***ФГБОУ ВО «Кубанский государственный аграрный университет имени И.Т. Трубилина»,***

***Российская Федерация***

**95 Инженирия знаний и интеллектуалные системы. Лабораторная 2. Разбор лабораторной 3.0.3 2020-09-23**

**Руководство по установке и настройке системы Aidos для лабораторной работы №2**

**Резюме**

**Введение и организационные моменты:**
Профессор Луценко Евгений Вениаминович проводит лабораторную работу №2 по дисциплине "Инженерия знаний и интеллектуальные системы" для группы БИ 1701 (23 сентября 2020 г.). Студентов просят отвечать микрофоном.

**Установка системы Aidos:**
Основная задача – установка системы Aidos на компьютеры студентов. Профессор напоминает, что на прошлом занятии показывал процесс установки, и просит студентов отметить в чате (+/-), установлена ли у них система. Выясняется, что у большинства система не установлена.

**Демонстрация установки и настройки:**
Профессор демонстрирует процесс скачивания системы Aidos со своего сайта (lc.kubagro.ru) и ее установки путем распаковки архива (самораспаковывающегося или .rar) в корневой каталог диска (например, C:\ или D:). Подчеркивается, что система портативная и не требует инсталляции.

**Настройка лабораторной работы:**
После установки профессор показывает, как подготовить систему к выполнению конкретной лабораторной работы:

1. Зайти в режим 1.11 и стереть все предыдущие приложения.
2. Зайти в режим 1.3 (Диспетчер приложений).
3. Выбрать "Добавить лабораторную работу".
4. Выбрать работу 3-го типа (установка путем ввода из внешних баз исходных данных).
5. Добавить работу 3.03 "Идентификация предметов по их признакам".
6. Подтвердить удаление старых данных в папке indata.
7. Принять параметры по умолчанию (диапазоны шкал и т.д.), которые объяснялись на прошлом занятии.

**Обзор системы Aidos и типов лабораторных работ:**
Кратко упоминаются четыре типа лабораторных работ, доступных в системе, включая работы с готовыми данными, с расчетными данными, с данными из внешних файлов (как 3.03) и с данными из интернета (парсинг). Также упоминается возможность скачивания дополнительных "облачных" лабораторных работ (курсовых, НИР, ВКР) с FTP-сервера через интерфейс системы, где каждая работа является темой для обсуждения на форуме.

**Структура входных данных (Excel):**
Объясняется структура входного Excel-файла для работ 3-го типа:

* Строки – наблюдения (объекты).
* Первая колонка – наименование объекта.
* Последующие колонки делятся на:
	+ Классификационные (желтый фон в примере): описывают принадлежность объекта к обобщающим категориям (например, "млекопитающее", "студент группы X").
	+ Описательные (зеленый фон в примере): описывают свойства объекта или факторы, на него влияющие (например, цвет: красный; вес: 60 кг; температура: -25°C).
* Шкалы бывают текстовые (номинальные, порядковые) и числовые. Система Aidos работает со всеми типами.
* Важно правильно форматировать данные в Excel, чтобы область значений (Ctrl+Shift+End) точно соответствовала фактическим данным без пустых строк/столбцов.

**Возможности системы и теоретические основы:**
Система Aidos позволяет на основе обучающей выборки (примеров) строить обобщенные модели (знания) для классификации и идентификации объектов. Упоминается связь между частотами и вероятностями, концепция нормализации данных (создание справочников), преобразование данных в информацию и знания, причинно-следственные связи, валидность моделей (внутренняя и внешняя), метод Bootstrap для оценки достоверности на подмножествах данных.

**Технические вопросы и устранение неполадок:**
Обсуждаются проблемы с запуском Aidos на новых версиях Windows и Excel (16+). Рекомендации:

* Проверить настройки совместимости Windows (эмуляция, режим совместимости).
* Использовать Excel 2010 или более ранний.
* Устанавливать Aidos в путь без пробелов и кириллицы.
* Попробовать запустить не start\_aidos.exe, а напрямую aidos\_X.exe.
* Установить на другой логический или физический диск.
* В крайнем случае, использовать виртуальную машину с Windows 7.
* Прислать скриншоты ошибок профессору на почту.

**Заключение:**
Занятие завершается. Профессор просит напомнить ему за 15 минут до конца следующего занятия.

**Детальная расшифровка текста**

**Введение и организационные моменты**

Здравствуйте, студенты. Вы отвечаете микрофоном, а не в чате, у кого есть возможность.
Здравствуйте.
Здравствуйте. Сейчас у нас лабораторная работа номер два. 23 сентября 2020 года. В группе БИ 1701 лабораторная работа номер номер два, которая идет с 17:20 до 18:50 по дисциплине Инженерия знаний и интеллектуальные системы.

Занятие ведет профессор Луценко Евгений Вениаминович.

Теперь, давайте посмотрим, на чем мы остановились на прошлом занятии. Тогда я еще не стал, не отмечал еще. Ну, наверное, я показывал вам, я так думаю, по плану, как устанавливать систему Aidos. Вы должны были ее установить у себя на компьютерах.

**Установка системы Aidos**

Сейчас в чате отметьте... как вы делитесь? Ну, все ко мне должны прийти. И у меня все должны заниматься. Вот. Если это возможно. Если у вас параллельно подгруппы занимаются в одно время со мной и с Георгием Александровичем. А если в разное время, тогда вот мы с Мариной занимаемся с ее подгруппой, а с Георгием Александровичем там уже он будет заниматься с вами.

Да, здравствуйте, ребята. Значит, у вас должна быть установлена система Aidos на компьютере, на вашем. Он должен быть рядом с вами, потому что это лабораторная работа. Поэтому, значит, давайте мне сейчас или микрофоном, но лучше в чате. У кого установлено, ставьте плюсики, а у кого не установлено – минусики.

Ну ссылочку на этот вот чат прямо пришлите им. Пусть входят тоже сюда в это совещание.

Ну тут я вижу только у двух установлено. Ну вообще-то вас и немного-то здесь. Ну, тогда давайте устанавливайте. Потому что дальше мы будем двигаться, нам нужно уже в системе работать.

**Демонстрация установки и настройки**

Сейчас я поделюсь экраном, чтобы видели экран, и буду тоже скачивать ее.

Значит, это делается, я вам объяснял на прошлом занятии, с моего сайта. Экран видно, ребят? Видно мой сайт, да? Вы отвечаете в чате, а мне в чате надо выключать и смотреть. Да, да, все видно.

Значит, на моем сайте, который адрес lc.kubagro.ru, второй пунктик нажимаем и двигаемся вот до картинки и кликаем по ссылочке "Скачать и запустить систему Aidos". Скачиваем либо самораспаковывающийся архив, либо rar-архив. Вот, скачиваем его и потом, когда он скачается...

Кликаем по самому уже скачанному архиву, запускаем и выбираем место для разархивирования. Это место желательно, чтобы это был корневой каталог какого-нибудь диска. Вот. В любом случае, ну вот здесь вот увидите сейчас. Можно это руками написать, не с помощью обзора, а руками. Корневой каталог какого-нибудь диска. И извлечь.

Собственно вот и вся установка системы. Я рассказывал на прошлом занятии, что она не инсталлируется, а просто разворачивается и работает. То есть это система portable. И вот на диске Е появилась папочка с системой. И там все лабораторные работы. Та лабораторная работа, которую мы сейчас будем проходить. Но я буду на диске C показывать вам по причине, которую я тоже объяснял, что этот диск у меня работает быстрее.

**Настройка лабораторной работы**

Я запускаю систему. Вот. И сразу же, сразу же захожу в режим 1.11 и стираю все приложения. Потом захожу в режим 1.3, жирненьким шрифтом "Добавить лабораторную работу". Выбираю работу третьего типа и добавляю работу 3.03. Она сообщает, что там все будет стерто. Потом все параметры по умолчанию. Я эти параметры вам подробно-подробно объяснял на прошлом занятии. Если не объяснял, то, пожалуйста, сейчас мне об этом скажите. Объяснял вам эти вот параметры или нет? Интерфейса.

Нет, вы не объясняли. Вы просто, да, объяснили, как её поставить, но не объясняли, как ставить лабораторные на неё.
Понятно. Ну тогда давайте сейчас этим и займемся. Не обращайте внимание, что на экране.

Вот систему поставили. Главное окно системы мы видим. Главное меню. Значит, сейчас что мы делаем? Мы должны зайти в режим 1.3. Я их все пронумеровал, все подсистемы пронумеровал от одного до семи, и в каждой подсистеме режимы пронумеровал. Вот мы заходим в режим Диспетчер приложений. Это режим, который как раз предназначен для установки учебных приложений, то есть лабораторных работ.

Вот, я кликаю по нему и нажимаю кнопочку "Добавить лабораторную работу". Видите, да? Появляется вот такая экранная форма. Да, все видно. Купи веб-камеру с микрофоном и говори голосом.

**Обзор системы Aidos и типов лабораторных работ**

Значит, здесь есть четыре типа лабораторных работ. Какие именно?
Первый тип лабораторных работ – полностью готовы для изучения.
Второй тип – исходные данные формируются расчетным путем. Вот эти работы и по ним есть описание.
Третье – лабораторные работы третьего типа устанавливаются путем ввода из внешних баз исходных данных. Путем ввода из внешних баз исходных данных. Это самый интересный тип лабораторных работ. С него я и начал, с работы 3.03.
Четвертый тип лабораторных работ, я тут одну работу сделал здесь и больше пока не думаю. Это работы, которые берут данные из интернета. Скачивается с сайтов информация для лабораторной работы. Так называемый парсинг.

Вот смотрим работы третьего типа. Выбираем работу 3.03. Мы ее первой будем изучать. Сейчас я на своем сайте дам вам ссылочку, ребят, со своего сайта, где есть описание того, что мы делаем, последовательности вот этой. Эта последовательность, она связана, то есть я вас веду по пути, чтобы такому пути, по которому вы бы освоили сами разработку облачных Aidos-приложений. Или, может быть, не облачных, а просто разработку приложений. Вот. Вот сейчас в чат вам пошлю эту ссылочку. И Марина, можешь другой группе тоже послать эту ссылочку, подгруппе.

Вот. Здесь вот видите, написано: читаем про систему Aidos – это на лабораторной работе уже, как ее устанавливать, что она включает в себя. Скачиваем здесь и устанавливаем. Вот это мы сделали на прошлой работе. Сейчас я вам показал, что там есть работы трех типов, встроенные локальные, четырех даже. И мы начинаем изучать работу 3.03. Потом 3.02, потом 3.04. Это уже на следующих занятиях. И потом будем двигаться дальше. Вот.

Еще есть в системе Aidos работы, это 31, 31 работа в самой инсталляции, локальные лабораторные работы. А еще есть лабораторные работы облачные, которые находятся на FTP сервере.

Вот. Ну давайте сейчас эту не будем устанавливать лабораторную работу. Здесь вот прервем этот процесс. А я вам покажу, что есть еще одна кнопочка, вот там, где жирным шрифтом, а правее кнопочка "Скачать приложение из облака". И здесь, видите, вот есть довольно много, 207 лабораторных работ. Но они, я их называю лабораторные, на самом деле это и курсовые работы, и научно-исследовательские, и лабораторные, и здесь даже есть дипломные работы, ВКР. Причем не из одного вуза, а из нескольких. И вот здесь вот есть описание этих работ, ссылочки на описание. Любую работу, любое описание мы можем прочитать, вот так два раза клацнув по ней. Скачивается сразу же браузер по умолчанию запускается и скачивается, устанавливается лабораторная работа, вернее, описание этой лабораторной работы, мы можем ее читать. Вот. А если мы хотим ее выполнить, тогда вот здесь вот есть кнопочка "Установка Aidos приложения из облака". Если мы по нему кликнем, тогда эта лабораторная работа будет установлена. И можно согласно описанию ее проходить. Значит, мы можем посмотреть, кто какие действия совершал здесь и когда. Мы увидим, что вот есть сообщение, допустим, 19.09 в 204 работе есть сообщение 19 числа. Вот. Это что значит? Это значит, что каждая работа является и каждая лабораторная работа вот эта облачная, является темой форума, которую можно обсуждать. Вот я нажал на обсуждение, показываю: привет там написано. Ну это я показывал студентам, что можно обсуждать. И можно сформировать гиперссылки на все файлы этого облачного приложения и можно скачать, какую хотите из этих файлов. Можно описание скачать. Readme – это описание обычно. Вот. Ну сейчас я вам рассказал про облачное приложение.

**Настройка лабораторной работы (продолжение)**

Вот. Теперь давайте установим лабораторную работу 3.03. Для этого кликаем по кнопочке, которая жирным шрифтом. Выбираем третий тип лабораторных работ и среди них выбираем третью работу 3.03. Идентификация предметов по их признакам. Нажимаем Окей. Возникает на экране маленькая форма экранная, на которой написано, что в папочке indata все данные будут удалены. Эта папочка вот с таким путем, она будет на вашем компьютере по месту расположения системы, именно на вашем компьютере.

Вот. Там могли быть какие-нибудь данные, которые вы использовали. Они сейчас стерты, и туда скопированы данные из лабораторной работы. Где они находятся сами эти лабораторные работы? Если мы войдем в папочку AidosData, в той папке, где система, то там есть папочка LabWorks, то есть лабораторные работы. И вот здесь вот в этих папочках есть все файлы этих лабораторных работ. Ну их по 10 штук, поэтому от с первой по десятую – это работы первого типа, с одиннадцатой по двадцатую – работы второго типа, и с двадцать первой по тридцатую – это работы третьего типа. Вот. Ну поскольку это работа третья, 3.03, то она будет двадцать третьей. То есть это третий десяток, третья работа в этом третьем десятке. Наименование этой лабораторной работы, идентификация предметов по их признакам, и исходные данные. И вот этот файл 2322 – это файл параметров экранной формы, которую сейчас вот мы видим на экране вот здесь.

Когда мы заходим в этот режим установки лабораторной работы, то сначала создается пустое приложение, записывается в папочку indata из файла исходных данных, и потом запускается универсальный программный интерфейс ввода данных из внешних источников данных. Вот этот программный интерфейс я вам не рассказывал, да? Так я понял? То есть мы сейчас впервые видим это, да, ребята? Вы видите, да, нет?
Впервые.
Впервые.

**Структура входных данных (Excel)**

Значит, смотрите. Значит, здесь есть такой хелпик, описание того, с какими, с каким стандартом внешних файлов работает этот интерфейс. Вообще, для чего он предназначен? Он предназначен для того, чтобы вводить данные из внешних файлов экселевских определенного стандарта. Таких программных интерфейсов в системе Aidos много. И с разными стандартами экселевских файлов, и с текстовыми файлами, и с графическими файлами, довольно много разных программных интерфейсов. Вот. Ну это тот один из наиболее развитых для ввода информации из экселевских файлов определенного стандарта. Что это за стандарт? Вот здесь вот требования к файлу исходные данные. Я вам показываю, какой это стандарт.

Значит, смотрите. Значит, строчками в файле являются данные наблюдений. В первой колоночке описано, откуда взяты эти наблюдения. А потом идут колоночки, в которых описывается каждое наблюдение. Оно описывается двумя способами. С одной, один способ описывает, описываются значения свойств объекта в каком-то состоянии. Ну, свойства, что значит значение свойств? Ну это немножко непривычное выражение. Ну я вам приведу пример, вам будет все понятно. Вот, допустим, свойство – цвет, а значение – красный, например. Или другое свойство – вес, а значение – 60 кг, например. Понятно, да? То есть, что такое свойство и его значение, понятно.

И здесь могут быть не свойства и их значения, а могут быть факторы и их значения. Ну, допустим, температура или там давление. То есть это фактор, который действует на объект наблюдения. Вот фактор – температура, там -25°. Температура, допустим, во время цветения. Вот. Понятно, да? Это уже вторая подгруппа здесь тоже, да, присоединилась? Второй подгруппе завтра занятие. Похоже, что сейчас у них нет занятия этого, как я понял.

Вот. А вот эти вот желтым фоном колоночки выделенные – это классификационные шкалы. То есть зеленые – это описательные шкалы, а желтым – это классификационные шкалы. Они предназначены для описания принадлежности вот этого объекта, который в этой строчке описан, в первой, например, к каким-то обобщающим категориям. Вот, допустим, мы хотим, чтобы система решила такую задачу: мы приведем описание разных животных, какие у них свойства, там сколько лапок, есть ли там шерсть там и все прочее. И среди прочих свойств этих животных будем описывать, как они выкармливают своих детенышей. И будем здесь вот писать в колонке классификационной, что это в первой колонке будет написано там кошка Мурка, там собачка Жучка, например. А вот в колонке с желтым фоном будет написано: млекопитающее, млекопитающее. А потом будет написано там лягушка, ящерица, будет написано земноводное, там пресмыкающееся. Вот. И будут описаны их свойства вот здесь вот. И будет много примеров приведено. И вот когда очень большой список этих животных, ну, допустим, там 200-300 строчек, то не очень понятно, а чем отличаются млекопитающие от земноводных, от пресмыкающихся? Какие у них особенности наиболее характерные? Ну я могу вам сказать, что в науке эта задача решена. Ламарк классифицировал животных там 300 лет назад, сейчас точно не помню когда, ну примерно. И предложил их разделять вот на такие категории. Вот. Но дело в том, что сейчас это может делать и программная система. Значит, мы можем посмотреть, как это выглядит. Вот. И будут созданы обобщенные образы млекопитающих. И самым характерным признаком млекопитающих окажется, знаете, какой признак? Что они выкармливают детенышей молоком. Потому что все остальные признаки, они встречаются и у других видов животных. А этот только у млекопитающих. Система это определит и покажет нам на основе вот этих конкретных определений создаст обобщенное определение. Что представляет собой описание объекта? С одной стороны его свойствами, признаками, а с другой стороны принадлежностью к обобщающей категории. А это определение. Только это определение не общее такое теоретическое, а это определение конкретного объекта. Вот, допустим, данный конкретный студент – это учащаяся вуза, конкретно Кубанского госуниверситета, группы БИ 1701. А здесь можно фамилию написать. Вот такие-то у него характеристики личностные. Можно психологический его портрет описать здесь. А можно чисто имиджевый, внешний вид его описать, как он выглядит. Вот. А здесь написать, какой группе он принадлежит и является студентом или студенткой. Вот. И система создаст обобщенный образ студентов этой группы и других групп и покажет, чем студенты группы БИ 1701 отличаются от студентов других групп. Выявит наиболее характерные их особенности, которыми они отличаются. Ну то есть она, то есть система позволяет решать задачи обобщения. То есть на основе примеров она формирует обобщенные образы, которые вот в этих колоночках вот желтеньких мы эти образы формируем. А объекты описываются своими свойствами, которые в колоночках зелеными, зеленым фоном.

Вот. Ну, как вы видели, там определенный диапазон колонок – классификационные шкалы и определенный диапазон колонок – описательные шкалы. Вот этот диапазон, его мы здесь и указываем. Видите, вот мы указываем, что тип Excel старый, Excel до 2003 года. Диапазон начальная классификационная шкала вторая, конечная классификационная шкала третья. А вот описательные шкалы идут с четвертой по двенадцатую.

Теперь, что я еще хочу сказать про эти шкалы? Что шкалы бывают трех типов. Значит, ну вообще-то, грубо говоря, двух типов – это текстовые и числовые. Но текстовые делятся еще на два подвида. Это номинальные шкалы и порядковые. В номинальных шкалах только отношение тождества или не тождества, эквивалентности, не эквивалентности. Ну, допустим, вот список студентов, не рассортированный. И студент каждый соответствует какой-то строчке или не соответствует. И все. И больше там ничего нет в этом списке. Второй вид текстовой шкалы – это порядковая шкала. Там кроме эквивалентности, не эквивалентности, есть еще порядок больше-меньше. Допустим, мы возьмем и всех студентов рассортируем по алфавиту фамилии. Тогда те, кто на букву А будут в начале списка, те, кто там на букву Ч – в конце списка. Вот. А можем рассортировать их не по фамилии, а по успеваемости или по весу или еще там как-то. Понимаете? Это называется порядковая шкала, когда есть отношение больше-меньше. А если там еще указана единица измерения и начало отсчета, тогда это числовая шкала. Тогда значения на этой шкале можно складывать, вычитать. Ну, допустим, на порядковой шкале мы говорим так: человек пожилой старше, чем молодой человек. А насколько старше, мы не можем сказать. Может он на 3 года старше, а может на 10 лет старше. А вот если мы возьмем числовую шкалу, там прямо будет написано, сколько ему лет. И мы можем вычесть из возраста пожилого человека, можем вычесть возраст молодого и сказать: он старше на 15 лет. То есть операции арифметические можно выполнять на числовой шкале. Система, которую мы будем изучать в лабораторной работе, система Aidos, работает со всеми видами шкал.

Значит, сейчас здесь ничего не нужно делать, потому что все параметры установлены по умолчанию. Просто я объяснил их смысл. Еще здесь тут такие опции, что если числовая шкала, то можно делить ее на некоторое число числовых диапазонов. Эти диапазоны могут быть либо одинаковые по размеру с разным числом наблюдений, либо они могут быть разными по величине, но с одинаковым числом наблюдений. И вот здесь вот тоже есть переключатель. Ну, более те, которые ниже, я не буду сейчас объяснять, потом расскажу. Значит, мы можем вот это первый вариант – формализация предметной области. Это создание справочников классов и признаков и кодирование исходных данных с помощью этих справочников и получение обучающей выборки. А это вот использование уже существующих справочников для того, чтобы закодировать распознаваемую выборку. Все эти параметры установлены по умолчанию в файле параметров, которые я, значит, копирую вместе с самим файлом исходных данных.

Теперь мы просто нажимаем Окей. Вот. И происходит тут всякие сообщения выводятся, которые означают вот что, что этот файл исходных данных преобразован в стандарт системы Aidos, проанализирован по этим диапазонам, которые мы указали. И найдено, что у нас есть две текстовых классификационных шкалы, описывающих обобщающие категории. И есть у нас одна числовая шкала описательная, и восемь есть шкал текстовых описательных. И в описательной числовой шкале предлагается 12 числовых диапазонов. Почему предлагается такое количество? Потому что так задано в этом вот файле параметров. Мы можем поменять это количество, тогда надо будет здесь вот пересчитать градации, шкалы и градации. Сейчас у нас будут построены модели. Смотрите, суммарное количество градаций классификационных шкал – 14, описательных шкал – 50. Выходим на создание модели. Это размерность матрицы.

Что произошло? Произошло, что система создала, смотрите, пункт первый из трех: сформированы классификационные и описательные шкалы и градации на основе базы данных indata. Сгенерирована обучающая выборка, то есть исходные данные закодированы с помощью этих справочников классификационных и описательных шкал и градаций. Вот. И потом все это переиндексировано.

Вот. И вот это сообщение для студентов. Студенты обычно никогда его не читают, что здесь написано. Вот из моего опыта я знаю, когда вот в аудиториях были занятия, обычно студенты, когда видят это вот окошко, они просто сразу же нажимают Окей и потом на меня смотрят умным взглядом и думают, и спрашивают: "А дальше что делать?" Я говорю: "А зачем вы нажали Окей?" Ну там как раз написано, что делать дальше. Вот пункт первый, что дальше нужно сделать? Нужно открыть файл исходных данных. Файл исходных данных всегда находится в папочке indata по пути, который здесь указан. Это предназначенная для этих целей папочка. Смотрим на эту папочку. Вот эта папочка, где система находится, indata, там все данные, indata. И там у нас файл исходных данных экселевский. Открываем этот файл. Что мы видим, ребята? Мы видим диапазон классификационных шкал, диапазон описательных шкал и наименование объектов обучающей выборки. Какая задача перед стоит перед системой интеллектуальной сейчас? Она должна создать обобщенный образ, который я называю конкретный класс и обобщающий класс. И тут у нас есть в конкретных классах у нас мышки есть, клавиатура, сумки, монитор, стул, вешалка, телефон, мяч. А в обобщающих классах у нас поменьше: элемент компьютера, аксессуар, мебель и средства связи, спорт-инвентарь. То есть ясно, что конкретные классы входят в обобщающие. Обобщающие более общий характер имеют. И свойства этих объектов: цвет и значение, материал и значение, размер и значение размера. И размер два. И здесь вот указано, видите, в виде числа размер, а здесь в виде текста. Наличие экрана, кнопок, проводов, форма и наличие ножек. Теперь смотрите, я ставлю клеточку А1 курсор. Вам видно, что я в клеточку А1 поставил, нет? Ребята? Или надо крупнее показывать?

Когда я вас спрашиваю, надо отвечать, ребята. Вот я поставил А1. Нажал Ctrl и Shift клавиши и нажал клавишу End. Что произошло? Выделилась область значений, видите, она блоком выделилась. Эта область значений точно соответствует фактически заполненным строкам и колоночкам. А иногда бывает, что эта область значений вот так получается. Вот. Вот так вот, например. То есть выходит за пределы фактически, фактических заполненных строк и колонок, понимаете? Вот если такой файл дать системе Aidos, она начнет ругаться, что вот у вас в колоночке М нет наименования, в колоночке N нет наименования. А колонку G она посчитает, что это текстовая, потому что здесь есть не числа. То есть я вам советую, что делать? Когда вы делаете файл исходных данных, вот так Shift+Ctrl и End выбираете или просто берете курсором. Ну если так вот, как я сейчас показал, получается, то значит, тогда ничего делать не надо. А если там вот шире область, то нужно выделить реальную область вот так, скопировать в буфер обмена и создать лист и туда вставить. Вот так надо делать тогда. Иначе система будет работать с более широким диапазоном строк и колонок, чем те, где есть данные. Это нежелательный вариант. Она будет работать, но будет выдавать разные сообщения. И вы не будете понимать, что они означают. Хотя там будет написано.

Значит, теперь смотрите, ребята. Значит, в Экселе может быть много листов. Всегда система работает с тем листом, с которым вы работали последним, когда закрывали Excel. Ну сейчас мы можем его закрыть. Значит, да, еще вот что. Значит, смотрите, если поставить клеточку А1 курсор и держать Shift и двигать вправо, то здесь вверху слева, видите, формула есть: 2С – вторая колонка, третья колонка, четвертая и двенадцатая. То есть я хочу сказать, что можно не считать вот так вот глазами там или еще другим способом каким-то, а можно просто блоком выделить и увидеть, Excel сам посчитает. Ну если их очень много, тогда можно сделать в отдельном листе строчку, где мы посчитаем эти номера этих колонок и зададим их для системы Aidos.

Вот. Дальше. Файл исходных данных мы посмотрели. Теперь предлагается нам прочитать описание по данной лабораторной работе в режиме 5.14. Значит, ну, если честно, вот эта работа как раз там и не описана. Есть, там почти все работы описаны, это разделы в учебных пособиях, но это не описано. Это я объясняю вот так подробно все для того, чтобы первый раз, когда студенты изучают систему и лабораторную работу первую, чтобы они базовые понятия усвоили. А потом там проще дальше. Дальше они там клацают так, что не успеваешь понять, что они там делают. Потом рекомендуется выполнить пункт третий: режимы 2.1, 2.2, 2.3, 3.5 и другие. Окей. Появляется в диспетчере приложений, появляется приложение, желтеньким выделено, потому что оно текущее. Если мы создадим много приложений, ребят, в системе можно их создавать неограниченное количество. Ну, понятно, что каждое из них занимает память, и желательно ограничиться каким-то разумным количеством. Вот. Ну можно какие-то разные варианты какого-то приложения здесь создавать и там писать название, в чем особенность. Ну, допустим, здесь можно написать 12 равных интервалов. Вот, а ниже, допустим, написать 10, допустим, адаптивных интервалов. И вот здесь мы можем написать любую буковку, в частности W, и это приложение будет текущим.

Теперь смотрим то, что нам было рекомендовано: режим 2.1. Сейчас уже установлено приложение. Да, ребят, я вам сейчас говорил, что есть много программных интерфейсов. Вот в режиме 1.3 мы устанавливаем именно учебные приложения, которые уже встроены либо находятся на FTP сервере. Для чего? Для того, чтобы учиться, понимаете, смотреть, как это делается вообще, в принципе. А если вы хотите сами сделать приложение, тогда режим 2.3.2 используется. А здесь много есть разных интерфейсов: с текстовыми файлами, вот этот, который сейчас мы использовали 2.3.2.2. Потом есть и другой 2.3.2.3, тоже экселевские файлы, но другой стандарт. Оцифровка изображений по внешним контурам, оцифровка изображений по пикселям и спектру. Понимаете? Также есть возможность исследовать динамику в числовых и символьных рядах, сценарный системно-когнитивный анализ. И в этом интерфейсе 2.3.2.2 тоже там есть режим, позволяющий это делать. То есть исследовать сценарии. Как влияла динамика предшествующая на динамику последующую? Причем, если вот здесь внутри мы можем это только в одном, в одной последовательности исследовать чисел или символов каких-то букв, например, да? То в режиме 2.3.2.2 мы это можем делать для нескольких колонок и обнаруживать их взаимосвязи между ними. Как сценарий изменения, допустим, числа банков влияет на сценарий изменения курса. То есть прошлые сценарии влияют на будущее.

Значит, сейчас мы смотрим 2.1. Что здесь? Да, кстати, когда я двигаю так вот по пунктам меню, то вы видите, там внизу появляется окошко, в котором объясняется, что этот пункт меню делает, какие функции выполняет. Так, теперь, значит, мы там блымкаете, поэтому я посмотрю, что вы там блымкаете. Ну, в чате ничего нет. Ответьте мне микрофоном, как вы понимаете, что я вам рассказываю, слышите, что я рассказываю? Улавливаете?
Слышим, но с трудом понимаем, потому что у большинства система не запускается. Ну, как бы...
Так, если она не запускается, ребят, то у меня какая рекомендация? Надо смотреть настройки в операционной системе. Вот. Причем, вы можете запускать не start\_aidos файл, а aidos\_X файл запускать. Файл aidos\_X – это исполнимый модуль системы. То есть когда вы в папочку заходите в систему, сортируете по алфавиту, то вот не start\_aidos запускаете, а вот прямо файл aidos\_X запускаете. Может быть, запустится. А вообще, вот надо вот эти настройки сделать, которые я вам сейчас показал, послал в чат.
Вот на нём как раз-таки у меня получается, он просит авторизацию, а потом выдаёт ошибку, как стандартная в коде ошибка бывает.
Ошибка исполнения.
Да. А подскажите, пожалуйста, она ещё на виртуальной машине, то есть если я сейчас семёрку себе поставлю, виртуальную, она будет работать?
Будет работать безупречно.
Хорошо, отлично.
Только единственное, VMware Player, он же тяжёлый, то есть он там еле шевелиться будет. Но будет работать. А на десятке ж можно, вот я ж там послал вам, эмулировать семёрку можно там, режим совместимости. Все вот эти пункты выполнить настроек. Обычно работает.
Да, я сейчас их и выполняю.
Угу. Если что-то не совсем понятно, то можно в интернете проконсультироваться, как вот экран снять, отменить для какого-то конкретного приложения защиту. Путь на систему должен быть досовским чисто или юниксовским точнее, то есть без пробелов и кириллицы.

Ну, я могу вам сказать, ребята, что программное обеспечение было разработано до создания восьмерки и десятки Windows. И поэтому возникают некоторые такие вот вопросы. Ну вообще десятка очень капризная в плане выполнения программ, разработанных ранее и не фирмой Microsoft. Вы, наверное, про это знаете. Ну вот приходится, так сказать, изворачиваться.

Значит, ребята, вот здесь вот мы сейчас зашли в режим 2.1 и видим вот здесь вот слева название классификационных шкал. Оно взято полностью из экселевского файла, который я показывал. Из этого вытекает такой вывод, что желательно эти названия в экселевском файле писать не очень длинные. Я советую сокращать там, но так, чтобы было понятно, что это значит. И вот мы видим, смотрите, когда мы выбираем какую-то шкалу классификационную, мы видим ее градации, видите? Эти градации текстовые, они уникальные. Значит, то есть если много раз повторяется какое-то наименование, то здесь будет один раз. С чем это связано? У вас же была дисциплина Базы данных, да, ребята?
Да.
Была. Значит, вы знаете, что такое нормализация. Если вы какое-то наименование где-то в форме какой-то несколько раз встречается, то это означает, что у нас, что нужно делать справочник, в который включать эти наименования и делать ссылки с этого справочника на этот файл, где были наименования. И тогда у вас это наименование будет храниться в базах данных в одном единственном месте. Вот. А все остальные места, где оно используется, будут просто ссылаться на это единственное место по отношению один ко многим. Называется первая нормальная форма. Слышали, да, про это? Так вот система Aidos, она когда берет файл исходных данных, она сразу же его анализирует и разрабатывает справочники для нормализации файла исходных данных. Вот эти справочники: конкретный класс, обобщающий класс – это те же самые колонки, только здесь их значения текстовые, они уже здесь являются уникальными.

И потом, сейчас я вам покажу обучающую выборку, сделано отношение вот справочников классов и строчек соответствующих обучающей выборки. То есть она кодируется с использованием этих справочников. Теперь 2.2 смотрим. Это здесь кодируется тот диапазон колонок, который описывал свойства объектов этих наблюдений. Здесь у нас все шкалы тоже являются текстовыми, кроме шкалы два, размер два. Здесь у нас 12 диапазонов, минимальный 1/12, максимальный 2/12. В этом эта шкала, как она закодирована? Поскольку мы там выбрали равные диапазоны, то было найдено минимальное и максимальное значение, и этот диапазон был разделен, изменение величины этой шкалы, диапазон изменения значений этой шкалы градации этой шкалы, был разделен на 12 равных диапазонов числовых. И вот эти диапазоны здесь мы видим. Внутренняя точность расчетов – семь знаков в системе. Вот. Размер один. По смыслу, конечно, это у нас порядковая шкала. Но здесь мы когда посмотрим на то, что здесь у нас написано, то видно, что она, конечно, здесь номинальная. Почему? Во-первых, здесь у нас градация большой, большая. Зачем в размере указывать род мужской или женский, непонятно. То есть здесь это было излишне делать. То есть в экселевском файле нужно, видимо, большая заменить на большой. Тогда здесь у нас получится уже не пять градаций будет, а четыре. Вот. Потом, чтобы они были расположены в нужном смысловом порядке от минимального к максимальному, для этого нужно писать: маленький, да, чтобы он был на первой позиции, этот эта градация, нужно писать: 1/4-маленький. Потом 2/4 средний, 3/4 под руку и потом 4/4-большой. Тогда это будет порядковая шкала. Ну это надо делать в экселевском файле, тогда вот так будет сформировано, как я сказал.

Теперь, у нас здесь есть такие шкалы, где бинарные их называют или логические, где мы отвечаем на какой-то вопрос: да, нет. Вот наличие экрана: есть, нет. Наличие кнопок: есть, нет. Наличие проводов: есть, нет. Так вот я здесь хотел вам сказать, что здесь есть такая тонкость, что нужно не только писать, когда есть провода что-то, но и когда их нет. Когда их нет – это такая же точная информация, как и та, когда они есть. То есть эта информация полезна для идентификации этих объектов, для обобщения, решения задач.

Смотрим теперь на саму обучающую выборку. Это можно сделать двумя способами. Вот здесь вот трехоконный интерфейс. В верхнем окне мы видим наименование объектов из первой колонки, объектов обучающей выборки. Эти наименования могут быть довольно длинные, до 130 символов, по-моему, или даже еще больше. Вот, сейчас я не помню уже. Но я их делал длинными. Там можно прямо писать там, что там, откуда, когда, что, где, когда это получено, эта информация, скажем так. Вот здесь вот в левом нижнем окошке, здесь вот коды классов, к которым этот объект относится. Здесь видите, он относится к двум классам. Один класс – это конкретный, а другой – обобщающий. А вот здесь у нас коды свойств, то есть значения свойств или значения факторов.

Значит, все это три таблицы, каждая из которых может иметь размер до 2 ГБ, потому что в этом языке программирования, которое я использовал, у них размеры таблиц, баз данных не больше 2 ГБ. Но, если взять 2 ГБ, то это довольно-таки большой объем данных. Иногда изредка бывает так, что не хватает этих 2 ГБ. Но это бывает очень редко. Есть способ с этим бороться, тогда я там кое-что придумал для этого. Вот. Но это бывает очень редко. Обычно хватает. Это что означает? Что вот этих вот кодах признаков здесь может быть очень много. И вот здесь вот будет прокрутка. То есть нет никакого принципиального ограничения на их количество, кроме размеров вот баз данных и диска. То же самое касается классов и то же самое касается числа объектов обучающей выборки. Я решал задачи, где сотни тысяч объектов обучающей выборки. В принципе, где-то до 700.000, по-моему, я пробовал решать. Ну, вообще-то их может быть до миллиона в экселевском файле. Вот. А классификационных шкал может быть до 16.000 в экселевском файле. Поэтому, соответственно, суммарное количество классов, кодов классов и кодов признаков, в принципе, может быть до 16.000. Но фактически классов не может быть больше полутора тысяч. Ну это тоже довольно много, на самом деле. Это связано с другими там обстоятельствами. Ну, в общем, классов до полутора тысяч, а вот признаков вот этих, их может быть и десятки тысяч, и сотни тысяч даже.

Вот. Есть другая форма, где мы видим обучающую выборку – 2.4 режим. Значит, что здесь интересного? Файл, который мы здесь, то есть экранная форма, которая отображает файл исходных данных, она устроена абсолютно так же, как файл исходных данных экселевский, который мы ввели в систему. Сейчас я попробую их одновременно показать.

Вот. И будет видно. Смотрите. Вот сейчас я беру прямо и вот это колоночки наименований, колоночка классификационных шкал и колоночка описательных шкал, колоночки. Вот. И вот мы видим, что они совпадают по структуре. Только здесь в экселевском файле мы видим тексты и числа, а уже в системе Aidos мы видим коды, коды этих текстов и чисел в соответствии со справочниками классификационных и описательных шкал и градаций. Вот, допустим, мышка, мышка, мышка, а мы здесь видим 4 4 4. А здесь элемент компьютера, элемент компьютера, мы видим, видите, 14 14 14. Ну и также вот все остальные свойства, это классы, коды классов, а это коды признаков. Они все закодированы в соответствии со справочником описательных шкал и градаций.

Теперь я хотел бы вам сказать, что есть две интерпретации. Да, в Экселе здесь вот номера строк есть, а здесь есть тоже номера строк. Если мы посмотрим на нижнюю шапочку вот эту, здесь мы видим от нуля до 12 пронумеровано. Причем классификационные шкалы вторая и третья, а описательная с четвертой по двенадцатую. Что точно соответствует тому, что и есть вот в этом экселевском файле. Значит, э-э... Что-то хотел вам еще сказать про эти шкалы, но что-то вылетело, что я хотел сказать. Ну ладно, дойдем до этого, скажу. Вот. Ну вот таким образом, э-э, вы поняли, что этот экселевский файл введен в систему. Мы это видели. То есть созданы справочники классификационных и описательных шкал градаций и обучающая выборка. Это означает, что создано все необходимое для того, чтобы разработать модели. А у нас на лекциях я вам рассказывал о преобразовании данных в информацию, а ее в знания. Вот сейчас вот скажите, кто помнит, что я рассказывал про преобразование данных в информацию, а ее в знания? Про концепцию смысла Шенка-Абельсона, про то, что нужно осмысливать данные, выявлять причинно-следственные зависимости в них, а точнее в реальной предметной области, описанной этими данными. Про меры причинно-следственных связей. Было дело такое, ребята? Вспоминаете? Я на одном из занятий должен был это все вам рассказывать. И, наверное, рассказывал, я так думаю. Сейчас посмотрим.

Даже скажу, на какой лекции это было. Вот. Это у нас было на первом занятии лекционном, ребята. Данные, информация, знания, определение, содержание понятий. Вот. То есть это было первое занятие, очень важное, понятийное. Я там объяснял вам, чем отличаются базы данных от информационных баз, от баз знаний. Это вот эти разделы. Глава один и пункты этой главы. Кстати, этот чат не очень хороший, он не понимает конца строки. Абзацы не сохраняет и не позволяет изображения записывать, передавать, что, в общем-то, не очень хорошо.

**Возможности системы и теоретические основы**

Значит, теперь мы выполняем следующий этап – преобразование данных в информацию. Мы обнаруживаем, ну не мы, а, вернее так, мы с помощью системы обнаруживаем причинно-следственные связи между наличием у объектов этих наблюдений определенных свойств, признаков и их принадлежностью к определенным категориям. Мы это делаем в режиме 3.5. В режиме 3.5 появляется такая экранная форма, которую тоже надо пояснить. Вам видна эта экранная форма, ребята? Ну видна должна быть. Просто вы стеснительные, ничего там не говорите, молчите.

Значит, смотрите, ребята. Значит, здесь в этом режиме осуществляется две важных операции. Это синтез моделей, то есть их расчет. И вторая операция – проверка этих моделей на достоверность, оценка их достоверности. Что это за модели? Это три модели статистических: модель абсолютных частот, условных и безусловных процентных распределений – две модели. И семь моделей, я их называю системно-когнитивные модели или модели знаний, посчитанные с разными частными критериями, мерами знаний, которыми являются количество информации по Харкевичу, хи-квадрат, Return on Investment, то есть коэффициент возврата инвестиций. И также критерии взаимосвязи, которые в разных областях науки эти вот коэффициенты, частные критерии были открыты. Но я их собрал в системе как критерии силы и направления причинно-следственных связей.

Теперь, следующий момент, который надо ваше внимание обратить. Здесь есть внизу справа есть возможность посчитать на графическом процессоре, а есть на центральном процессоре возможность. Сейчас давайте на центральном процессоре будем считать. Почему? Потому что на графическом процессоре это режим новый, и в нем есть одна неточность при расчете одной из моделей. ПРЦ-2 модель рассчитывается не совсем правильно. Хотя эта ошибка, которая там есть, прямо это вот ошибка, она не влияет на достоверность этой модели, но показывать, объяснять ее неудобно с этой ошибкой. Вот. Режим с графическим процессосом работает от 200 до 4.000 раз быстрее, чем на центральном процессоре. Ну на такой малюсенькой задачке, ну будет там не 1/100 секунды, а 1 там тысячная секунды, там, допустим. Какая разница? Вы все равно не заметите разницы особой. А вот если большая задача, которая считается сутки, например, на графическом процессоре, на центральном она может считаться 10 лет, например. Ну тогда вы заметите сразу разницу.

Значит, но когда не было режима работы на графическом процессоре, то тогда я придумал, как можно выкрутиться в ситуации, когда очень много объектов обучающей выборки. Допустим, у нас 700.000 объектов обучающей выборки, 400 классов, 11.000 признаков. То есть такие большой размерности задачи. Если мы начнем на 700.000 объектах проверять достоверность модели... Значит, что это, как это делается? Каждый объект, ну сначала цикл по объектам распознаваемой выборки, которая будет такая же, как обучающая в простейшем варианте. Потом каждый объект сравнивается обучающей выборки, сравнивается со всеми классами, а их 400. А чтобы сравнить, вычисляется сумма из 11.000 признаков, количество информации, содержащейся в 11.000 признаков о том, принадлежит, какой степени он этому классу принадлежит, какой степени он на него похож. Ну представляете себе, какой объем вычислений огромный? Это уже трудная задача для компьютера. И он ее обычно туговато выполняет. И для этого я сделал такое вот разделение. Значит, можно только синтез модели осуществить. Даже на больших моделях, большой размерности, большом числе объектов выборки, синтез модели достаточно быстро происходит. Ну, скажем, будет там не секунда, там, не минута, ну будет 15 минут там или 20 будет минут. Все равно это такое, ну, приемлемое время. А вот верификация модели путем распознавания всех объектов обучающей выборки, скопированных в распознаваемую выборку, это будет уже затягиваться. И вот этот режим, если мы запустим синтез и верификацию, он может разумно, это может быть разумным, но небольших моделях. Если модели большой размерности, с большим числом объектов, тогда лучше сначала синтез осуществить модели, а потом верификацию на каком-то подмножестве. Ну, например, можно выбрать 100 объектов, например, случайным образом. И с их помощью произвести оценку достоверности модели. Тогда у нас получится, скажем, что-то займет 7 минут, например. Отсюда мы можем посчитать, что если мы на это дело можем выделить там 7 часов, то пропорцию составим, что вот 7 минут – 100 объектов, а, значит, там 8 часов там X объектов. И мы поймем, что можно сделать, скажем, 200.000 объектов, к примеру. Тогда мы здесь пишем 200.000, запускаем верификацию, идем спать. Приходим утром, уже модель просчитана, оценена достоверность модели на максимальном числе объектов.

Теперь, мы можем объекты, на которых проверялась достоверность модели, удалять из обучающей выборки. А можем и не удалять. Значит, если удалять, тогда получается так, что мы одну часть обучающей выборки используем для синтеза модели, а оставшуюся используем для проверки ее достоверности. Этот подход называется Bootstrapный, Bootstrap. Bootstrap – это на, сейчас честно сказать, не помню на каком, но, по-моему, на шведском языке. В общем, на одном из скандинавских языков – это означает складной ножик. То есть у него много лезвий, вот мы можем их вытаскивать эти лезвия из ножика. И одно лезвие – это у нас, скажем, объекты обучающей выборки, а другое лезвие – это объекты распознаваемой выборки. Вот. Зачем удаляют? Ну для того, чтобы посчитать внешнюю достоверность модели, внешнюю валидность. Слово "валид" означает правильный, а "инвалид" означает неправильный. Вот. Ну слово "инвалид" есть в русском языке, а "валид" мы не знаем, в русском языке это слово не редко используется. Но оно иностранного происхождения. Когда мы говорим о том, что что-то нормальное, то мы говорим: это валидно. То есть валидность модели высокая. То есть есть внутренняя валидность и внешняя. Если мы не удаляем объекты из обучающей выборки, на основе которых мы оцениваем достоверность, то это называется внутренняя валидность. А если удаляем, то это внешняя валидность. Вот. Внешняя валидность всегда немножко поменьше. И модель-то для чего делается? Не для того, чтобы саму обучающую выборку исследовать обычно, а для того, чтобы создать модель, которая позволяет идентифицировать и изучать объекты, относящиеся к некоторой генеральной совокупности, по отношению к которой обучающая выборка репрезентативна. Что это значит, если перевести на русский язык? В обучающей выборке есть определенные причинно-следственные связи между признаками и принадлежностью объектов с этими признаками тем или иным классам, обобщающим категориям. И эти же закономерности действуют в более широкой совокупности какой-то, чем сама обучающая выборка. Вот эта более широкая совокупность, где действуют те же самые закономерности, называется генеральной совокупностью. А обучающая выборка называется репрезентативной по отношению к ней. У генеральной совокупности есть границы и в пространстве, и во времени. То есть в пространстве, например, мы взяли, сделали модель на основе, допустим, информации о вас, как студентах. В России так еще более-менее это будет похоже и на другие университеты, и другие группы, но уже в меньшей степени, чем в вашей группе будет проявляться эти закономерности. А если вы поедем куда-нибудь во Францию или в США, то там эта модель работать не будет вообще, потому что там вообще другие зависимости совершенно между признаками и принадлежностью объектов к категориям. И там надо модель пересоздавать. Это это границы генеральной совокупности в пространстве. Во времени. Взяли мы вот сейчас провели исследование, сделали сейчас модель, вот сейчас прямо. А потом стали ее использовать в следующем году, через год, через два, через три. И у нас мы заметили закономерность, что постепенно модель теряет достоверность. Сначала она работала неплохо, а потом прямо стало уже похуже работать, и потом просто перестала нас устраивать уже, как она работает. С чем это может быть связано? С тем, что существует естественная динамика предметной области, все изменяется. Вот эти закономерности, взаимосвязи между признаками и принадлежностью объектов классам тоже изменяются, и достоверность модели падает. То есть это означает, что генеральная совокупность ограничена во времени. И есть точки во времени, когда изменяются сами закономерности предметной области, эволюционным образом, скачкообразно, качественно изменяются. Сразу же эта модель станет непригодной для практического применения.

Вот. Сейчас мы, да, если очень много объектов обучающей выборки, то тогда без разницы, удаляем мы или не удаляем эти объекты из нее, когда проверяем достоверность модели. По какой причине? Ну представьте себе, если там у нас 700.000 примеров объектов, да, в этой обучающей выборке. И мы какой-то один из них удалим или не удалим. А там по каждому классу там по 20, по 30.000 примеров приводятся. Это означает, что если мы его удалим, то ничего особо не изменится, понимаете? Потому что вклад каждого объекта в формирование этих обобщенных образов ничтожно мала. Это тысячные доли процента там влияет на величины, количество информации там и так далее. Поэтому я считаю, что когда выборка большая, то никакой разницы нет, удалять или не удалять. Это без разницы.

Теперь смотрите, здесь вот внизу написано интересная вещь. Что база РДБФ будет иметь размер 14.200 байт, что составляет 60% от максимально возможного, то есть от 2 ГБ. То есть это что означает? Что если мы вот сейчас будем эту модель рассчитывать маленькую, то мы можем рассчитывать модель в системе Aidos, которая в десятки тысяч раз больше по размерности, потому что эта модель составляет 60%. От того, что может быть обработано в системе Aidos. Но если у нас здесь получится вот, допустим, не доли, а, допустим, 100% будет написано или даже будет написано 120%, это будет сообщение выдано, что система не может обработать этот объем информации и создать модель с такой размерностью, с таким числом объектов обучающей выборки. И будет рекомендовано удалить какую-то часть этих данных, наиболее плохо идентифицированных. Мы здесь можем написать, что если, допустим, у нас модель составляет 120% от возможного, ну то есть превосходит возможное, возможности системы, то мы можем оставить 5% наиболее значимых результатов. И тогда все дальше нормально пройдет, получится хорошая модель, а из результатов идентификации, распознавания будут убраны только те, которые наиболее сомнительные, с самым низким уровнем сходства. Ну то есть я сейчас вам про эту экранную форму все рассказал практически. Только не рассказал, как какие частные критерии используются для расчета этих моделей. Но это я рассказывал на первом занятии.

Нажмем теперь Окей. И потом еще покажу. Нажимаем Окей. Начинает там мелькать все на экране, потому что маленькая размерность модели, все очень быстро считается. Два процесса идет. Один процесс уже прошел – это процесс синтеза моделей. А сейчас идет, прошел уже процесс проверки их на достоверность. При этом, когда модели проверяются на достоверность, это делается следующим образом: обучающая выборка копируется в распознаваемую, распознается системой с использованием моделей всех по очереди, а их там 10. И определяется, сколько каких ошибок возникло. Система должна правильно относить объекты к тем классам, к которым они относятся – это True Positive называется, истинное положительное решение. Правильно не относить к тем, которым они в действительности не относятся на самом деле – это True Negative, уровень сходства отрицательный, но решение истинное, то есть объект не относится к классу. И система может ошибаться и в первом, и во втором случае. Если система ошибается при отнесении объекта к классу, то это называется ложное срабатывание или называется ложноположительное решение, False Positive. То есть система относит объект к классу, которому он на самом деле не относится. И то же самое есть ошибочное решение отрицательное, когда система не относит объект к классу, уровень сходства отрицательный, а фактически он к нему на самом деле относится. Это называется False Negative. Система подсчитывает все эти варианты и определяет достоверность модели.

Вот это все у нас произошло за 16 секунд. Нажимаем Окей. Теперь мы можем посмотреть, что ж там за модели у нас сформированы. Для этого мы используем режим 5.5.

Вот. Сейчас мы видим, имеем возможность выбрать эти модели. Значит, здесь вот у нас описано, каким образом эти модели рассчитываются. Ну здесь написано про системно-когнитивные, потому что остальные они очень примитивно рассчитываются, я объясню, скажу вам, как это делается. Ну когда будем их рассматривать, я это покажу еще. Самая простая модель – это модель абсолютных частот. Такая модель рассчитывается во всех системах статистических, вообще какие только существуют. В чем смысл этой модели? Значит, вертикальные, ну, есть некая матрица. Размерность матрицы: колоночки, число колонок – это число классов. То есть это суммарное число градаций всех классификационных шкал, всех типов – и текстовых, и числовых. Если классификационная шкала числовая, то классом будет являться числовой диапазон определенный. Вот. А строчки, вы видите, здесь у нас указано свойство и значение этого свойства. То есть это описательные шкалы и градации. А что у нас в колоночках? Причем вот эти шкалы и градации, вот видно здесь вот они номинальные, порядковые, но неупорядоченные, как положено. А вот это числовая, числовая шкала. Да, вот что я хотел еще сказать. Когда мы заходим, смотрим обучающую выборку в режиме 2.4 вот здесь, то мы уже не видим никакого различия между текстовыми шкалами и числовыми. Видите, то есть вот размер один, например, вот размер два. Уже не догадаешься, где там числовая шкала, а где порядковая, а где номинальная. Почему? А там только коды уже, коды градаций текстовых и числовых диапазонов. И дальше уже, после формирования обучающей выборки, система совершенно одинаково работает и с текстовыми шкалами, и с числовыми.

Вот. Теперь смотрим на сами модели. Вот здесь вот у нас классы слева – конкретные, а справа – классы обобщающие. В обобщающих классах там частоты побольше. То есть больше, что это за числа вообще? Это числа показывают, сколько раз встретился определенное значение свойства у объектов определенной категории, определенного класса. Классу соответствует колоночка. И сколько раз встретилось это свойство вообще у объектов обучающей выборки. Вот, допустим, размер под руку. Вот три раза встретился у средств связи и три раза встретился у компьютеров. Видите, вот три раза этот признак встретился под руку. Вот. И там, и там. А по всей выборке он встретился 12 раз. Вот, видите, всего у всех объектов обучающей выборки. Вопрос возникает такой: этот признак одинаково характерен для средств связи и для элементов компьютера или не одинаково? Чтобы на этот вопрос ответить, нам нужно опуститься вниз таблички и посмотреть, что средств связи у нас было три в обучающей выборке, а элементов компьютера – семь. Это что означает? Что 100% средств связи этим признаком обладают, и примерно половина элементов компьютера этим признаком обладают. Из этого можно сделать такой вывод, что надо сравнивать числа не для того, чтобы решать задачи, надо использовать не матрицу абсолютных частот, а хотя бы матрицу условных и безусловных процентных распределений.

Еще такой момент ваше внимание обращаю. Каждый объект обучающей выборки относится к двум классам. То есть он встречается где-то в конкретных классах, дает свой вклад, и где-то в обобщающих классах он встречается и тоже дает свой вклад. То есть у него есть два кода принадлежности к классу. Вот берем, допустим, клавиатура, видите, 2 14. Берем мышку, 4 14. Понимаете? То есть он относится к двум классам объект, каждый. А всего их 20, кстати. Вот посмотрите. Вот, видите, 20 всего примеров различных предметов, которые были на столе, когда мы эту задачу со студентами делали. Да, ребята, еще вот что я хотел попросить. Когда будет приближаться конец занятия... А это будет произойдет в 18:50 конец занятия. В 18:45 вы мне скажите, хорошо? Что уже дело идет к концу занятия. То есть через 10 минут.
Хорошо.
Кстати, насчёт программы помогла простая установка на другой диск. Ну я создал отдельный виртуальный образ диска и на него всё поставилось.
А почему, как вы думаете?
А не знаю, он чистый, новый.
Может быть, у вас там путь на систему был какой-то с пробелом, кириллицей?
Нет, без кириллицы, без пробелов, там корневая папка, сразу следующая папка и всё.
В общем, напишите мне письмо на почту. Сейчас я почту приведу. Назовём её новость. [prof.lucenko@gmail.com](https://www.google.com/url?sa=E&q=mailto%3Aprof.lucenko%40gmail.com). То есть я напишу там, что попробуйте другой диск. А вы его создали этот диск, да? Просто этот...
Да, я просто выделил место на диске C, отделил его в диск D и всё, и на диск D поставил систему.
Partition Magic, да, или что-нибудь такое, да?
Ну да.
Ну понятно. Ну вот этого я не знал, что такое может быть. Странно, я скажу так, между нами. Ну ладно, чего только не бывает, да, в этом мире, мой друг Горацио. Вот. Ну, напишите, чтобы я не забыл. Я это отмечу у себя там, что бывает и так. И даже так бывает. Хотя вы ж понимаете, что так не должно быть. Оно либо должно работать, либо не работать.

Так вот, теперь, значит, смотрим э-э матрицу ПРЦ-2, модель ПРЦ-2. Здесь мы сразу двигаемся вправо до конца. И мы здесь видим, смотрите, безусловная вероятность встретить какого-то признака у всех объектов обучающей выборки. И условная вероятность встретить этого признака у объектов разных классов, разных категорий. И вот здесь мы можем уже сказать, что можно было бы сравнить. Вот, скажем, э-э, у какого-то, э-э, объектов какого-то класса, элементов компьютеров, вот этот вот признак цвет белый 14% встречается, а у средств связи 33%. А всего по всей выборке 15%. То есть мы видим, что этот признак более характерно является для средств связи, чем для элементов компьютера, да? Правильно? И у средств связи он встречается чаще, чем по всей выборке в среднем, средневзвешенное значение. А у элементов компьютера чуть-чуть реже. То есть если мы этот признак обнаружим, то мы можем сделать вывод, что скорее всего, этот объект является средством связи – это обобщающая категория. А может быть, конкретно это телефон, видите, да, вот прямо. Вот. И похоже, что это не элемент компьютера. Но про то, что это не элемент компьютера, очень мало информации, потому что мало отличается условная вероятность его встречи в группе элемент компьютера от от вероятности встречи по всей выборке. А вот для средств связи довольно заметное различие, в два раза.

Значит, еще я сейчас использовал термин "вероятность". Значит, вы должны понимать, что это, конечно, не вероятности никакие, а относительные частоты, условные и безусловные. Но мы не привыкли этот термин использовать: относительная, условная относительная частота, безусловная относительная частота. Значит, я сейчас вам проведу сравнение этих двух понятий: частоты, относительные частоты и вероятности. Как они взаимосвязаны? Вероятность – это есть предел, вот слушайте внимательно, предел, к которому стремится относительная частота при неограниченном увеличении числа испытаний. Вот если мы кубик игральный кинули, выпала тройка, то вероятность выпадения тройки 1/6, а относительная частота – единица, потому что один раз кинули, один раз выпала тройка. 1 / 1 = относительная частота 1. Число испытаний одно, выпала тройка. 100% выпаданий. Понимаете, да? А если мы еще раз кинем, выпадет четверка, тогда тройка станет 0,5 и четверка 0,5. Вот. А вероятность так 1/6 и будет. И вот если мы сделаем программку на каком-нибудь там Паскале, Random, Randomize там, равномерное распределение используем, возьмем массив, а индекс массива будем случайным образом определять, округлять до целого и суммировать этому элементу массива единичку. А потом возьмем и посмотрим, насколько отличается распределение, которое получилось у нас случайное, от теоретического. И посмотрим, насколько, значит, оно отличается и насколько сильно при разном числе испытаний. Ну у нас получится, что, во-первых, вот эта относительная частота будет очень быстро стремиться к 1/6. Очень быстро. При 500 примерно 480 испытаниях уже получается различие меньше 5% погрешность. А если будет 1200 испытаний, тогда 2,5%. Ну потом замедляется приближение относительной частоты к вероятности, замедляется. И последующим она все медленнее и медленнее приближается, потому что асимптотически приближается. Ну при 10.000 испытаниях будет там уже 1%, например, различие. Ну то есть фактически мы видим, что она стремится к этой 1/6.

Так вот здесь у нас, конечно, не вероятности никакие, а относительные частоты. Но они будут, скажем так, они отличаются, конечно, от вероятности, но отличаются все меньше и меньше при увеличении объема выборки. И сходится частота, относительная частота сходится к вероятности довольно быстро.

Значит, у меня когда-то возникла такая мысль, ребята, что вот сравнивать глазами – это, конечно, ну я скажу так, по-вашему, на вашем языке, это полный отстой, да, сравнивать глазами вот эти вот частоты друг с другом там и со средним по всей выборке. И я это автоматизировал сравнение. Вот, допустим, мы берем э-э системно-когнитивную модель, э-э... наличие экрана там, допустим, да, вот они смотрим. Мы видим, здесь числа с плюсом и с минусом. Это в битах, ребята, в битах посчитано, какое количество информации мы получаем о том, что объект принадлежит или не принадлежит к определенной категории, к определенному классу. Если число положительное, то это о том, что он принадлежит. Если отрицательное, то это значит не принадлежит. Мы видим, что эти числа могут быть довольно большими, и положительные, и отрицательные. Вот. Ну здесь правда уже потом идут... Вот. И таким способом сравнения довольно много, их семь используется: Inf1, N2, N3. N3 – это хи-квадрат. Сравнивается фактическая наблюдаемая частота наблюдения этого итого признака у объектов житого класса с теоретически наблюдаемой, теоретической частотой наблюдения этого признака у объектов этого класса. И эти две величины, ребята, фактически наблюдаемая частота и теоретическая, теоретическую частоту наблюдения итого признака у объектов житого класса, их можно сравнить двумя способами. Один способ – это вычесть одно из другого. То, что сделал Пирсон, когда разрабатывал свой критерий наличия причинно-следственной связи, который называется хи-квадрат. Это элемент этого хи-квадрат. А можно путем деления. Вот если мы хотим сравнить два числа 5 и 7, то можно вычесть пятерку из семерки, а можно разделить семерку на пятерку, да? Вот если мы вычтем, то у нас будет либо положительное, либо отрицательное число. А если разделим, тогда у нас будет число либо больше единицы, либо меньше единицы, либо равное. Ну так же, когда мы вычитаем, тоже ноль может получиться. Вот если мы разделим N и Т житое на вот эту теоретически ожидаемую частоту, то у нас получится вот это выражение. Смотрите. N и Т житое получится, а это вот, видите, перевернутое теоретически ожидаемая частота перевернутая. То есть это N и Т житое разделенное на теоретическую частоту. Это и есть количество информации по Харкевичу. Я сейчас вам объяснил, ребята, что критерий Пирсона хи-квадрат тесно связан с теорией информации и в частности с мерой Харкевича, количество информации. Вам едва ли кто-нибудь про это скажет. Следующая мера – это коэффициент возврата инвестиций. Условная вероятность делится на безусловную. Получается, если нет никакой зависимости, то есть возврат такой же, как и инвестиции, такой же и возврат. То есть инвестиции бессмысленные, никакой прибыли нет, но и убытков нет. Но получается единица это отношение. А чтобы получился ноль, надо вычесть единицу. И это и сделается. А для чего нужно, чтобы получился ноль? Дело в том, что количество информации по смыслу – это такая величина, которая суммируется. Вот, допустим, мы знаем какое-то количество информации содержится в одном каком-то признаке о принадлежности объекта с этим признаком к какому-то классу. А если у этого объекта не один признак, а 10? То тогда надо просуммировать это количество информации во всех этих десяти признаках о принадлежности к этому классу и к другим классам. И потом все классы рассортировать в порядке убывания количества информации. Вот тогда мы получим...
Евгений Вениаминович, у нас пара кончилась уже.
Уже кончилась?
Да.
Ну а что ж вы не сказали заранее? Ну ладно, ребята. Значит, я сейчас... Хорошо, ладно. Тогда конец занятия. Всего самого хорошего вам. С этого с этого места продолжим в следующий раз.
Ещё у меня вопросы тут возникли. В общем, я устанавливаю уже лабораторную работу, но почему-то в формате XLS он выдаёт мне ошибку.
А у вас Excel установлен или Open GL или что-нибудь, ой, вернее, этот Open Office или что-нибудь там такое?
Excel у меня.
Какой версии? Шестнадцатая, наверное?
Да, да.
Ну... Ребят... Ну, драйвера вот эти вот, которые преобразовать используются системой для преобразования стандартов этих файлов, они не могут быть использованы, если такая версия Excel стоит. До десятого, до десятого можно. Очень новое у вас программное обеспечение, язык программирования куплен в 2012 году, ребята, понимаете? Тогда не было шестнадцатого Экселя.
Ну да.
Не было десятки. И вот то, что я на нём делаю, оно на очень новом программном обеспечении не работает. Пожалуйста, возьмите, установите 2010 Excel.
Хорошо, я понял. Спасибо большое.
И будет работать. А, кстати, при том на формате XLSX оно работает, то есть оно устанавливается, но просит просто выполнить некоторые...
Ну, может быть, там что-то другое. Пришлите мне скриншот вот на этот адрес.
Хорошо.
Я посмотрю, что там система просит там и так далее. Там могут быть и другие причины. Может быть, там файл какой-нибудь покореженный у вас там или не знаю, что там. Или диапазон неправильно задан. Или версия... Вот когда мы там задаем, какой тип данных, мы должны задать или старый Excel, или новый. А у вас сверхновый там. Ну, в общем, короче говоря... Ну я думаю, разберемся. Это все делали люди, и мы тоже не хуже них, и спокойно во всем разберемся и сделаем. Давайте, до свидания.