МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ
Федеральное государственное автономное образовательное учреждение высшего образования
«СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»
Невинномысский технологический институт (филиал)

Методические указания для в ыполнения лабораторных работ по дисциплине «Интеллектуальный анализ данных и машинное обучение»

(ЭЛЕКТРОННЫЙ ДОКУМЕНТ)

Направление подготовки 09.03.02 Информационные системы и технологии
Профиль: Цифровые технологии химических производств

Квалификация выпускника Бакалавр

Невинномысск 2024

Методические указания предназначены для проведения лабораторных работ по дисциплине «Интеллектуальный анализ данных и машинное обучение» для студентов направления подготовки 09.03.02 Информационные системы и технологии и соответствуют требованиям ФГОС ВО направления подготовки бакалавров.

Составитель: доцент кафедры ИСЭА Э.Е. Тихонов

Тематический план лабораторных работ

№ Темы дисциплины	Наименование тем дисциплины, их краткое содержание	ОФО	ЗФО	Из них в интерактивной форме
7 семестр				
Тема 1. Концепция Data Mining				
1	Знакомство с программой Deductor Academic	2		лабораторная работа
2	Анализ признаков и оценка их информативности в программе Deductor Academic	2		лабораторная работа
3	Анализ признаков и оценка их информативности в программе Deductor Academic	2		лабораторная работа
Тема 2. Задачи Data Mining. Классификация задач				
4	Базовые методы интеллектуального анализа данных в программе Deductor Academic	2		лабораторная работа
5	Методы интеллектуального анализа данных в программе Deductor Academic	2	2	лабораторная работа
6	Расширение возможностей интеллектуального анализа данных в программе Deductor Academic	2		лабораторная работа
Тема 3. Практическое применение Data Mining				
7	Классификация данных с помощью нейронной сети в программе Deductor Academic	2		лабораторная работа
8	Применение интеллектуального анализа данных в задачах поддержки принятия решений	2		лабораторная работа
Тема 4. Модели Data Mining				
9	Прогнозирование умножения с помощью нейронных сетей*	2		лабораторная работа
10	Прогнозирование данных на основе временного ряда*	2		лабораторная работа
Тема 5. Базовые методы Data Mining				
11	Нейросетевые технологии в интеллектуальном анализе данных*	2	2	лабораторная работа
12	Нейросетевые технологии в интеллектуальном анализе данных. Расширение возможностей нейронных сетей*	2		лабораторная работа
Тема 6. Процесс обнаружения знаний				
13	Пакеты Numpy, Scipy, математические операции в них.	2		лабораторная работа
14	Пакет Pandas, работа с данными в нем	2		лабораторная работа
15	Метрики качества алгоритмов машинного обучения, кросс-валидация.	2	2	лабораторная работа
16	Деревья решений, их построение	2		лабораторная работа
17	Композиции алгоритмов. Случайные леса	2		лабораторная работа
18	Поиск частых множеств и ассоциативных правил	2		лабораторная работа
	Итого за семестр	**36**	**6**	
8 семестр				
Тема 7. Математические объекты и методы в анализе данных				
1	Надстройки интеллектуального анализа данных для	2		лабораторная

	MicrosoftOffice. Установка и настройка*			работа
2	Надстройки интеллектуального анализа данных для MicrosoftOffice. Установка и настройка*	2		лабораторная работа
Тема 8. Линейная регрессия и классификация. Продвинутый уровень				
3	Использование инструментов "AnalyzeKeyInfluencers" и "DetectCategories"	2		лабораторная работа
4	Использование инструментов "AnalyzeKeyInfluencers" и "DetectCategories"	2		лабораторная работа
Тема 9. Оценивание качества алгоритмов				
5	Использование инструментов "FillFromExample" и "Forecast"	2	2	лабораторная работа
6	Использование инструментов "FillFromExample" и "Forecast"	2		лабораторная работа
Тема 10. Логические методы				
7	Использование инструментов "HighlightExceptions" и "ScenarioAnalysis"	2		лабораторная работа
8	Использование инструментов "HighlightExceptions" и "ScenarioAnalysis"	2		лабораторная работа
Тема 11. Композиции алгоритмов				
9	Анализ сценариев	2		лабораторная работа
10	Использование инструментов "Prediction Calculator" и "ShoppingbasketAnalysis"	2	2	лабораторная работа
11	Использование инструментов "Prediction Calculator" и "ShoppingbasketAnalysis"	2		лабораторная работа
Тема 12. Особенности реальных данных				
12	Использование инструментов Data Mining Client для Excel для подготовки данных.	2		лабораторная работа
13	Использование инструментов Data Mining Client для Excel для подготовки данных.	2		лабораторная работа
Тема 13. Анализ частых множеств признаков и ассоциативных правил				
14	Использование инструментов Data Mining Client для Excel для создания модели интеллектуального анализа данных.	2	2	лабораторная работа
15	Использование инструментов Data Mining Client для Excel для создания модели интеллектуального анализа данных.	2		лабораторная работа
16	Анализ точности прогноза и использование модели интеллектуального анализ	2		лабораторная работа
17	Анализ точности прогноза и использование модели интеллектуального анализ	2		лабораторная работа
18	Построение модели кластеризации, трассировка и перекрестная проверка	2		лабораторная работа
19	Построение модели кластеризации, трассировка и перекрестная проверка	2	2	лабораторная работа
20	Исследование и использование ресурса http://archive.ics.uci.edu/ml/index.php	2		лабораторная работа
	Итого за семестр	**40**	**8**	
	Итого	**76**	**14**	

* - с применением дистанционных образовательных технологий

По темам работ 1-13 предусмотрены занятия в виде практической подготовки в НТИ (филиал) СКФУ

Содержание

ВВЕДЕНИЕ

Аналитическая платформа *DEDUCTOR* состоит из следующих пяти компонентов: *Deductor Studio*, *Deductor Warehouse*, *Deductor Viewer*, *Deductor Studio* и *Deductor Client*.

Deductor Warehouse – многомерное кроссплатформенное хранилище данных, аккумулирующее всю необходимую для анализа предметной области информацию. Использование единого хранилища позволяет обеспечить непротиворечивость данных, их централизованное хранение автоматически обеспечивает всю необходимую поддержку процесса анализа данных.

Deductor Studio – это программа, предназначенная для анализа информации из различных источников данных. Она реализует функции импорта, обработки, визуализации и экспорта данных. Deductor Studio может функционировать и без хранилища данных.

Deductor Academic Studio предназначен только для образовательных целей. Использование данной версии в коммерческих целях запрещено. Для коммерческого применения необходимо приобрести Deductor Professional или Enterprise. Данную версию можно бесплатно получить на электронном ресурсе *http://www.basegroup.ru*

Deductor Viewer – это облегченная версия Deductor Studio, предназначенная для отображения построенных в Deductor Studio отчетов. Она не включает в себя механизмов создания сценариев, но обладает полноценными возможностями по их выполнению и визуализации результатов.

Deductor Server – сервер удаленной аналитической обработки. Он позволяет выполнять на сервере операции «прогона» данных через существующие сценарии и переобучение моделей. *Deductor Server* ориентирован на обработку больших объемов данных и работу в территориально-распределённой системе.

Лабораторная работа 1

Анализ признаков и оценка их информативности

Цель работы: ознакомиться с возможностями аналитического пакета *Deductor Academic*.

Программа работы

1. Выполнить импорт данных в программный комплекс *Deductor*.
2. Выполнить задание по предварительной парциальной обработке данных.
3. Выполнить задание по предварительной обработке путем удаления аномалий в данных.
4. Выполнить задание по предварительной обработке путем сглаживания данных методом спектральной обработки.
5. Выполнить задание по удаление шумов на этапе предварительной обработке данных.
6. Ознакомиться с возможностями автоматического анализа качества импортируемых данных.

Методические указания по выполнению работы

1.1 Импорт данных в программный комплекс Deductor Academic

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 (Только в коммерческой версии, в бесплатной версии возможен импорт из *.txt, *.*csv* и *.*ded*).

Импорт данных из текстового файла с разделителями осуществляется путем вызова мастера импорта на панели «Сценарии» (рис. 1.1).

Рис. 1.1. - Панель сценарии

После запуска мастера импорта укажем тип импорта «Текстовый файл» и перейдем к настройке импорта(рис. 1.2-3). Укажем имя файла, из которого необходимо получить данные. В окне просмотра, выбранного файла можно увидеть содержание данного файла.

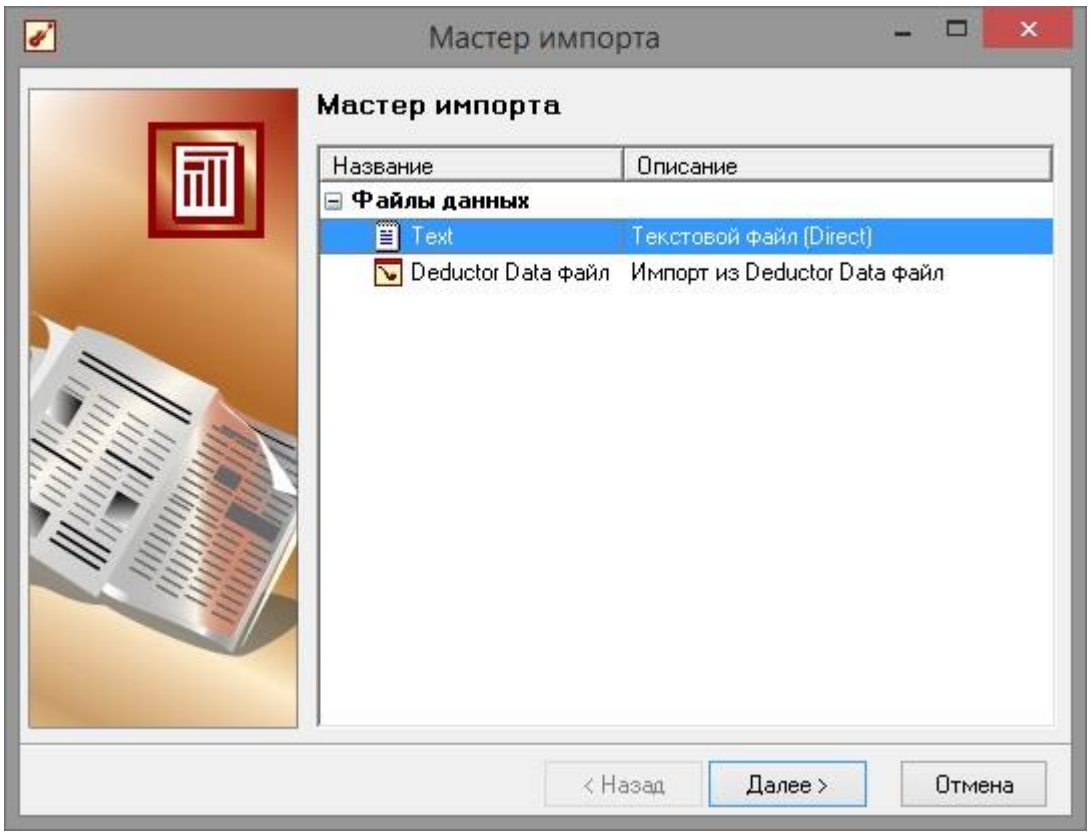


Рис. 1.2 - Мастер импорта

Далее перейдем к настройке параметров импорта (рис. 1.4). 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.

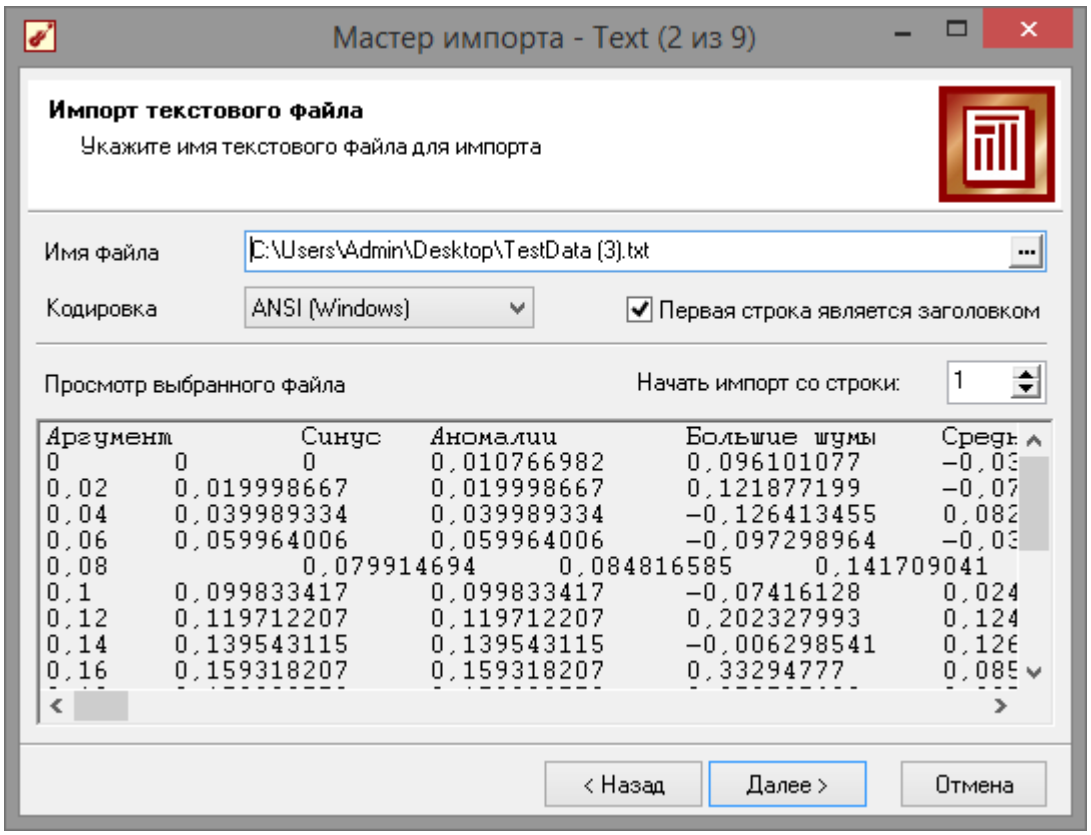


Рис. 1.3 - Выбор файла

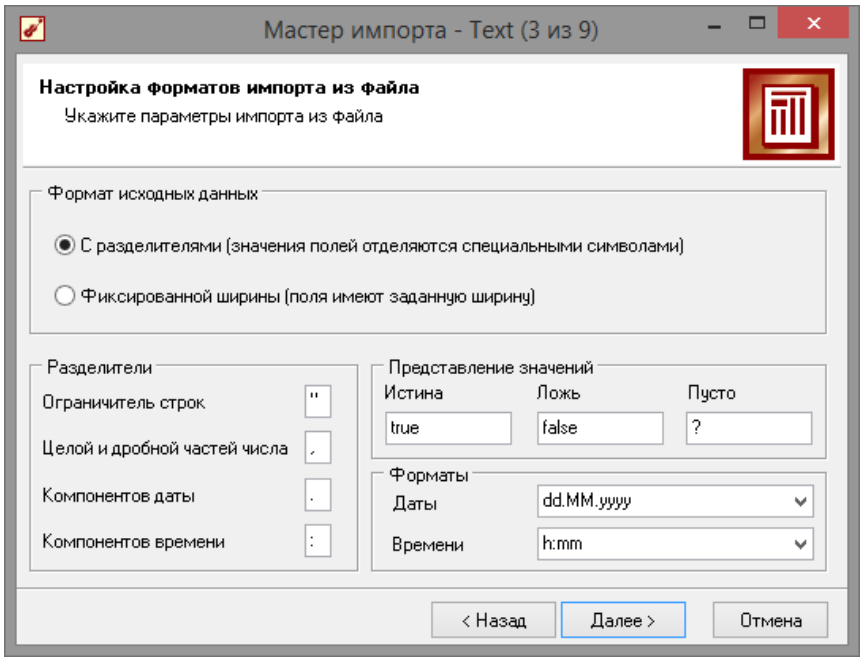


Рис. 1.4 - Параметры импорта

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 (рис. 1.5).

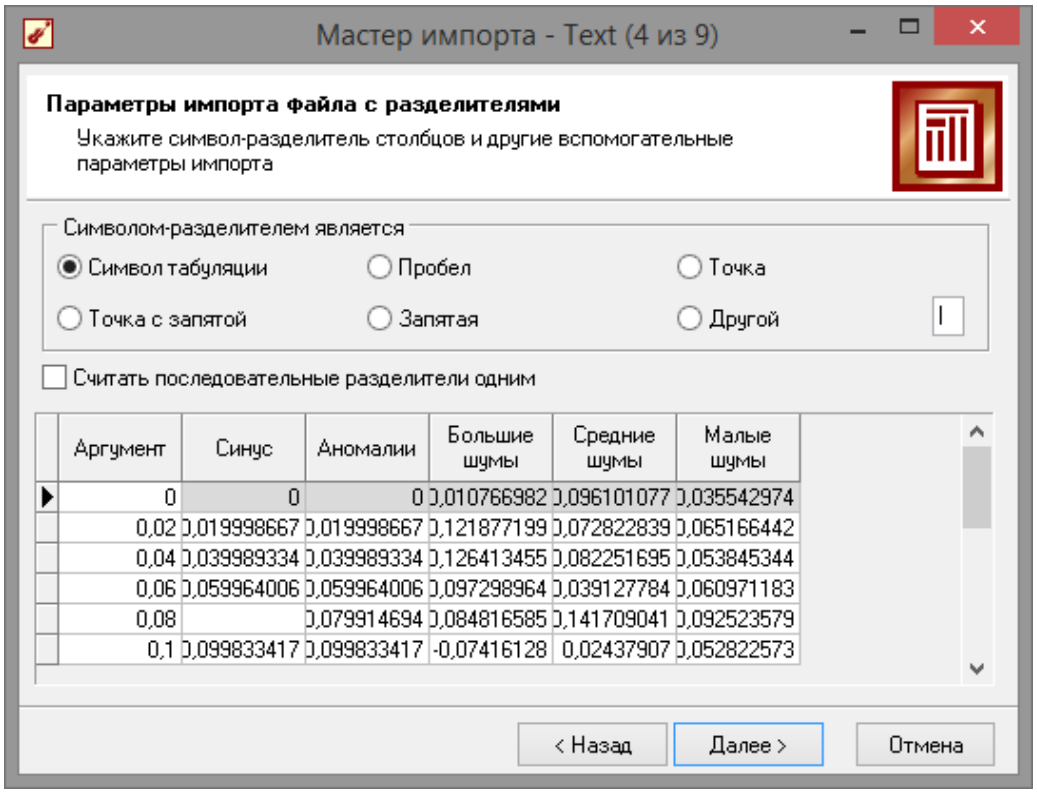


Рис. 1.5 - Параметры разделителей

На этом шаге мастера предоставляется возможность настроить имя, название (метку), размер, тип данных, вид данных и назначение (рис 1.6). 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). Необходимо убедиться, что в данном случае тип данных у все столбцов выставлен как вещественный, так как в более старых версиях тип определялся по первой строке, а в демо-примере столбцы «Аргумент», «Синус» и «Аномалии» имеют в первой строке значение «0», что могло приводить к неправильному определению типа данных.

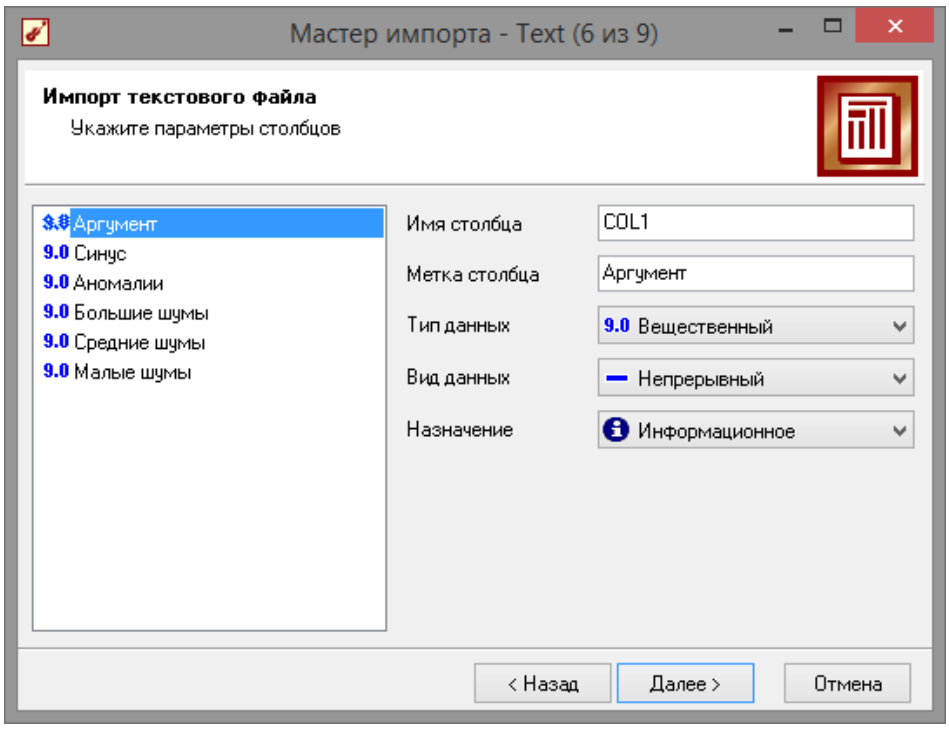


Рис. 1.6 - Параметры столбцов

Далее осталось только выполнить импорт данных, нажав на кнопку «Пуск» на следующем шаге мастера импорта (рис. 1.7). После импорта данных на следующем шаге мастера необходимо выбрать способ отображения данных (рис. 1.8). В данном случае самым информативным является диаграмма, выберем ее.

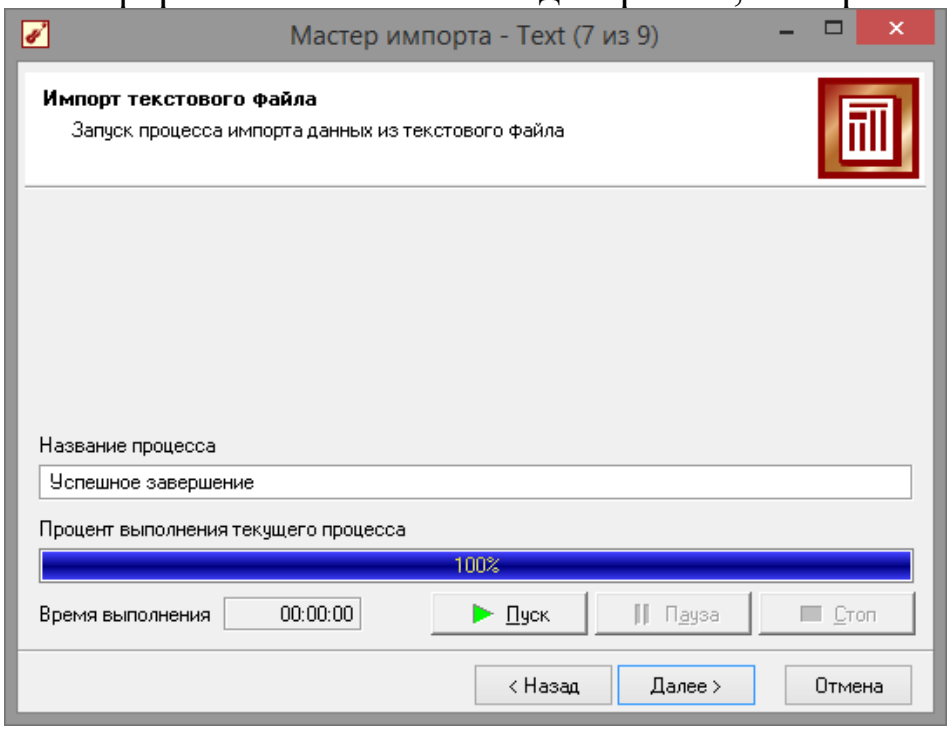


Рис. 1.7 - Импорт файла

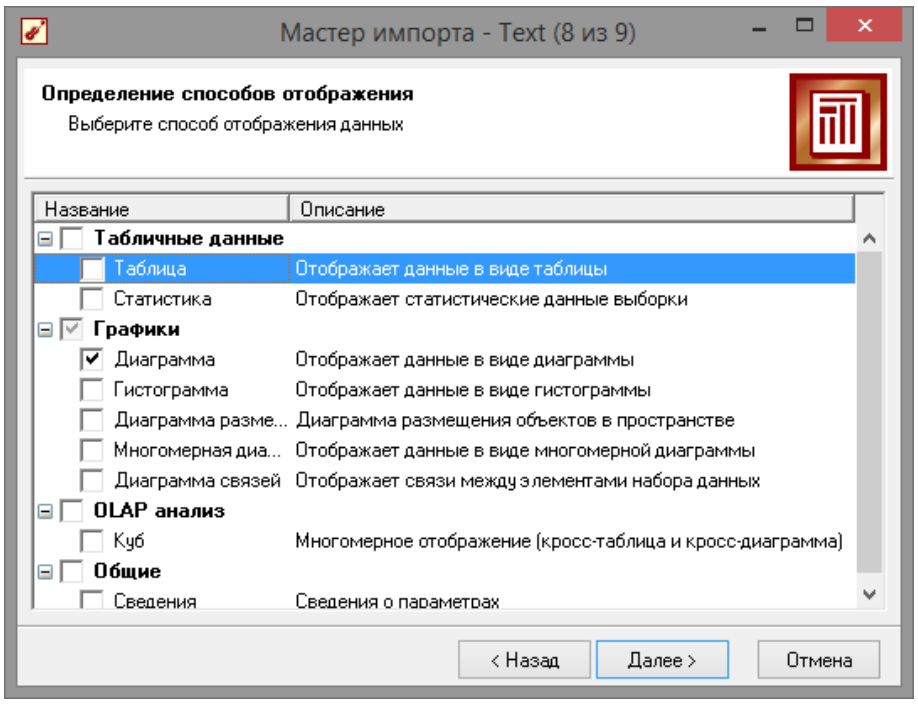


Рис. 1.8 - Способ отображения

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» (рис. 1.9).

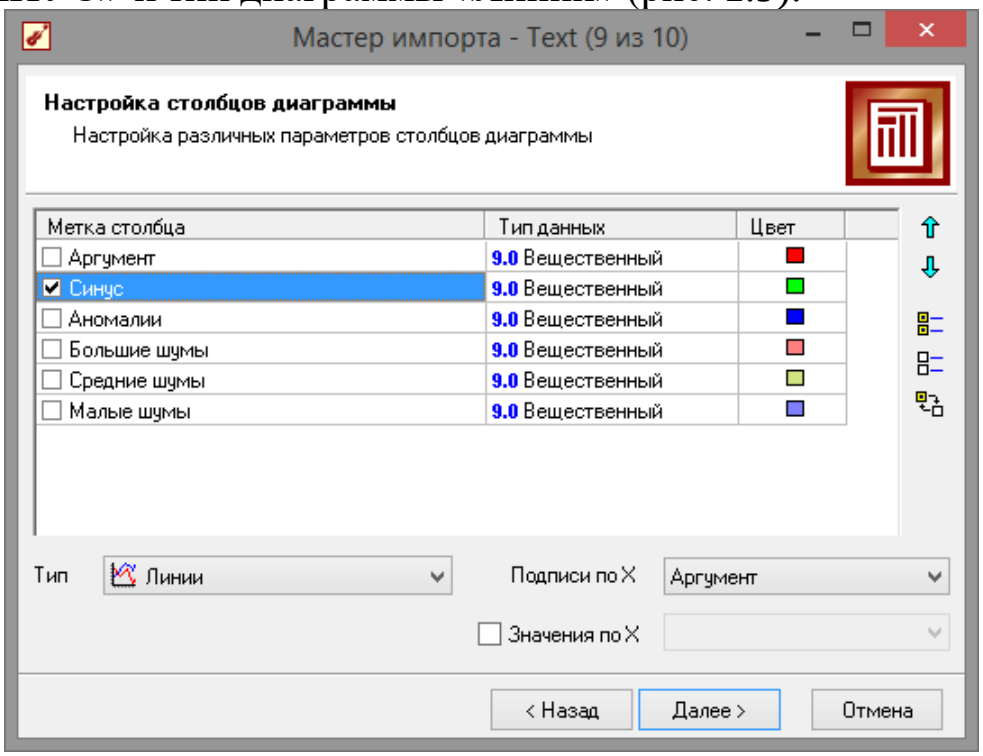


Рис. 1.9 - Настройка столбцов

На последнем шаге мастера необходимо указать название ветки

в дереве сценариев. Напишем в поле заголовка окна «Импорт примера для демонстрации предобработки данных» и нажмем «Готово» (рис. 1.10). 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. Примечание: для отображение диаграммы в 3D-виде, необходимо нажать кнопку

«3-х мерный вид» в левом верхнем углу панели «Диаграмма». А для просмотра другой диаграммы, нажать на значок лупы «Отображать поля» (рис. 1.11).

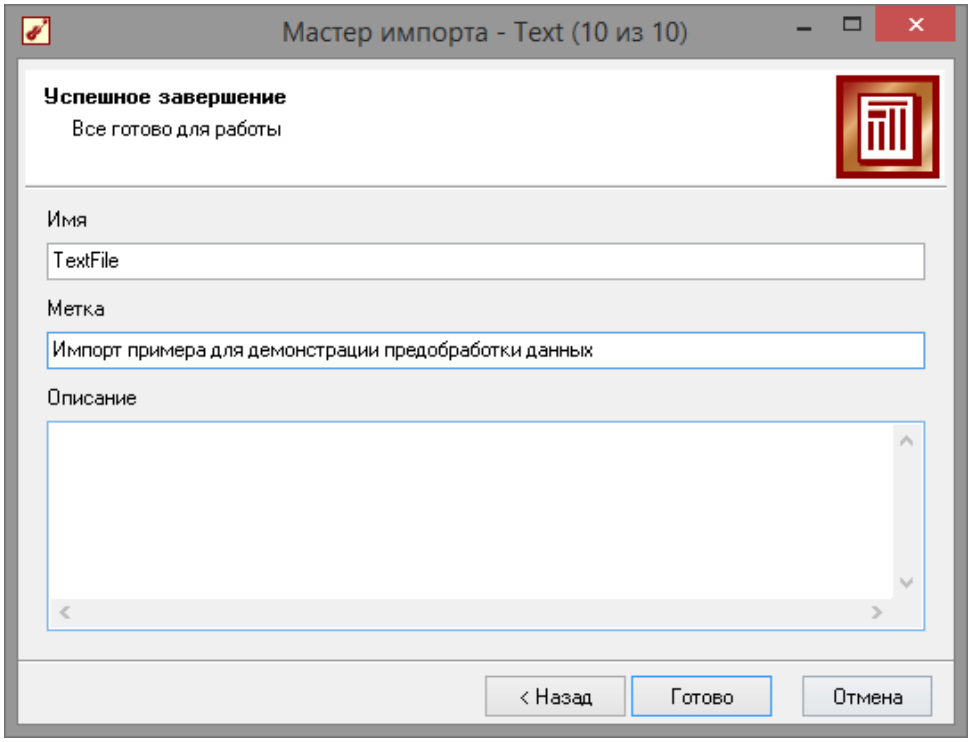


Рис. 1.10 – Завершение импорта

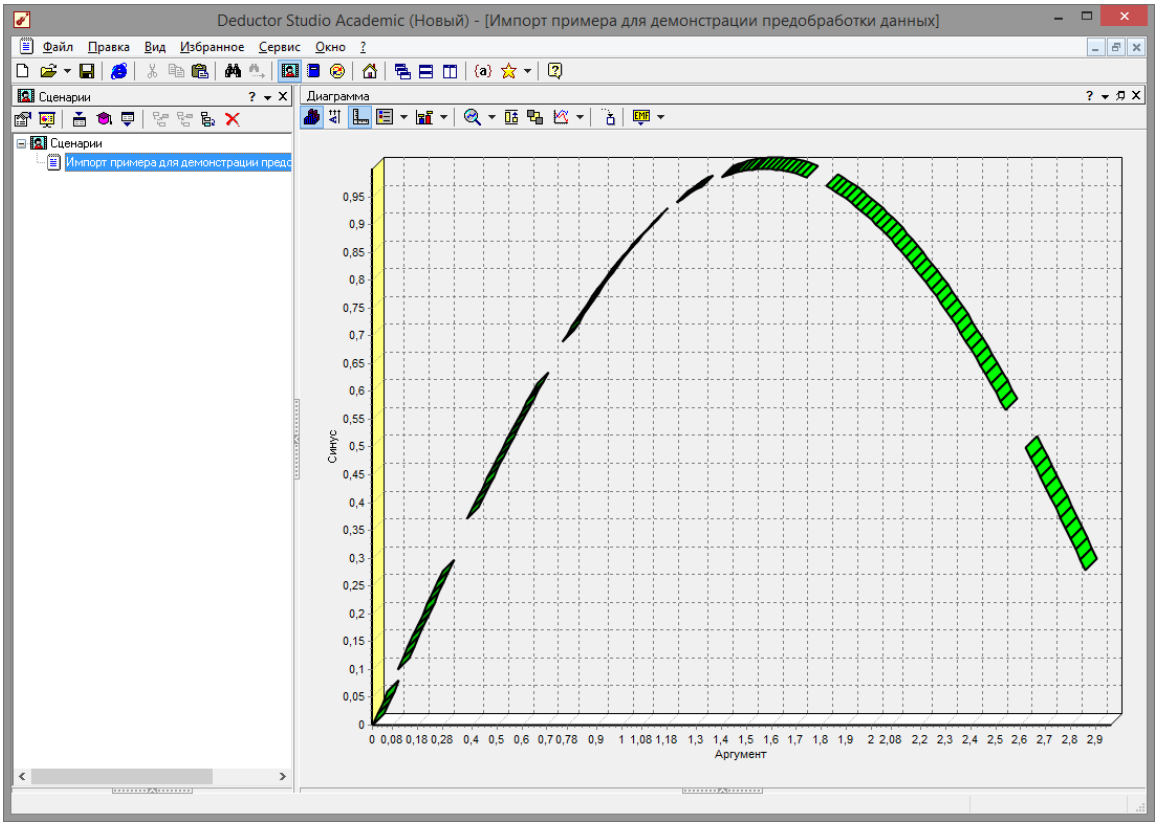


Рис. 1.11 - Диаграмма синуса

1.2 Предварительная парциальная обработка

Рис.

Часто исходные данные для анализа не годятся, а качество данных влияет на качество результатов, поэтому 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 служит для восстановления пропущенных данных, редактирования аномальных значений и спектральной обработке данных (например, сглаживания данных). Именно этот шаг часто проводится в первую очередь.

Рассмотрим применение обработки на примере данных из файла

«TestData.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.

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 заполнения пропусков. Импортировав файл можно увидеть, что в столбце

«СИНУС» содержатся пустые значения. На диаграмме выше видно, что некоторые значения синуса пропущены. Для дальнейшей обработки необходимо из восстановить. Для этого следует запустить мастер заполнения пропусков. На рис.1.12 показаны различные варианты столбцов: с пропущенными данными; с аномалиями (выбросами); с большими шумами; со средними шумами; с малыми шумами.

Для запуска мастера необходимо выделит нужный сценарий и нажать F7, либо правый клик по необходимому сценарию откроет контекстное меню, где так же можно выбрать мастер обработки. Поскольку данные в исходном наборе упорядочены, на следующем шаге мастера обработки поставим галочку – «обрабатывать как упорядоченный набор» (рис. 1.13).

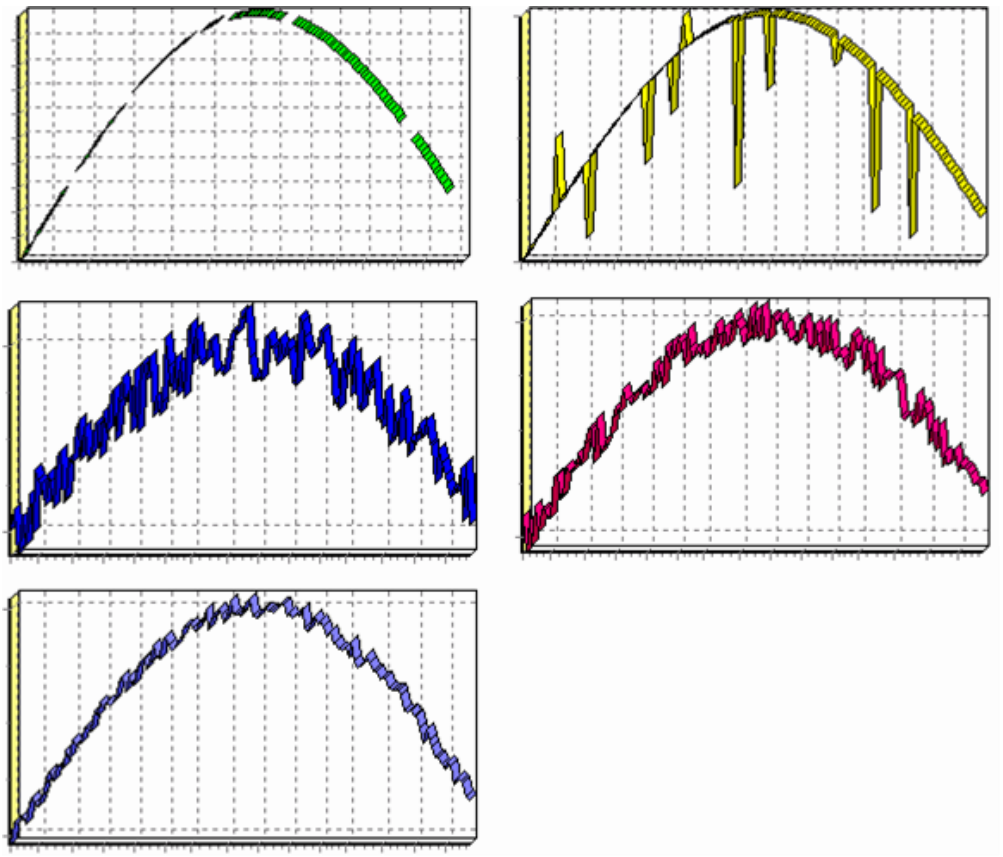


Рисунок 1.12 - Варианты столбцов

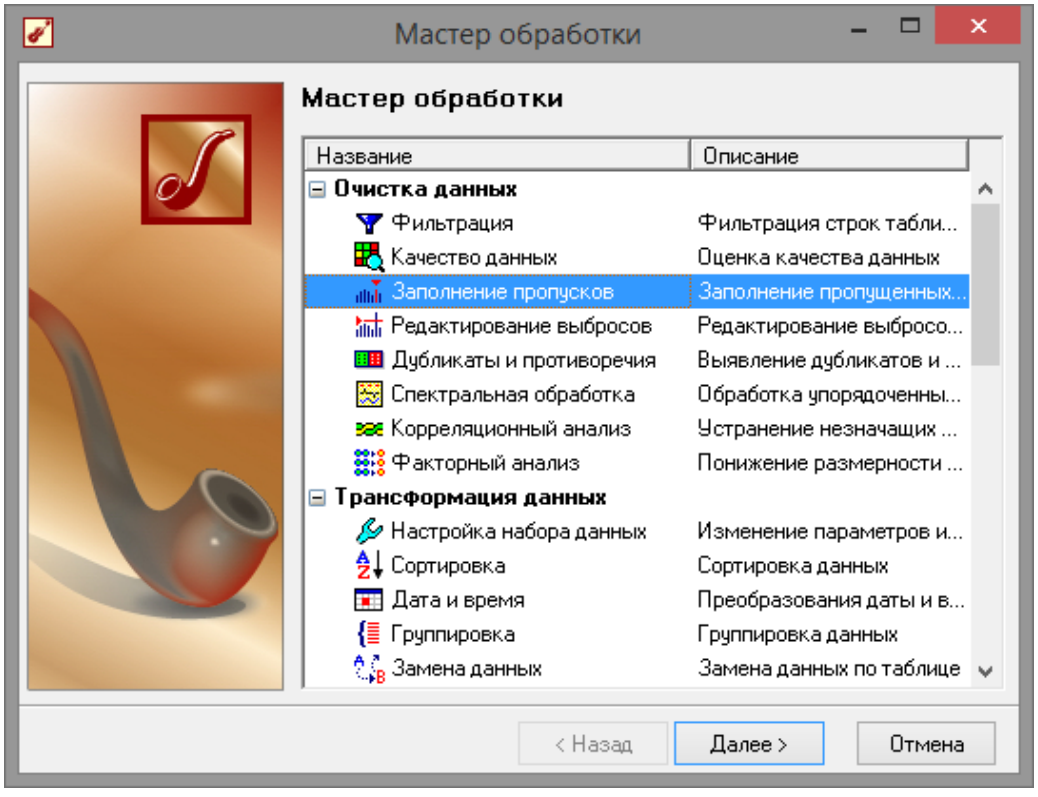


Рисунок 1.13 - Мастер обработки

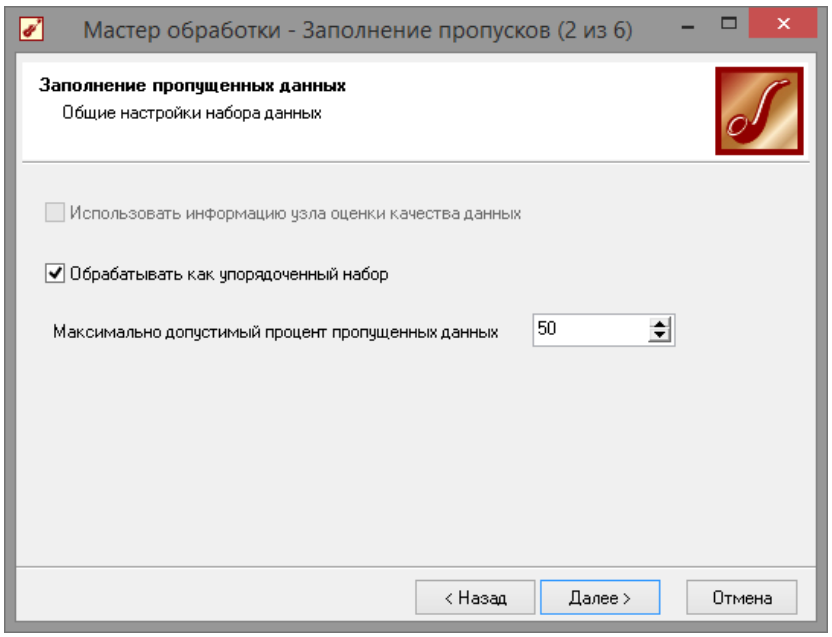


Рис. 1.14 - Мастер заполнения пропусков

Далее следует выбрать необходимый столбец и метод заполнения, в данном случае интерполяция (рис. 1.14). Перейдя на страницу запуска процесса обработки, выполняем ее, нажав на пуск, и далее выбираем тип визуализации обработанных данных (как в примере импорта) (рис. 1.15).. После выполнения процесса обработки на диаграмме видно, что пропуски в данных исчезли, что и было необходимо сделать (рис. 1.16).

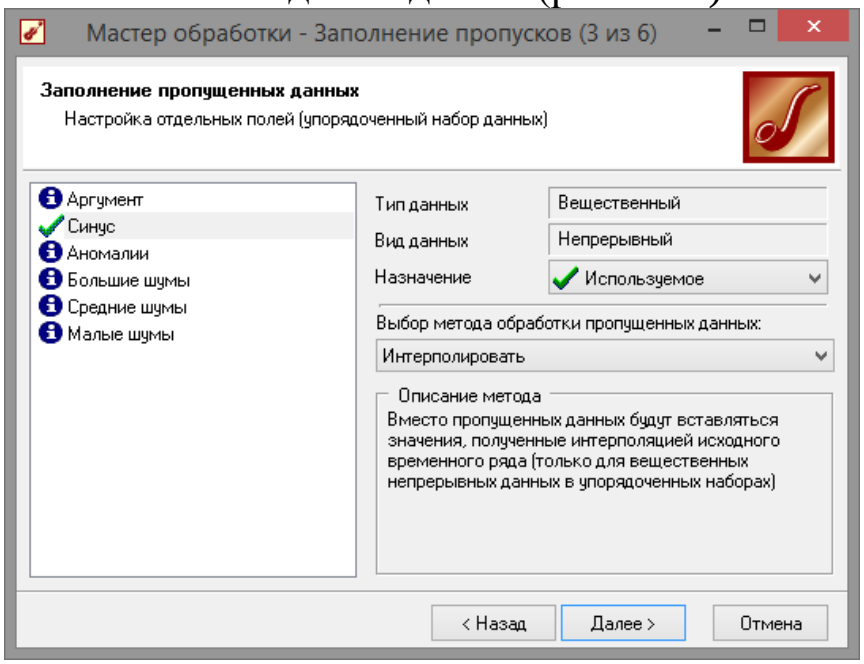


Рис. 1.15 - Настройки мастера заполнения пропусков

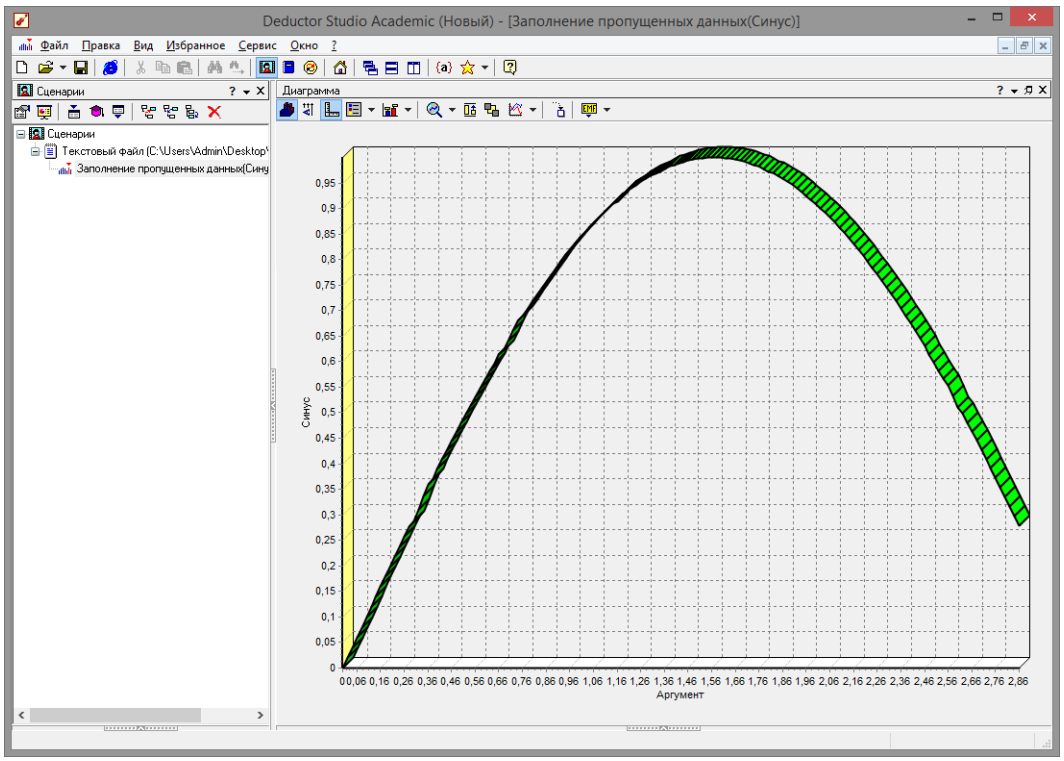


Рис.1.16 - Заполнение пропусков методом интерполяции

1.3 Удаление аномалий на этапе предобработки данных

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 и их предварительно необходимо устранить. Также они портят статистическую картину распределения данных. К примеру, вот как выглядят данные с аномалиями, а также гистограмма их распределения (рис. 1.17).

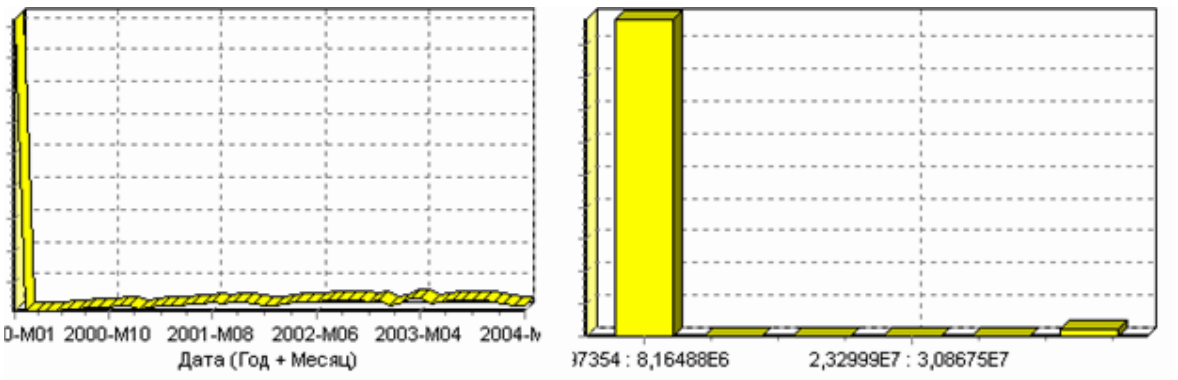


Рис. 1.17 -Гистограмма с аномалиями

Очевидно, что аномалии не позволяют определить, как характер самих данных, так и статистическую картину. После

устранения аномалий те же данные представляются как показано на рис. 1.18

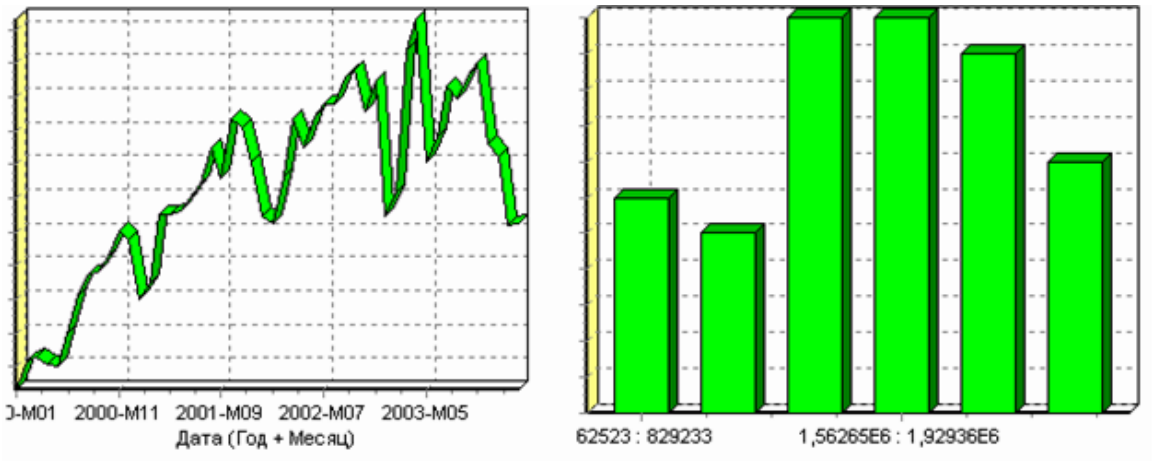


Рис. 1.18 - Гистограмма без аномалий

Следует открыть мастер обработки и выбрать редактирование выбросов (рис. 1.19). Поставить галочку - «Обрабатывать как упорядоченный набор данных» (рис. 1.20).

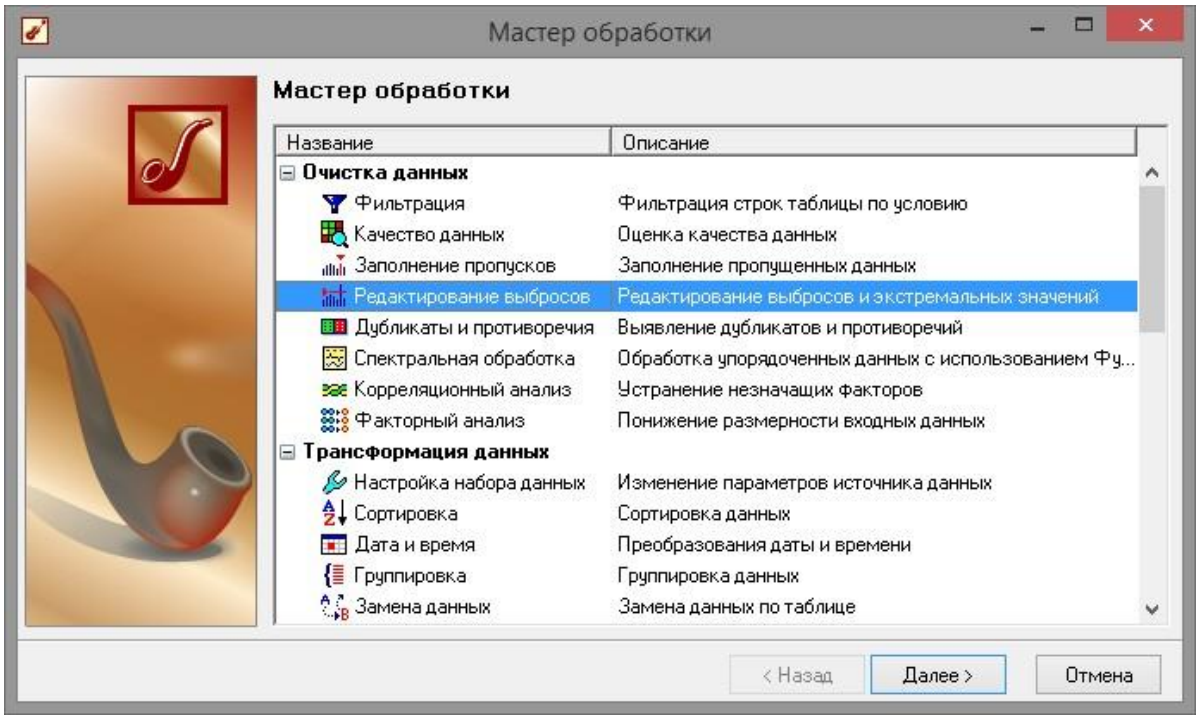


Рис. 1.19 - Мастер редактирования выбросов

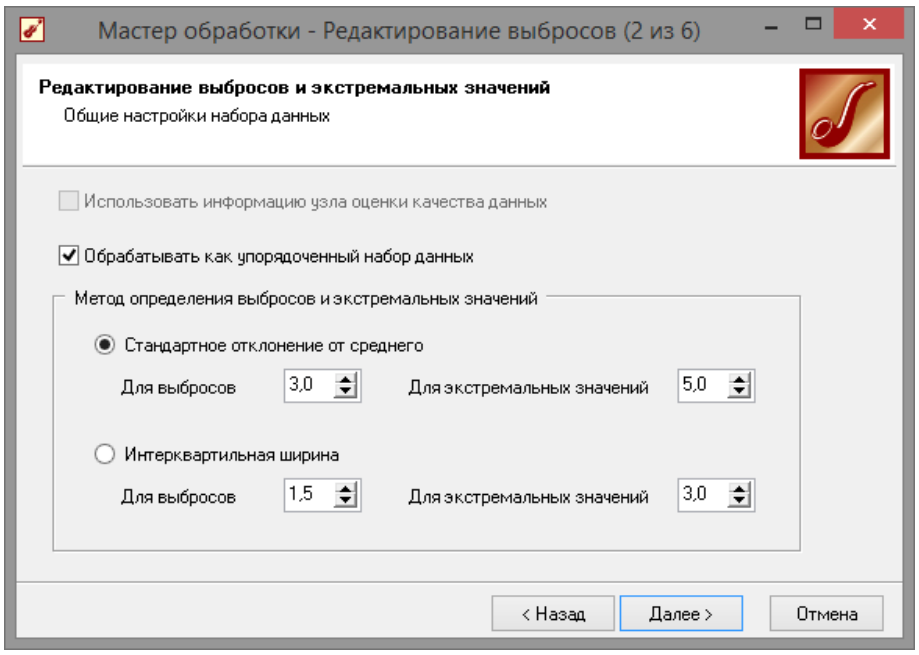


Рис. 1.20 - Настройки мастера редактирования выбросов

На следующем шаге (рис. 1.21) необходимо выбрать назначение «Используемое» только для необходимого столбца данных. И выставить большую степень подавления, так как выбросы существенны. Для множественного выделения столбцов можно использовать клавишу *Shift* и *Ctrl*.

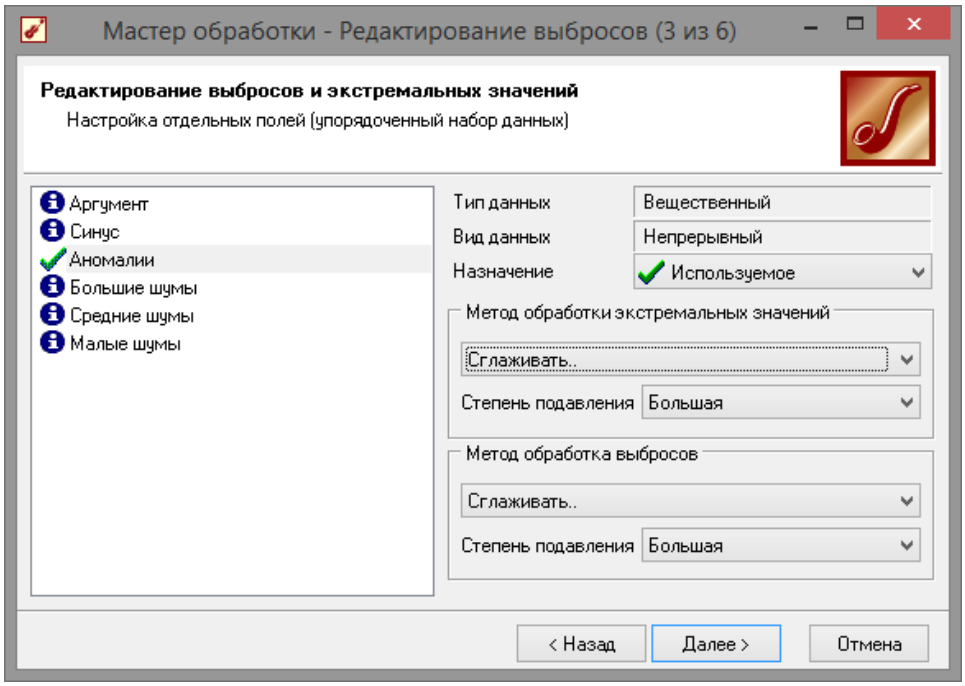


Рис. 1.21 - Настройки отдельных полей мастера редактирования выбросов

Далее нажать кнопку «Пуску» и выбрать данные для

отображения как в предыдущих пунктах. После выполнения процесса обработки на диаграмме видно, что выбросы исчезли, остались лишь небольшие возмущения, которые легко сгладить при помощи спектральной обработки (рис. 1.22).

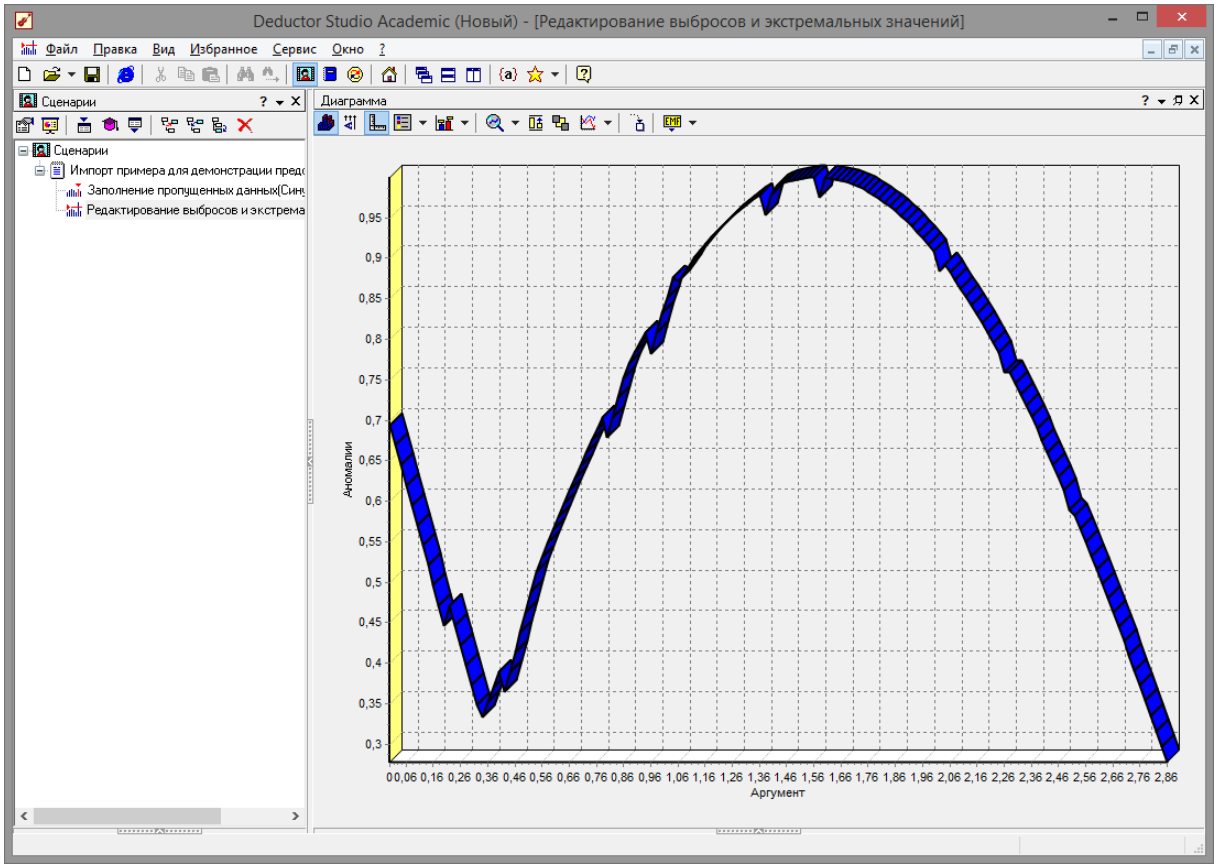


Рис. 1.22 - Диаграмма после удаления аномалий

1.4 Сглаживания данных методом спектральной обработки

Сглаживание данных применяется для удаления шумов из исходного набора, а также для выделения тенденции, которая трудно видна 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.

Для выбора способа вейвлет преобразования открыть мастер обработки для сценария «Редактирование выбросов и экстремальных

значений». В мастере следует выбрать пункт «спектральная

обработка» (рис. 1.23)

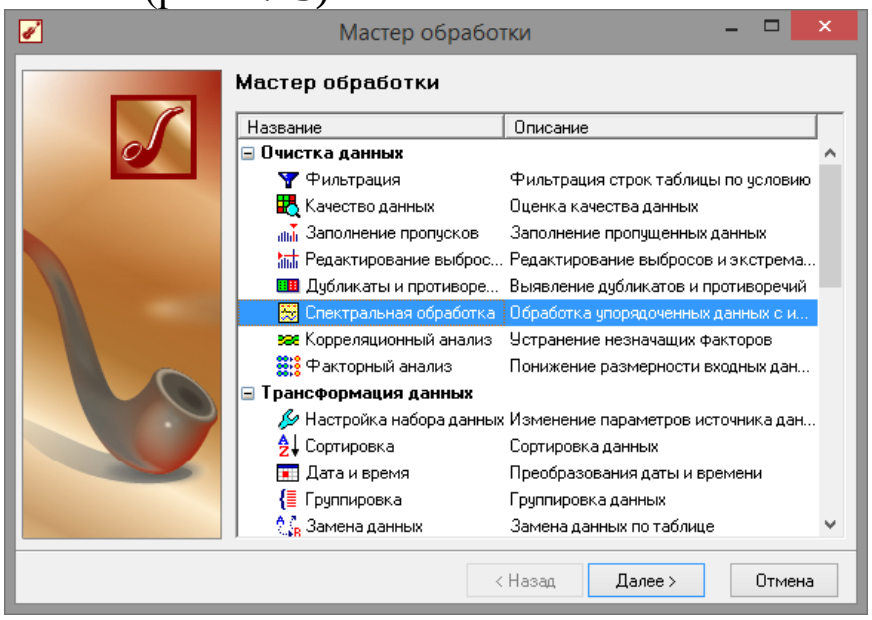


Рис. 1.23 - Мастер обработки

В мастере спектральной обработки необходимо выбрать назначение используемое для столбца «Аномалии», и в качестве метода сглаживания данных выбрать вейвлет преобразование (рис. 1.24).

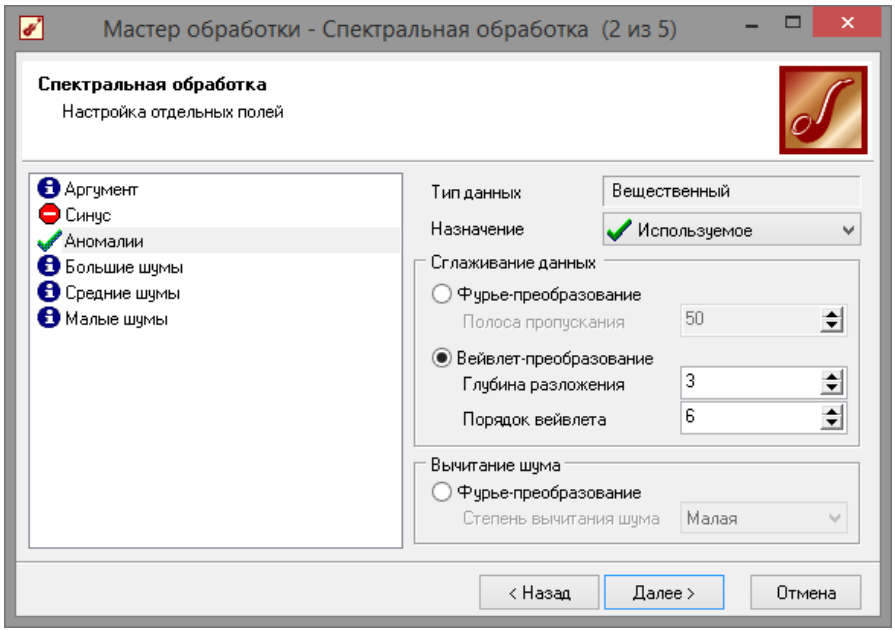


Рис. 1.24 - Мастер спектральной обработки

Далее выбрать визуализацию «Диаграмма» и столбец «Аномалии». После обработки можно убедиться на диаграмме в отсутствии выбросов (рис. 1.25).

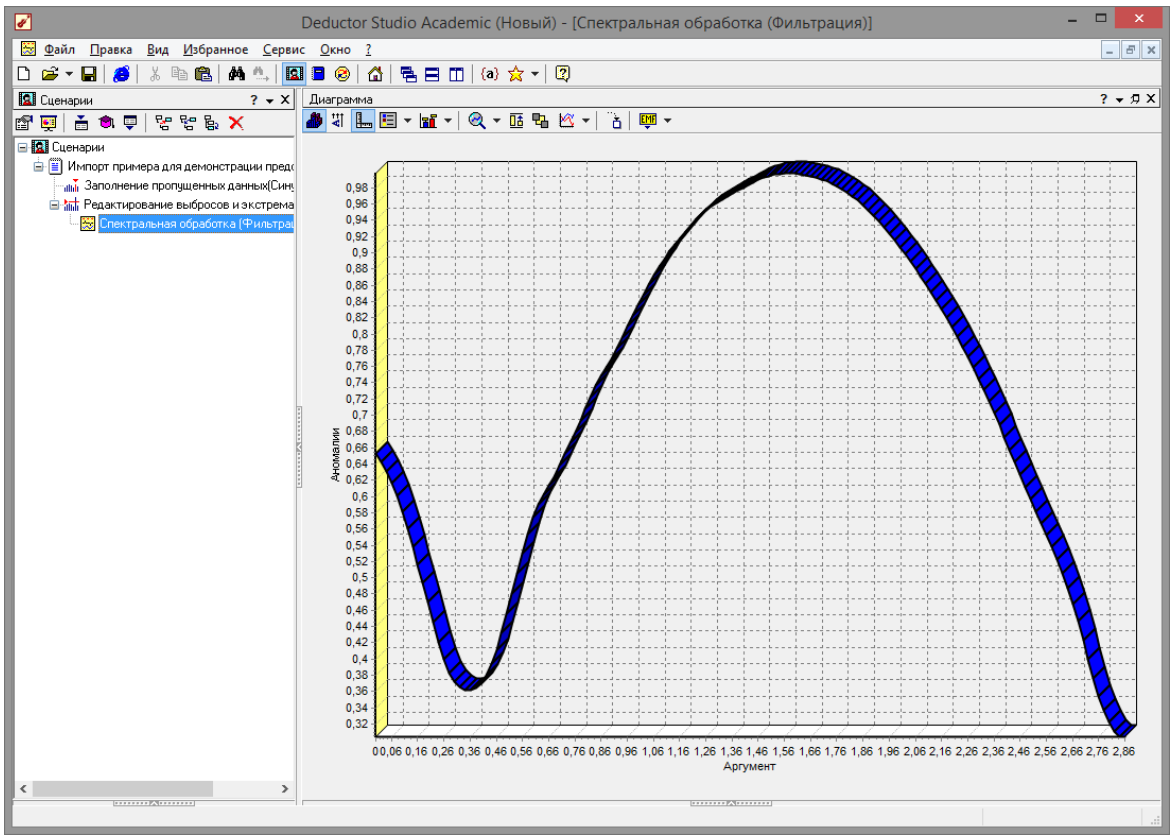


Рис. 1.25 - Диаграмма после применения спектральной обработки

1.5 Удаление шумов на этапе предварительной обработке данных

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 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 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.

В мастере спектральной обработки (рис. 1.26) по очереди

выбрать поля «БОЛЬШИЕ ШУМЫ», «СРЕДНИЕ ШУМЫ» и «МАЛЫЕ ШУМЫ», задать тип обработки «Вычитание шума» и указать степень подавления – «большая», «средняя» и «малая» соответственно. В некоторых случаях неплохие результаты удаления шумов дает вейвлет преобразование. Повысить качество сглаживания шумов таким способом можно, путем подбора удовлетворительных параметров обработки (рис. 1.27).

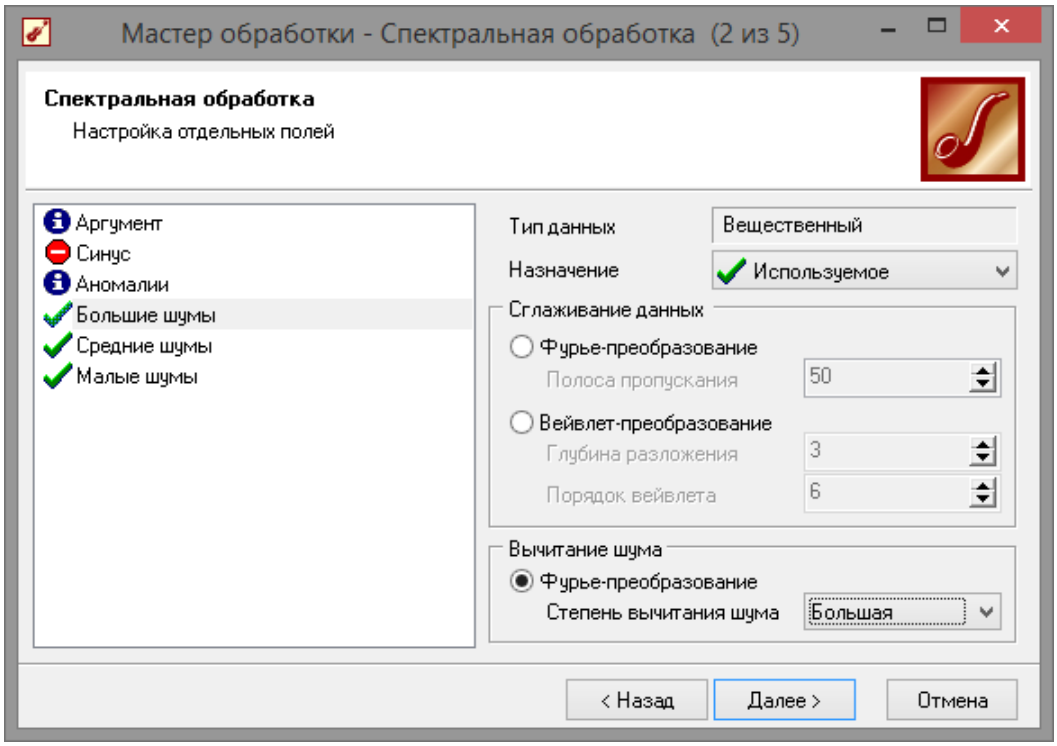


Рис. 1.26 - Настройки мастера спектральной обработки

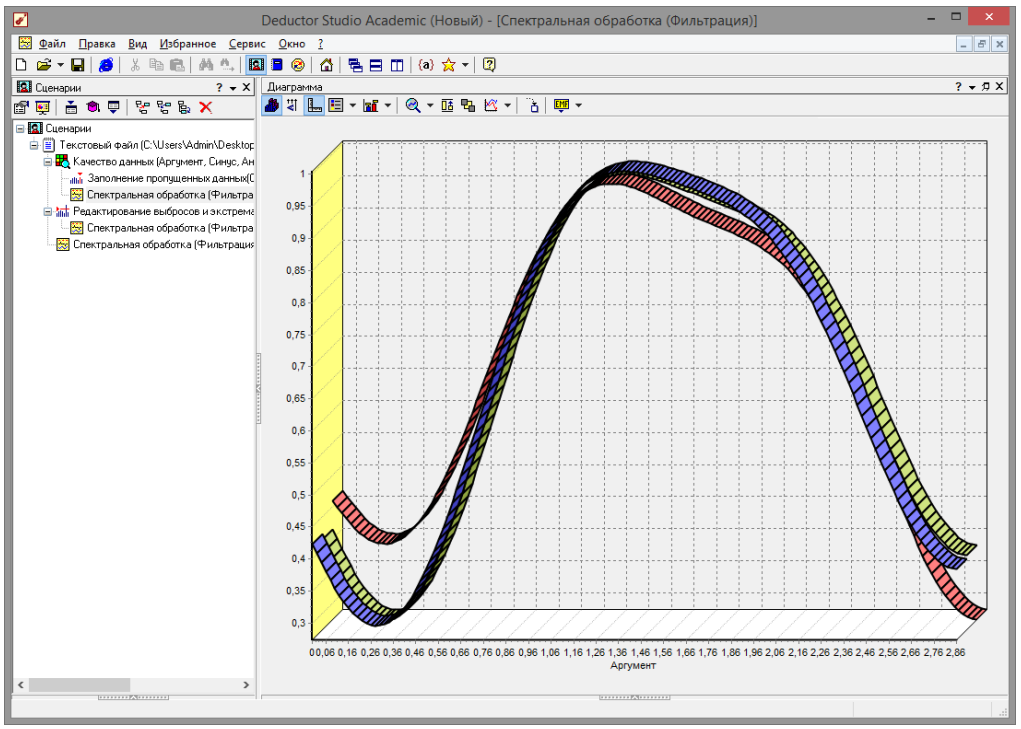


Рис 1.27 - Диаграмма после применения спектральной обработки

1.6 Возможности автоматического анализа качества импортируемых данных в *Deductor Academic*

В мастере обработки выбрать пункт «качество данных» (рис. 1.28).

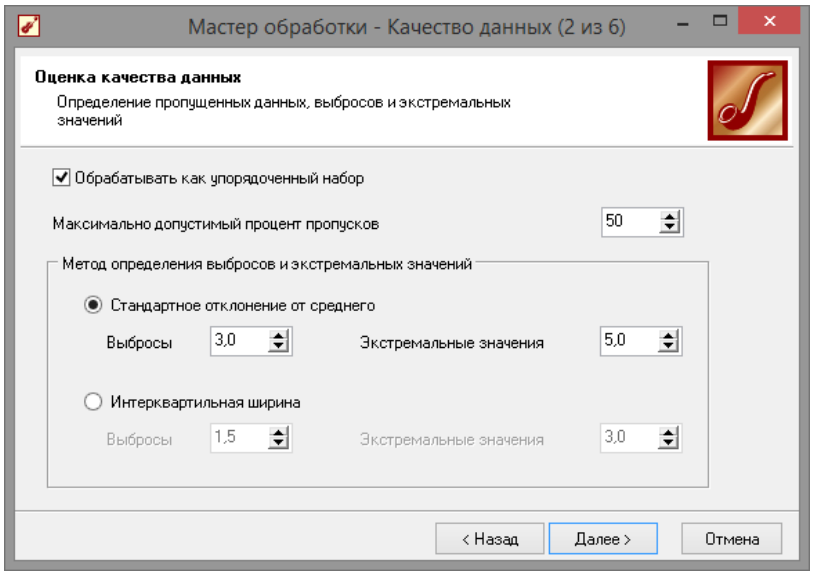


Рис. 1.28 - Мастер качества данных

После анализа мастер дает рекомендации к обработке данных и возможность автоматического исправления (рис. 1.29). Следует отметить, что автоматические исправление далеко не всегда дает желаемые результаты (рис. 1.30).

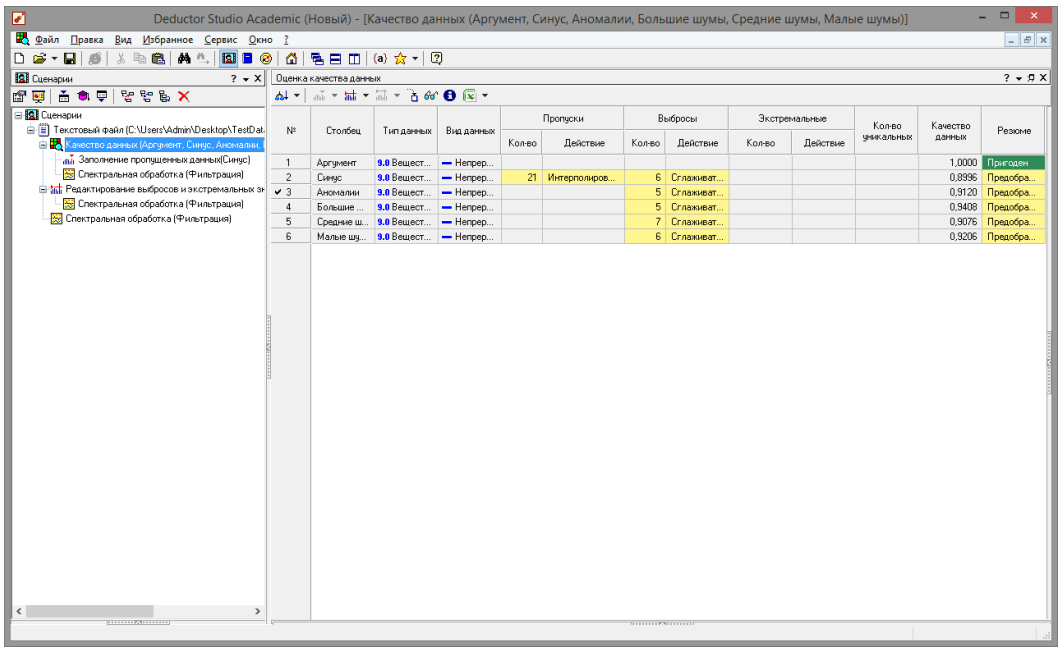


Рис. 1.29 - Результаты мастера качества до обработки данных

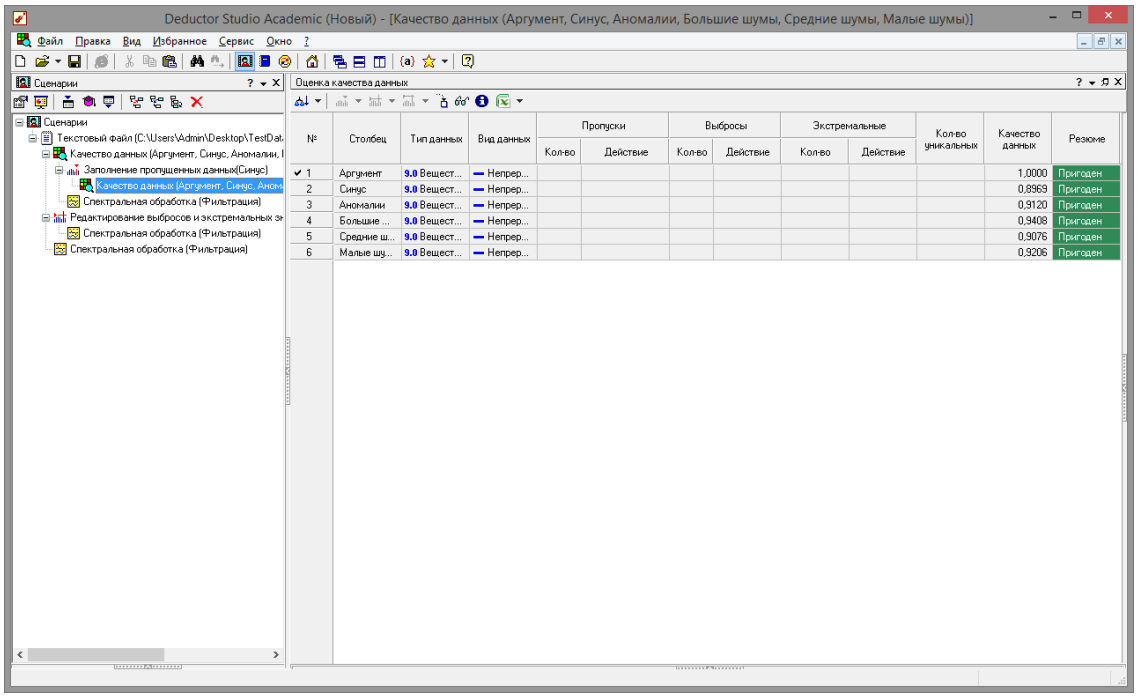


Рис. 1.30 - Вывод мастера после всех внесенных нами изменений

«Грязные данные» представляют собой очень большую проблему. Фактически они могут свести на нет все усилия по анализу данных. Причем, речь идет не о разовой операции, а о постоянной работе в этом направлении. Чисто не там, где не сорят, а там, где убирают.

Описанные выше варианты решения проблем не единственные. Есть еще достаточно много методов обработки, начиная от экспертных систем и заканчивая нейросетями. Главное, суметь грамотно ими воспользоваться. Обязательно нужно

учитывать то, что методы очистки сильно привязаны к предметной области. От сферы деятельности организации и назначения хранилища данных зависит практически все. То, что для одних является шумом для других очень ценная информация. Если у нас будет априорная информация о задаче, то качество очистки данных можно увеличить на порядки.

1.7 Задание на самостоятельную работу

Сгенерировать собственный набор данных провести его анализ и выполнить предобработку данных. Данные сгенерировать в электронной таблице с помощью *MS Excel* с использованием формул и автозаполнений. Готовый файл необходимо сохранить в формате

«**.csv*» (*MS-DOS*) и при импорте в *Deductor*, выбрать в качестве разделителя «Точка с запятой».

Содержание отчёта

1. Описание предметной области
2. Зашумлённые данные (90-100 векторов размерностью не менее 4)
3. Скорректированные данные
4. Краткий порядок обработки
5. Выводы

Контрольные вопросы

1. Для чего служит программа *Deductor Academic*?
2. Зачем нужна предобработка данных?
3. Что такое парциальная предобработка?
4. Что такое вейвлет?
5. Какие данные можно импортировать в программу?

Лабораторная работа 2

Базовые методы интеллектуального анализа данных

Продолжительность работы – 8 час.

Цель работы: ознакомиться с возможностями классификации данных с помощью аналитического пакета *Deductor Academic*.

Программа работы

1. Выполнить классификацию данных с использованием алгоритма *g-mean*.
2. Выполнить классификацию данных с использованием алгоритма *k-mean*.
3. Выполнить классификацию данных с использованием нейронной сети.

Методические указания по выполнению работы

Задача разбиения множества объектов или наблюдений на априорно заданные группы, называемые классами, внутри каждой из которых они предполагаются похожими друг на друга, имеющими примерно одинаковые свойства и признаки. При этом решение получается на основе анализа значений атрибутов (признаков).

Классификация является одной из важнейших задач *Data Mining*. Она применяется в маркетинге при оценке кредитоспособности заемщиков, определении лояльности клиентов, распознавании образов, медицинской диагностике и многих других приложениях. Если аналитику известны свойства объектов каждого класса, то когда новое наблюдение относится к определенному классу, данные свойства автоматически распространяются и на него.

Если число классов ограничено двумя, то имеет место бинарная классификация, к которой могут быть сведены многие более сложные задачи. Например, вместо определения таких степеней кредитного риска, как «Высокий», «Средний» или «Низкий», можно использовать всего две - «Выдать» или

«Отказать».

Для классификации в *Data Mining* используется множество различных моделей: нейронные сети, деревья решений, машины опорных векторов, метод k-ближайших соседей, алгоритмы покрытия и др., при построении которых применяется обучение с учителем, когда выходная переменная (метка класса) задана для каждого наблюдения. Формально классификация производится на основе разбиения пространства признаков на области, в пределах каждой из которых многомерные векторы рассматриваются как идентичные. Иными словами, если объект попал в область пространства, ассоциированную с определенным классом, он к нему и относится.

Рассмотрим классификацию данных на примере Ирисов Фишера. Ирисы Фишера - это набор данных для задачи классификации, на примере которого Рональд Фишер в 1936 году продемонстрировал работу разработанного им метода дискриминантного анализа. Этот набор данных стал уже классическим, и часто используется в литературе для иллюстрации работы различных статистических алгоритмов. Ирисы Фишера состоят из данных о 150 экземплярах ириса, по 50 экземпляров из трёх видов - Ирис щетинистый (*Iris setosa*), Ирис виргинский (*Iris virginica*) и Ирис разноцветный (*Iris versicolor*). Для каждого экземпляра измерялись четыре характеристики (в сантиметрах):

- длина чашелистика (англ. *sepal length*);
- ширина чашелистика (англ. *sepal width*);
- длина лепестка (англ. *petal length*);
- ширина лепестка (англ. *petal width*).

На основании этого набора данных требуется построить правило классификации, определяющее вид растения по данным измерений. Это задача многоклассовой классификации, так как имеется три класса - три вида ириса.

2.1 Классификация данных с использованием алгоритма «*g-mean*»

Импортируем в программу данные из файла «Ирисы.txt». Для начала попробуем провести классификацию ирисов, встроенным

методом кластеризации *g-mean* (рис. 2.1). Интересно то, что обучение будет проходить без учителя, т.е. выходные данные не будут указаны.

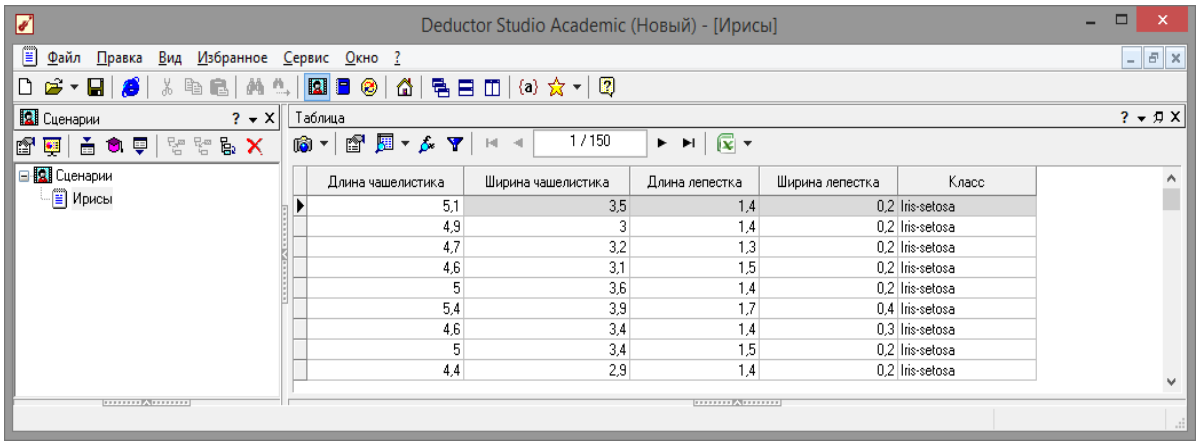


Рис. 2.1 - Импортированные данные

Более того, не будут даже указано количество кластеров, на которое необходимо разделить данную выборку, проверим эффективность встроенной системы кластеризации, для этого необходимо выбрать в мастере обработок пункт кластеризация (рис. 2.2).

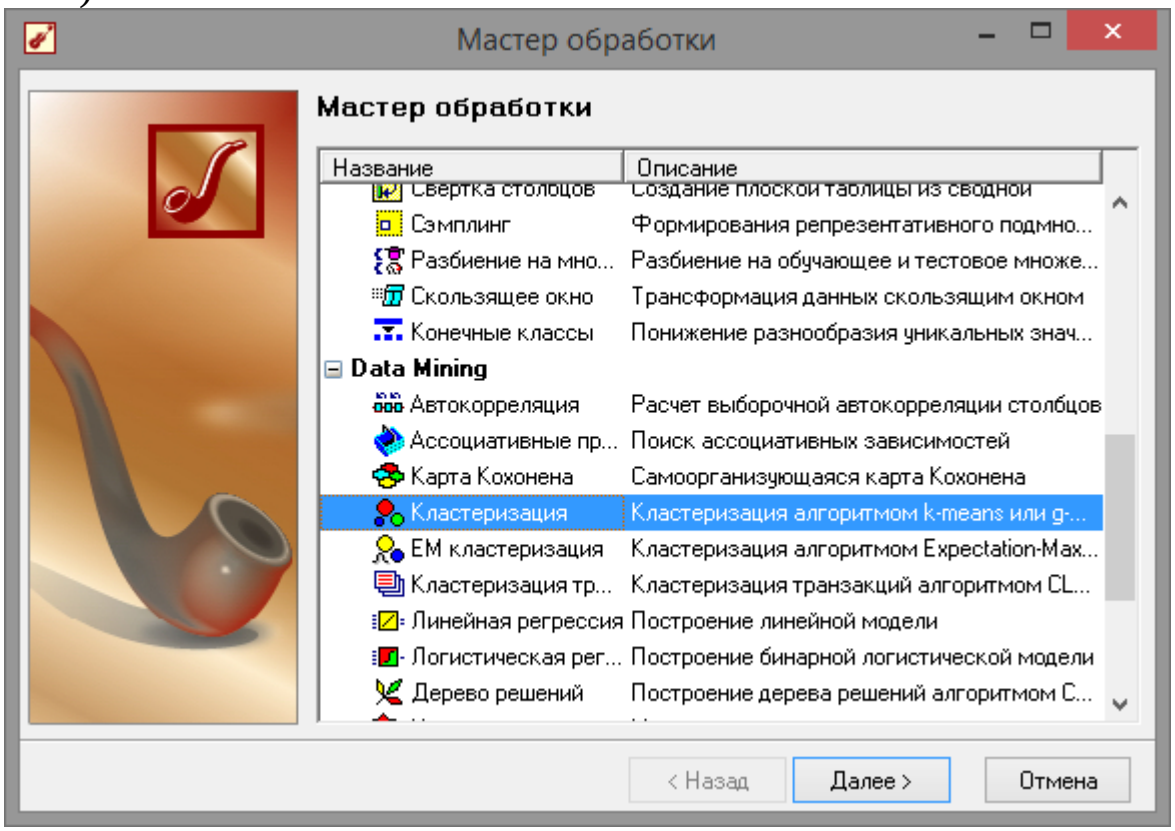


Рис. 2.2 - Мастер кластеризации

На следующем шаге (рис. 2.3) со столбца «Класс» уберем

назначение выходное и поставим информационное, теперь обучение будет происходит без учителя.

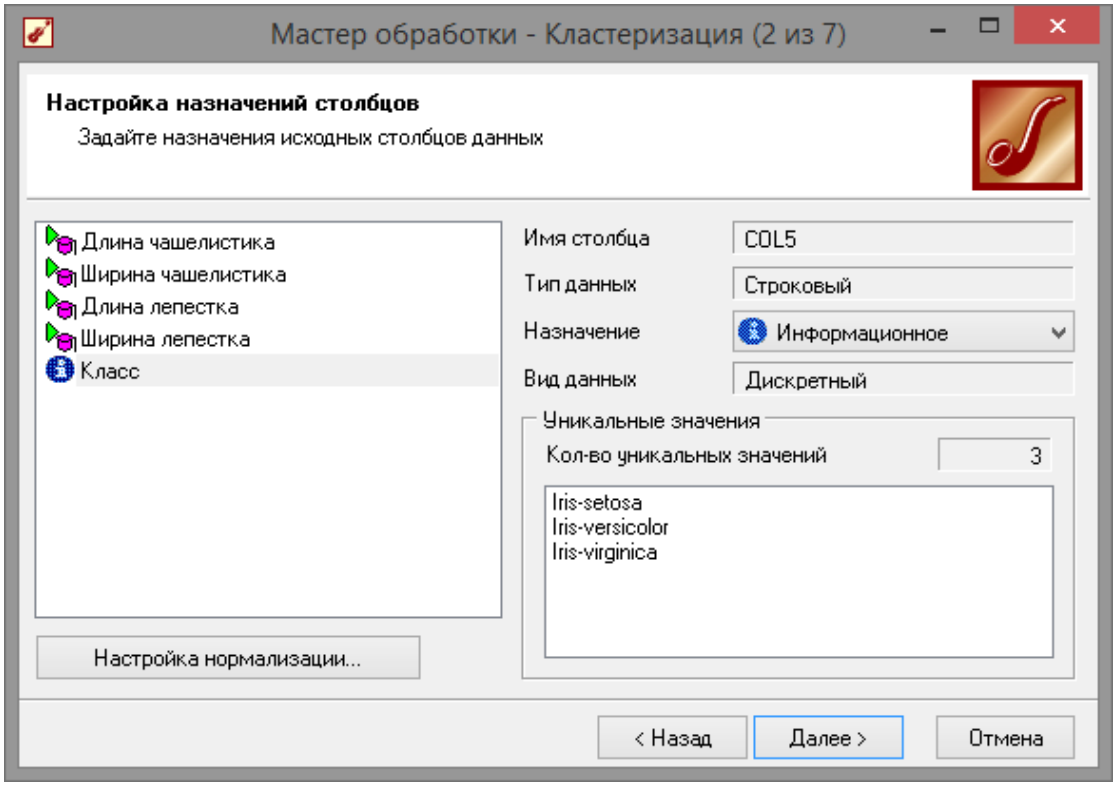


Рис. 2.3 - Настройка мастера кластеризации

Далее следует настроить параметры обучения, в данном конкретном случае, параметры по умолчанию отлично подходят (рис. 2.4).

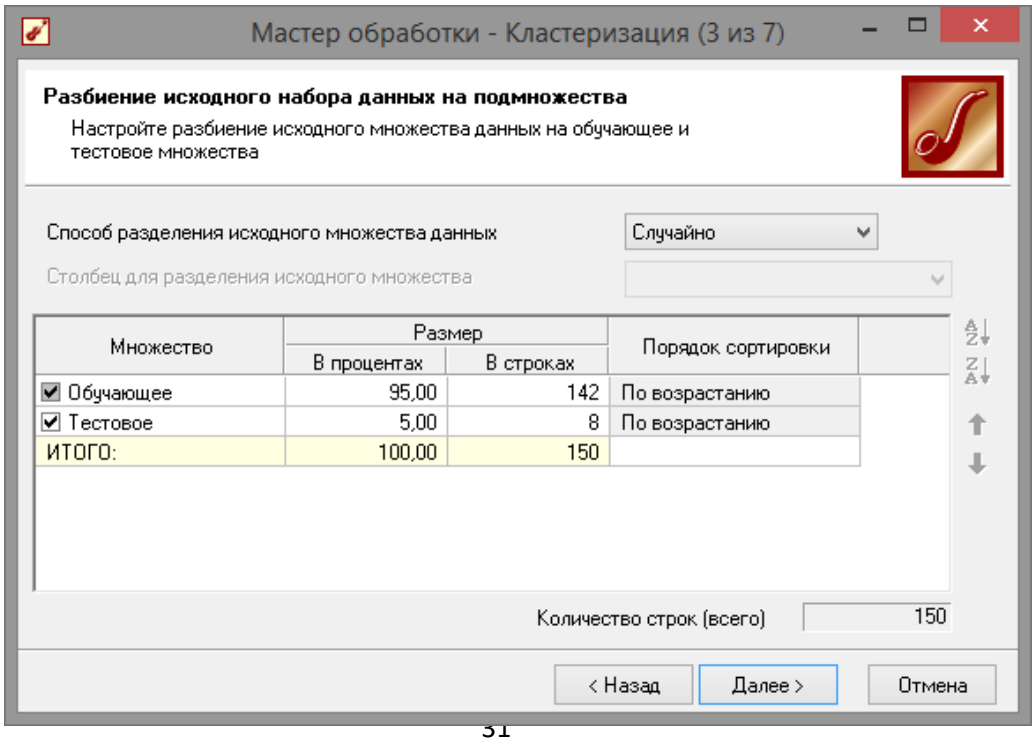


Рис. 2.4 - Настройка параметров обучения

На следующем шаге мастера необходимо выбрать алгоритм кластеризации. На и так заранее известно, что количество кластеров должно равняться «3», но протестируем возможности программы и предоставим ей самой выбрать количество кластеров. Это так же будет полезно, если не известно на какое количество групп следует разбивать выборку (рис. 2.5).

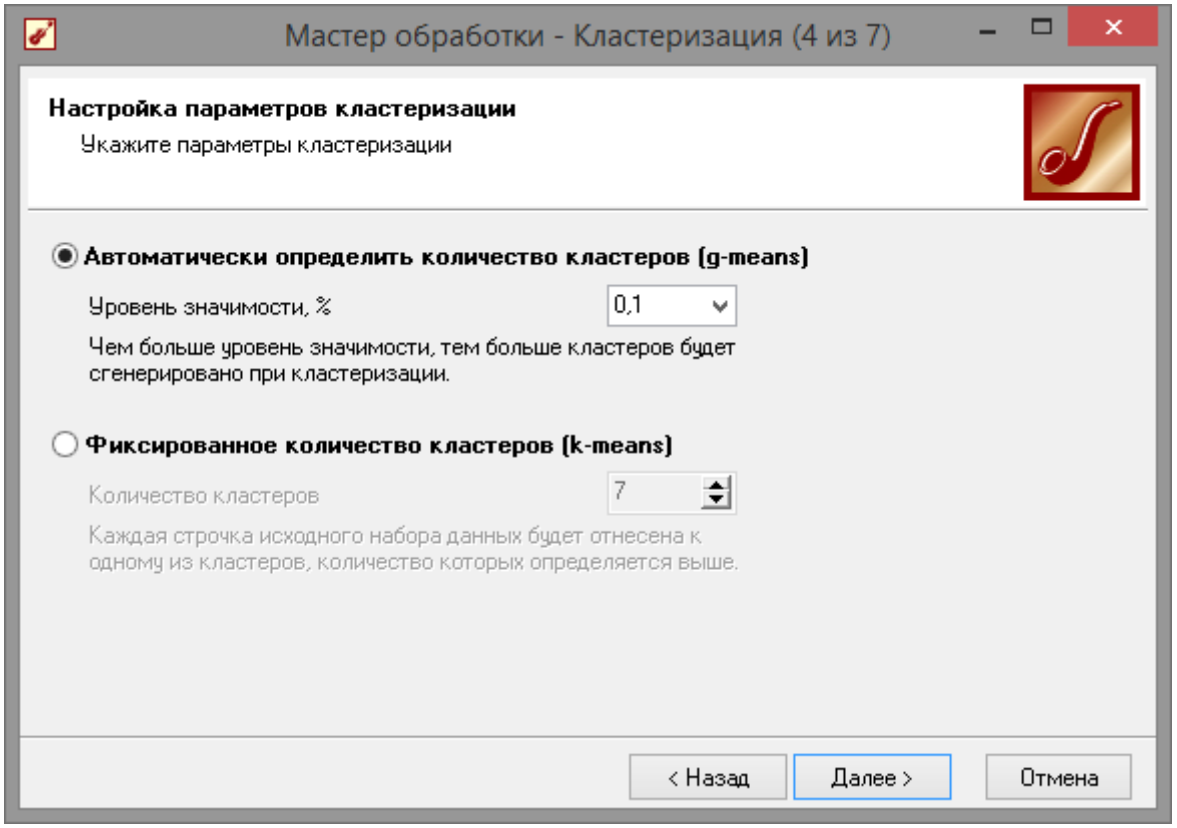


Рис. 2.5 - Выбор алгоритма кластеризации

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 (рис. 2.6).

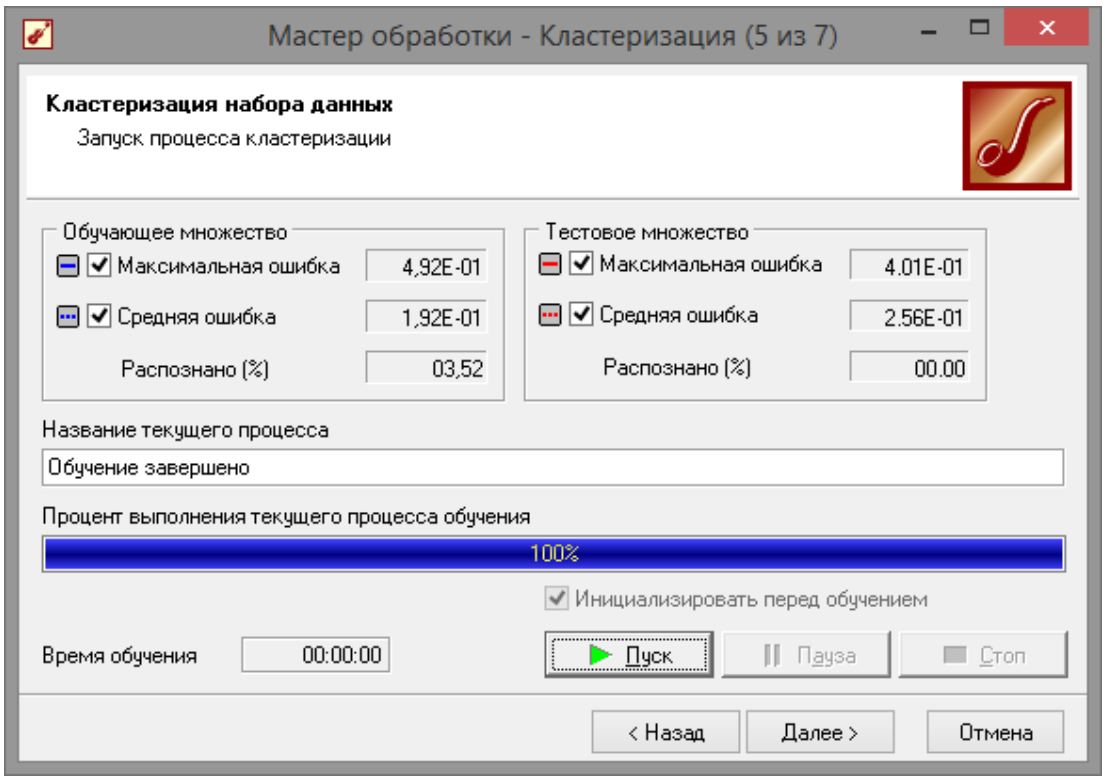


Рис. 2.6 - Обучение сети

После обучения сети, в качестве визуализаторов выберем: «Связи кластеров»; «Профили кластеров»; «Матрицу сравнения»; «Что-если» (рис. 2.7).

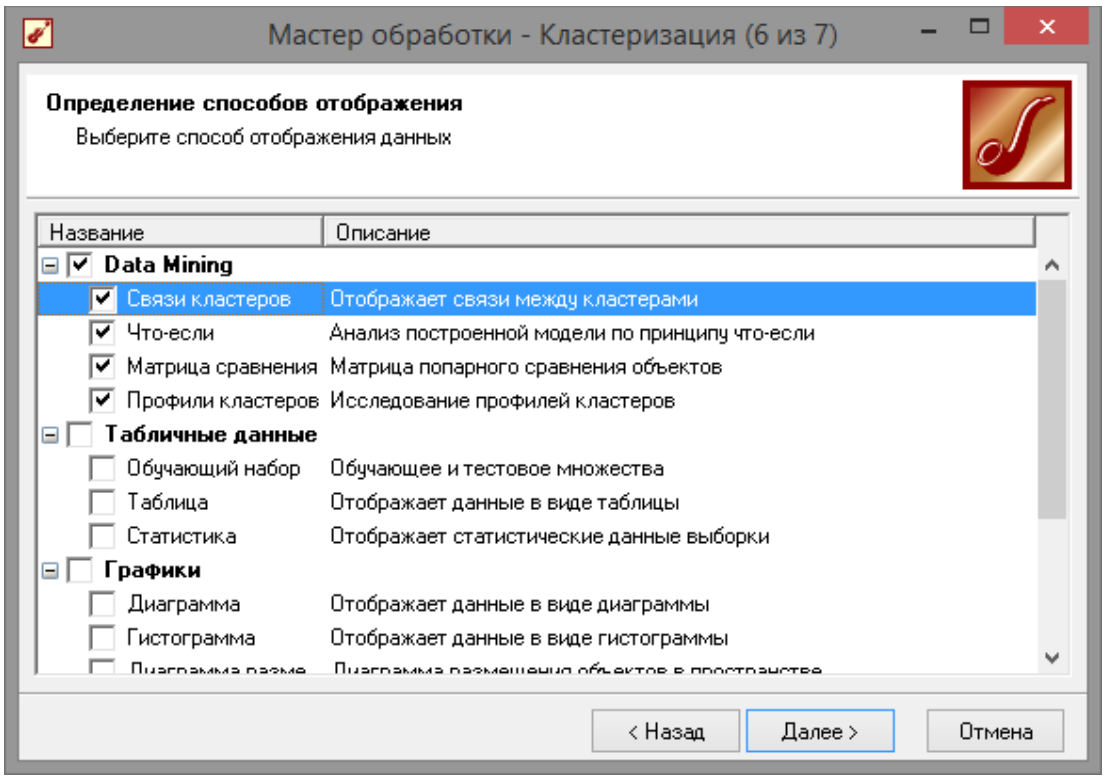


Рис. 2.7 - Визуализация данных

Как видно из рис. 2.8 приложение верно переделило количество групп, также здесь можно увидеть на сколько значим тот или иной параметр, для присвоения цветку того или иного вида.

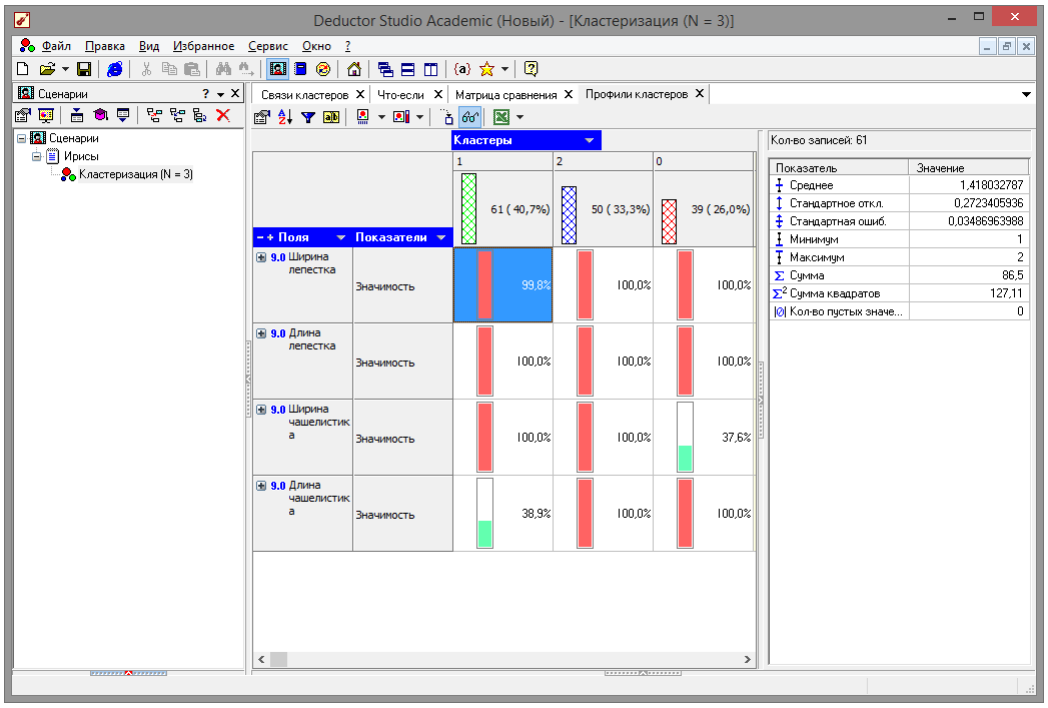


Рис. 2.8 - Профили кластеров

На матрице сравнения (рис. 2.9) видно что больше всех от остальных отличается «кластер 2» (ему соответствует «*iris-setosa*»).

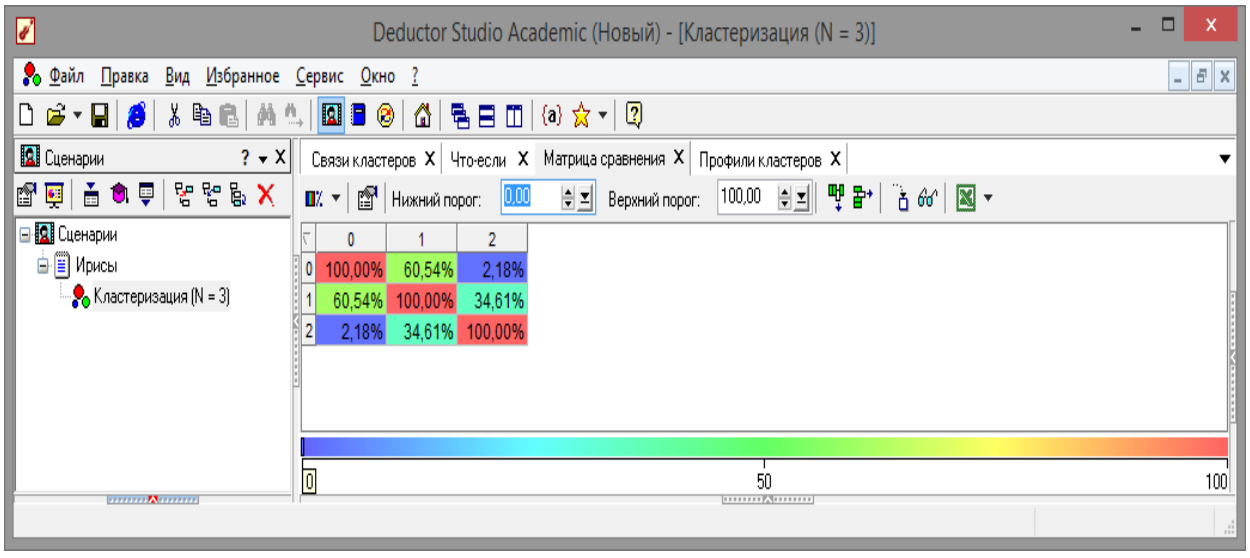


Рис. 2.9 - Матрица сравнения

Визуализатор «Что-если» (рис. 2.10) позволит провести эксперимент, введя любые значения параметров. Если же нажать кнопку «Загрузить данные» из исходной выборки, то можно

заметить неточности определения кластеров из-за схожих параметров цветков.

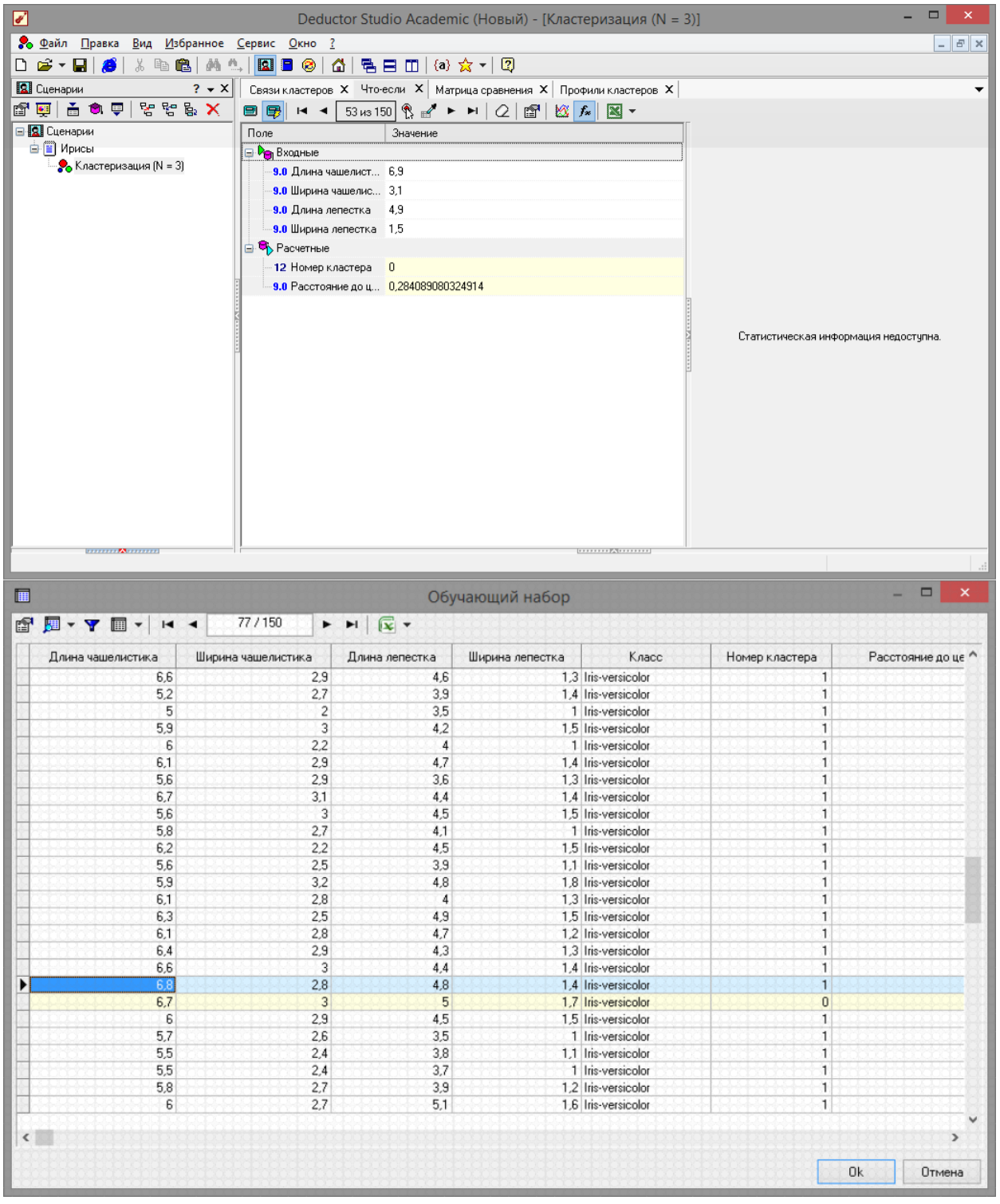


Рис. 2.10 - Инструмент «Что-Если»

2.2 Классификация данных с помощью нейронной сети

Теперь опробуем провести классификацию ирисов при помощи обычной нейронной сети. Входные данные брать из файла

«Ирисы.txt», только в данном случае обучение будет происходить с учителем. Импортируем данные и в мастере обработок выберем пункт нейросеть (рис. 2.11).

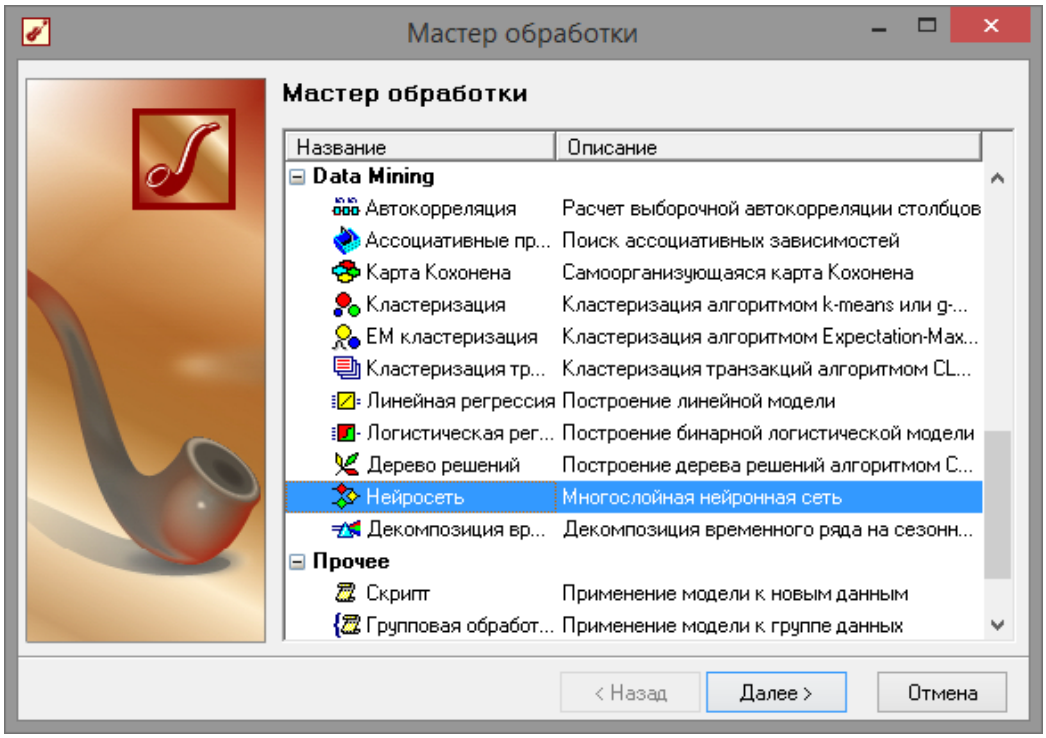


Рис. 2.11 - Мастер обработок

Установим в качестве выходного значения параметр класс. В качестве нормализатора для выхода «Класс» необходимо установить пункт «Уникальные значения».

Примечание. Уникальные значения используются для дискретных значений. Такими являются строки, числа или даты, заданные дискретно. Чтобы привести непрерывные числа в дискретные, можно, например, воспользоваться обработкой

«Квантование». Так следует поступать для величин, для которых можно задать отношение порядка, то есть, если для двух любых дискретных значений можно указать, какое больше, а какое меньше. Тогда все значения необходимо расположить в порядке возрастания (рис. 2.12-2.13). Далее они нумеруются по порядку, и значения заменяются их порядковым номером

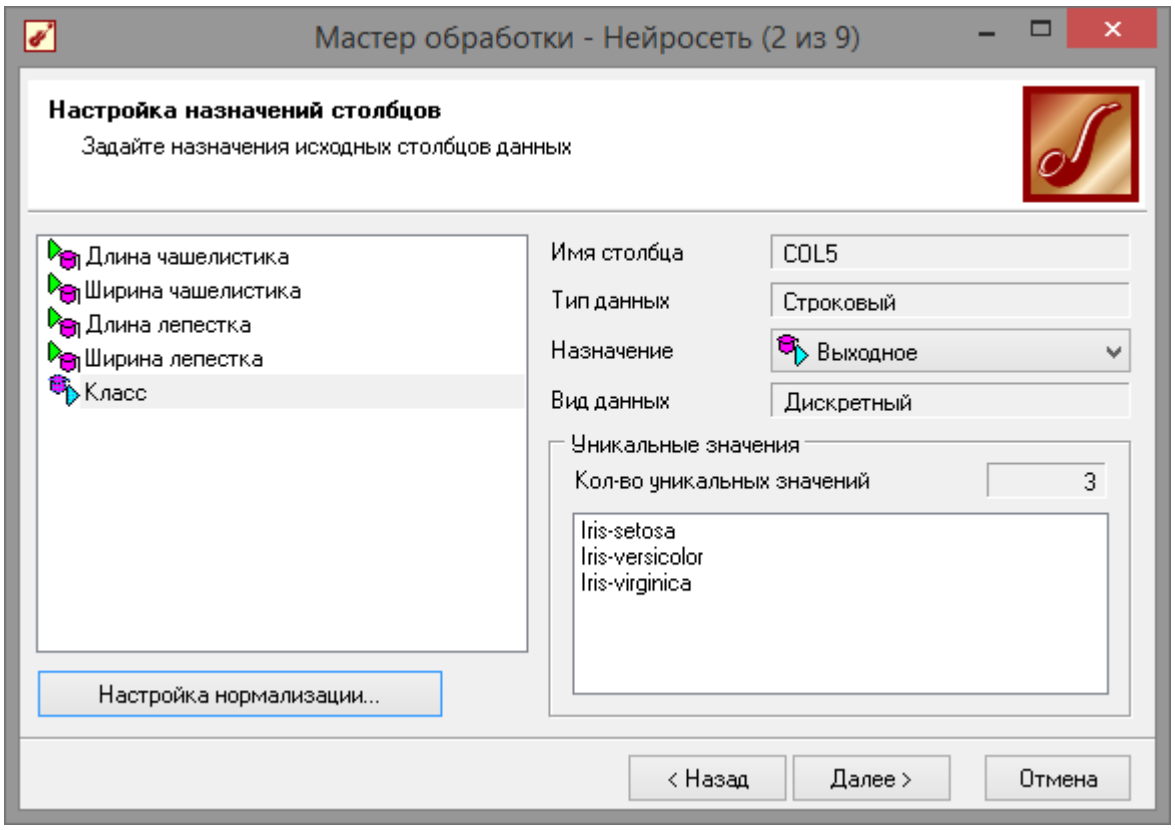


Рис. 2.12 - Назначение входов и выходов

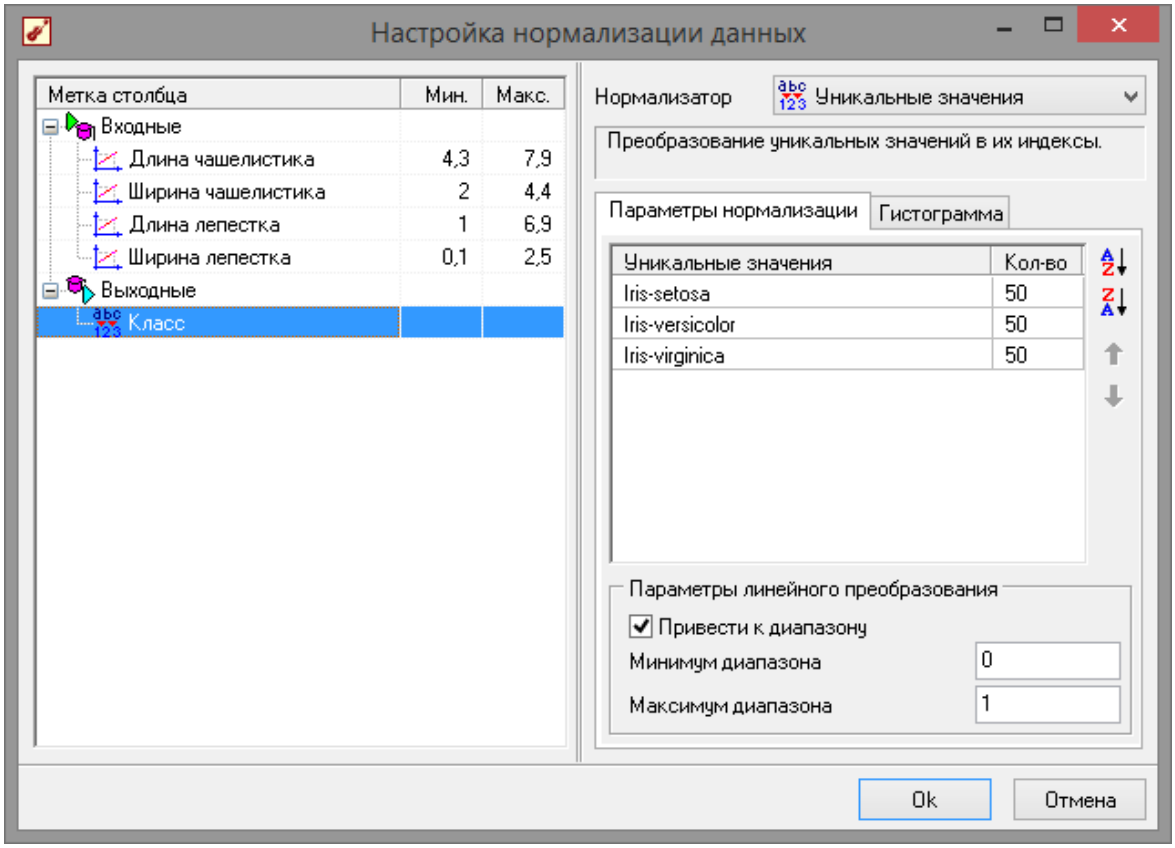


Рис. 2.13 - Настройка нормализации

Следующим шагом указываем параметры обучения (рис. 2.14).

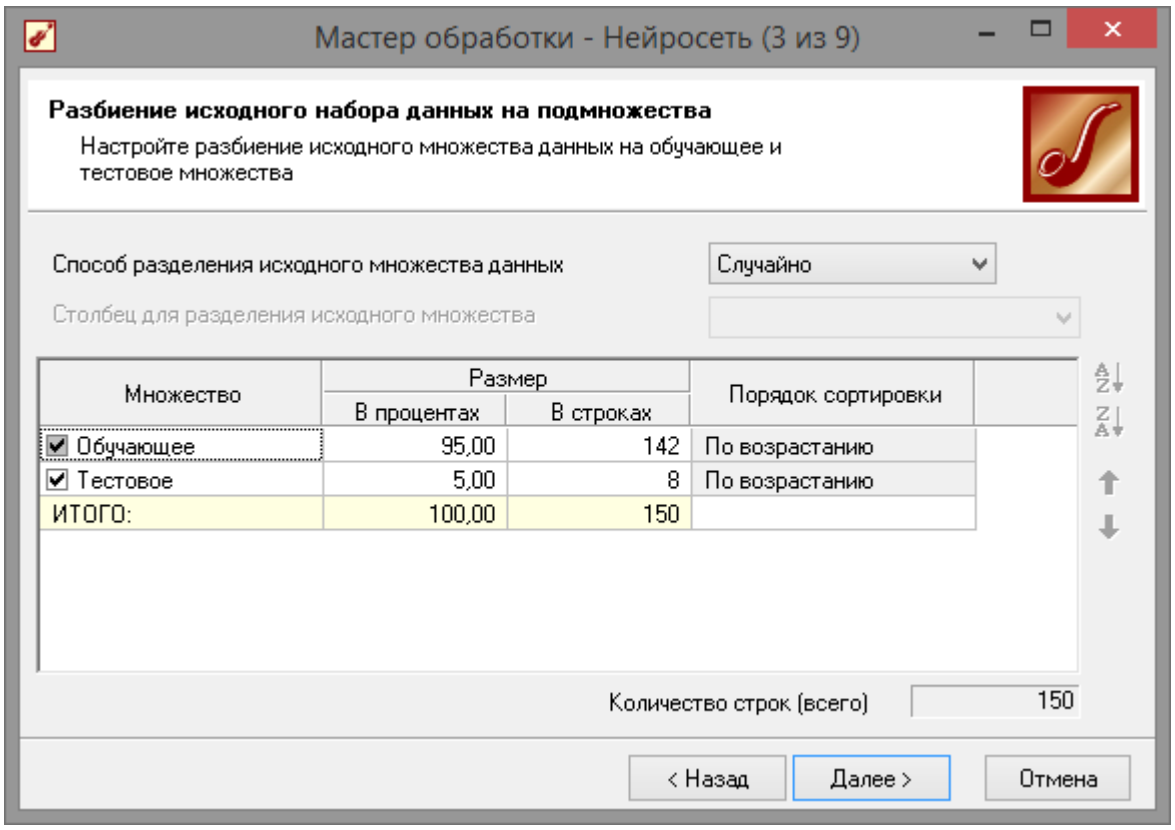


Рис. 2.14 - Параметры обучения

Количество нейронов первого слоя экспериментально было выставлено в значение 5 (рис. 2.15).

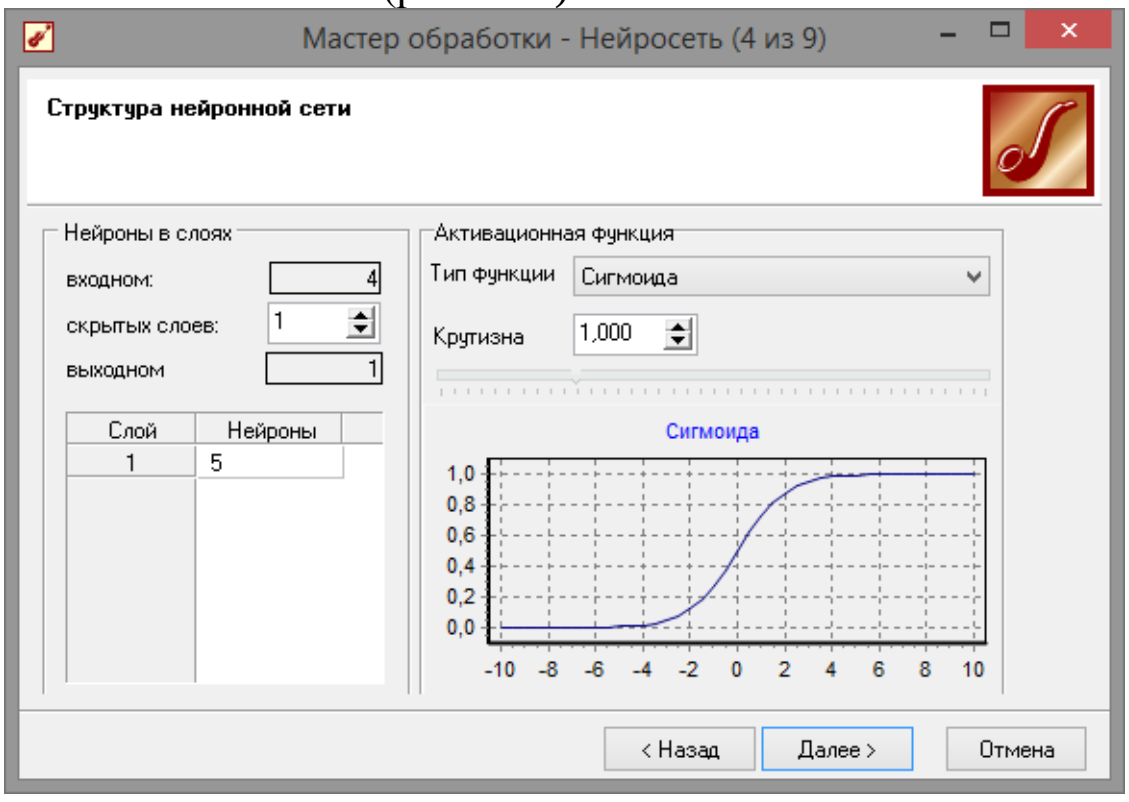


Рис. 2.15 - Параметры обучения

Далее выбираем алгоритм обучения (рис. 2.16).

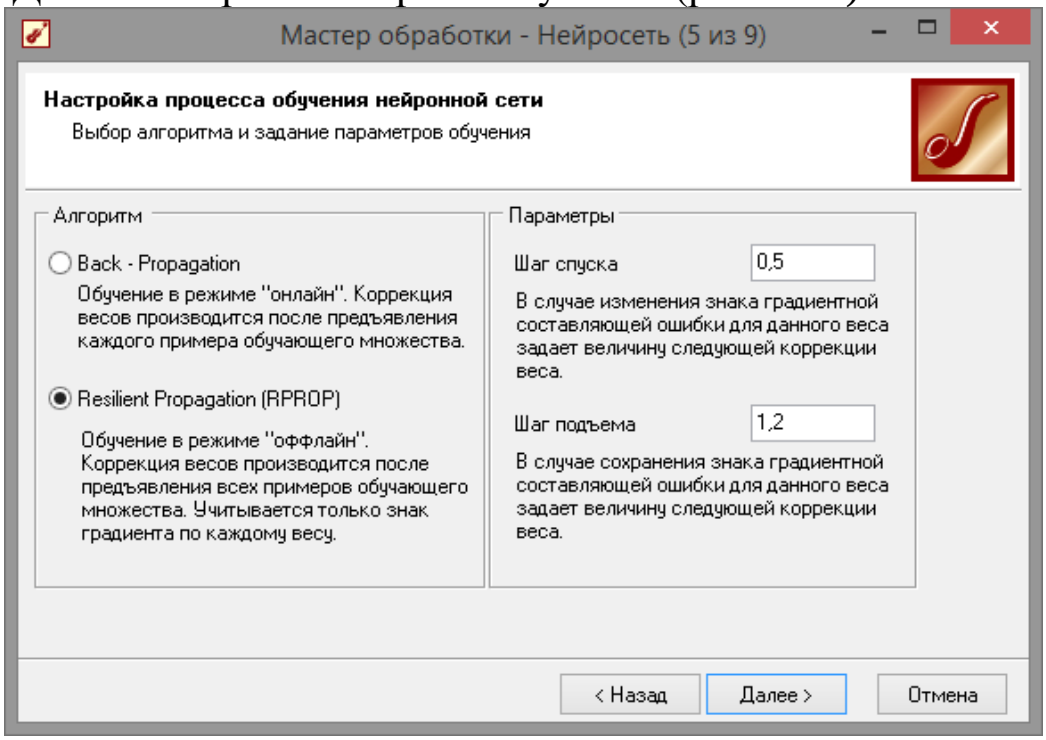


Рис. 2.16 - Алгоритм обучения

На следующем шаге (рис. 2.17) устанавливаем значение погрешности и количество эпох обучения. И запускаем обучение сети (рис. 2.18).

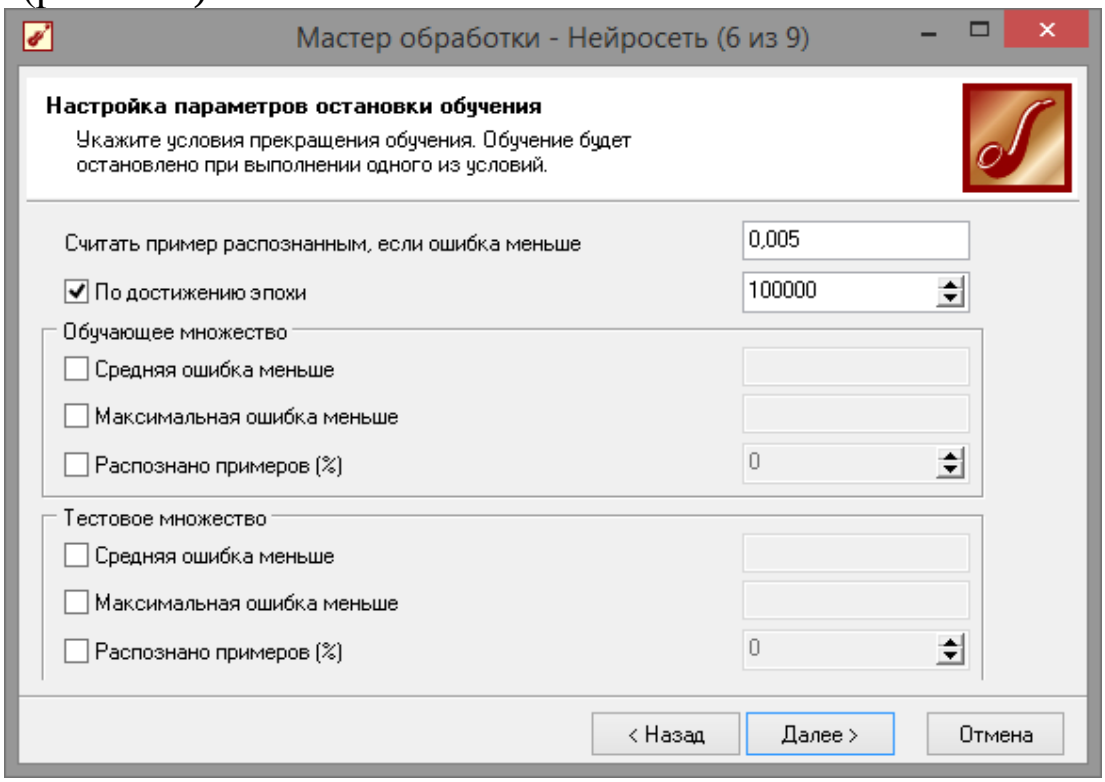


Рис. 2.17 - Установка погрешности и числа эпох обучения

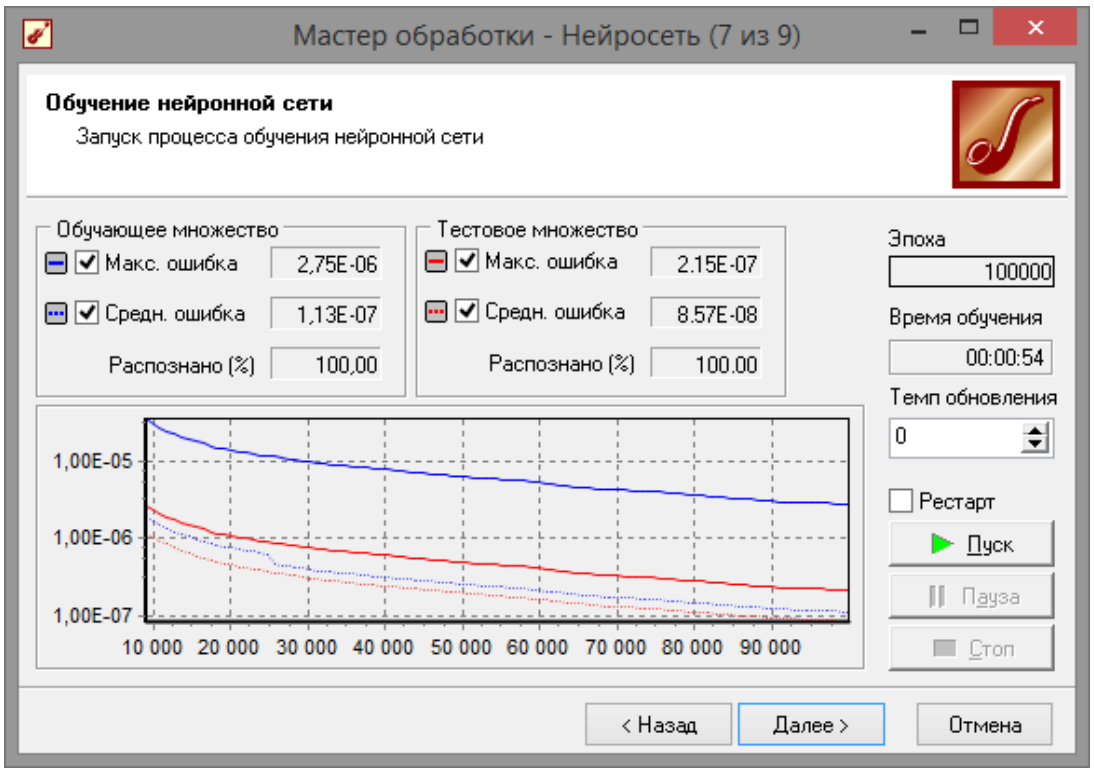


Рис. 2.18 - Обучение сети

Как видно из рис.2.18, сеть обучалась дольше, но дала более точные результаты. В качестве визуализаторов выбираем «Граф нейросети», далее пункт «Таблица сопряжения» и «Что-Если» (рис. 2.19).

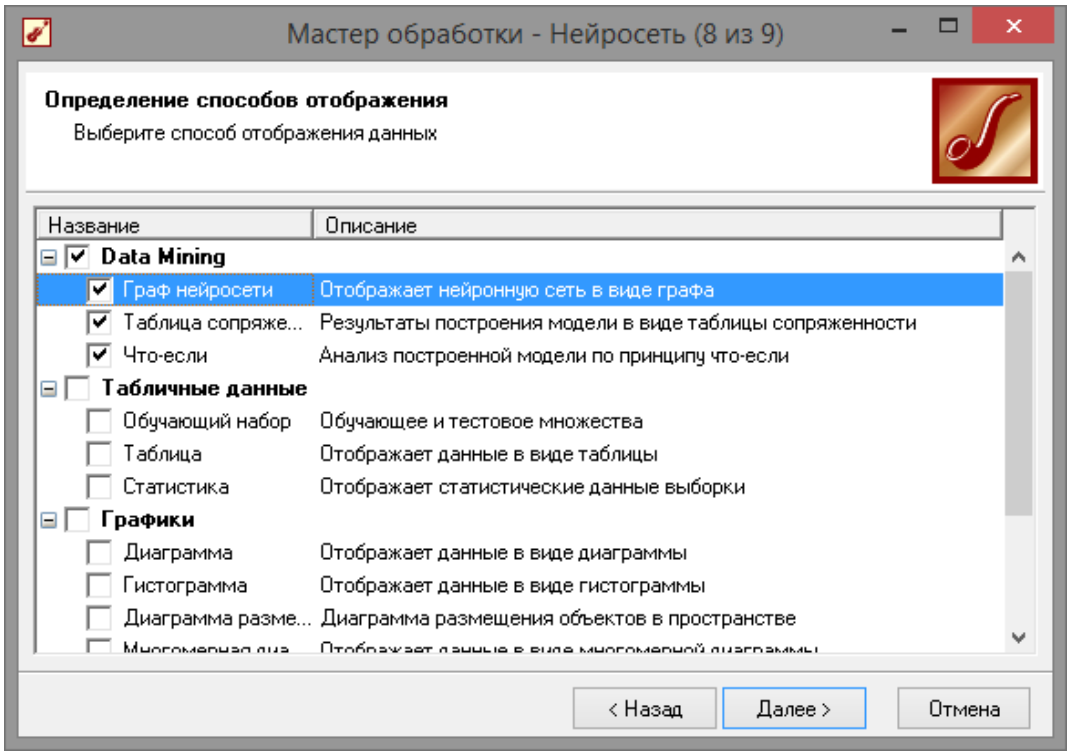


Рис. 2.19 - Визуализаторы

На графе нейросети видно, как выглядит обученная сеть (рис. 2.20).

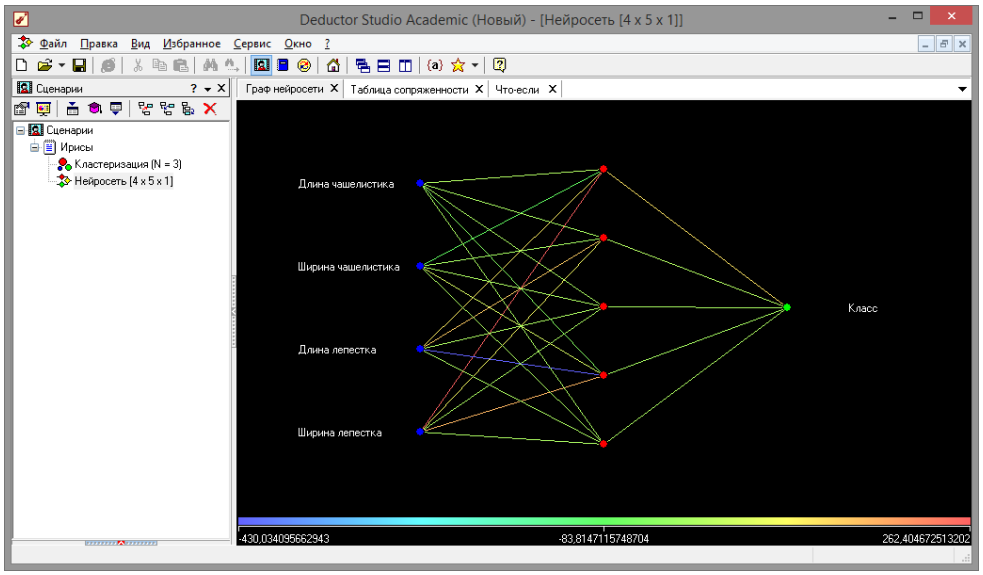


Рис. 2.20 - Граф нейросети

Диаграмма сопряженности (рис. 2.21) показывает как распределились входные значения. На ней видно что сеть обучилась абсолютно точно.

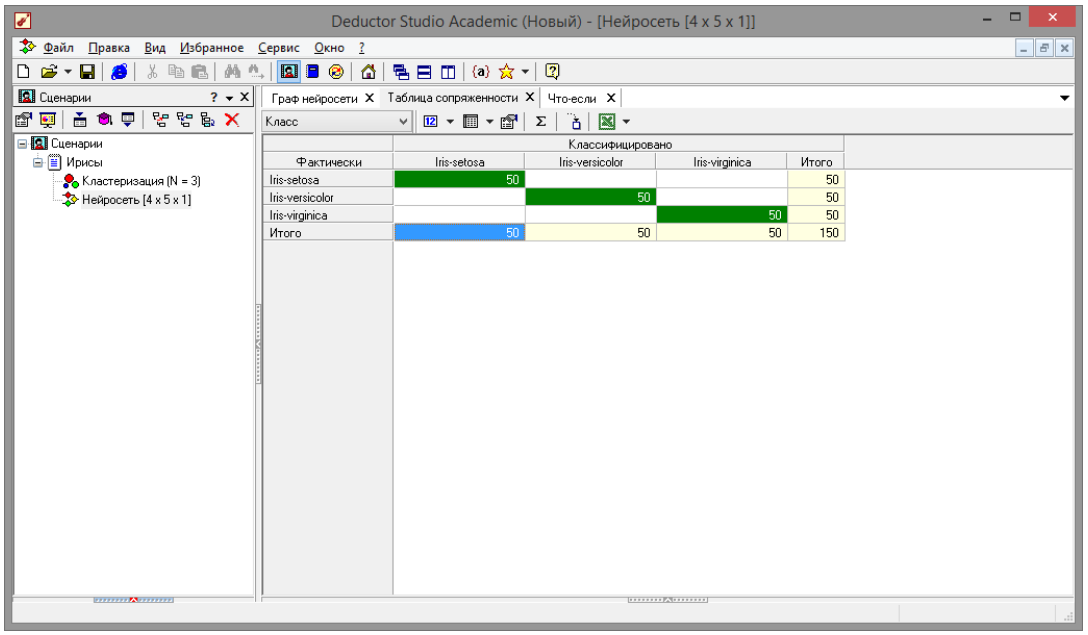
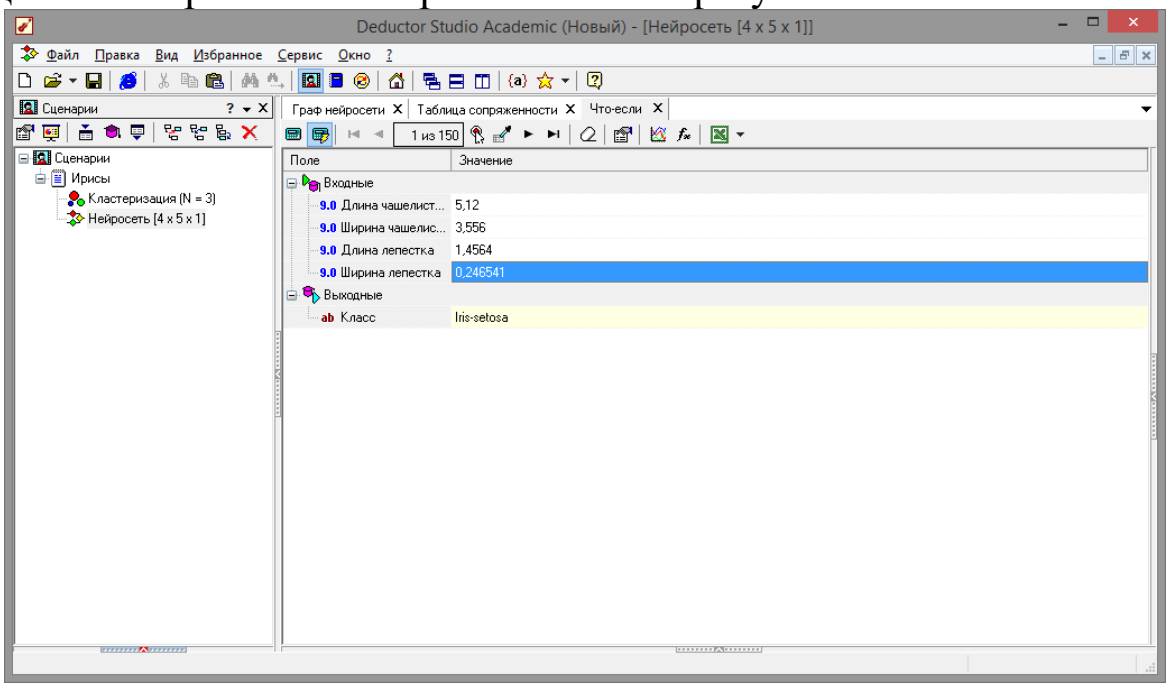


Рис. 2.21 - Диаграмма сопряженности

Убедимся в высказанном выше заключении взглянув на таблицу «Что-Если» (рис. 2.22). Даже при вводе значений отсутствующих в выборке, сеть верно реагирует на них. Далее можно опять нажать на

кнопку «Загрузить данные из исходной выборки» чтобы убедиться в правильности распознавания результатов.



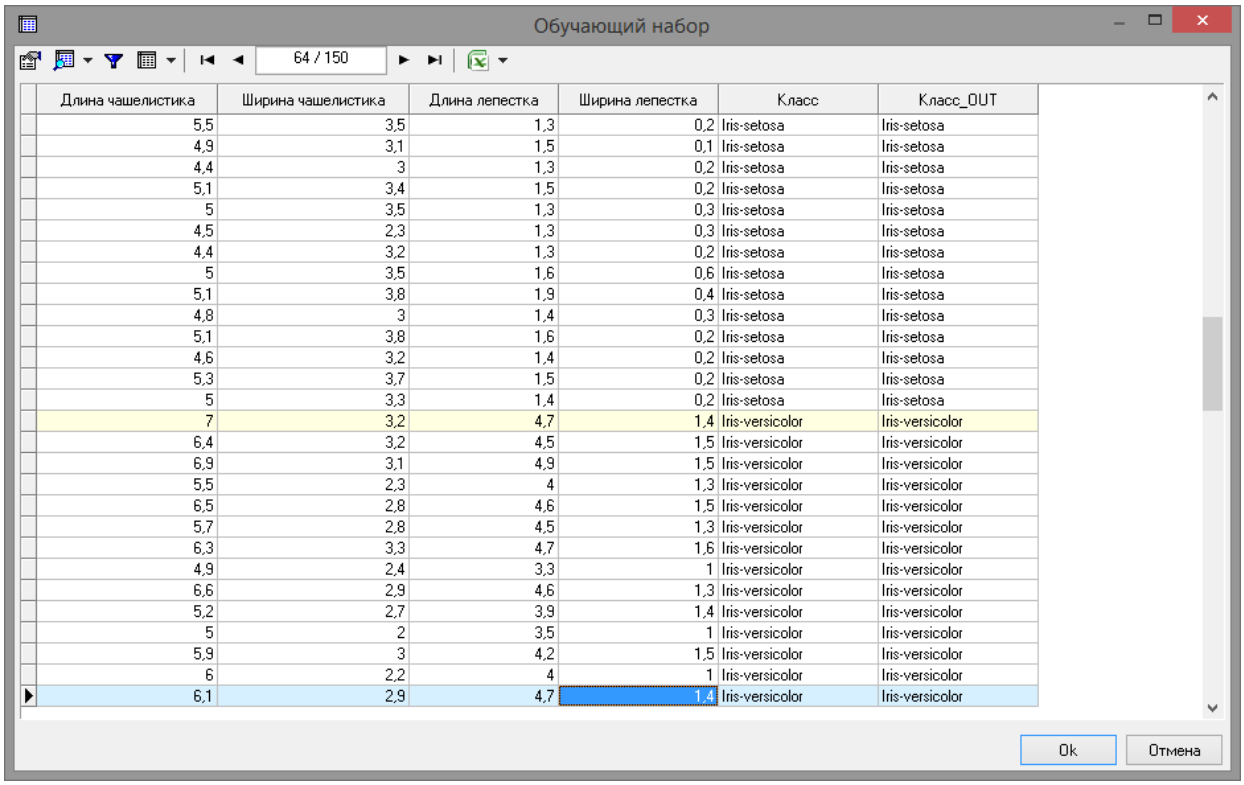


Рис. 2.22 - Инструмент «Что-Если»

Данный пример показал, как можно обучить сеть с учителем. Был изучен новый инструмент «Нейросеть» и сделано предположение, что обучение с учителем более эффективно, хотя и занимает больший промежуток времени.

Пример показал простоту и удобство применения «Нейросеть» для классификации Ирисов Фишера. Мастер предлагает широкие возможности по настройке процесса обучения. После обучения сети стали видны ее достоинства для анализа. Также были продемонстрированы широкие возможности визуализации простроенной сети. Все это говорит о эффективности инструмента

«Нейросеть».

2.3 Задания

1. Сгенерировать данные для классификации, провести классификацию двумя способами (инструмент «Кластеризация» и

«Нейронная сеть»), сделать выводы по эффективности этих двух способов.

2. Провести эксперимент по изменению параметров обучения нейросети, и сделать выводы по эффективности процесса обучения при разном количестве нейронов.

3.

Контрольные вопросы

1. Для чего служат алгоритмы g-mean и k-mean?
2. Какие алгоритмы обучения нейронной сети предлагает программный комплекс Deductor Academic?
3. В чем их отличие?
4. Что такое обучение с учителем?
5. Что такое обучение без учителя?

Лабораторная работа 3

Применение интеллектуального анализа данных в задачах поддержки принятия решений

Цель работы: освоить методы и средства прогнозирования в пакете *Deductor Academic* при интеллектуальном анализе данных

в задачах поддержки принятия решений.

Программа работы

1. Выполнить пример прогнозирования с помощью нейронных сетей в пакете Deductor Academic.
2. Выполнить пример прогнозирования с помощью временных рядов в пакете Deductor Academic.
3. Выполнить прогнозирование с помощью нейронных сетей и временных рядов на данных согласно индивидуальному заданию.

Методические указания по выполнению работы

Основное направление программы *Deductor Studo* – анализ, прогнозирование, классификация и кластеризация данных. Программа предоставляет следующие механизмы анализа: нейронные сети, линейный регрессионный анализ, построение деревьев решений, самоорганизующиеся карты Кохонена, прогнозирование временного ряда, обнаружение дубликатов и противоречий. Нейросети – механизм, который используют для прогнозирования и решения задач классификации. Они применяются в основном там, где существует нелинейные зависимости результата от входных факторов.

3.1 Прогнозирование умножения с помощью нейронных сетей

Рассмотрим прогнозирование с помощью нейронных сетей на примере прогнозирования результата умножения двух чисел – файл

«Произведение.txt». В нем содержится таблица со следующими полями: «АРГУМЕНТ1», «АРГУМЕНТ2» – множители,

«ПРОИЗВЕДЕНИЕ» – их произведение. Причем произведение некоторых чисел пропущено в обучающей выборке, например, 7x7=49. Импортировав данные из файла, можно посмотреть результат умножения, используя таблицу.

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, показанный на рис. 3.1. 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 и задать тип данных как «Целый» (рис. 3.2).

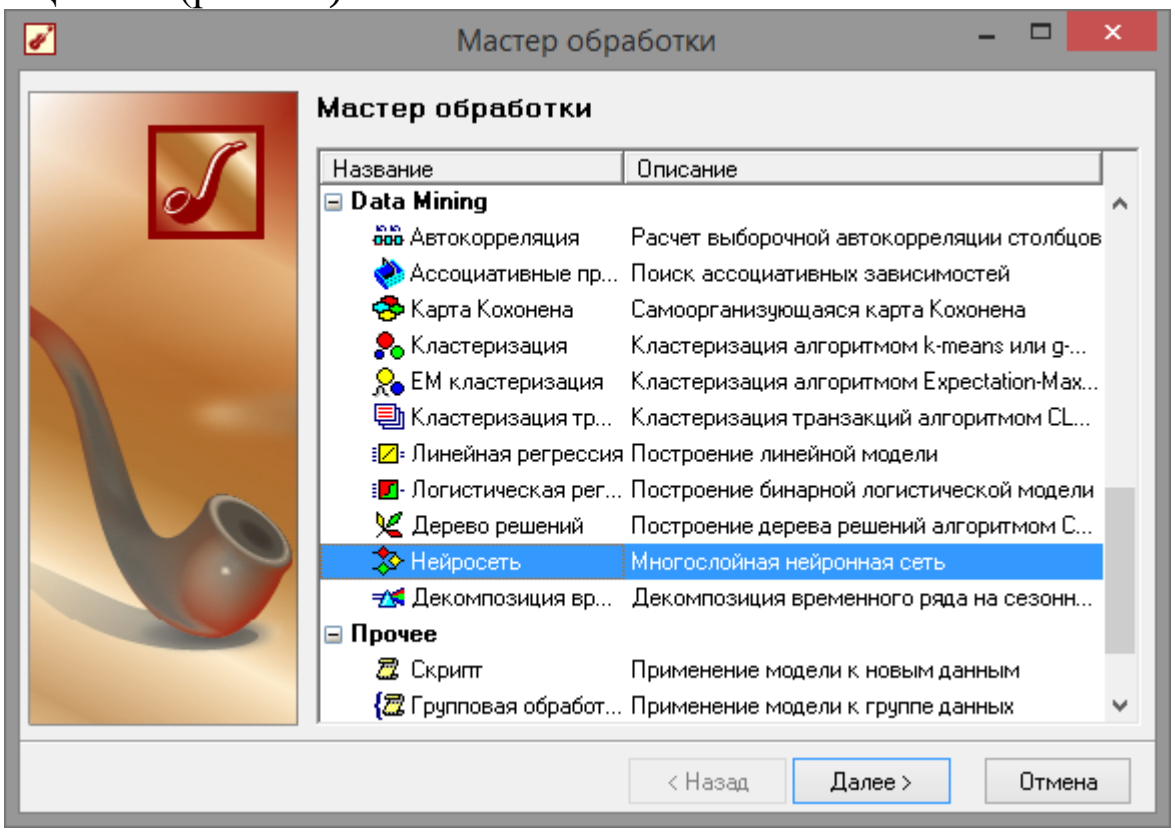


Рис. 3.1 - Мастер обработки

На следующем шаге предлагается настроить разбиение исходного множества данных на обучающее и тестовое. Здесь необходимо указать способ разбиения исходного множества данных

«Случайно» и поставить обучающее множество размером 100 %, так как выборка слишком мала (рис. 3.3).

Далее необходимо указать функцию активации, количество скрытых слоев, и количество нейронов в слое. Данные необходимо подбирать экспериментально, так как большое количество нейронов и слоев, может сильно замедлить процесс обучения. Эмпирически были выставлены настройки, показанные на рис. 3.4.

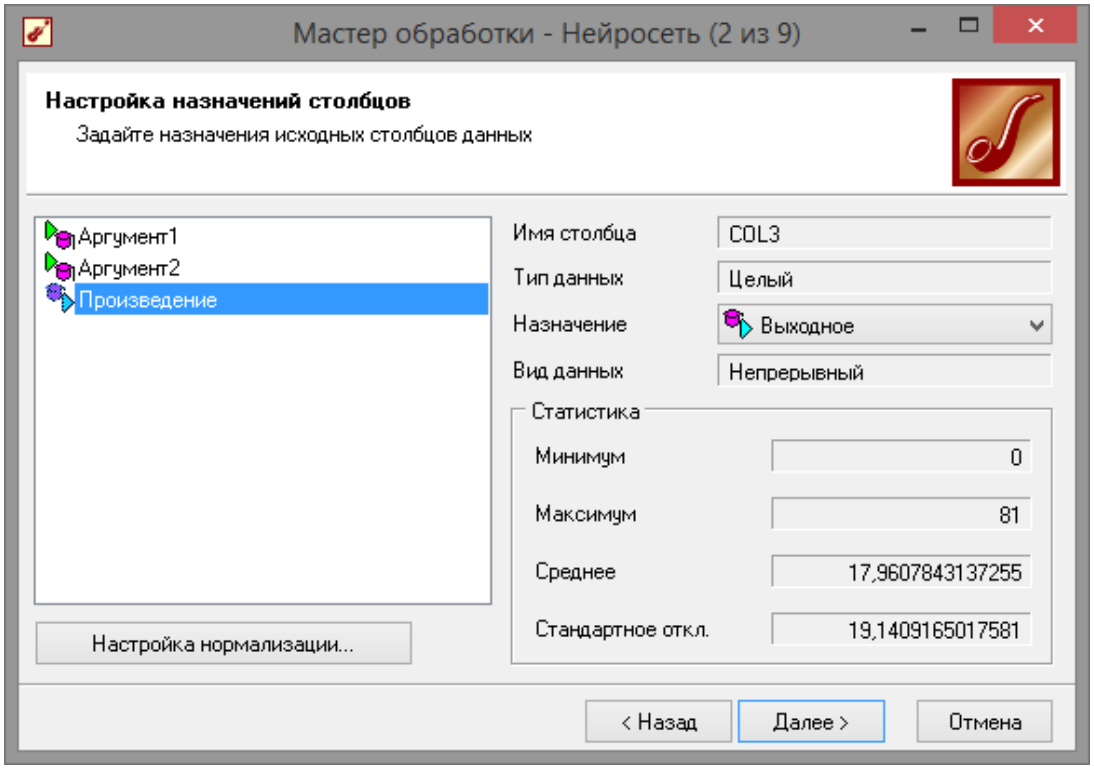


Рис. 3.2 - Мастер нейросети

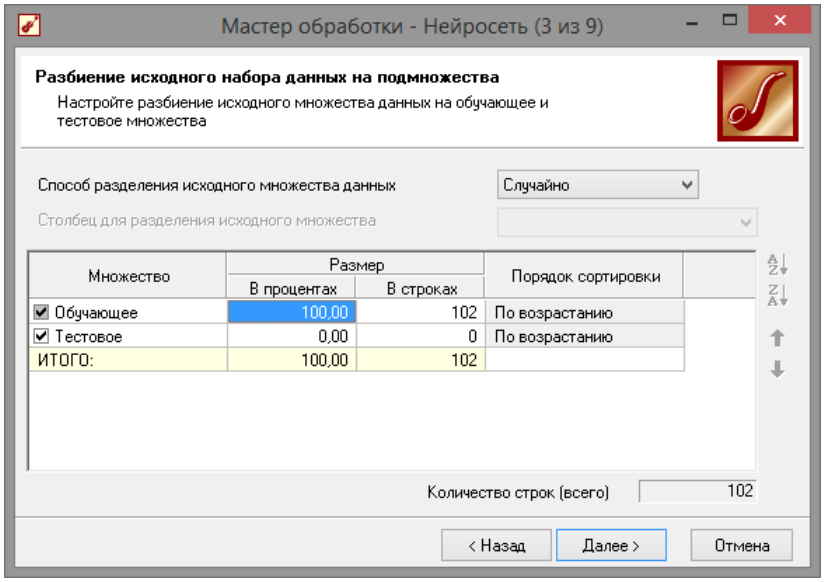


Рис. 3.3 - Настройки параметров обучения

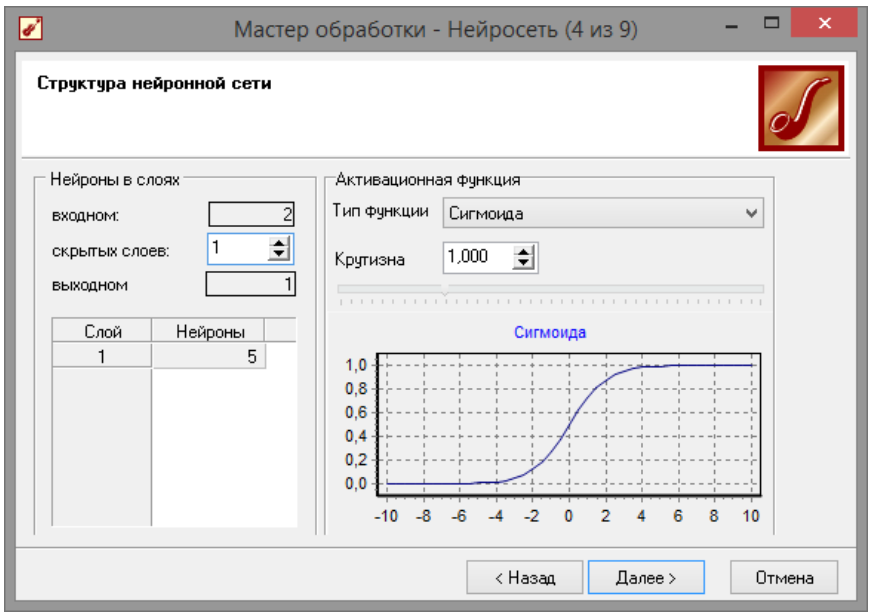


Рис. 3.4 - Структура нейронной сети

Следующий шаг предлагает выбрать алгоритм обучения и его параметры (рис. 3.5).

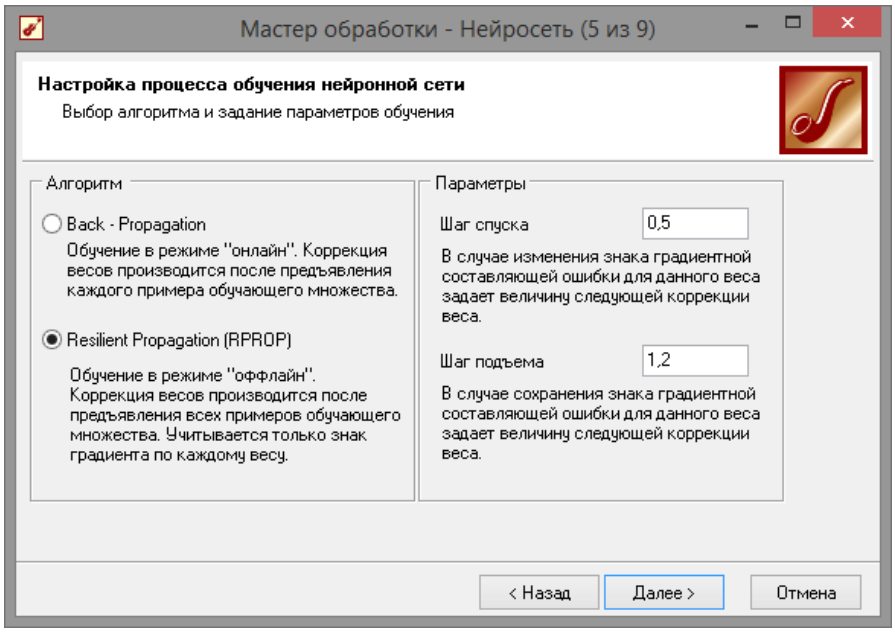


Рис. 3.5 - Алгоритм обучения

Далее настроим условия остановки обучения (рис. 3.6). Пусть пример считаем распознанным, если ошибка меньше 0,005. Также укажем условие остановки обучения при достижении эпохи 100000.

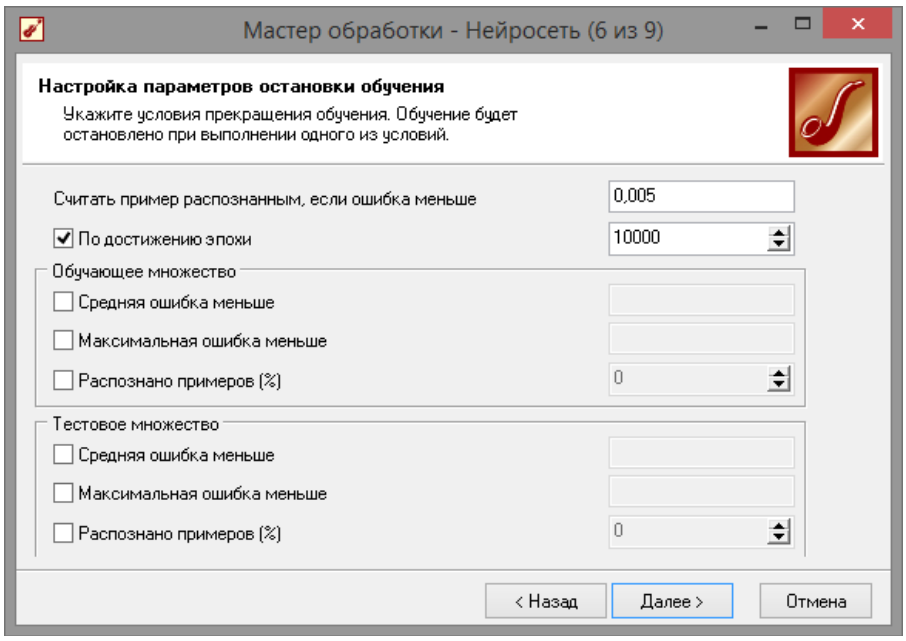


Рис. 3.6 - Настройка параметров остановки обучения

Следующий шаг мастера (рис. 3.7) 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.

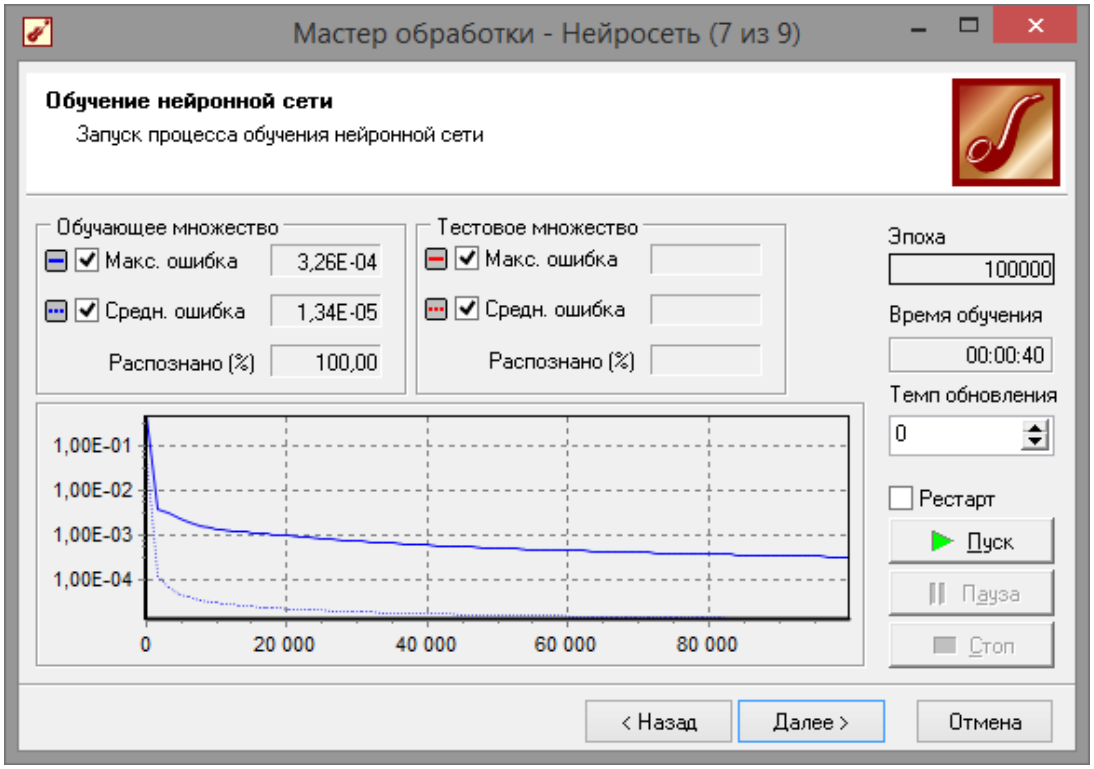


Рис. 3.7 - Обучение сети

После обучения сети, в качестве визуализаторов выберем

варианты, показанные на рис. 3.8: «Диаграмма», «Диаграмма рассеяния», «Граф нейросети», «Что-если».

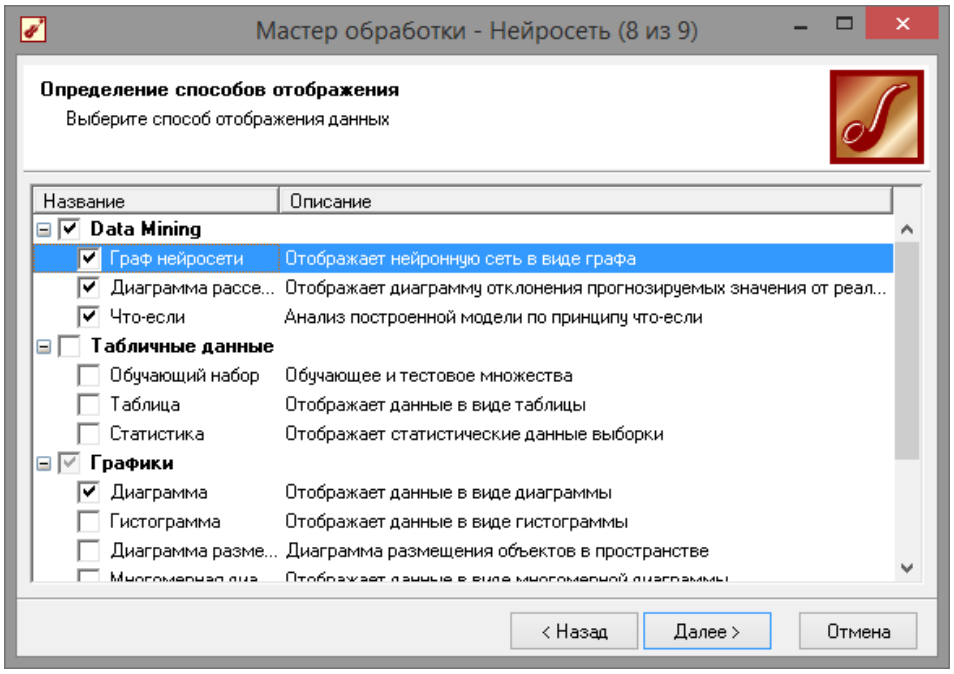


Рис. 3.8 - Визуализация данных

Результаты наглядно видны на диаграмме рассеяния (рис. 3.9), которая показывает рассеяние прогнозируемых данных относительно эталонных. По диаграмме можно судить, что сеть обучена недостаточно хорошо, из-за малого объема обучающей выборки. Также можно сравнить эталонные данные с прогнозируемыми, выбрав на обычной диаграмме два поля – «ПРОИЗВЕДЕНИЕ» и

«ПРОИЗВЕДЕНИЕ_OUT». Если масштабировать диаграмму и включить отображение меток, то можно увидеть достаточно большую ошибку (рис. 3.10), но на цели этого задания она сказываться не будет.

Визуализатор «Что-если» позволит провести эксперимент, введя любые значения множителей АРГУМЕНТ1 и АРГУМЕНТ2 и рассчитав результат их произведения. Попробуем ввести аргументы которые отсутствуют в обучающей выборке, например, 7x7. Как видно на рис. 3.11 сеть обучена достаточно точно, так как получен верный результат, несмотря на то, что сеть никогда не видела такую комбинацию аргументов.

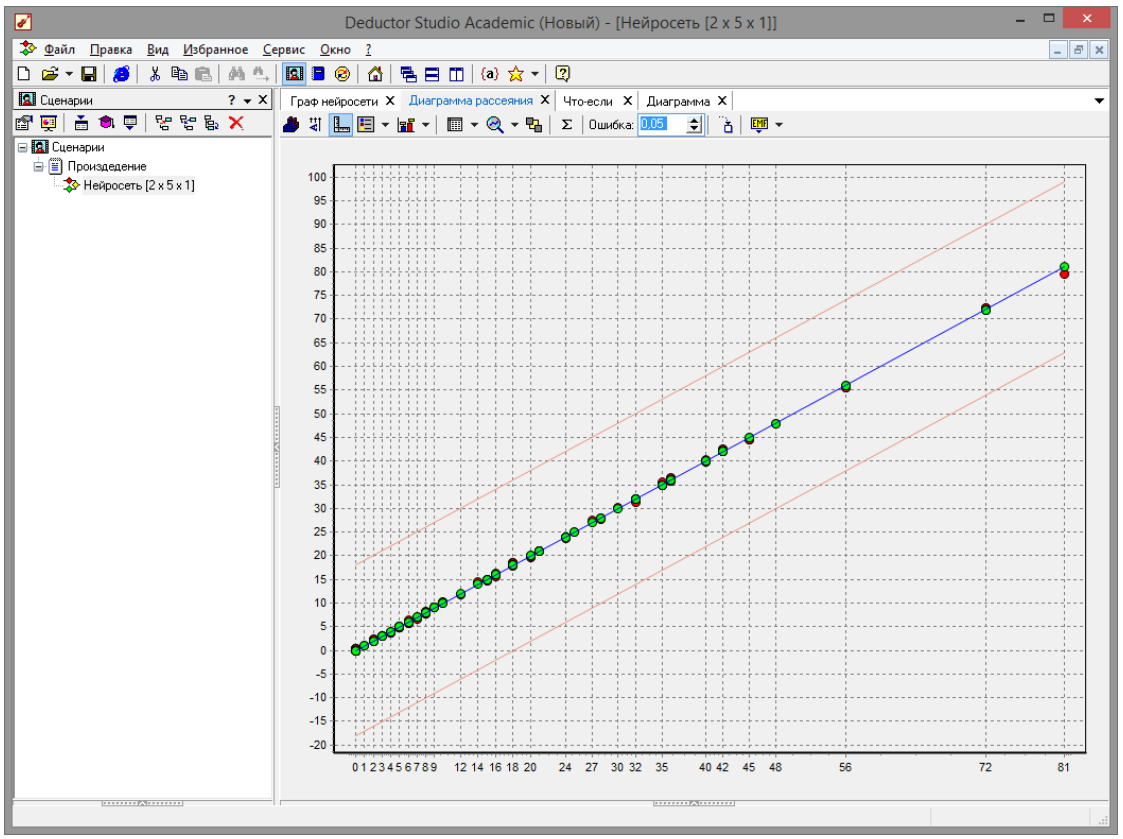


Рис. 3.9 - Диаграмма рассеяния

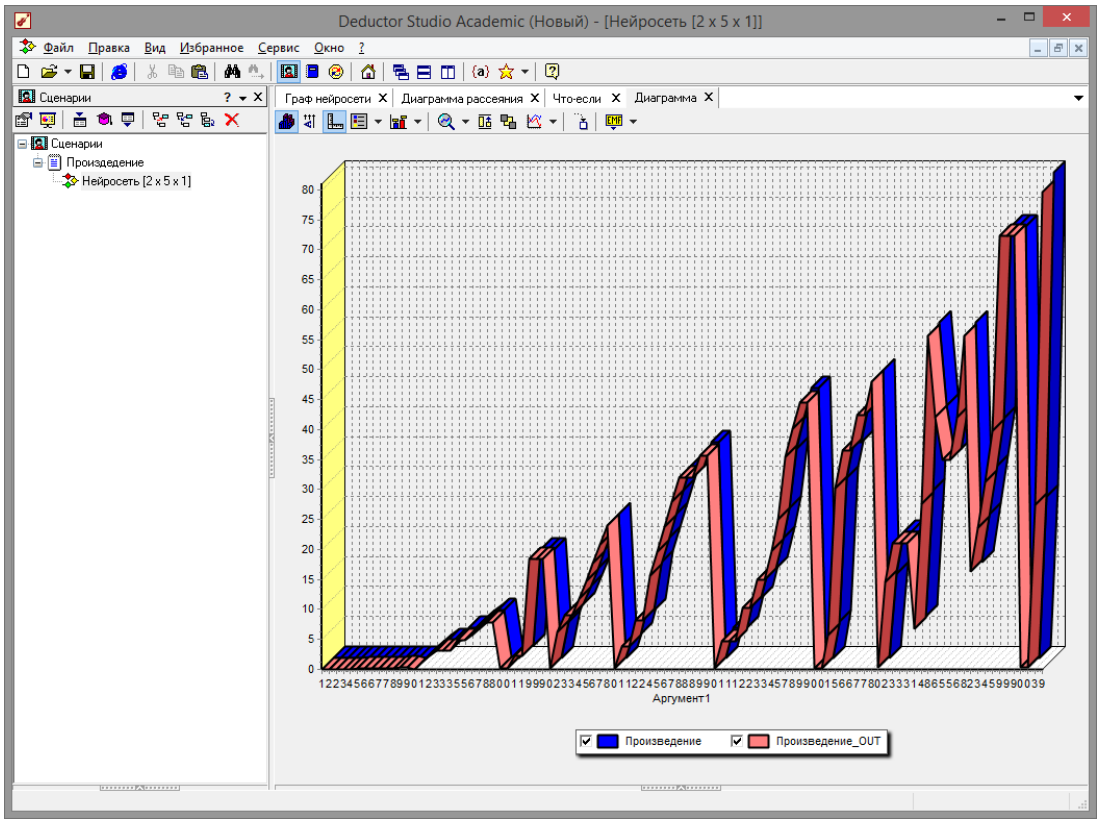


Рис. 3.10 - Сравнение эталонных данных с прогнозируемыми

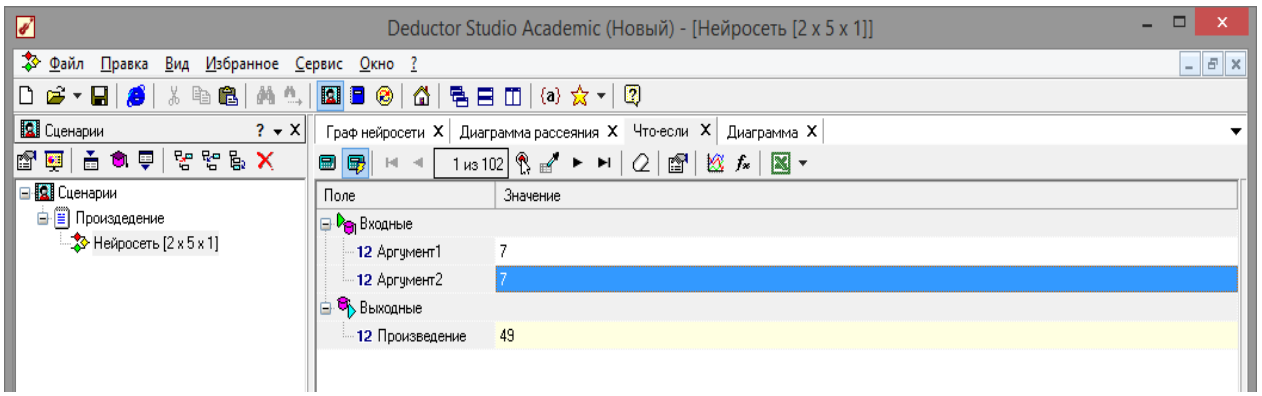


Рис. 3.11 - Инструмент «Что-Если»

Далее введем в окно «Что-Если» аргументы, которые как следует из диаграммы и диаграммы рассеивания большую ошибку. Как видим по произведению, данные диаграммы верни относительно ошибки.

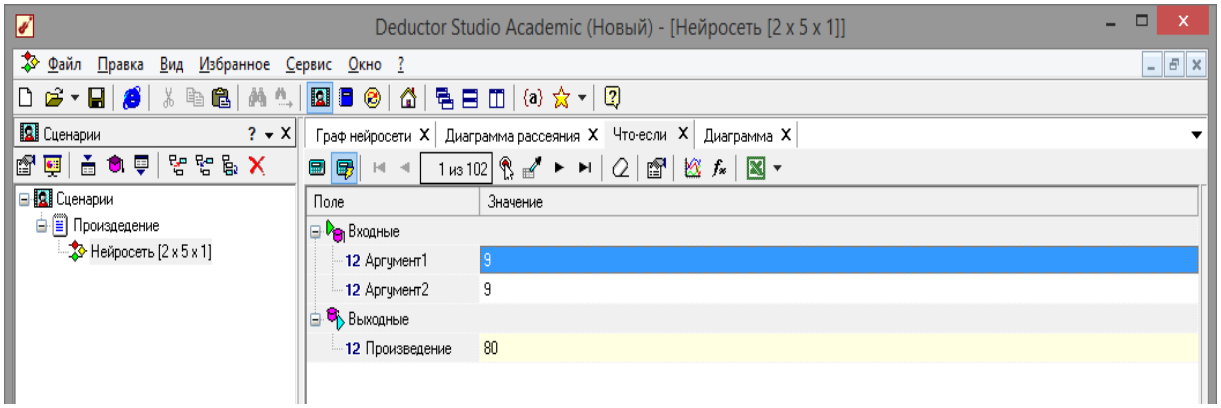


Рис. 3.12 - Ошибки прогнозирования

Вид построенной сети можно посмотреть, выбрав визуализатор «Граф нейронной сети» (рис. 3.13).

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.

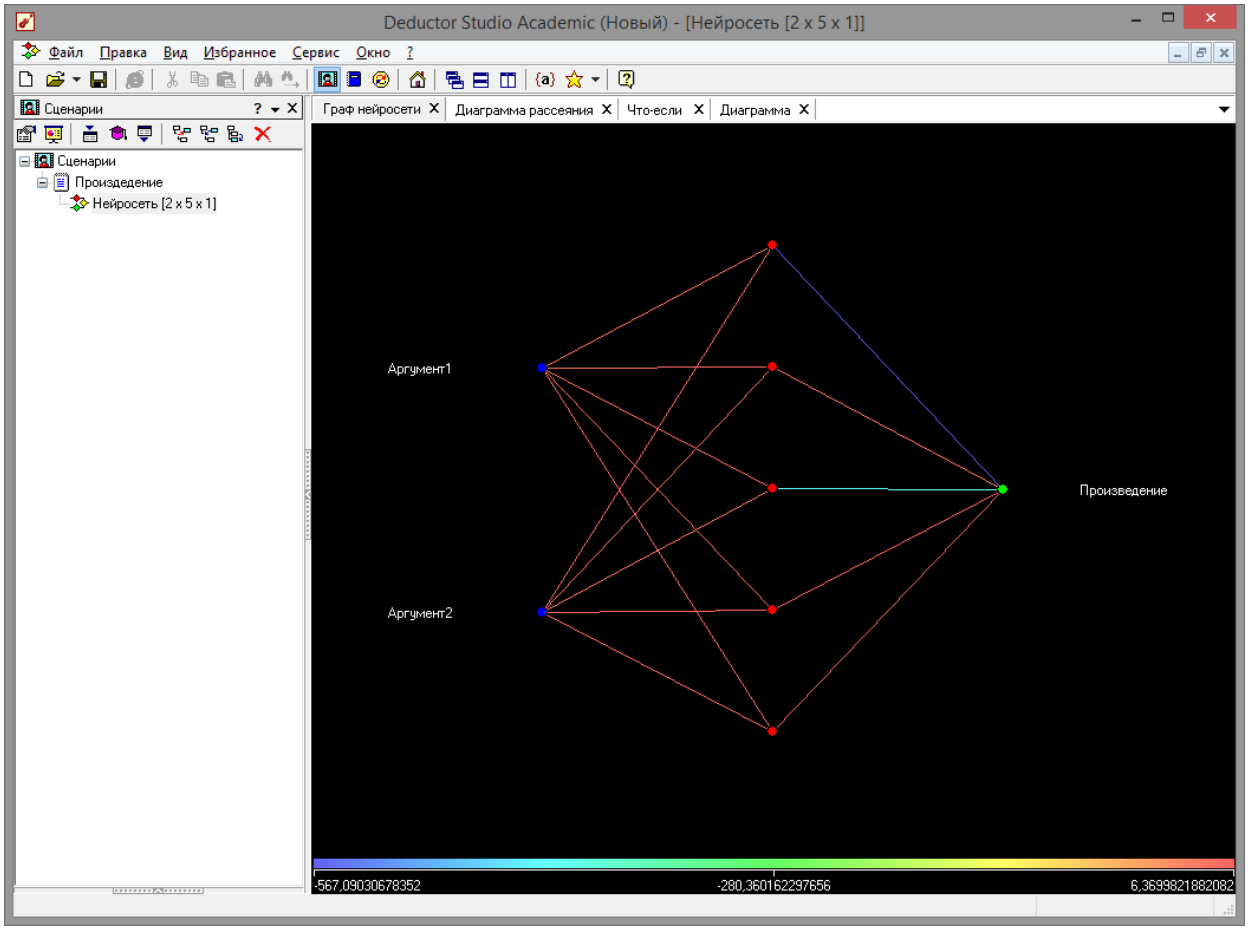


Рис. 3.13 - Граф нейронной сети

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». 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.

3.2 Прогнозирование данных на основе 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.

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 «Продажи.txt» Выполним импорт данных из файла (рис. 3.14).

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 спектральной обработки. Запустим мастер обработки (рис. 3.15), выберем в качестве обработки данных спектральную обработку и перейдем на следующий шаг мастера. Следующий шаг отвечает за удаление аномалий из исходного набора. Выберем поле для обработки «КОЛИЧЕСТВО» и укажем для него вычитание шума (степень вычитания – малая).

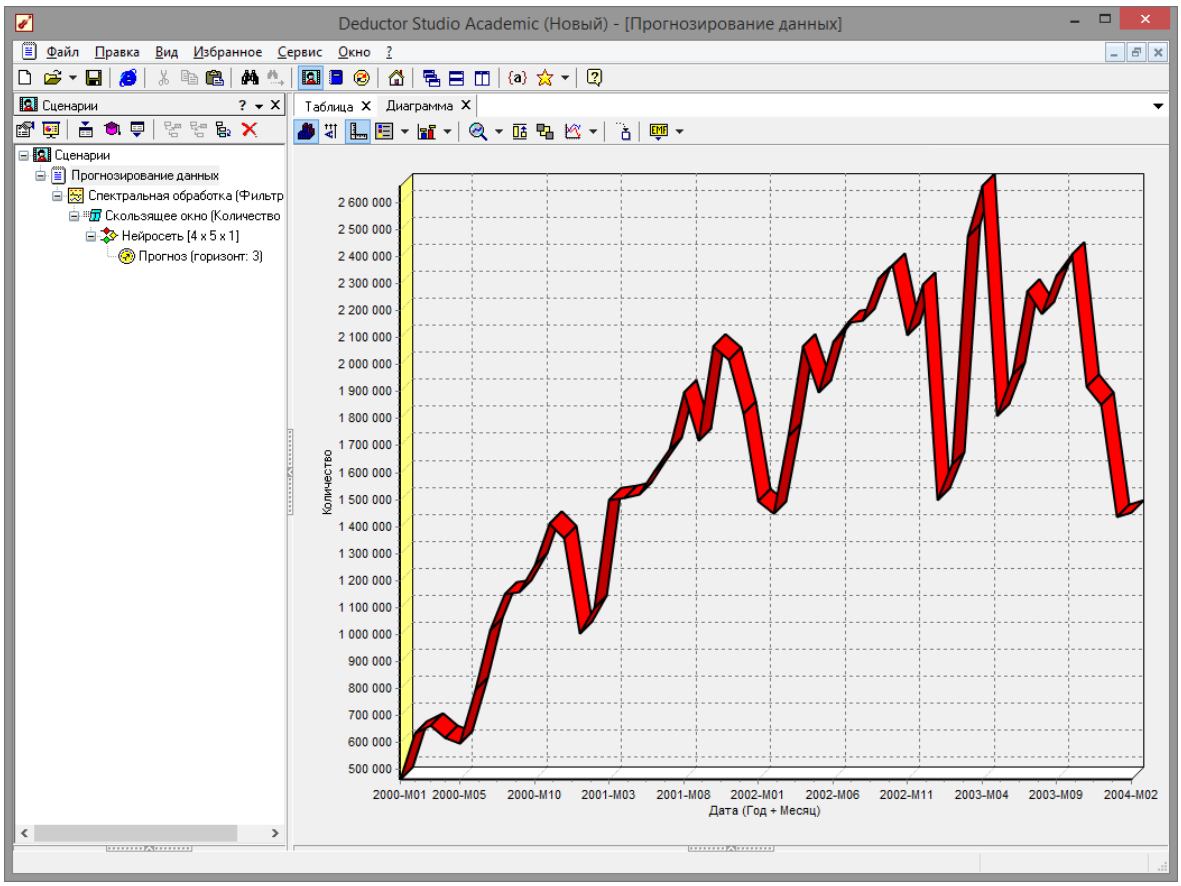


Рис. 3.14 - .Диаграмма продаж

На следующем шаге запустим обработку, нажав на «пуск» и посмотрим полученный результат (рис. 3.16). 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. Предполагается, что количество продаж на следующий месяц зависит от количества продаж за предыдущие месяцы. 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. Здесь явно необходимо трансформировать данные к скользящему окну.

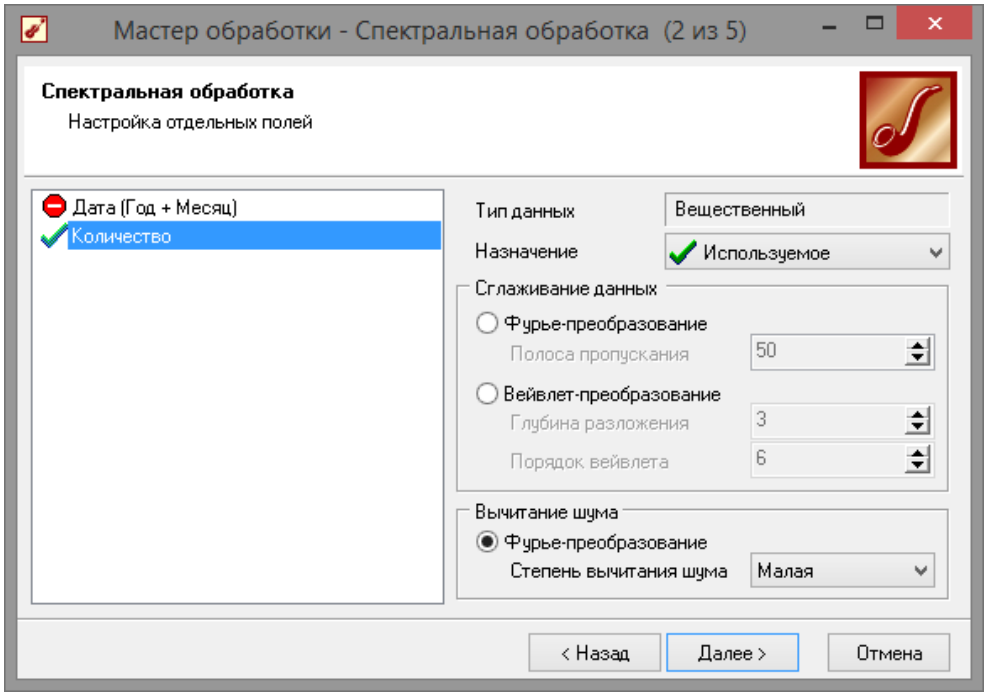


Рис. 3.15 - Выбор спектральной обработки

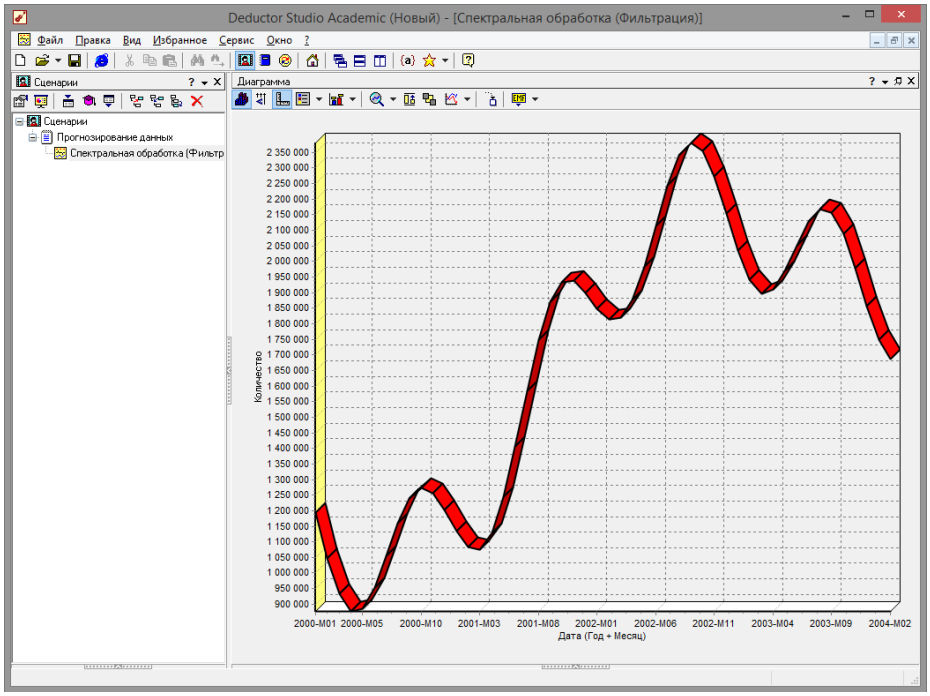


Рис. 3.16 - Результат спектральной обработки

Запустим мастер обработки, выберем в качестве обработчика скользящее окно и перейдем на следующий шаг. 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 (рис. 3.17).

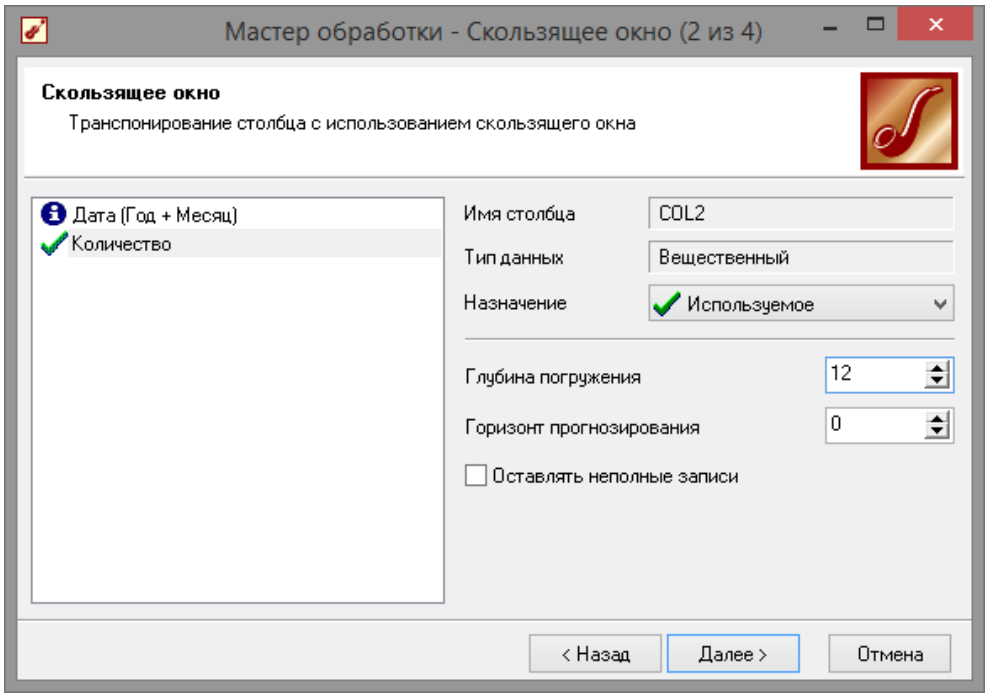


Рис. 3.17 - Скользящее окно

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 буден указан столбец «КОЛИЧЕСТВО».

Перейдем непосредственно к самому построению модели прогноза. Откроем мастер обработки и выберем в нем нейронную сеть (рис. 3.19). На втором шаге мастера, согласно с принятым ранее решениям, установим в качестве входных поля «КОЛИЧЕСТВО-12»,

«КОЛИЧЕСТВО-11», «КОЛИЧЕСТВО-2» и «КОЛИЧЕСТВО-1», а в

качестве выходного - «КОЛИЧЕСТВО». Остальные поля сделаем информационными.

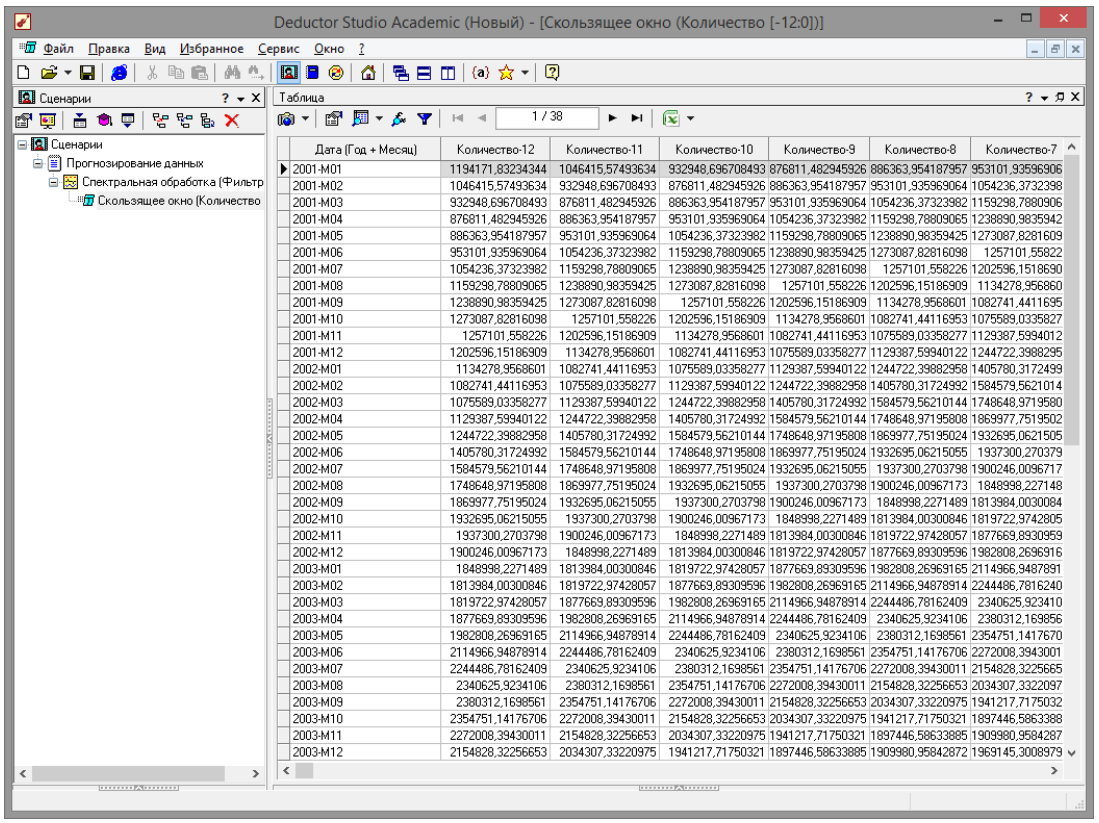


Рис. 3.18 - Таблица скользящего окна

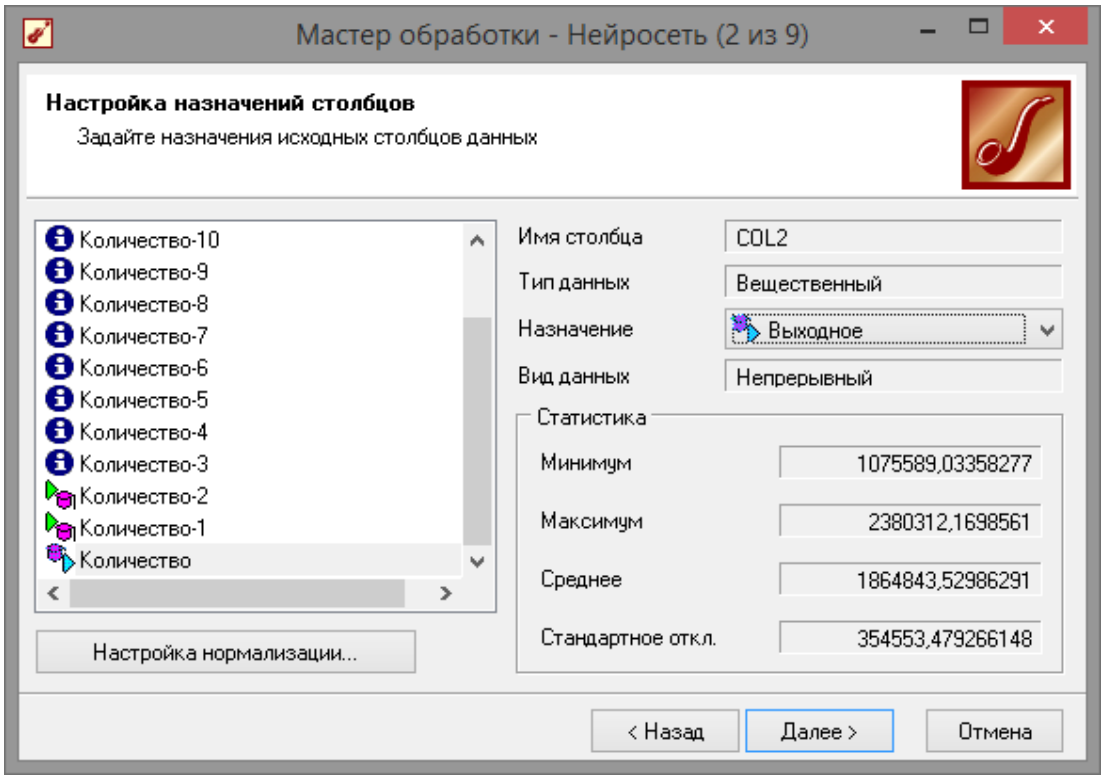


Рис. 3.19 - Настройки мастера спектральной обработки

Оставив остальные параметры построения модели по умолчанию, только количество нейронов скрытого слоя поставим равным 5

(рис. 3.20), обучим нейросеть (см. пример «прогнозирование умножения с помощью нейронной сети») (рис. 3.21). 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, показанные на рис.3.22.

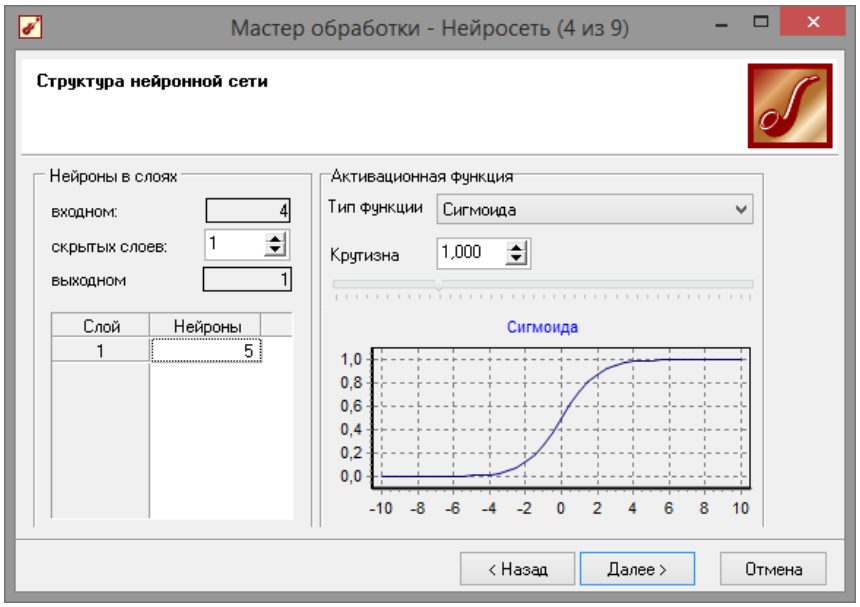


Рис. 3.20 - Настройки нейросети

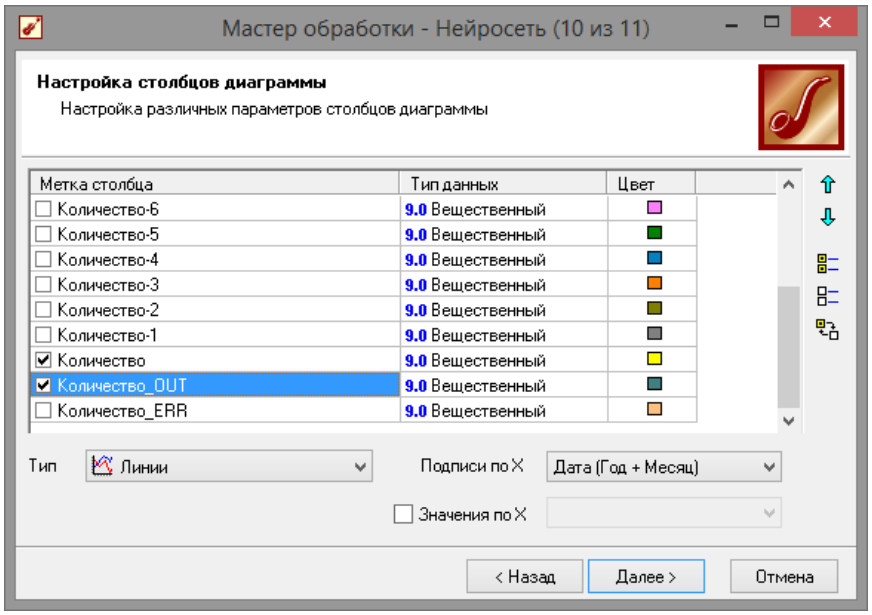


Рис. 3.21 - Обучение нейронной сети

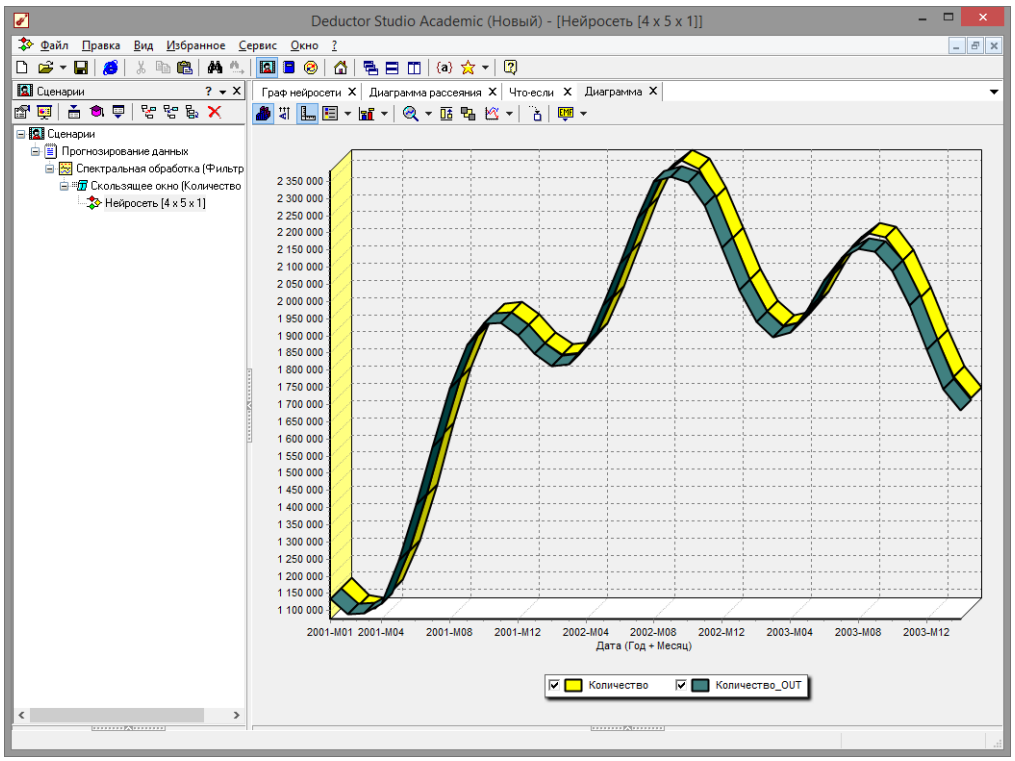


Рис. 3.22 - Сравнение эталонных данных с прогнозом

Диаграмма рассеяния более наглядно показывает качество обучения (рис. 3.23).

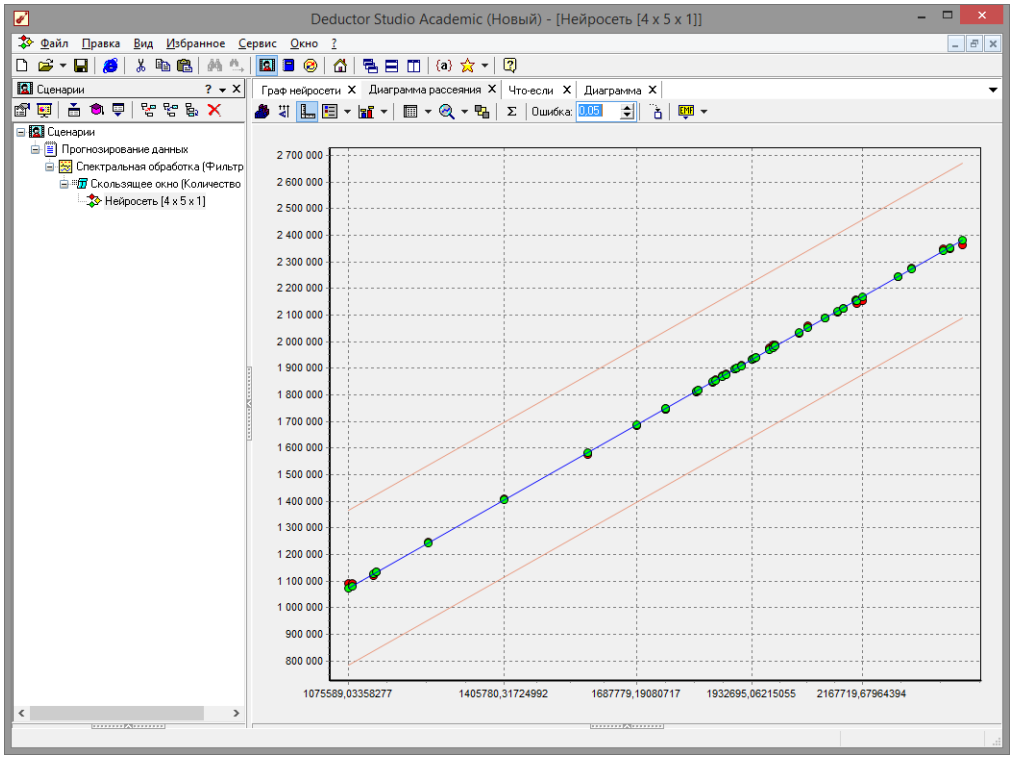


Рис. 3.23 - Диаграмма рассеяния

Нейросеть обучена, теперь осталось самое главное –

построить требуемый прогноз. Для этого открываем мастер обработки (рис. 3.24) 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 (рис. 3.25). 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.

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». Теперь аналитик может дать ответ на вопрос, какое количество товаров будет продано в следующем месяце и даже два месяца спустя (рис 3.26). Масштабировав результат и включив метки, можно увидеть расчетные значение на 3 месяца вперед (рис.3.27).

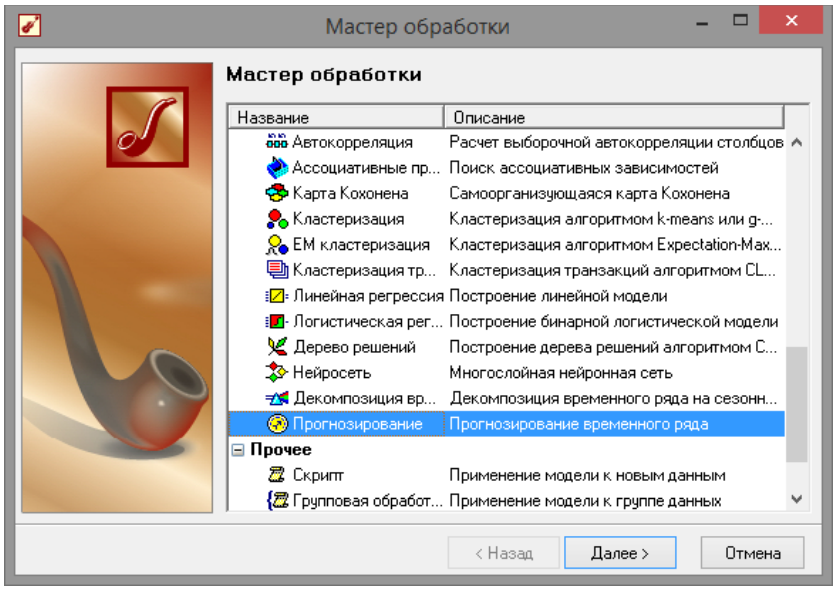


Рис. 3.24 - Мастер прогнозирования

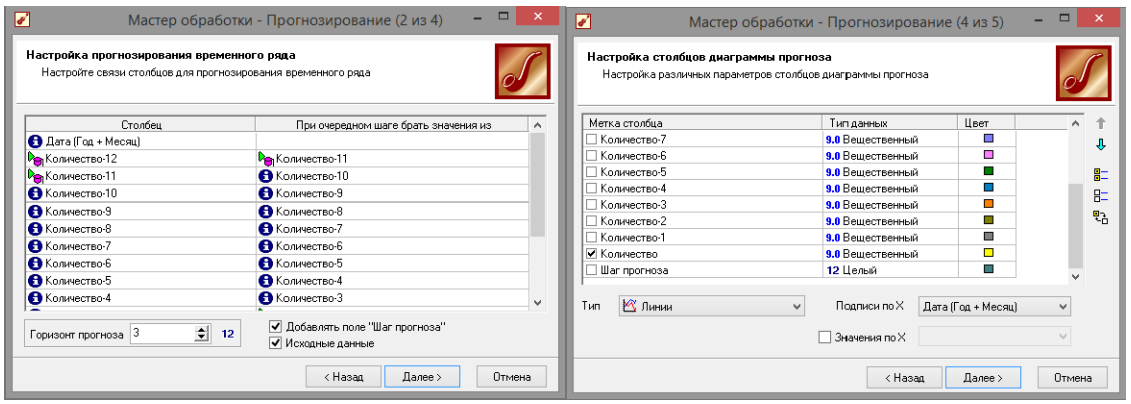


Рис. 3.25 - Мастер прогнозирования

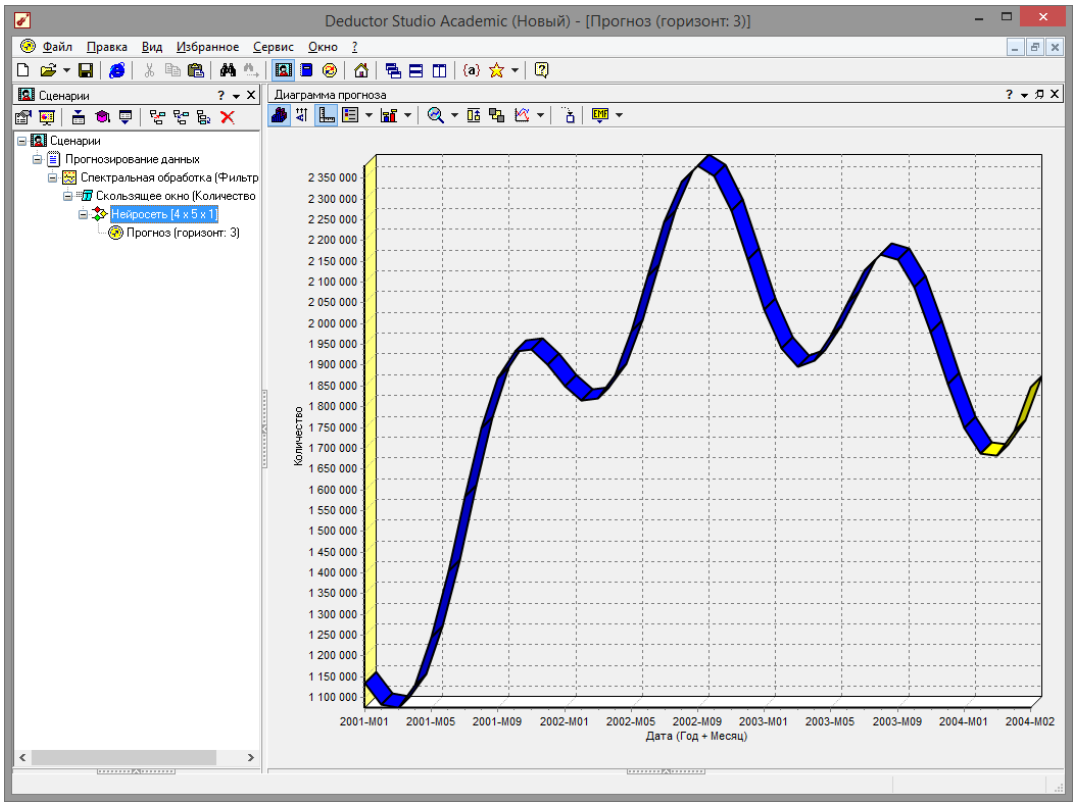


Рис. 3.26 - Расчетные значение на 3 месяца вперед

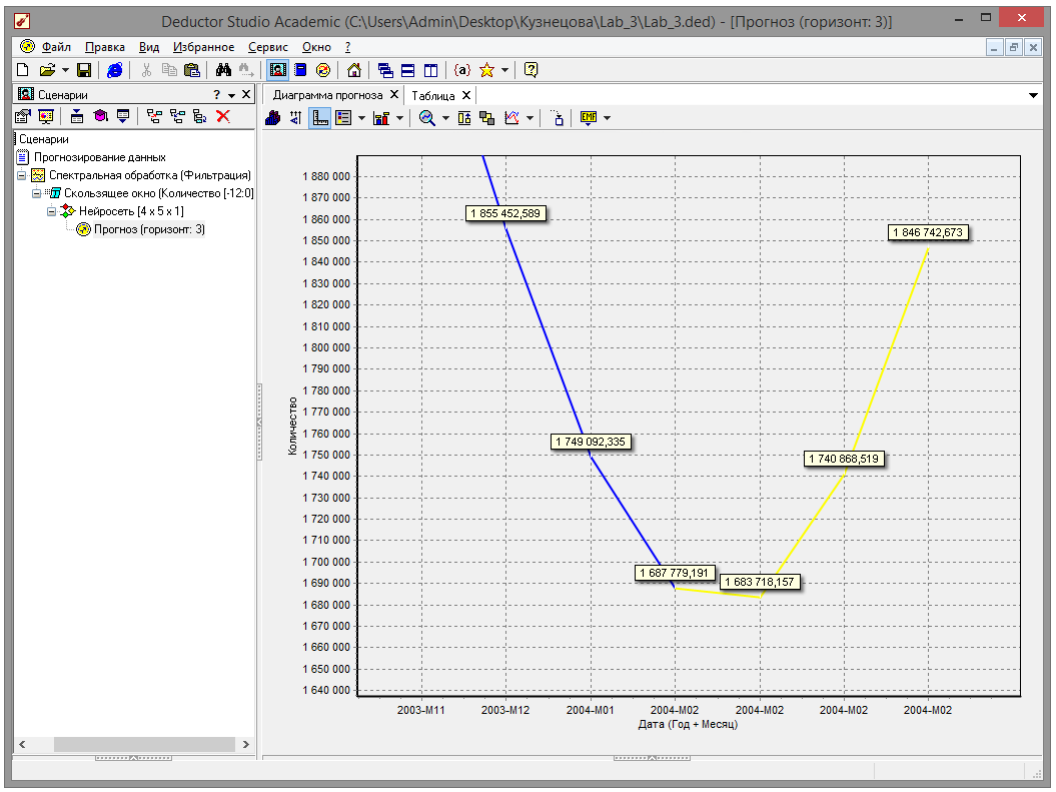


Рис. 3.27. Результаты мастера прогнозирования

После завершения анализа данные можно экспортировать. Так как данная версия является бесплатной для образовательных целей, то данные можно выгрузить либо в текстовый файл, либо в собственный проект программы с расширением «**.ded*». Мастер экспорта имеет точно такие же настройки, как и мастер импорта. Более того, если экспорт данных совершить в текстовый фал, то далее данные можно скопировать в файл табличного процессора *Excel*, и достаточно комфортно с ними работать.

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.

3.3 Задание на самостоятельную работу

Получить от преподавателя вариант задания для прогнозирования (изменение курса валют, график синусоиды, прогнозирование суммы или разности чисел и др.).

Контрольные вопросы

Что такое временной ряд?

В какие форматы можно экспортиров данные *Deductor Academic*?

Что такое обучающая и тестовая выборка?

Какие инструменты можно использовать д прогнозирования*?*

5. Для чего служит диаграмма рассеяния?

Лабораторная работа 4

Нейросетевые технологии в интеллектуальном анализе данных

Цель работы: изучить кластеризацию с помощью самоорганизующихся карт Кохонена в аналитическом пакете Deductor Academic.

Программа работы

1. Произвести импорт данных из подготовленного файла.
2. С помощью Карты Кохонена выполнить выполнить кластеризацию на данных контрольного примера.
3. Выполнить задачу кластеризацию для данных по индивидуальному заданию.

Методические указания по выполнению работы

4.1 Общие понятия о самоорганизующихся картах Кохонена

Самоорганизующаяся карта Кохонена (англ. *Self-organizing map*

- *SOM*) – соревновательная нейронная сеть с обучением без учителя, выполняющая задачу визуализации и кластеризации. Идея сети предложена финским учёным Т. Кохоненом. Является методом проецирования многомерного пространства в пространство с более низкой размерностью (чаще всего, двумерное), применяется также для решения задач моделирования, прогнозирования и др.

Самоорганизующаяся карта состоит из компонентов, называемых узлами или нейронами. Их количество задаётся аналитиком. Каждый из узлов описывается двумя векторами. Первый – т. н. вектор веса *m*, имеющий такую же размерность, что и входные данные. Второй — вектор *r*, представляющий собой координаты узла на карте. Обычно

узлы располагают в вершинах регулярной решётки с квадратными или шестиугольными ячейками.

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 и 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).

4.2. Пример кластеризации данных

Рассмотрим механизм кластеризации путем построения самоорганизующейся карты, основываясь на типичных характеристиках цветков. Исходная таблица находится в файле примеров «Ирисы.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.

Для начала необходимо импортировать данные из файла (рис. 4.1). После этого запустим, мастер обработки и выберем из списка метод обработки «Карта Кохонена» (рис. 4.2). На втором шаге мастера настроим назначения столбцов (рис. 4.3). Укажем столбцу «КЛАСС ЦВЕТКА» назначение «Выходной», а остальным – «Входной». Т.е. на основе данных о цветке будем относить его к тому или иному классу.

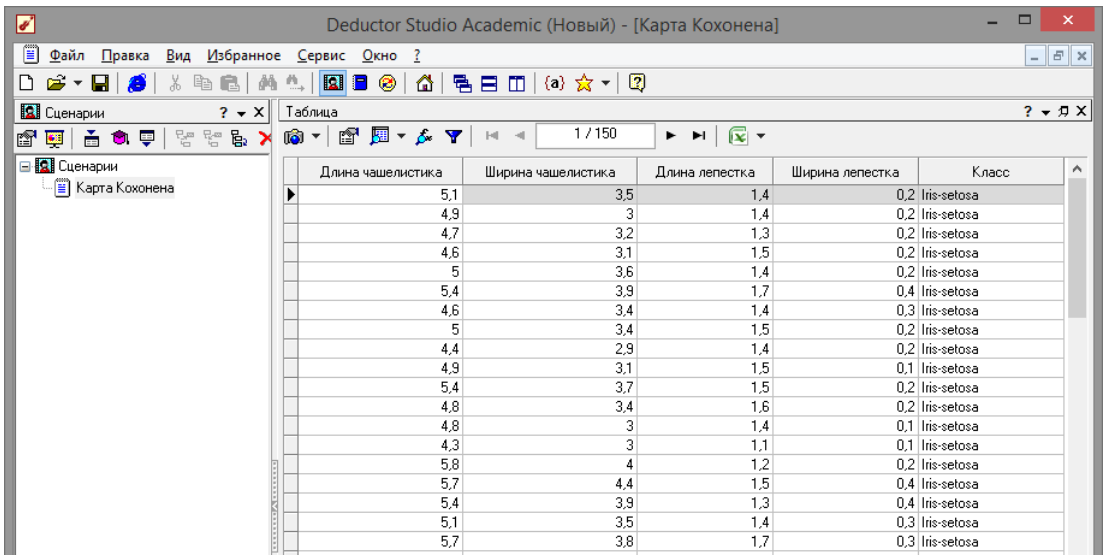


Рис. 4.1 - Импортированные данные

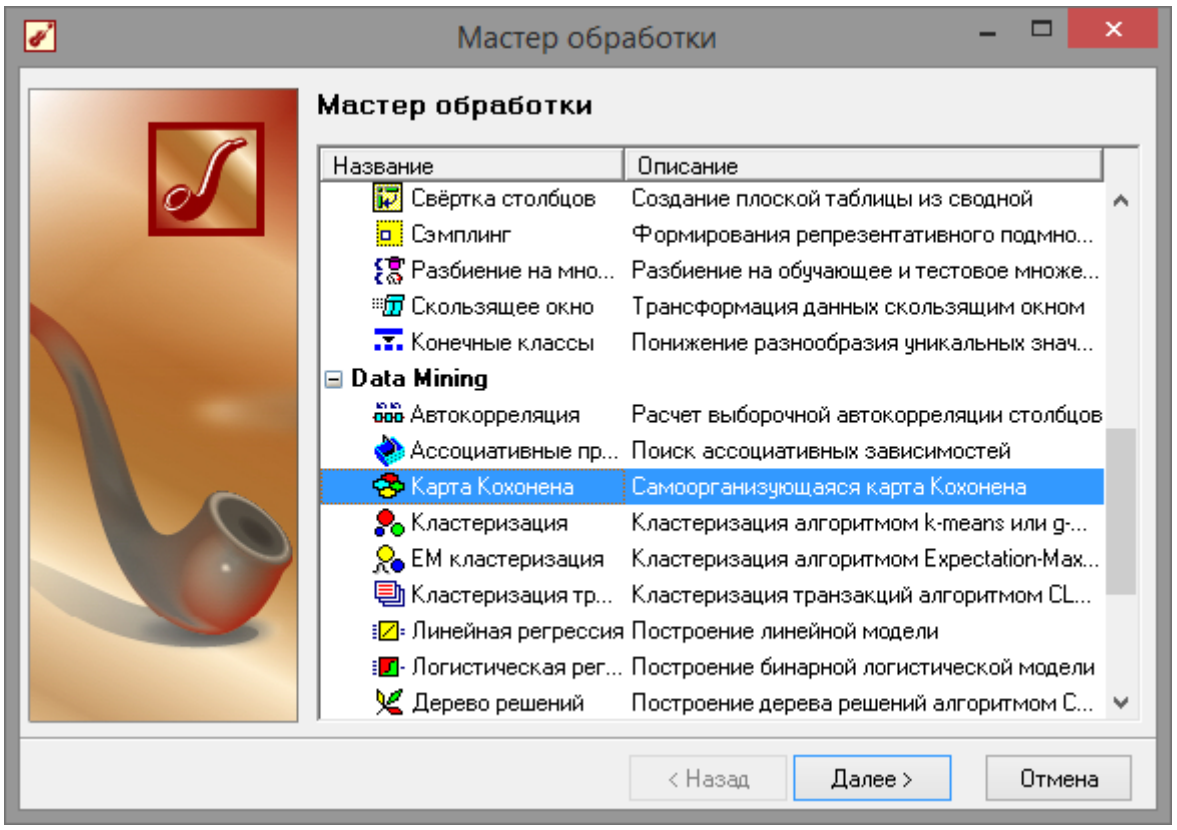


Рис. 4.2 - Мастер обработки «Карта Кохонена»

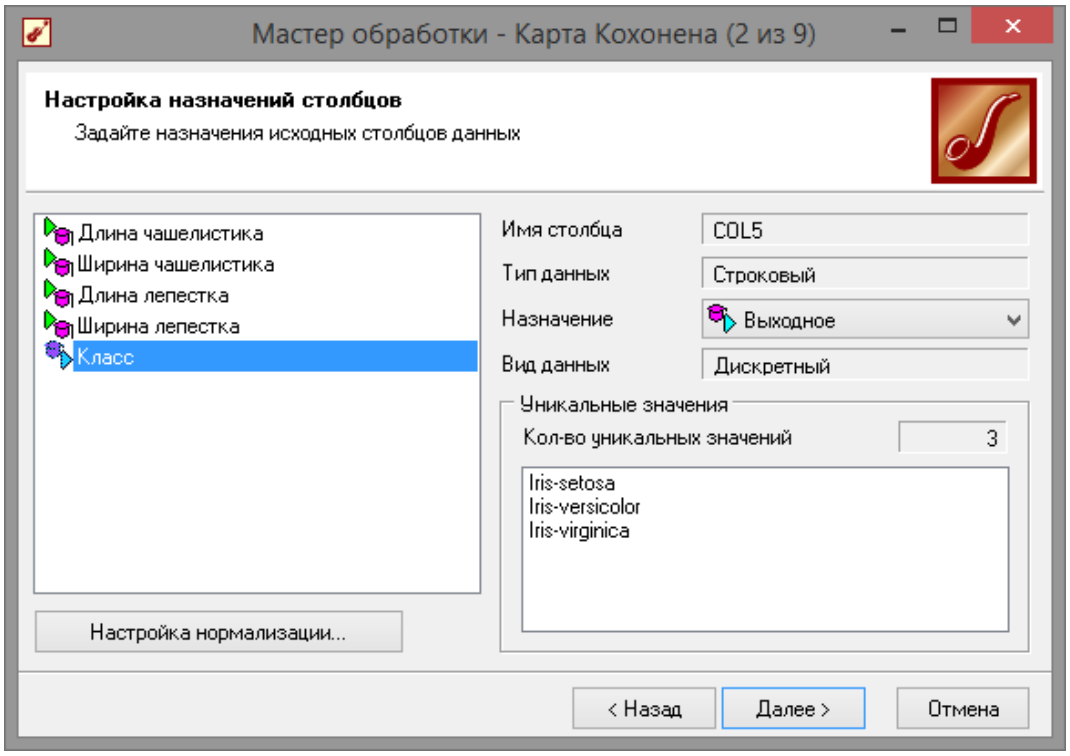


Рис. 4.3. Настройки мастера карт Кохонена

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»(рис. 4.4).

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 (рис. 4.5).

На пятом шаге мастера необходимо настроить параметры остановки обучения. Оставим параметры по умолчанию (рис. 4.6).

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 (рис. 4.7).

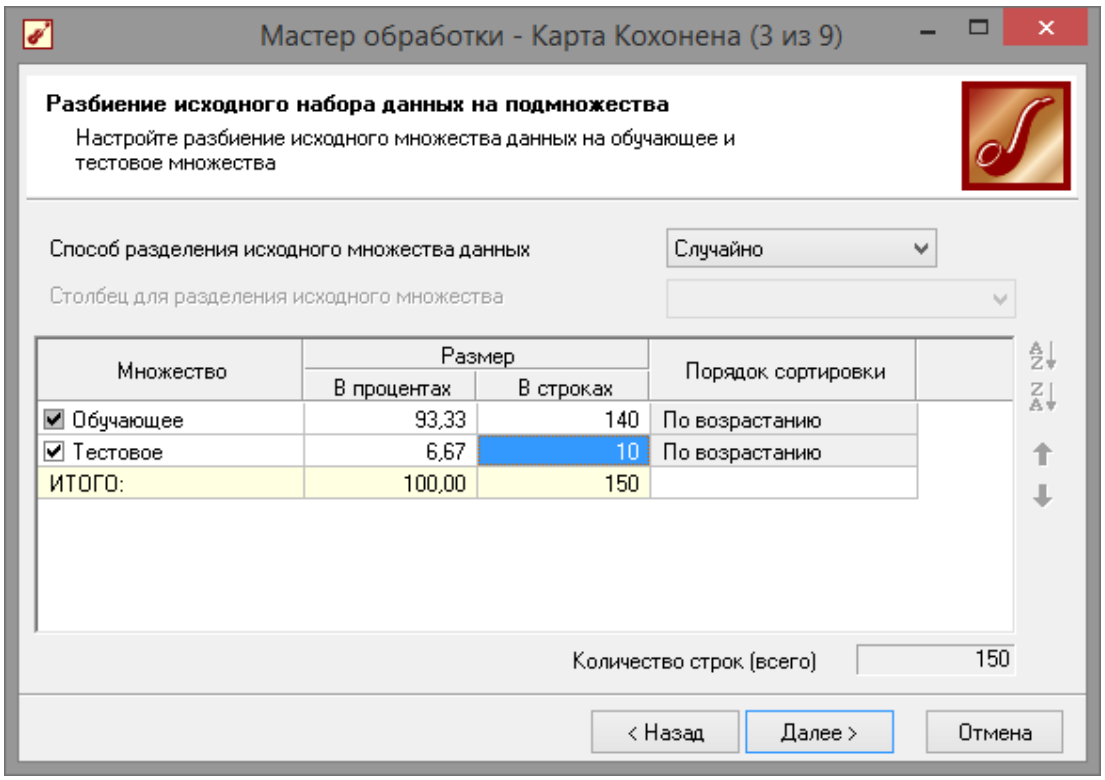


Рис.4.4 - Настройки тестового и обучающего множества

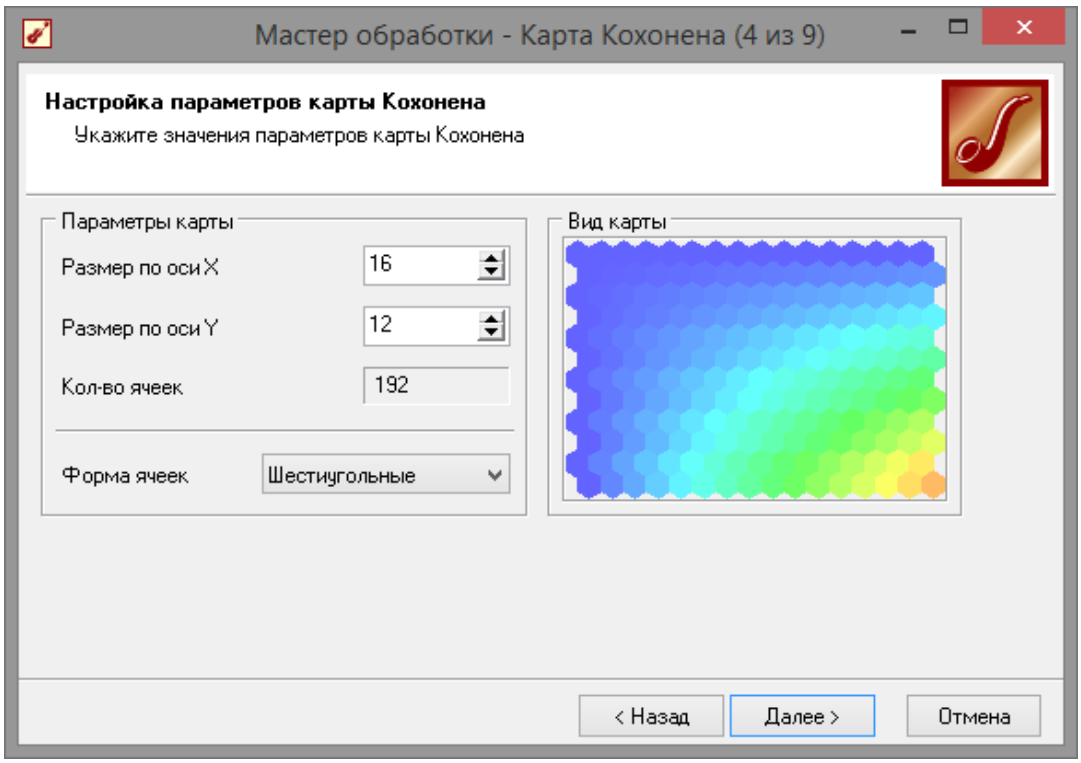


Рис. 4.5 - Настройки значения параметров карт

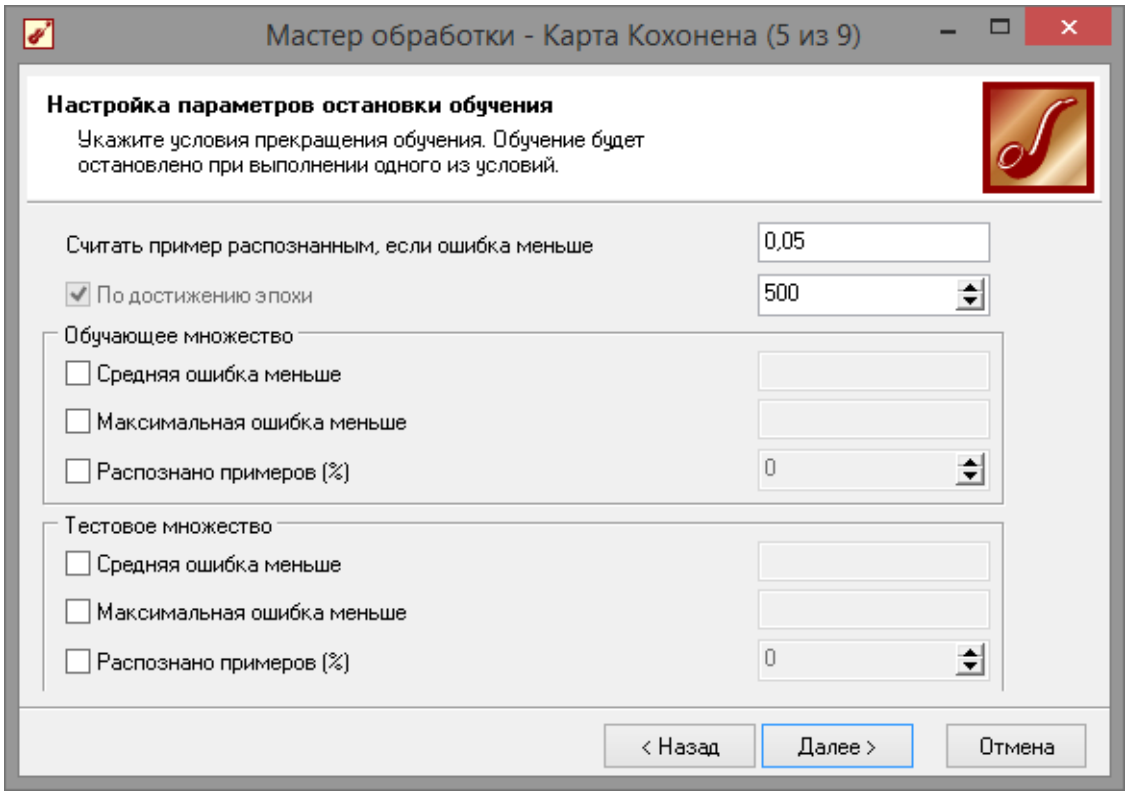


Рис.4.6 - Настройки параметров остановки обучения

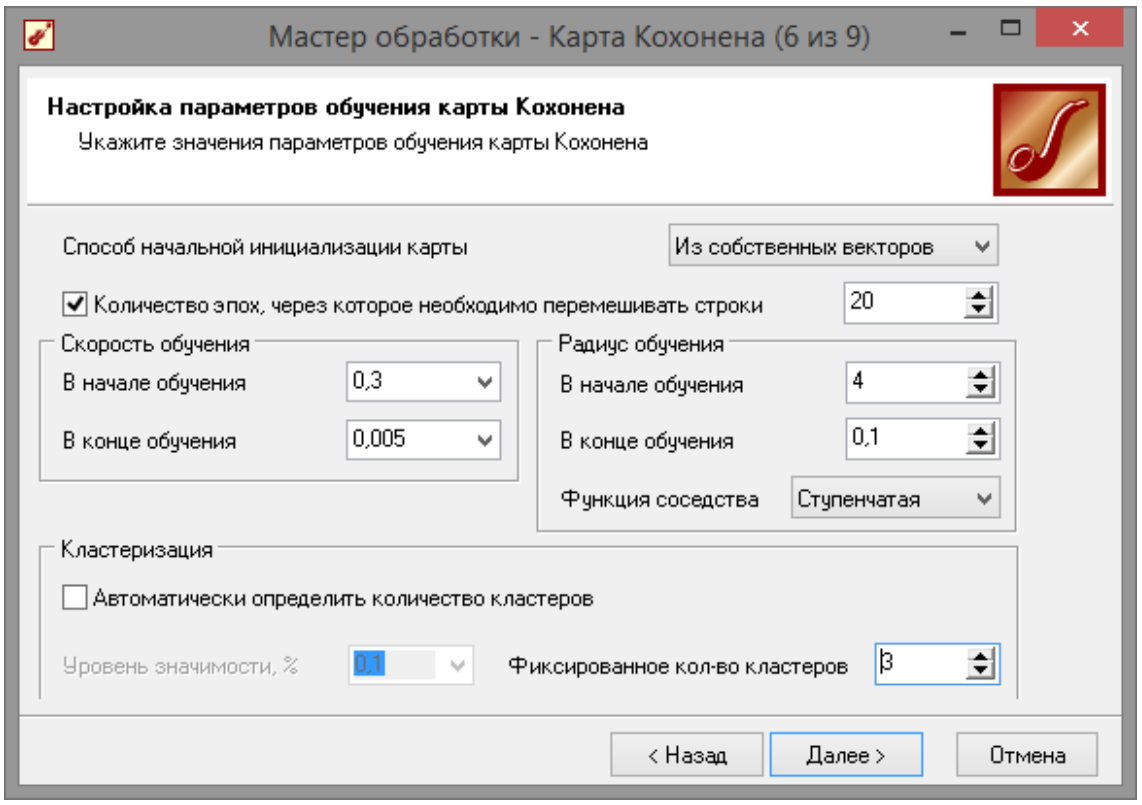


Рис. 4.7 - Настройки параметров обучения карты Кохонена

На седьмом шаге предлагается запустить сам процесс обучения (рис. 4.8). Во время обучения можно посмотреть количество распознанных примеров и текущие значения ошибок. Здесь необходимо нажать на кнопку пуск и дождаться завершения процесса обработки.

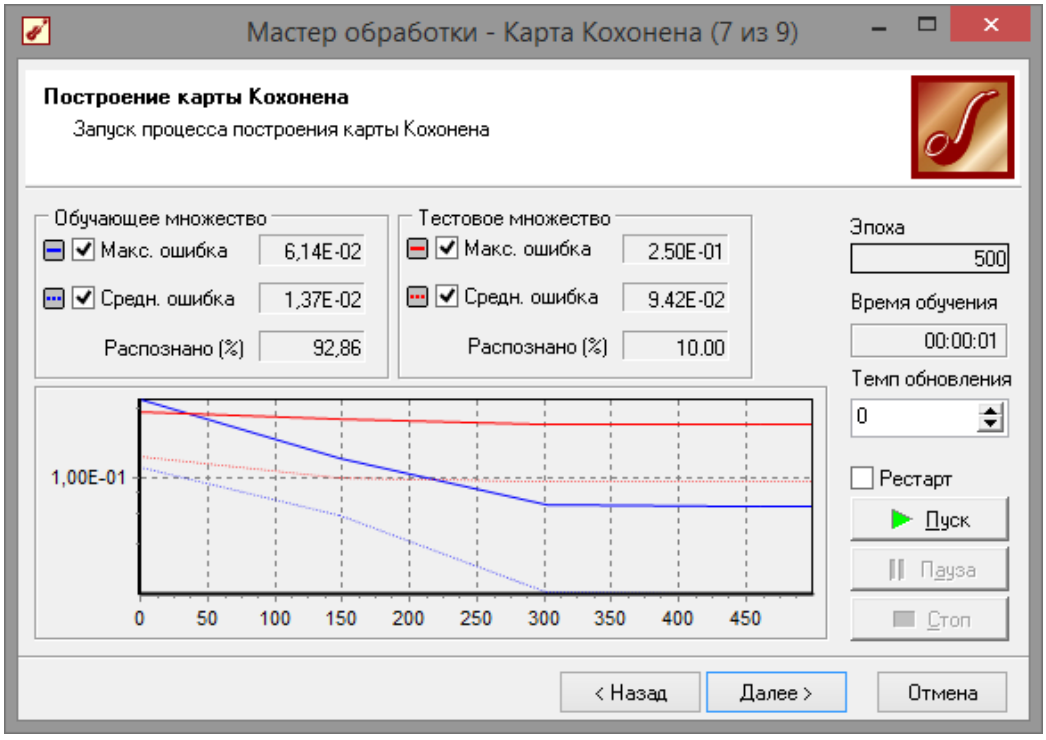


Рис. 4.8 - Процесс обучения

После этого необходимо в списке визуализаторов выбрать появившуюся теперь «Карту Кохонена» для просмотра результатов кластеризации, а также визуализатор «Что-если» для прогнозирования класса цветка(рис. 4.9).

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» (рис. 4.10).

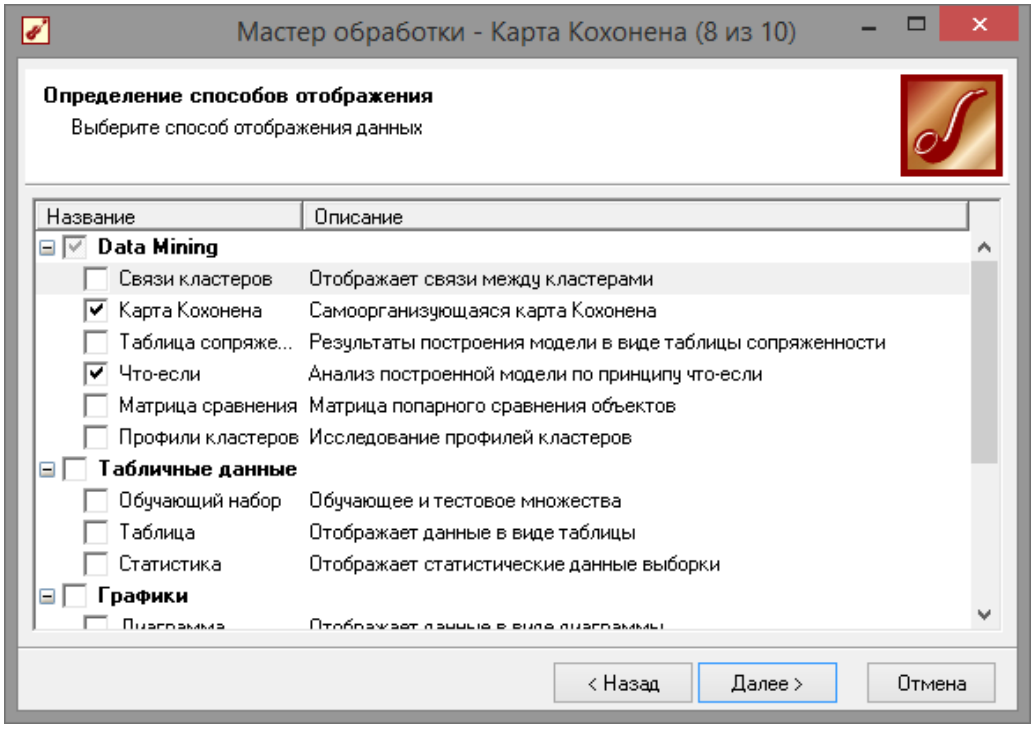


Рис. 4.9 - Способ отображения данных

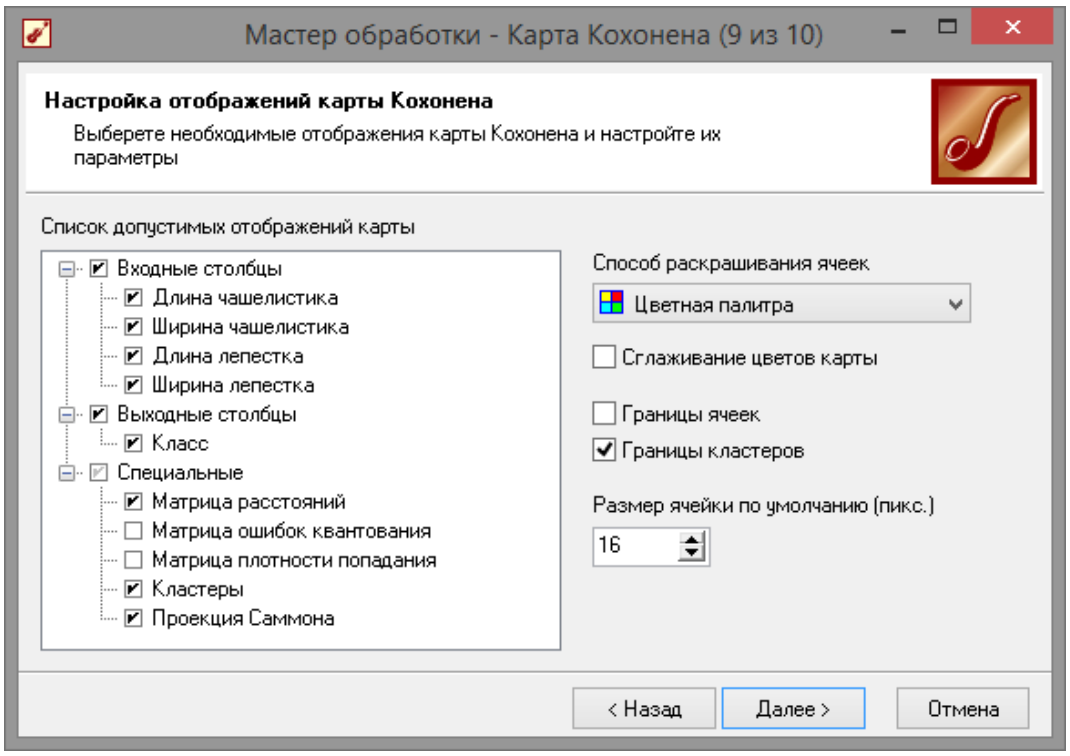


Рис. 4.10 - Настройка отображения кластеров

После этого можно увидеть полученные результаты (рис. 4.11). 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 *Versicolo* 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.

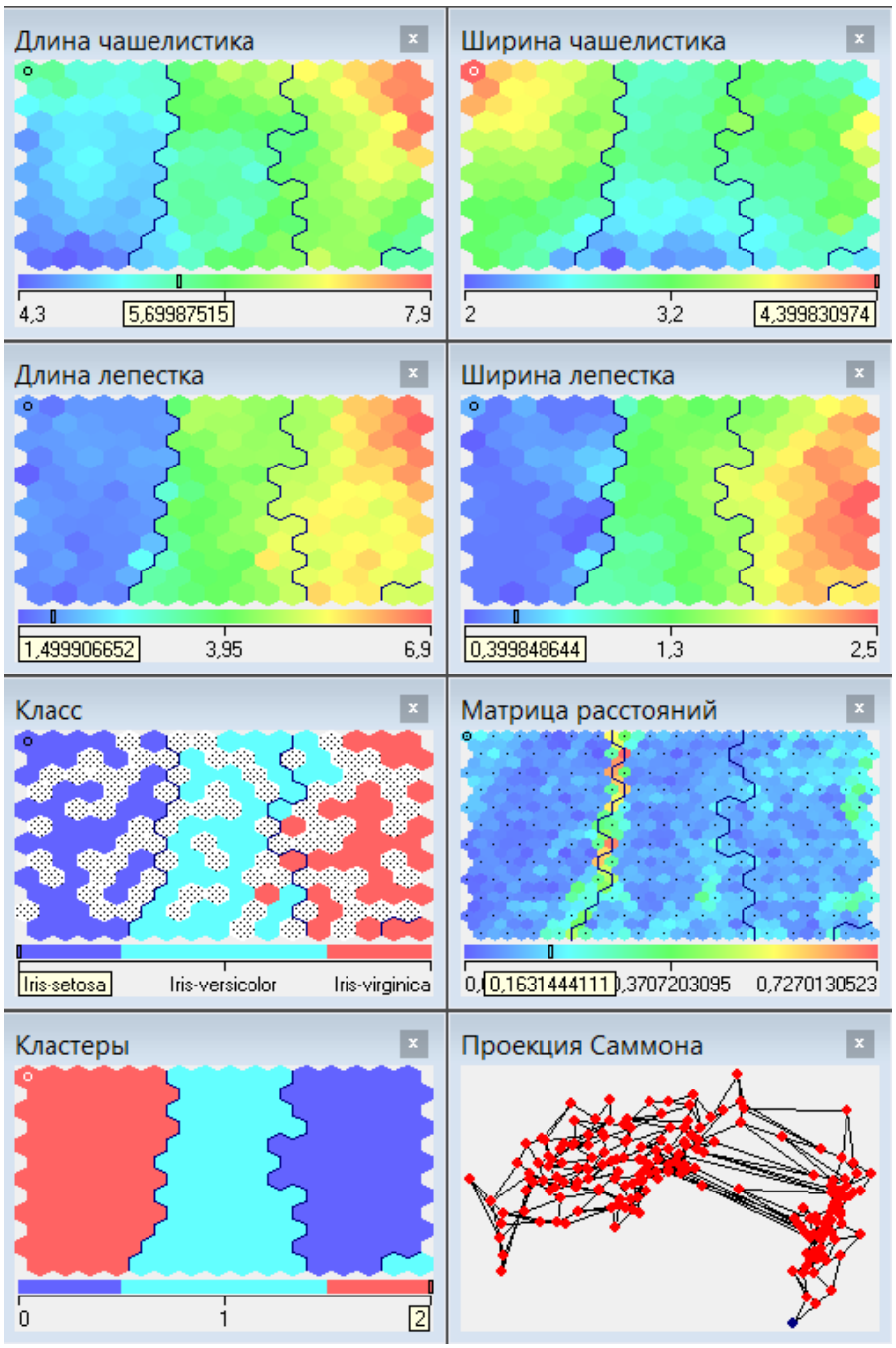


Рис. 4.11 - Карта Кохонена

Рассмотрим построенную таблицу «Что-Если». В верхней части таблицы отображаются входные поля, а в нижней – выходные и расчетные. Изменяя значения входных полей, пользователь дает команду на выполнение расчета и наблюдает за рассчитанными значениями выходов нейронной сети или дерева решений.

Расчетные поля отличаются от выходных тем, что они не существуют в исходном наборе данных и были созданы в ходе обработки. Такими полями являются, например, «Номер ячейки» или

«Номер кластера».

Каждое поле таблицы «Что-Если» представлено следующими атрибутами:

– «Тип» – указывается значок, соответствующий типу данных поля;

– «Поле» – имя входного или выходного поля;

– «Значение» – указывается текущее значение поля.

С помощью кнопки «Показать статистику» справа от таблицы можно вывести статистику по выделенному полю. Для непрерывных полей в ней отображается следующая информация:

– «Минимум» – минимальное значение поля в выборке;

– «Максимум» – максимальное значение поля в выборке;

– «Среднее» – среднее по выборке значение поля;

– «Стандартное откл.» – среднеквадратическое отклонение значений поля по выборке.

Знание диапазона входных данных (минимума и максимума), на котором строилась модель, позволит определить область устойчивости системы. Очевидно, что если подать на вход значения, существенно выходящие за диапазон, гарантировать правильную реакцию системы нельзя, и достоверность полученных данных может быть снижена. Если значение, присвоенное полю, выходит за границы диапазона, это поле окрашивается в красный цвет.

Для дискретных полей статистика содержит:

– «Значения» – список уникальных значений;

– «Кол-во» – число вхождений значения в выборку;

– «Итоговая информация» – общее число уникальных значений в выборке.

Для дискретных значений на вход можно подавать только значения, представленные в этом списке.

В таблице пользователь может менять лишь содержимое столбца

«Значение». Это осуществляется несколькими способами:

– непосредственно ввести данные с клавиатуры;

– заполнить записями из текущей выборки (при этом вводятся записи целиком и заполняются одновременно все поля);

– выбрать значения из статистики, находящейся справа от таблицы.

Чтобы ввести значения входов с клавиатуры, нужно выбрать ячейку «Значение» для соответствующего поля, и только потом вводить данные. Чтобы войти в режим редактирования, достаточно напечатать любой символ с клавиатуры в том числе «*Enter*» либо дважды "кликнуть" мышкой по соответствующей ячейке. Дискретные значения выбираются из выпадающего списка либо путем циклического перебора в следствии двойного "клика" мышкой. Для перехода к предыдущим или последующим строкам используются клавиши со стрелками. Если введенные вами значения выходят за диапазон значений выборки, соответствующая строка таблицы выделяется красным цветом. Если находясь на ячейке, нажать клавишу "*Del*", то значение соответствующего входного поля будет очищено.

Для автоматического ввода в таблицу «Что-Если» записей из текущей выборки используются кнопки на панели инструментов:

– Первая запись (Ctrl+PgUp) – позволяет выбрать для загрузки в таблицу «Что-Если» первую запись выборки;

– Предыдущая запись (PgUp) – позволяет загрузить предыдущую запись;

– Загрузить запись – загружает текущую запись в соответствующие входные поля таблицы «Что-Если»;

– Загрузить из исходной выборки – выводит на экран модальное окно с таблицей, из которой можно загрузить необходимую запись;

Как видно из таблицы «Что если» (рис. 4.12), даже данные отсутствующие в изначальной выборке определяются коректно.

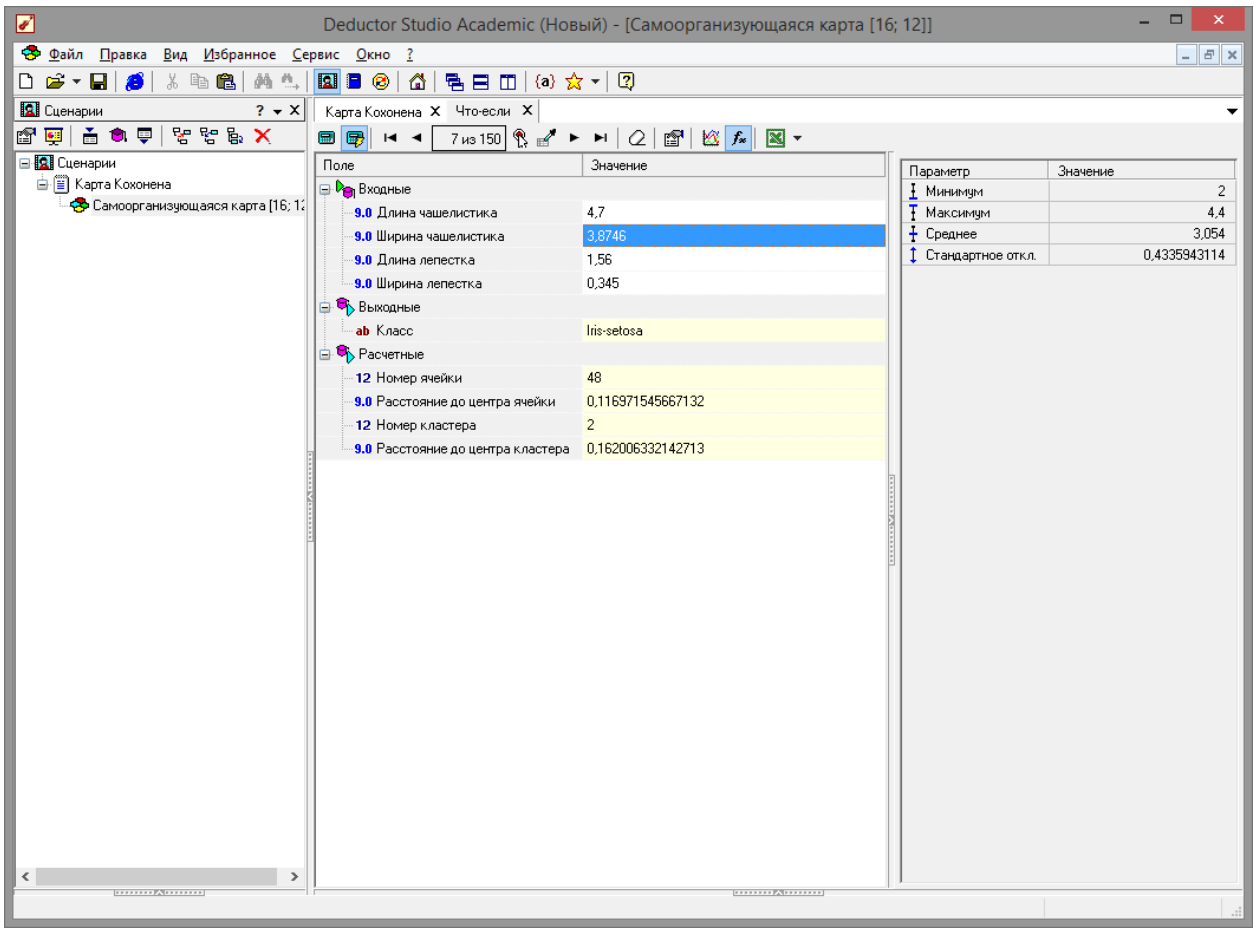


Рис. 4.12 - Таблица «Что-Если»

Данный пример показал область применения самоорганизующихся карт. Изначально имелось многомерное (четырехмерное) пространство входных факторов. Алгоритм представил его в двумерном виде, которое удобнее анализировать. Также исходные данные были отнесены к трем кластерам, по типу цветка – «*Setosa*», «*Versicolo*», «*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. информация о задаче, то качество очистки данных можно увеличить на порядки.

4.3 Задание на самостоятельную работу

Получить от преподавателя вариант задания на кластеризации (классификация помидоров по диаметру плода, весу плода, количеству плодов на кусте, высоте куста, или задача классификации призывников по категориям на основе их параметров и др.).

Решить задачу при помощи карт Кохонена.

Контрольные вопросы

1. Что такое карта Кохонена?
2. Что решают задачи кластеризации?
3. Для чего служит таблица «Что-Если»?

Контрольные вопросы промежуточной аттестации (по итогам изучения курса)

1. Данные и модели их представления.
2. Системы поддержки принятия решений (СППР).
3. Роль и место интеллектуального анализа данных в СППР.
4. Задачи ИАД.
5. Алгебра матриц.
6. Функции многих переменных.
7. Необходимые и достаточные условия существования экстремумов применительно к квадратичным формам.
8. Типы шкал.
9. Допустимые преобразования в шкалах.
10. Проверка истинности утверждений.
11. Статистическая выборка.
12. Числовые характеристики распределений.
13. Комплексные числа и их применение при визуализации многомерных данных.
14. Методы и алгоритмы оцифровки графиков
15. Методы и алгоритмы обработки изображений
16. Простые и сложные признаки и способы щценки информативности
17. Алгоритмы поиска систем информативных признаков.
18. Матрица объект-признак и её статистические характеристики.

19. Проблема сжатия данных
20. Разнотипные данные и методы их обработки
21. Задача поиска логических закономерностей
22. Методы классификации и прогнозирования
23. Задачи кластерного анализа
24. Иерархические и итеративные методы кластеризации
25. Особенности кластеризации в качественных количественных шкалах
26. Кластеризация данных по матрице объект-признак.
27. Кластеризация данных по матрице матрице связи.
28. Назначение компонентного и факторного анализа.
29. Сходство и различие компонентного и факторного анализа.
30. Применение компонентного и факторного анализа к задачам ИАД.
31. Методы распознавания образов с учителем и без учителя.
32. Задачи принятия решений.
33. Метод анализа иерархий.
34. Модификации метода анализа иерархий в интересах реализации интеллектуальных подсказок пользователям.
35. Основные понятия когнитивного моделирования
36. Инструментальные средства ИАД применительно задачам СППР
37. Направления развития ИАД
39. Краткая история нейрокомпьютинга.
40. Задачи ИАД на основе искусственных нейронных сетей.
41. Место нейронных сетей среди других методов решения задач
42. Информационный подход к моделированию нейрона.
43. Биологический подход к моделированию нейрона.
44. Структура искусственной нейронные сети.
45. Структура двухкровневого персептрона, многоуровневого персептрона (МСП).
46. Особенности структуры нейронных сетей и ее влияние на свойства сети.
47. Алгоритм решения задач с помощью МСП.
48. Классификация задач решаемых с помощью МСП.
49. Постановка задач распознавания, аппроксимации, прогнозирования. Примеры задач.
50. Топологии нейронных сетей.