ПЕРВОЕ ВЫСШЕЕ ТЕХНИЧЕСКОЕ УЧЕБНОЕ ЗАВЕДЕНИЕ РОССИИ

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
федеральное государственное бюджетное образовательное учреждение высшего образования
САНКТ-ПЕТЕРБУРГСКИЙ ГОРНЫЙ УНИВЕРСИТЕТ

УТВЕРЖДАЮ

Руководитель программы аспирантуры
доцент Ю.В. Ильюшин

МЕТОДИЧЕСКИЕ РЕКОМЕНДАЦИИ ДЛЯ САМОСТОЯТЕЛЬНОГО ИЗУЧЕНИЯ ДИСЦИПЛИНЫ

СИСТЕМЫ ИСКУССТВЕННОГО ИНТЕЛЛЕКТА

Подготовка научных и научно-педагогических кадров в аспирантуре

Область науки:	2. Технические науки
Группа научных специальностей:	2.3. Информационные технологии и телекоммуникации
Научная специальность:	2.3.1. Системный анализ, управление и обработка информации, статистика
Отрасли науки:	Технические
Форма освоения программы аспирантуры:	Очная
Срок освоения программы аспирантуры:	3 года
Составитель:	к.т.н., доц. Мазаков Е.Б.

Санкт-Петербург

Дисциплина «Системы искусственного интеллекта» входит в составляющую «Дисциплины (модули), в том числе элективные, факультативные дисциплины (модули), дисциплины, направленные на подготовку к сдаче кандидатских экзаменов» образовательного компонента программы подготовки научных и научно-педагогических кадров в аспирантуре по научной специальности 2.3.1. Системный анализ, управление и обработка информации, статистика и изучается в 4 семестре.

Цель изучения дисциплины – формирование и развитие у аспирантов знаний, навыков и умений применения методов теории интеллектуальных систем, приобретение навыков по использованию интеллектуальных систем, изучение основных методов представления знаний и моделирования рассуждений.

Основные задачи дисциплины:

– овладение навыками и знаниями в области искусственного интеллекта.

– формирование знаний, навыков и умений в области разработки алгоритмов решения комплексных задач с использованием искусственного интеллекта.

Самостоятельная работа аспирантов

Самостоятельная работа аспиранта включает:

- тематическую работу с рекомендованной научной литературой;
- самостоятельное изучение разделов дисциплины;
- исследовательскую работу, анализ научных публикаций по теме курса;
- подготовку к зачетам.

Самостоятельная работа обучающихся направлена на углубление и закрепление знаний, полученных на лекциях, выработку навыков самостоятельного активного приобретения новых, дополнительных знаний, подготовку к предстоящим учебным занятиям и промежуточному контролю.

Самостоятельная работа аспирантов - планируемая учебная и научно-исследовательская работа аспирантов, выполняемая во внеаудиторное время по заданию и при методическом руководстве преподавателя.

Целью самостоятельной работы аспирантов является овладение фундаментальными и профессиональными знаниями и умениями по профилю будущей специальности.

Основные задачи самостоятельной работы аспирантов

- изучение теоретического курса, углубление и расширение теоретического курса, углубление и расширение теоретической подготовки в области правовой охраны интеллектуальной собственности;
- формирование самостоятельного мышления, способностей к саморазвитию и самореализации;
- закрепление полученных теоретических знаний и практических умений;
- использование материала, полученного в ходе самостоятельных занятий в процессе ознакомления с нормативной, справочной документацией и специальной литературой.

Основными формами самостоятельной работы аспирантов являются:

- работа с учебной/научной литературой и углубление знаний при решении практических задач;
- подготовка к зачету.

Требуется изучить следующие темы и ответить на основные вопросы:

Тема 1. Введение в системы искусственного интеллекта

1. Понятие искусственного интеллекта.
2. Научные направления искусственного интеллекта
3. Технология экспертных систем
4. Эволюционные модели.
5. Нейросетевые модели

6. Модели представления знаний.
7. Логические модели представления знаний
8. Продукционная модель представления знаний

Тема 2. Инструментальные средства интеллектуального анализа данных

1. Инструментальные средства разработки систем искусственного интеллекта.
2. Инженерия знаний.
3. Методы извлечения знаний.
4. Методы структурирования знаний.
5. Коммуникативные методы извлечения знаний.
6. Текстологические методы извлечения знаний.
7. Управление знаниями.
8. Эволюционное моделирование.
9. Метод группового учета аргументов.
10. Генетические алгоритмы.
11. Теоретические аспекты применения генетических алгоритмов.
12. Кодирование фенотипов в хромосомы.
13. Функции приспособленности.
14. Операторы репродукции.
15. Формирование начальной, текущей популяции и родительского пула.
16. Биологический и искусственный нейрон.
17. Перцептрон.
18. Проблема исключающего «ИЛИ».
19. Виды искусственных нейронных сетей.
20. Карты Кохонена.
21. Машинное обучение.
22. Сеть Кохонена
23. Сети Хопфилда.
24. Сети встречного распространения.
25. Модели теории адаптивного резонанса.
26. Аналитический и информационный подходы к моделированию.
27. Структурированные данные: формы представления данных, типы данных, виды данных.
28. Основные этапы интеллектуального анализа данных (ИАД).
29. Машинное обучение и классы задач ИАД.
30. Структура и архитектура информационно-аналитических систем и систем поддержки принятия решений.
31. Организация облачных хранилищ данных.
32. Очистка данных. Оценка пригодности данных к анализу.
33. Оценка качества данных по их происхождению.
34. Предобработка данных и ее отличие от очистки.
35. Фильтрация данных. Обработка дубликатов и противоречий.
36. Обнаружение аномальных значений специальными методами.
37. Трансформация, объединение и квантование данных.
38. Дисперсионный анализ.
39. Ковариация и корреляция.
40. Простая и множественная линейная регрессия.
41. Оценка соответствия линейной регрессии реальным данным.
42. Регрессия с категориальными входными переменными.
43. Множественная логистическая регрессия.
44. Простой байесовский классификатор.

Тема 3. Нейросетевые технологии анализа данных

1. Временной ряд и его компоненты.

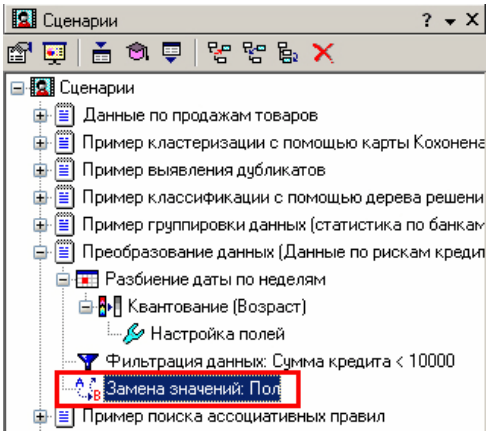
2. Трендовые модели прогнозирования.
3. Скользящее среднее и экспоненциальное сглаживание.
4. Ассоциативные правила. Алгоритм Apriori.
5. Методы поиска логических закономерностей.
6. Задачи кластерного анализа.
7. Иерархические и итеративные методы кластеризации.
8. Особенности кластеризации в качественных и количественных шкалах.
9. Кластеризация данных по матрице объект-признак.
10. Кластеризация данных по матрице связи.
11. Алгоритм кластеризации k-means.
12. Сети и карты Кохонена.
13. Назначение компонентного и факторного анализа.
14. Применение компонентного и факторного анализа к задачам ИАД.
15. Методы распознавания образов с учителем и без учителя.
16. Алгоритмы построения деревьев решений.
17. Информационный подход к моделированию нейрона
18. Принципы построения нейронных сетей.
19. Место нейронных сетей среди других методов решения задач ИАД.
20. Алгоритмы обучения нейронных сетей.
21. Алгоритм обратного распространения ошибки.
22. Особенности нейронных сетей и ее влияние на свойства сети.
23. Ансамбли моделей. Бэггинг и бустинг.

ОТРАБОТКА ПРАКТИЧЕСКИХ ЗАДАНИЙ

Практическая работа 1. 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.

Использование этого обработчика демонстрируется в выделенном фрагменте сценария проекта *"Демопример анализа данных.ded"*.

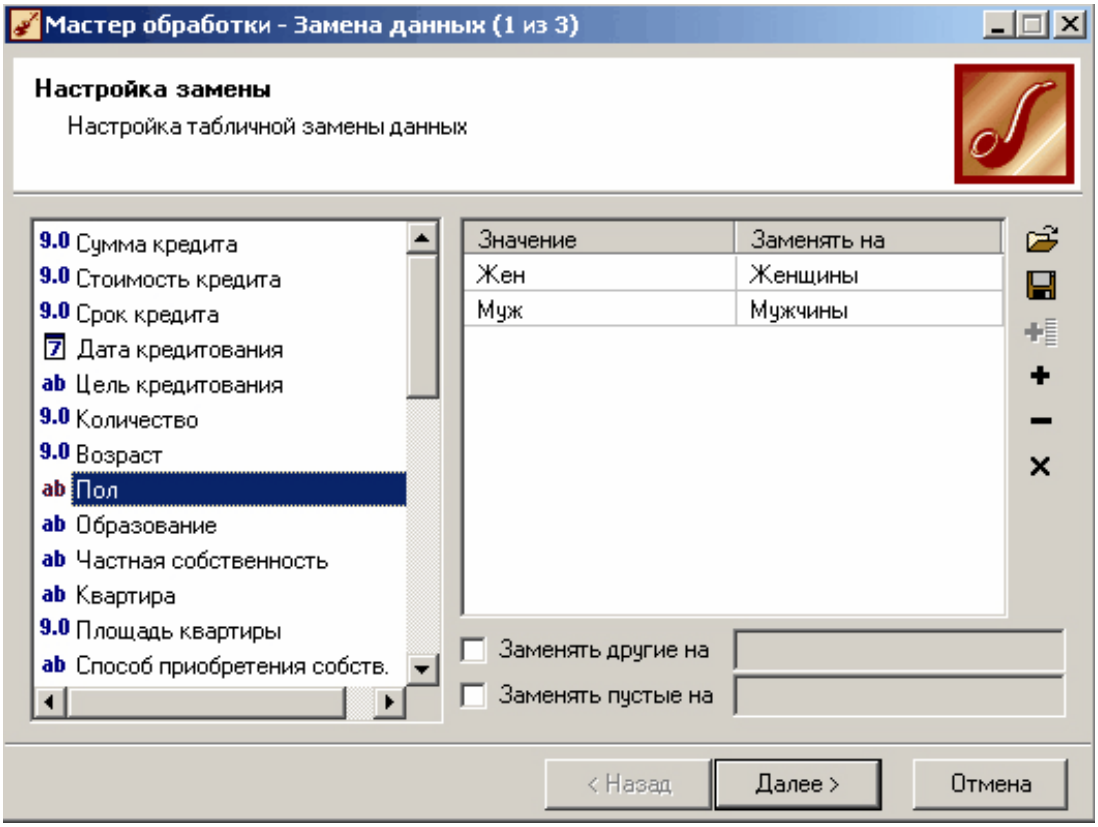


1. Исходные данные

Продемонстрируем применение 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, "муж" - "мужчины", "жен" - "женщины".

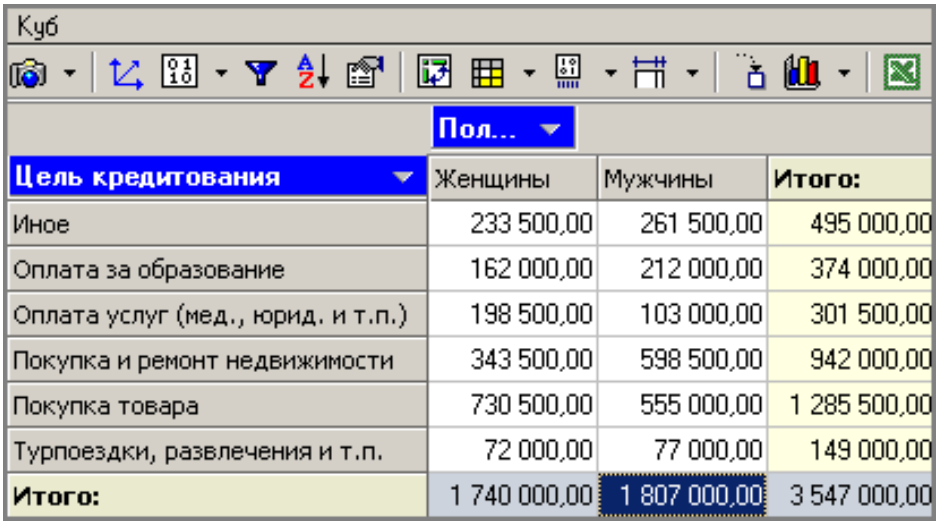
2. Выполнение замены

В Мастере замены следует 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 напишем напротив "муж" - "Мужчины", напротив "жен" - "Женщины". Затем 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 отметим как "неиспользуемый".



3. Результат

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

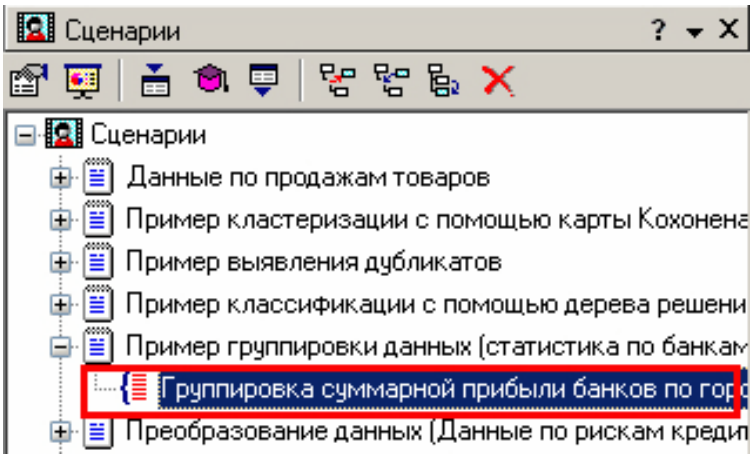


Цель кредитования	Женщины	Мужчины	Итого:
Иное	233 500,00	261 500,00	495 000,00
Оплата за образование	162 000,00	212 000,00	374 000,00
Оплата услуг (мед., юрид. и т.п.)	198 500,00	103 000,00	301 500,00
Покупка и ремонт недвижимости	343 500,00	598 500,00	942 000,00
Покупка товара	730 500,00	555 000,00	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

Группировка данных

Сложно делать выводы на основе необработанной первичной информации. Аналитику для принятия решения очень часто нужна сводная информация. Совокупные 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 агрегировать данные в полях-фактах для дальнейшего анализа.

Выделенная часть сценария, в которой демонстрируется данный обработчик, находится в проекте *"Демопример анализа данных.ded"*.



1. Исходные данные

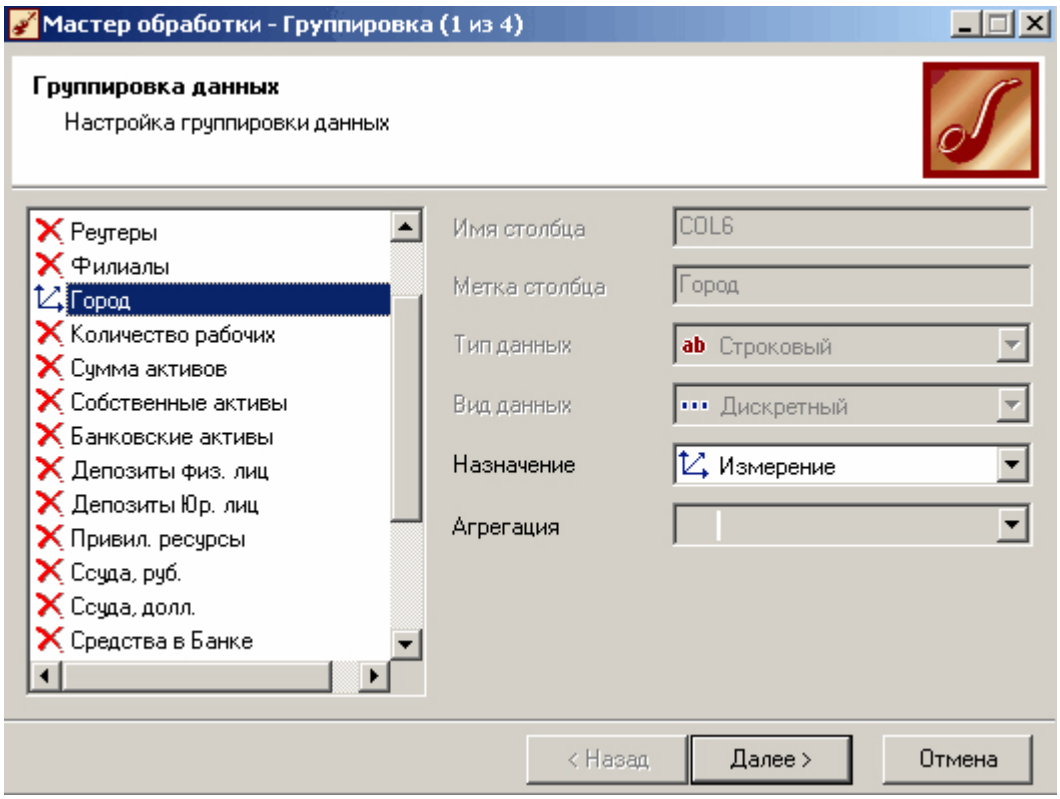
Допустим, что у аналитика имеется статистика по банкам России за определенный период. Она находится в файле "Banks.txt". Перед ним стоит задача выявления ряда городов, в которых прибыль банков самая большая, для использования этих данных в дальнейшем. Для этого аналитик должен обратить внимание на следующие 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 используется 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 сгруппированные данные, выбрав в качестве измерения поле "Город", а в качестве факта "Прибыль". Но нам нужно получить эти данные не только для визуализации, но и для последующей обработки, следовательно, необходимо применить обработчик "Группировка".

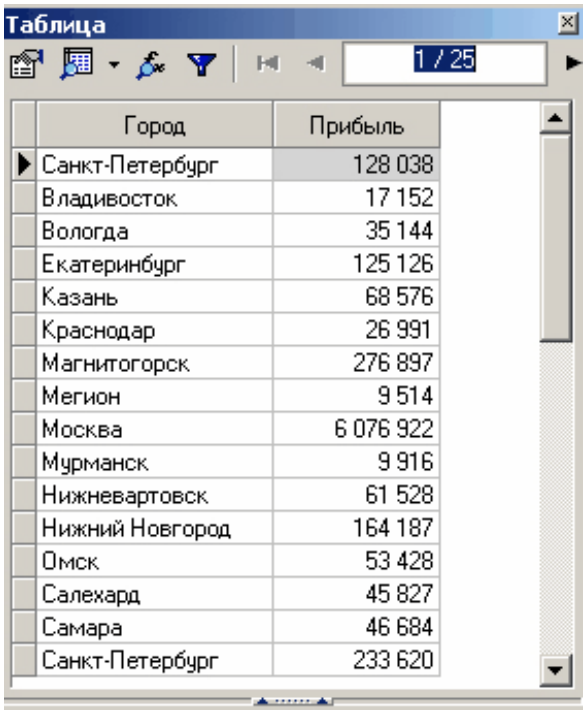
2. Группировка по городам

Находясь в узле импорта, запустим Мастер обработки. Выберем в качестве метода 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 "Сумма".



3. Результат

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.



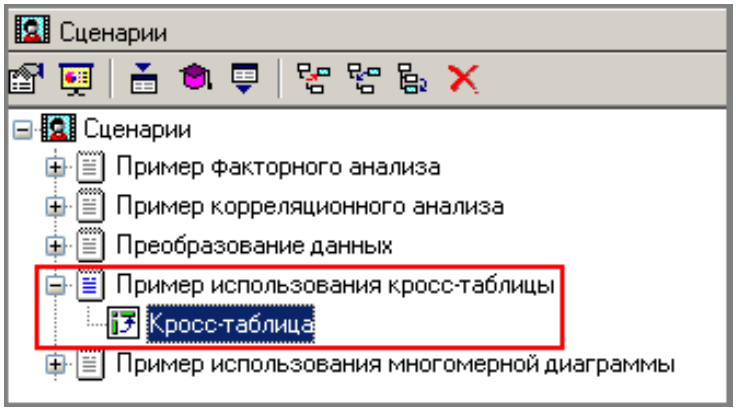
Город	Прибыль
Санкт-Петербург	128 038
Владивосток	17 152
Вологда	35 144
Екатеринбург	125 126
Казань	68 576
Краснодар	26 991
Магнитогорск	276 897
Мегион	9 514
Москва	6 076 922
Мурманск	9 916
Нижневартовск	61 528
Нижний Новгород	164 187
Омск	53 428
Салехард	45 827
Самара	46 684
Санкт-Петербург	233 620

Практическая работа 2. Кросс–таблица

Данный обработчик предназначен для преобразования исходной структуры таблицы данных в удобную для работы форму. С его помощью задаются новые поля таблицы из уже существующих, на основе преобразования значений выбранного поля в новые

поля с помощью встроенного обработчика фильтрации. Например: поле "месяц" содержащее в себе значения: "январь", "февраль", "март" преобразуется в три соответствующих поля. Значениями которого будут являться агрегированное поле фактов, заданное аналитиком. Данный обработчик можно заменить обработчиками: "Фильтр" — с помощью которого выбираются значения на основе которых будет строится первое поле таблицы, далее применяется "Калькулятор" — который формирует измерения нового поля и присваивает ему имя; данный алгоритм повторяется для всех предусмотренных полей; после чего все созданные поля собирают с помощью "Группировки" .
На основе кросс–таблицы удобно вычислять экономические показатели, рассчитываемые на основе прошедших месяцев. "Кросс–таблица" является одним из инструментов Deductor Studio.

Рассмотрим пример использования данного обработчика в фрагменте сценария проекта *"Демопример анализа данных.ded"*.



1. Исходные данные

Продемонстрируем применение "Кросс–таблицы", используя данные о стоимости продуктов, входящих в потребительскую корзину за четыре месяца. Исходные данные находятся в файле "basket_of_goods.txt".
Необходимо оценить индексы роста цен на продукты питания.

Таблица

19 / 132

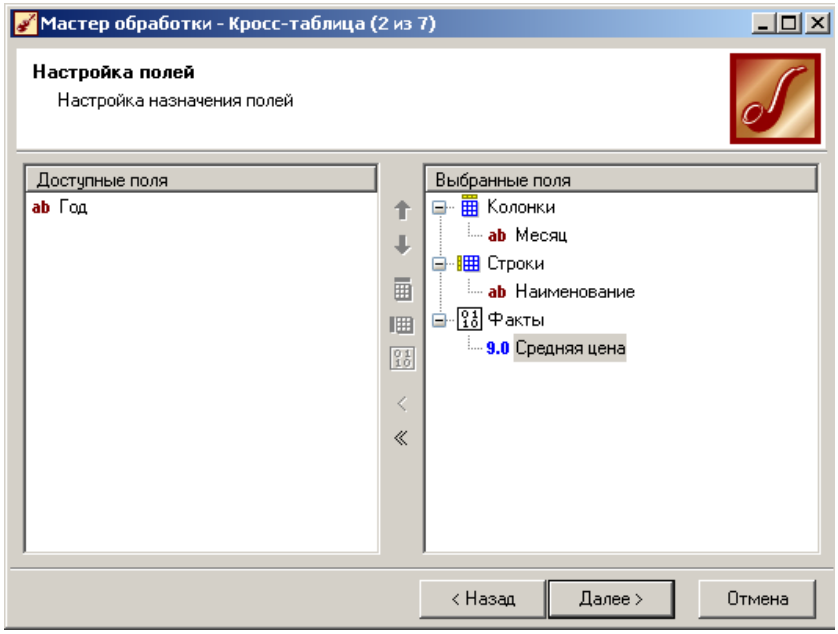
Наименование	Месяц	Год	Средняя цена
Баранина с костями, кг	сентябрь	2008	183,6
Баранина с костями, кг	октябрь	2008	185,9
Баранина с костями, кг	ноябрь	2008	187,3
Баранина с костями, кг	декабрь	2008	190,7
Вермишель, кг	сентябрь	2008	39,6
Вермишель, кг	октябрь	2008	40,9
Вермишель, кг	ноябрь	2008	41,1
Вермишель, кг	декабрь	2008	41,6
Говядина I кат (кроме бескостного мяса), кг	сентябрь	2008	164,7
Говядина I кат (кроме бескостного мяса), кг	октябрь	2008	167,4
Говядина I кат (кроме бескостного мяса), кг	ноябрь	2008	171,5
Говядина I кат (кроме бескостного мяса), кг	декабрь	2008	175,3

Вид исходной таблицы мало пригоден для вычислений индексов. Данную таблицу необходимо отредактировать, что бы в ней появились дополнительные поля. В которых содержалась бы информация о цене рассматриваемых продуктов питания за каждый месяц в отдельности. Применим обработчик "Кросс–таблица".

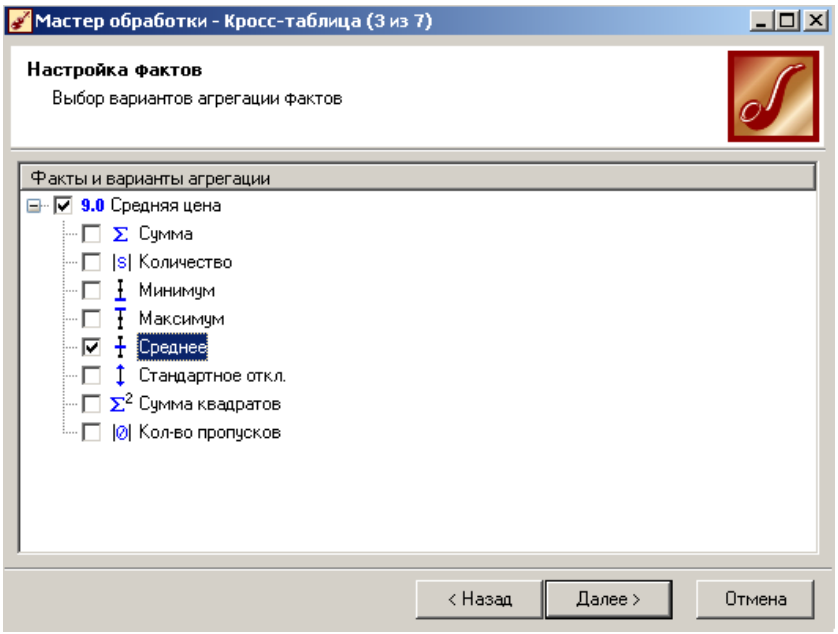
2. Преобразование исходной таблицы данных

Вызовем "Мастер обработки" и в появившемся окне выберем обработчик "Кросс–таблица".

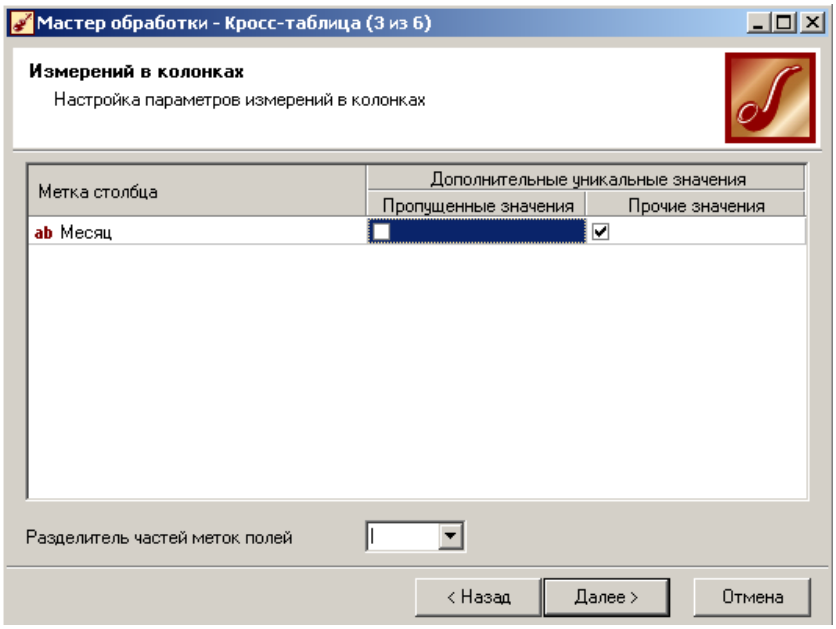
Следующим шагом будет настройка используемых полей для формирования таблицы. Используемые поля для построения должны находится либо в колонках, либо в строках. В колонки помещают поля на основе значений которых будут создаваться новые, их значениями будут выбранные факты. В строки помещаются поля, которые не нуждаются в изменении. Настроим данное окно: переместим "Месяц" в колонки, а "Наименование" в строки, при этом необходимо обязательно указать факты в данном случае — "Средняя цена". Новая таблица будет содержать поля: "Наименование" — название продуктов, входящих в потребительскую корзину; "Сентябрь" — средняя цена, данных продуктов за сентябрь месяц, "Декабрь" — средняя цена, продуктов за декабрь месяц.



Следующим шагом необходимо настроить параметры агрегации выбранных фактов. В нашем случае выберем среднюю агрегацию.



После нажатия кнопки "Далее" открывается следующее окно "Мастера обработки", в котором выбирается настройка параметров измерений в колонках. В нем резервируются дополнительные поля для возможного внесения изменений в значения исходного поля таблицы, а также для измерений, в названии которых содержатся пропуски.



Так как у нас нет данных о цене товара, с неопределенным месяцем, то галочку рядом с "Пропущенными значениями" ставить не будем. "Прочие значения" отметим галочкой, так как в дальнейшем мы рассчитываем пополнить исходную таблицу еще одним месяцем, данные которого запишутся в данный столбец.

Все настройки заданы, запустим процесс на выполнение.

3. Результат

Из множества предлагаемых визуализаторов выберем "Таблицу"

Таблица

1 / 33

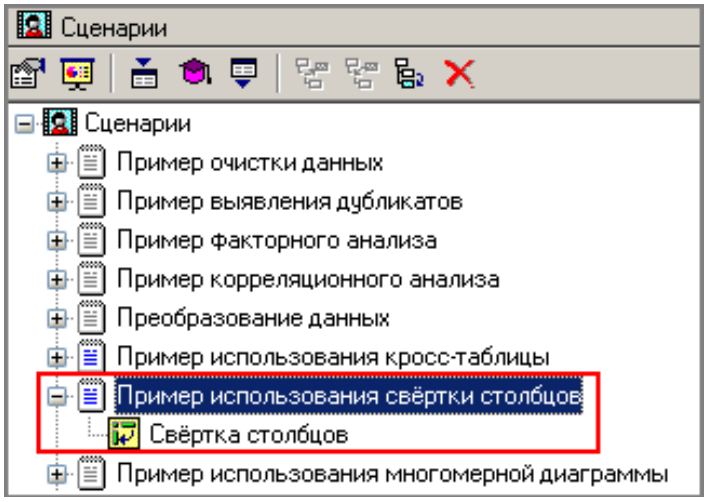
Наименование	декабрь	ноябрь	октябрь	сентябрь	<Прочее>
	Средняя цена	Средняя цена	Средняя цена	Средняя цена	Средняя цена
Баранина с костями, кг	190,7	187,3	185,9	183,6	
Вермишель, кг	41,6	41,1	40,9	39,6	
Говядина I кат (кроме бескостного мяса), кг	175,3	171,5	167,4	164,7	
Горох и фасоль, кг	27,2	26,9	26,8	26,3	
Капуста белокочанная свежая, кг	16,7	16,1	15,1	16,3	
Карамель, кг	83,1	81,1	79,6	77,5	
Картофель, кг	19	18,4	17,7	18,2	
Куры (кроме куриных окорочков), кг	90,5	89	85,9	84,6	
Лук репчатый, кг	19	19,4	19,7	21,5	
Маргарин, кг	64,4	63,6	63,5	62,4	
Масло подсолнечное, кг	76,5	76,7	76,8	76,3	
Масло сливочное, кг	173	172,5	172,2	168,8	
Молоко цельное разливное, л	23,9	23,8	23,5	23	
Морковь, кг	27	26,6	26,8	29	
Мука пшеничная, кг	24,4	24,2	24	23,7	
Огурцы, кг *	47,5	46	45,4	44,8	

Таким образом, после обработки получили новую таблицу данных, на основе которой удобно производить необходимые вычисления индексов.
Данную таблицу можно получить с помощью группы обработчиков: "Фильтр", "Калькулятор" и "Группировка", но они делают сценарий очень громоздким и неудобным к исправлению. Использование "Кросс– диаграммы" существенно сокращает время построения сценария и обработки.

Свёртка столбцов

Обработчик "Свёртка столбцов" как и "Кросс–таблица" служит для преобразования исходной структуры набора данных в форму удобную для обработки. Но в отличие от "Кросс–диаграммы" которая формирует из выбранного поля данных несколько новых полей со значениями, сформированными на основе заданных фактов. "Свёртка столбцов" наоборот собирает все обозначенные поля в одно. В Deductor Studio такую возможность предоставляет инструмент "Свёртка столбцов".
Ранее "Свертку столбцов" заменяла группа обработчиков: "Настройка полей" — которая отфильтровывала поля, которые в дальнейшем с помощью
"Слияние" собирались в один столбец. Но данный алгоритм мало подвержен изменению и его редактирование связано с большими трудностями, так как насчитывает много узлов, каждый из которых необходимо откорректировать при изменении.
Данный обработчик применяется, для подготовки данных к использованию в аналитических моделях.

Покажем на примере, использование данного обработчика в фрагменте сценария проекта *"Демопример анализа данных.ded"*.



1. Исходные данные

Рассмотрим алгоритм использования обработчика "Свёртка столбцов" на примере данных файла "region_servise.txt". В нем содержаться данные по объему предоставляемых платных услуг, выраженных в млн.руб с 1995 по 2006г. населению. Рассмотренные данные понадобятся аналитику для прогнозирования развития рынка услуг и отслеживания его динамики.

Необходимо преобразовать исходную таблицу, в такой вид что бы она содержала данные по объему предоставляемых услуг в одном столбце. Воспользуемся обработчиком "Свёртка столбцов".

Таблица

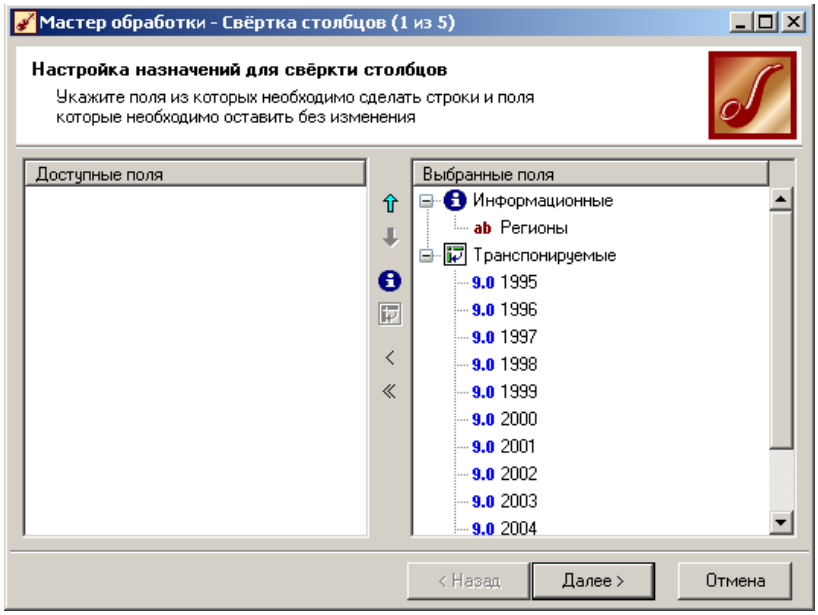
1 / 18

Регионы	1995	1996	1997	1998	1999	2000	2001
Белгородская область	733	1043	1539	1774	2239	3162	4726
Брянская область	901	1931	2479	1999	2676	3536	5026
Владимирская область	749	1165	1497	1565	2119	3095	4196
Воронежская область	1067	1724	2172	2541	3544	4936	7448
Ивановская область	622	1120	1501	1745	2097	2682	3265
Калужская область	529	964	1169	1279	1731	2321	3266
Костромская область	408	518	577	648	919	1263	1729
Курская область	554	960	1225	1226	1678	2075	3059
Липецкая область	511	785	1270	1505	1929	2736	3778
Московская область	4608	7053	9752	12403	19880	25454	36787
Орловская область	371	607	759	837	1199	1665	2435
Рязанская область	570	790	917	1094	1688	2172	3331
Смоленская область	625	1115	1391	1516	1849	2495	3094
Тамбовская область	455	761	1090	1365	1830	2615	3844
Тверская область	636	1113	1357	1576	2077	2822	4062
Тульская область	695	1398	1640	1950	2568	3802	5043
Ярославская область	856	1288	1904	2251	2850	3906	5654
г.Москва	20179	45541	74508	90069	134534	180724	226174

2. Преобразование исходной таблицы данных

Выберем обработчик "Свёртка–столбцов" из окна "Мастер обработки". Наша задача заключается в создании столбца фактов — "Объем предоставляемых платных услуг населению" и столбца измерения — "Год", где будут храниться рассматриваемые годы.

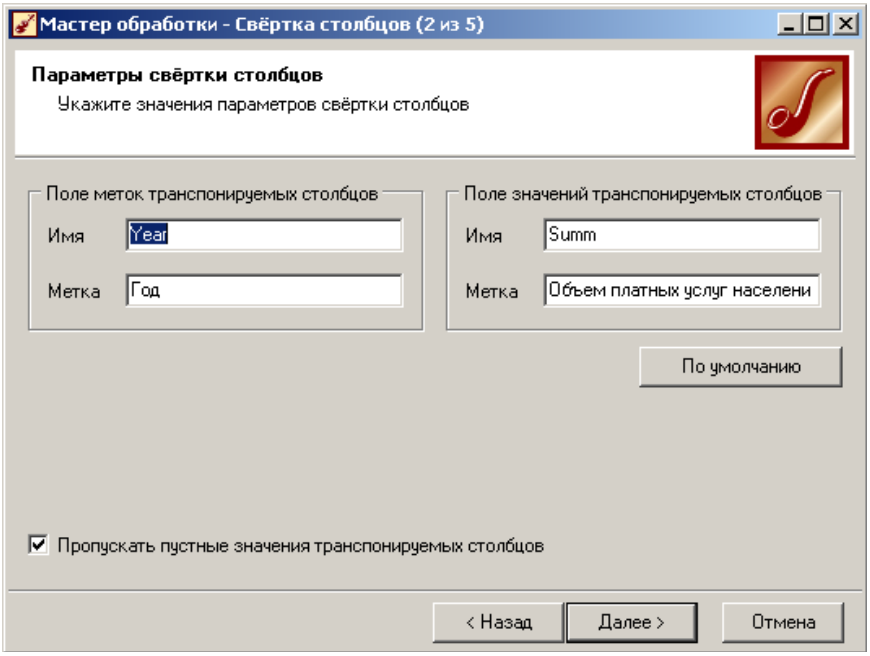
В появившемся окне "Мастера обработки" произведем настройку полей, переместим "Регионы" в информационные, а все рассматриваемые годы в транспонируемые. Поля, которые переместились в информационные, изменению не подлежат, транспонируемые поля объединяются в одно с помощью слияния их значений.



На следующем шаге задаем название новым полям. В "Поле меток транспонируемых столбцов" которое будет, содержать перечисление рассматриваемых лет присвоим значе-

ния: имя — "Year" и метка "Год". "Поле значений транспонируемых столбцов" содержащее данные по объему предоставляемых платных услуг населению по годам будет иметь имя "Summ" и метку "Объем платных услуг населению". Имеется возможность востановить значения по умолчанию нажатием соответствующей кнопки.

Поставим галочку в поле "Пропускать пустые значения транспонированных полей", в соответствии с чем пустые значения транспонированных полей будут исключаться из рассматриваемого набора данных.



После нажатия кнопки "Далее" запустим процесс на обработку.

3. Результат

На последнем шаге мастера выберем для просмотра результата визуализатор "Таблица".

Таблица

1 / 234

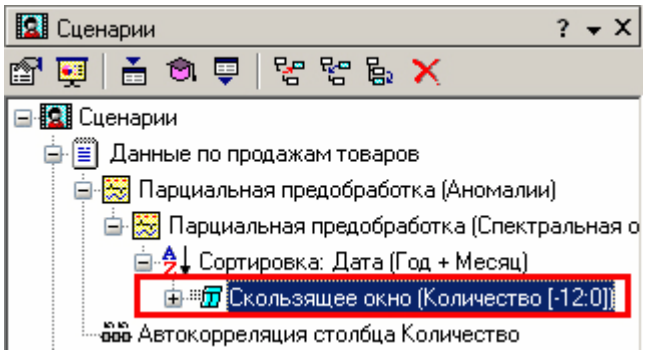
Регионы	Год	Объем платных услуг населению в млн. руб
Белгородская область	1995	733
Белгородская область	1996	1043
Белгородская область	1997	1539
Белгородская область	1998	1774
Белгородская область	1999	2239
Белгородская область	2000	3162
Белгородская область	2001	4726
Белгородская область	2002	5904
Белгородская область	2003	7776
Белгородская область	2004	10401
Белгородская область	2005	13641
Белгородская область	2006	17409
Белгородская область	2007	22574,6
Брянская область	1995	901
Брянская область	1996	1931
Брянская область	1997	2479
Брянская область	1998	1999

С помощью обработчика "Свёртка столбцов" получилась новая таблица данных на основе, которой осуществляется дальнейшее построение аналитических моделей. Данный обработчик делает редактирование структуры используемых таблиц данных намного легче. Он в отличие от применяемой до этого группы обработчиков: "Настройка наборов данных" и "Слияние" уменьшает размер сценария и делает его обработку намного быстрее.

Практическая работа 3. 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.

Рассмотренный пример находиться на выделенном участке сценария проекта *"Демопример анализа данных.ded"*.

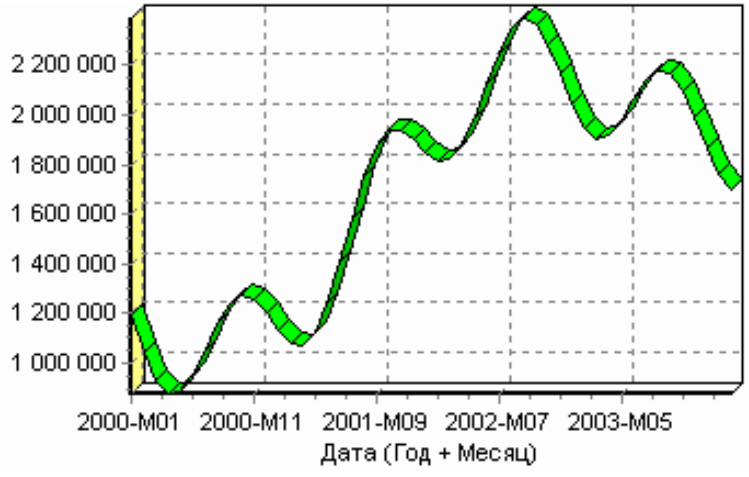


Обработка создает новые столбцы путем сдвига данных исходного столбца вниз и вверх (глубина погружения и горизонт прогноза).

1. 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.

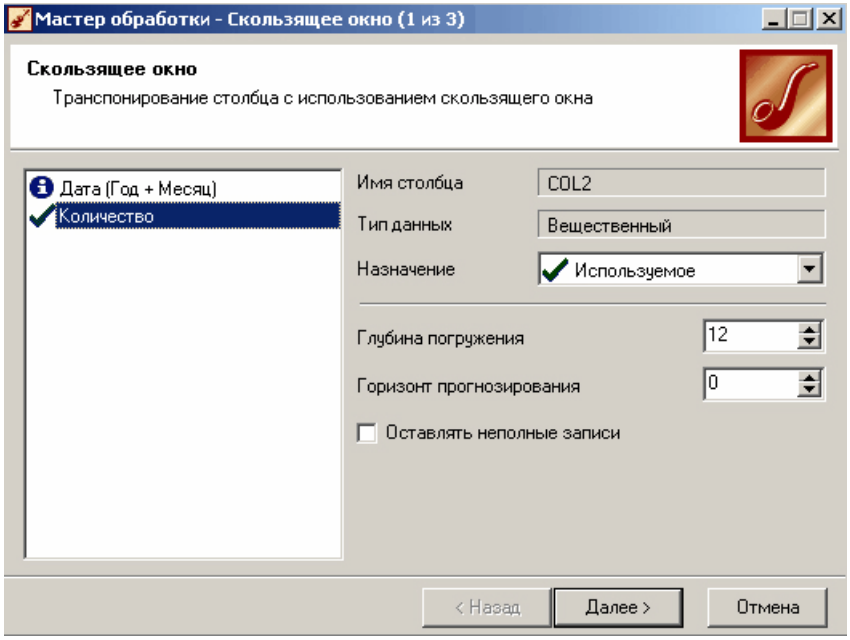
Исходные данные по продажам находятся в файле "Trade.txt". Выполним импорт данных из файла, не забыв указать в Мастере, чтобы в качестве разделителя дробной и целой части была точка, а не запятая. Выполнив удаление аномалий и сглаживание, получаем:



2. Приведение данных к скользящему окну

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

Можно использовать обработчик "Автокорреляция" и убедиться в наличии годовой сезонности. В связи с этим строить прогноз на месяц вперед можно, основываясь на данных за 1, 2, 11 и 12 месяцев 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.



3. Результат

Просмотреть полученные данные можно в виде таблицы.

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".

Таблица

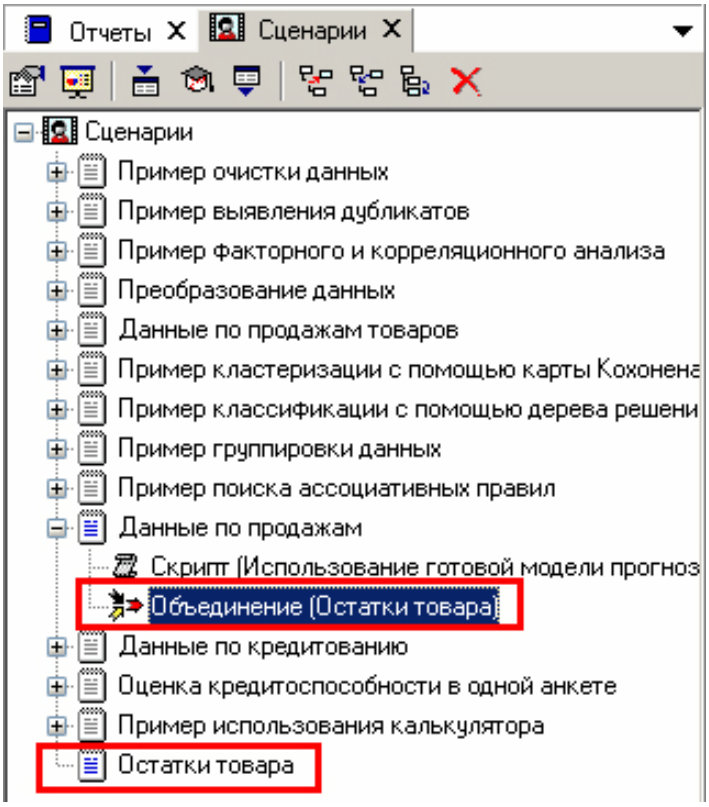
1 / 38

Дата (Год + Месяц)	Количество-12	Количество-11	Количество-10	Количество-9
2001-M01	1194372,66236191	1038792,30283372	919614,202815926	861513,586495192
2001-M02	1038792,30283372	919614,202815926	861513,586495192	873411,215927674
2001-M03	919614,202815926	861513,586495192	873411,215927674	946129,405485991
2001-M04	861513,586495192	873411,215927674	946129,405485991	1055162,67377219
2001-M05	873411,215927674	946129,405485991	1055162,67377219	1167771,49996288
2001-M06	946129,405485991	1055162,67377219	1167771,49996288	1252378,35295788
2001-M07	1055162,67377219	1167771,49996288	1252378,35295788	1287590,92307431
2001-M08	1167771,49996288	1252378,35295788	1287590,92307431	1268284,44787858
2001-M09	1252378,35295788	1287590,92307431	1268284,44787858	1207012,73191374
2001-M10	1287590,92307431	1268284,44787858	1207012,73191374	1130347,53164071
2001-M11	1268284,44787858	1207012,73191374	1130347,53164071	1071189,93073879
2001-M12	1207012,73191374	1130347,53164071	1071189,93073879	1059244,28215023
2002-M01	1130347,53164071	1071189,93073879	1059244,28215023	1112368,86802873
2002-M02	1071189,93073879	1059244,28215023	1112368,86802873	1231268,63186983
2002-M03	1059244,28215023	1112368,86802873	1231268,63186983	1399044,94789176
2002-M04	1112368,86802873	1231268,63186983	1399044,94789176	1585735,82560486

Слияние

Обработчик "Слияние" предназначен для объединения двух наборов данных по нескольким одинаковым полям. Обработчик применяется, например, для добавления в таблицу с данными о продажах данных по остаткам за те же месяцы. Операция производится над двумя таблицами: исходной и присоединяемой. К исходной таблице добавляются новые поля и/или строки, значения которых берутся из присоединяемой таблицы.

Демонстрацию данного обработчика можно наблюдать в выделенной части сценария проекта *"Демопример анализа данных.ded"*.



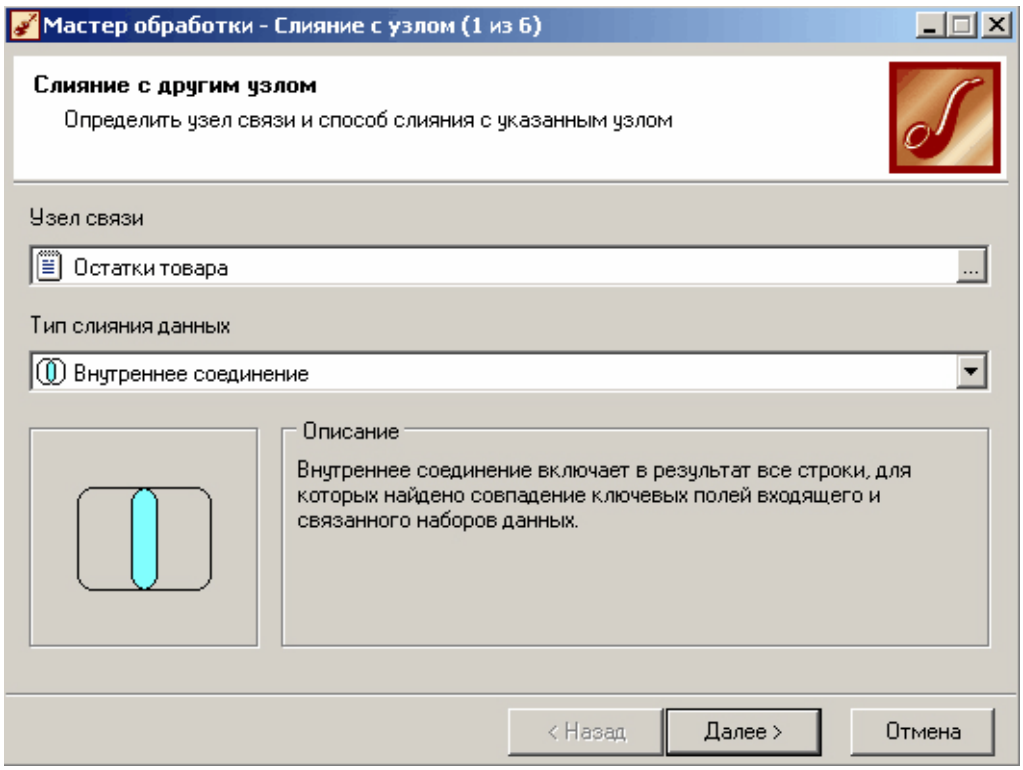
1. Исходные данные

Продемонстрируем механизм слияния, используя данные по продажам и остаткам (файлы "TradeSales.txt" и "TradeRest.txt" соответственно). Добавим к данным по продажам данные по остаткам.

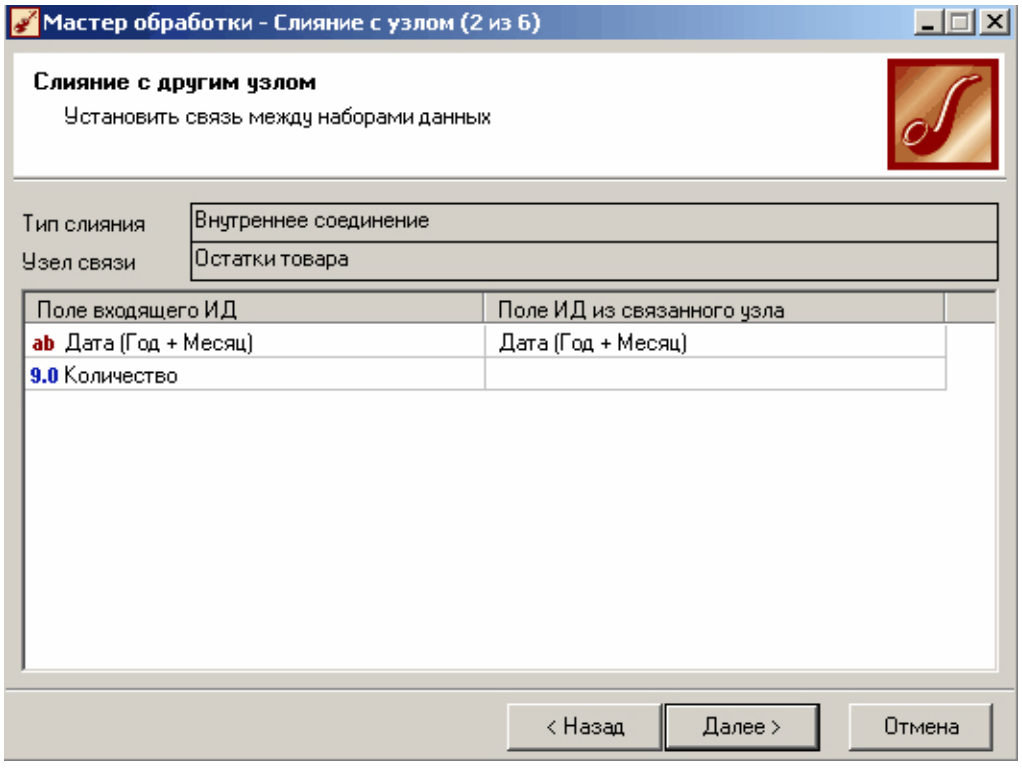
Для этого сначала импортируем данные из файла "TradeRes.txt", содержащего данные по остаткам. В сценарии появится новый узел — "Остатки товара". Далее импортируем данные по истории продаж из файла "TradeSales.txt". К сведениям по продажам добавим информацию об остатках, для этого запустим Мастер обработки и выберем обработчик "Слияние".

2. Выполнение слияния

В Мастере слияния сначала следует выбрать узел связи, с которым необходимо соединить данные. В данном случае это узел "Остатки товара". Также нужно указать тип слияния.



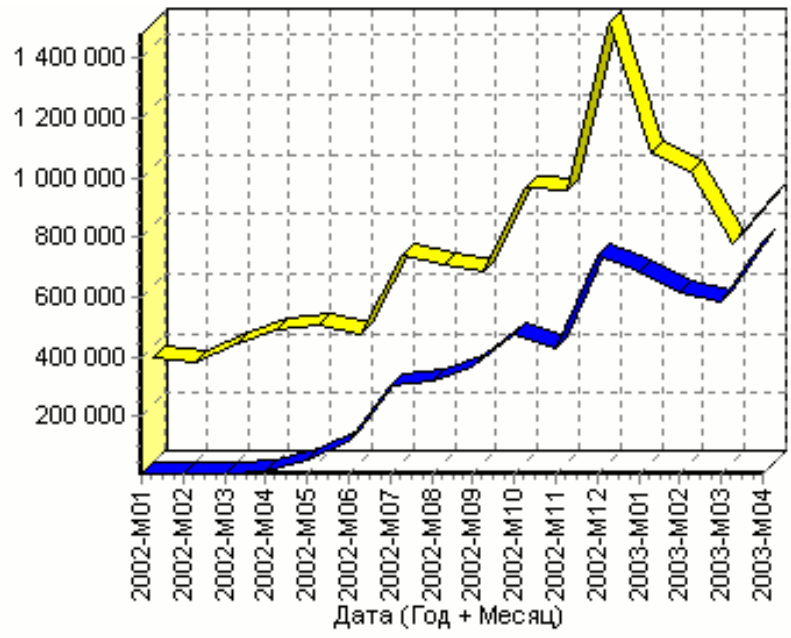
Далее укажем необходимые взаимосвязи между столбцами двух узлов сценария.



После указания параметров полей надлежит перейти на следующий шаг Мастера и запустить процесс слияния.

3. Результат

Полученные результаты, представленные в виде диаграммы, будут выглядеть следующим образом:



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

Вспомогательные механизмы

Платформа Deductor имеет ряд обработчиков, которые нельзя отнести к трансформации или очистке данных. Эти обработчики позволяют выполнить логическое или арифметическое действие, применить готовую модель и т.п.

Рассмотрим, как они работают на следующих примерах:

1. Применение калькулятора.
2. Применение скрипта.
3. Условное выполнение ветки сценария.

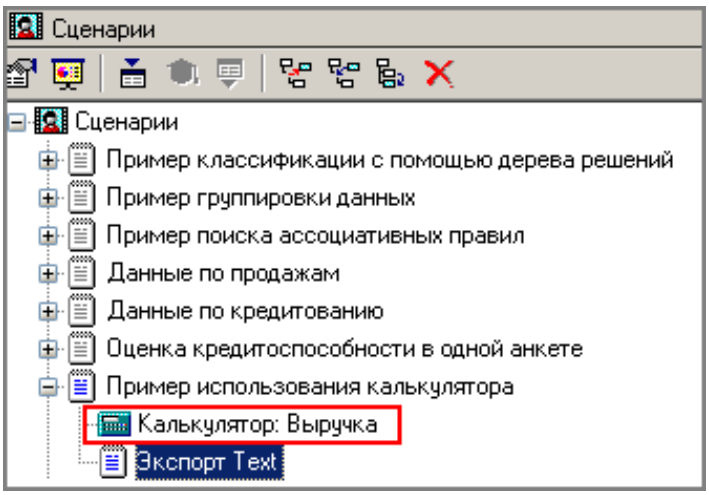
Применение калькулятора

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, либо получить на основе данных показатель и уже его использовать для дальнейшей обработки, либо 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. Там 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…).

Рассмотренный пример находится в проекте *"Демопример анализа данных.ded"*.



1. Исходные данные

В качестве исходных данных возьмем данные о продажах зубной пасты за 2 месяца 2007 года и вычислим выручку от ее продажи. Импортируем данные из файла "Trade2.txt". Данная таблица содержит информацию о дате продажи, наименовании зубной пасты и о объеме продаж и цене.

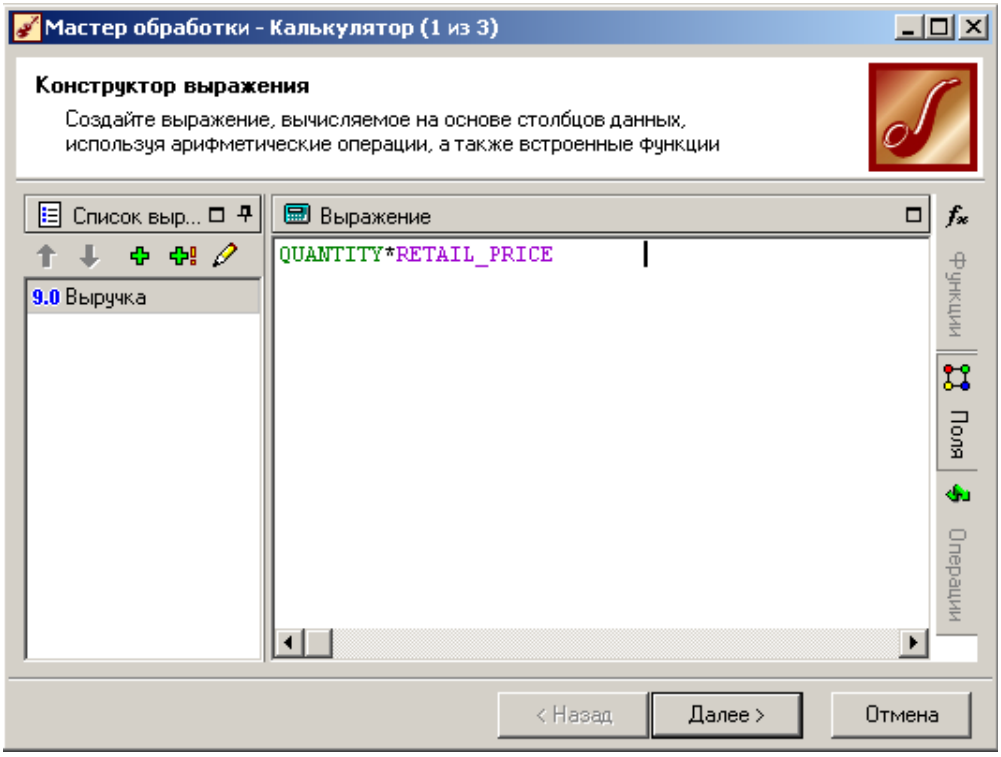
Таблица

1 / 163

Дата продажи	Наименование	Колличество	Цена продажи
10.01.2007	Зубная паста Силиция с морскими м	4	44,49
04.01.2007	Зубная паста Жемчуг белоснежный	5	9,78
05.01.2007	Зубная паста Мятная 100г	3	7,82
05.01.2007	Зубная паста Голубой жемчуг Экстра	3	8,7
05.01.2007	Зубная паста Голубой жемчуг Проти	3	8,7
05.01.2007	Зубная паста ФитоДент крапива и к	3	11,2
09.01.2007	Зубная паста Мятная 100г	3	7,82
09.01.2007	Зубная паста Мятная 100г	2	7,82
09.01.2007	Зубная паста Семейная 100г	5	7,82
09.01.2007	Зубная паста Семейная 100г	2	7,82
09.01.2007	Зубная паста Хвойная 100г	2	8,09
09.01.2007	Зубная паста Хвойная 100г	5	8,09
09.01.2007	Зубная паста Голубой жемчуг Отбели	10	8,7
09.01.2007	Зубная паста Голубой жемчуг Экстра	3	8,7
09.01.2007	Зубная паста Голубой жемчуг Проти	10	8,7
09.01.2007	Зубная паста Голубой жемчуг Проти	3	8,7
09.01.2007	Зубная паста Голубой жемчуг Свеже	10	8,7
09.01.2007	Зубная паста Голубой жемчуг Свеже	3	8,7
09.01.2007	Зубная паста Голубой жемчуг Свеже	3	8,7
09.01.2007	Зубная паста ФитоДент крапива и к	5	11,2
10.01.2007	Зубная паста Силиция с морскими м	4	44,49
10.01.2007	Зубная паста Силиция Special с морс	1	44,49

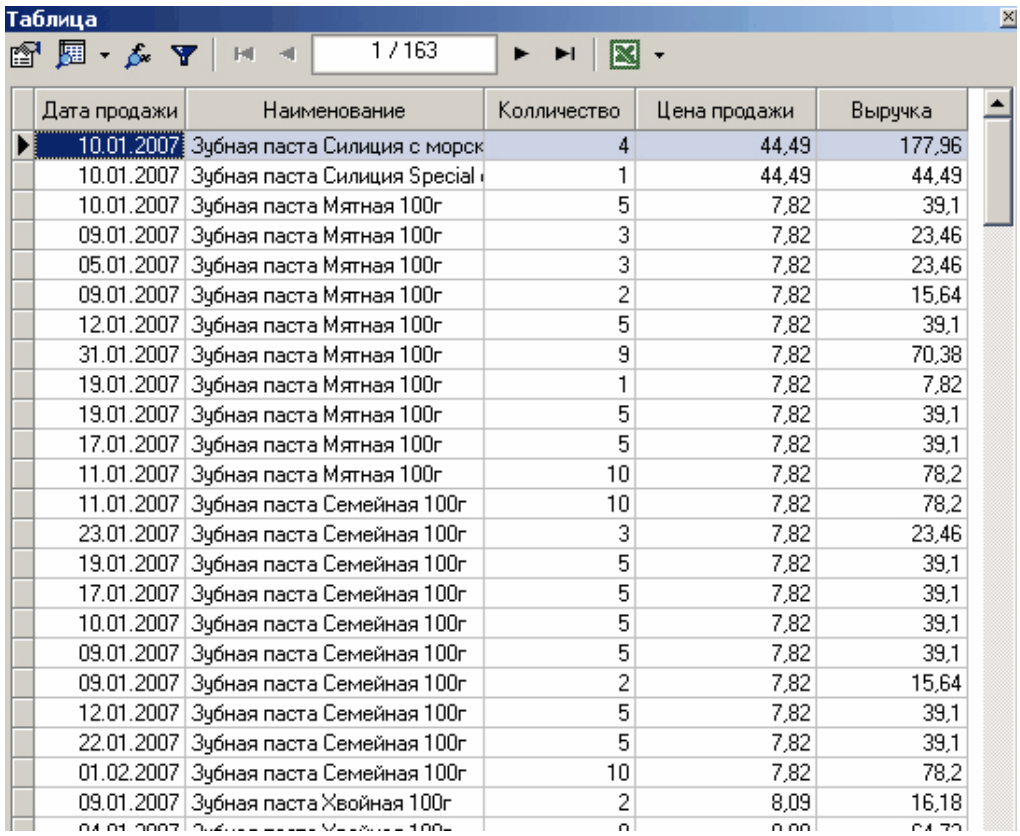
2. Расчет выручки с помощью калькулятора

Для расчета выручки воспользуемся Мастером обработки. В списке обработчиков выбираем калькулятор. Для нахождения суммы с помощью калькулятора запишем формулу для вычисления искомого выражения (в данном примере это произведение объема проданного товара на цену продажи).



3. Результат

На следующем шаге должен быть указан способ отображения. Выбираем для представления данных таблицу.



Дата продажи	Наименование	Колличество	Цена продажи	Выручка
10.01.2007	Зубная паста Силиция с морск	4	44,49	177,96
10.01.2007	Зубная паста Силиция Special	1	44,49	44,49
10.01.2007	Зубная паста Мятная 100г	5	7,82	39,1
09.01.2007	Зубная паста Мятная 100г	3	7,82	23,46
05.01.2007	Зубная паста Мятная 100г	3	7,82	23,46
09.01.2007	Зубная паста Мятная 100г	2	7,82	15,64
12.01.2007	Зубная паста Мятная 100г	5	7,82	39,1
31.01.2007	Зубная паста Мятная 100г	9	7,82	70,38
19.01.2007	Зубная паста Мятная 100г	1	7,82	7,82
19.01.2007	Зубная паста Мятная 100г	5	7,82	39,1
17.01.2007	Зубная паста Мятная 100г	5	7,82	39,1
11.01.2007	Зубная паста Мятная 100г	10	7,82	78,2
11.01.2007	Зубная паста Семейная 100г	10	7,82	78,2
23.01.2007	Зубная паста Семейная 100г	3	7,82	23,46
19.01.2007	Зубная паста Семейная 100г	5	7,82	39,1
17.01.2007	Зубная паста Семейная 100г	5	7,82	39,1
10.01.2007	Зубная паста Семейная 100г	5	7,82	39,1
09.01.2007	Зубная паста Семейная 100г	5	7,82	39,1
09.01.2007	Зубная паста Семейная 100г	2	7,82	15,64
12.01.2007	Зубная паста Семейная 100г	5	7,82	39,1
22.01.2007	Зубная паста Семейная 100г	5	7,82	39,1
01.02.2007	Зубная паста Семейная 100г	10	7,82	78,2
09.01.2007	Зубная паста Хвойная 100г	2	8,09	16,18

Как видно из рисунка, в таблице появился новый столбец с рассчитанной выручкой.

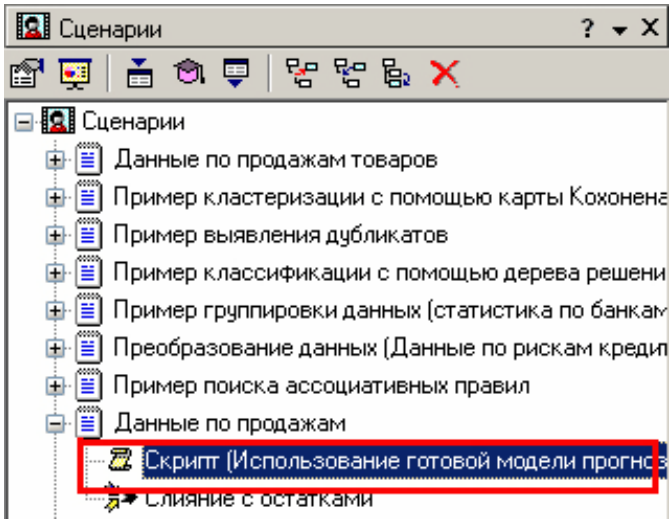
Применение скрипта

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. Скрипт является готовым фрагментом сценария, поэтому все составляющие данного фрагмента не могут быть изменены. Тем не менее, на скрипте отражаются все изменения, вносимые в ветку, на которую он ссылается, т. е. при переобучении или перенастройке узлов этой ветки все сделанные изменения будут учтены в работе 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.

Рассмотрим фрагмент проекта *"Демопример анализа данных.ded"*, где применяется обработчик "Скрипт".



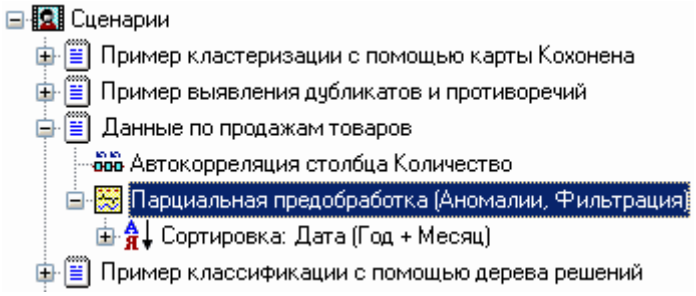
1. Исходные данные

Рассмотрим механизм использования скрипта на примере данных файла "TradeSales.txt". В нем находится информация по продажам некоторой группы товаров. Пусть необходимо сделать прогноз продаж на три месяца вперед. Поскольку данный пример описывается в примере анализа данных (прогнозирование с помощью нейронных сети), 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".

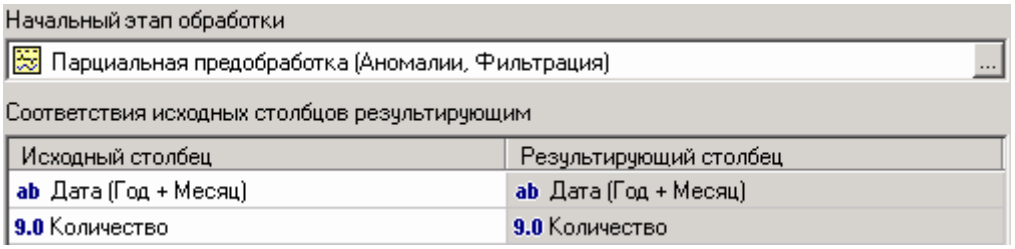
2. Указание цепочки выполняемых обработок

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

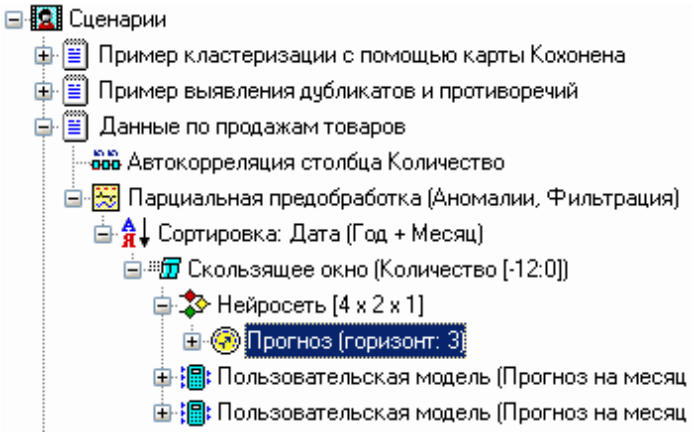
в правой части этой строки, после чего на экране появится окно "Выбор узла". В этом окне показано все дерево сценария. Выберем в качестве начального узел "Парциальная предобработка".



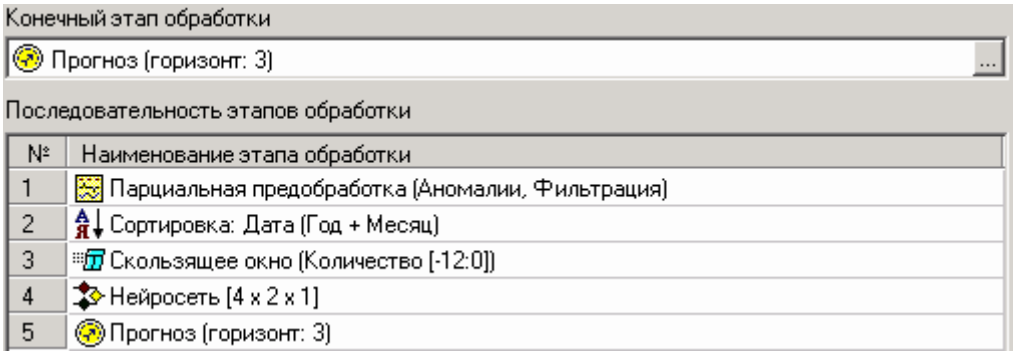
После выбора начального узла следует задать соответствия столбцов исходного набора данных полям выбранного узла. В нижней части экрана находится таблица со списком полей исходного набора в левом столбце и полей выбранного узла — в правом. Для каждого поля начального узла надо задать поле-источник исходного 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:



На следующем шаге Мастера аналогичным образом выбирается конечный узел обработки:



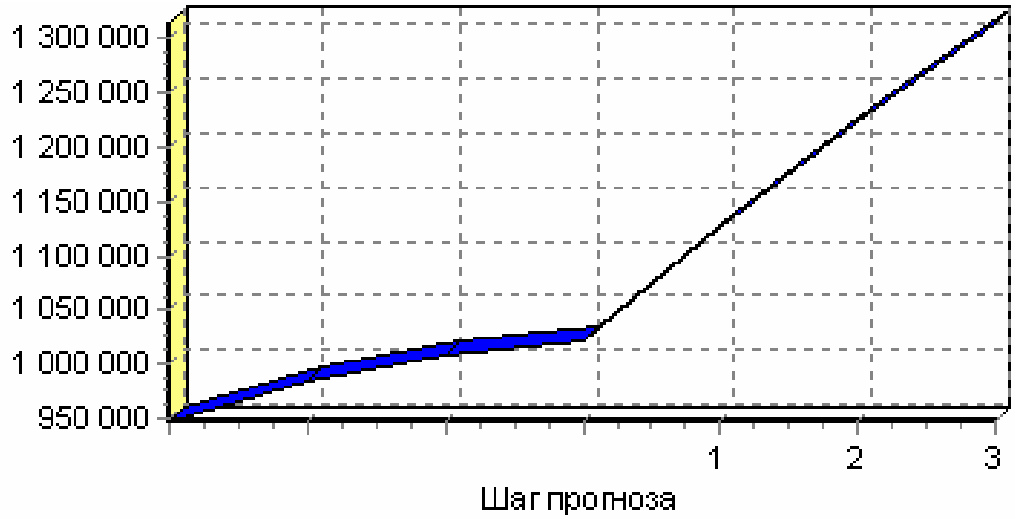
После выбора конечного узла в нижней части окна будет показан список узлов, входящих в скрипт. При выполнении скрипта последовательно будут выполнены все узлы сценария из этого списка:



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.

3. Результат

Вот, например, диаграмма с прогнозом объема продаж товара за некоторый период времени, полученного с использованием модели прогноза, построенной для другого товара:



4. Выводы

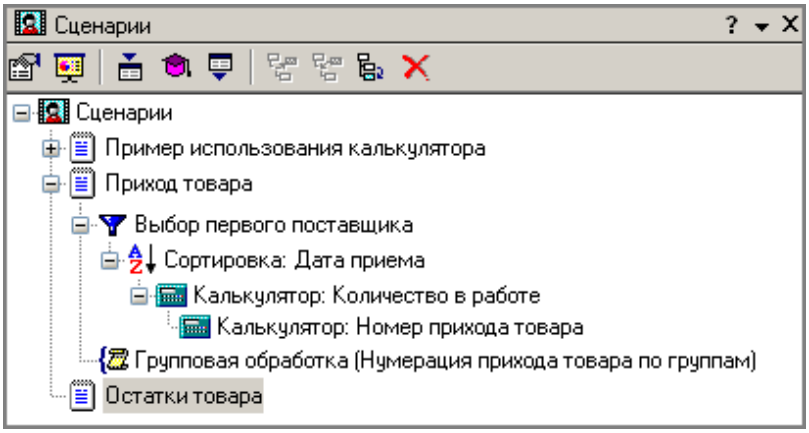
Данный пример показал, как применять одни и те же действия к различным данным, что позволяет намного быстрее создавать аналитические решения. Имея заранее подготовленные цепочки действий, например, для одной товарной группы, можно несколькими щелчками мыши провести очистку, сглаживание, прогноз и т.д.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

Применение групповой обработки

Сущность метода групповой обработки данных заключается в следующем. Все данные автоматически разбиваются на классы в соответствии с заданным параметром, признаком. Для каждой выделенной группы осуществляется выполнение однородной групповой ветки сценария. Построение веток сценария для каждой группы происходит в автоматическом режиме, по алгоритму, сформированному на первой выделенной группе (обучающей), на основе которого идет дальнейшая обработка всего набора данных. В конце работы обработчика "Групповая обработка данных", получается общая таблица набора данных, собранная по группам, для которых выполнены заданные аналитические операции. Перенастроить данный алгоритм работы сценария, можно только с помощью внесения изменения в ветку, на которую ссылается данный обработчик.

Рассмотрим простой пример использования групповой обработки данных. У нас имеются данные по приходу товара от различных поставщиков. Нам необходимо пронумеровать приход товара по каждому поставщику. Осуществить данную задачу можно на обработчике "Фильтр" и "Слияние", но это очень сильно увеличит размер сценария и при внесении изменения в состав поставщиков потребует его перестроения, обработчик "Групповая обработка данных" позволяет избежать данных проблем.

Рассмотрим фрагмент проекта *"Демопример анализа данных.ded"*, где применяется обработчик "Групповая обработка данных".



1. Исходные данные

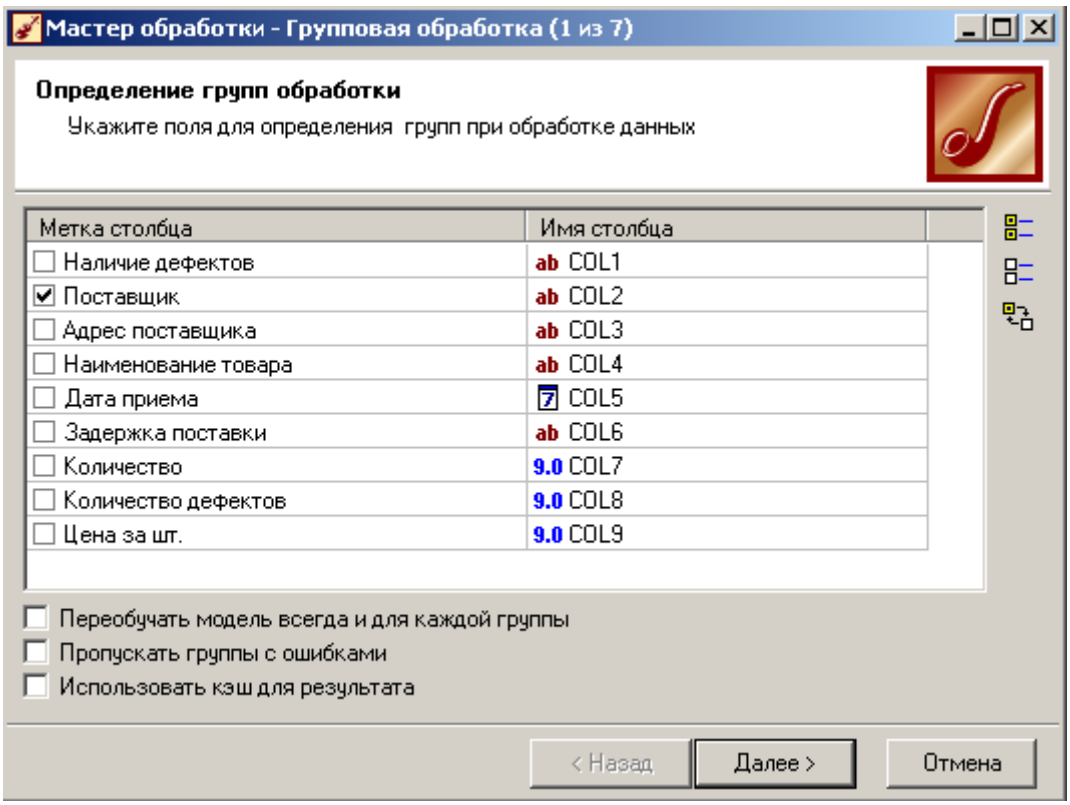
Продемонстрируем применение метода групповой обработки данных на примере данных файла "Goods". В нем находится информация по приходу некоторых групп товаров от поставщиков.

2. Указание цепочки выполняемых обработок

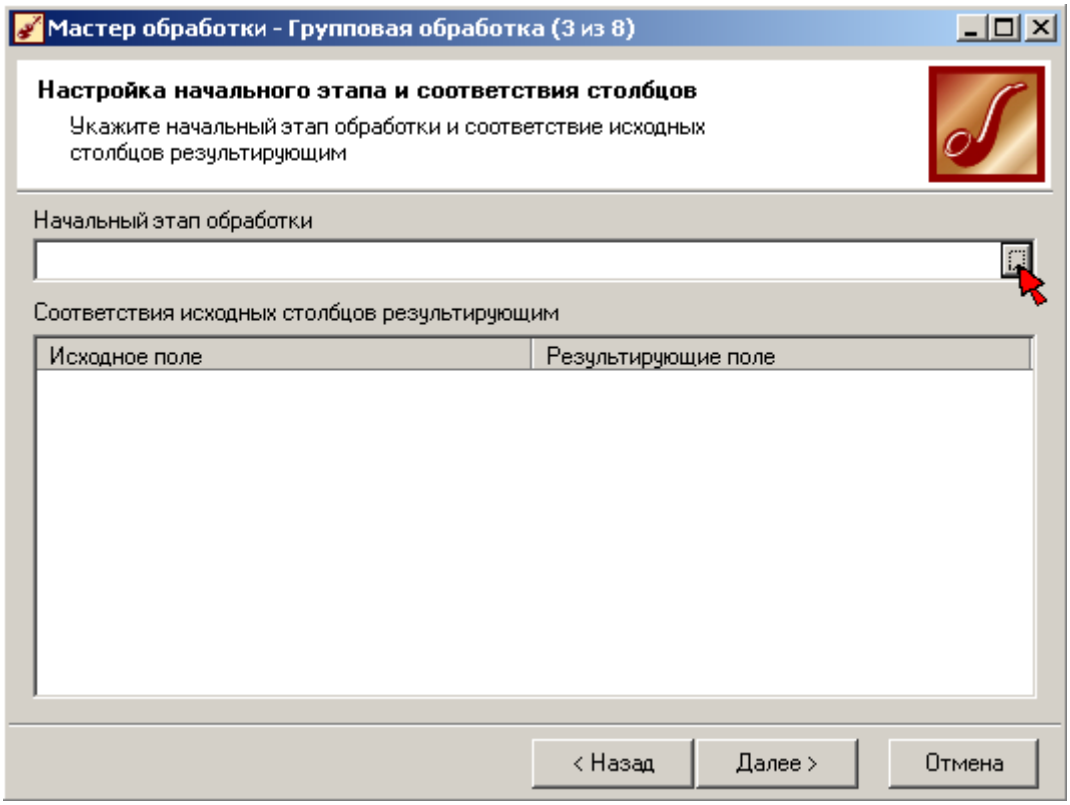
На первом шаге выполним импорт данных из текстового файла. Дальнейшие действия можно разделить на два этапа: первый — формирование обучающего алгоритма, по которому будет осуществляться групповая обработка данных, второй — настройка обработчика и его запуск. В начале необходимо определить поле на основе которого будет происходить группировка, в нашем случае это будет — Поставщик. Для этого применим обработчик "Фильтр" и выберем в нем одного из поставщиков. Данный обработчик можно пропустить и начать сразу построение сценария, который будет выполняться для каждой группы (фильтр встроен в групповой обработчик данных). Но это не очень хорошая идея, так как усложняется понимание выполняемых действий, осуществляемых над группой и нет возможности проверки построения сценария. Удобнее для аналитика использовать обработчик фильтр, он позволяет сократить время на отработку сценария и сразу же проверить правильность его работы.

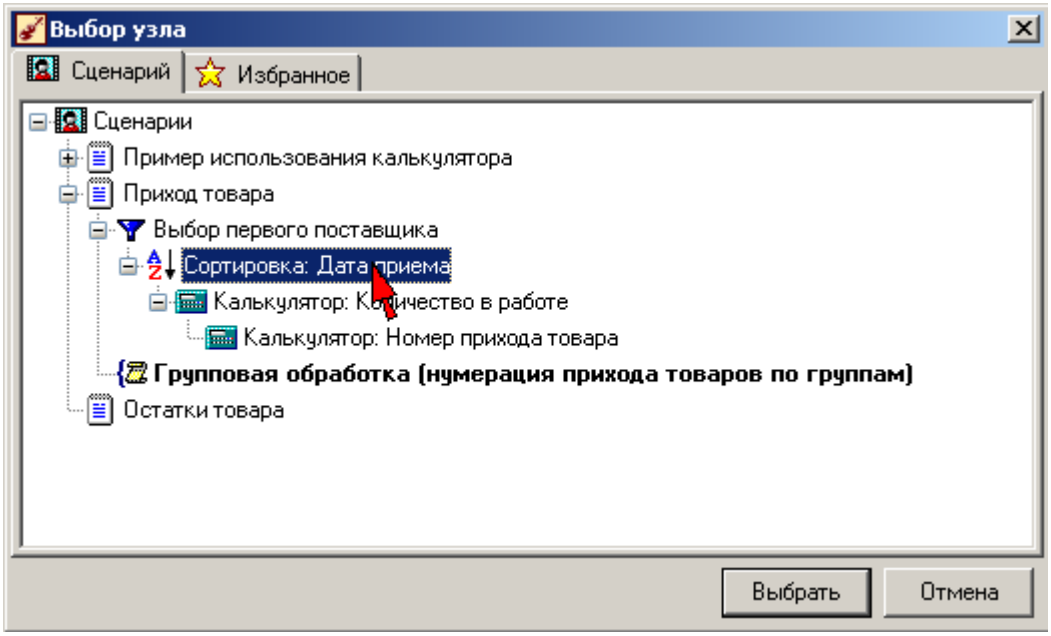
На следующем шаге осуществляется формирование сценария, который будет выполняться для каждой группы в отдельности. Пронумеруем все приходы товара от данного поставщика для этого сначала упорядочим их в порядке возрастания, а затем с помощью калькулятора выполним нумерацию
по порядку. Вставим дополнительные вычисления, которые могут понадобиться аналитику в дальнейшем для построения отчетов, такие как сумма покупки, количество товаров и произведем настройку полей данных. На этом формирование обучающего алгоритма выполнения необходимых операций над группой закончено.

Следующим этапом является запуск и настройка обработчика, для этого выберем в качестве узла построения групповой обработки узел, на котором решалось целесообразность его применения и запустим его. В нашем случае это узел импорта данных. После запуска обработчика необходимо выбрать поле по которому будет осуществляться группировка набора данных, выбираем поле поставщик и нажимаем кнопку далее.

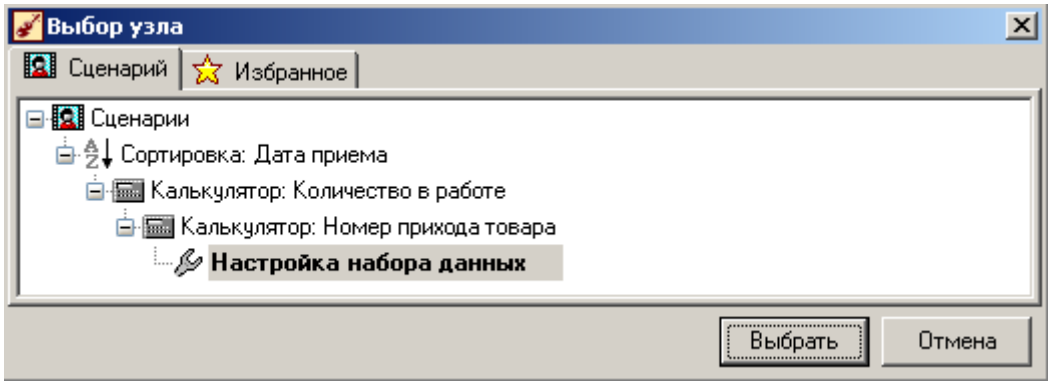


На следующем шаге следует выбрать узел сценария, с которого начнется исполнение групповой обработки данных. Для этого необходимо в строке "Начальный этап обработки" выбрать необходимый узел из дерева узлов.





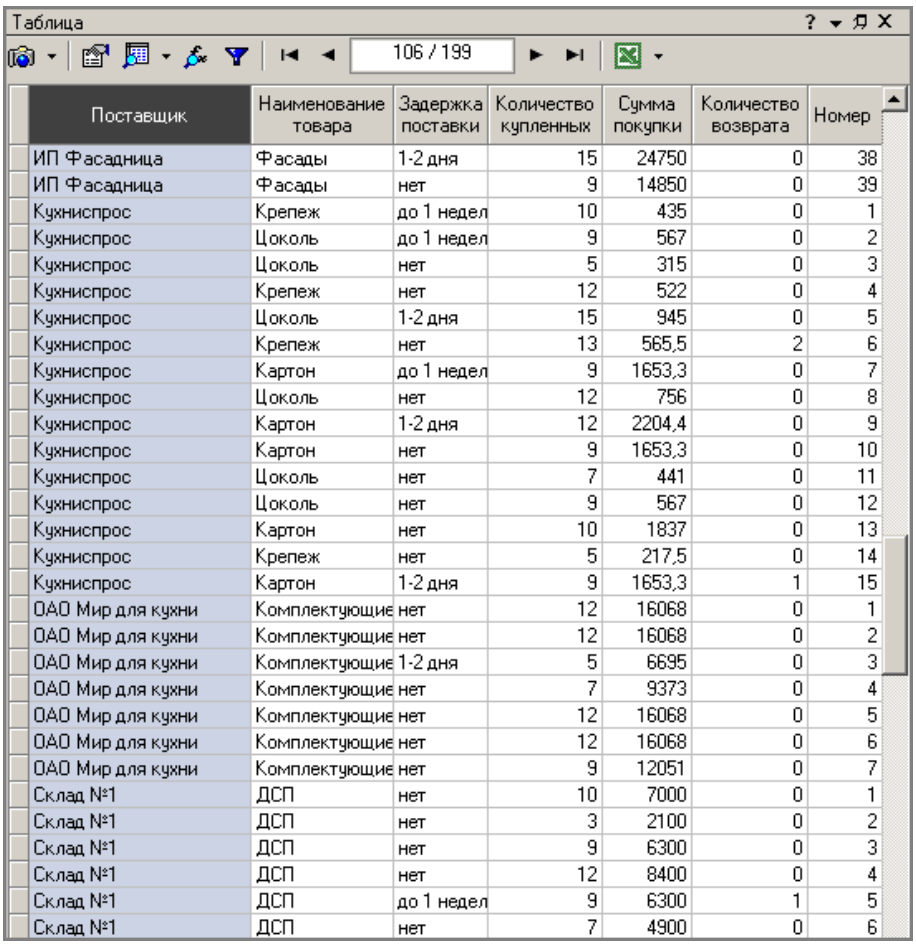
После выбора начального узла следует либо задать соответствия столбцов исходного набора данных полям выбранного узла, либо отредактировать автоматический настройки при необходимости. Оставим соответствие, полученное автоматически.
На следующем шаге Мастера аналогичным образом выбирается конечный узел обработки:



Запускаем процесс на обработку в следующем окне с помощью нажатия кнопки Пуск. После окончания выполнения процесса необходимо выбрать визуализаторы, выберем "Таблицу".

3. Результат

Полученные результаты, представленные в виде таблицы, будут выглядеть следующим образом:



Поставщик	Наименование товара	Задержка поставки	Количество купленных	Сумма покупки	Количество возврата	Номер
ИП Фасадница	Фасады	1-2 дня	15	24750	0	38
ИП Фасадница	Фасады	нет	9	14850	0	39
Кухниспрос	Крепеж	до 1 недел	10	435	0	1
Кухниспрос	Цоколь	до 1 недел	9	567	0	2
Кухниспрос	Цоколь	нет	5	315	0	3
Кухниспрос	Крепеж	нет	12	522	0	4
Кухниспрос	Цоколь	1-2 дня	15	945	0	5
Кухниспрос	Крепеж	нет	13	565,5	2	6
Кухниспрос	Картон	до 1 недел	9	1653,3	0	7
Кухниспрос	Цоколь	нет	12	756	0	8
Кухниспрос	Картон	1-2 дня	12	2204,4	0	9
Кухниспрос	Картон	нет	9	1653,3	0	10
Кухниспрос	Цоколь	нет	7	441	0	11
Кухниспрос	Цоколь	нет	9	567	0	12
Кухниспрос	Картон	нет	10	1837	0	13
Кухниспрос	Крепеж	нет	5	217,5	0	14
Кухниспрос	Картон	1-2 дня	9	1653,3	1	15
ОАО Мир для кухни	Комплектующие	нет	12	16068	0	1
ОАО Мир для кухни	Комплектующие	нет	12	16068	0	2
ОАО Мир для кухни	Комплектующие	1-2 дня	5	6695	0	3
ОАО Мир для кухни	Комплектующие	нет	7	9373	0	4
ОАО Мир для кухни	Комплектующие	нет	12	16068	0	5
ОАО Мир для кухни	Комплектующие	нет	12	16068	0	6
ОАО Мир для кухни	Комплектующие	нет	9	12051	0	7
Склад №1	ДСП	нет	10	7000	0	1
Склад №1	ДСП	нет	3	2100	0	2
Склад №1	ДСП	нет	9	6300	0	3
Склад №1	ДСП	нет	12	8400	0	4
Склад №1	ДСП	до 1 недел	9	6300	1	5
Склад №1	ДСП	нет	7	4900	0	6

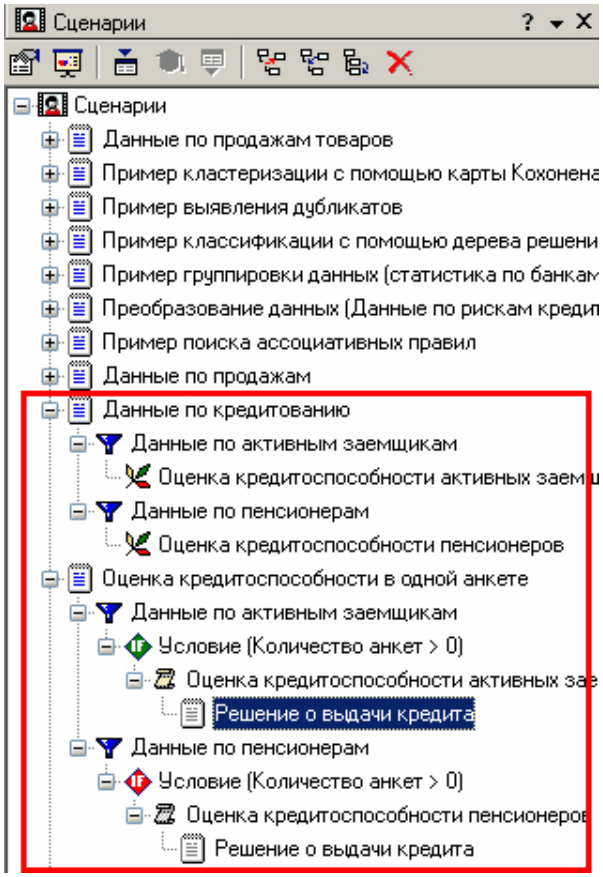
4. Выводы

Групповая обработка позволяет выполнять одинаковые действия над группами данных, что существенно сокращает время выполнения операций, увеличивает быстроту их редактирования и исходя из этого повышает правильность. Групповая обработка применяется как для подготовки данных для анализа, так и непосредственно при применении того или иного аналитического метода. Большое применение получила в прогнозирование данных по группам.

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.

Рассмотрим пример использования условия в проекте *"Демопример анализа данных.ded"*.



1. 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, какой это 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". Сценарий построения моделей кредитоспособности представляет собой импорт кредитной истории из текстового файла, далее фильтрацией набор разделяется на 2 части: активные заемщики и заемщики – пенсионеры. Затем 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.

2. 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 выбранному полю. 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.

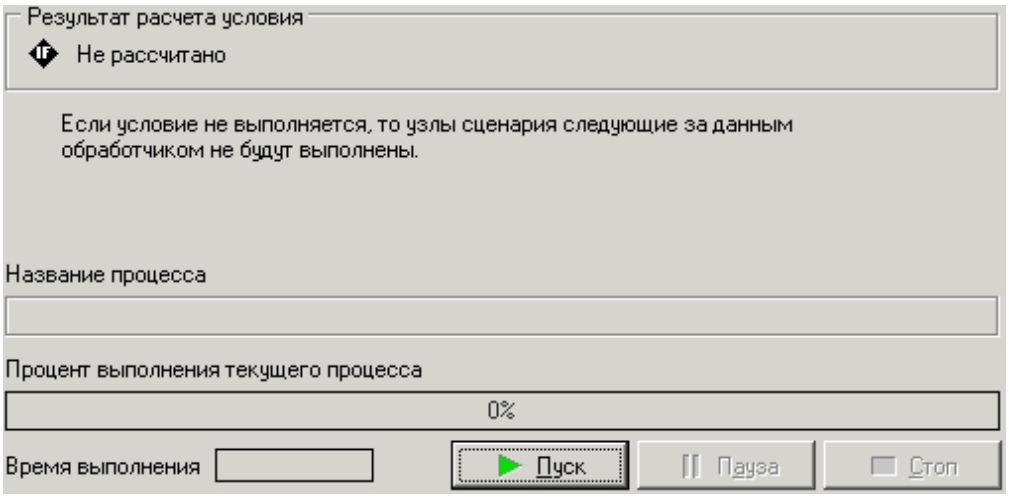
В данном случае необходимо установить поле "*", функцию агрегации "Количество", указать условие выполнения “> 0”:

Операция	Имя поля	Функция	Условие	Значение
	*	|s| Количество	>	0

Это означает, что в выборке должна быть хотя бы одна запись.

3. Расчет условия

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



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.

4. 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 сложной обработки исходных данных, что достаточно часто встречается на практике.

Практическая работа 3. Примеры анализа данных

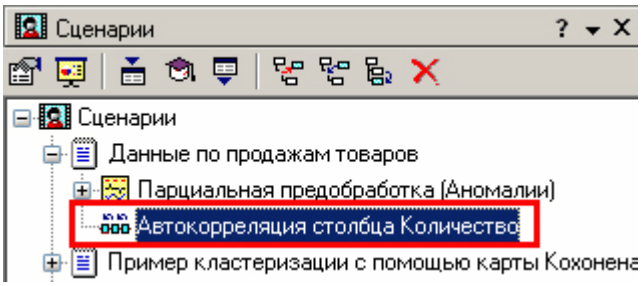
Основное направление программы Deductor Studo — анализ данных. Предыдущие примеры в основном касались только подготовки данных для последующего анализа. Программа предоставляет множество механизмов построения моделей. Некоторые из них будут продемонстрированы на примерах:

1. Классификация с помощью деревьев решений.
2. Прогнозирование с помощью линейной регрессии.
3. Кластеризация с помощью самоорганизующейся карты Кохонена.
4. Поиск ассоциативных правил.
5. Прогнозирование с помощью построения пользовательских моделей.
6. Пример расчета автокорреляции столбцов.
7. Пример прогноза временного ряда.

Пример расчета автокорреляции столбцов

Важным фактором для анализа временного ряда и прогноза является определение сезонности. В Deductor Studio инструментом, предназначенным для изучения сезонности, 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. Нахождение линейной автокорреляционной зависимости применяется для определения периодичности (сезонности) при обработке временных рядов.

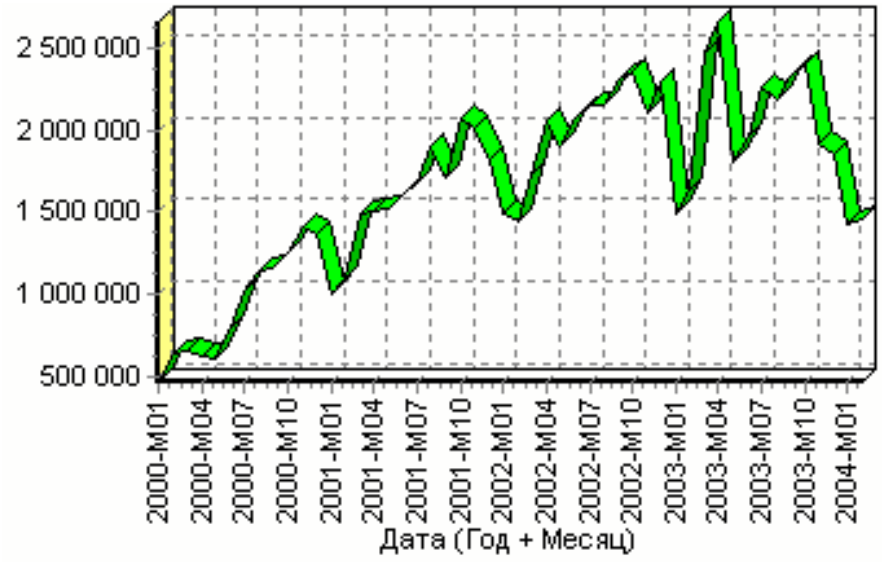
Рассмотрим часть проекта *"Демопример анализа данных.ded"*.



1. 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.

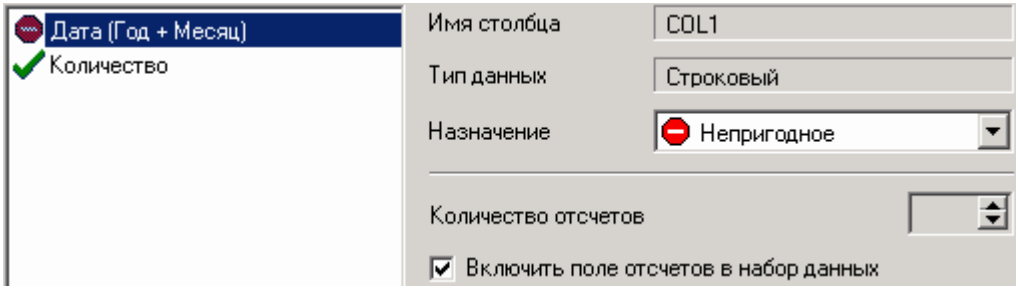
Импортируем данные из текстового файла. Обратите внимание на то, что в файле данные о количестве находятся не в стандартном формате: разделитель дробной и целой части числа не запятая, а точка, поэтому необходимо внести соответствующие изменения в настройки по умолчанию параметров импорта. Выберем в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



2. 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 искаться зависимость от месяца назад, двух, ..., пятнадцати месяцев назад). Количество отсчетов ставится больше 12 (хотя мы ищем наличие именно готовой сезонности, т.е. 12 месяцев) для того, чтобы убедиться, что на 12 месяцев приходится пик коэффициента автокорреляции, а далее следует его спад.

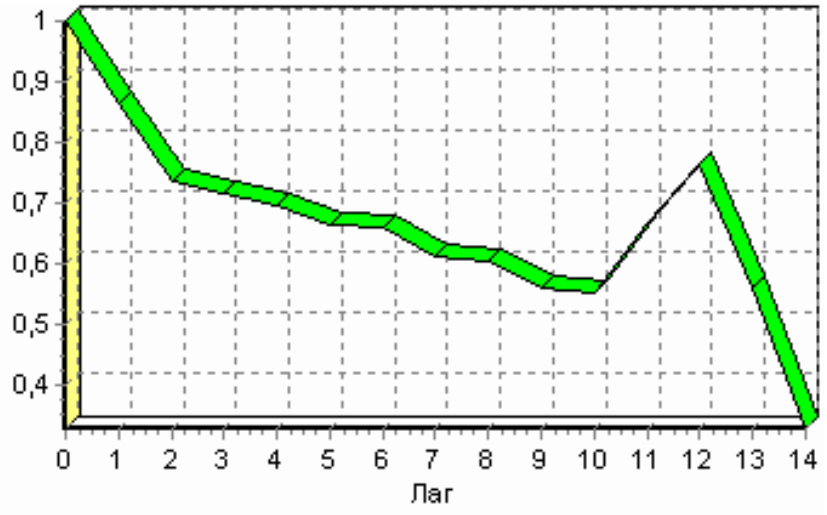
Также должен стоять флажок "Включить поле отсчетов набор данных". Он необходим для более удобной интерпретации автокорреляционного анализа.



Перейдем на следующий шаг Мастера и запустим процесс обработки.

3. Результат

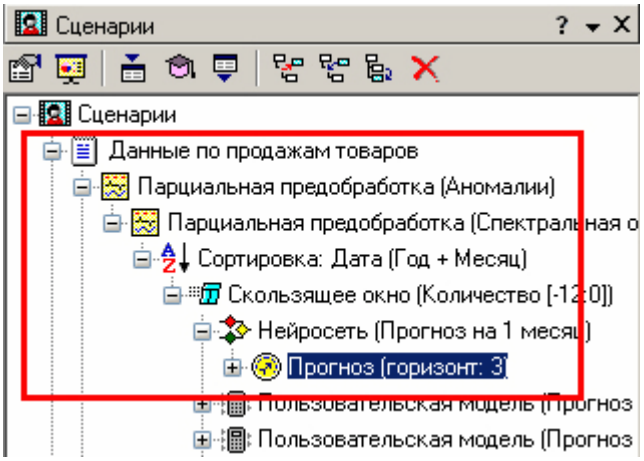
По окончанию результаты удобно анализировать как в виде таблицы, так и в виде диаграммы. После обработки были получены два столбца – "Лаг" (благодаря установленному флажку в Мастере) и "Количество" - результат автокорреляции.



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.

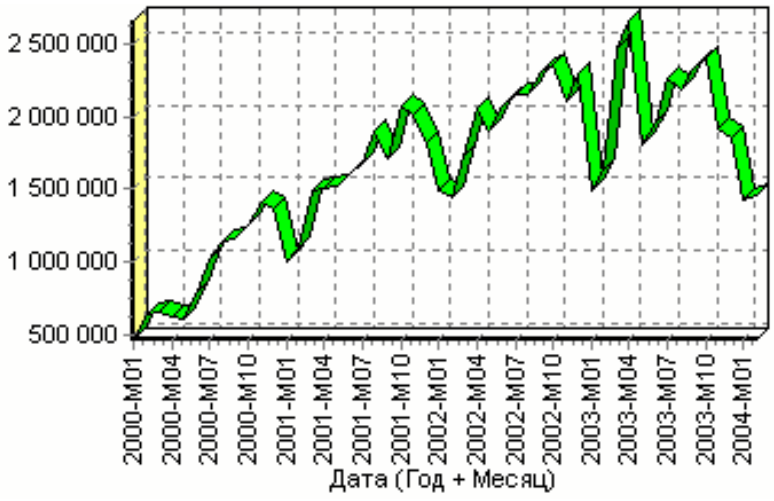
Прогнозирование с помощью нейронной сети

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, что 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. Данный пример анализа можно посмотреть в проекте *"Демопример анализа данных.ded"*.



У аналитика имеются данные о помесячном количестве проданного товара за несколько лет. Ему необходимо, основываясь на этих данных, определить, какое Количество товара будет продано через месяц и через два.

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



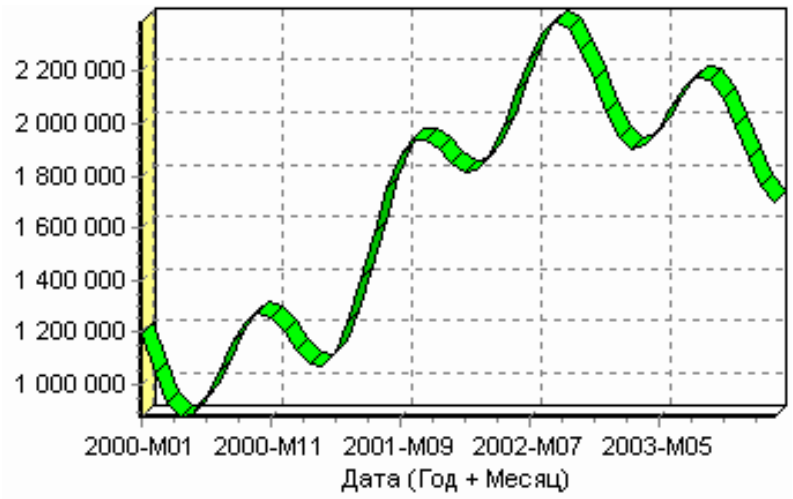
1. Удаление аномалий и сглаживание

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 просмотрим полученный результат на диаграмме.

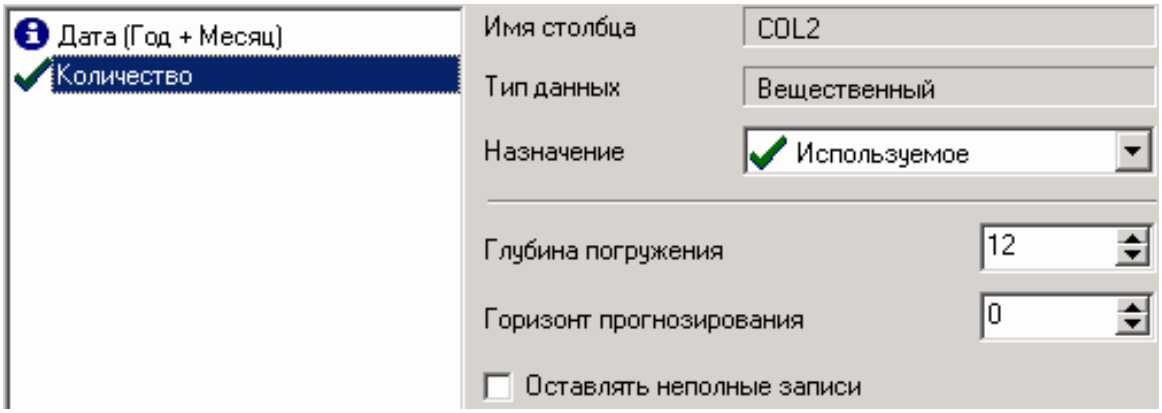


Видно, что данные сгладились, аномалии и шумы исчезли. Также видна тенденция.
Теперь перед аналитиком встает вопрос, а как, собственно, прогнозировать временной ряд. Во всех предыдущих примерах мы сталкивались с ситуацией, когда есть входные столбцы - факторы и есть выходные столбцы – результат. В данном случае столбец один. Строить прогноз на будущее будем, основываясь на данных прошлых периодов, т. е. предполагая, что количество продаж на следующий месяц зависит от количества продаж за предыдущие месяцы. Это значит, что 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.

2. Скользящее окно 12 месяцев назад

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

Проведя авторегрессионый анализ, обнаружили наличие годовой сезонности. В связи с этим было решено строить прогноз на месяц вперед, основываясь на данных за 1, 2, 11 и 12 месяцев назад. Поэтому требуется выбрать глубину погружения 12, назначив поле "Количество" используемым. Тогда данные трансформируются к скользящему окну так, что аналитику будут доступны все нужные факторы для построения прогноза.

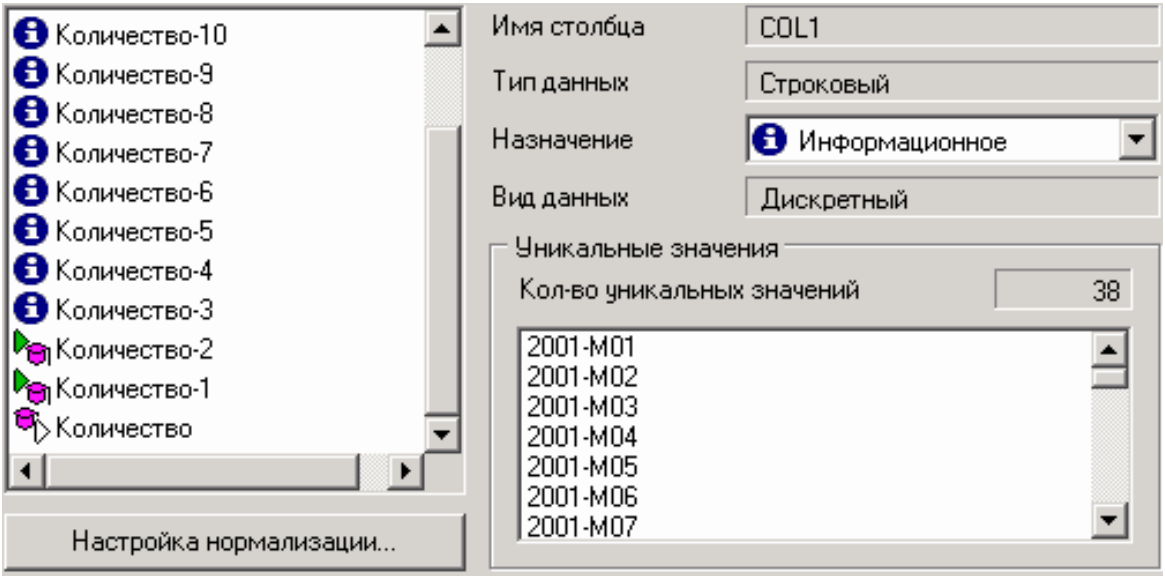


Теперь в качестве входных факторов можно использовать "Количество - 12", "Количество - 11" - данные по количеству 12 и 11 месяцев назад (относительно прогнозируемого месяца), а также "Количество - 2" и "Количество - 1" - данные за 2 предыдущих месяца. В качестве выходного поля укажем столбец "Количество".

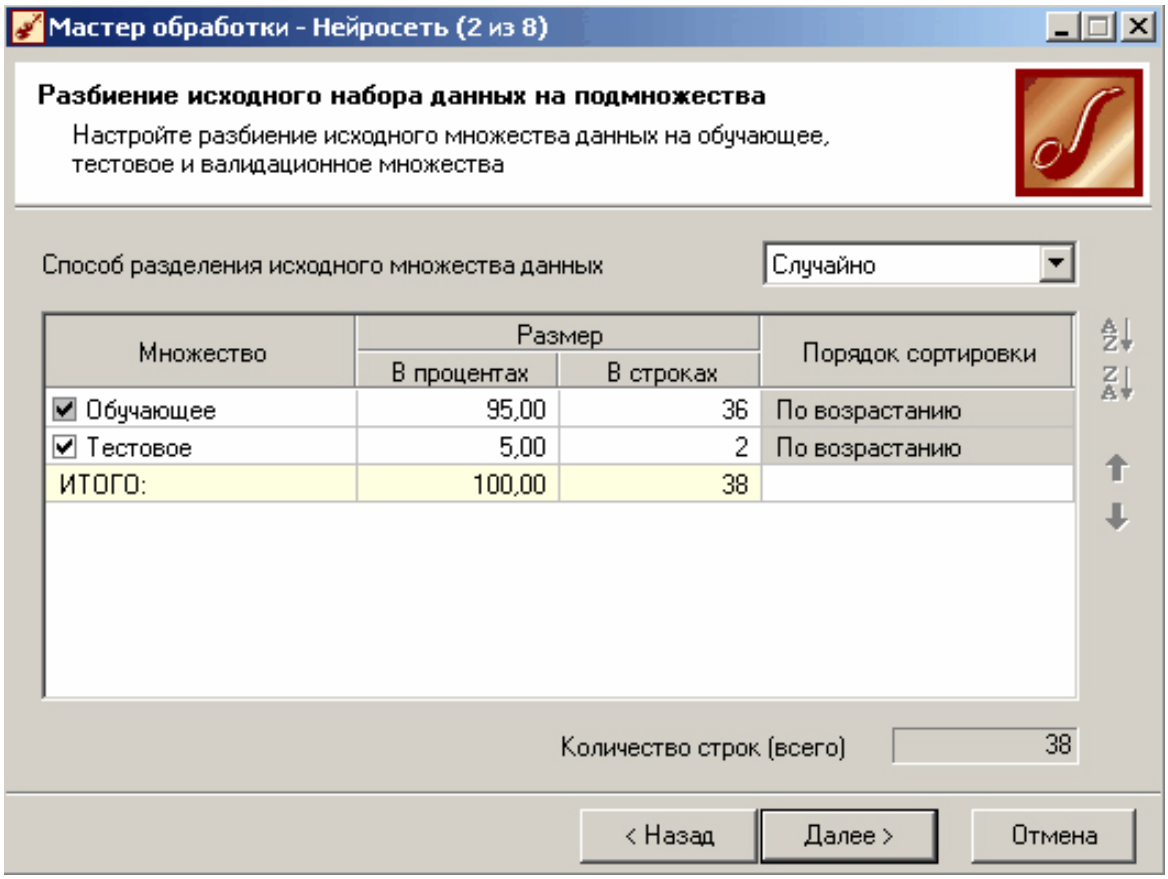
-

3. Обучение нейросети (прогноз на 1 месяц вперед)

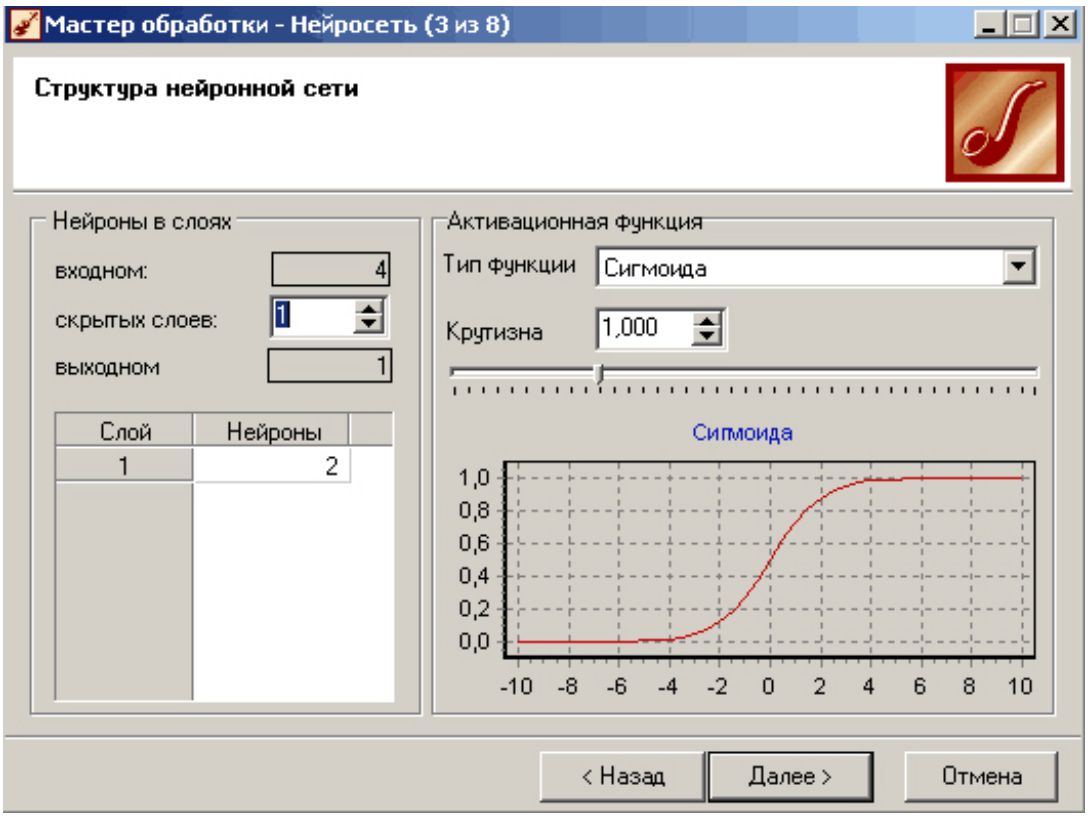
Перейдем непосредственно к самому построению модели прогноза. Откроем Мастер обработки и выберем в нем нейронную сеть. На втором шаге Мастера согласно с принятым ранее решением установим в качестве входных поля "Количество – 12", "Количество – 11", "Количество – 2" и "Количество – 1", а в качестве выходного — "Количество". Остальные поля сделаем информационными.



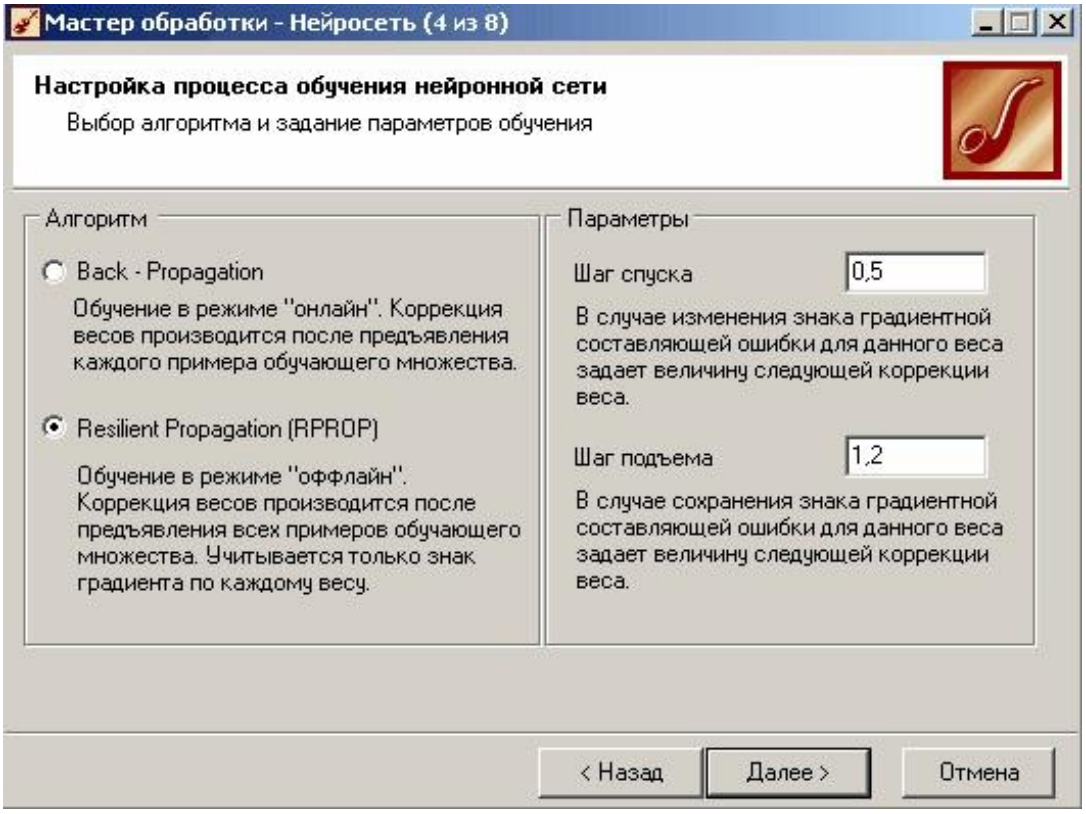
На следующем шаге укажем разбиение тестового и обучающего множеств.



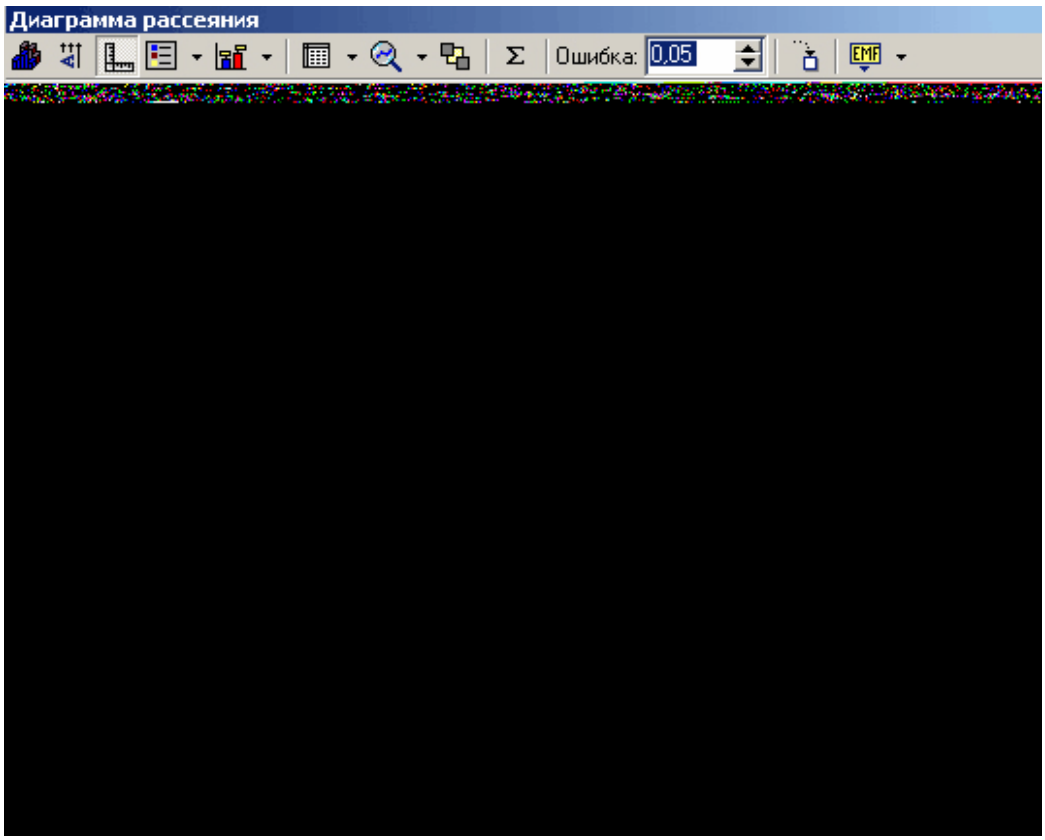
Перейдем к следующему шагу, на котором отметим необходимое количество слоев и нейронов в нейросети.



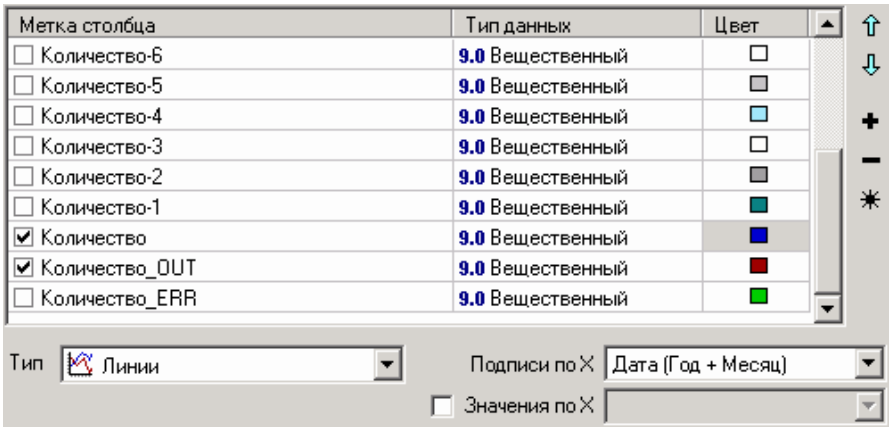
Перейдя далее, выберем алгоритм обучения нейросети.



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.



Результатом будет два графика.

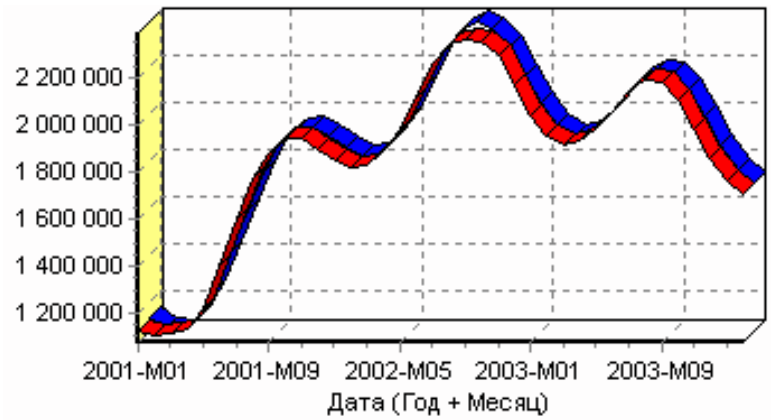
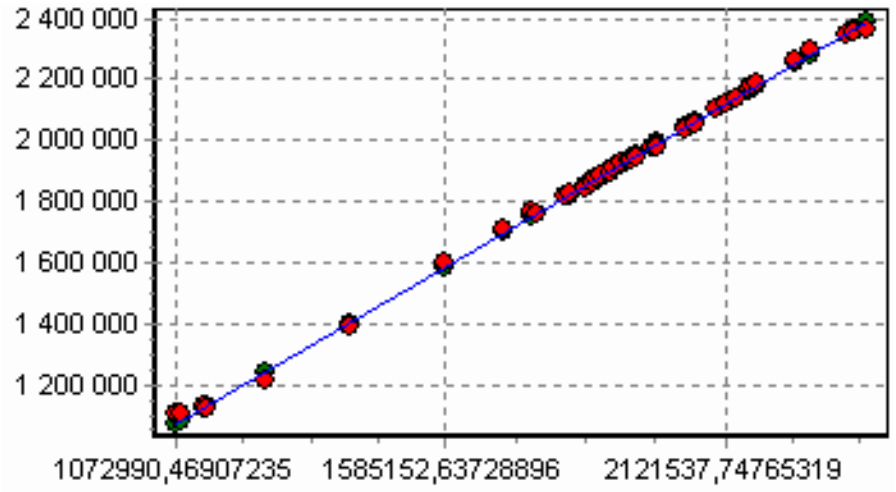


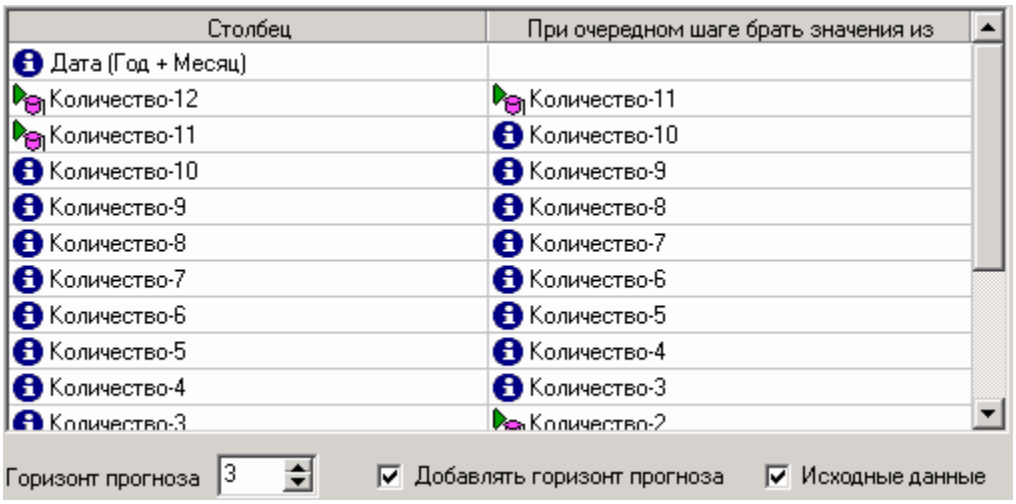
Диаграмма рассеяния более наглядно показывает качество обучения.



4. Построение прогноза

Нейросеть обучена, осталось получить 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) равный трем, а также для наглядности следует добавить к прогнозу исходные данные, установив в Мастере соответствующий флажок.

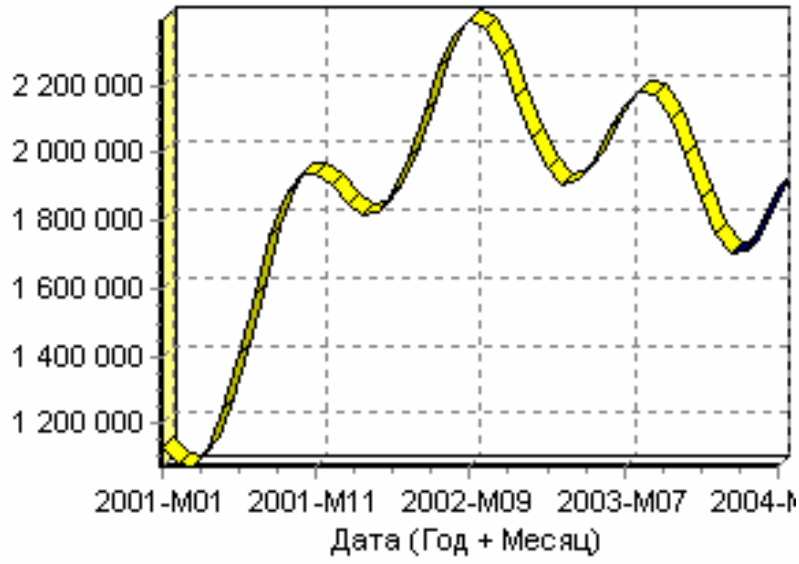


5. Результат

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

В Мастере настройки столбцов диаграммы прогноза надо 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.



Данный пример показал, как с помощью Deductor Studio прогнозировать временной ряд.

При решении задачи были применены механизмы очистки данных отшумов, 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 получать необходимый временной ряд посредством инструментов Deductor Studio (например, группировки), так и подбирать 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).

Критерии оценивания устных ответов аспирантов

Развернутый ответ аспиранта должен представлять собой связное, логически последовательное сообщение на определенную тему, показывать его умение применять определения, правила в конкретных случаях.

При оценке устного ответа аспиранта необходимо руководствоваться следующими критериями:

1) полнота и правильность ответа;

2) степень осознанности, понимания изучаемого материала;

3) знание терминологии и правильное ее использование;

4) соответствие требованиям рабочей программы по дисциплине.

Оценка **«зачтено»** за устный ответ ставится, если аспирант:

1) ориентируется в излагаемом материале, владеет базовой терминологией в объеме, предусмотренном рабочей программой дисциплины;

2) обнаруживает понимание материала, может обосновать свои суждения, подкрепляет теоретические положения примерами;

3) умеет структурировать содержание ответа в соответствии с поставленным вопросом;

4) не допускает (или допускает немногочисленные негрубые) ошибки при анализе языковых фактов; способен исправить допущенные им ошибки при помощи уточняющих вопросов преподавателя.

Порядок проведения дифференцированного зачета

Дифференцированный зачет используется для оценки соответствия результатов освоения дисциплины аспирантом планируемым.

Дифференцированный зачет проводится путем оценивания представления аспирантом индивидуального задания.

Задание выдается преподавателем и состоит из письменного выполнения следующих элементов:

- индивидуальный план работы преподавателя (фрагмент за семестр по одной дисциплине);
- календарный план занятий по дисциплине на семестр;
- рабочая программа дисциплины (фрагмент);
- план проведения занятия (любой формы);
- презентация занятия.

Аспирант в установленный преподавателем срок сдает преподавателю выполненное индивидуальное задание для проверки. При положительном результате проверки аспирант представляет презентацию и обсуждает выполненное индивидуальное задание с преподавателем, по итогам презентации и обсуждения преподаватель выставляет оценку. Оценка объявляется аспиранту и заносится в зачетную ведомость.

Выполненные индивидуальные задания в электронном виде и на бумажном носителе хранятся на кафедре системного анализа и управления.

Критерии и процедура оценивания результатов дифференцированного зачета

Оценки за представление аспирантом индивидуального задания выставляются, исходя из следующих критериев:

- **«отлично»**: если аспирант глубоко и прочно усвоил весь программный материал лекций и демонстрирует это в задании, все документы выполнены без ошибок, последовательно, грамотно и логически построены, излагает свои решения, хорошо их объясняя и обосновывая;
- **«хорошо»**: если аспирант твердо знает программный материал, не допускает существенных неточностей в его изложении, использует ограниченный круг источников, вместо своего решения в задании излагает одно из стандартных;
- **«удовлетворительно»**: если аспирант поверхностно усвоил основной материал лекций, не знает деталей, допускает неточности, при разработке задания привлекает мало оригинального материала, пользуясь, в основном, стандартными решениями и формулировками;
- **«неудовлетворительно»**: если аспирант не знает значительной части программного материала, в задании допущены существенные ошибки, с большими затруднениями выполняет или, по существу, не выполняет задания, не может его объяснить.

ПЕРЕЧЕНЬ УЧЕБНОЙ ЛИТЕРАТУРЫ И РЕСУРСОВ СЕТИ «ИНТЕРНЕТ»

. Основная литература

1. Бессмертный, И. А. Системы искусственного интеллекта: учебное пособие для академического бакалавриата / И. А. Бессмертный. – 2-е изд., испр. и доп. – М.: Юрайт, 2017. – 130 с.

2. Боровская, Е. Основы искусственного интеллекта [Текст] / Е. Боровская. – М.: Бином, 2015. – 128 с.

3. Бураков, М.В. Системы искусственного интеллекта. Учебное пособие [Текст] / М.В. Бураков. – М.: Проспект, 2017. – 440 с.

4. Кудрявцев, В. Б. Интеллектуальные системы: учебник и практикум для бакалавриата и магистратуры [Текст] / В. Б. Кудрявцев, Э. Э. Гасанов, А. С. Подколзин. – 2-е изд., испр. и доп.; МГУ им. М.В. Ломоносова. – М.: Юрайт, 2017. – 219 с.

5. Ясницкий, Л.Н. Введение в искусственный интеллект: учебное пособие [Текст] / Л.Н. Ясницкий. – М.: Академия, 2010. – 176 с.

Дополнительная литература

1. Заболеева-Зотова А.В. Лингвистическое обеспечение автоматизированных систем: учебное пособие [Текст] / А.В. Заболеева-Зотова, В.А. Камаев. – М.: Высш. шк., 2008. – 248 с.

2. Редько, В.Г. Эволюция. Нейронные сети. Интеллект: Модели и концепции эволюционной кибернетики [Текст] / В. Г. Редько. - М.: Едиториал УРСС, 2017. – 224 с.

3. Станкевич, Л.А. Интеллектуальные системы и технологии: учебник и практикум для бакалавриата и магистратуры [Текст] / Л. А. Станкевич. – М.: Юрайт, 2017. – 397 с.

4. Магола, Д. Логическое программирование в среде Visual Prolog [Текст] / Д. Магола. – М.: Palmarium Academic Publishing, 2014. – 136 с.

5. Марков, В. Современное логическое программирование на языке Visual Prolog 7.5. Учебник [Текст] / В. Марков. – СПб.: БХВ-Петербург, 2016. – 544 с.

Учебно-методическое обеспечение самостоятельной работы аспиранта

– Методические указания для самостоятельной работы аспирантов;

– Методические указания по практическим занятиям.

Ресурсы сети «Интернет»

1. Информационная справочная система «Консультант плюс».
2. Библиотека ГОСТов www.gostrf.com.
3. Сайт Российской государственной библиотеки. http://www.rsl.ru/
4. Сайт Государственной публичной научно-технической библиотеки России. http://www.gpntb.ru/
5. Каталог образовательных интернет ресурсов http://www.edu.ru/modules.php
6. Электронные библиотеки: http://www.pravoteka.ru/, http://www.zodchii.ws/, http://www.tehlit.ru/
7. Специализированный портал по информационно-коммуникационным технологиям в образовании http://www.ict.edu.ru

7.5. Электронно-библиотечные системы:

- ЭБС издательства «Лань» https://e.lanbook.com/
- ЭБС издательства «Юрайт» https://biblio-online.ru/
- ЭБС «Университетская библиотека онлайн» https://biblioclub.ru/
- ЭБС «ZNANIUM.COM» https://znanium.com
- ЭБС «IPRbooks» https://iprbookshop.ru
- ЭБС «Elibrary» https://elibrary.ru
- Автоматизированная информационно-библиотечная система «Mark -SQL» https://informsystema.ru
- Система автоматизации библиотек «ИРБИС 64» https://elnit.org

Информационные справочные системы:

1. Система ГАРАНТ: информационный правовой портал [Электронный ресурс]. – Электр.дан. http://www.garant.ru/
2. Консультант Плюс: справочно - поисковая система [Электронный ресурс]. – Электр.дан. www.consultant.ru/
3. ООО «Современные медиа-технологии в образовании и культуре». http://www.informio.ru/
4. Программное обеспечение Норма CS «Горное дело и полезные ископаемые» https://softmap.ru/normacs/normacs-gornoe-delo-i-poleznye-iskopaemye/
5. Информационно-справочная система «Техэксперт: Базовые нормативные документы» http://www.cntd.ru/