



Deductor

АНАЛИТИЧЕСКАЯ ПЛАТФОРМА

для эффективных бизнес решений

Описание демонстрационного примера

© BaseGroup Labs 1998-2006

www.basegroup.ru

© BaseGroup Labs 1998-2006

В документе описан демонстрационный пример работы с аналитической платформой Deductor 4.4: принципы построения сценариев обработки, описание реализованных механизмов анализа и обработки, настройка их параметров, импорт и экспорт данных. Демонстрируется пошаговое решение задач анализа бизнес данных при помощи Deductor 4.4 на тестовых примерах. Описание предназначено для аналитиков, руководителей подразделений и других специалистов, которые хотят использовать Deductor 4.4 для бизнес-анализа. Специальных знаний в области анализа данных не требуется, но предполагается, что читатель знаком с вводным курсом статистики и является квалифицированным пользователем компьютера.

Содержание

Примеры обработки данных с помощью инструментов Deductor Studio	5
Импорт данных	5
Импорт данных из 1С:Предприятия 7.7	9
Импорт данных из 1С:Предприятия 8.0	18
Примеры предобработки данных	28
Парциальная предобработка	28
Исходные данные	28
Восстановление пропущенных данных	29
Удаление аномалий	30
Спектральная обработка	31
Удаление шумов	32
Удаление больших, малых и средних шумов	32
Сглаживание больших, малых и средних шумов	33
Факторный анализ	33
Исходные данные	34
Понижение размерности пространства входных факторов	34
Корреляционный анализ	35
Исходные данные	35
Устранение незначущих входных факторов	36
Трансформация данных	37
Разбиение данных на группы	37
Разбиение даты (по неделям)	38
Квантование возраста кредиторов на 5 интервалов	40
Фильтрация данных	41
Калькулятор	43
Исходные данные	43
Функции F1(АРГУМЕНТ3), F2(АРГУМЕНТ3)	43
Функция от двух аргументов F3(АРГУМЕНТ1; АРГУМЕНТ2)	45
Вычисление отклонения АРГУМЕНТ1+1 от АРГУМЕНТ2+1	45
Пример кусочно-заданной функции	45
Группировка данных	46
Исходные данные	46
Группировка по городам	47
Преобразование данных к скользящему окну	47
Исходные данные	47
Преобразование скользящим окном	48
Настройка набора данных	48
Исходные данные	49
Выполнение настройки	49
Замена значений	49
Исходные данные	50
Выполнение замены	50
Слияние	50
Исходные данные	51
Выполнение слияния	51
Выявление дубликатов и противоречий	51
Исходные данные	52
Поиск дубликатов и противоречий	52
Примеры анализа данных	55
Прогнозирование умножения с помощью нейронных сетей	55
Исходные данные	55
Прогнозирование результата умножения	55
Выводы	59
Классификация с помощью деревьев решений	60
Исходные данные	60

Классификация на демократов и республиканцев	60
Выводы	64
Прогнозирование с помощью линейной регрессии.	65
Исходные данные	65
Прогнозирование суммы	65
Выводы	66
Кластеризация с помощью самоорганизующейся карты Кохонена	67
Исходные данные	67
Кластеризация ирисов	67
Выводы	70
Поиск ассоциативных правил	71
Исходные данные	71
Поиск ассоциативных правил	72
Выводы	75
Прогнозирование с помощью построения пользовательских моделей	76
Исходные данные	76
Прогнозирование с применением пользовательских моделей	76
Выводы	77
Пример расчета автокорреляции столбцов	78
Исходные данные	78
Автокорреляция столбца Количество	78
Пример прогноза временного ряда	79
Исходные данные	79
Удаление аномалий и сглаживание	80
Скользящее окно 12 месяцев назад	81
Обучение нейросети (прогноз на 1 месяц вперед)	81
Построение прогноза	83
Выводы	84
Применение скрипта	85
Исходные данные	85
Указание цепочки выполняемых обработок	85
Выводы	87
Условное выполнение ветки сценария	88
Исходные данные	88
Настройка условия	89
Расчет условия	89
Выводы	89
Экспорт данных.	90
Заключение	91

Примеры обработки данных с помощью инструментов Deductor Studio

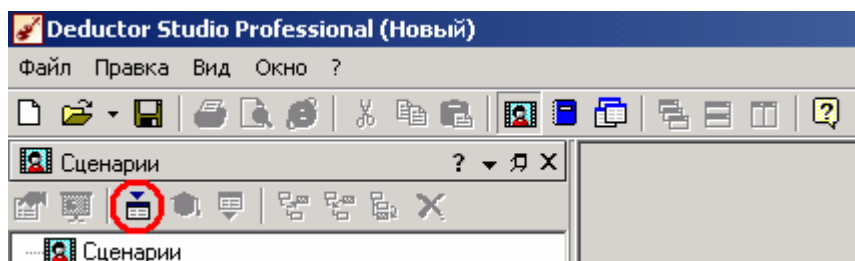
Deductor Studio – программа, являющаяся составной частью платформы Deductor. Она содержит механизмы импорта, обработки, визуализации и экспорта данных для быстрого и эффективного анализа и прогнозирования.

Данный файл позволит ознакомиться с возможностями, заложенными в Deductor Studio, рассмотреть каждую из них на конкретном примере, а также узнать, как набор механизмов действует в комплексе, ознакомившись в конце со способом построения законченного решения по прогнозированию объемов продаж товаров на три месяца вперед.

Все примеры разбиты на несколько категорий, в зависимости от цели, которая ставится перед аналитиком. Так, существует группа инструментов предобработки данных, которая приводит исходные «сырые» данные к виду, пригодному для анализа и обработки (устранение аномалий, сглаживание, заполнение пропусков...), группа инструментов преобразования данных, которая изменяет данные на основе настроек аналитика (группирует, дискретизирует, фильтрует...), группа инструментов анализа данных позволяет найти зависимости одних факторов от других, выявить противоречивые данные, найти сезонность во временных рядах, значимость влияния факторов на результат, а также построить модель прогноза и получить желаемый результат (провести эксперимент, спрогнозировать временной ряд).










Импорт данных

Импорт данных является отправной точкой анализа данных. Импорт в Deductor может осуществляться из популярных форматов хранения данных, таких как Excel, Access, MS SQL, Oracle, Текстовый файл и прочих. Кроме того, имеется универсальный доступ к любому источнику данных посредством ADO или ODBC.

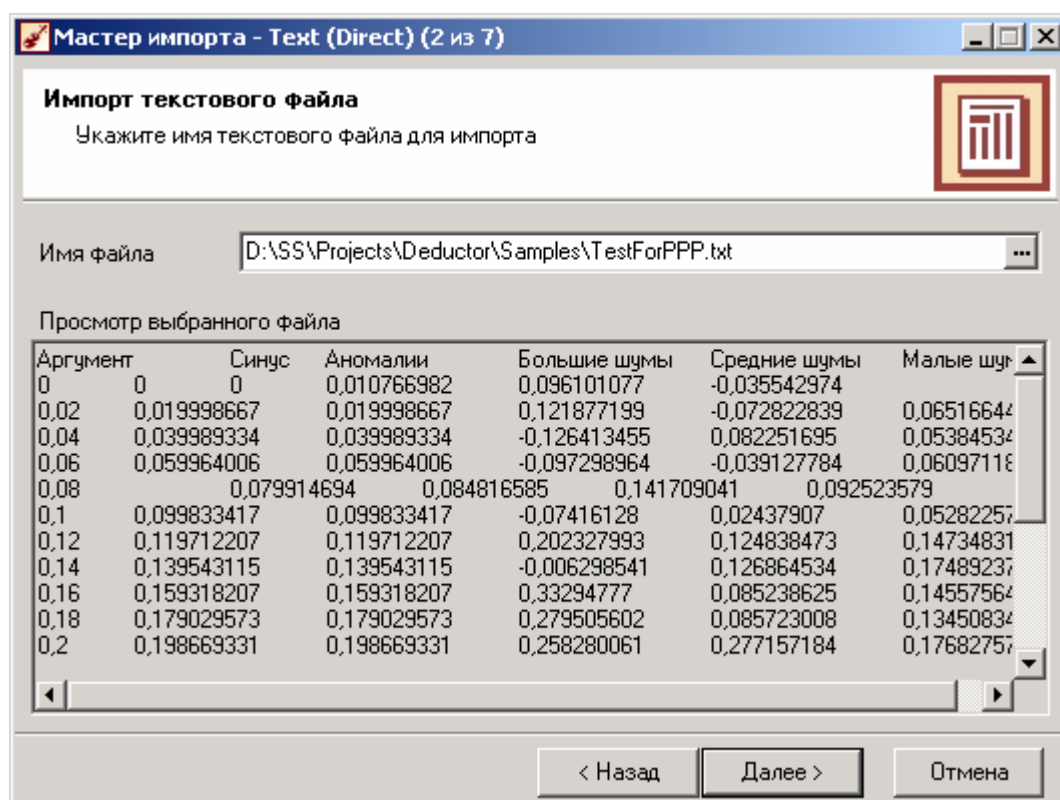


Рассмотрим пример импорта данных из текстового файла с разделителями, который будет необходим при апробировании технологий платформы Deductor на предлагаемых примерах.

Импорт осуществляется путем вызова мастера импорта на панели «Сценарии»

- [-] Бизнес-приложение
 -  1С:Предприятие 7.7 Импорт данных из 1С:Предприятия 7.7
- [-] Базы данных
 -  База данных Настроенный источник данных
- [-] Прямой доступ к файлам
 -  **Text (Direct)** Текстовый файл с разделителями (пря...
 -  DBase (Direct) Таблица в формате DBase (прямой дос...
- [-] Механизм MS ADO
 -  MS Excel Microsoft Excel (ADO)
 -  MS Access Microsoft Access (ADO)
 -  DBase (ADO) Таблицы в формате DBase (ADO)
 -  Text (ADO) Текстовый файл с разделителями (ADO)
 -  ADO источник ADO драйвер доступа к данным

После запуска мастера импорта укажем тип импорта “Текстовый файл с разделителями” и перейдем к настройке импорта. Укажем имя файла, из которого необходимо получить данные (пример для парциальной обработки). В окне просмотра выбранного файла можно увидеть содержание данного файла.



Далее перейдем к настройке параметров импорта. На этой странице мастера предоставляется возможность указать, с какой строки следует начать импорт, указать, то, что первая строка является заголовком, возможность добавить первичный ключ. Указать, что является символом-разделителем столбцов, а также указать ограничитель строк, разделитель целой и дробной части вещественного числа, разделитель компонентов даты и ее формат.

В данном случае параметры по умолчанию на этой странице мастера установлены правильно, а именно: начать импорт с первой строки, первая строка является заголовком, разделителем между столбцами является знак табуляции, разделителем целой и дробной частей является запятая.

Далее перейдем к настройке свойств полей.

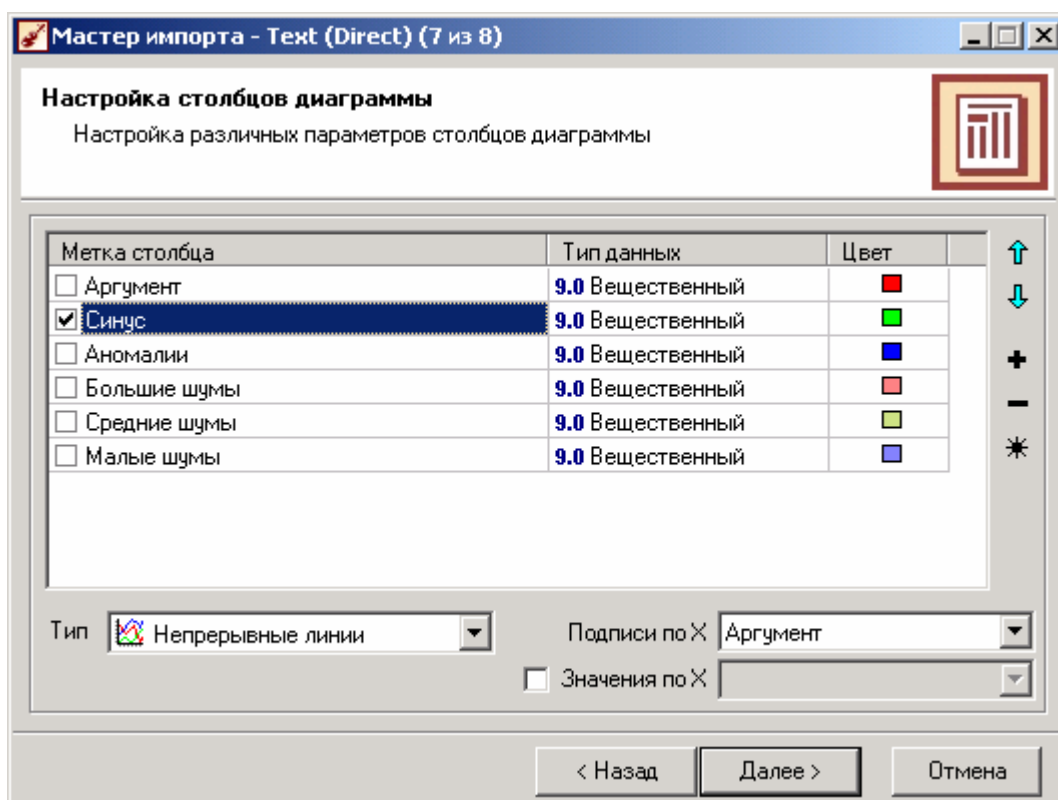
На этом шаге мастера предоставляется возможность настроить имя, название (метку), размер, тип данных, вид данных и назначение. Некоторые свойства (например, тип данных) можно задавать для выделенного набора столбцов. Вид данных определяет – конечный ли это набор (дискретные) или бесконечный (непрерывные). Назначение столбцов определяет характер их использования в алгоритмах обработки (при импорте можно оставить значение по умолчанию).

Для правильного импорта данных необходимо изменить тип данных у первых трех столбцов («АРГУМЕНТ», «СИНУС», «АНОМАЛИИ»). Тип данных по умолчанию неверный, поскольку программа определяет его, основываясь на значениях первой строки данных. В данном случае там находятся нули – целые числа. Поэтому программа определила, что столбец содержит целочисленные значения. Выделим их с помощью мыши и укажем им тип данных – «Вещественный». Далее осталось только выполнить импорт данных, нажав на кнопку «Пуск» на следующем шаге мастера импорта.

После импорта данных на следующем шаге мастера необходимо выбрать способ отображения данных. В данном случае самым информативным является диаграмма, выберем ее.

- ☒ Табличные данные
 - ☐ Таблица Отображает данные в виде таблицы
 - ☐ Статистика Отображает статистические данные выборки
 - ☒ **Диаграмма** Отображает данные в виде диаграммы
 - ☐ Гистограмма Отображает данные в виде гистограммы
- ☐ OLAP анализ
 - ☐ Куб Многомерное отображение (кросс-таблица и кросс-диаграмма)
- ☐ Прочее
 - ☐ Описание Сведения о параметрах

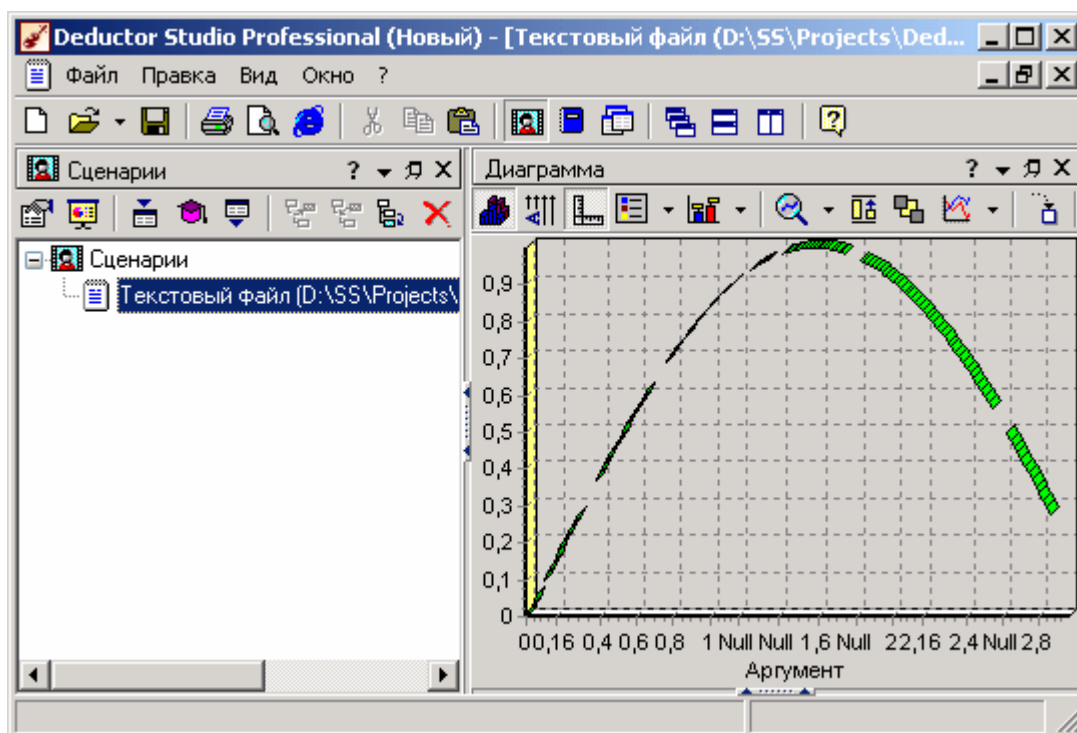
От того, какие способы отображения будут выбраны на этом этапе, зависят последующие шаги мастера. В данном случае необходимо настроить, какие столбцы диаграммы следует отображать и как именно.



Выберем для отображения поле «СИНУС» и тип диаграммы «Линии».

На последнем шаге мастера необходимо указать название ветки в дереве сценариев.

Напишем в поле заголовка окна «Импорт примера для демонстрации парциальной обработки» и нажмем «Готово». На этом работа мастера импорта заканчивается. Теперь в дереве сценариев появится новый узел с необходимыми данными. В главном окне программы представлены все выбранные отображения данных этого узла. В данном случае только диаграмма.

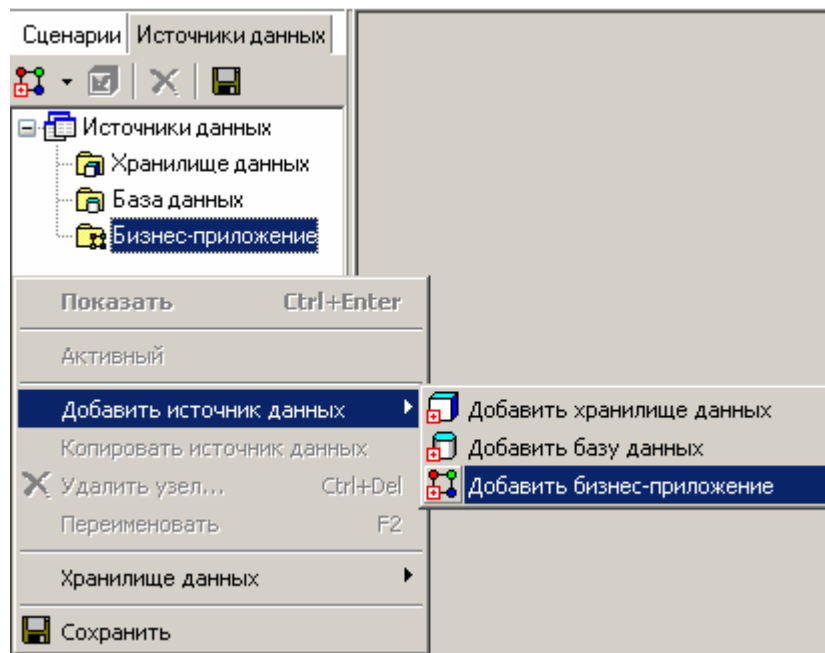


Импорт данных из 1С:Предприятия 7.7

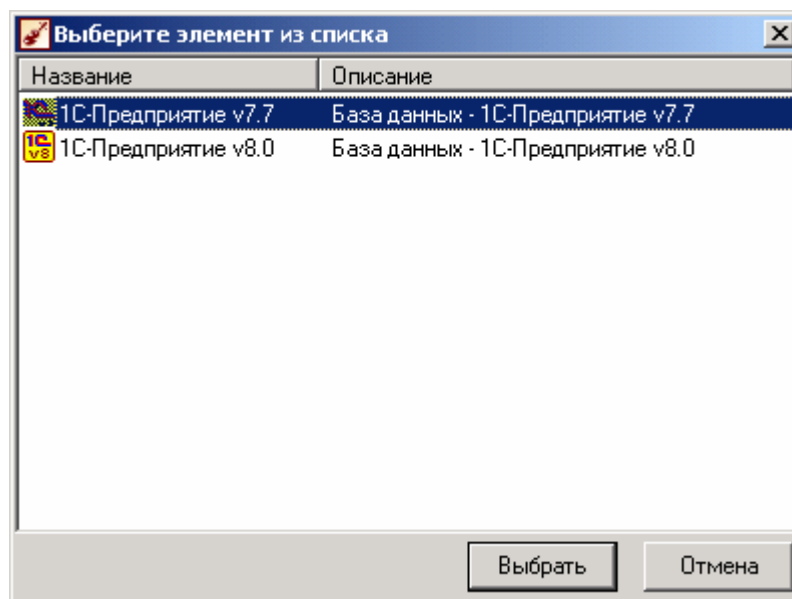
Одним из распространенных источников данных являются конфигурации 1С:Предприятия 7.7. Рассмотрим для примера подробно процесс импорта данных из демонстрационной конфигурации 1С:Комплексная конфигурация (демо). Предполагается, что на компьютере с Deductor установлено приложение 1С:Предприятие 7.7 с релизом не ниже 7.70.25 и демонстрационная конфигурация 1С:Комплексная.

Перед этапом импорта данных требуется настроить новый источник данных. Для этого следует сначала подключить конфигурацию в конфигураторе 1С:Предприятия (см. документацию к 1С:Предприятию 7.7). Следующим шагом следует запустить Deductor и открыть с помощью главного меню «Вид» панель «Источники данных».

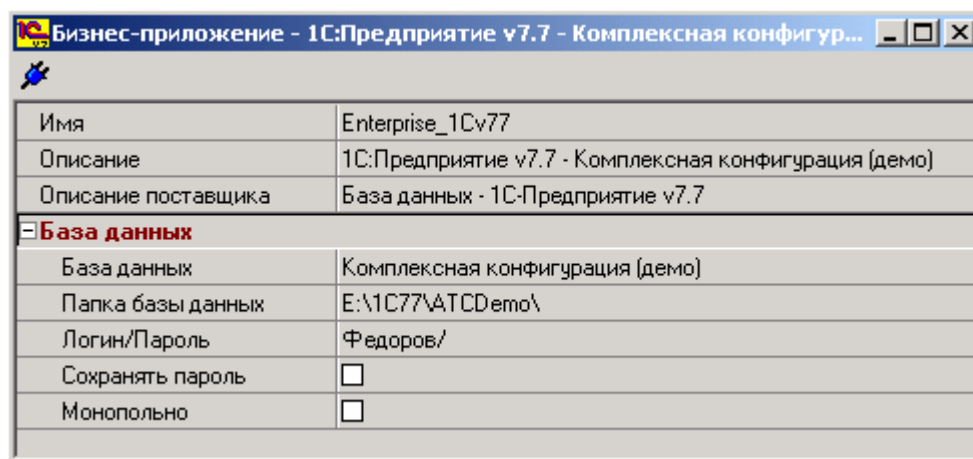
Для подключения в качестве источника данных конфигурации 1С:Предприятия 7.7 следует из всплывающего меню выбрать пункт «Добавить источник данных» - «Добавить бизнес-приложение».



В списке доступных бизнес-приложений следует выделить 1С:Предприятие 7.7 и нажать кнопку «Выбрать».



На экране после этого появится окно настройки параметров подключения. Здесь из выпадающего списка выберем поле «Комплексная конфигурация (демо)», в поле «Логин/Пароль» введем имя пользователя «Федоров», в поле описание – метку подключения «1С:Предприятие v7.7 - Комплексная конфигурация (демо)», а поле с именем подключения оставим без изменения (значение по умолчанию – Enterprise_1Cv77, именно на него настроена работа демо-примера). Данные настройки показаны на следующем рисунке.

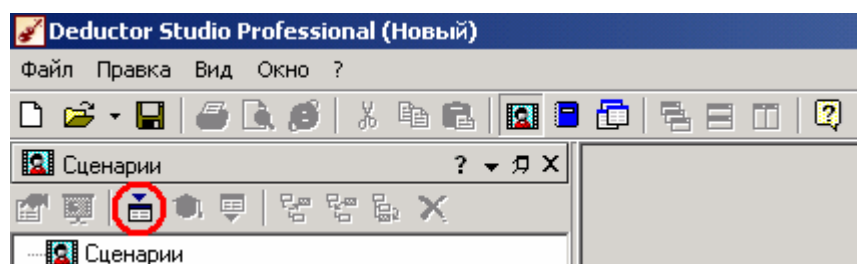


Если на компьютере не установлено 1C:Предприятие 7.7, либо в конфигураторе не зарегистрировано ни одной конфигурации, список «База данных» окажется пустым. Для продолжения работы следует установить само приложение, конфигурацию «1C:Комплексная (демо)» и зарегистрировать ее в Конфигураторе 1C:Предприятия.

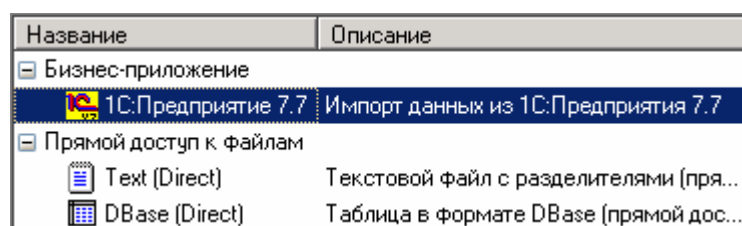
Проверить правильность настроек можно нажатием кнопки «Тестирование соединения» на панели инструментов этого окна. В процессе тестирования соединения 1C:Предприятие может вывести на экран окно с сообщением о том, что не открыт оперативный период. Для продолжения загрузки конфигурации следует закрыть это окно нажатием кнопки «Ок». Если все настройки указаны верно, на экране появится окно с сообщением «Тестирование соединения прошло успешно». В случае появления сообщения об ошибке следует еще раз проверить правильность настроек и попробовать загрузить конфигурацию 1C:Предприятия отдельно от Deductor. При работе в автономном режиме следует устранить все ошибки, выдаваемые сервером 1C:Предприятия, и после этого повторить проверку соединения.

Настроенный источник данных сохраним для дальнейшего использования, выбрав из всплывающего меню панели «Источники данных» пункт «Сохранить». Теперь можно перейти собственно к процессу импорта данных.

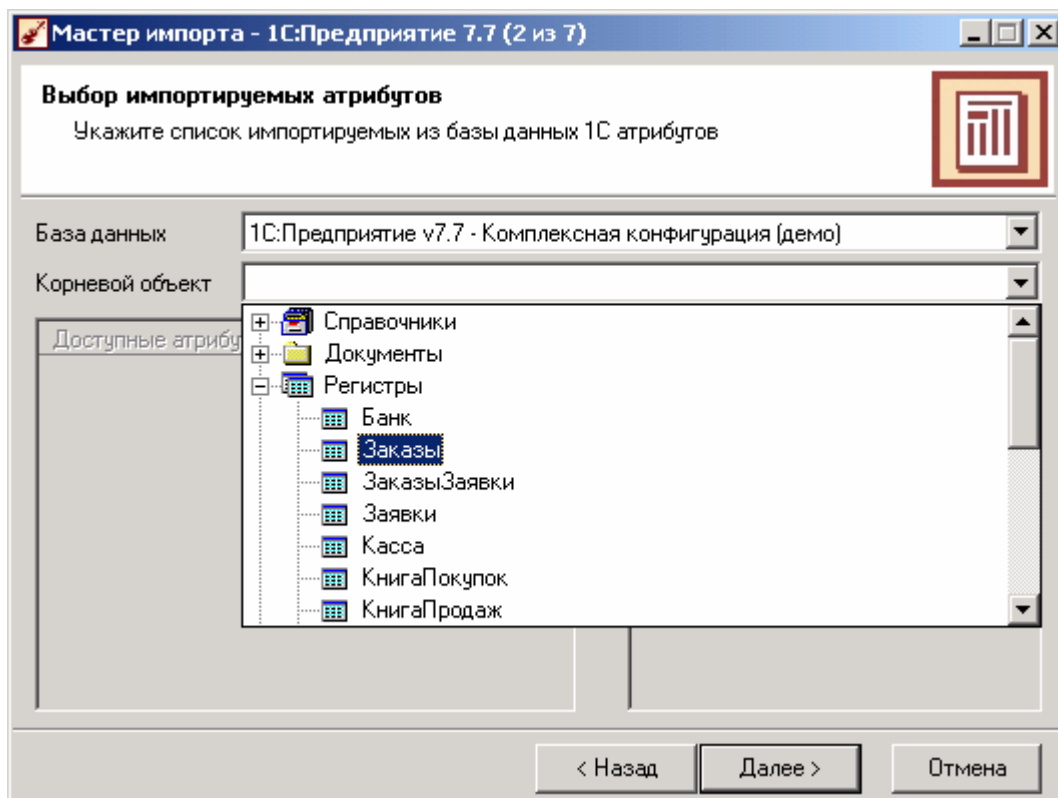
Вызовем на панели сценариев Мастер импорта.



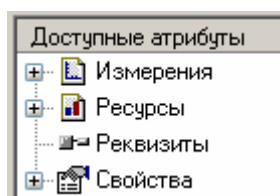
В появившемся списке доступных источников данных выберем «Бизнес-приложение» - «1C:Предприятие 7.7» и перейдем к выбору импортируемых атрибутов.



Из списка «База данных» выберем настроенный ранее источник данных «1С:Предприятие v7.7 - Комплексная конфигурация (демо)». После загрузки 1С:Предприятия в списке «Корневой объект» появится дерево объектов конфигурации, из которых возможен импорт данных.



Одной из наиболее часто встречающихся задач анализа данных является прогнозирование. Для построения прогнозов нужны данные о продажах товаров, поэтому будем импортировать информацию из регистра «Продажи». Выделим его в списке «Корневой объект», после чего в таблице «Доступные атрибуты» появится дерево атрибутов этого регистра, которые можно загрузить в программу.

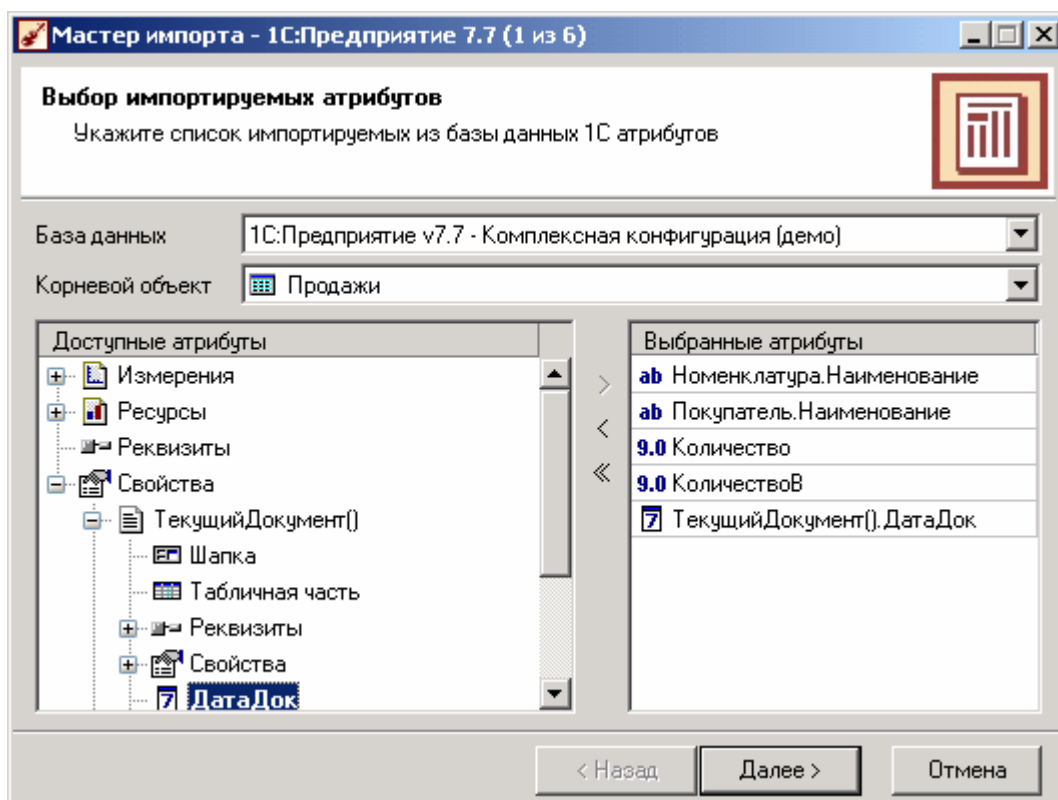


Нас будут интересовать следующие атрибуты:

- Измерения – Номенклатура – Наименование;
- Измерения – Покупатель – Наименование;
- Ресурсы – Количество;
- Ресурсы – КоличествоВ;
- Свойства – ТекущийДокумент() – ДатаДок.

Т.е. наименования товарной позиции и покупателя, объем продаж товара в натуральном выражении, количество возвратов и дата документа, по которому осуществлялась проводка (т.е. дата продажи).

Найдем последовательно эти атрибуты в дереве доступных атрибутов и выберем их двойным щелчком мыши или с помощью кнопок на панели инструментов справа. При выборе атрибута он подсвечивается полужирным шрифтом, и его имя перемещается в список «Выбранные атрибуты». После выбора указанных атрибутов этот список примет следующий вид:



Теперь перейдем на следующий шаг Мастера импорта с помощью кнопки «Далее».

В окне настройки дополнительных параметров импорта данных из регистра 1С:Предприятия в качестве способа получения данных оставим выбор движений из регистра, т.к. нас интересуют прежде всего именно движения, а не итоги. Способ задания диапазона дат оставим также по умолчанию – относительным. Точка отсчета – это конечная дата периода, за который будет производиться импорт данных из регистра. Если оставить значение по умолчанию («Имеющиеся данные»), то конечной точкой импорта данных станет точка актуальности, установленная в конфигурации. Тип периода зададим «Год», а количество периодов достаточно большим, например, 10, чтобы в выборку гарантированно попали все данные, имеющиеся в регистре.

Мастер импорта - 1С:Предприятие 7.7 (2 из 6)

Параметры импорта из 1С
Укажите параметры импорта данных из 1С

Способ получения данных: **Выбор движений**

Способ задания диапазона: **Относительный**

Точка отсчета:

- ☐ Текущая дата
- ☒ Имеющиеся данные
- ☐ Абсолютная дата: 09.10.2004

Тип периода: **Год**

Количество периодов: **10**

< Назад Далее > Отмена

Перейдем на следующую страницу Мастера с помощью кнопки «Далее». На этой странице Мастера импорта предоставляется возможность указать настройки импортируемых полей. Это окно не зависит от типа источника данных, пример настройки полей приведен выше, в описании импорта из текстового файла. Сейчас оставим эти настройки без изменений.

На следующем шаге Мастера можно запустить собственно процесс импорта данных, нажав кнопку «Пуск». Когда все данные будут загружены (в строке «Название процесса» появится сообщение «Успешное завершение»), перейдем далее на страницу определения способов отображения данных. Выберем здесь таблицу.

Название	Описание
<input checked="" type="checkbox"/> Табличные данные	
<input checked="" type="checkbox"/> Таблица	Отображает данные в виде таблицы
<input type="checkbox"/> Статистика	Отображает статистические данные выборки
<input type="checkbox"/> Диаграмма	Отображает данные в виде диаграммы

На следующем шаге можно ввести заголовок для окна, в котором будут отображаться полученные данные, например, так:

Заголовок для окна

Импорт из "1С:Предприятия v7.7 - Комплексная конфигурация" (Регистр.Продажи)

Завершив работу Мастера импорта нажатием кнопки «Готово». После этого на экране появится окно с таблицей, содержащей данные, полученные из регистра «Продажи» комплексной конфигурации.

Импорт из "1С:Предприятия v7.7 - Комплексная конфигурация" (Регистр.Продажи)				
Таблица				
1 из 88				
Номенклатура.Наименование	Покупатель.Наименование	Количество	КоличествоВ	ТекущийДокумент().ДатаДок
Вентилятор BINATONE ALPINE	Белявский-частное лицо	1	0	14.12.2001
Вентилятор настольный	Белявский-частное лицо	1	0	14.12.2001
Вафли "Причуда"	Магазин "Все для дома"	10	0	24.12.2001
Конфеты "Барбарис"	Магазин "Все для дома"	20	0	24.12.2001
Конфеты "Белочка"	Магазин "Все для дома"	10	0	24.12.2001

Далее с этими данными можно работать как с данными, полученными из любого другого источника. Например, с помощью Калькулятора рассчитаем фактические продажи каждого товара. Для этого следует из общего количества проданного товара вычесть количество возвратов. Вызовем Мастер обработки и выберем из дерева доступных обработчиков Калькулятор. На следующей странице для выражения «Выражение» укажем метку «Объем продаж» (для этого двойным щелчком на выражении вызовем редактор свойств и в поле «Метка» введем нужный текст).

Параметры выражения

Имя:

Метка:

Тип данных:

Описание:

Ok Отмена

В поле для ввода выражения напишем требуемую формулу: COL3 – COL4 (COL3 и COL4 имена для полей Количество и КоличествоВ соответственно). После этого нажмем кнопку «Далее», переходя к дальнейшим этапам Мастера обработки.

Мастер обработки - Калькулятор (2 из 4)

Конструктор выражения

Создайте выражение, вычисляемое на основе столбцов данных, используя арифметические операции, а также встроенные функции

Название выражения: 9.0 Объем продаж

↑ ↓ + +! -

COL3-COL4

Имя столбца	Метка столбца
ab COL1	Номенклатура.Наименован...
ab COL2	Покупатель.Наименование
9.0 COL3	Количество
9.0 COL4	КоличествоВ
7 COL5	ТекущийДокумент().ДатаД...

Операции

" " () + - * /

= <> < > <= >=

and or not xor true false

fx Функция

< Назад Далее > Отмена

На следующих шагах Мастера обработки оставим все настройки по умолчанию. В результате к нашему набору данных добавится новое поле с фактическими объемами продаж товаров.

Номенклатура.Наименование	Покупатель.Наименование	Количество	КоличествоВ	ТекущийДокумент().ДатаДок	Объем продаж
Вентилятор BINATONE ALPINE	Белявский-частное лицо	1	0	14.12.2001	1
Вентилятор настольный	Белявский-частное лицо	1	0	14.12.2001	1
Вафли "Причуда"	Магазин "Все для дома"	10	0	24.12.2001	10
Конфеты "Барбарис"	Магазин "Все для дома"	20	0	24.12.2001	20

Теперь поля Количество и КоличествоВ не нужны для последующей обработки, поэтому их можно исключить из набора данных с помощью обработчика «Настройка набора данных». Для этого в главном окне обработчика укажем для этих полей назначение «Неиспользуемое». Заодно для поля ТекущийДокумент().ДатаДок поменяем метку на «Дата продажи» (см. рисунок).

Мастер обработки - Настройка набора данных (2 из 4)

Изменение параметров набора данных
 Укажите новые параметры столбцов, а также необходимость кэширования данных

ab Номенклатура.Наименование
ab Покупатель.Наименование
9.0 *Количество*
9.0 *КоличествоВ*
☒ Дата продажи
9.0 Объем продаж

Имя столбца: COL4
 Метка столбца: КоличествоВ
 Тип данных: 9.0 Вещественный
 Вид данных: — Непрерывный
 Назначение: Неиспользуемое

Сброс параметров

☐ Кэшировать результирующий набор данных

< Назад Далее > Отмена

После завершения работы Мастера набор данных примет следующий вид:

Номенклатура.Наименование	Покупатель.Наименование	Дата продажи	Объем продаж
Вентилятор BINATONE ALPINE	Белявский-частное лицо	14.12.2001	1
Вентилятор настольный	Белявский-частное лицо	14.12.2001	1
Вафли "Причуда"	Магазин "Все для дома"	24.12.2001	10
Конфеты "Барбарис"	Магазин "Все для дома"	24.12.2001	20

Эти данные можно просмотреть в более удобном виде с помощью кросс-таблицы:

		Номенклатура.Наименование ▾			
Дата продажи ▾	Покупатель.Наименование ▾	Ботинки женски	Ботинки же	Вафли "Пр	Вентилятор В
14.12.2001	Белявский-частное лицо				1,00
	Итого				1,00
24.12.2001	Магазин "Все для дома"			10,00	
	Итого			10,00	
26.12.2001	Розничная продажа			10,00	
	Итого			10,00	
27.12.2001	Магазин "Продукты"			5,00	
	Розничная продажа			4,00	
	Итого			9,00	

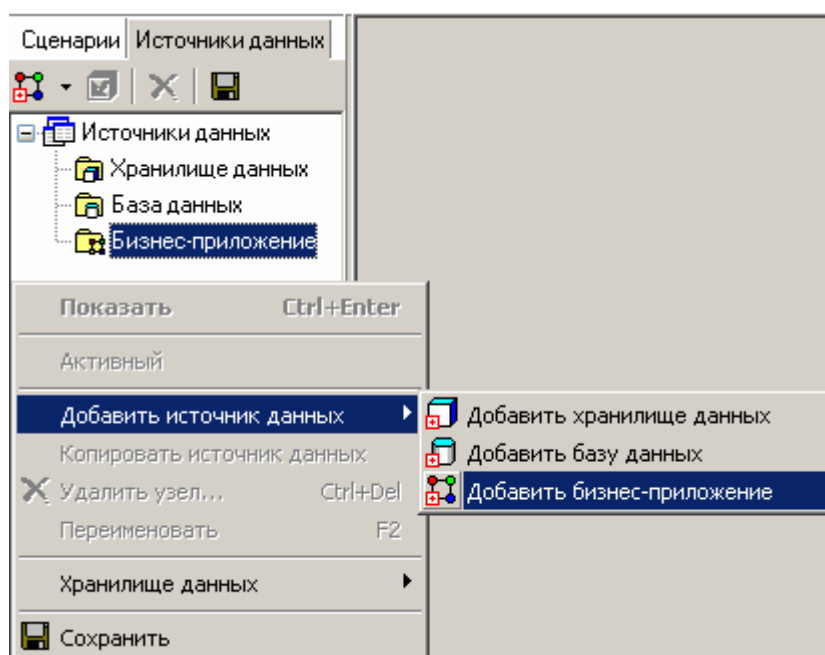
Дальнейшая обработка данных зависит от задач аналитика. Например, теперь можно перейти к прогнозированию объемов продаж товаров или построению аналитической отчетности. Решение этих задач рассматривается в соответствующих разделах «Описания демо-примера».

Импорт данных из 1С:Предприятия 8.0

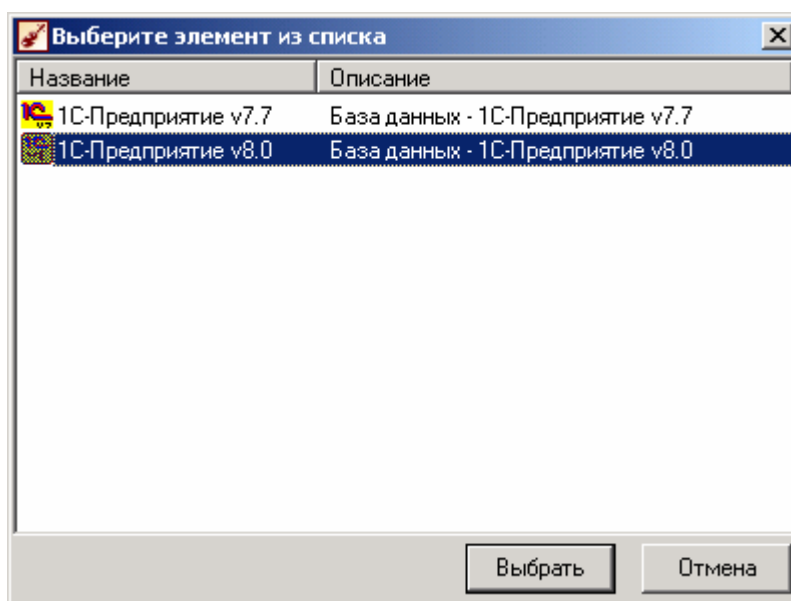
Процесс импорта данных из конфигурации учетной системы 1С:Предприятие 8.0 не сильно отличается от работы с версией 7.7. Рассмотрим процесс получения данных из демонстрационной конфигурации «Торговля и склад». Получим из нее данные о продажах товаров за весь период работы компании.

Первое, что необходимо сделать, - это настроить источник данных. Для этого сначала требуется подключить конфигурацию в конфигураторе 1С:Предприятия 8.0 (см. документацию к 1С:Предприятию 8.0). Далее следует запустить Deductor и открыть панель «Источники данных», выбрав соответствующий пункт в меню «Вид».

Для подключения в качестве источника данных конфигурации 1С:Предприятия 8.0 следует из всплывающего меню выбрать пункт «Добавить источник данных» - «Добавить бизнес-приложение».



В списке доступных бизнес-приложений следует выделить 1С:Предприятие 7.7 и нажать кнопку «Выбрать».



На экране появится окно настройки параметров подключения. Здесь из выпадающего списка выберем поле «DemoDB – Торговля и склад» (если в конфигураторе 1С:Предприятия для конфигурации «Торговля и склад» было выбрано другое имя, то здесь следует выделить его). В поле «Логин/Пароль» введем имя пользователя «Иванов», в поле описание – метку подключения «1С:Предприятие v8.0 – Торговля и склад (демо)», а поле с именем подключения оставим без изменения (значение по умолчанию – Enterprise_1Cv80, именно на него настроена работа демо-примера). Данные настройки показаны на следующем рисунке.

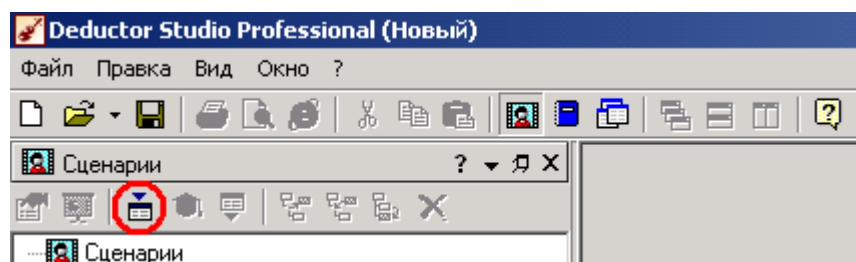
Имя	Enterprise_1Cv80
Описание	1С:Предприятие v8.0 - Торговля и склад (демо)
Описание поставщика	База данных - 1С:Предприятие v8.0
База данных	
База данных	DemoDB - Торговля и склад
Строка подключения к базе	File="E:\1C80\DemoDB";
Логин/Пароль	Иванов
Сохранять пароль	<input type="checkbox"/>
Спрашивать логин/пароль при подключении	<input type="checkbox"/>

Если на компьютере не установлено 1С:Предприятие 8.0, либо в конфигураторе для текущего пользователя не зарегистрировано ни одной конфигурации, то список «База данных» окажется пустым. Для продолжения работы следует установить само приложение, конфигурацию «1С:Торговля и склад» и зарегистрировать ее в Конфигураторе 1С:Предприятия 8.0.

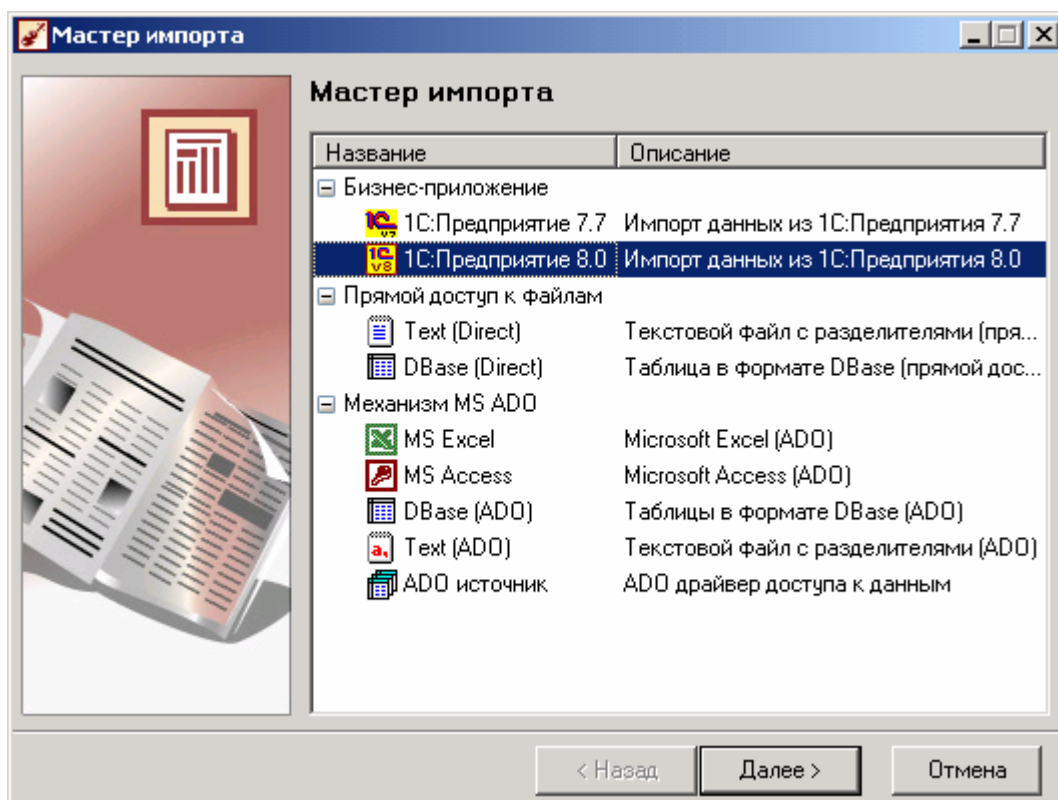
Проверить правильность настроек можно нажатием кнопки «Тестирование соединения» на панели инструментов этого окна. Если все настройки указаны верно, на экране появится окно с сообщением «Тестирование соединения прошло успешно». В случае появления сообщения об ошибке следует еще раз проверить правильность настроек и попробовать загрузить конфигурацию 1С:Предприятия отдельно от Deductor. При работе в автономном режиме следует устранить все ошибки, выдаваемые сервером 1С:Предприятия, и после этого повторить проверку соединения.

Настроенный источник данных сохраним для дальнейшего использования, выбрав из всплывающего меню панели «Источники данных» пункт «Сохранить». Теперь можно перейти собственно к процессу импорта данных.

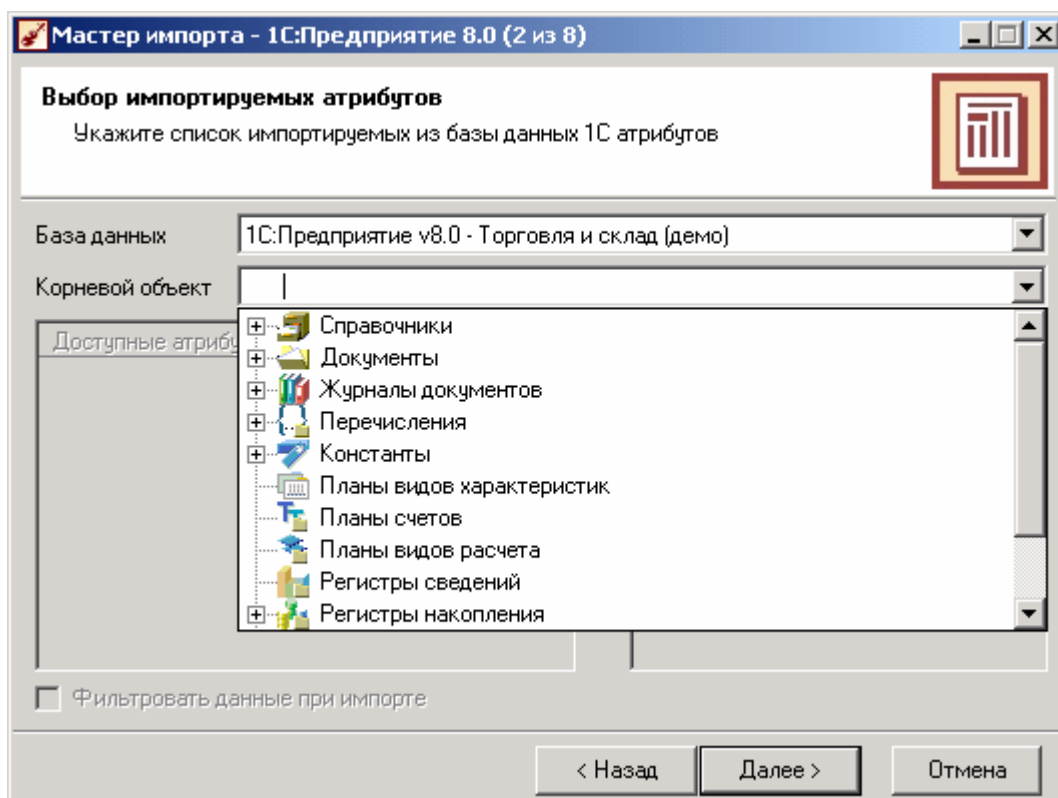
Вызовем на панели сценариев Мастер импорта.



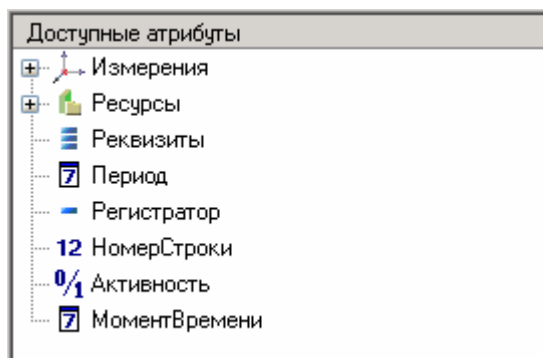
В появившемся списке доступных источников данных выберем «Бизнес-приложение» - «1С:Предприятие 8.0» и перейдем к выбору импортируемых атрибутов.



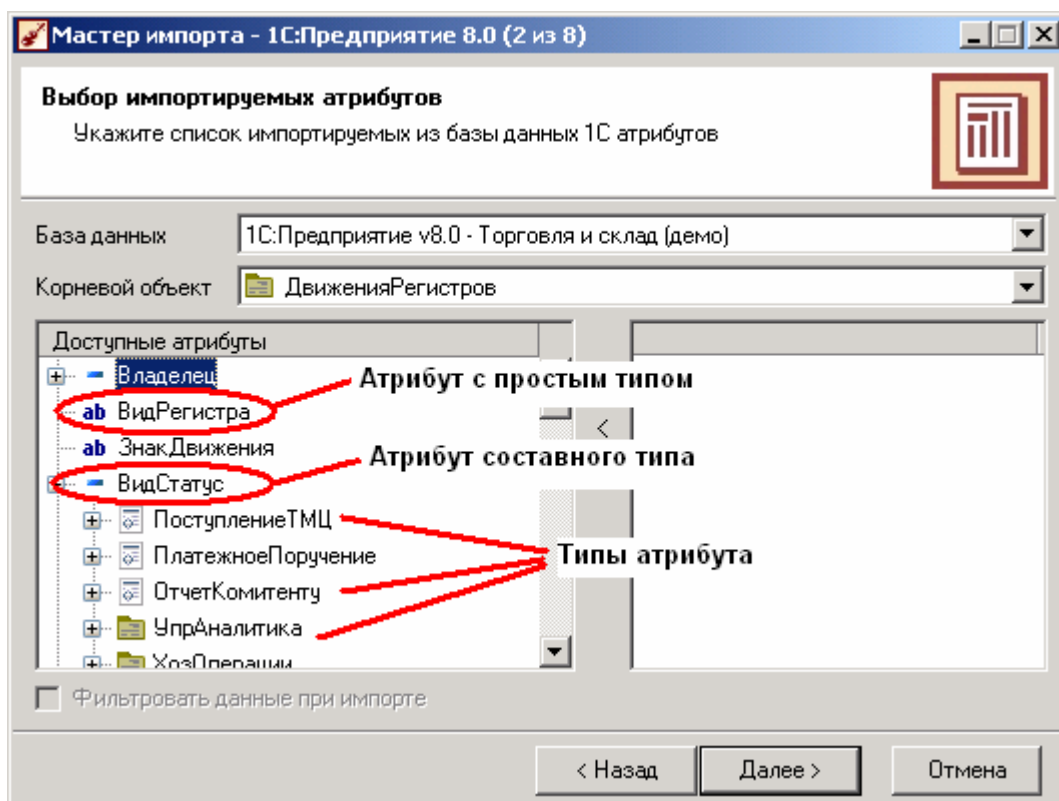
Из списка «База данных» выберем настроенный ранее источник данных «1С:Предприятие v8.0 – Торговля и склад (демо)». После загрузки 1С:Предприятия в списке «Корневой объект» появится дерево объектов конфигурации, из которых возможен импорт данных.



Данные о продажах хранятся в этой конфигурации в регистре накопления «Продажи». Выделим его в списке «Корневой объект», после чего в таблице «Доступные атрибуты» появится дерево атрибутов этого регистра, которые можно загрузить в программу.



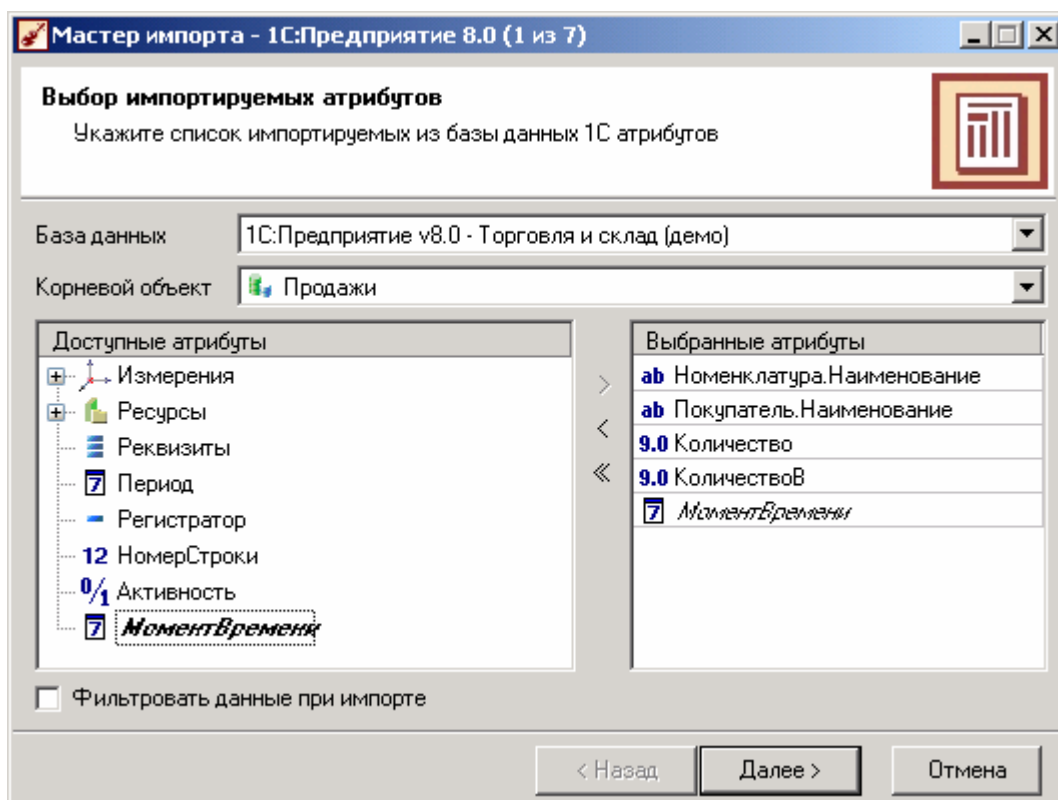
Так как в 1С:Предприятии 8.0 у каждого атрибута может быть несколько различных типов, то эти типы представляются также в виде узлов дерева (см. следующий рисунок). Отличие таких узлов состоит в том, что импорт данных непосредственно из них невозможен, поэтому их нельзя выбрать из этого дерева и перенести в список «Выбранные атрибуты».



Нас будут интересовать следующие атрибуты:

- Измерения – Номенклатура – Номенклатура – Наименование;
- Измерения – Покупатель – Контрагенты – Наименование;
- Ресурсы – Количество;
- Ресурсы – КоличествоВ;
- МоментВремени.

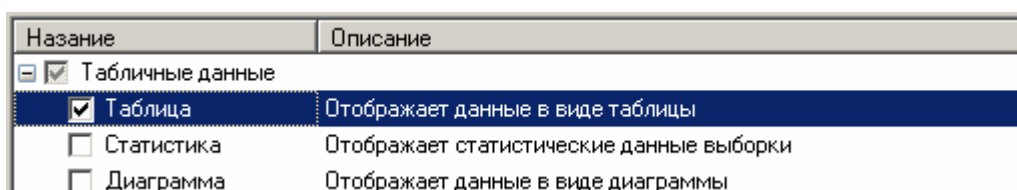
Здесь соответственно первая строка – наименование номенклатуры, вторая – наименование контрагента; Количество – продажи в штуках, КоличествоВ – возвраты в штуках; МоментВремени – дата и время проведения документа, оно в данном случае совпадает со значением поля Период. Все эти настройки показаны на следующем рисунке.



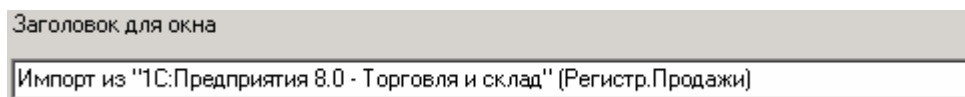
После выбора указанных атрибутов перейдем на следующий шаг Мастера импорта, нажав кнопку «Далее».

На этой закладке можно запустить собственно процесс импорта данных, нажав кнопку «Пуск». Когда все данные будут загружены (в строке «Название процесса» появится сообщение «Успешное завершение»), перейдем далее на страницу настройки импортируемых полей.

Это окно не зависит от типа источника данных, пример настройки полей приведен выше, в описании импорта из текстового файла. Сейчас оставим эти настройки без изменений. Перейдем дальше на страницу настройки способов отображения. Выберем таблицу.



На следующем шаге можно ввести заголовок для окна, в котором будут отображаться полученные данные, например, так:



Завершив работу Мастера импорта нажатием кнопки «Готово». После этого на экране появится окно с таблицей, содержащей данные, полученные из регистра «Продажи» комплексной конфигурации.

Импорт из "1С:Предприятия 8.0 - Торговля и склад" (Регистр.Продажи)				
Таблица				
Номенклатура.Наименование	Покупатель.Наименование	Количество	КоличествоВ	МоментВремени
▶ Вафли "Причуда"	Розничная продажа	0	1	29.04.2002 12:00:00
Печенье "Сердечко"	Розничная продажа	0	1	29.04.2002 12:00:00
Вафли "Причуда"	Розничная продажа	10	0	26.04.2002 12:02:00
Крупа гречневая	Розничная продажа	15	0	26.04.2002 12:02:00
Крупа "Геркулес"	Розничная продажа	14	0	26.04.2002 12:02:00

Далее с этими данными можно работать как с данными, полученными из любого другого источника. Предположим, что нам требуется проанализировать продажи магазину «Продукты». Для этого следует к имеющемуся набору данных применить обработчик «Фильтрация». Вызовем Мастер обработки и выберем из дерева доступных обработчиков «Фильтрацию». На следующей странице зададим условие фильтрации: «Номенклатура.Наименование = Магазин "Продукты"». Значение фильтра «Магазин "Продукты"» можно ввести вручную или выбрать из списка значений, который появляется при нажатии кнопки в правой части поля «Значение».

Мастер обработки - Фильтрация (2 из 5)

Фильтрация данных
Настройте условия фильтрации входных данных

Операция	Имя поля	Условие	Значение
	ab Покупатель.Наименов...	=	Магазин "Продукты" ...

☐ Учитывать регистр

[[Покупатель.Наименование] = 'Магазин "Продукты"']

< Назад Далее > Отмена

Пройдем по остальным страницам Мастера обработки, оставляя все настройки без изменений. В результате на выходе получим следующий набор данных:

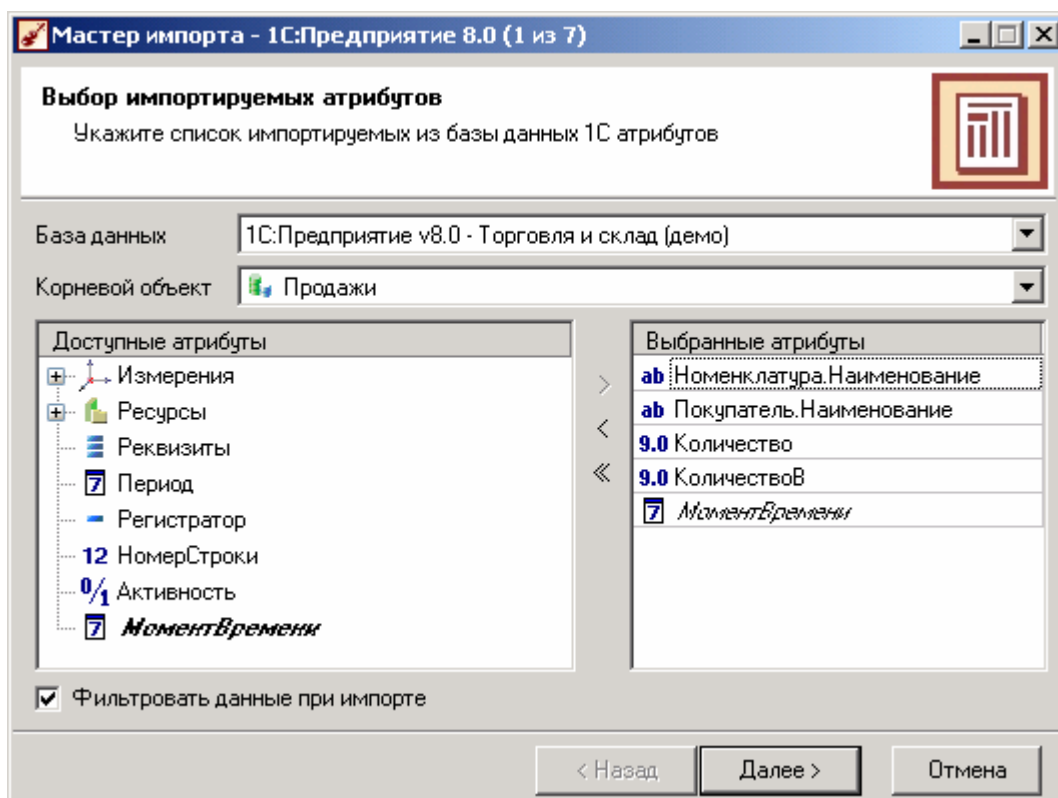
Номенклатура.Наименование	Покупатель.Наименование	Количество	КоличествоВ	МоментВремени
▶ Вафли "Причуда"	Магазин "Продукты"	5	0	27.04.2002 12:00:20
Крупа гречневая	Магазин "Продукты"	10	0	27.04.2002 12:00:20
Крупа манная	Магазин "Продукты"	10	0	27.04.2002 12:00:20
Крупа "Геркулес"	Магазин "Продукты"	5	0	27.04.2002 12:00:20
Крупа "Геркулес"	Магазин "Продукты"	5	0	27.04.2002 12:00:20
Сахарный песок	Магазин "Продукты"	30	0	27.04.2002 12:00:20
Вафли "Причуда"	Магазин "Продукты"	2	0	28.04.2002 12:01:40

Далее на этих данных можно, например, построить отчет об отгрузках товаров в этот магазин:

	МоментВремени ▾			
Номенклатура.Наименование ▾	27.04.2002	28.04.2002	30.04.2002	Итого
Вафли "Причуда"	5,00	2,00	3,00	10,00
Конфеты "Барбарис"			40,00	40,00
Конфеты "Белочка"		10,00	30,00	40,00
Конфеты "Грильяж"			20,00	20,00
Крупа "Геркулес"	10,00	5,00	10,00	25,00
Крупа гречневая	10,00	2,00	28,00	40,00
Крупа манная	10,00		10,00	20,00
Сахарный песок	30,00	10,00		40,00
Итого	65,00	29,00	141,00	235,00

Недостатком приведенного сценария является то, что для построения отчета по одному контрагенту потребовалось импортировать из 1С:Предприятия данные из всего регистра. Чтобы избежать лишних затрат времени и сразу избавиться от лишних данных, можно воспользоваться фильтром, встроенным в Мастер импорта из 1С:Предприятия.

Вызовем еще раз Мастер импорта, выберем импорт из 1С:Предприятия 8.0, источник данных «1С:Предприятие v8.0 – Торговля и склад (демо)» и дальше отметим все те же атрибуты, что и в предыдущий раз. Отличие будет состоять в том, что теперь на странице выбора атрибутов поставим флаг «Фильтровать данные при импорте».



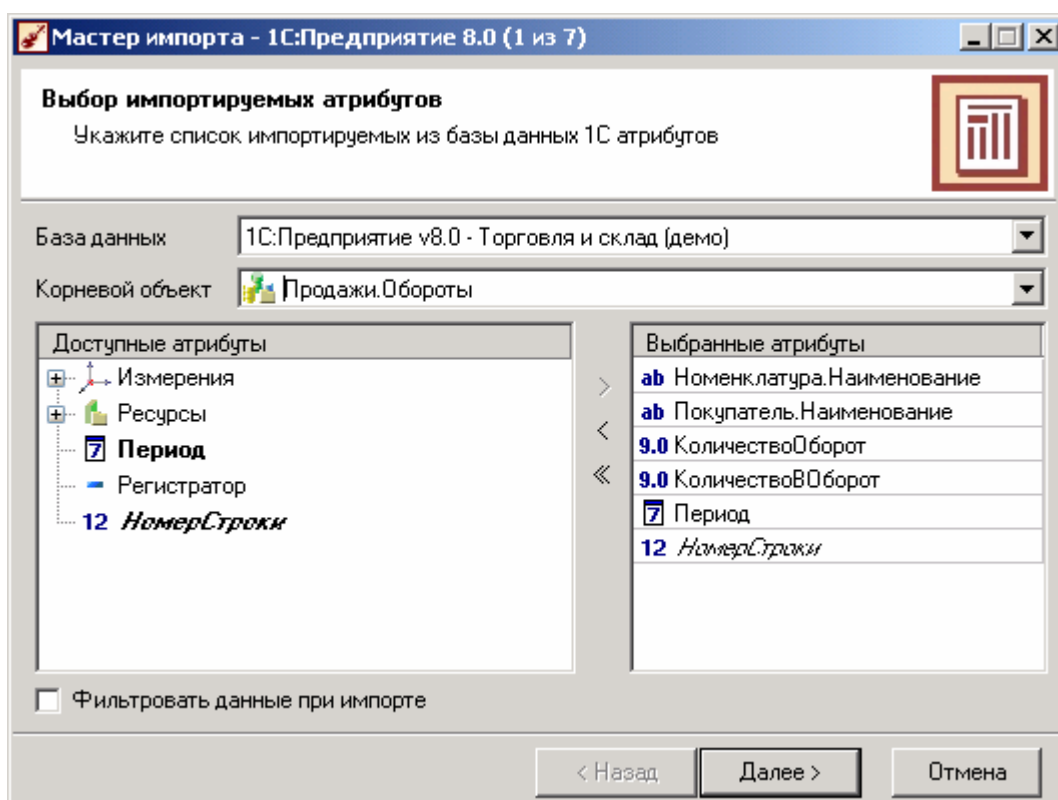
Не по всем импортируемым атрибутам можно настроить фильтрацию. Атрибуты, для которых такая возможность недоступна, отмечаются при выборе в дереве и в списке выбранных атрибутов курсивом (например, атрибут МоментВремени на предыдущем рисунке).

Перейдем на следующий шаг Мастера. Теперь можно увидеть окно фильтра, в котором можно задать условие выбора данных. В отличие от рассмотренного ранее варианта фильтрации, здесь придется поле «Значение» заполнять вручную. Так как данные из 1С:Предприятия еще не были получены, то нет возможности заполнить список значений. В этом окне укажем то же условие фильтрации, что и раньше (Номенклатура.Наименование = Магазин "Продукты"), и перейдем на следующие шаги обработки.

В итоге мы получили точно такой же набор данных, как и раньше, но теперь объем получаемых от сервера данных стал значительно меньше, а значит, ускорилась загрузка данных в Deductor.

Для некоторых объектов конфигурации 1С:Предприятия при импорте предусмотрены дополнительные настройки. Такими объектами являются дополнительные таблицы регистров. Рассмотрим работу с дополнительными параметрами на примере дополнительной таблицы «Обороты» регистра «Продажи».

Вызовем еще раз Мастер импорта данных из нашего источника данных. Выберем корневым объектом «Продажи.Обороты» и отметим атрибуты, как указано на следующем рисунке. Поле НомерСтроки содержит номер строки документа, по которому выполнялось движение регистра.



На следующей странице Мастера импорта будет предложено задать дополнительные параметры импорта.

Мастер импорта - 1С:Предприятие 8.0 (2 из 7)

Дополнительные параметры импорта

Задайте дополнительные параметры импорта данных

Для получения оборотов по регистру накопления нужно указать диапазон дат, за который следует выбирать итоги, и дополнительный разворот итогов по периодичности

☒ Относительный период
 ☐ Абсолютный период

Точка отсчета:

☒ Текущая дата

☐ Имеющиеся данные

☐ Абсолютная дата: 06.04.2006

Тип периода: Год

 Количество периодов: 10

Дополнительный разворот итогов по периодичности: Запись

Оставим относительный способ задания периода и текущую дату в качестве точки отсчета. Тип периода укажем год, а количество достаточно большим, чтобы в этот период гарантированно попали все данные из регистра. Чтобы выбранные на предыдущем этапе атрибуты Период и НомерСтроки были активны и заполнялись данными, потребуется установить дополнительный разворот итогов по периодичности в значение «Запись». Если будет выбрано другое значение, эти поля могут остаться пустыми. Подробнее об этом параметре, который есть у многих дополнительных таблиц, можно прочитать в документации к 1С:Предприятию 8.0.

Нажмем кнопку «Далее» для перехода к следующему шагу Мастера импорта. Если на странице «Выбор импортируемых атрибутов» был установлен флаг «Фильтровать импортируемый набор данных», то следующим шагом будет страница фильтра, а иначе – сразу страница запуска процесса импорта.

После завершения работы Мастера получим следующий набор данных:

Номенклатура.Наименование	Покупатель.Наименование	КоличествоОборот	КоличествоВОборот	Период	НомерСтроки
Вентилятор BINATONE ALPINE 1	Белявский-частное лицо	1	0	14.04.2002 12:00:00	1
Вентилятор настольный	Белявский-частное лицо	1	0	14.04.2002 12:00:00	2
Вафли "Прикуда"	Магазин "Все для дома"	10	0	24.04.2002 13:25:24	1
Конфеты "Барбарис"	Магазин "Все для дома"	20	0	24.04.2002 13:25:24	2
Конфеты "Белочка"	Магазин "Все для дома"	10	0	24.04.2002 13:25:24	3
Конфеты "Белочка"	Магазин "Все для дома"	10	0	24.04.2002 13:25:24	4
Конфеты "Грильяж"	Магазин "Все для дома"	10	0	24.04.2002 13:25:24	5
Крупа "Геркулес"	Магазин "Все для дома"	10	0	24.04.2002 13:25:24	6
Крупа "Геркулес"	Магазин "Все для дома"	10	0	24.04.2002 13:25:24	7
Крупа гречневая	Магазин "Все для дома"	20	0	24.04.2002 13:25:24	8
Крупа манная	Магазин "Все для дома"	10	0	24.04.2002 13:25:24	9
Сахарный песок	Магазин "Все для дома"	50	0	24.04.2002 13:25:24	10
Вафли "Прикуда"	Розничная продажа	10	0	26.04.2002 12:02:00	1
Крупа гречневая	Розничная продажа	15	0	26.04.2002 12:02:00	2

Данные, полученные из конфигурации 1С:Предприятия 8.0, могут обрабатываться средствами Deductor точно так же, как данные, полученные из любых других источников. Конкретные действия зависят от задач, стоящих перед аналитиком. Работа с различными инструментами и методы решения

общих проблем, встречающихся при анализе, рассматриваются в последующих разделах «Описания демо-примера».

Примеры предобработки данных

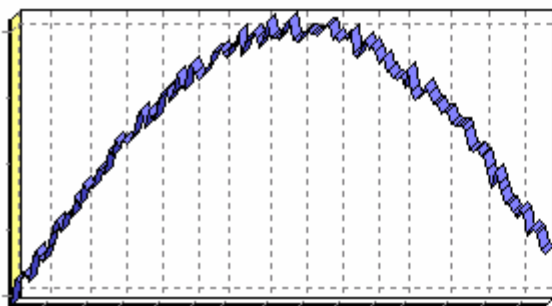
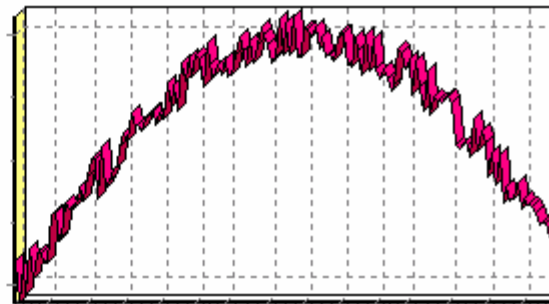
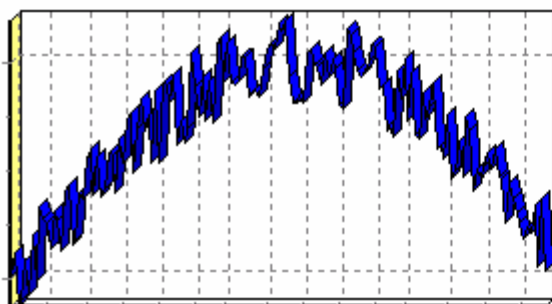
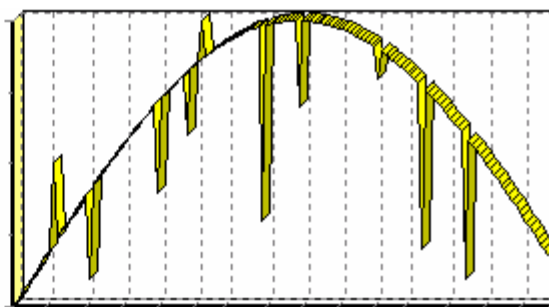
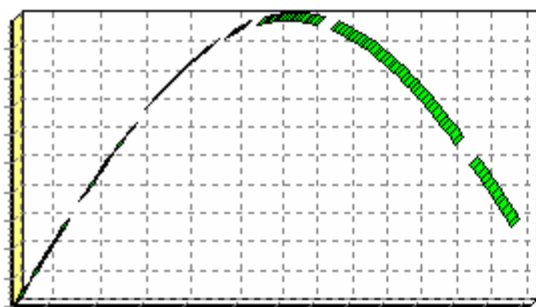
Часто исходные данные для анализа не годятся, а качество данных влияет на качество результатов. Так что вопрос подготовки данных для последующего анализа является очень важным. Обычно «сырые» данные содержат в себе различные шумы, за которыми трудно увидеть общую картину, а также аномалии – влияние случайно, либо редко происходивших событий. Очевидно, что влияние этих факторов на общую модель необходимо минимизировать, т.к. модель, учитывающая их, получится неадекватной.

Парциальная предобработка

Парциальная предобработка служит для восстановления пропущенных данных, редактирования аномальных значений и спектральной обработке данных (например, сглаживания данных). Именно этот шаг часто проводится в первую очередь.

Исходные данные

Рассмотрим применение обработки на примере данных из файла «TestForPPP.txt». Он содержит таблицу со следующими полями: «АРГУМЕНТ» – аргумент, «СИНУС» – значения синуса аргумента (некоторые значения пустые), «АНОМАЛИИ» – синус с выбросами, «БОЛЬШИЕ ШУМЫ» – значения синуса с большими шумами, «СРЕДНИЕ ШУМЫ» – значения синуса со средними шумами, «МАЛЫЕ ШУМЫ» – значения синуса с малыми шумами. Все данные можно увидеть на диаграмме после импорта из текстового файла.

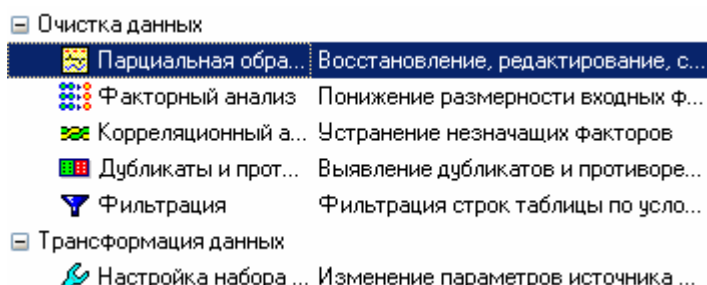


- а) Столбец с пропущенными данными
- б) Столбец с аномалиями (выбросами)
- в) Столбец с большими шумами
- г) Столбец со средними шумами
- д) Столбец с малыми шумами

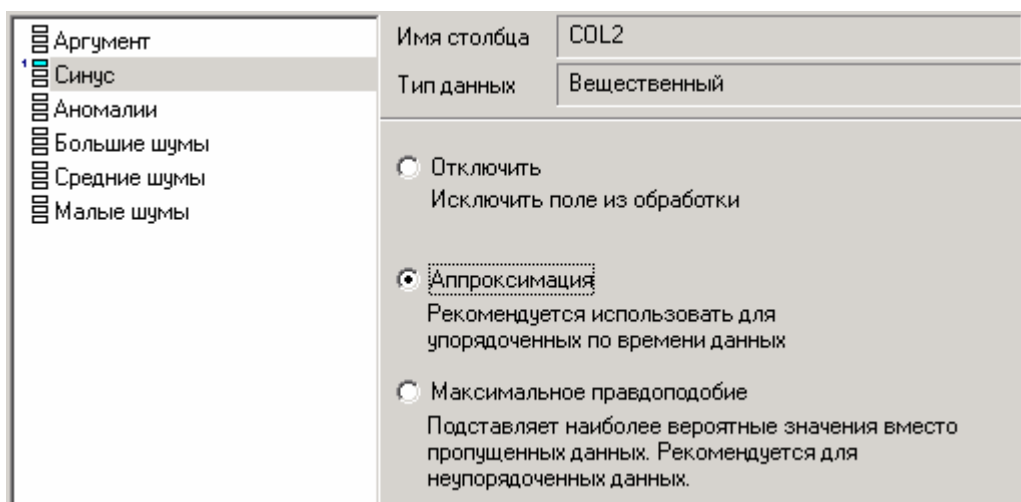
Восстановление пропущенных данных

Часто бывает так, что в столбце некоторые данные отсутствуют в силу каких либо причин (данные не известны, либо их забыли внести и т.п.). Обычно из-за этого пришлось бы убрать из обработки все строки, которые содержат пропущенные данные. Но механизмы Deductor Studio позволяют решить эту проблему. Один из шагов парциальной обработки как раз отвечает за восстановление пропущенных значений. Если данные упорядочены (например, по времени), то рекомендуется в качестве восстановления пропущенных значений использовать аппроксимацию. Алгоритм сам подберет значение, которое должно стоять на месте пропущенного значения, основываясь на близлежащих данных. Если же данные не упорядочены, то следует использовать режим максимального правдоподобия, когда алгоритм подставляет вместо пропущенных данных наиболее вероятные значения, основываясь на всей выборке.

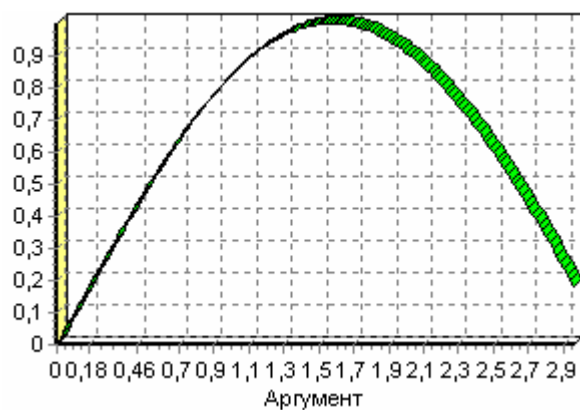
Для демонстрации воспользуемся мастером парциальной обработки. Импортировав файл можно увидеть, что в столбце «СИНУС» содержатся пустые значения. На диаграмме выше видно, что некоторые значения синуса пропущены. Для дальнейшей обработки необходимо их восстановить. Для этого следует запустить мастер парциальной обработки.



Поскольку данные в исходном наборе упорядочены, на следующем шаге мастера обработки выделим поле «СИНУС» и укажем для него тип обработки «Аппроксимация». Так как в данном случае больше ничего не требуется, то остальные параметры обработки оставляем отключенными. Перейдя на страницу запуска процесса обработки, выполняем ее, нажав на пуск, и далее выбираем тип визуализации обработанных данных (как в примере импорта).



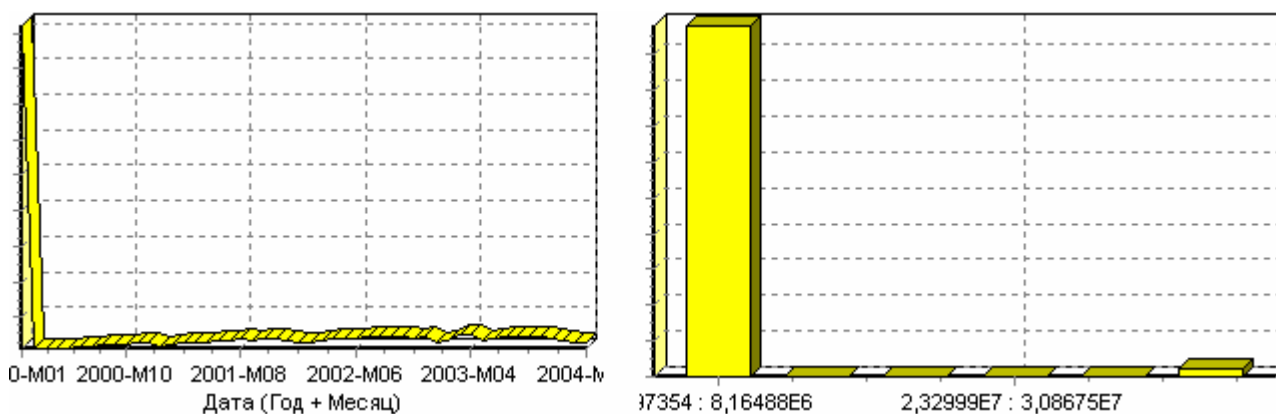
После выполнения процесса обработки на диаграмме видно, что пропуски в данных исчезли, что и было необходимо сделать.



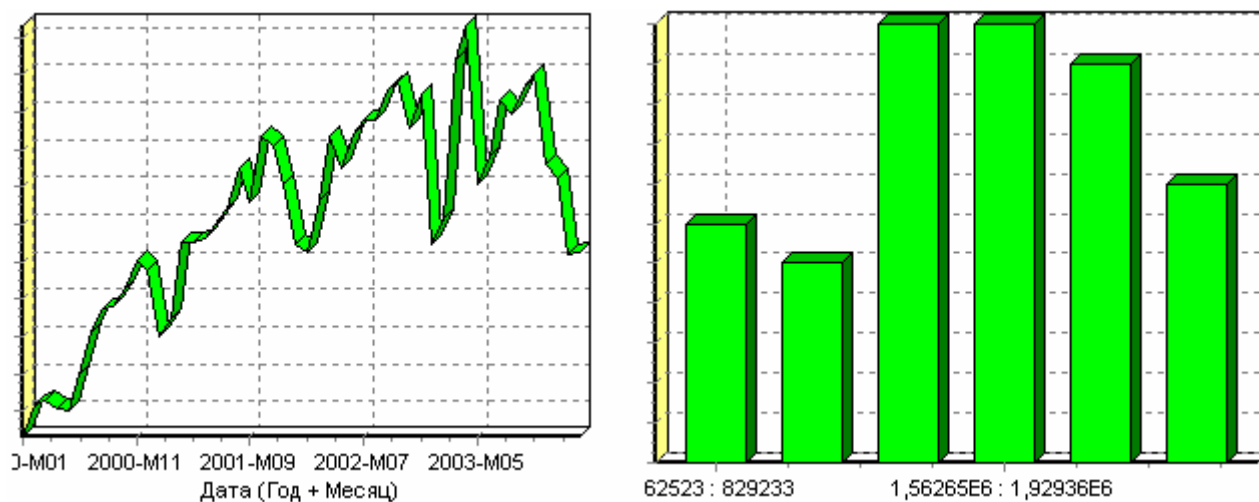
Удаление аномалий

Аномалии встречаются в «сырых» данных не реже шумов. По существу они вообще не должны оказывать никакого влияния на результат. Если же они присутствуют при построении модели, то оказывают на нее весьма большое влияние. Т.е. предварительно их необходимо устранить.

Также они портят статистическую картину распределения данных. К примеру, вот как выглядят данные с аномалиями, а также гистограмма их распределения:



Очевидно, что аномалии не позволяют определить как характер самих данных, так и статистическую картину. После устранения аномалий те же данные представляются в следующем виде:



Этот пример еще раз подчеркивает необходимость проведения парциальной обработки данных перед анализом.

Вернемся к примеру с удалением аномалий из поля «АНОМАЛИИ» импортированной таблицы.

Аргумент
Синус
*Аномалии
Большие шумы
Средние шумы
Малые шумы

Имя столбца COL3
Тип данных Вещественный

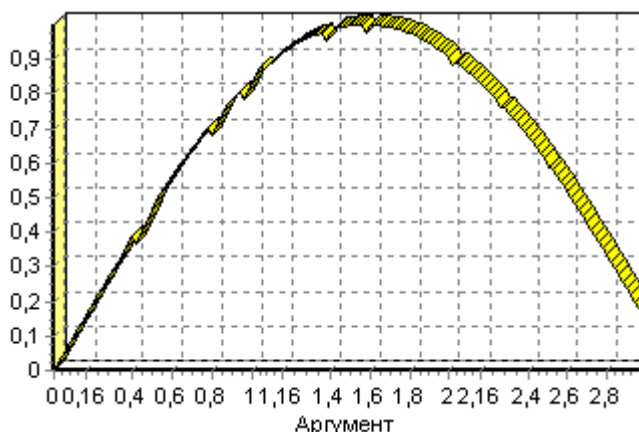
☒ Редактирование аномальных значений
Используется алгоритм робастной фильтрации

Степень подавления Большая

Степень подавления определяет допустимую величину отклонения от нормы (робастной оценки).

В мастере парциальной предобработки на третьем шаге выбираем поле «АНОМАЛИИ» и указываем ему тип обработки «Удаления аномальных явлений», степень подавления «Большая». Так как больше никаких обработок не планировалось, то переходим на шаг запуска процесса обработки и нажимаем «Пуск».

После выполнения процесса обработки на диаграмме видно, что выбросы исчезли, остались лишь небольшие возмущения, которые легко сгладить при помощи спектральной обработки.



Спектральная обработка.

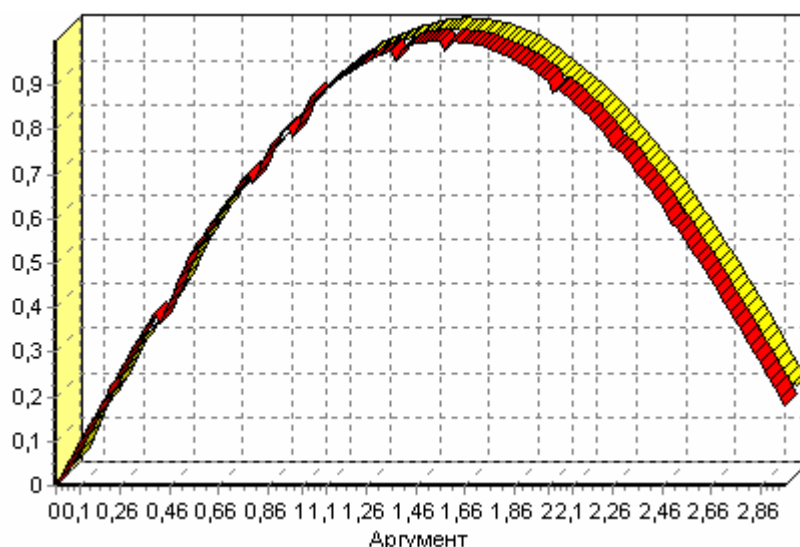
Данные, как мы видим из предыдущего примера, бывает необходимо сгладить. Сглаживание данных применяется для удаления шумов из исходного набора, (что будет продемонстрировано позднее) а также для выделения тенденции, трудно видимой в исходном наборе. Платформа Deductor Studio предлагает несколько видов спектральной обработки: сглаживание данных путем указания полосы пропускания, вычитание шума путем указания степени вычитания шума и вейвлет преобразование путем указания глубины разложения и порядка вейвлета.

Продemonстрируем такой метод спектральной обработки, как вейвлет преобразование. Для этого продолжим работу с данными, полученными в предыдущем примере. Как видно на рисунке, аномалии были устранены, однако небольшие возмущения остались. Сгладим их при помощи парциальной обработки. Для этого после удаления аномалий вновь запустим мастер парциальной обработки. В нем на четвертом шаге выберем поле «АНОМАЛИИ» и укажем ему тип обработки «Вейвлет преобразование» с параметрами по умолчанию (глубина разложения 3, порядок вейвлета 6).

Аргумент	Имя столбца	COL3
Синус	Тип данных	Вещественный
Аномалии		
Большие шумы		
Средние шумы		
Малые шумы		
	<input type="radio"/> Отключить	
	<input type="radio"/> Сглаживание данных	
	Полоса пропускания	50
	<input type="radio"/> Вычитание шума	
	Степень вычитания шума	Малая
	<input checked="" type="radio"/> Вейвлет преобразование	
	Глубина разложения	3
	Порядок вейвлета	6

Так как больше ничего не планировалось, то перейдем с шагу запуска процесса обработки и выполним ее. В качестве визуализатора укажем диаграмму.

После обработки можно убедиться на диаграмме в отсутствии выбросов и сравнить результат с эталонным значением синуса (столбец «СИНУС»). На рисунке красный (темный) график – значения синуса, желтый (светлый) – значения сглаженного синуса после устранения аномалий.



Удаление шумов

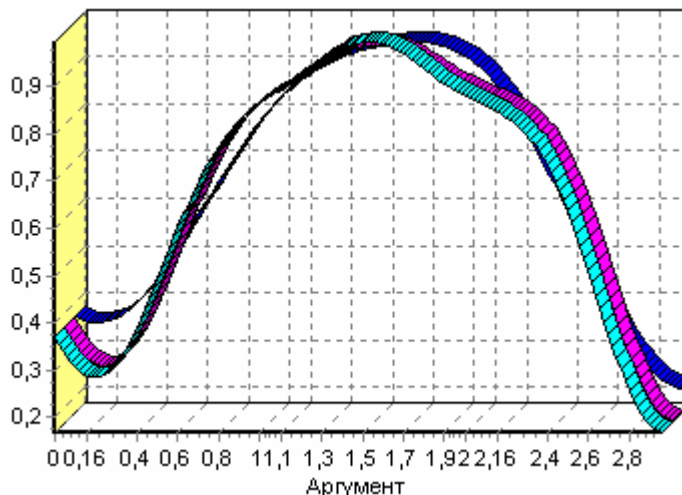
Шумы в данных не только скрывают общую тенденцию, но и проявляют себя при построении модели прогноза. Из-за них модель может получиться с плохими обобщающими качествами.

В примере по парциальной обработке, как было показано ранее, есть 3 столбца с шумами: «БОЛЬШИЕ ШУМЫ», «СРЕДНИЕ ШУМЫ», и «МАЛЫЕ ШУМЫ» - соответственно синус с большими, средними и малыми шумами. Ясно, что для дальнейшей работы с данными эти шумы необходимо устранить. Спектральная обработка, как говорилось ранее, позволяет сделать это с помощью указания для этих полей в качестве типа обработки «Вычитание шума». Настройки обладают определенной гибкостью. Так, существует большая, средняя и малая степень вычитания шума. Аналитик может подобрать степень, устраивающую его.

Удаление больших, малых и средних шумов.

Таким образом, в мастере парциальной обработки на четвертом шаге выберем по очереди поля «БОЛЬШИЕ ШУМЫ», «СРЕДНИЕ ШУМЫ» и «МАЛЫЕ ШУМЫ», зададим тип обработки «Вычитание шума» и укажем степень подавления – «большая», «средняя» и «малая» соответственно.

После выполнения обработки на диаграмме можно просмотреть полученные результаты.

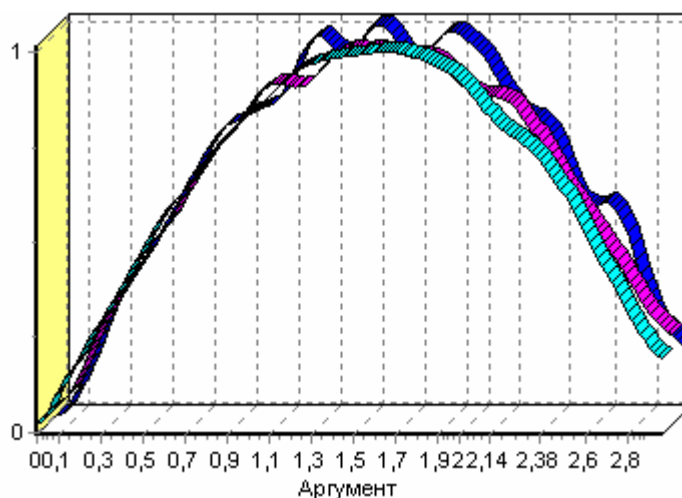


Сглаживание больших, малых и средних шумов

В некоторых случаях неплохие результаты удаления шумов дает вейвлет преобразование. Покажем, какие результаты показывает на этих же данных этот вид спектральной обработки.

В мастере парциальной обработки выберем поля «БОЛЬШИЕ ШУМЫ», «СРЕДНИЕ ШУМЫ» и «МАЛЫЕ ШУМЫ», укажем тип обработки «Вейвлет преобразование», оставив параметры обработки по умолчанию (глубина разложения – 3, порядок вейвлета – 6).

На диаграмме можно убедиться в том, что данные сгладлись. Синий график – сглаженные большие шумы, красный – сглаженные средние и желтый – сглаженные малые шумы. Повысить качество сглаживания шумов таким способом можно, путем подбора удовлетворительных параметров обработки.

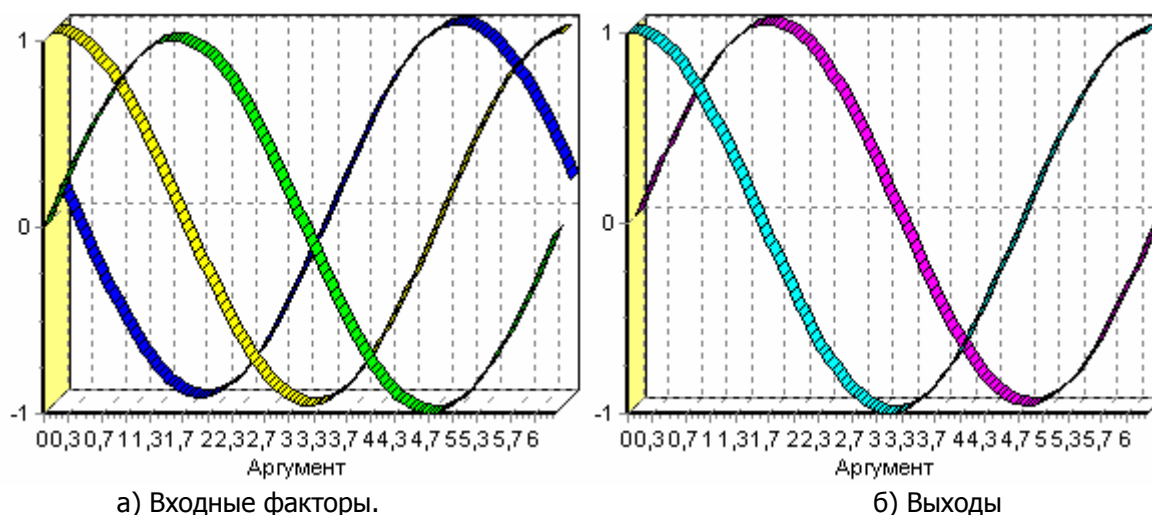


Факторный анализ

Факторный анализ служит для понижения размерности пространства входных факторов. Обработку можно выполнять как в автоматическом режиме (с указанием порога значимости), так и самостоятельно (основываясь на значениях матрицы значимости).

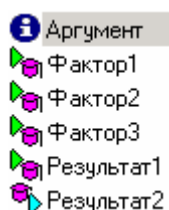
Исходные данные

Рассмотрим применение обработки на примере данных из файла «TestForCPP.txt». Он содержит таблицу со следующими полями: «АРГУМЕНТ» – аргумент, «ФАКТОР1», «ФАКТОР2», «ФАКТОР3» – входные значения, «РЕЗУЛЬТАТ1», «РЕЗУЛЬТАТ2» – выходные значения.



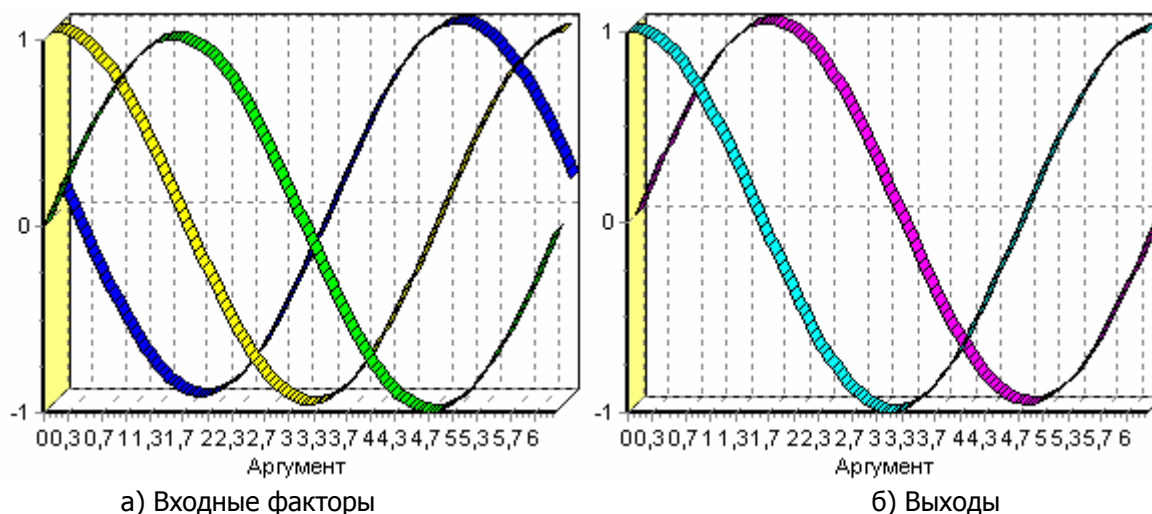
Понижение размерности пространства входных факторов

В мастере факторного анализа зададим «ФАКТОР1», «ФАКТОР2», «ФАКТОР3» входными полями, «РЕЗУЛЬТАТ1», «РЕЗУЛЬТАТ2» – выходными, а поле «АРГУМЕНТ» – непригодным.



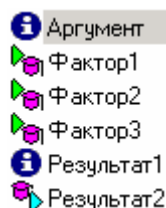
Следующий шаг предлагает запустить процесс понижения размерности пространства входных факторов. После завершения процесса на следующем шаге предлагается выбрать, какие из полученных в результате обработки факторы оставить для дальнейшей работы. Это делается путем указания необходимого порога значимости (по умолчанию порог значимости равен 90%, не будем его менять).

определим степень влияния входных факторов на один из выходов – «РЕЗУЛЬТАТ2» и оставим только значимые факторы.



Устранение незначущих входных факторов

В мастере корреляционного анализа зададим «ФАКТОР1», «ФАКТОР2», «ФАКТОР3» входными полями, «РЕЗУЛЬТАТ2» - выходными, а поля «АРГУМЕНТ» и «РЕЗУЛЬТАТ1» – информационным.



Следующий шаг предлагает запустить процесс корреляционного анализа. После завершения процесса на следующем шаге предлагается выбрать, какие факторы оставить для дальнейшей работы. Это делается либо вручную, основываясь на значениях матрицы ковариации, либо путем указания порога значимости (по умолчанию порог значимости равен 0.05). Из рассчитанной матрицы ковариации видно, что выходное поле «РЕЗУЛЬТАТ2» напрямую зависит от поля «ФАКТОР2» (вообще, значение коэффициента, равное 1.000 говорит о том, что эти поля идентичны), и в меньшей степени от остальных факторов. В данном случае без потери полезной информации можно исключить из дальнейшего рассмотрения «Фактор1» и «Фактор3».

Входные поля	Корреляция с выходными полями	
	Результат2	
<input type="checkbox"/> Фактор1		0,773
<input checked="" type="checkbox"/> Фактор2		1,000
<input type="checkbox"/> Фактор3		-0,773

☐ Ручной выбор незначущих факторов
☒ Автоматический выбор незначущих факторов в соответствии с порогом значимости

Порог значимости:

Теперь необходимо перейти на следующий шаг и выбрать способ визуализации: просмотрим результаты на диаграмме (например, можно убедиться в идентичности полей «Фактор2» и «Результат2»).

Таким образом, корреляционный анализ позволил проанализировать влияние входных факторов на результат и исключить незначущие факторы из дальнейшего анализа.

Трансформация данных

Часть примеров, входящих в группу трансформация данных показывает, как с помощью инструментов Deductor Studio можно добиться тех или иных промежуточных задач, касающихся сбора аналитической информации, либо разбиения данных на какие-либо группы по определенным критериям.

Разбиение данных на группы

Часто для проведения анализа или построения модели прогноза приходится разбивать данные на группы, исходя из определенных критериев. В первом случае такая необходимость возникает, если аналитик желает просмотреть, к примеру, информацию не по всей совокупности данных, а по определенным группам (например, какую сумму кредита берут на те или иные цели, либо кредиторы того или иного возраста). Во втором случае (прогнозирование) аналитику необходимо учитывать тот факт, что определенные группы (в данном случае группы кредиторов) ведут себя по-разному, и что модель прогноза, построенная на всех данных не будет учитывать нюансов, возникающих в этих группах. Т.е. лучше построить несколько моделей прогноза, например, в зависимости от суммовой группы кредита и строить прогноз на них, нежели построить одну модель прогноза. Исходя из этого и не только, в Deductor Studio предоставляется широкий набор инструментов, тем или иным способом позволяющие разбивать исходные данные на группы, группировать любым способом всевозможные показатели и т.п.

Рассмотрим разбиение данных на группы на примере данных по рискам кредитования физических лиц (Файл «Credit.txt»)

Интересующие нас столбцы: «СУММА КРЕДИТА», «ДАТА КРЕДИТОВАНИЯ», «ЦЕЛЬ КРЕДИТОВАНИЯ» и «ВОЗРАСТ».

После импорта данных из текстового файла наиболее информативно просмотреть данные можно с помощью визуализатора «Куб», выбрав в качестве измерений столбцы «ВОЗРАСТ» и «ЦЕЛЬ КРЕДИТОВАНИЯ», а в качестве факта – столбец «СУММА КРЕДИТА». Остальные столбцы установить как непригодные.

<div> <div>0110</div> <div>Сумма кредита</div> </div> <div> <div>✗</div> <div>Стоимость кредита</div> </div> <div> <div>✗</div> <div>Срок кредита</div> </div> <div> <div>✗</div> <div>Дата кредитования</div> </div> <div> <div>✗</div> <div>Дата кредитования (Год + Неделя)</div> </div> <div> <div>↕</div> <div>Цель кредитования</div> </div> <div> <div>✗</div> <div>Количество</div> </div> <div> <div>↕</div> <div>Возраст</div> </div> <div> <div>✗</div> <div>Пол</div> </div> <div> <div>✗</div> <div>Образование</div> </div> <div> <div>✗</div> <div>Частная собственность</div> </div> <div> <div>✗</div> <div>Квартира</div> </div> <div> <div>✗</div> <div>Площадь квартиры</div> </div>	<div>Имя столбца</div> <div>COL1</div> <div>Тип данных</div> <div>Целый</div> <div>Назначение</div> <div>0110 Факт</div> <div>Вид данных</div> <div>Непрерывный</div> <div>Статистическая информация недоступна</div>
--	---

На следующем шаге настройки куба следует указать измерение «ЦЕЛЬ КРЕДИТОВАНИЯ» как измерение в строках, а измерение «ВОЗРАСТ» как измерение в столбцах, перетаскив их с помощью мыши в соответствующие окна из области доступных измерений.

<div>Доступные измерения</div> <div></div>	<div>Измерения в столбцах</div> <div>Возраст</div>				
<div>Измерения в строках</div> <div>Цель кредитования</div>	<table border="1"> <thead> <tr> <th>Факт</th> <th>Агрегация</th> </tr> </thead> <tbody> <tr> <td><input checked="" type="checkbox"/> Сумма кредита</td> <td>Σ Сумма</td> </tr> </tbody> </table>	Факт	Агрегация	<input checked="" type="checkbox"/> Сумма кредита	Σ Сумма
Факт	Агрегация				
<input checked="" type="checkbox"/> Сумма кредита	Σ Сумма				

В итоге, на кросс диаграмме (одна из закладок визуализатора куб) можно просмотреть исходные данные.

	Возраст ▾		
Цель кредитования ▾	19	20	21
Иное	50 000,00	17 000,00	8 500,00
Оплата за образование		17 500,00	29 500,00
Оплата услуг (мед., юрид. и т.п.)			
Покупка и ремонт недвижимости	78 000,00		13 000,00
Покупка товара	46 500,00	73 500,00	76 500,00
Турпоездки, развлечения и т.п.		30 500,00	
Итого	174 500,00	138 500,00	127 500,00

Разбиение даты (по неделям)

Разбиение даты служит для анализа всевозможных показателей за определенный период (день, неделя, месяц, квартал, год). Суть разбиения заключается в том, что на основе столбца с

информацией о дате формируется другой столбец, в котором указывается, к какому заданному интервалу времени принадлежит строка данных. Тип интервала задается аналитиком, исходя из того, что он хочет получить – данные за год, квартал, месяц, неделю, день или сразу по всем интервалам.

Пусть нам необходимо получить данные по суммам взятых кредитов по неделям (в файле «Credit.txt» содержится информация за первые две недели 2003 года).

Для этого в мастере обработки «Дата и Время» на втором шаге выберем поле «ДАТА КРЕДИТОВАНИЯ» используемым, в появившейся после этого таблице настроек выберем назначение

«Используемое» в столбце «Строка» напротив строки «Год + Неделя».

Больше никакие настройки не понадобятся, поэтому перейдем далее к выбору типа визуализации.

Выберем в качестве визуализаторов «Таблицу» и «Куб», поставив галочки в соответствующих позициях. В мастере настройки полей куба выберем в качестве измерения появившийся после обработки столбец «ДАТА КРЕДИТОВАНИЯ_YWStr (Год + Неделя)» и столбец «ЦЕЛЬ КРЕДИТОВАНИЯ», а в качестве факта – «СУММА КРЕДИТА». Остальные поля сделаем неиспользуемыми.

На следующем шаге перенесем одно измерение из области «доступных» в область «Измерения в строках», а другое – в область «Измерения в столбцах».

Таким образом, на кросс диаграмме имеем суммы взятых кредитов по неделям (за первые две недели года) в разрезе целей кредитования.

	Дата кредитования (Год + Неделя) ▾		
Цель кредитования ▾	2003-W01	2003-W02	Итого
Иное	358 000,00	137 000,00	495 000,00
Оплата за образование	62 000,00	312 000,00	374 000,00
Оплата услуг (мед., юрид. и т.п.)	110 500,00	191 000,00	301 500,00
Покупка и ремонт недвижимости	404 000,00	538 000,00	942 000,00
Покупка товара	642 000,00	643 500,00	1 285 500,00
Турпоездки, развлечения и т.п.	35 500,00	113 500,00	149 000,00
Итого	1 612 000,00	1 935 000,00	3 547 000,00

В таблице с данными видно, что новое поле - «ДАТА КРЕДИТОВАНИЯ_YWStr (Год + Неделя)» содержит одинаковые значения (дата начала недели) для строк, которые попадают в одну и ту же неделю (дата начала недели или номер недели с начала года).

Срок кредита	Дата кредитования	Дата кредитования	Цель кредитования
24	05.01.2003	2003-W01	Покупка товара
12	05.01.2003	2003-W01	Покупка товара
30	05.01.2003	2003-W01	Иное
36	06.01.2003	2003-W02	Покупка и ремонт недвижимости
12	06.01.2003	2003-W02	Оплата за образование
18	06.01.2003	2003-W02	Иное
6	06.01.2003	2003-W02	Покупка товара

Квантование возраста кредиторов на 5 интервалов

Часто аналитику необходимо отнести непрерывные данные (например, количество продаж) к какому-либо конечному набору (например, всю совокупность данных о количестве продаж необходимо разбить на 5 интервалов – от 0 до 100, от 100 до 200 и т.д., и отнести каждую запись исходного набора к какому – то конкретному интервалу) для анализа или фильтрации исходя именно из этих интервалов. Для этого в Deductor Studio применяется инструмент квантования (или дискретизации).

Квантование предназначено для преобразования непрерывных данных в дискретные. Преобразование может проходить как по интервалам (данные разбиваются на заданное количество интервалов одинаковой длины), так и по квантилям (данные разбиваются на интервалы разной длины так, чтобы в каждом интервале находилось одинаковое количество данных). В качестве значений результирующего набора данных могут выступать номер интервала, нижняя или верхняя граница интервала, середина интервала, либо метка интервала (значения определяемые аналитиком).

Примером использования данного инструмента может служить разбиение данных о возрасте кредиторов на 5 интервалов (до 30 лет, от 30 до 40, от 40 до 50, от 50 до 60, старше 60 лет). Исходные данные распределяются по пяти интервалам именно так, поскольку, согласно статистике, минимальное значение возраста кредитора 19, а максимальное 69 лет. Это необходимо аналитику для оценки кредиторской активности разных возрастных групп, с целью принятия решения о стимулировании кредиторов в группах с низкой активностью (например, уменьшение стоимости кредита для этих групп) и, быть может, увеличение прибыли в возрастных группах кредиторов с высоким риском (путем предложения дополнительных платных услуг). Причем аналитик желает видеть данные в разрезе по неделям (поэтому продолжим работу на последних полученных данных предыдущего примера).

Воспользуемся мастером квантования.

<ul style="list-style-type: none"> Сумма кредита Стоимость кредита Срок кредита Дата кредитования Дата кредитования (Год + Неделя) Цель кредитования Количество Возраст Пол Образование Частная собственность Квартира Площадь квартиры 	Имя столбца: COL7 Тип данных: Целый Назначение: <input checked="" type="checkbox"/> Используемое Способ: По интервалам Интервалов: 5 Значение: Метка интервала Вид данных: ... Дискретный Минимум: Максимум: Стандартное откл.:
---	--

В нем выберем назначение поля «Возраст» используемым, укажем способ разбиения «По интервалам», зададим количество интервалов равное 5, в качестве значения выберем «Метку интервала».

На следующем шаге мастера определим сами метки соответственно возраста кредиторов: «до 30 лет», «от 30 до 40 лет» и т.д.

Столбцы		Интервалы (изменены)		
Имя	Интервалов	№	Граница	Метка
12 Возраст	5		0	
		0	29	До 30 лет
		1	39	От 30 до 40 лет
		2	49	От 40 до 50 лет
		3	59	От 50 до 60 лет
		4	1000	Старше 60 лет

После обработки выберем в качестве способа отображения «Куб». В мастере укажем «СУММА КРЕДИТА» в качестве факта, «ВОЗРАСТ» и поле «ДАТА КРЕДИТОВАНИЯ (Год + Неделя)» в качестве измерения, остальные поля укажем неиспользуемыми.

Далее перенесем «ВОЗРАСТ» из доступных измерений в «Измерения в строках», а «ДАТА КРЕДИТОВАНИЯ (Год + Неделя)» в «Измерения в столбцах».

На кросс диаграмме теперь видна информация о том, какие суммы кредитов берут кредиторы определенных возрастных групп в разрезе по неделям.

	Дата кредитования (Год + Неделя) ▼		
Возраст ▼	2003-W01	2003-W02	Итого
До 30 лет	721 500,00	795 000,00	1 516 500,00
От 30 до 40 лет	375 000,00	499 000,00	874 000,00
От 40 до 50 лет	195 000,00	362 500,00	557 500,00
От 50 до 60 лет	79 000,00	218 000,00	297 000,00
Старше 60 лет	241 500,00	60 500,00	302 000,00
Итого	1 612 000,00	1 935 000,00	3 547 000,00

Теперь аналитик, получив такие данные, может дать рекомендации о снижении стоимости кредита для лиц, старше 50 лет, либо о применении каких –нибудь других мер, способных привлечь большее количество кредиторов этих групп, либо мер, направленных на то, чтобы кредиторы брали кредит на большие суммы.

Фильтрация данных

Почти всегда исходный набор данных, или набор данных после обработки аналитику необходимо отфильтровать. Фильтрация бывает необходима для разбиения данных на какие либо группы (например, товарные группы) для последующей обработки или анализа данных уже отдельно по каждой группе. Также некоторые данные могут не подходить, или наоборот, подходить для дальнейшего анализа в силу накладываемых условий (например, если на каком – либо этапе

обработки данных были выявлены противоречивые записи, то их необходимо исключить из последующей обработки). Здесь тоже возникает необходимость фильтрации.

Фильтрация позволяет из базового набора данных получить набор данных, удовлетворяющий определенным аналитиком условиям. В Deductor Studio механизм построения условий фильтрации прост для понимания. В окне мастера можно определить несколько элементарных условий фильтрации (<ПОЛЕ> <ОТНОШЕНИЕ> <ЗНАЧЕНИЕ>), последовательно связанных логическими операциями (И, ИЛИ).

Рассмотрим ситуацию, когда аналитику необходимо спрогнозировать кредитоспособность потенциального кредитора. Предполагается, что кредиторы, берущие суммы разного диапазона ведут себя по-разному, следовательно, модели прогноза должны свои для каждой группы. Т.е. для дальнейшего построения моделей прогноза кредитоспособности определенных аналитиком категорий необходимо использовать фильтрацию.

Определим, для примера группу кредиторов, взявших кредит менее 10000 руб. Воспользуемся данными предыдущего примера. Для этого, находясь на узле импорта данных из текстового файла, запустим мастер обработки. В нем в качестве метода обработки выберем фильтрацию. На втором шаге мастера можно видеть одно неопределенное условие фильтрации (при необходимости их можно добавлять или удалять соответствующими кнопками на форме). Поскольку необходимо отфильтровать данные только по кредиторам, взявшим кредит менее 10000, то в графе «Имя поля» выбираем поле «СУММА КРЕДИТА», в графе «Условие» выбираем знак меньше, в графе «Значение» пишем «10000».

Операция	Имя поля	Условие	Значение
	12 Сумма кредита	<	10000

Больше никаких условий не требуется, поэтому переходим на следующий шаг мастера и запускаем процесс фильтрации. После выполнения обработки можно манипулировать уже только с данными по кредиторам выбранного кредитного диапазона. В правильности выполненной операции можно легко убедиться, выбрав в качестве визуализации данных статистику и просмотрев значения минимального и максимального значения поля «СУММА КРЕДИТА».

	Метка столбца		
		Минимум	Максимум
1	12 Сумма кредита	2000	9500
2	12 Стоимость кред...	400	1900
3	12 Срок кредита	6	6
4	7 Дата кредитова...	01.01.2003	11.01.2003

Калькулятор

Иногда возникает необходимость на каком-либо этапе обработки данных получить на их основе новые (производные) данные. Возможно, аналитику требуется вычислить процентное отклонение значения одного поля относительно другого, либо подсчитать сумму, разность полей, получить на основе данных показатель и уже его использовать для дальнейшей обработки, в зависимости от значения полей вычислить те или иные выражения. В Deductor Studio такую возможность предоставляет инструмент «Калькулятор». Он позволяет создавать новые поля, вычисляющие заданные аналитиком выражения. Т.е. калькулятор служит для получения производных данных на основе имеющихся в исходном наборе. Мастер предоставляет широкий набор функций различного направления. В мастере представлен список новых выражений, где добавляются необходимые аналитику выражения, список доступных функций с кратким описанием каждой, список доступных операций и также список доступных столбцов, которые можно задействовать при создании выражения.

Замечание: Реализованный в Deductor Studio конструктор выражений при построении использует не метки (Сумма, Количество, Цена ...), а имена полей таблицы, заданные в источнике данных (Summ, Count, Price...). При импорте в некоторых случаях (напр. из текстового файла) можно задать как метки, так и имена импортируемых полей. В следующем примере метками являются «АРГУМЕНТ1», «АРГУМЕНТ2», «АРГУМЕНТ3», а именами соответственно «COL1», «COL2», «COL3». При желании как метки, так и имена полей можно изменить на более информативные, используя обработчик «Настройка набора данных».

Исходные данные

Рассмотрим применение на примере данных из файла «Calculate.txt». В нем содержится таблица с полями «АРГУМЕНТ1», «АРГУМЕНТ2», «АРГУМЕНТ3» – набор аргументов. Для начала необходимо импортировать данный файл в программу. Для просмотра исходных данных в данном случае удобнее использовать визуализатор «Таблица».

	Аргумент1	Аргумент2	Аргумент3
	0	4	4
	0	5	5
	0	6	6
	0	7	7
	0	8	8

Пусть необходимо на основе аргументов рассчитать некоторые математические функции. Пусть это будут две функции одного аргумента (АРГУМЕНТ3), одна функция от двух аргументов, одна кусочно-заданная функция и функция, показывающая относительное отклонение (АРГУМЕНТ1 + 1 от АРГУМЕНТ2 + 1). Предполагается, что все эти функции будут использоваться для последующей обработки.

Функции $F1(\text{АРГУМЕНТ3})$, $F2(\text{АРГУМЕНТ3})$

Рассчитаем значение функций $\text{SIN}(\text{АРГУМЕНТ3} * \text{АРГУМЕНТ3}) * \text{LN}(\text{АРГУМЕНТ3} + 1) * \text{EXP}(-\text{АРГУМЕНТ3} / 10)$ и $10 * \text{SIN}(\text{АРГУМЕНТ3} * \text{АРГУМЕНТ3} / 100) / (\text{АРГУМЕНТ3} + 1) * \text{EXP}(-\text{АРГУМЕНТ3} / 10)$.

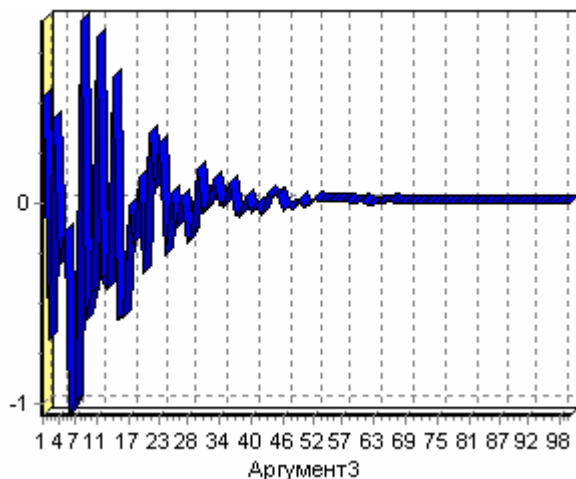
Для этого, находясь на узле импорта, запустим мастер обработки. Выберем в качестве обработчика калькулятор. На втором шаге мастера в списке выражений в первой строке в графе «Название выражения» вместо надписи «Выражение» напомним $F1(\text{АРГУМЕНТ3})$. В поле редактора выражения (в верхней части мастера) напомним « $\text{SIN}(\text{COL3} * \text{COL3}) * \text{LN}(\text{COL3} + 1) * \text{EXP}(-\text{COL3} / 10)$ ».

Название выражения	↑	$\text{SIN}(\text{COL3} * \text{COL3}) * \text{LN}(\text{COL3} + 1) * \text{EXP}(-\text{COL3} / 10)$
9.0 Функция F1(Аргумент3)	↓	
	+	
	+!	
	▲	
	▼	

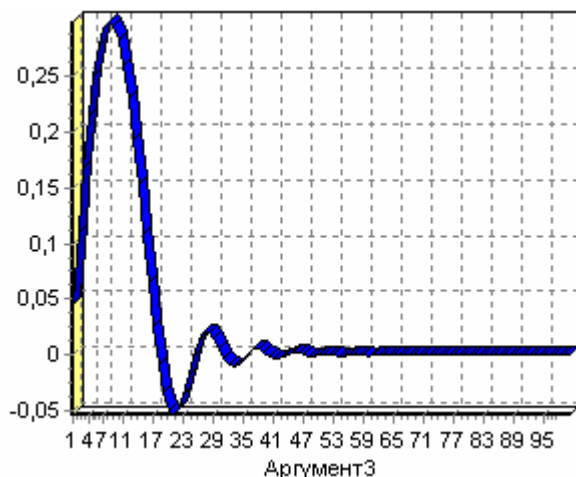
Имя столбца	Метка столбца
12 COL1	Аргумент1
12 COL2	Аргумент2
12 COL3	Аргумент3

Операции							
" "	{ }	+	-	*	/		
=	<>	<	>	<=	>=		
and	or	not	xor	true	false		
fx Функция							

Таким образом, мы создали новый столбец, задали ему название «F1(АРГУМЕНТ3)» и также определили, какие значения будут принимать записи этого поля. На этом создание вычисляемого значения окончено, поэтому переходим на следующий шаг мастера, где предлагается выбрать способ отображения данных. Самым информативным в данном случае является диаграмма, которую и следует выбрать. Далее, выбрав в мастере настроек диаграммы в качестве отображаемого поля «F1(АРГУМЕНТ3)», в качестве типа графика «Линии», в качестве подписей по оси X значения поля «АРГУМЕНТ3» можно увидеть график вычисленной функции.



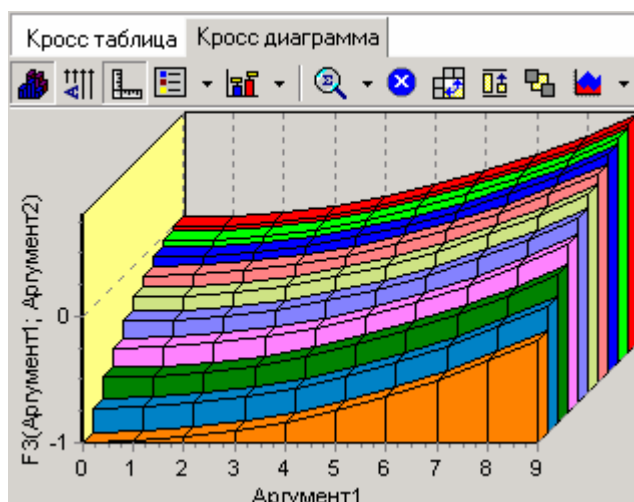
Сложная функция F2(АРГУМЕНТ3) отличается только видом функции («10*SIN(COL3 * COL3/100)/(COL3+1)*EXP(-COL3/10)»)



Функция от двух аргументов $F_3(\text{АРГУМЕНТ1}; \text{АРГУМЕНТ2})$

Данная функция интересна тем, что для ее просмотра в трех измерениях можно использовать визуализатор «Куб». Зададим название выражения « $F_3(\text{АРГУМЕНТ1}; \text{АРГУМЕНТ2})$ », в поле вычисляемого выражения напишем « $\text{COL1} * \text{COL1} / 100 - \text{COL2} * \text{COL2} / 100$ ». Выберем визуализатор «Куб» и настроим его так, что «АРГУМЕНТ1» и «АРГУМЕНТ2» являлись бы измерениями, $F_3(\text{АРГУМЕНТ1}; \text{АРГУМЕНТ2})$ – фактом, а «АРГУМЕНТ3» – неиспользуемым.

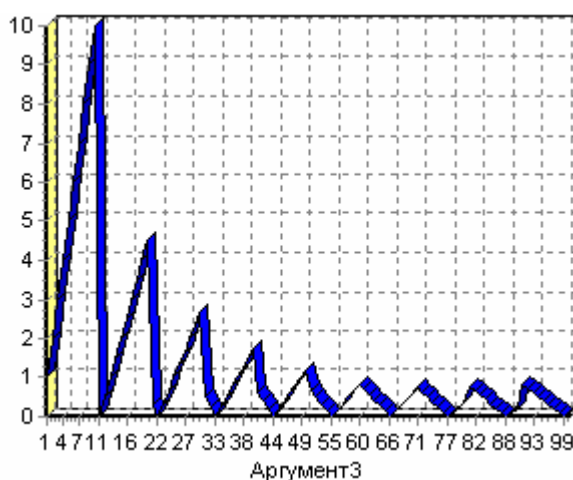
Выбрав «АРГУМЕНТ1» измерением в столбцах, а «АРГУМЕНТ2» – измерениям в строках перейдем к просмотру Кросс-диаграммы. Для более наглядного просмотра установим тип диаграммы «области». Теперь можно посмотреть вычисленную функцию в объемном виде.



Вычисление отклонения $\text{АРГУМЕНТ1}+1$ от $\text{АРГУМЕНТ2}+1$

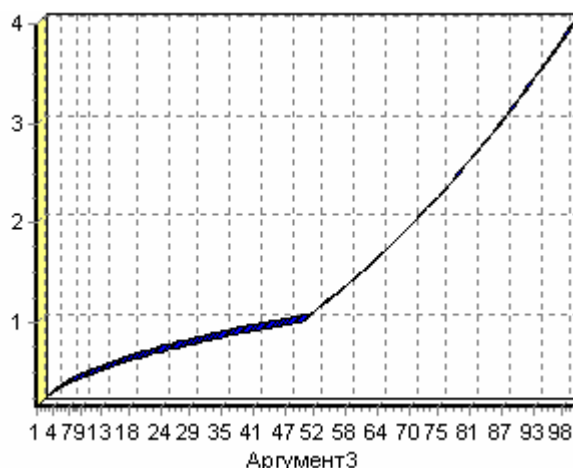
Покажем пример применения одной из встроенных функций – вычисление долевого отклонения одного аргумента от другого (RELDEV). Список всех встроенных функций вместе с описанием можно посмотреть в мастере нажав на кнопку Функция.

Задав в качестве вычисляемого выражения $\text{RELDEV}(\text{COL1} + 1; \text{COL2} + 1)$ можно на диаграмме увидеть данное отклонение.



Пример кусочно-заданной функции.

Пусть функция принимает значения $\text{SQRT}(\text{АРГУМЕНТ3}/50)$ (квадратный корень) при значениях АРГУМЕНТ3 от 0 до 50 и значения $\text{АРГУМЕНТ3} \cdot \text{АРГУМЕНТ3}/2500$ при остальных. Для вычисления подобной функции необходимо воспользоваться имеющейся в наличии функцией $\text{IFF}(\text{аргумент1}; \text{аргумент2}; \text{аргумент3})$, которая позволяет в зависимости от логического значения первого аргумента получить второй или третий аргумент. Согласно примеру, если значение аргумента больше нуля и меньше 50 необходимо получить выражение $\text{SQRT}(\text{АРГУМЕНТ3}/50)$, в противном случае – выражение $\text{АРГУМЕНТ3} \cdot \text{АРГУМЕНТ3}/2500$. Таким образом, в поле построения выражения необходимо написать « $\text{IFF}((\text{COL3} > 0) \text{ AND } (\text{COL3} < 50)); \text{SQRT}(\text{COL3}/50); \text{COL3} \cdot \text{COL3}/2500)$ ». Сделав это в мастере обработки «Калькулятор», и выбрав далее визуализатор «Диаграмма», и также выбрав в мастере настройки диаграммы поле со значениями кусочно-заданной функции, можно посмотреть на требуемый результат.



Группировка данных

Сложно делать выводы на основе необработанной первичной информации. Аналитика для принятия решения почти всегда нужна сводная информация. Сводные данные намного более информативны, тем более, если их можно получить в различных разрезах. В Deductor Studio предусмотрен инструмент, реализующий сбор сводной информации – «Группировка». Группировка позволяет объединять записи по полям - измерениям и агрегируя данные в полях-фактах для дальнейшего анализа.

Исходные данные

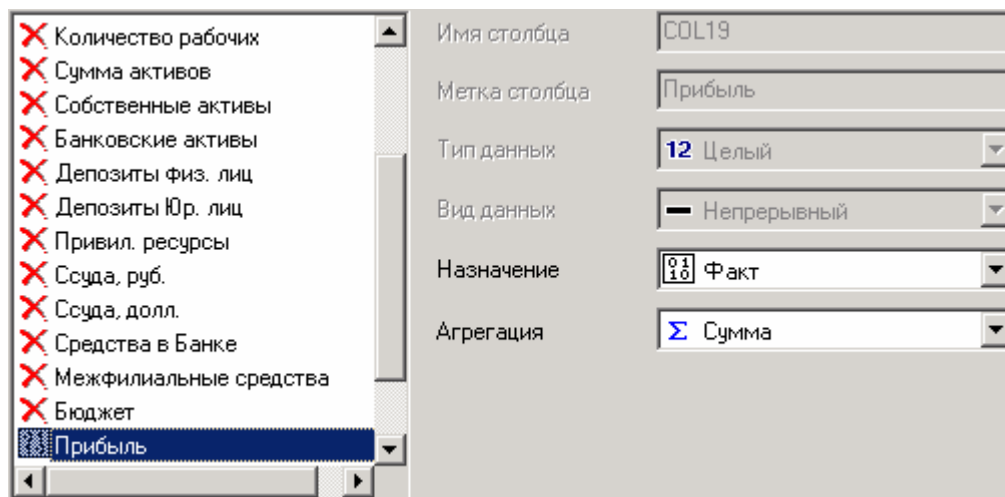
Допустим, что у аналитика имеется статистика по банкам России за определенный период. Она находится в файле «banks.txt». Перед ним стоит задача выявления ряда городов, в которых прибыль банков самая большая для использования этих данных в дальнейшем. Для этого аналитик должен обратить внимание на следующие поля таблицы из файла: «БАНК», «ФИЛИАЛЫ», «ГОРОД», «ПРИБЫЛЬ». Т.е. информация о названии банка, городе, в котором он находится (филиалы банка могут находиться в разных городах – следовательно, по одному и тому же банку может быть несколько записей с данными по разным городам) и прибыль банка.

Ясно, что для решения поставленной задачи первым делом необходимо найти суммарную прибыль всех банков в каждом городе. Для этого и необходима группировка.

Для начала следует импортировать данные по банкам из текстового файла. Просмотреть исходную информацию можно в виде куба, где по строкам будут названия банков, а по столбцам – города. С помощью визуализатора «Куб» также можно получить требуемую информацию, выбрав в качестве измерения поле «ГОРОД», а в качестве факта «ПРИБЫЛЬ». Но нам необходимо получить эти данные для последующей обработки, следовательно, необходимо сделать аналогичную группировку.

Группировка по городам

Находясь в узле импорта, запустим мастер обработки. Выберем в качестве обработки группировку данных. На втором шаге мастера установим назначение поля «ГОРОД» как измерение, а назначение поля «ПРИБЫЛЬ» как факт. В качестве функции агрегации у поля «ПРИБЫЛЬ» следует указать Сумму.



Таким образом, после обработки получим суммарные данные по прибыли всех банков по каждому городу. Их можно просмотреть, используя таблицу. Теперь аналитику можно выполнять следующий этап обработки данных.

Город	Прибыль
▶ Санкт-Петербург	128 038
Владивосток	17 152
Вологда	35 144
Екатеринбург	125 126
Казань	68 576
Краснодар	26 991

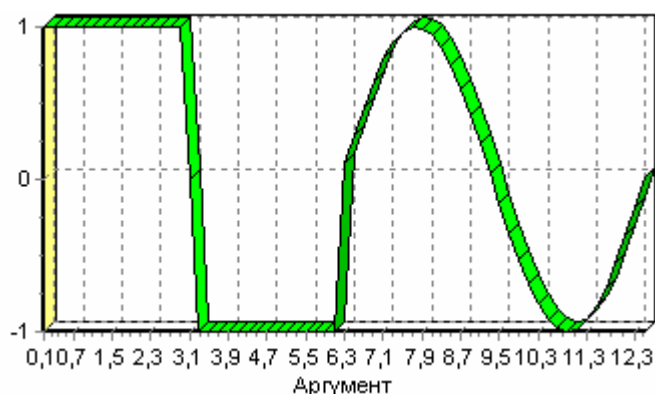
Преобразование данных к скользящему окну

Когда требуется прогнозировать временной ряд, тем более, если налицо его периодичность (сезонность), то лучшего результата можно добиться, учитывая значения факторов не только в данный момент времени, но и, например, за аналогичный период прошлого года. Такую возможность можно получить после трансформации данных к скользящему окну. То есть, например, при сезонности продаж с периодом 12 месяцев, для прогнозирования количества продаж на месяц вперед можно в качестве входного фактора указать не только значение количества продаж за предыдущий месяц, но и за 12 месяцев назад.

Обработка создает новые столбцы путем сдвига данных исходного столбца вниз и вверх (глубина погружения, горизонт прогноза).

Исходные данные

Продemonстрируем сам принцип трансформации данных, используя данные из файла «Sliding.txt». В нем всего 2 поля – «АРГУМЕНТ» - аргумент (время), «ФУНКЦИЯ» – временной ряд. Импортируем данные из файла (необходимо указать тип полей – вещественный) и построим диаграмму.

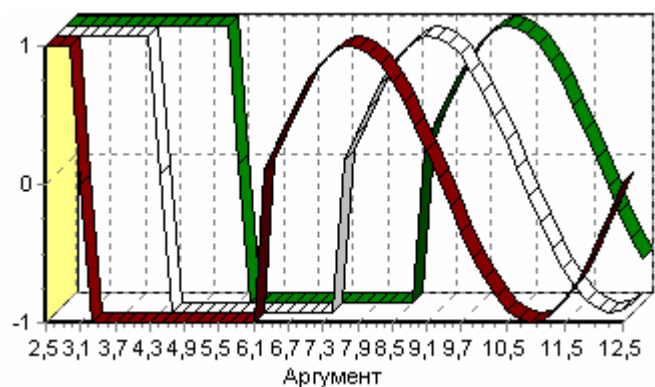


Преобразование скользящим окном

В мастере преобразования укажем назначение столбца «ФУНКЦИЯ» используемым, установим для него глубину погружения 12.

Аргумент	Имя столбца	COL2
✓ Функция	Тип данных	Вещественный
	Назначение	✓ Используемое
	Глубина погружения	12
	Горизонт прогнозирования	0
	<input type="checkbox"/> Оставлять неполные записи	

После трансформации были получены новые столбцы – «ФУНКЦИЯ - 12», ... «ФУНКЦИЯ - 2», «ФУНКЦИЯ - 1» на основе столбца «ФУНКЦИЯ». Если на диаграмме посмотреть несколько таких столбцов, то видно, что данные в них сдвинуты относительно друг друга.



Настройка набора данных

Настройка набора данных применяется, когда необходимо изменить имя, метку, размер, тип, вид и назначение полей текущей таблицы данных для более удобного дальнейшего использования.

Замечание: Данный обработчик аналогичен шагу мастера настройки полей при импорте данных в программу, рассмотренному выше.

Исходные данные

Продemonстрируем использование настройки полей, используя данные, полученные после квантования возраста кредиторов на интервалы из примера выше. Пусть необходимо изменить метку поля «Дата кредитования (Год + Неделя)» на более информативную метку при подготовке отчетности - «Год и неделя кредитования». Пусть также, для дальнейшего использования необходимо установить размер поля «Цель кредитования» 30 символов и необходимо использовать поле «Срок кредита» как дискретное.

Выполнение настройки

В мастере настройки выделим столбец «Дата кредитования (Год + Неделя)» и укажем ему новую метку. Подобные действия по изменению произведем и с другими полями.

12 Сумма кредита
12 Стоимость кредита
12 Срок кредита
7 Дата кредитования
ab **Год и Неделя кредитования**
ab Цель кредитования
12 Количество
ab Возраст
ab Пол
ab Образование
ab Частная собственность
ab Квартира

Имя столбца: COL4_YWStr
Метка столбца: Год и Неделя кредитования
Тип данных: ab Строковый
Вид данных: ... Дискретный
Назначение: i Информационное

Сброс настроек

☐ Кэшировать результирующий набор данных

После настройки полей, полученный отчет, представленный в виде кросс-таблицы, будет выглядеть следующим образом:

		Давать кредит ▼		
Год и Неделя кредитования ▼	Срок кредита ▼	Да	Нет	Итого
2003-w01	6	194 500,00	7 500,00	202 000,00
	12	244 500,00	176 500,00	421 000,00
	18	48 000,00	249 000,00	297 000,00
	24		195 500,00	195 500,00
	30		151 000,00	151 000,00
	36		229 000,00	229 000,00
	42		58 500,00	58 500,00
	48		58 000,00	58 000,00
	Итого	487 000,00	1 125 000,00	1 612 000,00
2003-w02	6	183 500,00	25 500,00	209 000,00
	12	109 000,00	373 500,00	482 500,00

Как видно, требуемый результат был достигнут.

Замена значений

Данный обработчик предназначен для замены значений по таблице подстановок, которая содержит пары, состоящие из исходного и измененного значения. Например: «кр» - «красный», «зел» -

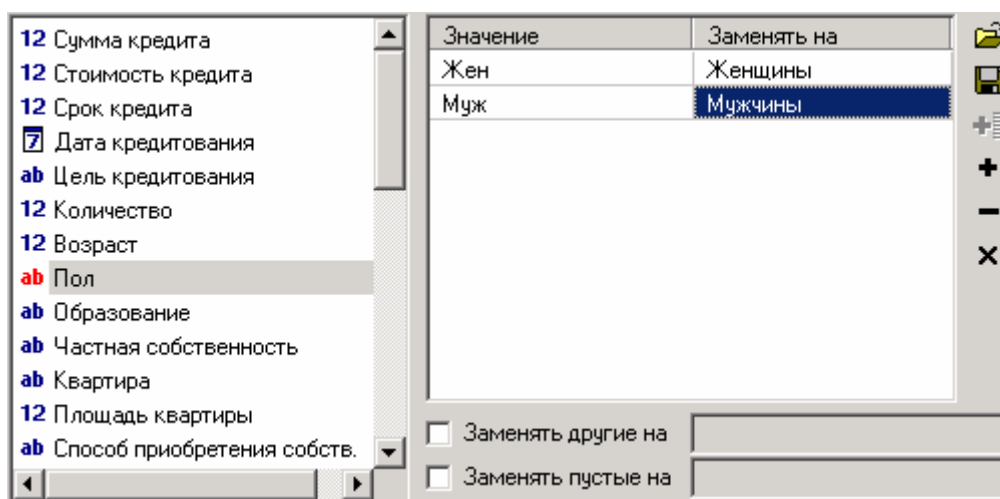
«зеленый», «син» - «синий». Или «зима» - «январь», «весна» - «апрель», «лето» - «июль», «осень» - «октябрь». Кроме того, замену часто используют для замены пустых значений на константу.

Исходные данные

Продemonстрируем использование замены значений, используя данные по кредитованию (файл «Credit.txt»). Пусть необходимо представить отчет о суммах кредитов на различные цели по мужчинам и женщинам. Для повышения информативности заменим значения столбца «Пол». Например, так: «муж» - «мужчины», «жен» - «женщины».

Выполнение замены

В мастере замены необходимо выделить столбец «Пол» и нажать на кнопку «Добавить список». В появившемся списке необходимо пометить галочками оба значения и нажать на «Ок». Выбранные значения добавятся в таблицу подстановок. Далее следует указать, на что заменять исходные значения. В соответствии с задачей напомним напротив «муж» - «Мужчины», напротив «жен» - «Женщины». Далее перейдем на следующий шаг мастера и выберем в качестве варианта визуализации «Куб». Укажем в качестве измерений поля «Пол» и «Цель кредитования», а в качестве факта «Сумма кредита». Остальные поля укажем «неиспользуемыми».



После замены значений полученный отчет, представленный в виде кросс-таблицы, будет выглядеть следующим образом:

	Пол		
Цель кредитования	Женщины	Мужчины	Итого
Иное	233 500,00	261 500,00	495 000,00
Оплата за образование	162 000,00	212 000,00	374 000,00
Оплата услуг (мед., юрид. и т.п.)	198 500,00	103 000,00	301 500,00
Покупка и ремонт недвижимости	343 500,00	598 500,00	942 000,00
Покупка товара	730 500,00	555 000,00	1 285 500,00
Турпоездки, развлечения и т.п.	72 000,00	77 000,00	149 000,00
Итого	1 740 000,00	1 807 000,00	3 547 000,00

Слияние

Обработчик "Слияние" предназначен для объединения двух таблиц по нескольким одинаковым полям. Обработчик применяется, например, для добавления в таблицу с данными о продажах данных по

остаткам за те же месяца. Различают две таблицы: исходная и присоединяемая. К исходной таблице добавляются новые поля, значения которых берутся из присоединяемой таблицы. Количество строк исходной таблицы остается неизменным.

Исходные данные

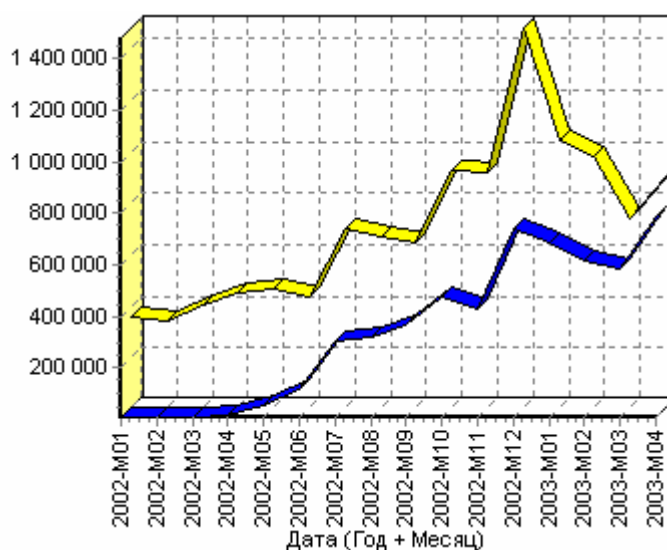
Продemonстрируем использование слияния, используя данные по продажам и остаткам (файлы «TradeSales.txt» и «TradeRest.txt» соответственно). Добавим к данным по продажам данные по остаткам. Для этого сначала импортируем данные из файла, содержащего данные по продажам, а затем запустим мастер обработки и выберем обработчик «Слияние».

Выполнение слияния

В мастере слияния сначала необходимо выбрать источник данных для слияния. Данные шаги аналогичны шагам мастера импорта данных. Так что импортируем данные из текстового файла с остатками «TradeRest.txt». Далее необходимо установить связь между наборами данных, а именно, указать соответствие импортируемого поля имеющемуся полю (**измерение** - для связи двух таблиц) и указать, какие новые поля добавить при слиянии (**факт** с указанием способа агрегации).

Импортируемое поле	Назначение	Поле/Агрегировать
ab Дата (Год + Месяц)	Измерение	Дата (Год + Месяц)
9.0 Остаток (количество)	Факт	Σ Сумма

После указания параметров полей, как показано на рисунке выше, необходимо перейти на следующий шаг мастера и запустить процесс слияния. Полученные результаты, представленные в виде диаграммы будут выглядеть следующим образом:



Как видно, при помощи слияния удалось объединить объем продаж с объемом остатков.

Выявление дубликатов и противоречий

Бывают ситуации, когда проблема неочищенных данных не позволяет построить хорошую модель прогнозирования вообще. Такое происходит, если в наборе данных для прогноза содержатся строки с одинаковыми входными факторами, но разными выходными. В такой ситуации непонятно, какое результирующее значение верное – налицо противоречие. Если противоречивые использовать для построения модели прогноза, то модель окажется неадекватной. Поэтому противоречивые данные, чаще всего, лучше вообще исключить из исходной выборки. Также в данных могут встречаться записи с одинаковыми входными факторами и одинаковыми выходными, т.е. дубликаты. Таким образом,

данные несут избыточность. Присутствие дубликатов в анализируемых данных можно рассматривать как способ повышения «значимости» дублирующейся информации. Иногда они даже необходимы, например, если при построении модели нужно особо выделить некоторые наборы значений. Но все равно, включение в выборку дублирующей информации должно происходить осознанно: в большинстве случаев дубликаты в данных являются следствием ошибок при подготовке данных.

Так или иначе, возникает задача выявления дубликатов и противоречий. В Deductor Studio для автоматизации этого процесса есть соответствующий инструмент – обработка «Дубликаты и противоречия».

Суть обработки состоит в том, что определяются входные (факторы) и выходные (результаты) поля. Алгоритм ищет во всем наборе записи, для которых одинаковым входным полям соответствуют одинаковые (дубликаты) или разные (противоречия) выходные поля. На основании этой информации создаются два дополнительных логических поля – «Дубликат» и «Противоречие», принимающие значения «правда» или «ложь». В дополнительные числовые поля «Группа дубликатов» и «Группа противоречий» записываются номер группы дубликатов и группы противоречий, в которые попадает данная запись. Если запись не является дубликатом или противоречием, то соответствующее поле будет пустым.

Исходные данные

Рассмотрим механизм выявления дубликатов и противоречий на примере данных файла «MultTable.txt». В нем находится таблица умножения двух целых аргументов в диапазоне от 1 до 10. Таблица имеет четыре поля: «АРГУМЕНТ1», «АРГУМЕНТ2» – аргументы, «ПРОИЗВЕДЕНИЕ», «ПРОИЗВЕДЕНИЕ С ПРОТИВОРЕЧИЯМИ» – произведение аргументов, содержащее противоречия. Данные подготовлены следующим образом: сначала идет 100 строк таблицы умножения (от 1*1 до 10*10), причем в поле «ПРОИЗВЕДЕНИЕ С ПРОТИВОРЕЧИЯМИ» в некоторых строках содержатся неверный результат умножения (например, «АРГУМЕНТ1» = 1, «АРГУМЕНТ2» = 5, «ПРОИЗВЕДЕНИЕ» = 5, «ПРОИЗВЕДЕНИЕ С ПРОТИВОРЕЧИЯМИ» = 10). Следующие 50 строк дублируют первые 50, причем значения поля «ПРОИЗВЕДЕНИЕ С ПРОТИВОРЕЧИЯМИ» содержат верный результат умножения. Таким образом, данные содержат ряд строк с одинаковыми входными значениями, но разными выходными и строки с одинаковыми входными и выходными значениями. Т.е. присутствуют дубликаты и противоречия. Остается только обнаружить их.

Импортируем данные из текстового файла и посмотрим их в виде таблицы.

	Аргумент1	Аргумент2	Произведение	Произведение с противоречиями
▶	1	1	1	1
	1	2	2	2
	1	3	3	3
	1	4	4	4
	1	5	5	10
	1	6	6	6

Поиск дубликатов и противоречий

Для выявления дубликатов и противоречий запустим мастер обработки. В нем выберем тип обработки «Дубликаты и противоречия».

На втором шаге мастера необходимо настроить назначение полей. В данном случае входными полями являются «АРГУМЕНТ1» и «АРГУМЕНТ2», а выходным «ПРОИЗВЕДЕНИЕ С ПРОТИВОРЕЧИЯМИ».

Аргумент1	Имя столбца	COL4
Аргумент2	Метка столбца	Произведение с противоречиями
Произведение	Тип данных	12 Целый
Произведение с противоречиями	Вид данных	Непрерывный
	Назначение	Выходное

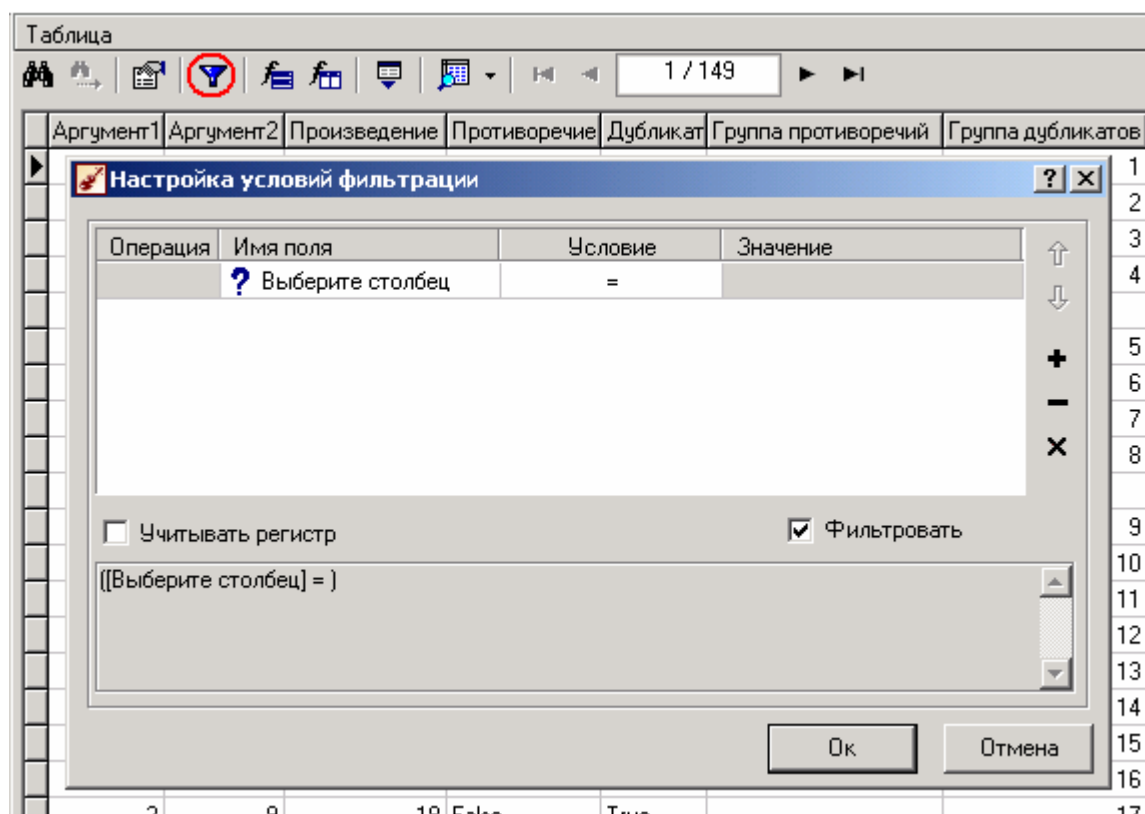
На следующем шаге необходимо запустить процесс обработки.

После завершения выявления дубликатов и противоречий посмотрим результат в виде таблицы.

	Аргумент1	Аргумент2	Произведение	Противоречие	Дубликат	Группа противоречий	Группа дубликатов
▶	1	1	1	False	True		1
■	1	2	2	False	True		2
■	1	3	3	False	True		3
■	1	4	4	False	True		4
■	1	5	5	True	False	1	
■	1	6	6	False	True		5
■	1	7	7	False	True		6
■	1	8	8	False	True		7
■	1	9	9	False	True		8
■	1	10	10	True	False	2	
■	2	1	2	False	True		9

В четырех новых столбцах как раз и находится интересующая нас информация: какие записи являются дубликатами, какие – противоречиями, к какой группе дубликатов или противоречий относятся.

Аналитик может также отфильтровать данные в таблице для просмотра только дубликатов или только противоречий. Покажем, как это можно сделать. Нажав на кнопку фильтрации таблицы, появится мастер настроек условий фильтра (аналогичный обработчику «Фильтрация», рассмотренного ранее).



Для просмотра только дубликатов необходимо задать условие «Дубликат истина»
После ввода условия необходимо нажать на кнопку «Ок» и в таблице будут только дубликаты.

	Аргумент1	Аргумент2	Произведение	Противоречие	Дубликат	Группа противоречий	Группа дубликатов
▶	1	1	1	False	True		1
	1	2	2	False	True		2
	1	3	3	False	True		3
	1	4	4	False	True		4
	1	6	6	False	True		5

Аналогично отфильтруем только противоречия.

	Аргумент1	Аргумент2	Произведение	Противоречие	Дубликат	Группа противоречий	Группа дубликатов
▶	1	5	5	True	False	1	
	1	10	10	True	False	2	
	1	5	5	True	False	1	
	1	10	10	True	False	2	

Примеры анализа данных

Основное направление программы Deductor Studio – анализ, прогнозирование, классификация и кластеризация данных. Предыдущие примеры в основном касались только подготовки данных для последующего анализа. Программа предоставляет следующие механизмы анализа: нейронные сети, линейный регрессионный анализ, построение деревьев решений, самоорганизующиеся карты Кохонена, прогнозирование временного ряда, обнаружение дубликатов и противоречий.

Рассмотрим принцип работы каждого из этих механизмов на последующих примерах.

Прогнозирование умножения с помощью нейронных сетей

Нейросети – механизм, который используют для прогнозирования и решения задач классификации. Они применяются в основном там, где существует нелинейные зависимости результата от входных факторов.

Исходные данные

Рассмотрим прогнозирование с помощью нейронных сетей на примере прогнозирования результата умножения двух чисел – файл «multi.txt»

В нем содержится таблица со следующими полями: «АРГУМЕНТ1», «АРГУМЕНТ2» – множители, «ПРОИЗВЕДЕНИЕ» – их произведение.

Импортировав данные из файла, можно посмотреть результат умножения, используя таблицу.

	Аргумент1	Аргумент2	Произведение
▶	1	0	0
	0	1	0
	3	0	0
	0	3	0
	5	0	0

Прогнозирование результата умножения

Пусть необходимо построить модель прогноза умножения, подавая на вход которой два множителя получать на выходе их произведение. Для этого необходимо, находясь на узле импорта, открыть мастер обработки. В нем выбрать в качестве обработки нейронную сеть и перейти к следующему шагу мастера. На втором шаге мастера необходимо установить назначение полей «АРГУМЕНТ1» и «АРГУМЕНТ2» как входные, а поле «ПРОИЗВЕДЕНИЕ» – как выходное.

Аргумент1	Имя столбца	COL1
Аргумент2	Тип данных	Целый
Произведение	Назначение	Входное
	Вид данных	Непрерывный
Статистика		
	Минимум	0
	Максимум	10
	Среднее	5,265625
	Стандартное откл.	3,21248632049617
Настройка нормализации...		

На следующем шаге предлагается настроить разбиение исходного множества данных на обучающее тестовое и валидационное. Здесь необходимо только указать способ разбиения исходного множества данных «Случайно».

Способ разделения исходного множества данных		Случайно	
Множество	Размер		Порядок сортировки
	В процентах	В строках	
<input checked="" type="checkbox"/> Обучающее	95,00	61	По возрастанию
<input checked="" type="checkbox"/> Тестовое	5,00	3	По возрастанию
<input type="checkbox"/> Валидационное	0,00	0	По возрастанию
ИТОГО:	100,00	64	

На следующем шаге необходимо указать количество нейронов в скрытом слое – 1, остальное можно оставить по умолчанию.

Нейроны в слоях		Активационная функция	
входном:	2	Тип функции	Сигмоида
скрытых слоев:	1	Крутизна	1,000
выходном:	1		
Слой	Нейроны		
1	1		

Следующий шаг предлагает выбрать алгоритм обучения и его параметры. Здесь тоже ничего менять не нужно.

Алгоритм	Параметры
<input type="radio"/> Back - Propagation Обучение в режиме "онлайн". Коррекция весов производится после предъявления каждого примера обучающего множества.	Шаг спуска <input type="text" value="0,5"/> В случае изменения знака градиентной составляющей ошибки для данного веса задает величину следующей коррекции веса.
<input checked="" type="radio"/> Resilent Propagation (RPROP) Обучение в режиме "оффлайн". Коррекция весов производится после предъявления всех примеров обучающего множества. Учитывается только знак градиента по каждому весу.	Шаг подъема <input type="text" value="1,2"/> В случае сохранения знака градиентной составляющей ошибки для данного веса задает величину следующей коррекции веса.

Следующий шаг предлагает настроить условия остановки обучения. Укажем, что следует считать пример распознанным, если ошибка меньше 0.005, и также укажем условие остановки обучения при достижении эпохи 10000.

Считать пример распознанным, если ошибка меньше	<input type="text" value="0,05"/>
<input checked="" type="checkbox"/> По достижению эпохи	<input type="text" value="10000"/>
Обучающее множество	
<input type="checkbox"/> Средняя ошибка меньше	<input type="text"/>
<input type="checkbox"/> Максимальная ошибка меньше	<input type="text"/>
<input type="checkbox"/> Распознано примеров (%)	<input type="text" value="0"/>
Тестовое множество	
<input type="checkbox"/> Средняя ошибка меньше	<input type="text"/>
<input type="checkbox"/> Максимальная ошибка меньше	<input type="text"/>
<input type="checkbox"/> Распознано примеров (%)	<input type="text" value="0"/>

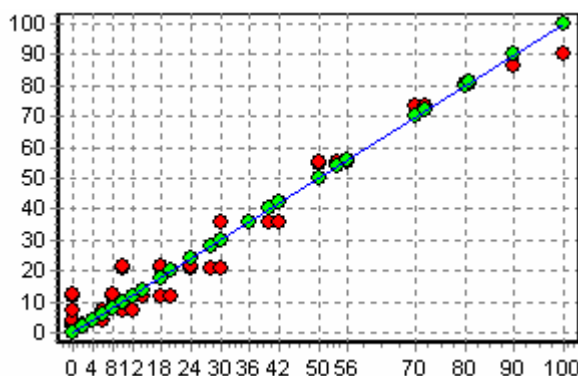
Следующий шаг мастера предлагает запустить процесс обучения и наблюдать в процессе обучения величину ошибки, а также процент распознанных примеров. Параметр «Частота обновления» отвечает за то, через какое количество эпох обучения выводится данная информация.



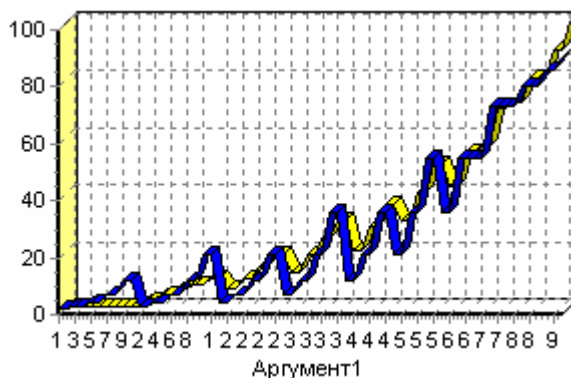
После обучения сети, в качестве визуализаторов выберем Диаграмму, Диаграмму рассеяния, Граф нейросети, Что-если.

<input checked="" type="checkbox"/> Data Mining	
<input checked="" type="checkbox"/> Граф нейросети Отображает нейронную сеть в виде графа	
<input checked="" type="checkbox"/> Что-если	Анализ построенной модели по принципу что-если
<input type="checkbox"/> Обучающий набор	Обучающее, тестовое и валидационное множества
<input checked="" type="checkbox"/> Диаграмма рассе...	Отображает диаграмму отклонения прогнозируемых значения ...
<input checked="" type="checkbox"/> Табличные данные	
<input type="checkbox"/> Таблица	Отображает данные в виде таблицы
<input type="checkbox"/> Статистика	Отображает статистические данные выборки
<input checked="" type="checkbox"/> Диаграмма	Отображает данные в виде диаграммы
<input type="checkbox"/> Гистограмма	Отображает данные в виде гистограммы
<input type="checkbox"/> OLAP анализ	
<input type="checkbox"/> Куб	Многомерное отображение (кросс-таблица и кросс-диаграмма)

Результаты наглядно видны на диаграмме рассеяния, которая показывает рассеяние прогнозируемых данных относительно эталонных.



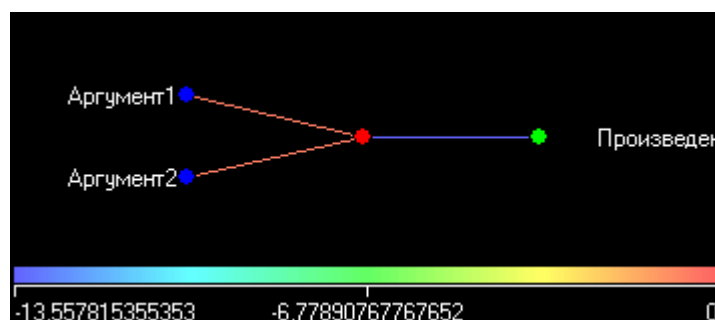
Также можно сравнить эталонные данные с прогнозируемыми, выбрав на обычной диаграмме два поля – «ПРОИЗВЕДЕНИЕ» и «ПРОИЗВЕДЕНИЕ_OUT».



Визуализатор «Что-если» позволит провести эксперимент, введя любые значения множителей АРГУМЕНТ1 и АРГУМЕНТ2 и рассчитав результат их произведения.

Граф нейросети		Что-если		Диаграмма рассеяния		Диаграмма	
1 из 64							
Поле		Значение		Параметр		Значение	
Входные				Минимум		0	
12 Аргумент1		1		Максимум		10	
12 Аргумент2		0		Среднее		5,2656	
Выходные				Стандартное откл.		3,2125	
12 Произведение		2					

Вид построенной сети можно посмотреть, выбрав визуализатор "Граф нейронной сети".



Выводы

Данный пример показал, как можно построить модель прогноза, используя нейронную сеть. Пример показал, что для построения нет необходимости в строгой математической спецификации модели, что особенно ценно при анализе плохо формализуемых процессов. А большинство бизнес задач плохо формализуется. Это означает, что наличие достаточно развитых и удобных инструментальных программных средств позволяет аналитику при построении модели прогнозируемого процесса руководствоваться такими понятиями, как опыт и интуиция.

Настройки мастера позволяют увидеть широкие возможности Deductor Studio касательно структуры сети, способов обучения и т.д. Аналитику предоставляется широкие возможности по настройке нормализации столбцов, разбиения данных на обучающее и тестовое множество, определения структуры сети, количества слоев и нейронов в каждом слое, выборе функции активации и ее параметров, выборе различных алгоритмов обучения и настройки их параметров. Все это позволяет построить модель описывающую практически любые закономерности. Также было показано, как можно спрогнозировать результат, введя любые значения входных факторов, используя визуализатор «Что-если». Понятно, что этап построения модели стоит на завершающих позициях анализа данных и перед тем, как его провести, необходимо должным образом подготовить данные, что позволяет сделать широкий набор инструментов Deductor Studio. Качество подготовки данных для модели, а также качество самой модели аналитик может оценить разными способами: посмотреть диаграмму рассеяния, провести ряд экспериментов при помощи «Что-если», построить гистограмму распределения ошибки и т.п.

Классификация с помощью деревьев решений

Деревья решений применяются для решения задачи классификации. Дерево представляет собой набор условий (правил), согласно которым данные относятся к тому или иному классу. Также после построения присутствует информация о достоверности того или иного правила, его значимость. С помощью данного инструмента можно узнать ранг значимости каждого фактора (наиболее значимые факторы находятся на верхних уровнях дерева).

Исходные данные

Пусть аналитик имеет данные по тому, как голосуют различные депутаты по различным законопроектам. Также известна партийная принадлежность каждого депутата – республиканец или демократ. Перед аналитиком поставлена задача: классифицировать депутатов на демократов и республиканцев в зависимости от того, как они голосуют. Данные по голосованию находятся в файле «Vote.txt». Таблица содержит следующие поля : «КОД» – порядковый номер, «КЛАСС» – класс голосующего (демократ или республиканец),

остальные поля информируют о том, как голосовали депутаты за принятие различных законопроектов («да» , «нет» , «воздержался»).

Импортируем данные из файла и посмотрим их в виде таблицы.

	Проект по преступности	Проект по таможенным пошлинам	Проект по экспорту	Класс
▶	да	нет	да	республиканец
	да	да	да	республиканец
	нет	да	да	демократ
	нет	нет	да	демократ
	да	нет	нет	республиканец

Классификация на демократов и республиканцев

Для решения задачи запустим мастер обработки. Выберем в качестве обработки дерево решений. В мастере построения дерева решения на втором шаге настроим «КОД» как информационный, «КЛАСС» – как выходной, остальные поля – входные. Далее предлагается настроить способ разбиения исходного множества данных на обучающее и тестовое. Зададим случайный способ разбиения, когда данные для тестового и обучающего множества берутся из исходного набора случайным образом. На следующем шаге мастера предлагается настроить параметры процесса обучения, а именно минимальное количество примеров, при котором будет создан новый узел (пусть узел создается, если в него попали два и более примеров), а также предлагается возможность строить дерево с более достоверными правилами, и параметры отсечения узлов. Включим данные опции.

Параметры ранней остановки

Минимальное количество примеров в узле, при котором будет создан новый

☒ Строить дерево с более достоверными правилами в ущерб компактности дерева
 Очередной узел будет разбиваться на подузлы, если количество нераспознанных примеров в узле больше значения параметра "минимальное количество примеров в узле".

Параметры отсечения

☒ Отсекать узлы дерева

Уровень доверия используемый при отсечении узлов дерева, %

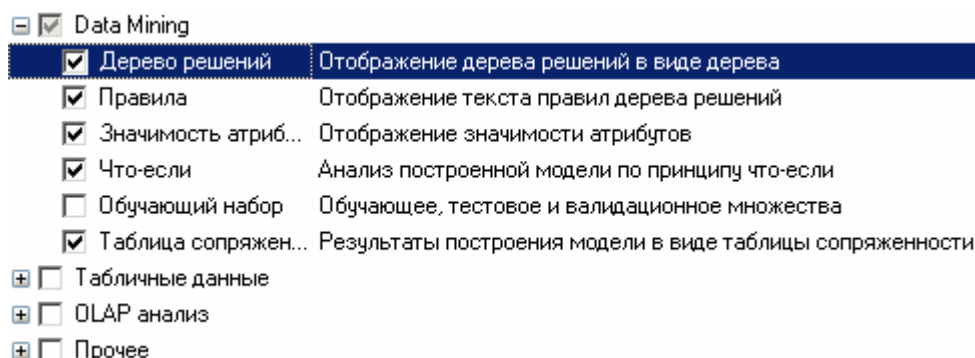
Чем меньше уровень доверия, тем больше узлов будет отсечено и тем компактнее будет дерево.

На следующем шаге мастера запускается сам процесс построения дерева. Также можно увидеть информацию о количестве распознанных примеров.

Распределено, шт.		Распознано, %		Кол-во узлов	
<input checked="" type="checkbox"/> Распознано	<input type="text" value="139"/>	Обучающее мн-во	<input type="text" value="97,89"/>	<input type="text" value="4"/>	
<input checked="" type="checkbox"/> Нераспознано	<input type="text" value="3"/>	Тестовое мн-во	<input type="text" value="100,00"/>	<input type="text" value="3"/>	

Время обучения		<input type="button" value="▶ Пуск"/> <input type="button" value="⏸ Пауза"/> <input type="button" value="⏹ Стоп"/>
<input type="text" value="0:00:00"/>		
Темп обновления <input type="text" value="100"/>		

После построения дерева можно увидеть, что почти все примеры и на обучающей и на тестовой выборке распознаны. Перейдем на следующий шаг мастера для выбора способа визуализации полученных результатов. Основной целью аналитика является отнесение депутата к той или иной партии. Механизм отнесения должен быть таким, чтобы депутат указал, как он будет голосовать за различные законопроекты, а дерево решений ответит на вопрос, кто он – демократ или республиканец. Такой механизм предлагает визуализатор «Что-если». Не менее важным является и просмотр самого дерева решений, на которое можно определить, какие факторы являются более важными (верхние узлы дерева), какие второстепенные, а какие вообще не оказывают влияния (входные факторы, вообще не присутствующие в дереве решений). Поэтому выберем также и визуализатор «Дерево решений». Формализованные правила классификации, выраженные в форме «Если <Условие>. Тогда <Класс>» можно увидеть, выбрав визуализатор «Правила (дерево решений)». Часто аналитику бывает полезно узнать, сколько примеров было распознано неверно, к каким именно примерам были отнесены к какому классу ошибочно. На этот вопрос дает ответ визуализатор «Таблица сопряженности». Очень важно знать, каким образом каждый фактор влияет на классификацию. Такую информацию предоставляет визуализатор «Значимость атрибутов».



Проанализируем данные на полученных визуализаторах. Для начала посмотрим на таблицу сопряженности.

Фактически	Классифицировано		
	демократ	республиканец	Итого
демократ	90	2	92
республиканец	1	57	58
Итого	91	59	150

По диагонали таблицы расположены примеры, которые были правильно распознаны, в остальных ячейках те, которые были отнесены к другому классу. В данном случае дерево правильно классифицировало практически все примеры.

Перейдем к основному визуализатору для данного алгоритма – «Дерево решений»

Как видно, дерево решений получилось не очень громоздкое, большая часть факторов (законопроектов) была отсечена, т.е. влияние их на принадлежность к партии минимальная или его вообще нет (по-видимому, по этим вопросам у партий нет принципиального противостояния).

Узел 3

Класс	№	%
демо...	2	3,45
рес...	56	96,60
Подд...	58	40,80

ЕСЛИ
Закон о врачах = да

Код	Проект по инвалидам	Проект по водным ресурсам	Проект по усы...
1	нет	нет	да
2	да	нет	да
5	нет	да	нет
11	нет	нет	да

Самым значимым фактором оказалась позиция, занимаемая депутатами по пакету законов касающихся врачей. Т.е. если депутат голосует против законопроекта о врачах, то он демократ (об это можно говорить с полной уверенностью, потому что в узел попало 83 примера). Достоверно судить о том, что депутат – республиканец можно, если он голосовал за законопроект о врачах, а также за законопроект по Сальвадору, а также был против законопроекта об усыновлении. Данный

визуализатор предоставляет возможность просмотра примеров, которые попали в тот или иной узел, а также информацию об узле.

Более удобно посмотреть значимость факторов или атрибутов в визуализаторе «Значимость атрибутов».

Целевой атрибут: Класс			
№	Атрибут	▲ Значимость, %	
4	Закон о врачах	92.207	<div><div></div></div>
16	Проект по экспорту	3.498	<div><div></div></div>
5	Проект по Сальвадору	2.455	<div><div></div></div>
3	Проект по усыновлению	1.840	<div><div></div></div>
12	Закон об образовании	0.000	<div><div></div></div>
11	Проект по альтернативным источникам топлива	0.000	<div><div></div></div>
13	Проект по фондам	0.000	<div><div></div></div>
15	Проект по таможенным пошлинам	0.000	<div><div></div></div>
14	Проект по преступности	0.000	<div><div></div></div>
10	Закон об иммигрантах	0.000	<div><div></div></div>
6	Закон о религиях	0.000	<div><div></div></div>
2	Проект по водным ресурсам	0.000	<div><div></div></div>
1	Проект по инвалидам	0.000	<div><div></div></div>
9	Проект по ракетам	0.000	<div><div></div></div>
8	Проект помощи Никарагуа	0.000	<div><div></div></div>
7	Антиспутниковый проект	0.000	<div><div></div></div>

С помощью данного визуализатора можно определить насколько сильно выходное поле зависит от каждого из входных факторов. Чем больше значимость атрибута, тем больший вклад он вносит при классификации. В данном случае самый большой вклад вносит закон о врачах, как и было сказано выше.

На визуализаторе «Правила» представлен список всех правил, согласно которым можно отнести депутата к той или иной партии. Правила можно сортировать по поддержке, достоверности, фильтровать по выходному классу (к примеру, показать только те правила, согласно которым депутат является демократом с сортировкой по поддержке).

№	Условие	Следствие (Класс)	Поддержка		Достоверность	
			%	Кол-во	%	Кол-во
1	<u>Закон о врачах</u> = воздержался	демократ	2,82	4	75,00	3
2	<u>Закон о врачах</u> = да И <u>Проект по Сальвадору</u> = воздержался	республиканец	0,00	0	0,00	0
3	<u>Закон о врачах</u> = да И <u>Проект по Сальвадору</u> = да И <u>Проект по усыновлению</u> = воздержался	республиканец	0,00	0	0,00	0

Данные представлены в виде таблицы. Полями этой таблицы являются:

- номер правила,
- условие, которое однозначно определяет принадлежность к партии,
- решение – то, кем является депутат, голосовавший согласно этому условию,
- поддержка – количество и процент примеров из исходной выборки, которые отвечают этому условию,
- достоверность – процентное отношение количества верно распознанных примеров, отвечающих данному условию к общему количеству примеров, отвечающих данному условию (сумма верно и ошибочно распознанных примеров).

Исходя из данных этой таблицы, аналитик может сказать, что именно влияет на то, что депутат демократ или республиканец, какова цена этого влияния (поддержка) и какова достоверность правила. В данном случае совершенно очевидно, что из всего списка правил с достаточно большим доверием можно отнести к двум – правилу №9 и правилу №7. Таким образом, получается, что демократы принципиально против законопроектов, касающихся врачей. Республиканцы же, наоборот, за принятие этих законопроектов и также за принятие законопроекта по Сальвадору, но категорически против законопроектов по усыновлению.

Теперь аналитик может точно сказать, кто есть кто.

Выводы

Пример показал простоту и удобство применения деревьев решений для классификации на республиканцев и демократов. Мастер предлагает широкие возможности по настройке процесса построения дерева решений. Это и настройка назначения столбцов, настройка их нормализации, настройка источника данных для учителя (тестовое и обучающее множества), настройка количества примеров в узле и настройка достоверности правил. После построения дерева стали видны его достоинства для анализа. Алгоритм сам отсекает несущественные факторы, выявил степень влияния тех или иных факторов на результат, описал при помощи формальных правил способ классификации, а также выдал информацию о достоверности и поддержке того или иного правила. Также были продемонстрированы широкие возможности визуализации построенного дерева. Все это говорит о незаменимости дерева решений для классификации.

Прогнозирование с помощью линейной регрессии.

Линейная регрессия необходима тогда, когда предполагается, что зависимость между входными факторами и результатом линейная. В основном ее применяют для прогнозирования временного ряда. Достоинством ее можно назвать быстроту обработки входных данных.

Исходные данные

Покажем нахождение линейных зависимостей на примере нахождения зависимости между двумя аргументами и их суммой.

Данные для решения задачи находятся в файле «Sum.txt». Он содержит таблицу с полями: «АРГУМЕНТ1», «АРГУМЕНТ2» – слагаемые, «СУММА» – их сумма.

Импортируем данные в Deductor Studio и посмотрим их в виде таблицы.

	Аргумент1	Аргумент2	Сумма
▶	1	1	2
	2	1	3
	3	1	4
	4	1	5
	5	1	6
	6	1	7

Прогнозирование суммы

Для линейного регрессионного анализа необходимо запустить мастер обработки и выбрать в качестве обработки данных линейную регрессию. На втором шаге мастера настроим поля исходных данных. Очевидно, что факторами будут являться аргументы, а результатом – сумма. Поэтому необходимо указать назначение поля «СУММА» как выходное, а назначение остальных полей – как входные.

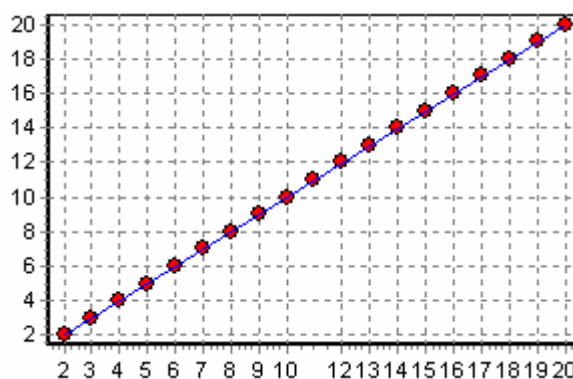
Аргумент1	Имя столбца	COL3
Аргумент2	Тип данных	Вещественный
Сумма	Назначение	Выходное
	Вид данных	Непрерывный

На следующем шаге необходимо настроить способ разделения исходного множества данных на тестовое и обучающее, а также количество примеров в том и другом множестве. Укажем, что данные из обоих множеств берутся случайным образом, а остальные параметры оставим без изменения.

Следующий шаг мастера позволяет выполнить обработку, нажав на кнопку «Пуск». Во время обучения отображаются текущая величина ошибки и процент распознанных примеров.

Обучающее множество		Тестовое множество	
<input checked="" type="checkbox"/> Максимальная ошибка	3,46E-01	<input checked="" type="checkbox"/> Максимальная ошибка	3.46E-01
<input checked="" type="checkbox"/> Средняя ошибка	1,22E-01	<input checked="" type="checkbox"/> Средняя ошибка	3.46E-01
Распознано (%)	31,58	Распознано (%)	00.00
Название текущего процесса			
Обучение завершено			
Процент выполнения текущего процесса обучения			
100%			
<input checked="" type="checkbox"/> Инициализировать перед обучением			
Время обучения	0:00:00	Пуск	Пауза
		Стоп	

После построения модели, можно, воспользовавшись визуализатором «Диаграмма рассеяния» для просмотра качества построенной модели.



На диаграмме рассеяния, что данную зависимость линейная регрессия распознала с большой точностью.

Выводы

Данный пример показал целесообразность применения линейного регрессионного анализа для прогнозирования линейных зависимостей. Простота настроек и быстрота построения модели иногда бывают необходимы. Аналитiku достаточно указать входные столбцы - факторы, выходные - результат, указать способ разбиения данных на тестовое и обучающее множество и запустить процесс обучения. Причем после этого будут доступны все механизмы визуализации и анализа данных, позволяющие построить прогноз, провести эксперимент по принципу «Что-если», исследовать зависимость результата от значений входных факторов, оценить качество построенной модели по диаграмме рассеяния. Также по результатам работы этого алгоритма можно подтвердить или опровергнуть гипотезу о наличии линейной зависимости.

Кластеризация с помощью самоорганизующейся карты Кохонена

Самоорганизующаяся карта Кохонена является разновидностью нейронной сети. Она применяется, когда необходимо решить задачу кластеризации, т.е. распределить данные по нескольким кластерам. Алгоритм определяет расположение кластеров в многомерном пространстве факторов. Исходные данные будут относиться к какому-либо кластеру в зависимости от расстояния до него. Многомерное пространство трудно для представления в графическом виде. Механизм же построения карты Кохонена позволяет отобразить многомерное пространство в двумерном, которое более удобно и для визуализации и для интерпретации результатов аналитиком.

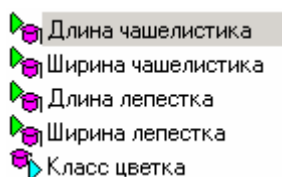
Также с помощью построенной карты Кохонена можно решить и задачу прогнозирования. В этом случае результирующее поле (то, которое необходимо спрогнозировать) в построении карты не участвует. После кластеризации используя диаграмму «Что-если» можно провести эксперимент. Алгоритм определяет точку пространства, где расположены введенные для прогноза данные, затем определяет, к какому кластеру принадлежит данная точка и подсчитывает среднее по результирующему полю всех точек этого кластера, что и будет результатом прогноза (для дискретных данных результатом прогноза является значение, больше всего встречающееся в результирующем поле всех ячеек кластера).

Исходные данные

Рассмотрим механизм кластеризации путем построения самоорганизующейся карты, основываясь на типичных характеристиках цветков. Исходная таблица находится в файле «Iris.txt». Она содержит следующие параметры цветов: «ДЛИНА ЧАШЕЛИСТИКА», «ШИРИНА ЧАШЕЛИСТИКА», «ДЛИНА ЛЕПЕСТКА», «ШИРИНА ЛЕПЕСТКА», «КЛАСС ЦВЕТКА». Задача состоит в том, чтобы определить по различным параметрам цветка его класс. Предполагается, что цветы одного класса имеют схожие параметры, поэтому они должны находиться в одном кластере.

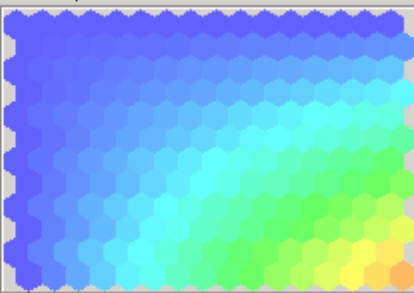
Кластеризация ирисов

Для начала необходимо импортировать данные из файла. После этого запустим, мастер обработки и выберем из списка метод обработки «Карта Кохонена». На втором шаге мастера настроим назначения столбцов. Укажем столбцу «КЛАСС ЦВЕТКА» назначение «Выходной», а остальным – «Входной». Т.е. на основе данных о цветке будем относить его к тому или иному классу.



На третьем шаге мастера необходимо настроить способ разделения исходного множества данных на тестовое и обучающее, а также количество примеров в том и другом множестве. Укажем, что данные обоих множеств берутся случайным образом, зададим размер тестового множества равным десяти примерам, путем изменения значения столбца «Размер в строках» строки «Тестовое множество».

Следующий шаг предлагает настроить параметры карты (количество ячеек по X и по Y, их форму) и параметры обучения (способ начальной инициализации, тип функции соседства, перемешивать ли строки обучающего множества и количество эпох, через которые необходимо перемешивание). Значения по умолчанию вполне подходят.

Параметры карты		Вид карты
Размер по оси X	16	
Размер по оси Y	12	
Кол-во ячеек	192	
Форма ячеек	Шестиугольные	

На пятом шаге мастера необходимо настроить параметры остановки обучения. Оставим параметры по умолчанию.

Считать пример распознанным, если ошибка меньше	0,05
<input checked="" type="checkbox"/> По достижению эпохи	130
Обучающее множество	
<input type="checkbox"/> Средняя ошибка меньше	
<input type="checkbox"/> Максимальная ошибка меньше	
<input type="checkbox"/> Распознано примеров (%)	0
Тестовое множество	
<input type="checkbox"/> Средняя ошибка меньше	
<input type="checkbox"/> Максимальная ошибка меньше	
<input type="checkbox"/> Распознано примеров (%)	0

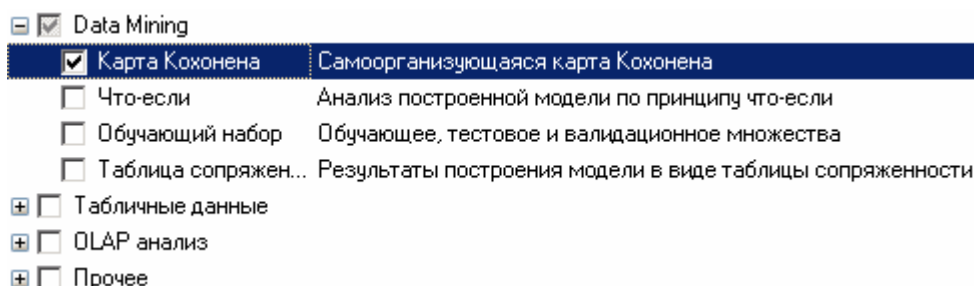
На шестом шаге настраиваются остальные параметры обучения – способ начальной инициализации, тип функции соседства и также параметры кластеризации – автоматическое определение числа кластеров с соответствующим уровнем значимости либо фиксированное количество кластеров предоставляется возможность настроить интервалы обучения. Каждый интервал задается количеством эпох, радиусом обучения и скоростью обучения. Укажем фиксированное количество кластеров, равное трем.

Способ начальной инициализации карты		Из собственных векторов
<input checked="" type="checkbox"/> Количество эпох, через которое необходимо перемешивать строки		20
Скорость обучения		
В начале обучения	0,3	
В конце обучения	0,005	
Радиус обучения		
В начале обучения	4	
В конце обучения	0,1	
Функция соседства		Ступенчатая
Кластеризация		
<input type="checkbox"/> Автоматически определить количество кластеров		
Уровень значимости, %	1	Фиксированное кол-во кластеров
		3

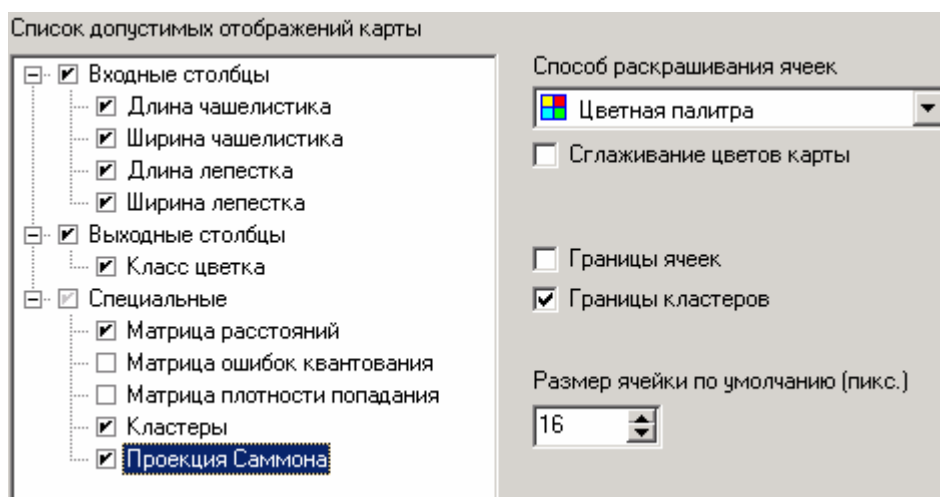
На седьмом шаге предлагается запустить сам процесс обучения. Во время обучения можно посмотреть количество распознанных примеров и текущие значения ошибок. Здесь необходимо нажать на кнопку пуск и дождаться завершения процесса обработки.



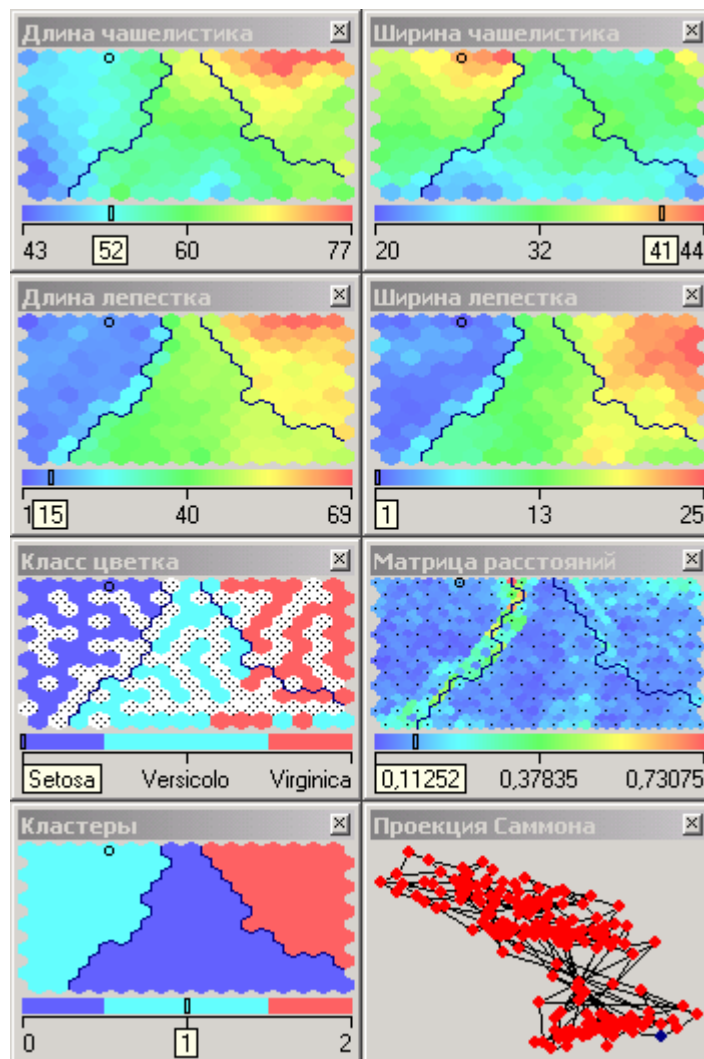
После этого необходимо в списке визуализаторов выбрать появившуюся теперь «Карту Кохонена» для просмотра результатов кластеризации, а также визуализатор «Что-если» для прогнозирования класса цветка.



Далее, в мастере настройки отображения карты Кохонена необходимо указать, чтобы отображались все поля, также следует установить количество кластеров равным трем и поставить флажок «Границы кластеров».



После этого можно увидеть полученные результаты.



Качество кластеризации можно оценить, просмотрев карту «КЛАСС ЦВЕТКА». На ней видно, что большинство цветов были классифицированы правильно. Заметим, что все цветы класса Setosa попали в один кластер. Это говорит о значительном отличии параметров цветов этого класса от других. Явное различие наблюдается по длине и ширине лепестка. То, что часть примеров Virginica попала в класс Versicolour и наоборот говорит о меньшем различии этих классов. На картах, в отличие от Setosa не видны резкие отличия параметров цветов этих двух классов. Этим как раз и объясняется «проникновение» некоторой части примеров в другой кластер.

Выводы

Данный пример показал область применения самоорганизующихся карт. Изначально имелось многомерное (четырёхмерное) пространство входных факторов. Алгоритм представил его в двумерном виде, которое удобнее анализировать. Также исходные данные были отнесены к трем кластерам, по типу цветка – «Setosa», «Versicolour», «Virginica». Основным визуализатором после построения является «Самоорганизующаяся карта». Здесь в первую очередь следует обратить внимание на матрицу расстояний и проекцию Саммона. На них явно видны расстояния между отдельными ячейками карты, т.е. четкие границы различных скоплений данных.

Мастер предоставляет широкий набор настройки параметров обучения: настройка нормализации столбцов, настройка разбиения на тестовое и обучающее множество, настройка условий остановки обучения, настройка параметров карты и параметров обучения, настройка интервалов обучения.

Поиск ассоциативных правил.

Ассоциативные правила позволяют находить закономерности между связанными событиями. Примером такого правила, служит утверждение, что покупатель, приобретающий «Хлеб», приобретет и «Молоко». Впервые эта задача была предложена для поиска ассоциативных правил для нахождения типичных шаблонов покупок, совершаемых в супермаркетах, поэтому иногда ее еще называют анализом рыночной корзины (market basket analysis). Пусть имеется база данных, состоящая из покупательских транзакций. Каждая транзакция – это набор товаров, купленных покупателем за один визит. Такую транзакцию еще называют рыночной корзиной. Целью анализа является установление следующих зависимостей: если в транзакции встретился некоторый набор элементов X , то на основании этого можно сделать вывод о том, что другой набор элементов Y также же должен появиться в этой транзакции. Установление таких зависимостей дает нам возможность находить очень простые и интуитивно понятные правила.

Основными характеристиками таких правил являются поддержка и достоверность. Правило «Из X следует Y » имеет поддержку s , если $s\%$ транзакций из всего набора, содержат наборы элементов X и Y . Достоверность правила показывает, какова вероятность того, что из X следует Y . Правило «Из X следует Y » справедливо с достоверностью c , если $c\%$ транзакций из всего множества, содержащих набор элементов X , также содержат набор элементов Y . Покажем на конкретном примере: пусть 75% транзакций, содержащих хлеб, также содержат молоко, а 3% от общего числа всех транзакций содержат оба товара. 75% – это достоверность правила, а 3% – это поддержка.

Алгоритмы поиска ассоциативных правил предназначены для нахождения всех правил вида «из X следует Y », причем поддержка и достоверность этих правил должны находиться в рамках некоторых наперед заданных границ, называемых соответственно минимальной и максимальной поддержкой и минимальной и максимальной достоверностью.

Границы значений параметров поддержки и достоверности выбираются таким образом, чтобы ограничить количество найденных правил. Если поддержка имеет большое значение, то алгоритмы будут находить правила, хорошо известные аналитикам или настолько очевидные, что нет никакого смысла проводить такой анализ. С другой стороны, низкое значение поддержки ведет к генерации огромного количества правил, что, конечно, требует существенных вычислительных ресурсов. Тем не менее, большинство интересных правил находится именно при низком значении порога поддержки. Хотя слишком низкое значение поддержки ведет к генерации статистически необоснованных правил. Таким образом, необходимо найти компромисс, обеспечивающий, во первых, интересность правил и, во вторых, их статистическую обоснованность. Поэтому значения этих границ напрямую зависят от характера анализируемых данных и подбираются индивидуально. Еще одним параметром, ограничивающим количество найденных правил является *максимальная мощность* часто встречающихся множеств. Если этот параметр указан, то при поиске правил будут рассматриваться только множества, количество элементов которых будет не больше данного параметра. И, следовательно, любое найденное правило будет состоять не больше, чем из *максимальная_мощность* элементов.

Исходные данные

Рассмотрим механизм поиска ассоциативных правил на примере данных о продажах товаров в некоторой торговой точке. Данные находятся в файле «Supermarket.txt». В таблице представлена информация по покупкам продуктов нескольких групп. Она имеет всего два поля «НОМЕР ЧЕКА» и «ТОВАР». Необходимо решить задачу анализа потребительской корзины с целью последующего применения результатов для стимулирования продаж.

Импортируем данные из текстового файла и посмотрим в виде таблицы:

Номер чека	Товар
160698	КЕТЧУПЫ, СОУСЫ, АДЖИКА
160698	МАКАРОННЫЕ ИЗДЕЛИЯ
160698	ЧАЙ
160747	МАКАРОННЫЕ ИЗДЕЛИЯ
160747	МЕД
160747	ЧАЙ
161217	КЕТЧУПЫ, СОУСЫ, АДЖИКА

Поиск ассоциативных правил

Для поиска ассоциативных правил запустим мастер обработки. В нем выберем тип обработки «Ассоциативные правила». На втором шаге мастера необходимо указать, какой столбец является идентификатором транзакции (чек), а какой элементом транзакции (товар).

Следующий шаг позволяет настроить параметры построения ассоциативных правил: минимальную и максимальную поддержку, минимальную и максимальную достоверность, а также максимальную мощность множества. Исходя из характера имеющихся данных, следует указать границы поддержки – 13% и 80%, и достоверности 60% и 90%.

Следующий шаг позволяет запустить процесс поиска ассоциативных правил. На экране отображается информация о количестве множеств, количестве найденных правил, а также гистограмма распределения найденных часто встречающихся множеств по мощности.



После завершения процесса поиска полученные результаты можно посмотреть, используя появившиеся специальные визуализаторы «Популярные наборы», «Правила», «Дерево правил», «Что-если».

Популярные наборы - это множества, состоящие из одного и более элементов, которые наиболее часто встречаются в транзакциях одновременно. На сколько часто встречается множество в исходном наборе транзакций можно судить по поддержке. Данный визуализатор отображает множества в виде списка.

№	Множество	↑ Поддержка	
		%	Кол-во
7	ЧАЙ	75,00	33
3	МАКАРОННЫЕ ИЗДЕЛИЯ	54,55	24
2	КЕТЧУПЫ, СОУСЫ, АДЖИКА	52,27	23
4	МЕД	50,00	22

Само название визуализатора говорит о том, как применить данные результаты на практике. Получившиеся наборы товаров наиболее часто покупают в данной торговой точке, следовательно можно принимать решения о поставках товаров, их размещении и т.д.

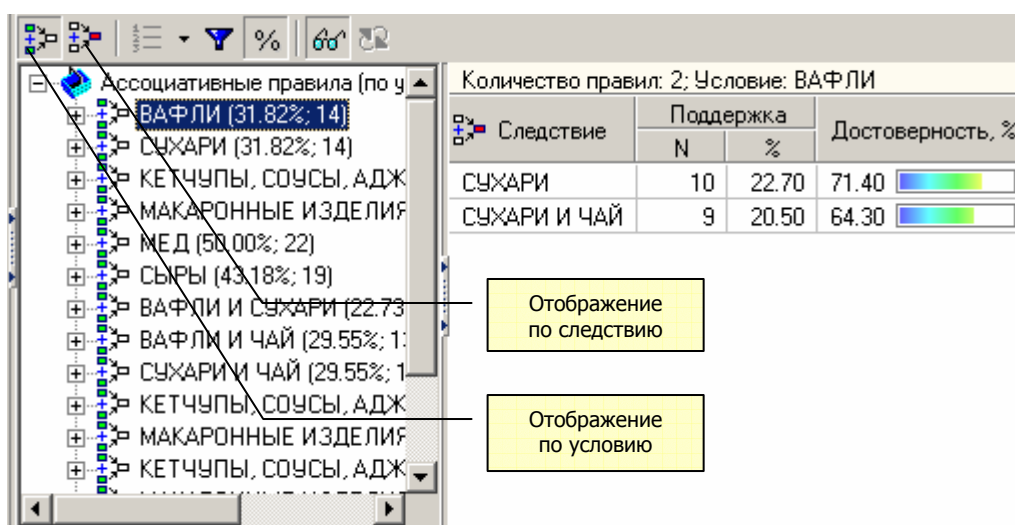
Визуализатор «Правила» отображает ассоциативные правила в виде списка правил. Этот список представлен таблицей со столбцами: «номер правила», «условие», «следствие», «поддержка, %», «поддержка, количество», «достоверность».

№	Условие	Следствие	Поддержка		Достоверность, %
			%	Кол-во	
1	ВАФЛИ	СУХАРИ	22,73	10	71,43
2	СУХАРИ	ВАФЛИ	22,73	10	71,43
3	КЕТЧУПЫ, СОУСЫ, АДЖИКА	МАКАРОННЫЕ ИЗДЕЛИЯ	45,45	20	86,96
4	МАКАРОННЫЕ ИЗДЕЛИЯ	КЕТЧУПЫ, СОУСЫ, АДЖИКА	45,45	20	83,33
5	МЕД	ЧАЙ	40,91	18	81,82

Таким образом, эксперту предоставляется набор правил, которые описывают поведение покупателей. Например, если покупатель купил вафли, то он с вероятностью 71% также купит и сухари.

Визуализатор «Дерево правил» - это всегда двухуровневое дерево. Оно может быть построено либо по условию, либо по следствию. При построении дерева правил по условию, на первом (верхнем) уровне находятся узлы с условиями, а на втором уровне - узлы со следствием. Второй вариант дерева правил - дерево, построенное по следствию. Здесь на первом уровне располагаются узлы со следствием.

Справа от дерева находится список правил, построенный по выбранному узлу дерева. Для каждого правила отображаются поддержка и достоверность. Если дерево построено по условию, то вверху списка отображается условие правила, а список состоит из его следствий. Тогда правила отвечают на вопрос, что будет при таком условии. Если же дерево построено по следствию, то вверху списка отображается следствие правила, а список состоит из его условий. Эти правила отвечают на вопрос, что нужно, чтобы было заданное следствие. Данный визуализатор отображает те же самые правила, что и предыдущий, но в более удобной для анализа форме.



В данном случае правила отображены по условию. Тогда отображаемый в данный момент результат можно интерпретировать как 2 правила:

1. Если покупатель приобрел вафли, то он с вероятностью 71% также приобретет сухари.
2. Если покупатель приобрел вафли, то он с вероятностью 64% также приобретет, сухари и чай.

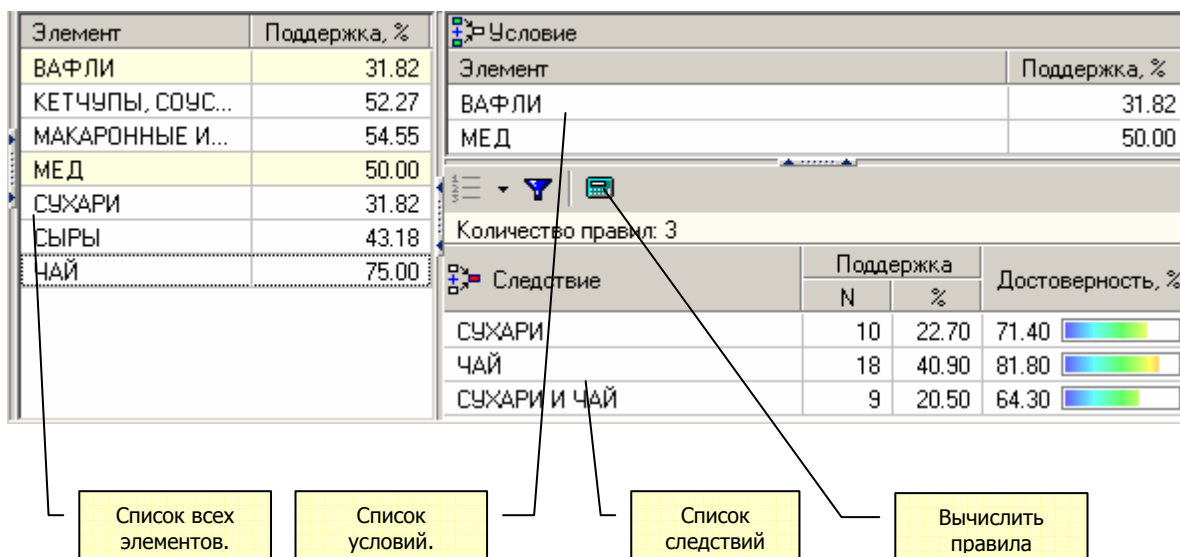
Аналогично интерпретируются и остальные правила.

Анализ «Что-если» в ассоциативных правилах позволяет ответить на вопрос что получим в качестве следствия, если выберем данные условия? Например, какие товары приобретаются совместно с выбранными товарами. В окне слева расположен список всех элементов транзакций. Справа от каждого элемента указана поддержка - сколько раз данный элемент встречается в транзакциях.

В правом верхнем углу расположен список элементов, входящих в условие. Это, например, список товаров, которые приобрел покупатель. Для них нужно найти следствие. Например, товары, приобретаемые совместно с ними. Чтобы предложить человеку то, что он возможно забыл купить.

В правом нижнем углу расположен список следствий. Справа от элементов списка отображается поддержка и достоверность.

Пусть необходимо проанализировать, что, возможно, забыл покупатель приобрести, если он уже взял вафли и мед? Для этого необходимо добавить в список условий эти товары (например, с помощью двойного щелчка мыши) и затем нажать на кнопку «Вычислить правила». При этом в списке следствий появятся товары, совместно приобретаемые с данными. В данном случае появятся «СУХАРИ», «ЧАЙ», «СУХАРИ И ЧАЙ». Т.е. возможно, покупатель забыл приобрести сухари или чай или и то и другое.



Выводы

Как было сказано, задача поиска ассоциативных правил впервые была представлена для анализа рыночной корзины. Как показал данный пример, результаты анализа можно применить и для сегментации покупателей по поведению при покупках, и для анализа предпочтений клиентов, и для планирования расположения товаров в супермаркетах, кросс-маркетинге. Предлагаемый набор визуализаторов позволяет эксперту найти интересные, необычные закономерности, понять, почему так происходит и применить их на практике.

В данном примере найденные правила можно использовать для сегментации клиентов на два сегмента: клиенты, покупающие макаронные изделия и соусы к ним и клиенты, покупающие все к чаю. В разрезе анализа предпочтений можно узнать, что наибольшей популярностью в данном магазине пользуются чай, мед, макаронные изделия, кетчупы, соусы и аджика. В разрезе размещения товаров в супермаркете можно применить результаты предыдущих двух анализов – располагать чай рядом с медом, а кетчупы, соусы и аджику рядом с макаронными изделиями и т.д.

Прогнозирование с помощью построения пользовательских моделей.

Пользовательская модель позволяет создавать аналитические модели на основании формул и экспертных оценок. Такая возможность требуется в тех случаях, когда объем исходной выборки мал, либо ее качество недостаточно для того, чтобы обучить нейронную сеть. В этом случае можно воспользоваться хорошо известными простыми моделями, задающимися с помощью формул. Примером такой модели может служить скользящее среднее или модель авторегрессии.

Исходные данные

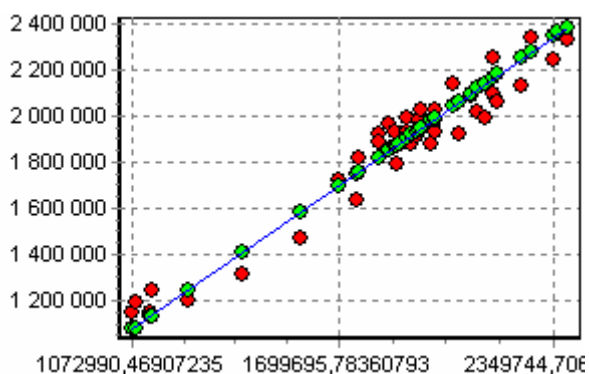
Рассмотрим применение пользовательской модели на примере данных по продажам, находящихся в файле «Trade.txt». Будем строить пользовательские модели на ветке «Данные по продажам товаров» сразу после обработчика «Скользящее окно». Рассмотрим две пользовательские модели. Пусть аналитику известен характер продаж определенных товаров. Так, например, известно, что каждый месяц наблюдается постоянный прирост объема продаж, равный 160000, также известно, что наблюдается спад продаж, равный 12% от аналогичного периода прошлого года, а также прирост в 2% по сравнению с текущим месяцем. Таким образом, аналитик может рассчитать прогноз по формуле:

$$\text{Прогноз} = 160000 - \text{ОбъемМесяцаГодНазад} * 0.12 + \text{ОбъемТекущегоМесяца} * 1.02.$$

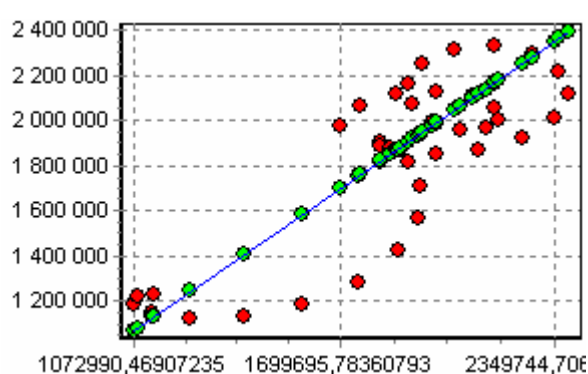
Также аналитик может воспользоваться реализованной моделью скользящего среднего, которая подразумевает, что объем продаж следующего месяца равен среднему объему продаж некоторого количества предшествующих месяцев. Рассмотрим их по очереди.

Прогнозирование с применением пользовательских моделей

Для построения пользовательской модели необходимо запустить мастер обработки и выбрать в качестве обработки данных пользовательскую модель. На втором шаге мастера настроим поля исходных данных. Для первой модели необходимо выбрать в качестве входных полей «Количество - 12» и «Количество - 1», а выходным будет поле «Количество». При построении второй пользовательской модели необходимо на данном этапе в качестве входов указать поля «Количество - 5» ... «Количество - 1». На следующем шаге мастера необходимо написать формулу получения прогноза. Интерфейс данного шага мастера практически не отличается от настройки обработчика «Калькулятор», рассмотренного выше. Так, в поле ввода выражения необходимо написать правую часть формулы, известную аналитику, а именно «160000 - 0.12 * COL2B12 + 1.02 * COL2B1» (COL2B12 и COL2B1 – соответственно имена полей «Количество - 12» и «Количество - 1»). При построении второй пользовательской модели выражение будет следующее: «MOVINGAVERAGE(COL2B1;COL2B2;COL2B3;COL2B4;COL2B5)» (здесь используется встроенная функция расчета среднего значения, в данном случае среднего объема продаж за пять предыдущих месяцев). Далее необходимо перейти на следующий шаг и выбрать способ визуализации. Вот как, например, выглядят диаграммы рассеяния обеих пользовательских моделей:

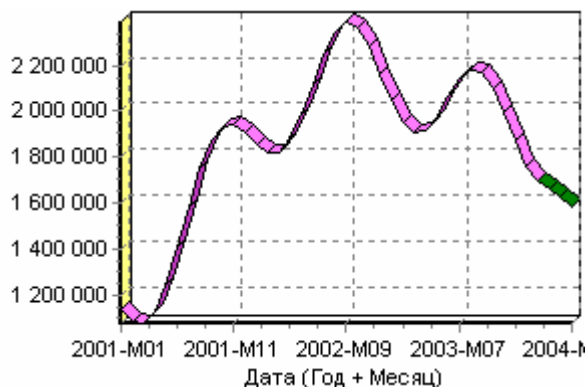


а) Модель, полученная по формуле

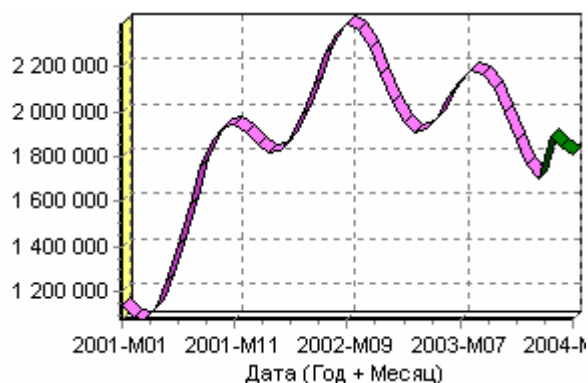


б) Модель скользящего среднего

Далее, также как и в примере построения прогноза объема продаж, после обеих пользовательских моделей построим прогноз на 3 месяца вперед. Вот как выглядят их диаграммы прогноза:



а) Модель, полученная по формуле



б) Модель скользящего среднего

Выводы

Данный пример показал целесообразность применения пользовательских моделей для прогнозирования простых или до определенной степени известных зависимостей. Простота настроек и быстрота построения модели иногда бывают необходимы. Причем после этого будут доступны все механизмы визуализации и анализа данных, позволяющие построить прогноз, провести эксперимент по принципу «Что-если», исследовать зависимость результата от значений входных факторов, оценить качество построенной модели по таблице сопряженности или диаграмме рассеяния и возможно скорректировать расчетную формулу для более точного отражения зависимости.

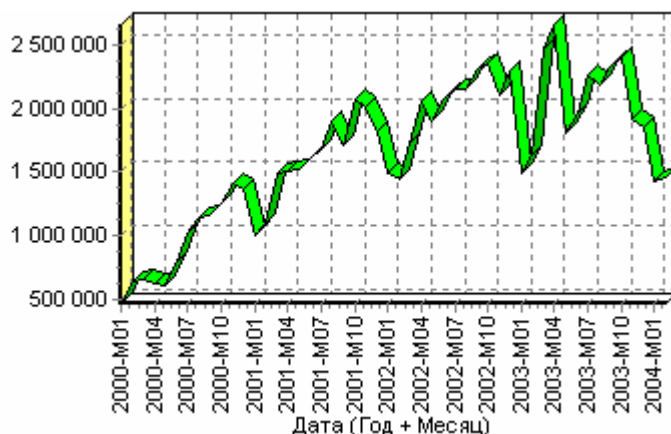
Пример расчета автокорреляции столбцов

Важным фактором для анализа временного ряда и прогноза является определение сезонности. В Deductor Studio таким инструментом является автокорреляция. Вообще корреляция подразумевает под собой зависимость значения одной величины от значения другой. Если их корреляция равна единице, то величины прямо зависимы друг от друга, если нулю – то нет, если минус единица, то зависимость обратная. Линейная автокорреляция ищет зависимости между значениями одной и той же величины, но в разное время. Поэтому нахождение линейной автокорреляционной зависимости и применяется для определения периодичности (сезонности) при обработке временных рядов.

Исходные данные

Пусть аналитик располагает данными по месячному количеству продаж за определенный период времени. Ему необходимо определить есть ли сезонность, и если есть, то какая. Данные по продажам находятся в файле «Trade.txt». Таблица содержит следующие столбцы: «ПЕРИОД» – год и месяц продаж, «КОЛИЧЕСТВО» – количество продаж за этот месяц.

Импортируем данные из текстового файла. Обратите внимание на то, что в файле данные о количестве находятся не в стандартном формате: разделитель дробной и целой части числа не запятая, а точка, поэтому необходимо внести соответствующие изменения в настройки по умолчанию параметров импорта. Выберем в качестве визуализатора Диаграмму для просмотра исходной информации.



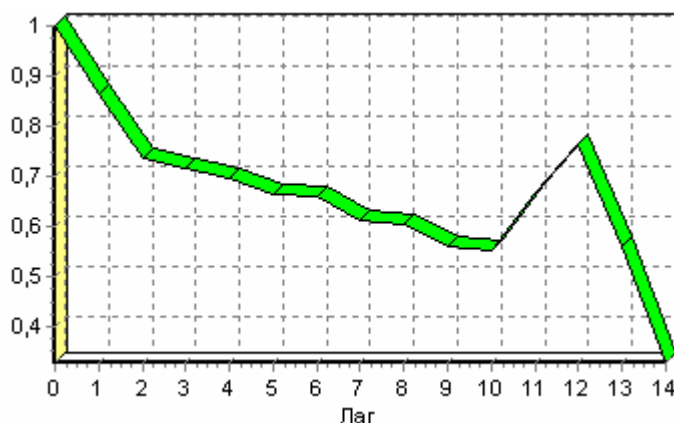
Автокорреляция столбца Количество

Как видно, не каждый аналитик сможет судить о сезонности по этим данным, поэтому необходимо воспользоваться автокорреляцией. Для этого откроем мастер обработки, выберем в качестве обработки автокорреляцию и перейдем на второй шаг мастера. В нем необходимо настроить параметры столбцов. Укажем поле «Дата (Год + Месяц)» неиспользуемым, а поле «КОЛИЧЕСТВО» используемым (ведь необходимо определить сезонность количества продаж). Предположим, что сезонность, если она имеет место, не больше года. В связи с этим зададим количество отсчетов равным 15 (тогда будет итаться зависимость от месяца назад, двух, ..., пятнадцати месяцев назад). Также должен стоять флажок «Включить поле отсчетов набор данных». Он необходим для более удобной интерпретации автокорреляционного анализа.

<div> <div>❌</div> <div>Дата (Год + Месяц)</div> </div> <div> <div>✅</div> <div>Количество</div> </div>	Имя столбца: COL1 Тип данных: Строковый Назначение: ❌ Непригодное Количество отсчетов: <input type="text"/> <input checked="" type="checkbox"/> Включить поле отсчетов в набор данных
---	--

Перейдем на следующий шаг мастера и запустим процесс обработки.

По окончании, результаты удобно анализировать как в виде таблицы, так и в виде диаграммы. После обработки были получены два столбца – «Лег» (благодаря установленному флажку в мастере) и «КОЛИЧЕСТВО» - результат автокорреляции.



Видно, что вначале корреляция равна единице – то как значение зависит само от себя. Далее зависимость убывает и затем виден пик зависимости от данных 12 месяцев назад. Это как раз и говорит о наличии годовой сезонности.

Пример прогноза временного ряда

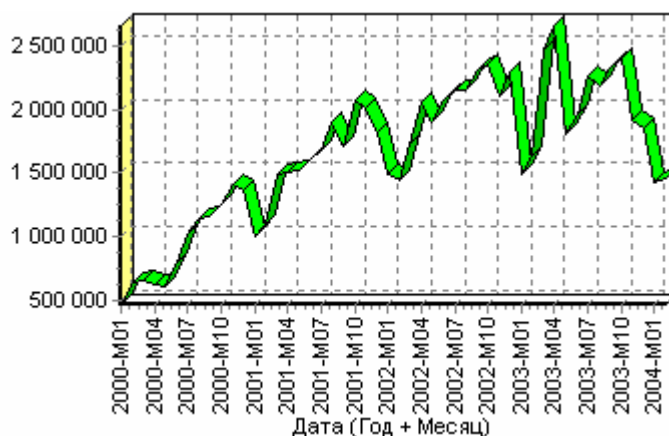
Прогнозирование результата на определенное время вперед, основываясь на данных за прошедшее время – задача, встречающаяся довольно часто (к примеру, перед большинством торговых фирм стоит задача оптимизации складских запасов, для решения которой требуется знать, чего и сколько должно быть продано через неделю, и т.п.; задача предсказания стоимости акций какого-нибудь предприятия через день и т.д. и другие подобные вопросы). Deductor Studio предлагает для этого инструмент «Прогнозирование».

Прогнозирование появляется в списке мастера обработки только после построения какой-либо модели прогноза: нейросети, линейной регрессии и т.д. Прогнозировать на несколько шагов вперед имеет смысл только временной ряд (к примеру, если есть данные по недельным суммам продаж за определенный период, можно спрогнозировать сумму продаж на две недели вперед). Поскольку при построении модели прогноза необходимо учитывать много факторов (зависимость результата от данных день, два, три, четыре назад), то методика имеет свои особенности. Покажем ее на примере.

Исходные данные

У аналитика имеются данные о месячном количестве проданного товара за несколько лет. Ему необходимо, основываясь на этих данных, сказать, какое количество товара будет продано через неделю и через две.

Исходные данные по продажам находятся в файле «Trade.txt», известном по предыдущему примеру (расчет автокорреляции). Выполним импорт данных из файла, не забыв указать в мастере, чтобы в качестве разделителя дробной целой частей была точка, а не запятая.



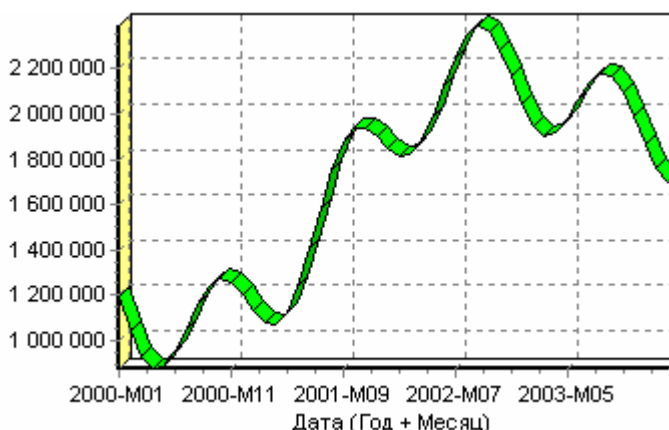
Удаление аномалий и сглаживание

После импорта данных воспользуемся диаграммой для их просмотра.

На ней видно, что данные содержат аномалии (выбросы) и шумы, за которыми трудно разглядеть тенденцию. Поэтому перед прогнозированием необходимо удалить аномалии и сгладить данные.

Сделать это можно при помощи парциальной обработки. Запустим мастер обработки, выберем в качестве обработки данных парциальную обработку и перейдем на следующий шаг мастера. Как известно, второй шаг мастера отвечает за обработку пропущенных значений, которых в исходных данных нет. Поэтому здесь ничего не настраиваем. Следующий шаг отвечает за удаление аномалий из исходного набора. Выберем поле для обработки «КОЛИЧЕСТВО» и укажем для него обработку аномальных явлений (степень подавления – малая).

Четвертый шаг мастера позволяет провести спектральную обработку. Из исходных данных необходимо исключить шумы, поэтому выбираем столбец «КОЛИЧЕСТВО» и указываем способ обработки «вычитание шума» (степень вычитания – малая). На следующем шаге запустим обработку, нажав на «пуск». После обработки посмотрим полученный результат на диаграмме.



Видно, что данные сгладились, аномалии и шумы исчезли. Также видна тенденция.

Теперь перед аналитиком встает вопрос, а как, собственно, прогнозировать временной ряд. Во всех предыдущих примерах мы сталкивались с ситуацией, когда есть входные столбцы - факторы и есть выходные столбцы – результат. В данном случае столбец один. Строить прогноз на будущее

необходимо, основываясь на данных прошлых периодов. Т.е. предполагается, что количество продаж на следующий месяц зависит от количества продаж за предыдущие месяцы. Т.е. входными факторами для модели могут быть продажи за текущий месяц, продажи за месяц ранее и т.д., а результатом должны быть продажи за следующий месяц. Т.е. здесь явно необходимо трансформировать данные к скользящему окну.

Скользящее окно 12 месяцев назад

Запустим мастер обработки, выберем в качестве обработчика скользящее окно и перейдем на следующий шаг.

Аналитик провел также авторегрессионный анализ и выяснил наличие годовой сезонности (см. пример с авторегрессией). В связи с этим было решено строить прогноз на неделю вперед, основываясь на данных за 12, 11 месяцев назад, два месяца назад и месяц назад. Поэтому необходимо, назначив поле «КОЛИЧЕСТВО» используемым, выбрать глубину погружения 12. Тогда данные трансформируются к скользящему окну так, что аналитику будут доступны все требуемые факторы для построения прогноза.

Просмотреть полученные данные можно в виде таблицы.

	Дата (Год -	Количество-12	Количество-11	Количество-10	Количество-9
►	2001-М01	1195750,32836624	1046730,3444785	932230,825412825	875457,294625339
	2001-М02	1046730,3444785	932230,825412825	875457,294625339	884830,92710038
	2001-М03	932230,825412825	875457,294625339	884830,92710038	951789,091701106
	2001-М04	875457,294625339	884830,92710038	951789,091701106	1053383,007105
	2001-М05	884830,92710038	951789,091701106	1053383,007105	1158930,16723578
	2001-М06	951789,091701106	1053383,007105	1158930,16723578	1238826,25455098
	2001-М07	1053383,007105	1158930,16723578	1238826,25455098	1273007,6581252

Как видно, теперь в качестве входных факторов можно использовать «КОЛИЧЕСТВО - 12», «КОЛИЧЕСТВО - 11» - данные по количеству 12 и 11 месяцев назад (относительно прогнозируемого месяца) и остальные необходимые факторы. В качестве результата прогноза будет указан столбец «КОЛИЧЕСТВО».

Обучение нейросети (прогноз на 1 месяц вперед)

Перейдем непосредственно к самому построению модели прогноза. Откроем мастер обработки и выберем в нем нейронную сеть. На втором шаге мастера, согласно с принятым ранее решением, установим в качестве входных поля «КОЛИЧЕСТВО - 12», «КОЛИЧЕСТВО - 11», «КОЛИЧЕСТВО - 2» и «КОЛИЧЕСТВО - 1», а в качестве выходного - «КОЛИЧЕСТВО». Остальные поля сделаем информационными.

Количество-10	Имя столбца	COL1
Количество-9	Тип данных	Строковый
Количество-8	Назначение	Информационное
Количество-7	Вид данных	Дискретный
Количество-6	Уникальные значения Кол-во уникальных значений: 38	
Количество-5	2001-M01 2001-M02 2001-M03 2001-M04 2001-M05 2001-M06 2001-M07	
Количество-4		
Количество-3		
Количество-2		
Количество-1		
Количество		

Настройка нормализации...

Оставив все остальные параметры построения модели по умолчанию, обучим нейросеть (см. пример «прогнозирование умножения с помощью нейронной сети»).

После построения модели для просмотра качества обучения представим полученные данные в виде диаграммы и диаграммы рассеяния.

В мастере настройки диаграммы выберем для отображения поля «КОЛИЧЕСТВО» и «КОЛИЧЕСТВО_OUT» - реальное и спрогнозированное значение.

Метка столбца	Тип данных	Цвет
<input type="checkbox"/> Количество-6	9.0 Вещественный	
<input type="checkbox"/> Количество-5	9.0 Вещественный	
<input type="checkbox"/> Количество-4	9.0 Вещественный	
<input type="checkbox"/> Количество-3	9.0 Вещественный	
<input type="checkbox"/> Количество-2	9.0 Вещественный	
<input type="checkbox"/> Количество-1	9.0 Вещественный	
<input checked="" type="checkbox"/> Количество	9.0 Вещественный	■
<input checked="" type="checkbox"/> Количество_OUT	9.0 Вещественный	■
<input type="checkbox"/> Количество_ERR	9.0 Вещественный	■

Тип: Линии

Подписи по X: Дата (Год + Месяц)

☐ Значения по X

Результатом будет два графика.

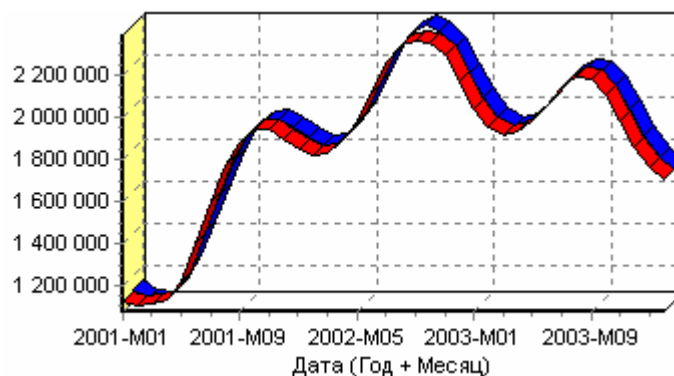
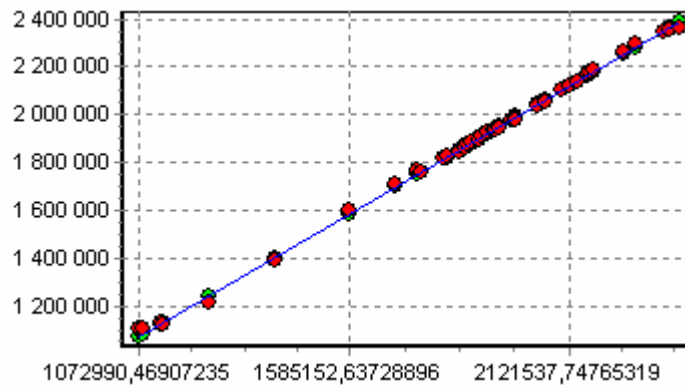


Диаграмма рассеяния более наглядно показывает качество обучения.



Построение прогноза

Нейросеть обучена, теперь осталось самое главное – построить требуемый прогноз. Для этого открываем мастер обработки и выбираем появившийся теперь обработчик «Прогнозирование».

На втором шаге мастера предлагается настроить связи столбцов для прогнозирования временного ряда – откуда брать данные для столбца при очередном шаге прогноза. Мастер сам верно настроил все переходы, поэтому остается только указать горизонт прогноза (на сколько вперед будем прогнозировать) равным трем, а также, для наглядности, необходимо добавить к прогнозу исходные данные, установив в мастере соответствующий флажок.

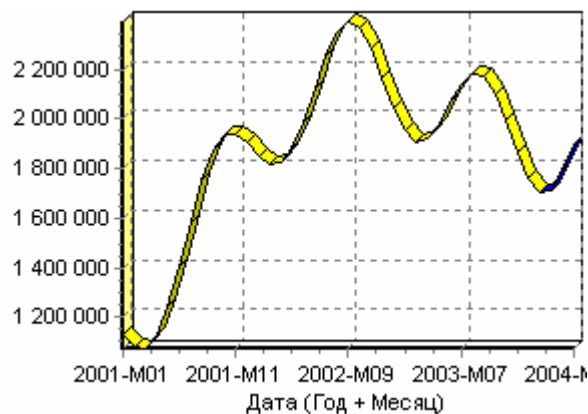
Столбец	При очередном шаге брать значения из
Дата (Год + Месяц)	
Количество-12	Количество-11
Количество-11	Количество-10
Количество-10	Количество-9
Количество-9	Количество-8
Количество-8	Количество-7
Количество-7	Количество-6
Количество-6	Количество-5
Количество-5	Количество-4
Количество-4	Количество-3
Количество-3	Количество-2

Горизонт прогноза: ☒ Добавлять горизонт прогноза ☒ Исходные данные

После этого необходимо в качестве визуализатора выбрать диаграмму прогноза, которая появляется только после прогнозирования временного ряда.

В мастере настройки столбцов диаграммы прогноза необходимо указать в качестве отображаемого столбец «КОЛИЧЕСТВО», а в качестве подписей по оси X указать столбец «ШАГ ПРОГНОЗА».

Теперь аналитик может дать ответ на вопрос, какое количество товаров будет продано в следующем месяце и даже два месяца спустя.



Выводы.

Данный пример показал, как с помощью Deductor Studio прогнозировать временной ряд. При решении задачи были применены механизмы очистки данных от шумов, аномалий, которые обеспечили качество построения модели прогноза далее и соответственно достоверный результат самого прогнозирования количества продаж на три месяца вперед. Также был продемонстрирован принцип прогнозирования временного ряда – импорт, выявление сезонности, очистка, сглаживание, построение модели прогноза и собственно построение прогноза временного ряда, а также экспорт результатов во внешний файл. Подобный сценарий – основа любого прогнозирования временного ряда с той разницей, что для каждого случая приходится, как получить необходимый временной ряд посредством инструментов Deductor Studio (например, группировки), так и подобрать параметры очистки данных и параметры модели прогноза (например, структуры сети, если используется обучение нейронной сети, определение значимых входных факторов). В данном случае приемлемые результаты получились с настройками по умолчанию, в большинстве же случаев предстоит работа по их подбору (например, оценивая качество модели по диаграмме рассеяния).

Применение скрипта

Скрипты предназначены для автоматизации процесса добавления в сценарий однотипных ветвей обработки. Скрипт позволяет применить имеющуюся в сценарии последовательность обработок одних данных к аналогичному набору других данных. Скрипт является готовой моделью, и поэтому входящие в него узлы не могут быть изменены отдельно от исходной ветки сценария. Тем не менее, на скрипте отражаются все изменения, вносимые в ветку, на которую он ссылается. То есть, при переобучении или перенастройке узлов этой ветки все сделанные изменения будут внесены в работу скрипта.

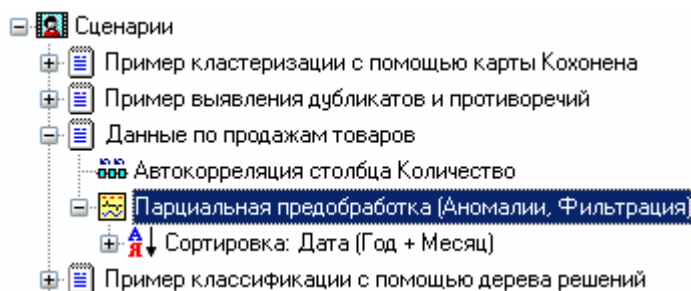
Предположим, что после импорта данных из двух разных баз данных требуется провести предобработку (очистить данные, сгладить, поменять названия столбцов, добавить несколько одинаковых выражений) и построить одинаковые модели прогноза, а затем экспортировать полученные данные обратно. Для первой ветви (первой БД) эти действия проводятся как обычно - последовательными шагами строится цепочка обработчиков. Для второго же источника (второй БД) достаточно создать узел импорта, к которому присоединить скрипт, основанный на уже построенной первой ветке. В этом скрипте будут выполнены точно такие же действия, как в оригинальной ветви. На выходе скрипта ставится узел экспорта, и вторая ветвь обработки готова к использованию.

Исходные данные

Рассмотрим механизм использования скрипта на примере данных файла «TradeSakes.txt». В нем находится информация по продажам некоторой группы товаров. Пусть необходимо сделать прогноз продаж на три месяца вперед. Поскольку мы уже сталкивались с подобной задачей выше (пример прогноза временного ряда), то уже имеется готовая цепочка действий для достижения данной цели. В данном примере именно ее мы и применим к исходным данным. После импорта данных запустим мастер обработки и выберем в качестве обработчика «Скрипт».

Указание цепочки выполняемых обработок

На следующем шаге следует выбрать узел сценария, с которого начнется исполнение скрипта. Имя выбранного начального узла отображается в строке «Начальный этап обработки». Для выбора другого узла нужно нажать кнопку в правой части этой строки, после чего на экране появится окно "Выбор узла". В этом окне показано все дерево сценария. Выберем в качестве начального узел «Парциальная предобработка (Аномалии, Фильтрация)».



После выбора начального узла следует задать соответствия столбцов исходного набора данных полям выбранного узла. В нижней части экрана находится таблица со списком полей исходного набора в левом столбце и полей выбранного узла - в правом. Для каждого поля начального узла надо задать поле-источник исходного набора. Для этого следует, щелкнув два раза в левом столбце напротив имени нужного поля, выбрать из выпадающего списка имя столбца входного набора. Настроим соответствие полей, как показано на рисунке ниже:

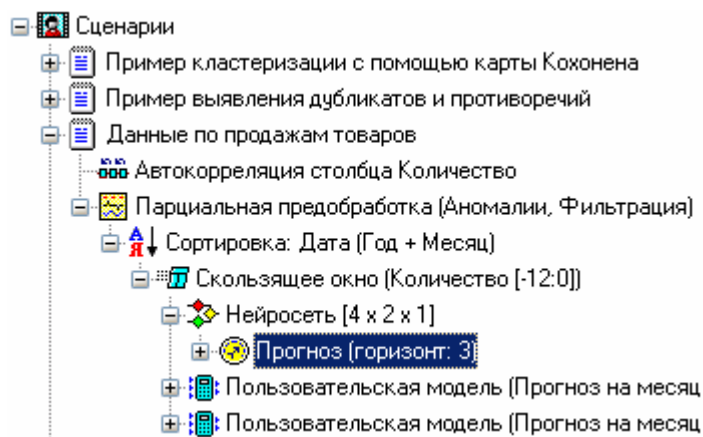
Начальный этап обработки

Парциальная предобработка (Аномалии, Фильтрация)

Соответствия исходных столбцов результирующим

Исходный столбец	Результирующий столбец
ab Дата (Год + Месяц)	ab Дата (Год + Месяц)
9.0 Количество	9.0 Количество

На следующем шаге мастера аналогичным образом выбирается конечный узел обработки:



После выбора конечного узла в нижней части окна будет показан список узлов, входящих в скрипт. При выполнении скрипта последовательно будут выполнены все узлы сценария из этого списка:

Конечный этап обработки

Прогноз (горизонт: 3)

Последовательность этапов обработки

№	Наименование этапа обработки
1	Парциальная предобработка (Аномалии, Фильтрация)
2	Сортировка: Дата (Год + Месяц)
3	Скользящее окно (Количество [-12:0])
4	Нейросеть [4 x 2 x 1]
5	Прогноз (горизонт: 3)

На следующем шаге запускается процесс анализа данных. Ход процесса обработки отображается с помощью прогресс-индикатора «Процент выполнения текущего процесса». В секции «Название процесса» отображается этап процесса обработки данных, выполняемый в данный момент. Запустим выполнение скрипта и перейдем на закладку выбора способа визуализации. Вот, например диаграмма с прогнозом объема продаж нашей группы товаров, полученного с использованием модели прогноза, построенной для другой группы товаров:



Выводы

Данный пример показал, как применять одни и те же действия к различным данным, что позволяет намного быстрее создавать аналитические решения. Имея заранее подготовленные цепочки действий для одной товарной группы можно несколькими щелчками мыши провести очистку, сглаживание, прогноз и т.д. для всех остальных товарных групп.

Условное выполнение ветки сценария

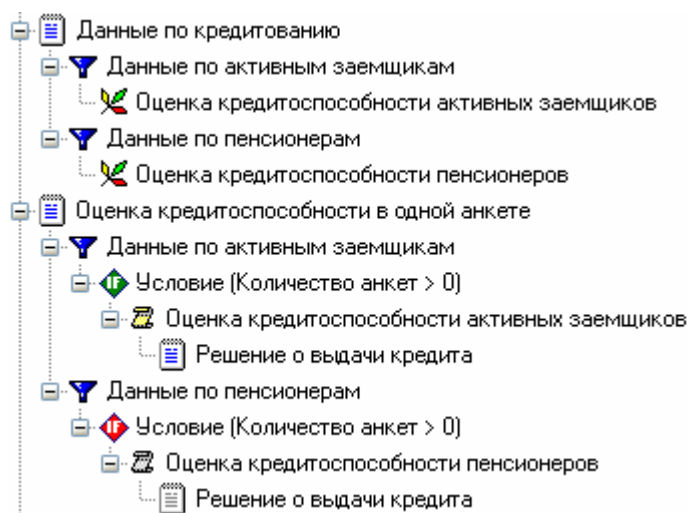
С помощью операции «Условие» можно организовать условное выполнение узлов сценария. При этом если заданное условие не выполняется, то узлы сценария, следующие за данным обработчиком, не будут обработаны.

Исходные данные

Рассмотрим механизм условного выполнения ветки сценария на примере задачи определения кредитоспособности физического лица. Аналитик построил две модели для разных сегментов заемщиков: лиц, моложе 50 лет, условно названными активными заемщиками и лиц, старше 50 лет, условно названными неактивными заемщиками. На вход системы подается одна анкета (находится в файле Credit1Sample.txt) и в зависимости от того, в какой сегмент попадает заемщик, необходимо применить первую или вторую модель оценки и затем записать результат в заранее определенный файл (Credit_Solution.txt). Далее предполагается использование этого файла в качестве связующего звена между моделью и остальными частями системы оценки кредитоспособности.

Цепочка обучения той или иной модели представлена в файле сценариев демонстрационного примера, в ветке с названием «Данные по кредитованию».

Цепочка «прогона» той или иной модели представлена в файле сценариев демонстрационного примера, в ветке с названием «Оценка кредитоспособности в одной анкете».



Сценарий построения моделей кредитоспособности представляет собой импорт кредитной истории из текстового файла, далее фильтрацией набор разделяется на 2 части: активные заемщики и заемщики – пенсионеры. Далее в каждой ветке обучаются различные модели оценки кредитоспособности.

Сценарий собственно оценки кредитоспособности представляет собой импорт единичной анкеты из текстового файла, далее фильтрацией набор разделяется на 2 части: активные заемщики и заемщики – пенсионеры. Поскольку в текстовом файле будет одна запись, то после фильтрации одна из ветвей окажется пустой. Тогда условие применения той или иной модели для оценки кредитоспособности заключается в существовании в ветке сценария строк для обработки, т.е. Количество анкет > 0. Далее в обеих ветках выполняются сценарии прогона анкеты через построенную скоринговую модель и экспорт результатов оценки в один и тот же текстовый файл – Credit_Solution.txt. Таким образом, вне зависимости от поданной на вход анкеты, результат обработки всегда будет попадать в один и тот же файл, который и будет использоваться для дальнейшей работы. Рассмотрим, каким образом задается такое условие.

Настройка условия

Запустим мастер обработки на узле фильтрации и выберем обработчик «Условие» и нажмем кнопку «Далее».


На следующем шаге указываются условия дальнейшего выполнения ветки сценария. Этот шаг мастера аналогичен шагу мастера фильтрации данных. Имя поля позволяет выбрать поле, по агрегированному значению которого должно быть проверено условие. В этом списке также присутствует имя «*». К этому полю можно применить функцию агрегации «количество». Агрегация позволяет установить функцию агрегации, применяемую к полю выбранному в предыдущем столбце. В поле «Условие» - указывается условие, по которому нужно проверить выражение. В поле «Значение» указывается значение, с которым сравнивается результат функции агрегации в соответствии с заданным условием. В данном случае необходимо установить поле «*», функцию агрегации «Количество», указать условие выполнения "> 0":

Операция	Имя поля	Функция	Условие	Значение
	*	s Количество	>	0

Расчет условия

На следующем шаге мастера осуществляется расчет факта выполнения или невыполнения условия:

Результат расчета условия

 Не рассчитано


Если условие не выполняется, то узлы сценария следующие за данным обработчиком не будут выполнены.


Название процесса


Процент выполнения текущего процесса

0%

Время выполнения

 Пуск

 Пауза

 Стоп

На этом шаге мастера необходимо нажать кнопку «Пуск». При этом будет рассчитано условие. Если условие выполняется, то на следующем шаге мастера доступны стандартные виды визуализации данных.

Выводы

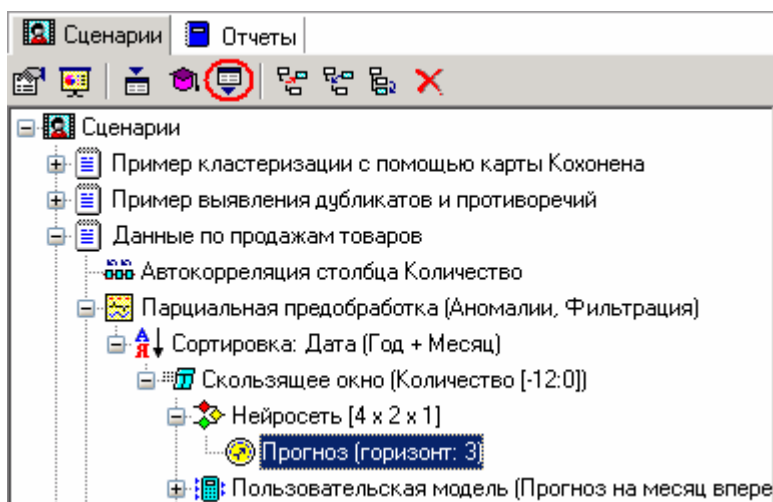
Данный пример показал, каким образом можно организовать условное выполнение узлов сценария. Как показала практика, данный механизм особенно актуален при организации взаимодействия с помощью текстовых файлов, обработке единичных записей. Также без него не обойтись при реализации нелинейной обработки исходных данных, что достаточно часто встречается на практике.

Экспорт данных.

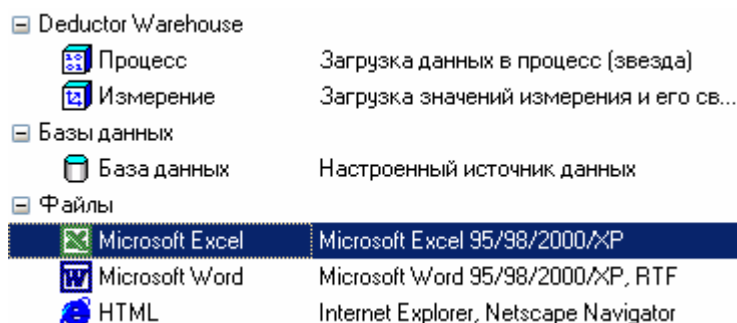
Продemonстрируем экспорт полученных результатов прогнозирования в Excel (пример прогноза временного ряда).

Экспорт позволяет сохранить результат обработки в одном из популярных форматов: Word, Excel, HTML, XML, DBF, Текстовый файл или скопировать в буфер обмена.

Мастер экспорта вызывается с панели сценариев.



После вызова мастера необходимо выбрать формат. Выберем экспорт в Microsoft Excel.



На следующем шаге необходимо указать поля, которые необходимо экспортировать. Результат прогноза находится в поле «КОЛИЧЕСТВО+1», информация о шаге прогноза в поле «ШАГ ПРОГНОЗА». Также необходимо указать год и месяц продаж, находящийся в поле «Дата (Год + Месяц)». Выберем три эти поля для экспорта. На следующем шаге необходимо указать имя файла, в который будут экспортированы результаты анализа. Также можно указать опцию «Открыть файл после экспорта». Пусть файл будет называться «Прогноз на 3 месяца.xls». В отображаемом после этого визуализаторе «Описание» можно узнать о настройках экспорта.

Заключение

Рассмотренные примеры позволяют говорить о том, что Deductor Studio предназначен для решения широкого спектра задач, связанных с обработкой структурированных, представленных в виде таблиц, данных. Он предоставляет аналитикам инструментальные средства необходимые для решения самых разнообразных аналитических задач, начиная от разнообразной аналитической отчетности и заканчивая созданием на его базе системы поддержки принятия решения. На базе приведенных примеров было рассмотрено применение таких технологий, как многомерный анализ, нейронные сети, деревья решений, самоорганизующиеся карты, спектральный анализ и множество других.

Одним из достоинств программы является то, что все механизмы унифицированы и выполняются при помощи мастеров. В примере прогнозирования временного ряда было показано, что все реализованные механизмы позволяют в рамках одного приложения пройти весь цикл анализа данных – получить информацию из произвольного источника, провести необходимую обработку (очистку, трансформацию данных, построение моделей), отобразить полученные результаты наиболее удобным образом и экспортировать результаты на сторону. Собственно в блоке обработки данных и производятся самые важные с точки зрения анализа действия. Наиболее важной особенностью механизмов обработки, реализованных в Deductor Studio, является то, что полученные в результате обработки данные можно опять обрабатывать любым из доступных в системе методов. Таким образом, можно строить сколь угодно сложные сценарии обработки.

Каждый из рассмотренных механизмов анализа и обработки дает ценные результаты. Но только их совместное применение и возможность комбинирования обеспечивают совершенно новое качество решений.

Завершающим шагом в сценарии обработки чаще всего является экспорт данных. Результаты обработки можно на любом шаге обработки экспортировать для последующего использования в других системах, например, учетных системах.

Объединение всех, описанных выше механизмов в рамках единой программы, позволяют уменьшить время создания законченных решений, упростить интеграцию с другими приложениями, увеличив при этом общую производительность. Все это сочетается с гибкостью и простотой использования. Наличие большого набора инструментов позволяет, начав с небольших подзадач, рассмотренных выше постепенно наращивать возможности, двигаясь к созданию системы поддержки принятия решений.

Одной из важных продемонстрированных возможностей является возможность при помощи Deductor Studio не только построить модели, но и провести анализ по принципу «что-если», т.е. оценить, как может измениться тот или иной показатель при изменении любого влияющего фактора. Для реализации этого простого в использовании и одновременно мощного механизма предназначен специальный визуализатор. Результаты «что-если» анализа можно просмотреть как в табличном виде, так и графическом. Такого рода механизмы визуализации являются готовым инструментом для оптимизации процессов.

Таким образом, на рассмотренных примерах были продемонстрированы следующие возможности:

- Возможность импорта данных из популярных форматов хранения информации.
- Возможность создания сценариев обработки, использующих все имеющиеся в программе обработки и комбинирующие их произвольным образом.
- Возможность применения всех доступных методов визуализации на любом шаге обработки.
- Возможность экспорта результатов обработки в сторонние системы.

Простота использования, высокая производительность, возможность комбинирования множества методов анализа, гибкие механизмы интеграции с существующими системами гарантируют быстрое создание качественных решений.