

Кафедра цифровой экономики

Лабораторный практикум по дисциплине

Анализ больших данных

Методические указания

к лабораторным работам для студентов

направлений подготовки:

38.03.05 «Бизнес-информатика» (степень - бакалавр)

Ульяновск
2018

ПРЕДИСЛОВИЕ

Лабораторный практикум содержит методические указания к лабораторным работам по курсу «Анализ больших данных».

Содержание практикума направлено на:

формирование навыков создания, наполнения и использования хранилищ данных для решения основных задач Data Mining;

создание законченных аналитических решений.

Лабораторный практикум состоит из восьми лабораторных работ.

Лабораторная работа №1 направлена на знакомство с аналитической платформой DEDUCTOR

Лабораторная работа №2 посвящена освоению навыков применения факторного и корреляционного анализа.

Лабораторная работа №3 направлена на получение навыков разбиения данных, квантования и фильтрации для трансформации данных

Лабораторная работа №4 направлена на освоение инструмента позволяющих решать и использовать математические функции.

Лабораторная работа №5 направлена на изучение ассоциативные правила и использовать визуализаторы «Популярные наборы», «Правила», «Дерево правил», «Что-если».

Лабораторная работа №6 посвящена применению методов Data Mining для решения задач прогнозирования временных рядов на примере построения модели.

Лабораторная работа №7 направлена на получение знаний о применении обработчика «скрипт» для решения задач прогнозирования на примере прогноза продаж.

Лабораторная работа №8 направлена на получение навыков решения задач кластеризации с применением карт Кохонена.

Все лабораторные работы имеют перечень вопросов для

самоконтроля и список рекомендуемой литературы.

Базовые фундаментальные знания, полученные при изучении курса «Анализ больших данных», позволяют перейти к изучению дисциплин:

«Современные финансовые инструменты социального предпринимательства»;

«Управление ИТ сервисами и контентом»;

«Инструменты цифровой экономики».

Знания, навыки и умения, приобретенные в результате прохождения курса, будут востребованы при выполнении курсовых и выпускной квалификационной работ, связанных с интеллектуальной обработкой больших объемов информации, информационными системами поддержки принятия решений.

РАЗВИТИЕ И НАЗНАЧЕНИЕ СПС DEDUCTOR

Deductor - это аналитическая платформа, основа для создания законченных прикладных решений в области анализа данных. Реализованные в Deductor технологии позволяют на базе единой архитектуры пройти все этапы построения аналитической системы: от консолидации данных до построения моделей и визуализации полученных результатов.

До появления аналитических платформ анализ данных осуществлялся в основном в статистических пакетах. Их использование требовало высокой квалификации пользователя.

Большинство алгоритмов, реализованных в статистических пакетах, не позволяло эффективно обрабатывать большие объемы информации. Для автоматизации рутинных операций приходилось использовать встроенные языки программирования.

В конце 80-х гг. произошел стремительный рост объемов информации, накапливаемый на машинных носителях и возросли потребности бизнеса по применению анализа данных.

Ответом этому стало появление новых парадигм в анализе: хранилища данных, машинное обучение, Data Mining, Knowledge Discovery in Databases.

Это позволило популяризировать анализ данных, вывести его на промышленную основу и решить огромное число бизнес-задач с большим экономическим эффектом.

Венцом развития анализа данных стали специализированные программные системы - аналитические платформы, которые полностью автоматизировали все этапы анализа от консолидации данных до эксплуатации моделей и интерпретации результатов.

Первая версия Deductor увидела свет в 2000 г. и с тех пор идет непрерывное развитие платформы. В 2007 г. выпущена пятая по счету версия системы, в 2009 г. - версия 5.2.

Сегодня Deductor - это яркий представитель как настольной, так и корпоративной системы анализа данных последнего поколения.

Общие сведения о Deductor

Аналитическая платформа Deductor состоит из пяти частей:

Warehouse - хранилище данных, консолидирующее информацию из разных источников;

Studio - приложение, позволяющее пройти все этапы построения прикладного решения, рабочее место аналитика;

Viewer - рабочее место конечного пользователя, одно из средств тиражирования знаний (т.е. когда построенные аналитиком модели используют пользователи, не владеющие технологиями анализа данных);

Server - служба, обеспечивающая удаленную аналитическую обработку данных;

Client - клиент доступа к Deductor Server. Обеспечивает доступ к серверу из сторонних приложений и управление его работой.

Существует три типа варианта поставки платформы Deductor:

Enterprise;

Professional;
Academic.

В зависимости от типа поставки набор доступных компонентов может различаться.

Версия Enterprise предназначена для корпоративного использования.

В ней присутствуют:

Серверные компоненты Deductor Server и Deductor Client.

Интерфейс доступа к Deductor через механизм OLE Automation.

Традиционное хранилище данных Deductor Warehouse на трех СУБД: Firebird, MS SQL, Oracle.

Виртуальное хранилище данных Deductor Virtual Warehouse.

Версия Professional предназначена для небольших компаний и однопользовательской работы. В ней отсутствуют серверные компоненты, поддержка OLE, виртуальное хранилище, а традиционное хранилище данных можно создавать только на СУБД FireBird. Автоматизация выполнения сценариев обработки данных осуществляется только через пакетный режим.

Версии Professional и Enterprise требуют установки драйверов Guardant для работы с лицензионным ключом.

Версия Academic предназначена для образовательных и обучающих целей. Ее функционал аналогичен версии Professional за исключением:

отсутствует пакетный запуск сценариев, т.е. работа в программе может вестись только в интерактивном режиме;

отсутствует импорт из промышленных источников данных: 1С, СУБД, файлы MS Excel, Deductor Data File;

некоторые другие возможности.

Категории

пользователей Deductor

В процессе развертывания и использования аналитической платформы с ней взаимодействуют различные категории пользователей. Можно выделить четыре основные категории:

- аналитик;
- пользователь;
- администратор;
- программист.

Функции аналитика:

создание в Deductor Studio сценариев - последовательности шагов, которую необходимо провести для получения нужного результата.

построение, оценка и интерпретация моделей.

настройка панели отчетов для пользователей Deductor Viewer.

настройка сценария на поточную обработку новых данных. Функции пользователя:

просмотр готовых отчетов в Deductor Viewer.

Функции администратора:

установка компонентов Deductor на рабочих местах и сервера ключей Guardant при необходимости.

развертывание традиционного хранилища данных на сервере.

контроль процедур регулярного пополнения хранилища данных.

конфигурирование сервера Deductor Server.

настройка пакетной и/или серверной обработки сценариев Deductor.

оптимизация доступа к источникам данных, в том числе к хранилищу данных.

Функции программиста:

интеграция Deductor с источниками и приемниками данных.

вызов Deductor из внешних программ различными способами, в том числе взаимодействие с Deductor Server.

Такая работа как проектирование и наполнение хранилище данных часто выполняется коллективно аналитиком, администратором и программистом. Аналитик проектирует семантический слой хранилища данных, то есть определяет, какие данные необходимо иметь в хранилище. Администратор создает хранилище данных и наполняет его данными. Программист при необходимости создает программные модули, выполняющие выгрузку информации из учетных систем в промежуточные источники (так называемые транспортные таблицы).

Установка Deductor

Установку Deductor рекомендуется проводить администратору системы, однако, при наличии прав администратора в Windows это может сделать и аналитик. Установка может быть произведена на компьютер с операционной системой MS Windows 2000 и выше. Системные требования к компьютеру изложены в справочной системе.

Для установки Deductor Professional/Academic запустите файл инсталлятора и следуйте инструкциям по установке. На странице Выбор компонентов программы установки предоставляется выбор, какой набор компонентов пакета Deductor необходимо установить на компьютер. В выпадающем списке можно выбрать predetermined configurations установки платформы, и программа установки сама предложит нужный набор компонентов.

После установки программ серии Professional и Enterprise дополнительно потребуется настроить работу с электронным ключом защиты от копирования. Установку и подключение электронного ключа осуществляет администратор.

Существуют два вида ключей - локальный и сетевой.

Локальный ключ устанавливается на том же компьютере, что и Deductor, и работать с ним можно только с этой рабочей станции. Сетевой ключ устанавливается на сервере, и к нему могут подключаться несколько пользователей одновременно (количество пользователей ограничивается типом приобретаемой лицензии).

При каждом запуске Deductor пытается найти доступный электронный ключ. В случае если ключ не найден, могут появиться следующие сообщения об ошибке:



ЗНАКОМСТВО С АНАЛИТИЧЕСКОЙ ПЛАТФОРМОЙ DEDUCTOR

1. Цель и содержание работы

Цель работы - ознакомиться с архитектурой, основными частями и пользовательским интерфейсом Deductor, получить навыки импорта данных, парциальная предобработка, восстановление пропущенных данных, удаление аномалий, спектральная обработка, удаление шумов.

Содержание работы:

Используя текстовый редактор "блокнот" создать файл «TestForPPP.txt» (рис.1), содержащий такие столбцы, как «Аргумент», «Синус», «Аномалии», «Больше шумы», «Средние шумы», «Малые шумы». Разделителем между столбцами является знак табуляции. Столбцу «Аргумент» присваиваются значения от 0 до 2,96 с шагом 0,02. В столбце «Синус» принимаются значения синуса (9 знаков после запятой). При любых двадцати значениях аргумента, ввод данных в значениях синуса пропустить. Значения столбца «Аномалии» равны значениям столбца «Синус», но не имеют пропущенных данных, однако 10 значений резко отклоняются от истинного значения синуса аргумента.

Значения столбцам «Больше шумы», «Средние шумы», «Малые шумы» имеют значения близкие к значению синуса аргумента, но имеют некоторое отклонение (дисперсию и выбираются из промежутка -1,5 до 1,5)(рис.1.).

Выполнить импорт данных, созданного файла, обработку данных, восстановить пропущенные значения синуса, выполнить парциальную обработку, удалить аномалии и шумы.

Аргумент	Синус	Аномалии	Большие шумы	Средние шумы	Малые шумы
0,14	0,139543115	0,139543115	-0,006298541	0,126864534	0,174892379
0,16	0,159318207	0,159318207	0,33294777	0,085238625	0,145575649
0,18	0,179029573	0,179029573	0,279505602	0,085723008	0,134508346
0,2	0,198669331	0,198669331	0,258280061	0,277157184	0,176827573
0,22	0,218229623	0,5	0,139744277	0,28969863	0,231458805
0,24	0,237702626	0,237702626	0,317646146	0,177087062	0,282627785
0,26	0,257080552	0,257080552	0,266913669	0,19389017	0,288192043
0,28	0,276355649	0,276355649	0,131932825	0,325878695	0,243445006
0,3		0,295520207	0,403496144	0,30362014	0,30123014
0,32		0,314566561	0,430780976	0,306273759	0,312952364

Рис.1. Пример заполнения файла «TestForPPP.txt»

Продолжительность работы 4 часа

2. Теоретические сведения

Deductor Studio - программа, реализующая функции импорта, обработки, визуализации и экспорта данных. Deductor Studio может функционировать и без хранилища данных, получая информацию из любых других источников, но наиболее оптимальным является их совместное использование. В Deductor Studio включен полный набор механизмов, позволяющий получить информацию из произвольного источника данных, провести весь цикл обработки (очистку, трансформацию данных, построение моделей), отобразить полученные результаты наиболее удобным образом (OLAP, диаграммы, деревья...) и экспортировать результаты на сторону. Это полностью соответствует концепции извлечения знаний из баз данных (KDD).

Интерфейс Deductor Studio состоит из главного окна, внутри которого располагаются панели сценариев, отчетов, источников данных и результаты моделирования (таблицы, графики, кросс-диаграммы, правила и т.д.).

Для автоматизации получения данных из любого источника, предусмотренного в системе, следует использовать мастер импорта. На первом шаге мастера импорта открывается список всех

предусмотренных в системе типов источников данных. Число шагов мастера импорта, а также набор настраиваемых параметров отличается для разных типов источников.

Мастер обработки предназначен для настройки всех параметров выбранного алгоритма.

Пошаговом режиме выбрать и настроить наиболее удобный способ представления данных можно с помощью мастера отображений. В зависимости от обработчика, в результате которого была получена ветвь сценария, список доступных для него видов отображений будет различным. Например, после построения деревьев решений их можно отобразить с помощью визуализаторов «Деревья решений» и «Правила». Эти способы отображения не доступны для других обработчиков.

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



Рис.2. Импорт данных

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).

Часто исходные данные являются не достаточно полными либо

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

Парциальная предобработка

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

Восстановление пропущенных данных

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

пропущенных данных наиболее вероятные значения, основываясь на всей выборке.

Удаление аномалий

Аномалии - это отклонения от нормального поведения чего-либо. Это может быть, например, резкое отклонение величины от ее ожидаемого значения.

Автоматическое редактирование аномальных значений осуществляется с использованием методов робастной фильтрации, в основе которых лежит использование робастных статистических оценок, таких, например, как медиана. При этом можно задать эмпирически подобранный критерий того, что считать аномалией. Например, задание в качестве степени подавления аномальных данных

значения «слабая» означает наиболее терпимое отношение к величине допустимых выбросов.

По существу аномалии вообще не должны оказывать никакого влияния на результат. Если же они присутствуют при построении модели, то оказывают на нее весьма большое влияние. Т.е. предварительно их необходимо устранить. Также они портят статистическую картину распределения данных. Данные с аномалиями, а также гистограмма их распределения представлены на рис.3:

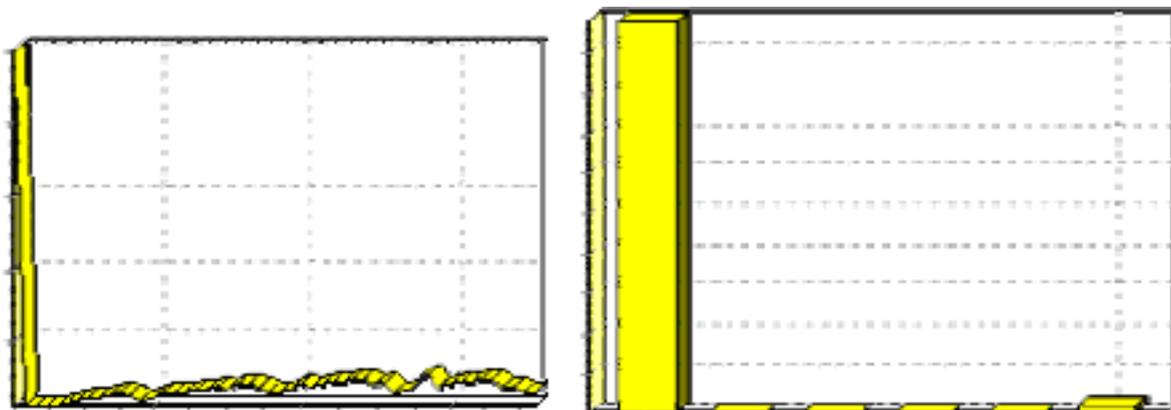


Рис.3. Данные с аномалиями и гистограмма распределения

Очевидно, что аномалии не позволяют определить как характер самих данных, так и статистическую картину. Данные после устранения аномалий представлены на рис.4.

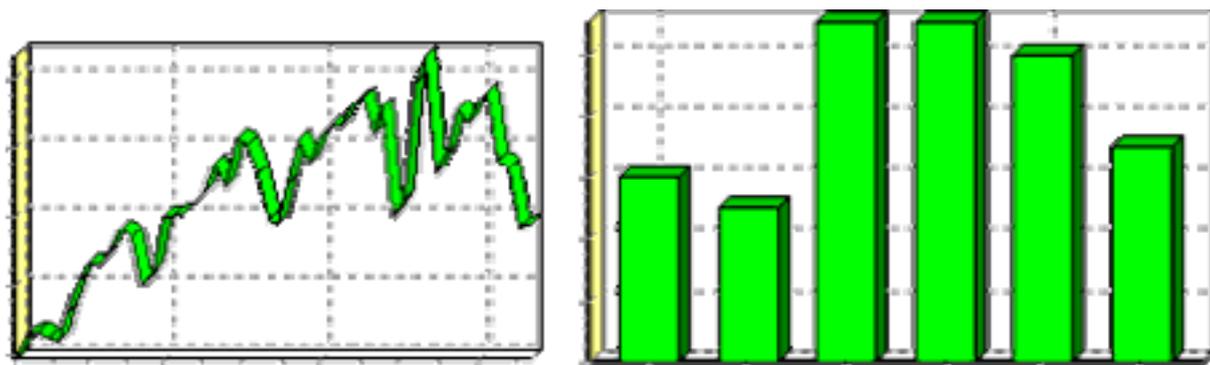


Рис.4. Результат удаления аномалий

Спектральная обработка.

Сглаживание данных применяется для удаления шумов из исходного набора. 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.

Удаление шумов

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

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 некоторых случаях неплохие результаты удаления шумов дает вейвлет преобразование.

3. Порядок выполнения работы

Импорт осуществляется путем вызова мастера импорта на панели «Сценарии»

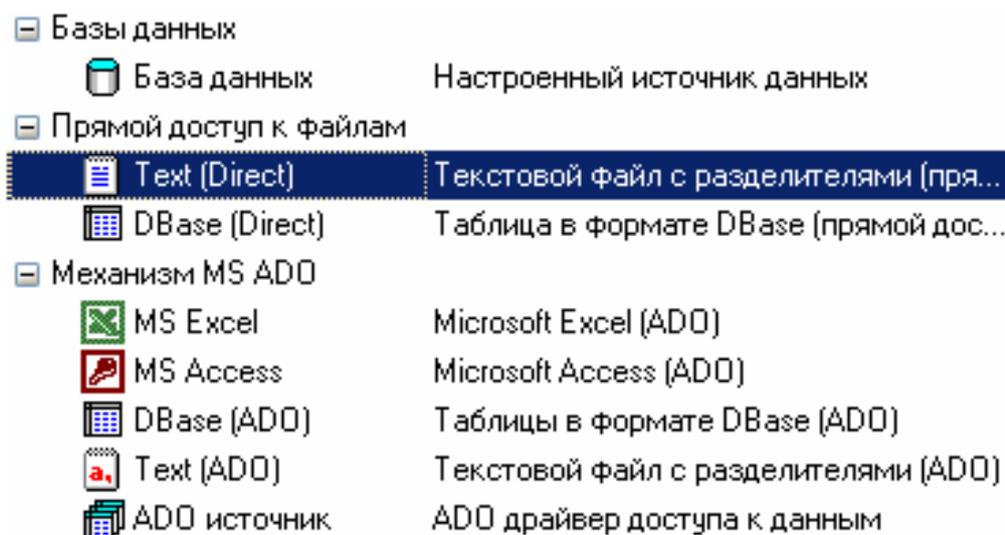


Рис.5. Импорт данных

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

Имя файла: C:\SSBG\PROJECTS\Deductor Studio\Samples\TestForPPP.txt

Просмотр выбранного файла

Аргумент	Синус	Аномалии	Большие шумы	Средние шумы	Малые шумы
0	0	0	0,010766982	0,096101077	-0,035542974
0,02	0,019998667	0,019998667	0,121877199	-0,072822839	0,06516644
0,04	0,039989334	0,039989334	-0,126413455	0,082251695	0,05384534
0,06	0,059964006	0,059964006	-0,097298964	-0,039127784	0,06097118
0,08	0,079914694	0,084816585	0,141709041	0,092523579	
0,1	0,099833417	0,099833417	-0,07416128	0,02437907	0,05282257
0,12	0,119712207	0,119712207	0,202327993	0,124838473	0,14734831
0,14	0,139543115	0,139543115	-0,006298541	0,126864534	0,17489237
0,16	0,159318207	0,159318207	0,33294777	0,085238625	0,14557564
0,18	0,179029573	0,179029573	0,279505602	0,085723008	0,13450834
0,2	0,198669331	0,198669331	0,258280061	0,277157184	0,17682757

Рис.6. Окно просмотра

Далее перейдем к настройке параметров импорта (рис.7). 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.

Начать импорт со строки: Первая строка является заголовком

Добавить первичный ключ

Символом-разделителем является

Символ табуляции
 Пробел
 Точка
 Точка с запятой
 Запятая
 Другой

Считать последовательные разделители одним

Ограничитель строк:

Разделитель целой и дробной частей числа:

Разделитель компонентов даты: Формат даты:

Разделитель компонентов времени: Формат времени:

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.

На следующем шаге мастера предоставляется возможность настроить имя, название (метку), размер, тип данных, вид данных и назначение. Некоторые свойства (например, тип данных) можно задавать для выделенного набора столбцов(рис.8).

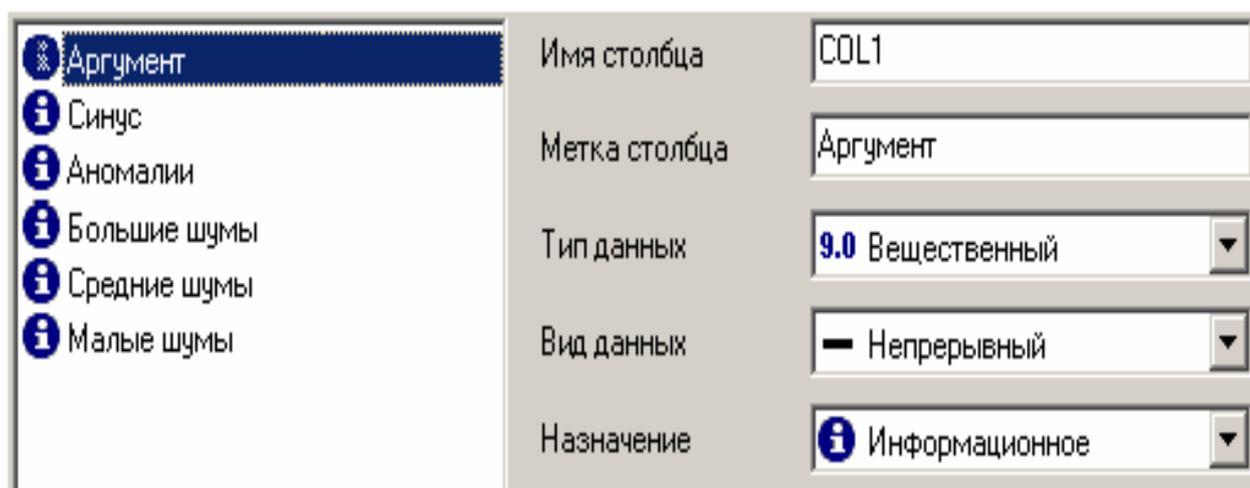


Рис.8. Настройка свойств столбцов

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

После импорта данных на следующем шаге мастера необходимо выбрать способ отображения данных (рис.9). В данном случае самым информативным является диаграмма, выберем ее.

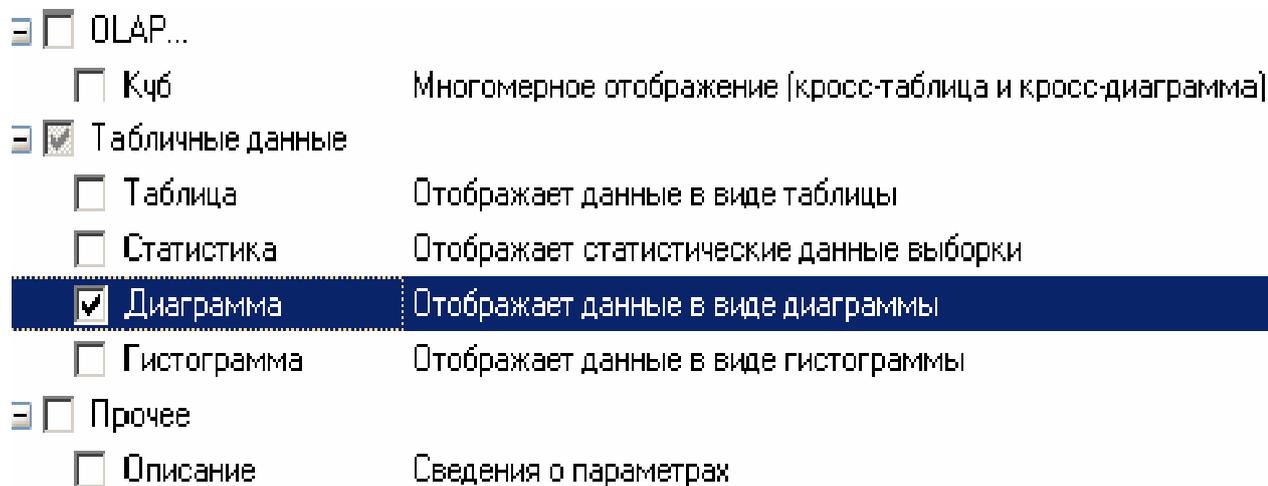


Рис.9. Способ отображения данных

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» (рис.10) и тип диаграммы «Линии».

<input type="checkbox"/> Аргумент	9.0 Вещественный	
<input checked="" type="checkbox"/> Синус	9.0 Вещественный	
<input type="checkbox"/> Аномалии	9.0 Вещественный	
<input type="checkbox"/> Большие шумы	9.0 Вещественный	
<input type="checkbox"/> Средние шумы	9.0 Вещественный	
<input type="checkbox"/> Малые шумы	9.0 Вещественный	

Рис.10. Настройка способа отображения

На последнем шаге мастера необходимо указать название ветки в дереве сценариев. 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(рис.11)

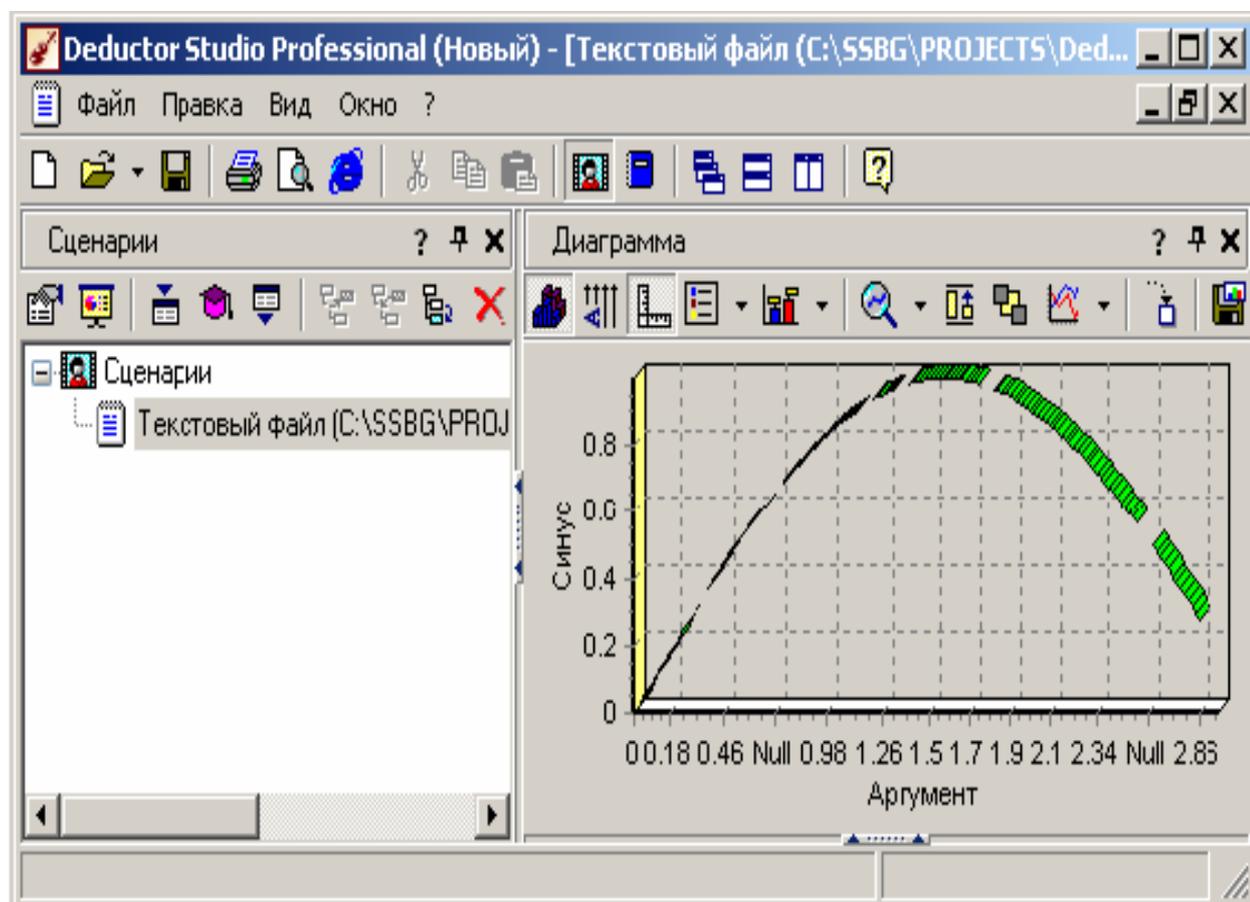


Рис.11. Главное окно программы

Далее сделаем обработку данных из файла «TestForPPP.txt». Он содержит таблицу со следующими полями: «АРГУМЕНТ» - аргумент, «СИНУС» - значения синуса аргумента (некоторые значения пустые), «АНОМАЛИИ» - синус с выбросами, «БОЛЬШИЕ ШУМЫ» - значения синуса с большими шумами, «СРЕДНИЕ ШУМЫ» - значения синуса со средними шумами, «МАЛЫЕ ШУМЫ» - значения синуса с малыми шумами. Все данные можно увидеть на диаграмме после импорта из текстового файла(рис.12).

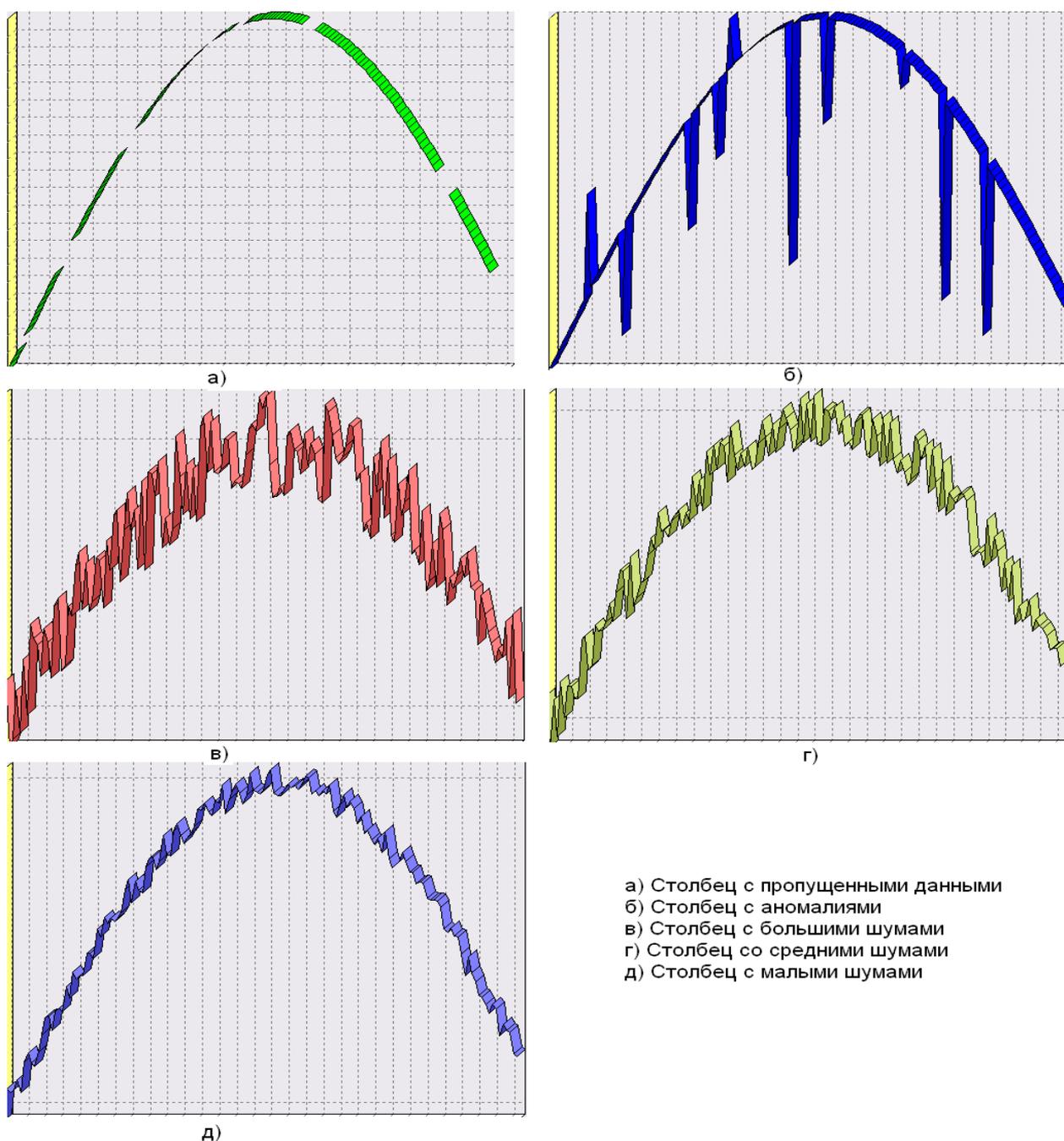


Рис.12. Диаграммы

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

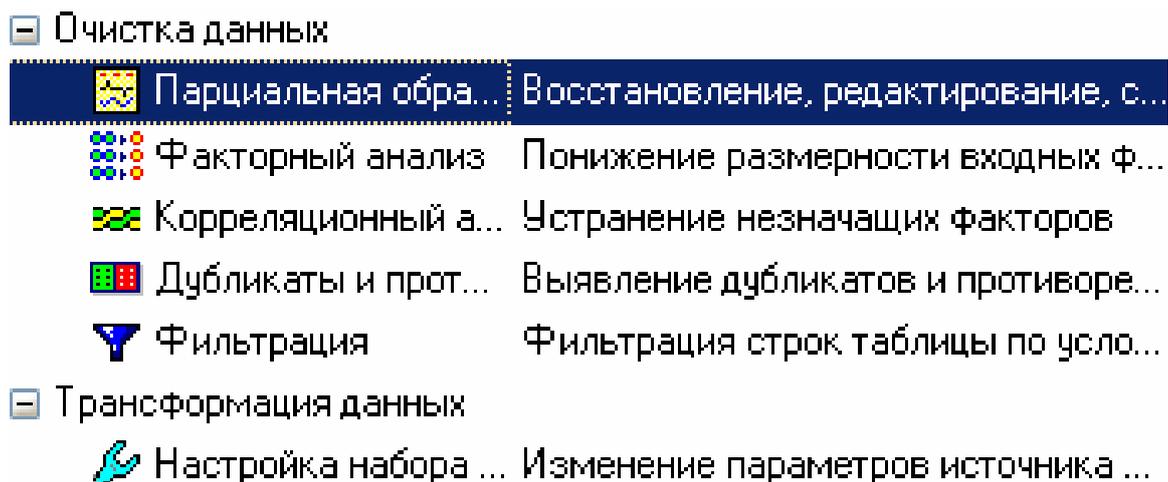


Рис.13. Запуск мастера парциальной обработки

Поскольку данные в исходном наборе упорядочены, на следующем шаге мастера обработки выделим поле «СИНУС» и укажем для него тип обработки «Аппроксимация» (рис.14). Так как в данном случае больше ничего не требуется, то остальные параметры обработки оставляем отключенными. Перейдя на страницу запуска процесса обработки, выполняем ее, нажав на пуск, и далее выбираем тип визуализации обработанных данных (как в примере импорта).

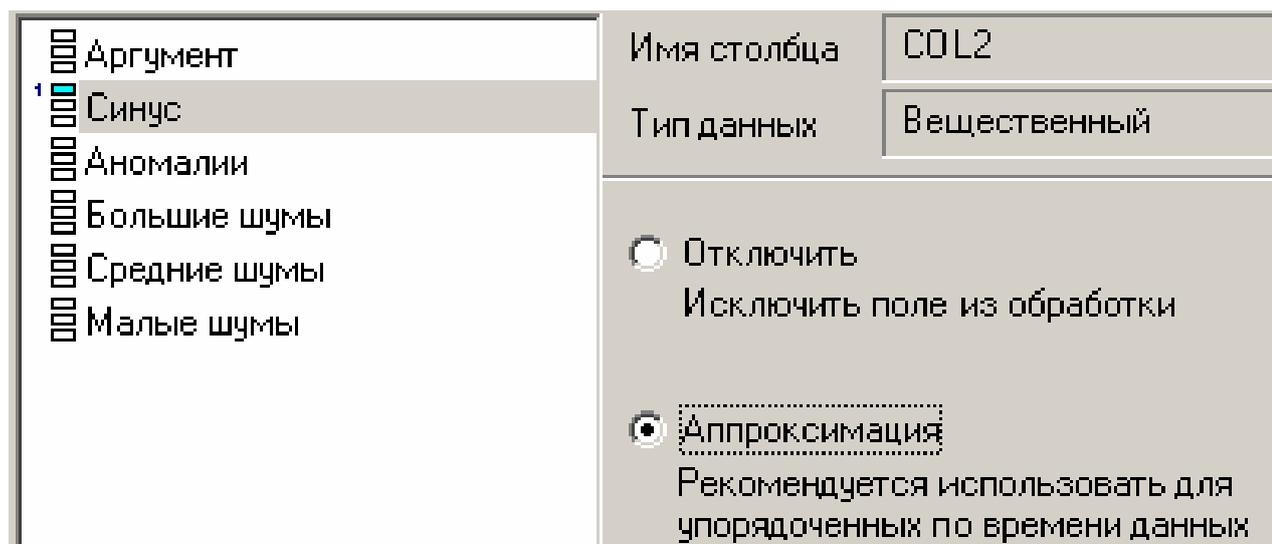


Рис.14 Окно мастера обработки

После выполнения процесса обработки, как видно из рисунка 15, на диаграмме пропуски в данных исчезли, что и было необходимо сделать.

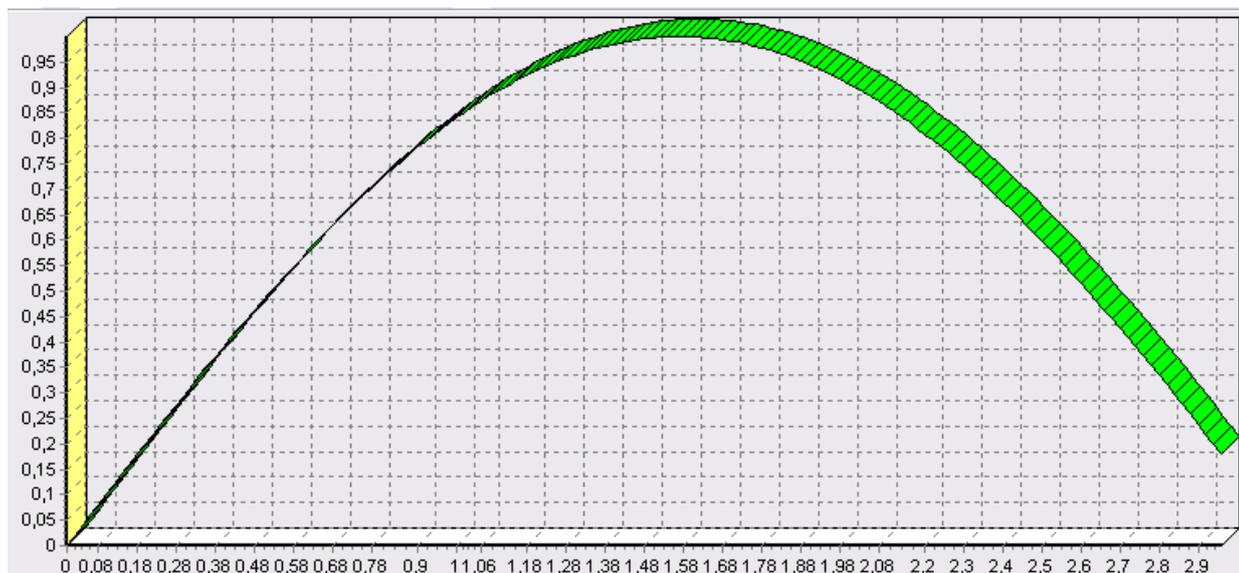


Рис.15. Диаграмма после процесса обработки

Далее удалим аномалий из поля «АНОМАЛИИ» импортированной таблицы.

В мастере парциальной предобработки на третьем шаге выбираем поле «АНОМАЛИИ» и указываем ему тип обработки «Удаления аномальных явлений», степень подавления «Большая» (рис.16).

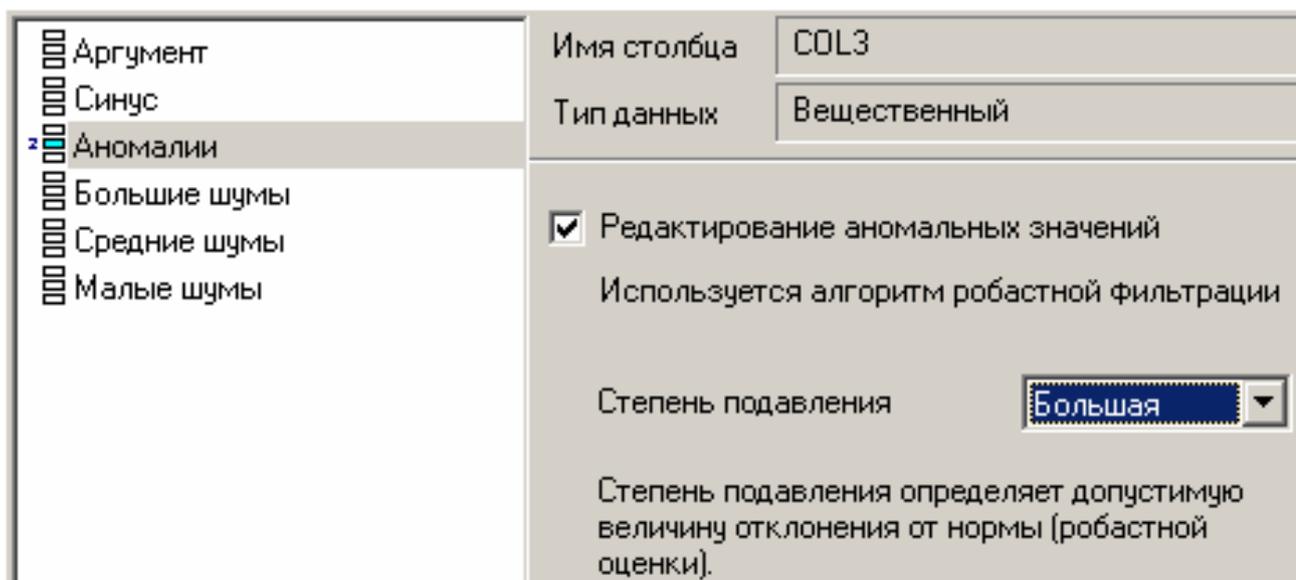


Рис.16. Окно мастера обработки

Так как больше никаких обработок не планировалось, то переходим на шаг запуска процесса обработки и нажимаем «Пуск».

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

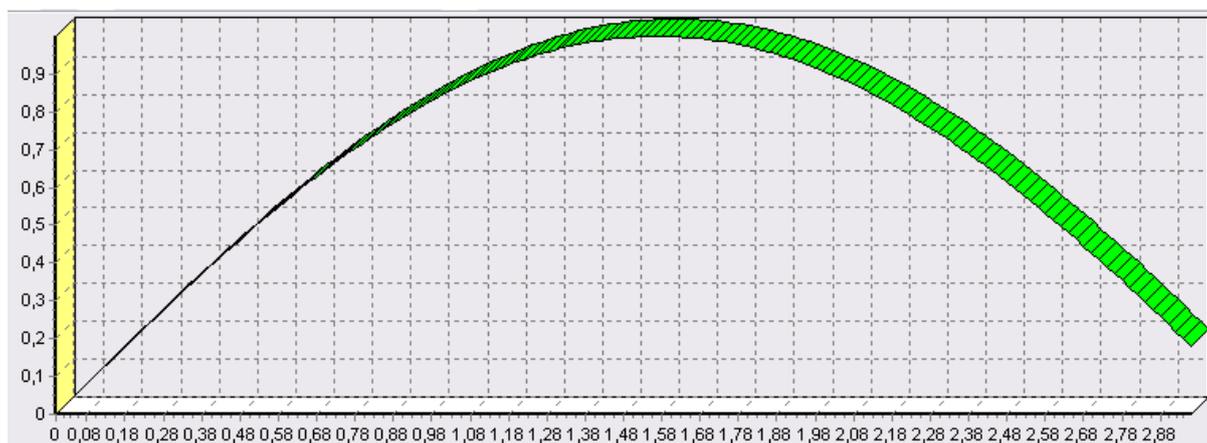


Рис.17. Итоговая диаграмма

Как видно на рисунке 17, аномалии были устранены, однако небольшие возмущения остались. Сгладим их при помощи парциальной обработки. Для этого после удаления аномалий вновь запустим мастер парциальной обработки. В нем на четвертом шаге выберем поле «АНОМАЛИИ» и укажем ему тип обработки

«Вейвлет преобразование» с параметрами по умолчанию (глубина разложения 3, порядок вейвлета 6).

<ul style="list-style-type: none"> ▢▢▢ Аргумент ▢▢▢ Синус <li style="background-color: #d3d3d3;">▢▢▢ Аномалии ▢▢▢ Большие шумы ▢▢▢ Средние шумы ▢▢▢ Малые шумы 	Имя столбца	COL3
	Тип данных	Вещественный
<input type="radio"/> Отключить		
<input type="radio"/> Сглаживание данных Полоса пропускания <input type="text" value="50"/>		
<input type="radio"/> Вычитание шума Степень вычитания шума <input type="text" value="Малая"/>		
<input checked="" type="radio"/> Вейвлет преобразование		
Глубина разложения <input type="text" value="3"/>		
Порядок вейвлета <input type="text" value="6"/>		

Рис.18. Окно мастера обработки

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

После обработки можно убедиться на диаграмме в отсутствии выбросов и сравнить результат с эталонным значением синуса (столбец «СИНУС»). На рисунке 19 зеленый (светлый) график - значения синуса, синий (темный) - значения сглаженного синуса после устранения аномалий.

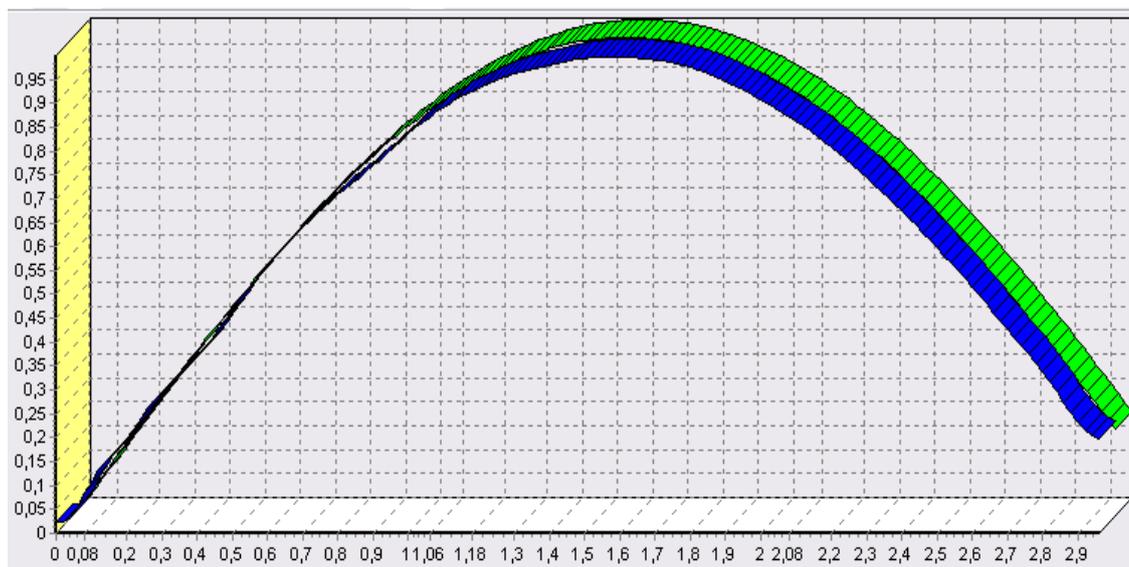


Рис.19. Диаграмма после обработки

В примере по парциальной обработке, как было показано ранее, есть 3 столбца с шумами: «БОЛЬШИЕ ШУМЫ», «СРЕДНИЕ ШУМЫ», и «МАЛЫЕ ШУМЫ» - соответственно синус с большими, средними и малыми шумами. Ясно, что для дальнейшей работы с данными эти шумы необходимо устранить.

Таким образом, в мастере парциальной обработки на четвертом шаге выберем по очереди поля «БОЛЬШИЕ ШУМЫ», «СРЕДНИЕ ШУМЫ» и «МАЛЫЕ ШУМЫ», зададим тип обработки «Вычитание шума» и укажем степень подавления - «большая», «средняя» и «малая» соответственно. После выполнения обработки на диаграмме можно просмотреть полученные результаты(рис.20).

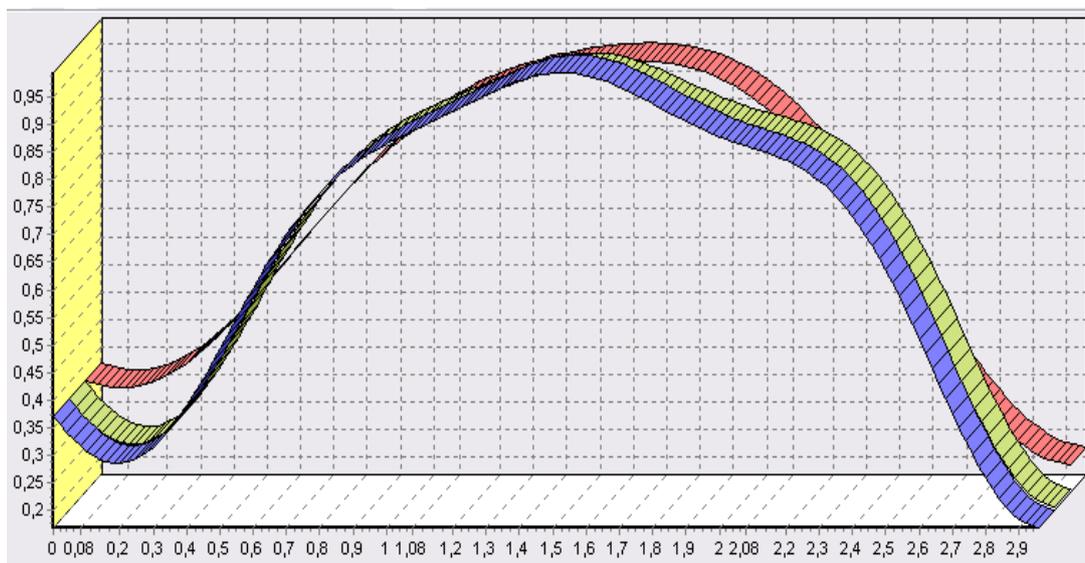


Рис.20. Результат парциальной обработки

Теперь удалим шумы с помощью вейвлет преобразования. В мастере парциальной обработки выберем поля «БОЛЬШИЕ ШУМЫ», «СРЕДНИЕ ШУМЫ» и «МАЛЫЕ ШУМЫ», укажем тип обработки «Вейвлет преобразование», оставив параметры обработки по умолчанию (глубина разложения - 3, порядок вейвлета - 6). На диаграмме можно убедиться в том, что данные сгладились (рис.21). Повысить качество сглаживания шумов таким способом можно, путем подбора удовлетворительных параметров обработки.

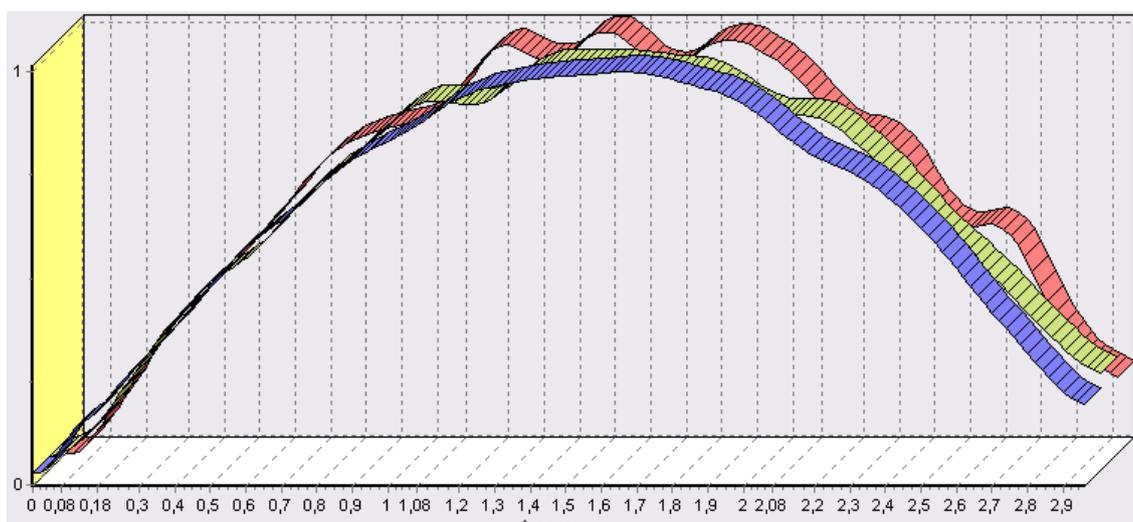


Рис.21. Результат удаления шума

4. Содержание отчета

Отчет по лабораторной работе представляется в виде документа Word. В состав документа входят:

1. Название работы
2. Цель работы
3. Копии экрана, иллюстрирующие выполнение задания лабораторной работы
4. Выводы по работе

5. Контрольные вопросы

1. Для чего предназначен мастер импорта программы Deductor Studio?
2. Для чего предназначен мастер обработки программы Deductor Studio?
3. Для чего предназначен мастер отображений программы Deductor Studio?
4. Для чего следует проводить подготовку данных для анализа?
5. Что такое шумы и аномалии в данных?
6. Какими методами можно убрать шумы в системе Deductor?
7. Какими методами можно убрать аномалии данных в системе Deductor?
8. Для чего используется парциальная предобработка?
9. Для чего используется спектральная обработка?
10. Какие виды спектральной обработки имеются в системе Deductor?

6. Список рекомендуемой литературы

1. Барсегян А.А., Куприянов М.С., Степаненко В.В., Холод И.И. Методы и модели анализа данных: OLAP и Data Mining. - Спб.: БХВ-Петербург, 2004. - 336 с.: ил
2. Загоруйко Н.Г. Прикладные методы анализа данных и знаний. - Новосибирск: Изд-во Ин-та математики, 1999. - 270 с.
3. Тюрин Ю.Н., Макаров А.А. Статистический анализ данных на компьютере / Под ред. В. Э. Фигурнова - М.: ИНФРА-М, 1998. - 528 с., ил.

ОБРАБОТКА ДАННЫХ ПРИ ФАКТОРНОМ И КОРРЕЛЯЦИОННОМ АНАЛИЗЕ

1. Цель и содержание работы

Цель работы - освоить и закрепить навыки применения факторного и корреляционного анализа.

Содержание работы:

В блокноте создать файл «TestForCPP.txt», содержащий такие столбцы, как «Аргумент», «Фактор1», «Фактор2», «Фактор3», «Результат1», «Результат2» (рис.1). Разделителем между столбцами является знак табуляции. Столбец «Аргумент» заполняется значениями в диапазоне от 0 до 6,25 с шагом 0,05. Столбцы «Фактор1», «Фактор2», «Фактор3» являются входными значениями и задаются в диапазоне от -1 до 1 (в любом порядке). Столбцы «Результат1», «Результат2» являются выходными значениями и принимаются равными значениям столбцов «Фактор1», «Фактор2» соответственно.

Значения столбцам нужно задавать такие, чтобы получилась циклическая функция.

Выполнить обработку данных факторным и корреляционным анализом.

Аргумент	Фактор1	Фактор2	Фактор3	Результат1	Результат2
0,00	0,00	1,00	0,14	0,00	1,00
0,05	0,05	1,00	0,09	0,05	1,00
0,10	0,10	1,00	0,04	0,10	1,00
0,15	0,15	0,99	-0,01	0,15	0,99
0,20	0,20	0,98	-0,06	0,20	0,98

0,25	0,25			0,97	-0,11	0,25
0,97						
0,30	0,30	0,96	-0,16	0,30		0,96
0,35	0,34	0,94	-0,21	0,34		0,94
0,40	0,39	0,92	-0,26	0,39		0,92
0,45	0,43	0,90	-0,30	0,43		0,90
0,50	0,48	0,88	-0,35	0,48		0,88
0,55	0,52	0,85	-0,40	0,52		0,85
0,60	0,56	0,83	-0,44	0,56		0,83
0,65	0,61	0,80	-0,49	0,61		0,80

Рис.1. Пример заполнения файла «TestForCPP.txt»

Продолжительность работы - 6 часов.

2. Теоретические сведения

Факторный анализ. Факторный анализ - группа методов многомерного статистического анализа, которые позволяют представить в компактной форме обобщенную информацию о структуре связей между наблюдаемыми признаками изучаемого объекта на основе выделения некоторых непосредственно не наблюдаемых факторов. Факторный анализ служит для понижения размерности пространства входных факторов. Обработку можно выполнять как в автоматическом режиме (с указанием порога значимости), так и самостоятельно (основываясь на значениях матрицы значимости).

Первым этапом факторного анализа является выбор новых признаков, которые являются линейными комбинациями прежних и «вбирают» в себя большую часть общей изменчивости входных факторов. Поэтому они содержат большую часть информации, заключенной в первоначальных данных. В обработчике «Факторный анализ» это осуществляется с помощью метода главных компонент. Этот метод сводится к выбору новой ортогональной системы

координат в пространстве наблюдений. В качестве первой главной компоненты избирают направление, вдоль которого массив данных имеет наибольший разброс. Выбор каждой последующей главной компоненты происходит так, чтобы разброс данных вдоль нее был максимальным и чтобы эта главная компонента была ортогональна другим главным компонентам, выбранным прежде.

Корреляционный анализ. Корреляционный анализ-совокупность основанных на математической теории корреляции методов обнаружения корреляционной зависимости между двумя случайными признаками или факторами. Корреляционный анализ применяется для оценки зависимости выходных полей данных от входных факторов и устранения незначущих факторов. Принцип корреляционного анализа состоит в поиске таких значений, которые в наименьшей степени взаимосвязаны с выходным результатом. Такие факторы могут быть исключены из результирующего набора данных практически без потери полезной информации. Критерием принятия решения об исключении является порог значимости. Если степень взаимозависимости между входным и выходным факторами меньше порога значимости, то соответствующий фактор отбрасывается как незначущий.

3. Порядок выполнения работы

Выполним обработку данных из файла «TestForCPP.txt» при помощи факторного анализа. Он содержит таблицу со следующими полями: «АРГУМЕНТ» - аргумент, «ФАКТОР1», «ФАКТОР2», «ФАКТОР3» - входные значения, «РЕЗУЛЬТАТ1», «РЕЗУЛЬТАТ2» - выходные значения (рис.2).

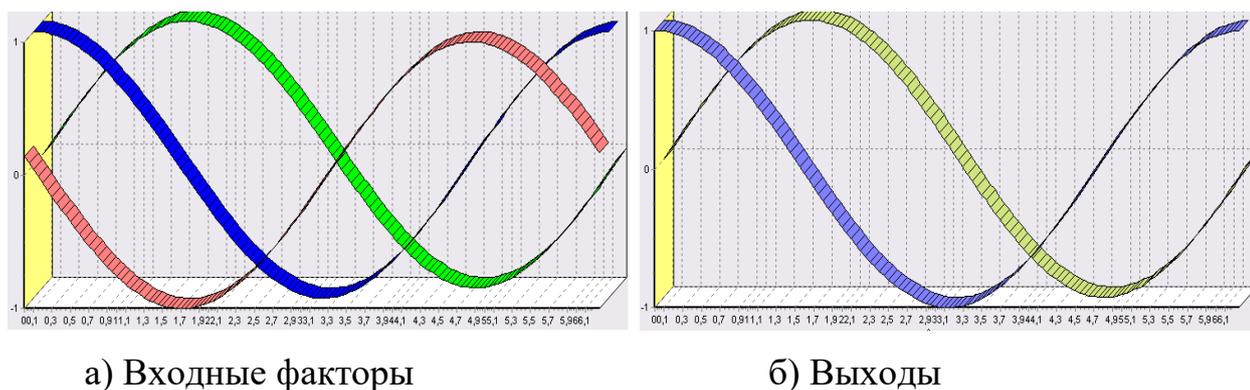


Рис.2. Данные для обработки

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

Следующий шаг предлагает запустить процесс понижения размерности пространства входных факторов. После завершения процесса на следующем шаге выбираем, какие из полученных в результате обработки факторы оставить для дальнейшей работы (рис.3). Это делается путем указания необходимого порога значимости (по умолчанию порог значимости равен 90%, не будем его менять).

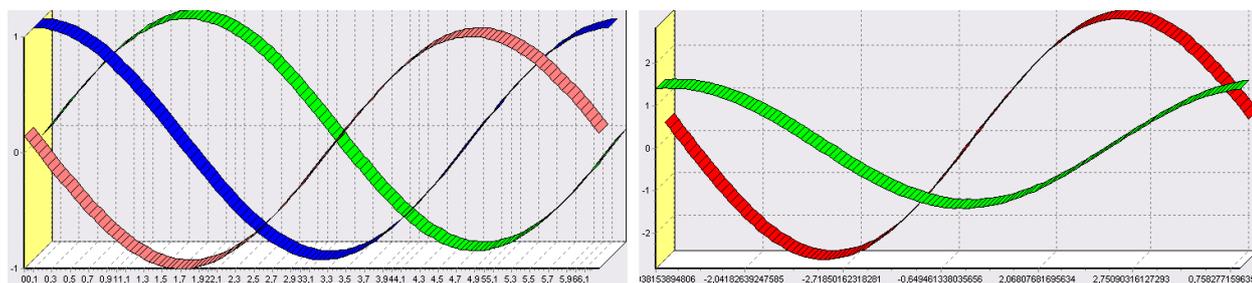
Главные компоненты	Собственные значения	Вклад в результат	Суммарный вклад
<input checked="" type="checkbox"/> Значение 1	2.000	66.66 %	66.66 %
<input checked="" type="checkbox"/> Значение 2	1.000	33.34 %	100.00 %
<input type="checkbox"/> Значение 3	0.000	00.00 %	

Порог значимости (%)

Рис.3. Указание порога значимости

Теперь необходимо перейти на следующий шаг и выбрать способ визуализации. Просмотрим результаты на диаграмме,

изображенной на рисунке 4.



а) Исходные входные факторы

б) Полученные входные факторы

Рис.4 Результаты факторного анализа

После обработки в наборе данных вместо трех исходных входных полей появились два новых поля - «Фактор1» и «Фактор2» - это результат понижения размерности (было 3 входных фактора, стало 2). На диаграмме видно, что «Фактор2» - близок к полю «ФАКТОР3», следовательно, «Фактор1» - это преобразованные факторы «ФАКТОР1» и «ФАКТОР2».

Далее выполним обработку данных из файла «TestForCPP.txt» при помощи корреляционного анализа. Он содержит таблицу со следующими полями: «АРГУМЕНТ» - аргумент, «ФАКТОР1», «ФАКТОР2», «ФАКТОР3» - входные значения, «РЕЗУЛЬТАТ1», «РЕЗУЛЬТАТ2» - выходные значения (рис.5).

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

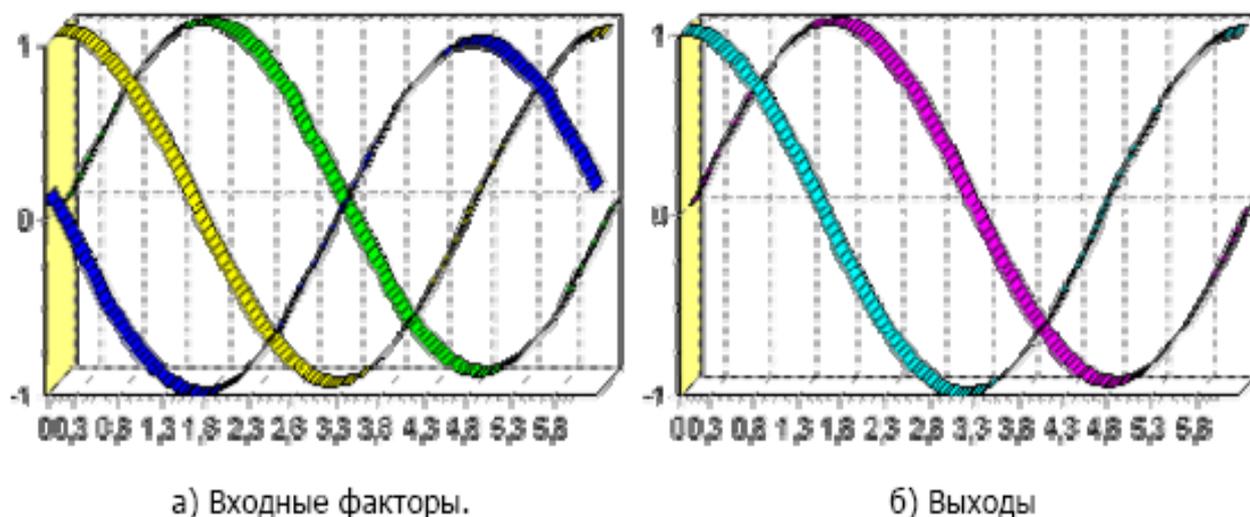


Рис.5. Данные для обработки

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

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

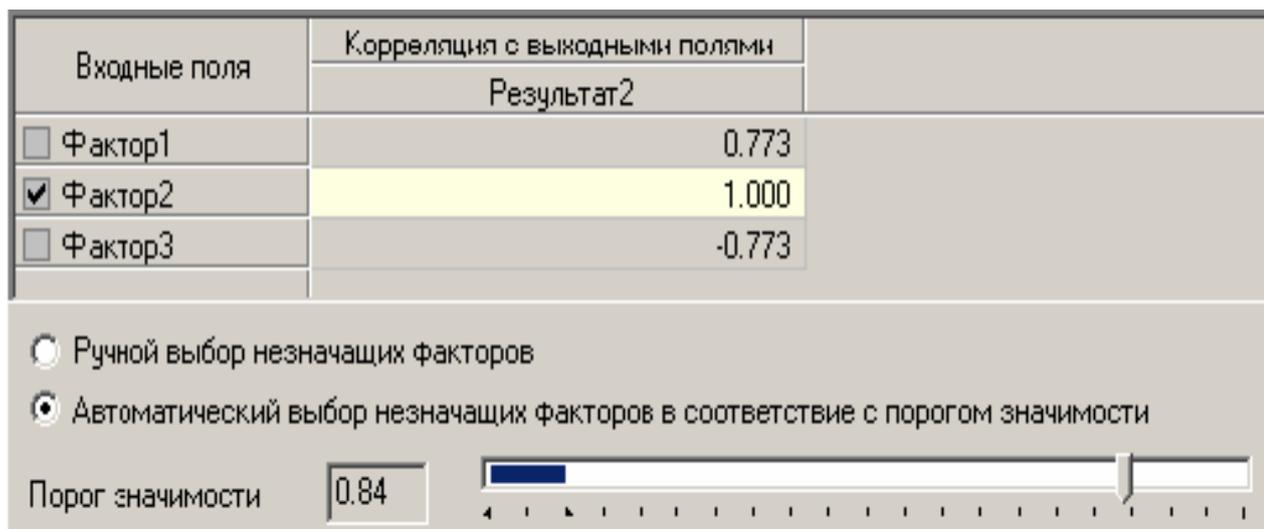


Рис.6. Указание порога значимости

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

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

4. Содержание отчета

Отчет по лабораторной работе представляется в виде документа Word. В состав документа входят:

1. Название работы
2. Цель работы
3. Копии экрана, иллюстрирующие выполнение лабораторной работы
4. Выводы по работе

5. Контрольные вопросы

1. Что такое факторный анализ?
2. Для чего используется факторный анализ, при обработке и анализе данных?
3. Что такое корреляционный анализ?
4. Для чего используется корреляционный анализ, при обработке и анализе данных?
5. Что является критерием принятия решения об исключении фактора при корреляционном анализе?
6. Какие методы статистического анализа Вы еще знаете?
7. Что такое матрица ковариации?

6. Список рекомендуемой литературы

1. Барсегян А.А., Куприянов М.С., Степаненко В.В., Холод И.И. Методы и модели анализа данных: OLAP и Data Mining. - Спб.: БХВ-Петербург, 2004. - 336 с.: ил
2. Загоруйко Н.Г. Прикладные методы анализа данных и знаний. - Новосибирск: Изд-во Ин-та математики, 1999. - 270 с.
3. Тюрин Ю.Н., Макаров А.А. Статистический анализ данных на компьютере / Под ред. В. Э. Фигурнова - М.: ИНФРА-М, 1998. - 528 с., ил.

ТРАНСФОРМАЦИЯ ДАННЫХ

1. Цель и содержание работы

Цель работы - научиться применять разбиение данных, квантование и фильтрацию для трансформации данных.

Содержание работы:

1. Создать в блокноте файл «Credit.txt», содержащий данные кредитования. В файле должны быть такие столбцы, как «Сумма кредита», «Дата кредитования» (в формате ДД.ММ.ГГ), «Цель кредитования», «Возраст» (рис.1.).

Сумма кредита	Дата кредитования	Цель кредитования	Возраст
7000	01.01.03	Оплата услуг	49
14578	01.01.03	Покупка товара	30
34567	03.01.03	Ремонт недвижимости	23
23567	04.02.03	<u>Турпоездка</u>	22
7500	05.02.03	Оплата услуг	56
12345	05.02.03	Покупка недвижимости	78

Рис.1. Пример заполнения файла «Credit.txt»

2. Импортировать в систему Deductor Studio текстовый файл «Credit.txt».

2.1. Произвести разбиение данных по рискам кредитования физических лиц.

2.2. Получить данные по суммам взятых кредитов по неделям.

2.3. Разбить данные о возрасте кредиторов на 5 интервалов (до 30 лет, от 30 до 40, от 40 до 50, от 50 до 60, старше 60 лет). Причем представить данные в разрезе по неделям.

3. Создать текстовый файл «banks.txt» и импортировать в систему. Файл должен содержать статистику по банкам России за определенный период («Банк», «Филиал», «Город», «Прибыль») (рис.2.).

Банк	Филиал	Город	Прибыль
Внешторгбанк	32	Москва	355197
Газпромбанк	1786	Казань	0
АВТОБАНК	100	Москва	4678389
Банк ЗЕНИТ	24	Челябинск	0
"ОАО ""АЛЬФА-БАНК"	ALFM	Москва	356564

Рис.2. Пример заполнения файла «banks.txt»»

3.1. Выявить ряд городов, в которых прибыль банков самая большая.

Продолжительность работы - 6 часов.

2. Теоретические сведения

Трансформация данных - перенос и преобразование проводится на основе правил трансформации, сопоставляющих аналитику двух "учетных зон" и определяющих критерии передачи данных.

Разбиение данных на группы. Часто для проведения анализа или построения модели прогноза приходится разбивать данные на группы, исходя из определенных критериев. В первом случае такая необходимость возникает, если аналитик желает просмотреть, к примеру, информацию не по всей совокупности данных, а по определенным группам (например, какую сумму кредита берут на те или иные цели, либо кредиторы того или иного возраста). Во втором

случае (прогнозирование) аналитику необходимо учитывать тот факт, что определенные группы (в данном случае группы кредиторов) ведут себя по разному, и что модель прогноза, построенная на всех данных не будет учитывать нюансов, возникающих в этих группах. Т.е. лучше построить несколько моделей прогноза, например, в зависимости от суммовой группы кредита и строить прогноз на них, нежели построить одну модель прогноза. Исходя из этого и не только, в Deductor Studio предоставляется широкий набор инструментов, тем или иным способом позволяющие разбивать исходные данные на группы, группировать любым способом всевозможные показатели и т.п.

Разбиение даты (по неделям) Разбиение даты служит для анализа всевозможных показателей за определенный период (день, неделя, месяц, квартал, год). Суть разбиения заключается в том, что на основе столбца с информацией о дате формируется другой столбец, в котором указывается, к какому заданному интервалу времени принадлежит строка данных. Тип интервала задается аналитиком, исходя из того, что он хочет получить - данные за год, квартал, месяц, неделю, день или сразу по всем интервалам.

Квантование. Часто аналитику необходимо отнести непрерывные данные (например, количество продаж) к какому-либо конечному набору (например, всю совокупность данных о количестве продаж необходимо разбить на 5 интервалов - от 0 до 100, от 100 до 200 и т.д., и отнести каждую запись исходного набора к какому - то конкретному интервалу) для анализа или фильтрации исходя именно из этих интервалов. Для этого в Deductor Studio применяется инструмент квантования (или дискретизации). Квантование предназначено для преобразования непрерывных данных в дискретные. Преобразование может проходить как по интервалам (данные разбиваются на заданное количество интервалов одинаковой длины), так и по квантилям (данные разбиваются на интервалы разной длины так, чтобы в каждом интервале находилось одинаковое количество данных). В качестве

значений результирующего набора данных могут выступать номер интервала, нижняя или верхняя граница интервала, середина интервала, либо метка интервала (значения определяемые аналитиком).

Фильтрация данных. Почти всегда исходный набор данных, или набор данных после обработки аналитику необходимо отфильтровать. Фильтрация бывает необходима для разбиения данных на какие либо группы (например, товарные группы) для последующей обработки или анализа данных уже отдельно по каждой группе. Также некоторые данные могут не подходить, или наоборот, подходить для дальнейшего анализа в силу накладываемых условий (например, если на каком - либо этапе обработки данных были выявлены противоречивые записи, то их необходимо исключить из последующей обработки). Здесь тоже возникает необходимость фильтрации. Фильтрация позволяет из базового набора данных получить набор данных, удовлетворяющий определенным аналитиком условиям. В Deductor Studio механизм построения условий фильтрации прост для понимания. В окне мастера можно определить несколько элементарных условий фильтрации (<ПОЛЕ> <ОТНОШЕНИЕ> <ЗНАЧЕНИЕ>), последовательно связанных логическими операциями (И, ИЛИ).

Группировка данных. Сложно делать выводы на основе необработанной первичной информации. Аналитику для принятия решения почти всегда 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 агрегируя данные в полях-фактах для дальнейшего анализа.

3. Порядок выполнения

работы

Интересующие нас столбцы: «СУММА КРЕДИТА», «ДАТА КРЕДИТОВАНИЯ», «ЦЕЛЬ КРЕДИТОВАНИЯ» и «ВОЗРАСТ». После импорта данных из текстового файла наиболее информативно просмотреть данные можно с помощью визуализатора «Куб», выбрав в качестве измерений столбцы «ВОЗРАСТ» и «ЦЕЛЬ КРЕДИТОВАНИЯ», а в качестве факта - столбец «СУММА КРЕДИТА». Остальные столбцы установить как непригодные (рис.1).

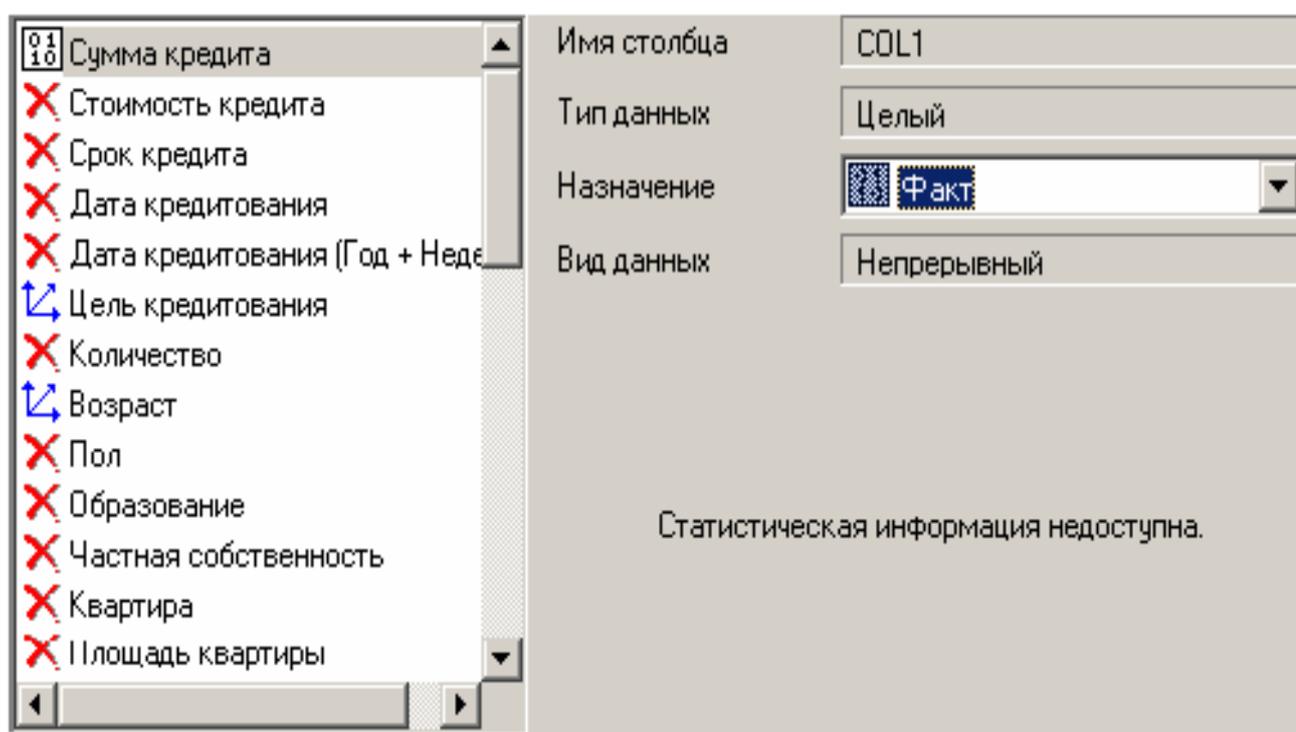


Рис.1. Настройка визуализатора «Куб»

На следующем шаге настройки куба следует указать измерение «ЦЕЛЬ КРЕДИТОВАНИЯ» как измерение в сроках, а измерение «ВОЗРАСТ» как измерение в столбцах, перетащив их с помощью мыши в соответствующие окна из области доступных измерений (рис.2).

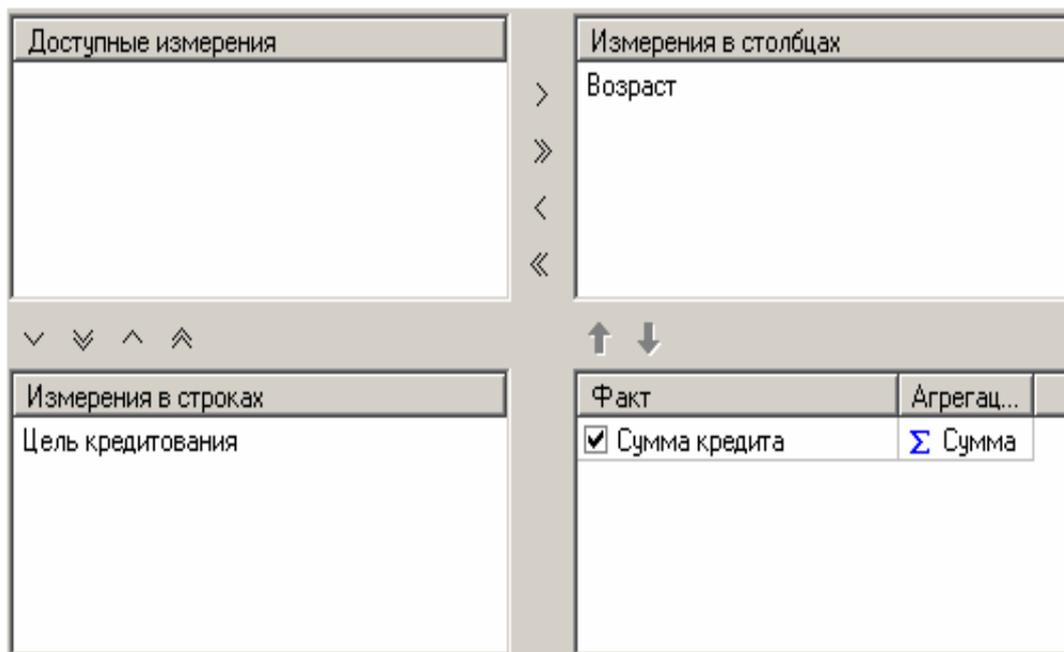


Рис.2. Настройка визуализатора «Куб»

В итоге, на кросс диаграмме (одна из закладок визуализатора куб) можно просмотреть исходные данные (рис.3).

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

Рис.3. Кросс-диаграмма

В мастере обработки «Дата и Время» на втором шаге выберем поле «ДАТА КРЕДИТОВАНИЯ» используемым, в появившейся после этого таблице настроек выберем назначение «Используемое» в столбце «Строка» напротив строки «Год + Неделя».

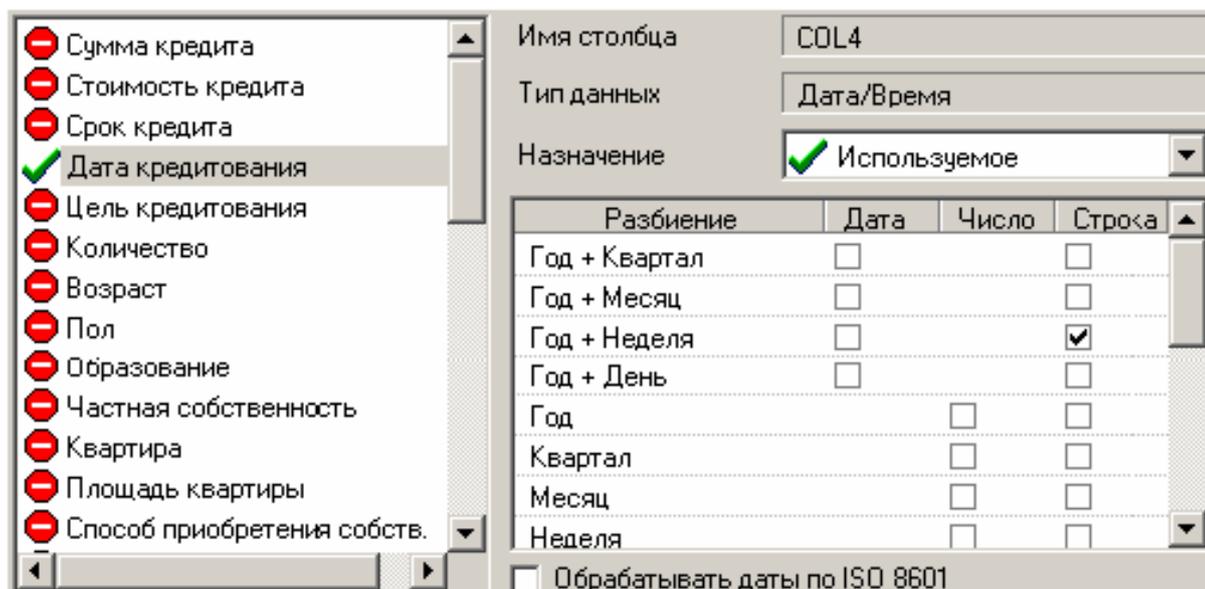


Рис.4 Окно мастера обработки «Дата и Время»

Больше никакие настройки не понадобятся, поэтому перейдем далее к выбору типа визуализации. Выберем в качестве визуализаторов «Таблицу» и «Куб», поставив галочки в соответствующих позициях. В мастере настройки полей куба выберем в качестве измерения появившийся после обработки столбец «ДАТА КРЕДИТОВАНИЯ_YWStr (Год + Неделя)» и столбец «ЦЕЛЬ КРЕДИТОВАНИЯ», а в качестве факта - «СУММА КРЕДИТА». Остальные поля сделаем неиспользуемыми. На следующем шаге перенесем одно измерение из области «доступных» в область «Измерения в строках», а другое - в область «Измерения в столбцах».

Таким образом, на кросс диаграмме имеем суммы взятых кредитов по неделям (за первые две недели года) в разрезе целей кредитования (рис.5).

	Дата кредитования (Год + Неделя) ▾		
Цель кредитования ▾	2003-w01	2003-w02	Итого
Иное	358 000,00	137 000,00	495 000,00
Оплата за образование	62 000,00	312 000,00	374 000,00
Оплата услуг (мед., юрид. и т.п.)	110 500,00	191 000,00	301 500,00
Покупка и ремонт недвижимости	404 000,00	538 000,00	942 000,00
Покупка товара	642 000,00	643 500,00	1 285 500,00
Турпоездки, развлечения и т.п.	35 500,00	113 500,00	149 000,00
Итого	1 612 000,00	1 935 000,00	3 547 000,00

Рис.5. Кросс диаграмма

В таблице с данными видно, что новое поле - «ДАТА КРЕДИТОВАНИЯ_YWStr (Год + Неделя)» содержит одинаковые значения (дата начала недели) для строк, которые попадают в одну и ту же неделю (дата начала недели или номер недели с начала года) (рис.6).

Срок кредита	Дата кредитов	Дата кред	Цель кредитования
24	05.01.03	2003-w01	Покупка товара
12	05.01.03	2003-w01	Покупка товара
30	05.01.03	2003-w01	Иное
36	06.01.03	2003-w02	Покупка и ремонт недвижимости
12	06.01.03	2003-w02	Оплата за образование
18	06.01.03	2003-w02	Иное
6	06.01.03	2003-w02	Покупка товара

Рис.6. Таблица данных

Далее используем инструмент квантования для разбиения данных о возрасте кредиторов на 5 интервалов (до 30 лет, от 30 до 40, от 40 до 50, от 50 до 60, старше 60 лет). Исходные данные распределятся по пяти интервалам именно так, поскольку, согласно статистике, минимальное значение возраста кредитора 19, а

максимальное 69 лет. Это необходимо аналитику для оценки кредиторской активности разных возрастных групп, с целью принятия решения о стимулировании кредиторов в группах с низкой активностью (например, уменьшение стоимости кредита для этих групп) и, быть может, увеличение прибыли в возрастных группах кредиторов с высоким риском (путем предложения дополнительных платных услуг). Причем аналитик желает видеть данные в разрезе по неделям. Воспользуемся мастером квантования (рис.7).

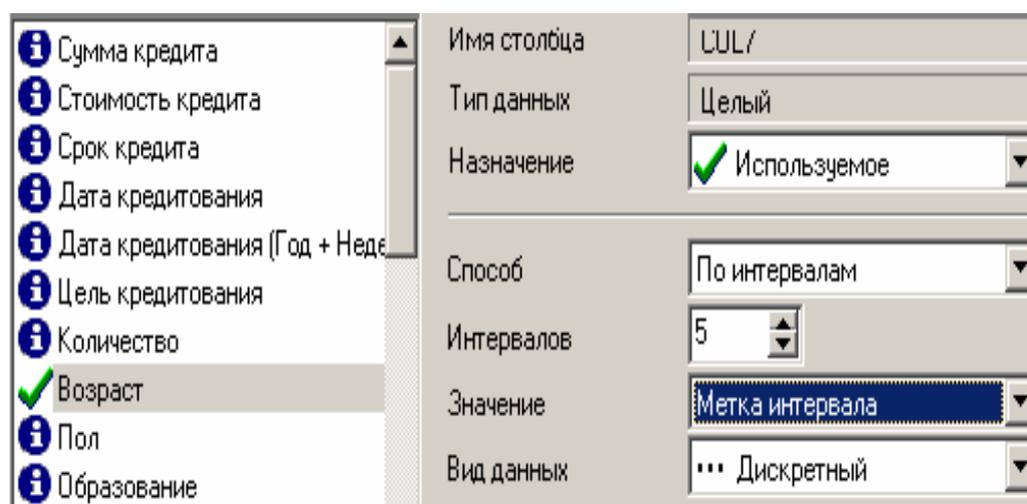


Рис.7. Мастер квантования

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

На следующем шаге мастера определим сами метки соответственно возраста кредиторов: «до 30 лет», «от 30 до 40 лет» и т.д. (рис.8)

Столбцы		Интервалы (изменены)		
Имя	Интервалов	NN	Граница	Метка
12 Возраст	5		0	
		0	29	До 30 лет
		1	39	От 30 до 40 лет
		2	49	От 40 до 50 лет
		3	59	От 50 до 60 лет
		4	1000	Старше 60 лет

Рис.8. Определение меток

После обработки выберем в качестве способа отображения «Куб». В мастере укажем «СУММА КРЕДИТА» в качестве факта, «ВОЗРАСТ» и поле «ДАТА КРЕДИТОВАНИЯ (Год + Неделя)» в качестве измерения, остальные поля укажем неиспользуемыми. Далее перенесем «ВОЗРАСТ» из доступных измерений в «Измерения в строках», а «ДАТА КРЕДИТОВАНИЯ (Год + Неделя)» в «Измерения в столбцах». На кросс диаграмме (рис.9) теперь видна информация о том, какие суммы кредитов берут кредиторы определенных возрастных групп в разрезе по неделям.

	Дата кредитования (Год + Неделя) ▼		
Возраст ▼	2003-W01	2003-W02	Итого
До 30 лет	798 500,00	808 500,00	1 607 000,00
От 30 до 40 лет	298 000,00	561 000,00	859 000,00
От 40 до 50 лет	195 000,00	295 500,00	490 500,00
От 50 до 60 лет	111 000,00	209 500,00	320 500,00
Старше 60 лет	209 500,00	60 500,00	270 000,00
Итого	1 612 000,00	1 935 000,00	3 547 000,00

Рис.9. Кросс диаграмма

Теперь аналитик, получив такие данные, может дать рекомендации о снижении стоимости кредита для лиц, старше 50 лет, либо о применении каких -нибудь других мер, способных привлечь большее количество кредиторов этих групп, либо мер,

направленных на то, чтобы кредитеры брали кредит на большие суммы.

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

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

Операция	Имя поля	Условие	Значение
	12 Сумма кредита	<	10000

Рис.10. Фильтр данных

Больше никаких условий не требуется, поэтому переходим на следующий шаг мастера и запускаем процесс фильтрации. После выполнения обработки можно манипулировать уже только с данными по кредитерам выбранного кредитного диапазона (рис.11). В правильности выполненной операции можно легко убедиться, выбрав в качестве визуализации данных статистику и просмотрев значения минимального и максимального значения поля «СУММА

КРЕДИТА».

Метка столбца	Тип данных		
		Минимум	Максимум
Сумма кредита	12 Целый	2000	9500
Стоимость кредита	12 Целый	400	1900
Срок кредита	12 Целый	6	6
Дата кредитования	7 Дата/Время	01.01.03	11.01.03
Дата кредитован...	ab Строковый		
Цель кредитования	ab Строковый		
Количество	12 Целый	1	1

Рис.11. Процесс фильтрации

Теперь допустим, что у аналитика имеется статистика по банкам России за определенный период. 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.

Для решения поставленной задачи первым делом необходимо найти суммарную прибыль всех банков в каждом городе. Для этого и необходима группировка. Для начала следует импортировать данные по банкам из текстового файла. Просмотреть исходную информацию можно в виде куба, где по строкам будут названия банков, а по столбцам - города. С помощью визуализатора «Куб» также можно получить требуемую информацию, выбрав в качестве измерения поле «ГОРОД», а в качестве факта «ПРИБЫЛЬ». Но нам необходимо получить эти данные для последующей обработки, следовательно, необходимо сделать аналогичную группировку.

Находясь в узле импорта, запустим мастер обработки. Выберем

в качестве обработки группировку данных. На втором шаге мастера установим назначение поля «ГОРОД» как измерение (рис.12), а назначение поля «ПРИБЫЛЬ» как факт. В качестве функции агрегации у поля «ПРИБЫЛЬ» следует указать Сумму.

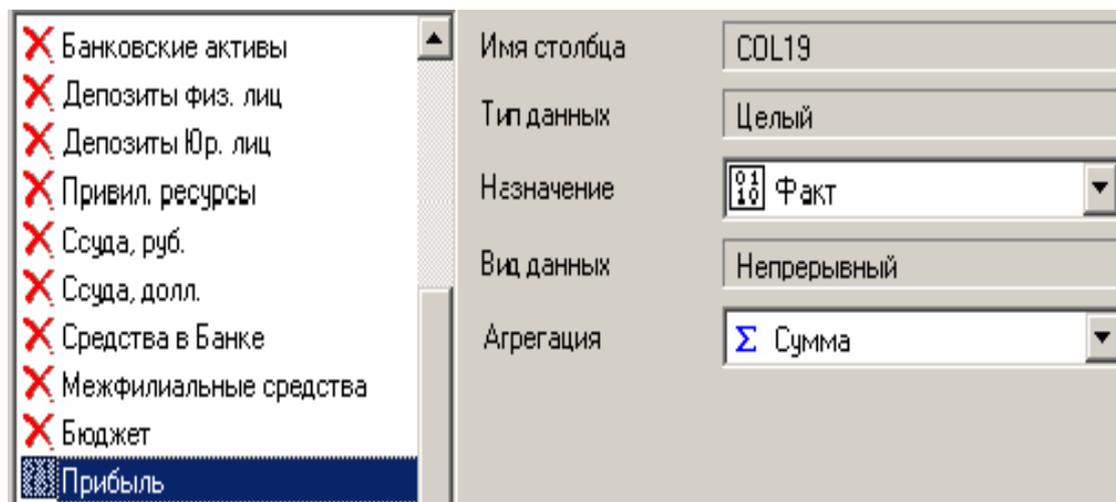


Рис.12. Окно мастера обработки

Таким образом, после обработки получим суммарные данные по прибыли всех банков по каждому городу. Их можно просмотреть, используя таблицу (рис.13).

	Город	Прибыль
	Москва	6076922
	Санкт-Петербург	233620
	Уфа	370468
	Санкт-Петербург	128038
	Ханты-Мансийск	30679
	Казань	68576
	Челябинск	63956

Рис.13. Таблица данных

4. Содержание отчета

Отчет по лабораторной работе представляется в виде документа Word. В состав документа входят:

1. Название работы
2. Цель работы
3. Копии экрана, иллюстрирующие выполнение задания лабораторной работы
4. Выводы по работе

5. Контрольные вопросы

1. Что такое разбиение данных на группы?
2. Для чего используется разбиение данных на группы в ходе анализа данных?
3. Для какого анализа используется разбиение по дате?
4. Что такое фильтрация данных?
5. Для чего используется фильтрация данных в анализе решений?
6. Как в системе Deductor Studio осуществляется фильтрация данных?
7. Что такое группировка данных?
8. Для чего используется группировка данных в анализе решений?
9. Как в системе Deductor Studio осуществляется группировка данных?
10. Для чего используется квантование данных в анализе решений?
11. Как в системе Deductor Studio осуществляется квантование данных?
12. Что такое кросс-диаграмма?

1. Загоруйко Н.Г. Прикладные методы анализа данных и знаний. - Новосибирск: Изд-во Ин-та математики, 1999. - 270 с.

ИСПОЛЬЗОВАНИЕ СТАНДАРТНЫХ МАТЕМАТИЧЕСКИХ ФУНКЦИЙ ПРИ АНАЛИЗЕ И ФОРМИРОВКЕ ДАННЫХ

1. Цель и содержание работы

Цель работы - научиться применять стандартные математические функции при анализе и формировке данных, заложенные в инструмент «Калькулятор» системы Deductor.

Содержание работы:

1. В блокноте создать файл «Calculate.txt», в котором содержатся столбцы «АРГУМЕНТ1», «АРГУМЕНТ2», «АРГУМЕНТ3» - набор аргументов в программу. Разделителем между столбцами является знак табуляции. В каждом столбце должно быть 100 значений. В столбце «Аргумент1» каждое значение повторяется десять раз. В столбце «Аргумент2» выбираются десять значений, которые повторяются. Столбец «Аргумент3» заполняется значениями от 1 до 100 (по возрастанию) (рис.1.).

2. На основе аргументов рассчитать математические функции:

- две функции одного аргумента (АРГУМЕНТ3)

$\sin(\text{Аргумент3} * \text{Аргумент3}) * \ln(\text{Аргумент3} + 1) - \exp(-\text{Аргумент3}/10),$

$10 * \sin(\text{Аргумент3} * \text{Аргумент3}/100) / (\text{Аргумент3} + 1) * \exp(-\text{Аргумент3}/10)$

- одну функцию от двух аргументов

$\text{Аргумент1} * \text{Аргумент1}/100 - \text{Аргумент2} * \text{Аргумент2}/100$

- функцию, показывающую относительное отклонение (Аргумент1+1 от Аргумент2+1).

Аргумент1	Аргумент2	Аргумент3
0	1	1
0	2	2
0	3	3
0	4	4
0	5	5
0	6	6
0	7	7
0	8	8
0	9	9
0	10	10
1	1	11
1	2	12
1	3	13
1	4	14
1	5	15
1	6	16

		55		
1			7	17
1		8		18
1		9		19
1		10		20

Рис.1. Пример заполнения файла «Calculate.txt»

Продолжительность выполнения работы - 4 часа.

2. Теоретические сведения

Иногда возникает необходимость на каком-либо этапе обработки данных получить на их основе новые (производные) данные. Возможно, аналитику требуется вычислить процентное отклонение значения одного поля относительно другого, либо подсчитать сумму, разность полей, получить на основе данных показатель и уже его использовать для дальнейшей обработки, в зависимости от значения полей вычислить те или иные выражения.

В Deductor Studio такую возможность предоставляет инструмент «Калькулятор». Он позволяет создавать новые поля, вычисляющие заданные аналитиком выражения. Т.е. калькулятор служит для получения производных данных на основе имеющихся в исходном наборе. Мастер предоставляет широкий набор функций различного направления. В мастере представлен список новых выражений, где добавляются необходимые аналитику выражения, список доступных функций с кратким описанием каждой, список доступных операций и также список доступных столбцов, которые можно задействовать при создании выражения.

Список всех встроенных функций вместе с описанием можно посмотреть в мастере нажав на кнопку Функция.

Реализованный в Deductor Studio конструктор выражений при

построении использует не метки (Сумма, Количество, Цена ...), а имена полей таблицы, заданные в источнике данных (Summ, Count, Price...). При импорте в некоторых случаях (напр. Из текстового файла) можно задать как метки, так и имена импортируемых полей.

Слева в окне конструктора находится список вычисляемых выражений. Изначально оно содержит одно пустое выражение. Для управления списком вычисляемых выражений предусмотрены следующие кнопки:

 - переместить текущее выражение на одну позицию вверх по списку;

 - переместить текущее выражение на одну позицию вниз по списку;

 - добавляет новое выражение с параметрами, устанавливаемыми по умолчанию, и пустой формулой. Затем вызывает диалог редактирования параметров выражения;

 - добавляет новое выражение с типом данных, описанием и формулой как у текущего выражения. Затем вызывает диалог редактирования параметров выражения;

 - вызывает диалог редактирования параметров выражения;

 - удаляет текущее выражение.

Для изменения свойств выражения используется конструктор выражений. В нем задаются следующие параметры:

- Имя - строка, которая будет служить идентификатором столбца в процедурах обработки. При желании пользователь может

ввести любые имена, которые точнее отражают назначение столбца;

- Метка - название, под которым данный столбец будет виден в таблице, кросс-таблице или на диаграмме после обработки. Желательно, чтобы оно отражало содержание столбца;

- Тип данных - тип данных вычисляемого выражения. Тип выбирается из списка, открываемого щелчком по кнопке в правой части поля.

Изначально при открытии страницы конструктора список выражений содержит только один элемент «Выражение». Для него следует установить нужные параметры и при необходимости добавить новые строки. По умолчанию для нового выражения назначается метка «Выражение_ N», где N - номер, обеспечивающий уникальность. Имена полей, формируемых в результате вычислений по данному выражению, назначаются автоматически и имеют вид: «EXPR_N», где N - уникальный номер.

После настройки параметров выражения в поле «Выражение» требуется ввести рассчитываемую формулу. Правила составления выражений соответствуют общепринятым, в частности, число открывающих скобок должно равняться числу закрывающих. Выражение может содержать:

- числа в явном виде;
- переменные в виде имен столбцов;
- скобки, определяющие порядок выполнения операций;
- знаки математических операций и отношений;
- имена функций;
- даты в формате «ДД.ММ.ГГ», обязательно указываемые в кавычках. Такой способ ввода даты, хотя и допускается, но может оказаться непереносимым между разными компьютерами. По этой причине лучше использовать функцию STRTODATE();

- строковые выражения в кавычках - «строковое выражение»;

- однострочные и многострочные комментарии. Однострочный комментарий начинается символами «//» (два слеша) и продолжается до конца строки. Многострочным комментарием считаются все символы, содержащиеся между скобками «/*» (слеш-звездочка) и «*/» (звездочка-слеш).

Выражение можно ввести вручную с клавиатуры, однако удобнее выбирать функции, переменные и знаки операций с помощью мыши. В поле «Выражение» всегда отображается то выражение, которое в данный момент выделено в списке. Для его редактирования достаточно щелкнуть в поле мышью, вызвав курсор, а затем редактировать как обычное текстовое поле.

Для добавления в выражение функций следует нажать кнопку «Функция» на панели «Операции». При этом будет открыто окно выбора функции. Чтобы ввести функцию нужно в поле «Список функций» открыть нужный вид функций, щелкнув по значку «+» справа от его наименования. В результате будет развернут список функций данного вида. Если выделить функцию в списке, то справа появится краткое описание функции.

Чтобы ввести функцию в выражение достаточно дважды щелкнуть по ее имени в списке. Имя функции в выражении появляется вместе со скобками, куда следует ввести аргумент или аргументы. Аргументами могут быть числа в явном виде, строки в кавычках, даты в кавычках, имена функций, имена полей, а также арифметические, логические и строковые выражения. Имена полей удобно вводить с помощью двойного щелчка в списке полей. Если в аргументе несколько полей, то их имена разделяются точкой с запятой. Знаки математических операций и отношений можно выбирать щелчком мыши в секции «Операции».

3. Порядок выполнения работы

Для просмотра исходных данных удобнее использовать

визуализатор «Таблица»:

	Аргумент1	Аргумент2	Аргумент3
	0	4	4
	0	5	5
	0	6	6
	0	7	7
	0	8	8
	0	9	9

Рис.2. Просмотр данных

Рассчитаем значение функций:

1) $\text{SIN}(\text{АРГУМЕНТ } 3 * \text{АРГУМЕНТ } 3) * \text{LN}(\text{АРГУМЕНТ } 3 + 1) * \text{EXP}(-\text{АРГУМЕНТ } 3 / 10)$

2) $10 * \text{SIN}(\text{АРГУМЕНТ } 3 * \text{АРГУМЕНТ } 3 / 100) / (\text{АРГУМЕНТ } 3 + 1) * \text{EXP}(-\text{АРГУМЕНТ } 3 / 10)$.

Для этого, находясь на узле импорта, запустим мастер обработки. Выберем в качестве обработчика калькулятор. На втором шаге мастера в списке выражений в первой строке в графе «Название выражения» вместо надписи «Выражение» напишем F1(АРГУМЕНТ3) (рис.3).

В поле редактора выражения (в верхней части мастера) напишем « $\text{SIN}(\text{COL3} * \text{COL3}) * \text{LN}(\text{COL3} + 1) * \text{EXP}(-\text{COL3} / 10)$ ».

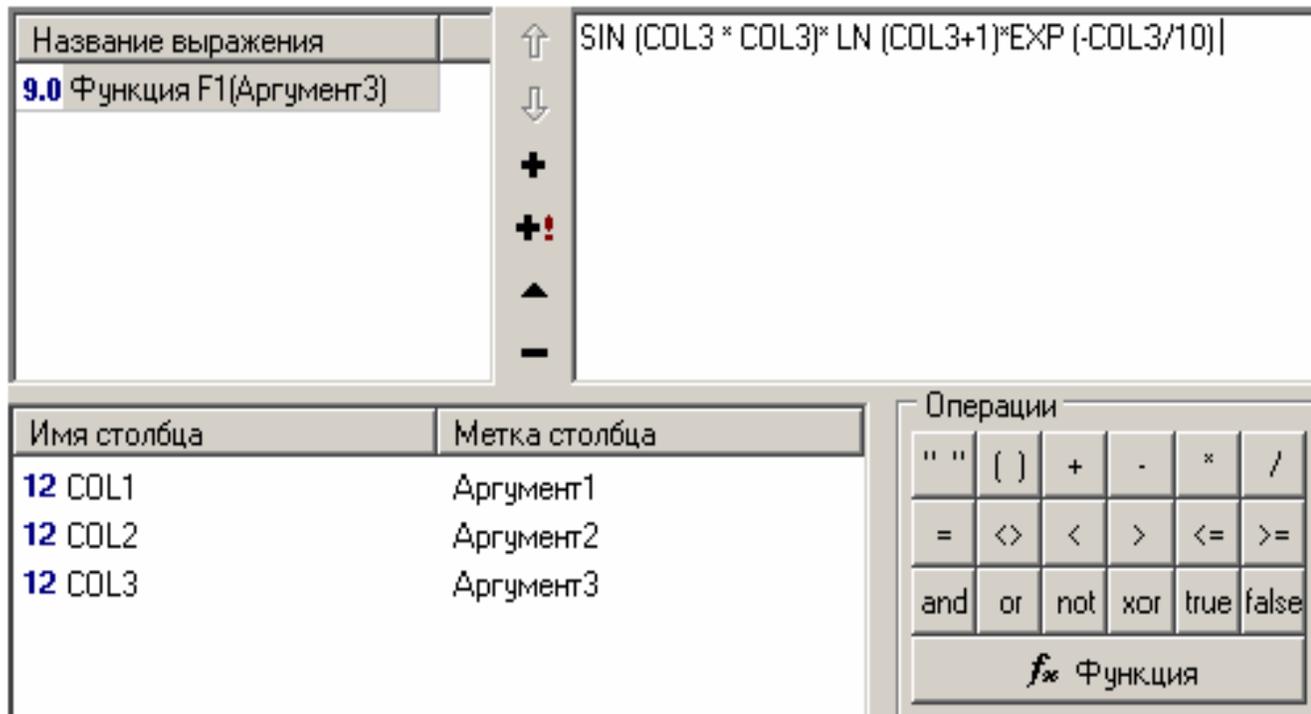


Рис.3. Мастер обработки

Таким образом, мы создали новый столбец, задали ему название «F1(АРГУМЕНТ3)» и также определили, какие значения будут принимать записи этого поля. На этом создание вычисляемого значения окончено, поэтому переходим на следующий шаг мастера, где предлагается выбрать способ отображения данных. Самым информативным в данном случае является диаграмма, которую и следует выбрать.

Далее, выбрав в мастере настроек диаграммы в качестве отображаемого поля «F1(АРГУМЕНТ3)», в качестве типа графика «Линии», в качестве подписей по оси X значения поля «АРГУМЕНТ3» можно увидеть график вычисленной функции, представленный на рисунке 4.

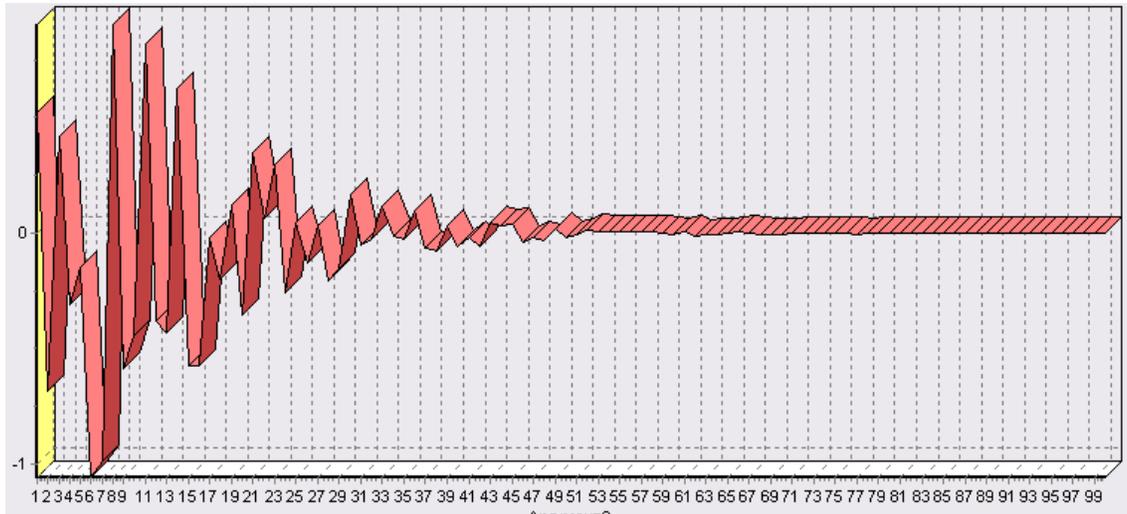


Рис.4. График вычисленной функции

Сложная функция $F2(\text{АРГУМЕНТ3})$ отличается только видом функции (« $10 * \sin(\text{COL3} * \text{COL3}/100) / (\text{COL3} + 1) * \exp(-\text{COL3}/10)$ »)

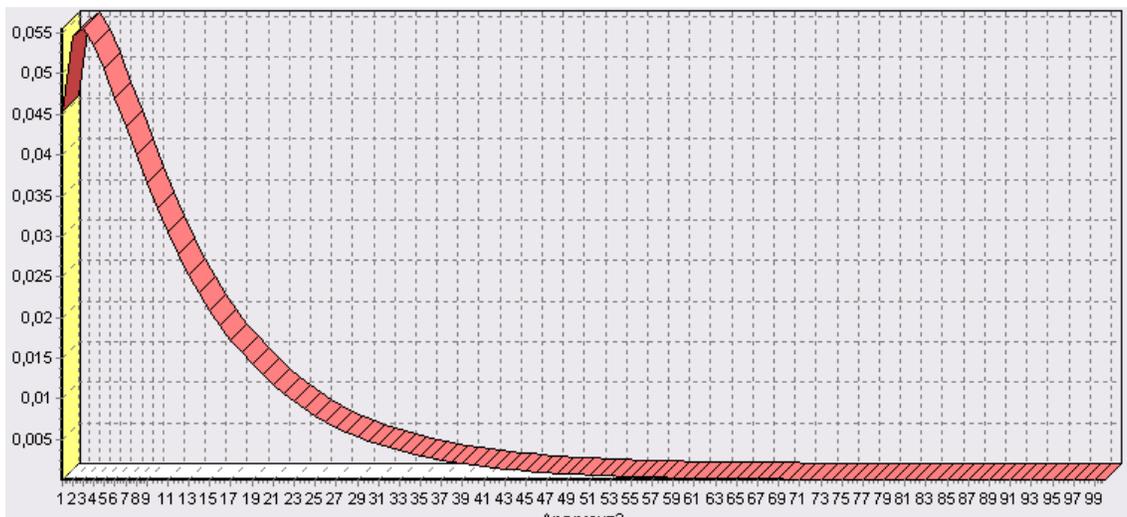


Рис.5. График сложной функции

Функция от двух аргументов $F3(\text{АРГУМЕНТ1}; \text{АРГУМЕНТ2})$

Данная функция интересна тем, что для ее просмотра в трех измерениях можно использовать визуализатор «Куб». Зададим название выражения « $F3(\text{АРГУМЕНТ1}; \text{АРГУМЕНТ2})$ », в поле

вычисляемого выражения напишем «COL1*COL1/100 - COL2*COL2/100». Выберем визуализатор «Куб» и настроим его так, что «АРГУМЕНТ1 и «АРГУМЕНТ2» являлись бы измерениями, F3 (АРГУМЕНТ1; АРГУМЕНТ2) - фактом, а «АРГУМЕНТ3» - неиспользуемым.

Выбрав «АРГУМЕНТ1» измерением в столбцах, а «АРГУМЕНТ2» - измерениям в строках перейдем к просмотру Кросс-диаграммы (рис.6). Для более наглядного просмотра установим тип диаграммы «области». Теперь можно посмотреть вычисленную функцию в объемном виде.

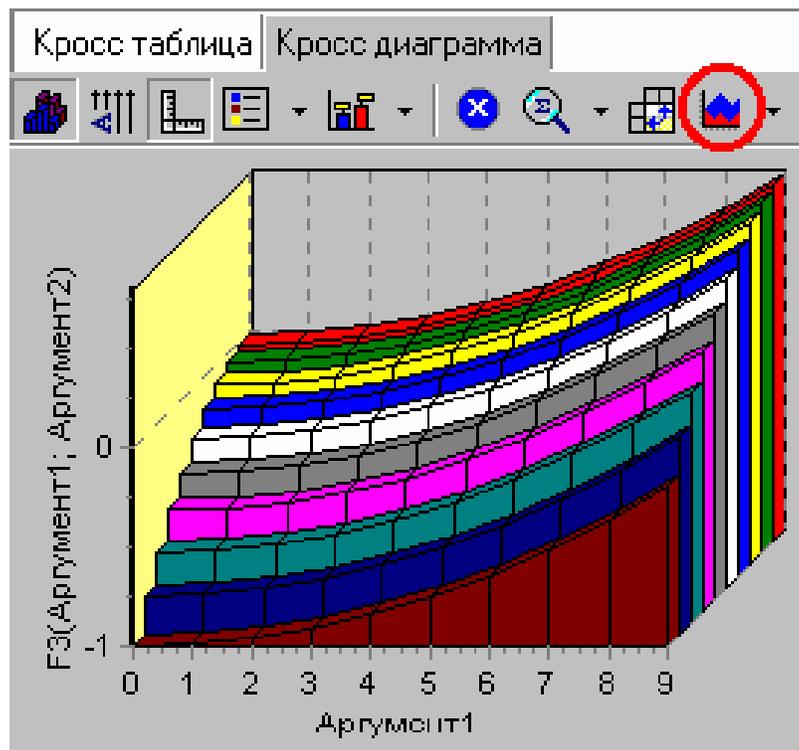


Рис.6. Кросс диаграмма

Теперь вычислим доленое отклонение АРГУМЕНТ1+1 от АРГУМЕНТ2+1 (RELDEV). Задав в качестве вычисляемого выражения RELDEV(COL1 + 1; COL2 + 1) можно на диаграмме увидеть данное отклонение (рис.7).

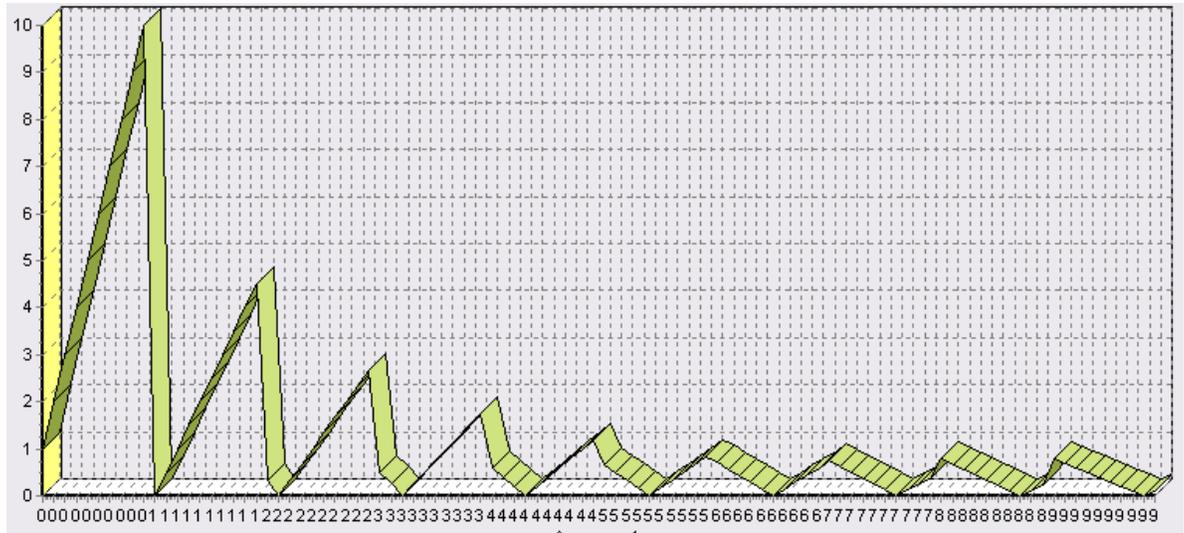


Рис.7. Диаграмма отклонения

Пусть функция принимает значения $\text{SQRT}(\text{АРГУМЕНТ3}/50)$ (квадратный корень) при значениях АРГУМЕНТ3 от 0 до 50 и значения $\text{АРГУМЕНТ3} * \text{АРГУМЕНТ} 3/2500$ при остальных. Для вычисления подобной функции необходимо воспользоваться имеющейся в наличии функцией $\text{IFF}(\text{арумент1}; \text{аргумент2}; \text{аргумент3})$, которая позволяет в зависимости от логического значения первого аргумента получить второй или третий аргумент. Если значение аргумента больше нуля и меньше 50 необходимо получить выражение $\text{SQRT}(\text{АРГУМЕНТ3}/50)$, в противном случае - выражение $\text{АРГУМЕНТ3} * \text{АРГУМЕНТ3}/2500$.

Таким образом, в поле построения выражения необходимо написать « $\text{IFF}((\text{COL3} > 0) \text{ AND } (\text{COL3} < 50)); \text{SQRT}(\text{COL3}/50); \text{COL3} * \text{COL3}/2500)$ ». Сделав это в мастере обработки «Калькулятор», и выбрав далее визуализатор «Диаграмма», и также выбрав в мастере настройки диаграммы поле со значениями кусочно-заданной функции, можно посмотреть на требуемый результат (рис.8).

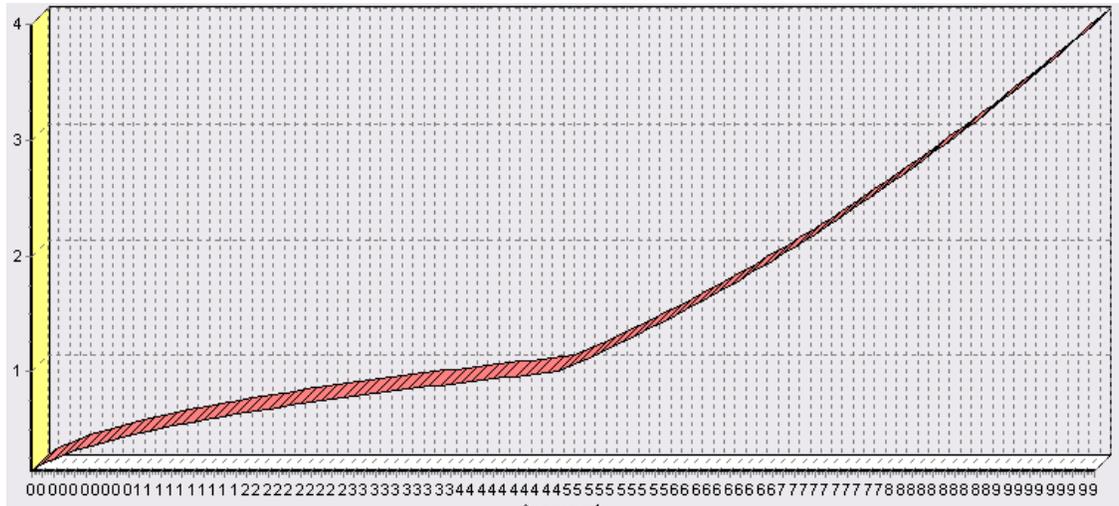


Рис.8. Диаграмма кусочно-заданной функции

4. Содержание отчета

Отчет по лабораторной работе представляется в виде документа Word. В состав документа входят:

1. Название работы
2. Цель работы
3. Копии экрана, иллюстрирующие выполнение задания лабораторной работы
4. Выводы по работе

5. Контрольные вопросы

1. Для чего необходимо формирование новых данных с использованием математических функций
2. Какой инструмент имеется в системе Deductor Studio для создания данных с помощью математических функций?

3. Приведите примеры необходимости использования математических функций для проведения анализа данных.
4. В чем особенности инструмента «Калькулятор»?

6. Список рекомендуемой литературы

1. Загоруйко Н.Г. Прикладные методы анализа данных и знаний. - Новосибирск: Изд-во Ин-та математики, 1999. - 270 с.
2. Барсегян А.А., Куприянов М.С., Степаненко В.В., Холод И.И. Методы и модели анализа данных: OLAP и Data Mining. - Спб.: БХВ Петербург, 2004. - 336 с.: ил.

ПОИСК АССОЦИАТИВНЫХ ПРАВИЛ ДЛЯ УСТАНОВЛЕНИЯ ЗАВИСИМОСТЕЙ МЕЖДУ СОБЫТИЯМ

1. Цель и содержание работы

Цель работы - научиться применять ассоциативные правила и использовать визуализаторы «Популярные наборы», «Правила», «Дерево правил», «Что-если».

Содержание работы:

В блокноте создать таблицу, в которой представлена информация по покупкам продуктов нескольких групп.

Сохранить файл с именем «Supermarket.txt». В созданном файле должны быть два столбца «Номер чека» «Товар», в каждом из которых должно быть 147 значений.

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

Рис.1. Пример заполнения файла «Supermarket.txt»

Необходимо решить задачу анализа потребительской корзины с целью последующего применения результатов для стимулирования продаж.

Продолжительность выполнения работы - 4 часа.

2. Теоретические сведения

Ассоциативные правила позволяют находить закономерности между связанными событиями. Примером такого правила, служит утверждение, что покупатель, приобретающий «Хлеб», приобретет и «Молоко». Впервые эта задача была предложена для поиска ассоциативных правил для нахождения типичных шаблонов покупок, совершаемых в супермаркетах, поэтому иногда ее еще называют анализом рыночной корзины (market basket analysis). Пусть имеется база данных, состоящая из покупательских транзакций. Каждая транзакция - это набор товаров, купленных покупателем за один визит. Такую транзакцию еще называют рыночной корзиной.

Целью анализа является установление следующих зависимостей: если в транзакции встретился некоторый набор элементов X , то на основании этого можно сделать вывод о том, что другой набор элементов Y также же должен появиться в этой транзакции. Установление таких зависимостей дает нам возможность находить очень простые и интуитивно понятные правила.

Основными характеристиками таких правил являются поддержка и достоверность.

Правило «Из X следует Y » имеет поддержку s , если $s\%$

транзакций из всего набора, содержат наборы элементов X и Y . Достоверность правила показывает, какова вероятность того, что из X следует Y .

Правило «Из X следует Y » справедливо с достоверностью s , если $s\%$ транзакций из всего множества, содержащих набор элементов X , также содержат набор элементов Y .

Покажем на конкретном примере: пусть 75% транзакций, содержащих хлеб, также содержат молоко, а 3% от общего числа всех транзакций содержат оба товара. 75% - это достоверность правила, а 3% - это поддержка.

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

Границы значений параметров поддержки и достоверности выбираются таким образом, чтобы ограничить количество найденных правил. Если поддержка имеет большое значение, то алгоритмы будут находить правила, хорошо известные аналитикам или настолько очевидные, что нет никакого смысла проводить такой анализ. С другой стороны, низкое значение поддержки ведет к генерации огромного количества правил, что, конечно, требует существенных вычислительных ресурсов. Тем не менее, большинство интересных правил находится именно при низком значении порога поддержки. Хотя слишком низкое значение поддержки ведет к генерации статистически необоснованных правил.

Таким образом, необходимо найти компромисс, обеспечивающий, во-первых, интересность правил и, во-вторых, их статистическую обоснованность. Поэтому значения этих границ напрямую зависят от характера анализируемых данных и подбираются

индивидуально. Еще одним параметром, ограничивающим количество найденных правил является максимальная мощность часто встречающихся множеств. Если этот параметр указан, то при поиске правил будут рассматриваться только множества, количество элементов которых будет не больше данного параметра. И, следовательно, любое найденное правило будет состоять не больше, чем из максимальной мощности элементов.

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

Визуализатор «*Правила*» отображает ассоциативные правила в виде списка правил.

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

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

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

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

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

Результаты анализа можно применить и для сегментации покупателей по поведению при покупках, и для анализа предпочтений клиентов, и для планирования расположения товаров в супермаркетах, кросс-маркетинге. Предлагаемый набор визуализаторов позволяет эксперту найти интересные, необычные закономерности, понять, почему так происходит и применить их на практике.

3. Порядок выполнения работы

Импортируем данные из файла «Supermarket.txt».

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

Рис.2. Таблица данных

Для поиска ассоциативных правил запустим мастер обработки (рис.3). В нем выберем тип обработки «Ассоциативные правила». На втором шаге мастера необходимо указать, какой столбец является идентификатором транзакции (чек), а какой элементом транзакции (товар).

The screenshot shows a wizard interface for data processing. On the left, a table is displayed with two columns: 'ID Номер чека' and 'Товар'. The 'Товар' column is selected. On the right, the configuration for this column is shown:

- Имя столбца: COL2
- Тип данных: Строковый
- Назначение: Элемент
- Вид данных: Дискретный
- Уникальные значения: Кол-во уникальных значений: 7
- Список уникальных значений: ВАФЛИ, КЕТЧУПЫ, СОУСЫ, АДЖИКА, МАКАРОННЫЕ ИЗДЕЛИЯ, МЕД, СУХАРИ, СЫРЫ, ЧАЙ

Рис3. Мастер обработки

Следующий шаг позволяет настроить параметры построения ассоциативных правил: минимальную и максимальную поддержку, минимальную и максимальную достоверность, а также максимальную мощность множества (рис.4). Исходя из характера имеющихся данных, следует указать границы поддержки - 13% и 80%, и достоверности 60% и 90%.

Часто встречающиеся множества	
Минимальная поддержка, %	13.00
Максимальная поддержка, %	80.00
<input type="checkbox"/> Максимальная мощность искомым часто встречающихся множеств	4

Ассоциативные правила	
Минимальная достоверность, %	60.00
Максимальная достоверность, %	90.00

Рис.4. Окно настройки параметров

Следующий шаг позволяет запустить процесс поиска ассоциативных правил. На экране отображается информация о количестве множеств, количестве найденных правил, а также гистограмма распределения найденных часто встречающихся множеств по мощности (рис.5).

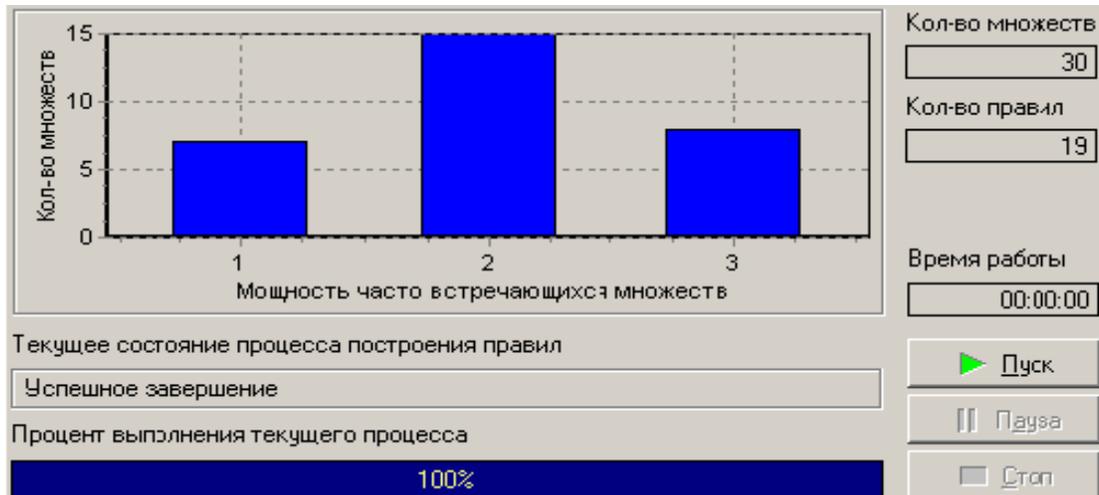


Рис.5. Результат процесса поиска

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

N	Множество	↑ Поддержка	
		%	Кол-во
7	ЧАЙ	75,00	33
3	МАКАРОННЫЕ ИЗДЕЛИЯ	54,55	24
2	КЕТЧУПЫ, СОУСЫ, АДЖИКА	52,27	23
4	МЕД	50,00	22
11	КЕТЧУПЫ, СОУСЫ, АДЖИКА И МАКАРОННЫЕ ИЗДЕЛИЯ	45,45	20
6	СЫРЫ	43,18	19
20	МЕД И ЧАЙ	40,91	18

Рис.6. Визуализатор «Популярные наборы»

При использовании визуализатора «Популярные наборы», получившиеся наборы товаров наиболее часто покупают в данной торговой точке, следовательно можно принимать решения о поставках товаров, их размещении и т.д. Визуализатор «Правила» выводит список ассоциативных правил, представленный таблицей со

столбцами: «номер правила», «условие», «следствие», «поддержка, %», «поддержка, количество», «достоверность».

N	Условие	Следствие	Поддержка		Достоверность, %
			%	Кол- во	
1	ВАФЛИ	СУХАРИ	22,73	10	71,43
2	СУХАРИ	ВАФЛИ	22,73	10	71,43
3	КЕТЧУПЫ, СОУСЫ, АДЖИКА	МАКАРОННЫЕ ИЗДЕЛИЯ	45,45	20	06,96
4	МАКАРОННЫЕ ИЗДЕЛИЯ	КЕТЧУПЫ, СОУСЫ, АДЖИКА	45,45	20	83,33
5	МЕД	ЧАЙ	40,91	18	81,82
6	СЫРЫ	ЧАЙ	29,55	13	68,42
7	ВАФЛИ И СУХАРИ	ЧАЙ	20,45	9	90,00
8	ВАФЛИ И ЧАЙ	СУХАРИ	20,45	9	69,23

Рис.7. Визуализатор «Правила»

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

Используя визуализатор «Дерево правил»получим:

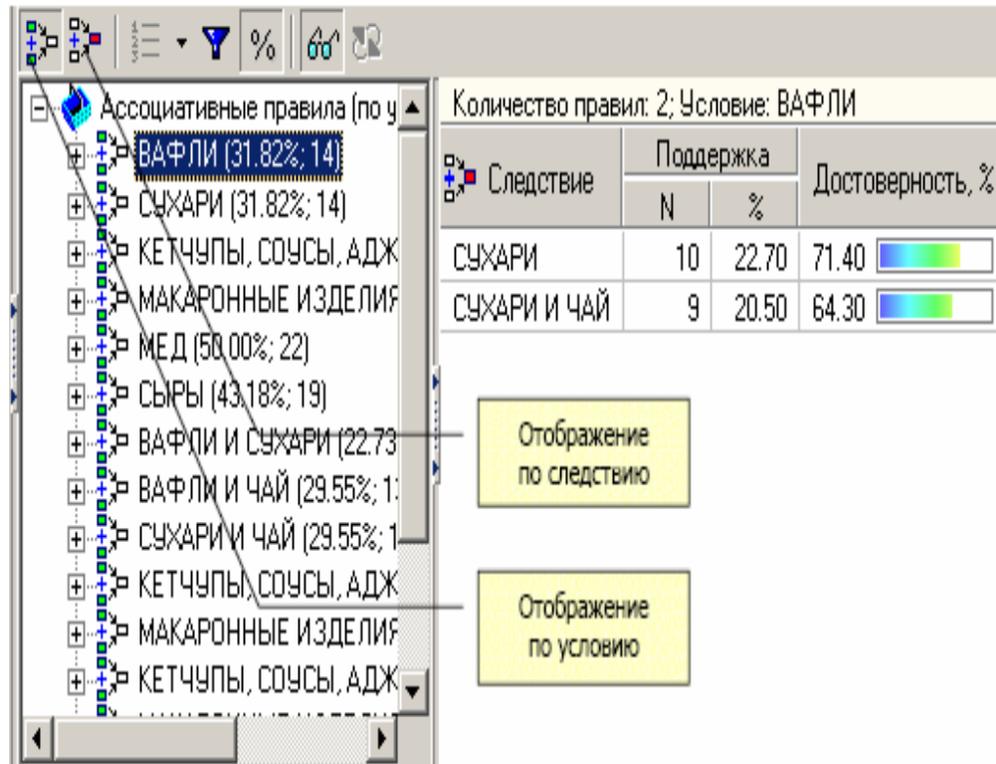


Рис.8. Визуализатор «Дерево правил»

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

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

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

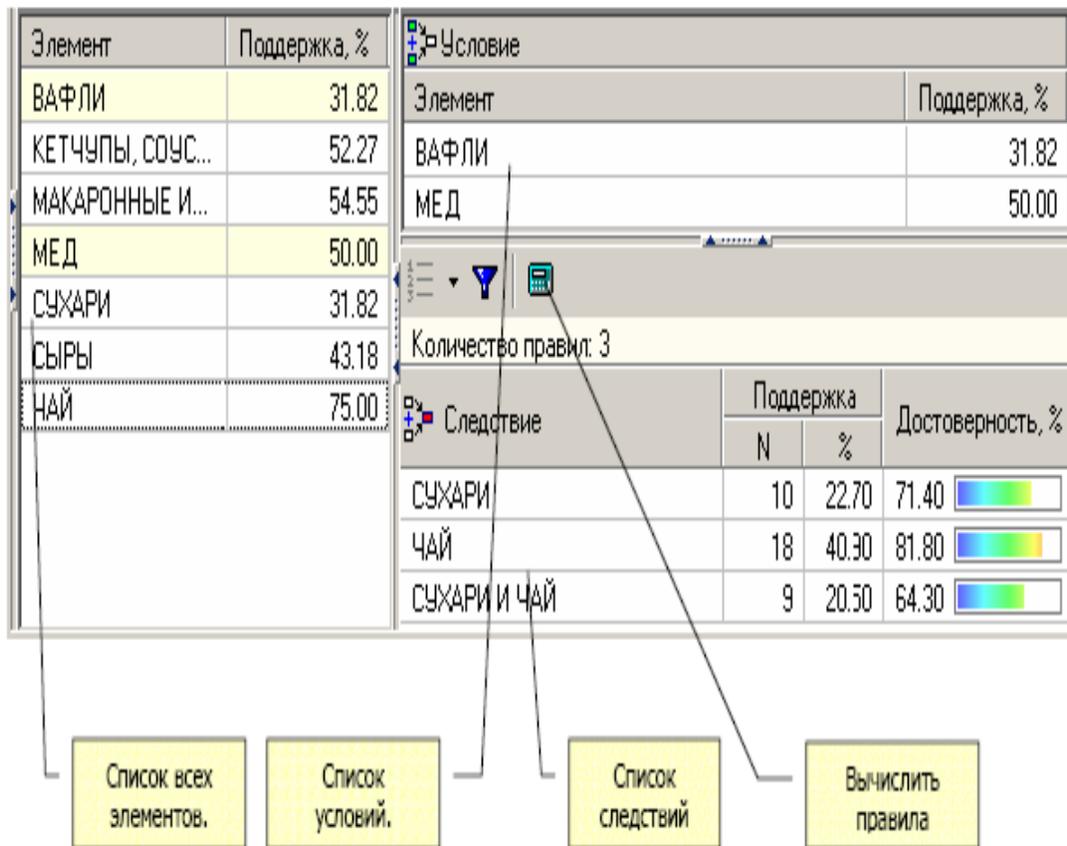


Рис.9. Визуализатор «Что - если»

4. Содержание отчета

Отчет по лабораторной работе представляется в виде документа Word. В состав документа входят:

1. Название работы
2. Цель работы
3. Копии экрана, иллюстрирующие выполнение лабораторной работы
4. Выводы по работе

5. Контрольные вопросы

1. Что такое ассоциативные правила?
2. Как создаются ассоциативные правила?
3. Для чего используются ассоциативные правила при анализе данных?
4. Что такое достоверность правила?
5. Что такое поддержка правила?
6. Какие инструменты для построения ассоциативных правил имеются в системе Deductor?
7. Что такое дерево правил?
8. Какие варианты создания дерева правил существуют в Deductore?
9. Приведите пример полученных результатов анализа данных с помощью ассоциативных правил.

6. Список рекомендуемой литературы

1. Барсегян А.А., Куприянов М.С., Степаненко В.В., Холод И.И. Методы и модели анализа
данных: OLAP и Data Mining. - Спб.: БХВ-Петербург,
2004. - 336 с.: ил.
2. Дюк В., Самойленко А. Data mining: учебный курс. - СПб:
Питер, 2001. - 368 с.: ил.

ПРОГНОЗИРОВАНИЕ ВРЕМЕННЫХ РЯДОВ**1. Цель и содержание работы**

Цель работы - научиться применять методы Data Mining для решения задач прогнозирования временных рядов на примере построения модели прогноза продаж

Содержание работы:

В блокноте создать файл «Trade.txt» с данными (рис.1), содержащими историю продаж за некоторый период. Файл должен содержать два столбца «Дата (Год+Месяц)» (формата ГГГГ-МММ) и «Количество» (десятичное число).

Определить есть ли сезонность, если есть, то какая.

Какое количество товара будет продано через неделю и через две.

Дата (Год + Месяц) Количество

2017-M01 462523.419

2017-M02 633208.196

2017-M03 660159.299

2017-M04 617455.3417

2017-M05 597354.4794

2017-M06 793517.4512

2017-M07 1015944.2862

2017-	M08	1148052.2523
2017-M09		1156623.1715
2017-M10		1255021.9423
2017-M11		1410114.5606
2017-M12		1357230.3388

Рис.1. Пример заполнения файла «Trade.txt»

Продолжительность выполнения работы - 4 часа.

2. Теоретические сведения

Важным фактором для анализа временного ряда и прогноза является определение сезонности. В Deductor Studio таким инструментом является автокорреляция.

Целью автокорреляционного анализа является выяснение степени статистической зависимости между различными значениями (отсчетами) случайной последовательности, которую образует поле выборки данных. Если их корреляция равна единице, то величины прямо зависимы друг от друга, если нулю - то нет, если минус единица, то зависимость обратная. В процессе автокорреляционного анализа рассчитываются коэффициенты корреляции (мера взаимной зависимости) для двух значений выборки, отстоящих друг от друга на определенное количество отсчетов, называемые также лагом.

Применительно к анализу временных рядов автокорреляция позволяет выделить месячную и годовую сезонность в данных. Видно, что пик зависимости на данных приходится на 12 месяц, что свидетельствует о годовой сезонности. Поэтому величину продаж годовой давности необходимо обязательно учитывать при построении модели (если используется нейронная сеть - то подавать на вход).

Линейная автокорреляция ищет зависимости между значениями

одной и той же величины, но в разное время. Поэтому нахождение линейной автокорреляционной зависимости и применяется для определения периодичности (сезонности) при обработке временных рядов.

Прогноз 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).

Обработчик «Нейросеть». Обработчик предназначен для решения задач регрессии и прогнозирования. В данном случае нейросеть строится для прогнозирования будущих значений временного ряда. Для проверки обобщающей способности нейросети рекомендуется разбить имеющееся множество данных на две части: обучающее и тестовое. Как правило, при прогнозировании временных рядов, доля тестового множества составляет не более 10-20%.

С помощью визуализатора «Диаграмма» оценивается способность построенной нейросетевой модели к обобщению. Для этого в одном окне выводятся графики исходного и спрогнозированного временных рядов.

3. Порядок выполнения работы

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 (рис.2).

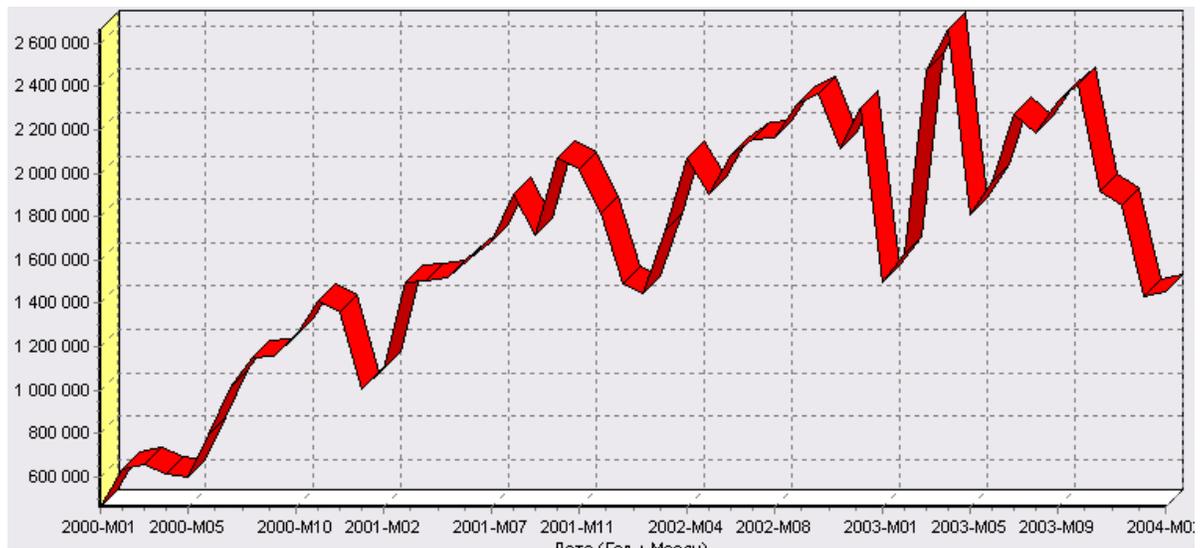


Рис.2. Диаграмма данных

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 (рис.3). 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 искажаться зависимость от месяца назад, двух, ..., пятнадцати месяцев назад). Также должен стоять флажок «Включить поле отсчетов набор данных». Он необходим для более удобной интерпретации автокорреляционного анализа.

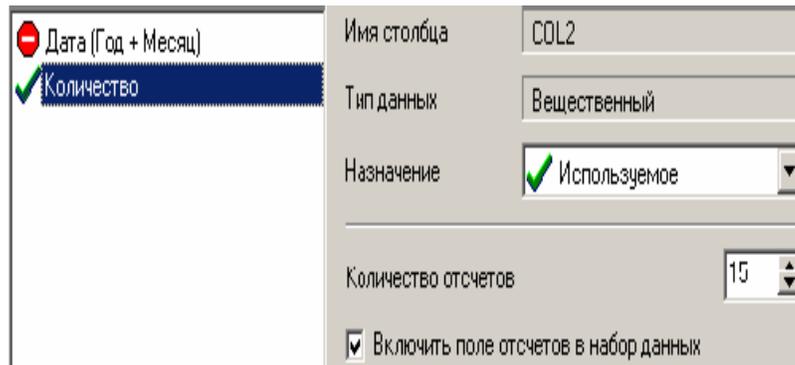


Рис.3. Мастер обработки

Перейдем на следующий шаг мастера и запустим процесс обработки.

По окончании, результаты удобно анализировать как в виде таблицы, так и в виде диаграммы (рис.4). После обработки были получены два столбца - «Лаг» (благодаря установленному флажку в мастере) и «КОЛИЧЕСТВО» - результат автокорреляции.

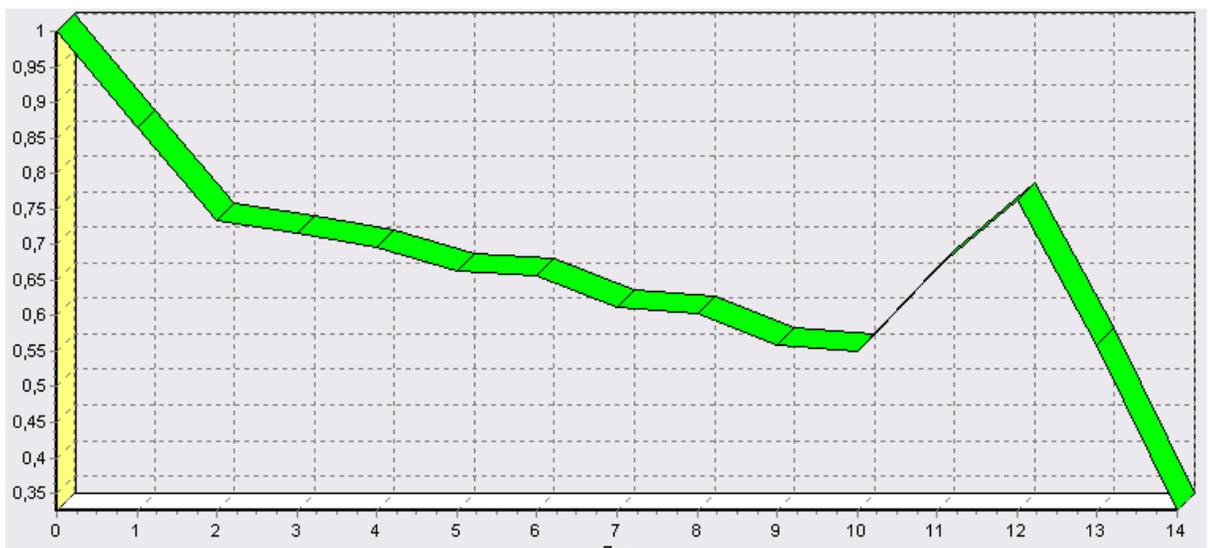


Рис.4. Итоговая диаграмма

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.

После импорта данных воспользуемся диаграммой для их просмотра. На ней видно, что данные содержат аномалии (выбросы) и шумы, за которыми трудно разглядеть тенденцию.

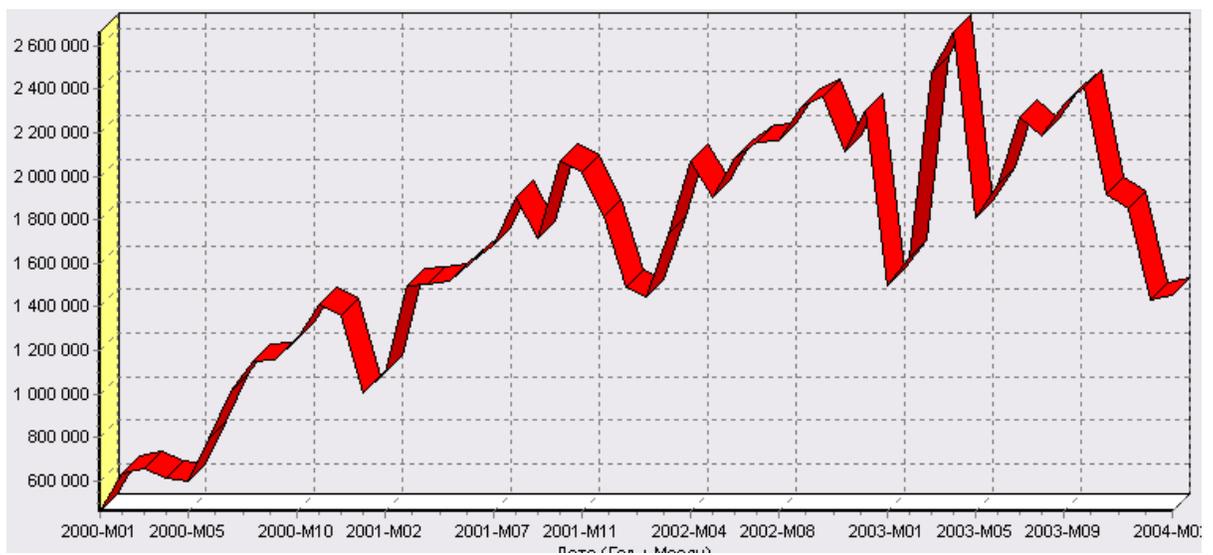


Рис.5 Диаграмма данных

Поэтому перед прогнозированием необходимо удалить аномалии и сгладить данные. Сделать это можно при помощи парциальной обработки.

Запустим мастер обработки, выберем в качестве обработки данных парциальную обработку и перейдем на следующий шаг мастера. Как известно, второй шаг мастера отвечает за обработку пропущенных значений, которых в исходных данных нет. Поэтому здесь ничего не настраиваем. Следующий шаг отвечает за удаление аномалий из исходного набора.

Выберем поле для обработки «КОЛИЧЕСТВО» и укажем для

него обработку аномальных явлений (степень подавления - малая).

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

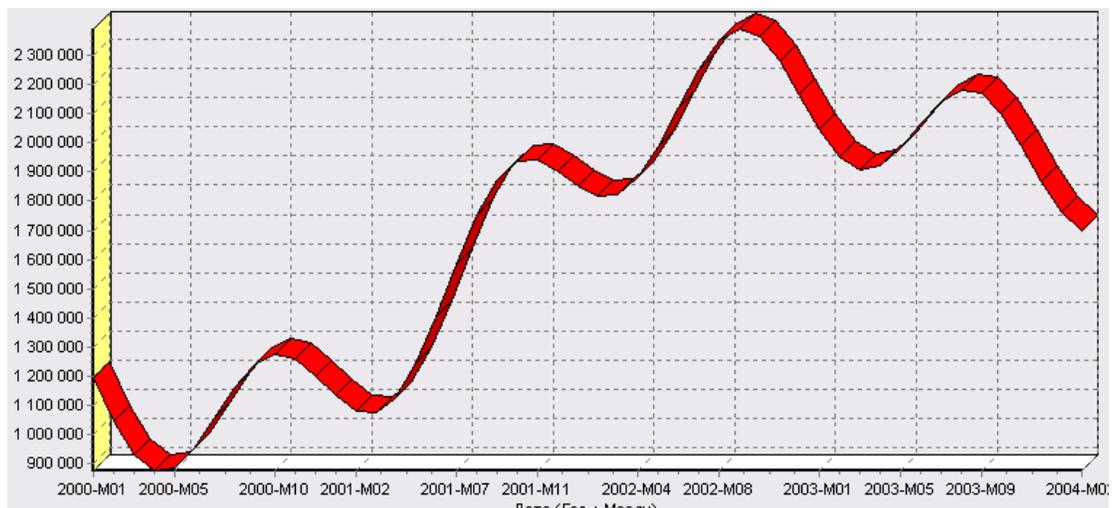


Рис.6. Диаграмма результата

Видно, что данные сгладились, аномалии и шумы исчезли. Также видна тенденция. Теперь перед аналитиком встает вопрос, а как, собственно, прогнозировать временной ряд. В данном случае столбец один. Строить прогноз на будущее необходимо, основываясь на данных прошлых периодов. Т.е. предполагается, что количество продаж на следующий месяц зависит от количества продаж за предыдущие месяцы. Т.е. входными факторами для модели могут быть продажи за текущий месяц, продажи за месяц ранее и т.д., а результатом должны быть продажи за следующий месяц. Т.е. здесь явно необходимо трансформировать данные к скользящему окну.

Запустим мастер обработки (рис.7), выберем в качестве

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

Рис.7. Окно мастера обработки

Просмотреть полученные данные можно в виде таблицы:

Дата (Год	Количество-12	Количество-11	Количество-10	Количество-9
2001-M01	1195750.32836624	1046730.3444785	932230.825412825	875457.294625339
2001-M02	1046730.3444785	932230.825412825	875457.294625339	884830.92710038
2001-M03	932230.825412825	875457.294625339	884830.92710038	951789.091701106
2001-M04	875457.294625339	884830.92710038	951789.091701106	1053383.007105

Рис.8 Таблица полученных данных

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».

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» (рис.9). Остальные поля сделаем информационными.

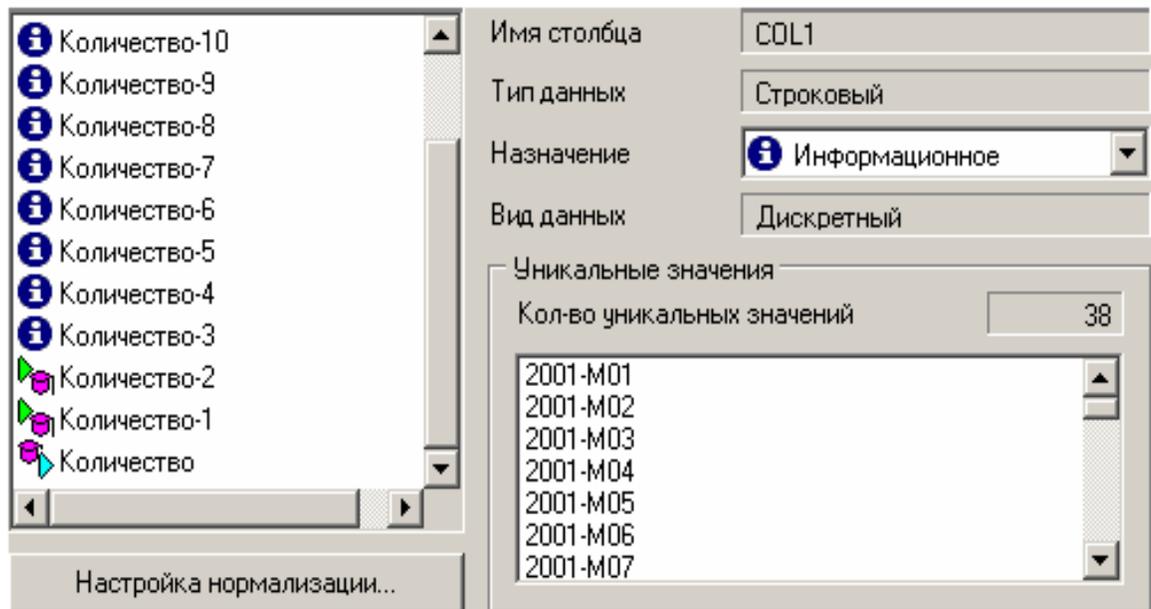


Рис.9. Окно мастера обработки

Оставив все остальные параметры построения модели по умолчанию, обучим нейросеть. После построения модели для просмотра качества обучения представим полученные данные в виде диаграммы и диаграммы рассеяния. В мастере настройки диаграммы (рис.10) выберем для отображения поля «КОЛИЧЕСТВО» и «КОЛИЧЕСТВО_OUT» - реальное и спрогнозированное значение.

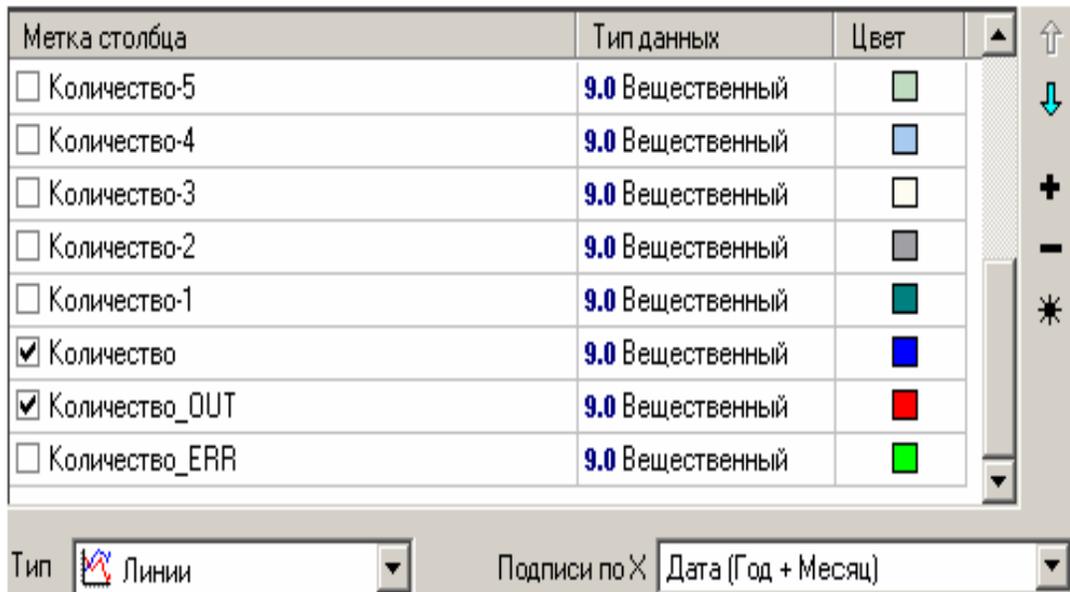


Рис.10. Окно мастера настройки диаграммы

Результатом будет два графика:

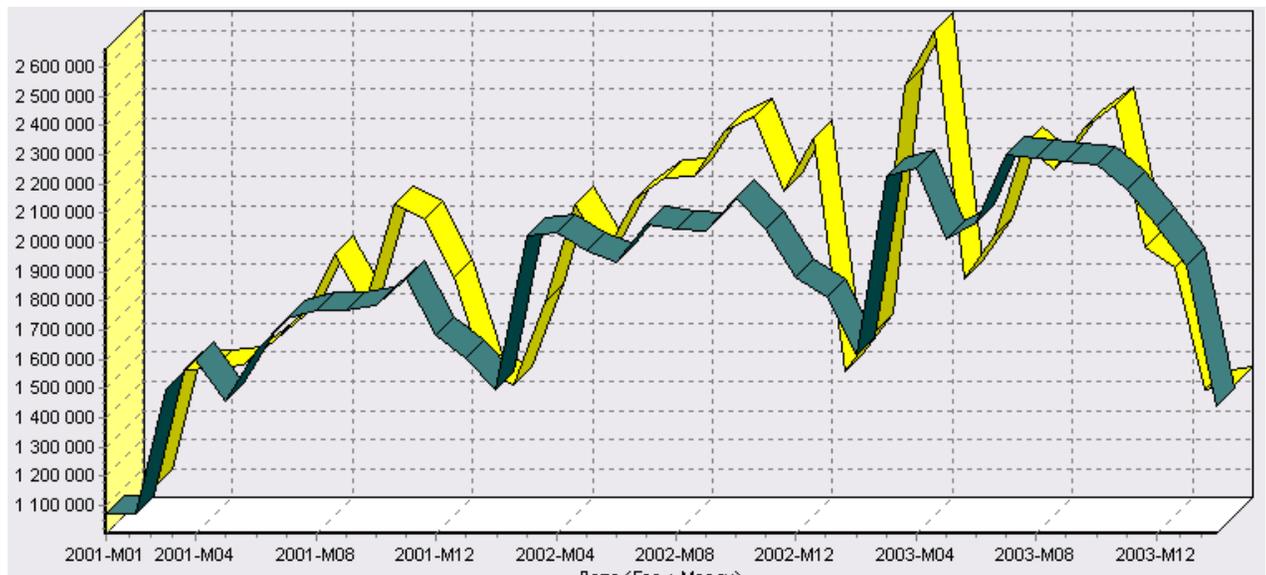


Рис.11. Диаграмма качества обучения

Диаграмма рассеяния более наглядно показывает качество обучения:

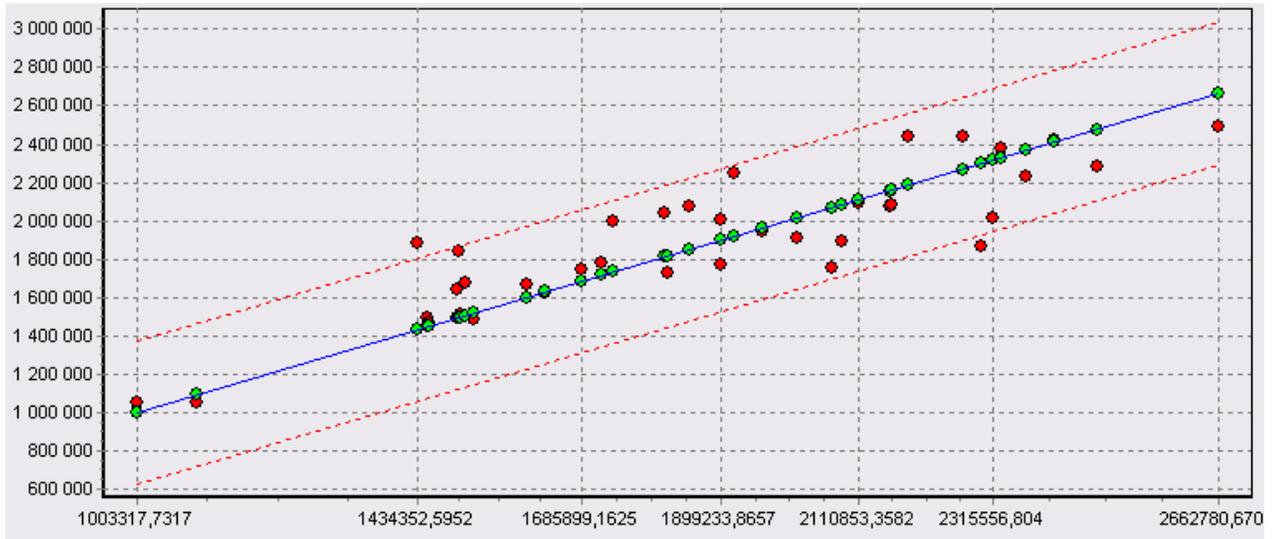


Рис.12. Диаграмма рассеяния

Нейросеть обучена, теперь осталось самое главное - построить требуемый прогноз. Для этого открываем мастер обработки и выбираем появившийся теперь обработчик «Прогнозирование» (рис13). На втором шаге мастера предлагается настроить связи столбцов для прогнозирования временного ряда - откуда брать данные для столбца при очередном шаге прогноза. Мастер сам верно настроил все переходы, поэтому остается только указать горизонт прогноза (на сколько вперед будем прогнозировать) равным трем, а также, для наглядности, необходимо добавить к прогнозу исходные данные, установив в мастере соответствующий флажок.

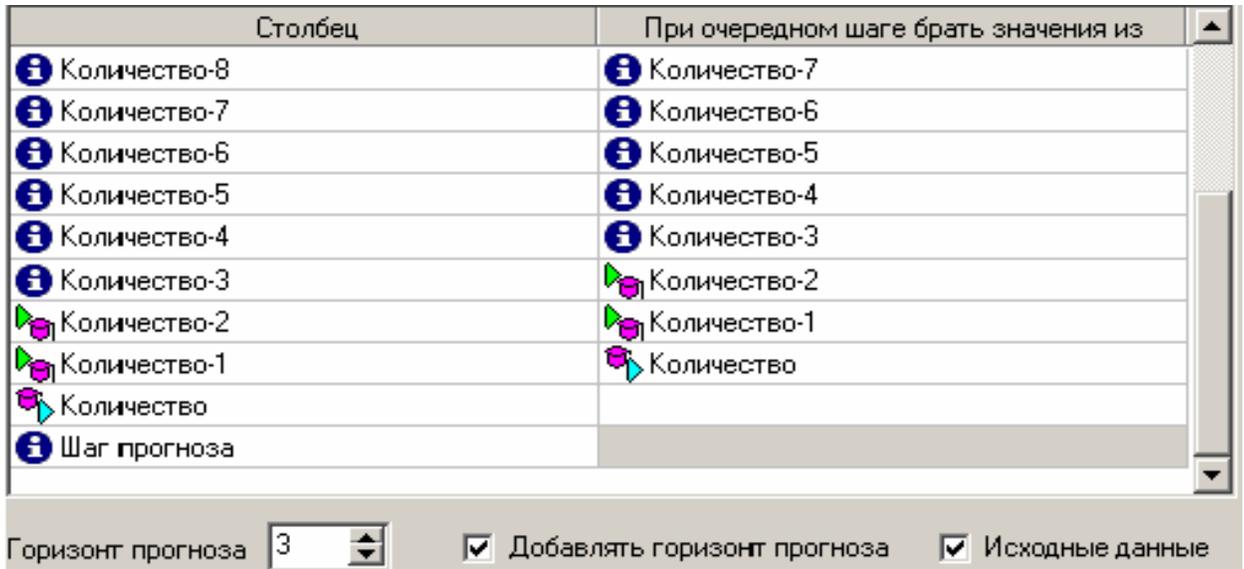


Рис.13. Окно мастера обработки

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

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

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

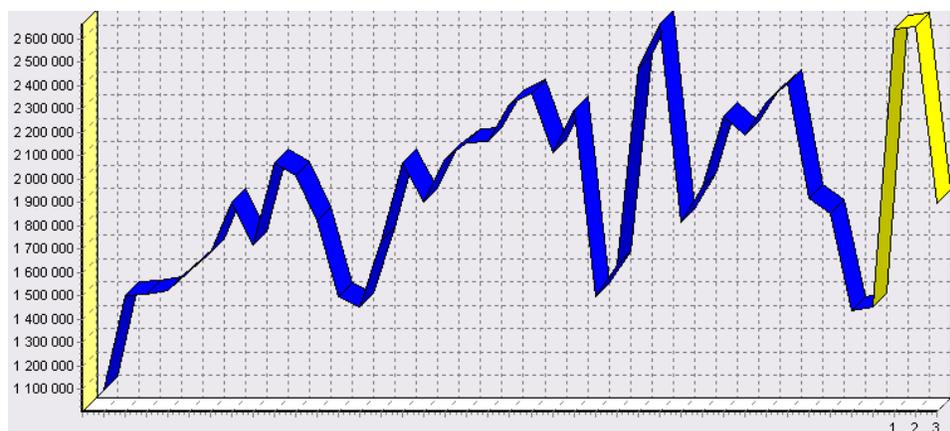


Рис.14. Диаграмма прогноза

4. Содержание отчета

Отчет по лабораторной работе представляется в виде документа Word. В состав документа входят:

1. Название работы
2. Цель работы
3. Копии экрана, иллюстрирующие выполнения задания лабораторной работы
4. Выводы по работе

5. Контрольные вопросы

1. Что такое сезонность?
2. Для чего используется автокорреляционный анализ?
3. Как определить существует зависимость между данными или нет?
4. Для чего нужен прогноз временного ряда?
5. Какой инструмент в системе Deduktor используется для прогнозирования временных рядов?
6. Какого назначения обработчика «Нейросеть» системы Deduktor?
7. Как обработчик «Нейросеть» можно использовать при прогнозировании?

6. Список рекомендуемой литературы

1. Ханк Д.Э., Уичерн Д.У., Райтс А.Дж. Бизнес-прогнозирование, 7-е издание.: Пер. с англ. - М.: Издательский дом «Вильямс», 2003. - 656 с.: ил. - Парал. тит. англ.
2. Загоруйко Н.Г. Прикладные методы анализа данных и знаний.-Новосибирск: Изд-во Ин-та математики, 1999. - 270с.

ПРИМЕНЕНИЕ СКРИПТОВ ДЛЯ АВТОМАТИЗАЦИИ ПРОЦЕССА ДОБАВЛЕНИЯ В СЦЕНАРИЙ ВЕТВЕЙ ОБРАБОТКИ

1. Цель и содержание работы

Цель работы - научиться применять обработчик «Скрипт» для решения задач прогнозирования на примере прогноза продаж

Содержание работы:

В блокноте создать файл «TradeSakes.txt», содержащий информацию по продажам некоторой группы товаров. Файл должен содержать два столбца «Дата (Год + Месяц)» (формата ГГГГ-МММ) и «Количество» (целое число) (рис.1.).

Выполнить описанные действия в пункте три и сделать прогноз продаж на три месяца вперед.

Дата (Год + Месяц)	Количество
2017-M01	355000
2017-M02	340000
2017-M03	405000
2017-M04	452000
2017-M05	464000
2017-M06	437000
2017-M07	697000
2017-M08	670000

	2017-M09	645000
2017-M10	923000	
2017-M11	915000	
2017-M12	1479000	
2018-M01	1040000	
2018-M02	978000	
2018-M03	739000	
2018-M04	900000	

Рис.1. Пример Заполнения файла «TradeSakes.txt»

Продолжительность выполнения работы - 4 часа.

2. Теоретические сведения

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

Предположим, что после импорта данных из двух разных баз данных требуется провести предобработку (очистить данные, сгладить, поменять названия столбцов, добавить несколько одинаковых выражений) и построить одинаковые модели прогноза, а затем экспортировать полученные данные обратно. Для первой ветви (первой БД) эти действия проводятся как обычно - последовательными

шагами строится цепочка обработчиков. Для второго же источника (второй БД) достаточно создать узел импорта, к которому присоединить скрипт, основанный на уже построенной первой ветке. В этом скрипте будут выполнены точно такие же действия, как в оригинальной ветви. На выходе скрипта ставится узел экспорта, и вторая ветвь обработки готова к использованию.

3. Порядок выполнения работы

Импортируем данные из файла «TradeSakes.txt». После импорта данных запустим мастер обработки и выберем в качестве обработчика «Скрипт». 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» (рис2).

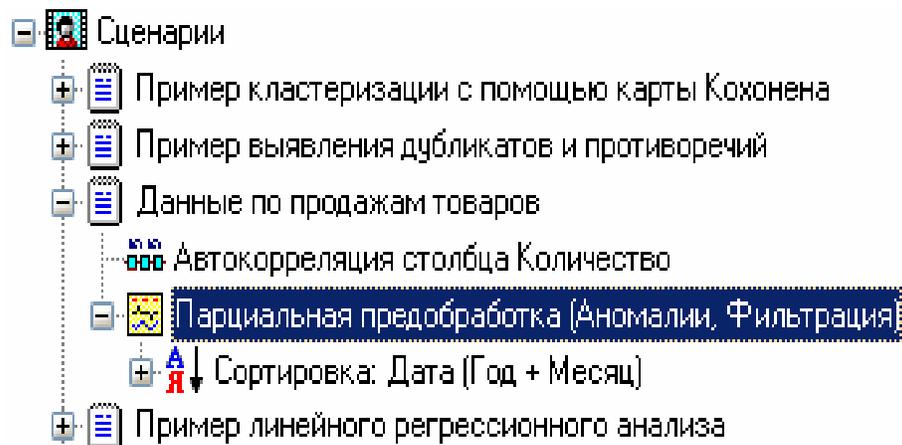


Рис.2. Выбор начального узла

После выбора начального узла задаем 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 задаем 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:

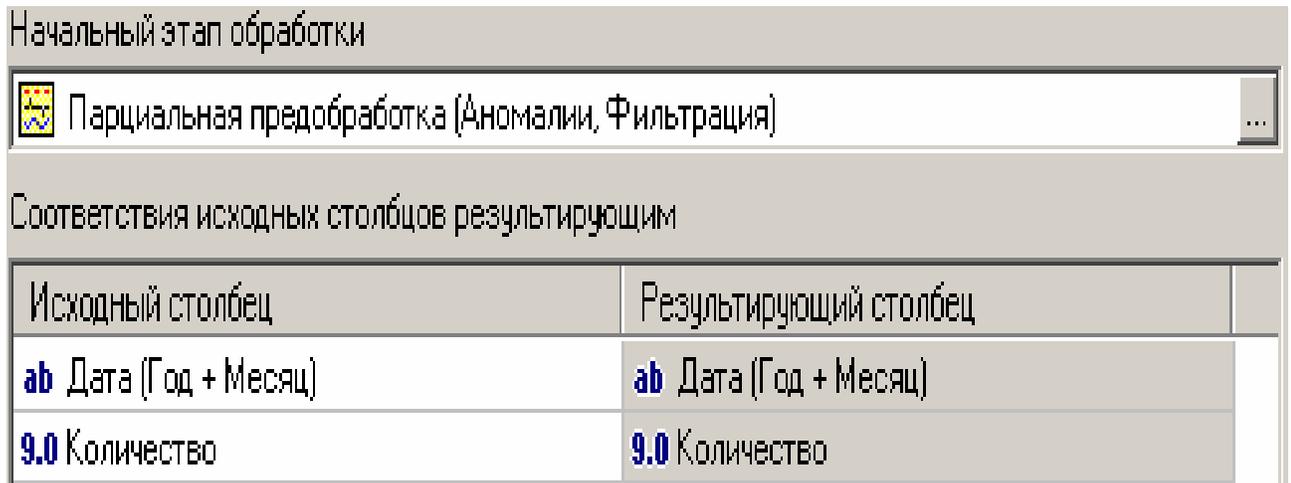
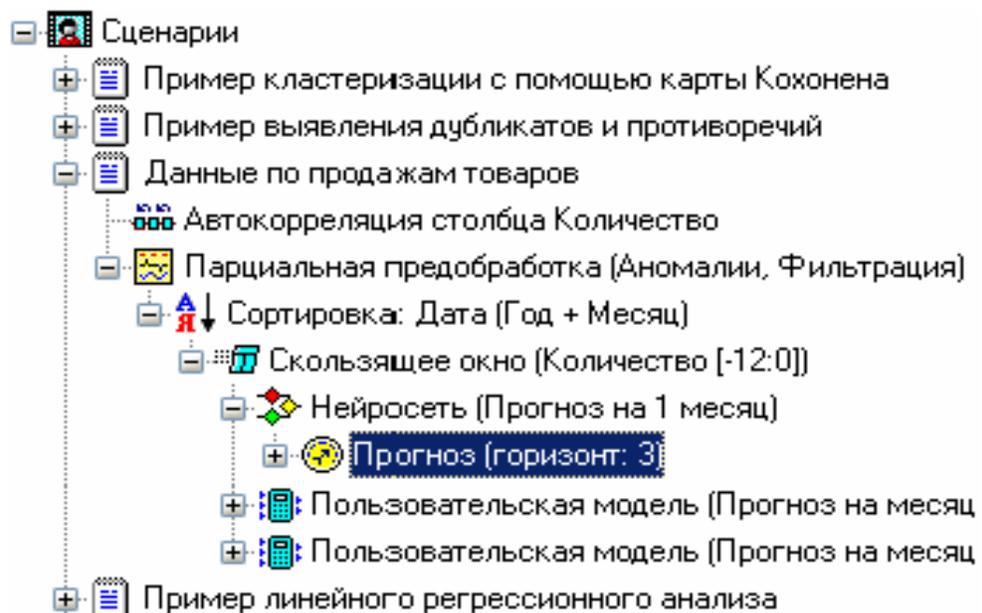


Рис.3. Настройка соответствия полей

На следующем шаге мастера аналогичным образом выбирается конечный узел обработки:



После выбора конечного узла в нижней части окна будет показан список узлов (рис5), входящих в скрипт. При выполнении скрипта последовательно будут выполнены все узлы сценария из этого списка:

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

Рис.5. Список узлов

На следующем шаге запускается процесс анализа данных. Ход процесса обработки отображается с помощью прогресс-индикатора «Процент выполнения текущего процесса». В секции «Название процесса» отображается этап процесса обработки данных, выполняемый в данный момент. Запустим выполнение скрипта и перейдем на закладку выбора способа визуализации. Вот, например диаграмма с прогнозом объема продаж нашей группы товаров (рис.6), полученного с использованием модели прогноза, построенной для другой группы товаров:

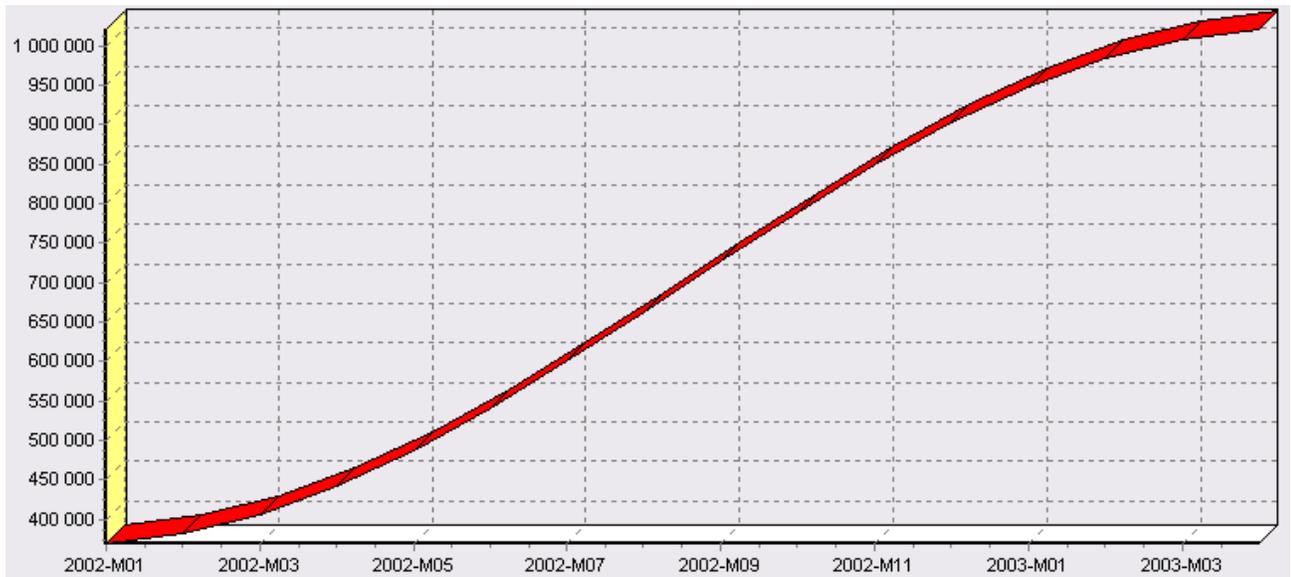


Рис.6. Диаграмма прогноза объема продаж

4. Содержание отчета

Отчет по лабораторной работе представляется в виде документа Word. В состав документа входят:

1. Название работы
2. Цель работы
3. Копии экрана, иллюстрирующие выполнение лабораторной работы
4. Выводы по работе

5. Контрольные вопросы

1. Что такое скрипт?

2. Для чего предназначен скрипт?
3. В каких случаях его следует использовать при проведении анализа данных?
4. Какой инструмент в системе Deductor позволяет применить имеющуюся в сценарии последовательность обработок одних данных к аналогичному набору других данных

6. Список рекомендуемой литературы

1. Загоруйко Н.Г. Прикладные методы анализа данных и знаний.-Новосибирск: Изд-во Ин-та математики, 1999. - 270 с.
2. Ханк Д.Э., Уичерн Д.У., Райтс А.Дж. Бизнес-прогнозирование, 7-е издание.: Пер. с англ. - М.: Издательский дом «Вильямс», 2003. - 656 с.: ил. - Парал. тит. англ.

КЛАСТЕРНЫЙ АНАЛИЗ С ИСПОЛЬЗОВАНИЕМ КАРТ КОХОНЕНА

1. Цель и содержание работы

Цель работы - освоение основных методов и способов кластеризации с использованием самоорганизующихся карт Кохонена, освоение принципов построения и использования простейших нейронных сетей, приобретение практических навыков по использованию инструментария Deductor Studio.

Содержание работы:

Разработать сценарии построения самоорганизующихся карт Кохонена.

Создать сводную таблицу (например, банков), включив в нее суммарные сведения. Таблицу получить путем слияния соответствующих полей из разных таблиц и последующей группировки.

Для подготовленной сводной таблицы разработать сценарий кластеризации с использованием самоорганизующихся карт Кохонена.

Создать отчеты по всем разработанным сценариям.

Продолжительность выполнения работы - 4 часа.

2. Теоретические сведения

Самоорганизующиеся карты признаков Кохонена.

Самоорганизующиеся карты признаков (СКП) являются разновидностью неуправляемых нейросетей. Они были предложены Тьюво Кохоненом в начале 80-х годов прошлого века и нашли широкое применение в инженерной области (для распознавания речи, в робототехнике и др.).

Технология СКП представляет собой набор аналитических процедур и алгоритмов, позволяющих преобразовывать традиционное описание множества объектов, заданных в многомерном ($n > 3$) пространстве признаков плоской базы данных, в двумерную карту. Полученная карта устроена таким образом, что близким объектам в многомерном пространстве отвечают рядом стоящие точки (их образы) на карте.

В результате, трудно анализируемые в совокупности многомерные объекты получают простой и наглядный вид на двумерной карте, которая сохраняет их основные свойства (топологию и распределение в многомерном пространстве).

Применение технологии СКП дает ряд преимуществ:

обнаружение групп объектов с одинаковыми характеристиками (далее - кластеров) по их локализованному расположению на специально создаваемой карте кластеров;

проверка содержательного описания обнаруженных групп по специфическим особенностям, обнаруженным на карте признаков, а также на проекциях карты кластеров на каждый признак в отдельности;

выявление неявных связей и закономерностей между признаками;

проведение оценки объектов в динамике, оценка изменений как в целом по структуре кластеров, так и по отдельности;

позиционирование на карту новых объектов для придания им статуса (рейтинга);

прогнозирование значений одних признаков объектов через другие;

фильтрация объектов за счет поисковых уникальных критериев, формируемых в терминах СКП.

Разберем действие нейросетевой модели самоорганизующихся карт Кохонена для маркетингового анализа.

Как уже было указано, характеристики организаций на рынке можно получить, анализируя различные показатели их работы и связи между ними. Для этого следует использовать данные финансовых отчетов. Из них эксперты извлекают значения различных параметров (активы, капитализацию, прибыль и т. д.). Но для получения достаточно достоверной информации приходится анализировать много взаимосвязей большого количества параметров. Эта задача непростая. Часто для описания субъекта рынка используется несколько десятков различных показателей, а человек обычно не может оперировать более чем тремя параметрами одновременно. Поскольку информации для анализа нужно много и чаще всего она разнородна, то невозможно окинуть одним взглядом весь этот набор.

В современном маркетинге достаточно часто возникает задача анализа данных, которые с трудом можно представить в математической числовой форме. Это случай, когда нужно извлечь данные, принципы отбора которых заданы нечетко: выделить надежных партнеров, определить перспективный товар, выявить основных конкурентов.

Предположим, что имеется информация о деятельности нескольких десятков фирм на рынке (их открытая финансовая отчетность) за некоторый период времени. По окончании этого периода исследователю известно, какие из этих фирм обанкротились, а какие продолжают стабильно работать (на момент окончания периода). И теперь необходимо решить вопрос о том, какие из них являются приоритетными с точки зрения сотрудничества. Значит, следует каким-то образом решить задачу анализа рисков сотрудничества с

различными коммерческими структурами.

На первый взгляд, решить эту проблему несложно - есть данные о работе фирм и результат их деятельности. Но при этом возникает сложность, связанная с тем, что существующие данные описывают прошедший период, а исследователю интересно то, что будет в дальнейшем. Таким образом, необходимо на основании имеющихся априорных данных получить прогноз на дальнейший период. Для решения этой задачи можно использовать различные методы.

Так, например, наиболее очевидным является применение методов математической статистики. Однако недостатком подобных методов является потребность в большом объеме априорных данных, а в выбранном примере может быть ограниченное их количество. При этом статистические методы зачастую не могут гарантировать успешный результат.

Следовательно, нужно попытаться найти эти закономерности, с тем, чтобы использовать их в дальнейшем. И тут возникает вопрос: как найти эти закономерности? Для этого, если будут применяться методы статистики, исследователь должен определить, какие критерии «похожести» использовать, а это может потребовать от него каких-либо дополнительных знаний о характере задачи.

Другим путем решения этой задачи может быть применение нейронных сетей. Метод анализа с использованием самоорганизующихся карт Кохонена - это метод, позволяющий автоматизировать все действия по поиску закономерностей. Рассмотрим, как решаются такие задачи и как карты Кохонена находят закономерности в исходных данных. Для общности рассмотрения здесь и далее будем использовать термин объект (например, объектом может быть фирма-клиент, как в рассмотренном выше примере, но описываемый метод без изменений подходит для решения и других задач, например, анализа конкуренции, поиска оптимальной стратегии поведения на рынке). В данной работе описывается способ применения указанного метода для анализа клиентов на рынке (реальных и

потенциальных).

Каждый объект характеризуется набором различных параметров, которые описывают его состояние. Для примера по анализу фирм-клиентов параметрами можно взять данные из финансовых отчетов. Эти параметры часто имеют числовую форму или могут быть приведены к ней.

Как уже было указано выше, решение задачи предполагает на основании анализа параметров объектов выделение схожих объектов и представление результата в форме, удобной для восприятия. Все эти подзадачи успешно и эффективно решаются самоорганизующимися картами Кохонена. В целях упрощения рассмотрения будем считать, что объекты имеют три признака (на самом деле их может быть любое количество).

Предположим, что все эти три параметра объектов представляют собой их координаты в трехмерном пространстве. Например, для промышленного предприятия это могут быть следующие показатели: капитализация, объем реализованной продукции, прибыль. Тогда каждый объект можно представить в виде точки в этом пространстве, что и сделаем (чтобы не было проблем с различным масштабом по осям, пронормируем все эти признаки в интервал $[0,1]$ любым подходящим способом). В результате проведенной нормировки все точки попадут в куб единичного размера.

Отообразим эти точки (рис. 1).

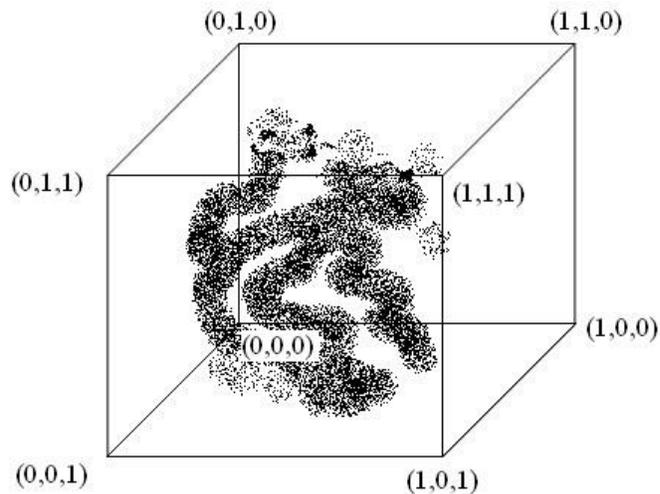


Рис. 1. Расположение объектов в трехмерном пространстве

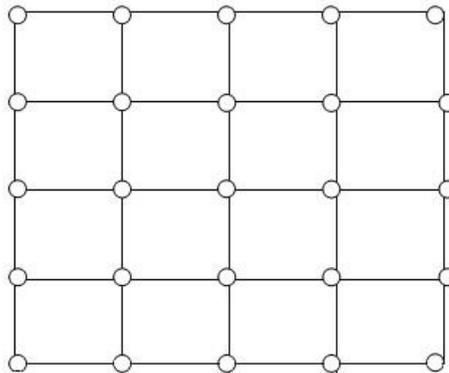


Рис. 2. Карта Кохонена

Анализ полученного рисунка позволяет увидеть, как расположены объекты в пространстве, причем легко заметить участки, где объекты группируются (сгущения). Распределение объектов таким образом означает, что у них схожи параметры, значит, и сами эти объекты принадлежат одной группе. Очевидно, что такой легкий способ можно применить только в том случае, когда признаков немного, поскольку человеческий разум не может представить изображение четырехмерного пространства. Следовательно, необходимо найти способ, которым можно преобразовать данную систему в простую для восприятия, желательно двумерную систему (потому что уже трехмерную картинку невозможно корректно отобразить на плоскости) так, чтобы соседние в изучаемом пространстве объекты оказались рядом и на полученной картинке. Для

этого используем самоорганизующуюся карту Кохонена. В первом приближении ее можно представить в виде гибкой сети (рис. 2).

Эластичную сеть карты исследователь помещает в пространство признаков, где уже имеются объекты, которые необходимо проанализировать. Далее система работает следующим образом: берется один объект (точка в исследуемом пространстве) и выявляется ближайший к нему узел сети. После этого данный узел подтягивается к объекту (сетка эластична, поэтому вместе с этим узлом так же, но с меньшей силой подтягиваются и соседние узлы). Затем выбирается другой объект (точка), и процедура повторяется. В результате строится карта, расположение узлов которой совпадает с расположением основных скоплений объектов в исходном пространстве. Кроме того, полученная карта обладает следующим замечательным свойством - узлы ее расположились таким образом, что объектам, похожим между собой, соответствуют соседние узлы карты (рис. 3). Теперь следует определить, в какие узлы карты попали те или иные объекты. Это также определяется ближайшим узлом - объект попадает в тот узел, который находится ближе к нему. В результате всех этих операций объекты со схожими параметрами попадут в один узел или в соседние узлы. Таким образом, можно считать, что благодаря системе самоорганизующихся карт Кохонена исследователь решает задачу поиска похожих объектов и их группировки.

Самоорганизующиеся карты Кохонена обладают и другими возможностями. Они позволяют также представить полученную информацию в простой и наглядной форме путем нанесения раскраски. Для этого исследователь раскрашивает полученную карту цветами, соответствующими интересующим признакам объектов. Возвращаясь к примеру с анализом фирмклиентов на рынке, можно раскрасить одним цветом те узлы, куда попала хотя бы одна фирма, у которой наблюдаются убытки. Тогда после нанесения цвета мы получим зону, которую можно назвать зоной риска, и попадание интересующей нас фирмы в эту зону говорит о ее ненадежности.

С помощью карт можно также получить информацию о зависимостях между параметрами. Отмечая на карте различные статьи финансовых и экономических отчетов отдельными цветами, менеджер-исследователь получит атлас, хранящий в себе информацию о состоянии рынка. Сравнивая расположение цветов на раскрашенных картах, подготовленных таким образом, руководитель получает полную информацию о финансовом и экономическом портрете фирм-клиентов - банкротов, неудачников, процветающих фирм, «средняков». Например, таким показателем может быть чистая прибыль фирмы.

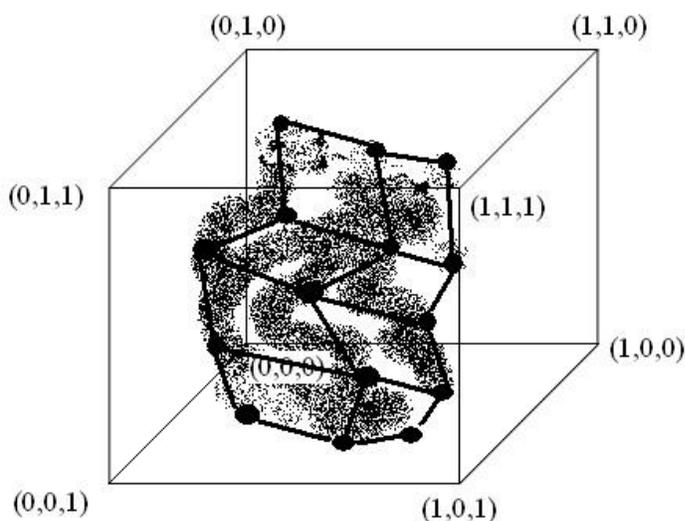


Рис. 4. Вид пространства после наложения карты

При всем этом, описанная технология является универсальным методом анализа. С ее помощью можно анализировать различные стратегии деятельности, производить анализ результатов маркетинговых исследований, проверять кредитоспособность клиентов и т. д. В данной работе технология самоорганизующихся карт Кохонена применяется для анализа клиентской базы. Этот универсальный многофункциональный инструмент анализа способен представить достаточно четкую картину реальных и потенциальных клиентов-фирм, работающих на рынке. Эта технология также полезна и потому, что в России большая часть деловой информации является

засекреченной, и в результате информация, на основании которой приходится работать, крайне искажена и часто носит неправдоподобный характер.

Таким образом, имея перед собой карту, исследователь может достаточно достоверно судить об объектах, даже если имеет неполную информацию об этих объектах. В результате, можно извлекать информацию из базы данных, основываясь на нечетких характеристиках.

В отличие от классических методов, самоорганизующиеся карты обеспечивают простую визуализацию данных, навязывают несколько меньшее количество предположений и ограничений и обнаруживают изолированные структуры в данных, оперируя с большим количеством комплексных данных.

Учитывая показанные возможности самоорганизующихся карт, можно определить следующие основные области применения этих карт в маркетинге:

- анализ товарных рынков на основании потребительских предпочтений;

- сегментирование покупателей и клиентов;

- информационное обеспечение выработки маркетинговых решений и анализа рынка;

- конкурентный анализ.

Алгоритм функционирования самоорганизующихся карт (Self Organizing Maps - SOM) представляет собой один из вариантов кластеризации многомерных векторов - алгоритм проецирования с сохранением топологического подобия.

Примером таких алгоритмов может служить алгоритм k -ближайших средних (k -means). Важным отличием алгоритма SOM является то, что в нем все нейроны (узлы, центры классов) упорядочены в некоторую структуру (обычно двумерную сетку). При

этом в ходе обучения модифицируется не только нейрон-победитель (нейрон карты, который в наибольшей степени соответствует вектору входов, и определяет, к какому классу относится пример), но и его соседи, хотя и в меньшей степени. За счет этого SOM можно считать одним из методов проецирования многомерного пространства в пространство с более низкой размерностью. При использовании этого алгоритма векторы, схожие в исходном пространстве, оказываются рядом и на полученной карте.

SOM подразумевает использование упорядоченной структуры нейронов. Обычно применяются одно- и двумерные сетки. При этом каждый нейрон представляет собой n -мерный вектор-столбец, где n определяется размерностью исходного пространства (размерностью входных векторов).

Применение одно- и двумерных сеток связано с тем, что возникают проблемы при отображении пространственных структур большей размерности (при этом опять возникают проблемы с понижением размерности до двумерной, представимой на мониторе).

Обычно нейроны располагаются в узлах двумерной сетки с прямоугольными или шестиугольными ячейками. При этом, как было сказано выше, нейроны также взаимодействуют друг с другом. Величина этого взаимодействия определяется расстоянием между нейронами на карте.

При реализации алгоритма SOM заранее задается конфигурация сетки (прямоугольная или шестиугольная), а также количество нейронов в сети. Некоторые источники рекомендуют использовать максимально возможное количество нейронов в карте. При этом начальный радиус обучения (*neighborhood* в англоязычной литературе) в значительной степени влияет на способность обобщения при помощи полученной карты. В случае, когда количество узлов карты превышает количество примеров в обучающей выборке, успех использования алгоритма в большой степени зависит от подходящего выбора начального радиуса обучения. Однако в случае, когда размер карты

составляет десятки тысяч нейронов, время, требуемое на обучение карты, обычно бывает слишком велико для решения практических задач. Таким образом, необходимо достигать допустимый компромисс при выборе количества узлов.

Перед началом обучения карты необходимо проинициализировать весовые коэффициенты нейронов. Удачно выбранный способ инициализации может существенно ускорить обучение и привести к получению более качественных результатов.

Существуют три способа инициирования начальных весов:

инициализация случайными значениями, когда всем весам даются малые случайные величины;

инициализация примерами, когда в качестве начальных значений задаются значения случайно выбранных примеров из обучающей выборки;

линейная инициализация, в этом случае веса иницируются значениями векторов, линейно упорядоченных вдоль линейного подпространства, проходящего между двумя главными собственными векторами исходного набора данных.

Обучение карты заключается в последовательности коррекции векторов, представляющих собой нейроны. На каждом шаге обучения из исходного набора данных случайно выбирается один из векторов, а затем производится поиск наиболее похожего на него вектора коэффициентов нейронов. При этом выбирается нейрон-победитель, который наиболее похож на вектор входов. Под похожестью в данной задаче понимается расстояние между векторами, обычно вычисляемое в евклидовом пространстве.

После того, как найден нейрон-победитель, производится корректировка весов карты. При этом вектор, описывающий нейрон-победитель, и векторы, описывающие его соседей в сетке, перемещаются в направлении входного вектора.

Обучение состоит из двух основных фаз: на первоначальном этапе выбирается достаточно большое значение скорости обучения и радиуса обучения, что позволяет расположить векторы нейронов в соответствии с распределением примеров в выборке, а затем производится точная подстройка весов, когда значения параметров скорости обучения много меньше начальных. В случае использования линейной инициализации, первоначальный этап грубой подстройки может быть пропущен.

Самоорганизующиеся карты могут использоваться для решения таких задач, как моделирование, прогнозирование, поиск закономерностей в больших массивах данных, выявление наборов независимых признаков и сжатие информации.

В результате обучения самоорганизующейся карты в исходную выборку данных будут добавлены следующие поля:

1. *<ИМЯ ПОЛЯ>_OUT* - содержит значения выходных полей, рассчитанные картой.
2. *Номер ячейки* - содержит номер ячейки карты, в которую попала данная запись.
3. *Расстояние до центра ячейки* - содержит значение расстояния от данной записи до центра ячейки, в которую эта запись попала.
4. *Номер кластера* - указывается номер кластера, где расположена ячейка, в которую попала данная запись исходной выборки.
5. *Расстояние до центра кластера* - указывается значение расстояния от ячейки, куда попала данная запись исходной выборки, до центра кластера.
6. *<ИМЯ ПОЛЯ>_ERR* - содержит среднеквадратичную ошибку рассогласования реального значения поля и значения, рассчитанного картой.

Настройка назначения полей. Здесь необходимо определить, как будут использоваться поля исходного набора данных при обучении самоорганизующейся карты и практической работе с ней. Для настройки поля следует выделить его в списке, при этом в правой части окна будут отображены его параметры:

1. *Имя поля* - идентификатор поля, определенный для него в источнике данных. Изменить его здесь нельзя.

2. *Тип данных* - тип данных, содержащихся в поле (вещественный, строковый, дата). Он также задается в источнике данных и здесь изменен быть не может.

3. *Назначение* - здесь необходимо выбрать порядок использования данного поля при обучении и работе самоорганизующейся карты Кохонена. Выбор производится с помощью списка, открываемого кнопкой и содержащего следующие варианты:

Входное - поле будет использовано как одна из координат входного вектора, которые алгоритм построения карты Кохонена будет кластеризовать. По этому полю можно будет впоследствии посмотреть карту распределения значений этого поля;

Выходное - при построении карты это поле использоваться не будет, однако после построения по этому полю будет собрана статистика для каждой ячейки и для каждого кластера полученной карты. Таким образом, можно говорить о выходе (классе) ячейки по этому выходному полю. Например, если выходное поле - это дискретное поле, то выходом ячейки (по этому выходному полю) будет являться самое распространенное значение выходного поля тех строчек данных, которые «попали» в данную ячейку. Если же выходное поле - это непрерывное поле, то выходом ячейки (по этому выходному полю) будет являться среднее значение выходного поля

тех строчек данных, которые «попали» в данную ячейку. При таком подходе это поле можно рассматривать как целевое, как если бы мы рассматривали задачу регрессии или классификации;

Информационное - поле не будет использоваться при обучении карты, но будет помещено в результирующий набор в исходном состоянии;

Неиспользуемое - поле не будет использоваться при построении и работе с картой и будет исключено из результирующего набора. В отличие от непригодного, такое поле может быть использовано, если в этом возникнет необходимость;

Непригодное - поле не может быть использовано при построении и работе алгоритма, но будет помещено в результирующий набор в исходном состоянии.

4. *Вид данных* - указывает на характер данных, содержащихся в поле (непрерывный или дискретный). Изменить это свойство здесь нельзя.

Статус непригодного поля устанавливается только автоматически и в дальнейшем может быть изменен только на неиспользуемое или информационное. Поле будет запрещено к использованию, если:

поле является дискретным и содержит всего одно уникальное значение;

непрерывное поле с нулевой дисперсией;

поле содержит пропущенные значения.

3. Порядок выполнения работы

Рассмотрим механизм кластеризации путем построения

самоорганизующейся карты, основываясь на информации по банкам. Исходная таблица находится в файле "Banks.txt". Задача состоит в том, чтобы определить по различным данным банка его прибыль и наличие скрытых закономерностей.

Для начала необходимо импортировать данные из файла. После этого запустим Мастер обработки и выберем из списка метод обработки "Карта Кохонена". На втором шаге Мастера настроим назначения столбцов. Укажем столбцу "Прибыль" назначение "Выходной", а "Филиалы", "Сумма активов", "Собственные активы", "Банковские активы", "Средства в банке" - "Входной", т. е. на основе данных о банке будем относить его к тому или иному классу.

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

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.

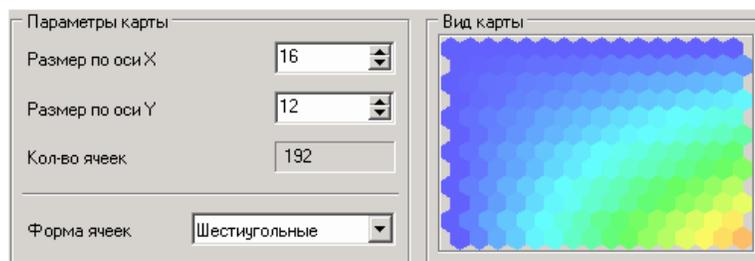


Рис. 5. Настройка параметров карты

На пятом шаге Мастера следует настроить параметры остановки обучения. Оставим параметры по умолчанию.

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

Рис. 6. Параметры остановки обучения

На шестом шаге настраиваются остальные параметры обучения: способ начальной инициализации, тип функции соседства, а также параметры кластеризации - автоматическое определение числа кластеров с соответствующим уровнем значимости либо фиксированное количество кластеров.

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

Рис. 7. Настройка дополнительных параметров карты

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

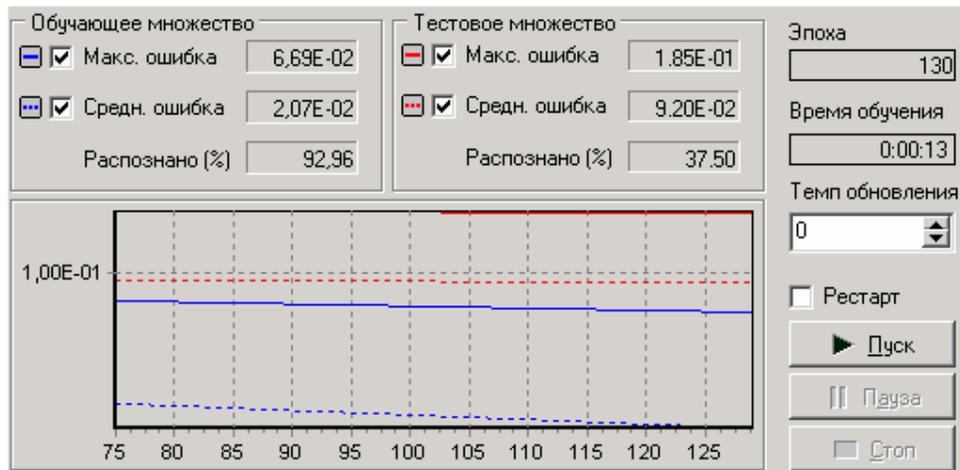


Рис. 8. Запуск процесса обучения

После этого требуется в списке визуализаторов выбрать появившуюся теперь Карту Кохонена для просмотра результатов кластеризации, а также визуализатор "Что-если" для прогнозирования прибыли банков.

Далее в Мастере настройки отображения карты Кохонена надлежит указать поля, которые необходимы для отображения.

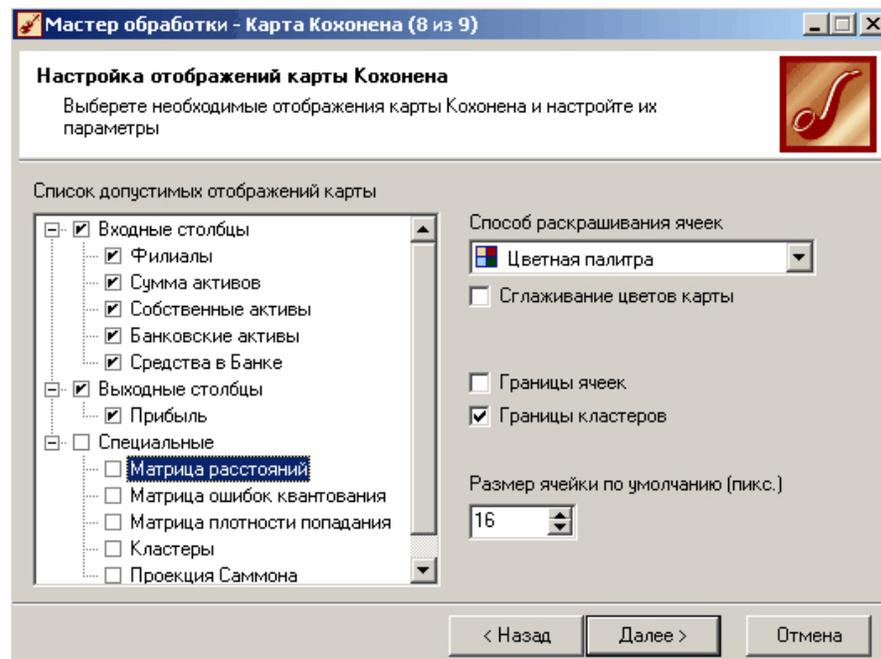


Рис. 9. Настройка визуализации результатов анализа

В итоге получаем Карту Кохонена.

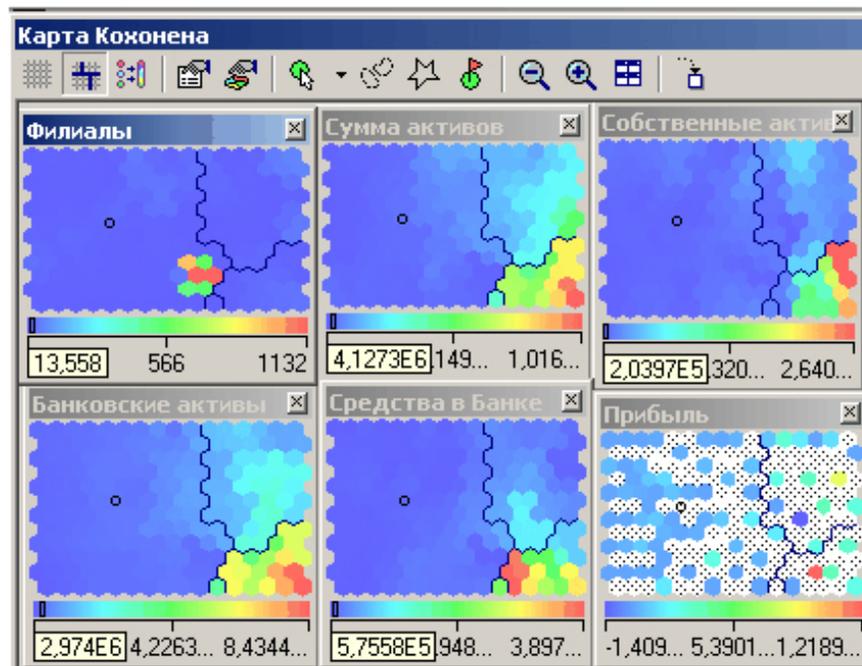


Рис. 10. Карты Кохонена

Можно видеть, что наиболее прибыльные банки попали в кластеры, что находятся в правой части карты. Для этих банков характерны большая сумма активов и средств в банке. Количество же филиалов не оказывает существенного влияния на прибыльность, т.к. банки с большим количеством филиалов разместились в левом не самом прибыльном кластере (см. проекцию "Филиалы").

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

4. Содержание отчета

Отчет по лабораторной работе представляется в виде документа Word. В состав документа входят:

1. Название работы
2. Цель работы

3. Копии экрана, иллюстрирующие выполнение лабораторной работы
4. Выводы по работе

5. Контрольные вопросы

1. Поясните необходимость использования карт Кохонена при кластеризации.
2. Объясните общий принцип построения самоорганизующейся карты признаков Кохонена.
3. Каким образом производится назначение размеров и формы ячеек (нейронов) карты Кохонена?
4. Как осуществляется назначение начальных значений весовых коэффициентов нейронов?
5. Поясните понятия скорости и радиуса обучения нейросети.
6. Какие критерии используются для остановки процесса обучения?

6. Список рекомендуемой литературы

1. Анализ данных и процессов / А. А. Барсегян [и др.]. - СПб. : БХВПетербург, 2009. - 512 с.
2. Афонин, А. Ю. Оперативный и интеллектуальный анализ данных / А. Ю. Афонин, П. П. Макарычев. - Пермь : Изд-во ПГУ,

2010. - 142 с.

ОГЛАВЛЕНИЕ

ЛАБОРАТОРНАЯ РАБОТА №1	10
ЛАБОРАТОРНАЯ РАБОТА №2	30
ЛАБОРАТОРНАЯ РАБОТА №3	38
ЛАБОРАТОРНАЯ РАБОТА №4	53
ЛАБОРАТОРНАЯ РАБОТА №5	66
ЛАБОРАТОРНАЯ РАБОТА №6	78
ЛАБОРАТОРНАЯ РАБОТА №7	92
ЛАБОРАТОРНАЯ РАБОТА №8	99