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

***Российская Федерация***

**110 Интеллектуальные информационные системы и технологии.**

**Лабораторная 3. 2020-09-30**

Заголовок: Обзор системы Эйдос: Лабораторные работы, типы данных и ускорение на GPU

Резюме текста:

Занятие посвящено лабораторной работе №3 по дисциплине "Интеллектуальные информационные системы и технологии" для группы ИТ-1823. В начале обсуждается установка системы Эйдос, проверяется успешность установки у студентов и разбираются типичные проблемы (отсутствие папки Aidata или файлов в ней, необходимость полной инсталляции, проблемы с путями, содержащими кириллицу или пробелы).

Далее дается подробный обзор типов лабораторных работ в системе Эйдос:

Локальные работы (4 типа):

Тип 1: Полностью готовые, все файлы присутствуют.

Тип 2: Исходные данные генерируются расчетным путем.

Тип 3: Данные импортируются из внешних источников (текст, таблицы); считаются наиболее ценными для практики.

Тип 4: Парсинг данных с сайтов (веб-скрейпинг).

Облачные приложения Эйдос: Загружаются с FTP-сервера системы, могут представлять собой готовые проекты (лабораторные, курсовые, научные работы, ВКР).

Обсуждается взаимодействие системы Эйдос с внешними ресурсами:

Веб-сайт (LC.kubagro.ru): Используется для скачивания инсталляции, обновлений и как источник файлов помощи (учебных пособий).

FTP-сервер: Хранит базу данных запусков системы, облачные приложения, инсталляционные файлы и обновления. Позволяет скачивать и устанавливать облачные приложения, обсуждать их.

Затрагиваются теоретические аспекты работы системы:

Преобразование данных в информацию и знания.

Роль программных интерфейсов (API) для конвертации данных (Excel, текст в формат базы данных Эйдос).

Работа с различными типами шкал (текстовые: номинальные, порядковые; числовые: отношений).

Статическая (свойства, классы) и динамическая (факторы, последствия) интерпретация данных.

Основы онтологий и фреймового представления знаний.

Особое внимание уделяется возможности ускорения вычислений с помощью графического процессора (GPU):

Система Эйдос может использовать видеокарты Nvidia для расчетов (синтез моделей, распознавание).

Это дает значительное ускорение (в сотни и тысячи раз) по сравнению с центральным процессором.

Упоминаются специальные модули для вычислений на GPU.

В конце даются практические указания по выполнению лабораторной работы №3:

Установка работы через режим 1.3 (Диспетчер приложений).

Знакомство с интерфейсом ввода данных (API 2.3.2.2), его параметрами и структурой исходного файла Excel.

Объясняется процесс формализации предметной области: создание классификационных и описательных шкал, кодирование данных.

Кратко упоминается следующий этап – синтез и верификация моделей (режим 3.5), который будет рассмотрен на следующем занятии.

Детальная расшифровка текста:

1. Введение и организационные моменты

Ведём запись. Да, три человека уже. Это круто. Ну, на самом деле побольше даже. Да, побольше уже, уже целая группа. Да.

Ну что, ребята. Значит, у нас сегодня 30 сентября, последний день сентября.

2. Обзор лабораторной работы №3

И на первой паре, которая с 8:00 до 9:30, лабораторная работа номер три в группе ИТ-1823 по дисциплине информационно-интеллектуальные информационные системы и технологии.

3. Установка и настройка системы Эйдос

3.1. Проверка установки и повторение

На прошлой работе в этой группе мы что делали? Ребят, напоминайте мне, потому что тогда, когда я проводил её, это было 25 сентября. Я тогда ещё не писал, на чём мы остановились. Ну, наверное, надо устанавливать лабораторную работу, я так понимаю, и изучать её.

Ребят, кто у вас тут староста есть, нет? Группы? Ну не староста, кто-нибудь есть, кто помнит, на чём мы остановились на лабораторной работе?

Ребята, ответьте. Установка Эйдоса. Да. Ну и как у вас получилось? В чате это, в чате ответьте, у кого получилось. Да, здравствуйте, Денис. Значит, плюсики поставьте, у кого не получилось. Или у кого, давайте плюсики, у кого получилось, а минусы, у кого не получилось. Давайте в чате.

Ну так, более-менее. А остальные просто не слышат, что я говорю, да? Ни плюсиков, ни минусиков не ставят. Ну, наверное, надо ещё задать, а кто, а кто не знает, что такое система Эйдос, поставить нолик, да? Просто некоторые отвечают плюсики, а те, кто минусики, никто не ничего не ответил. Ну да, не густо.

Ну сейчас же лабораторная работа, то есть вы должны на своих компьютерах повторять то, что я делаю. Вот в чём всё дело. Понимаете? И на лабораторной работе вы вполне можете задавать вопросы какие-то. Вдруг у вас какие-то возникли проблемы. Пожалуйста.

3.2. Решение проблем с установкой

Во. У Виктора не получилось. А что у вас не получилось, Виктор? Что там выдаётся, какое сообщение? Ионов. Что в папке Aidata?

А. Ну это элементарно. Там должно быть написано, что делать. А путь на, путь на систему какой у вас? Главная папка, где система, какой путь на неё? Там не должно быть кириллицы и пробелов.

В общем, надо скачать полную версию. Ну, вроде нормально. Полную версию надо скачать, ребят. Потому что там папка-то есть, а вот в ней может быть пусто. То есть там есть даже поддиректории есть, а файлов нет в поддиректориях. То есть вы могли скачать минимальную инсталляцию и там нет лабораторных работ. Скачайте, ну, заново полную инсталляцию.

И в папках, нет, в папке там в лабораторных работах, там ещё есть папки в самих лабораторных работах, а вот в них уже там файлы. Хм. Лучше пользуйтесь микрофоном, чем чатом, потому что чатом не очень удобно мне.

Я зашёл в эти папки, ну, в сами лабораторные, и там есть эти файлики. Мм. Тогда попробуйте режим 1.11 запустить. То есть запускаете систему и режим 1.11.

Значит, сейчас, ребят, старайтесь вообще старт Эйдос запускать по возможности. Иногда он выдаёт ошибку на некоторых компьютерах. Тогда в этом случае тогда уже сам исполнимый модуль запускаете.

4. Обзор типов лабораторных работ и структуры системы Эйдос

4.1. Классификация лабораторных работ

Значит, у нас тогда сейчас такая тема нашей лабораторной работы. Я вам рассказываю, какие есть лабораторные работы, как они классифицируются и как они устанавливаются. И после этого мы начинаем проходить лабораторную работу 3.03. И про неё я вам всё подробно объясняю про эту лабораторную работу.

Я вам ещё не рассказывал про классификацию лабораторных работ? Что в режиме 1.3 устанавливаются лабораторные работы, что есть встроенные. Это я рассказывал или нет? А есть облачные. Вспоминаете, нет? Или нет?

Да или нет? Не расслышал я, вы знаете, не очень-то слышно. Скажите, мы это проходили или нет? Не рассказывали, не рассказывали. Понятно. Значит, ну тогда давайте начну с этого, что буду вам рассказывать.

Вот. Советую полную версию скачать и заново развернуть в корневом диске. И тогда уже смотреть, что там получается, что не получается.

Вот, ребята. Значит, в режиме, режим 1.3 – это диспетчер приложений, он предназначен для установки лабораторных работ. И для записи в облако ещё он позволяет записывать приложения в облако.

Значит, если мы… Значит, лабораторные работы есть двух типов: локальные – это не требуют интернета для изучения. Ну, вообще-то интернет всегда желателен, потому что хелп лабораторных работ всё равно всё-таки на моём сайте находится. Вот. Но всё-таки работы будут работать без интернета. И облачные Эйдос-приложения, которые можно скачать с FTP-сервера системы, нажав вот кнопочку "Скачать приложение из облака".

4.2. Локальные лабораторные работы (4 типа)

Значит, локальные лабораторные работы есть четырёх типов. Вот я сейчас нажал на кнопочку, кликнул, которая жирным шрифтом "Добавить лабораторную работу". И здесь есть четыре типа лабораторных работ: первый, второй, третий и четвёртый.

Первый тип. Эти лабораторные работы полностью готовы для изучения, все файлы там есть. Когда мы выбираем какую-то из этих работ, то создаётся пустое приложение сначала, и потом туда копируются все файлы из соответствующей лабораторной работы. Они находятся в папочке Aidata. Сами эти источники, откуда копируются, копируются лабораторные работы. И там есть папочка LabWorks. И в ней, как вы видите, 30 поддиректорий. И с первой по десятую там находятся работы полностью уже… Что-то вообще пусто. А, ну, наверное, как раз у меня тот самый случай, как что надо полную версию установить. Сейчас бы… Ну, установка происходит достаточно быстро. Кстати, вчера я новую версию выставил. Сделал там ряд исправлений. Когда вот я вам рассказываю, показываю, то замечаю всякие нюансы, которые я, в общем, мне не нравятся, я их исправил.

Значит, смотрите, ребята. Значит, сейчас система версии 30 сентября, видите? 30.09. То есть я её ночью записал, когда уже было тридцатое число. Теперь смотрите, где находятся лабораторные работы. Вот папочка системы, в ней есть папочка Aidata. Ну, это база данных. LabWorks. А там уже вот сами работы находятся, видите? Первые 10. Значит, сейчас мы посмотрим, как они называются. Значит, это всё видно в режиме 1.3. Я их все режимы пронумеровал, чтобы удобнее было называть. Добавить лабораторную работу. Вот когда мы переключаем этот переключатель, то меняются названия работ. Вот. Первого типа работы полностью готовы.

Второго типа. Они, э-э, скажем так, э-э, никаких данных там в этой папочке соответствующих лабораторных работ нет, кроме названия. А исходные данные для выполнения этой работы формируются расчётным путём. То есть вообще не требует ничего, э-э, для того, чтобы её изучать.

Работа третьего типа. Устанавливается путём ввода данных из внешних баз данных, из внешних источников данных.

И четвёртого типа – это, если вы такой термин слышали, может быть, парсинг. То есть это обращение к сайту и скачивание информации с сайта либо по HTTP протоколу, либо по FTP. Система Эйдос обеспечивает и то, и то.

Когда-то раньше я изучал работы именно в таком порядке со студентами, очень давно. Эту дисциплину преподаю уже 18 лет в пожарном университете, можете себе представить. Вот эти первые работы, они как раз были сделаны и изучались в 2002, 2003 году, вот так вот. Тогда этой системы не было, которую сейчас мы используем, она с 2011 года появилась. А была ДОСовская версия. И в этой ДОСовской версии эти работы были, ребята. То есть именно вот эти работы, которые сейчас у нас в первом пункте. Вот. И, может быть, и вторые тоже, второго типа. И третьего типа тогда были работы тоже. Так вот, работы третьего типа, ребята, на мой взгляд, это самые ценные работы, потому что все наши э-э, исследования, которые мы сами проводим, они основаны на каких-то источниках данных, которые могут быть в виде текстов, в виде графических объектов и в виде таблиц. Вот. И как раз это, этот тип лабораторных работ э-э, учит нас, как с ними обращаться с этими внешними таблицами, текстами. Ну, с графикой здесь, правда, нет работ. Графика – это более сложный вопрос, и я его на уровне лабораторных работ не излагаю. Но есть очень много публикаций по этому поводу. Ну, чтобы не отходить далеко, раз уж я упомянул, сейчас сразу вам эти публикации покажу. Вот. Второй пункт моего сайта. Выходим, скачать, запустить систему Эйдос. И если пролистать, тут структура системы, монографии, пособия, а потом тематические подборки публикаций. И вот здесь вот и есть у нас публикации по обработке изображений. Довольно много статей у меня, связанных с этим, и учебные пособия есть. Вот. В чате, в чате посылаю вам ссылочку. Кто заинтересуется возможностями обработки графики средствами Эйдос, пожалуйста, вот я дал ссылочку вам. Ну, работы разных периодов времени. И 2009 год, и 2019 год. Можете посмотреть. 2015. В общем, много разных работ разного периода времени.

4.3. Облачные приложения Эйдос и взаимодействие с внешними ресурсами

И ещё есть работы, я их называю облачные Эйдос-приложения. Ну, система Эйдос, то есть система Эйдос – это не просто программа, которая работает на компьютере, она взаимодействует с моим сайтом широко, который вот я показывал вам, lc.kubagro.ru. Почему? Потому что на этом сайте, во-первых, сама инсталляция системы находится, обновления на нём находятся системы, отсюда они именно скачиваются. И третье, на нём находятся учебные пособия, разделы которых являются хелпами к лабораторным работам. Вот. Ну, там размещаю, я обращаюсь не к этому учебному пособию 2004 года, а вот к этим вот зелёненьким. Здесь есть два тома. Первый том – это лекции, а второй – лабораторный практикум. И там есть лабораторные работы описанные. И прямо эти описанные работы, эти описания являются хелпами соответствующим работам. Прямо вот по этому адресу они запускаются, загружаются вот и отображаются. То есть интернет нужен в этом для этого.

И второй сайт, который использует система Эйдос – это FTP-сервер. На нём, э-э, какие у него функции? Две функции основных. Ну, сразу вам покажу, не отходя, так сказать, далеко. Смотрите, вот это мой сайт по FTP доступу я сейчас открыл. И вот, видите, здесь инсталляции и обновления системы. Вот, файлы обновлений. Вот. И если мы посмотрим здесь вот, тут есть инсталляция. Вот это самораспаковывающийся архив, максимальная инсталляция, обычный архив и минимальная инсталляция без лабораторных работ.

И можем посмотреть FTP-сервер. У него две функции основных.

Первая функция. На нём есть база данных текстовая, в которой э-э, фиксируется вся информация о запусках системы Эйдос с декабря 2016 года. Первое посещение из Соединённых Штатов, Айдахо. Вот, Бойс. Вот. Потом из России, потом из Словакии, потом опять из США, потом из Германии. Вот. Смотрите, вот сейчас. Сегодня тридцатое число. Вот уже сегодня у нас семь запусков. Краснодарский край, Адыгея. Подозреваю, что, возможно, это вы и есть. Сейчас мы даже посмотрим, можем, где вы там есть. И здесь вот все, все посещения системы Эйдос, они здесь все фиксируются в этом файле. Ну, потом у нас будет занятие, где я расскажу, как это делается.

А вот в этой папочке у нас есть ещё поддиректории. И в этих поддиректориях у нас э-э, то, что я называю облачное Эйдос-приложение. То есть это папочки, в которых есть исходные данные, есть описание приложения. Вот. Ну и есть некоторые ещё служебные файлы, которые нужны для работы, которые я сейчас покажу в работе, зачем они нужны.

Значит, смотрите. Значит, мы можем открыть, кликнуть по кнопочке "Скачать приложение из облака". У нас скачивается каталог этих приложений. Естественно, там есть каталог. И сейчас мы видим, что их 207 на данный момент. Но они сюда могут добавляться без программирования. И вот здесь есть ссылочка на эти приложения, на описание этих приложений. Чтобы эта ссылка активировалась, чтобы запустился, запустилось соответствующее описание, открылось, значит, для этого нужно всего лишь два раза кликнуть по строчке. При этом в браузере по умолчанию открывается описание данного облачного приложения. Вот. Этот файлик заменю, потому что сейчас есть получше этот. Вот. Ну в общем, открывается описание. Всё там есть. И можно в этом, э-э… Это 205 приложение. А можно просто кликнуть "Установка Эйдос-приложения", тогда будут скачаны все эти файлы из приложения, будут скачаны. Можно кликнуть "Каталог обсуждений". Тогда мы увидим, что, когда делалось э-э, с этим облачным приложением. Вот, допустим, в сентябре 2018 года было сообщение появилось в семидесятом приложении. Ну так вот мы можем узнать, когда там появлялись сообщения и когда они скачивались эти приложения. Мы это видим. Мы можем на любое приложение выйти, обсуждение его. Вот. И нажать кнопочку "Обсуждение приложения". У нас появится возможность здесь что-то написать, ответить, спросить. Можно получить ссылочки на файлы приложения. Это абсолютные гиперссылки, которые можно использовать в строку адреса его указывать эту ссылочку. И будет соответствующий файл загружаться. Вот мы видим, как это выглядит.

В общем, эти вот ссылки – это прямо правильные ссылки на эти файлы. И эти файлы именно и загружаются, если мы это приложение решим загрузить. Таким образом, мы видим, что здесь есть сами приложения, вернее, исходные файлы для них. Есть каталог, показывающий нам, что там происходит с этими приложениями. И есть возможность обсуждения каждого из приложений. Можно его это приложение просто почитать, что там есть, э-э, что там входит туда. А можно и э-э, установить его и изучать. Вот, собственно, основные возможности, какие есть с лабораторными работами. Конечно, вот эти облачные приложения, они не только уровня лабораторной работы. Тут есть и курсовые работы, и научные работы, и лабораторные работы, и даже есть выпускные квалификационные работы. То есть они разного уровня. Вот. Ну а довольно-таки научно-исследовательских работ. То есть это какие-то исследования. Ну я иногда писал лабораторные работы, иногда НИР. Вот ВКР, например, видите, ВКР, ВКР, ВКР. Это вообще э-э, такие серьёзные работы уровня выпускной квалификационной работы. Причём они из нескольких университетов, не только из нашего.

Вот такие вот дела, ребята.

5. Практические шаги по лабораторной работе №3

5.1. Установка работы 3.03

Такие вот дела. И теперь, э-э, сейчас я, пожалуй, не знаю, стоит ли это сейчас делать. Есть некое сомнение, но попробую. Ну, наверное, не буду я сейчас это делать. В общем, есть описание получше вот этой 205 работы, я загружу получше описание.

Вот. Ну теперь чем займёмся? Значит, давайте установим первую работу, которую мы изучаем сейчас. Сейчас я начинаю изучение прямо с работы 3.03. То есть работа третьего типа, третья лабораторная работа. Установить её очень просто. Просто вот в режиме 1.3 "Установить лабораторную работу", выбираем какую, какую именно. Можно несколько выбрать, кстати, но смысла особого нет, потому что всё равно по очереди их проходим мы. Вот. И просто нажимаем О'кей.

5.2. Интерфейс ввода данных (API 2.3.2.2) и структура исходного файла

Появляется предупреждение, что вот в этой папке EidosX\Aidata\Indata будут стёрты все файлы. Что это за папка? Это папка, которая по умолчанию предназначена для исходных данных. То есть когда устанавливается лабораторная работа, то система создаёт сначала пустое приложение. Вот. То есть создаётся папочка вот здесь для приложения. А потом в это приложение, потом э-э, в папку Indata копируются исходные данные, что там было, всё стирается. Вот. И э-э, потом уже запускается программный интерфейс ввода данных из этих внешних источников данных. Ну, нажимаем О'кей здесь. Значит, для чего это нужно сообщение? Если тут мы что-то делали э-э, с с исходными файлами, с файлами исходных данных, с описанием работы, мало ли, что мы здесь делали, то нас система предупреждает, что сейчас это всё будет стёрто.

А вот появилась папочка пустого приложения с начальными файлами. В начале их довольно мало этих файлов, это вот минимальный набор файлов. Что сюда входит? Сюда входят классификационные описательные шкалы и градации, обучающая выборка. Но все эти базы данных пустые. Да, кстати, они открываются в Экселе эти базы данных. Видите, стоит X. Ну я ассоциировал Excel с файлами стандарта DBF, DBM. И он спокойно их открывает абсолютно. Чем мы и будем пользоваться.

Вот. Появляется э-э, окошко программного интерфейса, которое на первый взгляд такое довольно сложное, много разных параметров. И это вот место, начиная с которого требуются пояснения. То есть здесь я останавливаюсь и начинаю вам рассказывать, какие требования к файлу исходных данных и так далее.

Нужно сказать, что в системе Эйдос есть много автоматизированных программных интерфейсов с различными типами внешних данных. Автоматизированный программный интерфейс (Automatic Program Interface, API) сокращённо называется. API, наверное, вы этот термин слышали. Есть API Windows, есть API там каких-то программных систем иногда. Значит, что это такое? Это программы, которые предназначены для преобразования стандарта представления данных в каких-то файлах. То есть вот, допустим, были файлы в одном стандарте представления, они конвертируются в другой стандарт представления. Вот, это делает программный интерфейс. В принципе, все конвертеры представляют собой API. Ну то есть, если мы, допустим, преобразуем Вордовский файл в PDF, то это API, значит, уже. Понятно, да? Или наоборот. Но в системе Эйдос есть программные интерфейсы, которые встроены прямо в саму систему, реализованы на языке программирования, на котором она написана, которые преобразуют экселевские файлы в DBF, текстовые файлы тоже в DBF. Ну, DBF – это просто стандарт баз данных этого языка программирования. И графические даже файлы тоже э-э, анализируются средствами языка, который содержит развитую графическую подсистему, э-э, в которой есть всё, что только можно вообразить. Вот. И то, чего нельзя вообразить, тоже там есть. И всё это, значит, э-э, анализируются графические объекты. Ну я, допустим, спектральный анализ их провожу, э-э, контуры там выявляю, сравниваю их по спектрам, по контурам и просто по пикселям. И так далее, и так далее. То есть там можно, в общем-то, э-э, различные операции с с графикой осуществлять. Вот. Ну и рисовать можно тоже примитивы есть там, команды рисующие точки, линии, всякие фигуры, заливки там и так далее.

Так вот, э-э, можно анализировать графические объекты, тоже создавать э-э, базы данных, в которых могут как сами эти объекты содержаться, так и их описание. Э-э, ну, допустим, значение пикселей, например, или значение там, которое позволяет восстановить контур.

И вот благодаря наличию этих вот автоматизированных программных интерфейсов, у системы очень широкая область применения. Потому что все эти типы данных, они охватывают очень широкую предметную область, в которой можно применять систему.

Ну и сейчас могу вам сказать, что есть два стандарта ввода данных из табличных данных. Вот этот стандарт, который сейчас у нас используется. Значит, его надо описать, для чтобы вы поняли, как организованы исходные данные. Тогда вы будете понимать вот этот смысл всех этих параметров.

Значит, э-э, требования к файлу исходных данных этого стандарта API 2.3.2.2. Здесь вот есть в хелпе. Здесь я довольно долго рассказываю, это читать, может быть, смысла нет, там так довольно занудно написано. Ну, в общем, могу вам сказать вот что, что файл исходных данных представляет собой экселевскую, экселевскую таблицу в старом или новом, э-э, Экселе. Но желательно не не позднее, ну, в общем, 2016, 2019 не подходят. Потому что там почему-то отсутствуют или изменены драйвера, преобразующие вот экселевские файлы в DBF. А система их использует. То есть, э-э, на компьютере должен быть установлен Excel, начиная с 2003 по 2010 год версия. Более поздние, там могут вопросы возникать. И э-э, никакой там Open Office, а именно вот э-э, Microsoft Excel, Microsoft Office.

Значит, файл исходных данных имеет следующую структуру. Каждая строчка этого файла является наблюдением. Ну, то есть содержит информацию об одном наблюдении. Система обрабатывает эмпирические данные. Вот эти строчки – это и есть вот как раз э-э, информация о наблюдениях. Первая колонка содержит информацию о том, где, когда и что наблюдалось для самого исследователя. То есть вот, допустим, если мы хотим изучить, как влияют агротехнологии и факторы окружающей среды на количественные и качественные и стоимостные показатели финансово-экономические показатели результатов выращивания пшеницы, то мы здесь вот описываем, какие поля у нас в базе данных есть. Вот поле номер такой-то, в таком-то хозяйстве, в таком-то году, там выращивалась пшеница такого-то сорта. Вот здесь уже, там, где зелёного фона колоночки, там мы пишем, какого сорт, допустим, пшеницы. И потом пишем, какие применялись технологии. Какая вспашка, какой высев, какие нормы высева, способ высева, какие применялись э-э, средства полива, удобрения, средства защиты, всё-всё-всё здесь описываем. Значит, каждая колоночка содержит э-э, предназначена для описания одного фактора какого-то. А в значениях, в самих строках, там значение этого фактора. Ну, допустим, если мы здесь колоночка у нас э-э, дозы полива, то в колоночке будет написано, сколько кубометров на гектар, например. Понятно, да? И вот так вот все эти колоночки. Если, допустим, способ вспашки, то там может быть колоночка, допустим, глубина вспашки, и там глубина там 5 см, 10, 15 см. А также может быть колоночка способ вспашки. Там плуг, там отвальная, там дисковая, там, ну то есть какие какие-то будут способы. Я в них не слишком разбираюсь, но можно их тут указывать.

И вот так вот мы описываем все технологии, которые применялись. А слева мы описываем результаты, какие получились. Результаты описываются вот здесь, где жёлтым фоном колоночки. Описываются обычно в натуральном выражении и в стоимостном выражении. В натуральном выражении результаты – это количество продукции и качество продукции, прежде всего. Ну, могут быть также э-э, потребительские свойства, товарные свойства продукции. Потребительские – это насколько она приятна покупателю на вид. Вот. А и вкус там, допустим, цвет, запах, вот такие показатели, общий вид внешний. А технические свойства – это насколько хорошо она хранится, насколько хорошо перерабатывается при изготовлении там продукции, как из сырья вот это из этой продукции. Ну и так далее, и так далее. То есть как она в технологиях себя проявляет эта продукция.

И вот таких наблюдений, ну, в новой версии Экселя, э-э, начиная с 2007 года, э-э, уже может быть до миллиона наблюдений. А колоночек всех вместе взятых до 16.000. Ну я вам скажу так, по-простому говоря, что этого более чем достаточно, чтобы описать даже очень большие задачи, очень большой размерности. То есть 16.000 факторов, ну там, может быть, не 16, а скажем, там 12.000 факторов. И, допустим, 500 результатов выращивания. Этого более чем достаточно, чтобы описать даже очень сложные ситуации, очень сложные задачи.

Вот. И, собственно говоря, это так вкратце я всё рассказал. Теперь более подробно. Значит, э-э, значение в колонках может быть либо текстовое, либо числовое. Текстовое – это просто что-то там может быть написано, причём на любом языке, абсолютно без разницы, на каком языке. Или с какими с помощью каких-то кодов или случайных этих э-э, шифров, символов каких-то, может быть, написано, понятное только самому тому, кто разрабатывает это приложение. Вот. Ну, числа просто в числах какие-то могут быть. Значит, если шкала, значит, содержит, э-э, вернее, если колонка содержит какое-то текстовое значение, то она вся считается текстового типа. Тогда числа считаются тоже текстами. Вот. А если там указаны числа, а текстов никаких нет, э-э, которые не являются числами. Числа тоже являются текстами, но текст – это более широкое понятие, чем числа. Бывают тексты и не числа. Так вот, если там есть тексты не числа, то то числа как тексты рассматриваются. А если там только числа, то тогда э-э, обрабатывается следующим образом эта колонка. Э-э, находятся минимальное и максимальное значение. Это делается путём индексирования, то есть очень быстро, логического индексирования. И потом делится на заданное пользователем число числовых диапазонов, которые могут быть либо равными по величине с разным числом наблюдений, либо э-э, разными по величине с одинаковым числом наблюдений. Вот. Если же текстовая переменная там, то тогда просто сортировка производится по алфавиту с признаком Unique. То есть каждое наименование встречается только один раз. И формируются справочники, которые потом используются для кодирования исходных данных. То есть получается, что у нас есть справочник, в котором есть наименования, числовые диапазоны, и появляется в результате у нас возможность вместо файла исходных данных создать файл, в котором будут коды соответствующих текстов и числовых диапазонов. Сейчас мы это посмотрим.

Значит, это я сейчас вам рассказал о динамической интерпретации того, что в этой таблице. То есть колоночки вот эти зелёненькие – это факторы и значения в ячейках. А жёлтые – это будущее состояние объекта управления, в которое он переходит под действием этих факторов. Вот. Их может быть много, как вы поняли, разных способов классификации этих состояний.

И есть статичная интерпретация, что вот эти зелёненькие колоночки – это свойства, а в клеточках – значения этих свойств. Ну, например, свойством является цвет, а значением – красный. Или там возраст там, или вес, а значением там 22 года, там 60 кг, там вот такое. То есть мы можем здесь рассматривать свойства объектов и значения, а здесь каким категориям эти объекты относятся, обобщающим, каким классам.

Значит, здесь очень важный момент есть, что каждое наблюдение представляет собой описание двумя способами какого-то объекта или состояния объекта, или явления, процесса. Вот, двумя способами. С одной стороны, его характеристиками, его значениями его свойств или факторами, действующими на него. А с другой стороны, принадлежностью к обобщающим категориям различного вида. Такое, такая конструкция, когда объект описывается с одной стороны признаками, а с другой стороны обобщающей категорией, называется определением, если по-русски это говорить, что это такое, это определение, но определение конкретного объекта. Ну, например, стул. Вот стул один, стул прямо, да? Что это такое? Это э-э, мебель, во-первых, то есть это обобщающая категория. Вот. А свойства какие? Четыре ножки, можно на нём сидеть и спинка есть. Вот. И это у нас определение конкретного стула. Но это определение вообще стула. Но конкретного стула можно сказать, что этот вот конкретный стул, вот он маленький там, например, там, деревянный. Вот, какого-то цвета. То есть это всё его свойства. Значит, сами эти свойства можно рассматривать как свойства э-э, физические. Вот я уже сказал, что вес там, допустим, э-э, температура, вот такие свойства – это физические свойства. А также социальные могут быть свойства. Это его э-э, назначение с точки зрения людей. Нравится он людям или нет, и другие э-э, отношения людей к этому объекту. А также э-э, такие свойства социальные, опять же, как стоимость. Вот стоимость – это свойство не физическое, это свойство именно в отношении с людьми. Какие здесь свойства у нас перечисляются, э-э, физические или социальные свойства – это совершенно всё равно, потом всё обрабатывается совершенно одинаково.

5.3. Формализация предметной области и создание модели

Вот. Теперь текстовые шкалы, я сказал, числовые и текстовые. Текстовые могут быть двух видов: номинальные шкалы и порядковые. В номинальных только отношение эквивалентности, не эквивалентности. Скажем, вот список студентов, и какой-то конкретный студент соответствует какой-то строчке, или какая-то строчка соответствует какому-то конкретному студенту. И порядковые. Ну, допустим, у студентов стоит оценка какая-то сбоку там. И можно по оценке рассортировать, ну по средней, к примеру. И сказать, что вот впереди там идут студенты, которые очень хорошо учатся, потом чуть похуже, ещё похуже, в конце, которые плохо учатся. Это называется ранжировано уже или отсортированный э-э, список. И э-э, соответственно, шкала является порядковой. То есть на ней есть отношение больше-меньше. Если же на шкале есть ещё единица измерения какая-то и начало отсчёта, то на ней можно производить арифметические операции. Тогда это называется шкала отношений или числовая шкала. Вот все эти типы шкал в системе Эйдос могут обрабатываться. И э-э, в части вот этих описательных шкал и градаций, которые описывают свойства и их значения или факторы и их значения. И в классификационных шкалах и градациях, которые описывают принадлежность объекта к обобщающим категориям, классам.

Так вот, такая структура, я вам сказал, называется определением по-русски. А вообще, в технологиях искусственного интеллекта она называется онтологией. Но онтологии есть двух видов. Вообще-то есть много разных моделей представления знаний в области искусственного интеллекта. И, значит, онтология – это общий термин. А вот, скажем, э-э, в теории фреймов, модели фреймовые представления знаний, разработанные Марвином Мински в семьдесят пятом году опубликована эта модель. Э-э, такая конструкция называется фреймом экземпляром. И таких фреймов экземпляров может быть много, как вы поняли. А потом система что делает, собственно говоря? Она берёт э-э, фреймы экземпляры, которые относятся к одной какой-то обобщающей категории, и обобщает эти конкретные определения и получает обобщённое определение, которое называется в теории фреймов фрейм-прототип. Обобщающий фрейм. Вот. Ну и также можно вот эти конкретные определения э-э, обобщить и получить научное определение. Ну вот, допустим, мы знаем такое определение, что млекопитающее – это животное, выкармливающее своих детёнышей молоком. Откуда это мы взяли вообще определение? Ну, мы это взяли его из интернета, кто-то нам сказал там. А можно ли получить с помощью интеллектуальной системы такое определение? Можно, ребята, нет проблем. Для этого мы должны в качестве строчек описать разных животных: мышек, кошек, собак, лошадей там, вот, лягушек, ящериц, крокодилов там, в общем, всё-всё-всё, рыб, насекомых там, бабочек, жучков всяких, птиц. Всё мы должны описать. И здесь мы должны описать, значит, э-э, являются ли они живородящими или или э-э, яйца используются у них, или икра. Вот. Чем они выкармливают там, комарами там, молочком выкармливают там, или чем, рыбой выкармливают там, креветками, выкармливают молоком там, да? Вот так перечислить их довольно большое количество разных животных, разных видов животных. А вот здесь вот в обобщающей категории писать: рыба, пресмыкающаяся, земноводная, где там э-э, рождается, где живёт там и так далее. И вот, э-э, я вам могу сказать, что система Эйдос обобщит эти представления о земноводных, о пресмыкающихся, о птицах, насекомых, э-э, млекопитающих. И наиболее важным признаком млекопитающих будет не то, что у них есть шерсть или хвост, вот, или что у них там четыре лапки, а то, что они выкармливают детёнышей молоком. Она это выявит и сообщит вам, что это самый существенный признак для млекопитающих. То есть она способна обобщить конкретные определения объектов или определения конкретных объектов и сформировать обобщённое определение.

Вот. Ну теперь давайте посмотрим ещё немножко. Значит, вот это диапазон классификационных шкал, а это диапазон описательных шкал. Да, общая терминология, которая не касается того вопроса о том, статичная или динамическая интерпретация. Мне эта терминология нравится, она универсальная. Это использование понятия шкалы. Шкала – это способ формализации, скажем так, линейка какая-то. Вот, или ось в каком-то пространстве. Так вот, мы имеем, значит, используем для описания этих объектов, онтологий, описательные шкалы и градации и классификационные шкалы и градации. Описательные используем для того, чтобы описывать, какие свойства у объектов наблюдения или какие значения факторов на них действовали. А классификационные – для того, чтобы описывать, каким обобщающим категориям они относятся, в какие состояния будущие, соответствующие каким классам, они перешли под действием этих факторов.

Вот. И вот здесь вот мы указываем, смотрите, вверху слева тип данных указываем исходного файла, файла исходных данных. Потом указываем диапазон классификационных шкал, потом диапазон описательных шкал. Остальные термины, эти опции пока мы не будем рассматривать. Ну, скажу единственное, что здесь вот мы задаём тип числового диапазона: равные по величине числовые диапазоны с разным числом наблюдений или разные по величине числовые диапазоны с равным числом наблюдений. А здесь мы ставим либо "Формализация предметной области", либо "Генерация распознаваемой выборки". Если формализация, то создаются описательные, классификационные описательные шкалы и градации, и с их помощью кодируются исходные данные. Если мы выбираем второй вариант, тогда уже должны существовать, должно существовать приложение, в котором уже есть э-э, шкалы классификационные описательные и градации. И с их помощью кодируется такой же точный файл, как файл исходных данных, но получается выборка не для создания модели, а для её использования э-э, для решения задач. Вот. Или для тестирования. То есть это обучающая выборка, её иногда называют тренировочной, для того, чтобы подчеркнуть, что она используется для создания модели. А вот эту распознаваемую выборку, её используют, этот термин для решения задач означает решение задач распознавания. Вот. Но может быть, иногда её называют и тестовая выборка для проверки достоверности модели. И переходим на О'кей.

Что происходит? Копируются файлы исходных данных в папочку, которую я показывал. И файл анализируется, преобразуется во внутренний стандарт баз данных. Это достаточно быстро происходит, даже когда файлы большого размера. И анализируется диапазон классификационных шкал и описательных шкал. И определяются типы данных в этих шкалах, число градаций. Значит, если шкалы текстовые, тогда просто градациями являются уникальные текстовые значения. Если, значит, есть числовая шкала, тогда возникает вопрос о том, сколько здесь у нас числовых диапазонов. Если бы в классификационных шкалах была бы числовая шкала, а здесь её нету, тут ноль числовых шкал, а в описательных есть, то здесь бы тоже был вопрос о том, сколько диапазонов. Мы можем это число поменять здесь, пересчитать э-э, градации, шкалы и градации, и потом выйти на создание модели. Вверху вот здесь вот у нас показано нетрадиционно немножко, потому что здесь связано с фразой с этой. Суммарное количество градаций классификационных шкал и описательных шкал. То есть сначала идут классификационные шкалы, их 14, не шкал, а градаций классификационных шкал, то есть классов. И 50 строк, то есть градаций описательных шкал. Это малюсенькая матрица, которая на глаз, правда, глазами просматривается на одной страничке. Ну, выходим, значит, на создание модели теперь.

Произошла формализация предметной области. То есть автоматизированный программный интерфейс 2.3.2.2, он э-э, очень серьёзные функции выполняет. Что сейчас сделано? Пункт первый. Сформированы классификационные описательные шкалы и градации. С их помощью закодированы исходные данные из базы данных Indata. И создана база событий и обучающая выборка. И потом всё это переиндексировано. Вот. Потом сообщение студентам. Студенты, вы видите, что тут написано, нет? Читабельно? Да, читабельно. Вот. Значит, смотрите, значит, пункт первый. Посмотреть сам файл исходных данных по этому пути. Смотрим мы этот файл исходных данных. Вот. Смотрим. Он э-э, это маленький очень файл по размерности, сделанный студентами, которые сидели на лабораторной работе и прямо его и делали. При этом, значит, они допускали массу разных ошибок, неточностей, которые я сейчас вам потом расскажу. Значит, здесь мы видим в первой колонке название конкретного объекта. Ну здесь некоторые объекты были фантастические, то есть они их придумали просто и здесь написали. А в общем, значит, в начале были те, которые реально лежат на столе. Три классификационные шкалы разной степени общности. Но они могут быть и по-разному принципу классифицировать результаты, не только разной степени общности, но разный принцип, способ классификации по разным критериям. И потом идут описательные шкалы. Все эти шкалы текстовые, кроме одной шкалы – размер два, которую я специально включил, чтобы показать, как работа идёт с числовыми шкалами. Вот. И мы, в общем, этот файл обрабатываем в системе.

Значит, вы должны посмотреть режим 2.1, 2.2, 2.3. Почему? Потому что в этих режимах как раз мы видим результаты работы программного интерфейса. Вот диспетчер приложений появилось наименование созданного приложения. Э-э, использование лабораторных работ – не единственный способ создания приложений. Есть другие способы. Мы можем прямо запустить программный интерфейс системы, не из лабораторной работы он запускается. А вот здесь вот у нас есть эти программные интерфейсы: импорт данных из текстовых файлов. Вот этот интерфейс, который сейчас я вам объяснял. Другой экселевский интерфейс тоже, э-э, но там э-э, файлы, другой стандарт файлов. Оцифровка изображений по контурам, оцифровка изображений по пикселям и спектрам. Вот, сценарный системно-когнитивный анализ числовых и символьных и числовых рядов. И некоторые служебные режимы. Специальный программный интерфейс для ввода данных для прогнозирования землетрясений. Тут я включил его, потому что у нас там книга есть с авторами.

И вот, э-э, любой из этих интерфейсов, он тоже создаёт приложение, справочники классификационных описательных шкал градаций и обучающую выборку. Это это способ создания уже научных приложений, не лабораторных работ, учебных приложений, а именно вот самостоятельная разработка новых приложений позволяет эти режимы осуществить.

Заходим в режим 2.1, видим классификационные шкалы и градации. Сразу же можно что сказать? Что названия этих классификационных шкал, они просто взяты из экселевского файла. Вот у нас здесь написано: конкретный класс, обобщающий класс. Вот, вот вам название шкал: конкретный класс, обобщающий класс классификационных шкал. И в каждой из них есть, видите, градации. Они здесь расположены в порядке алфавита. Вот эти градации. То есть это идёт речь о том, что это номинальные шкалы. То есть в данном случае классификационные шкалы – это номинальные шкалы. Здесь никакого разумного порядка нет э-э, между этими значениями. Это просто по алфавиту они рассортированы.

Вот. Теперь смотрим, да, что касается наименований. Желательно, чтобы они были с одной стороны понятными, а с другой стороны как можно короче. Потому что потом они используются в выходных формах многочисленных. И если они очень длинные, то там могут возникать проблемы с их размещением в выходных формах. Хотя там очень слабые ограничения, то есть можно довольно длинные названия использовать, но всё-таки ограничения есть. Ну, примерно 65 символов там можно без проблем использовать, можно и больше, оно будет работать, но будет не всё понятно, что там написано.

Режим 2.2. Это описательные шкалы и градации. Здесь мы видим градация, то есть шкала цвет. Градации здесь у нас, как в номинальной шкале, просто по алфавиту. А напрашивается такой вариант, чтобы они были по спектру там, например, расположены, или по длине волны, грубо говоря. Тогда, если это сделать, тогда это будет порядковая шкала. Материал, скорее всего, он так номинальной шкалой и останется. Размер один – это точно порядковая шкала. Но здесь он очень у нас э-э, недоработан этот вопрос, чтобы это было, то есть сейчас у нас фактически это номинальная шкала, причём такая странная немножко. Вот смотрите, у нас есть градация 20 большая и 21 большой. То есть в наименовании градации есть род: мужской, женский род, мужской род, да? Ну я бы посоветовал э-э, все в одном роде, в одном падеже писать. Ну, допустим, мужской и именительный падеж. Ну, например, средний. Какой? Чтобы отвечал на вопрос какой, например. Какой? Большой, маленький, под руку, средний, например. Большую надо убрать, заменить на большой. Но это надо делать не здесь, а в экселевском файле. Тогда при формировании, выполнении этого режима 2.3.2.2, здесь будет не пять э-э, градаций, а четыре. Теперь, хотелось бы, чтобы маленькая была первая, средняя – вторая, вот здесь вот расположена, под руку – третья, а большой – четвёртая. Значит, чтобы это было, надо просто их пронумеровать. То есть в самом экселевском файле писать: 1 наклонная черта 4 - маленький. Э-э, 2 наклонная черта 4 - средний. 3 наклонная черта 4 - под руку. И 4 наклонная черта 4 - большой. Тогда при сортировке по алфавиту они приобретут разумный порядок, соответствующий смыслу шкалы. И это будет такая классическая порядковая шкала. Размер два. Это числовая шкала. В ней найдено было минимальное и максимальное значение в этой шкале – это 7 и 200. И разделено на равное число диапазонов, то есть, вернее, на э-э, 12 равных диапазонов, извините. Вот, они посчитаны и написано здесь: 1/12 – минимальный э-э, диапазон по величине, попадающих в этот э-э, диапазон чисел. Э-э, 2/12 – это следующий, и максимальный – 12/12. И дальше потом идут шкалы, которые можно назвать логическими. Ну, они являются номинальными у нас формально, но вообще-то это логические: есть, нет, да, нет. Вот. Значит, при этом я хочу ваше внимание на что обратить? Что признак заключается не только в том, что он есть, но и в том, что его нет, тоже это признак. Ну, допустим, провода. Есть провода – это признак, у него код. А если нет, то что, ничего не писать, что ли? Ничего не отмечать? Значит, если мы не будем ничего отмечать, когда проводов нет, то тогда мы теряем информацию. То есть я думаю, что нужно писать, и когда они есть, и когда их нет. Вот, тогда у нас получается два варианта, булевы, так сказать, булева шкала. True, false там, да, нет, истина там, ложь, вот это что я сказал, 1, 0. Вот. Форма – это классическая номинальная шкала. Наличие ножек – булева, да? Вот. Ну, у нас она булева шкала является текстовой шкалой.

И смотрим, что у нас получилось в результате. Значит, вот это наша обучающая выборка. Э-э, вернее, файл исходных данных, он преобразован в обучающую выборку. Каким образом? Значит, э-э, наименование объекта обучающей выборки, которое было в первой колонке, взято прямо один к одному оттуда. Причём про него я говорил, что оно может быть довольно длинным это наименование. Потом, э-э, в левом нижнем окошке идут коды классов в соответствии со справочниками классификационных шкал. Поскольку у нас две классификационных шкалы, то каждый объект выборки обучающей относится к двум классам: к одному классу по первой шкале и к одному классу по второй шкале. То есть он как бы два раза используется этот объект для формирования образов, э-э, сначала первого класса, потом второго класса. У него есть ряд признаков у этого объекта, мышка три. Эти признаки есть, коды этих признаков в правом нижнем окошке. Здесь нет каких-либо принципиальных ограничений по числу этих кодов. Их там могут быть тысячи. И единственное ограничение – файл, э-э, этот, эта база обучающей, вернее, так, эта обучающая выборка состоит из трёх таблиц, каждая из которых не может быть больше 2 ГБ. Это довольно большое количество. Реально при решении задач это примерно соответствует 5 млн примеров. То есть вот здесь вот у нас, э-э, сейчас 20 примеров. Вот если их сделать около 5 млн, тогда эти вот файлы по размерам, файлы таблиц этих соответствующих, будут приближаться к 2 ГБ. Это из опыта. Ну, там плюс-минус, смотря сколько кодов у каждого класса там, то есть там есть, конечно, э-э, может 4 млн, может 6 млн, в зависимости от того, какие вот параметры описания этих объектов обучающей выборки.

И есть другой э-э, способ отображения результатов кодирования исходных данных, который очень напоминает э-э, ту форму экселевскую, которая была исходная. Но в отличие от исходной формы, значит, даже сейчас вот так возьму и сделаю. Вот смотрите. Так, значит, номера, наименования объектов, шкалы классификационные и описательные шкалы. Ну тут я не буду особо так выравнивать, но ясно, что они соответствуют. Вот, то есть эти шкалы описательные соответствуют тем, которые в исходном файле. Но что здесь интересного, ребята? Обратите внимание. У нас вместо текстов и чисел, которые были в файле исходных данных, теперь коды этих текстов и этих чисел, которые э-э, в соответствующих справочниках классификационных шкал и градаций и описательных шкал и градаций. В данном случае это свойства и их значения. И вот начиная с этого момента, ребята, сейчас вот, пожалуйста, слушайте внимательно. Начиная с этого момента, когда закодированы исходные данные, сформирована обучающая выборка, дальше весь процесс обработки текстовой информации и числовой информации, которая в исходных данных была текстовой и числовой, она осуществляется совершенно одинаково. Вот в этом, э-э, в этой обучающей выборке мы уже не видим никакого различия между теми шкалами, которые были числовыми и текстовыми. Там только коды.

5.4. Преобразование данных в информацию и знания

Теперь мы внизу здесь вот видим ещё нумерацию колонок, которая совпадает с фактической нумерацией колонок в файле исходных данных. Вот этот номер в Экселе, он не нумеруется, это просто номер строки. Здесь тоже есть номер строки. А вот наименование объектов выборки, здесь вот у нас первая колонка. И в Экселе это колоночка А. Тоже первая. Потом идут э-э, вторая и третья – это классификационные шкалы, как мы задавали в этой в экранной форме в интерфейсе, да? У нас здесь тоже вторая и третья – это классификационные шкалы. А потом идут описательные шкалы, вот с четвёртой по двенадцатую. И так вот они идут и в экселевском файле. У нас здесь 20 строчек, каждая из которых описывает один объект обучающей выборки. И в экселевском файле у нас 20 строчек, которые точно эти же самые объекты и описывают. То есть мы закодировали исходные данные и получили в результате обучающую выборку.

Вот. Теперь, когда мы получили обучающую выборку, у нас созданы все необходимые предпосылки для выявления причинно-следственных связей между наличием у объекта тех или иных свойств или действием на него тех или иных факторов и переходом его в те или иные будущие состояния, соответствующие каким-то классам, или просто и принадлежностью его к каким-то обобщающим категориям, классам. Э-э, всё это, то, что я сейчас рассказываю вам, описано в режиме 6.4. На лекциях я вам, наверное, рассказывал, ребята, напомните мне, пожалуйста. На лекциях я вам рассказывал про преобразование, про то, что система Эйдос и вообще преобразование данных в информацию, а её в знания при решении задач на основе этих знаний. Была такая у нас тема? Была. Ну, конечно, потому что это базовые понятия. Э-э, я пытался вам объяснить, что делают интеллектуальные системы, что делают информационные системы, что делают базы данных, да? Вот. СУБД и базы данных, информационные базы и информационные системы или аналитические системы, базы знаний и интеллектуальные системы. То есть интеллектуальные системы выполняют такую же функцию по отношению к базам знаний, как СУБД по отношению к базам данных. Вот. Но, э-э, конечно, информация – это тоже данные, но не любые данные, а такие, которые осмыслены, то есть в которых выявлены причинно-следственные зависимости. Конечно, знания – это тоже и информация, и данные. Но это такая информация, которая полезна для достижения целей. То есть, э-э, знание – это информация полезная для достижения цели. А есть ещё другая информация, которая информацией является, то есть она, это осмысленные данные, но она не используется для достижения цели. Тогда это не знание, это просто информация. То есть я вам это всё рассказывал. А в системе Эйдос, значит, это осуществляется следующим образом процедура преобразования данных в информацию, а её в знания. Первый этап мы решаем сами. Это первый этап не автоматизированный в системно-когнитивном анализе. Когда мы просто решаем для себя, что нас интересует, э-э, что мы хотим, собственно говоря, изучать. Какие-то причины влияют на какие-то последствия. Какие-то факторы что-то обуславливают. И что мы в качестве факторов хотим рассматривать? Вот, допустим, когда мы собираемся решать задачу по управлению агротехнологиями, э-э, в процессами выращивания, то мы понимаем, что в качестве факторов мы хотим рассматривать агротехнологические факторы, то есть что там делается на поле, а также факторы окружающей среды. А в качестве результатов их влияния мы хотим узнать количественные, качественные и финансово-экономические результаты выращивания различных культур. Это вот неформализованный этап, который просто означает понятие, понимание того, что мы собираемся сделать. Когнитивно-целевая структуризация. То есть что мы хотим понять в результате создания нашей модели, что мы хотим узнать. Поэтому и называется когнитивно-целевая структуризация. Ну, по сути, это постановка задачи, но такая не формальная постановка задачи, детальная, а понятийная, на уровне идеи. Вот что мы хотим, собственно говоря, от этой модели? Вот мы хотим узнать, как это влияет на это. Всё. Этот этап мы выполнили. Первое. Второй этап и последующие все этапы автоматизированы. Формализация предметной области заключается в чём? В том, что мы анализируем файлы исходных данных, разрабатываем классификационные описательные шкалы и градации, кодируем исходные данные с их использованием, получаем обучающую выборку. Вот я вам показывал эти таблицы, они прямо здесь реальные названия имеют баз, те, которые базы данных, такие вот названия имеют в папочке приложения. Э-э, это база заголовков, то, что было верх, верхнее окошко. Э-э, коды классов, то, что нижнее левое окошко, и коды обучаю, этих, коды э-э, признаков. Э-э, это правое нижнее окошко. Теперь следующий этап, э-э, обеспечивающий преобразование данных в информацию. Он создаёт, рассчитывает статистические модели и модели системно-когнитивные, модели знаний, которые и отражают причинно-следственные связи между причинами и последствиями, теми, которые были определены на этапе когнитивно-структуризации предметной области.

5.5. Режим синтеза и верификации моделей (3.5)

Ну вот, в данном случае, в данной задаче мы выявляем зависимость между наличием у объектов тех или иных значений свойств и их принадлежностью к определённым категориям. Вот. И здесь довольно много разных опций есть, про которые я вам сейчас расскажу, потому что потом я это не планирую делать. Я все вот эти рассказы делаю, так сказать, э-э, на первом лабораторном занятии. На лекции это не получается, потому что нужно показывать экранные формы, выполнять их, э-э, соответствующие режимы. А на лабораторной первой я всё подробно объясняю, а потом дальше уже быстренько мы всё делаем, ничего я там особо не объясняю, только новое объясняю. А нового получается намного меньше, потому что я всё подробно объяснил.

5.6. Использование графического процессора (GPU)

Значит, здесь смотрите, ребята, что здесь есть? У нас здесь есть возможность использовать либо центральный процессор, либо графический. По умолчанию – графический процессор. Что такое графический процессор? Кто-нибудь знает, ребята? Что такое графический процессор? Ну, у всех ли компьютерах есть графический процессор? Ну, давайте, отвечайте. Я вас спрашиваю, ребята, лабораторная работа, я спрашиваю студентов. Что такое графический процессор? И в ответ слышу тиканье часов, вентилятора компьютера. Ну, это отдельное устройство компьютера, которое создаёт графику, текстуры там, цвета. Ну, грубо говоря, это видеокарта или или чипсет, или чип на материнке, который выполняет функции видеокарты. Ну, обычно видеокарта на компьютере есть. Так вот, даже у самых примитивных видеокарт э-э, 96 там шейдерских процессоров и больше. Вот у меня видеокарта на компьютере стоит очень такая простенькая, старинная, которая сейчас там, не знаю, 500 руб. стоит. Э-э, GT 240, э-э, GeForce. На ней 96 шейдерских процессоров. Каждый из них э-э, в несколько раз мощнее, чем центральный процессор. Использование видеокарты для, вот моей э-э, хиленькой, слабенькой видеокарты, вот, э-э, обеспечивает ускорение расчётов от 200 раз до 4.000 раз, в зависимости от того, какая модель. Это то, что я фиксировал, ну, видел просто вот своими глазами. А есть более мощные видеокарты, э-э, которые, скажем, Nvidia 780, например. У неё около 4.000 шейдерских процессоров. И каждый из них мощнее, чем мой на моей видеокарте в несколько раз. А есть видеокарты очень дорогие, которые, ну, скажем, Nvidia 1700. На ней около 7.000 процессоров шейдерских. Что такое эти э-э, видеокарты с такой точки зрения? Они представляют собой суперкомпьютеры практически. Вот их используют для майнинга, может, слышали вы там, для этих блокчейнов там, для всех этих вот криптовалют. Для для майнинга – это значит, для вам дают, начисляют вам эти э-э, криптовалюты, биткоины начисляют в зависимости от того, какие вы вычислительные ресурсы задействованы. И сейчас целые фабрики есть, где молотят эти процессоры день и ночь, водяное охлаждение, у них бассейн какой-нибудь есть для охлаждения этой системы. Представляете себе, вообще, к чему, что происходит, собственно говоря? До конца теоретически непонятно, не осознано, что происходит, собственно говоря. То есть огромные вычислительные мощности, колоссальные просто, тратятся на то, чтобы нагревать воздух, а не для того, чтобы решать реальные задачи, которые кому-то нужны. И за это платят деньги за то, что вы эти ресурсы выбрасываете на ветер, можно сказать. Вот. Ну это вот криптовалюты как раз построены на этом.

Вот. Так вот, э-э, есть возможность использовать эти графические процессоры видеокарт для расчётов. Но, правда, первые фирмы, которые э-э, эту возможность обеспечила, была фирма Nvidia. Поэтому мы можем это использовать на видеокартах фирмы Nvidia и на тех видеокартах, которые имеют чипсет Nvidia. Ну, то есть там те же самые микросхемы, чипы. Вот. Но другие фирмы их просто покупают у фирмы Nvidia и изготавливают свои варианты видеокарт. Это, допустим, фирма GeForce. Ну ещё, может быть, несколько фирм есть. Значит, на всех этих видеокартах работает язык OpenCL, на низкоуровневый язык параллельного программирования, который довольно сложен в освоении, потому что программирование совершенно не такое, как на центральном процессоре. Значит, представьте себе, что у вас есть э-э, э-э, граф, на котором э-э, дугами или пути указаны, э-э, выполнение определённых работ. Вот, допустим, строительство дома идёт. И эти работы есть, есть работы, которые можно параллельно осуществлять, а есть работы, которые можно осуществлять только тогда, когда закончен пучок целый параллельных работ. Ну, например, сколько там квартир в этом доме, там 100, например. Все квартиры можно отделывать, отделку осуществлять одновременно. Вот. Но если какая-то квартира ещё не, не произведена там отделка, то тогда на следующий этап выйти нельзя. То есть дом не сдашь, если он не является домом, где уже квартиры с отделкой. Ну, в общем, вот так вот, такой сетевой график называется работ.

Так вот, точно такой же сетевой график разрабатывается для параллельной программы. В ней есть потоки обработки данных. Эти потоки, они идут параллельно, выполняются разными процессорами, вообще говоря. Вот. Потом собираются результаты всех этих процессов и начинается следующий этап. На следующем этапе тоже может быть распараллеливание. Но есть такие вот точки синхронизации и светофоры, где объединяются данные, где определяется, куда их направлять там и так далее. Значит, с точки зрения э-э, компьютерной техники, э-э, видеокарта представляет собой матричный процессор. То есть там есть и конвейеры, возможность эти процессоры использовать как конвейеры шейдерские, и как вектора, как векторные конвейеры. Э-э, то есть, вернее, как векторный процессор и конвейерный процессор. Причём много конвейеров, много векторов. А когда и то, и то, то это называется матричный процессор. Матричный процессор – это конвейер векторов или вектор конвейеров. Значит, что такое конвейер и вектор? Значит, конвейер – один процессор обработал данные, передал следующему, тот обработал, передал следующему. Ну то есть как вот машину делают на конвейере. Въезжает туда там э-э, кузов, например, пустой, выезжает машина своим ходом, да? Вот. И на каждом этапе, э-э, каждый обрабатывающий центр что-то с ней делает. На одном э-э, обрабатывающем центре одно делает, на другом – другое, на третьем – третье. Вот. Ну и кузов тоже так же делают на конвейере, тоже красят его там, шпаклюют, красят, потом начинают его напихивать всеми вот этими э-э, ручками там, двигателями, колёсами там, всё сиденьями, всё это вот начинают. В общем, есть определённая последовательность этого процесса. Это вот конвейер. Значит, то же самое и на компьютерах. То есть когда одна какая-то информация поступает на вход, обрабатывается немного, передаётся следующему процессору, тот её обрабатывает, передаёт следующему и так далее, и так далее. А то есть у нас что получается? Одна информация проходит сквозь ряд обрабатывающих центров, процессоров. Сама, естественно, видоизменяясь в результате этого. Это вот конвейерный процессор. Так устроены, кстати, между прочим, и центральные процессоры, как конвейерный процессор, принцип его устройства именно конвейер. А векторный процессор – это процессор, который состоит из многих э-э, ядер вычислительных, которые работают, слушайте внимательно, по одной программе, совершенно одинаковую выполняют программу, одну и ту же, все ядра вектора. Но на них подаётся разная информация. Ну вот очень ярким примером векторного процессора является процессор, перемножающий две матрицы. Значит, в матрице произведений каждый элемент равен сумме произведений э-э, матрицы, э-э, элементов соответствующей строки первой матрицы и соответствующего столбца второй матрицы. Вот на их пересечении получается сумма произведений этих элементов строки и столбца, получается элемент матрицы произведения. Так вот, все элементы матрицы произведения рассчитываются совершенно одинаково по одной программе. Но данные разные, из разных строк и разных столбцов исходных матриц. Так вот, если у нас э-э, перемножение матриц осуществляет векторный процессор, то это происходит во столько раз быстрее, сколько там ядер. Ну, допустим, там 4.000 ядер. Значит, он рассчитает матрицу произведения 4.000 элементов за такое же время, как э-э, центральный процессор посчитает одно произведение, ну, один элемент матрицы произведения. То есть он в 4.000 раз будет быстрее считать просто-напросто. Ну, там чуть-чуть по-другому, потому что там ещё есть управление всеми этими процессами, передача данных на сети, ядра и так далее, и так далее. Но это э-э, оптимизировано, стараются, чтобы это как-то не очень тормозило. И, в общем, получается так очень упрощённо, можно сказать, что во столько раз быстрее всё это будет делаться, сколько там шейдерских процессоров.

Так вот, система Эйдос, она может использовать видеокарты для расчётов. Потому что э-э, программы синтеза моделей и программы распознавания реализованы и на центральном процессоре изначально в исходном этом тексте и в исполнимом модуле. И ещё как в качестве внешних модулей э-э, используются программы, разработанные на Дельфи, которые используют язык OpenCL, позволяет. И на нём реализованы вот операции синтеза модели и распознавания. Операция распознавания является крайне трудоёмкой в вычислительном отношении. Вот если у вас, допустим, 700.000 объектов обучающей выборки, и каждый, и, допустим, 400 классов, и каждый описан, допустим, там, ну, 5.000 признаков. Значит, что делается? Цикл сначала по объектам обучающей выборки, которых 700.000. Потом цикл по классам, которых 400. Потом цикл по признакам, которых 5.000. Для того, чтобы сравнить, это при распознавании, для того, чтобы сравнить один объект распознаваемой выборки с одним классом, нужно произвести 5.000 операций сложения количества информации в его признаках о принадлежности к этому классу. Классов 400, объектов 700.000. И вот можете перемножить и представить себе. То есть это может считаться месяцами, понимаете, на центральном процессоре. На графическом вот матричном процессоре такие операции распараллеливаются очень хорошо. Причём это автоматически, э-э, сейчас я расскажу, когда автоматически, когда нет. Э-э, может быть сделано автоматически. И получается, что ускорение до нескольких тысяч раз, даже на моей этой примитивной видеокарте. Это означает, что, по сути дела, какой-нибудь слабенький, старенький ноутбук, по сути дела, является суперкомпьютером в вычислительных, по своей вычислительной мощности. Но мы эту мощность не используем, ребята. Мы это используем только для того, чтобы там туман какой-то был в игре там, или блёск, блеск снежинок, или ещё что-нибудь такое. Ну, чтобы качественно было, короче говоря, изображение, трассировка там лучей и всё прочее. Что действительно требует колоссальных вычислительных ресурсов. Они у нас есть под рукой, но мы ими не пользуемся.

Так вот, э-э, сейчас единственный язык высокого уровня, где распараллеливание, э-э, распараллеливание программы осуществляется самим компилятором, который анализирует исходный текст, находит там потоки, синхронизирует эти потоки, и всё это вот делает, является C#, ребята. Вот я хочу обратить внимание, что C# позволяет комфортно работать с видеокартой для вычислений. Значит, Дельфи позволяет, э-э, ну, образ Паскаль, можно сказать, позволяет работать, э-э, в языке OpenCL, писать самому его, всё это вот дело. Что более, гораздо более трудоёмко. И там есть много своих специфических моментов по поводу того, как представляются числа в памяти видеокарт и так далее. А в языке C# это всё очень комфортно, э-э, сделано, автоматизировано.

5.7. Дополнительные опции режима 3.5

На что обращаю ваше внимание? Ну, может быть, потом будет и в Питоне, или в Пайтоне, вот, как иногда говорят. Теперь, что здесь ещё есть в этой, э-э, в экранной форме по управлению процессами синтеза и верификации модели? Значит, если очень большая выборка, ну, скажем, 700.000 объектов обучающей выборки, то тогда, э-э, верификация модели на центральном процессоре может сильно затянуться. А когда не было графических вот этих модулей, да, кстати, графические модули для графических вычислений, то есть для, вернее, так скажем, э-э, модули для не графических вычислений на видеокарте, разработаны разработчиками из Белоруссии, Димой Бандык. Э-э, значит, эти модули разработаны специально для системы Эйдос. В качестве исходных данных они берут базы данных системы Эйдос, прямо те, которые есть у неё. Решают задачу и потом результаты решения этой задачи тоже выдают в виде форм стандарта системы Эйдос.

Значит, э-э, ну когда этого не было, то тогда, э-э, я что придумал? Значит, я только синтез модели осуществлял. Потом только верификацию осуществлял на небольшом числе примеров, ну, объектов выборки, ну, например, на 100. И определял время, за которое это происходит. В результате я получал такую информацию, что 100 объектов обучающей выборки используются для верификации, это занимает 7 минут. Составлял, ну, к примеру, составлял пропорцию: 100 объектов – 7 минут, X объектов – 480 минут, ну то есть 8 часов. То есть я думал так: вечером запущу, утром получу результат. Вот. И это, соответственно, означало где-то 200.000 объектов. Я их задаю эти 200.000 объектов, иду себе спокойно спать, монитор выключаю, утром у меня готовый результат. Это называется бутстрепный подход. Bootstrap – складной нож. То есть часть обучающей выборки используется для синтеза модели, а часть – для верификации. Когда появился графический процессор, в этом необходимость исчезла. И можно осуществлять и синтез, и верификацию даже на очень больших данных. Но если мы всё-таки это делаем, тогда можно либо удалять, либо не удалять э-э, объекты, которые мы используем из обучающей выборки, объекты, которые мы используем для э-э, тестовой выборки.

Теперь смотрите, ребята, система показывает нам, что вот эта модель, которая сейчас у нас э-э, на в системе Эйдос, она составляет 60% от того, что система Эйдос могла бы обработать. То есть это совершенно такая мизерная модель, вообще миниатюрная. Специально для учебных целей, чтобы было понятно. То есть можно обрабатывать намного, намного больше размерности модели. И всё-таки, если здесь мы видим, что 120%, то нам выдаётся сообщение, что обработка невозможна, и просим вот здесь указать соответствующий процент, э-э, ну, допустим, 5% э-э, оставить результатов распознавания, а все остальные удалить. Причём удалять, всегда удаляются наименее достоверные результаты, то есть с минимальным уровнем сходства. Теперь мы запускаем О'кей на графическом процессоре при параметрах по умолчанию. У нас получается, что не видно вообще, когда эти модели были синтезированы. Ну есть когда на центральном, то там видно сам процесс. А сейчас вообще не видно. Но кое-что видно. Модели были созданы, а потом э-э, создаётся 10 выходных форм по результатам э-э, идентификации, решения задачи идентификации. И получается у нас уже эти модели. Мы их можем просмотреть эти модели.

6. Заключение и планы на следующее занятие

Но, значит, я так думаю, что на этом мы задание, то есть сегодняшнее занятие завершим. Я запишу себе в расписание, что мы начинаем, то есть мы синтез модели осуществили. Вот. И начинаем смотреть на сами модели, а потом на то, какова достоверность этих моделей. Это то, что мы будем изучать на следующей лабораторной работе.

7. Вопросы и ответы

Какие вопросы есть, ребята? Есть вопросы, нет? У матросов? Есть? Ну, спрашивайте, если есть вопросы. Сейчас вы можете задать вопросы, ребята, по материалу.

Так, ну я так понимаю, что вопросов нет. Наверное, всё понятно, а может, не слушали, я не знаю. Ну, будем предполагать, что всё понятно. Можно спросить? Да, пожалуйста. То есть мы, получается, вот делаем всё, что делали сейчас на этом, демонстрировали сейчас, да? Конечно, конечно. Вы должны это повторять на своих компьютерах. Если у вас что-то не получается, вы спрашиваете, что делать. Бывает иногда, что из-за десятки, ну, я же вам рассказывал, что инсталляция проблематична. Сейчас, кстати, я на этом сайте, на своём, написал более подробно, как настраивать э-э… Ну я не буду это переносить, потому что оно не поместится в чате целиком. А просто ссылочку дам. Ну там же, где мы скачиваем систему Эйдос, там описано, э-э, как настраивать операционную систему, чтобы она работала. Вот она, описание это. Оно более длинное стало, больше пунктов. И не помещается в чате. Вот. Если что-то не получается, попробуйте это сделать. Если всё равно не получается, ну тогда у меня спрашивайте. Вот у многих всё нормально получается, я знаю.

Вот, ребята. Таким образом. Желательно, чтобы вы всё это повторяли. Но я понимаю, что не у всех там компьютеры, не у всех э-э, удалось установить систему. Значит, если не удалось, задавайте вопросы. Можете, в частности, вот писать мне на адрес, на почту. Я тоже буду отвечать, что-то рекомендовать. Вот моя почта. Вот. Но если не получается повторять, то желательно, конечно, сделать, чтобы получалось. Но если не получается, то, в принципе, можно смотреть, слушать и понимать, что я делаю.

Ну всё, ребята, на этом конец занятия. До свидания. До следующего занятия. До свидания. До свидания. До свидания.