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

***Российская Федерация***

***Мальцева Эмма Эдуардовна, ПИ2102  
emmamaltseva0@gmail.com***

**27 Лабораторная работа №4. п.2 Инструкция по разработке собственных Эйдос-приложений, ЛР 3.03. 2020-10-31**

**Заголовок:**  
Обзор системы Aidos: разработка приложений, обработка данных и оценка моделей

**Резюме:**

**I. Введение и обзор курса**  
Лектор представляет план курса, включающий работу с таблицами, текстами и графикой в системе Aidos. Основная цель – научить студентов создавать собственные приложения. Подчеркивается важность практической работы для освоения материала, сравнивая теоретическое изучение языка программирования с необходимостью написать хотя бы простую программу. Будет предложено выбрать тему и найти данные для самостоятельного проекта.

**II. Система Aidos: Сложность и проблемы разработки**  
Система Aidos описывается как сложный и масштабный программный продукт. Приводятся примеры ее сложности: 810 функций, объем исходного кода около 3000 листов (127 000 строк), что эквивалентно большой коробке бумаги. Разработка такой системы требует значительных ресурсов (оценивается в сотни тысяч долларов и несколько лет работы команды программистов). Поднимается проблема устаревания языков программирования и сложности миграции больших систем на новые платформы.

**III. Практическая лабораторная работа: Начало работы с Aidos**  
Демонстрируется запуск системы Aidos (рекомендуется использовать Aidos.exe, так как StartAidos может вызывать ошибки). Представлена лабораторная работа № 3.03, посвященная основам разработки собственных приложений в Aidos (пункт меню 2). Описываются первые шаги: очистка предыдущих данных (режим 1.11), проверка и обновление версии системы (актуальная на момент записи – 27.10.2020), добавление лабораторной работы (режим 1.3, тип 3).

**IV. Подготовка и понимание данных**  
Объясняется структура входного файла данных на примере описания предметов студентами. Вводятся ключевые понятия:

* **Классификационные шкалы:** Определяют принадлежность объекта к категориям (например, тип материала, размер). В примере выделены желтым.
* **Описательные шкалы:** Описывают свойства объектов или факторы (например, цвет, наличие экрана/кнопок). Могут быть текстовыми или числовыми.
* **Обработка данных:** Необходимо корректно форматировать числовые данные (разделитель – запятая), избегать текста в числовых колонках. Нули и пробелы могут интерпретироваться как отсутствие данных или как конкретное значение.
* **Область данных:** Важно правильно выделять область данных в Excel (Ctrl+Shift+End), чтобы избежать проблем при импорте в Aidos.
* **Типы файлов:** Aidos может работать с XLS, DBF, CSV. Обсуждаются проблемы конвертации CSV и рекомендуются онлайн-конвертеры.

**V. Синтез и верификация моделей в Aidos**  
После импорта данных система автоматически определяет типы шкал и количество градаций. Запускается режим 3.5 для синтеза и верификации моделей. Упоминается возможность настройки процента данных для идентификации (полезно при работе с большими объемами на CPU вместо GPU). Система строит модели, связывающие описательные признаки с классификационными.

**VI. Оценка производительности и критика моделей**  
Рассматриваются критерии достоверности моделей (режим 3.4). Вводятся понятия:

* **Истинно/ложно положительные и отрицательные решения (TP, TN, FP, FN).**
* **F-мера Ван Рисбергена:** Классический критерий, основанный на подсчете TP, FP, FN (полнота, точность, достоверность).
* **Критика F-меры:** Лектор критикует бинарный подход (суммирование единиц при ошибках/верных решениях), так как он не учитывает степень уверенности системы в своем решении. Система Aidos сама оценивает достоверность каждого решения (уровень сходства).
* **Предлагаемая модификация:** Суммировать не единицы, а саму степень уверенности (уровень сходства) для более адекватной оценки.
* **Зависимость от объема выборки:** Классическая F-мера зависит от абсолютного числа ошибок, которое растет с объемом выборки. Предлагается использовать относительные частоты или инвариантную меру.
* **Связь критериев:** Показывается, что Хи-квадрат, мера Харкевича и ROI (коэффициент возврата инвестиций) по сути сравнивают фактическую и теоретическую частоту встречи признака, но используют разные математические операции (вычитание, деление логарифмов, деление с нормировкой).

**VII. Анализ результатов идентификации**  
Демонстрируется просмотр результатов идентификации (режим 3.4). Показывается, как система относит объекты к классам с определенным уровнем сходства. Визуализация (гистограммы) помогает определить порог достоверности, выше которого ложных решений практически нет. Обсуждается проблема неоднозначности (когда объект похож на несколько классов) и нечеткости (когда степень принадлежности не 100%).

**VIII. Заключение и дальнейшая работа**  
Подводятся итоги лабораторной работы. Анонсируются следующие темы: обработка текстов и графики. Ставится задача для студентов – разработать собственное приложение под руководством преподавателя, возможно, с использованием данных из интернета (например, с Kaggle) или собственных. Упоминается возможность сравнения своих результатов с мировым рейтингом разработчиков ИИ.

**Детальная расшифровка:**

**I. Введение и обзор курса**

Покажу с графикой. Три направления: обработка таблиц, обработка текстов, обработка графики. И потом уже надо будет попытаться что-то своё сделать вам. Выбрать тему, ну и про это тоже расскажу, где можно данные взять, как их там можно ввести и обработать.

То есть всё-таки будет обязательно сделать какое-либо приложение на зачёт?  
Ну, как сказать? Я, конечно, поставлю так, например, если честно. Я же не буду вам… Поставлю. Но просто, чтобы вот научиться чему-то, надо что-то попробовать сделать.  
Ну да, скорее всего. Это же будет на следующем, может, курсе, семестре, будем всё-таки делать приложение?  
Будем, надо, да, надо до этого дойти, до этого уровня, чтобы сделать. Потому что вот вы, допустим, изучаете языки программирования, да? Вот представьте себе, что вам рассказали, что вот есть такой язык программирования, что там есть скобочки, там команды, операторы, функции там, да?  
Ну да, да. Понял.  
И вы, и вы, значит, вроде как уже всё знаете про этот язык программирования. Ну я имею в виду C#, например. А теперь говорят: "А теперь садись вот и сделай, ну хотя бы там 2+2, чтобы она написала на экране". И вы не можете этого сделать. Ну и что тогда? Ну что, что это стоит?  
А, ну хорошо, если будет довольно лёгкое приложение, в принципе, будет понятно, я думаю.  
Конечно. То есть, э-э, ну, короче, всё это, то, что я рассказываю, оно, конечно, я так надеюсь, что это интересно. Вот, но, скажем так, это не увеличит ваши компетенции в плане того, чтобы вы сели и сделали что-то своё, правильно?

Вот. А для чего это всё нужно в конце концов? Для того, чтобы вы могли какие-то свои задачи решать. Может быть, вы даже не будете в системе Aidos их решать, но будете знать, как там это делается. Это вам поможет в будущем, может быть, своё что-то придумать, там соответствующее, реализацию.

**II. Система Aidos: Сложность и проблемы разработки**

Хотя скажу, что это дело такое сложное. Вот. Там огромное количество работы потребуется, чтобы это сделать. Я даже вам сейчас покажу. Я вот недавно стал заниматься тем, чтоб структурировать систему. Вот, по модулям. И оказалось, что у неё 810 функций в системе. Понимаете?

Ну есть исходный текст. Ну вы знаете, что она большая. Вот, но не знаете, насколько большая. Ну я могу сказать, что 3.000 листов. Вот так вот. Строк 127.000. 127.000 строк. Это, это 3.200, где-то ещё десятым шрифтом если печатать. То есть это коробка бумаги, там 2.500 листов. Пять пачек по 500 листов. Вот коробка бумаги и ещё одна пачка сверху. Это вот распечатать десятым шрифтом текст системы.

Ну то есть она большая. Правильно? Вот. Такую большую систему сделать сложно. Это просто требует определённого времени. Даже я скажу вам больше того, очень интересный момент во всём этом есть, что вот если начнёшь её делать, ну вот сейчас, например, начнёшь и сделаешь, допустим, через 3 года такого упорного труда. Вот. Ну 3.000 листов надо забабахать. Может быть, там короче получится, но как-то основные функции какие-то, не все. Ну, в общем, всё равно это большой труд.  
Москвичи пытались как-то оценивать, сколько это может стоить такой труд. У них получалось там 400-500 тысяч долларов, несколько лет. Ну, работа программистов, оплачиваемых по тем тарифам, окладам, которые у них есть там в Москве. Ну там тысяч по 100, по 200 там зарплаты в месяц. Ну получаются такие вот суммы.

Вот. И потом эти программисты скажут: "Ну, пошли. Финансирование прекращено, мы пошли". И это всё колом встанет, и никто не сможет ничего там изменить и вообще понять. То есть это вообще такое дело серьёзное.

И вот к тому времени, когда вы её сделаете эту систему, ребята, если вы начнёте вот её сегодня делать, сделаете через 2-3 года, вот этот язык программирования, на котором вы её делали, он устареет. Вот. И на новый будет трудно перейти, мигрировать приложение.

Вот такая вот проблема существует.

**III. Практическая лабораторная работа: Начало работы с Aidos**

Ну теперь давайте запустим систему. Сейчас я тоже это сделаю. И вы тоже запустите её. И будем проходить лабораторную работу первую, которую мы изучаем.

Это работа 3.03.

Так, значит, вот смотрите. Значит, мы, э-э, значит, скажите мне, пожалуйста, запускается ли у вас система, когда вы запускаете файл StartAidos?  
Со StartAidosом не запускается, мы вам скидывали ошибки, но запускается просто через Aidos X.  
Хорошо, хорошо. Значит, тогда просто запускаете исполнимый модуль. И оказываемся вот в главном меню.

Сейчас мы изучаем, ребята, пункт два, лабораторная работа 3.03.

Давайте тогда у нас, тем более, раз ведётся запись, я коротко скажу, что у нас за занятие и какая работа, что мы делаем.

**IV. Подготовка и понимание данных**

Значит, у нас сегодня 1 октября 2020 года, суббота, пятая пара, 15:00-15:30. Лабораторная работа номер четыре, на которой мы изучаем лабораторную работу 3.03. Пункт два. Инструкции по разработке собственных Aidos приложений.

Ну вот, то есть мы изучаем сейчас пункт два инструкции по разработке собственных Aidos приложений. Это лабораторная работа 3.03, и потом там другие ещё есть.  
Дисциплина: Персональная интеллектуальная онлайн-среда Aidos. Группа 102/2. Занятие ведёт профессор Луценко Евгений Венеминович. Это для того, чтобы потом, если видео кто-то посмотрит, не дай бог, то чтобы они поняли хотя бы, о чём там речь идёт вообще.

Вот. Теперь, значит, мы запускаем систему. Я уже рассказывал про то, как её найти инсталляцию, установить. Это уж я точно рассказывал, да, ребята?  
Вот, теперь мы переходим в режим 1.11. Для чего? Для того, чтобы стереть все приложения, которые там были, если они там были у вас.

Теперь, ребят, новая версия должна быть 27.10.2020. Значит, если у кого-то другая версия системы, то нужно скачать обновление, патч и заменить э-э в папочке, где находится система, заменить все файлы, которые там находятся, э-э на файлы этого архивчика. Но будут заменяться, конечно, не все файлы, а только некоторые будут заменяться, те, которые в этом есть архивчики небольшом, 10 МБ по размеру.

Вот этот патч. Значит, вы должны его скачать и развернуть в папке системы, но система при этом должна быть не запущена, иначе исполнимый модуль системы не заменится.

Вот. Потом переходим в режим 1.3. И здесь у нас есть, видите, добавить лабораторную работу. Добавляем работу третьего типа, то есть вверху птичку ставим на три. Появляется список работ третьего типа. Выбираем работу третью, 3.03. И нажимаем О'кей.

Будет предупреждение, что все файлы в папочке исходных данных будут удалены, потому что туда будут скопированы исходные данные по этой лабораторной работе. То есть если там что-то вы делали, то надо иметь в виду это. Но она об этом сообщает, что сейчас можно скопировать оттуда то, что там нужно вам, если вы там что-то делали. Вот. И нажимаем О'кей. Там всё стирается, туда копируются файлы исходных данных. Можем в этой папочке на этот файл исходных данных посмотреть, что он собой представляет.

Вот, это файл такой полушуточный. Фактически это студенты на занятии все предметы, которые лежали на столе, описали, причём описали не совсем корректно. Сейчас мы коснёмся этого. И, значит, эти данные будут обрабатываться в системе.  
Какая задача будет решаться? Значит, две задачи. Сначала будут созданы модели, в которых будут отражены взаимосвязи между наличием у предметов определённых свойств, которые вот я блоком выделил колоночки, и их принадлежностью к определённой категории. А предметы эти находятся, описание их, то есть их названия находятся в первой колоночке.  
Эти колоночки, которые выделены жёлтым фоном, я их называю классификационные шкалы. Те, которые без фона – это описательные шкалы. Описательные шкалы – обозначают, это такой общий термин, он может описывать эти шкалы либо свойства и их значения, либо факторы и их значения. Ну, допустим, свойства и их значения: цвет, свойство – это цвет, а значение – чёрный, там, красный, серый. Понятно, да?  
А классификационные шкалы описывают принадлежность объекта к каким-то обобщающим категориям. Принципы формирования этих обобщающих категорий могут быть различные. Если мы, допустим, анализируем предприятия или фирмы, или какую-то такую деятельность производственную, то обычно результаты описываются в натуральном выражении и в стоимостном выражении. Ну, когда вот я рассказывал вам про алгоритм принятия решения, я об этом говорил. И натуральное выражение – обычно это количество и качество продукции. Колоночки: количество продукции, качество продукции. Бывает, что качество описывается несколькими колоночками: содержание там, допустим, витамина С, содержание сахара, содержание э-э каких-то пластических веществ, вот такое вот. То есть может быть много таких колоночек, описывающих качество, так же, как и количество.  
Вот. И в стоимостном выражении – это обычно прибыль и рентабельность, но тоже может быть разные. Допустим, прибыль и рентабельность с поля, прибыль и рентабельность по сорту э-э пшеницы, например. Какой-то сорт, допустим, даёт урожайность больше, но но рентабельность у него ниже. Такое может быть. Ну то есть какие-то вот такие моменты. То есть там есть э-э эффективность работы фирмы в натуральном выражении и в стоимостном выражении. И факторы, действующие на фирму. Факторы делятся на внутренние факторы, управляющие, и на факторы окружающей среды, которые тоже точно так же действуют, но мы не можем на них влиять.

Ну вот примерно так. То есть мы должны сейчас выявить взаимосвязи между свойствами и их значениями и принадлежностью объектов к этим обобщающим категориям, классам. Я их называю классами. Колоночки и описательные шкалы, и классификационных могут быть и текстовые, и числовые. Если они числовые, то должны быть э-э запятые разделитель между целой частью и десятичной. И никаких текстов в этой колоночке, кроме названия. То есть если здесь где-нибудь написать вместо нуля буковку О или вместо запятой поставить точку, то вся колоночка будет считаться текстовой. Почему? Потому что текстовый тип данных, он является неизменным по отношению к числовому. Почему? Потому что любое число является текстом. Вот, где символы – это цифры, да? Но не любой текст является числом. То есть любое число можно текстом представить, но не любой текст является числом. И вот если у нас здесь появляется не число, а некий текст, но не число, то тогда уже тип данных должен быть текст, иначе этот текст будет потерян, не сможет быть представлен в том типе данных, который там есть. Поэтому система Aidos, когда есть какое-нибудь не число в колоночке, она тогда считает, что это уже не числовая колоночка.

Теперь смотрите про область данных, важный момент. Вот я ставлю курсор сейчас в клеточку А1, нажимаю Ctrl и Shift клавиши, держу и нажимаю End. Получается такая, видите, ситуация, что блоком выделена область данных, которая действительно соответствует области данных. А могут быть случаи, когда, особенно удаляешь колоночки, добавляешь, удаляешь, корректируешь. Вот, допустим, буковку А я нажму, а потом возьму и сотру. Вроде как у нас область данных та же самая осталась, но это уже не так. Появилась, то есть система Aidos будет думать, что ещё и колоночка М появилась, понимаете? Но сначала выдаст вам сообщение, что там нет наименования, а потом выдаст сообщение то, что там все значения в этой колоночке одинаковые, нет вариабельности. Потом будет работать дальше. Но оно вам надо это? Ну вы должны знать, что область данных должна быть та, которая фактически. Если у вас это не так, вы должны фактически выделить ту, какую надо вам, которая является фактической областью данных, создать новый лист, вставить туда, ну, сделать ширину колонок, а старый удалить. Это самый простой способ, ребята. Старый лист удалить, где неправильно был э-э блок данных, неправильно выделялся. Значит, вот это нужно сделать будет для того, чтобы дальше нормально работало.

Вот. Теперь мы задаём в системе диапазоны классификационных шкал и описательных шкал. И тип данных. То есть мы здесь должны задать тип данных Экселя старого. Можем нового, можем DBF файл, можем CSV файл. Недавно я сделал CSV DBF конвертер в системе, включил в систему. Он неплохо работает. В некоторых отношениях он лучше, чем Excel. Вот, но в некоторых случаях он даёт ошибки, когда какие-то нестандартные CSV файлы. В этом случае я советую пользоваться э-э онлайн-конвертером, который я сейчас вам ссылочку дам. CSV XLS онлайн. Э-э, очень хороший конвертер, который корректно работает. Вот. И вот я его вам рекомендую. Ну, конечно, есть и другие. Может быть, другие тоже нормальные, но вот этот вот я пробовал, и, в общем, мне он нравится, как он работает. Он часто обрабатывает вот эти вот некорректные CSV файлы. А что там значит, я имею в виду под некорректностью? Ну бывают такие сложно организованные, какие-то заморочистые, где, скажем, поля разделены не просто запятой, а, допустим, где разделены точкой с запятой, что уже странно несколько. Вот, но э-э вот эти вот э-э блоки, которые разделены с точкой с запятой, внутри них есть, допустим, поля, разделённые запятой. Ну что делать с такими данными, не очень понятно. Вот. По идее, можно включить целиком это в значение поля, вот этот вот блок весь, разделённый запятой. Но дело в том, что он разделён запятой, а это э-э предполагается, что это поля раздельные. То есть стандартный конвертер CSV файлов, он даст э-э неправильное преобразование. Если Excel преобразовывать CSV файлы, то часто очень числа воспринимаются как даты. То есть присваивается тип даты колоночке там или ячейке. И, в общем, потом не поймёшь, что там было до этого. То есть Excel он, скажем, слишком интеллектуальный, наверное, перемудрённый. То есть они хотели сделать хорошо, потом хотели сделать ещё лучше, а получилось хуже, а не лучше. То есть лучше бы было бы, если бы он просто э-э туповато так преобразовывал числа в числа, тексты в тексты, даты в даты. А он иногда там что-то там думает, что это число – это дата, непонятно, в общем. Вот этот конвертер онлайн, он в этом смысле гораздо корректнее работает. Вот его советую. Почему я про него упомянул вообще про этот конвертер? Потому что э-э часто исходные данные для расчётов находятся в CSV формате, форматированный текст. То есть это, в общем-то, обычный досовский текст, но значения разделены запятой.

И вот мы можем посмотреть, что здесь вот у нас э-э задан тип данных. Это все параметры, которые здесь заданы, они соответствуют тем, которые необходимы, потому что они э-э вместе с этим файлом исходных данных считываются вот эти вот параметры и показываются на экране. Этот можно убрать. Теперь смотрите, значит, нули и пробелы можно считать отсутствием данных, а можно считать, что это значение. Вот если считать, что это значение, то в числовых шкалах э-э при сортировках, если ноль будет минимальным значением, к примеру, получится, если там нет отрицательных, то э-э диапазоны числовые будут от нуля идти и дальше там до максимального, на заданное число диапазонов. А если не считать его э-э значением ноль, а считать отсутствием данных, то минимальное значение области, значит, значений в этом, в этой колонке будет считаться минимальным, но это будет не ноль, а там, скажем, пять, например, или там шесть там. То есть какое там есть минимальное значение, оно будет считаться минимальным, и потом от него и до максимального будет задано число диапазонов создано. И вот здесь вот у нас мы видим, что у нас диапазоны числовые, в числовых шкалах есть равного размера с разным числом наблюдений и разного размера с одинаковым числом наблюдений, примерно одинаковым. И мы задаём здесь диапазон классификационных шкал. Там у нас они жёлтым фоном выделены. 2-3. А описательные четыре, с четвёртой по двенадцатую. Как можно быстро узнать э-э э-э без нумерации, какие номера колонок? Ставим курсор в клеточку А1, нажимаем Shift и двигаемся вправо. Вот здесь вот у нас в уголке появляется формула, на которой написан номер колонки: 2C там и так далее. И вот до конца двигаем, получается 12. Ну то, что там вторая, третья, оно и так понятно. Это вот четвёртая тоже можно догадаться. И до двенадцатой. Вот у нас эти диапазоны колонок. Из этого ясно что? Что э-э типы шкал э-э, ну, классификационные и описательные, они должны идти подряд. То есть, допустим, сначала идут все э-э классификационные шкалы, мне так больше нравится, а потом идут все описательные. Но если взять, допустим, сайт с источниками данных для машинных обучения, для машинного обучения Kaggle, очень хороший известный сайт, то там э-э чаще всего идут сначала описательные шкалы, а потом одна, обычно, классификационная шкала в самом конце. То есть можно, это тоже возможно. То есть тогда просто нужно указать вот здесь соответствующие диапазоны. Ну, допустим, здесь указать 4-12, а здесь указать 2-3, если там наоборот они были, стояли. Но просто подряд они должны быть. Подряд классификационные и подряд описательные. И не должны быть там внутри них там, допустим, внутри описательных одна классификационная, потом опять описательная. Такого не должно быть.

**V. Синтез и верификация моделей в Aidos**

Вот. Теперь такой момент ещё. Система Aidos может строить многослойные нейронные сети. Но для этого нужно, чтобы у нас в исходном файле данных данные были расположены в таком порядке, что сначала должны идти нейроны э-э верхнего слоя, потом рецепторы верхнего слоя. А потом э-э рецепторы верхнего слоя являются нейронами второго, нижнего слоя. И потом рецепторы нижнего слоя. И таких слоёв может быть сколько угодно, в принципе, но обычно делается или один слой вообще в сети, или два. Больше очень редко. Вот. Можно построить их, посмотреть модели в этих слоях. Э-э, для чего это нужно? Ну, например, мы можем по ответам на вопросы респондентов определить у них, какие у них личностные свойства психологические. Это первый слой. То есть вот здесь вот ответ на вопрос, здесь вариант ответа. А здесь колоночка классификационная шкала э-э первого слоя, то есть нейронный первого слоя. А варианты ответа на вопросы – это рецепторы первого слоя. Ну, узнаём, какие у него личностные свойства. А этот же слой используется в качестве входного, это слой рецепторов для второго слоя. А второй слой, значит, он определяет по тому, какие у человека личностные свойства, какая их степень выраженности, определяет, насколько он похож на образы обобщённые тех, кто хорошо или плохо работает по тем или иным должностям данной фирмы, и кто себя хорошо или плохо проявляет с точки зрения этики, там, поведения, криминогенности и тому подобное. То есть мы можем и сразу связать, как человек работает с ответами на вопросы или другой стимульный материал. Но тогда будет усложнена интерпретация. То есть нам тогда будет непонятно, почему хорошо или плохо он работает. А если сначала выявить личностные свойства, а потом уже связать их с успешностью деятельности, то тогда можно сказать так, что вот такая-то категория работников, вот которые хорошо работают на такой-то должности, для них характерным является такие-то личностные свойства. То есть составить профессиограмму можно будет. И потом эти профессиограммы использовать в системе Aidos, причём в адаптивном режиме на данных самой фирмы.

Вот. Теперь смотрим, что у нас дальше происходит. Значит, дальше мы здесь вот задали диапазоны, всё остальное я сейчас вам не показываю, это будем смотреть на работах другого, другой направленности. Ну, допустим, связанной с обработкой текстов. Нажимаем О'кей. Файл вводится в систему. Вот этот экселевский. В системе есть встроенный конвертер XLS DBF конвертер, то есть он преобразуется во внутренний стандарт баз данных. И система обнаружила две текстовых классификационных шкалы с суммарным числом интервальных значений, градаций 14. Это у нас колоночки будут в матрицах моделей. Вот. И обнаружила одну описательную шкалу числовую с двенадцатью градациями и восемь обнаружила текстовых описательных шкал с суммарным числом градаций 38. Всего девять шкал с числом градаций 50. Мы пока здесь ничего менять не будем, но если здесь поменять, вот скажем, на пять, например, тогда нужно вот здесь пересчитать. И тогда мы видим, что изменилась размерность матриц модели. Ну сейчас мы это делать не будем, потому что у нас там всякие есть недоработки э-э в том, как мы задаём эти параметры, в том, как заданы исходные данные. Мы их увидим эти недоработки. Как э-э это будет в системе проявляться, мы всё это увидим, и это я на это ваше внимание обращу.

Значит, теперь смотрите, что у нас произошло? У нас всё происходит в соответствии со схемой преобразования данных в информацию, информации в знания. То есть мы взяли исходные данные, проанализировали их, сформировали классификационные описательные шкалы и градации и с помощью них закодировали исходные данные. В результате у нас получилась обучающая выборка. Обучающая выборка является базой исходных данных, нормализованной с применением вот этих справочников, которые сейчас были созданы. Ребята, извините. Четвёртая пара – это вам не то, чтобы там, правильно?

Вот. Продолжим. Вот это вот окошко появляется с рекомендациями для студентов. Мой опыт показывает, когда вот ещё было личное образование, что студенты никогда не читают, что здесь написано, сразу о'кей нажимают. Наверное, у них это вызывает скуку, наверное, то, что там написано. А там написано, что им дальше делать. Написано: открыть этот файл, который мы уже открыли, посмотрели. А потом написано, что выполнить режимы 2.1, 2.2, 2.3.1, 3.5, 5.5, перечислены режимы. И написано, что это надо сделать в соответствии с схемой преобразования данных в информацию, а её в знания, которая приведена в режиме 6.4.

В диспетчере приложений появляются приложения. Название мы можем поменять. Вот. Здесь может быть много приложений. Рабочее – это то, которое сейчас является текущим, буковка W, Work. Ну, можно, в принципе, любой буковкой обозначить, система потом заменит на W. Первое приложение, где есть э-э не пробел в этой колоночке, будет рассматриваться как текущее. Удаляется всегда текущее приложение, если мы удаляем.

Смотрим, как она нам рекомендует э-э классификационные шкалы и градации. У нас две шкалы, как две колонки были в экселевском файле, так вот у нас и две шкалы. Здесь у нас э-э уникальные значения в этих шкалах. Шкалы текстовые. Значит, там была сортировка с признаком Unique, то есть повторы э-э отсутствуют в наименованиях. Но никаких там э-э приведений к леме или э-э выравнивания э-э размеров букв, большие, маленькие, тут не было. Просто как один к одному, как оно было, так оно и сделано. Это у нас номинальная шкала, где нет никакого порядка на шкале, а есть только операция эквивалентности, не эквивалентности. Они рассортированы по алфавиту, эти наименования.

Описательные шкалы. Это у нас название колонок в экселевском файле. Кстати, из этого следует, из-за того, что это название колонок, что они должны быть не очень длинными, потому что потом они будут участвовать в выходных формах, и может так получиться, что вообще не поместятся в выходных формах. Хотя там э-э довольно много места я старался для этого уделить, но можно настолько длинное наименование написать, что они не будут помещаться.

Ну здесь мы видим э-э шкалы э-э номинальные, порядковые. Вот материал – это явно номинальная шкала, только отношение эквивалентности. Размер, по идее, это у нас порядковая шкала, но здесь мы видим, что у нас градации этой шкалы текстовые и расположены в таком порядке, который не соответствует их смыслу, их смыслу. Значит, что мы можем сделать, чтобы это было э-э порядковая шкала? То есть формально это номинальная шкала тоже. Чтобы она была порядковая, для этого нужно зайти в экселевский файл и во всей колоночке вот этой вот, где написано название размер один, именно в этой колонке, поменять маленькие на один, наклонная черта 4 - маленький. Э-э, средний поменять на два, наклонная черта 4 - средний. Под руку поменять на один, наклонная черта три. Ой, вернее, извините, три, наклонная черта 4 - под руку. То есть э-э номер градации, короче, указать и через из скольки градаций. Первая из э-э четырёх, вторая из четырёх, третья из четырёх, четвёртая из четырёх. А вот эта большая тоже заменить на э-э четвёртая из четырёх - большой. Вот. И тогда у нас здесь будет четыре градации, и она будет порядковая шкала это у нас. То есть будут отношения больше-меньше между градациями. А это у нас интервальная шкала числовая. У нас здесь найдено минимальное, максимальное значение и разделено на заданное число одинаковых по размеру интервалов.

Вот. И потом дальше здесь есть у нас логические шкалы. Вот, скажем, наличие экрана, кнопок. Здесь у нас есть, нет. Вы давно встречали такие анкеты, где нужно было ответить, отметить птичкой вопросы или варианты ответа, на которые вы отвечаете да. А на которые нет, не надо было отмечать птичкой. То есть там не было варианта нет. Если вариант да, то вы отмечали птичкой, а если нет, то вообще ничего не отмечали. Это вас немножко, ребята, разводят на этом деле. То есть такая обработка такой информации, она приводит к тому, что ответа нет вообще не учитывается. Вот. То есть получается, у нас три варианта есть ответа на вопрос. Вот смотрите, ребята. В анкете можно ответить да, можно ответить нет, а можно вообще не ответить. И вот если у нас варианта нет нету, то у нас получается, что у нас есть два варианта тогда: да или не ответить. Вот не ответить – это не то же самое, что нет. Улавливаете, нет, ребята? Когда человек отказывается от ответа или когда он отвечает нет – это разные вещи. Вы понимаете, да, это? Да, да. Получается искажение информации обрабатываемой. То есть модель становится менее адекватной в результате. Ну, например, если мы обрабатываем двоичные числа, там у нас 0, 1, да? Вот если один, то это есть. А если ноль, то это что, нету, что ли? Тоже есть, только ноль. Вот нужно это понимать.

**VI. Оценка производительности и критика моделей**

Вот. Теперь следующий этап – это э-э смотрим на саму обучающую выборку. Вот у нас то, что было написано на левой колоночке, это не является, она не является шкалой. Это вот здесь у нас есть, и может здесь вот как раз могут быть длинные наименования. И здесь вот у нас в нижнем правом окошечке коды э-э значений свойств в соответствии с этими справочниками соответствующими, которые мы я вам сейчас показывал. А в левой нижней окошечке показаны коды классов, к которым относятся эти объекты. Каждый объект относится к двум классам. То есть фактически это означает, что один физический объект является двумя логическими объектами. Это в терминологии, которую я там применял, когда описывал это всё.

Теперь мы после этого создаём модель. Это всё делается в соответствии э-э с процессом преобразования данных в информацию, а её в знания, который описан вот на этой страничке. Вот. То есть мы поняли, что по признакам объектов хотим их идентифицировать, относить к категориям каким-то. Потом мы создали на основе исходных данных классификационные шкалы и градации, описательные шкалы и градации, и закодировали исходные данные с их помощью, и получили обучающую выборку. А теперь мы будем выполнять этап синтеза и верификации моделей. Это делается в режиме 3.5.

Значит, здесь есть такие нюансы. Вот с чем я буквально вчера столкнулся с одним из нюансов. Студенты решили сделать приложение на очень большом объёме данных. Ну, сравнительно большом, там около 70.000 примеров. Ну, как много, мало. Ну, в общем, довольно много. Вот. И не получалось это сделать на графическом процессоре, хотя это предпочтительно. Обычно на нём всегда всё нормально получается, но очень быстро, но тогда, когда всё-таки данных поменьше. Вот. И пришлось что сделать? Значит, э-э на центральном процессоре считать. Но если там э-э такой объём данных анализировать, то это будет довольно долго. Поэтому я взял вот тут вот и написал вместо 100, написал один. И запустил. И тогда что получилось? Что из э-э результатов идентификации, которых будет огромное количество, потому что каждый объект обучающей выборки, который 70.000, будет сравниваться с каждым из классов, которых там тоже там штук 10 там или 20 там. Вот. А так из этого всего количества результатов распознавания огромного, оставлено будет 1% всего тех, кто тех результатов, которые имеют наиболее высокую достоверность результатов. Достоверность оценивается системой. И, как показывает опыт, она оценивает очень э-э с высокой, ну, скажем так, её её оценки, они являются адекватными. Оценки, которые даёт система результатам идентификации, прогнозирования, принятия решений, они адекватные эти оценки. Это подтверждается э-э статистикой истинно положительных, истинно отрицательных, ложно положительных и ложно отрицательных решений в зависимости от уровня сходства. Мы видим, что чем выше уровень сходства, тем э-э больше э-э доля истинных решений. Это означает, что уровень сходства является адекватной мерой степени истинности.

Сейчас создаются модели и проверяются на достоверность.

**VII. Анализ результатов идентификации**

Вот. Получили мы, э-э, все модели созданы и проверены на достоверность, верифицированы на тех же исходных данных, э-э, обучающей выборки, на основе которых они были и созданы эти модели. Теперь смотрим в режиме 3-4 на достоверность моделей. Значит, здесь есть несколько разных критериев, э-э, достоверности. А сейчас мне, пожалуйста, ответьте, ребята. Неужели я вам не рассказывал про критерии достоверности модели, критерии F, критерии Ван Рисбергена, его обобщение? Было дело или не было?  
Ну, если помните, учёный.  
Ну, в общем, там э-э идёт речь о чём? О том, что есть истинно положительные, э-э, истинно отрицательные решения, ложно положительные и ложно отрицательные. И подсчитывается на основе этого э-э полнота модели, точность и достоверность. Это называется F-мера достоверности модели Ван Рисбергена. Не было?  
Ну, насчёт ложных этих я не помню.  
Угу. Ну тогда сейчас рассмотрим. Значит, смотрите, ребята, когда мы решили задачу идентификации, то есть мы создали модели, посмотрим на эти модели. Значит, колоночки – это у нас классы, обобщающие категории, а строки – это значения свойств. А вот эти вот единички стоят там, где э-э это свойство встретилось у объектов этого класса. И поскольку объектов у нас очень мало в обучающей выборке, то получается, что и свойства практически там везде единички стоят. Один раз встретилось, два раза. Здесь вот, допустим, у нас это свойство три раза встретилось вот в этом классе. Вот. А это всего, сколько по всей выборке встретилось. Здесь у нас не 20, а 40. Почему? Потому что у нас каждый объект два раза использовался. Один раз использовался для того, чтобы э-э учесть, какие в нём признаки по первому, первой классификационной шкале, а потом то же самое по второй классификационной шкале. То есть фактически вот здесь вот у нас в строке, в колоночке сумма, числа будут в два раза больше, потому что две классификационных шкалы. И сумма по этой колоночке, по этой строке, вот где у нас число объектов, по строке последней, число объектов обучающей выборки по различным категориям, по классам, будет тоже в два раза больше, соответственно. Потому что здесь у нас как бы подсчитываются логические объекты, а не физические. И вот мы видим, что здесь у нас есть и двойки, и тройки, то есть какие-то эти вот э-э свойства, признаки встретились там по несколько раз в разных категориях объектов. А это вот у нас то же самое уже в виде э-э условных и безусловных процентных распределений, условные относительные частоты посчитаны. На что, допустим, 25% объектов этой категории обладают этим признаком. А вот этим, а этим же признаком 14% объектов этой категории обладают. А всего у нас э-э 15% объектов обучающей выборки обладают этим признаком. И, соответственно, мы можем посчитать, теперь сравнить можем э-э вероятность встретить признак в объектах определённой группы и вероятность встретить этот признак по всей выборке. Если его вероятность встретить в группе больше, чем по всей выборке, то он характерен для этих объектов этой выборки. Если меньше, тогда значит не характерен для объектов этой выборки. Вот. И, значит, это можно увидеть э-э в критерии Хи-квадрат, который как раз и означает, насколько э-э, во-первых, э-э сравнивается фактическая частота встречи признака и теоретическая. И если фактическая выше, чем теоретическая, тогда это значит признак характерный для объектов этой категории. Если равна, тогда значит ничего нельзя сказать об этом признаке и объекте, никак они не связаны между собой практически. А если э-э какой-то признак у объекта встречается реже, чем по всей выборке, то он не характерен для этого, для этого класса. Вот, а больше характерен для других классов. Ну, например, я могу вам привести такой наглядный пример, который я всегда студентам привожу, такой напрашивается он сам собой. Вот представьте себе, что есть у нас студенты и студентки, нужно их друг от друга отличить по каким-то внешним признакам. И вот у нас один из таких признаков – это длина волос. Вот я спрашиваю: "А длинные волосы?" А там за дверью кто-то стоит, я не знаю, кто там. Мне говорят: "Да, длинные". Я думаю: "Ну, наверное, студентка". А потом спрашиваю: "А в брюках или в юбке?" Мне говорят: "В брюках". Вот. То есть есть брюки? Мне говорят: "Да, в брюках". Тогда я всё равно думаю: "Ну всё равно, наверное, студентка". Почему? Потому что признак волосы длинные у студенток намного чаще, чем в среднем встречается. И много несёт информации о том, что это студентка. А признак брюки у ребят встречается чаще, чем у студенток. Но чаще совсем чуть-чуть, потому что студентки тоже очень часто в брюках ходят. То есть различие по этому признаку между ними меньше, чем по длине волос. Поэтому признак длинные брюки несёт информацию о том, что это студенты и не студентки, но очень мало несёт информации. А признак длинные волосы очень много несёт информации о том, что это студентка это или студент. Если студенты бывают с длинными волосами тоже, но очень редко. Получается, что этот признак несёт очень много информации о том, что это не студент, а очень много информации о том, что это студентка. А вот признак, допустим, мобильный телефон, есть или нет мобильный телефон, вообще никакой информации ни о чём не несёт. Потому что он у всех встречается с одинаковой вероятностью и в каждой группе, и по всей выборке.

Теперь смотрим на результаты решения задачи идентификации. Значит, смотрим, что у нас получилось. У нас получилось, что объект э-э системой отнесён к какому-то классу, конкретный объект обучающей выборки отнесён к каким-то классам. И отнесён с определённым уровнем сходства. Вот здесь они показаны. И есть решения истинные и ложные. Значит, э-э где истина стоит птичка, то есть действительно объект к этому классу относится. А ложное – там птички нет. Существует э-э несколько видов ошибок идентификации, не идентификации. Смотрим, ребята. Если система э-э считает, что объект больше, то есть похож на какой-то класс с уровнем сходства больше нуля, значит она считает, что он похож на этот класс. То есть она его относит к этому классу. Это может быть действительно, что он к этому классу относится, если птичка стоит, да, вот истинно положительное решение. А может быть, что она его относит, то есть уровень сходства больше нуля, но это на самом деле не так. То есть это ложно положительное решение. То же самое касается решения отрицательного, о непринадлежности объекта к классу. Если уровень сходства объекта с классом меньше нуля, то это называется решение отрицательное. То есть когда больше нуля – это положительное решение, когда меньше нуля – отрицательное. И положительные решения, отрицательные бывают истинные и ложные. Если объект э-э уровень сходства больше нуля, и действительно он к этому классу относится, тогда это истинно положительное решение. Если уровень сходства больше нуля, но объект на самом деле не относится к классу, то это ложно положительное решение. И если э-э уровень сходства меньше нуля, и объект не относится к классу действительно, то это истинно отрицательное решение. А если уровень сходства меньше нуля, а объект на самом деле относится к этому классу, то есть система его не относит к нему, а он на самом деле к нему относится, то это называется ложно отрицательное решение.

Ну, что можно здесь сказать? Что э-э система должна правильно относить объект к классу и не относить объект, то есть правильно относить к тем классам, к которым они относятся, и правильно не относить к тем, к которым они не относятся. И она может ошибаться и в первом, и во втором случае. То есть могут быть истинно положительные, истинно отрицательные решения, а могут быть ложно положительные, ложно отрицательные решения. И Ван Рисберген э-э как раз и предложил меры достоверности, которые основаны на подсчёте этих вот частот. То есть он в программистской аннотации, я здесь пишу не как математики, а как вот программисты пишут переменные. Вот он ввёл переменные: True Positive, True Negative, False Positive, False Negative. Вот. Это переменные э-э программистские языка программирования, которые суммируются в эти сумматоры. Сначала они все равны нулю, потом какое бы ни было принято решение э-э с помощью модели о принадлежности, непринадлежности объекта к классу, э-э оно будет всё равно в любом случае положительным или отрицательным, то есть принадлежность или непринадлежность. И оно будет либо истинным, либо ложным. Других вариантов не бывает. Поэтому к какому-то из этих сумматоров всё равно просуммируется единичка. Вот. Ну и в результате после распознавания всех объектов обучающей выборки, мы можем посчитать по этим формулам, которые обоснованы были Рисбергеном, Ван Рисбергеном, посчитать точность модели, полноту и достоверность, которая называется F-мерой.

Какие есть э-э возражения у публики? Значит, э-э я могу сказать, что мера F-мера Ван Рисбергена является классической мерой э-э достоверности модели и очень широко применяется э-э и в основном варианте, и ещё бывает в других вариантах. Есть несколько разных вариантов этой меры для случая, когда только два класса, например, есть. Э-э, ну и для других. Бывают смещённые всякие меры, когда больше играет точность, чем полнота, или полнота больше, чем точность. Ну, в общем, короче, есть много разных вариантов этой меры. И э-э, кроме полноты и точности модели, с помощью этих счётчиков, сумматоров, рассчитывается очень большое количество ещё других показателей различных. Э-э, я видел статьи по F-мере Ван Рисбергена, в которых э-э целый лист этих формул приводятся, которые можно посчитать на основе вот этих значений этих сумматоров.

Вот. Теперь такой момент интересный. Вот с чем я буквально вчера столкнулся с одним из нюансов. Вот он берёт и предлагает суммировать единички в случае ошибочных решений к сумматорам False Positive, False Negative. Ну, False – это ложь, True – истина, вы, наверное, знаете. И вот у меня возникает такой естественный вопрос: а на каком основании он вообще э-э считает, что нужно суммировать единички к сумматорам False Positive? Она ведь сама сомневается система, она только на 4 сотых уверена в том, что это правильное решение. А максимум единичка. Вот. Ну или если у нас четыре здесь написано процента, то максимум 100 тогда. Это говорю, несправедливо. Если бы система не знала э-э степень уверенности своего решения, тогда да, я согласен. То есть для чётких систем, которые говорят: да, нет, относится к классу, не относится к классу. То для них, согласен, можно суммировать единички к этим сумматорам. Но если система нечёткая и определяет степень принадлежности объекта к классу, то есть она не просто определяет, что он принадлежит, она ещё определяет, какой степенью он принадлежит. Как вот мера расстояния в этой, нечёткой логике, степень истинности, понимаете? Вот. То есть она сама определяет эту степень истинности. И сейчас она определяет как очень низкую степень истинности этого решения, принадлежности, допустим, сумки к клавиатуре. То есть э-э клавиатуры два к классу сумки или аксессуара. Она сама считает, что это очень маловероятно. А вы берёте, суммируете единичку к этому сумматору False Positive. Ну или это как-то не несправедливо. Он говорит: "А что справедливо?" Ну это я так мысленно с ним как бы это, скажем так, обсуждаю этот вопрос. И говорю, он говорит: "А что справедливо? Как вы считаете?" Я считаю, что надо суммировать 4 сотых. Вот когда истинное решение, тогда надо суммировать единичку. То есть, вернее так, когда м-м уровень сходства очень высокий, 100%, то тогда надо суммировать единичку. А если уровень сходства 95%, тогда нужно суммировать 0,95. А если 56%, то нужно суммировать 0,56. А если 4% уверенность, то надо суммировать 0,04. Вот что я считаю. То есть я считаю, что нужно суммировать сам уровень сходства. Э-э, и то есть это и есть степень уверенности в решении.

Это что касается нечёткости. Вот так я предложил её решить. И ещё такой момент интересный. При увеличении объёма выборки мера Ван Рисбергена, вот все эти сумматоры, которые он считает, они все увеличивают свои значения практически линейно объёму с объёмом обучающей выборки. Вот я сейчас вам нарисую здесь картиночку и в ваш чат пошлю. Раз такая возможность есть, то почему бы это не сделать? Вот. И вот смотрим. Значит, э-э, ну давай сначала напишем. Вот, это объём выборки. Вот, а это у нас число решений. Вот, а это у нас какие решения? Это вот у нас True Negative. Почему? Потому что опыт показывает, что я вам объясню, могу объяснить, почему это так. Э-э, истинно положительные, то есть истинно отрицательные решения растут, их число растёт быстрее всего. Вот. Это вот у нас как бы больше желаемое, чем фактическое, но в хороших моделях так и есть. А вот ложные решения тоже растут их число. Вот. False Positive. И медленнее всего растёт False Negative. Чаще всего вообще она близко к этому идёт к оси X. И посылаю вам в чат ваш замечательный, который такие вещи воспринимает.

Вот, вот примерно так они растут. Значит, в хороших моделях вот эта вот э-э линия False Negative, она вообще ползёт практически параллельно оси X. Очень медленно растёт. Вот так она растёт. Вот False Negative. Это чаще всего так вот происходит. Она растёт, но очень медленно. Вот. В результате этого, когда мы подсчитываем по этим вот формулам, предложенным Ван Рисбергеном, меня заинтересовало, как будет вести себя мера Ван Рисбергена в зависимости от объёма выборки. Я взял и посчитал, сделал модель. Здесь сначала вы видели, описывается классический, классический подход к мере Ван Рисбергена, потом описываются её недостатки, потом предлагаются решения этих недостатков. И э-э самый главный недостаток, ну, вернее так, там вот эти все недостатки все существенны. Все их нужно как-то учитывать, обрабатывать. И, короче говоря, я предложил решение этих всех проблем математическое. И исследовал, как ведёт себя мера э-э сама классическая мера Ван Рисбергена. Да, это произошло в 2017 году, 3 года назад я статью эту написал. Ну, на самом деле, я их использовал эти критерии гораздо раньше, но вот такую статью развёрнутую написал вот 3 года назад. А до этого писал, но в другой форме немножко, более коротко, в хелпах там и такое вот. Вот мы смотрим здесь, что при увеличении объёма обучающей выборки классическая F-мера, она при малых значениях э-э объёма выборки, когда вообще мало объектов обучающей выборки, ну где-то до 300 примерно, она вообще очень неустойчивая и даёт очень сильно заниженную оценку достоверности. То есть если мы эту линию продолжим мысленно, то, конечно, идёт она по максимуму. Вот здесь вот где колебания, она там идёт по максимуму. Вы же видите, да? Вот здесь, вот сюда она идёт. Не вот так вот сюда, а вот она вот сюда идёт именно. То есть получается, что при э-э малых объёмах выборки классическая мера Ван Рисбергена может давать неоправданно заниженные оценки достоверности модели. Такой вот вывод. Я не помню, написал здесь статьи или нет, но в общем, я вам сейчас говорю.

Теперь смотрите, из-за того, что мы суммируем не единички к сумматорам ложных решений, а суммируем степень уверенности, из этого получается, что у нас достоверность модели где-то на 10 пунктов увеличивается. Ну, на 1/10, если максимум единицы, то где-то на 1/10 увеличивается достоверность. Вот это вот сиреневая – это э-э как ведёт себя мера мультиклассовая и нечёткая обобщение, 0,864. Вот, а э-э инвариантно относительно объёма данных 0,928. Ну есть же разница, 0,928, ребята, или 0, там, сколько там, 645, например. Ну есть разница большая. То есть 0,645 – это где-то на три с плюсом, три с минусом, не поймёшь, насколько там. Ну такая слабенькая модель, в общем, если честно. 0,864 – это уже так более-менее, ну, на четыре балла, там, если взять в экономику, психологию. А 0,928 – так это вообще отличная модель.

Смотрим теперь, как у нас э-э распределены количество истинных и ложных, положительных и отрицательных решений при разных э-э моделях и разных интегральных критериях. Значит, вот мы смотрим. Э-э модель F4 - сумма знаний. У неё вот так они распределены. Ну, здесь, может быть, не очень наглядно. Сейчас я вам, может быть, другую покажу, где более будет наглядно. Значит, вот смотрите, это модель F3 - Хи-квадрат. Обычно эта модель, она очень такая, как вот говорят сомелье, округлая, ну, выпуклая, то есть она очень хорошо выявляет зависимости действительные. То есть Пирсон молодец, что эту модель предложил. Она во многих отношениях очень интересная, там сумма же на нулю, средние равны нулю. Ну то есть это как бы очень такое интересное особенности. То есть она хорошо отцентрирована, скажем так. Вот мы видим, что это вот по оси X у нас уровень сходства. Здесь он положительный уровень сходства, здесь отрицательный. Истинные решения красным цветом и коричневым, а ложные решения синим. Ну видим, что решения отрицательные всегда истинные, ложных вообще нету ни одного. А положительные есть и ложные решения, и истинные. Но ложные решения, они все при маленьких уровнях сходства, до тридцати там трёх, там тридцати двух. А начиная с тридцати шести, там тридцати семи, только истинные решения. А это вот линия сглаженная, сглаженная с интервалом сглаживания 17 центральным, центральное сглаживание. То есть берётся точка, берётся влево и вправо одинаковое число отчётов, вычисляется среднее, ставится точка. И вот так вот сглаживание идёт. То есть мы видим, что у нас э-э при высоких уровнях сходства только одни истинные решения, при низких – только ложные решения. И есть зона промежуточная, где есть и истинные, и ложные решения. Да, давайте я эту картинку вам пошлю для прикола. Я не знаю, увидите вы её или нет, эту картинку там в чате. Наверное, увидите. О! Увидели.

И смотрим разность истинных и ложных решений. Ну здесь видим, что при до тридцати пяти, это тоже сглаженная линия 17, до тридцати пяти преобладают ложные решения положительные. Начиная с сорока, вообще ложных нет решений, преобладают истинные решения. То есть у нас есть конкретный критерий, который позволяет оценивать вот эту гистограммку, которую мы получили. Сейчас мы берём, задаём эту модель текущую, и в ней решаем задачу идентификации. Смотрим результат. То есть мы перешли на решение задачи, ребята. Смотрим результат. У нас э-э теперь э-э нету ни одной, ни одного ошибочного решения э-э положительного э-э с уровнем сходства больше тридцати пяти. То есть если так вот пролистать, есть там ложные решения, но нет ни одного, где уровень сходства выше тридцати пяти. Вот 35,8 самое большее. Все остальные… Что там? F3? Странно. F4. То есть у нас появляется чёткий критерий для того, чтобы отличить истинное решение от ложных. И можно сказать ещё вот что. Тот вывод, который я один раз уже сформулировал, что поскольку мы видим, что с увеличением уровня сходства доля истинных решений э-э растёт, то это означает, что система имеет внутренний э-э А, понял. Э-э, это у нас вот э-э по критерию сумма знаний. В системе есть чёткий критерий совершенно, позволяющий отличить э-э истинное решение от ложных.

**VIII. Заключение и дальнейшая работа**

Ну, ребята, значит, на этом мы это занятие заканчиваем. А для себя в расписании я пишу, что на следующем занятии мы рассматриваем решение задачи принятия решений, исследования классов самих и исследования модели. То есть обратная задача прогнозирования и исследования модели. И после этого, это не очень долго займёт времени, это там, не знаю, минут 20 займёт. А потом мы после этого начнём рассматривать решение задач обработки текстов, интеллектуальной обработки текстов, потом обработка графики, изображений, интеллектуальная тоже, формирование обобщённых э-э образов изображений, спектральных образов, там, э-э контурных образов, обобщённых. И после этого вы будете готовы э-э что-нибудь своё сделать под моим, так сказать, чутким руководством. То есть я буду вам помогать, подсказывать, проверять ваши модели, э-э все проблемы ваши буду стараться снять, которые у вас возникают. И расскажу вам, как определить своё место в рейтинге, международном рейтинге разработчиков интеллектуальных технологий. Чтобы вы там знали, где вы там находитесь. Ну, э-э с использованием инструмента системы Aidos или с инструментом, инструментов Python. Сейчас очень популярный. Вот, набирает быстро популярность. Уже Java обогнал, кажется. Да? Обогнал Java Python? Ну? Кто скажет?  
Что-что? Python Java обогнал по популярности?  
Собирает.  
Ну, на на пятки наступает, да? Ну ладно, ребят. Java всё-таки она тоже расплатформенная, всё-таки серьёзная очень. Ну вот. На этом занятие наше заканчивается, ребята. Всего самого хорошего вам. До свидания.  
До свидания. До свидания. До свидания. До свидания. До свидания.