

*Руководство по сценариям и
автоматизации Python для IBM
SPSS Modeler 16*

IBM

Примечание

Прежде чем использовать эту информацию и продукт, описанный в ней, прочтите сведения в разделе “Уведомления” на стр. 257.

Информация о продукте

Это издание применимо к версии 16, выпуск 0, модификация 0 IBM SPSS Modeler и ко всем последующим версиям и модификациям до тех пор, пока в новых изданиях не будет указано иное.

Содержание

Глава 1. Скрипты 1

Обзор сценариев	1
Типы сценариев	1
Сценарии потока	2
Автономные сценарии	3
Сценарии надузла	3
Потоки с циклами и условиями	4
Циклы в потоках	5
Выполнение с условиями в потоках	8
Выполнение и прерывание сценариев	9
Найти и заменить	10

Глава 2. Язык сценариев. 13

Обзор языка сценария	13
Rython и Jython	13
Сценарий Rython	14
Операции	14
Списки	14
Строки	15
Комментарии	16
Синтаксис операторов	17
Идентификаторы	17
Блоки кода	17
Передача аргументов в сценарий	18
Примеры	18
Математические методы	19
Использование символов не из кодового набора ASCII	20
Объектно-ориентированное программирование.	21
Определение класса	22
Создание экземпляра класса	22
Добавление атрибутов к экземпляру класса	22
Определение атрибутов классов и методов	23
Скрытые переменные	23
Наследование.	23

Глава 3. Сценарии в IBM SPSS Modeler 25

Типы сценариев	25
потоки, потоки надузлов и диаграммы	25
Потоки	25
Потоки надузлов	25
Диаграммы	25
Выполнение потока	25
Контекст сценариев	26
Ссылки на существующие узлы	27
Поиск узлов	27
Задание свойств	28
Создание узлов и изменение потоков	29
Создание узлов	29
Соединение и отсоединение узлов	29
Импорт, замена и удаление узлов	30
Перемещение по узлам в потоке	31
Получение информации об узлах	32

Глава 4. API сценариев 35

Введение в API сценариев	35
Пример: поиск узлов с помощью пользовательского фильтра	35
Метаданные: Информация о данных	35
Доступ к сгенерированным объектам	38
Обработка ошибок	39
Параметры потока, сеанса и надузла	40
Глобальные значения	44
Работа с несколькими потоками: автономные сценарии	45

Глава 5. Подсказки для сценариев . . . 47

Изменение выполнения потока	47
Работа с моделями	47
Генерирование закодированного пароля	47
Проверка сценария	48
Работа со сценариями из командной строки	48
Задание путей файлов	48
Совместимость с предыдущими выпусками	48

Глава 6. Аргументы командной строки. 49

Вызов программного обеспечения	49
Использование аргументов командной строки	49
Системные аргументы	50
Аргументы параметров	51
Аргументы соединения с сервером	51
Аргументы соединения с IBM SPSS Collaboration and Deployment Services Repository	52
Объединение нескольких аргументов	53

Глава 7. Справочник по свойствам . . . 55

Справочный обзор свойств	55
Сокращения	55
Примера свойств узла и потока	55
Обзор свойств узлов	56
Общие свойства узлов	56

Глава 8. Свойства потока 57

Глава 9. Свойства узла источника . . . 61

Общие свойства узлов источников	61
Свойства узла asimport	62
Свойства узла cognosimport	63
Свойства узла database	64
Свойства узла datacollectionimport	65
Свойства узла excelimport	67
Свойства узла evimport	68
Свойства узла fixedfile	69
Свойства узла sasimport	71
Свойства узла simgen	71
Свойства узла statisticsimport	74
Свойства узла userinput	74

Свойства узла variablefile	75
Свойства узла xmlimport	78

Глава 10. Запись свойств узла операций 79

Свойства узла append	79
Свойства узла aggregate	79
Свойства узла balance	80
Свойства узла derive_stb	80
Свойства узла distinct	82
Свойства узла merge	82
Свойства узла rfnaggregate	83
Свойства узла Rprocess	84
Свойства узла sample	85
Свойства узла select	86
Свойства узла sort	87
Свойства узла streamingts	87

Глава 11. Поле Свойства узла операций 91

Свойства узла anonymize	91
Свойства узла autodataprep	91
Свойства узла binning	94
Свойства узла derive	97
Свойства узла ensemble	98
Свойства узла filler	98
Свойства узла filter	99
Свойства узла history	99
Свойства узла partition	100
Свойства узла reclassify	101
Свойства узла reorder	102
Свойства узла restructure	102
Свойства узла rfinanalysis	102
Свойства узла settoflag	104
Свойства узла statistictransform	104
Свойства узла timeintervals	104
Свойства узла transpose	108
Свойства узла type	109

Глава 12. Свойства узла графика 115

Общие свойства узла графика	115
Свойства узла collection	116
Свойства узла distribution	117
Свойства узла evaluation	117
Свойства узла graphboard	119
Свойства узла histogram	121
Свойства узла multiplot	122
Свойства узла plot	123
Свойства узла timeplot	125
Свойства узла web	126

Глава 13. Свойства узла моделирования 127

Общие свойства узлов моделирования	127
Свойства узла anomalydetection	128
Свойства узла apriori	129
Свойства узла autclassifier	130
Задание свойств алгоритмов	131
Свойства узла autcluster	132

Свойства узла autonumeric	133
Свойства узла bayesnet	134
Свойства узла buildr	135
Свойства узла c50	136
Свойства узла carma	137
Свойства узла cart	138
Свойства узла chaid	139
Свойства узла coxreg	141
Свойства узла decisionlist	142
Свойства узла discriminant	144
Свойства узла factor	145
Свойства узла featureselection	146
Свойства узла genlin	148
Свойства узла glmm	151
Свойства узла kmeans	155
Свойства узла knn	155
Свойства узла kohonen	157
Свойства узла linear	157
Свойства узла logreg	159
Свойства узла neuralnet	161
Свойства узла neuralnetwork	163
Свойства узла quest	165
Свойства узла regression	166
Свойства узла sequence	168
Свойства узла slrm	168
Свойства узла statisticsmodel	169
Свойства узла svm	169
Свойства узла timeseries	170
Свойства узла twostep	172

Глава 14. Свойства узла слепков моделей 175

Свойства узла applyanomalydetection	175
Свойства узла applyapriori	175
Свойства узла applyautclassifier	176
Свойства узла applyautcluster	176
Свойства узла applyautnumeric	176
Свойства узла applybayesnet	177
Свойства узла applyc50	177
Свойства узла applycarma	177
Свойства узла applycart	177
Свойства узла applychaid	178
Свойства узла applycoxreg	178
Свойства узла applydecisionlist	179
Свойства узла applydiscriminant	179
Свойства узла applyfactor	179
Свойства узла applyfeatureselection	179
Свойства узла applygeneralizedlinear	180
Свойства узла applyglmm	180
Свойства узла applykmeans	180
Свойства узла applyknn	181
Свойства узла applykohonen	181
Свойства узла applylinear	181
Свойства узла applylogreg	181
Свойства узла applyneuralnet	181
Свойства узла applyneuralnetwork	182
Свойства узла applyquest	182
Свойства узла applyregression	183
Свойства узла applyr	183
Свойства узла applyselflearning	183
Свойства узла applysequence	183

Свойства узла <code>applysvm</code>	184
Свойства узла <code>applytimeseries</code>	184
Свойства узла <code>applytwestep</code>	184

Глава 15. Свойства узла моделирования базы данных 185

Свойства узлов для моделирования Microsoft	185
Свойства узлов моделирования Microsoft	185
Свойства слепков моделей Microsoft	187
Свойства узлов для моделирования Oracle	188
Свойства узлов моделирования Oracle	188
Свойства слепков моделей Oracle	194
Свойства узла для моделирования IBM DB2	195
Свойства узла моделирования IBM DB2	195
Свойства слепков моделей IBM DB2	200
Свойства узлов для моделирования IBM Netezza Analytics	201
Свойства узлов моделирования Netezza	201
Свойства слепков моделей Netezza	210

Глава 16. Свойства узлов вывода 213

Свойства узла <code>analysis</code>	213
Свойства узла <code>dataaudit</code>	214
Свойства узла <code>matrix</code>	215
Свойства узла <code>means</code>	216
Свойства узла <code>report</code>	218
Свойства узла <code>Routput</code>	218
Свойства узла <code>setglobals</code>	219
Свойства узла <code>simeval</code>	219
Свойства узла <code>simfit</code>	220
Свойства узла <code>statistics</code>	221
Свойства узла <code>statisticsoutput</code>	222
Свойства узла <code>table</code>	222
Свойства узла <code>transform</code>	224

Глава 17. Свойства узла экспорта 227

Общие свойства узлов экспорта	227
Свойства узла <code>asexport</code>	227
Свойства узла <code>cognosexport</code>	228
Свойства узла <code>databaseexport</code>	229
Свойства узла <code>datacollectionexport</code>	232
Свойства узла <code>excelexport</code>	232
Свойства узла <code>outputfile</code>	233
Свойства узла <code>sasexport</code>	234
Свойства узла <code>statisticsexport</code>	234
Свойства узла <code>xmlexport</code>	234

Глава 18. Свойства узла IBM SPSS Statistics 237

Свойства узла <code>statisticsimport</code>	237
---	-----

Свойства узла <code>statisticstransform</code>	237
Свойства узла <code>statisticsmodel</code>	238
Свойства узла <code>statisticsoutput</code>	238
Свойства узла <code>statisticsexport</code>	238

Глава 19. Свойства надузлов 241

Приложение А. Ссылки на имена узлов 243

Имена слепков моделей	243
Исключение дублирования имен моделей	245
Имена типов вывода	245

Приложение В. Перенастройка от унаследованных сценарием к сценариям Python 247

Обзор перенастройки унаследованных сценариев	247
Общие отличия	247
Контекст сценариев	247
Команды или функции	247
Литералы и комментарии	248
Операторы	248
Условное выполнение и циклы	249
Переменные	250
Типы узлов, объектов вывода и моделей.	250
Имена свойств	250
Ссылки на узлы.	250
Получение и задание свойств	251
Редактирование потоков	251
Операции с узлами.	252
Циклы.	252
выполнение потоков	253
Доступ к объектам через файловую систему и репозиторий	254
Операции с потоками	254
Операции с моделями.	255
Операции вывода документов	255
Другие различия между унаследованными сценариями и сценариями Python	255

Уведомления 257

Товарные знаки.	258
-------------------------	-----

Индекс 261

Глава 1. Скрипты

Обзор сценариев

Сценарии в IBM® SPSS Modeler - это мощный инструмент для автоматизации процессов в пользовательском интерфейсе. Сценарии могут выполнять действия того же типа, которые выполняются с помощью мыши или клавиатуры, и их можно использовать для автоматизации задач, которые при ручном выполнении могли бы быть многократно повторяемыми или требующими большого времени.

Вы можете использовать сценарии для следующего:

- Установить конкретный порядок для выполнений узлов в потоке и выполнять узлы в зависимости от того, соблюдается ли условие для их выполнения.
- Создавать циклы для последовательного выполнения узлов в потоке.
- Задавать автоматическую последовательность действий, которые обычно выполняются с участием пользователя, например, можно построить модель и затем испытать ее.
- Сконфигурировать сложные процессы, требующие существенных взаимодействий с пользователями, например, процедуры перекрестной проверки, для которых нужны повторяющиеся действия по созданию и испытанию моделей.
- Сконфигурировать процессы, обращающиеся с потоками, например, можно взять поток обучения модели, запустить его, а затем автоматически создать соответствующий поток испытания модели.

В этой части представлены описания высокого уровня и примеры сценариев на уровне потока, автономных сценариев и сценариев на надузлах в интерфейсе IBM SPSS Modeler. Более подробная информация о языке сценариев, синтаксисе и командах представлена в последующих главах.¹

Примечание: Нельзя импортировать и запускать сценарии, созданные в IBM SPSS Statistics в составе IBM SPSS Modeler.

Типы сценариев

IBM SPSS Modeler использует три типа сценариев:

- **Потоковые сценарии** хранятся как свойство потока и поэтому сохраняются и загружаются вместе с конкретным потоком. Например, можно написать потоковый сценарий для автоматизации процесса обучения и применения слепка модели. Можно задать также, чтобы при любом выполнении конкретного потока вместо содержимого холста потока запускался сценарий.
- **Автономные сценарии** не связаны с каким-то конкретным потоком и сохраняются во внешних текстовых файлах. Автономный сценарий можно использовать, например, для одновременной работы с несколькими потоками.
- **Сценарии надузла** хранятся как свойство потока надузла. Сценарии надузла доступны только на конечных надузлах. Сценарий надузла можно использовать для управления выполнением последовательности содержимого надузла. Для не конечных надузлов (источников или процессов) непосредственно в вашем потоковом сценарии можно определить свойства для надузла или содержащихся в нем узлов.

1. Язык унаследованных сценариев IBM SPSS Modeler по-прежнему доступен для использования совместно с IBM SPSS Modeler 16. Дополнительную информацию смотрите в документе *IBM SPSS Modeler 16 Руководство по сценариям и автоматизации*. В разделе Приложение В, “Перенастройка от унаследованных сценариев к сценариям Python”, на стр. 247 вы найдете указания по отображению существующих унаследованных сценариев IBM SPSS Modeler в сценарии Python.

Сценарии потока

Сценарии могут использоваться для настройки операций в конкретном потоке, и они сохраняются с этим потоком. Сценарии потоков можно использовать для указания конкретного порядка выполнения для конечных узлов в потоке. Диалоговое окно сценариев потоков используется для изменения сценария, который сохранен в текущем потоке.

Чтобы получить доступ на вкладку сценариев потоков в диалоговом окне Свойства потоков:

1. В меню Инструменты выберите:
Свойства потоков > Выполнение
2. Перейдите на вкладку **Выполнение** для работы со сценариями для текущего потока.
3. Выберите режим выполнения: **По умолчанию (необязательный сценарий)**.

Значки панели инструментов в верхней части диалогового окна сценария потока позволяют вам выполнить следующие операции:

- Импортировать в это окно содержимое уже существующего автономного сценария.
- Сохранит сценарий в виде текстового файла.
- Распечатать сценарий.
- Добавить в конец сценарий по умолчанию.
- Отредактировать сценарий (функции откат, вырезать, копировать, вставить и другие обычные функции редактирования).
- Выполнить весь текущий сценарий.
- Выполнить выбранные в сценарии строки.
- Остановить сценарий во время выполнения. (Этот значок доступен, только когда выполняется сценарий.)
- Проверить синтаксис сценария и в случае обнаружения каких-то ошибок вывести их для изучения на нижней панели диалогового окна.

Дополнительно можно указать, будет или нет запущен этот сценарий при выполнении потока. Можно выбрать опцию **Запускать этот сценарий** для запуска сценария всякий раз при выполнении потока с учетом порядка выполнения сценария. Этот параметр на уровне потока обеспечивает автоматизацию для ускорения построения модели. Однако по умолчанию этот сценарий игнорируется при выполнении сценариев. Даже если выбрана опция **Игнорировать этот сценарий**, вы всегда можете запустить сценарий непосредственно из этого диалогового окна.

Кроме того, можно переключиться со сценариев Python на унаследованные сценарии.

Редактор сценариев поддерживает следующие возможности, помогающие разрабатывать сценарии:

- Выделение синтаксиса; выделяются ключевые слова, литеральные значения (строчные и числовые), комментарии.
- Нумерация строк.
- Выявление парных блоков; когда указатель помещают у начала программного блока, выделяется также соответствующий конечный блок.
- Предлагаемое автозаполнение.

Цвета и стили текста при выделении синтаксиса можно настроить в предпочтениях экрана IBM SPSS Modeler. Для доступа к предпочтениям экрана выберите **Инструменты > Опции > Опции пользователя** и щелкните по вкладке **Синтаксис**.

Для вывода списка предлагаемых завершений синтаксиса выберите **Автопредложение** в контекстном меню или нажмите клавиши Ctrl + Пробел. При помощи клавиш со стрелками можно перемещаться вверх и вниз по списку; клавиша Enter вставляет выделенный текст. При помощи клавиши Esc можно выйти из режима автозаполнений без изменения существующего текста.

Вкладка **Отладка** содержит сообщения отладки и на ней можно оценить состояние сценария после выполнения. Вкладка **Отладка** состоит из текстовой области, доступной только для чтения, и текстового поля, в которое можно ввести одну строку текста. Текстовая область содержит текст, отправленный либо на стандартное устройство вывода, например, через команду Python `print`, или в стандартный поток ошибок через команду в сценарии, например, через текст сообщения об ошибке. В текстовое поле пишет пользователь. Этот текст оценивается в контексте последнего выполненного сценария в этом диалоговом окне (так называемый *сценарный контекст*). Текстовая область содержит команду и полученный вывод, так что пользователь может просматривать трассировку команд. Текстовое поле всегда содержит приглашение командной строки (для сценариев Python это `>>>`).

Новый сценарный контекст создается в следующих ситуациях:

- Запущен некоторый сценарий при помощи кнопки “Выполнить этот сценарий” или “Выполнить выбранные строки”.
- Изменен язык сценариев.

Когда создается новый сценарный контекст, текстовая область очищается.

Примечание: Если выполнить поток вне панели сценариев, контекст сценария панели сценариев не изменяется. Значения переменных, созданных в ходе такого выполнения, не будут видны в диалоговом окне сценария.

Автономные сценарии

Диалоговое окно Автономный сценарий используется, чтобы создать или изменить сценарий, сохраненный как текстовый файл. Он выводит имя файла и предоставляет возможности для загрузки, сохранения, импорта и выполнения сценариев.

Чтобы получить доступ к диалоговому окну автономного сценария:

В основном меню выберите:

Инструменты > Автономный сценарий

Для автономных и потоковых сценариев доступна одинаковая панель инструментов и опции проверки синтаксиса сценариев. Дополнительную информацию смотрите в разделе “Сценарии потока” на стр. 2.

Сценарии надузла

Используя язык сценариев IBM SPSS Modeler можно создавать и сохранять сценарии на любом конечном надузле. Эти сценарии доступны только для конечных надузлов и часто используются при создании потоков шаблонов или для установления особого порядка выполнения для содержимого надузлов. Сценарии надузлов позволяют также одновременный запуск нескольких сценариев в потоке.

Например, допустим, что вам нужно было задать порядок выполнения для сложного потока, а надузел содержит несколько узлов, в том числе узел Задать глобальные значения, который нужно выполнить до получения нового поля, используемого на узле График. В этом случае можно создать сценарий надузла, первоначально выполняющий узел Задать глобальные значения. Вычисленные этим узлом значения, такие как среднее или среднеквадратичное отклонение, могут затем использоваться при выполнении узла График.

Свойства узлов можно задать в сценарии надузла таким же образом, как это делается в других сценариях. Как вариант, изменить или определить свойства для любого надузла или содержащихся в нем узлов можно непосредственно в потоковом сценарии. Дополнительную информацию смотрите в разделе Глава 19, “Свойства надузлов”, на стр. 241. Этот способ работает для надузлов источников и процессов, а также для конечных надузлов.

Примечание: Так как свои собственные сценарии могут выполнять только конечные надузлы, только для них доступна вкладка Сценарии в диалоговом окне Надузел.

Чтобы открыть диалоговое окно сценария надузла на главном холсте:

Выберите конечный надузел на холсте потоков и в меню Надузел перейдите к следующему пункту:

Сценарий надузла...

Чтобы открыть диалоговое окно сценария надузла на раскрывающемся холсте надузла:

Щелкните правой кнопкой мыши по холсту надузла и в контекстном меню выберите:

Сценарий надузла...

Потоки с циклами и условиями

Начиная с версии 16.0, простые сценарии в SPSS Modeler можно создавать внутри потока, не вводя сами инструкции на языке сценариев, а выбирая значения в различных диалоговых окнах. Два основных типа сценария, доступных для создания этим способом, - это простые циклы и возможность выполнять узлы в зависимости от выполнения условия.

Возможно сочетание в одном потоке и циклов, и условий. Пусть, например, вы располагаете данными о продажах автомобилей, произведенных по всему миру. Вы можете создать цикл для обработки данных в потоке, задав подробности о стране производителя, и вывести данные на различные диаграммы, чтобы показать такие подробности, как объем продаж по моделям, уровни эмиссии вредных веществ в выхлопных газах по производителю и рабочему объему двигателя и так далее. Если вам нужна только информация по Европе, вы можете также добавить в цикл условия, чтобы не создавать диаграмм для производителей из Америки и Азии.

Примечание: Поскольку циклы и условия основаны на фоновых сценариях, они применяются только к потоку в целом, при его выполнении.

- **Выполнение в цикле** Циклы служат для автоматизации повторяющихся задач. Например, можно добавить в поток заданное число узлов, изменяя каждый раз один параметр узла. Также можно устроить многократное (с заданным числом раз) выполнение потока или его ветви, как в следующих примерах:
 - Выполнить поток заданное число раз, каждый раз изменяя источник.
 - Выполнить поток заданное число раз, каждый раз изменяя значение некоторой переменной.
 - Выполнить поток заданное число раз, каждый раз вводя дополнительное поле.
 - Построить модель заданное число раз, каждый раз изменяя параметр модели.
- **Выполнение с условиями** С его помощью можно управлять запуском конечных узлов с учетом заданных условий; вот несколько возможных примеров:
 - Управлять запуском узла с учетом полученного значения true или false.
 - Определить, что узлы в цикле должны выполняться параллельно или последовательно.

Циклы и условия задаются на вкладке Выполнение в диалоговом окне Свойства потока. Все узлы, которые используются в требованиях с условиями или циклами, изображаются на холсте с добавлением специального символа.

Есть 3 способа открыть вкладку Выполнение:

- При помощи меню в верхней части главного диалогового окна:
 1. В меню Инструменты выберите:
Свойства потоков > Выполнение
 2. Перейдите на вкладку Выполнение для работы со сценариями для текущего потока.

- Из потока:
 1. Щелкните правой кнопкой по узлу и выберите **С циклами и условиями**.
 2. Выберите нужный пункт подменю.
- На панели инструментов графики в верхней части главного диалогового окна щелкните по значку свойств потока.

Если вы впервые задаете подробности циклов или условий, выберите на вкладке Выполнение режим выполнения **С циклами и условиями**, затем выберите подвкладку **С условием** или **В цикле**.

Циклы в потоках

При помощи циклов можно автоматизировать повторяющиеся задачи в потоках; вот некоторые возможные примеры:

- Выполнить поток заданное число раз, каждый раз изменяя источник.
- Выполнить поток заданное число раз, каждый раз изменяя значение некоторой переменной.
- Выполнить поток заданное число раз, каждый раз вводя дополнительное поле.
- Построить модель заданное число раз, каждый раз изменяя параметр модели.

Необходимые условия задаются на подвкладке **В цикле** на вкладке выполнения потока. Чтобы увидеть эту подвкладку, выберите режим выполнения **С циклами и условиями**.

Все определенные вами требования выполнения в цикле действуют при запуске потока в режиме **С циклами и условиями**. Вы можете сгенерировать код сценария для требований цикла и вставить этот код в редактор сценариев, выбрав **Вставить...** в нижнем правом углу подвкладки **В цикле**; основная вкладка выполнения изменится - на ней появится режим выполнения **По умолчанию (дополнительный сценарий)** и сценарий в верхней части вкладки. Таким образом можно определить структуру цикла, пользуясь различными опциями диалогового окна циклов; сгенерированный сценарий потом можно дополнительно изменить в редакторе сценариев. Обратите внимание на то, что при нажатии на кнопку **Вставить...** в сгенерированный сценарий будут вставлены также все требования выполнения с условиями.

Чтобы задать цикл:

1. Создайте ключ итерации, чтобы определить основную структуру цикла, выполняемую в потоке. Дополнительную информацию смотрите в разделе Создать ключ итерации.
2. Если нужно, определите одну или несколько переменных итерации. Дополнительную информацию смотрите в разделе Создать переменную итерации.
3. Итерации и любые созданные вами переменные показаны в основной части подвкладки. По умолчанию итерации выполняются и оцениваются в том порядке, в каком показаны; чтобы изменить порядок, переместите итерацию вверх или вниз в списке, для чего щелкните по ней, чтобы выделить, и затем щелкайте по стрелкам вверх и вниз в столбце в правой части подвкладки, чтобы переместить выделенное.

Создание ключа итерации для циклов в потоках

При помощи ключа итерации определяется основная структура цикла, выполняемая в потоке. Например, если вы анализируете продажи автомобилей, можно создать параметр потока *Страна изготовителя* и использовать его как ключ итерации; при выполнении потока на каждой итерации для этого ключа будет задаваться значение другой страны. Для задания ключа служит диалоговое окно Определить ключ итерации.

Чтобы открыть это диалоговое окно, нажмите кнопку **Ключ итерации...** в нижнем левом углу подвкладки циклов или щелкните правой кнопкой мыши по любому узлу потока и выберите **С циклами и условиями > Определить ключ итерации (поля)** или **С циклами и условиями > Определить ключ итерации (значения)**. Если открыть диалоговое окно из потока, некоторые поля могут быть заполнены автоматически, например, имя узла.

Чтобы задать ключ итерации, заполните следующие поля:

При итерации перебирать. Можно выбрать один из следующих вариантов:

- **Параметр потока - Поля.** При помощи этой опции можно создать цикл, в котором все заданные поля поочередно используются как значение для существующего параметра потока.
- **Параметр потока - Значения.** При помощи этой опции можно создать цикл, в котором все заданные значения поочередно используются для существующего параметра потока.
- **Свойство узла - Поля.** При помощи этой опции можно создать цикл, в котором все заданные поля поочередно используются как значение для свойства узла.
- **Свойство узла - Значения.** При помощи этой опции можно создать цикл, в котором все заданные значения поочередно используются для свойства узла.

Что задать. Выберите элемент, значение которого будет задаваться при каждом выполнении цикла. Можно выбрать один из следующих вариантов:

- **Параметр.** Доступно, только если выбрать **Параметр потока - Поля** или **Параметр потока - Значения**. Выберите нужный параметр из списка.
- **Узел.** Доступно, только если выбрать **Свойство узла - Поля** или **Свойство узла - Поля**. Выберите узел, для которого хотите задать цикл. Нажмите кнопку просмотра, чтобы открыть диалоговое окно **Выбрать узел** и выбрать нужный узел; если узлов в этом окне слишком много, можете отфильтровать список, чтобы в нем остались только некоторые типы узлов, для чего выберите одну из следующих категорий: узел источника, узел обработки, узел диаграмм, узел моделирования, узел вывода, узел экспорта или узел применения модели.
- **Свойство.** Доступно, только если выбрать **Свойство узла - Поля** или **Свойство узла - Значения**. Выберите свойство узла из списка.

Используемые поля. Доступно, только если выбрать **Параметр потока - поля** или **Свойство узла - поля**. Выберите одно или несколько полей на узле, которые послужат данными для итерации. Можно выбрать один из следующих вариантов:

- **Узел.** Доступно, только если выбрать **Параметр потока - поля**. Выберите узел, содержащий информацию, для которой вы хотите задать цикл. Нажмите кнопку просмотра, чтобы открыть диалоговое окно **Выбрать узел** и выбрать нужный узел; если узлов в этом окне слишком много, можете отфильтровать список, чтобы в нем остались только некоторые типы узлов, для чего выберите одну из следующих категорий: узел источника, узел обработки, узел диаграмм, узел моделирования, узел вывода, узел экспорта или узел применения модели.
- **Список полей.** Щелкните по кнопке списка в столбце справа, чтобы открыть диалоговое окно **Выбрать поля** и выбрать в нем поля узла, которые послужат данными для итерации. Дополнительную информацию смотрите в разделе “Выбор полей для итераций” на стр. 7.

Используемые значения. Доступно, только если выбрать **Параметр потока - значения** или **Свойство узла - значения**. Выберите одно или несколько значений в выбранном поле, которые будут использоваться как значения итерации. Можно выбрать один из следующих вариантов:

- **Узел.** Доступно, только если выбрать **Параметр потока - значения**. Выберите узел, содержащий информацию, для которой вы хотите задать цикл. Нажмите кнопку просмотра, чтобы открыть диалоговое окно **Выбрать узел** и выбрать нужный узел; если узлов в этом окне слишком много, можете отфильтровать список, чтобы в нем остались только некоторые типы узлов, для чего выберите одну из следующих категорий: узел источника, узел обработки, узел диаграмм, узел моделирования, узел вывода, узел экспорта или узел применения модели.
- **Список полей.** Выберите поле на узле, из которого нужно брать данные для итерации.
- **Список значений.** Нажмите кнопку списка в столбце справа, чтобы открыть диалоговое окно **Выбрать значения** и выбрать в нем значения поля, которые будут перебираться при итерации.

Создание переменной итерации для циклов в потоках

С помощью переменных итерации можно при каждом выполнении цикла изменять значения параметров потока или свойств выбранных узлов в этом потоке. Например, пусть в вашем цикле анализируются данные

о продажах автомобилей и в качестве ключа итерации используется *Страна изготовителя*, и пусть у вас на одной выходной диаграмме показаны продажи по моделям и на другой - информация о выхлопных газах. В таком случае вы можете создать переменные итерации, которыми будут создаваться заголовки для новых графов, например, *Выхлоп шведских автомобилей* и *Продажи японских машин по модели*. Все нужные переменные можно задать в диалоговом окне **Определить переменную итерации**.

Чтобы открыть это диалоговое окно, нажмите кнопку **Переменная итерации...** в нижнем левом углу подвкладки циклов или щелкните правой кнопкой по любому узлу потока и выберите: **С циклами и условиями > Определить переменную итерации**.

Чтобы задать переменную итерации, заполните следующие поля:

Изменить. Выберите тип атрибута, который хотите корректировать. Доступные варианты - **Параметр потока** и **Свойство узла**.

- Если вы выбрали **Параметр потока**, выберите нужный параметр, затем воспользуйтесь одной из описанных ниже опций, если она доступна в вашем потоке, чтобы определить значение параметра, задаваемое на каждой итерации цикла:
 - **Глобальная переменная.** Выберите глобальную переменную, значение которой будет использовано, чтобы задать значение параметра потока.
 - **Ячейка вывода таблицы.** Чтобы задать для параметра потока значение из ячейки табличного вывода, выберите таблицу в списке и введите нужные **Строку** и **Столбец**.
 - **Ввести вручную.** Выберите этот вариант, если нужно вручную вводить значение этого параметра на каждой итерации. Когда вы вернетесь на подвкладку циклов, будет создан новый столбец, в который вы введете нужные тексты.
- Если вы выбрали **Свойство узла**, выберите нужный узел и одно из его свойств, затем задайте значение для этого свойства. При задании нового значения свойства используйте одну из следующих возможностей:
 - **Отдельно.** В качестве значения свойства будет использоваться значение ключа итерации. Дополнительную информацию смотрите в разделе “Создание ключа итерации для циклов в потоках” на стр. 5.
 - **Как префикс для основы.** Значение ключа итерации используется как префикс перед значением, которое вы введете в поле **Основа**.
 - **Как суффикс для основы.** Значение ключа итерации используется как суффикс после значения, которое вы введете в поле **Основа**.

Если вы выбрали вариант с префиксом или суффиксом, появится приглашение ввести также текст в поле **Основа**. Например, пусть у вас значение ключа итерации - *Страна производителя* и вы выбрали опцию **Как префикс для основы**, тогда в это поле можно ввести, например, такой текст: - *продажи по модели*.

Выбор полей для итераций

Создавая итерацию, вы можете выбрать одно или несколько полей в диалоговом окне **Выбрать поля**.

Сортировать по. Доступные поля можно отсортировать для просмотра, выбрав одну из следующих опций:

- **Естественный.** Просмотр полей в порядке их поступления через поток данных на текущий узел.
- **Имя.** Использовать алфавитный порядок сортировки полей для просмотра.
- **Тип.** Просмотр полей, отсортированных по их шкалам измерений. Эта опция полезна при выборе полей с конкретной шкалой измерений.

Выберите поля в списке по одному или сразу несколько, удерживая при их выборе нажатой клавишу Shift или Ctrl. При помощи опций под этим списком можно также выбрать группы полей на основе их шкалы измерений, а также выбрать сразу все поля или отменить выбор всех полей в таблице.

Обратите внимание на то, что доступные для выбора поля фильтруются: показаны только те поля, которые допустимы для выбранного параметра потока или свойства узла. Например, если используется параметр потока со строковым типом хранения, показаны только поля со строковым типом хранения.

Выполнение с условиями в потоках

При помощи выполнения с условиями можно управлять запуском конечных узлов с учетом того, соответствует ли содержимое потока задаваемым вами условиям; вот несколько возможных примеров:

- Управлять запуском узла с учетом полученного значения true или false.
- Определить, что узлы в цикле должны выполняться параллельно или последовательно.

Необходимые условия задаются на подвкладке **С условиями** на вкладке выполнения потока. Чтобы увидеть эту подкладку, выберите режим выполнения **С циклами и условиями**.

Все определенные вами требования выполнения с условиями действуют при запуске потока в режиме **С циклами и условиями**. Вы можете сгенерировать код сценария для требований условного выполнения и вставить этот код в редактор сценариев, выбрав **Вставить...** в нижнем правом углу подвкладки **С условиями**; основная вкладка выполнения изменится - на ней появится режим выполнения **По умолчанию (дополнительный сценарий)** и сценарий в верхней части вкладки. Таким образом можно определить условия, пользуясь различными опциями диалогового окна циклов; сгенерированный сценарий потом можно дополнительно изменить в редакторе сценариев. Обратите внимание на то, что при нажатии на кнопку **Вставить...** в сгенерированный сценарий будут вставлены также все требования циклов.

Чтобы задать условие:

1. В правом столбце на подвкладке **С условиями** нажмите кнопку **Добавить выполняемый оператор** , чтобы открыть диалоговое окно **Оператор выполнения с условиями**. В этом диалоговом окне можно задать условие, только при выполнении которого выполняется узел.
2. В диалоговом окне **Оператор выполнения с условиями** укажите следующее:
 - a. **Узел**. Выберите узел, для которого хотите задать условное выполнение. Нажмите кнопку просмотра, чтобы открыть диалоговое окно **Выбрать узел** и выбрать нужный узел; если узлов в этом окне слишком много, можете отфильтровать список по одной из следующих категорий: узел экспорта, узел диаграмм, узел моделирования или узел вывода.
 - b. **Условие на основе**. Задайте условие, только при выполнении которого выполняется узел. Можно выбрать один из четырех вариантов: **Параметр потока**, **Глобальная переменная**, **Ячейка табличного вывода** или **Всегда true**. Подробности, которые вводятся в нижней части диалогового окна, зависят от выбранного условия.
 - **Параметр потока**. Выберите параметр в списке, затем для этого параметра выберите **операцию**, например, **Более**, **Равен**, **Меньше**, **Между** и так далее. Затем, в зависимости от выбранной операции, введите одно **Значение** или минимальное и максимальное значения.
 - **Глобальная переменная**. Выберите переменную в списке, например, **Среднее**, **Сумма**, **Минимальное значение**, **Максимальное значение**, **Среднеквадратичное отклонение**. Затем выберите **Операцию** и задайте необходимые значения.
 - **Ячейка вывода таблицы**. Выберите табличный узел в списке, затем выберите **строку** и **столбец** таблицы. Затем выберите **Операцию** и задайте необходимые значения.
 - **Всегда true**. Выберите эту опцию, если узел нужно выполнять всегда. Если вы выбрали эту опцию, никакие другие параметры задавать не нужно.
3. Повторите шаги 1 и 2, сколько требуется, чтобы задать все нужные условия. Выбранный узел и условие выполнения этого узла появятся в основной области подвкладки в столбцах **Выполнить узел** и **При этом условии** соответственно.

4. По умолчанию узлы и условия выполняются и оцениваются в том порядке, в каком показаны; чтобы изменить порядок, переместите узел и условие вверх или вниз в списке, для чего щелкните по ним, чтобы выделить, и затем щелкните по стрелкам вверх и вниз в столбце в правой части подвкладки, чтобы переместить выделенное.

Кроме того, в нижней части вкладки С условиями можно задать следующие опции:

- **Оценивать все по порядку.** Выберите эту опцию, чтобы оценивать все условия в том порядке, в каком они показаны на этой подвкладке. После оценки всех условий будут выполнены те узлы, для которых при оценке условия было получено значение "True".
- **Выполнять по очереди.** Эта опция доступна, только если выбрано **Оценивать все по порядку**. Выбор этой опции означает, что если при оценке очередного условия получено значение "True", то сначала выполняется соответствующий узел, а затем оценивается следующее условие.
- **Оценивать до первого совпадения.** Выбор этой опции означает, что выполняется только первый узел, для которого при оценке заданных вами условий возвращено значение "True".

Выполнение и прерывание сценариев

Доступно несколько способов выполнения сценариев. Например, в диалоговом окне потокового или автономного сценария есть кнопка "Запустить этот сценарий", после нажатия которой выполняется полный сценарий:



Рисунок 1. Кнопка Запустить этот сценарий

При нажатии кнопки "Запустить выбранные строки" выполняется одна строка или блок соседних строк, которые вы выбрали в сценарии:



Рисунок 2. Кнопка Запустить выбранные строки

Сценарий можно выполнить с использованием любого из следующих способов:

- Нажмите кнопку "Запустить этот сценарий" или "Запустить выбранные строки" в диалоговом окне потокового или автономного сценария.
- Запустить поток, когда в качестве способа выполнения по умолчанию задано **Запустить этот сценарий**.
- Используйте флаг `-execute` для запуска в интерактивном режиме. Дополнительную информацию смотрите в разделе "Использование аргументов командной строки" на стр. 49.

Примечание: Сценарий супер узла выполняется, когда выполняется сам надузел, а также в диалоговом окне сценария надузла выбрана опция **Запустить этот сценарий**.

Прерывание выполнения сценария

В диалоговом окне потокового сценария красная кнопка остановки активируется во время выполнения сценария. При нажатии этой кнопки прерывается выполнение сценария и любого текущего потока.

Найти и заменить

Диалоговое окно Найти/заменить доступно там, где вы редактируете текст сценария или выражения, в том числе в редакторе сценариев, или при определении шаблонов на узле Отчет. При изменении текста в любой из этих областей нажмите комбинацию клавиш **Ctrl+F** для перехода в диалоговое окно изменений и убедитесь, что указатель мыши сфокусирован в области текста. Например, при работе на узле Заполнитель можно получить доступ к диалоговому окну из любой текстовой области на вкладке Параметры или из поля текста в построителе выражений.

1. При указателе в области текста нажмите клавиши **Ctrl+F** для доступа к диалоговому окну Найти/заменить.
2. Введите текст для поиска или выберите из раскрывающегося списка элементы недавнего поиска.
3. Введите текст для замещения, если такой есть.
4. Нажмите кнопку **Найти далее** для запуска поиска.
5. Нажмите кнопку **Заменить** для замены текущего выбранного фрагмента или кнопку **Заменить все** для замены всех или выбранных экземпляров.
6. Диалоговое окно закрывается после каждой операции. Находясь в любой области текста, нажмите **F3** для повторения последней операции поиска или **Ctrl+F** для повторного доступа к диалоговому окну.

Опции поиска

Учитывать регистр. Задаёт, учитывается ли при поиске регистр символов; например, совпадает ли *myvar* с *myVar*. Текст замещения всегда вставляется так, как он введен, независимо от этого параметра.

Только слова целиком. Задаёт, учитывает ли операция поиска совпадения текста внутри слов. При выборе этой опции поиск для *spider* не найдет соответствий с вариантами *spiderman* или *spider-man*.

Регулярные выражения. Задаёт, используется ли синтаксис регулярных выражений (смотрите следующий раздел). При выборе этой опции отключается опция **Только слова целиком** и ее значение игнорируется.

Только выбранный текст. Управляет областью выполнения поиска при использовании опции **Заменить все**.

Синтаксис регулярных выражений

Регулярные выражения позволяют искать специальные символы, такие как символы табуляции или перехода на новую строку, классы или диапазоны символов, такие как от *a* до *d* при любых цифровых или нецифровых параметрах и граничные положения, такие как начало или конец строки. Шаблон регулярного выражения описывает структуру строки, которую выражение попытается найти во входной строке. Поддерживаются следующие типы конструкций регулярного выражения.

Таблица 1. Совпадения символов

Символы	Совпадает
x	Символ x
\\	Символ обратной дробной черты
\\0n	Символ с восьмеричным значением 0n (0 <= n <= 7)
\\0nn	Символ с восьмеричным значением 0nn (0 <= n <= 7)
\\0mnn	Символ с восьмеричным значением (0 <= m <= 3, 0 <= n <= 7)
\\xhh	Символ с шестнадцатиричным значением 0xhh
\\uhhhh	Символ с шестнадцатиричным значением 0xhhhh
\\t	Символ табуляции ('\\u0009')
\\n	Символ новой строки (начало строки) ('\\u000A')
\\r	Символ возврата каретки ('\\u000D')

Таблица 1. Совпадения символов (продолжение)

Символы	Совпадает
\f	Символ перехода к новой странице ("u000C")
\a	Символ предупреждения (звонок) ("u0007")
\e	Символ перехода ESC ("u001B")
\cx	Управляющий символ, соответствующий x

Таблица 2. Совпадения классов символов

Классы символов	Совпадает
[abc]	a, b или c (простой класс)
[^abc]	Любой символ, кроме a, b или c (вычитание)
[a-zA-Z]	От a до z или от A до Z, включительно (диапазон)
[a-d[m-p]]	От a до d или от m до p (объединение). Альтернативно это можно задать как [a-dm-p]
[a-z&&[def]]	От a до z и d, e или f (пересечение)
[a-z&&[^bc]]	От a до z, кроме b и c (вычитание). Альтернативно это можно задать как [ad-z]
[a-z&&[^m-p]]	От a до z, но не от m до p (вычитание). Альтернативно это можно задать как [a-lq-z]

Таблица 3. Предварительно определенные классы символов

Предварительно определенные классы символов	Совпадает
.	Любой символ (разделители строк могут и совпадать, и не совпадать)
\d	Любая цифра: [0-9]
\D	Не цифра: [^0-9]
\s	Пробельный символ: [\t\n\r0B\f\r]
\S	Не пробельный символ: [^\s]
\w	Символ слова: [a-zA-Z_0-9]
\W	Символ не слова: [^\w]

Таблица 4. Совпадения границ

Обнаружители совпадений границ	Совпадает
^	Начало строки
\$	Конец строки
\b	Граница слова
\B	Граница не слова
\A	Начало ввода
\Z	Конец ввода кроме заключительного разделителя, если он есть
\z	Конец ввода

Дополнительную информацию об использовании регулярных выражений и некоторые примеры смотрите в разделе <http://www.ibm.com/developerworks/java/tutorials/j-introjava2/section9.html>.

Примеры

Следующий код ищет и находит три цифры в начале строки:

```
^[0-9]{3}
```

Следующий код ищет и находит три цифры в конце строки:

```
[0-9]{3}$
```

Глава 2. Язык сценариев

Обзор языка сценария

Возможность написания сценариев для IBM SPSS Modeler позволяет вам создавать сценарии, работающие в пользовательском интерфейсе SPSS Modeler, манипулирующие объектами вывода и запускающие командный синтаксис. Запустить сценарии можно непосредственно изнутри SPSS Modeler.

Сценарии в IBM SPSS Modeler пишутся на языке сценариев Python. Используемая IBM SPSS Modeler реализация Python на основе Java называется Jython. Язык написания сценариев состоит из следующих возможностей:

- Формат для ссылки на узлы, потоки, проекты, выходные данные и другие объекты IBM SPSS Modeler.
- Набор операторов сценариев или команд, которые можно задать для манипулирования этими объектами.
- Язык выражений сценариев для задания значений переменных, параметров и других объектов.
- Поддержка комментариев, продолжений и блоков литерального текста.

В следующих разделах описывается язык сценариев, его конкретная реализация Jython и основной синтаксис для начала работы со сценариями в IBM SPSS Modeler. Информация о конкретных свойствах и командах представлена в последующем разделе.

Python и Jython

Jython - это реализация языка сценариев Python, написанная на языке Java и интегрированная с платформой Java. Python - богатый объектно-ориентированный язык сценариев. Jython полезен тем, что предлагает повышающие производительность труда программистов возможности развитого языка сценариев, но, в отличие от языка Python, работает в любой среде, поддерживающей виртуальную Java-машину (JVM). Это значит, что при написании программ доступны библиотеки Java для JVM. Работая на языке Jython, можно пользоваться этим преимуществом и вместе с тем синтаксисом и большинством возможностей языка Python

Как язык сценариев, Python (и его реализация Jython) удобен для изучения, эффективен при программировании и дает возможность создать работающую программу при минимальных требованиях к структуре. Код можно вводить интерактивно, по одной строке. Python работает как интерпретатор языка сценариев, не требуя стадии компиляции, которая необходима в Java. Программы Python - это просто текстовые файлы, которые интерпретируются по мере ввода (после синтаксического анализа для выявления ошибок). Простые выражения, такие как задание значений, и более сложные действия, такие как определение функций, выполняются немедленно, и их результаты доступны для использования. Все изменения, вносимые в код, можно быстро протестировать. Однако у интерпретаторов сценариев есть свои недостатки. Так, использование переменной до ее определения не считается ошибкой для компилятора и обнаруживается только при попытке выполнить оператор, содержащий такую переменную. В этом случае программу можно редактировать и запускать для исправления ошибок.

Python рассматривает все как объекты, например, объекты данных и объекты кода. Со всеми объектами можно работать, редактируя строки кода. Некоторые типы выбора, такие как числа и строки, удобнее считать не объектами, а значениями, и это поддерживается в Python. Поддерживается одно пустое значение. Ему присвоено зарезервированное имя None.

Более подробное введение в сценарии Python и Jython, с примерами сценариев, есть в разделах www.ibm.com/developerworks/java/tutorials/j-jython1 и www.ibm.com/developerworks/java/tutorials/j-jython2.

Сценарий Python

Это руководство по языку написания сценариев Python представляет собой введение в компоненты, которые чаще всего используются для сценариев в IBM SPSS Modeler, а также в концепции и основы программирования. Вы получите достаточные знания для начала разработки собственных сценариев Python и их использования в IBM SPSS Modeler.

Операции

Назначение производится с помощью знака равенства (=). Например, чтобы назначить значение "3" переменной с именем "x", используется следующий оператор:

```
x = 3
```

Знак равенства используется также для назначения переменной строковых значений. Например, чтобы назначить значение "строковое значение" переменной "y", используется следующий оператор:

```
y = "строковое значение"
```

В следующей таблице перечисляются общие операции с числами и операции сравнения, а также их описания.

Таблица 5. Общие операции с числами и операции сравнения

Операция	Описание
$x < y$	x меньше y?
$x > y$	x больше y?
$x \leq y$	x меньше или равно y?
$x \geq y$	x больше или равно y?
$x == y$	x равно y?
$x != y$	x не равно y?
$x \lt;> y$	x не равно y?
$x + y$	Сложить y и x
$x - y$	Вычесть y из x
$x * y$	Умножить x на y
x / y	Разделить x на y
$x ** y$	Возвести x в степень y

Списки

Перечисляет последовательность элементов. Список может содержать любое количество элементов, и эти элементы списка могут быть любых типов объектов. Списки можно рассматривать как массивы. Число элементов в списке может увеличиваться или уменьшаться при добавлении, удалении или замене элементов.

Примеры

<code>[]</code>	Любой пустой список.
<code>[1]</code>	Список с одним элементом, целое число.
<code>["Mike", 10, "Don", 20]</code>	Список с четырьмя элементами - двумя строковыми и двумя целочисленными.
<code>[[], [7], [8, 9]]</code>	Список списков. Каждый подсписок - это или пустой список, или список целочисленных элементов.
<code>x = 7; y = 2; z = 3; [1, x, y, x + y]</code>	Список целых чисел. В этом примере демонстрируется использование переменных и выражений.

Список можно назначить переменной, например:

```
mylist1 = ["один", "два", "три"]
```

Затем вы можете обратиться к любому конкретному элементу в этом списке, например:

```
mylist[0]
```

При этом вывод будет следующим:

```
один
```

Число в прямых скобках ([]) называют *индексом*; оно указывает на конкретный элемент в списке. Элементы в списке индексируются, начиная с нуля.

Вы можете выбрать также диапазон элементов в списке; такой диапазон называется *срезом*. Например, `x[1:3]` выбирает второй и третий элемент `x`. Обозначенный конечный индекс - это индекс следующего элемента после последнего выбранного.

Строки

Строка - это неизменная последовательность символов, которая рассматривается как значение. Строки поддерживают все функции и операторы с неизменяемыми последовательностями, которые дают на выходе новую строку. Например, `"abcdef"[1:4]` дает в результате `"bcd"`.

В Python символы представлены строками единичной длины.

Строковые литералы определяются использованием одинарных или тройных кавычек. Строки, определенные одинарными кавычками, не могут распространяться на несколько строк кода, а строки с тройными кавычками могут. Строка может заключаться в одинарные (') или в двойные кавычки ("). Между символами кавычек могут содержаться другие символы кавычек, не обозначающие переход, или символы кавычек с переходом, перед которыми должен стоять знак обратной дробной черты (\).

Примеры

```
"Это строка"  
'Это тоже строка'  
"It's a string"  
'Эта книга называется "Руководство по автоматизации и сценариям Python".'  
"Это кавычка, заданная с использованием эскейп-символа (\") в закавыченной строке"
```

Несколько строк, разделенных пробельным символом, автоматически объединяются синтаксическим анализатором Python. Так проще вводить длинные строки и совместно использовать разные типы кавычек в одной строке, например:

```
"В этой строке используется ', а " 'в этой строке используется ".'
```

Это приводит к следующему выводу:

```
В этой строке используется ', а в этой строке используется ".
```

Строки поддерживают несколько полезных методов. Некоторые из них приведены в следующей таблице.

Таблица 6. Строковые методы

Метод	Использование
<code>s.capitalize()</code>	Переводит в верхний регистр начальные символы <code>s</code>
<code>s.count(ss {,start {,end}})</code>	Считает количество <code>ss</code> в <code>s[start:end]</code>
<code>s.startswith(str {, start {, end}})</code> <code>s.endswith(str {, start {, end}})</code>	Проверяет, начинается ли <code>s</code> с <code>str</code> Проверяет, заканчивается ли <code>s</code> на <code>str</code>
<code>s.expandtabs({size})</code>	Заменяет символы табуляции на пробелы, по умолчанию значение <code>size</code> - 8

Таблица 6. Строковые методы (продолжение)

Метод	Использование
s.find(str {, start {, end}}) s.rfind(str {, start {, end}})	Ищет первое вхождение str в s; если не находит, возвращает -1. rfind ищет справа налево.
s.index(str {, start {, end}}) s.rindex(str {, start {, end}})	Находит первое вхождение str в s; если не находит, выводит ошибку ValueError. rindex ищет справа налево.
s.isalnum	Проверяет, состоит ли строка из букв и цифр
s.isalpha	Проверяет, состоит ли строка из букв
s.isnum	Проверяет, состоит ли строка из цифр
s.isupper	Проверяет, используется ли в строке только верхний регистр
s.islower	Проверяет, используется ли в строке только нижний регистр
s.isspace	Проверяет, состоит ли строка только из пробельных символов
s.istitle	Проверяет, состоит ли строка из последовательности алфавитно-цифровых строк, начинающихся символом капители
s.lower() s.upper() s.swapcase() s.title()	Преобразует все в нижний регистр Преобразует все в верхний регистр Преобразует все в противоположный регистр Преобразует первую букву каждого слова в верхний регистр, а прочие буквы - в нижний регистр
s.join(seq)	Объединяет строки в seq, используя s как разделитель
s.splitlines({keep})	Разбивает s на строки; если значение keep - true, сохраняет переход на новые строки
s.split({sep {, max}})	Разделяет s на "слова", используя sep (по умолчанию sep - это пробельный символ); число повторений действия - max
s.ljust(width) s.rjust(width) s.center(width) s.zfill(width)	Выравнивает строку по левому краю в поле ширины width Выравнивает строку по правому краю в поле ширины width Выравнивает строку по центру в поле ширины width Дополняет символами 0.
s.lstrip() s.rstrip() s.strip()	Удаляет пробельные символы в начале Удаляет пробельные символы в конце Удаляет начальные и конечные пробельные символы
s.translate(str {, delc})	Переводит s с помощью таблицы, после удаления всех символов delc. str должна быть строкой длины == 256.
s.replace(old, new {, max})	Заменяет все или max строк old на строку new

Комментарии

Комментарии (remarks) вводятся в текст кода с помощью символа решетки (#). Весь текст в одной строке после этого знака рассматривается как комментарий и игнорируется. Комментарий может начинаться в любом столбце. В следующем примере показано использование комментариев:

```
#Программа HelloWorld - одна из самых простых программ
print 'Hello World' # печатает строку Hello World
```

Синтаксис операторов

Синтаксис операторов Python очень прост. В общем случае каждая исходная строка - это один оператор. Исключения составляют операторы `expression` и `assignment`, каждый оператор вводится именем ключевого слова, таким как `if` или `for`. Пустые строки или строки комментариев можно вставить в произвольном месте между любыми операторами в коде. Если в строке располагается несколько операторов, их нужно разделить точкой с запятой (;).

Очень длинные операторы могут располагаться на нескольких строках. В этом случае оператор, который нужно продолжить на следующую строку должен заканчиваться знаком обратной дробной черты на данной строке (`\`), например:

```
x = "Длиииииииииииииииииииинная строка" + \  
    "еще одна длиииииииииииииииииииинная строка"
```

Когда некоторая структура кода заключена в скобки (`()`), квадратные скобки (`[]`) или фигурные скобки (`{}`), оператор можно продолжить на новую строку после любой запятой, не используя знака обратной дробной черты, например:

```
x = (1, 2, 3, "привет",  
    "пока", 4, 5, 6)
```

Идентификаторы

Идентификаторы используются для именования переменных, функций, классов и ключевых слов. Идентификаторы могут быть любой длины, но должны начинаться или с буквы, или с символа подчеркивания (`_`). Имена, начинающиеся с подчеркивания, обычно резервируются для внутренних или частных имен. После первого символа идентификатор может содержать любую комбинацию произвольной длины букв, чисел от 0 до 9 и символов подчеркивания.

В Python есть несколько зарезервированных слов, которые нельзя использовать в качестве имен переменных, функций или классов. Они попадают в следующие категории:

- **Вводные для операторов:** `assert`, `break`, `class`, `continue`, `def`, `del`, `elif`, `else`, `except`, `exec`, `finally`, `for`, `from`, `global`, `if`, `import`, `pass`, `print`, `raise`, `return`, `try` и `while`
- **Вводные параметров:** `as`, `import` и `in`
- **Операции:** `and`, `in`, `is`, `lambda`, `not` и `or`

Неправильное использование ключевых слов обычно приводит к ошибке `SyntaxError`.

Блоки кода

Блоки кода - это группы операторов, используемые там, где предполагаются отдельные операторы. Блоки кода могут следовать после любого из следующих операторов: `if`, `elif`, `else`, `for`, `while`, `try`, `except`, `def` и `class`. Эти операторы вводят блоки кода с помощью символа двоеточия (`:`), например:

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

Отступ используется для разделения блоков кода (в отличие от фигурных скобок, используемых в Java). Все строки в блоке должны иметь одинаковый отступ. Это связано с тем, что изменение отступа обозначает конец блока кода. Обычно производится сдвиг по четыре пробела на каждый уровень. Рекомендуется использовать для отступа строки именно пробелы, а не знаки табуляции. Пробелы и табуляции нельзя смешивать. Строки самого внешнего блока в модуле должны начинаться в первом столбце, иначе возникнет ошибка `SyntaxError`.

Операторы, составляющие блок кода (и следующие за двоеточием), могут располагаться также на одной строке и разделяться точками с запятой, например:

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

Передача аргументов в сценарий

Передача аргументов в сценарий полезна тем, что сценарий можно использовать повторно без изменений. Аргументы, переданные в командной строке, передаются как значения в списке `sys.argv`. Количество передаваемых значений можно получить с использованием команды `len(sys.argv)`. Например:

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

В этом примере команда `import` импортирует весь класс `sys`, поэтому можно использовать методы, существующие для этого класса, такие как `argv`.

Сценарий из этого примера можно вызвать, используя следующую строку:

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

Будет получен следующий вывод:

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

Примеры

Ключевое слово `print` выводит непосредственно следующие за ним аргументы на печать. Если после оператора стоит запятая, переход на новую строку в выводе не происходит. Например:

```
print "Показано использование запятой",
print " в конце оператора print."
```

При этом вывод будет следующим:

```
Показано использование запятой в конце оператора print.
```

Оператор `for` используется для итераций по блоку кода. Например:

```
mylist1 = ["один", "два", "три"]
for lv in mylist1:
    print lv
    continue
```

В этом примере списку `mylist1` назначаются три строки. Затем элементы этого списка выводятся на печать по одному на строке. При этом вывод будет следующим:

```
один
два
три
```

В этом примере итератор `lv` принимает значения каждого элемента списка `mylist1` по порядку, как цикл реализует блок кода для каждого элемента. Итератор может быть любым допустимым идентификатором произвольной длины.

Оператор `if` - это условный оператор. Он вычисляет некоторое условие и возвращает значение `true` или `false` в зависимости от результата вычислений. Например:

```

mylist1 = ["один", "два", "три"]
for lv in mylist1:
    if lv == "два"
        print "Значение lv - ", lv
    else
        print "Значение lv - не два, а ", lv
        continue

```

В этом примере вычисляется значение итератора lv. Если значение lv равно два, возвращается не та строка, которая возвращается в случаях, когда значение lv не равно два. Это приводит к следующему выводу:

```

Значение lv - не два, а один
Значение lv - два
Значение lv - не два, а три

```

Математические методы

Из модуля math можно получить доступ к полезным математическим методам. Некоторые из них приведены в следующей таблице. Если не указано иное, все значения возвращаются в виде чисел с плавающей запятой.

Таблица 7. Математические методы

Метод	Использование
math.ceil(x)	Возвращает в виде значения с плавающей запятой потолочное значение x, то есть наименьшее целое число, большее или равное x
math.copysign(x, y)	Возвращает x со знаком y. copysign(1, -0.0) возвращает -1
math.fabs(x)	Возвращает абсолютное значение x
math.factorial(x)	Возвращает факториал x. Если x отрицательное или не целое, возникает ошибка ValueError.
math.floor(x)	Возвращает в виде значения с плавающей запятой поддерживающее значение x, то есть максимальное целое, меньшее или равное x
math.frexp(x)	Возвращает мантиссу (m) и экспоненту (e) числа x как пару чисел (m, e). m - это число с плавающей запятой, а e - целое, так что выполняется точное равенство $x == m * 2^{**}e$. Если x равно нулю, возвращается (0.0, 0), в противном случае $0.5 <= \text{abs}(m) < 1$.
math.fsum(iterable)	Возвращает в формате числа с плавающей запятой точную сумму значений в iterable
math.isinf(x)	Проверяет переполнение числа с плавающей запятой x (бесконечно большое отрицательное или положительное)
math.isnan(x)	Проверяет, принимает ли x значение NaN (not a number, не число)
math.ldexp(x, i)	Возвращает $x * (2^{**}i)$. По сути это функция, обратная к frexp.
math.modf(x)	Возвращает дробную и целую часть числа x. Оба полученных числа - с плавающей запятой, и у обоих значений знак x.
math.trunc(x)	Возвращает значение Real числа x, усеченного до значения Integral.
math.exp(x)	Возвращает $e^{**}x$
math.log(x[, base])	Возвращает значение логарифма x по данному основанию base. Если значение base не задано, возвращается значение натурального логарифма x.

Таблица 7. Математические методы (продолжение)

Метод	Использование
<code>math.log1p(x)</code>	Возвращает натуральный логарифм $1+x$ (основание e)
<code>math.log10(x)</code>	Возвращает десятичный логарифм x
<code>math.pow(x, y)</code>	Возвращает x в степени y . <code>pow(1.0, x)</code> и <code>pow(x, 0.0)</code> всегда возвращают 1, даже если x равно нулю или NaN.
<code>math.sqrt(x)</code>	Возвращает квадратный корень из x

В дополнение к математическим функциям есть несколько полезных тригонометрических методов. Эти методы показаны в следующей таблице.

Таблица 8. Тригонометрические методы

Метод	Использование
<code>math.acos(x)</code>	Возвращает арккосинус x в радианах
<code>math.asin(x)</code>	Возвращает арксинус x в радианах
<code>math.atan(x)</code>	Возвращает арктангенс x в радианах
<code>math.atan2(y, x)</code>	Возвращает арктангенс $\text{atan}(y / x)$ в радианах.
<code>math.cos(x)</code>	Возвращает косинус x .
<code>math.hypot(x, y)</code>	Возвращает евклидову норму $\text{sqrt}(x*x + y*y)$. Это длина вектора из начала координат в точку (x, y) .
<code>math.sin(x)</code>	Возвращает синус x
<code>math.tan(x)</code>	Возвращает тангенс x
<code>math.degrees(x)</code>	Преобразует угол x из радиан в градусы
<code>math.radians(x)</code>	Преобразует угол x из градусов в радианы
<code>math.acosh(x)</code>	Возвращает обратный гиперболический косинус x
<code>math.asinh(x)</code>	Возвращает обратный гиперболический синус x
<code>math.atanh(x)</code>	Возвращает обратный гиперболический тангенс x
<code>math.cosh(x)</code>	Возвращает гиперболический косинус x
<code>math.sinh(x)</code>	Возвращает гиперболический синус x
<code>math.tanh(x)</code>	Возвращает гиперболический тангенс x

Есть также две математические константы. Значение `math.pi` - это число пи. Значение `math.e` - это основание натуральных логарифмов e .

Использование символов не из кодового набора ASCII

Чтобы использовать символы не из кодового набора ASCII, в Python требуется явное кодирование и декодирование строк в Unicode. В IBM SPSS Modeler сценарии Python считаются написанными в формате UTF-8, представляющем собой стандарт кодировки Unicode с поддержкой символов не из кодового набора ASCII. Следующий сценарий пройдет компиляцию, поскольку компилятор Python был настроен на UTF-8 SPSS Modeler.

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

Однако созданный в результате узел будет иметь неправильную метку.



ãfã, 'ãf^ãf ãf%ãf%

Рисунок 3. Метка узла, содержащая символы не из кодового набора ASCII, которые выводятся неправильно

Неправильная метка - результат преобразования строкового литерала в строку ASCII, выполненного Python.

Python разрешает задавать строковые литералы Unicode, добавляя перед литералом префикс u:

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

Здесь создается строка Unicode, и метка будет выведена правильно.



テストノード

Рисунок 4. Метка узла, содержащая символы не из кодового набора ASCII, которые выводятся правильно

Использование Python и Unicode - большая тема, выходящая за рамки этого документа. Подробное обсуждение этой темы имеется во многих книгах и сетевых ресурсах.

Объектно-ориентированное программирование

Объектно-ориентированное программирование основано на идее создания модели основной проблемы в ваших программах. Объектно-ориентированное программирование сокращает количество программных ошибок и способствует повторному использованию кода. Python - это объектно-ориентированный язык. У определенных в Python объектов есть следующие характеристики:

- **Тождество.** Каждый объект должен быть индивидуальным, и должна существовать возможность проверки этого. Для этой цели существуют проверки `is` и `is not`.
- **Состояние.** У каждого объекта должна быть возможность сохранения состояния. Для этой цели существуют атрибуты, такие как поля и переменные экземпляров.
- **Поведение.** Для каждого объекта должна существовать возможность изменения состояния. Для этой цели существуют методы.

Python включает в себя следующие возможности поддержки объектно-ориентированного программирования:

- **Создание объектов на основе классов.** Классы - это шаблоны для создания объектов. Объекты - это структуры данных со связанным поведением.
- **Наследование с полиморфизмом.** Python поддерживает и одиночное, и множественное наследование. Методы всех экземпляров Python полиморфичны и могут быть перезаписаны подклассами.

- **Инкапсуляция с сокрытием данных.** Python допускает сокрытие атрибутов. Получить доступ к скрытым атрибутам извне класса можно только через методы класса. Классы реализуют методы для изменения данных.

Определение класса

В классе Python можно определять и переменные, и методы. В отличие от Java, в Python можно определить любое количество доступных классов на файл источника (или *модуль*). Таким образом, модуль в Python можно рассматривать аналогично пакету в Java.

В Python классы определяются с использованием оператора `class`. Оператор `class` выглядит следующим образом:

```
имя класса (надклассы): оператор
```

или

```
имя класса (надклассы):  
    назначение  
    .  
    .  
    функция  
    .  
    .
```

При определении класса есть возможность задать несколько операторов *assignment* или не задавать их вовсе. Эти операторы создают атрибуты класса, которые совместно используются всеми экземплярами класса. Можно задать также несколько определений *function* или не задавать их вовсе. Эти определения функций создают методы. Список надклассов не обязателен.

Имя класса должно быть уникальным в одной области действия, то есть в модуле, функции или классе. Для ссылки на один класс можно определить несколько переменных.

Создание экземпляра класса

Классы используются для содержания атрибутов класса (или атрибутов совместного использования) или для создания экземпляров классов. Чтобы создать экземпляр класса, класс вызывается так же, как функция. Например, рассмотрим следующий класс:

```
class MyClass:  
    pass
```

Здесь используется оператор `pass`, поскольку для завершения определения класса нужен оператор, но никакое действие программно не требуется.

Следующий оператор создает экземпляр класса `MyClass`:

```
x = MyClass()
```

Добавление атрибутов к экземпляру класса

В отличие от Java, в Python клиенты могут добавить атрибуты к экземпляру класса. Изменяется только один экземпляр. Например, чтобы добавить атрибуты к экземпляру `x`, задайте новые значения для этого экземпляра:

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

Определение атрибутов классов и методов

Любая переменная, связанная в класс, - это *атрибут класса*. Любая определенная в классе функция - это *метод*. Методы получают экземпляр класса, условно называемый `self`, в качестве первого аргумента. Например, для определения нескольких атрибутов и методов класса можно использовать следующий код:

```
class MyClass
    attr1 = 10      #атрибуты класса
    attr2 = "hello"

    def method1(self):
        print MyClass.attr1  #ссылка на атрибут класса

    def method2(self):
        print MyClass.attr2  #ссылка на атрибут класса

    def method3(self, text):
        self.text = text      #атрибут экземпляра
        print text, self.text #напечатать мой аргумент и мой атрибут

    method4 = method3  #создать алиас для method3
```

Внутри класса все ссылки на атрибуты класса необходимо специфицировать с помощью имени класса; например, `MyClass.attr1`. Все ссылки на атрибуты экземпляра должны специфицироваться переменной `self`; например, `self.text`. Вне класса все ссылки на атрибуты класса должны специфицироваться именем класса (например, `MyClass.attr1`) или экземпляром класса (например, `x.attr1`, где `x` - это экземпляр класса). Вне класса все ссылки на переменные экземпляра должны специфицироваться экземпляром класса; например, `x.text`.

Скрытые переменные

Данные можно скрыть, создав *Приватные* переменные. К приватным переменным может обратиться только сам класс. Если вы объявляете имена в виде `__xxx` или `__xxx_yyy`, то есть с двумя начальными знаками подчеркивания, синтаксический анализатор Python автоматически добавит имя класса к объявленным именам, создавая скрытые переменные, например:

```
class MyClass:
    __attr = 10  #атрибут приватного класса

    def method1(self):
        pass

    def method2(self, p1, p2):
        pass

    def __privateMethod(self, text):
        self.__text = text  #приватный атрибут
```

В отличие от Java, в Python все указания на переменные экземпляра должны быть специфицированы с `self`; использование `this` не применяется.

Наследование

Возможность наследования от классов - это фундаментальное свойство объектно-ориентированного программирования. Python поддерживает и одиночное, и множественное наследование. *Одиночное наследование* означает, что может быть только один надкласс. *Множественное наследование* означает, что может быть несколько надклассов.

Наследование реализуется определением других классов в качестве подклассов. Надклассами может быть любое число классов Python. В Python, реализации Python, прямо или косвенно можно наследовать только от одного класса Java. Представление надкласса не требуется.

Любой атрибут или метод надкласса содержится также в любом подклассе и может использоваться самим классом или любым клиентом, если атрибут или метод не скрыт. Любой экземпляр подкласса можно использовать там, где допустимо использование и экземпляра надкласса; это пример *полиморфизма*. Эти возможности допускают повторное использование и облегчают работу с расширением.

Пример

```
class Class1: pass    #нет наследования

class Class2: pass

class Class3(Class1): pass    #одиночное наследование

class Class4(Class3, Class2): pass    #множественное наследование
```

Глава 3. Сценарии в IBM SPSS Modeler

Типы сценариев

В IBM SPSS Modeler есть три типа сценариев:

- *Сценарии потока* используются для управления выполнением одного потока и хранятся в этом потоке.
- *Сценарии надузлов* используются для управления поведением надузлов.
- *Автономные сценарии и сценарии сеансов* можно использовать для координирования выполнения по нескольким разным потокам.

Для использования в сценариях в IBM SPSS Modeler доступны различные методы, с помощью которых можно получить доступ к широкому диапазону функциональных возможностей SPSS Modeler. Эти методы используются в Глава 4, “API сценариев”, на стр. 35 для создания также более расширенных функций.

Потоки, потоки надузлов и диаграммы

В большинстве случаев термин *поток* имеет один и тот же смысл как в отношении потока, загруженного из файла, так и потока, используемого внутри надузла. Обычно он означает собрание соединенных друг с другом узлов, которые можно выполнять. Но в сценариях поддерживаются не все операции и не в любом месте, так что автор сценария должен знать, какую разновидность потока использует.

Потоки

Поток - это основной тип документа IBM SPSS Modeler. Его можно сохранять, загружать, редактировать и выполнять. Кроме того, с потоками может связываться такая информация, как параметры, глобальные переменные, сценарии и другое.

Потоки надузлов

Поток надузла - это тип потока, используемый внутри надузла. Подобно обычному потоку, он состоит из соединенных между собой узлов. У потоков надузла есть ряд отличий от обычного потока:

- Параметры и сценарии связываются не с самим потоком надузла, а с надузлом, которому принадлежит поток надузла.
- Потоки надузла содержат дополнительные узлы входящих и исходящих соединений, зависящие от типа надузла. Эти узлы соединений служат для передачи информации в поток надузла и из потока надузла; такие узлы создаются автоматически при создании надузла.

Диаграммы

Термин *диаграмма* охватывает функции, которые поддерживаются и обычными потоками, и потоками надузла, такими как добавление и удаление узлов и изменение соединений между узлами.

Выполнение потока

В следующем примере запускаются все исполняемые узлы в потоке, и это простейший тип сценария потока:
`modeler.script.stream().runAll(None)`

В следующем примере также запускаются все исполняемые узлы в потоке:

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

В этом примере поток хранится в переменной, называемой `stream`. Хранение потока в переменной полезно, так как сценарий обычно используется для изменения или потока, или узлов в потоке. Создание переменной, хранящей поток, приводит к более лаконичному сценарию.

Контекст сценариев

Модуль `modeler.script` предоставляет контекст, в котором выполняется сценарий. Этот модуль автоматически импортируется в сценарий SPSS Modeler во время выполнения. Этот модуль определяет четыре функции, обеспечивающие сценарию доступ в его среде выполнения:

- Функция `session()` возвращает сеанс для сценария. Сеанс определяет такую информацию, как локаль и механизм обработки SPSS Modeler (или локальный процесс, или сетевой процесс сервер SPSS Modeler), используемые для запуска любых потоков.
- Функцию `stream()` можно использовать со сценариями потоков и надузлов. Эта функция возвращает поток, который владеет исполняемым сценарием потока или сценарием надузла.
- Функцию `diagram()` можно использовать со сценариями надузлов. Эта функция возвращает диаграмму в надузле. Для остальных типов сценариев возвращаемое значение то же, что у функции `stream()`.
- Функцию `supernode()` можно использовать со сценариями надузлов. Эта функция возвращает надузел, который владеет исполняемым сценарием.

Эти четыре функции и их выходы сведены в следующей таблице.

Таблица 9. Сводка функций `modeler.script`

Тип сценария	<code>session()</code>	<code>stream()</code>	<code>diagram()</code>	<code>supernode()</code>
Локальный режим	Возвращает сеанс	Возвращает текущий управляемый поток на момент вызова сценария (например, поток, прошедший через опцию <code>-stream</code> пакетного режима) или значение <code>None</code> .	То же, что и <code>stream()</code>	Неприменимо
Поток	Возвращает сеанс	Возвращает поток	То же, что и <code>stream()</code>	Неприменимо
Надузел	Возвращает сеанс	Возвращает поток	Возвращает поток надузла	Возвращает надузел

Модуль `modeler.script` определяет также способ завершения сценария при помощи кода выхода. Функция `exit(код-выхода)` останавливает выполнение сценария и возвращает предоставленный целочисленный код выхода.

Один из определенных для потока методов - это `runAll(List)`. Этот метод запускает все выполняемые узлы. Все модели и выводы, генерируемые при выполнении узлов, добавляются к предоставленному списку.

Обычно при выполнении потока генерируются такие выводы, как модели, графики и другие объекты. Для захвата этих выводов сценарий может использовать переменную, инициализированную для списка, например:

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

Когда выполнение закончено, ко всем сгенерированным объектам можно получить доступ в списке `results`.

Ссылки на существующие узлы

Часто поток строится предварительно с использованием некоторых параметров, которые нужно изменить до выполнения узла. Изменение этих параметров включает в себя следующие задачи:

1. Обнаружение узлов в соответствующем потоке.
2. Изменение параметров узла и/или потока.

Поиск узлов

Потоки предоставляют несколько способов обнаружения существующего узла. Соответствующие методы сведены в следующей таблице.

Таблица 10. Методы для обнаружения существующего узла

Метод	Возвращаемый тип	Описание
<code>s.findAll(type, label)</code>	Набор	Возвращает список всех узлов с заданным типом и меткой. И у типа, и у метки может быть значение <code>None</code> , в этом случае используется другой параметр.
<code>s.findAll(filter, recursive)</code>	Набор	Возвращает собрание всех узлов, которые приняты заданным фильтром. Если для рекурсивного флага задано значение <code>True</code> , поиск проводится и для всех надузлов в заданном потоке.
<code>s.findById(id)</code>	Узел	Возвращает узел с указанным значением ID или значение <code>None</code> , если такого узла не существует. Поиск ограничен текущим потоком.
<code>s.findbyType(type, label)</code>	Узел	Возвращает узел с указанным значением типа и/или метки. И у типа, и у имени может быть значение <code>None</code> , в этом случае используется другой параметр. Если совпадение получено для нескольких узлов, выбирается и возвращается произвольный из них. Если критериям поиска не удовлетворяет ни один узел, возвращается значение <code>None</code> .
<code>s.findDownstream(fromNodes)</code>	Набор	Проводит поиск в предоставленном списке узлов и возвращает набор узлов ниже по потоку от предоставленных узлов. Возвращаемый список включает в себя исходные предоставленные узлы.
<code>s.findUpstream(fromNodes)</code>	Набор	Проводит поиск в предоставленном списке узлов и возвращает набор узлов выше по потоку от предоставленных узлов. Возвращаемый список включает в себя исходные предоставленные узлы.

Например, если в потоке есть один узел фильтра, к которому должен обратиться сценарий, этот узел можно найти с помощью следующего кода в сценарии:

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

Другой вариант - если известен ID узла (показанный на вкладке Аннотации диалогового окна узла), можно использовать для поиска узла этот ID, например:

```
stream = modeler.script.stream()
node = stream.findById("id32FJT71G2") # ID узла фильтра
...
```

Задание свойств

У узлов, потоков, моделей и выводов есть свойства, доступные для обращения и в большинстве случаев - для задания. Обычно свойства используются для изменения поведения или условий создания объектов. Методы, доступные для обращения к свойствам объектов и для задания этих свойств, сведены в следующей таблице.

Таблица 11. Методы для доступа к свойствам объектов и для задания этих свойств

Метод	Возвращаемый тип	Описание
<code>p.getPropertyValue(propertyName)</code>	Объект	Возвращает значение именованного свойства или значение <code>None</code> , если такого свойства не существует.
<code>p.setPropertyValue(propertyName, value)</code>	Неприменимо	Задаёт значение именованного свойства.
<code>p.setPropertyValues(properties)</code>	Неприменимо	Задаёт значения именованных свойств. Каждая запись в карте свойств состоит из ключа, представляющего имя свойства, и значения, которое должно быть назначено этому свойству.
<code>p.getKeyedPropertyValue(propertyName, keyName)</code>	Объект	Возвращает значение именованного свойства и связанный ключ или значение <code>None</code> , если такого свойства или ключа не существует.
<code>p.setKeyedPropertyValue(propertyName, keyName, value)</code>	Неприменимо	Задаёт значение именованного свойства и ключ.

Например, если вы хотите задать значение узла файла переменных при запуске потока, можно использовать следующий сценарий:

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

Другой вариант - вам может потребоваться отфильтровать поле с узла Фильтр. В этом случае значение также сопровождается ключом для имени поля, например:

```
stream = modeler.script.stream()
# Найти узел Фильтр ...
node = stream.findByType("filter", None)
# ... и отфильтровать поле "Na"
node.setKeyedPropertyValue("include", "Na", False)
```

Создание узлов и изменение потоков

В некоторых ситуациях вам может потребоваться добавить новые узлы к существующим потокам. Добавление узлов к существующим потокам обычно включает в себя следующие задачи:

1. Создание узлов.
2. Соединение узлов с процессами существующего потока.

Создание узлов

Потоки предоставляют несколько способов создания узлов. Соответствующие методы сведены в следующей таблице.

Таблица 12. Методы для создания узлов

Метод	Возвращаемый тип	Описание
<code>s.create(nodeType, name)</code>	Узел	Создает узел заданного типа и добавляет его в заданный поток.
<code>s.createAt(nodeType, name, x, y)</code>	Узел	Создает узел заданного типа и добавляет его в заданный поток в заданном положении. Если $x < 0$ или $y < 0$, положение не задается.
<code>s.createModelApplier(modelOutput, name)</code>	Узел	Создает узел применения модели, полученный из предоставленного объекта вывода модели.

Например, чтобы создать в потоке новый узел типа, вы можете использовать следующий сценарий:

```
stream = modeler.script.stream()
# Создать новый узел типа
node = stream.create("type", "My Type")
```

Соединение и отсоединение узлов

Когда в потоке создается новый узел, до использования его нужно соединить с последовательностью узлов. Потоки обеспечивают несколько способов соединения и отсоединения узлов. Соответствующие методы сведены в следующей таблице.

Таблица 13. Методы для соединения и отсоединения узлов

Метод	Возвращаемый тип	Описание
<code>s.link(source, target)</code>	Неприменимо	Создает новое соединение между узлом источника и узлом назначения.
<code>s.link(source, targets)</code>	Неприменимо	Создает новые соединения между узлом источника и каждым узлом назначения в предоставленном списке.
<code>s.linkBetween(inserted, source, target)</code>	Неприменимо	Соединяет узел между двумя другими экземплярами узлов (узлами источника и назначения) и задает положение для вставки узла между ними. Сначала удаляются все непосредственные соединения между узлами источника и назначения.
<code>s.linkPath(path)</code>	Неприменимо	Создает новый путь между экземплярами узла. Первый узел соединяется со вторым, второй с третьим и так далее.

Таблица 13. Методы для соединения и отсоединения узлов (продолжение)

Метод	Возвращаемый тип	Описание
<code>s.unlink(source, target)</code>	Неприменимо	Удаляет прямые соединения между узлами источника и назначения.
<code>s.unlink(source, targets)</code>	Неприменимо	Удаляет все прямые соединения между узлом источника и каждым объектом в списке назначения.
<code>s.unlinkPath(path)</code>	Неприменимо	Удаляет все пути, существующие между экземплярами узла.
<code>s.disconnect(node)</code>	Неприменимо	Удаляет все соединения между предоставленным узлом и всеми другими узлами в заданном потоке.
<code>s.isValidLink(source, target)</code>	<i>логическое</i>	Возвращает значение True, если можно создать соединение между заданными узлами источника и назначения. При этом методе проверяется, что оба объекта принадлежат заданному потоку, что узел источника позволяет устанавливать соединение, а узел назначения - принять его, и что при создании такого соединения в потоке не возникнет заикливания.

Приведенный пример сценария выполняет пять задач:

1. Создает входной узел файла переменных и выходной узел Таблица.
2. Соединяет узлы.
3. Задает имя входного узла файла переменных.
4. Фильтрует поле "Drug" в полученном выводе.
5. Выполняет узел Таблица.

```
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)
```

Импорт, замена и удаление узлов

Кроме создания узлов и соединения с ними, часто необходима замена или удаление узлов из потока. Доступные для импорта, замены и удаления узлов методы сведены в следующей таблице.

Таблица 14. Методы для импорта, замены и удаления узлов

Метод	Возвращаемый тип	Описание
<code>s.replace(originalNode, replacementNode, discardOriginal)</code>	Неприменимо	Заменяет заданный узел в указанном потоке. Заданному потоку должны принадлежать и исходный узел, и узел для замены.

Таблица 14. Методы для импорта, замены и удаления узлов (продолжение)

Метод	Возвращаемый тип	Описание
<code>s.insert(source, nodes, newIDs)</code>	Список	Вставляет копии узлов из предоставленного списка. Предполагается, что все узлы из предоставленного списка содержатся в заданном потоке. Флаг <code>newIDs</code> указывает, должны ли генерироваться новые ID для всех узлов, или нужно копировать и использовать только существующий ID. Предполагается, что у всех узлов в потоке уникальные ID, поэтому для этого флага нужно задать значение <code>True</code> , если поток источника совпадает с заданным. Этот метод возвращает список вновь вставленных узлов, где порядок узлов не определен (то есть порядок не обязан быть тем же, что во входном списке узлов).
<code>s.delete(node)</code>	Неприменимо	Удаляет заданный узел из указанного потока. Владельцем узла должен быть заданный поток.
<code>s.deleteAll(nodes)</code>	Неприменимо	Удаляет все заданные узлы из заданного потока. Все узлы в собрании должны принадлежать к заданному потоку.
<code>s.clear()</code>	Неприменимо	Удаляет все узлы из заданного потока.

Перемещение по узлам в потоке

Общее требование - это определение узлов, расположенных дальше по потоку или предшествующих в потоке данному узлу. Поток обеспечивает несколько методов, которые можно использовать для обнаружения таких узлов. Соответствующие методы сведены в следующей таблице.

Таблица 15. Методы для обнаружения узлов выше и ниже по потоку

Метод	Возвращаемый тип	Описание
<code>s.iterator()</code>	Итератор	Возвращает итератор по объектам узла, содержащимся в заданном потоке. Если поток изменяется между вызовами функции <code>next()</code> , поведение итератора не определено.
<code>s.predecessorAt(node, index)</code>	Узел	Возвращает заданный непосредственный предшественник предоставленного узла или значение <code>None</code> , если значение индекса выходит за границы.
<code>s.predecessorCount(node)</code>	<i>int</i>	Возвращает количество непосредственных предшественников предоставленного узла.
<code>s.predecessors(node)</code>	Список	Возвращает непосредственные предшественники предоставленного узла.

Таблица 15. Методы для обнаружения узлов выше и ниже по потоку (продолжение)

Метод	Возвращаемый тип	Описание
<code>s.successorAt(node, index)</code>	Узел	Возвращает заданный непосредственный преемник предоставленного узла или значение <code>None</code> , если значение индекса выходит за границы.
<code>s.successorCount(node)</code>	<i>int</i>	Возвращает количество непосредственных преемников предоставленного узла.
<code>s.successors(node)</code>	Список	Возвращает непосредственные преемники предоставленного узла.

Получение информации об узлах

Узлы относятся к нескольким разным категориям, таким как узлы импорта и экспорта данных, узлы построения моделей и узлы других типов. Каждый узел обеспечивает несколько методов, которые можно использовать для поиска информации об этом узле.

Методы, которые можно использовать для получения ID, имени и метки узла, сведены в следующей таблице.

Таблица 16. Методы для получения ID, имени и метки узла

Метод	Возвращаемый тип	Описание
<code>n.getLabel()</code>	<i>строка</i>	Возвращает метку вывода на экран для заданного узла. Данная метка - это значение свойства <code>custom_name</code> , если это свойство - не пустая строка и свойство <code>use_custom_name</code> не задано; в противном случае метка - это значение <code>getName()</code> .
<code>n.setLabel(label)</code>	Неприменимо	Задаёт метку вывода на экран для заданного узла. Если новая метка - это не пустая строка, она назначается свойству <code>custom_name</code> , а свойству <code>use_custom_name</code> назначается значение <code>False</code> , так что у заданной метки есть преимущество использования; в противном случае свойству <code>custom_name</code> назначается пустая строка, а свойству <code>use_custom_name</code> назначается значение <code>True</code> .
<code>n.getName()</code>	<i>строка</i>	Возвращает имя заданного узла.
<code>n.getID()</code>	<i>строка</i>	Возвращает ID заданного узла. Новый ID создается при всяком создании нового узла. Этот ID остается с узлом при его сохранении как части потока, то есть при открытии потока ID узлов сохраняются. Однако если сохраненный узел вставляется в поток, он рассматривается как новый объект, и для него выделяется новый ID.

Методы, которые можно использовать для получения другой информации об узле, сведены в следующей таблице.

Таблица 17. Методы для получения информации об узле

Метод	Возвращаемый тип	Описание
<code>n.getTypeName()</code>	<i>строка</i>	Возвращает имя в сценарии для этого узла. Это то же самое имя, которое можно использовать для создания нового экземпляра этого узла.
<code>n.isInitial()</code>	<i>Логический</i>	Возвращает значение True, если это <i>исходный</i> узел, существовавший при запуске потока.
<code>n.isInline()</code>	<i>Логический</i>	Возвращает значение True, если это <i>встроенный</i> узел, появившийся в середине потока.
<code>n.isTerminal()</code>	<i>Логический</i>	Возвращает значение True, если это <i>конечный</i> узел, создаваемый при завершении потока.
<code>n.getXPosition()</code>	<i>int</i>	Возвращает смещение положения x для узла в потоке.
<code>n.getYPosition()</code>	<i>int</i>	Возвращает смещение положения y для узла в потоке.
<code>n.setXYPosition(x, y)</code>	Неприменимо	Задаёт положение узла в потоке.
<code>n.setPositionBetween(source, target)</code>	Неприменимо	Задаёт положение узла в потоке, чтобы он располагался между двумя указанными узлами.
<code>n.isCacheEnabled()</code>	<i>Логический</i>	Возвращает значение True, если включено кэширование; в противном случае возвращает False.
<code>n.setCacheEnabled(val)</code>	Неприменимо	Включает или отключает кэш для этого объекта. Если кэш заполнен и возможность кэширования отключается, происходит очистка кэша.
<code>n.isCacheFull()</code>	<i>Логический</i>	Возвращает значение True, если кэш заполнен; в противном случае возвращает False.
<code>n.flushCache()</code>	Неприменимо	Очищает кэш для этого узла. Не приводит ни к какому действию, если кэш не включен или не заполнен.

Глава 4. API сценариев

Введение в API сценариев

API сценариев предоставляет доступ к широкому диапазону функциональных возможностей SPSS Modeler. Все описанные до сих пор методы - это часть API, к ним можно неявно обратиться в сценарии без дальнейшего импорта. Однако если вы хотите сослаться на классы API, необходимо явным образом импортировать API с помощью следующего оператора:

```
import modeler.api
```

Этот оператор `import` требуется во многих примерах API сценариев.

Пример: поиск узлов с помощью пользовательского фильтра

В раздел “Поиск узлов” на стр. 27 включен пример поиска для узла в потоке с помощью имени типа узла в качестве критерия поиска. В некоторых ситуациях требуется более общий поиск, и его можно реализовать с помощью класса `NodeFilter` и метода потока `findAll()`. Поиск такого типа включает в себя следующие два шага:

1. Создание нового класса, который расширяет `NodeFilter`, реализующий пользовательскую версию метода `accept()`.
2. Вызов метода потока `findAll()` с помощью экземпляра этого нового класса. При этом возвращаются все узлы, для которых выполнен критерий, определенный в методе `accept()`.

В следующем примере показано, как искать узлы в потоке, для которых включен кэш узлов. Возвращенный список узлов можно использовать для очищения или отключения их кэша.

```
import modeler.api
```

```
class CacheFilter(modeler.api.NodeFilter):  
    """Фильтр узлов с включенным кэшированием"""  
    def accept(this, node):  
        return node.isCacheEnabled()
```

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

Метаданные: Информация о данных

Поскольку узлы соединены, образуя поток, доступна информация о столбцах или полях на каждом узле. Например, в пользовательском интерфейсе Modeler это дает возможность выбрать поля, по которым нужно выполнить сортировку или объединение. Такая информация называется моделью данных.

Кроме того, сценарии могут обращаться к модели данных путем поиска по входным или выходным полям узла. Для некоторых узлов входная и выходная модели данных одинаковы; например, узел сортировки изменяет только порядок записей, не изменяя модель данных. Некоторые узлы, такие как узел вычислений, могут добавлять новые поля. Другие, например, узел фильтра, могут переименовывать и удалять поля.

В приведенном ниже примере сценарий обращается к стандартному потоку IBM SPSS Modeler `druglearn.str` и для каждого поля строит модель, в которой одно входное поле отбрасывается. Для этого выполняются следующие действия:

1. Оценка модели данных на выходе узла типа.
2. Цикл, перебирающий все поля входной модели данных.
3. Изменение узла фильтра для каждого входного поля.
4. Изменение имени создаваемой модели.

5. Выполнение узла построения модели.

Примечание: Перед запуском сценария в потоке `druglean.str` не забудьте задать язык сценариев Python (поскольку поток был создан в прошлой версии IBM SPSS Modeler, для него задан унаследованный язык сценариев).

```
import modeler.api

stream = modeler.script.stream()
filternode = stream.findByType("filter", None)
typenode = stream.findByType("type", None)
c50node = stream.findByType("c50", None)
# Всегда используйте пользовательское имя модели
c50node.setPropertyValue("use_model_name", True)

lastRemoved = None
fields = typenode.getOutputDataModel()
for field in fields:
    # Если поле выходное, оно игнорируется
    if field.getModelingRole() == modeler.api.ModelingRole.OUT:
        continue

    # Снова активировать последнее удаленное поле
    if lastRemoved != None:
        filternode.setKeyedPropertyValue("include", lastRemoved, True)

    # Удалить поле
    lastRemoved = field.getColumnName()
    filternode.setKeyedPropertyValue("include", lastRemoved, False)

# Задать имя новой модели и выполнить построение
c50node.setPropertyValue("model_name", "Exclude " + lastRemoved)
c50node.run([])
```

Объектом `DataModel` поддерживается ряд методов для оценки информации о полях и столбцах в модели данных. Соответствующие методы сведены в следующей таблице.

Таблица 18. Методы объекта модели данных для доступа к информации о полях или столбцах

Метод	Возвращаемый тип	Описание
<code>d.getColumnCount()</code>	<i>int</i>	Возвращает количество столбцов в модели данных.
<code>d.columnIterator()</code>	Итератор	Возвращает итератор, который возвращает каждый столбец в "естественном" порядке вставки. Итератор возвращает экземпляры столбца.
<code>d.nameIterator()</code>	Итератор	Возвращает итератор, который возвращает имя каждого столбца в "естественном" порядке вставки.
<code>d.contains(name)</code>	<i>Логический</i>	Возвращает значение <code>True</code> , если столбец с указанным именем существует в этой модели данных, и значение <code>False</code> - в противном случае.
<code>d.getColumn(name)</code>	Столбец	Возвращает столбец с указанным именем.
<code>d.getColumnGroup(name)</code>	<code>ColumnGroup</code>	Возвращает именованную группу столбцов или <code>None</code> , если такая группа столбцов не существует.
<code>d.getColumnGroupCount()</code>	<i>int</i>	Возвращает количество групп столбцов в этой модели данных.

Таблица 18. Методы объекта модели данных для доступа к информации о полях или столбцах (продолжение)

Метод	Возвращаемый тип	Описание
d.columnGroupIterator()	Итератор	Возвращает итератор, который возвращает каждый столбец группы по очереди.
d.toArray()	Column[]	Возвращает модель данных в виде массива столбцов. Столбцы упорядочены в "естественном" порядке вставки.

Каждый объект поля (столбца) содержит ряд методов для доступа к информации об этом столбце. Некоторые из них представлены в следующей таблице.

Таблица 19. Методы объекта столбца для доступа к информации о столбце

Метод	Возвращаемый тип	Описание
c.getColumnName()	строка	Возвращает имя столбца.
c.getColumnLabel()	строка	Возвращает метку столбца или пустую строку, если со столбцом не связана никакая метка.
c.getMeasureType()	MeasureType	Возвращает тип показателя для столбца.
c.getStorageType()	StorageType	Возвращает тип хранения для столбца.
c.isMeasureDiscrete()	Логический	Возвращает значение True, если столбец - дискретный. Дискретными считаются столбцы типа набора или флага.
c.isModelOutputColumn()	Логический	Возвращает значение True, если столбец - это выходной столбец модели.
c.isStorageDatetime()	Логический	Возвращает значение True, если тип хранения столбца - время, дата или отметка времени.
c.isStorageNumeric()	Логический	Возвращает значение True, если тип хранения столбца - целое или действительное число.
c.isValidValue(value)	Логический	Возвращает значение True, если заданное значение допустимо для этого типа хранения, и valid, если для столбца известны допустимые значения.
c.getModelingRole()	ModelingRole	Возвращает роль моделирования для столбца.
c.getSetValues()	Object[]	Возвращает массив допустимых значений для столбца или None, если нет известных значений или столбец не задан.
c.getValueLabel(value)	строка	Возвращает метку для значения в столбце или пустую строку, если с этим значением не связана никакая метка.

Таблица 19. Методы объекта столбца для доступа к информации о столбце (продолжение)

Метод	Возвращаемый тип	Описание
<code>c.getFalseFlag()</code>	Объект	Возвращает индикаторное значение "false" для столбца или None, если значение неизвестно или столбец - не флаг.
<code>c.getTrueFlag()</code>	Объект	Возвращает индикаторное значение "true" для столбца или None, если значение неизвестно или столбец - не флаг.
<code>c.getLowerBound()</code>	Объект	Возвращает значение нижней границы для значений в столбце или None, если значение неизвестно или столбец - не непрерывный.
<code>c.getUpperBound()</code>	Объект	Возвращает значение верхней границы для значений в столбце или None, если значение неизвестно или столбец - не непрерывный.

Имейте в виду, что для большинства методов доступа к информации о столбце есть эквивалентные методы, определенные в самом объекте модели данных. Например, следующие два оператора эквивалентны:

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

Доступ к сгенерированным объектам

При выполнении потока обычно создаются дополнительные объекты вывода. Этими дополнительными объектами могут быть новая модель или часть вывода, предоставляющая информацию для последующих выполнений.

В приведенном ниже примере поток `druglearn.str` снова используется как стартовая точка для потока. В этом примере выполняются все узлы в потоке и результаты сохраняются в списке. Затем сценарий перебирает результаты в цикле, а все выводы модели, получаемые при выполнении, сохраняются как файл модели IBM SPSS Modeler (.gm), и модель экспортируется в формате PMML.

```
import modeler.api

stream = modeler.script.stream()

# Задайте здесь существующую папку в вашей системе.
# Включите в имя завершающий разделитель каталогов
modelFolder = "C:/temp/models/"

# Выполнить поток
models = []
stream.runAll(models)

# Сохранить все созданные модели
taskrunner = modeler.script.session().getTaskRunner()
for model in models:
    # Если при выполнении потока создаются другие объекты вывода, игнорировать их
    if not(isinstance(model, modeler.api.ModelOutput)):
        continue

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

    # сохранить каждую модель...
    modelFile = modelFolder + label + algorithm + ".gm"
```

```

taskrunner.saveModelToFile(model, modelFile)

# ...и экспортировать PMML каждой модели...
modelFile = modelFolder + label + algorithm + ".xml"
taskrunner.exportModelToFile(model, modelFile, modeler.api.FileFormat.XML)

```

Класс запуска задач предоставляет удобный способ запуска различных общих задач. Доступные в этом классе методы сведены в следующей таблице.

Таблица 20. Методы класса запуска задач для выполнения общих задач

Метод	Возвращаемый тип	Описание
<code>t.createStream(name, autoConnect, autoManage)</code>	Stream	Создает и возвращает новый поток. Обратите внимание на то, что в коде, который должен создавать скрытые потоки, невидимые для пользователя, для флага <code>autoManage</code> должно быть задано значение <code>False</code> .
<code>t.exportDocumentToFile(documentOutput, filename, fileFormat)</code>	Неприменимо	Экспортирует описание потока в файл с использованием заданного формата файла.
<code>t.exportModelToFile(modelOutput, filename, fileFormat)</code>	Неприменимо	Экспортирует модель в файл с использованием заданного формата файла.
<code>t.exportStreamToFile(stream, filename, fileFormat)</code>	Неприменимо	Экспортирует поток в файл с использованием заданного формата файла.
<code>t.insertNodeFromFile(filename, diagram)</code>	Node	Читает и возвращает узел из заданного файла, вставляя его в предоставленную диаграмму. Обратите внимание на то, что это может использоваться для чтения и объектов узлов, и объектов надузлов.
<code>t.openDocumentFromFile(filename, autoManage)</code>	DocumentOutput	Читает и возвращает документ из заданного файла.
<code>t.openModelFromFile(filename, autoManage)</code>	ModelOutput	Читает и возвращает модель из заданного файла.
<code>t.openStreamFromFile(filename, autoManage)</code>	Stream	Читает и возвращает поток из заданного файла.
<code>t.saveDocumentToFile(documentOutput, filename)</code>	Неприменимо	Сохраняет документ в заданное положение файла.
<code>t.saveModelToFile(modelOutput, filename)</code>	Неприменимо	Сохраняет модель в заданное положение файла.
<code>t.saveStreamToFile(stream, filename)</code>	Неприменимо	Сохраняет поток в заданное положение файла.

Обработка ошибок

Язык программирования Python предоставляет средства обработки ошибок через блок кода `try...except`. Его можно использовать в сценариях для локализации исключительных ситуаций и исправления ошибок, которые в противном случае привели бы к прерыванию сценария.

В приведенном ниже примере сценария сделана попытка получить модель из IBM SPSS Collaboration and Deployment Services Repository. Эта операция может привести к возникновению исключительной ситуации, например, могут быть неправильно заданы регистрационные данные при входе в репозиторий, или

указанный путь к репозиторию может быть неправильным. В данном сценарии из-за этого может возникнуть исключительная ситуация `ModelerException` (все исключительные ситуации, генерируемые IBM SPSS Modeler, являются производными от `modeler.api.ModelerException`).

```
import modeler.api

session = modeler.script.session()
try:
    repo = session.getRepository()
    m = repo.retrieveModel("/some-non-existent-path", None, None, True)
    # печать направляется на вкладку Отладка панели сценариев пользовательского интерфейса моделирования
    print "Все в порядке"
except modeler.api.ModelerException, e:
    print "Произошла ошибка:", e.getMessage()
```

Примечание: Некоторые операции сценариев могут вызвать появление стандартных исключительных ситуаций Java; они не являются производными от `ModelerException`. Чтобы учесть эти исключительные ситуации, можно использовать дополнительный блок отслеживания всех исключительных ситуаций Java, например:

```
import modeler.api

session = modeler.script.session()
try:
    repo = session.getRepository()
    m = repo.retrieveModel("/some-non-existent-path", None, None, True)
    # печать направляется на вкладку Отладка панели сценариев пользовательского интерфейса моделирования
    print "Все в порядке"
except modeler.api.ModelerException, e:
    print "Произошла ошибка:", e.getMessage()
except java.lang.Exception, e:
    print "Произошла исключительная ситуация Java:", e.getMessage()
```

Параметры потока, сеанса и надузла

Параметры предоставляют удобный способ передачи значений во время выполнения, чтобы не кодировать их непосредственно в сценарии. Параметры и их значения определяются так же, как для потоков, то есть как записи в таблице потока или надузла или параметрами в командной строке. Классы потока и надузла реализуют набор функций, определенный объектом `ParameterProvider`, как показано в следующей таблице. Сеанс обеспечивает вызов `getParameters()`, в результате чего возвращается объект, определяющий эти функции.

Таблица 21. Функции, определенные объектом `ParameterProvider`

Метод	Возвращаемый тип	Описание
<code>p.parameterIterator()</code>	Итератор	Возвращает итератор имен параметров для этого объекта.
<code>p.getParameterDefinition(parameterName)</code>	<code>ParameterDefinition</code>	Возвращает определение для параметра с заданным именем или значение <code>None</code> , если у провайдера такого параметра нет. Результат может быть снимком определения на момент вызова метода и не должен отображать последовательные изменения, произведенные данным провайдером для параметра.
<code>p.getParameterLabel(parameterName)</code>	<i>строка</i>	Возвращает метку именованного параметра или значение <code>None</code> , если такого параметра не существует.
<code>p.setParameterLabel(parameterName, label)</code>	Неприменимо	Задает метку именованного параметра.

Таблица 21. Функции, определенные объектом *ParameterProvider* (продолжение)

Метод	Возвращаемый тип	Описание
<code>p.getParameterStorage(parameterName)</code>	<code>ParameterStorage</code>	Возвращает систему хранения именованного параметра или значение <code>None</code> , если такого параметра не существует.
<code>p.setParameterStorage(parameterName, storage)</code>	Неприменимо	Задаёт систему хранения именованного параметра.
<code>p.getParameterType(parameterName)</code>	<code>ParameterType</code>	Возвращает тип именованного параметра или значение <code>None</code> , если такого параметра не существует.
<code>p.setParameterType(parameterName, type)</code>	Неприменимо	Задаёт тип именованного параметра.
<code>p.getParameterValue(parameterName)</code>	Объект	Возвращает значение именованного параметра или значение <code>None</code> , если такого параметра не существует.
<code>p.setParameterValue(parameterName, value)</code>	Неприменимо	Задаёт значение именованного параметра.

В следующем примере сценарий агрегирует некоторые данные Telco для определения, в каком регионе самые низкие значения среднего дохода. Затем для этого региона задается параметр потока. После этого данный параметр потока используется на узле `Выбор` для исключения данных этого региона до построения модели перехода в конкурирующую компанию на основании оставшихся данных.

Это искусственный пример, так как сценарий сам генерирует узел `Выбор`, то есть может использовать сгенерированные точные значения непосредственно в выражении узла `Выбор`. Однако потоки обычно строятся заранее, поэтому задание параметров таким образом предоставляет полезный пример.

В первой части сценария для этого примера создается параметр потока, который будет определять регион с наименьшим средним доходом. Сценарий создает также узлы на ветви агрегации и на ветви построения модели и соединяет их.

```
import modeler.api

stream = modeler.script.stream()

# Инициализировать параметр потока
stream.setParameterStorage("LowestRegion", modeler.api.ParameterStorage.INTEGER)

# Сначала создаем ветвь агрегации для вычисления среднего дохода по регионам
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")
# Указываем параметр потока при выборе
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)

```

В этом примере сценарий создает следующий поток.

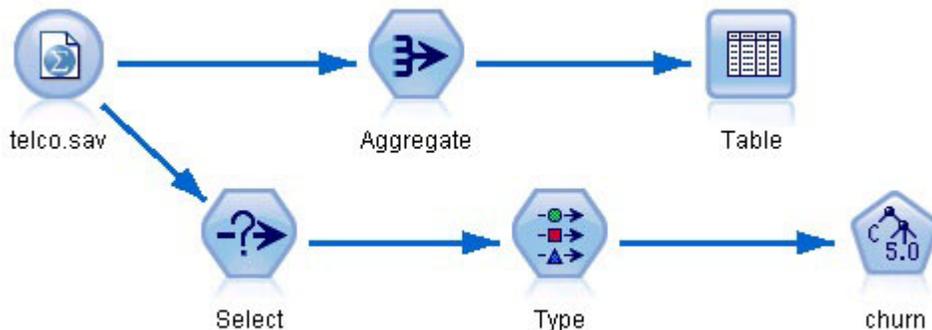


Рисунок 5. Поток, возникающий при выполнении сценария в примере

В следующей части сценария примера выполняется узел Таблица в конце ветви агрегации.

```

# Сначала выполним узел Таблица
results = []
tablenode.run(results)

```

В следующей части примера сценарий обращается к табличному выводу, сгенерированному при выполнении узла Таблица. Затем сценарий проводит итерации по строкам таблицы, находя регион с минимальным средним доходом.

```

# При выполнении узла Таблица в качестве вывода должна быть создана одна таблица
table = results[0]

# Табличный вывод содержит RowSet, поэтому к значениям можно обращаться, как к строкам и столбцам
rowset = table.getRowSet()
min_income = 1000000.0
min_region = None

# По тому, как определен узел агрегации, первый столбец
# содержит регион, а второй - средний доход
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

```

В следующей части сценария примера используется регион с наименьшим средним доходом, чтобы задать параметр потока "LowestRegion", созданный ранее. Затем сценарий запускает построитель моделей, в котором заданный регион исключен из данных обучения.

```

# Проверяем, что значение было назначено
if min_region != None:
    stream.setParameterValue("LowestRegion", min_region)
else:

```

```

    stream.setParameterValue("LowestRegion", -1)

# В завершение запускаем построитель модели с критерием выбора
c50node.run([])

Полный пример сценария показан ниже.
import modeler.api

stream = modeler.script.stream()

# Создаем параметр потока
stream.setParameterStorage("LowestRegion", modeler.api.ParameterStorage.INTEGER)

# Сначала создаем ветвь агрегации для вычисления среднего дохода по регионам
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")
# Указываем параметр потока при выборе
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)

# Сначала выполним узел Таблица
results = []
tablenode.run(results)

# При выполнении узла Таблица в качестве вывода должна быть создана одна таблица
table = results[0]

# Табличный вывод содержит RowSet, поэтому к значениям можно обращаться, как к строкам и столбцам
rowset = table.getRowSet()
min_income = 1000000.0
min_region = None

# По тому, как определен узел агрегации, первый столбец
# содержит регион, а второй - средний доход
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

# Проверяем, что значение было назначено
if min_region != None:

```

```

    stream.setParameterValue("LowestRegion", min_region)
else:
    stream.setParameterValue("LowestRegion", -1)

# В завершение запускаем построитель модели с критерием выбора
c50node.run([])

```

Глобальные значения

Глобальные значения используются при вычислении разнообразных сводных статистических показателей для заданных полей. К этим сводным значениям можно получить доступ отовсюду в потоке. Глобальные значения похожи на параметры потока тем, что к ним можно обратиться отовсюду в потоке по имени. Они отличаются от параметров потока тем, что связанные значения изменяются автоматически при запуске узла. Задать глобальные значения, а не назначаются сценарием или из командной строки. К глобальным значениям для потока можно обратиться, вызвав метод потока `getGlobalValues()`.

Объект `GlobalValues` определяет функции, показанные в следующей таблице.

Таблица 22. Функции, определенные объектом `GlobalValues`

Метод	Возвращаемый тип	Описание
<code>g.fieldNameIterator()</code>	Итератор	Возвращает итератор для каждого имени поля по крайней мере с одним глобальным значением.
<code>g.getValue(type, fieldName)</code>	Объект	Возвращает глобальное значение для заданного типа и имени поля или <code>None</code> , если значение не удастся обнаружить. В общем случае предполагается, что возвращаемое значение - это число, хотя будущие функциональные возможности позволят возвращать различные типы данных.
<code>g.getValues(fieldName)</code>	Отобразить	Возвращает карту, содержащую известные записи для заданного имени поля, или значение <code>None</code> , если для этого поля не существует записей.

`GlobalValues.Type` определяет тип доступных сводных статистических показателей. Доступны следующие сводные статистики:

- `MAX`: максимальное значение в поле.
- `MEAN`: среднее значение в поле.
- `MIN`: минимальное значение в поле.
- `STDDEV`: среднеквадратичное отклонение значений в поле.
- `SUM`: сумма значений в поле.

Например, следующий сценарий обращается к среднему значению поля "income", вычисляемому узлом. Задать глобальные значения:

```

import modeler.api

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

```

Работа с несколькими потоками: автономные сценарии

Для работы с несколькими потоками нужно использовать автономный сценарий. Автономный сценарий можно изменить и запустить в пользовательском интерфейсе IBM SPSS Modeler или передать как параметр командной строки в пакетном режиме.

Следующий автономный сценарий открывает два потока. Один из этих потоков строит модель, а второй создает графики распределения предсказанных значений.

```
# Перейти в нужный каталог вашей системы
demosDir = "C:/Program Files/IBM/SPSS/Modeler/16/DEMOS/streams/"

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

# Открыть поток построения модели, найти узел C5.0 и запустить его
buildstream = tasks.openStreamFromFile(demosDir + "druglearn.str", True)
c50node = buildstream.findByType("c50", None)
results = []
c50node.run(results)

# Теперь откроем поток построения графиков, найдем извлеченные значения Na_to_K и гистограмму
plotstream = tasks.openStreamFromFile(demosDir + "drugplot.str", True)
derivenode = plotstream.findByType("derive", None)
histogramnode = plotstream.findByType("histogram", None)

# Создаем узел применения модели, вставляем его между узлами извлечения данных и гистограммы
# и затем запускаем гистограмму
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([])

# Наконец, завершаем потоки
buildstream.close()
plotstream.close()
```

Глава 5. Подсказки для сценариев

В этом разделе представлен обзор подсказок и способов использования сценариев, в том числе изменение выполнения потоков, и использование в сценариях закодированных паролей.

Изменение выполнения потока

Когда поток запущен, его конечные узлы выполняются в порядке, который оптимизирован для ситуации по умолчанию. В некоторых случаях вы можете предпочесть другой порядок выполнения. Чтобы изменить порядок выполнения потока, выполните следующие шаги на вкладке Выполнение диалогового окна свойств потока:

1. Начните с пустого сценария.
2. Нажмите кнопку **Присоединить сценарий по умолчанию** на панели инструментов, чтобы добавить сценарий потока по умолчанию.
3. Измените порядок операторов в сценарии потока по умолчанию на такой, в котором должны выполняться операторы.

Работа с моделями

Если в IBM SPSS Modeler разрешена автоматическая замена модели и узел построителя модели выполняется через пользовательский интерфейс IBM SPSS Modeler, слепок существующей модели, связанный с узлом построителя модели, заменяется на слепок новой модели. Если узел построителя модели выполняется при помощи сценария, слепок существующей модели не заменяется. Чтобы заменить слепок существующей модели, надо явно указать замену слепка в вашем сценарии.

Генерирование закодированного пароля

В определенных случаях вам может потребоваться включить пароль в сценарий; например, если нужен доступ к защищенному паролем источнику данных. Закодированные пароли можно использовать в следующих ситуациях:

- Свойства узлов Источник базы данных и Выходные данные
- Аргументы командной строки для регистрации на сервере
- Свойства соединения с базой данных, хранящиеся в файле *.par* (файл параметров, сгенерированный на вкладке Опубликовать узла экспорта)

Через пользовательский интерфейс доступен инструмент генерирования закодированных паролей на основе алгоритма Blowfish (дополнительную информацию смотрите на сайте <http://www.schneier.com/blowfish.html>). После кодирования пароль можно копировать и сохранять в файлы сценария и в аргументы командной строки. Закодированный пароль хранит свойство узла `epassword`, используемое для `database` и `databaseexport`.

1. Чтобы сгенерировать закодированный пароль, в меню Инструменты выберите:
Закодировать пароль...
2. Задайте пароль в текстовом поле Пароль.
3. Нажмите кнопку **Кодировать**, чтобы сгенерировать случайное кодирование вашего пароля.
4. Нажмите кнопку Копировать, чтобы скопировать закодированный пароль в буфер обмена.
5. Вставьте пароль в нужный сценарий или параметр.

Проверка сценария

Синтаксис всех типов сценариев можно быстро проверить, нажав красную кнопку проверки на панели инструментов диалогового окна Автономный сценарий.



Рисунок 6. Значки панели инструментов потокового сценария

Проверка сценариев предупреждает вас об ошибках в коде и предлагает рекомендации по улучшению. Для просмотра строки с ошибками щелкните по ссылке обратной связи внизу диалогового окна. При этом ошибка будет выделена красным цветом.

Работа со сценариями из командной строки

Сценарии позволяют запускать операции, обычно выполняемые в пользовательском интерфейсе. Просто задайте и запустите автономный сценарий в командной строке при запуске IBM SPSS Modeler.

Например:

```
client -script scores.py -execute
```

При наличии флага `-script` загружается заданный сценарий, а при флаге `-execute` выполняются все команды в файле сценария.

Задание путей файлов

При задании путей для каталогов и файлов можно использовать в качестве разделителей каталогов либо прямую дробную черту (/), либо двойную обратную дробную черту (\), например,

```
c:/demos/druglearn.str
```

или

```
c:\\demos\\druglearn.str
```

Совместимость с предыдущими выпусками

Обычно унаследованные сценарии, созданные в предыдущих выпусках IBM SPSS Modeler, должны без изменений работать в текущем выпуске. Чтобы эти сценарии работали, включите переключатель **Унаследованные** на вкладке сценариев потока в диалоговом окне Свойства потока или в диалоговом окне Автономный сценарий. Сейчас слепки моделей можно автоматически вставлять в поток (это параметр по умолчанию), чтобы или заменить существующий слепок такого типа в потоке, или присоединить его. Произойдет ли это на самом деле, зависит от заданных опций **Добавить в модель в поток** и **Заменить предыдущую модель (Инструменты > Опции > Пользовательские опции > Уведомления)**. Например, вам может потребоваться изменить сценарий предыдущего выпуска, в котором замена производится посредством удаления существующего слепка и вставки нового.

Созданные в текущем выпуске сценарии могут не работать в более старых выпусках.

Созданные в текущем выпуске сценарии Python не будут работать в более старых выпусках.

Если созданный в более старом выпуске сценарий использует команду, которая была заменена (или объявлена устаревшей), старая форма будет по-прежнему поддерживаться, но появится предупреждение. Например, старое ключевое слово `generated` было заменено на `model`, а `clear generated` - на `clear generated palette`. Использующие старые формы сценарии будут по-прежнему работать, но появится предупреждение.

Глава 6. Аргументы командной строки

Вызов программного обеспечения

Для запуска IBM SPSS Modeler можно использовать командную строку операционной системы, как описано ниже.

1. Откройте окно DOS (окно командной строки) на компьютере с IBM SPSS Modeler.
2. Чтобы запустить интерфейс IBM SPSS Modeler в интерактивном режиме, введите команду `modelerclient` с необходимыми аргументами, например:

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

Доступные аргументы (флаги) позволяют подключаться к серверу, загружать потоки, выполнять сценарии и указывать при необходимости прочие параметры выполнения.

Использование аргументов командной строки

Аргументы командной строки (также известные как **флаги**) можно присоединить к начальной команде `modelerclient` для изменения вызова IBM SPSS Modeler.

Доступно несколько типов аргументов командной строки; они описаны ниже в этом разделе.

Таблица 23. Типы аргументов командной строки.

Тип аргумента	Где описано
Системные аргументы	Дополнительную информацию смотрите в разделе “Системные аргументы” на стр. 50.
Аргументы параметров	Дополнительную информацию смотрите в разделе “Аргументы параметров” на стр. 51.
Аргументы соединений с сервером	Дополнительную информацию смотрите в разделе “Аргументы соединения с сервером” на стр. 51.
Аргументы соединения с IBM SPSS Collaboration and Deployment Services Repository	Дополнительную информацию смотрите в разделе “Аргументы соединения с IBM SPSS Collaboration and Deployment Services Repository” на стр. 52.

Например, можно использовать флаги `-server`, `-stream` и `-execute` для соединения с сервером, а затем загрузить и запустить поток, как в следующем примере:

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

Обратите внимание на то, что при запуске для локальной установки клиента аргументы соединения с сервером не требуются.

Значения параметров, содержащие пробелы, могут быть заключены в двойные кавычки, например:

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

Таким же образом можно запустить и сценарии IBM SPSS Modeler, указав флаг `-script`.

Аргументы отладки командной строки

Для отладки командной строки используйте команду `modelerclient`, чтобы запустить IBM SPSS Modeler с нужными аргументами. Это позволяет вам проверить, что команды будут выполняться, как предполагается.

Вы можете подтвердить также значения любых параметров, переданных из командной строки в диалоговом окне Параметры сеанса (меню Инструменты, параметры Задать сеанс).

Системные аргументы

В следующей таблице описаны системные параметры, доступные для вызова из командной строки пользовательского интерфейса.

Таблица 24. Системные аргументы

Аргумент	Поведение/описание
@ <commandFile>	Символ @ с последующим именем файла определяет список команд. Когда modelerclient встречает аргумент, начинающийся с @, он работает с командами из этого файла, как будто они используются в командной строке. Дополнительную информацию смотрите в разделе “Объединение нескольких аргументов” на стр. 53.
-directory <dir>	Задает рабочий каталог по умолчанию. В локальном режиме этот каталог используется и для данных, и для вывода. Пример: -directory c:/ или -directory c:\\
-server_directory <dir>	Задает каталог сервера для данных по умолчанию. Рабочий каталог, обозначенный флагом -directory, используется для выходных данных.
-execute	После запуска выполняет любой поток, состояние или сценарий, загруженные при запуске. Если сценарий загружается в дополнение к потоку или состоянию, будет выполняться один этот сценарий.
-stream <stream>	При запуске загрузить указанный поток. Можно задать несколько потоков, но в качестве текущего потока будет использоваться последний из них.
-script <script>	При запуске загрузить указанный автономный сценарий. Он может быть задан в дополнение к потоку или состоянию, как описано ниже, но при запуске можно загрузить только один сценарий. Если суффикс файла сценария - .py, файл считается сценарием Python, если нет - унаследованным сценарием.
-model <model>	При запуске загрузить заданную сгенерированную модель (файл формата .gm).
-state <state>	При запуске загрузить указанное сохраненное состояние.
-project <project>	Загрузить заданный проект. При запуске можно загрузить только один проект.
-output <output>	При запуске загрузить сохраненный объект вывода (файл формата .cou).
-help	Выводит список аргументов командной строки. При указании этой опции все другие аргументы игнорируются и выводится окно Справка.
-P <имя>=<значение>	Используется для задания параметра запуска. Может использоваться также для задания свойств узла (параметров слота).
-scriptlang <python legacy>	Так можно задать язык сценария, связанный с опцией -script, независимо от суффикса файла сценария. Пример client -scriptlang python -script scores.txt -execute Указанный файл сценария будет запущен при помощи Python, даже если суффикс файла - не .py.

Примечание: Каталоги по умолчанию можно задать также в пользовательском интерфейсе. Для доступа к этим опциям в меню Файл выберите **Задать рабочий каталог** или **Задать каталог сервера**.

Загрузка нескольких файлов

Из командной строки можно загрузить несколько потоков, состояний и файлов выходных данных при запуске, повторив соответствующий аргумент для каждого загружаемого объекта. Например, чтобы загрузить и запустить два потока с названиями *report.str* и *train.str*, можно использовать следующую команду:

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

Загрузка объектов из IBM SPSS Collaboration and Deployment Services Repository

Так как определенные объекты можно загрузить из файла или из IBM SPSS Collaboration and Deployment Services Repository (если есть лицензия), префикс имени файла `spsscr:` и, дополнительно, `file:` (для объектов на диске) указывает IBM SPSS Modeler, где искать данный объект. Префикс работает со следующими флагами:

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

Этот префикс используется для создания URI, указывающего положение объекта, например, `-stream "spsscr:///folder_1/scoring_stream.str"`. Наличие префикса `spsscr:` требует, чтобы в той же команде было задано допустимое соединение с IBM SPSS Collaboration and Deployment Services Repository. Поэтому полная команда может выглядеть, как в следующем примере:

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

Обратите внимание на то, что в командной строке вы *должны* использовать URI. Более простая форма `REPOSITORY_PATH` не поддерживается. (Это работает только в сценариях).

Аргументы параметров

Во время выполнения командной строки IBM SPSS Modeler параметры можно использовать как флаги. В аргументах командной строки флаг `-P` используется для обозначения параметра в форме `-P <имя>=<значение>`.

Параметрами могут быть любые из следующих:

- **Простые параметры**
- **Параметры слота**, также называемые **свойства узла**. Эти параметры используются для изменения параметров узлов в потоке. Дополнительную информацию смотрите в разделе “Обзор свойств узлов” на стр. 56.
- **Параметры командной строки**, используемые для изменения вызова IBM SPSS Modeler.

Например, вы можете предоставить имена пользователей источников данных и пароли в виде флага командной строки следующим образом:

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

Формат тот же, что и у параметра `datasource` свойства узла `database`. Дополнительную информацию смотрите в разделе “Свойства узла `database`” на стр. 64.

Аргументы соединения с сервером

Флаг `-server` сообщает IBM SPSS Modeler, что нужно соединиться с общедоступным сервером, а флаги `-hostname`, `-use_ssl`, `-port`, `-username`, `-password` и `-domain` используются, чтобы указать IBM SPSS Modeler, как соединиться с этим общедоступным сервером. Если аргумент `-server` не задан, используется сервер по умолчанию или локальный сервер.

Примеры

Чтобы соединиться с общедоступным сервером:

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

Чтобы соединиться с кластером серверов:

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

Обратите внимание на то, что для соединения с кластером серверов требуется координатор процессов через IBM SPSS Collaboration and Deployment Services, поэтому аргумент `-cluster` нужно использовать в комбинации с опциями соединения с репозиторием (`spsscr_*`). Дополнительную информацию смотрите в разделе “Аргументы соединения с IBM SPSS Collaboration and Deployment Services Repository”.

Таблица 25. Аргументы соединения с сервером.

Аргумент	Поведение/описание
<code>-server</code>	Запускает IBM SPSS Modeler в режиме сервера, соединяясь с публичным сервером при помощи флагов <code>-hostname</code> , <code>-port</code> , <code>-username</code> , <code>-password</code> и <code>-domain</code> .
<code>-hostname <имя></code>	Имя хоста компьютера сервера. Доступно только в режиме сервера.
<code>-use_ssl</code>	Задаёт, что соединение должно использовать SSL. Этот флаг не обязательный; по умолчанию SSL <i>не</i> используется.
<code>-port <номер></code>	Номер порта заданного сервера. Доступно только в режиме сервера.
<code>-cluster <имя></code>	Задаёт соединение с кластером серверов, а не с указанным по имени сервером; этот аргумент используется как альтернатива для аргументов <code>hostname</code> , <code>port</code> и <code>use_ssl</code> . Имя - это имя кластера или уникальный URI, определяющий кластер в IBM SPSS Collaboration and Deployment Services Repository. Кластер серверов управляется координатором процессов через IBM SPSS Collaboration and Deployment Services. Дополнительную информацию смотрите в разделе “Аргументы соединения с IBM SPSS Collaboration and Deployment Services Repository”.
<code>-username <имя></code>	Имя пользователя для регистрации на сервере. Доступно только в режиме сервера.
<code>-password <пароль></code>	Пароль для регистрации на сервере. Доступно только в режиме сервера. <i>Примечание:</i> Если аргумент <code>-password</code> не используется, последует предложение ввести пароль.
<code>-epassword <строка_закодир_пароля></code>	Закодированный пароль для регистрации на сервере. Доступно только в режиме сервера. <i>Примечание:</i> Закодированный пароль можно сгенерировать в меню Инструменты прикладной программы IBM SPSS Modeler.
<code>-domain <имя></code>	Домен, используемый для регистрации на сервере. Доступно только в режиме сервера.
<code>-P <имя>=<значение></code>	Используется для задания параметра запуска. Может использоваться также для задания свойств узла (параметров слота).

Аргументы соединения с IBM SPSS Collaboration and Deployment Services Repository

Примечание: Для получения доступа к репозиторию IBM SPSS Collaboration and Deployment Services требуется отдельная лицензия. Дополнительную информацию смотрите в разделе <http://www.ibm.com/software/analytics/spss/products/deployment/cds/>

Если вы хотите сохранять данные в IBM SPSS Collaboration and Deployment Services или извлекать их оттуда с помощью командной строки, необходимо задать допустимое соединение с IBM SPSS Collaboration and Deployment Services Repository. Например:

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

В следующей таблице перечислены аргументы, которые можно использовать для конфигурирования соединения.

Таблица 26. Аргументы соединения с IBM SPSS Collaboration and Deployment Services Repository

Аргумент	Поведение/описание
-spsscr_hostname <имя хоста или IP-адрес>	Имя хоста или IP-адрес сервера, на котором установлен IBM SPSS Collaboration and Deployment Services Repository.
-spsscr_port <номер>	Номер порта, через который IBM SPSS Collaboration and Deployment Services Repository получает доступ к соединению (обычно по умолчанию это 8080).
-spsscr_use_ssl	Задаёт, что соединение должно использовать SSL. Этот флаг не обязательный; по умолчанию SSL <i>не</i> используется.
-spsscr_username <имя>	Имя пользователя для регистрации в IBM SPSS Collaboration and Deployment Services Repository.
-spsscr_password <пароль>	Пароль для регистрации в IBM SPSS Collaboration and Deployment Services Repository.
-spsscr_epassword <закодированный пароль>	Закодированный пароль для регистрации в IBM SPSS Collaboration and Deployment Services Repository.
-spsscr_domain <имя>	Домен, используемый для регистрации в IBM SPSS Collaboration and Deployment Services Repository. Этот флаг необязательный, не используйте его, если вы не зарегистрировались при помощи LDAP или Active Directory.

Объединение нескольких аргументов

Несколько аргументов можно объединить в один командный файл, задаваемый при вызове, используя символ @ перед именем файла. Это позволяет укоротить вызов командной строки и преодолеть все ограничения операционных систем на длину команды. Например, следующая команда запуска использует аргументы, заданные в файле, указанном <commandFileName>.

```
modelerclient @<commandFileName>
```

Если требуются пробелы, заключите имя файла и путь к командному файлу в кавычки следующим образом:

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

При запуске командный файл может содержать все ранее заданные аргументы индивидуально, по одному аргументу на строку. Например:

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

При записи в командные файлы и ссылках на них убедитесь, что выполнены следующие ограничения:

- Использовать только одну команду на строку.
- Не включайте аргумент @CommandFile в командный файл.

Глава 7. Справочник по свойствам

Справочный обзор свойств

Для узлов, потоков, надузлов и проектов можно задать много различных свойств. Некоторые свойства общие для всех узлов, например, имя, аннотация и подсказки, а другие свойства специфичны для определенных типов узлов. Есть и свойства, относящиеся к операциям потоков высокого уровня, таким как кэширование или поведение надузлов. К свойствам можно обратиться через стандартный пользовательский интерфейс (например, когда вы открываете диалоговое окно для изменения опций узла), а также использовать их в самых разных ситуациях.

- Свойства можно изменять через сценарии, как описано в этом разделе. Дополнительную информацию смотрите в разделе “Задание свойств” на стр. 28.
- Свойства узлов можно использовать в параметрах надузла.
- Свойства узлов можно использовать также как часть опций командной строки (используя флаг -P) при запуске IBM SPSS Modeler.

В контексту сценариев в IBM SPSS Modeler свойства узлов и потоков часто называются **параметрами слота**. В этом руководстве на них указывается как на свойства узлов и потоков.

Дополнительную информацию о языке сценариев смотрите в разделе Глава 2, “Язык сценариев”, на стр. 13.

Сокращения

Во всех синтаксических формах для свойств узлов используются стандартные сокращения. Изучение сокращений полезно для конструирования сценариев.

Таблица 27. Стандартные сокращения, используемые во всех синтаксических формах.

Сокращение	Значение
abs	Абсолютное значение
len	Длина
мин	Минимум
макс	Максимум
correl	Корреляция
covar	Ковариация
num	Число или числовой
pct	Процент или процентная доля
transp	Прозрачность
xval	Перекрестная проверка
пер	Дисперсия или переменная (в узлах источника)

Примера свойств узла и потока

Свойства узлов и потоков можно использовать с IBM SPSS Modeler разными способами. Чаще всего они применяются как часть сценария, **автономного**, используемого для автоматизации нескольких потоков или операций, или **потокового**, используемого для автоматизации процессов в одном потоке. Можно задать также параметры узла, используя свойства узла в надузле. На самом базовом уровне свойства можно использовать также как опцию в командной строке для запуска IBM SPSS Modeler. Применив аргумент -p как часть вызова из командной строки, можно использовать свойство потока для изменения настроек в потоке.

Дальнейшие примеры сценариев смотрите в темах “Параметры потока, сеанса и надузла” на стр. 40 и “Системные аргументы” на стр. 50.

Обзор свойств узлов

У узла каждого типа есть свой собственный набор разрешенных свойств, и у каждого свойства есть тип. Это может быть общий тип (число, флаг, строка), и в этом случае параметры для свойства будут принудительно преобразованы к правильному типу. Ошибка возникает, если такое принудительное преобразование невозможно. Как вариант, ссылка на свойство может задавать диапазон допустимых значений, таких как Discard, PairAndDiscard и IncludeAsText, и в этом случае ошибка возникает при использовании другого значения. Свойства флага должны читаться или задаваться с использованием значений True и False. В справочных таблицах этого руководства структурированные свойства указаны в столбце *Описание свойств*, приводятся также форматы их использования.

Общие свойства узлов

Многие свойства общие для всех узлов (в том числе надузлов) в IBM SPSS Modeler.

Таблица 28. Общие свойства узлов.

Имя свойства	Тип переменной	Описание свойства
use_custom_name	логический	
name	строка	Предназначенное только для чтения свойство, по которому читается имя для узла (автоматически сгенерированное или пользовательское) на холсте.
custom_name	строка	Задает пользовательское имя для узла.
tooltip	строка	
annotation	строка	
keywords	строка	Структурированный слот, задающий список ключевых слов, связанных с объектом
cache_enabled	логический	
node_type	source_supernode process_supernode terminal_supernode Все имена узлов, как задано для сценариев	Предназначенное только для чтения свойство, используемое для ссылки на узел по типу. Например, вместо ссылки на узел только по имени, как real_income, можно задать и тип, такой как userInput или filter.

Специфичные для надузла свойства обсуждаются отдельно от всех других узлов. Дополнительную информацию смотрите в разделе Глава 19, “Свойства надузлов”, на стр. 241.

Глава 8. Свойства потока

При написании сценариев можно управлять многими свойствами потоков.

Сценарий может работать с текущим потоком при помощи функции `stream()` в модуле `modeler.script`, например:

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

Для указания на свойства потока необходимо использовать специальную переменную потока, обозначаемую символом `^` перед потоком.

Свойство узлов используется для ссылки на узлы в текущем потоке.

Свойства потока описываются в следующей таблице.

Таблица 29. Свойства потока.

Имя свойства	Тип переменной	Описание свойства
<code>execute_method</code>	Нормальный Сценарий	
<code>date_format</code>	"ДДММГГ" "ММДДГГ" "ГГММДД" "ГГГГММДД" "ГГГГДД" ДЕНЬ МЕСЯЦ "ДД-ММ-ГГ" "ДД-ММ-ГГГГ" "ММ-ДД-ГГ" "ММ-ДД-ГГГГ" "ДД-МЕС-ГГ" "ДД-МЕС-ГГГГ" "ГГГГ-ММ-ДД" "ДД.ММ.ГГ" "ДД.ММ.ГГГГ" "ММ.ДД.ГГ" "ММ.ДД.ГГГГ" "ДД.МЕС.ГГ" "ДД.МЕС.ГГГГ" "ДД/ММ/ГГ" "ДД/ММ/ГГГГ" "ММ/ДД/ГГ" "ММ/ДД/ГГГГ" "ДД/МЕС/ГГ" "ДД/МЕС/ГГГГ" МЕС ГГГГ к К ГГГГ нн НД ГГГГ	
<code>date_baseline</code>	<i>число</i>	
<code>date_2digit_baseline</code>	<i>число</i>	

Таблица 29. Свойства потока (продолжение).

Имя свойства	Тип переменной	Описание свойства
time_format	"ччммсс" "ччмм" "ммсс" "чч:мм:сс" "чч:мм" "мм:сс" "(ч)ч:(м)м:(с)с" "(ч)ч:(м)м" "(м)м:(с)с" "чч.мм.сс" "чч.мм" "мм.сс" "(ч)ч.(м)м.(с)с" "(ч)ч.(м)м" "(м)м.(с)с"	
time_rollover	логический	
import_datetime_as_string	логический	
decimal_places	число	
decimal_symbol	По умолчанию Точка Запятая	
angles_in_radians	логический	
use_max_set_size	логический	
max_set_size	число	
ruleset_evaluation	Голосование FirstHit	
refresh_source_nodes	логический	Используйте для автоматического обновления узлов источника после выполнения потока.
script	строка	
script_language	Python Legacy	Задаёт язык сценария потока.
annotation	строка	
encoding	SystemDefault "UTF-8"	
stream_rewriting	логический	
stream_rewriting_maximise_sql	логический	
stream_rewriting_optimise_clem_execution	логический	
stream_rewriting_optimise_syntax_execution	логический	
enable_parallelism	логический	
sql_generation	логический	
database_caching	логический	
sql_logging	логический	
sql_generation_logging	логический	
sql_log_native	логический	
sql_log_prettyprint	логический	

Таблица 29. Свойства потока (продолжение).

Имя свойства	Тип переменной	Описание свойства
record_count_suppress_input	<i>логический</i>	
record_count_feedback_interval	<i>целое</i>	
use_stream_auto_create_node_параметры	<i>логическое</i>	При значении true используются параметры, заданные для потока; в противном случае используются предпочтения пользователя.
create_model_applier_for_new_модели	<i>логическое</i>	Если значение - true, при создании в построителе моделей новой модели без активных ссылок обновления добавляется новый применитель модели.
create_model_applier_update_links	createEnabled createDisabled doNotCreate	Задает тип создаваемой ссылки при автоматическом добавлении узла применителя модели.
create_source_node_from_builders	<i>логическое</i>	Если значение - true, при создании в построителе источников нового вывода источника без активных ссылок обновления добавляется новый узел источника.
create_source_node_update_links	createEnabled createDisabled doNotCreate	Задает тип создаваемой ссылки при автоматическом добавлении узла источника.

Глава 9. Свойства узла источника

Общие свойства узлов источников

Ниже перечислены все свойства, общие для узлов источников, а информация о конкретных узлах представлена в следующих разделах.

Таблица 30. Общие свойства узлов источников.

Имя свойства	Тип переменной	Описание свойства
direction	Ввод Назначение Both Нет Подмножества Разбиения Частота RecordID	Ключевое свойство для ролей полей. <i>Примечание:</i> Значения In и Out в настоящее время объявлены устаревшими. Их поддержка может быть прекращена в следующем выпуске.
type	Диапазон Флаг Установить Без типа Дискретное Порядковое По умолчанию	Тип поля. При задании этого свойства <i>По умолчанию</i> будут очищены все параметры свойства значения, и если для режим_значения будет установлено <i>Задать</i> , эта настройка сбросится до <i>Читать</i> . Если режим_значения уже задан как <i>Передать</i> или <i>Читать</i> , на него не будет влиять параметр тип.
storage	Нет данных Строка Целое Действительное число Время Дата Отметка времени	Предназначенное только для чтения ключевое свойство для типа хранения поля.
check	Нет Аннулировать Принуждать Исключение Предупреждение Прервать	Ключевое свойство для проверки типа и диапазона поля.
values	[значение значение]	Для количественного поля (диапазона) первое значение - это минимум, а последнее - максимум. Для номинальных полей (набора) задайте все значения. Для флаговых полей первое значение представляет <i>false</i> , а последнее - <i>true</i> . Задание этого свойства автоматически устанавливает для свойства режим_значения значение <i>Задать</i> .
value_mode	Чтение Успех Read+ Текущий Задать	Определяет, как при следующем проходе данных устанавливаются значения для поля. Обратите внимание на то, что вы не можете автоматически установить для этого свойства значение <i>Задать</i> ; чтобы использовать конкретные значения, задайте свойство значения.
default_value_mode	Чтение Успех	Задаёт способ по умолчанию для задания значений для всех полей. Этот параметр может быть перезаписан для конкретных полей с помощью свойства режим_значения.

Таблица 30. Общие свойства узлов источников (продолжение).

Имя свойства	Тип переменной	Описание свойства
extend_values	логический	Применяется, когда для режим_значения задано Чтение. Задайте T, чтобы добавить вновь прочитанные значения к любым существующим значениям для этого поля. Задайте F, чтобы отбросить существующие значения и заменить их на вновь прочитанные значения.
value_labels	строка	Используется, чтобы задать метку значения. Обратите внимание на то, что сначала должны быть заданы значения.
enable_missing	логический	Когда задано T, активирует отслеживание пропущенных значений для поля.
missing_values	[значение значение ...]	Задаёт значения данных, отмечающие пропущенные данные.
range_missing	логический	Когда для этого свойства задано T, указывает, определен ли для этого поля диапазон пропущенных (пустых) значений.
missing_lower	строка	Когда для значения диапазон_отсутствия задано true, указывает нижнюю границу диапазона значений отсутствия.
missing_upper	строка	Когда для значения диапазон_отсутствия задано true, указывает верхнюю границу диапазона значений отсутствия.
null_missing	логический	Когда для этого свойства задано T, значения nulls (не определенные значения, обозначаемые в программах как \$null\$) рассматриваются как значения отсутствия.
whitespace_missing	логический	Когда для этого свойства задано T, значения, содержащие только пробельные символы (пробелы, знаки табуляции и новой строки) рассматриваются как значения отсутствия.
description	строка	Использовано, чтобы задать метку или описание поля.
default_include	логический	Ключевое свойство для указания, каким будет поведение по умолчанию, передать или отфильтровать поля:
include	логический	Ключевое свойство, используемое для определения, включаются или отфильтровываются индивидуальные поля:
new_name	строка	

Свойства узла asimport

При помощи источника Analytic Server (сервер аналитических служб) поток можно выполнить в файловой системе HDFS (Hadoop Distributed File System).

Таблица 31. свойства узла asimport.

Свойства узла asimport	Тип переменной	Описание свойства
data_source	строка	Имя источника данных.
host	строка	Имя хоста Analytic Server (сервер аналитических служб).
port	целое	Порт, на котором ожидает приема информации Analytic Server (сервер аналитических служб).

Таблица 31. свойства узла *asimport* (продолжение).

Свойства узла <i>asimport</i>	Тип переменной	Описание свойства
tenant	строка	В мультиарендной среде - имя того арендатора, которому вы принадлежите. В одноарендной среде используется значение по умолчанию - ibm .
set_credentials	логическое	Если значения аутентификации пользователя для Analytic Server (сервер аналитических служб) такие же, как на сервере SPSS Modeler, задайте для этого параметра значение false . В противном случае задайте значение true .
user_name	строка	Имя пользователя для регистрации в Analytic Server (сервер аналитических служб). Нужно только в том случае, если значение set_credentials - true.
password	строка	Пароль для регистрации в Analytic Server (сервер аналитических служб). Нужен только в том случае, если значение set_credentials - true.

Свойства узла *cognosimport*



Узел источника IBM Cognos BI импортирует данные из баз данных Cognos BI.

Таблица 32. Свойства узла *cognosimport*.

Свойства узла <i>cognosimport</i>	Тип переменной	Описание свойства
mode	Данные Отчет	Задаёт, импортировать ли данные Cognos BI (по умолчанию), или отчеты.
cognos_connection	{ <i>"строка"</i> , <i>boolean</i> , <i>"строка"</i> , <i>"строка"</i> , <i>"строка"</i> }	Свойство списка, содержащего подробности соединения для сервера Cognos. Формат следующий: { <i>"URL_сервера_Cognos"</i> , <i>режим_регистрации</i> , <i>"пространство_имен"</i> , <i>"имя_пользователя"</i> , <i>"пароль"</i> } где: <i>URL_сервера_Cognos</i> - это URL сервера Cognos, содержащего источник данных <i>режим_регистрации</i> обозначает, используется ли анонимная регистрация и может быть значением true или false; если задано true, следующие поля должны быть заданы как <i>"пространство_имен"</i> задает провайдера аутентификации защиты, используемого для регистрации на сервере <i>имя_пользователя</i> и <i>пароль</i> - это значения для регистрации на сервере Cognos
cognos_package_name	строка	Путь и имя пакета Cognos, из которого импортируются объекты данных, например: /Public Folders/GOSALES Примечание: Допустимы символы только прямой дробной черты.

Таблица 32. Свойства узла *cognosimport* (продолжение).

Свойства узла <i>cognosimport</i>	Тип переменной	Описание свойства
<i>cognos_items</i>	{ <i>"поле"</i> , <i>"поле"</i> , ... <i>"поле"</i> }	Имя одного или нескольких объектов данных, которые будут импортированы. Формат <i>поле</i> следующий: [<i>пространство_имен</i>].[<i>субъект_запроса</i>].[<i>элемент_запроса</i>]
<i>cognos_filters</i>	<i>поле</i>	Имя одного или нескольких фильтров для применения перед импортом данных.
<i>cognos_data_parameters</i>	<i>список</i>	Значения для предложения параметров данных. Пары имя-и-значение заключены в фигурные скобки, а несколько пар разделяются запятыми, и вся строка заключается в квадратные скобки. Формат: [{" <i>парам1</i> ", " <i>значение</i> "}, ..., {" <i>парамN</i> ", " <i>значение</i> "}]
<i>cognos_report_directory</i>	<i>поле</i>	Путь Cognos к папке или пакету, из которой будут импортироваться отчеты, например: /Public Folders/GOSALES Примечание: Допустимы символы только прямой дробной черты.
<i>cognos_report_name</i>	<i>поле</i>	Путь и имя в положении отчетов того отчета, который будет импортироваться.
<i>cognos_report_parameters</i>	<i>список</i>	Значения для параметров отчета. Пары имя-и-значение заключены в Формат: [{" <i>парам1</i> ", " <i>значение</i> "}, ..., {" <i>парамN</i> ", " <i>значение</i> "}]

Свойства узла *database*



Узел базы данных можно использовать для импорта данных из множества других пакетов при помощи ODBC (Open Database Connectivity), в том числе Microsoft SQL Server, DB2, Oracle и других.

Таблица 33. Свойства узла *database*.

Свойства узла <i>database</i>	Тип переменной	Описание свойства
<i>mode</i>	Таблица Query	Укажите <i>Таблицу</i> для соединения с таблицей базы данных, используя элементы управления диалогового окна, или задайте <i>Запрос</i> для запроса в выбранной базе данных, используя SQL.
<i>datasource</i>	<i>строка</i>	Имя базы данных (смотрите также примечание ниже).
<i>username</i>	<i>строка</i>	Подробности соединения с базой данных (смотрите также примечания ниже).
<i>password</i>	<i>строка</i>	
<i>epassword</i>	<i>строка</i>	Задаёт закодированный пароль как альтернативу жестко закодированному в сценарии паролю. Дополнительную информацию смотрите в разделе “Генерирование закодированного пароля” на стр. 47. При выполнении это свойство предназначено только для чтения.

Таблица 33. Свойства узла database (продолжение).

Свойства узла database	Тип переменной	Описание свойства
tablename	строка	Имя таблицы, к которой вы хотите обратиться.
strip_spaces	Нет Слева Справа Both	Опции для отброса начальных и завершающих пробелов в строках.
use_quotes	AsNeeded Всегда Никогда	Задайте, заключаются ли имена таблиц и столбцов в кавычки при отправлении запросов в базу данных (например, если они содержат пробелы или знаки пунктуации).
query	строка	Задаёт код SQL для запроса, который вы хотите передать.

Примечание: Если имя базы данных (в свойстве источник_данных) содержит пробелы, вместо индивидуальных свойств для источника_данных, имени_пользователя и пароля используйте одно свойство источника данных в следующем формате:

Таблица 34. Свойства узла базы данных - для конкретного источника данных.

Свойства узла database	Тип переменной	Описание свойства
datasource	строка	<p>Формат: [имя_базы_данных, имя_пользователя, пароль [,true false]]</p> <p>Последний параметр предназначен для использования с зашифрованными паролями. Если для него задано значение true, перед использованием пароль будет расшифрован.</p>

Используйте этот формат также в том случае, если вы изменяете источник данных; однако если нужно изменить только имя пользователя или пароль, можно использовать свойства имя_пользователя или пароль.

Свойства узла datacollectionimport



Узел импорта данных IBM SPSS Data Collection импортирует материалы обследования на основании модели данных IBM SPSS Data Collection, используемой продуктами изучения рынка IBM Corp.. Для использования этого узла должна быть установлена библиотека данных IBM SPSS Data Collection.

Рисунок 7. Узел импорта данных измерений

Таблица 35. Свойства узла `datacollectionimport`.

Свойства узла <code>datacollectionimport</code>	Тип переменной	Описание свойства
<code>metadata_name</code>	<i>строка</i>	Имя MDSC. Специальное значение <code>DimensionsMDD</code> указывает, что нужно использовать стандартный документ метаданных IBM SPSS Data Collection. Другие возможные значения включают в себя: <code>mrAD0Dsc</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> Специальное значение <code>none</code> означает, что нет MDSC.
<code>metadata_file</code>	<i>строка</i>	Имя файла, где хранятся метаданные.
<code>casedata_name</code>	<i>строка</i>	Имя CDSC. Возможные значения включают в себя: <code>mrAD0Dsc</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>mrXm1Dsc</code> Специальное значение <code>none</code> означает, что нет CDSC.
<code>casedata_source_type</code>	Нет данных File Папка UDL DSN	Указывает тип источника CDSC.
<code>casedata_file</code>	<i>строка</i>	Когда <code>casedata_source_type</code> - это <i>Файл</i> , задает файл, содержащий данные наблюдения.
<code>casedata_folder</code>	<i>строка</i>	Когда <code>casedata_source_type</code> - это <i>Папка</i> , задает папку, содержащую данные наблюдения.
<code>casedata_udl_string</code>	<i>строка</i>	Когда <code>casedata_source_type</code> - это <i>UDL</i> , задает строку соединения OLD-DB для источника данных, содержащего данные наблюдения.
<code>casedata_dsn_string</code>	<i>строка</i>	Когда <code>casedata_source_type</code> - это <i>DSN</i> , задает строку соединения ODBC для источника данных.
<code>casedata_project</code>	<i>строка</i>	При чтении данных наблюдения из базы данных IBM SPSS Data Collection можно ввести имя проекта. Для всех остальных типов данных наблюдений значение этого параметра следует оставить пустым.

Таблица 35. Свойства узла `datacollectionimport` (продолжение).

Свойства узла <code>datacollectionimport</code>	Тип переменной	Описание свойства
<code>version_import_mode</code>	All Последняя Задать	Определяет, как должны обрабатываться версии.
<code>specific_version</code>	<i>строка</i>	Когда <code>version_import_mode</code> - это <i>Specify</i> , определяет версию данных наблюдения для импорта.
<code>use_language</code>	<i>строка</i>	Определяет, должны ли использоваться метки конкретного языка.
язык	<i>строка</i>	Если значение <code>use_language</code> - это <code>true</code> , определяет используемый при импорте код языка. Код языка должен быть одним из доступных в данных наблюдения.
<code>use_context</code>	<i>строка</i>	Определяет, нужно ли импортировать конкретный контекст. Контексты используются для изменения описания, связанного с откликами.
<code>context</code>	<i>строка</i>	Если значение <code>use_context</code> - это <code>true</code> , определяет контекст для импорта. Контекст должен быть одним из доступных в данных наблюдения.
<code>use_label_type</code>	<i>строка</i>	Определяет, нужно ли импортировать конкретный тип метки.
<code>label_type</code>	<i>строка</i>	Если значение <code>use_label_type</code> - это <code>true</code> , определяет тип метки для импорта. Тип метки должен быть одним из доступных в данных наблюдения.
<code>user_id</code>	<i>строка</i>	Для баз данных, требующих непосредственной регистрации, можно предоставить ID пользователя и пароль для доступа к источнику данных.
<code>password</code>	<i>строка</i>	
<code>import_system_variables</code>	Общий Нет All	Задаёт, какие системные переменные импортируются.
<code>import_codes_variables</code>	<i>логический</i>	
<code>import_sourcefile_variables</code>	<i>логический</i>	
<code>import_multi_response</code>	MultipleFlags Одиночный	

Свойства узла `excelimport`



Узел импорта Excel импортирует данные из любой версии Microsoft Excel. Источник данных ODBC не требуется.

Таблица 36. Свойства узла *excelimport*.

Свойства узла <i>excelimport</i>	Тип переменной	Описание свойства
<i>excel_file_type</i>	Excel2003 Excel2007	
<i>full_filename</i>	<i>строка</i>	Полное имя файла, включающее путь.
<i>use_named_range</i>	<i>Логический</i>	Использовать ли названный диапазон. При значении true свойство <i>named_range</i> используется для задания диапазона чтения, а другие параметры данных и рабочего листа игнорируются.
<i>named_range</i>	<i>строка</i>	
<i>worksheet_mode</i>	Индекс Имя	Задаёт, чем определяется рабочий лист, индексом или именем.
<i>worksheet_index</i>	<i>целое</i>	Индекс рабочего листа для чтения, начинается с 0 для первого листа, 1 - для второго и так далее.
<i>worksheet_name</i>	<i>строка</i>	Имя рабочего листа для чтения.
<i>data_range_mode</i>	FirstNonBlank ExplicitRange	Задаёт, как нужно определить диапазон.
<i>blank_rows</i>	StopReading ReturnBlankRows	Когда значение <i>data_range_mode</i> - это <i>FirstNonBlank</i> , определяет, как должны обрабатываться пустые строки.
<i>explicit_range_start</i>	<i>строка</i>	Когда значение <i>data_range_mode</i> - это <i>ExplicitRange</i> , задаёт начальную точку диапазона для чтения.
<i>explicit_range_end</i>	<i>строка</i>	
<i>read_field_names</i>	<i>Логический</i>	Задаёт, должна ли первая строка в заданном диапазоне использоваться для имен полей (столбцов).

Свойства узла *evimport*



Узел Представление предприятия создает соединение с IBM SPSS Collaboration and Deployment Services Repository, позволяя прочесть данные представления предприятия в поток и создать пакет модели в сценарии, к которому другим пользователям можно обратиться из репозитория.

Таблица 37. Свойства узла *evimport*.

Свойства узла <i>evimport</i>	Тип переменной	Описание свойства
<i>connection</i>	<i>список</i>	Структурированное свойство, список параметров для установки соединения с представлением предприятия.
<i>tablename</i>	<i>строка</i>	Имя таблицы в представлении прикладной программы.

Свойства узла fixedfile



Узел фиксированного файла импортирует данные из текстовых файлов с фиксированными полями, то есть, файлов, поля которых не разделяются, но начинаются с одного положения и у них фиксированная длина. Сгенерированные компьютером или устаревшие данные часто хранятся в формате фиксированных полей.

Таблица 38. Свойства узла fixedfile.

Свойства узла fixedfile	Тип переменной	Описание свойства
record_len	число	Задаёт количество символов в каждой записи.
line_oriented	логический	Пропускает символ новой строки в конце каждой записи.
decimal_symbol	По умолчанию Запятая Точка	Тип десятичного разделителя, используемого в вашем источнике данных.
skip_header	число	Задаёт число строк, которые будут игнорироваться в начале первой записи. Полезно, чтобы игнорировать заголовки столбцов.
auto_recognize_datetime	логический	Задаёт, идентифицируются ли автоматически в исходных данных значения даты и времени.
lines_to_scan	число	
fields	список	Структурированное свойство.
full_filename	строка	Полное имя файла для чтения, включая каталог.
strip_spaces	Нет Слева Справа Both	Отбрасывает начальные и завершающие пробелы в строках для импорта.
invalid_char_mode	Исключение Replace	Удаляет недопустимые символы (null, 0 или любой символ, которого нет в текущей кодировке) из входных данных или замещает недопустимые символы на заданный однозначный символ.
invalid_char_replacement	строка	
use_custom_values	логический	
custom_storage	Нет данных Строка Целое Действительное число Время Дата Отметка времени	

Таблица 38. Свойства узла *fixedfile* (продолжение).

Свойства узла <i>fixedfile</i>	Тип переменной	Описание свойства
custom_date_format	"ДДММГГ" "ММДДГГ" "ГГММДД" "ГГГГММДД" "ГГГГДДД" ДЕНЬ МЕСЯЦ "ДД-ММ-ГГ" "ДД-ММ-ГГГГ" "ММ-ДД-ГГ" "ММ-ДД-ГГГГ" "ДД-МЕС-ГГ" "ДД-МЕС-ГГГГ" "ГГГГ-ММ-ДД" "ДД.ММ.ГГ" "ДД.ММ.ГГГГ" "ММ.ДД.ГГ" "ММ.ДД.ГГГГ" "ДД.МЕС.ГГ" "ДД.МЕС.ГГГГ" "ДД/ММ/ГГ" "ДД/ММ/ГГГГ" "ММ/ДД/ГГ" "ММ/ДД/ГГГГ" "ДД/МЕС/ГГ" "ДД/МЕС/ГГГГ" МЕС ГГГГ к К ГГГГ нн НД ГГГГ	Это свойство применимо только в том случае, если была задана пользовательская система хранения.
custom_time_format	"ЧЧММСС" "ЧЧММ" "ММСС" "ЧЧ:ММ:СС" "ЧЧ:ММ" "ММ:СС" "(Ч)Ч:(М)М:(С)С" "(Ч)Ч:(М)М" "(М)М:(С)С" "ЧЧ.ММ.СС" "ЧЧ.ММ" "ММ.СС" "(Ч)Ч.(М)М.(С)С" "(Ч)Ч.(М)М" "(М)М.(С)С"	Это свойство применимо только в том случае, если была задана пользовательская система хранения.
custom_decimal_symbol	<i>поле</i>	Это свойство применимо только в том случае, если была задана пользовательская система хранения.
encoding	StreamDefault SystemDefault "UTF-8"	Задает способ текстового кодирования.

Свойства узла sasimport



Узел импорта SAS импортирует данные SAS в IBM SPSS Modeler.

Таблица 39. Свойства узла sasimport.

Свойства узла sasimport	Тип переменной	Описание свойства
формат	Windows UNIX Transport SAS7 SAS8 SAS9	Формат файла для импорта.
full_filename	строка	Полное вводимое имя файла, в том числе его путь.
member_name	строка	Задать участника для импорта из заданного транспортного файла SAS.
read_formats	логический	Читает форматы данных (такие как метки переменных) из заданного файла форматов.
full_format_filename	строка	
import_names	NamesAndLabels LabelsasNames	Задает способ для отображения имен переменных и меток при импорте.

Свойства узла simgen



Узел генерирования имитации обеспечивает удобный путь сгенерировать имитационные данные - либо с нуля, используя указанные пользователем статистические распределения, либо автоматически, используя распределения, полученные при выполнении узла подгонки имитации для существующих данных хронологии. Это полезно, когда нужно оценить вывод прогнозной модели при наличии неопределенности во входных данных модели.

Таблица 40. Свойства узла simgen.

Свойства узла simgen	Тип переменной	Описание свойства
fields	Структурированное свойство	
корреляции	Структурированное свойство	
max_cases	целое	Минимальное значение - 1000, максимальное значение - 2 147 483 647
create_iteration_field	логическое	
iteration_field_name	строка	
replicate_results	логическое	
random_seed	целое	
overwrite_when_refitting	логическое	
parameter_xml	строка	Возвращает параметр Xml как строку

Таблица 40. Свойства узла *simgen* (продолжение).

Свойства узла <i>simgen</i>	Тип переменной	Описание свойства
распределение	Бернулли Бета Биномиальное Категориальная Экспоненциальная Фиксированная Гамма Логнормальное NegativeBinomialFailures NegativeBinomialTrials Нормальное Пуассона Диапазон Треугольник Равные Вейбулл	
bernoulli_prob	число	$0 \leq \text{bernoulli_prob} \leq 1$
beta_shape1	число	Должно быть ≥ 0
beta_shape2	число	Должно быть ≥ 0
beta_min	число	Необязательно. Должно быть меньше, чем beta_max.
beta_max	число	Необязательно. Должно быть больше, чем beta_min.
binomial_n	целое	Должно быть > 0
binomial_prob	число	$0 \leq \text{binomial_prob} \leq 1$
binomial_min	число	Необязательно. Должно быть меньше, чем binomial_max.
binomial_max	число	Необязательно. Должно быть больше, чем binomial_min.
exponential_scale	число	Должно быть > 0
exponential_min	число	Необязательно. Должно быть меньше, чем exponential_max.
exponential_max	число	Необязательно. Должно быть больше, чем exponential_min.
fixed_value	строка	
gamma_shape	число	Должно быть ≥ 0
gamma_scale	число	Должно быть ≥ 0
gamma_min	число	Необязательно. Должно быть меньше, чем gamma_max.
gamma_max	число	Необязательно. Должно быть больше, чем gamma_min.
lognormal_shape1	число	Должно быть ≥ 0
lognormal_shape2	число	Должно быть ≥ 0
lognormal_min	число	Необязательно. Должно быть меньше, чем lognormal_max.
lognormal_max	число	Необязательно. Должно быть больше, чем lognormal_min.
negative_bin_failures_threshold	число	Должно быть ≥ 0

Таблица 40. Свойства узла *simgen* (продолжение).

Свойства узла <i>simgen</i>	Тип переменной	Описание свойства
<code>negative_bin_failures_prob</code>	<i>число</i>	$0 \leq \text{negative_bin_failures_prob} \leq 1$
<code>negative_bin_failures_min</code>	<i>число</i>	Необязательно. Должно быть меньше, чем <code>negative_bin_failures_max</code> .
<code>negative_bin_failures_max</code>	<i>число</i>	Необязательно. Должно быть больше, чем <code>negative_bin_failures_min</code> .
<code>negative_bin_trials_threshold</code>	<i>число</i>	Должно быть ≥ 0
<code>negative_bin_trials_prob</code>	<i>число</i>	$0 \leq \text{negative_bin_trials_prob} \leq 1$
<code>negative_bin_trials_min</code>	<i>число</i>	Необязательно. Должно быть меньше, чем <code>negative_bin_trials_max</code> .
<code>negative_bin_trials_max</code>	<i>число</i>	Необязательно. Должно быть меньше, чем <code>negative_bin_trials_min</code> .
<code>normal_mean</code>	<i>число</i>	
<code>normal_sd</code>	<i>число</i>	Должно быть > 0
<code>normal_min</code>	<i>число</i>	Необязательно. Должно быть меньше, чем <code>normal_max</code> .
<code>normal_max</code>	<i>число</i>	Необязательно. Должно быть больше, чем <code>normal_min</code> .
<code>poisson_mean</code>	<i>число</i>	Должно быть ≥ 0
<code>poisson_min</code>	<i>число</i>	Необязательно. Должно быть меньше, чем <code>poisson_max</code> .
<code>poisson_max</code>	<i>число</i>	Необязательно. Должно быть больше, чем <code>poisson_min</code> .
<code>triangular_mode</code>	<i>число</i>	$\text{triangular_min} \leq \text{triangular_mode} \leq \text{triangular_max}$
<code>triangular_min</code>	<i>число</i>	Должно быть меньше, чем <code>triangular_mode</code>
<code>triangular_max</code>	<i>число</i>	Должно быть больше, чем <code>triangular_mode</code>
<code>uniform_min</code>	<i>число</i>	Должно быть меньше, чем <code>uniform_max</code>
<code>uniform_max</code>	<i>число</i>	Должно быть больше, чем <code>uniform_min</code>
<code>weibull_rate</code>	<i>число</i>	Должно быть ≥ 0
<code>weibull_scale</code>	<i>число</i>	Должно быть ≥ 0
<code>weibull_location</code>	<i>число</i>	Должно быть ≥ 0
<code>weibull_min</code>	<i>число</i>	Необязательно. Должно быть меньше, чем <code>weibull_max</code> .
<code>weibull_max</code>	<i>число</i>	Необязательно. Должно быть больше, чем <code>weibull_min</code> .

Корреляция может быть любым числом от +1 до -1. Можно задать любое число корреляций. Незаданной корреляции присваивается значение ноль. Если какие-то из полей неизвестны, значение корреляции должно быть задано в корреляционной матрице (или таблице) и текст показан красным. Если есть неизвестные поля, выполнить узел нельзя.

Свойства узла statisticsimport



Узел Файл IBM SPSS Statistics читает данные из файла формата `.sav`, используемого IBM SPSS Statistics, а также файлы кэша, сохраненные в IBM SPSS Modeler, которые также используют тот же формат.

Свойства этого узла описаны в разделе “Свойства узла statisticsimport” на стр. 237.

Свойства узла userinput



Узел пользовательского ввода представляет простой способ создания синтетических данных, или от нуля, или изменением существующих данных. Например, это полезно, если вы хотите создать испытательный набор данных для моделирования.

Таблица 41. Свойства узла userinput.

Свойства узла userinput	Тип переменной	Описание свойства
data		Данные для каждого поля могут быть разной длины, но должны быть согласованы с объемом, занимаемым полем в системе хранения. Задание значений для несуществующего поля создает это поле. Кроме этого, задание для значений поля пустой строки (" ") удаляет заданное поле. Примечание: Введенные для этого свойства значения должны быть строками, а не числами. Например, числа 1, 2, 3 и 4 надо вводить как "1 2 3 4".
names		Структурированный слот, устанавливающий или возвращающий список имен полей, сгенерированных узлом.
custom_storage	Нет данных Строка Целое Действительное число Время Дата Отметка времени	Ключевой слот, задающий или возвращающий систему хранения для поля.
data_mode	Combined Ordered	Если задана опция Combined, записи генерируются для каждого объединения заданных значений и минимального/максимального значений. Количество сгенерированных записей равно произведению числа значений в каждом поле. Если задана опция Упорядоченная, из каждого столбца берется одно значение для каждой записи по порядку, чтобы сгенерировать строку данных. Количество сгенерированных записей равно максимальному числу значений, связанных с полем. Любые поля с меньшим числом данных будут дополнены значениями null.

Таблица 41. Свойства узла *userinput* (продолжение).

Свойства узла <i>userinput</i>	Тип переменной	Описание свойства
values		Это свойство было объявлено устаревшим с заменой на <i>data</i> и больше не должно использоваться.

Свойства узла *variablefile*



Узел файла переменных читает данные из текстовых файлов со свободными полями, то есть, такие файлы, записи которых содержат фиксированное количество полей, но переменное число символов. Этот узел полезен также для файлов с текстовыми заголовками фиксированной длины и с некоторыми типами аннотаций.

Таблица 42. Свойства узла *variablefile*.

Свойства узла <i>variablefile</i>	Тип переменной	Описание свойства
skip_header	число	Задаёт число символов, которые будут игнорироваться в начале первой записи.
num_fields_auto	логический	Автоматически определяет количество полей в каждой записи. Записи должны быть ограничены символом новой строки.
num_fields	число	Вручную задаёт количество полей в каждой записи.
delimit_space	логический	Задаёт символ, который будет использоваться для разделения границ полей в файле.
delimit_tab	логический	
delimit_new_line	логический	
delimit_non_printing	логический	
delimit_comma	логический	В случаях, когда запятая - это и разделитель полей, и десятичный разграничитель для потоков, задайте для <i>delimit_other</i> значение <i>true</i> и укажите запятую в качестве разделителя, используя свойство <i>other</i> .
delimit_other	логический	Позволяет задать пользовательский разделитель, используя свойство <i>other</i> .
other	строка	Задаёт разделитель, используемый при установленном для <i>delimit_other</i> значении <i>true</i> .
decimal_symbol	По умолчанию Запятая Точка	Задаёт десятичный разделитель, используемый в источнике данных.
multi_blank	логический	Рассматривает несколько смежных пустых символов разделителя как один разделитель.
read_field_names	логический	Рассматривает первую строку в файле данных как метки для столбца.
strip_spaces	Нет Слева Справа Both	Отбрасывает начальные и завершающие пробелы в строках при импорте.

Таблица 42. Свойства узла variablefile (продолжение).

Свойства узла variablefile	Тип переменной	Описание свойства
invalid_char_mode	Исключение Replace	Удаляет недопустимые символы (null, 0 или любой символ, которого нет в текущей кодировке) из входных данных или замещает недопустимые символы на заданный однозначный символ.
invalid_char_replacement	<i>строка</i>	
break_case_by_newline	<i>логическое</i>	Задаёт, что разделитель строк - это символ новой строки.
lines_to_scan	<i>число</i>	Задаёт, сколько строк просматривать для заданных типов данных.
auto_recognize_datetime	<i>логический</i>	Задаёт, идентифицируются ли автоматически в исходных данных значения даты или времени.
quotes_1	Исключение PairAndDiscard IncludeAsText	Задаёт, как рассматриваются при импорте одинарные кавычки.
quotes_2	Исключение PairAndDiscard IncludeAsText	Задаёт, как рассматриваются при импорте двойные кавычки.
full_filename	<i>строка</i>	Полное имя файла для чтения, включая каталог.
use_custom_values	<i>логический</i>	
custom_storage	Нет данных Строка Целое Действительное число Время Дата Отметка времени	

Таблица 42. Свойства узла `variablefile` (продолжение).

Свойства узла <code>variablefile</code>	Тип переменной	Описание свойства
<code>custom_date_format</code>	"ДДММГГ" "ММДДГГ" "ГГММДД" "ГГГГММДД" "ГГГГДДД" ДЕНЬ МЕСЯЦ "ДД-ММ-ГГ" "ДД-ММ-ГГГГ" "ММ-ДД-ГГ" "ММ-ДД-ГГГГ" "ДД-МЕС-ГГ" "ДД-МЕС-ГГГГ" "ГГГГ-ММ-ДД" "ДД.ММ.ГГ" "ДД.ММ.ГГГГ" "ММ.ДД.ГГ" "ММ.ДД.ГГГГ" "ДД.МЕС.ГГ" "ДД.МЕС.ГГГГ" "ДД/ММ/ГГ" "ДД/ММ/ГГГГ" "ММ/ДД/ГГ" "ММ/ДД/ГГГГ" "ДД/МЕС/ГГ" "ДД/МЕС/ГГГГ" МЕС ГГГГ к К ГГГГ нн НД ГГГГ	Это свойство применимо только в том случае, если была задана пользовательская система хранения.
<code>custom_time_format</code>	"ЧЧММСС" "ЧЧММ" "ММСС" "ЧЧ:ММ:СС" "ЧЧ:ММ" "ММ:СС" "(Ч)Ч:(М)М:(С)С" "(Ч)Ч:(М)М" "(М)М:(С)С" "ЧЧ.ММ.СС" "ЧЧ.ММ" "ММ.СС" "(Ч)Ч.(М)М.(С)С" "(Ч)Ч.(М)М" "(М)М.(С)С"	Это свойство применимо только в том случае, если была задана пользовательская система хранения.
<code>custom_decimal_symbol</code>	<i>поле</i>	Это свойство применимо только в том случае, если была задана пользовательская система хранения.
<code>encoding</code>	StreamDefault SystemDefault "UTF-8"	Задаёт способ текстового кодирования.

Свойства узла xmlimport



Узел источника XML импортирует данные в формате XML в поток. Вы можете импортировать в каталог один файл или все файлы. Дополнительно вы можете задать файл схемы, из которой можно прочесть структуру XML.

Таблица 43. Свойства узла xmlimport.

Свойства узла xmlimport	Тип переменной	Описание свойства
read	single directory	Читает один файл данных (по умолчанию) или все файлы XML в каталоге.
recurse	логический	Задаёт, читать ли дополнительно файлы XML из всех подкаталогов заданного каталога.
full_filename	строка	(обязательно) Полный путь и имя файла импорта XML (если read = single).
directory_name	строка	(обязательно) Полный путь и имя каталога, из которого будут импортироваться файлы XML (если read = directory).
full_schema_filename	строка	Полный путь и имя файла XSD или DTD, из которого будет читаться структура XML. Если опустить этот параметр, структура будет читаться из файла источника XML.
records	строка	Выражение XPath (например, /author/name) для определения границы записи. При всякой встрече этого элемента в файле источника будет создаваться новая запись.
mode	read specify	Читает все записи (по умолчанию) или задать, какие элементы читать.
fields		Список объектов (элементов и атрибутов) для импорта. Каждый элемент в списке - это выражение XPath.

Глава 10. Запись свойств узла операций

Свойства узла append



Узел присоединения проводит конкатенацию наборов записей. Это полезно для объединения наборов данных с похожими структурами, но различными данными.

Таблица 44. Свойства узла append.

Свойства узла append	Тип переменной	Описание свойства
match_by	Положению Имя	Вы можете присоединить наборы данных на основе положения полей в главном источнике полей или имени полей во входных наборах полей.
match_case	<i>логический</i>	Включает учет регистра при сравнении имен полей.
include_fields_from	Главные All	
create_tag_field	<i>логический</i>	
tag_field_name	<i>строка</i>	

Свойства узла aggregate



Узел Агрегат замещает последовательность входных записей на итоговые, агрегированные выходные записи.

Таблица 45. Свойства узла aggregate.

Свойства узла aggregate	Тип переменной	Описание свойства
ключи	<i>[поле поле ... поле]</i>	Перечисляет поля, которые можно использовать как ключи для агрегации. Например, если Пол и Область будут вашими полями ключа, то у каждой уникальной комбинации М и Ж с областями С и Ю (четыре уникальных комбинации) будет агрегированная запись.
contiguous	<i>логический</i>	Выберите эту опцию, если вам известно, что все записи с одинаковыми значениями ключа группируются совместно во входных данных (например, если входные данные отсортированы по полям ключей). Выбрав эту опцию, можно увеличить производительность.
aggregates		Структурированное свойство, перечисляющее числовые поля, значения которых будут агрегированы, а также выбранные режимы агрегации.
extension	<i>строка</i>	Задайте префикс или суффикс для дубликатов агрегированных полей (пример ниже).

Таблица 45. Свойства узла aggregate (продолжение).

Свойства узла aggregate	Тип переменной	Описание свойства
add_as	Суффикс Префикс	
inc_record_count	логический	Создает дополнительное поле, в котором указывается, сколько входных записей было агрегировано для образования каждой записи агрегата.
count_field	строка	Задаёт имя поля количества записей.

Свойства узла balance



Узел Баланс исправляет дисбаланс в наборе данных, чтобы он соответствовал заданному условию. Директива балансировки корректирует часть записей, где выполнено условие по заданному фактору.

Таблица 46. Свойства узла balance.

Свойства узла balance	Тип переменной	Описание свойства
directives		Структурированное свойство для балансировки доли значений поля на основании заданного числа (см. пример ниже).
training_data_only	логический	Задаёт, что балансировка должна выполняться только для данных обучения. Если поле раздела отсутствует в потоке, эта опция игнорируется.

Свойство узла directives использует следующий формат:

[{ строка чисел } \ { строка чисел } \ ... { строка чисел }].

Примечание: Если строки заключены в выражении (с использованием знаков двойных кавычек), перед ними должен быть символ обратной дробной черты " \ ". Символ " \ " служит также для обозначения продолжения строки, позволяя для ясности выделить аргументы.

Свойства узла derive_stb



Узел Пространственно-временные диапазоны выводит пространственно-временные диапазоны из полей широты, долготы и отметок времени. Вы также можете указать часто встречающиеся пространственно-временные диапазоны как аттракторы.

Таблица 47. Свойства узла derive_stb.

Свойства узла derive_stb	Тип переменной	Описание свойства
mode	IndividualRecords Hangouts	
latitude_field	field	
longitude_field	field	
timestamp_field	field	
hangout_density	density	Отдельная плотность. Допустимые значения плотности смотрите в описании плотности.

Таблица 47. Свойства узла *derive_stb* (продолжение).

Свойства узла <i>derive_stb</i>	Тип переменной	Описание свойства
<i>densities</i>	<i>[density,density,..., density]</i>	Каждая плотность представляет собой строку, например, STB_GH8_1DAY. Примечание: Существуют пределы допустимой плотности. Для геохеша можно использовать значения от GH1 до GH15. Для временной части можно использовать следующие значения: ВСЕГДА 1ГОД 1МЕСЯЦ 1ДЕНЬ 12ЧАСОВ 8ЧАСОВ 6ЧАСОВ 4ЧАСА 3ЧАСА 2ЧАСА 1ЧАС 30МИН 15МИН 10МИН 5МИН 2МИН 1МИН 30СЕК 15СЕК 10СЕК 5СЕК 2СЕК 1СЕК
<i>id_field</i>	<i>field</i>	
<i>qualifying_duration</i>	1ДЕНЬ 12ЧАСОВ 8ЧАСОВ 6ЧАСОВ 4ЧАСА 3ЧАСА 2ЧАСА 1ЧАС 30МИН 15МИН 10МИН 5МИН 2МИН 1МИН 30СЕК 15СЕК 10СЕК 5СЕК 2СЕК 1СЕК	Должно быть строковым.
<i>min_events</i>	<i>целое</i>	Минимальное целое значение - 2.
<i>qualifying_pct</i>	<i>целое</i>	Должно быть в диапазоне от 1 до 100.
<i>add_extension_as</i>	Prefix Suffix	
<i>name_extension</i>	<i>строка</i>	

Свойства узла distinct



Отдельный узел удаляет дублированные записи, или передавая первую отдельную запись в поток данных, или отбрасывая первую запись и вместо этого передавая в поток данных любые дубликаты.

Таблица 48. Свойства узла distinct.

Свойства узла distinct	Тип переменной	Описание свойства
mode	Включать Исключение	Вы можете включить первую отдельную запись в потоке данных или отбросить первую отдельную запись и передать вместо этого в поток любые дублированные записи.
grouping_fields	[поле поле поле]	Перечисляет поля, используемые для определения, идентичны ли записи. Примечание: Это свойство устарело, начиная с IBM SPSS Modeler 16.
composite_value	Структурированный слот	
composite_values	Структурированный слот	
inc_record_count	логическое	Создает дополнительное поле, в котором указывается, сколько входных записей было агрегировано для образования каждой записи агрегата.
count_field	строка	Задает имя поля количества записей.
sort_keys	Структурированный слот.	Примечание: Это свойство устарело, начиная с IBM SPSS Modeler 16.
default_ascending	логическое	
low_distinct_key_count	логический	Задает, что у вас только небольшое число записей и/или небольшое количество уникальных значений ключевых полей.
keys_pre_sorted	логический	Задает, что все записи с одинаковыми значениями ключа совместно группируются во входных данных.
disable_sql_generation	логическое	

Свойства узла merge



Узел слияния берет несколько входных записей и создает одну выходную запись, содержащую некоторые или все из входных полей. Он полезен для слияния данных из разных источников, например, из внутренних данных о клиентах и приобретенных демографических данных.

Таблица 49. Свойства узла merge.

Свойства узла merge	Тип переменной	Описание свойства
method	Порядок Ключи Условие	Задайте, будут ли сливаться записи в том порядке, как они перечислены в файлах данных, если одно или несколько полей ключей будет использовано для слияния записей с одинаковым значением в полях ключей, или записи будут сливаться при выполнении заданного условия.

Таблица 49. Свойства узла merge (продолжение).

Свойства узла merge	Тип переменной	Описание свойства
condition	строка	Если значение method - это Condition, задает условие для включения или отбрасывания записей.
key_fields	[поле поле поле]	
common_keys	логический	
join	Внутреннее FullOuter PartialOuter Anti	
outer_join_tag.n	логический	В этом свойстве <i>n</i> - это имя тега, как оно выведено на экран в диалоговом окне Выбрать набор данных. Обратите внимание на то, что можно задать несколько имен тегов, так как вклад может внести любое число наборов данных.
single_large_input	логический	Задает, будет ли использоваться оптимизация, чтобы одно входное множество было относительно большим в сравнении с другими вводами.
single_large_input_tag	строка	Задает имя тега, как оно показывается в диалоговом окне Выбрать большой набор данных. Обратите внимание на то, что использование этого свойства немного отличается от свойства outer_join_tag (логическое по сравнению со строкой), так как можно задать только один входной набор данных.
use_existing_sort_keys	логический	Определяет, отсортированы ли уже входные поля по одному или нескольким полям ключей.
existing_sort_keys	[{строка По возрастанию} \ {строка По убыванию}]	Задает поля, которые уже отсортированы, и направление, в котором они отсортированы.

Свойства узла rfmaggregate



Узел агрегата Новизна, частота, деньги (Recency, Frequency, Monetary - RFM) позволяет рассмотреть хронологические данные транзакций клиента, исключить любые неиспользуемые данные и объединить все оставшиеся данные транзакций в одну строку, где будет представлено, когда в последний раз обращался клиент, сколько транзакций он произвел и какова общая денежная сумма этих транзакций.

Таблица 50. Свойства узла rfmaggregate.

Свойства узла rfmaggregate	Тип переменной	Описание свойства
relative_to	Фиксированная Сегодня	Задать дату, по сравнению с которой будет вычисляться новизна транзакций.
reference_date	дата	Доступно только в том случае, если для relative_to выбрано Fixed.
contiguous	логический	Если ваши данные предварительно отсортированы, так что все записи с одинаковым ID сгруппированы совместно в потоке данных, выбор этой опции ускоряет обработку.

Таблица 50. Свойства узла rfmaggregate (продолжение).

Свойства узла rfmaggregate	Тип переменной	Описание свойства
id_field	поле	Задать поле, используемое для идентификации клиентов и их транзакций.
date_field	поле	Задать поле даты, которое будет использоваться для вычисления новизны.
value_field	поле	Задать поле, которое будет использоваться для вычисления денежного значения.
extension	строка	Задать префикс или суффикс для дублированных агрегированных полей.
add_as	Суффикс Префикс	Указать использование для расширение - как суффикс или как префикс.
discard_low_value_records	логический	Включить использование параметра discard_records_below.
discard_records_below	число	Задать минимальное значение, ниже которого любые подробности транзакций не используются при вычислении суммарных значений RFM. Единицы измерения относятся к выбранному полю значение.
only_recent_transactions	логический	Включить использование одного из параметров, specify_transaction_date или transaction_within_last.
specify_transaction_date	логический	
transaction_date_after	дата	Доступно только в том случае, если выбрано specify_transaction_date. Определить дату транзакции, после которой записи будут включены в ваш анализ.
transaction_within_last	число	Доступно только в том случае, если выбрано transaction_within_last. Задать количество и тип периодов времени (дни, недели, месяцы и годы) для отсчета назад от даты для вычисления недавних значений и включения этих записей в ваш анализ.
transaction_scale	Дни Недели Месяцы Годы	Доступно только в том случае, если выбрано transaction_within_last. Задать количество и тип периодов времени (дни, недели, месяцы и годы) для отсчета назад от даты для вычисления недавних значений и включения этих записей в ваш анализ.
save_r2	логический	Задает дату предпоследней из недавних транзакций для каждого клиента.
save_r3	логический	Доступно только в том случае, если выбрано save_r2. Задает дату третьей по счету из самых недавних транзакций для каждого клиента.

Свойства узла Rprocess



Узел обработки R дает возможность брать данные из потока IBM(r) SPSS(r) Modeler и модифицировать их при помощи пользовательского сценария R. После модификации данные возвращаются в поток.

Таблица 51. Свойства узла Rprocess.

Свойства узла Rprocess	Тип переменной	Описание свойства
синтаксис	строка	
convert_flags	StringsAndDoubles LogicalValues	
convert_datetime	логическое	
convert_datetime_class	POSIXct POSIXlt	
convert_missing	логическое	

Свойства узла sample



Узел Выборка отбирает подмножество записей. Поддерживается несколько типов выборки, в том числе стратифицированные, кластеризованные и неслучайные (структурированные) выборки. Выборки могут быть полезны для повышения производительности и для выбора групп связанных записей или транзакций для анализа.

Таблица 52. Свойства узла sample.

Свойства узла sample	Тип переменной	Описание свойства
method	Простой Сложный	
mode	Включать Исключение	Включить или отбросить записи, для которых выполнено заданное условие.
sample_type	Первое OneInN RandomPct	Задаёт способ выборки.
first_n	целое	Записи вплоть до заданной точки отсечения будут включены или отброшены.
one_in_n	число	Включить или отбросить каждую n-ную запись.
rand_pct	число	Задать процентную долю записей для включения или отбрасывания.
use_max_size	логический	Включить использование параметра maximum_size.
maximum_size	целое	Задать самую большую выборку, которая будет включена в поток данных или отброшена из него. Эта опция избыточна и поэтому будет отключена, когда заданы First и Include.
set_random_seed	логический	Включение использования параметра начального значения генератора псевдослучайных чисел.
random_seed	целое	Задать значение, используемое как начальное значение генератора псевдослучайных чисел.
complex_sample_type	Переменный Систематическая	
sample_units	Доли Количества	
sample_size_proportions	Фиксированная Пользовательские Переменная	

Таблица 52. Свойства узла *sample* (продолжение).

Свойства узла <i>sample</i>	Тип переменной	Описание свойства
<code>sample_size_counts</code>	Фиксированная Пользовательские Переменная	
<code>fixed_proportions</code>	<i>число</i>	
<code>fixed_counts</code>	<i>целое</i>	
<code>variable_proportions</code>	<i>поле</i>	
<code>variable_counts</code>	<i>поле</i>	
<code>use_min_stratum_size</code>	<i>логический</i>	
<code>minimum_stratum_size</code>	<i>целое</i>	Эта опция применяется только в том случае, когда берется сложная выборка с опцией <code>Sample units=Proportions</code> .
<code>use_max_stratum_size</code>	<i>логический</i>	
<code>maximum_stratum_size</code>	<i>целое</i>	Эта опция применяется только в том случае, когда берется сложная выборка с опцией <code>Sample units=Proportions</code> .
кластеры	<i>поле</i>	
<code>stratify_by</code>	<i>[поле1 ... полеN]</i>	
<code>specify_input_weight</code>	<i>логический</i>	
<code>input_weight</code>	<i>поле</i>	
<code>new_output_weight</code>	<i>строка</i>	
<code>sizes_proportions</code>	<i>[{строка строковое значение}{строка строковое значение}...]</i>	Если задано <code>sample_units=proportions</code> и <code>sample_size_proportions=Custom</code> , указывает значение для каждой возможной комбинации значений полей стратификации.
<code>default_proportion</code>	<i>число</i>	
<code>sizes_counts</code>	<i>[{строка строковое значение}{строка строковое значение}...]</i>	Указывает значение для каждой возможной комбинации значений полей стратификации. Использование аналогично <code>sizes_proportions</code> , но указывается целое число, а не доля.
<code>default_count</code>	<i>число</i>	

Свойства узла *select*



Узел Выбор отбирает или отбрасывает подмножество записей из потока данных на основе конкретного условия. Например, вы можете выбрать записи, принадлежащие определенному району продаж.

Таблица 53. Свойства узла *select*.

Свойства узла <i>select</i>	Тип переменной	Описание свойства
<code>mode</code>	Включать Исключение	Задает, включать или отбрасывать выбранные записи.
<code>condition</code>	<i>строка</i>	Условие для включения или отбрасывания записей.

Свойства узла sort



Узел Сортировка сортирует записи в восходящем или убывающем порядке на основании значений одного или нескольких полей.

Таблица 54. Свойства узла sort.

Свойства узла sort	Тип переменной	Описание свойства
ключи	[{строка По возрастанию} \ {строка По убыванию}]	Задаёт поля, которые вам нужно отсортировать. Если направление сортировки не задано, используются значения по умолчанию.
default_ascending	логический	Задаёт порядок сортировки по умолчанию.
use_existing_keys	логический	Задаёт, оптимизирована ли сортировка посредством использования предыдущего порядка сортировки для полей, которые уже сортировались.
existing_keys		Задаёт поля, которые уже сортировались, и направление, в котором они сортировались. Использует тот же формат, что и свойство ключи.

Свойства узла streamingts



Узел Поточковые временные ряды строит и оценивает модели временных рядов за один шаг, без необходимости использования узла Временные интервалы.

Таблица 55. Свойства узла streamingts.

Свойства узла streamingts	Тип переменной	Описание свойства
custom_fields	логическое	Если custom_fields=false, используются параметры с вышележащего узла типа. Если custom_fields=true, надо задать поля назначения и входные поля.
targets	[поле_1...поле_N]	
inputs	[поле_1...поле_N]	
method	ExpertModeler Exsmooth Arima	
calculate_conf	логическое	
conf_limit_pct	действительное число	
use_time_intervals_node	логическое	Если use_time_intervals_node=true, используются параметры с вышележащего узла временных интервалов. В противном случае необходимо задать interval_offset_position, interval_offset и interval_type.
interval_offset_position	LastObservation LastRecord	LastObservation - это Последнее допустимое наблюдение . LastRecord - это Обратный отсчет с последней записи .
interval_offset	число	

Таблица 55. Свойства узла *streamingts* (продолжение).

Свойства узла <i>streamingts</i>	Тип переменной	Описание свойства
<i>interval_type</i>	Periods Years Quarters Months WeeksNonPeriodic DaysNonPeriodic HoursNonPeriodic MinutesNonPeriodic SecondsNonPeriodic	
<i>events</i>	<i>fields</i>	
<i>expert_modeler_method</i>	AllModels Exsmooth Arima	
<i>consider_seasonal</i>	<i>логическое</i>	
<i>detect_outliers</i>	<i>логическое</i>	
<i>expert_outlier_additive</i>	<i>логическое</i>	
<i>expert_outlier_level_shift</i>	<i>логическое</i>	
<i>expert_outlier_innovational</i>	<i>логическое</i>	
<i>expert_outlier_transient</i>	<i>логическое</i>	
<i>expert_outlier_seasonal_additive</i>	<i>логическое</i>	
<i>expert_outlier_local_trend</i>	<i>логическое</i>	
<i>expert_outlier_additive_patch</i>	<i>логическое</i>	
<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>целое</i>	То же самое свойство, что для узла моделирования временных рядов
<i>arima_d</i>	<i>целое</i>	То же самое свойство, что для узла моделирования временных рядов
<i>arima_q</i>	<i>целое</i>	То же самое свойство, что для узла моделирования временных рядов
<i>arima_sp</i>	<i>целое</i>	То же самое свойство, что для узла моделирования временных рядов
<i>arima_sd</i>	<i>целое</i>	То же самое свойство, что для узла моделирования временных рядов
<i>arima_sq</i>	<i>целое</i>	То же самое свойство, что для узла моделирования временных рядов
<i>arima_transformation_type</i>	None SquareRoot NaturalLog	То же самое свойство, что для узла моделирования временных рядов
<i>arima_include_constant</i>	<i>логическое</i>	То же самое свойство, что для узла моделирования временных рядов

Таблица 55. Свойства узла *streamingts* (продолжение).

Свойства узла <i>streamingts</i>	Тип переменной	Описание свойства
<i>tf_arma_p.fieldname</i>	<i>целое</i>	То же самое свойство, что для узла моделирования временных рядов. Для передаточных функций.
<i>tf_arma_d.fieldname</i>	<i>целое</i>	То же самое свойство, что для узла моделирования временных рядов. Для передаточных функций.
<i>tf_arma_q.fieldname</i>	<i>целое</i>	То же самое свойство, что для узла моделирования временных рядов. Для передаточных функций.
<i>tf_arma_sp.fieldname</i>	<i>целое</i>	То же самое свойство, что для узла моделирования временных рядов. Для передаточных функций.
<i>tf_arma_sd.fieldname</i>	<i>целое</i>	То же самое свойство, что для узла моделирования временных рядов. Для передаточных функций.
<i>tf_arma_sq.fieldname</i>	<i>целое</i>	То же самое свойство, что для узла моделирования временных рядов. Для передаточных функций.
<i>tf_arma_delay.fieldname</i>	<i>целое</i>	То же самое свойство, что для узла моделирования временных рядов. Для передаточных функций.
<i>tf_arma_transformation_type.имя_поля</i>	None SquareRoot NaturalLog	
<i>arma_detect_outlier_mode</i>	None Automatic	
<i>arma_outlier_additive</i>	<i>логическое</i>	
<i>arma_outlier_level_shift</i>	<i>логическое</i>	
<i>arma_outlier_innovational</i>	<i>логическое</i>	
<i>arma_outlier_transient</i>	<i>логическое</i>	
<i>arma_outlier_seasonal_additive</i>	<i>логическое</i>	
<i>arma_outlier_local_trend</i>	<i>логическое</i>	
<i>arma_outlier_additive_patch</i>	<i>логическое</i>	
<i>deployment_force_rebuild</i>	<i>логическое</i>	
<i>deployment_rebuild_mode</i>	Count Percent	
<i>deployment_rebuild_count</i>	<i>число</i>	
<i>deployment_rebuild_pct</i>	<i>число</i>	
<i>deployment_rebuild_field</i>	<i><field></i>	

Глава 11. Поле Свойства узла операций

Свойства узла anonymize



Узел анонимизации преобразует способ представления имен и значений полей уровнем ниже, маскируя таким образом исходные данные. Это может быть полезно, если вы хотите разрешить другим пользователям построить модели, используя чувствительные данные, такие как имена клиентов или другие подробности.

Таблица 56. Свойства узла anonymize.

Свойства узла anonymize	Тип переменной	Описание свойства
enable_anonymize	<i>логический</i>	Если задано Т, активирует анонимизацию значений полей (эквивалентно выбору варианта Да для этого поля в столбце Анонимизировать значения).
use_prefix	<i>логический</i>	Если задано Т, будет использоваться пользовательский префикс, если он задан. Применяется к полям, которые будут анонимизированы хеш-методом, что эквивалентно включению радиокнопки Настроить в диалоговом окне Заменить значения для этого поля.
prefix	<i>строка</i>	Эквивалентно вводу префикса в текстовом поле диалогового окна Заменить значения. Если ничего дополнительного не задано, префикс по умолчанию - это значение по умолчанию.
преобразование	Переменный Фиксированная	Определяет, какими будут параметры преобразования для анонимизируемого методом Преобразование поля - случайными или фиксированными.
set_random_seed	<i>логический</i>	Если задано Т, будет использоваться указанное начальное значение генератора псевдослучайных чисел (если для преобразования также задано значение Случайное).
random_seed	<i>целое</i>	Если для set_random_seed задано значение Т, это начальное значение генератора псевдослучайных чисел.
scale	<i>число</i>	Если преобразование задано как Фиксированное, это значение используется для функции "умножить на". Максимальное значение масштабирования обычно равно 10, но его можно уменьшить для предотвращения переполнения.
translate	<i>число</i>	Если преобразование задано как Фиксированное, это значение используется для функции "увеличить на". Максимальное значение увеличения обычно равно 1000, но его можно уменьшить для предотвращения переполнения.

Свойства узла autodataprep



Узел автоматической подготовки данных (Automated Data Preparation, ADP) может анализировать ваши данные и находит исправления, выявляет проблемные и малополезные поля, создает при необходимости производные атрибуты и повышает производительность, применяя интеллектуальные способы анализа и выборки. Этот узел можно использовать в полностью автоматическом режиме, позволив ему выбирать и применять исправления или предварительно просматривать изменения перед тем, как они сделаны и приняты, а при желании применять, отклонять или исправлять их.

Таблица 57. Свойства узла autodataprep.

Свойства узла autodataprep	Тип переменной	Описание свойства
objective	Balanced Скорость Точность Пользовательские	
custom_fields	<i>логический</i>	Если значение флага true, это позволяет вам задать поле назначения, входные и другие поля для текущего узла. При значении false используются текущие параметры с вышележащего узла Тип.
target	<i>поле</i>	Задаёт одно поле назначения.
inputs	[поле1 ... полеN]	Входные (или предикторные) поля, используемые в модели.
use_frequency	<i>логический</i>	
frequency_field	<i>поле</i>	
use_weight	<i>логический</i>	
weight_field	<i>поле</i>	
excluded_fields	Фильтр Нет	
if_fields_do_not_match	StopExecution ClearAnalysis	
prepare_dates_and_times	<i>логический</i>	Управлять доступом ко всем полям даты и времени
compute_time_until_date	<i>логический</i>	
reference_date	Сегодня Фиксированная	
fixed_date	<i>дата</i>	
units_for_date_durations	Автоматически Фиксированная	
fixed_date_units	Годы Месяцы Дни	
compute_time_until_time	<i>логический</i>	
reference_time	CurrentTime Фиксированная	
fixed_time	<i>время</i>	
units_for_time_durations	Автоматически Фиксированная	
fixed_date_units	Часы Мин. Секунды	
extract_year_from_date	<i>логический</i>	
extract_month_from_date	<i>логический</i>	
extract_day_from_date	<i>логический</i>	
extract_hour_from_time	<i>логический</i>	
extract_minute_from_time	<i>логический</i>	
extract_second_from_time	<i>логический</i>	

Таблица 57. Свойства узла autodataprep (продолжение).

Свойства узла autodataprep	Тип переменной	Описание свойства
exclude_low_quality_inputs	логический	
exclude_too_many_missing	логический	
maximum_percentage_missing	число	
exclude_too_many_categories	логический	
maximum_number_categories	число	
exclude_if_large_category	логический	
maximum_percentage_category	число	
prepare_inputs_and_target	логический	
adjust_type_inputs	логический	
adjust_type_target	логический	
reorder_nominal_inputs	логический	
reorder_nominal_target	логический	
replace_outliers_inputs	логический	
replace_outliers_target	логический	
replace_missing_continuous_inputs	логический	
replace_missing_continuous_target	логический	
replace_missing_nominal_inputs	логический	
replace_missing_nominal_target	логический	
replace_missing_ordinal_inputs	логический	
replace_missing_ordinal_target	логический	
maximum_values_for_ordinal	число	
minimum_values_for_continuous	число	
outlier_cutoff_value	число	
outlier_method	Replace Удалить	
rescale_continuous_inputs	логический	
rescaling_method	MinMax ZScore	
min_max_minimum	число	
min_max_maximum	число	
z_score_final_mean	число	
z_score_final_sd	число	
rescale_continuous_target	логический	
target_final_mean	число	
target_final_sd	число	
transform_select_input_fields	логический	
maximize_association_with_target	логический	
p_value_for_merging	число	
merge_ordinal_features	логический	
merge_nominal_features	логический	

Таблица 57. Свойства узла autodataprep (продолжение).

Свойства узла autodataprep	Тип переменной	Описание свойства
minimum_cases_in_category	число	
bin_continuous_fields	логический	
p_value_for_binning	число	
perform_feature_selection	логический	
p_value_for_selection	число	
perform_feature_construction	логический	
transformed_target_name_extension	строка	
transformed_inputs_name_extension	строка	
constructed_features_root_name	строка	
years_duration_name_extension	строка	
months_duration_name_extension	строка	
days_duration_name_extension	строка	
hours_duration_name_extension	строка	
minutes_duration_name_extension	строка	
seconds_duration_name_extension	строка	
year_cyclical_name_extension	строка	
month_cyclical_name_extension	строка	
day_cyclical_name_extension	строка	
hour_cyclical_name_extension	строка	
minute_cyclical_name_extension	строка	
second_cyclical_name_extension	строка	

Свойства узла binning



Узел разделения на интервалы автоматически создает новые номинальные поля на основе значений одного или нескольких существующих количественных полей (числового диапазона). Например, можно преобразовать количественное входное поле в новое категориальное поле, содержащее группы входных данных, как отклонения от среднего. После создания интервалов для нового поля вы можете сгенерировать узел извлечения на основе точек деления.

Таблица 58. Свойства узла binning.

Свойства узла binning	Тип переменной	Описание свойства
fields	[поле1 поле2 ... полен]	Отложенное преобразование количественных полей (числовой диапазон). Можно разделить на интервалы несколько полей одновременно.
method	FixedWidth EqualCount Ранг SDev Оптимальная	Способ, используемый для определения точек разделения для новых интервалов поля (категорий).

Таблица 58. Свойства узла binning (продолжение).

Свойства узла binning	Тип переменной	Описание свойства
rcalculate_bins	Всегда IfNecessary	Задаёт, вычисляются ли повторно интервалы для размещения данных в соответствующем интервале при каждом выполнении узла, или же данные добавляются только в существующие и любые новые добавленные интервалы.
fixed_width_name_extension	строка	Расширение по умолчанию - это <i>_BIN</i> .
fixed_width_add_as	Суффикс Префикс	Задаёт, куда добавляется расширение имени поля, в конец имени (суффикс) или в начало (префикс). Расширение по умолчанию - это <i>income_BIN</i> .
fixed_bin_method	Ширина Count	
fixed_bin_count	целое	Задаёт целое число, используемое для определения количества интервалов фиксированной ширины (категорий) для нового поля или полей.
fixed_bin_width	real	Значение (целое или действительное) для вычисления ширины интервала.
equal_count_name_extension	строка	Расширение по умолчанию - это <i>_TILE</i> .
equal_count_add_as	Суффикс Префикс	Задаёт расширение, суффикс или префикс, используемое для имени поля, сгенерированного при помощи стандартных процентилей. Расширение по умолчанию - это <i>_TILE</i> плюс <i>N</i> , где <i>N</i> - это порядок процентиля.
tile4	логический	Задаёт четыре интервала квантили, каждая из которых содержит 25% наблюдений.
tile5	логический	Генерирует пять интервалов квинтили.
tile10	логический	Генерирует десять интервалов децили.
tile20	логический	Генерирует 20 интервалов вингтили.
tile100	логический	Генерирует 100 интервалов процентиля.
use_custom_tile	логический	
custom_tile_name_extension	строка	Расширение по умолчанию - это <i>_TILEN</i> .
custom_tile_add_as	Суффикс Префикс	
custom_tile	целое	
equal_count_method	RecordCount ValueSum	Способ RecordCount направлен на назначение одинакового числа записей в каждый интервал, в то время как ValueSum назначает записи так, чтобы сумма значений в каждом интервале была одинакова.
tied_values_method	Следующее Текущий Переменный	Задаёт, в какой интервал будут помещены связанные со значением данные.

Таблица 58. Свойства узла binning (продолжение).

Свойства узла binning	Тип переменной	Описание свойства
rank_order	По возрастанию По убыванию	Это свойство включает в себя опции По возрастанию (минимальное значение помечается как первое) или По убыванию (максимальное значение помечается как первое).
rank_add_as	Суффикс Префикс	Эта опция применяется к рангу, дробному рангу и процентному рангу.
rank	логический	
rank_name_extension	строка	Расширение по умолчанию - это <i>_RANK</i> .
rank_fractional	логический	Ранжирует наблюдения, причем значение в новом поле равно рангу, деленному на сумму весов непропущенных наблюдений. Дробные ранги лежат в диапазоне 0–1.
rank_fractional_name_extension	строка	Расширение по умолчанию - это <i>_F_RANK</i> .
rank_pct	логический	Каждый ранг делится на число записей с непропущенными значениями и умножается на 100. Процентные дробные ранги лежат в диапазоне 1–100.
rank_pct_name_extension	строка	Расширение по умолчанию - это <i>_P_RANK</i> .
sdev_name_extension	строка	
sdev_add_as	Суффикс Префикс	
sdev_count	Один Два Три	
optimal_name_extension	строка	Расширение по умолчанию - это <i>_OPTIMAL</i> .
optimal_add_as	Суффикс Префикс	
optimal_supervisor_field	поле	Поле, выбранное как контрольное, с которым связаны поля, выбранные для разделения на интервалы.
optimal_merge_bins	логический	Задаёт, что любые интервалы с малым количеством наблюдений будут добавлены к большему соседнему интервалу.
optimal_small_bin_threshold	целое	
optimal_pre_bin	логический	Обозначает, что должно выполняться предварительное разделение набора данных на интервалы.
optimal_max_bins	целое	Задаёт верхний предел, чтобы исключить создания неограниченно большого числа интервалов.
optimal_lower_end_point	Inclusive Exclusive	
optimal_first_bin	Неограниченный Bounded	
optimal_last_bin	Неограниченный Bounded	

Свойства узла derive



Узел извлечения изменяет значения данных или создает новые поля из одного или нескольких существующих полей. Он создает поля формулы типа, флага, номинала, состояния, количества и условного выражения.

Таблица 59. Свойства узла derive.

Свойства узла derive	Тип переменной	Описание свойства
new_name	строка	Имя нового поля.
mode	Одиночный Несколько	Создает одно или несколько полей.
fields	[поле поле поле]	Используется только в режиме Несколько для выбора несколько полей.
name_extension	строка	Задаёт расширение для имен новых полей.
add_as	Суффикс Префикс	Добавляет расширение как префикс (в начале) или как суффикс (в конце) имени поля.
result_type	Формула Флаг Установить Состояние Частота Условное	Шесть типов новых полей, которые можно создать.
formula_expr	строка	Выражение для вычисления значения в новом поле на узле Извлечение.
flag_expr	строка	
flag_true	строка	
flag_false	строка	
set_default	строка	
set_value_cond	строка	Структурировано для предоставления условия, связанного с данным значением.
state_on_val	строка	Задаёт значение для нового поля, когда выполнено условие On.
state_off_val	строка	Задаёт значение для нового поля, когда выполнено условие Off.
state_on_expression	строка	
state_off_expression	строка	
state_initial	Вкл Выкл	Назначает каждой записи нового поля начальное значение On или Off. Это значение можно изменить при выполнении любого условия.
count_initial_val	строка	
count_inc_condition	строка	
count_inc_expression	строка	
count_reset_condition	строка	
cond_if_cond	строка	
cond_then_expr	строка	
cond_else_expr	строка	

Свойства узла ensemble



Узел Ансамбль объединяет два или более слепков моделей для получения более точных предсказаний, чем можно получить от любой модели.

Таблица 60. Свойства узла ensemble.

Свойства узла ensemble	Тип переменной	Описание свойства
ensemble_target_field	поле	Задаёт поле назначения для всех моделей, используемых в ансамбле.
filter_individual_model_output	логический	Задаёт, нужно ли отключать результаты скоринга из индивидуальных моделей.
flag_ensemble_method	Голосование ConfidenceWeightedVoting RawPropensityWeightedVoting AdjustedPropensityWeightedVoting HighestConfidence AverageRawPropensity AverageAdjustedPropensity	Задаёт способ, используемый для определения оценки ансамбля. Этот параметр применяется только в том случае, если выбранное поле назначения - это флаговое поле.
set_ensemble_method	Голосование ConfidenceWeightedVoting HighestConfidence	Задаёт способ, используемый для определения оценки ансамбля. Этот параметр применим только в том случае, если выбранное поле назначения номинальное.
flag_voting_tie_selection	Переменный HighestConfidence RawPropensity AdjustedPropensity	Если выбран способ голосования, задаёт, как разрешаются связи. Этот параметр применим только в том случае, если выбранное поле назначения флаговое.
set_voting_tie_selection	Переменный HighestConfidence	Если выбран способ голосования, задаёт, как разрешаются связи. Этот параметр применим только в том случае, если выбранное поле назначения номинальное.
calculate_standard_error	логический	Если поле назначения количественное, по умолчанию запускается вычисление среднеквадратичной ошибки для определения различий между измеренными или оцененными значениями и действительными значениями, а также для демонстрации, насколько эти оценки совпали.

Свойства узла filler



Узел заполнителя замещает значения полей и заменяет систему хранения. Вы можете заменить значения на основе условия CLEM, такого как @BLANK(@FIELD). Можно также выбрать замену всех пробелов или всех пробельных значений на заданное значение. Узел заполнителя часто используется вместе с узлом Тип для замены пропущенных значений.

Таблица 61. Свойства узла filler.

Свойства узла filler	Тип переменной	Описание свойства
fields	[поле поле поле]	Поля из набора данных, значения которого будут проверены и заменены.
replace_mode	Всегда Условное Пробел Ноль BlankAndNull	Вы можете заменить все значения, пустые значения или значения null, а также выполнить замену на основе заданного условия.
condition	строка	
replace_with	строка	

Свойства узла filter



Узел Фильтр фильтрует (отбрасывает) поля, переименовывает поля и отображает поля с одного узла источника на другой.

Использование свойства default_include. Обратите внимание на то, что задание значения для свойства default_include автоматически не включает и не исключает все поля; оно просто определяет значения по умолчанию для текущего выбора. Это функциональный эквивалент нажатия кнопки **Включить поля по умолчанию** в диалоговом окне узла Фильтр.

Таблица 62. Свойства узла filter.

Свойства узла filter	Тип переменной	Описание свойства
default_include	логический	Ключевое свойство для указания, каким будет поведение по умолчанию - передать или отфильтровать поля. Обратите внимание на то, что задание значения для этого свойства автоматически не включает и не исключает все поля; оно просто определяет, включаются или исключаются выбранные поля по умолчанию.
include	логический	Ключевое свойство для включения или удаления поля.
new_name	строка	

Свойства узла history



Узел Хронология создает новые поля, содержащие данные из полей в предыдущих записях. Хронологические узлы чаще всего используются для последовательных данных, таких как данные временных рядов. Перед использованием узла Хронология может потребоваться отсортировать данные с использованием узла Сортировка.

Таблица 63. Свойства узла history.

Свойства узла history	Тип переменной	Описание свойства
fields	[поле поле поле]	Поля, для которых требуется хронология.

Таблица 63. Свойства узла history (продолжение).

Свойства узла history	Тип переменной	Описание свойства
смещение	число	Задаёт последнюю запись (предшествующую текущей записи), из которой нужно извлекать значения хронологического поля.
span	число	Задаёт количество предшествующих записей, из которых нужно извлекать значения.
unavailable	Исключение Leave Заполнить	Обсуждая обработку записей без хронологических значений, обычно имеют в виду первые несколько записей (наверху набора данных), у которых ещё нет предыдущих записей, которые можно было бы использовать как хронологические.
fill_with	Строка Число	Задаёт значение или строку, которые будут использоваться для записей без доступных хронологических значений.

Свойства узла partition



Узел Разделы генерирует поле раздела, которое разбивает данные на отдельные подмножества для стадий обучения, испытания и проверки при построении моделей.

Таблица 64. Свойства узла partition.

Свойства узла partition	Тип переменной	Описание свойства
new_name	строка	Имя поля раздела, сгенерированного узлом.
create_validation	логический	Задаёт, должен ли создаваться раздел проверки.
training_size	целое	Процентная доля записей (0–100) для выделения разделу обучения.
testing_size	целое	Процентная доля записей (0–100) для выделения разделу испытания.
validation_size	целое	Процентная доля записей (0–100) для выделения разделу проверки. Игнорируется, если раздел проверки не создан.
training_label	строка	Метка для раздела обучения.
testing_label	строка	Метка для раздела испытания.
validation_label	строка	Метка для раздела проверки. Игнорируется, если раздел проверки не создан.
value_mode	Системная SystemAndLabel Метка	Задаёт значения, используемые для представления каждого раздела в данных. Например, обучающая выборка может быть представлена системным целым числом 1, меткой Обучение или комбинацией обоих значений 1_Обучение.
set_random_seed	логический	Задаёт, нужно ли использовать определенное пользователем начальное значение генератора псевдослучайных чисел.

Таблица 64. Свойства узла *partition* (продолжение).

Свойства узла <i>partition</i>	Тип переменной	Описание свойства
<code>random_seed</code>	<i>целое</i>	Заданное пользователем случайное начальное значение генератора псевдослучайных чисел. Чтобы использовать это значение, для <code>set_random_seed</code> должно быть задано <code>True</code> .
<code>enable_sql_generation</code>	<i>логический</i>	Задаёт, использовать ли SQL pushback для назначения записей разделам.
<code>unique_field</code>		Задаёт входное поле, используемое для обеспечения назначения записей разделам случайным, но повторяемым образом. Чтобы использовать это значение, для <code>enable_sql_generation</code> должно быть задано значение <code>True</code> .

Свойства узла *reclassify*



Узел переклассификации преобразует один набор категориальных значений в другой. Переклассификация полезна для сворачивания категорий или для перегруппировки данных для анализа.

Таблица 65. Свойства узла *reclassify*.

Свойства узла <i>reclassify</i>	Тип переменной	Описание свойства
<code>mode</code>	Одиночный Несколько	Режим Единичный переклассифицирует категории для одного поля. Несколько активирует опции, включающие одновременное преобразование нескольких полей.
<code>replace_field</code>	<i>логический</i>	
<code>поле</code>	<i>строка</i>	Используется только в режиме Единичный.
<code>new_name</code>	<i>строка</i>	Используется только в режиме Единичный.
<code>fields</code>	<i>[поле1 поле2 ... полен]</i>	Используется только в режиме Несколько.
<code>name_extension</code>	<i>строка</i>	Используется только в режиме Несколько.
<code>add_as</code>	Суффикс Префикс	Используется только в режиме Несколько.
<code>reclassify</code>	<i>строка</i>	Структурированное свойство для значений полей.
<code>use_default</code>	<i>логический</i>	Использовать значение по умолчанию.
<code>по умолчанию</code>	<i>строка</i>	Задать значение по умолчанию.
<code>pick_list</code>	<i>[строка строка ... строка]</i>	Позволяет пользователю импортировать список известных новых значений для заполнения раскрывающегося списка в таблице.

Свойства узла reorder



Узел переупорядочения полей определяет естественный порядок, используемый для вывода полей нижележащего уровня. Этот порядок влияет на показ полей во многих положениях, таких как таблицы, списки и средство выбора полей. Эта операция полезна при работе с обширными наборами данных, чтобы сделать нужные поля более наглядными.

Таблица 66. Свойства узла reorder.

Свойства узла reorder	Тип переменной	Описание свойства
mode	Пользовательские Авто	Значения можно сортировать автоматически или задавать пользовательский порядок.
sort_by	Имя Тип Хранение	
ascending	логический	
start_fields	[поле1 поле2 ... fieldn]	Новые поля вставляются после этих полей.
end_fields	[поле1 поле2 ... fieldn]	Новые поля вставляются перед этими полями.

Свойства узла restructure



Узел реструктуризации преобразует номинальное или флаговое поле в группу полей, которые можно заполнить значениями еще одного поля. Например, если задано поле с именем *тип платежа*, у которого могут быть значения *кредит*, *наличные* и *дебет*, могут быть заданы три новые поля (*кредит*, *наличные*, *дебет*), каждое из которых может содержать значение фактического выполненного платежа.

Таблица 67. Свойства узла restructure.

Свойства узла restructure	Тип переменной	Описание свойства
fields_from	[категория категория категория] все	
include_field_name	логический	Указывает, использовать ли имя поля в реструктурированном имени поля.
value_mode	OtherFields Флаги	Указывает на режим для того, чтобы определить значения для реструктурированных полей. Используя OtherFields, нужно указать, какие поля использовать (см. ниже). При использовании Flags значения - это числовые флаги.
value_fields	[поле поле поле]	Требуется, если для value_mode задано OtherFields. Задаст, какие поля использовать как поля значений.

Свойства узла rfmanalysis



Узел анализа Новизна, частота, деньги (Recency, Frequency, Monetary - RFM) позволяет вам количественно определить, какие клиенты вероятно будут лучшими, исследуя, насколько недавно они сделали свои последние покупки (новизна), как часто они покупали (частота) и сколько денег потратили на все транзакции (деньги).

Таблица 68. Свойства узла *rfanalysis*.

Свойства узла <i>rfanalysis</i>	Тип переменной	Описание свойства
<i>recency</i>	<i>поле</i>	Задать поле новизны. Это может быть дата, отметка времени или просто число.
<i>frequency</i>	<i>поле</i>	Задать поле частоты.
<i>monetary</i>	<i>поле</i>	Задать поле денег.
<i>recency_bins</i>	<i>целое</i>	Задать количество интервалов новизны, которые будут сгенерированы.
<i>recency_weight</i>	<i>число</i>	Задать веса для применения к недавним данным. Значение по умолчанию - 100.
<i>frequency_bins</i>	<i>целое</i>	Задать количество интервалов частоты, которые будут сгенерированы.
<i>frequency_weight</i>	<i>число</i>	Задать веса для применения к данным частоты. Значение по умолчанию - 10.
<i>monetary_bins</i>	<i>целое</i>	Задать количество интервалов денег, которые будут сгенерированы.
<i>monetary_weight</i>	<i>число</i>	Задать веса для применения к денежным данным. Значение по умолчанию - 1.
<i>tied_values_method</i>	Следующее Текущий	Задаёт, в какой интервал будут помещены связанные со значением данные.
<i>recalculate_bins</i>	Всегда IfNecessary	
<i>add_outliers</i>	<i>логический</i>	Доступно только в том случае, если для <i>recalculate_bins</i> задано значение <i>IfNecessary</i> . Если задана эта опция, записи, лежащие ниже самого нижнего интервала, будут добавлены к нему, а записи, лежащие выше самого верхнего интервала, будут добавлены к этому интервалу.
<i>binned_field</i>	Недавность Частота Monetary	
<i>recency_thresholds</i>	<i>значение значение</i>	Доступно только в том случае, если для <i>recalculate_bins</i> задано значение <i>Always</i> . Задать верхний и нижний порог для интервалов новизны. Верхний порог одного интервала используется как нижний порог следующего, например, [10 30 60] будет определять два интервала, первый интервал с верхним и нижним порогами 10 и 30, а второй интервал с порогами 30 и 60.
<i>frequency_thresholds</i>	<i>значение значение</i>	Доступно только в том случае, если для <i>recalculate_bins</i> задано значение <i>Always</i> .
<i>monetary_thresholds</i>	<i>значение значение</i>	Доступно только в том случае, если для <i>recalculate_bins</i> задано значение <i>Always</i> .

Свойства узла settoflag



Узел Задать как флаг извлекает несколько полей флагов на основании категориальных значений, определенных для одного или нескольких номинальных полей.

Таблица 69. Свойства узла settoflag.

Свойства узла settoflag	Тип переменной	Описание свойства
fields_from	[категория категория категория] все	
true_value	строка	Задаёт значение true, используемое узлом при установке флага. Значение по умолчанию - Т.
false_value	строка	Задаёт значение false, используемое узлом при установке флага. Значение по умолчанию - F.
use_extension	логический	Использовать расширение как префикс или суффикс для нового поля флага.
extension	строка	
add_as	Суффикс Префикс	Задаёт, как добавляется расширение, в виде префикса или суффикса.
aggregate	логический	Группирует совместно записи на основании ключевых полей. Если для какой-то записи задается значение true, включаются все флаговые поля в группе.
ключи	[поле поле поле]	Ключевые поля.

Свойства узла statistictransform



Узел Преобразование статистики запускает разнообразные команды синтаксиса IBM SPSS Statistics для источников данных в IBM SPSS Modeler. Этому узлу требуется лицензированная копия IBM SPSS Statistics.

Свойства этого узла описаны в разделе “Свойства узла statistictransform” на стр. 237.

Свойства узла timeintervals



Узел Интервалы времени задает интервалы и создает метки (при необходимости) для моделирования временных рядов. Если значения явно не разделены пробелами, этот узел может заполнить или агрегировать значения, как требуется для генерирования однородных интервалов между записями.

Таблица 70. Свойства узла *timeintervals*.

Свойства узла <i>timeintervals</i>	Тип переменной	Описание свойства
<i>interval_type</i>	Нет Периоды CyclicPeriods Годы Кварталы Месяцы DaysPerWeek DaysNonPeriodic HoursPerDay HoursNonPeriodic MinutesPerDay MinutesNonPeriodic SecondsPerDay SecondsNonPeriodic	
<i>mode</i>	Метка Создать	Определяет, хотите ли вы маркировать записи последовательно или построить ряд на основе указанного поля даты, метки времени или времени.
<i>поле</i>	<i>поле</i>	При построении ряда из данных задает поле, обозначающее дату или время для каждой записи.
<i>period_start</i>	<i>целое</i>	Задает начальный интервал для периодов или периодов циклов
<i>cycle_start</i>	<i>целое</i>	Начальный цикл для периодов циклов.
<i>year_start</i>	<i>целое</i>	Для применимых типов интервалов - год, в который попадает первый интервал.
<i>quarter_start</i>	<i>целое</i>	Для применимых типов интервалов - квартал, в который попадает первый интервал.
<i>month_start</i>	Январь Февраль Март Апрель Май Июнь Июль Август Сентябрь Октябрь Ноябрь Декабрь	
<i>day_start</i>	<i>целое</i>	
<i>hour_start</i>	<i>целое</i>	
<i>minute_start</i>	<i>целое</i>	
<i>second_start</i>	<i>целое</i>	
<i>periods_per_cycle</i>	<i>целое</i>	Для периодов циклов - число в каждом цикле.

Таблица 70. Свойства узла *timeintervals* (продолжение).

Свойства узла <i>timeintervals</i>	Тип переменной	Описание свойства
<i>fiscal_year_begins</i>	Январь Февраль Март Апрель Май Июнь Июль Август Сентябрь Октябрь Ноябрь Декабрь	Для квартальных интервалов задает месяц начала финансового года.
<i>week_begins_on</i>	Воскресенье Понедельник Вторник Среда Четверг Пятница Суббота Воскресенье	Для периодических интервалов (дней в неделю, часов в день, минут в день и секунд в день) задает день, с которого начинается неделя.
<i>day_begins_hour</i>	<i>целое</i>	Для периодических интервалов (часов в день, минут в день и секунд в день) задает час, с которого начинается день. Можно использовать вместе с <i>day_begins_minute</i> и <i>day_begins_second</i> , чтобы задать точное время, такое как <i>8:05:01</i> . Смотрите пример использования ниже.
<i>day_begins_minute</i>	<i>целое</i>	Для периодических интервалов (часов в день, минут в день и секунд в день) задает минуту, в которую начинается день (например, <i>5</i> в <i>8:05</i>).
<i>day_begins_second</i>	<i>целое</i>	Для периодических интервалов (часов в день, минут в день и секунд в день) задает секунду, в которую начинается день (например, <i>17</i> в <i>8:05:17</i>).
<i>days_per_week</i>	<i>целое</i>	Для периодических интервалов (дней в неделю, часов в день, минут в день и секунд в день) задает количество дней в неделю.
<i>hours_per_day</i>	<i>целое</i>	Для периодических интервалов (часов в день, минут в день и секунд в день) задает количество часов в день.
<i>interval_increment</i>	1 2 3 4 5 6 10 15 20 30	Для минут в день и секунд в день задает количество минут или секунд для инкремента при переходе к следующей записи.
<i>field_name_extension</i>	<i>строка</i>	
<i>field_name_extension_as_prefix</i>	<i>логический</i>	

Таблица 70. Свойства узла `timeintervals` (продолжение).

Свойства узла <code>timeintervals</code>	Тип переменной	Описание свойства
<code>date_format</code>	"ДДММГГ" "ММДДГГ" "ГГММДД" "ГГГГММДД" "ГГГГДД" ДЕНЬ МЕСЯЦ "ДД-ММ-ГГ" "ДД-ММ-ГГГГ" "ММ-ДД-ГГ" "ММ-ДД-ГГГГ" "ДД-МЕС-ГГ" "ДД-МЕС-ГГГГ" "ГГГГ-ММ-ДД" "ДД.ММ.ГГ" "ДД.ММ.ГГГГ" "ММ.ДД.ГГ" "ММ.ДД.ГГГГ" "ДД.МЕС.ГГ" "ДД.МЕС.ГГГГ" "ДД/ММ/ГГ" "ДД/ММ/ГГГГ" "ММ/ДД/ГГ" "ММ/ДД/ГГГГ" "ДД/МЕС/ГГ" "ДД/МЕС/ГГГГ" МЕС ГГГГ к К ГГГГ нн НД ГГГГ	
<code>time_format</code>	"ЧЧММСС" "ЧЧММ" "ММСС" "ЧЧ:ММ:СС" "ЧЧ:ММ" "ММ:СС" "(Ч)Ч:(М)М:(С)С" "(Ч)Ч:(М)М" "(М)М:(С)С" "ЧЧ.ММ.СС" "ЧЧ.ММ" "ММ.СС" "(Ч)Ч.(М)М.(С)С" "(Ч)Ч.(М)М" "(М)М.(С)С"	
<code>aggregate</code>	Mean Sum Мода Min Максимум Первое Последнее TrueIfAnyTrue	Задает метод агрегации для поля.
<code>pad</code>	Пробел MeanOfRecentPoints True False	Задает метод дополнения значений переменной длины для поля.

Таблица 70. Свойства узла *timeintervals* (продолжение).

Свойства узла <i>timeintervals</i>	Тип переменной	Описание свойства
agg_mode	All Задать	Задаёт, как агрегировать или заполнять поля - используя нужные функции по умолчанию или задавая поля и функции для использования.
agg_range_default	Mean Sum Мода Min Максимум	Задаёт функцию по умолчанию для агрегирования количественных полей.
agg_set_default	Мода Первое Последнее	Задаёт функцию по умолчанию для агрегирования номинальных полей.
agg_flag_default	TrueIfAnyTrue Мода Первое Последнее	
pad_range_default	Пробел MeanOfRecentPoints	Задаёт функцию по умолчанию для заполнения количественных полей.
pad_set_default	Пробел MostRecentValue	
pad_flag_default	Пробел True False	
max_records_to_create	<i>целое</i>	Задаёт максимальное количество записей для создания при заполнении ряда.
estimation_from_beginning	<i>логический</i>	
estimation_to_end	<i>логический</i>	
estimation_start_offset	<i>целое</i>	
estimation_num_holdouts	<i>целое</i>	
create_future_records	<i>логический</i>	
num_future_records	<i>целое</i>	
create_future_field	<i>логический</i>	
future_field_name	<i>строка</i>	

Свойства узла *transpose*



Узел Транспонирование меняет данные в строках и столбцах, чтобы записи становились полями, а поля записями.

Таблица 71. Свойства узла *transpose*.

Свойства узла <i>transpose</i>	Тип переменной	Описание свойства
transposed_names	Префикс Чтение	Новые имена полей могут быть сгенерированы автоматически на основе указанного префикса, или они могут быть считаны из уже существующего поля в данных.

Таблица 71. Свойства узла transpose (продолжение).

Свойства узла transpose	Тип переменной	Описание свойства
prefix	строка	
num_new_fields	целое	При использовании префикса задает максимальное количество новых полей для создания.
read_from_field	поле	Поле, из которого читаются имена. Это должно быть полностью определенное поле, или при выполнении узла произойдет ошибка.
max_num_fields	целое	При чтении имен из поля задает верхний предел для исключения создания неограниченно большого числа полей.
transpose_type	Числовой Строка Пользовательские	По умолчанию транспонируются только количественные поля (числового диапазона), но можно выбрать пользовательское подмножество числовых полей или вместо этого транспонировать все поля строки.
transpose_fields	[поле поле поле]	Задает поля, которые будут транспонироваться, если используется опция Настроить.
id_field_name	поле	

Свойства узла type



Узел Тип задает метаданные и свойства полей. Например, можно задать уровень измерений (количественный, номинальный, порядковый или флаговый) для каждого поля, задать опции для обработки отсутствующих значений и системных null, задавать роль поля для целей моделирования, задавать метки полей и значений и задавать значения для поля.

Обратите внимание на то, что в некоторых случаях вам может потребоваться полностью определить узел Тип, чтобы правильно работали другие узлы, например, свойство поля из узла Задать как флаг. Вы можете просто соединиться с узлом Таблица и выполнить его для полного определения полей.

Таблица 72. Свойства узла type.

Свойства узла type	Тип переменной	Описание свойства
direction	Input Target Both None Partition Split Frequency RecordID	Ключевое свойство для ролей полей. <i>Примечание:</i> Значения In и Out в настоящее время объявлены устаревшими. Их поддержка может быть прекращена в следующем выпуске.

Таблица 72. Свойства узла type (продолжение).

Свойства узла type	Тип переменной	Описание свойства
type	Range Flag Set Typeless Discrete OrderedSet Default	Тип измерений поля (прежнее название - тип поля). При задании для type значения Default будут очищены все значения параметров values, а если для value_mode задано значение Specify, оно будет изменено на Read. Если для value_mode задано Pass или Read, значение type не повлияет на value_mode. Примечание: Типы данных, используемые внутренне, отличаются от типов, видимых в узле типа. Соответствие выглядит так: Range -> Continuous Set -> Nominal OrderedSet -> Ordinal Discrete- -> Categorical
storage	Нет данных Строка Целое Действительное число Время Дата Отметка времени	Предназначенное только для чтения ключевое свойство для типа хранения поля.
check	Нет Аннулировать Принуждать Исключение Предупреждение Прервать	Ключевое свойство для проверки типа и диапазона поля.
values	[значение значение]	Для количественных полей первое значение - это минимум, а последнее - максимум. Для номинальных полей задайте все значения. Для флаговых полей первое значение представляет false, а последнее - true. Задание этого свойства автоматически устанавливает для свойства режим_значения значение Задать.
value_mode	Чтение Успех Read+ Текущий Задать	Определяет, как установлены значения. Обратите внимание на то, что вы не можете непосредственно установить для этого свойства значение Задать; чтобы использовать конкретные значения, задайте свойство значения.
extend_values	логический	Применяется, когда для режим_значения задано Чтение. Задайте T, чтобы добавить вновь прочитанные значения к любым существующим значениям для этого поля. Задайте F, чтобы отбросить существующие значения и заменить их на вновь прочитанные значения.
enable_missing	логический	Когда задано T, активирует отслеживание пропущенных значений для поля.
missing_values	[значение значение ...]	Задаёт значения данных, отмечающие пропущенные данные.
range_missing	логический	Указывает, определен ли для этого поля диапазон пропущенных (пустых) значений.

Таблица 72. Свойства узла type (продолжение).

Свойства узла type	Тип переменной	Описание свойства
missing_lower	строка	Когда для значения диапазон_отсутствия задано true, указывает нижнюю границу диапазона значений отсутствия.
missing_upper	строка	Когда для значения диапазон_отсутствия задано true, указывает верхнюю границу диапазона значений отсутствия.
null_missing	логический	Когда для этого свойства задано T, значения <i>nulls</i> (не определенные значения, обозначаемые в программах как \$null\$) рассматриваются как значения отсутствия.
whitespace_missing	логический	Когда для этого свойства задано T, значения, содержащие только пробельные символы (пробелы, знаки табуляции и новой строки) рассматриваются как значения отсутствия.
description	строка	Задаёт описание для поля.
value_labels	[{строка значение метка} {строка значение метка} ...]	Используется для задания меток парам значений.
display_places	целое	Задаёт количество десятичных разрядов при выводе поля (применимо только к полям с системой хранения REAL). При значении -1 будут использоваться значения потока по умолчанию.
export_places	целое	Задаёт количество десятичных разрядов при экспорте поля (применимо только к полям с системой хранения REAL). При значении -1 будут использоваться значения потока по умолчанию.
decimal_separator	DEFAULT PERIOD COMMA	Задаёт десятичный разделитель для поля (применимо только к полям с системой хранения REAL).

Таблица 72. Свойства узла type (продолжение).

Свойства узла type	Тип переменной	Описание свойства
date_format	"ДДММГГ" "ММДДГГ" "ГГММДД" "ГГГГММДД" "ГГГГДД" ДЕНЬ МЕСЯЦ "ДД-ММ-ГГ" "ДД-ММ-ГГГГ" "ММ-ДД-ГГ" "ММ-ДД-ГГГГ" "ДД-МЕС-ГГ" "ДД-МЕС-ГГГГ" "ГГГГ-ММ-ДД" "ДД.ММ.ГГ" "ДД.ММ.ГГГГ" "ММ.ДД.ГГ" "ММ.ДД.ГГГГ" "ДД.МЕС.ГГ" "ДД.МЕС.ГГГГ" "ДД/ММ/ГГ" "ДД/ММ/ГГГГ" "ММ/ДД/ГГ" "ММ/ДД/ГГГГ" "ДД/МЕС/ГГ" "ДД/МЕС/ГГГГ" МЕС ГГГГ к К ГГГГ нн НД ГГГГ	Задает формат даты для поля (применимо только к полям с системой хранения DATE или TIMESTAMP).
time_format	"ЧЧММСС" "ЧЧММ" "ММСС" "ЧЧ:ММ:СС" "ЧЧ:ММ" "ММ:СС" "(Ч)Ч:(М)М:(С)С" "(Ч)Ч:(М)М" "(М)М:(С)С" "ЧЧ.ММ.СС" "ЧЧ.ММ" "ММ.СС" "(Ч)Ч.(М)М.(С)С" "(Ч)Ч.(М)М" "(М)М.(С)С"	Задает формат времени для поля (применимо только к полям с системой хранения TIME или TIMESTAMP).
number_format	DEFAULT STANDARD SCIENTIFIC CURRENCY	Задает формат вывода чисел для поля.
standard_places	<i>целое</i>	Задает количество десятичных разрядов при выводе поля в стандартном формате. При значении -1 будут использоваться значения потока по умолчанию. Обратите внимание на то, что существующий слот разряды_вывода может также использоваться для изменения, но сейчас он объявлен устаревшим.
scientific_places	<i>целое</i>	Задает количество десятичных разрядов при выводе поля в экспоненциальном представлении. При значении -1 будут использоваться значения потока по умолчанию.

Таблица 72. Свойства узла type (продолжение).

Свойства узла type	Тип переменной	Описание свойства
currency_places	<i>целое</i>	Задаёт количество десятичных разрядов при выводе поля в формате валюты. При значении -1 будут использоваться значения потока по умолчанию.
grouping_symbol	DEFAULT NONE LOCALE PERIOD COMMA SPACE	Задаёт знак группировки для поля.
column_width	<i>целое</i>	Задаёт ширину столбца для поля. При значении -1 для ширины столбца будет задано Auto.
justify	AUTO CENTER LEFT RIGHT	Задаёт выравнивание столбцов для поля.

Глава 12. Свойства узла графика

Общие свойства узла графика

В этом разделе описаны свойства, доступные для узлов графиков, в том числе общие свойства и свойства, специфичные для каждого типа узла.

Таблица 73. Свойства *Common graph node*.

Свойства <i>Common graph node</i>	Тип переменной	Описание свойства
title	строка	Задаёт заголовок. Пример: "Это заголовок".
caption	строка	Задаёт подпись. Пример: "Это подпись".
output_mode	Screen File	Задаёт способ обработки выходных данных с узла графика - будут ли они выводиться на экран, или записываться в файл.
output_format	BMP JPEG PNG HTML output (.cou)	Задаёт тип выходных данных. Точный тип выходных данных, разрешенный для каждого из узлов, варьируется.
full_filename	строка	Задаёт путь назначения и имя файла для выходных данных, сгенерированных на узле графика.
use_graph_size	логический	Управляет точностью определения размера графика, используя свойства ширины и высоты ниже. Влияет только на графики, которые выводятся на экран. Недоступно для узла Распределение.
graph_width	число	Когда значение use_graph_size - это True, задаёт ширину графика в пикселях.
graph_height	число	Когда значение use_graph_size - это True, задаёт высоту графика в пикселях.

Примечания

Выключение дополнительных полей. Дополнительные поля, такие как поля наложения для диаграмм, можно выключить при задании для свойства значения " " (пустая строка).

Задание цветов. Цвета для заголовков, подписей, фона и меток можно задать, используя шестнадцатиричные строки, начинающиеся с символа решетки (#).

Первые две цифры задают долю красного; средние две цифры - зеленого; последние две цифры - синего. У каждой цифры диапазон изменения 0–9 или A–F. Совместно эти значения задают цвет в RGB (красный, зеленый, голубой).

Примечание: При задании цвета в RGB можно использовать средство выбора полей в пользовательском интерфейсе, чтобы определить правильный код цвета. Просто наведите указатель мыши на цвет, чтобы активировать подсказку с нужной информацией.

Свойства узла collection



Узел Собрание показывает распределение значений для одного числового поля относительно значений другого. (При этом создаются диаграммы, похожие на гистограммы). Это полезно для иллюстрации переменной или поля, значения которых изменяется во времени. Используя 3D-представление, вы можете включить также символическую ось, показывающую распределения по категориям.

Таблица 74. Свойства узла collection.

Свойства узла collection	Тип переменной	Описание свойства
over_field	поле	
over_label_auto	логический	
over_label	строка	
collect_field	поле	
collect_label_auto	логический	
collect_label	строка	
three_D	логический	
by_field	поле	
by_label_auto	логический	
by_label	строка	
operation	Sum Mean Min Max SDev	
color_field	строка	
panel_field	строка	
animation_field	строка	
range_mode	Автоматически UserDefined	
range_min	число	
range_max	число	
интервалы	ByNumber ByWidth	
num_bins	число	
bin_width	число	
use_grid	логический	
graph_background	цвет	Стандартные цвета графиков описаны в начале этого раздела.
page_background	цвет	Стандартные цвета графиков описаны в начале этого раздела.

Свойства узла distribution



Узел распределения показывает появление символических (категориальных) значений, таких как тип закладных или пол. Обычно узел распределения используется для показа разбалансировки данных, которую можно выправить при помощи узла балансировки до создания модели.

Таблица 75. Свойства узла distribution.

Свойства узла distribution	Тип переменной	Описание свойства
plot	SelectedFields Флаги	
x_field	поле	
color_field	поле	Поле наложения.
normalize	логический	
sort_mode	ByOccurrence По алфавиту	
use_proportional_scale	логический	

Свойства узла evaluation



Узел Оценка помогает оценить и сравнить прогнозирующие модели. Диаграмма оценки показывает, насколько хорошо модели предсказывают конкретные выходные данные. Он сортирует записи на основе предсказанного значения и доверительного интервала предсказания. Он разбивает записи на группы равного размера (**квантили**) и затем выводит значение бизнес-критерия для каждой квантили от самой высокой до самой низкой. Несколько моделей представляются разными линиями на графике.

Таблица 76. Свойства узла evaluation.

Свойства узла evaluation	Тип переменной	Описание свойства
chart_type	Рост Отклик Рост Доход ROI ROC	
inc_baseline	логический	
field_detection_method	Метаданные Имя	
use_fixed_cost	логический	
cost_value	число	
cost_field	строка	
use_fixed_revenue	логический	
revenue_value	число	
revenue_field	строка	
use_fixed_weight	логический	
weight_value	число	
weight_field	поле	

Таблица 76. Свойства узла *evaluation* (продолжение).

Свойства узла <i>evaluation</i>	Тип переменной	Описание свойства
<i>n_tile</i>	Квартили Quintles Децили Вингтили Процентили 1000-тили	
<i>cumulative</i>	<i>флаг</i>	
<i>style</i>	Line Точка	
<i>point_type</i>	Прямоугольник Точки Треугольник Шестиугольник Плюс Пятиугольник Звезда BowTie HorizontalDash VerticalDash IronCross Фабрика Дом Собор OnionDome ConcaveTriangle OblateGlobe CatEye FourSidedPillow RoundRectangle Веер	
<i>export_data</i>	<i>логический</i>	
<i>data_filename</i>	<i>строка</i>	
<i>delimiter</i>	<i>строка</i>	
<i>new_line</i>	<i>логический</i>	
<i>inc_field_names</i>	<i>логический</i>	
<i>inc_best_line</i>	<i>логический</i>	
<i>inc_business_rule</i>	<i>логический</i>	
<i>business_rule_condition</i>	<i>строка</i>	
<i>plot_score_fields</i>	<i>логический</i>	
<i>score_fields</i>	<i>[поле1 ... полеN]</i>	
<i>target_field</i>	<i>поле</i>	
<i>use_hit_condition</i>	<i>логический</i>	
<i>hit_condition</i>	<i>строка</i>	
<i>use_score_expression</i>	<i>логический</i>	
<i>score_expression</i>	<i>строка</i>	
<i>caption_auto</i>	<i>логический</i>	

Свойства узла graphboard



Узел Панель выбора диаграмм предлагает много разных типов диаграмм на одном узле. Используя этот узел, можно выбрать поля данных, которые вы хотите изучать, а затем выбрать диаграмму из доступных для выбранных данных. Узел автоматически отфильтровывает все типы диаграмм, которые нельзя использовать для работы с выбранными полями.

Примечание: Если задается свойство, недопустимое для типа диаграммы (например, для гистограммы указывается поле_y), это свойство игнорируется.

Таблица 77. Свойства узла graphboard.

Свойства узла graphboard	Тип переменной	Описание свойства
graph_type	2DDotplot 3DArea 3DBar 3DDensity 3DHistogram 3DPie 3DScatterplot Область ArrowMap Столбцы BarCounts BarCountsMap BarMap BinnedScatter Коробчатая диаграмма Пузырьковая ChoroplethMeans ChoroplethMedians ChoroplethSums ChoroplethValues ChoroplethCounts CoordinateMap CoordinateChoroplethMeans CoordinateChoroplethMedians CoordinateChoroplethSums CoordinateChoroplethValues CoordinateChoroplethCounts Точечная диаграмма Тепловая карта HexBinScatter Гистограмма Line LineChartMap LineOverlayMap Параллельная Путь Круг PieCountMap PieCounts PieMap PointOverlayMap PolygonOverlayMap Лента Диаграмма рассеяния SPLOM Поверхность	Определяет тип диаграммы.
x_field	поле	Задаёт пользовательскую метку для оси x. Доступно только для меток.
y_field	поле	Задаёт пользовательскую метку для оси y. Доступно только для меток.
z_field	поле	Используется в некоторых трехмерных диаграммах.
color_field	поле	Используется на картах интенсивности.
size_field	поле	Используется на диаграммах с пузырями.

Таблица 77. Свойства узла *graphboard* (продолжение).

Свойства узла <i>graphboard</i>	Тип переменной	Описание свойства
categories_field	поле	
values_field	поле	
rows_field	поле	
columns_field	поле	
fields	поле	
start_longitude_field	поле	Используется со стрелками на опорной карте.
end_longitude_field	поле	
start_latitude_field	поле	
end_latitude_field	поле	
data_key_field	поле	Используется на различных картах.
panelrow_field	строка	
panelcol_field	строка	
animation_field	строка	
longitude_field	поле	Используется с координатами на картах.
latitude_field	поле	
map_color_field	поле	

Свойства узла *histogram*



Узел Гистограмма показывает существующие значения для числовых полей. Он часто используется для изучения данных перед работой с ними и построением моделей. Аналогично узлу Распределение узел Гистограмма часто выявляет несбалансированность данных.

Таблица 78. Свойства узла *histogram*.

Свойства узла <i>histogram</i>	Тип переменной	Описание свойства
поле	поле	
color_field	поле	
panel_field	поле	
animation_field	поле	
range_mode	Автоматически UserDefined	
range_min	число	
range_max	число	
интервалы	ByNumber ByWidth	
num_bins	число	
bin_width	число	
normalize	логический	
separate_bands	логический	
x_label_auto	логический	

Таблица 78. Свойства узла histogram (продолжение).

Свойства узла histogram	Тип переменной	Описание свойства
x_label	строка	
y_label_auto	логический	
y_label	строка	
use_grid	логический	
graph_background	цвет	Стандартные цвета графиков описаны в начале этого раздела.
page_background	цвет	Стандартные цвета графиков описаны в начале этого раздела.
normal_curve	логический	Указывает, должна ли кривая нормального распределения показываться при выводе.

Свойства узла multiplot



Узел нескольких графиков (Multiplot) создает график, выводящий несколько полей *Y* по отношению к одному полю *X*. Значения полей *Y* изображаются на графике как цветные линии, каждая из которых эквивалентна графику на узле График при заданном значении стиля **Линия** и режиме **X Сортировка**. Узел Несколько графиков полезен, когда нужно исследовать флуктуации нескольких переменных во времени.

Таблица 79. Свойства узла multiplot.

Свойства узла multiplot	Тип переменной	Описание свойства
x_field	поле	
y_fields	[поле поле поле]	
panel_field	поле	
animation_field	поле	
normalize	логический	
use_overlay_expr	логический	
overlay_expression	строка	
records_limit	число	
if_over_limit	PlotBins PlotSample PlotAll	
x_label_auto	логический	
x_label	строка	
y_label_auto	логический	
y_label	строка	
use_grid	логический	
graph_background	цвет	Стандартные цвета графиков описаны в начале этого раздела.
page_background	цвет	Стандартные цвета графиков описаны в начале этого раздела.

Свойства узла plot



Узел График показывает взаимосвязь между численными полями. Графики можно создавать, используя точки (диаграммы рассеяния) или линии.

Таблица 80. Свойства узла plot.

Свойства узла plot	Тип переменной	Описание свойства
x_field	поле	Задаёт пользовательскую метку для оси x. Доступно только для меток.
y_field	поле	Задаёт пользовательскую метку для оси y. Доступно только для меток.
three_D	логический	Задаёт пользовательскую метку для оси z. Доступно только для меток трёхмерных графиков.
z_field	поле	
color_field	поле	Поле наложения.
size_field	поле	
shape_field	поле	
panel_field	поле	Задаёт номинальное или флаговое поле для использования при построении отдельной диаграммы для каждой категории. Диаграммы располагаются совместно в одном выходном окне.
animation_field	поле	Задаёт номинальное или флаговое поле для иллюстрации категорий значений данных посредством создания набора диаграмм, показываемых последовательно с использованием анимации.
transp_field	поле	Задаёт номинальное или флаговое поле для иллюстрации категорий значений данных посредством различных уровней прозрачности для каждой категории. Недоступно для линейных графиков.
overlay_type	Нет Сглаживатель Function	Задаёт, что будет выводиться, функция наложения или сглаживатель LOESS.
overlay_expression	строка	Задаёт выражение, которое используется, когда для overlay_type задано значение Function.
style	Точка Line	

Таблица 80. Свойства узла plot (продолжение).

Свойства узла plot	Тип переменной	Описание свойства
point_type	Прямоугольник Точки Треугольник Шестиугольник Плюс Пятиугольник Звезда BowTie HorizontalDash VerticalDash IronCross Фабрика Дом Собор OnionDome ConcaveTriangle OblateGlobe CatEye FourSidedPillow RoundRectangle Веер	
x_mode	Сортировка Наложение AsRead	
x_range_mode	Автоматически UserDefined	
x_range_min	<i>число</i>	
x_range_max	<i>число</i>	
y_range_mode	Автоматически UserDefined	
y_range_min	<i>число</i>	
y_range_max	<i>число</i>	
z_range_mode	Автоматически UserDefined	
z_range_min	<i>число</i>	
z_range_max	<i>число</i>	
jitter	<i>логический</i>	
records_limit	<i>число</i>	
if_over_limit	PlotBins PlotSample PlotAll	
x_label_auto	<i>логический</i>	
x_label	<i>строка</i>	
y_label_auto	<i>логический</i>	
y_label	<i>строка</i>	
z_label_auto	<i>логический</i>	
z_label	<i>строка</i>	
use_grid	<i>логический</i>	
graph_background	<i>цвет</i>	Стандартные цвета графиков описаны в начале этого раздела.

Таблица 80. Свойства узла plot (продолжение).

Свойства узла plot	Тип переменной	Описание свойства
page_background	цвет	Стандартные цвета графиков описаны в начале этого раздела.
use_overlay_expr	логический	Объявлено устаревшим с заменой на overlay_type.

Свойства узла timeplot



Узел Временной график выводит один или несколько наборов данных временных рядов. Обычно вы сначала используете узел Временные интервалы для создания поля *TimeLabel*, которое будет использовано для отметок по оси *x*.

Таблица 81. Свойства узла timeplot.

Свойства узла timeplot	Тип переменной	Описание свойства
plot_series	Ряды Модели	
use_custom_x_field	логический	
x_field	поле	
y_fields	[поле поле поле]	
panel	логический	
normalize	логический	
line	логический	
points	логический	
point_type	Прямоугольник Точки Треугольник Шестиугольник Плюс Пятиугольник Звезда BowTie HorizontalDash VerticalDash IronCross Фабрика Дом Собор OnionDome ConcaveTriangle OblateGlobe CatEye FourSidedPillow RoundRectangle Веер	
smoother	логический	Добавить к графику сглаживатели можно только в том случае, если для panel вы задали значение True.
use_records_limit	логический	
records_limit	целое	
symbol_size	число	Задаёт размер знака.

Таблица 81. Свойства узла *timeplot* (продолжение).

Свойства узла <i>timeplot</i>	Тип переменной	Описание свойства
panel_layout	Горизонтально Вертикальный	

Свойства узла *web*



Узел *Web* иллюстрирует силу взаимосвязи между значениями двух или более символических (категориальных) полей. На графике используются линии разной ширины для обозначения силы соединения. Например, вы можете использовать узел *Web* для изучения взаимосвязи между покупкой набора товаров на сайте интернет-магазина.

Таблица 82. Свойства узла *web*.

Свойства узла <i>web</i>	Тип переменной	Описание свойства
use_directed_web	логический	
fields	[поле поле поле]	
to_field	поле	
from_fields	[поле поле поле]	
true_flags_only	логический	
line_values	Абсолютная OverallPct PctLarger PctSmaller	
strong_links_heavier	логический	
num_links	ShowMaximum ShowLinksAbove ShowAll	
max_num_links	число	
links_above	число	
discard_links_min	логический	
links_min_records	число	
discard_links_max	логический	
links_max_records	число	
weak_below	число	
strong_above	число	
link_size_continuous	логический	
web_display	Circular Network Directed Сетка	
graph_background	цвет	Стандартные цвета графиков описаны в начале этого раздела.
symbol_size	число	Задает размер знака.

Глава 13. Свойства узла моделирования

Общие свойства узлов моделирования

Следующие свойства общие для всех или некоторых узлов моделирования. Любые исключительные ситуации соответствующим образом отмечаются в документации для индивидуальных узлов моделирования.

Таблица 83. Общие свойства узлов моделирования.

Свойство	Значения	Описание свойства
custom_fields	логический	Если значение флага true, это позволяет вам задать поле назначения, входные и другие поля для текущего узла. При значении false используются текущие параметры с вышележащего узла Тип.
target или targets	поле или [поле1 ... полеN]	Задаёт одно поле назначения или несколько полей назначения в зависимости от типа модели.
inputs	[поле1 ... полеN]	Входные (или предикторные) поля, используемые в модели.
partition	поле	
use_partitioned_data	логический	Если определено поле раздела, эта опция обеспечивает, что для построения модели используются только данные из раздела обучения.
use_split_data	логический	
splits	[поле1 ... полеN]	Задаёт поле или поля, которые будут использоваться для моделирования разбиения. Действует только в том случае, если для use_split_data задано значение True.
use_frequency	логический	Поля веса и частоты используются конкретными моделями, как отмечено для каждого типа модели.
frequency_field	поле	
use_weight	логический	
weight_field	поле	
use_model_name	логический	
model_name	строка	Пользовательское имя для новой модели.
mode	Простые Эксперт	

Свойства узла anomalydetection



Узел выявления аномалий определяет необычные наблюдения, или выбросы, которые не соответствуют структуре “нормальных” данных. При помощи этого узла можно находить выбросы даже в том случае, если они не подходят ни под какие ранее известные шаблоны или вы точно не уверены, что именно ищете.

Таблица 84. Свойства узла anomalydetection.

Свойства узла anomalydetection	Значения	Описание свойства
inputs	[поле1 ... полеN]	Модели обнаружения аномалий изучают данные на основе заданных входных полей. Они не используют поле назначения. Поля веса и частоты также не используются. Дополнительную информацию смотрите в разделе “Общие свойства узлов моделирования” на стр. 127.
mode	Эксперт Простые	
anomaly_method	IndexLevel PerRecords NumRecords	Задаёт способ определения значения отсечения для пометки записей как аномальных.
index_level	число	Задаёт минимальное значение отсечения для пометки аномалий.
percent_records	число	Задаёт порог для отметки записей на основании процентной доли записей в данных обучения.
num_records	число	Задаёт порог для отметки записей на основании количества записей в данных обучения.
num_fields	целое	Количество полей для отчета о каждой аномальной записи.
impute_missing_values	логический	
adjustment_coeff	число	Значение для балансировки относительного веса, предназначенное для количественных и категориальных полей при вычислении расстояния.
peer_group_num_auto	логический	Автоматически вычисляет количество равноправных групп.
min_num_peer_groups	целое	Задаёт минимальное количество равноправных групп, используемых, если для поля peer_group_num_auto задано значение True.
max_num_per_groups	целое	Задаёт максимальное количество равноправных групп.
num_peer_groups	целое	Задаёт минимальное количество равноправных групп, используемых, если для поля peer_group_num_auto задано значение False.
noise_level	число	Определяет, как при кластеризации обрабатываются выбросы. Задайте значение от 0 до 0,5.

Таблица 84. Свойства узла anomalydetection (продолжение).

Свойства узла anomalydetection	Значения	Описание свойства
noise_ratio	число	Задаёт выделенную для компонента долю памяти, которая должна использоваться для буферизации шума. Задайте значение от 0 до 0,5.

Свойства узла apriori



Узел Априори извлекает набор правил из данных, выделяя правила с наибольшим информационным содержанием. Узел Априори предлагает пять различных способов выбора правил и использует сложные схемы индексирования для эффективной обработки больших наборов данных. Для больших задач узел Априори обычно быстрее при обучении; у него нет произвольного ограничения количества правил, которые можно сохранить, и он может обрабатывать правила с количеством предварительных условий до 32. Для узла Априори требуются категориальные входные и выходные поля, он был оптимизирован для полей такого типа и показывает с ними высокую производительность.

Таблица 85. Свойства узла apriori.

Свойства узла apriori	Значения	Описание свойства
consequents	поле	Априорные модели используют консеквенты и antecedentes вместо стандартных полей назначения и входных полей. Поля веса и частоты не используются. Дополнительную информацию смотрите в разделе “Общие свойства узлов моделирования” на стр. 127.
antecedents	[поле1 ... полеN]	
min_supp	число	
min_conf	число	
max_antecedents	число	
true_flags	логический	
optimize	Скорость Память	
use_transactional_data	логический	
contiguous	логический	
id_field	строка	
content_field	строка	
mode	Простые Эксперт	
evaluation	RuleConfidence DifferenceToPrior ConfidenceRatio InformationDifference NormalizedChiSquare	
lower_bound	число	
optimize	Скорость Память	Используется для определения, нужно ли при построении модели оптимизировать скорость или использование памяти.

Свойства узла autoclassifier



Узел автоклассификации создает и сравнивает несколько различных моделей для двоичных выходных данных (да или нет, уйдет клиент или останется и так далее), что позволяет выбрать лучший подход для данного анализа. Поддерживается несколько алгоритмов моделирования, что делает возможным выбор желательных для использования способов, конкретных опций для каждого из них и критериев сравнения результатов. Этот узел генерирует набор моделей на основе заданных опций и ранжирует лучших кандидатов в соответствии с заданными вами критериями.

Таблица 86. Свойства узла autoclassifier.

Свойства узла autoclassifier	Значения	Описание свойства
target	поле	Для флаговых полей назначения узлу автоклассификации требуется одно поле назначения и одно или несколько входных полей. Можно задать также поля веса и частоты. Дополнительную информацию смотрите в разделе “Общие свойства узлов моделирования” на стр. 127.
ranking_measure	Точность Area_under_curve Доход Рост Num_variables	
ranking_dataset	Обучающее Критерий	
number_of_models	целое	Количество моделей для включения в слепок моделей. Указать целое число от 1 до 100.
calculate_variable_importance	логический	
enable_accuracy_limit	логический	
accuracy_limit	целое	Целое число от 0 до 100.
enable_area_under_curve_limit	логический	
area_under_curve_limit	число	Действительное число от 0,0 до 1,0.
enable_profit_limit	логический	
profit_limit	число	Целое число больше 0.
enable_lift_limit	логический	
lift_limit	число	Действительное число, большее 1,0.
enable_number_of_variables_limit	логический	
number_of_variables_limit	число	Целое число больше 0.
use_fixed_cost	логический	
fixed_cost	число	Действительное число, большее 0,0.
variable_cost	поле	
use_fixed_revenue	логический	
fixed_revenue	число	Действительное число, большее 0,0.
variable_revenue	поле	
use_fixed_weight	логический	
fixed_weight	число	Действительное число, большее 0,0
variable_weight	поле	

Таблица 86. Свойства узла *autoclassifier* (продолжение).

Свойства узла <i>autoclassifier</i>	Значения	Описание свойства
<code>lift_percentile</code>	<i>число</i>	Целое число от 0 до 100.
<code>enable_model_build_time_limit</code>	<i>логический</i>	
<code>model_build_time_limit</code>	<i>число</i>	Целое число, заданное для числа минут, чтобы ограничить время на построение каждой конкретной модели.
<code>enable_stop_after_time_limit</code>	<i>логический</i>	
<code>stop_after_time_limit</code>	<i>число</i>	Действительное число, заданное для количества часов для ограничения общего используемого времени на выполнение автоклассификации.
<code>enable_stop_after_valid_model_produced</code>	<i>логический</i>	
<code>use_costs</code>	<i>логический</i>	
<алгоритм>	<i>логический</i>	Включает или отключает использование конкретного алгоритма.
<алгоритм>.<свойство>	<i>строка</i>	Задаёт значение свойства для конкретного алгоритма. Дополнительную информацию смотрите в разделе “Задание свойств алгоритмов”.

Задание свойств алгоритмов

Имена алгоритмов для узла автоклассификации - это `cart`, `chaid`, `quest`, `c50`, `logreg`, `decisionlist`, `bayesnet`, `discriminant`, `svm` и `knn`.

Имена алгоритмов для узла автонумерации - это `cart`, `chaid`, `neuralnetwork`, `genlin`, `svm`, `regression`, `linear` и `knn`.

Имена алгоритмов для узла автокластеризации - это `twostep`, `k-means` и `kohonen`.

Имена свойств стандартные, как документировано для каждого узла алгоритмов.

Свойства алгоритмов, содержащие точки или другие знаки пунктуации, надо заключать в одинарные кавычки.

Для свойства можно назначить также несколько значений.

Замечания :

- При задании значений `true` и `false` нужно использовать нижний регистр (например, не `False`).
- В тех случаях, когда некоторые опции алгоритмов недоступны на узле автоклассификации, или можно задать только одно значение, а не диапазон значений, такие же ограничения применяются к сценариям и при обращении к узлу обычным образом.

Свойства узла autocluster



Узел автоматической кластеризации оценивает и сравнивает модели кластеризации, идентифицирующие группы записей со сходными характеристиками. Этот узел работает аналогично другим узлам автоматического моделирования, допуская экспериментирование с несколькими комбинациями опций при одном проходе моделирования. Модели можно сравнивать при помощи базовых показателей, пытаясь фильтровать и ранжировать с их использованием полезность моделей кластеризации и предоставить показатель на основе важности конкретных полей.

Таблица 87. Свойства узла autocluster.

Свойства узла autocluster	Значения	Описание свойства
evaluation	поле	<i>Примечание:</i> Только для узла автокластеризации. Определяет поле, для которого будет вычислено значение важности. Как вариант, может использоваться для определения, насколько хорошо кластер различает значение этого поля, и как следствие - насколько хорошо модель предскажет значение в этом поле.
ranking_measure	Силуэтная мера Num_clusters Size_smallest_cluster Size_largest_cluster Smallest_to_largest Важность	
ranking_dataset	Обучающее Критерий	
summary_limit	целое	Количество моделей для перечисления в отчете. Указать целое число от 1 до 100.
enable_silhouette_limit	логический	
silhouette_limit	целое	Целое число от 0 до 100.
enable_number_less_limit	логический	
number_less_limit	число	Действительное число от 0,0 до 1,0.
enable_number_greater_limit	логический	
number_greater_limit	число	Целое число больше 0.
enable_smallest_cluster_limit	логический	
smallest_cluster_units	Процент Количества	
smallest_cluster_limit_percentage	число	
smallest_cluster_limit_count	целое	Целое число больше 0.
enable_largest_cluster_limit	логический	
largest_cluster_units	Процент Количества	
largest_cluster_limit_percentage	число	
largest_cluster_limit_count	целое	
enable_smallest_largest_limit	логический	
smallest_largest_limit	число	
enable_importance_limit	логический	

Таблица 87. Свойства узла *autocluster* (продолжение).

Свойства узла <i>autocluster</i>	Значения	Описание свойства
<i>importance_limit_condition</i>	Greater_than Less_than	
<i>importance_limit_greater_than</i>	<i>число</i>	Целое число от 0 до 100.
<i>importance_limit_less_than</i>	<i>число</i>	Целое число от 0 до 100.
<алгоритм>	<i>логический</i>	Включает или отключает использование конкретного алгоритма.
<алгоритм>.<свойство>	<i>строка</i>	Задает значение свойства для конкретного алгоритма. Дополнительную информацию смотрите в разделе “Задание свойств алгоритмов” на стр. 131.

Свойства узла *autonumeric*



Узел автономерации оценивает и сравнивает модели для выходных данных в количественном числовом диапазоне при помощи нескольких разных способов. Этот узел работает аналогично другим узлам автоклассификации, допуская выбор алгоритмов для использования и экспериментирование с несколькими комбинациями опций при одном проходе моделирования. Поддерживаемые алгоритмы включают в себя нейросети, дерево C&R, CHAID, линейную регрессию, обобщенную линейную регрессию и механизмы опорных векторов (support vector machines, SVM). Модели можно сравнивать на основе корреляции, относительной ошибки или числа используемых переменных.

Таблица 88. Свойства узла *autonumeric*.

Свойства узла <i>autonumeric</i>	Значения	Описание свойства
<i>custom_fields</i>	<i>логический</i>	При значении True вместо параметров узла Тип будут использоваться пользовательские параметры полей.
<i>target</i>	<i>поле</i>	Для узла автономерации требуется одно поле назначения и одно или несколько входных полей. Можно задать также поля веса и частоты. Дополнительную информацию смотрите в разделе “Общие свойства узлов моделирования” на стр. 127.
<i>inputs</i>	[<i>поле1 ... поле2</i>]	
<i>partition</i>	<i>поле</i>	
<i>use_frequency</i>	<i>логический</i>	
<i>frequency_field</i>	<i>поле</i>	
<i>use_weight</i>	<i>логический</i>	
<i>weight_field</i>	<i>поле</i>	
<i>use_partitioned_data</i>	<i>логический</i>	Если определено поле раздела, для построения модели используются только данные из раздела обучения.
<i>ranking_measure</i>	Корреляция NumberOfFields	
<i>ranking_dataset</i>	Критерий Обучающее	

Таблица 88. Свойства узла *autonumeric* (продолжение).

Свойства узла <i>autonumeric</i>	Значения	Описание свойства
number_of_models	целое	Количество моделей для включения в слепок моделей. Указать целое число от 1 до 100.
calculate_variable_importance	логический	
enable_correlation_limit	логический	
correlation_limit	целое	
enable_number_of_fields_limit	логический	
number_of_fields_limit	целое	
enable_relative_error_limit	логический	
relative_error_limit	целое	
enable_model_build_time_limit	логический	
model_build_time_limit	целое	
enable_stop_after_time_limit	логический	
stop_after_time_limit	целое	
stop_if_valid_model	логический	
<алгоритм>	логический	Включает или отключает использование конкретного алгоритма.
<алгоритм>.<свойство>	строка	Задаёт значение свойства для конкретного алгоритма. Дополнительную информацию смотрите в разделе “Задание свойств алгоритмов” на стр. 131.

Свойства узла *bayesnet*



Узел Байесовская сеть позволяет построить вероятностную модель, комбинируя наблюдаемые и записанные сведения с очевидными с точки зрения здравого смысла данными, чтобы установить правдоподобие возникновения событий. Этот узел в основном работает с усиленным деревом наивными байесовскими сетями (Tree Augmented Naïve Bayes, TAN) и полными марковскими сетями, которые изначально используются для классификации.

Таблица 89. Свойства узла *bayesnet*.

Свойства узла <i>bayesnet</i>	Значения	Описание свойства
inputs	[поле1 ... полеN]	Модели Байесовской сети используют одно поле назначения и одно или несколько входных полей. Количественные поля автоматически разбиваются на интервалы. Дополнительную информацию смотрите в разделе “Общие свойства узлов моделирования” на стр. 127.
continue_training_existing_model	логический	
structure_type	TAN MarkovBlanket	Выберите структуру, которая будет использоваться при построении Байесовской сети.
use_feature_selection	логический	

Таблица 89. Свойства узла bayesnet (продолжение).

Свойства узла bayesnet	Значения	Описание свойства
parameter_learning_method	Вероятность Bayes	Задаёт способ, используемый для оценки таблиц условной вероятности между узлами, когда родительские элементы известны.
mode	Эксперт Простые	
missing_values	логический	
all_probabilities	логический	
independence	Вероятность Пирсона	Задаёт способ, используемый для определения, независимы ли друг от друга парные наблюдения двух переменных.
significance_level	число	Задаёт значение отсечения для определения независимости.
maximal_conditioning_set	число	Задаёт максимальное количество переменных настройки, которые будут использоваться для проверки независимости.
inputs_always_selected	[поле1 ... полеN]	Задаёт, какие поля из набора данных всегда будут использоваться при построении Байесовской сети. <i>Примечание:</i> Поле назначения выбирается всегда.
maximum_number_inputs	число	Задаёт максимальное количество входных полей, которые будут использоваться при построении Байесовской сети.
calculate_variable_importance	логический	
calculate_raw_propensities	логический	
calculate_adjusted_propensities	логический	
adjusted_propensity_partition	Критерий Проверка	

Свойства узла buildr



Узел построения R позволяет ввести пользовательский сценарий R для выполнения построения и скоринга модели, внедренной в IBM SPSS Modeler.

Таблица 90. свойства buildr.

Свойства узла buildr	Значения	Описание свойства
build_syntax	строка	Синтаксис сценария R для построения моделей.
score_syntax	строка	Синтаксис сценария R для скоринга моделей.
convert_flags	StringsAndDoubles LogicalValues	Опция для преобразования флаговых полей.

Таблица 90. свойства buildr (продолжение).

Свойства узла buildr	Значения	Описание свойства
convert_datetime	логическое	Опция для преобразования переменных с форматом данных даты или даты-времени в форматы даты/времени R.
convert_datetime_class	POSIXct POSIXlt	Опции для указания, в какой формат нужно конвертировать переменные из формата даты или даты-времени.
convert_missing	логический	Опция для преобразования пропущенных значений в значение R NA.
output_html	логический	Опция для вывода графиков на вкладке в слепке модели R.
output_text	логическое	Опция для записи текстового вывода консоли R на вкладку в слепке модели R.

Свойства узла c50



Узел C5.0 строит или дерево решений, или набор правил. Эта модель работает, разделяя выборку на основании значения в поле, дающего максимальный информационный выигрыш на каждом уровне. Поле назначения должно быть категориальным. Разрешено несколько разделений на подгруппы, и таких подгрупп может быть больше двух.

Таблица 91. Свойства узла c50.

Свойства узла c50	Значения	Описание свойства
target	поле	Модели C50 используют одно поле назначения и одно или несколько входных полей. Можно задать также поле веса. Дополнительную информацию смотрите в разделе “Общие свойства узлов моделирования” на стр. 127.
output_type	DecisionTree RuleSet	
group_symbolics	логический	
use_boost	логический	
boost_num_trials	число	
use_xval	логический	
xval_num_folds	число	
mode	Простые Эксперт	
favor	Точность Общность	Предпочтение точности или общности.
expected_noise	число	
min_child_records	число	
pruning_severity	число	
use_costs	логический	
costs	структурированный	Это структурированное свойство.
use_winnowing	логический	
use_global_pruning	логический	По умолчанию значение (True).

Таблица 91. Свойства узла c50 (продолжение).

Свойства узла c50	Значения	Описание свойства
calculate_variable_importance	логический	
calculate_raw_propensities	логический	
calculate_adjusted_propensities	логический	
adjusted_propensity_partition	Критерий Проверка	

Свойства узла carma



Модель CARMA извлекает из данных набор правил, не требуя, чтобы вы задавали входные или выходные поля. В отличие от узла Априори, узел CARMA предлагает параметры построения для поддержки правил (поддержка относится и к антецедентам, и к консеквентам), а не только для поддержки антецедентов. Это означает, что сгенерированные правила можно использовать в более широком наборе прикладных программ, например, чтобы найти список продуктов или услуг (антецедентов), консеквент которых - это товар, который вы хотите продвигать в этом летнем сезоне.

Таблица 92. Свойства узла carma.

Свойства узла carma	Значения	Описание свойства
inputs	[поле1 ... полеN]	Модели CARMA используют список входных полей, но не поля назначения. Поля веса и частоты не используются. Дополнительную информацию смотрите в разделе “Общие свойства узлов моделирования” на стр. 127.
id_field	поле	Поле, используемое в качестве поля ID для построения модели.
contiguous	логический	Используется для указания, последовательные ли ID содержатся в поле ID.
use_transactional_data	логический	
content_field	поле	
min_supp	число(процент)	Относится к поддержке правила, а не антецедента. Значение по умолчанию - 20%.
min_conf	число(процент)	Значение по умолчанию - 20%.
max_size	число	Значение по умолчанию - 10.
mode	Простые Эксперт	Значение по умолчанию - Simple.
exclude_multiple	логический	Исключает правила с несколькими консеквентами. Значение по умолчанию: Ложь .
use_pruning	логический	Значение по умолчанию: Ложь .
pruning_value	число	Значение по умолчанию - 500.
vary_support	логический	
estimated_transactions	целое	
rules_without_antecedents	логический	

Свойства узла cart



Узел дерева классификации и регрессии (Classification and Regression, C&R) генерирует дерево решений, позволяющее предсказывать или классифицировать будущие наблюдения. Этот метод использует рекурсивное разделение, чтобы расщепить обучающие записи на сегменты, на каждом шаге минимизируя неоднородность, причем узел дерева считается “чистым”, если все 100% наблюдений в узле попадают в конкретную категорию поля назначения. Входные поля и поля назначения могут быть из числового диапазона или категориальными (номинальными, порядковыми или флагами); все расщепления бинарны (только две подгруппы).

Таблица 93. Свойства узла cart.

Свойства узла cart	Значения	Описание свойства
target	поле	Моделям дерева C&R требуется одно поле назначения и одно или несколько входных полей. Может быть задано также поле частоты. Дополнительную информацию смотрите в разделе “Общие свойства узлов моделирования” на стр. 127.
continue_training_existing_model	логический	
objective	Стандартные Бустинг Бэггинг psm	psm используется для очень больших наборов данных и требует соединения с сервером.
model_output_type	Одиночный InteractiveBuilder	
use_tree_directives	логический	
tree_directives	строка	Задайте директивы роста дерева. Чтобы не использовать символов перехода на новую строку или двойных кавычек, директивы можно заключить в тройные знаки кавычек. Обратите внимание на то, что директивы могут быть очень чувствительны к небольшим изменениям данных или опций моделирования, и их нельзя обобщать на другие наборы данных.
use_max_depth	По умолчанию Пользовательские	
max_depth	целое	Максимальное количество уровней в дереве, от 0 до 1000. Используется только в том случае, если use_max_depth = Custom.
prune_tree	логический	Отсекать ветви, чтобы избежать переобучения.
use_std_err	логический	Использовать максимальную разницу в риске (в стандартных ошибках).
std_err_multiplier	число	Максимальная разность.
max_surrogates	число	Максимум суррогатов.
use_percentage	логический	
min_parent_records_pc	число	
min_child_records_pc	число	
min_parent_records_abs	число	
min_child_records_abs	число	

Таблица 93. Свойства узла cart (продолжение).

Свойства узла cart	Значения	Описание свойства
use_costs	логический	
costs	структурированный	Структурированное свойство.
priors	Данные Равенство Пользовательские	
custom_priors	структурированный	Структурированное свойство.
adjust_priors	логический	
trails	число	Число моделей компонентов для бустинга или бэггинга.
set_ensemble_method	Голосование HighestProbability HighestMeanProbability	Принятое по умолчанию правило объединения для категориальных целевых полей.
range_ensemble_method	Mean Медиана	Принятое по умолчанию правило объединения для непрерывных целевых полей.
large_boost	логический	Применить бустинг к очень большим наборам данных.
min_impurity	число	
impurity_measure	Gini Бинаризация Упорядоченная	
train_pct	число	Множество предотвращения переобучения.
set_random_seed	логический	Опция репликации результатов.
seed	число	
calculate_variable_importance	логический	
calculate_raw_propensities	логический	
calculate_adjusted_propensities	логический	
adjusted_propensity_partition	Критерий Проверка	

Свойства узла chaid



Узел CHAID генерирует деревья решений, используя статистику хи-квадрат для определения оптимальных расщеплений. В отличие от узлов дерева C&R и QUEST, CHAID может генерировать не только бинарные деревья, то есть у некоторых расщеплений может быть больше двух ветвей. Входные поля и поле назначения могут быть количественными (числовой диапазон) или категориальными. Исчерпывающий CHAID - это модификация метода CHAID, при котором продельвается более тщательная работа по изучению всех возможных расщеплений для каждого предиктора, но это требует больше времени для вычислений.

Таблица 94. Свойства узла chaid.

Свойства узла chaid	Значения	Описание свойства
target	поле	Моделям CHAID требуется одно поле назначения и одно или несколько входных полей. Может быть задано также поле частоты. Дополнительную информацию смотрите в разделе “Общие свойства узлов моделирования” на стр. 127.
continue_training_existing_model	логический	
objective	Стандартные Бустинг Бэггинг psm	psm используется для очень больших наборов данных и требует соединения с сервером.
model_output_type	Одиночный InteractiveBuilder	
use_tree_directives	логический	
tree_directives	строка	
method	Chaid ExhaustiveChaid	
use_max_depth	По умолчанию Пользовательские	
max_depth	целое	Максимальное количество уровней в дереве, от 0 до 1000. Используется только в том случае, если use_max_depth = Custom.
use_percentage	логический	
min_parent_records_pc	число	
min_child_records_pc	число	
min_parent_records_abs	число	
min_child_records_abs	число	
use_costs	логический	
costs	структурированный	Структурированное свойство.
trails	число	Число моделей компонентов для бустинга или бэггинга.
set_ensemble_method	Голосование HighestProbability HighestMeanProbability	Принятое по умолчанию правило объединения для категориальных целевых полей.
range_ensemble_method	Mean Медиана	Принятое по умолчанию правило объединения для непрерывных целевых полей.
large_boost	логический	Применить бустинг к очень большим наборам данных.
split_alpha	число	Уровень значимости для разбиения.
merge_alpha	число	Уровень значимости для слияния.
bonferroni_adjustment	логический	Скорректировать уровни значимости, используя метод Бонферрони.
split_merged_categories	логический	Допускать разбиение объединенных категорий.

Таблица 94. Свойства узла chaid (продолжение).

Свойства узла chaid	Значения	Описание свойства
chi_square	Пирсона LR	Используемый для вычисления статистики хи-квадрат метод: Пирсона или отношения правдоподобия
epsilon	число	Минимальное изменение ожидаемых частот в ячейках.
max_iterations	число	Максимум итераций до сходимости.
set_random_seed	целое	
seed	число	
calculate_variable_importance	логический	
calculate_raw_propensities	логический	
calculate_adjusted_propensities	логический	
adjusted_propensity_partition	Критерий Проверка	
maximum_number_of_models	целое	

Свойства узла coxreg



Узел регрессии Кокса позволяет построить модель дожития для данных времени-до-события в присутствии цензурируемых записей. Эта модель создает функцию дожития, которая предсказывает вероятность, что изучаемое событие произойдет в данное время (t) для данных значений входных переменных.

Таблица 95. Свойства узла coxreg.

Свойства узла coxreg	Значения	Описание свойства
survival_time	поле	Модели регрессии Кокса требуют одного поля, содержащего времена выживания.
target	поле	Моделям Кокса требуется одно поле назначения и одно или несколько входных полей. Дополнительную информацию смотрите в разделе “Общие свойства узлов моделирования” на стр. 127.
method	Ввод Пошаговый BackwardsStepwise	
groups	поле	
model_type	MainEffects Пользовательские	
custom_terms	["BP*Sex" "BP*Age"]	
mode	Эксперт Простые	
max_iterations	число	
p_converge	1,0E-4 1,0E-5 1,0E-6 1,0E-7 1,0E-8 0	

Таблица 95. Свойства узла *soxreg* (продолжение).

Свойства узла <i>soxreg</i>	Значения	Описание свойства
<i>p_converge</i>	1,0E-4 1,0E-5 1,0E-6 1,0E-7 1,0E-8 0	
<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 Вальда Условное	
<i>probability_entry</i>	<i>число</i>	
<i>probability_removal</i>	<i>число</i>	
<i>output_display</i>	EachStep LastStep	
<i>ci_enable</i>	<i>логический</i>	
<i>ci_value</i>	90 95 99	
<i>корреляция</i>	<i>логический</i>	
<i>display_baseline</i>	<i>логический</i>	
<i>survival</i>	<i>логический</i>	
<i>hazard</i>	<i>логический</i>	
<i>log_minus_log</i>	<i>логический</i>	
<i>one_minus_survival</i>	<i>логический</i>	
<i>separate_line</i>	<i>поле</i>	
<i>value</i>	<i>число или строка</i>	Если никакое значение для поля не задано, для него будет использоваться опция по умолчанию "Среднее".

Свойства узла *decisionlist*



Узел списка решений определяет подгруппы или сегменты, которые показывают более высокое или более низкое правдоподобие для данного бинарного результата по сравнению с полной совокупностью. Например, вы могли бы искать клиентов с низкой вероятностью оттока или с высокой вероятностью отклика на кампанию. Вы можете включить свои знания о бизнесе в модель, добавляя свои собственные пользовательские сегменты и параллельно просматривая альтернативные модели, чтобы сравнить результаты. Модели списка решений состоят из списка правил, в котором каждое правило имеет условие и следствие. Правила применяются по очереди, и первое подходящее правило определяет результат.

Таблица 96. Свойства узла decisionlist.

Свойства узла decisionlist	Значения	Описание свойства
target	поле	Модели списка решений используют одно поле назначения и одно или несколько входных полей. Может быть задано также поле частоты. Дополнительную информацию смотрите в разделе “Общие свойства узлов моделирования” на стр. 127.
model_output_type	Модель InteractiveBuilder	
search_direction	Вверх Вниз	Относится к нахождению сегментов; где Вверх - это эквивалент высокой вероятности, а Вниз - низкой.
target_value	строка	Если не задано, для флагов предполагается значение true.
max_rules	целое	Максимальное количество сегментов за исключением остатка.
min_group_size	целое	Минимальный размер сегмента.
min_group_size_pct	число	Минимальный размер сегмента, выраженный как процентная доля.
confidence_level	число	Минимальный порог, насколько входное поле должно повысить правдоподобие отклика (обеспечить подъем), чтобы был смысл добавить определение сегмента.
max_segments_per_rule	целое	
mode	Простые Эксперт	
bin_method	EqualWidth EqualCount	
bin_count	число	
max_models_per_cycle	целое	Ширина поиска для списков.
max_rules_per_cycle	целое	Ширина поиска для правил сегментов.
segment_growth	число	
include_missing	логический	
final_results_only	логический	
reuse_fields	логический	Позволяет повторное использование атрибутов (входных полей, появляющихся в правилах).
max_alternatives	целое	
calculate_raw_propensities	логический	
calculate_adjusted_propensities	логический	
adjusted_propensity_partition	Критерий Проверка	

Свойства узла discriminant



Дискриминантный анализ делает более строгие предположения, чем логистическая регрессия, но он может быть ценной альтернативой или дополнением к анализу логистической регрессии, когда эти предположения оказываются правильными.

Таблица 97. Свойства узла discriminant.

Свойства узла discriminant	Значения	Описание свойства
target	поле	Моделям дискриминанта требуется одно поле назначения и одно или несколько входных полей. Поля веса и частоты не используются. Дополнительную информацию смотрите в разделе “Общие свойства узлов моделирования” на стр. 127.
method	Ввод Пошаговый	
mode	Простые Эксперт	
prior_probabilities	AllEqual ComputeFromSizes	
covariance_matrix	WithinGroups SeparateGroups	
средние значения	логический	Опции статистики в диалоговом окне расширенного вывода.
univariate_anovas	логический	
box_m	логический	
within_group_covariance	логический	
within_groups_correlation	логический	
separate_groups_covariance	логический	
total_covariance	логический	
fishers	логический	
unstandardized	логический	
casewise_results	логический	Опции классификации в диалоговом окне расширенного вывода.
limit_to_first	число	Значение по умолчанию - 10.
summary_table	логический	
leave_one_classification	логический	
combined_groups	логический	
separate_groups_covariance	логический	Опция матриц Ковариации отдельно по группам .
territorial_map	логический	
combined_groups	логический	Опция графика Объединенные группы .
separate_groups	логический	Опция графика Отдельные группы .
summary_of_steps	логический	
F_pairwise	логический	

Таблица 97. Свойства узла discriminant (продолжение).

Свойства узла discriminant	Значения	Описание свойства
stepwise_method	WilksLambda UnexplainedVariance MahalanobisDistance SmallestF RaosV	
V_to_enter	число	
критерии	UseValue UseProbability	
F_value_entry	число	Значение по умолчанию - 3,84.
F_value_removal	число	Значение по умолчанию - 2,71.
probability_entry	число	Значение по умолчанию - 0,05.
probability_removal	число	Значение по умолчанию - 0,10.
calculate_variable_importance	логический	
calculate_raw_propensities	логический	
calculate_adjusted_propensities	логический	
adjusted_propensity_partition	Критерий Проверка	

Свойства узла factor



Узел PCA/фактора предоставляет мощные средства сокращения числа данных для уменьшения сложности ваших данных. Анализ главных компонент (principal components analysis, PCA) находит линейные комбинации входных полей, которыми главным образом определяются изменения в целом наборе полей, где компоненты ортогональны друг другу. Факторный анализ направлен на выявление скрытых факторов, объясняющих структуру корреляций в наборе наблюдаемых полей. Цель обоих подходов - найти небольшое количество производных полей, которые эффективно суммируют информацию исходного набора входных полей.

Таблица 98. Свойства узла factor.

Свойства узла factor	Значения	Описание свойства
inputs	[поле1 ... полеN]	Модели PCA/факторов используют список входных полей, но не поле назначения. Поля веса и частоты не используются. Дополнительную информацию смотрите в разделе "Общие свойства узлов моделирования" на стр. 127.
method	PC ULS GLS ML PAF Альфа Рисунок	
mode	Простые Эксперт	
max_iterations	число	
complete_records	логический	

Таблица 98. Свойства узла factor (продолжение).

Свойства узла factor	Значения	Описание свойства
матрица	Корреляция Ковариация	
extract_factors	ByEigenvalues ByFactors	
min_eigenvalue	число	
max_factor	число	
rotation	Нет Варимакс DirectOblimin Эквимакс Quartimax Промакс	
delta	число	При выборе DirectOblimin в качестве типа данных вращения, можно задать значение для delta. Если значение не задавать, будет использоваться значение delta по умолчанию.
каппа	число	Если в качестве типа данных вращения выбрать Promax, можно задать значение для каппа. Если значение не задавать, будет использоваться значение каппа по умолчанию.
sort_values	логический	
hide_values	логический	
hide_below	число	

Свойства узла featureselection



Узел выбора возможностей изучает входные поля на возможность удаления, основываясь на наборе критериев (таких как процентная доля пропущенных значений); затем этот узел ранжирует важность оставшихся полей по отношению к заданному полю назначения. Например, если у набора данных сотни потенциальных входных полей, какие из них потенциально наиболее полезны при моделировании исхода лечения пациента?

Таблица 99. Свойства узла featureselection.

Свойства узла featureselection	Значения	Описание свойства
target	поле	Модели выбора возможностей ранжируют предикторы относительно заданного назначения. Поля веса и частоты не используются. Дополнительную информацию смотрите в разделе “Общие свойства узлов моделирования” на стр. 127.

Таблица 99. Свойства узла *featureselection* (продолжение).

Свойства узла <i>featureselection</i>	Значения	Описание свойства
<code>screen_single_category</code>	<i>логический</i>	При значении <code>True</code> экранирует поля, у которых слишком много записей относительно общего количества записей попадает в одну категорию.
<code>max_single_category</code>	<i>число</i>	Задаёт порог, используемый при значении <code>screen_single_category</code> , равном <code>True</code> .
<code>screen_missing_values</code>	<i>логический</i>	При значении <code>True</code> экранирует поля, у которых слишком много пропущенных значений, выражается процентной долей от общего числа записей.
<code>max_missing_values</code>	<i>число</i>	
<code>screen_num_categories</code>	<i>логический</i>	При значении <code>True</code> экранирует поля, у которых слишком много категорий относительно общего числа записей.
<code>max_num_categories</code>	<i>число</i>	
<code>screen_std_dev</code>	<i>логический</i>	При значении <code>True</code> экранирует поля, для которых среднеквадратичные отклонения меньше или равны заданному минимуму.
<code>min_std_dev</code>	<i>число</i>	
<code>screen_coeff_of_var</code>	<i>логический</i>	При значении <code>True</code> экранирует поля, для которых коэффициент изменчивости меньше или равен заданному минимуму.
<code>min_coeff_of_var</code>	<i>число</i>	
критерии	Пирсона Вероятность CramersV Лямбда	При ранжировании категориальных предикторов по категориальным полям назначения задаёт показатель, на котором основывается значение важности.
<code>unimportant_below</code>	<i>число</i>	Задаёт пороговые значения <i>p</i> , используемые для ранжирования переменных как важных, пограничных или не важных. Принимает значения от 0,0 до 1,0.
<code>important_above</code>	<i>число</i>	Принимает значения от 0,0 до 1,0.
<code>unimportant_label</code>	<i>строка</i>	Задаёт метку для ранжирования как не важного.
<code>marginal_label</code>	<i>строка</i>	
<code>important_label</code>	<i>строка</i>	
<code>selection_mode</code>	ImportanceLevel ImportanceValue TopN	
<code>select_important</code>	<i>логический</i>	Когда для <code>selection_mode</code> задано значение <code>ImportanceLevel</code> , задаёт, выбрать ли важные поля.
<code>select_marginal</code>	<i>логический</i>	Когда для <code>selection_mode</code> задано значение <code>ImportanceLevel</code> , задаёт, выбрать ли пограничные поля.
<code>select_unimportant</code>	<i>логический</i>	Когда для <code>selection_mode</code> задано значение <code>ImportanceLevel</code> , задаёт, выбрать ли не важные поля.

Таблица 99. Свойства узла *featureselection* (продолжение).

Свойства узла <i>featureselection</i>	Значения	Описание свойства
<i>importance_value</i>	<i>число</i>	Когда для <i>selection_mode</i> задано значение <i>ImportanceValue</i> , задает значение отсечения для использования. Принимает значения от 0 до 100.
<i>top_n</i>	<i>целое</i>	Когда для <i>selection_mode</i> задано значение <i>TopN</i> , задает значение отсечения для использования. Принимает значения от 0 до 1000.

Свойства узла *genlin*



Обобщенная линейная модель расширяет общую линейную модель, так что зависимая переменная считается линейно связанной с факторами и ковариатами через заданную функцию связи. Более того, модель допускает наличие у зависимой переменной распределения, отличающегося от нормального. Она включает в себя функциональные возможности большого количества статистических моделей, в том числе линейной регрессии, логистической регрессии, логлинейных моделей для количества данных и интервал-цензурированных моделей выживания.

Таблица 100. Свойства узла *genlin*.

Свойства узла <i>genlin</i>	Значения	Описание свойства
<i>target</i>	<i>поле</i>	Обобщенным линейным моделям требуется одно поле назначения, которое должно быть номинальным или флаговым, и одно или несколько входных полей. Можно задать также поле веса. Дополнительную информацию смотрите в разделе “Общие свойства узлов моделирования” на стр. 127.
<i>use_weight</i>	<i>логический</i>	
<i>weight_field</i>	<i>поле</i>	Тип поля - только количественный.
<i>target_represents_trials</i>	<i>логический</i>	
<i>trials_type</i>	Переменная <i>FixedValue</i>	
<i>trials_field</i>	<i>поле</i>	Тип поля - количественный, флаговый или порядковый.
<i>trials_number</i>	<i>число</i>	Значение по умолчанию - 10.
<i>model_type</i>	<i>MainEffects</i> <i>MainAndAllTwoWayEffects</i>	
<i>offset_type</i>	Переменная <i>FixedValue</i>	
<i>offset_field</i>	<i>поле</i>	Тип поля - только количественный.
<i>offset_value</i>	<i>число</i>	Должно быть действительным числом.
<i>base_category</i>	Последнее Первое	
<i>include_intercept</i>	<i>логический</i>	
<i>mode</i>	Простые Эксперт	

Таблица 100. Свойства узла *genlin* (продолжение).

Свойства узла <i>genlin</i>	Значения	Описание свойства
распределение	BINOMIAL GAMMA IGAUSS NEGBIN NORMAL POISSON TWEEDIE MULTINOMIAL	IGAUSS: Обратный гауссиан. NEGBIN: Отрицательное биномиальное.
negbin_para_type	Задать Оценка	
negbin_parameter	<i>число</i>	Значение по умолчанию 1. Должно содержать неотрицательное действительное число.
tweedie_parameter	<i>число</i>	
link_function	IDENTITY CLOGLOG LOG LOGC LOGIT NEGBIN NLOGLOG ODDSPower PROBIT POWER CUMCAUCHIT CUMCLOGLOG CUMLOGIT CUMNLOGLOG CUMPROBIT	CLOGLOG: Дополнительное лог-лог. LOGC: Дополняющее лог. NEGBIN: Отрицательное биномиальное. NLOGLOG: Отрицательное лог-лог. CUMCAUCHIT: Кумулятивное Коши. CUMCLOGLOG: Кумулятивное дополняющее лог-лог. CUMLOGIT: Кумулятивное логит. CUMNLOGLOG: Кумулятивное отрицательное лог-лог. CUMPROBIT: Кумулятивное пробит.
Степень	<i>число</i>	Значение должно быть действительным ненулевым числом.
method	Гибридный Фишера NewtonRaphson	
max_fisher_iterations	<i>число</i>	Значение по умолчанию - 1; допустимы только положительные целые.
scale_method	MaxLikelihoodEstimate Отклонение PearsonChiSquare FixedValue	
scale_value	<i>число</i>	Значение по умолчанию - 1; должно быть больше 0.
covariance_matrix	ModelEstimator RobustEstimator	
max_iterations	<i>число</i>	Значение по умолчанию - 100; допустимы только неотрицательные целые.
max_step_halving	<i>число</i>	Значение по умолчанию - 5; допустимы только положительные целые.
check_separation	<i>логический</i>	
start_iteration	<i>число</i>	Значение по умолчанию - 20; допустимы только положительные целые.

Таблица 100. Свойства узла *genlin* (продолжение).

Свойства узла <i>genlin</i>	Значения	Описание свойства
<i>estimates_change</i>	<i>логический</i>	
<i>estimates_change_min</i>	<i>число</i>	Значение по умолчанию - 1E-006; допустимы только положительные числа.
<i>estimates_change_type</i>	Абсолютная Относительный	
<i>loglikelihood_change</i>	<i>логический</i>	
<i>loglikelihood_change_min</i>	<i>число</i>	Допускаются только положительные числа.
<i>loglikelihood_change_type</i>	Абсолютная Относительный	
<i>hessian_convergence</i>	<i>логический</i>	
<i>hessian_convergence_min</i>	<i>число</i>	Допускаются только положительные числа.
<i>hessian_convergence_type</i>	Абсолютная Относительный	
<i>case_summary</i>	<i>логический</i>	
<i>contrast_matrices</i>	<i>логический</i>	
<i>descriptive_statistics</i>	<i>логический</i>	
<i>estimable_functions</i>	<i>логический</i>	
<i>model_info</i>	<i>логический</i>	
<i>iteration_history</i>	<i>логический</i>	
<i>goodness_of_fit</i>	<i>логический</i>	
<i>print_interval</i>	<i>число</i>	Значение по умолчанию - 1; допустимы только положительные целые.
<i>model_summary</i>	<i>логический</i>	
<i>lagrange_multiplier</i>	<i>логический</i>	
<i>parameter_estimates</i>	<i>логический</i>	
<i>include_exponential</i>	<i>логический</i>	
<i>covariance_estimates</i>	<i>логический</i>	
<i>correlation_estimates</i>	<i>логический</i>	
<i>analysis_type</i>	TypeI TypeIII TypeIAndTypeIII	
<i>statistics</i>	Вальда LR	
<i>citype</i>	Вальда Профиль	
<i>tolerancelevel</i>	<i>число</i>	Значение по умолчанию - 0,0001.
<i>confidence_interval</i>	<i>число</i>	Значение по умолчанию - 95.
<i>loglikelihood_function</i>	Заполнено Ядро	

Таблица 100. Свойства узла *genlin* (продолжение).

Свойства узла <i>genlin</i>	Значения	Описание свойства
<code>singularity_tolerance</code>	1E-007 1E-008 1E-009 1E-010 1E-011 1E-012	
<code>value_order</code>	По возрастанию По убыванию DataOrder	
<code>calculate_variable_importance</code>	логический	
<code>calculate_raw_propensities</code>	логический	
<code>calculate_adjusted_propensities</code>	логический	
<code>adjusted_propensity_partition</code>	Критерий Проверка	

Свойства узла *glm*



Обобщенная линейная смешанная модель (generalized linear mixed model, GLMM) обобщает линейную модель таким образом, что у значений назначения может быть отличное от нормального распределение и оно будет линейно связано с факторами и ковариатами через задаваемую функцию связи, так что наблюдения могут быть скоррелированными. Обобщенные линейные смешанные модели включают широкий набор моделей, начиная от простой линейной регрессии и кончая сложными многоуровневыми моделями для не нормально распределенных данных с повторными измерениями.

Таблица 101. Свойства узла *glm*.

Свойства узла <i>glm</i>	Значения	Описание свойства
<code>residual_subject_spec</code>	структурированный	Сочетание значений заданных категориальных полей, уникальным образом определяющее субъекты в наборе данных
<code>repeated_measures</code>	структурированный	Поля, используемые для идентификации повторных наблюдений.
<code>residual_group_spec</code>	[поле1 ... полеN]	Поля, определяющие независимые наборы параметров ковариации повторяющихся эффектов.
<code>residual_covariance_type</code>	Диагональная AR1 ARMA11 COMPOUND_SYMMETRY IDENTITY TOEPLITZ UNSTRUCTURED VARIANCE_COMPONENTS	Задаёт ковариационную структуру для остатков.
<code>custom_target</code>	логический	Обозначает, использовать ли поле назначения, определенное на вышележащем узле (<code>false</code>), или пользовательское поле назначения, заданное в поле <code>target_field</code> (<code>true</code>).

Таблица 101. Свойства узла glmт (продолжение).

Свойства узла glmт	Значения	Описание свойства
target_field	поле	Поле для использования в качестве поля назначения, если значение custom_target - это true.
use_trials	логический	Обозначает, что будет использоваться, когда отклик назначения - это число произошедших в наборе испытаний событий, дополнительное поле или значение, задающее число испытаний. Значение по умолчанию - false.
use_field_or_value	Поле Значение	Обозначает, что используется для задания числа испытаний, поле (по умолчанию) или значение.
trials_field	поле	Поле, используемое для задания числа испытаний.
trials_value	целое	Значение, используемое для задания числа испытаний. Если задано, минимальное значение равно 1.
use_custom_target_reference	логический	Обозначает, будет ли использоваться для категориального поля пользовательская опорная категория. Значение по умолчанию - false.
target_reference_value	строка	Опорная категория для использования, если значение use_custom_target_reference равно true.
dist_link_combination	Номинальная Логит ГаммаLog БиномиальнЛогит ПуассонLog БиномиальнПробит ОтрицБиномLog БиномиальнLogC Пользовательские	Общие модели для распределения значений поля назначения. Выберите Custom, чтобы задать распределение из списка в поле target_distribution.
target_distribution	Нормальный Биномиальное Multinomial Гамма Обратная NegativeBinomial Poisson	Распределение значений для поля назначения, когда dist_link_combination равно Custom.
link_function_type	IDENTITY LOGC LOG CLOGLOG LOGIT NLOGLOG PROBIT POWER CAUCHIT	Функция связи для установления соотношений между значениями назначения и предикторами. Если target_distribution - Binomial, можно использовать любую из перечисленных функций связи. Если target_distribution - Multinomial, можно использовать CLOGLOG, CAUCHIT, LOGIT, NLOGLOG или PROBIT. Если target_distribution - не Binomial и не Multinomial, можно использовать IDENTITY, LOG или POWER.

Таблица 101. Свойства узла *glmm* (продолжение).

Свойства узла <i>glmm</i>	Значения	Описание свойства
<code>link_function_param</code>	<i>число</i>	Значение параметра функции связи для использования. Применимо только в том случае, если <code>normal_link_function</code> или <code>link_function_type</code> - это POWER.
<code>use_predefined_inputs</code>	<i>логический</i>	Обозначает, что представляют собой фиксированные поля эффектов - определяются ли они вышележащими полями как входные (<code>true</code>), или берутся из списка <code>fixed_effects_list</code> (<code>false</code>). Значение по умолчанию - <code>false</code> .
<code>fixed_effects_list</code>	<i>структурированный</i>	Если <code>use_predefined_inputs</code> - это <code>false</code> , задается, что входные поля будут использоваться как фиксированные поля эффектов.
<code>use_intercept</code>	<i>логический</i>	При значении <code>true</code> (по умолчанию) в модель включается свободный член.
<code>random_effects_list</code>	<i>структурированный</i>	Список полей для задания в качестве случайных эффектов.
<code>regression_weight_field</code>	<i>поле</i>	Поле для использования в качестве поля веса анализа.
<code>use_offset</code>	Нет <code>offset_value</code> <code>offset_field</code>	Обозначает, как задается смещение. Значение Нет означает, что смещение не используется.
<code>offset_value</code>	<i>число</i>	Значение для использования в качестве смещения, если для <code>use_offset</code> задано <code>offset_value</code> .
<code>offset_field</code>	<i>поле</i>	Поле, используемое для значения смещения, если для <code>use_offset</code> задано <code>offset_field</code> .
<code>target_category_order</code>	По возрастанию По убыванию Данные	Порядок сортировки для категориальных целевых переменных. Значение Данные задает использование порядка сортировки, найденного в данных. Значение по умолчанию - Ascending (по возрастанию).
<code>inputs_category_order</code>	По возрастанию По убыванию Данные	Порядок сортировки для категориальных предикторов. Значение Данные задает использование порядка сортировки, найденного в данных. Значение по умолчанию - Ascending (по возрастанию).
<code>max_iterations</code>	<i>целое</i>	Максимальное количество итераций, которые могут быть выполнены алгоритмом. Неотрицательное целое число; значение по умолчанию - 100.
<code>confidence_level</code>	<i>целое</i>	Доверительный уровень, используемый для вычисления оценок интервалов коэффициентов модели. Неотрицательное целое число; максимальное значение 100, значение по умолчанию 95.
<code>degrees_of_freedom_method</code>	Фиксированная Varied	Задаёт, как вычисляется число степеней свободы для критерия значимости.
<code>test_fixed_effects_coefficients</code>	Модель Робастная	Способ вычисления матрицы ковариации оценок параметров.

Таблица 101. Свойства узла *glm* (продолжение).

Свойства узла <i>glm</i>	Значения	Описание свойства
<code>use_p_converge</code>	<i>логическое</i>	Опция для сходимости параметра.
<code>p_converge</code>	<i>число</i>	Пробел или любое положительное значение.
<code>p_converge_type</code>	Абсолютная Относительная	
<code>use_l_converge</code>	<i>логическое</i>	Опция для сходимости логарифмического правдоподобия.
<code>l_converge</code>	<i>число</i>	Пробел или любое положительное значение.
<code>l_converge_type</code>	Абсолютная Относительная	
<code>use_h_converge</code>	<i>логическое</i>	Опция для сходимости гессiana.
<code>h_converge</code>	<i>число</i>	Пробел или любое положительное значение.
<code>h_converge_type</code>	Абсолютная Относительная	
<code>max_fisher_steps</code>	<i>целое</i>	
<code>singularity_tolerance</code>	<i>число</i>	
<code>use_model_name</code>	<i>логический</i>	Обозначает, как определять имя модели, задавать пользовательское имя (<i>true</i>) или использовать сгенерированное системой имя (<i>false</i>). Значение по умолчанию - <i>false</i> .
<code>model_name</code>	<i>строка</i>	Если значение поля <code>use_model_name</code> равно <i>true</i> , задает имя модели для использования.
<code>confidence</code>	<code>onProbability</code> <code>onIncrease</code>	Основание для вычисления доверительного значения оценки: максимальная предсказанная вероятность или разность между максимальной и второй по значению предсказанной вероятностью.
<code>score_category_probabilities</code>	<i>логический</i>	При значении <i>true</i> создает предсказанные вероятности для категориальных полей назначения. Значение по умолчанию - <i>false</i> .
<code>max_categories</code>	<i>целое</i>	Если значение <code>score_category_probabilities</code> - это <i>true</i> , задает максимальное число категорий для сохранения.
<code>score_propensity</code>	<i>логический</i>	При значении <i>true</i> создает оценки склонности для флаговых полей назначения, определяющие правдоподобие выходного значения "true" для поля.
<code>emean</code>	<i>structure</i>	Для каждого категориального поля из списка фиксированных эффектов задает, создавать ли оценочные маргинальные средние.
<code>covariance_list</code>	<i>structure</i>	Для каждого количественного поля из списка фиксированных эффектов задает, что использовать при вычислении оценочных маргинальных средних, среднее или пользовательское значение.
<code>mean_scale</code>	Исходное Преобразование	Задает, как вычислять оценочные маргинальные средние, на основе исходной шкалы назначения (по умолчанию) или преобразования функции связи.

Таблица 101. Свойства узла *glm* (продолжение).

Свойства узла <i>glm</i>	Значения	Описание свойства
comparison_adjustment_method	НЗР SEQBONFERRONI SEQSIDAK	Способ корректировки для использования при выполнении испытаний гипотез с несколькими контрастами.

Свойства узла *kmeans*



Узел *K*-средних кластеризует набор данных в отдельные группы (или кластеры). Этот метод определяет фиксированное количество кластеров, итерационно распределяет записи по кластерам и настраивает центры кластеров, пока дальнейшие уточнения более не улучшают модель. Вместо попытки предсказать выходное значение *k*-средние используют процесс, называемый неконтролируемым обучением, чтобы обнаружить структуры в наборе входных полей.

Таблица 102. Свойства узла *kmeans*.

Свойства узла <i>kmeans</i>	Значения	Описание свойства
inputs	[поле1 ... полеN]	Модели <i>k</i> -средних выполняют кластерный анализ для набора входных полей, но не используют поле назначения. Поля веса и частоты не используются. Дополнительную информацию смотрите в разделе “Общие свойства узлов моделирования” на стр. 127.
num_clusters	число	
gen_distance	логический	
cluster_label	Строка Число	
label_prefix	строка	
mode	Простые Эксперт	
stop_on	По умолчанию Пользовательские	
max_iterations	число	
tolerance	число	
encoding_value	число	
optimize	Скорость Память	Используется для определения, нужно ли при построении модели оптимизировать скорость или использование памяти.

Свойства узла *knn*



Узел *k* ближайших соседей (*k*-Nearest Neighbor, *KNN*) связывает новое наблюдение с категорией или значением *k* объектов, ближайших к нему в пространстве предикторов, где *k* - это целое число. Подобные наблюдения близки друг к другу, а непохожие наблюдения, наоборот, удалены друг от друга.

Таблица 103. Свойства узла knn.

Свойства узла knn	Значения	Описание свойства
analysis	PredictTarget IdentifyNeighbors	
objective	Баланс Скорость Точность Пользовательские	
normalize_ranges	логический	
use_case_labels	логический	Переключатель для включения следующей опции.
case_labels_field	поле	
identify_focal_cases	логический	Переключатель для включения следующей опции.
focal_cases_field	поле	
automatic_k_selection	логический	
fixed_k	целое	Включается только в том случае, если значение automatic_k_selectio - это False.
minimum_k	целое	Включается только в том случае, если значение automatic_k_selectio - это True.
maximum_k	целое	
distance_computation	Евклидова CityBlock	
weight_by_importance	логический	
range_predictions	Mean Медиана	
perform_feature_selection	логический	
forced_entry_inputs	[поле1 ... полеN]	
stop_on_error_ratio	логический	
number_to_select	целое	
minimum_change	число	
validation_fold_assign_by_field	логический	
number_of_folds	целое	Включается только в том случае, если значение validation_fold_assign_by_field - это False
set_random_seed	логический	
random_seed	число	
folds_field	поле	Включается только в том случае, если значение validation_fold_assign_by_field - это True
all_probabilities	логический	
save_distances	логический	
calculate_raw_propensities	логический	
calculate_adjusted_propensities	логический	
adjusted_propensity_partition	Критерий Проверка	

Свойства узла kohonen



Узел Коонена генерирует тип нейросети, которую можно использовать для кластеризации набора данных в отдельные группы. Когда сеть полностью обучена, похожие записи должны быть близко друг от друга на выходной карте, а отличающиеся записи должны быть сильно разделены. По количеству наблюдений, захваченных каждым нейроном в слепке модели, можно определить сильные нейроны. Это может дать представление об оправданном количестве кластеров.

Таблица 104. Свойства узла kohonen.

Свойства узла kohonen	Значения	Описание свойства
inputs	[поле1 ... полеN]	Модели Коонена используют список входных полей, но не поля назначения. Поля веса и частоты не используются. Дополнительную информацию смотрите в разделе “Общие свойства узлов моделирования” на стр. 127.
continue	логический	
show_feedback	логический	
stop_on	По умолчанию Время	
время	число	
optimize	Скорость Память	Используется для определения, нужно ли при построении модели оптимизировать скорость или использование памяти.
cluster_label	логический	
mode	Простые Эксперт	
ширина	число	
length	число	
decay_style	Линейная Экспоненциальное	
phase1_neighborhood	число	
phase1_eta	число	
phase1_cycles	число	
phase2_neighborhood	число	
phase2_eta	число	
phase2_cycles	число	

Свойства узла linear



Модели линейной регрессии предсказывают значения непрерывного целевого поля на основе линейных взаимосвязей между целевым полем и одним или несколькими предикторами.

Таблица 105. Свойства узла linear.

Свойства узла linear	Значения	Описание свойства
target	поле	Задаёт одно поле назначения.
inputs	[поле1 ... полеN]	Предикторные поля, используемые моделью.
continue_training_existing_model	логический	
objective	Стандартные Бэггинг Бустинг psm	psm используется для очень больших наборов данных и требует соединения с сервером.
use_auto_data_preparation	логический	
confidence_level	число	
model_selection	ForwardStepwise BestSubsets Нет	
criteria_forward_stepwise	AICC Fstatistics AdjustedRSquare ASE	
probability_entry	число	
probability_removal	число	
use_max_effects	логический	
max_effects	число	
use_max_steps	логический	
max_steps	число	
criteria_best_subsets	AICC AdjustedRSquare ASE	
combining_rule_continuous	Mean Медиана	
component_models_n	число	
use_random_seed	логический	
random_seed	число	
use_custom_model_name	логический	
custom_model_name	строка	
use_custom_name	логический	
custom_name	строка	
tooltip	строка	
keywords	строка	
annotation	строка	

Свойства узла logreg



Логистическая регрессия - это статистический метод для классификации записей на основании значений входных полей. Она аналогична линейной регрессии, но логистическая регрессия использует категориальные поля назначения вместо численных.

Таблица 106. Свойства узла logreg.

Свойства узла logreg	Значения	Описание свойства
target	поле	Моделям логистической регрессии требуется одно поле назначения и одно или несколько входных полей. Поля веса и частоты не используются. Дополнительную информацию смотрите в разделе “Общие свойства узлов моделирования” на стр. 127.
logistic_procedure	Биномиальное Multinomial	
include_constant	логический	
mode	Простые Эксперт	
method	Ввод Пошаговый Вперед Backwards BackwardsStepwise	
binomial_method	Ввод Вперед Backwards	
model_type	MainEffects FullFactorial Пользовательские	Когда в качестве типа модели задано FullFactorial, пошаговые способы не будут запущены, даже если они заданы. Вместо этого будет использоваться способ Ввод. Если для типа модели задано Пользовательский, но сами пользовательские поля не заданы, будет построена модель главных эффектов.
custom_terms	[{BP Sex}{BP}{Age}]	
multinomial_base_category	строка	Задаёт, как определяется опорная категория.
binomial_categorical_input	строка	
binomial_input_contrast	Индикатор Простые Разность Хелмерт Повторяемый Полиномиальный Отклонение	Ключевое свойство для категориального входного поля, которое задаёт, как определяется контраст.
binomial_input_category	Первое Последнее	Ключевое свойство для категориального входного поля, которое задаёт, как определяется контраст.

Таблица 106. Свойства узла logreg (продолжение).

Свойства узла logreg	Значения	Описание свойства
scale	Нет UserDefined Пирсона Отклонение	
scale_value	число	
all_probabilities	логический	
tolerance	1,0E-5 1,0E-6 1,0E-7 1,0E-8 1,0E-9 1,0E-10	
min_terms	число	
use_max_terms	логический	
max_terms	число	
entry_criterion	Оценка LR	
removal_criterion	LR Вальда	
probability_entry	число	
probability_removal	число	
binomial_probability_entry	число	
binomial_probability_removal	число	
requirements	HierarchyDiscrete HierarchyAll Containment Нет	
max_iterations	число	
max_steps	число	
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	число	
iteration_history	логический	
history_steps	число	
summary	логический	
likelihood_ratio	логический	
asymptotic_correlation	логический	
goodness_fit	логический	

Таблица 106. Свойства узла *logreg* (продолжение).

Свойства узла <i>logreg</i>	Значения	Описание свойства
<code>parameters</code>	логический	
<code>confidence_interval</code>	число	
<code>asymptotic_covariance</code>	логический	
<code>classification_table</code>	логический	
<code>stepwise_summary</code>	логический	
<code>info_criteria</code>	логический	
<code>monotonicity_measures</code>	логический	
<code>binomial_output_display</code>	at_each_step at_last_step	
<code>binomial_goodness_of_fit</code>	логический	
<code>binomial_parameters</code>	логический	
<code>binomial_iteration_history</code>	логический	
<code>binomial_classification_plots</code>	логический	
<code>binomial_ci_enable</code>	логический	
<code>binomial_ci</code>	число	
<code>binomial_residual</code>	выбросы все	
<code>binomial_residual_enable</code>	логический	
<code>binomial_outlier_threshold</code>	число	
<code>binomial_classification_cutoff</code>	число	
<code>binomial_removal_criterion</code>	LR Вальда Условное	
<code>calculate_variable_importance</code>	логический	
<code>calculate_raw_propensities</code>	логический	

Свойства узла *neuralnet*

Внимание: В этом выпуске доступна более новая версия узла моделирования нейронных сетей с расширенными возможностями, которая обсуждается в следующем разделе (*neuralnetwork*). Хотя вы по-прежнему можете построить и оценить модель в предыдущей версии, рекомендуется изменить сценарии для использования новой версии. Подробности предыдущей версии приведены здесь для справки.

Таблица 107. Свойства узла *neuralnet*.

Свойства узла <i>neuralnet</i>	Значения	Описание свойства
<code>targets</code>	[поле1 ... полеN]	Для узла нейронных сетей предполагаются одно или несколько полей назначения и одно или несколько входных полей. Поля веса и частоты игнорируются. Дополнительную информацию смотрите в разделе “Общие свойства узлов моделирования” на стр. 127.

Таблица 107. Свойства узла neuralnet (продолжение).

Свойства узла neuralnet	Значения	Описание свойства
method	Быстрый Динамический Несколько Сокращение ExhaustivePrune RBFN	
prevent_overtrain	логический	
train_pct	число	
set_random_seed	логический	
random_seed	число	
mode	Простые Эксперт	
stop_on	По умолчанию Точность Циклы Время	Режим остановки.
accuracy	число	Точность остановки.
cycles	число	Циклов для обучения.
время	число	Время для обучения (минуты).
continue	логический	
show_feedback	логический	
binary_encode	логический	
use_last_model	логический	
gen_logfile	логический	
logfile_name	строка	
alpha	число	
initial_eta	число	
high_eta	число	
low_eta	число	
eta_decay_cycles	число	
hid_layers	Один Два Три	
h1_units_one	число	
h1_units_two	число	
h1_units_three	число	
persistence	число	
m_topologies	строка	
m_non_pyramids	логический	
m_persistence	число	
p_hid_layers	Один Два Три	
p_h1_units_one	число	

Таблица 107. Свойства узла *neuralnet* (продолжение).

Свойства узла <i>neuralnet</i>	Значения	Описание свойства
<code>p_hl_units_two</code>	<i>число</i>	
<code>p_hl_units_three</code>	<i>число</i>	
<code>p_persistence</code>	<i>число</i>	
<code>p_hid_rate</code>	<i>число</i>	
<code>p_hid_pers</code>	<i>число</i>	
<code>p_inp_rate</code>	<i>число</i>	
<code>p_inp_pers</code>	<i>число</i>	
<code>p_overall_pers</code>	<i>число</i>	
<code>r_persistence</code>	<i>число</i>	
<code>r_num_clusters</code>	<i>число</i>	
<code>r_eta_auto</code>	<i>логический</i>	
<code>r_alpha</code>	<i>число</i>	
<code>r_eta</code>	<i>число</i>	
<code>optimize</code>	Скорость Память	Используется для определения, нужно ли при построении модели оптимизировать скорость или использование памяти.
<code>calculate_variable_importance</code>	<i>логический</i>	Комментарий: Свойство <code>sensitivity_analysis</code> , использованное в предыдущих выпусках, объявлено устаревшим и заменяется этим свойством. Старое свойство еще поддерживается, но рекомендуется использовать <code>calculate_variable_importance</code> .
<code>calculate_raw_propensities</code>	<i>логический</i>	
<code>calculate_adjusted_propensities</code>	<i>логический</i>	
<code>adjusted_propensity_partition</code>	Критерий Проверка	

Свойства узла *neuralnetwork*

Узел нейросетей использует упрощенную модель обработки информации человеческим мозгом. Нейросети работают, обчитывая большое количество связанных между собой обрабатываемых элементов, которые представляют абстрактную версию нейронов. Нейросети - это мощные средства оценки общих функциональных зависимостей, требующие минимальных знаний статистики и математики для их обучения и применения.

Таблица 108. Свойства узла *neuralnetwork*.

Свойства узла <i>neuralnetwork</i>	Значения	Описание свойства
<code>targets</code>	<i>[поле1 ... полеN]</i>	Задаёт поля назначения.
<code>inputs</code>	<i>[поле1 ... полеN]</i>	Предикторные поля, используемые моделью.
<code>splits</code>	<i>[поле1 ... полеN]</i>	Задаёт поле или поля для использования для моделирования разбиения.

Таблица 108. Свойства узла neuralnetwork (продолжение).

Свойства узла neuralnetwork	Значения	Описание свойства
use_partition	логический	Если определено поле раздела, эта опция обеспечивает, что для построения модели используются только данные из раздела обучения.
continue	логический	Продолжить обучение существующей модели.
objective	Стандартные Бэггинг Бустинг psm	psm используется для очень больших наборов данных и требует соединения с сервером.
method	MultilayerPerceptron RadialBasisFunction	
use_custom_layers	логический	
first_layer_units	число	
second_layer_units	число	
use_max_time	логический	
max_time	число	
use_max_cycles	логический	
max_cycles	число	
use_min_accuracy	логический	
min_accuracy	число	
combining_rule_categorical	Голосование HighestProbability HighestMeanProbability	
combining_rule_continuous	Mean Медиана	
component_models_n	число	
overfit_prevention_pct	число	
use_random_seed	логический	
random_seed	число	
missing_values	listwiseDeletion missingValueImputation	
use_custom_model_name	логический	
custom_model_name	строка	
confidence	onProbability onIncrease	
score_category_probabilities	логический	
max_categories	число	
score_propensity	логический	
use_custom_name	логический	
custom_name	строка	
tooltip	строка	

Таблица 108. Свойства узла *neuralnetwork* (продолжение).

Свойства узла <i>neuralnetwork</i>	Значения	Описание свойства
keywords	строка	
annotation	строка	

Свойства узла *quest*



Узел QUEST предоставляет метод бинарной классификации для построения деревьев решений, разработанный для уменьшения времени обработки, требуемого для анализа больших деревьев C&R, при одновременном подавлении обнаруженного в способах деревьев классификации предпочтения входных полей, допускающих больше расщеплений. Входные поля могут быть в числовом диапазоне (количественными), но поле назначения должно быть категориальным. Все расщепления бинарные.

Таблица 109. Свойства узла *quest*.

Свойства узла <i>quest</i>	Значения	Описание свойства
target	поле	Моделям QUEST требуется одно поле назначения и одно или несколько входных полей. Может быть задано также поле частоты. Дополнительную информацию смотрите в разделе “Общие свойства узлов моделирования” на стр. 127.
continue_training_existing_model	логический	
objective	Стандартные Бустинг Бэггинг psm	psm используется для очень больших наборов данных и требует соединения с сервером.
model_output_type	Одиночный InteractiveBuilder	
use_tree_directives	логический	
tree_directives	строка	
use_max_depth	По умолчанию Пользовательские	
max_depth	целое	Максимальное количество уровней в дереве, от 0 до 1000. Используется только в том случае, если use_max_depth = Custom.
prune_tree	логический	Отсекать ветви, чтобы избежать переобучения.
use_std_err	логический	Использовать максимальную разницу в риске (в стандартных ошибках).
std_err_multiplier	число	Максимальная разность.
max_surrogates	число	Максимум суррогатов.
use_percentage	логический	
min_parent_records_pc	число	
min_child_records_pc	число	
min_parent_records_abs	число	
min_child_records_abs	число	
use_costs	логический	

Таблица 109. Свойства узла *quest* (продолжение).

Свойства узла <i>quest</i>	Значения	Описание свойства
<i>costs</i>	<i>структурированный</i>	Структурированное свойство.
<i>priors</i>	Данные Равенство Пользовательские	
<i>custom_priors</i>	<i>структурированный</i>	Структурированное свойство.
<i>adjust_priors</i>	<i>логический</i>	
<i>trails</i>	<i>число</i>	Число моделей компонентов для бустинга или бэггинга.
<i>set_ensemble_method</i>	Голосование HighestProbability HighestMeanProbability	Принятое по умолчанию правило объединения для категориальных целевых полей.
<i>range_ensemble_method</i>	Mean Медиана	Принятое по умолчанию правило объединения для непрерывных целевых полей.
<i>large_boost</i>	<i>логический</i>	Применить бустинг к очень большим наборам данных.
<i>split_alpha</i>	<i>число</i>	Уровень значимости для разбиения.
<i>train_pct</i>	<i>число</i>	Множество предотвращения сверхобучения.
<i>set_random_seed</i>	<i>логический</i>	Опция репликации результатов.
<i>seed</i>	<i>число</i>	
<i>calculate_variable_importance</i>	<i>логический</i>	
<i>calculate_raw_propensities</i>	<i>логический</i>	
<i>calculate_adjusted_propensities</i>	<i>логический</i>	
<i>adjusted_propensity_partition</i>	Критерий Проверка	

Свойства узла *regression*



Линейная регрессия - это общепринятый статистический метод обработки данных и вычисления предсказаний при подгонке прямой линии или плоскости, минимизирующих разности между предсказанными и фактическими выходными значениями.

Примечание: В следующем выпуске узел регрессии будет заменен узлом линейных моделей. Рекомендуется для линейной регрессии с этого момента перейти на использование узла *Линейные модели*.

Таблица 110. Свойства узла *regression*.

Свойства узла <i>regression</i>	Значения	Описание свойства
<i>target</i>	<i>поле</i>	Моделям регрессии требуется одно поле назначения и одно или несколько входных полей. Можно задать также поле веса. Дополнительную информацию смотрите в разделе “Общие свойства узлов моделирования” на стр. 127.

Таблица 110. Свойства узла regression (продолжение).

Свойства узла regression	Значения	Описание свойства
method	Ввод Пошаговый Backwards Вперед	
include_constant	логический	
use_weight	логический	
weight_field	поле	
mode	Простые Эксперт	
complete_records	логический	
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	Используйте двойные кавычки.
stepping_method	useP useF	useP : использовать вероятность F useF: использовать значение F
probability_entry	число	
probability_removal	число	
F_value_entry	число	
F_value_removal	число	
selection_criteria	логический	
confidence_interval	логический	
covariance_matrix	логический	
collinearity_diagnostics	логический	
regression_coefficients	логический	
exclude_fields	логический	
durbin_watson	логический	
model_fit	логический	
r_squared_change	логический	
p_correlations	логический	
описательные статистики	логический	
calculate_variable_importance	логический	

Свойства узла sequence



Узел последовательности обнаруживает правила связывания для последовательных или зависящих от времени данных. Последовательность - это список наборов элементов с тенденцией появления в предсказуемом порядке. Например, покупатель, который приобрел лезвия и лосьон после бритья, с большой вероятностью в следующий раз купит крем для бритья. Узел последовательности основан на алгоритме правил связывания CARMA, использующем эффективный двухпроходный способ обнаружения последовательностей.

Таблица 111. Свойства узла sequence.

Свойства узла sequence	Значения	Описание свойства
id_field	поле	Для создания модели Последовательности необходимо определить поле ID, дополнительное поле времени и одно или несколько полей содержимого. Поля веса и частоты не используются. Дополнительную информацию смотрите в разделе “Общие свойства узлов моделирования” на стр. 127.
time_field	поле	
use_time_field	логический	
content_fields	[поле1 ... полеn]	
contiguous	логический	
min_supp	число	
min_conf	число	
max_size	число	
max_predictions	число	
mode	Простые Эксперт	
use_max_duration	логический	
max_duration	число	
use_gaps	логический	
min_item_gap	число	
max_item_gap	число	
use_pruning	логический	
pruning_value	число	
set_mem_sequences	логический	
mem_sequences	целое	

Свойства узла slrm



Узел Самообучаемая модель откликов (Self-Learning Response Model, SLRM) позволяет построить модель, в которой одно новое наблюдение или всего несколько наблюдений могут быть использованы для повторной оценки модели без необходимости повторного обучения модели с использованием всех данных.

Таблица 112. Свойства узла *slrm*.

Свойства узла <i>slrm</i>	Значения	Описание свойства
target	поле	Поле назначения должно быть номинальным или флаговым. Может быть задано также поле частоты. Дополнительную информацию смотрите в разделе “Общие свойства узлов моделирования” на стр. 127.
target_response	поле	Тип должен быть флаговым.
continue_training_existing_model	логический	
target_field_values	логический	Использовать все: использовать все значения источника. Задать: выбрать нужные значения.
target_field_values_specify	[поле1 ... полеN]	
include_model_assessment	логический	
model_assessment_random_seed	число	Должно быть действительным числом.
model_assessment_sample_size	число	Должно быть действительным числом.
model_assessment_iterations	число	Число итераций.
display_model_evaluation	логический	
max_predictions	число	
randomization	число	
scoring_random_seed	число	
sort	По возрастанию По убыванию	Задаёт, в каком порядке будут показываться предложения, начиная с максимальной или минимальной оценки.
model_reliability	логический	
calculate_variable_importance	логический	

Свойства узла *statisticsmodel*



Узел Статистическая модель позволяет проанализировать свои данные и работать с ними, запустив процедуры IBM SPSS Statistics, создающие PMML. Этому узлу требуется лицензированная копия IBM SPSS Statistics.

Свойства этого узла описаны в разделе “Свойства узла *statisticsmodel*” на стр. 238.

Свойства узла *svm*



Узел механизма опорных векторов (Support Vector Machine, SVM) позволяет классифицировать данные по одной или двум группам без переобучения. SVM хорошо работает с широкими наборами данных, в частности, в случае очень большого числа входных полей.

Таблица 113. Свойства узла svm.

Свойства узла svm	Значения	Описание свойства
all_probabilities	логический	
stopping_criteria	1,0E-1 1,0E-2 1,0E-3 (по умолчанию) 1,0E-4 1,0E-5 1,0E-6	Определяет, когда остановить алгоритм оптимизации.
regularization	число	Известно также как C-параметр.
precision	число	Используется только в том случае, если уровень измерения поля назначения - Количественный.
kernel	RBF(default) Полиномиальный Сигмоид Линейная	Тип функции ядра, используемой для преобразования.
rbf_gamma	число	Используется только в том случае, если kernel - это RBF.
gamma	число	Используется только в том случае, если kernel - это Polynomial или Sigmoid.
bias	число	
degree	число	Используется только в том случае, если kernel - это Polynomial.
calculate_variable_importance	логический	
calculate_raw_propensities	логический	
calculate_adjusted_propensities	логический	
adjusted_propensity_partition	Критерий Проверка	

Свойства узла timeseries



Узел временных рядов оценивает экспоненциальное сглаживание, а также одномерные и многомерные модели авторегрессии и проинтегрированного скользящего среднего (Autoregressive Integrated Moving Average, ARIMA) для временных рядов и создает прогнозы будущего выполнения. Предшественником узла временных рядов всегда должен быть узел Интервалы времени.

Таблица 114. Свойства узла timeseries.

Свойства узла timeseries	Значения	Описание свойства
targets	поле	Узел временных рядов прогнозирует значения одного или нескольких полей назначения, используя в качестве предикторов одно или несколько входных полей. Поля веса и частоты не используются. Дополнительную информацию смотрите в разделе “Общие свойства узлов моделирования” на стр. 127.
continue	логический	
method	ExpertModeler Exsmooth Arima Reuse	
expert_modeler_method	логический	
consider_seasonal	логический	
detect_outliers	логический	
expert_outlier_additive	логический	
expert_outlier_level_shift	логический	
expert_outlier_innovational	логический	
expert_outlier_level_shift	логический	
expert_outlier_transient	логический	
expert_outlier_seasonal_additive	логический	
expert_outlier_local_trend	логический	
expert_outlier_additive_patch	логический	
exsmooth_model_type	Простые HoltLinearTrend BrownsLinearTrend DampedTrend SimpleSeasonal WintersAdditive WintersMultiplicative	
exsmooth_transformation_type	Нет SquareRoot NaturalLog	
arima_p	целое	
arima_d	целое	
arima_q	целое	
arima_sp	целое	
arima_sd	целое	
arima_sq	целое	
arima_transformation_type	Нет SquareRoot NaturalLog	

Таблица 114. Свойства узла *timeseries* (продолжение).

Свойства узла <i>timeseries</i>	Значения	Описание свойства
<i>arma_include_constant</i>	<i>логический</i>	
<i>tf_arma_p.fieldname</i>	<i>целое</i>	Для передаточных функций.
<i>tf_arma_d.fieldname</i>	<i>целое</i>	Для передаточных функций.
<i>tf_arma_q.fieldname</i>	<i>целое</i>	Для передаточных функций.
<i>tf_arma_sp.fieldname</i>	<i>целое</i>	Для передаточных функций.
<i>tf_arma_sd.fieldname</i>	<i>целое</i>	Для передаточных функций.
<i>tf_arma_sq. имя_поля</i>	<i>целое</i>	Для передаточных функций.
<i>tf_arma_delay. имя_поля</i>	<i>целое</i>	Для передаточных функций.
<i>tf_arma_transformation_type. имя_поля</i>	Нет SquareRoot NaturalLog	Для передаточных функций.
<i>arma_detect_outlier_mode</i>	Нет Автоматически	
<i>arma_outlier_additive</i>	<i>логический</i>	
<i>arma_outlier_level_shift</i>	<i>логический</i>	
<i>arma_outlier_innovational</i>	<i>логический</i>	
<i>arma_outlier_transient</i>	<i>логический</i>	
<i>arma_outlier_seasonal_additive</i>	<i>логический</i>	
<i>arma_outlier_local_trend</i>	<i>логический</i>	
<i>arma_outlier_additive_patch</i>	<i>логический</i>	
<i>conf_limit_pct</i>	<i>real</i>	
<i>max_lags</i>	<i>целое</i>	
события	<i>fields</i>	
<i>scoring_model_only</i>	<i>логический</i>	Использовать для моделей с очень большим числом временных рядов (десятки тысяч).

Свойства узла *twostep*



Узел Двухшаговый использует метод двухшаговой кластеризации. На первом шаге проводится первый проход по данным, при котором необработанные входные данные сжимаются в управляемый набор подкластеров. На втором шаге используется способ иерархической кластеризации для все большего слияния подкластеров в крупные и еще более крупные кластеры. У двухшагового метода есть преимущество автоматической оценки оптимального числа кластеров для обучающих данных. Он может эффективно обрабатывать поля смешанных типов и большие наборы данных.

Таблица 115. Свойства узла *twostep*.

Свойства узла <i>twostep</i>	Значения	Описание свойства
<code>inputs</code>	[поле1 ... полеN]	Двухшаговые модели используют список входных полей, но не поле назначения. Поля веса и частоты не распознаются. Дополнительную информацию смотрите в разделе “Общие свойства узлов моделирования” на стр. 127.
стандартизация	<i>логический</i>	
<code>exclude_outliers</code>	<i>логический</i>	
процент	<i>число</i>	
<code>cluster_num_auto</code>	<i>логический</i>	
<code>min_num_clusters</code>	<i>число</i>	
<code>max_num_clusters</code>	<i>число</i>	
<code>num_clusters</code>	<i>число</i>	
<code>cluster_label</code>	Строка Число	
<code>label_prefix</code>	<i>строка</i>	
<code>distance_measure</code>	Евклидова Loglikelihood	
<code>clustering_criterion</code>	AIC BIC	

Глава 14. Свойства узла слепков моделей

У узлов слепков моделей те же общие свойства, что и у других узлов. Дополнительную информацию смотрите в разделе “Общие свойства узлов” на стр. 56.

Свойства узла `applyanomalydetection`

Узлы моделирования обнаружения аномалий можно использовать для генерирования слепка модели обнаружения аномалий. Имя сценария этого слепка модели - `applyanomalydetection`. Более подробную информацию о сценарии самого узла моделирования смотрите в разделе “Свойства узла `anomalydetection`” на стр. 128.

Таблица 116. Свойства узла `applyanomalydetection`.

Свойства узла <code>applyanomalydetection</code>	Значения	Описание свойства
<code>anomaly_score_method</code>	FlagAndScore FlagOnly ScoreOnly	Определяет, какие выходные значения создаются для скоринга.
<code>num_fields</code>	<i>целое</i>	Поля для отчета.
<code>discard_records</code>	<i>логический</i>	Указывает, отбрасываются или нет записи из выходных данных.
<code>discard_anomalous_records</code>	<i>логический</i>	Индикатор, какие записи отбрасывать - аномальные или <i>не</i> аномальные. Значение по умолчанию <code>off</code> соответствует отбрасыванию <i>не</i> аномальных полей. В противном случае, если задано <code>on</code> , будут отброшены аномальные записи. Это свойство включается только в том случае, если включено свойство <code>discard_records</code> .

Свойства узла `applyapriori`

Узлы моделирования Априори можно использовать для генерирования слепка априорной модели. Имя сценария этого слепка модели - `applyapriori`. Более подробную информацию о сценарии самого узла моделирования смотрите в разделе “Свойства узла `apriori`” на стр. 129.

Таблица 117. Свойства узла `applyapriori`.

Свойства узла <code>applyapriori</code>	Значения	Описание свойства
<code>max_predictions</code>	<i>количество (целое число)</i>	
<code>ignore_unmatched</code>	<i>логический</i>	
<code>allow_repeats</code>	<i>логический</i>	
<code>check_basket</code>	NoPredictions Predictions NoCheck	
<code>criterion</code>	Показатель доверия Поддержка RuleSupport Рост Deployability	

Свойства узла applyautoclassifier

Узлы моделирования автоклассификации можно использовать для генерирования слепка модели автоклассификации. Имя сценария этого слепка модели - *applyautoclassifier*. Более подробную информацию о сценарии самого узла моделирования смотрите в разделе “Свойства узла autoclassifier” на стр. 130.

Таблица 118. Свойства узла *applyautoclassifier*.

Свойства узла applyautoclassifier	Значения	Описание свойства
flag_ensemble_method	Голосование ConfidenceWeightedVoting RawPropensityWeightedVoting HighestConfidence AverageRawPropensity	Задает способ, используемый для определения оценки ансамбля. Этот параметр применим только в том случае, если выбранное поле назначения флаговое.
flag_voting_tie_selection	Переменный HighestConfidence RawPropensity	Если выбран способ голосования, задает, как разрешаются связи. Этот параметр применим только в том случае, если выбранное поле назначения флаговое.
set_ensemble_method	Голосование ConfidenceWeightedVoting HighestConfidence	Задает способ, используемый для определения оценки ансамбля. Этот параметр применим только в том случае, если выбранное поле назначения - это поле набора.
set_voting_tie_selection	Переменный HighestConfidence	Если выбран способ голосования, задает, как разрешаются связи. Этот параметр применим только в том случае, если выбранное поле назначения номинальное.

Свойства узла applyautocluster

Узлы моделирования автокластеризации можно использовать для генерирования слепка модели автокластеризации. Имя сценария этого слепка модели - *applyautocluster*. Других свойств для этого слепка модели нет. Более подробную информацию о сценарии самого узла моделирования смотрите в разделе “Свойства узла autocluster” на стр. 132.

Свойства узла applyautonumeric

Узлы моделирования автонумерации можно использовать для генерирования слепка модели автонумерации. Имя сценария этого слепка модели - *applyautonumeric*. Более подробную информацию о сценарии самого узла моделирования смотрите в разделе “Свойства узла autonumeric” на стр. 133.

Таблица 119. Свойства узла *applyautonumeric*.

Свойства узла applyautonumeric	Значения	Описание свойства
calculate_standard_error	логический	

Свойства узла applybayesnet

Узлы моделирования Байесовской сети можно использовать для генерирования слепка модели Байесовской сети. Имя сценария этого слепка модели - *applybayesnet*. Более подробную информацию о сценарии самого узла моделирования смотрите в разделе “Свойства узла bayesnet” на стр. 134.

Таблица 120. Свойства узла applybayesnet.

Свойства узла applybayesnet	Значения	Описание свойства
all_probabilities	логический	
raw_propensity	логический	
adjusted_propensity	логический	
calculate_raw_propensities	логический	
calculate_adjusted_propensities	логический	

Свойства узла applyc50

Узлы моделирования C5.0 можно использовать для генерирования слепка модели C5.0. Имя сценария этого слепка модели - *applyc50*. Более подробную информацию о сценарии самого узла моделирования смотрите в разделе “Свойства узла c50” на стр. 136.

Таблица 121. Свойства узла applyc50.

Свойства узла applyc50	Значения	Описание свойства
sql_generate	Никогда NoMissingValues	Используется для задания опций генерирования SQL во время выполнения набора правил.
calculate_conf	логический	Доступно, когда включено генерирование SQL; это свойство включает в себя вычисления доверительных показателей в сгенерированном дереве.
calculate_raw_propensities	логический	
calculate_adjusted_propensities	логический	

Свойства узла applycarma

Узлы моделирования CARMA можно использовать для генерирования слепка модели CARMA. Имя сценария этого слепка модели - *applycarma*. Других свойств для этого слепка модели нет. Более подробную информацию о сценарии самого узла моделирования смотрите в разделе “Свойства узла carma” на стр. 137.

Свойства узла applycart

Узлы моделирования дерева C&R можно использовать для генерирования слепка модели дерева C&R. Имя сценария этого слепка модели - *applycart*. Более подробную информацию о сценарии самого узла моделирования смотрите в разделе “Свойства узла cart” на стр. 138.

Таблица 122. Свойства узла applycart.

Свойства узла applycart	Значения	Описание свойства
sql_generate	Никогда MissingValues NoMissingValues	Используется для задания опций генерирования SQL во время выполнения набора правил.

Таблица 122. Свойства узла *applycart* (продолжение).

Свойства узла <i>applycart</i>	Значения	Описание свойства
calculate_conf	логический	Доступно, когда включено генерирование SQL; это свойство включает в себя вычисления доверительных показателей в сгенерированном дереве.
display_rule_id	логический	Добавляет поле в выводе данных скоринга, обозначающее ID для конечного узла, которому назначена каждая запись.
calculate_raw_propensities	логический	
calculate_adjusted_propensities	логический	

Свойства узла *applychaid*

Узлы моделирования CHAID можно использовать для генерирования слепка модели CHAID. Имя сценария этого слепка модели - *applychaid*. Более подробную информацию о сценарии самого узла моделирования смотрите в разделе “Свойства узла *chaid*” на стр. 139.

Таблица 123. Свойства узла *applychaid*.

Свойства узла <i>applychaid</i>	Значения	Описание свойства
sql_generate	Никогда MissingValues	
calculate_conf	логический	
display_rule_id	логический	Добавляет поле в выводе данных скоринга, обозначающее ID для конечного узла, которому назначена каждая запись.
calculate_raw_propensities	логический	
calculate_adjusted_propensities	логический	

Свойства узла *applycoxreg*

Узлы моделирования Кокса можно использовать для генерации слепка модели Кокса. Имя сценария этого слепка модели - *applycoxreg*. Более подробную информацию о сценарии самого узла моделирования смотрите в разделе “Свойства узла *coxreg*” на стр. 141.

Таблица 124. Свойства узла *applycoxreg*.

Свойства узла <i>applycoxreg</i>	Значения	Описание свойства
future_time_as	Интервалы Поля	
time_interval	число	
num_future_times	целое	
time_field	поле	
past_survival_time	поле	
all_probabilities	логический	
cumulative_hazard	логический	

Свойства узла applydecisionlist

Узлы моделирования списка решений можно использовать для генерирования слепка модели списка решений. Имя сценария этого слепка модели - *applydecisionlist*. Более подробную информацию о сценарии самого узла моделирования смотрите в разделе “Свойства узла decisionlist” на стр. 142.

Таблица 125. Свойства узла applydecisionlist.

Свойства узла applydecisionlist	Значения	Описание свойства
enable_sql_generation	логический	При значении true IBM SPSS Modeler будет стараться продвинуть модель списка решений обратно в SQL.
calculate_raw_propensities	логический	
calculate_adjusted_propensities	логический	

Свойства узла applydiscriminant

Узлы моделирования дискриминанта можно использовать для генерирования слепка модели дискриминанта. Имя сценария этого слепка модели - *applydiscriminant*. Более подробную информацию о сценарии самого узла моделирования смотрите в разделе “Свойства узла discriminant” на стр. 144.

Таблица 126. Свойства узла applydiscriminant.

Свойства узла applydiscriminant	Значения	Описание свойства
calculate_raw_propensities	логический	
calculate_adjusted_propensities	логический	

Свойства узла applyfactor

Узлы моделирования PCA/факторов можно использовать для генерирования слепка модели PCA/факторов. Имя сценария этого слепка модели - *applyfactor*. Других свойств для этого слепка модели нет. Более подробную информацию о сценарии самого узла моделирования смотрите в разделе “Свойства узла factor” на стр. 145.

Свойства узла applyfeatureselection

Узлы моделирования выбора возможностей можно использовать для генерирования слепка модели выбора возможностей. Имя сценария этого слепка модели - *applyfeatureselection*. Более подробную информацию о сценарии самого узла моделирования смотрите в разделе “Свойства узла featureselection” на стр. 146.

Таблица 127. Свойства узла applyfeatureselection.

Свойства узла applyfeatureselection	Значения	Описание свойства
selected_ranked_fields		Задаёт, какие ранжированные поля отмечаются для выбора в браузере моделей.
selected_screened_fields		Задаёт, какие экранированные поля отмечены для выбора в браузере моделей.

Свойства узла `applygeneralizedlinear`

Узлы моделирования обобщенной линейной регрессии (`genlin`) можно использовать для генерирования слепка обобщенной линейной модели. Имя сценария этого слепка модели - `applygeneralizedlinear`. Более подробную информацию о сценарии самого узла моделирования смотрите в разделе “Свойства узла `genlin`” на стр. 148.

Таблица 128. Свойства узла `applygeneralizedlinear`.

Свойства узла <code>applygeneralizedlinear</code>	Значения	Описание свойства
<code>calculate_raw_propensities</code>	<i>логический</i>	
<code>calculate_adjusted_propensities</code>	<i>логический</i>	

Свойства узла `applyglm`

Узлы моделирования GLMM можно использовать для генерирования слепка модели GLMM. Имя сценария этого слепка модели - `applyglm`. Более подробную информацию о сценарии самого узла моделирования смотрите в разделе “Свойства узла `glm`” на стр. 151.

Таблица 129. Свойства узла `applyglm`.

Свойства узла <code>applyglm</code>	Значения	Описание свойства
<code>confidence</code>	<code>onProbability</code> <code>onIncrease</code>	Основание для вычисления доверительного значения оценки: максимальная предсказанная вероятность или разность между максимальной и второй по значению предсказанной вероятностью.
<code>score_category_probabilities</code>	<i>логический</i>	При значении <code>True</code> создает предсказанные вероятности для категориальных полей назначения. Для каждой категории создается поле. Значение по умолчанию - <code>False</code> .
<code>max_categories</code>	<i>целое</i>	Максимальное количество категорий, для которых будут предсказываться вероятности. Используется только в том случае, когда значение <code>score_category_probabilities</code> - это <code>True</code> .
<code>score_propensity</code>	<i>логический</i>	Если задано значение <code>True</code> , создает простые оценки склонности (правдоподобие выхода "True") для моделей с флаговыми полями назначения. Если используются разделы, создаются также скорректированные оценки склонности на основании обучающего раздела. Значение по умолчанию - <code>False</code> .

Свойства узла `applykmeans`

Узлы моделирования К-средних можно использовать для генерирования слепка модели К-средних. Имя сценария этого слепка модели - `applykmeans`. Других свойств для этого слепка модели нет. Более подробную информацию о сценарии самого узла моделирования смотрите в разделе “Свойства узла `kmeans`” на стр. 155.

Свойства узла applyknn

Узлы моделирования KNN можно использовать для генерирования слепка модели KNN. Имя сценария этого слепка модели - *applyknn*. Более подробную информацию о сценарии самого узла моделирования смотрите в разделе “Свойства узла knn” на стр. 155.

Таблица 130. Свойства узла applyknn.

Свойства узла applyknn	Значения	Описание свойства
all_probabilities	логический	
save_distances	логический	

Свойства узла applykohonen

Узлы моделирования Коонена можно использовать для генерирования слепка модели Коонена. Имя сценария этого слепка модели - *applykohonen*. Других свойств для этого слепка модели нет. Более подробную информацию о сценарии самого узла моделирования смотрите в разделе “Свойства узла kohonen” на стр. 157.

Свойства узла applylinear

Узлы линейного моделирования можно использовать для генерирования слепка линейной модели. Имя сценария этого слепка модели - *applylinear*. Более подробную информацию о сценарии самого узла моделирования смотрите в разделе “Свойства узла linear” на стр. 157.

Таблица 131. Свойства узла applylinear.

Свойства узла applylinear	Значения	Описание свойства
use_custom_name	логический	
custom_name	строка	
enable_sql_generation	логический	

Свойства узла applylogreg

Узлы моделирования логистической регрессии можно использовать для генерирования слепка модели логистической регрессии. Имя сценария этого слепка модели - *applylogreg*. Более подробную информацию о сценарии самого узла моделирования смотрите в разделе “Свойства узла logreg” на стр. 159.

Таблица 132. Свойства узла applylogreg.

Свойства узла applylogreg	Значения	Описание свойства
calculate_raw_propensities	логический	
calculate_conf	логическое	
enable_sql_generation	логическое	

Свойства узла applyneuralnet

Узлы моделирования нейронной сети можно использовать для генерирования слепка модели нейронной сети. Имя сценария этого слепка модели - *applyneuralnet*. Более подробную информацию о сценарии самого узла моделирования смотрите в разделе “Свойства узла neuralnet” на стр. 161.

Внимание: В этом выпуске доступна более новая версия узла моделирования нейронных сетей с расширенными возможностями, которая обсуждается в следующем разделе (*applyneuralnetwork*). Несмотря

на то, что предыдущая версия все еще доступна, мы рекомендуем обновить ваши сценарии для использования новой версии. Подробности предыдущей версии приведены здесь для справки, но ее поддержка будет прекращена в следующем выпуске.

Таблица 133. Свойства узла *applyneuralnet*.

Свойства узла <i>applyneuralnet</i>	Значения	Описание свойства
calculate_conf	логический	Доступно, когда включено генерирование SQL; это свойство включает в себя вычисления доверительных показателей в сгенерированном дереве.
enable_sql_generation	логический	
nn_score_method	Разность SoftMax	
calculate_raw_propensities	логический	
calculate_adjusted_propensities	логический	

Свойства узла *applyneuralnetwork*

Узлы моделирования нейронной сети можно использовать для генерирования слепка модели нейронной сети. Имя сценария этого слепка модели - *applyneuralnetwork*. Более подробную информацию о сценарии самого узла моделирования смотрите в разделе “Свойства узла *neuralnetwork*” на стр. 163.

Таблица 134. Свойства узла *applyneuralnetwork*.

Свойства узла <i>applyneuralnetwork</i>	Значения	Описание свойства
use_custom_name	логический	
custom_name	строка	
confidence	onProbability onIncrease	
score_category_probabilities	логический	
max_categories	число	
score_propensity	логический	

Свойства узла *applyquest*

Узлы моделирования QUEST можно использовать для генерирования слепка модели QUEST. Имя сценария этого слепка модели - *applyquest*. Более подробную информацию о сценарии самого узла моделирования смотрите в разделе “Свойства узла *quest*” на стр. 165.

Таблица 135. Свойства узла *applyquest*.

Свойства узла <i>applyquest</i>	Значения	Описание свойства
sql_generate	Никогда MissingValues NoMissingValues	
calculate_conf	логический	
display_rule_id	логический	Добавляет поле в выводе данных скоринга, обозначающее ID для конечного узла, которому назначена каждая запись.
calculate_raw_propensities	логический	
calculate_adjusted_propensities	логический	

Свойства узла applyregression

Узлы моделирования линейной регрессии можно использовать для генерирования слепка модели линейной регрессии. Имя сценария этого слепка модели - *applyregression*. Других свойств для этого слепка модели нет. Более подробную информацию о сценарии самого узла моделирования смотрите в разделе “Свойства узла regression” на стр. 166.

Свойства узла applyr

Узлы моделирования R можно использовать для генерирования слепка модели R. Имя сценария этого слепка модели - *applyr*. Более подробную информацию о сценарии самого узла моделирования смотрите в разделе “Свойства узла buildr” на стр. 135.

Таблица 136. Свойства узла applyr

Свойства узла applyr	Значения	Описание свойства
score_syntax	<i>строка</i>	Синтаксис сценария R для скоринга моделей.
convert_flags	StringsAndDoubles LogicalValues	Опция для преобразования флаговых полей.
convert_datetime	<i>логическое</i>	Опции для преобразования переменных с форматом данных даты или даты-времени в форматы даты/времени R.
convert_datetime_class	POSIXct POSIXlt	Опции для указания, в какой формат нужно конвертировать переменные из формата даты или даты-времени.
convert_missing	<i>логическое</i>	Опция для преобразования значений отсутствия в значение R NA.

Свойства узла applyselflearning

Узлы моделирования откликов самообучения (Self-Learning Response Model, SLRM) можно использовать для генерирования слепка модели SLRM. Имя сценария этого слепка модели - *applyselflearning*. Более подробную информацию о сценарии самого узла моделирования смотрите в разделе “Свойства узла slrm” на стр. 168.

Таблица 137. Свойства узла applyselflearning.

Свойства узла applyselflearning	Значения	Описание свойства
max_predictions	<i>число</i>	
randomization	<i>число</i>	
scoring_random_seed	<i>число</i>	
sort	ascending descending	Задаёт, в каком порядке будут показываться предложения, начиная с максимальной или минимальной оценки.
model_reliability	<i>логический</i>	Принимает во внимание опцию надежности модели на вкладке Параметры.

Свойства узла applysequence

Узлы моделирования последовательности можно использовать для генерации слепка модели последовательности. Имя сценария этого слепка модели - *applysequence*. Других свойств для этого слепка модели нет. Более подробную информацию о сценарии самого узла моделирования смотрите в разделе “Свойства узла sequence” на стр. 168.

Свойства узла `applysvm`

Узлы моделирования SVM можно использовать для генерирования слепка модели SVM. Имя сценария этого слепка модели - `applysvm`. Более подробную информацию о сценарии самого узла моделирования смотрите в разделе “Свойства узла `svm`” на стр. 169.

Таблица 138. Свойства узла `applysvm`.

Свойства узла <code>applysvm</code>	Значения	Описание свойства
<code>all_probabilities</code>	<i>логический</i>	
<code>calculate_raw_propensities</code>	<i>логический</i>	
<code>calculate_adjusted_propensities</code>	<i>логический</i>	

Свойства узла `applytimeseries`

Узлы моделирования временных рядов можно использовать для генерирования слепка модели временных рядов. Имя сценария этого слепка модели - `applytimeseries`. Более подробную информацию о сценарии самого узла моделирования смотрите в разделе “Свойства узла `timeseries`” на стр. 170.

Таблица 139. Свойства узла `applytimeseries`.

Свойства узла <code>applytimeseries</code>	Значения	Описание свойства
<code>calculate_conf</code>	<i>логический</i>	
<code>calculate_residuals</code>	<i>логический</i>	

Свойства узла `applytwostep`

Узлы двухшагового моделирования можно использовать для генерирования слепка двухшаговой модели. Имя сценария этого слепка модели - `applytwostep`. Других свойств для этого слепка модели нет. Более подробную информацию о сценарии самого узла моделирования смотрите в разделе “Свойства узла `twostep`” на стр. 172.

Глава 15. Свойства узла моделирования базы данных

IBM SPSS Modeler поддерживает интеграцию с инструментами анализа данных и моделирования, доступными от поставщиков баз данных, в том числе в Microsoft SQL Server Analysis Services, Oracle Data Mining, IBM DB2 InfoSphere Warehouse и IBM Netezza Analytics. Вы можете построить и оценить модели с помощью собственных алгоритмов баз данных внутри прикладной программы IBM SPSS Modeler. Модели баз данных можно создавать также и работать с ними через сценарии, используя описанные в этом разделе свойства.

Свойства узлов для моделирования Microsoft

Свойства узлов моделирования Microsoft

Общие свойства

Следующие свойства общие для узлов моделирования баз данных Microsoft.

Таблица 140. Общие свойства узлов Microsoft.

Общие свойства узлов Microsoft	Значения	Описание свойства
analysis_database_name	строка	Имя базы данных Analysis Services.
analysis_server_name	строка	Имя хоста Analysis Services.
use_transactional_data	логический	Задаёт, в каком формате используются входные данные, в табличном или транзакционном.
inputs	[поле поле поле]	Входные поля для табличных данных.
target	поле	Предсказанное поле (неприменимо для узлов кластеризации MS или кластеризации последовательностей).
unique_field	поле	Ключевое поле.
msas_parameters	структурированный	Параметры алгоритмов. Дополнительную информацию смотрите в разделе “Параметры алгоритма” на стр. 186.
with_drillthrough	логический	С опцией детализации.

Дерево решений MS

Не существует специфических свойств, определенных для узлов типа `mstree`. Смотрите описание общих свойств Microsoft в начале этого раздела.

Кластеризация MS

Не существует специфических свойств, определенных для узлов типа `mscluster`. Смотрите описание общих свойств Microsoft в начале этого раздела.

Правила связывания MS

Следующие конкретные свойства доступны для узлов типа `msassoc`:

Таблица 141. Свойства узла `msassoc`.

Свойства узла <code>msassoc</code>	Значения	Описание свойства
id_field	поле	Идентифицирует каждую транзакцию в данных.

Таблица 141. Свойства узла *msassoc* (продолжение).

Свойства узла <i>msassoc</i>	Значения	Описание свойства
<i>trans_inputs</i>	[поле поле поле]	Входные поля для транзакционных данных.
<i>transactional_target</i>	поле	Предсказанное поле (транзакционные данные).

Наивный критерий Байеса MS

Не существует специфических свойств, определенных для узлов типа *msbayes*. Смотрите описание общих свойств Microsoft в начале этого раздела.

Линейная регрессия MS

Не существует специфических свойств, определенных для узлов типа *msregression*. Смотрите описание общих свойств Microsoft в начале этого раздела.

Нейросеть MS

Не существует специфических свойств, определенных для узлов типа *msneuralnetwork*. Смотрите описание общих свойств Microsoft в начале этого раздела.

Логистическая регрессия MS

Не существует специфических свойств, определенных для узлов типа *mslogistic*. Смотрите описание общих свойств Microsoft в начале этого раздела.

Временные ряды MS

Не существует специфических свойств, определенных для узлов типа *mstimeseries*. Смотрите описание общих свойств Microsoft в начале этого раздела.

Кластеризация последовательностей MS

Следующие конкретные свойства доступны для узлов типа *mssequencecluster*:

Таблица 142. Свойства узла *mssequencecluster*.

Свойства узла <i>mssequencecluster</i>	Значения	Описание свойства
<i>id_field</i>	поле	Идентифицирует каждую транзакцию в данных.
<i>input_fields</i>	[поле поле поле]	Входные поля для транзакционных данных.
<i>sequence_field</i>	поле	Идентификатор последовательности.
<i>target_field</i>	поле	Предсказанное поле (табличные данные).

Параметры алгоритма

У каждого типа моделей баз данных Microsoft есть специфические параметры, которые можно задать с использованием свойства *msas_parameters*.

Эти параметры получены от SQL Server. Чтобы увидеть соответствующие параметры для каждого узла:

1. Поместите узел источника базы данных на холст.
2. Откройте узел источника базы данных.
3. Выберите допустимый источник из раскрывающегося списка **Источник данных**.
4. Выберите допустимую таблицу из списка **Имя таблицы**.
5. Нажмите кнопку **ОК**, чтобы закрыть узел источника базы данных.

6. Присоедините узел моделирования базы данных Microsoft, свойства которого вы хотите перечислить.
7. Откройте узел моделирования баз данных.
8. Выберите вкладку **Эксперт**.

Будут выведены доступные свойства `msas_parameters` для этого узла.

Свойства слепков моделей Microsoft

Следующие свойства предназначены для слепков моделей, созданных с использованием узлов моделирования баз данных Microsoft.

Дерево решений MS

Таблица 143. Свойства дерева решений MS.

Свойства узла <code>appliedtree</code>	Значения	Описание
<code>analysis_database_name</code>	<i>строка</i>	Этот узел может быть оценен непосредственно в потоке. Это свойство используется для идентификации имени базы данных Analysis Services.
<code>analysis_server_name</code>	<i>строка</i>	Имя хоста сервера анализа.
<code>datasource</code>	<i>строка</i>	Имя источника данных (data source name, DSN) SQL Server ODBC.
<code>sql_generate</code>	<i>логический</i>	Включает генерирование SQL.

Линейная регрессия MS

Таблица 144. Свойства линейной регрессии MS.

Свойства узла <code>appliedmsregression</code>	Значения	Описание
<code>analysis_database_name</code>	<i>строка</i>	Этот узел может быть оценен непосредственно в потоке. Это свойство используется для идентификации имени базы данных Analysis Services.
<code>analysis_server_name</code>	<i>строка</i>	Имя хоста сервера анализа.

Нейросеть MS

Таблица 145. Свойства нейронной сети MS.

Свойства узла <code>appliedmsneuralnetwork</code>	Значения	Описание
<code>analysis_database_name</code>	<i>строка</i>	Этот узел может быть оценен непосредственно в потоке. Это свойство используется для идентификации имени базы данных Analysis Services.
<code>analysis_server_name</code>	<i>строка</i>	Имя хоста сервера анализа.

Логистическая регрессия MS

Таблица 146. Свойства логистической регрессии MS.

Свойства узла applieslogistic	Значения	Описание
analysis_database_name	строка	Этот узел может быть оценен непосредственно в потоке. Это свойство используется для идентификации имени базы данных Analysis Services.
analysis_server_name	строка	Имя хоста сервера анализа.

Временные ряды MS

Таблица 147. Свойства MS Time Series.

Свойства узла appliestimeseries	Значения	Описание
analysis_database_name	строка	Этот узел может быть оценен непосредственно в потоке. Это свойство используется для идентификации имени базы данных Analysis Services.
analysis_server_name	строка	Имя хоста сервера анализа.
start_from	new_prediction historical_prediction	Задаёт, какие предсказания будут делаться, будущие или хронологические.
new_step	число	Определяет начальный период времени для будущих предсказаний.
historical_step	число	Определяет начальный период времени для будущих предсказаний.
end_step	число	Определяет конечный период времени для предсказаний.

Кластеризация последовательностей MS

Таблица 148. Свойства кластеризации последовательностей MS.

Свойства узла appliessequencecluster	Значения	Описание
analysis_database_name	строка	Этот узел может быть оценен непосредственно в потоке. Это свойство используется для идентификации имени базы данных Analysis Services.
analysis_server_name	строка	Имя хоста сервера анализа.

Свойства узлов для моделирования Oracle

Свойства узлов моделирования Oracle

Следующие свойства общие для узлов моделирования баз данных Oracle.

Таблица 149. Общие свойства узлов Oracle.

Общие свойства узлов Oracle	Значения	Описание свойства
target	поле	
inputs	Список полей	

Таблица 149. Общие свойства узлов Oracle (продолжение).

Общие свойства узлов Oracle	Значения	Описание свойства
partition	поле	Поле, которое будет использоваться для разделения данных на отдельные выборки для стадий обучения, испытания и проверки при построении моделей.
datasource		
username		
password		
epassword		
use_model_name	логический	
model_name	строка	Пользовательское имя для новой модели.
use_partitioned_data	логический	Если определено поле раздела, эта опция обеспечивает, что для построения модели используются только данные из раздела обучения.
unique_field	поле	
auto_data_prep	логический	Включает или отключает возможность автоматической подготовки данных (только для баз данных 11g).
costs	структурированный	Структурированное свойство.
mode	Простые Эксперт	Приводит к тому, что при заданной опции Простые некоторые свойства игнорируются, как отмечено в свойствах конкретных узлов.
use_prediction_probability	логический	
prediction_probability	строка	
use_prediction_set	логический	

Наивный критерий Байеса Oracle

Следующие свойства доступны для узлов типа oranb.

Таблица 150. Свойства узла oranb.

Свойства узла oranb	Значения	Описание свойства
singleton_threshold	число	0,0–1,0.*
pairwise_threshold	число	0,0–1,0.*
priors	Данные Равенство Пользовательские	
custom_priors	структурированный	Структурированное свойство.

* Свойства, которые игнорируются, если для mode задано значение Простые.

Адаптивный критерий Байеса Oracle

Следующие свойства доступны для узлов типа oraabn.

Таблица 151. Свойства узла *oraabn*.

Свойства узла <i>oraabn</i>	Значения	Описание свойства
<i>model_type</i>	SingleFeature MultiFeature NaiveBayes	
<i>use_execution_time_limit</i>	логический	*
<i>execution_time_limit</i>	целое	Значение должно быть больше 0,*
<i>max_naive_bayes_predictors</i>	целое	Значение должно быть больше 0,*
<i>max_predictors</i>	целое	Значение должно быть больше 0,*
<i>priors</i>	Данные Равенство Пользовательские	
<i>custom_priors</i>	структурированный	Структурированное свойство.

* Свойства, которые игнорируются, если для *mode* задано значение Простые.

Механизмы опорных векторов Oracle

Следующие свойства доступны для узлов типа *orasvm*.

Таблица 152. Свойства узла *orasvm*.

Свойства узла <i>orasvm</i>	Значения	Описание свойства
<i>active_learning</i>	Включить Отключить	
<i>kernel_function</i>	Линейная Нормальная Системная	
<i>normalization_method</i>	zscore minmax none	
<i>kernel_cache_size</i>	целое	Только гауссовское ядро. Значение должно быть больше 0,*
<i>convergence_tolerance</i>	число	Значение должно быть больше 0,*
<i>use_standard_deviation</i>	логический	Только гауссовское ядро.*
<i>standard_deviation</i>	число	Значение должно быть больше 0,*
<i>use_epsilon</i>	логический	Только модели регрессии.*
<i>epsilon</i>	число	Значение должно быть больше 0,*
<i>use_complexity_factor</i>	логический	*
<i>complexity_factor</i>	число	*
<i>use_outlier_rate</i>	логический	Только вариант одного класса.*
<i>outlier_rate</i>	число	Только вариант одного класса. 0,0–1,0.*
<i>weights</i>	Данные Равенство Пользовательские	
<i>custom_weights</i>	структурированный	Структурированное свойство.

* Свойства, которые игнорируются, если для *mode* задано значение Простые.

Обобщенные линейные модели Oracle

Следующие свойства доступны для узлов типа `oraglm`.

Таблица 153. Свойства узла `oraglm`.

Свойства узла <code>oraglm</code>	Значения	Описание свойства
<code>normalization_method</code>	<code>zscore</code> <code>minmax</code> <code>none</code>	
<code>missing_value_handling</code>	<code>ReplaceWithMean</code> <code>UseCompleteRecords</code>	
<code>use_row_weights</code>	<i>логический</i>	*
<code>row_weights_field</code>	<i>поле</i>	*
<code>save_row_diagnostics</code>	<i>логический</i>	*
<code>row_diagnostics_table</code>	<i>строка</i>	*
<code>coefficient_confidence</code>	<i>число</i>	*
<code>use_reference_category</code>	<i>логический</i>	*
<code>reference_category</code>	<i>строка</i>	*
<code>ridge_regression</code>	Авто Выкл Вкл	*
<code>parameter_value</code>	<i>число</i>	*
<code>vif_for_ridge</code>	<i>логический</i>	*

* Свойства, которые игнорируются, если для `mode` задано значение Простые.

Дерево решений Oracle

Следующие свойства доступны для узлов типа `oradecisiontree`.

Таблица 154. Свойства узла `oradecisiontree`.

Свойства узла <code>oradecisiontree</code>	Значения	Описание свойства
<code>use_costs</code>	<i>логический</i>	
<code>impurity_metric</code>	Энтропия Gini	
<code>term_max_depth</code>	<i>целое</i>	2–20.*
<code>term_minpct_node</code>	<i>число</i>	0,0–10,0.*
<code>term_minpct_split</code>	<i>число</i>	0,0–20,0.*
<code>term_minrec_node</code>	<i>целое</i>	Значение должно быть больше 0,*
<code>term_minrec_split</code>	<i>целое</i>	Значение должно быть больше 0,*
<code>display_rule_ids</code>	<i>логический</i>	*

* Свойства, которые игнорируются, если для `mode` задано значение Простые.

O-кластер Oracle

Следующие свойства доступны для узлов типа `oraoccluster`.

Таблица 155. Свойства узла oraocluster.

Свойства узла oraocluster	Значения	Описание свойства
max_num_clusters	целое	Значение должно быть больше 0.
max_buffer	целое	Значение должно быть больше 0,*
sensitivity	число	0,0–1,0.*

* Свойства, которые игнорируются, если для mode задано значение Простые.

K-средние Oracle

Следующие свойства доступны для узлов типа orakmeans.

Таблица 156. Свойства узла orakmeans.

Свойства узла orakmeans	Значения	Описание свойства
num_clusters	целое	Значение должно быть больше 0.
normalization_method	zscore minmax none	
distance_function	Евклидова Косинус	
iterations	целое	0–20.*
conv_tolerance	число	0,0–0,5.*
split_criterion	Дисперсия Размер	По умолчанию используется дисперсия.*
num_bins	целое	Значение должно быть больше 0,*
block_growth	целое	1–5.*
min_pct_attr_support	число	0,0–1,0.*

* Свойства, которые игнорируются, если для mode задано значение Простые.

NMF Oracle

Следующие свойства доступны для узлов типа oranmf.

Таблица 157. Свойства узла oranmf.

Свойства узла oranmf	Значения	Описание свойства
normalization_method	minmax none	
use_num_features	логический	*
num_features	целое	0–1. Значение по умолчанию оценено алгоритмом по данным.*
random_seed	число	*
num_iterations	целое	0–500.*
conv_tolerance	число	0,0–0,5.*
display_all_features	логический	*

* Свойства, которые игнорируются, если для mode задано значение Простые.

Априорный анализ Oracle

Следующие свойства доступны для узлов типа oraapriori.

Таблица 158. Свойства узла oraapriori.

Свойства узла oraapriori	Значения	Описание свойства
content_field	поле	
id_field	поле	
max_rule_length	целое	2–20.
min_confidence	число	0,0–1,0.
min_support	число	0,0–1,0.
use_transactional_data	логический	

Минимальная длина описания (Minimum Description Length, MDL) Oracle

Не существует специфических свойств, определенных для узлов типа oramd1. Смотрите описание общих свойств Oracle в начале этого раздела.

Важность атрибутов (Attribute Importance, AI) Oracle

Следующие свойства доступны для узлов типа oraai.

Таблица 159. Свойства узла oraai.

Свойства узла oraai	Значения	Описание свойства
custom_fields	логический	Если значение флага true, это позволяет вам задать поле назначения, входные и другие поля для текущего узла. При значении false используются текущие параметры с вышележащего узла Тип.
selection_mode	ImportanceLevel ImportanceValue TopN	
select_important	логический	Когда для selection_mode задано значение ImportanceLevel, задает, выбрать ли важные поля.
important_label	строка	Задает метку для ранжирования "важно".
select_marginal	логический	Когда для selection_mode задано значение ImportanceLevel, задает, выбрать ли пограничные поля.
marginal_label	строка	Задает метку для ранжирования "пограничное".
important_above	число	0,0–1,0.
select_unimportant	логический	Когда для selection_mode задано значение ImportanceLevel, задает, выбрать ли не важные поля.
unimportant_label	строка	Задает метку для ранжирования "не важно".
unimportant_below	число	0,0–1,0.
importance_value	число	Когда для selection_mode задано значение ImportanceValue, задает значение отсечения для использования. Принимает значения от 0 до 100.

Таблица 159. Свойства узла *oraai* (продолжение).

Свойства узла oraai	Значения	Описание свойства
top_n	<i>число</i>	Когда для <code>selection_mode</code> задано значение TopN, задает значение отсечения для использования. Принимает значения от 0 до 1000.

Свойства слепков моделей Oracle

Следующие свойства предназначены для слепков моделей, созданных с использованием моделей Oracle.

Наивный критерий Байеса Oracle

Не существует специфических свойств, определенных для узлов типа `applyoranb`.

Адаптивный критерий Байеса Oracle

Не существует специфических свойств, определенных для узлов типа `applyoraabn`.

Механизмы векторов поддержки Oracle

Не существует специфических свойств, определенных для узлов типа `applyorasvm`.

Дерево решений Oracle

Следующие свойства доступны для узлов типа `applyoradecisiontree`.

Таблица 160. Свойства узла *applyoradecisiontree*.

Свойства узла applyoradecisiontree	Значения	Описание свойства
use_costs	<i>логический</i>	
display_rule_ids	<i>логический</i>	

O-кластер Oracle

Не существует специфических свойств, определенных для узлов типа `applyoraocluster`.

K-средние Oracle

Не существует специфических свойств, определенных для узлов типа `applyorakmeans`.

NMF Oracle

Следующие конкретные свойства доступны для узлов типа `applyoranmf`:

Таблица 161. Свойства узла *applyoranmf*.

Свойства узла applyoranmf	Значения	Описание свойства
display_all_features	<i>логический</i>	

Априорный анализ Oracle

Этот слепок модели нельзя применять в сценарии.

MDL Oracle

Этот слепок модели нельзя применять в сценарии.

Свойства узла для моделирования IBM DB2

Свойства узла моделирования IBM DB2

Следующие свойства общие для узлов моделирования баз данных IBM InfoSphere Warehouse (ISW).

Таблица 162. Свойства Common ISW node.

Общие свойства узла ISW	Значения	Описание свойства
inputs	Список полей	
datasource		
username		
password		
epassword		
enable_power_options	логический	
power_options_max_memory	целое	Значение должно быть больше 32.
power_options_cmdline	строка	
mining_data_custom_sql	строка	
logical_data_custom_sql	строка	
mining_settings_custom_sql		

Дерево решений ISW

Следующие свойства доступны для узлов типа db2imtree.

Таблица 163. Свойства узла db2imtree.

Свойства узла db2imtree	Значения	Описание свойства
target	поле	
perform_test_run	логический	
use_max_tree_depth	логический	
max_tree_depth	целое	Значение больше 0.
use_maximum_purity	логический	
maximum_purity	число	Число между 0 и 100.
use_minimum_internal_cases	логический	
minimum_internal_cases	целое	Значение больше 1.
use_costs	логический	
costs	структурированный	Структурированное свойство.

Связывание ISW

Следующие свойства доступны для узлов типа db2imassoc.

Таблица 164. Свойства узла db2imassoc.

Свойства узла db2imassoc	Значения	Описание свойства
use_transactional_data	логический	

Таблица 164. Свойства узла db2imassoc (продолжение).

Свойства узла db2imassoc	Значения	Описание свойства
id_field	поле	
content_field	поле	
data_table_layout	basic limited_length	
max_rule_size	целое	Значение должно быть больше 2.
min_rule_support	число	0–100%
min_rule_confidence	число	0–100%
use_item_constraints	логический	
item_constraints_type	Включать Exclude	
use_taxonomy	логический	
taxonomy_table_name	строка	Имя таблицы DB2 для хранения подробностей таксономии.
taxonomy_child_column_name	строка	Имя дочернего столбца в таблице таксономии. Дочерний столбец содержит имена элементов или категорий.
taxonomy_parent_column_name	строка	Имя родительского столбца в таблице таксономии. Родительский столбец содержит имена категорий.
load_taxonomy_to_table	логический	Управляет опцией, должна ли информация, хранящаяся в IBM SPSS Modeler, записываться в таблицу таксономии во время построения модели. Обратите внимание на то, что таблица таксономии отбрасывается, если она уже существует. Информация таксономии хранится на узле построения модели и изменяется с помощью кнопок Изменить категории и Изменить таксономию .

Последовательность ISW

Следующие свойства доступны для узлов типа db2imsequence.

Таблица 165. Свойства узла db2imsequence.

Свойства узла db2imsequence	Значения	Описание свойства
id_field	поле	
group_field	поле	
content_field	поле	
max_rule_size	целое	Значение должно быть больше 2.
min_rule_support	число	0–100%
min_rule_confidence	число	0–100%
use_item_constraints	логический	
item_constraints_type	Включать Exclude	
use_taxonomy	логический	
taxonomy_table_name	строка	Имя таблицы DB2 для хранения подробностей таксономии.
taxonomy_child_column_name	строка	Имя дочернего столбца в таблице таксономии. Дочерний столбец содержит имена элементов или категорий.

Таблица 165. Свойства узла db2imsequence (продолжение).

Свойства узла db2imsequence	Значения	Описание свойства
taxonomy_parent_column_name	строка	Имя родительского столбца в таблице таксономии. Родительский столбец содержит имена категорий.
load_taxonomy_to_table	логический	Управляет опцией, должна ли информация, хранимая в IBM SPSS Modeler, закачиваться в таблицу таксономии во время построения модели. Обратите внимание на то, что таблица таксономии отбрасывается, если она уже существует. Информация таксономии хранится на узле построения модели и изменяется с помощью кнопок Изменить категории и Изменить таксономию .

Регрессия ISW

Следующие свойства доступны для узлов типа db2imreg.

Таблица 166. Свойства узла db2imreg.

Свойства узла db2imreg	Значения	Описание свойства
target	поле	
regression_method	transform Линейная polynomial rbf	Смотрите в следующей таблице свойства, применимые только в том случае, если для regression_method задано значение rbf.
perform_test_run	поле	
limit_rsquared_value	логический	
max_rsquared_value	число	Значение от 0,0 до 1,0.
use_execution_time_limit	логический	
execution_time_limit_mins	целое	Значение больше 0.
use_max_degree_polynomial	логический	
max_degree_polynomial	целое	
use_intercept	логический	
use_auto_feature_selection_method	логический	
auto_feature_selection_method	Нормальное скорректировано	
use_min_significance_level	логический	
min_significance_level	число	
use_min_significance_level	логический	

Следующие свойства применимы только в том случае, если для regression_method задано значение rbf.

Таблица 167. Свойства узла db2imreg при заданном значении rbf для regression_method.

Свойства узла db2imreg	Значения	Описание свойства
use_output_sample_size	логический	При значении true автоматически задать значение по умолчанию.
output_sample_size	целое	Значение по умолчанию - 2. Минимум - 1.

Таблица 167. Свойства узла db2imreg при заданном значении rbf для regression_method (продолжение).

use_input_sample_size	логический	При значении true автоматически задать значение по умолчанию.
input_sample_size	целое	Значение по умолчанию - 2. Минимум - 1.
use_max_num_centers	логический	При значении true автоматически задать значение по умолчанию.
max_num_centers	целое	Значение по умолчанию - 20. Минимум - 1.
use_min_region_size	логический	При значении true автоматически задать значение по умолчанию.
min_region_size	целое	Значение по умолчанию - 15. Минимум - 1.
use_max_data_passes	логический	При значении true автоматически задать значение по умолчанию.
max_data_passes	целое	Значение по умолчанию - 5. Минимум равен 2.
use_min_data_passes	логический	При значении true автоматически задать значение по умолчанию.
min_data_passes	целое	Значение по умолчанию - 5. Минимум равен 2.

Кластеризация ISW

Следующие свойства доступны для узлов типа db2imcluster.

Таблица 168. Свойства узла db2imcluster.

Свойства узла db2imcluster	Значения	Описание свойства
cluster_method	demographic kohonen birch	
kohonen_num_rows	целое	
kohonen_num_columns	целое	
kohonen_passes	целое	
use_num_passes_limit	логический	
use_num_clusters_limit	логический	
max_num_clusters	целое	Значение больше 1.
birch_dist_measure	log_likelihood euclidean	Значение по умолчанию - log_likelihood.
birch_num_cfleaves	целое	Значение по умолчанию - 1000.
birch_num_refine_passes	целое	Значение по умолчанию - 3; минимум - 1.
use_execution_time_limit	логический	
execution_time_limit_mins	целое	Значение больше 0.

Таблица 168. Свойства узла db2imcluster (продолжение).

Свойства узла db2imcluster	Значения	Описание свойства
min_data_percentage	число	0–100%
use_similarity_threshold	логический	
similarity_threshold	число	Значение от 0,0 до 1,0.

Наивный байесовский анализ ISW

Следующие свойства доступны для узлов типа db2imnbs.

Таблица 169. Свойства узла db2imnb.

Свойства узла db2imnb	Значения	Описание свойства
perform_test_run	логический	
probability_threshold	число	Значение по умолчанию - 0,001. Минимальное значение - 0; максимальное значение - 1,000
use_costs	логический	
costs	структурированный	Структурированное свойство.

Логистическая регрессия ISW

Следующие свойства доступны для узлов типа db2imlog.

Таблица 170. Свойства узла db2imlog.

Свойства узла db2imlog	Значения	Описание свойства
perform_test_run	логический	
use_costs	логический	
costs	структурированный	Структурированное свойство.

Временные ряды ISW

Примечание: Для этого узла параметр входного поля не используется. Если в сценарии обнаружен параметр входного поля, выводится предупреждение, что на узле есть *время* и *назначения* как входящие поля, но нет полей ввода.

Следующие свойства доступны для узлов типа db2imtimeseries.

Таблица 171. Свойства узла db2imtimeseries.

Свойства узла db2imtimeseries	Значения	Описание свойства
время	поле	Допускается целое, время или дата.
назначения	список полей	
forecasting_algorithm	arima exponential_smoothing seasonal_trend_decomposition	
forecasting_end_time	auto целое дата время	

Таблица 171. Свойства узла db2imtimeseries (продолжение).

Свойства узла db2imtimeseries	Значения	Описание свойства
use_records_all	логический	При значении false должны быть заданы use_records_start и use_records_end.
use_records_start	целое/ время/ дата	Зависит от типа поля времени
use_records_end	целое/ время/ дата	Зависит от типа поля времени
interpolation_method	none Линейная exponential_splines cubic_splines	

Свойства слепков моделей IBM DB2

Следующие свойства предназначены для слепков моделей, созданных с использованием моделей IBM DB2 ISW.

Дерево решений ISW

Не существует специфических свойств, определенных для узлов типа applydb2imtree.

Связывание ISW

Этот слепок модели нельзя применять в сценарии.

Последовательность ISW

Этот слепок модели нельзя применять в сценарии.

Регрессия ISW

Не существует специфических свойств, определенных для узлов типа applydb2imreg.

Кластеризация ISW

Не существует специфических свойств, определенных для узлов типа applydb2imcluster.

Наивный байесовский анализ ISW

Не существует специфических свойств, определенных для узлов типа applydb2imnb.

Логистическая регрессия ISW

Не существует специфических свойств, определенных для узлов типа applydb2imlog.

Временные ряды ISW

Этот слепок модели нельзя применять в сценарии.

Свойства узлов для моделирования IBM Netezza Analytics

Свойства узлов моделирования Netezza

Следующие свойства общие для узлов моделирования баз данных IBM Netezza.

Таблица 172. Свойства Common Netezza node.

Общие свойства узлов Netezza	Значения	Описание свойства
custom_fields	логический	Если значение флага true, это позволяет вам задать поле назначения, входные и другие поля для текущего узла. При значении false используются текущие параметры с вышележащего узла Тип.
inputs	[поле1 ... полеN]	Входные (или предикторные) поля, используемые в модели.
target	поле	Поле назначения (количественное или категориальные).
record_id	поле	Поле для использования в качестве уникального идентификатора записей.
use_upstream_connection	логический	При значении true (по умолчанию) подробности соединения задаются на узле более высокого уровня. Не используется, если задано move_data_to_connection.
move_data_connection	логический	При значении true перемещает данные в базу данных, заданную connection. Не используется, если задано use_upstream_connection.
connection	структурированный	Строка соединения для базы данных Netezza, где хранится модель. Структурированное свойство в следующей форме: ['odbc' '<dsn>' '<имя_пользователя>' '<пароль>' '<имя_каталога>' '<атрибуты_соединения>' {true false}] где: <dsn> - это имя источника данных <имя_пользователя> и <пароль> - это имя пользователя и пароль для базы данных <имя_каталога> - это имя каталога <атрибуты_соединения> - это атрибуты соединения true false обозначает, нужен ли пароль.
table_name	строка	Имя таблицы базы данных, в которой будет храниться модель.
use_model_name	логический	При значении true использует имя, заданное опцией имя_модели как имя модели, в противном случае имя модели создается системой.
model_name	строка	Пользовательское имя для новой модели.
include_input_fields	логический	При значении true передает все входные поля ниже уровнем, в противном случае передает только ID_записи и поля, сгенерированные моделью.

Дерево решений Netezza

Следующие свойства доступны для узлов типа netezzadectree.

Таблица 173. Свойства узла *netezzadectree*.

Свойства узла <i>netezzadectree</i>	Значения	Описание свойства
<code>impurity_measure</code>	Энтропия Gini	Мера неоднородности используется для оценки лучшего положения для разветвления дерева.
<code>max_tree_depth</code>	<i>целое</i>	Максимальное количество уровней, до которых может расти дерево. Значение по умолчанию 62 (максимально возможное).
<code>min_improvement_splits</code>	<i>число</i>	Минимальное значение улучшения неоднородности, чтобы произошло разделение. Значение по умолчанию 0,01.
<code>min_instances_split</code>	<i>целое</i>	Минимальное количество остающихся неразделенных записей до возможного разделения. Значение по умолчанию 2 (минимально возможное).
<code>weights</code>	<i>структурированный</i>	Относительные значения веса для классов. Структурированное свойство. По умолчанию вес 1 задается для всех классов.
<code>pruning_measure</code>	Acc wAcc	Значение по умолчанию Acc (точность). Альтернативное значение wAcc (взвешенная точность) при применении усечения учитывает веса классов.
<code>prune_tree_options</code>	<code>allTrainingData</code> <code>partitionTrainingData</code> <code>useOtherTable</code>	По умолчанию для оценки точности модели используется <code>allTrainingData</code> . Используйте <code>partitionTrainingData</code> , чтобы задать для использования процентную долю данных обучения, или <code>useOtherTable</code> , чтобы использовать набор данных обучения из заданной таблицы базы данных.
<code>perc_training_data</code>	<i>число</i>	Если для <code>prune_tree_options</code> задано значение <code>partitionTrainingData</code> , определяет процентную долю данных для использования при обучении.
<code>prune_seed</code>	<i>целое</i>	Начальное значение генератора псевдослучайных чисел, используемое для репликации результатов анализа, когда для <code>prune_tree_options</code> задано значение <code>partitionTrainingData</code> ; значение по умолчанию равно 1.
<code>pruning_table</code>	<i>строка</i>	Имя таблицы отдельного набора данных усечения для оценки точности модели.
<code>compute_probabilities</code>	<i>логический</i>	При значении <code>true</code> создает поле доверительного интервала (вероятности), а также поле предсказания.

Следующие свойства доступны для узлов типа `netezzakmeans`.

Таблица 174. Свойства узла `netezzakmeans`.

Свойства узла <code>netezzakmeans</code>	Значения	Описание свойства
<code>distance_measure</code>	Евклидова Манхеттен Канберра максимум	Способ, используемый для измерения расстояния между точками данных.
<code>num_clusters</code>	<i>целое</i>	Количество кластеров, которые будут созданы; значение по умолчанию 3.
<code>max_iterations</code>	<i>целое</i>	Количество итераций алгоритма, после которого останавливается обучение модели; значение по умолчанию 5.
<code>rand_seed</code>	<i>целое</i>	Начальное значение генератора псевдослучайных чисел, используемое для репликации результатов анализа; значение по умолчанию 12345.

Байесовская сеть Netezza

Следующие свойства доступны для узлов типа `netezزابayes`.

Таблица 175. Свойства узла `netezزابayes`.

Свойства узла <code>netezزابayes</code>	Значения	Описание свойства
<code>base_index</code>	<i>целое</i>	Числовой идентификатор, назначенный первому входному полю для внутреннего управления; значение по умолчанию 777.
<code>sample_size</code>	<i>целое</i>	Размер выборки, которую требуется сделать при очень большом числе атрибутов; значение по умолчанию 10 000.
<code>display_additional_information</code>	<i>логический</i>	При значении <code>true</code> в диалоговом окне сообщений выводит дополнительную информацию о ходе выполнения.
<code>type_of_prediction</code>	<code>best</code> <code>соседи</code> <code>nn-neighbors</code>	Тип алгоритма предсказания для использования: наилучший (наиболее скоррелированные соседи), соседи (взвешенное предсказание соседей) или <code>nn-соседи</code> (непустые соседние).

Наивный байесовский анализ Netezza

Следующие свойства доступны для узлов типа `netezзанаivebayes`.

Таблица 176. Свойства узла `netezзанаivebayes`.

Свойства узла <code>netezзанаivebayes</code>	Значения	Описание свойства
<code>compute_probabilities</code>	<i>логический</i>	При значении <code>true</code> создает поле доверительного интервала (вероятности), а также поле предсказания.
<code>use_m_estimation</code>	<i>логический</i>	При значении <code>true</code> использует способ <code>m-оценки</code> для исключения при оценке нулевых вероятностей.

KNN Netezza

Следующие свойства доступны для узлов типа `netezзакnn`.

Таблица 177. Свойства узла netezzaknn.

Свойства узла netezzaknn	Значения	Описание свойства
weights	<i>структурированный</i>	Структурированное свойство, используемое для назначения весов индивидуальным классам.
distance_measure	Евклидова Манхеттен Канберра Максимум	Способ, используемый для измерения расстояния между точками данных.
num_nearest_neighbors	<i>целое</i>	Количество ближайших соседей для конкретного наблюдения; значение по умолчанию 3.
standardize_measurements	<i>логический</i>	При значении true стандартизует измерения для количественных входных полей перед вычислением значений расстояния.
use_coresets	<i>логический</i>	При значении true использует выборку базового набора для ускорения вычислений для больших наборов данных.

Разделительная кластеризация Netezza

Следующие свойства доступны для узлов типа netezadivcluster.

Таблица 178. Свойства узла netezadivcluster.

Свойства узла netezadivcluster	Значения	Описание свойства
distance_measure	Евклидова Манхеттен Канберра Максимум	Способ, используемый для измерения расстояния между точками данных.
max_iterations	<i>целое</i>	Максимальное количество итераций алгоритма для выполнения перед остановками обучения модели; значение по умолчанию - 5.
max_tree_depth	<i>целое</i>	Максимальное число уровней, на которые можно разделить набор данных; значение по умолчанию 3.
rand_seed	<i>целое</i>	Начальное значение генератора псевдослучайных чисел, используемое для репликации результатов анализа; значение для умолчанию 12345.
min_instances_split	<i>целое</i>	Минимальное количество записей, которые можно разделить; значение по умолчанию 5.
level	<i>целое</i>	Уровень иерархии, до которого будут оцениваться записи; значение по умолчанию равно -1.

PCA Netezza

Следующие свойства доступны для узлов типа netezzarca.

Таблица 179. Свойства узла netezzarca.

Свойства узла netezzarca	Значения	Описание свойства
center_data	<i>логический</i>	При значении true (по умолчанию), эта опция перед анализом выполняет центрирование данных (называемое также "извлечение среднего").

Таблица 179. Свойства узла *netezzapca* (продолжение).

Свойства узла <i>netezzapca</i>	Значения	Описание свойства
<code>perform_data_scaling</code>	<i>логический</i>	При значении <code>true</code> выполняет масштабирование данных перед анализом. Это делает анализ менее произвольным, когда различные переменные измеряются разными единицами.
<code>force_eigenolve</code>	<i>логический</i>	При значении <code>true</code> использует менее точный, но быстрый способ нахождения главных компонент.
<code>pc_number</code>	<i>целое</i>	Количество главных компонент, до которого будет сокращен набор данных; значение по умолчанию 1.

Дерево регрессии Netezza

Следующие свойства доступны для узлов типа *netezzaregtree*.

Таблица 180. Свойства узла *netezzaregtree*.

Свойства узла <i>netezzaregtree</i>	Значения	Описание свойства
<code>max_tree_depth</code>	<i>целое</i>	Максимальное количество уровней, до которых может расти дерево ниже корневого узла; значение по умолчанию 10.
<code>split_evaluation_measure</code>	Дисперсия	Показатель неоднородности классов, используется для оценки лучшего положения для разделения дерева; значение по умолчанию (в настоящее время единственное) - Дисперсия.
<code>min_improvement_splits</code>	<i>число</i>	Минимальное сокращение неоднородности перед созданием нового разделения в дереве.
<code>min_instances_split</code>	<i>целое</i>	Минимальное количество записей, которые можно разделить.
<code>pruning_measure</code>	mse r2 Пирсона Спирмана	Способ, который будет использоваться для усечения.
<code>prune_tree_options</code>	allTrainingData partitionTrainingData useOtherTable	По умолчанию для оценки точности модели используется <code>allTrainingData</code> . Используйте <code>partitionTrainingData</code> , чтобы задать для использования процентную долю данных обучения, или <code>useOtherTable</code> , чтобы использовать набор данных обучения из заданной таблицы базы данных.
<code>perc_training_data</code>	<i>число</i>	Если для <code>prune_tree_options</code> задано значение <code>PercTrainingData</code> , определяет процентную долю данных для использования при обучении.
<code>prune_seed</code>	<i>целое</i>	Начальное значение генератора псевдослучайных чисел, используемое для репликации результатов анализа, когда для <code>prune_tree_options</code> задано значение <code>PercTrainingData</code> ; значение по умолчанию равно 1.

Таблица 180. Свойства узла *netezzaregtree* (продолжение).

Свойства узла <i>netezzaregtree</i>	Значения	Описание свойства
<code>pruning_table</code>	<i>строка</i>	Имя таблицы отдельного набора данных усечения для оценки точности модели.
<code>compute_probabilities</code>	<i>логический</i>	При значении <code>true</code> задает, что дисперсии назначенных классов должны включаться в выходные данные.

Линейная регрессия Netezza

Следующие свойства доступны для узлов типа *netezzalineression*.

Таблица 181. Свойства узла *netezzalineression*.

Свойства узла <i>netezzalineression</i>	Значения	Описание свойства
<code>use_svd</code>	<i>логический</i>	При значении <code>true</code> использует матрицу декомпозиции единичного значения вместо исходной матрицы для повышения скорости и точности вычислений.
<code>include_intercept</code>	<i>логический</i>	При значении <code>true</code> (по умолчанию) повышает общую точность решения.
<code>calculate_model_diagnostics</code>	<i>логический</i>	При значении <code>true</code> вычисляет диагностику для модели.

Временные ряды Netezza

Следующие свойства доступны для узлов типа *netezzatimeseries*.

Таблица 182. Свойства узла *netezzatimeseries*.

Свойства узла <i>netezzatimeseries</i>	Значения	Описание свойства
<code>time_points</code>	<i>поле</i>	Входное поле, содержащее значения даты или времени для временных рядов.
<code>time_series_ids</code>	<i>поле</i>	Входное поле, содержащее значения ID временных рядов; используется, если входные данные содержат больше одного временного ряда.
<code>model_table</code>	<i>поле</i>	Имя таблицы базы данных, где будет храниться модель временного ряда Netezza.
<code>description_table</code>	<i>поле</i>	Имя входной таблицы, содержащей имена и описания временных рядов.
<code>seasonal_adjustment_table</code>	<i>поле</i>	Имя выходной таблицы, где будут храниться значения, скорректированные по сезону, вычисленные по алгоритмам экспоненциального сглаживания или декомпозиции сезонных тенденций.
<code>algorithm_name</code>	<code>SpectralAnalysis</code> или спектральный <code>ExponentialSmoothing</code> или <code>esoothing</code> АРПСС <code>SeasonalTrendDecomposition</code> или <code>std</code>	Алгоритмы, которые будут использованы для моделирования временных рядов.

Таблица 182. Свойства узла *netezzatimeseries* (продолжение).

Свойства узла <i>netezzatimeseries</i>	Значения	Описание свойства
trend_name	N A DA M DM	Тип тенденции для экспоненциального сглаживания: N - нет A - аддитивный DA - демпфированный аддитивный M - мультипликативный DM - демпфированный мультипликативный
seasonality_type	N A M	Тип сезонности для экспоненциального сглаживания: N - нет A - аддитивный M - мультипликативный
interpolation_method	Линейная cubicspline exponential spline	Способ интерполяции, который будет использован.
timerange_setting	SD SP	Параметр диапазона времени для использования: SD - определяется системой (system-determined), использует полный диапазон данных временного ряда SP - задается пользователем в полях начальное_время и конечное_время
earliest_time	<i>Дата</i>	Начальное и конечное время, если для timerange_setting задано значение SP. Формат: <гггг>-<мм>-<дд>
latest_time		
arima_setting	SD SP	Параметр для алгоритма ARIMA (используется только в том случае, если для имя_алгоритма задано значение ARIMA): SD - определяется системой SP - задается пользователем Если arima_setting = SP, используйте следующие параметры, чтобы задать сезонные и несезонные значения.
p_symbol	less eq lesseq	ARIMA - оператор для параметров p, d, q, sp, sd и sq: less - меньше чем eq - равно lesseq - less than or equal to
d_symbol		
q_symbol		
sp_symbol		
sd_symbol		
sq_symbol		
p	<i>целое</i>	ARIMA - несезонные степени автокорреляции.

Таблица 182. Свойства узла *netezzatimeseries* (продолжение).

Свойства узла <i>netezzatimeseries</i>	Значения	Описание свойства
q	<i>целое</i>	ARIMA - несезонное значение отклонения.
d	<i>целое</i>	ARIMA - несезонное количество порядков скользящего среднего в модели.
sp	<i>целое</i>	ARIMA - сезонные степени автокорреляции.
sq	<i>целое</i>	ARIMA - сезонное значение отклонения.
sd	<i>целое</i>	ARIMA - сезонное количество порядков скользящего среднего в модели.
advanced_setting	SD SP	Определяет, как будут обрабатываться расширенные параметры: SD - определяется системой SP - задается пользователем через период, единицы_периода и параметр_прогноза.
period	<i>целое</i>	Длина сезонного цикла, задается в сочетании с единицами_периода. Не применяется для спектрального анализа.
units_period	ms s min h d wk q y	Единицы, в которых выражается период: ms - миллисекунды s - секунды min - минуты h - часы d - дни wk - недели q - кварталы y - годы Например, для понедельных временных рядов используйте 1 для периода и wk для единиц_периода.
forecast_setting	forecasthorizon forecasttimes	Задает, как будет делаться прогноз.
forecast_horizon	<i>строка</i>	Если forecast_setting = forecasthorizon, задает конечную точку прогнозирования. Формат: <гггг>-<мм>-<дд>
forecast_times	[{'дата'}, {'дата'},..., {'дата'}]	Если forecast_setting = forecasttimes, задает значения времени для того, чтобы делать прогнозы. Формат: <гггг>-<мм>-<дд>

Таблица 182. Свойства узла *netezzatimeseries* (продолжение).

Свойства узла <i>netezzatimeseries</i>	Значения	Описание свойства
<code>include_history</code>	<i>логический</i>	Обозначает, будут ли включены хронологические значения в выходные данные.
<code>include_interpolated_values</code>	<i>логический</i>	Обозначает, будут ли включены интерполированные значения в выходные данные. Не применяется, если для <code>include_history</code> задано значение <code>false</code> .

Обобщенный линейный анализ Netezza

Следующие свойства доступны для узлов типа *netezzaglm*.

Таблица 183. Свойства узла *netezzaglm*.

Свойства узла <i>netezzaglm</i>	Значения	Описание свойства
<code>dist_family</code>	<code>bernoulli</code> Нормальное <code>poisson</code> <code>negativebinomial</code> <code>wald</code> <code>gamma</code>	Тип распределения; значение по умолчанию <code>bernoulli</code> .
<code>dist_params</code>	<i>число</i>	Значение параметра распределения для использования. Применимо только в том случае, если <code>distribution</code> - это <code>Negativebinomial</code> .
<code>trials</code>	<i>целое</i>	Применимо только в том случае, если <code>distribution</code> - это <code>Binomial</code> . Если отклик назначения - это количество событий, произошедших в наборе испытаний, поле <code>target</code> содержит количество событий, а поле <code>trials</code> - число испытаний.
<code>model_table</code>	<i>поле</i>	Имя таблицы базы данных, где будет храниться обобщенная линейная модель Netezza.
<code>maxit</code>	<i>целое</i>	Максимальное количество итераций, которые должны быть выполнены алгоритмом; значение по умолчанию 20.
<code>eps</code>	<i>число</i>	Максимальное значение ошибки (в экспоненциальном представлении), при котором алгоритм должен остановиться при поиске модели с наилучшей подгонкой. Значение по умолчанию -3, что означает $1E-3$, или 0,001.

Таблица 183. Свойства узла *netezzaglm* (продолжение).

Свойства узла <i>netezzaglm</i>	Значения	Описание свойства
<i>tol</i>	<i>число</i>	Значение (в экспоненциальном представлении), ниже которого ошибки рассматриваются как имеющие нулевое значение. Значение по умолчанию -7, то есть значения ошибки меньше 1E-7 (или 0,0000001) рассматриваются как несущественные.
<i>link_func</i>	Тождество обратное <i>invnegative</i> <i>invsquare</i> <i>sqrt</i> Степень <i>oddspower</i> <i>log</i> <i>clog</i> <i>loglog</i> <i>cloglog</i> Логит Пробит <i>gaussit</i> <i>cauchit</i> <i>canbinom</i> <i>cangeom</i> <i>cannegbinom</i>	Функция связи, которая будет использоваться; значение по умолчанию <i>logit</i> .
<i>link_params</i>	<i>число</i>	Значение параметра функции связи для использования. Применимо только в том случае, если <i>link_function</i> - это <i>power</i> или <i>oddspower</i> .
<i>interaction</i>	[{[<i>имена_столбцов1</i>],[<i>уровни1</i>]}, {[<i>имена_столбцов2</i>], [<i>уровни2</i>]},..., {[<i>имена_столбцовN</i>], [<i>уровниN</i>]},.]	Задаёт взаимодействия между полями. <i>имена_столбцов</i> - это список входных полей, а <i>уровень</i> - это всегда 0 для каждого поля.
<i>intercept</i>	<i>логический</i>	При значении <i>true</i> включает в модель свободный член.

Свойства слепков моделей Netezza

Следующие свойства общие для слепков моделей баз данных Netezza.

Таблица 184. Общие свойства слепков моделей Netezza.

Общие свойства слепков моделей Netezza	Значения	Описание свойства
<i>connection</i>	<i>строка</i>	Строка соединения для базы данных Netezza, где хранится модель.
<i>table_name</i>	<i>строка</i>	Имя таблицы базы данных, в которой хранится модель.

Другие свойства слепков моделей - те же, что для соответствующего узла моделирования.

Имена сценариев слепков моделей следующие.

Таблица 185. Имена сценариев слепков моделей Netezza.

Образец модели	Имя сценария
Деревья решений	applynetezzadectree
К-средних	applynetezzakmeans
Байесовская сеть	applynetezzabayes
Наивный Байес	applynetezzanaivebayes
KNN	applynetezzaknn
Разделительная кластеризация	applynetezzadivcluster
РСА	applynetezzapca
Дерево регрессии	applynetezzaregtree
линейная регрессия	applynetezzalineregression
Временные ряды	applynetezzatimeseries
Обобщенная линейная	applynetezzaglm

Глава 16. Свойства узлов вывода

Свойства узлов вывода немного отличаются от свойств других типов узлов. Вместо того, чтобы относиться к определенной опции узла, свойства узлов вывода хранят ссылку на выходной объект. Это полезно, когда нужно взять значение из таблицы, а затем задать его как параметр потока.

В этом разделе описываются свойства сценариев, доступные для полей вывода.

Свойства узла analysis



Узел Анализ оценивает способность прогнозирующих моделей генерировать точные предсказания. Узлы анализа выполняют различные сравнения между предсказанными и фактическими значениями для одного или нескольких слепков моделей. Они могут сравнивать также прогнозирующие модели друг с другом.

Таблица 186. Свойства узла analysis.

Свойства узла analysis	Тип переменной	Описание свойства
output_mode	Screen File	Используется для задания положения назначения для выходных данных, сгенерированных на узле выходных данных.
use_output_name	логический	Указывает, используется ли пользовательское имя выходного поля.
output_name	строка	Если для use_output_name задано true, определяет имя для использования.
output_format	Text (.txt) HTML (.html) Output (.cou)	Используется для указания типа выхода.
by_fields	[поле поле поле]	
full_filename	строка	Имя выходного файла для диска, данных или HTML.
coincidence	логический	
performance	логический	
evaluation_binary	логическое	
confidence	логический	
threshold	число	
improve_accuracy	число	
inc_user_measure	логический	
user_if	expr	
user_then	expr	
user_else	expr	
user_compute	[Среднее Сумма Мин Макс Среднеквадр_отклон]	

Свойства узла dataaudit



Узел Аудит данных предоставляет всесторонний первый взгляд на данные, в том числе сводную статистику, гистограммы и распределение для каждого поля, а также информацию о выбросах, значениях отсутствия и экстремумах. Результаты выводятся в виде простой для чтения матрицы, которую можно отсортировать и использовать для генерирования узлов полноразмерных графиков и подготовки данных.

Таблица 187. Свойства узла dataaudit.

Свойства узла dataaudit	Тип переменной	Описание свойства
custom_fields	логический	
fields	[поле1 ... полеN]	
наложение	поле	
display_graphs	логический	Используется для включения или выключения вывода графиков в выходной матрице.
basic_stats	логический	
advanced_stats	логический	
median_stats	логический	
calculate	Count Breakdown	Используется для вычисления пропущенных значений. Выберите один из способов вычисления, оба или ни одного.
outlier_detection_method	std iqr	Используется для задания способа детектирования выбросов и экстремальных значений.
outlier_detection_std_outlier	число	Если способ_обнаружения_выбросов - это std, задает число, используемое для определения выбросов.
outlier_detection_std_extreme	число	Если способ_обнаружения_выбросов - это std, задает число, используемое для определения экстремальных значений.
outlier_detection_iqr_outlier	число	Если способ_обнаружения_выбросов - это iqr, задает число, используемое для определения выбросов.
outlier_detection_iqr_extreme	число	Если способ_обнаружения_выбросов - это iqr, задает число, используемое для определения экстремальных значений.
use_output_name	логический	Задает, используется ли пользовательское имя вывода.
output_name	строка	Если для use_output_name задано true, определяет имя для использования.
output_mode	Screen File	Используется для задания положения назначения для выходных данных, сгенерированных на узле выходных данных.

Таблица 187. Свойства узла dataaudit (продолжение).

Свойства узла dataaudit	Тип переменной	Описание свойства
output_format	Formatted (.tab) Delimited (.csv) HTML (.html) Output (.cou)	Используется для указания типа выходных данных.
paginate_output	логический	Когда output_format - это HTML, выходные данные будут разделяться на страницы.
lines_per_page	число	При использовании с paginate_output задает количество строк на страницу в выходных данных.
full_filename	строка	

Свойства узла matrix



Узел Матрица создает таблицу, показывающую взаимосвязи между полями. Чаще всего он используется для показа взаимосвязи между двумя символическими полями, но он же может показывать взаимосвязи между флаговыми или числовыми полями.

Таблица 188. Свойства узла matrix.

Свойства узла matrix	Тип переменной	Описание свойства
fields	Выбранные Флаги Numerics	
строка	поле	
столбец	поле	
include_missing_values	логический	Задает, включается ли в вывод строк и столбцов пользовательские значения отсутствия (пустые) и системные значения отсутствия (null).
cell_contents	CrossTabs Function	
function_field	строка	
function	Sum Mean Min Max SDev	
sort_mode	Без сортировки По возрастанию По убыванию	
highlight_top	число	Если отлично от нуля, то true.
highlight_bottom	число	Если отлично от нуля, то true.

Таблица 188. Свойства узла *matrix* (продолжение).

Свойства узла <i>matrix</i>	Тип переменной	Описание свойства
display	[Количества Ожидаемые Остатки RowPct ColumnPct TotalPct]	
include_totals	<i>логический</i>	
use_output_name	<i>логический</i>	Задает, используется ли пользовательское имя вывода.
output_name	<i>строка</i>	Если для use_output_name задано true, определяет имя для использования.
output_mode	Screen File	Используется для задания положения назначения для выходных данных, сгенерированных на узле выходных данных.
output_format	Formatted (.tab) Delimited (.csv) HTML (.html) Output (.cou)	Используется для указания типа выходных данных. Оба формата, Formatted и Delimited, могут использовать модификатор transposed, который транспонирует строки и столбцы в таблице.
paginate_output	<i>логический</i>	Когда output_format - это HTML, выходные данные будут разделяться на страницы.
lines_per_page	<i>число</i>	При использовании с paginate_output задает количество строк на страницу в выходных данных.
full_filename	<i>строка</i>	

Свойства узла *means*



Узел средних значений сравнивает независимые группы или пары связанных полей для проверки, существует ли между ними существенное различие. Например, можно сравнить средние прибыли до и после рекламной кампании, или сравнить прибыли от клиентов, не получивших рекламы, и клиентов, участвовавших в программе продвижения товара.

Таблица 189. Свойства узла *means*.

Свойства узла <i>means</i>	Тип переменной	Описание свойства
means_mode	BetweenGroups BetweenFields	Задает тип статистики средних для выполнения для данных.
test_fields	[поле1 ... полеN]	Задает проверяемое поле, когда для means_mode задано BetweenGroups.
grouping_field	<i>поле</i>	Задает поле группировки.
paired_fields	[{поле1 поле2} {поле3 поле4} ...]	Задает пары полей, когда для means_mode задано BetweenFields.

Таблица 189. Свойства узла means (продолжение).

Свойства узла means	Тип переменной	Описание свойства
label_correlations	логический	Задаёт, показываются ли при выводе метки корреляции. Этот параметр применим, когда для means_mode задано BetweenFields.
correlation_mode	Вероятность Абсолютная	Задаёт, как помечать корреляции - по вероятности или по абсолютному значению.
weak_label	строка	
medium_label	строка	
strong_label	строка	
weak_below_probability	число	Когда для correlation_mode задано значение Probability, задаёт значение отсечения для слабых корреляций. Это должно быть значение от 0 до 1, например, 0,90.
strong_above_probability	число	Значение отсечения для сильных корреляций.
weak_below_absolute	число	Когда для correlation_mode задано значение Absolute, задаёт значение отсечения для слабых корреляций. Это должно быть значение от 0 до 1, например, 0,90.
strong_above_absolute	число	Значение отсечения для сильных корреляций.
unimportant_label	строка	
marginal_label	строка	
important_label	строка	
unimportant_below	число	Значение отсечения для низкой важности полей. Это должно быть значение от 0 до 1, например, 0,90.
important_above	число	
use_output_name	логический	Задаёт, используется ли пользовательское имя вывода.
output_name	строка	Имя для использования.
output_mode	Screen File	Задаёт положение назначения для выходных данных, сгенерированных на узле выходных данных.
output_format	Formatted (.tab) Delimited (.csv) HTML (.html) Output (.cou)	Задаёт тип выходных данных.
full_filename	строка	
output_view	Простые Дополнительные параметры	Задаёт, какое представление используется для вывода - простое или расширенное.

Свойства узла report



Узел отчетов создает форматированные отчеты, содержащие фиксированный текст, а также данные и другие выражения, полученные из данных. Вы задаете формат отчета, используя текстовые шаблоны, чтобы определить конструкции фиксированного текста и вывода данных. Вы можете предоставить пользовательское форматирование текста с помощью тегов HTML в шаблоне и задав опции на вкладке Вывод. Значения данных и другой условный вывод можно включить в отчет с использованием выражений CLEM в шаблоне.

Таблица 190. Свойства узла report.

Свойства узла report	Тип переменной	Описание свойства
output_mode	Screen File	Используется для задания положения назначения для выходных данных, сгенерированных на узле выходных данных.
output_format	HTML (.html) Text (.txt) Output (.cou)	Используется для указания типа выхода.
use_output_name	логический	Задаёт, используется ли пользовательское имя вывода.
output_name	строка	Если для use_output_name задано true, определяет имя для использования.
text	строка	
full_filename	строка	
highlights	логический	
title	строка	
lines_per_page	число	

Свойства узла Routput



Узел вывода R дает возможность проанализировать данные и результаты оценки модели, используя пользовательский сценарий R. Вывод анализа может быть текстовым или графическим. Кроме того, вывод добавляется на вкладку **Вывод** панели менеджеров; другой вариант - направить вывод в файл.

Таблица 191. Свойства узла Routput.

Свойства узла Routput	Тип переменной	Описание свойства
синтаксис	строка	
convert_flags	StringsAndDoubles LogicalValues	
convert_datetime	логическое	
convert_datetime_class	POSIXct POSIXlt	
convert_missing	логическое	
output_name	Авто Пользовательское:	

Таблица 191. Свойства узла *Routput* (продолжение).

Свойства узла <i>Routput</i>	Тип переменной	Описание свойства
custom_name	строка	
output_to	Screen File	
output_type	Graph Text	
full_filename	строка	
graph_file_type	HTML COU	
text_file_type	HTML TXT COU	

Свойства узла *setglobals*



Узел *Задать глобальные значения* просматривает данные и вычисляет сводные значения, которые можно использовать в выражениях CLEM. Например, можно использовать этот узел для вычисления статистических показателей для поля с именем *age*, а затем использовать общее среднее *age* в выражениях CLEM, вставив функцию @GLOBAL_MEAN(*age*).

Таблица 192. Свойства узла *setglobals*.

Свойства узла <i>setglobals</i>	Тип переменной	Описание свойства
globals	[Сумма Среднее Мин Макс Среднеквадр_отклон]	Структурированное свойство
clear_first	логический	
show_preview	логический	

Свойства узла *simeval*



Узел *оценки имитации* оценивает заданное предсказанное поле назначения и представляет информацию о распределении и корреляции этого поля назначения.

Таблица 193. Свойства узла *simeval*.

Свойства узла <i>simeval</i>	Тип переменной	Описание свойства
target	<i>field</i>	
iteration	<i>field</i>	
presorted_by_iteration	логическое	
max_iterations	число	
tornado_fields	[поле_1...поле_N]	
plot_pdf	логическое	
plot_cdf	логическое	
show_ref_mean	логическое	

Таблица 193. Свойства узла *simeval* (продолжение).

Свойства узла <i>simeval</i>	Тип переменной	Описание свойства
show_ref_median	логическое	
show_ref_sigma	логическое	
num_ref_sigma	число	
show_ref_pct	логическое	
ref_pct_bottom	число	
ref_pct_top	число	
show_ref_custom	логическое	
ref_custom_values	[число_1...число_N]	
category_values	Категория Вероятности И те, и другие	
category_groups	Категории Итерации	
create_pct_table	логическое	
pct_table	Квартили Интервалы Пользовательское:	
pct_intervals_num	число	
pct_custom_values	[число_1...число_N]	

Свойства узла *simfit*



Узел подгонки имитации исследует статистическое распределение данных в каждом поле и генерирует (или обновляет) узел генерирования имитации, используя для каждого поля оптимально подогнанное распределение. Затем узел генерирования имитации можно использовать для генерирования данных имитации.

Таблица 194. Свойства узла *simfit*.

Свойства узла <i>simfit</i>	Тип переменной	Описание свойства
build	Узел XMLExport И те, и другие	
use_source_node_name	логическое	
source_node_name	строка	Пользовательское имя генерируемого или изменяемого узла источника.
use_cases	Все LimitFirstN	
use_case_limit	целое	
fit_criterion	AndersonDarling KolmogorovSmirnov	
num_bins	целое	
parameter_xml_filename	строка	
generate_parameter_import	логическое	

Свойства узла statistics



Узел Статистика предоставляет базовую сводную информацию о числовых полях. Здесь вычисляется сводная статистика для индивидуальных полей и корреляции между полями.

Таблица 195. Свойства узла statistics.

Свойства узла statistics	Тип переменной	Описание свойства
use_output_name	логический	Задаёт, используется ли пользовательское имя вывода.
output_name	строка	Если для use_output_name задано true, определяет имя для использования.
output_mode	Screen File	Используется для задания положения назначения для выходных данных, сгенерированных на узле выходных данных.
output_format	Text (.txt) HTML (.html) Output (.cou)	Используется для указания типа выхода.
full_filename	строка	
examine	[поле поле поле]	
correlate	[поле поле поле]	
statistics	[Число Среднее сумма Мин Макс Диапазон Дисперсия Среднеквадратичное отклонение Квадрат ошибки Медиана Режим]	
correlation_mode	Вероятность Абсолютная	Задаёт, как метить корреляции, по вероятности или по абсолютному значению.
label_correlations	логический	
weak_label	строка	
medium_label	строка	
strong_label	строка	
weak_below_probability	число	Когда для correlation_mode задано значение Probability, определяет значение отсечения для слабых корреляций. Это должно быть значение от 0 до 1, например, 0,90.
strong_above_probability	число	Значение отсечения для сильных корреляций.
weak_below_absolute	число	Когда для correlation_mode задано значение Absolute, определяет значение отсечения для слабых корреляций. Это должно быть значение от 0 до 1, например, 0,90.
strong_above_absolute	число	Значение отсечения для сильных корреляций.

Свойства узла statisticsoutput



Узел Вывод статистики позволяет вызвать процедуру IBM SPSS Statistics для анализа ваших данных IBM SPSS Modeler. Доступны разнообразные аналитические процедуры IBM SPSS Statistics. Этому узлу требуется лицензированная копия IBM SPSS Statistics.

Свойства этого узла описаны в разделе “Свойства узла statisticsoutput” на стр. 238.

Свойства узла table



Узел Таблица выводит данные в табличном формате, которые можно также записать в файл. Это полезно всякий раз, когда вам нужно проверить значения своих данных или экспортировать их в просто читаемую форму.

Таблица 196. Свойства узла table.

Свойства узла table	Тип переменной	Описание свойства
full_filename	строка	Имя выходного файла для диска, данных или HTML.
use_output_name	логический	Задаёт, используется ли пользовательское имя вывода.
output_name	строка	Если для use_output_name задано true, определяет имя для использования.
output_mode	Screen File	Используется для задания положения назначения для выходных данных, сгенерированных на узле выходных данных.
output_format	Formatted (.tab) Delimited (.csv) HTML (.html) Output (.cou)	Используется для указания типа выходных данных.
transpose_data	логический	Транспонирует данные перед экспортом, чтобы строки представляли поля, а столбцы - записи.
paginate_output	логический	Когда output_format - это HTML, выходные данные будут разделяться на страницы.
lines_per_page	число	При использовании с paginate_output задаёт количество строк на страницу в выходных данных.
highlight_expr	строка	
output	строка	Свойство только для чтения, содержащее ссылку на последнюю таблицу, созданную узлом.
value_labels	<i>[{строка значение метка} {строка значение метка} ...]</i>	Используется для задания меток параметрам значений.

Таблица 196. Свойства узла table (продолжение).

Свойства узла table	Тип переменной	Описание свойства
display_places	целое	Задает количество десятичных разрядов при выводе поля (применимо только к полям с системой хранения REAL). При значении -1 будут использоваться значения потока по умолчанию.
export_places	целое	Задает количество десятичных разрядов при экспорте (применимо только к полям с системой хранения REAL). При значении -1 будут использоваться значения потока по умолчанию.
decimal_separator	DEFAULT PERIOD COMMA	Задает десятичный разделитель для поля (применимо только к полям с системой хранения REAL).
date_format	"ДДММГГ" "ММДДГГ" "ГГММДД" "ГГГГММДД" "ГГГГДД" ДЕНЬ МЕСЯЦ "ДД-ММ-ГГ" "ДД-ММ-ГГГГ" "ММ-ДД-ГГ" "ММ-ДД-ГГГГ" "ДД-МЕС-ГГ" "ДД-МЕС-ГГГГ" "ГГГГ-ММ-ДД" "ДД.ММ.ГГ" "ДД.ММ.ГГГГ" "ММ.ДД.ГГ" "ММ.ДД.ГГГГ" "ДД.МЕС.ГГ" "ДД.МЕС.ГГГГ" "ДД/ММ/ГГ" "ДД/ММ/ГГГГ" "ММ/ДД/ГГ" "ММ/ДД/ГГГГ" "ДД/МЕС/ГГ" "ДД/МЕС/ГГГГ" МЕС ГГГГ к К ГГГГ нн НД ГГГГ	Задает формат даты для поля (применимо только к полям с системой хранения DATE или TIMESTAMP).
time_format	"ЧЧММСС" "ЧЧММ" "ММСС" "ЧЧ:ММ:СС" "ЧЧ:ММ" "ММ:СС" "(Ч)Ч:(М)М:(С)С" "(Ч)Ч:(М)М" "(М)М:(С)С" "ЧЧ.ММ.СС" "ЧЧ.ММ" "ММ.СС" "(Ч)Ч.(М)М.(С)С" "(Ч)Ч.(М)М" "(М)М.(С)С"	Задает формат времени для поля (применимо только к полям с системой хранения TIME или TIMESTAMP).

Таблица 196. Свойства узла table (продолжение).

Свойства узла table	Тип переменной	Описание свойства
column_width	целое	Задает ширину столбца для поля. При значении -1 для ширины столбца будет задано Auto.
justify	AUTO CENTER LEFT RIGHT	Задает выравнивание столбцов для поля.

Свойства узла transform



Узел Преобразование позволяет выбрать и предварительно просмотреть результаты преобразований, прежде чем применить их к выбранным полям.

Таблица 197. Свойства узла transform.

Свойства узла transform	Тип переменной	Описание свойства
fields	[поле1... полеn]	Поля для использования в преобразовании.
formula	All Выбрать	Обозначает, какие преобразования будут вычисляться, все или выбранные.
formula_inverse	логический	Обозначает, должно ли использоваться обратное преобразование.
formula_inverse_offset	number	Обозначает смещение данных для использования в формуле. По умолчанию задано 0, в противном случае указывается пользователем.
formula_log_n	логический	Обозначает, должно ли использоваться преобразование \log_n .
formula_log_n_offset	number	
formula_log_10	логический	Обозначает, должно ли использоваться преобразование \log_{10} .
formula_log_10_offset	number	
formula_exponential	логический	Обозначает, должно ли использоваться экспоненциальное преобразование (e^x).
formula_square_root	логический	Обозначает, должно ли использоваться преобразование квадратного корня.
use_output_name	логический	Задает, используется ли пользовательское имя вывода.
output_name	строка	Если значение поля use_output_name - это true, задает имя для использования.

Таблица 197. Свойства узла *transform* (продолжение).

Свойства узла <i>transform</i>	Тип переменной	Описание свойства
<code>output_mode</code>	Screen File	Используется для задания положения назначения для выходных данных, сгенерированных на узле выходных данных.
<code>output_format</code>	HTML (<i>.html</i>) Output (<i>.cou</i>)	Используется для указания типа выходных данных.
<code>paginate_output</code>	<i>логический</i>	Когда <code>output_format</code> - это HTML, выходные данные будут разделяться на страницы.
<code>lines_per_page</code>	<i>число</i>	При использовании с <code>paginate_output</code> задает количество строк на страницу в выходных данных.
<code>full_filename</code>	<i>строка</i>	Обозначает имя файла для использования при файловых выходных данных.

Глава 17. Свойства узла экспорта

Общие свойства узлов экспорта

Следующие свойства общие для всех узлов экспорта.

Таблица 198. Общие свойства узлов экспорта.

Свойство	Значения	Описание свойства
publish_path	строка	Введите имя rootname, которое будет использоваться для опубликованного образа и файлов параметров.
publish_metadata	логический	Задает, будет ли создаваться файл метаданных, описывающий входные и выходные данные образа и их модели.
publish_use_parameters	логический	Задает, включаются ли параметры потока в файл *.par.
publish_parameters	список строк	Задайте параметры, которые будут включены.
execute_mode	export_data publish	Задает, будет ли выполняться узел без публикации потока, или при выполнении узла поток публикуется автоматически.

Свойства узла asexport

При помощи экспорта Analytic Server (сервер аналитических служб) поток можно выполнить в файловой системе HDFS (Hadoop Distributed File System).

Таблица 199. свойства узла asexport.

Свойства узла asexport	Тип переменной	Описание свойства
data_source	строка	Имя источника данных.
export_mode	строка	Задает, что нужно присоединить (append) экспортированные данные к существующему источнику данных или перезаписать (overwrite) существующий источник данных.
host	строка	Имя хоста Analytic Server (сервер аналитических служб).
port	целое	Порт, по которому ожидает приема информации Analytic Server (сервер аналитических служб).
tenant	строка	В мультиарендной среде - имя того арендатора, которому вы принадлежите. В среде с одним арендатором используется значение по умолчанию - ibm .

Таблица 199. свойства узла *asexport* (продолжение).

Свойства узла <i>asexport</i>	Тип переменной	Описание свойства
set_credentials	логическое	Если значения аутентификации пользователя для Analytic Server (сервер аналитических служб) такие же, как на сервере SPSS Modeler, задайте для этого параметра значение false . В противном случае задайте значение true .
user_name	строка	Имя пользователя для регистрации в Analytic Server (сервер аналитических служб). Нужно только в том случае, если значение set_credentials - true.
password	строка	Пароль для регистрации в Analytic Server (сервер аналитических служб). Нужно только в том случае, если значение set_credentials - true.

Свойства узла *cognosexport*



Узел экспорта IBM Cognos BI экспортирует данные в формате, который могут прочесть базы данных Cognos BI.

Примечание: Для этого узла необходимо определить подключение Cognos и подключение ODBC.

Подключение Cognos

Свойства соединения Cognos следующие.

Таблица 200. Свойства узла *cognosexport*.

Свойства узла <i>cognosexport</i>	Тип переменной	Описание свойства
cognos_connection	{ "поле", "поле", ... ,"поле" }	Свойство списка, содержащего подробности соединения для сервера Cognos. Формат следующий: { "URL_сервера_Cognos", режим_регистрации, "пространство_имен", "имя_пользователя", "пароль" } где: <i>URL_сервера_Cognos</i> - это URL сервера Cognos, куда производится экспорт <i>режим_регистрации</i> обозначает, используется ли анонимная регистрация и может быть значением true или false; если задано true, следующие поля должны быть заданы как "" <i>пространство_имен</i> задает провайдера аутентификации защиты, используемого для регистрации на сервере <i>имя_пользователя</i> и <i>пароль</i> - это значения для регистрации на сервере Cognos

Таблица 200. Свойства узла *cognosexport* (продолжение).

Свойства узла <i>cognosexport</i>	Тип переменной	Описание свойства
<i>cognos_package_name</i>	<i>строка</i>	Путь и имя пакета Cognos, в который экспортируются данные, например: /Public Folders/MyPackage
<i>cognos_datasource</i>	<i>строка</i>	
<i>cognos_export_mode</i>	Опубликовать ExportFile	
<i>cognos_filename</i>	<i>строка</i>	

Подключение ODBC

Свойства соединения ODBC идентичны перечисленным для *databaseexport* в следующем разделе, за тем исключением, что свойство *datasource* не используется.

Свойства узла *databaseexport*



Узел экспорта баз данных записывает данные в совместимый с ODBC источник реляционных данных. Чтобы произвести запись в источник данных ODBC, этот источник данных должен существовать и у вас должны быть разрешения записи для него.

Таблица 201. Свойства узла *databaseexport*.

Свойства узла <i>databaseexport</i>	Тип переменной	Описание свойства
<i>datasource</i>	<i>строка</i>	
<i>username</i>	<i>строка</i>	
<i>password</i>	<i>строка</i>	
<i>epassword</i>	<i>строка</i>	При выполнении этот слот предназначен только для чтения. Чтобы сгенерировать закодированный пароль, используйте Инструмент паролей из меню Инструменты. Дополнительную информацию смотрите в разделе “Генерирование закодированного пароля” на стр. 47.
<i>table_name</i>	<i>строка</i>	
<i>write_mode</i>	Создать Добавлять в конец Merge	
<i>map</i>	<i>строка</i>	Отображает имя поля потока на имя столбца базы данных (допустимо только в том случае, если режим_записи - это Слияние). При слиянии все поля должны быть отображены, чтобы их можно было экспортировать. Имена полей, не существующие в базе данных, добавляются как новые столбцы.
<i>key_fields</i>	<i>[поле поле ... поле]</i>	Задаёт поле потока, используемое для ключа; свойство <i>map</i> показывает, чему оно соответствует в базе данных.

Таблица 201. Свойства узла databaseexport (продолжение).

Свойства узла databaseexport	Тип переменной	Описание свойства
join	База данных Add	
drop_existing_table	логический	
delete_existing_rows	логический	
default_string_size	целое	
type		Структурированное свойство, используемое для задания типа схемы.
generate_import	логический	
use_custom_create_table_command	логический	Используйте слот <i>пользовательское_создание_таблицы</i> для изменения стандартной команды SQL CREATE TABLE.
custom_create_table_command	строка	Задает строковую команду для использования вместо стандартной команды SQL CREATE TABLE.
use_batch	логический	Следующие свойства - это расширенные опции для массовой загрузки базы данных. Значение true для опции <i>Использовать_пакет</i> выключает принятия строки за строкой в базу данных.
batch_size	число	Задает количество записей для отправки в базу данных до принятия в память.
bulk_loading	Выкл ODBC Внешнее	Задает тип массовой загрузки. Дополнительные опции для ODBC и Внешнее перечислены ниже.
not_logged	логическое	
odbc_binding	Строка Столбец	Задайте построчное или постолбцовое связывание для массовой загрузки через ODBC.
loader_delimit_mode	Табуляция Пробел Другое	Для массовой загрузки через внешнюю программу задайте тип разделителя. Выберите Другое в сочетании со свойством <i>loader_other_delimiter</i> для указания других разделителей, например, запятой (,).
loader_other_delimiter	строка	
specify_data_file	логический	Значение флага true активирует свойство <i>файл_данных</i> ниже, где можно задать имя файла и путь для записи в него при массовой загрузке в базу данных.
data_file	строка	
specify_loader_program	логический	Значение флага true активирует свойство <i>программа_загрузчика</i> ниже, где можно задать имя файла и положение сценария или программы внешнего загрузчика.

Таблица 201. Свойства узла databaseexport (продолжение).

Свойства узла databaseexport	Тип переменной	Описание свойства
loader_program	строка	
gen_logfile	логический	Значение флага true активирует свойство имя_файла_журнала ниже, где можно задать имя файла на сервере для генерирования журнала ошибок.
logfile_name	строка	
check_table_size	логический	Флаг true позволяет проверить таблицу, чтобы увеличение размера таблицы базы данных соответствовало количеству строк, экспортированных из IBM SPSS Modeler.
loader_options	строка	Задайте дополнительные аргументы, такие как -comment и -specialdir, в программу загрузчика.
export_db_primarykey	логический	Указывает, представляет ли собой данное поле первичный ключ.
use_custom_create_index_command	логический	Если true, включает пользовательский SQL для всех индексов.
custom_create_index_command	строка	Задает команду SQL, используемую для создания индексов при включении пользовательского SQL. (Это значение можно переопределить для конкретных индексов, как описано ниже).
indexes.INDEXNAME.fields		При необходимости создает заданный индекс и перечисляет имена полей для включения в этот индекс.
indexes.INDEXNAME.use_custom_create_index_command	логический	Используется для включения или отключения пользовательского SQL для конкретного индекса.
indexes.INDEXNAME.custom_create_command		Задает пользовательский SQL, используемый для заданного индекса.
indexes.INDEXNAME.remove	логический	Если true, удаляет заданный индекс из набора индекса.
table_space	строка	Задает табличное пространство, которое будет создано.
use_partition	логический	Задает, что будет использоваться хеш-поле распределения.
partition_field	строка	Задает содержимое хеш-поля распределения.

Примечание: Для некоторых баз данных можно задать, чтобы таблицы базы данных создавались для экспорта со сжатием (например, эквивалент CREATE TABLE MYTABLE (...) COMPRESS YES; в SQL). Для поддержки этой возможности предоставляются свойства использовать_сжатие и режим_сжатия, как это описано ниже.

Таблица 202. Свойства узла экспорта базы данных (databaseexport) с использованием возможностей сжатия.

Свойства узла databaseexport	Тип переменной	Описание свойства
use_compression	логический	Если задано true, создает таблицы для экспорта со сжатием.
compression_mode	Строка Страница	Задаёт уровень сжатия для баз данных SQL Server.
	По умолчанию Direct_Load_Operations All_Operations Тип OLTP Query_High Query_Low Archive_High Archive_Low	Задаёт уровень сжатия для баз данных Oracle. Обратите внимание на то, что для использования значений OLTP, Query_High, Query_Low, Archive_High и Archive_Low требуется как минимум Oracle 11gR2.

Свойства узла datacollectionexport



Узел экспорта IBM SPSS Data Collection выводит данные в формате, используемом программным обеспечением изучения рынка IBM SPSS Data Collection. Для использования этого узла должна быть установлена библиотека данных IBM SPSS Data Collection.

Таблица 203. Свойства узла datacollectionexport.

Свойства узла datacollectionexport	Тип переменной	Описание свойства
metadata_file	строка	Имя файла метаданных для экспорта.
merge_metadata	Перезаписать MergeCurrent	
enable_system_variables	логический	Задаёт, должен ли экспортируемый файл .mdd включать в себя системные переменные IBM SPSS Data Collection.
casedata_file	строка	Имя файла .sav, в который экспортируются данные наблюдения.
generate_import	логический	

Свойства узла exceleport



Выходные данные узла экспорта Excel в формате Microsoft Excel (.xls). Дополнительно можно выбрать автоматический запуск Excel и открытие экспортированного файла при выполнении узла.

Таблица 204. Свойства узла exceleport.

Свойства узла exceleport	Тип переменной	Описание свойства
full_filename	строка	
excel_file_type	Excel2003 Excel2007	

Таблица 204. Свойства узла `excelexport` (продолжение).

Свойства узла <code>excelexport</code>	Тип переменной	Описание свойства
<code>export_mode</code>	Создать Добавлять в конец	
<code>inc_field_names</code>	<i>логический</i>	Задаёт, должны ли имена полей включаться в первую строку рабочего листа.
<code>start_cell</code>	<i>строка</i>	Задаёт начальную ячейку для экспорта.
<code>worksheet_name</code>	<i>строка</i>	Имя рабочего листа для записи.
<code>launch_application</code>	<i>логический</i>	Задаёт, должен ли Excel вызываться для итогового файла. Обратите внимание на то, что путь для запуска Excel должен быть задан в диалоговом окне Прикладные программы помощника (меню Инструменты, прикладные программы помощника).
<code>generate_import</code>	<i>логический</i>	Задаёт, должен ли быть сгенерирован узел импорта Excel, который будет читать файл экспортированных данных.

Свойства узла `outputfile`



Узел экспорта плоских файлов выводит данные в текстовом формате с разделителями. Он полезен для экспорта данных, которые может читать другое программное обеспечение анализа или электронных таблиц.

Таблица 205. Свойства узла `outputfile`.

Свойства узла <code>outputfile</code>	Тип переменной	Описание свойства
<code>full_filename</code>	<i>строка</i>	Имя выходного файла.
<code>write_mode</code>	Перезаписать Добавлять в конец	
<code>inc_field_names</code>	<i>логический</i>	
<code>use_newline_after_records</code>	<i>логический</i>	
<code>delimit_mode</code>	Запятая Табуляция Пробел Другое	
<code>other_delimiter</code>	<i>char</i>	
<code>quote_mode</code>	Нет Одиночный Двойной точности Другое	
<code>other_quote</code>	<i>логический</i>	
<code>generate_import</code>	<i>логический</i>	

Таблица 205. Свойства узла *outputfile* (продолжение).

Свойства узла <i>outputfile</i>	Тип переменной	Описание свойства
encoding	StreamDefault SystemDefault "UTF-8"	

Свойства узла *sasexport*



Узел экспорта SAS выводит данные в формате SAS, которые можно прочесть в программных пакетах SAS или SAS-совместимых. Доступно три формата файлов SAS: SAS для Windows/OS2, SAS для UNIX или SAS Версии 7/8.

Таблица 206. Свойства узла *sasexport*.

Свойства узла <i>sasexport</i>	Тип переменной	Описание свойства
формат	Windows UNIX SAS7 SAS8	Поля меток различных свойств.
full_filename	<i>строка</i>	
export_names	NamesAndLabels NamesAsLabels	Используется для отображения имен полей из IBM SPSS Modeler при экспорте на IBM SPSS Statistics или на имена переменных SAS.
generate_import	<i>логический</i>	

Свойства узла *statisticsexport*



Узел Экспорт статистики выводит данные в формате IBM SPSS Statistics *.sav*. Файлы *.sav* могут читать базовый и другие продукты IBM SPSS Statistics. Это формат, используемый также для файлов кэша в IBM SPSS Modeler.

Свойства этого узла описаны в разделе “Свойства узла *statisticsexport*” на стр. 238.

Свойства узла *xmlexport*



Узел экспорта XML выводит данные в файл в формате XML. Дополнительно вы можете создать узел источника XML, чтобы прочесть экспортированные данные обратно в поток.

Таблица 207. Свойства узла *xmlexport*.

Свойства узла <i>xmlexport</i>	Тип переменной	Описание свойства
full_filename	<i>строка</i>	(обязательно) Полный путь и имя файла экспорта XML.
use_xml_schema	<i>логический</i>	Задаёт, использовать ли схему XML (файл XSD или DTD) для управления структурой экспортированных данных.

Таблица 207. Свойства узла `xml:export` (продолжение).

Свойства узла <code>xml:export</code>	Тип переменной	Описание свойства
<code>full_schema_filename</code>	<i>строка</i>	Полный путь и имя файла XSD или DTD для использования. Обязательно, если для <code>use_xml_schema</code> задано значение <code>true</code> .
<code>generate_import</code>	<i>логический</i>	Генерирует узел источника XML, который будет читать файл экспортированных данных обратно в поток.
<code>records</code>	<i>строка</i>	Выражение XPath, обозначающее границу записи.
<code>map</code>	<i>строка</i>	Отображает имя поля на структуру XML.

Глава 18. Свойства узла IBM SPSS Statistics

Свойства узла statisticsimport



Узел Файл статистики читает данные в формате файлов *.sav*, используемом IBM SPSS Statistics, а также файлы кэша, сохраненные IBM SPSS Modeler, которые также используют тот же формат.

Таблица 208. Свойства узла statisticsimport.

Свойства узла statisticsimport	Тип переменной	Описание свойства
full_filename	строка	Полное имя файла, включающее путь.
import_names	NamesAndLabels LabelsAsNames	Способ для обработки имен и меток переменных.
import_data	DataAndLabels LabelsAsData	Способ для обработки значений и меток.
use_field_format_for_storage	Логический	Указывает, использовать ли при импорте информацию о формате полей IBM SPSS Statistics.

Свойства узла statistictransform



Узел Преобразование статистики запускает разнообразные команды синтаксиса IBM SPSS Statistics для источников данных в IBM SPSS Modeler. Этому узлу требуется лицензированная копия IBM SPSS Statistics.

Таблица 209. Свойства узла statistictransform.

Свойства узла statistictransform	Тип переменной	Описание свойства
синтаксис	строка	
check_before_saving	логический	Проверяет введенный синтаксис перед сохранением записей. Выводит сообщение об ошибке, если синтаксис неправильный.
default_include	логический	Дополнительную информацию смотрите в разделе “Свойства узла filter” на стр. 99.
include	логический	Дополнительную информацию смотрите в разделе “Свойства узла filter” на стр. 99.
new_name	строка	Дополнительную информацию смотрите в разделе “Свойства узла filter” на стр. 99.

Свойства узла statisticsmodel



Узел Статистическая модель позволяет проанализировать свои данные и работать с ними, запустив процедуры IBM SPSS Statistics, создающие PMML. Этому узлу требуется лицензированная копия IBM SPSS Statistics.

Свойства узла statisticsmodel	Тип переменной	Описание свойства
синтаксис	<i>строка</i>	
default_include	<i>логический</i>	Дополнительную информацию смотрите в разделе “Свойства узла filter” на стр. 99.
include	<i>логический</i>	Дополнительную информацию смотрите в разделе “Свойства узла filter” на стр. 99.
new_name	<i>строка</i>	Дополнительную информацию смотрите в разделе “Свойства узла filter” на стр. 99.

Свойства узла statisticsoutput



Узел Вывод статистики позволяет вызвать процедуру IBM SPSS Statistics для анализа ваших данных IBM SPSS Modeler. Доступны разнообразные аналитические процедуры IBM SPSS Statistics. Этому узлу требуется лицензированная копия IBM SPSS Statistics.

Таблица 210. Свойства узла statisticsoutput.

Свойства узла statisticsoutput	Тип переменной	Описание свойства
mode	Dialog Синтаксис	Выбирает опцию "Диалоговое окно IBM SPSS Statistics" или редактор синтаксиса
синтаксис	<i>строка</i>	
use_output_name	<i>логический</i>	
output_name	<i>строка</i>	
output_mode	Screen File	
full_filename	<i>строка</i>	
file_type	HTML SPV SPW	

Свойства узла statisticsexport



Узел Экспорт статистики выводит данные в формате IBM SPSS Statistics *.sav*. Файлы *.sav* могут читать базовый и другие продукты IBM SPSS Statistics. Это формат, используемый также для файлов кэша в IBM SPSS Modeler.

Таблица 211. Свойства узла *statisticsexport*.

Свойства узла statisticsexport	Тип переменной	Описание свойства
full_filename	<i>строка</i>	
launch_application	<i>логический</i>	
export_names	NamesAndLabels NamesAsLabels	Используется для отображения имен полей из IBM SPSS Modeler при экспорте на IBM SPSS Statistics или на имена переменных SAS.
generate_import	<i>логический</i>	

Глава 19. Свойства надузлов

В следующих таблицах описываются свойства, специфичные для надузлов. Обратите внимание на то, что общие свойства узлов применимы также к надузлам.

Таблица 212. Свойства конечных надузлов.

Имя свойства	Тип свойства/Список значений	Описание свойства
execute_method	Сценарий Нормальный	
script	<i>строка</i>	
script_language	Python Legacy	Задаёт язык сценария надузла.

Параметры надузлов

Сценарии можно использовать для создания или задания параметров надузлов, используя те же функции, что и для модификации параметров потока. Дополнительную информацию смотрите в разделе “Параметры потока, сеанса и надузла” на стр. 40.

Задание свойств для инкапсулированных узлов

Для задания свойств узлов в составе надузла нужно открыть диаграмму, принадлежащую этому надузлу, и затем использовать различные методы поиска (такие как `findByName()` и `findById()`), чтобы найти нужные узлы. Например, в сценарии надузла, содержащего только один узел типа:

```
supernode = modeler.script.supernode()
diagram = supernode.getCompositeProcessorDiagram()
# Найти узел типа на внутренней диаграмме надузла
typenode = diagram.findByName("type", None)
typenode.setKeyedProperty("direction", "Drug", "Input")
typenode.setKeyedProperty("direction", "Age", "Target")
```

Ограничения сценариев надузлов. Надузлы не могут манипулировать другими потоками и изменять текущий поток.

Приложение А. Ссылки на имена узлов

В этом разделе представлены ссылки на имена сценариев узлов в IBM SPSS Modeler.

Имена слепков моделей

На слепки моделей (также известные как сгенерированные модели) можно сослаться по типу, как и на узел и выходные объекты. В следующей таблице перечислены имена ссылок на объекты моделей.

Обратите внимание на то, что эти имена используются в частности для ссылки на слепки моделей на палитре Модели (в верхнем правом углу окна IBM SPSS Modeler). Для ссылки на узлы модели, добавленные в поток для целей скоринга, используется другой набор имен с префиксом `app1u...` Дополнительную информацию смотрите в разделе Глава 14, “Свойства узла слепков моделей”, на стр. 175.

Примечание: При обычных обстоятельствах рекомендуется, чтобы исключить путаницу, сослаться на модели по имени *и* по типу.

Таблица 213. Имена слепков моделей (палитра моделирования).

Имя модели	Модель
anomalydetection	Аномалия
apriori	Априорный анализ
autoclassifier	Автоклассификатор
autocluster	Автокластер
autonumeric	Автономерация
bayesnet	Байесовская сеть
c50	C5.0
carma	Carma
cart	C&R Tree
chaid	CHAID
coxreg	Регрессия Кокса
decisionlist	Список решений
discriminant	Дискриминантный
factor	РСА/фактор
featureselection	Отбор показателей
genlin	Обобщенная линейная регрессия
glm	GLMM
kmeans	К-средних
knn	<i>k</i> ближайших соседей
kohonen	Коонена
Линейная	Линейное
logreg	Логистическая регрессия
neuralnetwork	Нейросеть
quest	QUEST
регрессия	Линейная регрессия

Таблица 213. Имена слепков моделей (палитра моделирования) (продолжение).

Имя модели	Модель
sequence	Порядковый номер
slrm	Самообучаемая модель откликов
statisticsmodel	Модель IBM SPSS Statistics
svm	Механизм опорных векторов
timeseries	Временные ряды
twostep	TwoStep

Таблица 214. Имена слепков моделей (палитра моделирования базы данных).

Имя модели	Модель
db2imcluster	Кластеризация IBM ISW
db2imlog	Логистическая регрессия IBM ISW
db2imnb	Наивный критерий Байеса IBM ISW
db2imreg	Регрессия IBM ISW
db2imtree	Дерево решений IBM ISW
msassoc	Правила связывания MS
msbayes	Наивный критерий Байеса MS
mscluster	Кластеризация MS
mslogistic	Логистическая регрессия MS
msneuralnetwork	Нейросеть MS
msregression	Линейная регрессия MS
mssequencecluster	Кластеризация последовательностей MS
mstimeseries	Временные ряды MS
mstree	Дерево решений MS
netezزابayes	Байесовская сеть Netezza
netezзадectree	Дерево решений Netezza
netezзадdivcluster	Разделительная кластеризация Netezza
netezзаglm	Обобщенный линейный анализ Netezza
netezзаkmeans	K-средние Netezza
netezзаknn	KNN Netezza
netezзаlineregression	Линейная регрессия Netezza
netezзаnaivebayes	Наивный байесовский анализ Netezza
netezзаpca	PCA Netezza
netezзаregtree	Дерево регрессии Netezza
netezзаtimeseries	Временные ряды Netezza
oraabn	Адаптивный критерий Байеса Oracle
oraai	AI Oracle
oradecisiontree	Дерево решений Oracle
oraglm	ОЛМ Oracle
orakmeans	k-средние Oracle
oranb	Наивный критерий Байеса Oracle

Таблица 214. Имена слепков моделей (палитра моделирования базы данных) (продолжение).

Имя модели	Модель
oranmf	NMF Oracle
oracluster	О-кластер Oracle
orasvm	SVM Oracle

Исключение дублирования имен моделей

При использовании сценариев для работы со сгенерированными моделями имейте в виду, что разрешение дублирования имен моделей может привести к неоднозначным ссылкам. Для исключения этого рекомендуется при написании сценариев потребовать уникальности имен для сгенерированных моделей.

Чтобы задать опции для дублирования имен моделей:

1. Выберите в меню:
Инструменты > Пользовательские опции
2. Щелкните по вкладке **Уведомления**.
3. Выберите **Заменить предыдущую модель**, чтобы ограничить дублирование имен для сгенерированных моделей.

Поведение выполнения сценариев может различаться для SPSS Modeler и IBM SPSS Collaboration and Deployment Services, когда есть неоднозначные ссылки на модель. Клиент SPSS Modeler включает в себя опцию "Заменить предыдущую модель", которая автоматически заменяет модели с одинаковыми именами (например, когда в сценарии происходит цикл для создания всякий раз новой модели). Однако эта опция недоступна, когда тот же сценарий запущен в IBM SPSS Collaboration and Deployment Services. Исключить эту ситуацию можно, или переименовав модель, сгенерированную при каждой итерации, чтобы исключить неоднозначные ссылки на модели, или очистив текущую модель (например, добавляя оператор `clear generated palette`) перед окончанием цикла.

Имена типов вывода

В следующей таблице перечислены все типы объектов вывода и узлы, которые их создали. Полный список форматов экспорта, доступных для каждого типа объектов вывода, смотрите в описании свойств для узла, создающего тип вывода, в разделах "Общие свойства узла графика" на стр. 115 и Глава 16, "Свойства узлов вывода", на стр. 213.

Таблица 215. Типы объектов вывода и создающие их узлы.

Тип объектов вывода	Узел
analysisoutput	Анализ
collectionoutput	Собрание
dataauditoutput	Аудит данных
distributionoutput	Распределение
evaluationoutput	Оценка
histogramoutput	Гистограмма
matrixoutput	Матрица
meansoutput	Средние
multiplotoutput	Несколько графиков
plotoutput	График
qualityoutput	Качество

Таблица 215. Типы объектов вывода и создающие их узлы (продолжение).

Тип объектов вывода	Узел
reportdocumentoutput	Этот тип объектов создан не узлом, такой вывод создается отчетом проекта
reportoutput	Отчет
statisticsprocedureoutput	Вывод Statistics
statisticsoutput	Статистики
tableoutput	Таблицу
timeplotoutput	график зависимости от времени
weboutput	Web

Приложение В. Перенастройка от унаследованных сценарием к сценариям Python

Обзор перенастройки унаследованных сценариев

В этом разделе дается обзор различий между сценариями Python и унаследованными сценариями в IBM SPSS Modeler, а также информация о том, как перенастроить унаследованные сценарии в сценарии Python. В этом разделе вы найдете список унаследованных стандартных команд SPSS Modeler и эквивалентных команд Python.

Общие отличия

Во многом структура унаследованных сценариев происходит от командных сценариев операционной системы. Унаследованные сценарии ориентированы на строки, и отступы обычно не играют роли, несмотря на то, что используются некоторые блочные структуры, например, `if...then...else...endif` и `for...endfor`.

В сценариях Python отступы важны, и у строк одного логического блока должен быть один и тот же уровень отступа.

Примечание: Будьте осторожны при копировании кода Python. Строка с отступами, введенными клавишей Tab, может выглядеть в редакторе так же, как строка с отступами, введенными клавишей пробела. Однако сценарий Python сгенерирует ошибку, поскольку он не считает такие отступы строк одинаковыми.

Контекст сценариев

Контекст сценариев определяет среду, в которой выполняется сценарий, например, поток или надузел, вызывающий сценарий. В унаследованных сценариях контекст задается неявно; например, ссылки на узлы в сценарии потока считаются ссылками на узлы в потоке, запустившем сценарий.

В сценариях Python контекст сценариев задается явным образом через модуль `modeler.script`. Например, сценарий потока Python может обратиться к потоку, который выполняет сценарий с таким кодом:

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

После этого через возвращенный объект можно вызывать функции, связанные с потоком.

Команды или функции

Унаследованные сценарии ориентированы на команды. Это значит, что обычно строка сценария начинается с выполняемой команды, за которой следуют параметры, например:

```
connect 'Type':typenode to :filternode  
rename :derivenode as "Compute Total"
```

Python uses functions that are usually invoked through an object (a module, class or object) that defines the function, for example:

```
stream = modeler.script.stream()  
typenode = stream.findByName("type", "Type")  
filternode = stream.findByName("filter", None)  
stream.link(typenode, filternode)  
derive.setLabel("Compute Total")
```

Литералы и комментарии

Для некоторых команд литералов и комментариев, широко используемых в IBM SPSS Modeler, есть эквивалентные команды в сценариях Python. Это может помочь при преобразовании ваших существующих сценариев SPSS Modeler прежних версий в сценарии Python для использования в IBM SPSS Modeler 16.

Таблица 216. Отображение унаследованных сценариев на сценарии Python для литералов и комментариев.

Унаследованный сценарий	Сценарий Python
Целое число, например, 4	То же
Число с плавающей точкой, например, 0.003	То же
Строки в одинарных кавычках, например 'Привет'	То же Примечание: Перед строковыми литералами, содержащими не входящие в ASCII символы, надо указывать <code>u</code> , чтобы они интерпретировались как символы Unicode.
Строки в двойных кавычках, например, "Привет еще раз"	То же Примечание: Перед строковыми литералами, содержащими не входящие в ASCII символы, надо указывать <code>u</code> , чтобы они интерпретировались как символы Unicode.
Длинные строки, например """Это строковое значение, которое занимает несколько строк"""	То же
Списки, например, [1 2 3]	[1, 2, 3]
Ссылка на переменную, например, set x = 3	x = 3
Продолжение строки (<code>\</code>), например set x = [1 2 \ 3 4]	x = [1, 2,\n3, 4]
Блок комментариев, например /* Это длинный комментарий с переносом на другую строку. */	/* Это длинный комментарий с переносом на другую строку. */
Однострочные комментарии, например, set x = 3 # задать для x значение 3	x = 3 # задать для x значение 3
undef	Нет
true	True
false	False

Операторы

Для некоторых операционных команд, широко используемых в IBM SPSS Modeler, есть эквивалентные команды в сценариях Python. Это может помочь при преобразовании ваших существующих сценариев SPSS Modeler прежних версий в сценарии Python для использования в IBM SPSS Modeler 16.

Таблица 217. Отображение унаследованных сценариев на сценарии Python для операций.

Унаследованный сценарий	Сценарий 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

Таблица 217. Отображение унаследованных сценариев на сценарии Python для операций (продолжение).

Унаследованный сценарий	Сценарий Python
NUM1 / NUM2	NUM1 / NUM2
= ==	==
/= /==	!=
X ** Y	X ** Y
X < Y X <= Y X > Y X >= Y	X < Y X <= Y X > Y X >= Y
X div Y X rem Y X mod Y	X // Y X % Y X % Y
и или not(EXPR)	и или not EXPR

Условное выполнение и циклы

Для некоторых команд условного выполнения и циклов, широко используемых в IBM SPSS Modeler, есть эквивалентные команды в сценариях Python. Это может помочь при преобразовании ваших существующих сценариев SPSS Modeler прежних версий в сценарии Python для использования в IBM SPSS Modeler 16.

Таблица 218. Отображение унаследованных сценариев на сценарии Python для условных выполнений и циклов.

Унаследованный сценарий	Сценарий Python
for VAR from INT1 to INT2 ... endfor	for VAR in range(INT1, INT2): ... или 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	Нет эквивалента

Таблица 218. Отображение унаследованных сценариев на сценарии Python для условных выполнений и циклов (продолжение).

Унаследованный сценарий	Сценарий Python
var VAR1	Объявление переменной не требуется

Переменные

В унаследованных сценариях переменные объявляются до их использования, например:

```
var mynode
set mynode = create typenode at 96 96
```

В сценариях Python переменные создаются при первом использовании, например:

```
mynode = stream.createAt("type", "Type", 96, 96)
```

В унаследованных сценариях ссылки на переменные нужно удалять явным образом при помощи операции ^, например:

```
var mynode
set mynode = create typenode at 96 96
set ^mynode.direction."Age" = Input
```

В сценариях Python, как и в большинстве языков сценариев, такой необходимости нет, например:

```
mynode = stream.createAt("type", "Type", 96, 96)
mynode.setKeyedPropertyValue("direction", "Age", "Input")
```

Типы узлов, объектов вывода и моделей

В унаследованных сценариях к конкретному типу узла, объекта вывода и модели обычно добавляется общий тип (node, output и model). Например, узел вычислений имеет тип `derivenode`:

```
set feature_name_node = create derivenode at 96 96
```

В API Python IBM SPSS Modeler не добавляет суффикс `node`, так что тип узла вычислений - `derive`, например:

```
feature_name_node = stream.createAt("derive", "Feature", 96, 96)
```

Отсутствие суффикса типа - единственное различие между именами типов в унаследованных сценариях и языке сценариев Python.

Имена свойств

Имена свойств в унаследованных сценариях и в сценариях Python одни и те же. Так, на узле файла переменных свойство, задающее положение файла, называется `full_filename` в обеих языковых средах.

Ссылки на узлы

Во многих унаследованных сценариях для доступа к редактируемому узлу используется неявный поиск. Например, приведенные ниже команды ищут в текущем потоке узел типа с меткой "Type", а затем задают направление (или роль моделирования) поля "Age" как Входное поле и поля "Drug" - как Поле назначения, то есть предсказываемое:

```
set 'Type':typenode.direction."Age" = Input
set 'Type':typenode.direction."Drug" = Target
```

В сценариях Python поиск объектов нужно выполнять явно и до того, как вызывается функция, задающая значение свойства, например:

```
typenode = stream.findByType("type", "Type")
typenode.setKeyedPropertyValue("direction", "Age", "Input")
typenode.setKeyedPropertyValue("direction", "Drug", "Target")
```

Примечание: В этом случае "Target" берется в кавычки.

Другой вариант - в сценариях Python может использоваться нумерация `ModelingRole` в пакете `modeler.api`.

Сценарии Python могут получаться более громоздкими, зато производительность при их выполнении выше, поскольку поиск узла в них, как правило, производится только один раз. В примере унаследованных сценариев поиск узла выполняется для каждой команды.

Поддерживается также поиск узлов по ID (ID узла можно видеть в диалоговом окне узла на вкладке аннотаций). Например, в унаследованных сценариях:

```
# id65EMPB9VL87 - это ID узла типа
set @id65EMPB9VL87.direction."Age" = Input
```

В следующем сценарии показан пример на языке Python:

```
typenode = stream.findByID("id65EMPB9VL87")
typenode.setKeyedPropertyValue("direction", "Age", "Input")
```

Получение и задание свойств

Унаследованные сценарии используют команду `set` для задания значения. Элемент после команды `set` может быть определением свойства. В приведенном ниже сценарии показано два возможных формата для задания свойства в сценарии:

```
set <ссылка на узел>.<свойство> = <значение>
set <ссылка на узел>.<ключевое свойство>.<ключ> = <значение>
```

В сценариях Python тот же результат достигается использованием функций `setProperty()` и `setKeyedPropertyValue()`, например:

```
object.setProperty(property, value)
object.setKeyedPropertyValue(keyed-property, key, value)
```

В унаследованных сценариях для доступа к значениям свойств можно было использовать команду `get`, например:

```
var n v
set n = get node :filternode
set v = ^n.name
```

В сценариях Python тот же результат достигается использованием функции `getProperty()`, например:

```
n = stream.findByName("filter", None)
v = n.getProperty("name")
```

Редактирование потоков

В унаследованных сценариях для создания нового узла служит команда `create`, например:

```
var agg select
set agg = create aggregatenode at 96 96
set select = create selectnode at 164 96
```

В сценариях Python есть ряд методов для создания узлов в потоках, например:

```
stream = modeler.script.stream()
agg = stream.createAt("aggregate", "Aggregate", 96, 96)
select = stream.createAt("select", "Select", 164, 96)
```

В унаследованных сценариях для создания связей между узлами служит команда `connect`, например:

```
connect ^agg to ^select
```

В сценариях Python для создания связей между узлами служит метод `link`, например:

```
stream.link(agg, select)
```

В унаследованных сценариях для удаления связей между узлами служит команда `disconnect`, например:

```
disconnect ^agg from ^select
```

В сценариях Python для удаления связей между узлами служит метод `unlink`, например:

```
stream.unlink(agg, select)
```

В унаследованных сценариях для размещения узлов на холсте потока или между другими узлами служит команда `position`, например:

```
position ^agg at 256 256  
position ^agg between ^myselect and ^mydistinct
```

В сценариях Python тот же результат достигается использованием двух различных методов - `setXYPosition` и `setPositionBetween`. Например:

```
agg.setXYPosition(256, 256)  
agg.setPositionBetween(myselect, mydistinct)
```

Операции с узлами

Для некоторых команд операций с узлами, широко используемых в IBM SPSS Modeler, есть эквивалентные команды в сценариях Python. Это может помочь при преобразовании ваших существующих сценариев SPSS Modeler прежних версий в сценарии Python для использования в IBM SPSS Modeler 16.

Таблица 219. Отображение унаследованных сценариев на сценарии Python для операций с узлами.

Унаследованный сценарий	Сценарий Python
<code>create nodespec at x y</code>	<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>
<code>connect fromNode to toNode</code>	<code>stream.link(fromNode, toNode)</code>
<code>delete node</code>	<code>stream.delete(node)</code>
<code>disable node</code>	<code>stream.setEnabled(node, False)</code>
<code>enable node</code>	<code>stream.setEnabled(node, True)</code>
<code>disconnect fromNode from toNode</code>	<code>stream.unlink(fromNode, toNode)</code> <code>stream.disconnect(node)</code>
<code>duplicate node</code>	<code>node.duplicate()</code>
<code>execute node</code>	<code>stream.runSelected(nodes, results)</code> <code>stream.runAll(results)</code>
<code>flush node</code>	<code>node.flushCache()</code>
<code>position node at x y</code>	<code>node.setXYPosition(x, y)</code>
<code>position node between node1 and node2</code>	<code>node.setPositionBetween(node1, node2)</code>
<code>rename node as name</code>	<code>node.setLabel(name)</code>

Циклы

В унаследованных сценариях поддерживаются следующие две основных опции:

- *Циклы с подсчетом*, в которых переменная индекса изменяется в диапазоне между двумя целыми числами.

- *Циклы последовательности*, в которых перебирается последовательность значений, которые привязывают текущее значение к переменной цикла.

Следующий сценарий служит примером цикла с подсчетом в унаследованных сценариях:

```
for i from 1 to 10
  println ^i
endfor
```

Следующий сценарий служит примером цикла последовательности в унаследованных сценариях:

```
var items
set items = [a b c d]

for i in items
  println ^i
endfor
```

Доступны и другие типы циклов:

- Итерация по моделям на палитре моделей или по объектам вывода на палитре объектов вывода.
- Итерация по входным или выходным полям узла.

В сценариях Python тоже поддерживаются различные типы циклов. Следующий сценарий служит примером цикла с подсчетом в языке сценариях Python:

```
i = 1
while i <= 10:
  print i
  i += 1
```

Следующий сценарий служит примером цикла последовательности в сценариях Python:

```
items = ["a", "b", "c", "d"]
for i in items:
  print i
```

Цикл последовательности - весьма гибкая конструкция, а в сочетании с методами API IBM SPSS Modeler способен поддерживать большинство ситуаций в унаследованных сценариях. Приведенный ниже пример показывает, как использовать цикл последовательности в сценариях Python для итерации по выходным полям узла:

```
node = modeler.script.stream().findByType("filter", None)
for column in node.getOutputDataModel().columnIterator():
  print column.getColumnname()
```

ВЫПОЛНЕНИЕ ПОТОКОВ

Во время выполнения потока генерируемые объекты моделей и выходные объекты добавляются в один из менеджеров объектов. В унаследованных сценариях построенные объекты нужно либо найти в менеджере объектов, либо вызвать как последний сгенерированный вывод соответствующего узла.

В сценариях Python выполнение потока отличается в том отношении, что все модели и выходные объекты, сгенерированные при выполнении потока, возвращаются в виде списка, передаваемого внешней функции. Это упрощает доступ к результатам выполнения потока.

В унаследованных сценариях поддерживаются три команды выполнения потока:

- `execute_all` выполняет все выполняемые конечные узлы потока.
- `execute_script` выполняет сценарий потока независимо от того, задано ли выполнение сценариев.
- `execute узел` выполняет указанный узел.

В сценариях Python поддерживается аналогичный набор функций:

- `поток.runAll` (*список-результатов*) выполняет все выполняемые конечные узлы в потоке.
- `поток.runScript` (*список-результатов*) выполняет сценарий потока независимо от того, задано ли выполнение сценариев.
- `поток.runSelected` (*массив-узлов, список-результатов*) выполняет заданный набор узлов в том порядке, в котором они заданы.
- `поток.run` (*список-результатов*) выполняет указанный узел.

В унаследованных сценариях можно завершить выполнение потока командой `exit` с необязательным целочисленным кодом, например:

```
exit 1
```

В языке Python того же результата можно достичь таким сценарием:

```
modeler.script.exit(1)
```

Доступ к объектам через файловую систему и репозиторий

В унаследованных сценариях можно открыть существующий поток, модель или выходной объект командой `open`, например:

```
var s
set s = open stream "c:/my streams/modeling.str"
```

В языке сценариев Python есть класс `TaskRunner`, доступный из сеанса и поддерживающий аналогичные задачи, например:

```
taskrunner = modeler.script.session().getTaskRunner()
s = taskrunner.openStreamFromFile("c:/my streams/modeling.str", True)
```

Чтобы сохранить объект в унаследованных сценариях, можно использовать команду `save`, например:

```
save stream s as "c:/my streams/new_modeling.str"
```

Эквивалентный подход в языке сценариев Python - использовать класс `TaskRunner`, например:

```
taskrunner.saveStreamToFile(s, "c:/my streams/new_modeling.str")
```

Операции на основе IBM SPSS Collaboration and Deployment Services Repository поддерживаются в унаследованных сценариях через команды `retrieve` и `store`, например:

```
var s
set s = retrieve stream "/my repository folder/my_stream.str"
store stream ^s as "/my repository folder/my_stream_copy.str"
```

В языке сценариев Python более удобен доступ к эквивалентным возможностям через объект репозитория, связанный с сеансом:

```
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)
```

Примечание: Для доступа к репозиторию требуется, чтобы сеанс был сконфигурирован с использованием допустимого соединения с репозиторием.

Операции с потоками

Для некоторых команд операций с потоками, широко используемых в IBM SPSS Modeler, есть эквивалентные команды в сценариях Python. Это может помочь при преобразовании ваших существующих сценариев SPSS Modeler прежних версий в сценарии Python для использования в IBM SPSS Modeler 16.

Таблица 220. Отображение унаследованных сценариев на сценарии Python для операций с потоками.

Унаследованный сценарий	Сценарий 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>	Нет эквивалента
load stream <i>path</i>	Нет эквивалента
open stream <i>path</i>	<i>taskrunner.openStreamFromFile(path, autoManage)</i>
save <i>stream</i> as <i>path</i>	<i>taskrunner.saveStreamToFile(stream, path)</i>
retrieve stream <i>path</i>	<i>repository.retrieveStream(path, version, label, autoManage)</i>
store <i>stream</i> as <i>path</i>	<i>repository.storeStream(stream, path, label)</i>

Операции с моделями

Для некоторых команд операций с моделями, широко используемых в IBM SPSS Modeler, есть эквивалентные команды в сценариях Python. Это может помочь при преобразовании ваших существующих сценариев SPSS Modeler прежних версий в сценарии Python для использования в IBM SPSS Modeler 16.

Таблица 221. Отображение унаследованных сценариев на сценарии Python для операций с моделями.

Унаследованный сценарий	Сценарий 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>

Операции вывода документов

Для некоторых команд операций вывода документов, широко используемых в IBM SPSS Modeler, есть эквивалентные команды в сценариях Python. Это может помочь при преобразовании ваших существующих сценариев SPSS Modeler прежних версий в сценарии Python для использования в IBM SPSS Modeler 16.

Таблица 222. Отображение унаследованных сценариев на сценарии Python для операций вывода документов.

Унаследованный сценарий	Сценарий 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>

Другие различие между унаследованными сценариями и сценариями Python

В унаследованных сценариях есть поддержка работы с проектами IBM SPSS Modeler. Сценарии Python в настоящее время это не поддерживают.

В унаследованных сценариях до некоторой степени поддерживается загрузка *объектов состояния* (сочетание потоков и моделей). Объекты состояния устарели, начиная с версии IBM SPSS Modeler 8.0. Сценарии Python не поддерживают объекты состояния.

В сценариях Python предлагаются следующие дополнительные возможности, недоступные в унаследованных сценариях:

- Определения классов и функций
- Обработка ошибок
- Более современная поддержка ввода-вывода
- Внешние модули и модули других производителей

Уведомления

Эта информация относится к продуктам и сервису, предлагаемым по всему миру.

IBM может не предоставлять в других странах продукты, услуги и аппаратные средства, описанные в данном документе. За информацией о продуктах и услугах, предоставляемых в вашей стране, обращайтесь к местному представителю IBM. Ссылки на продукты, программы или услуги IBM не означают и не предполагают, что можно использовать только указанные продукты, программы или услуги IBM. Разрешается использовать любые функционально эквивалентные продукты, программы или услуги, если при этом не нарушаются права IBM на интеллектуальную собственность. Однако ответственность за оценку и проверку работы любого продукта, программы или сервиса, не произведенного корпорацией IBM, лежит на пользователе.

IBM может располагать патентами или рассматриваемыми заявками на патенты, относящимися к предмету данного документа. Предъявление данного документа не предоставляет какую-либо лицензию на эти патенты. Вы можете послать письменный запрос о лицензии по адресу:

IBM Director of Licensing
IBM Corporation
North Castle Drive
Armonk, NY 10504-1785
U.S.A.

По поводу лицензий, связанных с использованием наборов двухбайтных символов (DBCS), обращайтесь в отдел интеллектуальной собственности IBM в вашей стране или направьте запрос в письменной форме по адресу:

Intellectual Property Licensing
Legal and Intellectual Property Law
IBM Japan Ltd.
1623-14, Shimotsuruma, Yamato-shi
Kanagawa 242-8502 Japan

Следующий абзац не применяется в Великобритании или в любой другой стране, где подобные заявления противоречат местным законам: INTERNATIONAL BUSINESS MACHINES CORPORATION ПРЕДСТАВЛЯЕТ ДАННУЮ ПУБЛИКАЦИЮ "КАК ЕСТЬ", БЕЗ КАКИХ-ЛИБО ГАРАНТИЙ, КАК ЯВНЫХ, ТАК И ПОДРАЗУМЕВАЕМЫХ, ВКЛЮЧАЯ, НО НЕ ОГРАНИЧИВАЯСЬ ТАКОВЫМИ, ПРЕДПОЛАГАЕМЫЕ ГАРАНТИИ СОБЛЮДЕНИЯ ЧЬИХ-ЛИБО АВТОРСКИХ ПРАВ, ВОЗМОЖНОСТИ КОММЕРЧЕСКОГО ИСПОЛЬЗОВАНИЯ ИЛИ ПРИГОДНОСТИ ДЛЯ КАКИХ-ЛИБО ЦЕЛЕЙ И СООТВЕТСТВИЯ ОПРЕДЕЛЕННОЙ ЦЕЛИ. В некоторых штатах при определенных соглашениях не допускается отказ от выраженных или подразумеваемых гарантий, поэтому данное заявление может к вам не относиться.

Эта информация может содержать технические неточности и типографские ошибки. В представленную здесь информацию периодически вносятся изменения; эти изменения будут включаться в новые издания данной публикации. Фирма IBM может в любое время без уведомления вносить изменения и усовершенствования в продукты и программы, описанные в этой публикации.

Любые приводимые здесь ссылки на web-сайты, не относящиеся к компании IBM, даются исключительно для удобства и ни в коей мере не служат целям поддержки или рекламы этих web-сайтов. Материалы этих Web-сайтов не являются частью данного продукта IBM, и вы можете использовать их только на собственную ответственность.

Любую предоставленную вами информацию IBM может использовать или распространять любым способом, какой сочтет нужным, не беря на себя никаких обязательств по отношению к вам.

Если обладателю лицензии на данную программу понадобятся сведения о возможности: (i) обмена данными между независимо разработанными программами и другими программами (включая данную) и (ii) совместного использования таких данных, он может обратиться по адресу:

IBM Software Group
ATTN: Licensing
200 W. Madison St.
Chicago, IL; 60606
U.S.A.

Такая информация может быть доступна при соответствующих условиях и соглашениях, включая в некоторых случаях взимание платы.

Описанную в данном документе лицензионную программу и все прилагаемые к ней лицензированные материалы IBM предоставляет на основе положений Соглашения между IBM и Заказчиком, Международного Соглашения о Лицензиях на Программы IBM или любого эквивалентного соглашения между IBM и заказчиком.

Любые данные о выполнении, содержащиеся здесь, были определены в контролируемой среде. Поэтому результаты, полученные в других операционных средах, могут существенно отличаться. Некоторые измерения могли быть сделаны на системах в стадии разработки, и поэтому нет гарантии, что соответствующие показатели останутся теми же на общедоступных системах. Более того, некоторые показатели могли быть оценены путем экстраполяции. Реальные результаты могут отличаться. Пользователи этого документа должны проверить приводимые данные в их конкретной среде.

Информация о продуктах, не принадлежащих компании IBM, была получена от поставщиков этих продуктов, из их опубликованных сообщений или других общедоступных источников. Компания IBM не тестировала эти продукты и не может подтвердить правильность их работы, совместимость и другие утверждения, касающиеся продуктов, не принадлежащих компании IBM. Вопросы о возможностях этих продуктов следует направлять их поставщикам.

Все заявления, касающиеся будущих направлений деятельности или намерений корпорации IBM, подвержены изменению или отмене без предупреждения и являются не более чем выражением целей или намерений.

Эти сведения содержат примеры данных и отчетов, используемых в повседневных деловых операциях. Чтобы проиллюстрировать их настолько полно, насколько это возможно, данные примеры включают имена индивидуумов, названия компаний, брендов и продуктов. Все эти имена и названия являются вымышленными, и любое совпадение с названиями и адресами, используемыми реально действующими компаниями, является чисто случайными.

При просмотре данного электронного информационного документа фотографии и цветные иллюстрации могут не показываться.

Товарные знаки

IBM, логотип IBM, и ibm.com являются товарными знаками или зарегистрированными товарными знаками компании International Business Machines Corp., зарегистрированными во многих странах мира. Прочие наименования продуктов и услуг могут быть товарными знаками, принадлежащими IBM или другим компаниям. Текущий список товарных знаков IBM можно найти в Интернете “Copyright and trademark information” по адресу: www.ibm.com/legal/copytrade.shtml.

Intel, логотип Intel, Intel Inside, логотип Intel Inside, Intel Centrino, логотип Intel Centrino, Celeron, Intel Xeon, Intel SpeedStep, Itanium и Pentium являются товарными знаками или зарегистрированными товарными знаками компании Intel или ее дочерних компаний в Соединенных Штатах и других странах.

Linux является зарегистрированным товарным знаком Linus Torvalds в Соединенных Штатах и других странах.

Microsoft, Windows, Windows NT и логотип Windows являются товарными знаками корпорации Microsoft в Соединенных Штатах и других странах.

UNIX является зарегистрированным товарным знаком The Open Group в Соединенных Штатах и других странах.

Java и все основанные на Java товарные знаки и логотипы - товарные знаки или зарегистрированные товарные знаки Oracle и/или его филиалов.

Другие названия продуктов и услуг могут являться товарными знаками IBM или других компаний.

Индекс

A

- API сценариев
 - автономные сценарии 45
 - введение 35
 - глобальные значения 44
 - доступ к сгенерированным объектам 38
 - метаданные 35
 - несколько потоков 45
 - обработка ошибок 39
 - параметры надузла 40
 - параметры потока 40
 - параметры сеанса 40
 - поиск 35
 - пример 35

I

- IBM SPSS Modeler
 - выполнение из командной строки 49

J

- Jython 13

M

- model nuggets
 - имена сценариев 243, 245
 - Свойства node scripting 175

N

- Netezza K-Means models
 - Свойства node scripting 201, 210

O

- O-кластер Oracle
 - Свойства node scripting 188, 194

P

- Python 13
 - сценарии 14

A

- автоматическая подготовка данных
 - свойства 91
- автономные сценарии 1, 3, 25
- адаптивные модели Байеса Oracle
 - Свойства node scripting 188, 194
- априорные модели
 - Свойства node scripting 129, 175
- аргументы
 - system 50

- аргументы (*продолжение*)
 - командный файл 53
 - соединение с репозиторием IBM SPSS Collaboration and Deployment Services 52
 - соединение с сервером 51

B

- Байесовские модели сети
 - Свойства node scripting 134
- блоки кода 17

V

- Временные ряды MS
 - Свойства node scripting 187
- выполнение потоков 25
- выполнение сценариев 9

D

- двухшаговые модели
 - Свойства node scripting 172, 184
- Дерево решений MS
 - Свойства node scripting 185, 187
- диаграммы 25
- дискриминантные модели
 - Свойства node scripting 144, 179
- добавление атрибутов 22

3

- задание свойств 28
- закодированные пароли
 - добавление к сценариям 47
- защита
 - закодированные пароли 47, 51

I

- идентификаторы 17
- изменение потоков 29, 31
- исходные узлы
 - свойства 61
- исходный узел IBM Cognos BI
 - свойства 63
- Исходный узел IBM SPSS Data Collection
 - свойства 65

K

- Кластеризация последовательностей MS
 - Свойства node scripting 187
- ключ итерации
 - циклы в сценариях 5
- команда clear generated palette 48

- командная строка
 - выполнение IBM SPSS Modeler 49
 - несколько аргументов 53
 - параметры 51
 - список аргументов 50, 51, 52
 - сценарии 48
- комментарии 16

L

- Линейная регрессия MS
 - Свойства node scripting 185, 187
- линейные модели
 - Свойства node scripting 157, 181
- Логистическая регрессия MS
 - Свойства node scripting 185, 187

M

- математические методы 19
- модели
 - имена сценариев 243, 245
- модели AI Oracle
 - Свойства node scripting 188
- модели C5.0
 - Свойства node scripting 136, 177
- модели CARMA
 - Свойства node scripting 137, 177
- модели CHAID
 - Свойства node scripting 139, 178
- модели GLMM
 - Свойства node scripting 151, 180
- модели IBM DB2
 - Свойства node scripting 195
- Модели IBM SPSS Statistics
 - Свойства node scripting 238
- модели k-средних
 - Свойства node scripting 155, 180
- модели K-средних Oracle
 - Свойства node scripting 188, 194
- модели KNN
 - Свойства node scripting 181
- модели KNN Netezza
 - Свойства node scripting 201, 210
- модели MDL Oracle
 - Свойства node scripting 188, 194
- модели Microsoft
 - Свойства node scripting 185, 187
- модели Netezza
 - Свойства node scripting 201
- модели NMF Oracle
 - Свойства node scripting 188, 194
- модели Oracle
 - Свойства node scripting 188
- модели PCA
 - Свойства node scripting 145, 179
- модели PCA Netezza
 - Свойства node scripting 201, 210
- модели PCA/факторов
 - Свойства node scripting 145, 179

- Модели QUEST
 - Свойства node scripting 165, 182
- модели SLRM
 - Свойства node scripting 168, 183
- модели SVM
 - Свойства node scripting 169
- модели автоклассификации
 - Свойства node scripting 176
- модели автокластеризации
 - Свойства node scripting 176
- модели автонумерации
 - Свойства node scripting 133, 176
- модели априори Oracle
 - Свойства node scripting 188, 194
- модели Байесовской сети
 - Свойства node scripting 177
- модели Байесовской сети Netezza
 - Свойства node scripting 201, 210
- модели ближайших соседей
 - Свойства node scripting 155
- модели временных рядов
 - Свойства node scripting 170, 184
- модели временных рядов IBM ISW
 - Свойства node scripting 195
- модели временных рядов Netezza
 - Свойства node scripting 201
- модели выбора возможностей
 - Свойства node scripting 146, 179
- модели выявления аномалий
 - Свойства node scripting 175
- модели дерева C&R
 - Свойства node scripting 138, 177
- модели дерева регрессии Netezza
 - Свойства node scripting 201, 210
- модели дерева решений IBM ISW
 - Свойства node scripting 195, 200
- модели дерева решений Oracle
 - Свойства node scripting 188, 194
- модели деревьев решений Netezza
 - Свойства node scripting 201, 210
- модели кластеризации IBM ISW
 - Свойства node scripting 195, 200
- модели Коонена
 - Свойства node scripting 157, 181
- модели линейной регрессии
 - Свойства node scripting 166, 183
- модели линейной регрессии Netezza
 - Свойства node scripting 201, 210
- модели логистической регрессии
 - Свойства node scripting 159, 181
- модели логистической регрессии IBM ISW
 - Свойства node scripting 195, 200
- модели механизмов опорных векторов
 - Свойства node scripting 169, 184
- модели механизмов опорных векторов Oracle
 - Свойства node scripting 188, 194
- модели нейронной сети
 - Свойства node scripting 161, 181
- модели обнаружения аномалий
 - Свойства node scripting 128
- Модели откликов самообучения
 - Свойства node scripting 168, 183
- модели последовательностей
 - Свойства node scripting 183
- модели последовательностей IBM ISW
 - Свойства node scripting 195, 200

- модели последовательности
 - Свойства node scripting 168
- модели разделительной кластеризации Netezza
 - Свойства node scripting 201, 210
- модели регрессии IBM ISW
 - Свойства node scripting 195, 200
- модели регрессии Кокса
 - Свойства node scripting 141, 178
- модели связывания IBM ISW
 - Свойства node scripting 195, 200
- модели списка решений
 - Свойства node scripting 142, 179
- моделирование базы данных 185

Н

- надузел 55
- Надузел
 - поток 25
- надузлы
 - сценарии 1, 3, 25
- Надузлы
 - задание свойств в 241
 - параметры 241
 - потoki 25
 - свойства 241
 - сценарии 241
- наивные модели Байеса
 - Свойства node scripting 210
- наивные модели Байеса IBM ISW
 - Свойства node scripting 195, 200
- наивные модели Байеса Netezza
 - Свойства node scripting 201
- наивные модели Байеса Oracle
 - Свойства node scripting 188, 194
- направленный узел Web
 - свойства 126
- наследование 23
- нейронные сети
 - Свойства node scripting 182
- нейросети
 - Свойства node scripting 163
- Нейросеть MS
 - Свойства node scripting 185, 187

О

- обобщенные линейные модели
 - Свойства node scripting 148, 180
- обобщенные линейные модели Netezza
 - Свойства node scripting 201
- обобщенные линейные модели Oracle
 - Свойства node scripting 188
- объектно-ориентированное 21
- объекты вывода
 - имена сценариев 245
- объекты моделей
 - имена сценариев 243
- объекты модели
 - имена сценариев 245
- операторы 17
- операции 14
- определение атрибутов 23
- определение класса 22
- определение методов 23

П

- параметры 3, 55, 57
 - Надузлы 241
 - сценарии 14
- Параметры slot 3, 55, 56
- пароли
 - добавление к сценариям 47
 - закодированные 51
- передача аргументов 18
- переменная итерации
 - циклы в сценариях 6
- переменные
 - сценарии 14
- перемещение по узлам 31
- перенастройка
 - выполнение потоков 253
 - доступ к объектам 254
 - задание свойств 251
 - имена свойств 250
 - команды 247
 - контекст сценариев 247
 - обзор 247
 - общие отличия 247
 - переменные 250
 - получение свойств 251
 - разное 255
 - редактирование потоков 251
 - репозиторий 254
 - ссылки на узлы 250
 - типы вывода 250
 - типы моделей 250
 - типы узлов 250
 - файловая система 254
 - функции 247
 - циклы 252
- поиск и замена 10
- поиск узлов 27
- поля
 - выключение в сценариях 115
- порядок выполнения
 - изменение через сценарии 47
- порядок выполнения потока
 - изменение через сценарии 47
- потоки
 - выполнение 25
 - изменение 29
 - Команда multiset 55
 - свойства 57
 - сценарии 1, 2, 25
 - условное выполнение 4, 8
 - циклы 4, 5
- прерывание сценариев 9
- примеры 18
- проверка ошибок
 - сценарии 48

Р

- регулярные выражения 10
- Репозиторий IBM SPSS Collaboration and Deployment Services
 - аргументы командной строки 52

С

свойства

- Надузлы 241
- общие сценарии 56
- поток 57
- сценарии 55, 56, 127, 175, 227
- узлы моделирования баз данных 185
- свойства applyr 183
- свойства buildr 135
- Свойства flatfilenode 233
- Свойства node scripting 185
 - model nuggets 175
 - узлы моделирования 127
 - узлы экспорта 227
- Свойства узла aggregate 79
- Свойства узла analysis 213
- Свойства узла anomalydetection 128
- Свойства узла anonymize 91
- Свойства узла append 79
- Свойства узла applyanomalydetection 175
- Свойства узла applyapriori 175
- Свойства узла applyautoclassifier 176
- Свойства узла applyautocluster 176
- Свойства узла applyautonumeric 176
- Свойства узла applybayesnet 177
- Свойства узла applyc50 177
- Свойства узла applycarma 177
- Свойства узла applycart 177
- Свойства узла applychaid 178
- Свойства узла applycoxreg 178
- Свойства узла applydb2imcluster 200
- Свойства узла applydb2imlog 200
- Свойства узла applydb2imnb 200
- Свойства узла applydb2imreg 200
- Свойства узла applydb2imtree 200
- Свойства узла applydecisionlist 179
- Свойства узла applydiscriminant 179
- Свойства узла applyfactor 179
- Свойства узла applyfeatureselection 179
- Свойства узла applygeneralizedlinear 180
- Свойства узла applyglmm 180
- Свойства узла applykmeans 180
- Свойства узла applyknn 181
- Свойства узла applykohonen 181
- Свойства узла applylinear 181
- Свойства узла applylogreg 181
- Свойства узла applymslogistic 187
- Свойства узла applymsneuralnetwork 187
- Свойства узла applymsregression 187
- Свойства узла applymssequencecluster 187
- Свойства узла applymstimeseries 187
- Свойства узла applymstree 187
- Свойства узла applynetezzbayes 210
- Свойства узла applynetez zadectree 210
- Свойства узла applynetez zadivcluster 210
- Свойства узла applynetez zakmeans 210
- Свойства узла applynetez zaknn 210
- Свойства узла
 - applynetez zalineregression 210
 - applynetez zanaivebayes 210
 - applynetez zapca 210
 - applynetez zaregtree 210
- Свойства узла applyneuralnet 181
- Свойства узла applyneuralnetwork 182
- Свойства узла applyoraabn 194
- Свойства узла applyoradecisiontree 194
- Свойства узла applyorakmeans 194
- Свойства узла applyoranb 194
- Свойства узла applyoranmf 194
- Свойства узла applyoracluster 194
- Свойства узла applyorasvm 194
- Свойства узла applyquest 182
- Свойства узла applyregression 183
- Свойства узла applyselflearning 183
- Свойства узла applysequence 183
- Свойства узла applysvm 184
- Свойства узла applytimeseries 184
- Свойства узла applytwestep 184
- Свойства узла apriori 129
- свойства узла asexport 227
- свойства узла asimport 62
- Свойства узла autoclassifier 130
- Свойства узла autocluster 132
- Свойства узла autodataprep 91
- Свойства узла autonumeric 133
- Свойства узла balance 80
- Свойства узла bayesnet 134
- Свойства узла binning 94
- Свойства узла c50 136
- Свойства узла carma 137
- Свойства узла cart 138
- Свойства узла chaid 139
- Свойства узла cognosimport 63
- Свойства узла collection 116
- Свойства узла coxreg 141
- Свойства узла dataaudit 214
- Свойства узла database 64
- Свойства узла databaseexport 229
- Свойства узла datacollectionexport 232
- Свойства узла datacollectionimport 65
- Свойства узла db2imassoc 195
- Свойства узла db2imcluster 195
- Свойства узла db2imlog 195
- Свойства узла db2imnb 195
- Свойства узла db2imreg 195
- Свойства узла db2imsequence 195
- Свойства узла db2imtimeseries 195
- Свойства узла db2imtree 195
- Свойства узла decisionlist 142
- Свойства узла derive 97
- свойства узла derive_stb 80
- Свойства узла directedweb 126
- Свойства узла discriminant 144
- Свойства узла distinct 82
- Свойства узла distribution 117
- Свойства узла ensemble 98
- Свойства узла evaluation 117
- Свойства узла evimport 68
- Свойства узла excelexport 232
- Свойства узла excelimport 67
- Свойства узла factor 145
- Свойства узла featureselection 146
- Свойства узла filler 98
- Свойства узла filter 99
- Свойства узла fixedfile 69
- Свойства узла genlin 148
- Свойства узла glmm 151
- Свойства узла graphboard 119
- Свойства узла histogram 121
- Свойства узла history 99
- Свойства узла kmeans 155
- Свойства узла knn 155
- Свойства узла kohonen 157
- Свойства узла linear 157
- Свойства узла logreg 159
- Свойства узла matrix 215
- Свойства узла means 216
- Свойства узла merge 82
- Свойства узла msassoc 185
- Свойства узла msbayes 185
- Свойства узла mscluster 185
- Свойства узла mslogistic 185
- Свойства узла msneuralnetwork 185
- Свойства узла msregression 185
- Свойства узла mssequencecluster 185
- Свойства узла mstimeseries 185
- Свойства узла mstree 185
- Свойства узла multiplot 122
- Свойства узла netezzbayes 201
- Свойства узла netez zadectree 201
- Свойства узла netez zadivcluster 201
- Свойства узла netez zaglm 201
- Свойства узла netez zakmeans 201
- Свойства узла netez zaknn 201
- Свойства узла netez zalineregression 201
- Свойства узла netez zanaivebayes 201
- Свойства узла netez zapca 201
- Свойства узла netez zaregtree 201
- Свойства узла netez zartimeseries 201
- Свойства узла neuralnet 161
- Свойства узла neuralnetwork 163
- Свойства узла numericpredictor 133
- Свойства узла oraabn 188
- Свойства узла oraai 188
- Свойства узла oraapriori 188
- Свойства узла oradecisiontree 188
- Свойства узла oraglm 188
- Свойства узла orakmeans 188
- Свойства узла oramdl 188
- Свойства узла oranb 188
- Свойства узла oranmf 188
- Свойства узла oracluster 188
- Свойства узла orasvm 188
- Свойства узла outputfile 233
- Свойства узла partition 100
- Свойства узла plot 123
- Свойства узла quest 165
- Свойства узла reclassify 101
- Свойства узла regression 166
- Свойства узла reorder 102
- Свойства узла report 218
- Свойства узла restructure 102
- Свойства узла rfaggregate 83
- Свойства узла rfanalysis 102
- свойства узла Routput 218
- свойства узла Rprocessnode 84
- Свойства узла sample 85
- Свойства узла sasexport 234
- Свойства узла sasimport 71
- Свойства узла select 86
- Свойства узла sequence 168
- Свойства узла setglobals 219
- Свойства узла settotflag 104
- Свойства узла simeval 219
- Свойства узла simfit 220
- Свойства узла simgen 71
- Свойства узла slrm 168
- Свойства узла sort 87
- Свойства узла statistics 221
- Свойства узла statisticsexport 238
- Свойства узла statisticsimport 237

- Свойства узла statisticsmodel 238
- Свойства узла statisticsoutput 238
- Свойства узла statisticstransform 237
- свойства узла streamingts 87
- Свойства узла svm 169
- Свойства узла table 222
- Свойства узла timeintervals 104
- Свойства узла timeplot 125
- Свойства узла timeseries 170
- Свойства узла transform 224
- Свойства узла transpose 108
- Свойства узла twostep 172
- Свойства узла type 109
- Свойства узла userinput 74
- Свойства узла variablefile 75
- Свойства узла web 126
- Свойства узла xmlexport 234
- Свойства узла xmlimport 78
- сгенерированное ключевое слово 48
- сгенерированные модели
 - имена сценариев 243, 245
- сервер
 - аргументы командной строки 51
- символы не из кодового набора ASCII 20
- системные
 - аргументы командной строки 50
- скрытые переменные 23
- слепки
 - Свойства node scripting 175
- создание класса 22
- создание узлов 29, 30
- списки 14
- ссылки на узлы 27
 - задание свойств 28
 - поиск узлов 27
- строки 15
- сценарии
 - автономные сценарии 1, 25
 - в надузлах 3
 - выбор полей 7
 - выполнение 9
 - диаграммы 25
 - из командной строки 48
 - импорт из текстовых файлов 2
 - используемые сокращения 55
 - ключ итерации 5
 - контекст 26
 - наглядное изображение циклов 4, 5
 - общие свойства 56
 - переменная итерации 6
 - пользовательский интерфейс 3, 10
 - потoki 1, 25
 - потoki надузлов 25
 - прерывание 9
 - пути файлов 48
 - синтаксис 14, 15, 16, 17, 18, 19, 20, 21, 22, 23
 - совместимость с более старыми версиями 48
 - сохранение 2
 - сценарии надузла 1, 25
 - Сценарий Python 248, 249, 252, 255
 - узлы вывода 213
 - узлы диаграммы 115
 - унаследованные сценарии 248, 249, 252, 255
 - условное выполнение 4, 8

- сценарии (*продолжение*)
 - циклы 4, 5
- сценарий
 - выполнение узла моделирования 47
 - замена модели 47
 - обзор 1, 13
 - пользовательский интерфейс 2
 - порядок выполнения потока 47
 - проверка ошибок 48

У

- узел simeval (оценка имитации)
 - свойства 219
- узел SimFit (подгонка имитации)
 - свойства 220
- узел SimGen
 - свойства 71
- узел Web
 - свойства 126
- узел автоклассификации
 - Свойства node scripting 130
- узел автокластеризации
 - Свойства node scripting 132
- узел агрегации
 - свойства 79
- узел агрегации RFM
 - свойства 83
- узел анализа
 - свойства 213
- узел анализа RFM
 - свойства 102
- узел анонимизации
 - свойства 91
- узел ансамбля
 - свойства 98
- узел Аудит данных
 - свойства 214
- узел базы данных
 - свойства 64
- узел балансировки
 - свойства 80
- узел выбора
 - свойства 86
- Узел выборки
 - свойства 85
- Узел вывода IBM SPSS Statistics
 - свойства 238
- узел вывода R
 - свойства 218
- узел генерирования имитации
 - свойства 71
- узел Гистограмма
 - свойства 121
- узел График
 - свойства 123
- узел График зависимости от времени
 - свойства 125
- узел Задать глобальные значения
 - свойства 219
- узел Задать как флаг
 - свойства 104
- Узел заполнения
 - свойства 98
- Узел извлечения
 - свойства 97

- узел интервалов времени
 - свойства 104
- узел источника Analytic Server
 - свойства 62
- узел источника Excel
 - свойства 67
- узел источника IBM SPSS Statistics
 - свойства 237
- узел источника SAS
 - свойства 71
- узел источника XML
 - свойства 78
- узел Матрица
 - свойства 215
- узел Несколько графиков
 - свойства 122
- узел обработки R
 - свойства 84
- узел особого типа
 - свойства 82
- узел отчета
 - свойства 218
- узел Оценка
 - свойства 117
- узел оценки имитации
 - свойства 219
- узел Панель выбора диаграмм
 - свойства 119
- Узел переклассификации
 - свойства 101
- узел переупорядочения
 - свойства 102
- узел переупорядочения полей
 - свойства 102
- узел плоского файла
 - свойства 233
- узел подгонки имитации
 - свойства 220
- узел пользовательского ввода
 - свойства 74
- узел потоковых временных рядов
 - свойства 87
- Узел Представление предприятия
 - свойства 68
- узел преобразования
 - свойства 224
- узел преобразования IBM SPSS Statistics
 - свойства 237
- узел присоединения
 - свойства 79
- узел раздела
 - свойства 100
- Узел разделения на интервалы
 - свойства 94
- узел Распределение
 - свойства 117
- узел реструктуризации
 - свойства 102
- узел сборки R
 - Свойства node scripting 135
- узел Слияние
 - свойства 82
- узел Собрание
 - свойства 116
- узел сортировки
 - свойства 87

узел средних
 свойства 216
узел статистики
 свойства 221
узел таблицы
 свойства 222
узел Тип
 свойства 109
узел транспонирования
 свойства 108
узел файла переменных
 свойства 75
узел фиксированного файла
 свойства 69
Узел фильтра
 свойства 99
узел Хронология
 свойства 99
узел экспорта Excel
 свойства 232
Узел экспорта IBM SPSS Data Collection
 свойства 232
узел экспорта IBM SPSS Statistics
 свойства 238
узел экспорта SAS
 свойства 234
узел экспорта XML
 свойства 234
узел экспорта базы данных
 свойства 229
узлы
 замена 30
 импорт 30
 информация 32
 отсоединение узлов 29
 соединение узлов 29
 ссылки на имена 243
 удаление 30
узлы вывода
 свойства сценариев 213
узлы диаграммы
 свойства сценариев 115
узлы моделирования
 Свойства node scripting 127
узлы пространственно-временных
 интервалов
 свойства 80
узлы экспорта
 Свойства node scripting 227
условное выполнение потоков 4, 8

Ц

циклы в потоках 4, 5

Ф

флаги
 аргументы командной строки 49
 объединение нескольких флагов 53
функции
 комментарии 248
 литералы 248
 операции 248
 операции вывода документов 255
 операции с моделями 255
 операции с потоками 255
 операции с узлами 252
 ссылки на объекты 248
 условное выполнение 249
 циклы 249



Напечатано в Дании