości węzła applyapriorinode 255

właściwości węzła
applyassociationrulesnode 256

właściwości węzła
applyautoclassifiernode 256

właściwości węzła applyautoclusternode 257

właściwości węzła
applyautonumericnode 257

właściwości węzła applybayesnetnode 257

właściwości węzła applyc50node 257

właściwości węzła applycarmanode 258

właściwości węzła applycartnode 258

właściwości węzła applychaidnode 258

Właściwości węzła applycoxregnode 259

właściwości węzła applydecisionlistnode 259

właściwości węzła
applydiscriminantnode 259

właściwości węzła applyextension 260

właściwości węzła applyfactornode 261

właściwości węzła
applyfeatureselectionnode 262

właściwości węzła
applygeneralizedlinearnode 262

właściwości węzła applygle 263

właściwości węzła applyglmnode 262

właściwości węzła applykmeansnode 263

Właściwości węzła applyknnnode 263

właściwości węzła applykohonenode 264

właściwości węzła applylinearasnode 264

właściwości węzła applylinearnode 264

właściwości węzła applylogregnode 264

właściwości węzła applylsvmnode 265

właściwości węzła applymlslogisticnode 275

właściwości węzła
applymlsneuralnetworknode 275

właściwości węzła
applymlsregressionnode 275

właściwości węzła
applymlssequenceclusternode 275

właściwości węzła
applymlstimeseriesnode 275

właściwości węzła applymlstreenode 275

właściwości węzła
applynetzezzabayesnode 294

właściwości węzła
applynetzezzadectreenode 294

właściwości węzła
applynetzezzadivclusternode 294

właściwości węzła
applynetzezzakmeansnode 294

właściwości węzła applynetzezzaknnnode 294

właściwości węzła
applynetzezzalineregressionnode 294

właściwości węzła
applynetzezzanaivebayesnode 294

właściwości węzła
applynetzezzapcanode 294

właściwości węzła
applynetzezzaregtreenode 294

Właściwości węzła applyneuralnetnode 265

właściwości węzła
applyneuralnetworknode 265

właściwości węzła applyoraabnode 283

właściwości węzła
applyoradecisiontreenode 283

właściwości węzła applyorakmeansnode 283

właściwości węzła applyoranbnode 283

właściwości węzła applyoranmfnode 283

właściwości węzła applyoraoclusternode 283

właściwości węzła applyorasvmnode 283

właściwości węzła applyquestnode 266

właściwości węzła applyr 267

właściwości węzła applyrandomtrees 267

właściwości węzła applyregressionnode 267

właściwości węzła
applyselflearningnode 268

właściwości węzła applysequencenode 268

właściwości węzła applystpnode 268

właściwości węzła applysvmnode 268

właściwości węzła applytcmnode 269

właściwości węzła applytimeseriesnode 269

właściwości węzła applytreeas 270

właściwości węzła `applyts` 269
właściwości węzła `applytwostepAS` 270
właściwości węzła `applytwostepnode` 270
właściwości węzła `apriorinode` 175
właściwości węzła `asexport` 313
właściwości węzła `asimport` 79
właściwości węzła `associationrulesnode` 176
właściwości węzła `astimeintervalsnode` 129
właściwości węzła `autoclassifiernode` 178
właściwości węzła `autoclusternode` 180
właściwości węzła `autodataprepnode` 126
Właściwości węzła `autonumericnode` 182
właściwości węzła `balancenode` 104
właściwości węzła `binningnode` 129
właściwości węzła `builldr` 184
właściwości węzła `c50node` 185
właściwości węzła `carmanode` 186
właściwości węzła `cartnode` 187
właściwości węzła `chaidnode` 190
właściwości węzła `cognosimport` 80
właściwości węzła `collectionnode` 156
właściwości węzła `coxregnode` 192
właściwości węzła `dataauditnode` 298
właściwości węzła `databaseexportnode` 316
właściwości węzła `basenode` 82
właściwości węzła
 `datacollectionexportnode` 320
właściwości węzła
 `datacollectionimportnode` 84
właściwości węzła `dataviewimport` 101
właściwości węzła `decisionlist` 193
właściwości węzła `derivenode` 132
właściwości węzła `directedwebnode` 171
właściwości węzła `discriminantnode` 195
właściwości węzła `distinctnode` 107
Właściwości węzła `distributionnode` 157
Właściwości węzła `dopasowania symulacji`
 właściwości 307
właściwości węzła `ensemblenode` 134
właściwości węzła `evaluationnode` 158
właściwości węzła `excelexportnode` 320
właściwości węzła `excelimportnode` 86
właściwości węzła `extensionexportnode` 321
właściwości węzła `extensionimportnode` 87
właściwości węzła `extensionmodelnode` 196
właściwości węzła `extensionoutputnode` 300
właściwości węzła `extensionprocessnode` 108
właściwości węzła `factornode` 199
właściwości węzła `featureselectionnode` 5,
 200
właściwości węzła `fillernode` 135
właściwości węzła `filternode` 136
właściwości węzła `fixedfilenode` 89
właściwości węzła `flatfilenode` 322
właściwości węzła `genlinnode` 202
właściwości węzła `gle` 209
właściwości węzła `glmnode` 205
właściwości węzła `graphboardnode` 160
właściwości węzła `gsdata_import` 92
właściwości węzła `histogramnode` 162
właściwości węzła `historynode` 137
właściwości węzła `kmeansnode` 213
właściwości węzła `knnnode` 214
właściwości węzła `kohonenode` 216
właściwości węzła `linear` 217
właściwości węzła `Liniowy-AS` 218
właściwości węzła `logregnode` 219
właściwości węzła `lsvmnode` 223
właściwości węzła `mapvisualization` 163
właściwości węzła `matrixnode` 301
właściwości węzła `meansnode` 302
właściwości węzła `mergenode` 110
właściwości węzła `msassocnode` 273
właściwości węzła `msbayesnode` 273
właściwości węzła `msclusternode` 273
właściwości węzła `mslogisticnode` 273
właściwości węzła
 `msneuralnetworknode` 273
Właściwości węzła `msregressionnode` 273
właściwości węzła
 `mssequenceclusternode` 273
właściwości węzła `mstimeseriesnode` 273
właściwości węzła `mstreenode` 273
właściwości węzła `multiplotnode` 167
właściwości węzła `netezزابayesnode` 284
właściwości węzła `netezzadectreenode` 284
właściwości węzła
 `netezzadivclusternode` 284
właściwości węzła `netezzaglmnode` 284
właściwości węzła `netezzakmeansnode` 284
właściwości węzła `netezzaknnnode` 284
właściwości węzła
 `netezzalinerregressionnode` 284
właściwości węzła
 `netezzanaivebayesnode` 284
właściwości węzła `netezzapcanode` 284
właściwości węzła `netezzaregtreenode` 284
właściwości węzła
 `netezzatimeseriesnode` 284
Właściwości węzła `neuralnetnode` 224
właściwości węzła `neuralnetworknode` 226
właściwości węzła
 `numericpredictornode` 182
właściwości węzła `oraabnnode` 277
właściwości węzła `oraainode` 277
właściwości węzła `oraapriorinode` 277
właściwości węzła `oradecisiontreenode` 277
właściwości węzła `oraglmnode` 277
właściwości węzła `orakmeansnode` 277
właściwości węzła `oramdlnode` 277
właściwości węzła `oranbnnode` 277
właściwości węzła `oranmfnode` 277
właściwości węzła `oraoclusternode` 277
właściwości węzła `orasvmnode` 277
właściwości węzła `outputfilenode` 322
Właściwości węzła `partitionnode` 137
właściwości węzła `plotnode` 168
właściwości węzła `questnode` 228
właściwości węzła `randomtrees` 230
właściwości węzła `reclassifynode` 138
właściwości węzła `regressionnode` 231
właściwości węzła `reordernode` 139
właściwości węzła `reportnode` 304
właściwości węzła `reprojectnode` 140
właściwości węzła `restructurenode` 140
właściwości węzła `rfmaggregatenode` 111
właściwości węzła `rfanalysisnode` 141
właściwości węzła `routputnode` 305
właściwości węzła `Rprocessnode` 113
właściwości węzła `samplenode` 113
właściwości węzła `sasexportnode` 323
właściwości węzła `sasimportnode` 92
właściwości węzła `selectnode` 115
właściwości węzła `sequencenode` 233
właściwości węzła `setglobalsnode` 305
Właściwości węzła `settoflagnode` 142
właściwości węzła `Siatka`
 czasoprzestrzeni 105
właściwości węzła `simevalnode` 306
właściwości węzła `simfitnode` 307
właściwości węzła `simgennode` 93
Właściwości węzła `slrmnode` 235
właściwości węzła `sortnode` 115
właściwości węzła `spacetimeboxes` 116
właściwości węzła `statisticsexportnode` 331
właściwości węzła `statisticsimportnode` 329
Właściwości węzła `statisticsimportnode` 5
właściwości węzła `statisticsmodelnode` 330
właściwości węzła `statisticsnode` 307
właściwości węzła `statisticsoutputnode` 331
właściwości węzła
 `statisticstransformnode` 329
Właściwości węzła `stpnode` 236
właściwości węzła `streamingtimeseries` 118
właściwości węzła `streamingts` 121
właściwości węzła `svmnode` 239
właściwości węzła `tablenode` 309
Właściwości węzła `tcnode` 240
właściwości węzła `timeintervalsnode` 143
właściwości węzła `timeplotnode` 170
właściwości węzła `timeseriesnode` 248
właściwości węzła `tmlimport` 95
właściwości węzła `tmlodataimport` 95
właściwości węzła `transformnode` 311
właściwości węzła `transposenode` 147
właściwości węzła `treeas` 250
właściwości węzła `ts` 244
właściwości węzła `twcimport` 96
właściwości węzła `twostepAS` 253
właściwości węzła `twostepnode` 252
właściwości węzła `typenode` 149
Właściwości węzła `typenode` 5
właściwości węzła `userinputnode` 97
właściwości węzła `variablefilenode` 98
właściwości węzła w skryptach 273
 modele użytkowe 255
 węzły modelowania 173
Właściwości węzła w skryptach
 węzły eksportu 313
właściwości węzła `webnode` 171
właściwości węzła `xmllexportnode` 327
właściwości węzła `xmlimportnode` 101
właściwości węzła `xgboostlinearnode` 335
właściwości węzła `xgboosttreenode` 333
właściwość `stream.nodes` 49
wygenerowane modele
 nazwy używane w skryptach 341, 343
wykonywanie skryptów 11
Wykonywanie strumieni 27

Z

zmiana rzutowania na nowy układ
 współrzędnych
 właściwości 140
zmienność iteracji
 pętle w skryptach 9
zmienne
 skrypty 16
 wyłączanie w skryptach 155
zmienne ukryte 25

znajdowanie węzłów 29
znaki spoza zestawu ASCII 22
zorientowane obiektowo 23



Drukowane w USA