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

***Российская Федерация***

## **100 Интеллектуальные информационные системы и технологии. Лабораторная 2. 2020-09-25**

## **Заголовок:** Обзор и классификация лабораторных работ в системе Aidos: Установка и формализация данных ЛР 3.03

### Резюме

Занятие представляет собой вторую лабораторную работу по дисциплине "Интеллектуальные информационные системы и технологии" для группы ИТ\_18-21, проводимую профессорами Луценко Е. В. и Аршиновым Г. А. 25 сентября 2020 года. Основное внимание уделяется работе с системой Aidos.

**1. Введение и проверка готовности:**
Подтверждается наличие системы Aidos у большинства студентов. Уточняется, что классификация лабораторных работ ранее не рассматривалась.

**2. Обзор системы Aidos и план занятия:**
Система Aidos предназначена для учебных и научно-исследовательских целей. Занятие включает:

* Классификацию встроенных (локальных) лабораторных работ.
* Процесс установки лабораторной работы на примере ЛР 3.03.
* Начало изучения ЛР 3.03, включая формализацию данных.

**3. Типы приложений в Aidos:**

* **Локальные:** Встроены в инсталляцию, не требуют интернета (кроме файлов помощи).
* **Облачные:** Скачиваются из интернета (более 200), доступны через FTP-сервер системы, включают форум для обсуждений.

**4. Классификация локальных лабораторных работ (4 типа):**

* **Тип 1:** Полностью готовые, просчитанные работы для изучения.
* **Тип 2:** Исходные данные формируются расчетным путем самой системой Aidos.
* **Тип 3:** Основаны на обработке данных из внешних файлов (Excel, текстовых). Считаются наиболее интересными для текущего этапа обучения. План включает изучение ЛР 3.03, 3.02, 3.04.
* **Тип 4:** Демонстрируют парсинг данных из интернета (сбор информации с сайтов по HTTP/FTP).

**5. Установка и формализация данных (на примере ЛР 3.03 - Тип 3):**

* **Установка:** Выполняется через режим 1.3 (Диспетчер приложений), с предварительной очисткой предыдущих данных в режиме 1.11. Установка ЛР 3.03 копирует необходимые файлы (включая Excel-файл с исходными данными) в рабочие папки системы.
* **Формализация данных (Режим 3.5):** Объясняется стандарт структуры исходных данных в Excel-файле:
	+ **Строки:** Объекты обучающей выборки.
	+ **Столбцы:** Атрибуты (шкалы).
	+ **1-й столбец:** Идентификатор объекта.
	+ **Желтые столбцы:** Классификационные шкалы (результаты, зависимые переменные).
	+ **Зеленые столбцы:** Описательные шкалы (факторы, свойства, независимые переменные).
* **Интерпретация:** Статическая (свойства и классы одновременны) и динамическая (факторы приводят к будущим результатам).
* **Типы шкал:** Номинальные, порядковые, числовые. Обсуждается автоматическая квантизация числовых шкал на заданное число интервалов (равной ширины или адаптивных).
* **Процесс:** Система анализирует Excel-файл, определяет шкалы, градации (уникальные значения), создает внутренние справочники и кодирует обучающую выборку.

**6. Просмотр результатов формализации:**

* **Режим 2.1:** Отображает созданные классификационные и описательные шкалы и их градации.
* **Режим 2.2:** Показывает обучающую выборку в закодированном виде (ссылки на справочники).
* **Режим 5.5:** Демонстрирует матрицу абсолютных частот (сходную с корреляционной), показывающую совместную встречаемость градаций описательных шкал и классов.

**7. Завершение и домашнее задание:**
Студентам предлагается самостоятельно установить систему Aidos (если еще не сделано) и повторить шаги по установке ЛР 3.03 и выполнению формализации данных (запуск режима 3.5).

### Детальная расшифровка текста

**00:00**

**Раздел 1: Введение и проверка готовности**

Здравствуйте, ребята.
Ребята должны сказать: "Здравствуйте, здравствуйте".
Здравствуйте.
Молодцы.
Вот. Ну что, начинаем наше занятие.

Сегодня, это уже идет запись, я под запись говорю. Сегодня 25 сентября 2020 года.
Шестая пара. 17:20-18:50.
Вторая лабораторная работа у группы ИТ\_18-21 по дисциплине интеллектуальные информационные системы и технологии.
Занятия ведут профессор Луценко Евгений Вениаминович и профессор Аршинов Георгий Александрович.

У нас, значит, теперь, э-э, вопрос к вам, ребята.
Э-э, кто-нибудь из вас помнит, на чем мы остановились на прошлом занятии? Что я вам там рассказывал на прошлом занятии?

Ну давайте тогда так попробуем. Значит, в чат: "Здравствуйте, здравствуйте".
О. Владислав поздоровался. Значит, ребята, значит, сейчас вы в чате ответьте, поставьте плюсики, у кого на компьютере стоит уже система Aidos.
Она должна у вас у всех стоять на компьютерах, установлена быть.
Дима, Владислав. Ну я так смотрю, неплохо так.
Вот.
Алина, Артур. Молодцы. А почему вы по два плюсика поставили? Два раза стоит, что ли?

Ладно. Значит, э-э, я вам рассказывал классификацию лабораторных работ или нет?
Или просто рассказывал про систему, как установить и все, и на этом мы закончили, да?
Дайте мне знать.

Классификацию, классификацию лабораторных работ мы изучали или нет?

Какие-то вы молчаливые попались, ребята.
Отвечайте. Если я спрашиваю, я вас спрашиваю, понимаете? Не Георгия Александровича, я именно от студентов спрашиваю.
Не изучали. Вот, молодцы. Но лучше не в чате отвечать, а микрофоном, потому что я сейчас включил доступ к экрану.
Вот, совместный доступ. И поэтому сейчас я просто не буду видеть этот чат. Вы должны видеть мой рабочий стол сейчас.
Ну, экран компьютера моего должны видеть.
Видите, да?
Должны подтвердить: "Да, видим".
Видим.
Молодцы.

**Раздел 2: Обзор системы Aidos и план занятия**

Значит, ребята, вот сейчас вот мы, значит, я что-то делаю, вы делаете то же самое на своих компьютерах.
Извините.
Значит, сейчас я буду рассказывать вам вопрос о том, какая классификация лабораторных работ, встроенных в инсталляцию системы Aidos.
Это первый вопрос.
Второй вопрос - это как установить лабораторную работу. Третий вопрос - начинаем изучать лабораторную работу 3.03.

Откуда эти вопросы берутся? Вот смотрите, ребята, сейчас я вам покажу.
Входим на мой сайт.
Второй пункт.
И инструкция для учащихся по разработке собственного приложения.
Вот мы идем, можно сказать так, по этому плану, буквально вот выполняя этот план.
Вот ссылочка на эту инструкцию.

Значит, здесь пункт первый. Рассказ о системе Aidos, коротко, очень такое упрощенное, короткий рассказ.
Потом мы ее скачиваем и устанавливаем. Это у нас была первая лабораторная работа.

**Раздел 3: Типы приложений в Aidos**

Теперь у нас вторая работа. Мы начинаем этой системой пользоваться для того, чтобы реально установить лабораторную работу. Но для этого сначала я показываю вам, какие типы лабораторных работ в этой системе есть. Почему я, э-э, встроил в ее инсталляцию лабораторные работы? Потому что система предназначена для учебных целей и проведения научных исследований. Но раз для учебных целей, то в ней есть учебные приложения. Все эти учебные приложения находятся в режиме 1.3.
Я пронумеровал все подсистемы вот так. И в каждой подсистеме режимы тоже пронумеровал. И, к сожалению, кое-где есть высокая степень вложенности этих режимов. Вот. Бывает иногда там. Вот, высокая степень вложенности, потому что многое.

Значит, переходим мы в диспетчер приложений. Это режим 1.3.
Все переходим в диспетчер приложений. Здесь у меня уже лабораторная работа есть, поэтому я сначала все стираю, чтобы с нуля начинать. Все стираем мы все приложения в режиме 1.11.

Ребят, у нас лабораторная работа - это работа диалоги. То есть я вам все показываю, рассказываю, подробно объясняю. Но вы полное право имеете и даже должны меня спрашивать, если что-то у вас не получается. Что-то непонятно, пожалуйста, спрашивайте, проявляйте активность.

Вот. Значит, есть локальные лабораторные работы, встроенные в самую инсталляцию. Только спрашивайте микрофоном, а не э-э, в чате. Потому что в чате мне надо специально там нажимать, чтобы посмотреть, как вы поняли. Вот.

Значит, есть локальные, встроенные в самую инсталляцию, э-э, учебные приложения, то есть лабораторные работы, э-э, которые не связаны с интернетом. То есть вот есть у вас инсталляция на флешке, взяли, записали на компьютер, и, пожалуйста, себе, интернет не нужен, чтобы ими э-э, пользоваться, чтобы их изучать. Единственное, хелпы скачиваются с моего сайта, поэтому не будут, не будет помощи по лабораторным работам. Вот. Ну так вот, э-э, можно будет их изучать как э-э, без этого.

Вот. И второе, есть лабораторные работы, которые скачиваются из интернета. Э-э, их называю облачные Aidos-приложения. На данный момент их 207 этих облачных Aidos-приложений, но их количество постоянно увеличивается.

**Раздел 4: Классификация локальных лабораторных работ (4 типа)**

Локальные работы. Как они классифицируются? Они четырех типов.

* **Подраздел 4.1: Тип 1**
Первый тип лабораторных работ, он полностью готов для изучения. Они полностью просчитаны эти работы, все выходные формы. То есть они полностью готовы для изучения.
Извините, секундочку.
Да, Степан.
Алло, здравствуйте. Подъезжаешь? Я выступаю к вам еду, я сейчас на Вишняков встану. А, ну ты когда будешь там остановки две-три, позвони, ладно? Выйдет моя жена с собачкой такой маленькой, пушистенькой. Вот. Шпиц, бело-бело-рыженькая такая. Она к тебе подойдет там на остановке. Я думаю, вы друг друга там увидите. Потому что я занятие веду. Я веду занятие, я не могу просто. Хорошо. Угу. Спасибо тебе большое за все, за работу.
На Вишнякова пока что. На Вишнякова он.

Вот. Значит, первого типа лабораторные работы полностью готовы для изучения.

* **Подраздел 4.2: Тип 2**
Второго типа. Это лабораторные работы, у которых исходные данные формируются расчетным путем самой системой Aidos. Когда мы здесь вот переключаем вверху, то меняется вот этот список, видите, другие работы. Здесь вот 10 работ первого типа, 10 работ второго типа.
* **Подраздел 4.3: Тип 3**
10 работ третьего типа вот здесь. Работа третьего типа, на мой взгляд, наиболее интересная. Хотя раньше, когда я их изучал, в таком порядке, как здесь вот они у нас расположены. То есть сначала первого типа, потом второго. На занятиях мы изучали со студентами. Но времена меняются, и сейчас студенты стали более такие смышлёные, подготовленные. И мы сразу начинаем работу третьего типа изучать. Я всех ориентирую на то, что вы должны сами разработать новое приложение после того, как мы изучим с вами три лабораторных работы.
Вот я вам показываю, это инструкция, как сделать собственное приложение. Здесь вот пункт второй, видите? Изучаем лабораторную работу 3.03, потом 3.02, потом 3.04, вот в таком порядке. И вот сейчас мы выполняем пункт два. То есть мы изучаем первую из этих лабораторных работ. Но сначала вот вообще мы их классификацию изучаем, а потом начнем ее изучать.
Вот. Работа третьего типа. Чем они интересны? Тем, что это работы, которые основаны на обработке данных, которые находятся во внешнем файле. Ну, допустим, вот в экселевском файле или в текстовом файле, или в графическом файле. Вот. Ну, работа с текстовыми и тут графики нет, а с, в смысле, исходных данных графических нет, а в других местах они есть. Вот. А здесь табличные текстовые данные обрабатываются.
* **Подраздел 4.4: Тип 4**
И четвертого типа работы. Может быть, вы слышали, ребята, что такое парсинг? Кто-нибудь слышал, нет? Термин такой, парсинг. Это означает, что какая-то программа автоматически обращается к сайтам по HTTP протоколу и FTP протоколу и скачивает оттуда какую-то информацию с сайтов с этих. И потом с этой информацией работает. Вот этот язык программирования, на котором написана система...
Да? Евгений Винеминович, извините, вас беспокоит помощник... О, Господи! Давайте я вам позвоню, потому что я веду занятия. У меня большая нагрузка, я веду занятия. Вы мне звоните, когда я занятия не в перерыве, а во время занятия звоните. Сейчас тоже у вас занятие? Да, сейчас занятие. Оно с 17:20 началось и в 18:50 закончится. Хорошо, я позвоню в 18:50. Спасибо. Ага, спасибо вам.

Вот. Значит, э-э, в системе э-э, реализован как пример такая работа, которая основана на скачивании по FTP данных с сайта. Ну, можно и по HTTP скачивать. И потом анализировать, находить там гиперссылки, по этим гиперссылкам переходить, опять скачивать. То есть можно сделать на локальном компьютере программу, которая будет лазить по интернету, по ссылкам и там собирать информацию, анализировать её. Ну это сейчас я просто вам показал, что есть работа, которая демонстрирует такую возможность техническую.

**Раздел 5: Установка и формализация данных (на примере ЛР 3.03 - Тип 3)**

* **Подраздел 5.1: Обзор облачных приложений (перед установкой)**
Вот. И рассмотрим, какие работы есть у нас облачные Aidos-приложения. Они находятся на FTP сервере системы. У неё есть свой FTP сервер. Сейчас я на него выйду по FTP протоколу и покажу вам, что там находится.

Прежде всего, там находится база данных, в которой отмечаются все запуски системы Aidos в мире. Этот режим я сделал в конце 2016 года, и с тех пор там отмечаются все запуски системы Aidos в мире. Вот сегодня, вот видите, 25-го числа довольно много было запусков из разных мест.
Тут описано дата, время, IP-адрес. Ну здесь они в таком виде компактном. Я вам покажу, как когда будем систему, значит, смотреть, там как-то покажу вам в другом, в другой форме. А вот здесь вот у нас папочка, где находятся облачные Aidos-приложения. Вот их 207 папочек. Открываем любую, там находятся файлы этого приложения, описание, каталог приложений. То есть всё там есть.

Значит, и мы можем любое приложение рассматривать как э-э, тему форума. То есть мы можем это блок обсуждений посмотреть, увидеть, что, когда делали пользователи, какого числа они устанавливали приложение, какого числа писали сообщения, в каком, в каком приложении они писали эти сообщения. И мы можем выйти в любой момент на любое приложение и написать там какое-то сообщение. Люди, которые это увидят это сообщение, они могут э-э, пользователи во всём мире могут открыть это сообщение и прочитать.

Вот. Мы видим, что можно получить гиперссылки абсолютные на файлы этого приложения. То есть мы можем вот такую гиперссылку в строку адреса поместить, и у нас этот файл скачается. Тут есть файл, видите, описание Redmi. Вот. Можно это скачать и по-другому, посмотреть. Можно просто э-э, клацнуть по какому-то полю, вот, допустим, два раза, и загрузится описание соответствующее в браузере по умолчанию. Вот это описание приложения.

И это приложение, оно написано как лабораторная работа, в смысле, описание это как написано как лабораторная работа, и можно полностью это всё выполнять, и всё так же на экране у вас и будет. И исходные данные, и все формы, всё будет получаться.

Вот. Если вас это приложение заинтересует, прочитав описание, вы посмотрели, и вам понравилось, и э-э, вы приняли решение ознакомиться более внимательно, можно его установить. Тогда это будет скачано, вот эти файлы будут скачаны этого приложения на локальный компьютер и запущен программный интерфейс э-э, преобразования исходных данных, которые в этом приложении, э-э, в стандарт системы Aidos, создание соответствующего локального приложения в системе, и потом дальше его изучение.

Значит, здесь также есть э-э, ссылки на мой сайт, на форум Роджера Донна, разработчика этого языка программирования, на котором написана система. И на этом форуме очень много есть обсуждений. И я там тоже принимал участие в обсуждениях. Вот видите, там вот Луценко там. Иногда, э-э, чаще всего я что-то спрашиваю, и мне помогают. Но иногда бывает, что-то и я кому-то там подсказываю. В общем, можно вот э-э, как бы общаться здесь. И пользователи могут размещать здесь новые приложения в этом облаке. Но сейчас пока что никто это сам не делает, а только когда э-э, я э-э, вижу, что нормальное приложение, ну, со студентами, студенты, допустим, сделают нормальное приложение под моим руководством, и потом я его размещаю. Хотя в системе не это открыто, этот, это открыта возможность, не заблокирована, и любой пользователь может это сделать.

И вот здесь следующий пункт - это записать приложение, но только в том случае это возможно сделать, если оно уже есть на компьютере и если есть описание его, этого приложения.

* **Подраздел 5.2: Установка ЛР 3.03**
Ну теперь мы что делаем? Выбираем, второй пункт рассматриваем, как установить лабораторную работу. Выбираем лабораторную работу, вот, которую мы устанавливаем, 3.03. Нажимаем Окей. Появляется сообщение о том, что все файлы вот в этой папочке, предназначенные для исходных данных, будут удалены. Идём, смотрим на эту папочку. Не исключено, что мы там что-то делали, там какие-то есть файлы. Значит, вы имеете в виду, что дальше, сейчас, вот если я кликнуть по Окей, то эти файлы будут удалены. То есть вы начинаете изучение нового приложения. То, что было раньше, э-э, в этой папке исходных данных стирается.

Значит, что сейчас произошло? Смотрите. Вот появился файл. Видите, там было много файлов, теперь только один файл там. И мы видим здесь вот, смотрите, LabWorks папочку. Это всё в папочке AidData, AidData, то есть это данные системы Aidos. База данных системы Aidos. Там довольно много папок, есть и вложенные папки есть там и так далее. Значит, вот сейчас вот смотрим. Э-э, LabWorks. Я вам говорил, что каждого типа лабораторных работ по 10 штук. С первой по десятую - это первого типа, с одиннадцатой по двадцатую - второго типа, с двадцать первого по тридцатого - третьего типа. А четвёртого типа она вообще здесь э-э, отсутствует папочка, потому что исходные данные скачиваются из интернета. А это вот работа 3.03, она вот в двадцать третьей папочке находится. И мы можем посмотреть, вот наименование этой работы, а вот файл исходных данных. А этот файл 2322 - это файл, в котором хранятся параметры вот этого окошка, которое открылось сейчас, когда мы устанавливаем лабораторную работу. Ну и дальше сейчас будем двигаться, как это принято в системе предполагается.

* **Подраздел 5.3: Формализация данных (Режим 3.5): Структура и параметры**
Значит, смотрите, ребята, значит, вот здесь вот в этом окошке есть такие много разных параметров. Значит, один параметр вот здесь слева вверху - тип данных. Какой тип данных? Значит, мы в данном случае пишем, что это 2003 Excel. Э-э, потом э-э, указывается, как считать нули и пробелы? Данными или отсутствием данных? Я советую считать нули и пробелы отсутствием данных. А если вы хотите какое-то очень маленькое число записать, писать там, допустим, одна стотысячная, например, что-то очень близкое к нулю. Вот эта опция нам обычно не нужна - создавать базы данных средних по классам из исходных баз данных. Ну потом здесь задаются ещё параметры, которые сейчас я вам объясню на хелпе. Вот в этом режиме есть свой хелп, который предназначен для того, чтобы объяснить учащимся там или тем, кто осваивает систему, какова структура или стандарт этого экселевского файла, внешнего, в котором находятся исходные данные для приложения. Это довольно-таки много надо рассказывать сейчас, я это и сделаю.

Значит, прежде всего, значит, есть у системы Aidos, много есть программных интерфейсов автоматизированных. Automatic Program Interface, API называется сокращённо, которые позволяют вводить в систему данные различных типов из внешних файлов: текстовых, экселевских табличных файлов, вот, графических файлов и CSV. Ну CSV - это для специальных случаев только. Вот. И э-э, все эти интерфейсы, их довольно много, э-э, ну, скажем, экселевских файлов два стандарта, графических тоже два стандарта э-э, для ввода в систему. Два способа как бы ввода информации в систему. И текстовых тоже несколько стандартов есть. Короче говоря, э-э, я хочу сказать, что вот то, что я сейчас рассказываю, это один из программных интерфейсов, который называется API 2322. Там есть API 2321, 2323 там и так далее.

Вот. Значит, в этом интерфейсе такой стандарт. Значит, э-э, экселевский файл, он может быть до миллиона строчек в этом файле. В новых версиях Экселя, в старых 65.000 строчек. Ну, до миллиона строчек. И в каждой строчке описан один объект обучающей выборки. Обучающая выборка в нейронных сетях называется тренировочной выборкой, на которой тренируют сеть, разбираться с тем, какие взаимосвязи между значениями на рецепторах и принадлежностью каким э-э, нейронам, какие нейроны должны активироваться. Это делается в тренировочной выборке информация для этого исходная. А у нас называется обучающая выборка. Это синонимы.

Колоночки, колоночки этого файла экселевского, они описывают э-э, различные аспекты этих объектов наблюдения. Первая колоночка, она описывает, что это вообще за объект. То есть прямо может быть написано. Если, допустим, мы разрабатываем методику, э-э, которая будет выяснять, как влияют техно- агротехнологии на результаты выращивания, то здесь могут быть приведены примеры, как мы выращивали там пшеницу, к примеру, в течение там 10 лет. И здесь будет написано: в 2000 там 15 году на поле номер 16 э-э, э-э, выращивали пшеницу сорта такого-то. Вот.

А дальше идут результаты выращивания. Обычно это результат в натуральном выражении и в стоимостном выражении. В натуральном выражении чаще всего это количество продукции и качество продукции, а в стоимостном выражении чаще всего это прибыль и рентабельность. Вот.

Э-э, вот эти, это жёлтым фоном выделено. Таких вот, я их называю такие колоночки классификационные шкалы, потому что они позволяют описать принадлежность объекта наблюдения какому-то обобщающему классу, обобщающей категории. И потом все наблюдения, которые тоже принадлежат этой категории, они будут сгруппированы и использованы для формирования обобщённого образа этой категории. То есть система будет обобщать эти вот описания конкретных ситуаций, конкретных объектов, их состояний.

А вот то, что зелёным фоном выделено, это колоночки, которые позволяют описать, какие свойства были у этого объекта наблюдения, и какие значения этих свойств в строчечках, в клеточках будет написано. Или какие факторы действовали на объект наблюдения, и какие у них были значения. Ну я так немножко специфически выражаюсь, но я выражаюсь точно. То есть я использую точную терминологию для того, чтобы об этом говорить.

Значит, что такое свойство? Значит, свойства бывают физические и социальные у объектов... И не дала, да? А. Значит, свойства физические, ну, допустим, может быть температура, там, вес. Физические свойства, э-э, физическое свойство вес, а значение 65 кг.
Да, Степан. Алло, да, я уже вышел. А, ну сейчас вот жена подойдёт. Она с собачкой, с собачкой такой беленькой маленькой, лохматенькой. Угу. Ну давай, спасибо тебе большое. А номер дома ещё раз? 121. Ага, всё. Угу. Ну на угловой дом, девятиэтажный, белый. Давай. Спасибо.

Вот, это физическое свойство. Э-э, или э-э, значит, социальное свойство - стоимость. Стоимость телефона. Вот здесь вот телефон такой-то марки, такой-то модели, а здесь, значит, его стоимость. Вот. Или какие-то его характеристики там, объём памяти там, разрешение его экранчика там и так далее, ёмкость батареи. То есть это его свойства физические и социальные. Социальное свойство, чем отличается от физических? Это в отношении к людям эти свойства. Стоимость - это свойство не физическое, а это его отношение к этому устройству людей. То есть люди его ценят вот так-то вот, настолько-то его оценивают. И ещё могут быть такие социальные свойства, как, допустим, он мне нравится этот телефон. Ну, приятный его взять в руки там, допустим, там и так далее. То есть он приятно сделан, то есть дизайн хороший там. Вот. Это социальное свойство. То есть сам по себе физически он это свойство у него отсутствует, а вот людям он нравится эстетически, допустим. Вот. Ну то есть есть физические свойства, социальные. Мы здесь можем все эти свойства обрабатывать.

А также, значит, эти, значит, существует три э-э, вида терминологии, позволяющих описать этот вот файл исходных данных. Один вид терминологии - это э-э, универсальная терминология. Я его говорю так: описательные шкалы и градации, классификационные шкалы и градации. А статистичная терминология, которая предполагает, что свойства объекта или явления наблюдаются у него сейчас, прямо вот сейчас, в текущий момент времени, и принадлежность его каким-то обобщающим категориям, классам э-э, этого объекта или явления тоже сейчас, одновременно с его свойствами.

Вот. Но тут я скажу, что так не бывает на самом деле. Но это понятие сейчас, оно может быть таким довольно расплывчатым. Ну, допустим, если у человека какие-то есть симптомы, и ему ставят диагноз, то симптомы и диагноз, они одновременно или нет по времени? Ну все понимают прекрасно, что для того, чтобы поставить диагноз, нужно некоторое время потратить. А ещё бывает эти анализы надо собрать какие-то, кроме внешне наблюдаемых симптомов, клинической картины, надо какие-то ещё анализы сделать. А анализы, может быть, могут быть длительными, кстати. Может быть, и неделю там их надо сдавать разные, и две даже. Вот. И потом врач всё это собирает всю эту информацию, анализирует и ставит диагноз. И считается, что у него симптоматика и результаты анализов, анамнез сейчас, и диагноз тоже сейчас. Хотя на самом деле фактически симптоматика, информация о пациенте, анамнез и результаты анализов относятся к прошлому, а диагноз сейчас как бы к настоящему. То есть между ними есть расстояние во времени. Но считается, что они одновременно. То есть в медицине сейчас это может там неделю занимать это сейчас. Это не миллисекунды, как вот субъективно мы думаем, что сейчас это то, что одновременно. А в медицине то, что неделю в течение недели - это тоже сейчас при постановке диагноза.

Вот. Но есть и динамическая э-э, интерпретация этой терминологии. Когда мы говорим так: вот эти зелёненькие э-э, колоночки, они соответствуют факторам каким-то, которые влияют на объект исследования, на объект моделирования. Этот объект моделирования может быть в разных состояниях, которые описаны в разных строчках. И может быть много строчек, которые описывают одно состояние, или там у них разные сочетания принадлежности к разным классам. В одной строчке там одно сочетание, в другой - другое. И вот мы говорим так: на этот объект моделирования действуют какие-то факторы, которые перечислены вот в этих колоночках: давление, температура, там, допустим, там, ещё что-то, вибрация, например, там какие-то действуют, излучения какие-то действуют. Конкретно указывается в каждой колоночке, какой вид излучения, скажем, бета-излучение, гамма-излучение, альфа-излучение там. Вот.

А здесь э-э, написана величина. Ну, допустим, если мы говорим о том, какое давление, там говорим, три атмосферы, допустим, здесь пишем, какая температура, там пишем, -15° там. И, в общем, всё мы вот это пишем, какие значения этих факторов. И говорим так, что вот здесь вот описываются причины, действующие на объект моделирования, различные факторы. А здесь описывается результат действия этих факторов, который, естественно, относится к другому времени, чем сами эти факторы. То есть факторы действуют на объект моделирования, и он под их действием, этих факторов, переходит в некоторое будущее состояние определённое. Вот такая возможная интерпретация.

Теперь, что же в нашей задаче? Мы видим здесь, что у нас здесь указано диапазон классификационных шкал 2-3. Это то, что здесь вот жёлтеньким указано. А сколько может быть таких шкал в задаче, ребята? Как вы думаете? Значит, э-э, современные, то есть новые версии Экселя, ну я имею в виду там 2010 Excel, э-э, он позволяет до 16.000 колонок использовать. То есть мы можем сделать, скажем, там, 150 классификационных шкал и 600, например, описательных шкал. Понимаете? Вот. Но в системе Aidos есть ограничения определённые, ну, связанные с тем, как она реализована программно. Вот. Можно было по-другому реализовать, чтобы этих ограничений не было. Но я реализовал так, что у неё есть эти ограничения. Когда я её реализовывал, то мне казалось, что никогда она не мы не натолкнёмся на то, чтобы эти ограничения сыграли роль. А уже это произошло. Но я могу вам сказать, что можно до полутора тысяч классов исследовать в системе Aidos. Это суммарное количество уникальных градаций классификационных шкал. И э-э, э-э, скажем так, э-э, десятки тысяч описательных шкал. Много это или мало? Это, ребята, колоссальное количество. Почему? Колоссальное просто. Потому что в системе SPSS, например, можно не больше 70 шкал использовать. А здесь я говорю, ну там 5-7.000, там, 10.000. Я решал такие задачи, когда там эти вот десятки, сотни тысяч этих вот значений этих описательных шкал там было. И самих шкал были, допустим, там 15.000, например. То есть это намного, просто на порядке превосходит возможности статистических систем. И тем более таких методов математических, как факторный анализ, где всего считается, что многофакторный он тогда, когда уже семь факторов считается он многофакторный. А когда здесь вот семь факторов, то мы просто так улыбаемся, что это такая простенькая задачка, совсем простенькая. Вот сейчас у нас, например, будет восемь факторов, ребята.

Вот смотрите, мы задаём диапазон. Классификационные шкалы со второй по третью, а описательные с четвёртой по двенадцатую. То есть восемь факторов. Теперь вот здесь вот у нас есть какие опции? Формализация предметной области. Это что включает в себя? Разработку классификационных и описательных шкал и градаций и кодирование исходных данных с этим, с их использованием и формирование обучающей выборки. А этот вот режим генерации распознаваемой выборки или тестовой выборки, как говорят э-э, в нейронных сетях. Э-э, дело в том, что тестовая - это не совсем правильно. Тестовая - это предполагает проверку э-э, модели по каким-то критериям. А если мы просто ей пользуемся, не проверяем её, а просто пользуемся для решения задач, тогда более правильно говорить, что это просто там э-э, выборка распознаваемая или прогнозируемая выборка. Но не тестовая. Но если мы её используем для того, чтобы оценить достоверность модели, тогда тестовая, да. А если для решения задач, тогда не тестовая, а просто распознаваемая выборка. Тогда используются уже существующие шкалы и градации для того, чтобы закодировать файл для распознавания, который такую же структуру имеет, как и файл исходных данных.

Если есть, да, теперь вопрос о том, какие типы данных могут быть в этих колонках. Они могут быть текстовые и числовые, ребята. И такие, и такие. Причём, э-э, если они текстовые, то они могут быть двух видов: номинальные шкалы и порядковые шкалы. Чем они отличаются? В номинальных шкалах есть только операция эквивалентности и неэквивалентности. Список студентов, например, и конкретный некий студент, он каким-то пунктом этого списка не соответствует, а одному соответствует. Порядковые шкалы. Если этот список рассортирован по какому-то критерию: по успеваемости, по алфавиту там или ещё каким-то способом, или по сложному какому-то ключу составному. Тогда это порядковая шкала.

Если же у нас есть числа вот здесь в колоночках, то тогда это числовая шкала. Тогда можно эти числа складывать, вычитать. Но нужно иметь в виду, что если там везде числа, а в одной строчечке там текст, то тогда все остальные числа будут рассматриваться как текст. Потому что иначе этот текст рассмотреть нельзя, это более общий тип данных, чем числовой. То есть числа являются текстом, но не все тексты являются числами. Вот в этом смысле. Вот. И классификационные описательные шкалы могут быть номинальными, порядковыми и числовыми.

Если шкала числовая попадается в классификационных или описательных шкалах, то тогда можно задать число числовых диапазонов, на которые разбивать э-э, диапазон изменения э-э, значений этой шкалы. Ищется минимальное и максимальное значение в шкале, э-э, в колоночке, и разбивается на заданное число интервалов одним из способов. Этих способов сейчас два в системе Aidos. Один способ - равные по величине диапазоны числовые, интервалы, но при этом в этих диапазонах получится разное число наблюдений. Или разные по величине диапазоны, разные, но при этом в них будет одинаковое число наблюдений с точностью до единицы, то есть в целых. Ну по какой причине? Ну если вот три землекопа копают две ямы, то сколько землекопов копает одну яму? Берём мы три делим на два, получаем полтора, и ответ такой: Одну значит, э-э, три землекопа копают две ямы, одну яму копает полтора землекопа. Значит, полтора землекопа в мультике показывали, это такое, непонятно, как он копает, во-первых. Но не в этом дело даже. Дело в том, что он не то что там копать, он даже и двигаться не сможет, если он будет половина землекопа. Короче говоря, ясно, что одну яму копают два землекопа, а другую - один. То есть вот такие задачи, которые решение имеют в целых числах, там э-э, когда я говорю, что равное число наблюдений, то нужно иметь в виду, что равное с точностью до единицы в целых. Потому что наблюдений вот этих вот строчек не бывает полторы строчки, бывает либо одна строчка, либо две, либо три, а полторы не бывает наблюдений, понимаете?

Значит, сейчас мы по эти параметры по умолчанию используем. Нажимаем Окей. Что происходит? Этот экселевский файл, который я пока я ещё его не показывал, но я показывал, откуда он взялся и где он находится. Сейчас я его покажу потом. Он проанализирован. При этом проанализирован диапазон классификационных шкал. Ну он преобразован в стандарт системы Aidos внутренний. DBS файл э-э, получился. При этом, кстати, ребята, надо вам сказать, это момент такой, ну, существенный момент, в общем, э-э, который заключается вот в чём. Для того, чтобы преобразовать экселевский файл во внутренний стандарт данных, для этого система Aidos э-э, использует... Сейчас я вам в чат пошлю э-э, не было интернета. Добрый вечер, добрый вечер. Отлично. Значит, ребята, я вам пошлю некоторые требования э-э, или настройки э-э, системы Windows 10 для того, чтобы работала система Aidos. Здесь одна из настроек, но строка, где эти, то есть этот, это место сайта, где есть эти настройки, вот здесь. Я вижу, что в чате не полностью поместилось это сообщение.
Значит, э-э, там есть, в частности, такой пунктик в этом сообщении, что должен быть на компьютере установлен MS Excel версии от 2003 до 2010. Значит, если его там нет Экселя, вот, а есть там Open Office или что-нибудь такое вот аналогичное Экселю, но не Excel, то тогда э-э, не будет преобразовываться экселевский файл во внутренний стандарт данных, потому что система Aidos пользуется э-э, конвертером, который есть в Экселе встроенном, для того, чтобы это сделать. Это первое. Второе, этот конвертер нормально работает у версий 2003-2010. А у 2016 Экселя этот конвертер э-э, не позволяет сделать то, что делает система Aidos, когда преобразует внешний внешний файл экселевский в DBS файл. Вот такая вот новость.

И э-э, что произошло, собственно говоря? Этот файл был преобразован во внутренний стандарт данных, и проанализирован, и было обнаружено, что у нас есть две текстовые классификационные шкалы с суммарным числом градаций текстовых 14. А когда шкала текстовая, то там просто э-э, градациями являются уникальные текстовые значения в этих шкалах. Это и есть классы, ребята, классов 14. В матрице модели классы идут колоночками, их 14 в данном случае. А строчек 50. А как они образуются? Это восемь шкал описательных текстовых, в которых 38 градаций уникальных текстовых, и одна числовая шкала описательная, в которой 12 градаций. Всего 50 градаций. Вот здесь мы можем задать число диапазонов числовых, на которые разбивается диапазон изменения э-э, значений этой единственной шкалы. Если бы было несколько, тогда это было бы для всех шкал действовало. Если мы зададим здесь пять, например, то нужно пересчитать шкалы и градации, размерность модели изменится. Вот. Но сейчас мы это делать не будем, выйдем на создание модели.

**Раздел 6: Просмотр результатов формализации**

Что произошло, ребята? Произошла формализация предметной области. То есть были сформированы классификационные описательные шкалы и градации на основе файла исходных данных. А потом с их использованием было закодировано, закодированы исходные данные, в результате чего получилась обучающая выборка. Потом всё это было переиндексировано. Потом появляется сообщение для студентов, которые студенты, насколько я знаю, всегда Окей сразу нажимают, не читая. Такие вот сообщения, их раздражают студентов. Они просто раз, клац, сразу Окей и дальше двигают. А дальше что нам делать, спрашивают. Я говорю: "А дальше надо вернуться назад, ребят, чтобы появилось опять это окошко. Вот. И прочитать, что здесь написано. А здесь написано, вот смотрите, что нужно открыть файл исходных данных по пути. Давайте это сделаем. Этот путь будет отличаться на разных компьютерах, потому что система Aidos на разных компьютерах в разных местах установлена. И мы видим файл исходных данных. Строчки - это различные предметы, которые лежали на столе, когда студенты делали эту лабораторную работу и создали этот экселевский файл, и прямо на лабораторной работе вводили туда информацию. И я взял один из этих файлов. Каждый делал это на своём компьютере по-своему, совершая собственные ошибки. Вот. Ну я один из этих файлов скопировал и сделал лабораторную работу на его основе. Он не очень корректный этот файл. Значит, сейчас опишу, что здесь не совсем так.

Значит, здесь вот, видите, в первой колоночке просто название объектов, которые лежали на столе. И ещё там были в комнате, я так понял, там, допустим, там вешалка. Ну а здесь уже ребята немножко включили фантазию, написали, что там мячики ещё лежали там. Вот. Две классификационных шкалы: конкретная и обобщающая. Ну то есть разная степень обобщения. Свойства этих объектов. И, значит, само свойство и значение этого свойства. Скажем, размер и прямо величина этого размера. Ну, конечно, здесь надо будет писать в сантиметрах там или в чём там, в километрах. Так вот, ну здесь это сделано очень упрощённо.

Есть шкалы так называемые логические, где два значения всего бинарных: да, нет, есть, нет. Можно писать 1, 0, по идее. Но лучше словами, тогда понятнее потом в выходных формах получается. И вот одна шкала числовая, где размер указан в каких-то единицах измерения. Получилось, Лизочка? Ну хорошо. А здесь он написан текстом.

Какие здесь есть моменты некорректные? Вот смотрите, значит, вот здесь вот есть белый, а вот здесь вот есть белая. Это разные значения свойства цвет. Но мы по смыслу понимаем, что это одно и то же значение, но поскольку оно написано по-разному, то это будут разные свойства. То же самое касается чёрный и вот здесь вот чёрная, видите? И вот чёрный. Здесь вот чёрный написан с маленькой буквы, здесь с большой буквы, а здесь написан с маленькой, да ещё женского рода. Вот такие вот э-э, особенности здесь есть. Но это я не стал их исправлять, чтобы показать вам, что это неправильно. Вот. Теперь вот этот размер один - это явно порядковая шкала, правильно? То есть у нас есть самые маленькие, средние и большие. А здесь вот есть большой, видите? А ещё есть большая. Наверное, конечно, писать всё в одном роде, в среднем там или в мужском, или в женском. Потому что большой и большая получилось разные будут это свойства, э-э, объекты по шкале размер один. И это свойство будет отражать не только размер, но и род в русском языке этого объекта. Ну, скажем, вешалка она большая, а вот сумка почему-то большой. Кто написал сумка большой, не знаю. Ну, наверное, не буду говорить, чтобы никого не обижать. Вот. Сумка большой. Какой сумка? Большой. Понятно, да? Ну вот.

Так вот мы, в общем, видим, как это организована таблица. Теперь мы э-э, это выполнили этот пункт. Следующие пункты мы должны выполнить 2.1, 2.2, 2.3. А? Ага, ну давай, аккуратненько.

Вот мы видим, что появилось сообщение в диспетчере приложений, что приложение установлено. То есть этим приложением можно пользоваться, с чем мы сейчас и займёмся. Здесь может быть много приложений, но активным должно быть только одно. Там должна быть W, буковка Work, рабочая. Но если будет несколько указано, в принципе, это возможно указать несколько приложений с буковкой W или другой буковкой, системе всё равно, не пробел, главное. Вот. То тогда используется первое из них без, которое не пробел, а указана буковка там.

Ну, конечно, если эти приложения, много приложений, которые не имеют отношения к делу, не интересуют вас, то можно их всех удалить э-э, используя вот режим 1.11. Значит, теперь выполним рекомендации, которые были вот в этом окошке. Режим 2.1 посмотрим. Что это за режим? Это режим, который показывает нам, где слева в левом окошке наименование классификационных шкал: конкретный класс, обобщающий класс. Вот они есть. Описательные шкалы. Прямо вот то, что здесь написано, было сюда перенесено. Значит, это название шкал классификационных. А что в качестве градаций этих шкал мы видим? А мы видим уникальные текстовые значения в этих шкалах. Они являются градациями.

Переходим в режим 2.2 и смотрим, что у нас тут. А здесь у нас тоже вот эти наименования в левом окошке - это наименование описательных шкал: цвет, материал, размер. Видите, вот они соответствуют тем, которые в Экселе. Это что означает? Поскольку эти названия шкал и классификационных, и описательных появляются в выходных формах, то я советую всегда э-э, писать эти названия минимальной длины, но так, чтобы было ещё понятно, чтобы это не сказывалось на понимании, чтобы было понятно, что это значит.

И вот мы смотрим на то, какие это шкалы. Цвет. Какая это шкала? Номинальная. Здесь они просто цвета по алфавиту рассортированы, никакого особого смысла в них нет, в их порядке, я имею в виду. Если бы мы хотели бы, мы могли бы рассортировать их по э-э, спектру, не по алфавиту, а по спектру. И тогда был бы порядок осмысленный. Ну, допустим, длина волны возрастает или уменьшается от одному к другому. И тогда было бы, чтобы это сделать, нужно было бы здесь в экселевском файле, там где цвет, указать для того, который должен быть на первой позиции, 1-12, для второго 2-12, 3-12, 4-12, 12-12. При сортировке по алфавиту они были бы в нужном порядке. Ну было бы примерно так: 5 наклонная черта 12 - жёлтый. Вот она будет у нас пятой позицией находиться.

Материал - это точно номинальная шкала, не порядковая. Размер - тоже порядковая шкала. Но мы здесь видим, что у нас порядок совершенно не тот, который смысловой в данном случае. Маленький должен быть на первой позиции, потом средний. И вопрос возникает о том, что здесь вот у нас есть большая и большой. Ясно, что в экселевском файле надо их одинаково называть. Вот. Ну, допустим, в мужском роде там. И большую убрать просто и написать большой в этом месте. Тогда здесь вот уже будет четыре градации. А чтобы они были по алфавиту э-э, и при этом в нужном смысловом порядке, для этого нужно, чтобы маленький был назван 1 наклонная черта 4 - маленький. Средний - 2 наклонная черта 4 - средний. То есть первая э-э, числитель этой дроби, он будет показывать позицию, э-э, а потом наклонная черта и знаменатель - число этих позиций, количество, количество градаций. Тогда они будут все в нужном порядке. А потом будет 4 - то есть наклонная черта 4 - большой. Это будет максимальная градация.

А это вот числовые градации. Здесь они автоматически так нумеруются с такой нумерацией, как я сейчас вам сказал. Находится минимальное и максимальное значение, делится на заданное число диапазонов. Здесь эти диапазоны равного размера. А если бы там поставили опцию э-э, адаптивные интервалы, разного размера, они были бы разного размера эти интервалы. Наличие экрана - есть, нет. Казалось бы, вот так интуитивно нам иногда кажется, что если вот наличие кнопок есть, вот написали, что есть. А если их нет, то ничего писать не надо. Это неправильно, потому что отсутствие кнопок - это тоже такой же признак, как их наличие. То есть надо писать и если они есть, и если их нет.

В результате мы эти шкалы, система эти шкалы использовала и сформировала обучающую выборку, которая выглядит следующим образом. Это три таблицы. Э-э, верхняя таблица содержит наименование объекта обучающей выборки. Оно может быть очень большим, ну, до 250 символов, по-моему, я делал. Ну, по крайней мере, тут подробно написано, откуда это взята информация. Левое нижнее окошко - это код классов в соответствии с классификационными шкалами и градациями, которые мы видели в режиме 2.1. Я ваше внимание обращаю на то, что здесь у нас э-э, каждый объект обучающей выборки относится к двум классам. То есть, по сути дела, это можно сказать так, два объекта. Один относится к этому классу, а другой - к этому. Если было бы три э-э, классификационных шкалы, тогда бы ещё третий код появился бы.

А вот здесь вот мы видим коды э-э, описательных шкал, которых много этих шкал восемь. И здесь вот коды их градаций перечислены для каждого объекта. Поскольку это таблицы баз данных, то они э-э, по размеру могут быть очень большими. Каждая из этих таблиц может быть размером до 2 ГБ. Если здесь будет 2 ГБ, то здесь несколько миллионов, около 5 млн строк получится, как минимум. Вот. То есть это что означает? Что э-э, на каждый объект выборки может приходиться очень много вот этих кодов, что их надо будет прокручивать, чтобы их рассмотреть. Это мы можем увидеть, когда будем тексты анализировать в системе. Сейчас пока что это не очень так сказать наглядно.

И э-э, другой вывод, э-э, другая форма вывода информации об обучающей выборке, которая очень похожа на экселевскую форму. Сейчас я вам её покажу. Экселевская форма. А вот у нас, э-э, так, экселевская форма. Вот так надо сделать. Вот здесь номер строки в Экселе, а здесь номер строки указан э-э, в этом файле. Здесь у нас классификационные шкалы жёлтым фоном выделены. Вот они есть. Описательные шкалы. Наименования все те же самые у них. Ну я не буду там прямо скрупулёзно всё выравнивать, но скажу, что, в принципе, она, эта форма, система Aidos, она точно такую же структуру имеет, как файл исходных данных. Больше того, здесь точно так же пронумерованы колонки. Вот. Вторая, третья, четвёртая, пятая. Первая, вторая, значит, э-э, ну, первая не нумеруется, она нулевая, потому что её нет в экселевском файле, а есть только сами, сами номера строк в экселевском, в Экселе. Вот, а нумеруются именно колонка А - первая, потом Б. Вот так они тут и идут. Первая, вторая, то есть вторая, третья - это классификационные шкалы, а с четвёртой по двенадцатую - описательные шкалы.

Но мы видим, что здесь у нас тексты находятся и числа, а здесь у нас только коды этих текстов и чисел в справочниках. И если какой-то текст повторяется здесь, то в справочнике он один раз встречается, а здесь коды повторяются. Это отношение один ко многим, первая нормальная форма, э-э, по сути дела, это означает, что обучающая выборка представляет собой нормализованную базу исходных данных. Вот эта вот база исходных данных, которая в Экселе мы видели, она нормализована. То есть все повторения исключены за счёт того, что разработаны справочники соответствующие и сделаны ссылочки э-э, здесь уже указаны в обучающей выборке не сами эти тексты и числа, а ссылки на эти справочники. С этого момента, ребята, дальше в системе Aidos числа и тексты обрабатываются совершенно одинаково. Что и позволяет мне обрабатывать и числа, и тексты.

Вот у нас текстовая шкала, а вот размер два - числовая шкала. Вот так вот, когда мы смотрим на обучающую выборку в закодированном виде, уже не догадаешься, где там текстовая шкала, где э-э, числовая. Вот.

**Раздел 7: Завершение и домашнее задание**

Дальше двигаемся.
Значит, теперь мы можем, когда у нас созданы, созданы э-э, справочники классификационных и описательных шкал и градаций, исходные данные с помощью них закодированы, получилась обучающая выборка, которая представляет собой нормализованную базу исходных данных. Теперь мы можем создать модели. И мы при этом э-э, выполняем тот порядок преобразования данных в информацию, а её в знание, который я вам рассказывал на лекциях. Данные преобразуются в информацию тогда, когда в этих данных формируется база событий на основе баз данных исходных. И потом выявляются причинно-следственные связи между событиями. То есть мы что сделали сейчас? Мы создали справочники прошлых и будущих событий, разработали классификационные описательные шкалы и градации и закодировали исходные данные с помощью этих справочников, получилось обучающая выборка или база событий.

Теперь нам нужно осуществить синтез и верификацию модели, то есть создать модели и проверить их на достоверность. Этим самым мы завершим преобразование исходных данных в информацию и завершим процесс анализа этих исходных данных. А потом мы выберем наиболее достоверную модель по ряду критериев, которые там мы рассмотрим. Основным из которых является критерий F-критерий Ван Рейсбергена. И решим ряд задач: задачи распознавания, идентификации, прогнозирования, принятия решений и исследования объекта моделирования путём исследования его модели. Вот такой у нас план, ребята.

Значит, теперь мы должны перейти к синтезу модели и проверки их на достоверность. Для этого мы запускаем режим 3.5.
Ребят, скажите, пожалуйста, мне, как вы вообще ещё там не выключились, нет? Вы воспринимаете, что я рассказываю?
Всё понятно.
Ну отлично. Я не повторяюсь? Я это не рассказывал вам раньше, нет?
Нет, не рассказывали.
Слава Богу! А то я так могу два раза или три раза рассказать, понимаете? Ну сейчас я, правда, решил вот здесь вот в расписании, прямо я пишу, что я рассказал и что мы будем изучать на следующем занятии. Поэтому э-э, больше я вас не буду спрашивать, что мы там прошли, что не прошли. Я и сам уже теперь буду знать это. По каждой группе, по каждой, по каждому занятию я сейчас это фиксирую эту информацию.

Значит, смотрим на эту экранную форму. Она довольно такая сложная, много здесь параметров есть. Значит, прежде всего, мы будем создавать статистические модели и системно-когнитивные модели или модели знаний. Второе, мы можем это сделать, используя центральный процессор или графический процессор. Я вам советую в данном случае, вот в данной работе использовать центральный процессор. Графический процессор, ребята, это процессор видеокарты, э-э, которая является многопроцессорной э-э, системой, видеокарта, матричного типа, причём, программируемого матричного типа. Ну то есть это что означает? Что можно рассматривать э-э, графический процессор как систему конвейеров, конвейеров. Вот. И тогда это э-э, мультипроцессорная конвейерная система. А можно рассматривать как вектора. И тогда это мультипроцессорная векторная система. А фактически она является универсальной и позволяет и тот, и тот вариант, и является просто матричной системой, мультипроцессорной.

Что такое конвейер? Конвейер - это когда один процессор что-то сделал с исходными данными, как-то их преобразовал и передал на следующий процессор. Тот продолжил этот процесс, и на выходе мы получили результат. Значит, система Aidos устроена наподобие конвейера, который преобразует исходные данные в информацию, а потом в знание. А потом на основе этих знаний решает ряд задач, которые и являются результатом обработки этих данных, информации и знаний и использования этих знаний. Вот.

А матричный процессор, э-э, вернее, векторный процессор - это процессор, который осуществляет одни и те же операции над над разными данными. То есть конвейер осуществляет разные операции над одними данными, которые преобразуются по ходу процесса, процессов в конвейере. Вот. А э-э, векторный процессор, он одна и та же программа, совершенно одинаково обрабатывает разные данные. Например, если мы перемножаем две матрицы, то в матрице произведений каждый элемент матрицы произведения рассчитывается совершенно одинаково. Это сумма произведений элементов соответствующих э-э, матрицы первой и второй, сомножителей. При этом сами эти элементы, строки-то и колонки разные. Ну то есть получается, что каждый элемент матрицы произведения рассчитывается одинаковым, одинаковой программой, но данные разные берутся. Это вот и есть векторный процессор.

Так вот, в самой простенькой даже э-э, видеокарте, э-э, скажем, ЖТ-240, в ней уже 96 шейдерских процессоров, и каждый мощнее, чем обычный наш центральный процессор в разы. Вот. А если там Nvidia 1.500 или 1.700, то там уже больше 4.000 шейдерских процессоров. И каждый из них уже там на порядок мощнее центрального. То есть это мощнейшая э-э, матричная э-э, мультипроцессорная система вычислительная.

Значит, э-э, есть разные виды, типы карт, видеокарт. Но вот э-э, фирма Nvidia, она первая догадалась, наверное, можно так предположить. Ну, по крайней мере, это проявилось на рынке, что можно сделать открытым процесс программирования графических процессоров. Открыла язык OpenGL, на котором это можно сделать. И появились системы программирования, скажем, Delphi, в которой можно программировать на языке OpenGL графический процессор. И там приходится продумывать самому потоки конвейерные, собирать данные на светофорах со всех потоков, объединять их там. То есть это довольно-таки сложный процесс, э-э, намного отличающийся от линейного программирования, когда просто вот от одного, от одной команды к другой идёт счётчик команд, он их выполняет. Там совершенно по-другому. Там может быть у нас десятки и даже сотни и тысячи этих счётчиков команд в разных конвейерах, потом они объединяются. Это эти конвейеры объединяются в группы конвейеров, пучки объединяются в вектора. В общем, там довольно сложная, может быть, организация этих параллельных вычислений.

Сейчас есть, насколько мне известно, это, может быть, я и ошибаюсь, может быть, уже, уже ошибаюсь. Но недавно было так, что только C# язык высокого уровня обеспечивал программирование графических процессоров без э-э, того, чтобы программист влезал во все эти детали, которые я сейчас описал. А работал на высоком уровне, а уже сама система программирования, компилятор э-э, этого языка сам уже анализирует э-э, алгоритмы программы и распараллеливает их и образует из них э-э, и конвейеры, и вектора, так что это считается за минимальное время на конкретном числе шейдерских процессоров. Вот. Значит, возможно, Питон скоро это будет обеспечивать, я не знаю. Но вот C# уже обеспечивает. Это огромнейшее преимущество этого языка перед другими. Ну, кроме как великолепнейшие, великолепные средства отладки там есть. Ну, значит, хочу что сказать? Что э-э, графический процессор позволяет вести расчёты в сотни и в тысячи раз быстрее, чем центральный в результате. Ну, я приведу пример. Я решал задачи э-э, вот в системе Aidos и м-м, видел, что от 200 раз ускорение, до 4.000 раз ускорение получается расчётов.

Да, кстати, модули соответствующие расчёта самих моделей и э-э, в модуле распознавания на графическом процессоре реализовал разработчик из Белоруссии, Дима Бандык, для системы Aidos. То есть он в качестве исходных данных использует базы данных системы Aidos обычные, стандартные. И в качестве выхода, результата работы графических этих программ, работающих на графическом процессоре, тоже получаются совершенно обычные, стандартные базы системы Aidos, те же самые, которые на центральном процессоре. Вот. И сама система Aidos не замечает, какой процессор мы использовали: или центральный, или графический, по результатам работы. Но, к сожалению, на графическом процессоре есть ошибка в расчёте модели PRC2. Но я подчёркиваю, не влияющая на э-э, достоверность этих моделей и не влияющая на результаты решения задач. Такая неважная, то есть ошибка, но не приводящая к каким-то таким тяжёлым последствиям. Вот. Но всё-таки, когда объясняешь, рассказываешь, лучше центральный процессор использовать.

Теперь следующий момент. Значит, если очень большой объём данных обрабатывается, а вы ещё туда, и это я сделал тогда, когда ещё не было расчётов на центральном процессоре, э-э, в системе Aidos, на графическом когда ещё не было, только были на центральном. Я тогда предусмотрел такую возможность, что мы осуществляем не синтез и верификацию модели, а только синтез или только верификацию. Почему? Потому что синтез модели осуществляется сравнительно быстро, даже на очень больших выборках. Ну, в пределах там 15-20 минут, там, часа, вот такие времена, затраты времени. А вот верификация может занять недели, месяцы, например. Если взять модель, в которой 700.000 э-э, объектов обучающей выборки, 700.000, а можно до миллиона брать. И взять, скажем, 400 классов, и э-э, объекты обучающей выборки описаны 11.000 признаков. Это совершенно реальные размерности, которые я решал задачи, и гораздо больше размерности.

И вот, значит, что получается? Чтобы проверить, насколько модель достоверна, мы распознаём объекты обучающей выборки и смотрим, сколько получилось у нас истинно положительных, истинно отрицательных, ложноположительных, ложноотрицательных решений. И потом по, используя меру, F-меру Ван Рейсбергена, чтобы оценить достоверность модели.

Что такое положительные, отрицательные решения? Это решения положительные, это когда уровень сходства объекта с классом выше нуля, а отрицательные, когда уровень сходства ниже нуля. Ну что такое истинное и ложное, понятно, наверное. Ну, в общем, смысл такой, я сейчас опишу все эти виды решений. Значит, если система, значит, система обязана или должна, желательно, чтобы она правильно относила объекты к тем классам, которым они относятся, и правильно не относила к тем, которым они не относятся. Значит, э-э, и она может ошибаться и в первом, и во втором случае. Значит, истинно положительное решение - это система правильно определяет, к какому классу относится объект, и тоже относится его к этому классу, то есть правильно определяет. Э-э, истинно отрицательное решение - система правильно определяет, к какому классу объект не относится, вот, и не относится его к этому классу.

А ложноположительное решение. Значит, объект не относится к классу, а система его относит. То есть уровень сходства больше нуля, то есть система думает, что этот объект относится к этому классу, а он к нему на самом деле не относится. Это называется ложное положительное решение или ложное срабатывание. Вот летит самолёт, делается запрос: свой, я свой самолёт. Он отвечает что-то этот самолёт. Система идентификации комплекса ПВО думает, что это не свой, что это чужой самолёт, и берёт его и сшибает. Вот. Представляете себе, что может произойти? Значит, это э-э, значит, э-э, ложноотрицательное решение. То есть система подумала, что он не относится к категории, к классу свой самолёт, а на самом деле он относится к нему. Понимаете? То есть это вообще-то такие очень опасные шуточки, когда такие ошибки, они могут приводить к очень серьёзным последствиям. Ну это я описал такой уже экстремальный прямо случай. Ну, что ж, это э-э, в принципе, может быть. Э-э, мы видели, как пролетел самолёт перед нашим самолётом в Сирии, и ракета перенаправилась и сбила наш самолёт, а не израильский истребитель, который проскочил перед его носом и, используя очень высокую манёвренность, улетел, а ракета перехватилась, э-э, перенавелась на наш и сшибла его самолёт-разведки радиокационной. Это было не так давно, в общем-то. Вот такие случаи бывают.

Так вот, э-э, что здесь предусмотрено в системе, когда мы хотим проверить достоверность модели очень высокой размерности? Мы проводим тогда только синтез, потом проводим только верификацию на подмножестве объектов. Я чаще всего пользуюсь таким вот вариантом, когда мы выбираем N случайных объектов. Ну здесь есть другие варианты. Каждый ный объект, копировать только текущий, копировать диапазон объектов. Вот. И можем здесь что сделать? Мы можем указать здесь, скажем, 100 объектов и провести только верификацию и по секундомеру заметить время, которое на это было затрачено. Можно было бы, конечно, компьютерные часы использовать, но я что-то как-то не использовал, потому что на самом деле времена эти довольно заметные. Ну, скажем, там 7 минут. Вот эти 100 верификация модели по 100 объектам, где 400 классов, и каждый объект 11.000 признаков описан. Чтобы его чтобы провести верификацию, мы организуем цикл по объектам, э-э, по которым будет осуществляться верификация модели, проверка её достоверности. Эти объекты копируются в распознаваемую выборку. Организуется по ним цикл с первого по сотый. И каждый объект сравнивается, потом организуется цикл по классам с первого по четырёхсотый. И потом, э-э, э-э, чтобы сравнить объект с каждым из этих классов, э-э, производится суммирование количества информации, которое содержится в 11.000 признаков. И этот процесс продолжается. И, допустим, эти 100 объектов верифицируется модель по 100 объектам э-э, случайным, выбранным случайным образом за 7 минут. После этого мы строим пропорцию, ребята. 100% э-э, верификация прошла 7 минут, а X объектов, которые мы хотим узнать, сколько надо их взять, будет идти э-э, ну, скажем, 8 часов. 8 часов умножить на 60 минут, получается 480 минут. Вот. И мы находим это X. У нас получается, скажем, э-э, ну, 300 там 30 или 200.000 объектов получается. То есть вместо 100 мы можем вот здесь указать теперь 200.000, запустить вечером, а утром получим довольно точную оценку достоверности модели. Этот подход, который я сейчас вам описал, называется бутстрепный метод, бутстреп. Бутстреп - это название на одном из скандинавских языков, по-моему, на шведском, название складного ножа. То есть мы одни лезвия, то есть нож - это вот у нас все данные. Одно лезвие вытаскиваем - это у нас обучающая выборка, другое лезвие вытаскиваем - это у нас распознаваемая выборка.

Мы можем либо не удалять распознаваемую выборку из обучающей, либо удалять. Значит, если не удалять, то те объекты, на основе которых будет проверяться достоверность модели, будут использоваться и для синтеза модели. Некоторые начинают говорить: "О, они используются для синтеза, они там повлияли на эту модель. Конечно, она теперь должна их обязательно распознать. Достоверность будет завышенная при такой проверке". Вот. А если удалять, тогда что получается? Тогда часть объектов, э-э, которая у нас была описана в обучающей выборке, используется для синтеза модели, а часть - для проверки её на достоверность. Но я думаю, что особого смысла удалять нет. Почему? Сейчас объясню. Если у нас 700.000 объектов обучающей выборки, то каждый из них влияет, э-э, даёт, так сказать, одну семисоттысячную влияния на то, какой будет модель сформирована. Если мы оттуда возьмём и удалим там, допустим, даже 50.000, то останется 650.000. Ну то есть, короче говоря, вот это влияние тех объектов, которые мы можем удалить для того, чтобы использовать их для тестирования модели, оно будет настолько малым, что мы этого не заметим там в семи знаках после запятой. Ну сейчас мы, конечно, синтез и верификацию осуществим, потому что модель маленькая.

Теперь смотрите, система оценивает, насколько э-э, полно будет задействованы её ресурсы для при решении данных задач, синтеза и верификации. И говорит нам, что сейчас вот эта модель, которую мы сейчас будем использовать, исследовать, она задействует шесть десятитысячных процента от того, что могла бы делать система Aidos. Ребят, вдумайтесь. Шесть десятитысячных процента. Вот. То есть это э-э, можно в сотни раз больше там э-э, модели исследовать. Если она напишет, что здесь вот у нас используется 120% ресурсов системы, то это означает, что она это не может решить эту задачу. То есть её ресурсов не хватает для этого. То есть её надо, надо систему в полтора раза более мощную, чтобы она это позволила сделать. Тогда мы вот здесь вот можем указать, ребята, 5% оставить от файла результатов распознавания. И тогда у нас э-э, получится там, допустим, 70% ресурсов используется. И тогда система решит эту задачу. При этом из файла результатов распознавания будут удалены те результаты, которые наименее достоверны, наиболее сомнительные, с наиболее низким уровнем сходства, близким к нулю. Чем ближе э-э, к нулю уровень сходства, тем чаще ошибки бывают, ложные решения и положительные, и отрицательные. Это известно из опыта и теоретически понятно.

Значит, сейчас мы э-э, нажимаем Окей. То есть я вам долго рассказывал всё это. Наверное, замучил уже.
Да, ребят, хочу вас попросить, э-э, когда будет 5 минут до конца занятия, напомните мне. Конец занятия будет у нас 18:50.

Вот мы видим, ребята, что у нас за 14 секунд все созданы модели статистические и системно-когнитивные. А потом в каждой из них решена задача э-э, верификации модели, проверки её на достоверность. И потом, после этого, э-э, созданы 11 выходных форм по результатам решения задачи идентификации.

Ну, конечно, нам интересно узнать, а что ж там за модели были созданы? А потом уже узнать, ну посмотреть на эти модели глазами, как говорится. А потом уже интересно оценить их достоверность. А потом уже решать различные задачи. Ну, я думаю, что вот это дальнейшее решать задачи, скорее всего, это на следующем занятии получится сделать. А сейчас мы можем посмотреть на сами модели, ребята. Смотрим режим 5.5.

Э-э, да, кстати, вот то, что я сейчас всё вам рассказываю, в диспетчере приложений в помощи это всё и есть. Всё это написано. Пункт шестой - посмотреть сформированные модели в режиме 5.5. Видите, да? Создать э-э, осуществить синтез и верификацию модели в режиме 3.5. Всё-всё вот это, что я сейчас делаю, всё это здесь описано в хелпе диспетчера приложений. То есть когда мы создаём приложение, там есть помощь, где прямо вот написано вот это всё, что я сейчас рассказываю. Ну, не так подробно, а написано просто, что это нужно сделать.

Смотрим режим 5.5. Мы видим, что в этом режиме у нас есть э-э, 10 моделей: три модели статистических и семь моделей системно-когнитивных. И есть помощь, которая описывает, как эти модели рассчитываются. Я потом к ней обращусь. Э-э, самая примитивная модель, э-э, которая первая рассчитывается, на основе неё всё остальное делается, - это модель абсолютных частот. Ещё её называют в статистике корреляционная матрица, вот эту модель.

Что она собой представляет? Значит, в системе Aidos эта модель выглядит следующим образом. Значит, э-э, колонки, то есть это некая матрица, таблица. Колонки, которые представляют собой классы, а строчки - значения свойства или значения факторов. А классы - это текущие или будущие состояния объекта моделирования. У нас две классификационных шкалы, поэтому каждый объект обучающей выборки относится к двум классам. Это означает, что в этой матрице есть результаты э-э, учёта этого объекта два раза: один раз по одной шкале, другой раз по другой шкале. А эти классы - это колоночки у нас. И у нас есть здесь две классификационных шкалы: конкретные классы, вот они, конкретный класс вешалка, конкретный класс клавиатура, и обобщающий класс, элемент компьютера, средства связи.

И у нас есть единички там и другие числа на пересечении строк, соответствующих значению свойства или значению фактора, и колонки какой-то, которая соответствует э-э, обобщающей категории, к которой относятся данные объекты, относятся объекты обучающей выборки. И в самом низу мы видим, сколько объектов обучающей выборки э-э, относились к тому или иному классу. И на основе какого числа объектов созданы эти вот э-э, получены эти числа, распределение свойств по классам объектов. И мы видим, что здесь есть и 1, и 2, и 3. И там, где вот обобщённые классы, там у нас и 7, и 4 есть объекта на класс. И мы видим, что всего-то у нас 40 объектов э-э, обучающей выборки. А если мы посмотрим на экселевский файл исходный, то там явно, что их не 40, а 20. Так же? 20 объектов. Ну, первая строка - это наименование. Кстати, наименование должно быть в одну строчку. И в этой строчке вот здесь вот не должно быть внутри наименования символов переноса строки или конца абзаца. Иначе они будут разбиваться на два, когда будут носиться в систему.

Теперь, давайте ещё я вам кое-что скажу, э-э, уровня ноу-хау насчёт этого файла. Давайте вот проверим его область значений. Я беру Ctrl+Shift и нажимаю End. Выделяется область значений. Мы видим, что здесь она выделена абсолютно правильно. То есть действительно, и колоночки выбраны только те, где есть значения, и строчки только те, где есть значения. Теперь я делаю такой вот э-э, э-э, такие операции провожу, которые вообще-то э-э, неправильные. Вот я взял, ошибочно ввёл сюда какой-то символ, а потом догадался, что я сделал это ошибочно, и убрал его. То же самое делаю в строке. Вношу символ и потом удаляю его. Теперь перехожу в первую, а, в колон, ну, в общем, в ячейку А1, нажимаю Shift+Ctrl, держу их и нажимаю End. У меня выделяется, смотрите, колонка М и двадцать вторая строка. Почему? Потому что теперь Excel думает, что я с ними работаю, понимаете? А зря. Почему? Потому что раз колонка есть вот такая, то это, значит, если мы сейчас попробуем в систему Aidos этот файл ввести и укажем там у нас диапазон колонок, диапазон строк, то она нам напишет, что вот в этой колонке у нас нет наименования. А потом позже напишет, что все значения в этой колонке одинаковые, нет вариабельности. А вот в этой колонке, смотрите, ребята, где числа, у нас появится текст, пустая колонка с пробелом. Это означает, что теперь эти числа будут использоваться системой Aidos не как числа, а как тексты некоторые. И она будет уникальные находить тексты, кодировать их, как вот она с текстами работает.

Ну чтобы ничего этого не было, что нужно сделать? Берём, выделяем область данных. Если всё нормально, ну и слава Богу. Если нет, тогда мы выделяем искусственно, ограничиваем только фактическую область данных, копируем её в буфер обмена, создаём пустой листик, пустой листик, и вставляем. После этого выделяем всю страницу, выравниваем э-э, ширину колоночек. А ту страницу, которая у нас была э-э, не совсем хорошая, мы её удаляем этот листик. И теперь у нас получается чистенький, прекрасный листик, лист с исходными данными, который никаких проблем не вызовет в системе. Вот.

Теперь дальше мы э-э, это выполнили. Мы посмотрели на эту модель, и нам нужно ответить на вопрос. Вот, допустим, э-э, для средства связи, какой признак более характерный, а какой менее характерный? Ну, смотрим, вот размер под руку. Все три средства связи, которые у нас были, а их было три, они все три обладают этим признаком. А вот из семи элементов компьютера только три обладают этим признаком. То есть в 100% случаев у средств связи есть этот признак, а у элементов компьютера только в 43% случаев, 3/7. Это означает, что этот признак гораздо более характерен для средств связи, чем для элементов компьютера.

Насколько более характерен? В этом вопросе нам поможет следующая модель, которая называется модель относительных частот. Давайте её посмотрим. Вот она. Мы видим здесь 100%. А здесь 42,9, почти 43%. То есть мы можем сказать, что в 2,3 раза чаще этот признак наблюдается у средств связи, чем у элементов компьютера.

Ну вот это мы изучили с вами режим 5.5, который является всего лишь вспомогательным режимом для просмотра того, что нам насчитали режимы 3.5.

**Раздел 8: Оценка достоверности моделей (Режим 4.1)**

Теперь посмотрим режим 4.1, который позволяет нам оценить достоверность моделей.
Значит, в левом верхнем углу табличка, в которой ранжированы, упорядочены модели по убыванию их достоверности. И мы видим, что наиболее достоверная модель - это ДСМ-метод автоматического порождения правил Квайна-МакКласки. Она распознаёт 97,5% объектов обучающей выборки. А на втором месте - метод просто Квайна, который распознаёт 90% объектов обучающей выборки. А метод средневзвешенного Байеса - всего 62,5% объектов обучающей выборки. Ну и так далее. То есть мы видим, что эта модель лучше всех.

Если мы посмотрим на другие критерии, например, F-критерий Ван Рейсбергена, то мы увидим, что здесь у нас достоверность этой модели, э-э, она, э-э, 0,839 тысячных. То есть она не 97,5%, а 84%. Почему? Потому что F-критерий Ван Рейсбергена, он является средним гармоническим между полнотой и точностью. То есть он учитывает не только то, как правильно относятся объекты к своим классам, но и то, как они не относятся к чужим классам. И поэтому он является более объективным критерием, чем критерий процента распознавания объектов обучающей выборки.

Дальше. Смотрим на эту вот карту классификации. Здесь у нас э-э, оси - это классы. У нас их всего 14 классов. И мы можем посмотреть, какой объект, как распознаётся. Вот, допустим, смотрим на ноутбук. Ноутбук - это элемент компьютера. И вот здесь написано, как он распознался. По модели абсолютных частот он не распознался. По модели относительных частот он распознался как элемент компьютера, но с уровнем сходства 0,123. Это близко к нулю. Поэтому он мог бы быть и не распознан. Но положительное значение, поэтому он распознан правильно. А по методу Байеса, средневзвешенного, он тоже распознан правильно, но с уровнем сходства ещё меньше, 0,046. Тоже близко к нулю. А по методу Квайна он распознан с уровнем сходства 0,286. Это уже заметно больше нуля. А по методу ДСМ-методу Квайна-МакКласки, он распознан с уровнем сходства 0,488. То есть почти 0,5. Это уже довольно-таки уверенно он распознан.

То есть мы можем посмотреть, как каждый объект распознаётся каждой моделью. И это позволяет нам понять, почему та или иная модель лучше или хуже. Ну, допустим, смотрим на модель абсолютных частот. Мы видим, что она многие объекты не распознаёт. Видите, нули. То есть она не может принять решение, к какому классу отнести объект. Это означает, что эта модель очень такая, ну, скажем так, осторожная. Она лучше промолчит, чем ошибётся. Вот. А вот модель ДСМ-метода Квайна-МакКласки, она распознаёт все объекты, кроме одного. Вот здесь вот ноль. Это означает, что этот объект она не смогла распознать. Но зато она не ошиблась ни разу. То есть у неё нет ложных решений. Все решения, которые она приняла, они все правильные. Вот.

А вот метод Квайна, он распознал тоже все объекты, кроме этого. Но при этом он ошибся. Вот здесь вот, смотрите, э-э, флешка, она относится к классу носителя информации, а он её распознал как элемент компьютера. То есть он ошибся. И ещё он ошибся вот здесь. Э-э, записная книжка - это канцелярский товар, а он её распознал как средство связи. Тоже ошибся. То есть у него две ошибки. И один пропуск. А у этого метода ДСМ-метода Квайна-МакКласки только один пропуск, а ошибок нет. Поэтому он и лучше.

Вот так вот мы можем анализировать результаты работы моделей и выбирать наилучшую модель.

**Раздел 9: Завершение**

Ну вот, собственно говоря, и всё, что я хотел вам рассказать сегодня. Пожалуйста, ваши вопросы.

Пожалуйста, вам нужно какое-то отчёт высылать по поводу? Нет, нет, нет, не нужно. Вы это раньше было нужно, теперь вроде говорят, не нужно. Всё, понимаете? Вот всё. То есть вот то, что я так долго вам всё объяснял, это я объяснял смысл этого того, что мы делаем, смысл этих понятий там и так далее. А если просто нажимать на кнопочки, то вот всё, это минута буквально. Понимаете? Вот. Ну ещё не нажал на кнопочку 5.5, но вы поняли, что это, собственно говоря, она помелькала, помелькала. Если бы я вот так вот вам показывал, как я сейчас показал вот в конце занятия, то вы бы ничего вообще не поняли, что я делаю. Я всё объяснил вам.

Теперь в последующем вы будете это делать уже осмысленно, вы будете понимать, что вы делаете. А делать будете быстро. Привет, Аня! Рад тебя видеть. Или, вернее, видеть твоё имя и фамилию. Аня, ты что там? А! Привет.

Ну ладно. Значит, ну на этом конец занятия, ребят. Всего самого хорошего вам.
До свидания. До следующего занятия.