МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ
Федеральное государственное автономное образовательное учреждение
высшего образования
«ТЮМЕНСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»

ИНСТИТУТ МАТЕМАТИКИ И КОМПЬЮТЕРНЫХ НАУК
Кафедра программного обеспечения

РЕКОМЕНДОВАНО К ЗАЩИТЕ В ГЭК
Заведующий кафедрой, к.т.н, доцент
_____М. С. Воробьева
23.06. 2023 г.

ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА
магистерская диссертация

РАЗРАБОТКА ОНЛАЙН ИНСТРУМЕНТА ДЛЯ ИНТЕРПРЕТАЦИИ ДАННЫХ МОНИТОРИНГА ОБРАЗОВАТЕЛЬНЫХ РЕЗУЛЬТАТОВ ПО ДИСЦИПЛИНАМ CORE

02.04.03 Математическое обеспечение и администрирование информационных систем
Магистерская программа «Разработка технологий Интернета вещей и больших данных»

Выполнил работу
студент 2 курса
очной
формы обучения
Фомин
Александр
Юрьевич

Научный руководитель
к.т.н., доцент
Воробьева
Марина
Сергеевна

Рецензент
заместитель директора,
Институт математики и
компьютерных наук
Перевалова
Мария
Николаевна

Тюмень
2023

ОГЛАВЛЕНИЕ

ВВЕДЕНИЕ

В результате продолжающейся цифровизации учебного процесса студентов происходит накопление большого массива данных об их учебной деятельности – их образовательных результатах. Наличием этого факта обуславливается необходимость в использовании информационных и интеллектуальных информационных систем.

Некоторые образовательные учреждения разрабатывают собственные информационные системы. Одним из примеров таких систем является информационная система (ИС) “Деканат”, являющаяся частью интегрированной системы управления учебным процессом (ИСУП) “Герцен” и разработанная отделом РГПУ им. А. И. Герцена [1]. ИС “Деканат” была разработана в результате потребности в вэб-ресурсах университета, способствующих эффективному обеспечению проектирования и реализации образовательных программ. Данная ИС взаимодействует с данными, передаваемыми другими ИС и веб-ресурсами в рамках ИСУП. ИС “Деканат” позволяет просматривать и редактировать информацию о студентах факультетов, формировать различные документы (ведомости на сдачу и пересдачу экзаменов, стипендиальные приказы, академические справки), а также получать статистическую информацию об успеваемости студентов.

Образовательные организации также используют и сторонние ИС, которые интегрируются в имеющуюся инфраструктуру. Примерами таких систем можно считать образовательную информационную систему Modeus и систему управления обучением Moodle.

Modeus является платформой, обеспечивающую автоматизацию учебного процесса в рамках модульного подхода и предоставляющую возможность управления индивидуальными образовательными траекториями в университетах [2]. С помощью данной системы автоматизируется процесс планирования нагрузки преподавателей и создания расписания. Также в системе ведется учет об успеваемости и результатах образовательной деятельности студентов.

Moodle является открытой системой для создания и управления онлайн-курсами [3]. В пределах курса, данная платформа позволяет формировать тесты и публиковать различные учебные материалы. В Moodle имеются функционал для оценки студентов и отслеживания их прогресса по курсу.

При этом важной частью таких систем являются инструменты анализа и визуализации данных о студентах, их образовательном процессе и результате. Данные инструменты помогают на индивидуальном уровне помочь понять происходящие процессы преподавателям дисциплин, а также администрации этих образовательных учреждений для выбора дальнейших организационных действий.

Для ИС “Деканат” такой инструмент анализа образовательной деятельности студентов реализован в виде отдельного веб-ресурса [4]. Данный инструмент позволяет формировать графики и отчеты по количеству отчисленных студентов в различных разрезах для последующей корректировки образовательного процесса. Например, имеется возможность формирования детальных отчетов об отчисленных студентах различных направлений в разрезе образовательных программ, что используются для выявления сложно осваиваемых дисциплин и отслеживания динамики движения контингента университета по годам.

В Moodle имеется базовый функционал для формирования отчетов об успеваемости и графиков активности на основе прохождения курса студентами. Также Moodle предоставляет возможность установки сторонних плагинов, позволяющих многократно расширить функционал платформы. Среди них есть и плагины, созданные для более детального отслеживания и визуализации данных студентов, например “Analytics graphs” [5] и “Overview statistics” [6]

Целью выпускной квалификационной работы является создание веб-дашборда, позволяющего визуализировать данные мониторинга образовательных результатов студентов дисциплин CORE для их последующей интерпретации.

Исходя из описанной цели были поставлены следующие задачи:

- Изучить предметную область – ознакомиться с существующими программами, позволяющими визуализировать и анализировать результаты образовательной деятельности студентов.
- Получить данные мониторинга образовательных результатов студентов дисциплин CORE.
- Провести предобработку и анализ полученных данных.
- Определить функционал приложения.
- Определить средства разработки для реализации приложения.
- Разработать и протестировать приложение.

Для успешной подготовки и защиты выпускной квалификационной работы обучающимся использовались средства и методы физической культуры и спорта с целью поддержания должного уровня физической подготовленности, обеспечивающую высокую умственную и физической работоспособность. В режим рабочего дня включались различные формы организации занятий физической культурой (физкультпаузы, физкультминутки, занятия избранным видом спорта) с целью профилактики утомления, появления хронических заболеваний и нормализации деятельности различных систем организма.

В рамках подготовки к защите выпускной квалификационной работы автором созданы и поддерживались безопасные условия жизнедеятельности, учитывающие возможность возникновении чрезвычайных ситуаций.

ГЛАВА 1. ОПИСАНИЕ ПРЕДМЕТНОЙ ОБЛАСТИ

Современное образование постоянно развивается, находя новые подходы и методы обучения. Однако, для эффективного обучения необходимы не только новые методы и подходы, но и глубокое понимание процесса обучения, и, что немаловажно, способность анализировать результаты этого процесса. Данные об учениках и процессе обучения являются ценным ресурсом, который, при правильном использовании, может улучшить качество обучения и сделать его более целевым и персонализированным.

Важным же компонентом в анализе данных об обучении является Learning Analytics (LA) или "аналитика обучения" - область, которая использует аналитику и методы анализа данных для изучения образовательного процесса и улучшения обучения и окружающей среды [7]. Цель LA – предоставить студентам, преподавателям и администрации университета информацию, которая поможет им принять обоснованные решения и улучшить образовательные результаты.

Существует множество разных методов LA в зависимости от решаемой задачи. Примером такой задачи является проверка данных на зависимости. Для проверки на зависимости производится корреляционный анализ. В статье [8] автором А. Н. Сокольниковым описывается применение корреляционного анализа с использованием коэффициента корреляции Пирсона. В описываемом исследовании проводится проверка на зависимости между контрольными точками (как по отдельности, так и среднее по всем значениям) и результатами экзамена по математической дисциплине, а также между контрольными точками и баллом ЕГЭ по предмету “Математика”. Исходя из значений коэффициента корреляции выдвигаются предположения о влиянии обучения студента до конкретного контрольного среза на итоговую оценку и влиянии баллов ЕГЭ на продолжающуюся учебную деятельность студента. Так, по данным рассматриваемого вуза МГИМО, для предмета “Математика” корреляция ЕГЭ и результатов контрольных срезов уменьшается для каждого последующего среза, что интерпретируется автором как уменьшение общего

влияния учебы студента в школе по сравнению с учебой в вузе.

Другой актуальной задачей является группирование студентов на основе каких-либо выделенных признаков. Чаще всего имеется необходимость формировать группы по неразмеченным данным, что производится с помощью методов кластерного анализа. В статье [9] авторами В. П. Арефьевым, А. А. Михальчуком и Н. М. Филипенко описывается проведение кластерного анализа для качественного оценивания результатов дистанционного тест-экзамена по высшей математике. Кластеризация велась по признакам DT (время, затраченное на экзамен), ЭКЗ (набранные баллы за экзамен) и ИДЗ (баллы за индивидуальные домашние задания) с использованием метода Kmeans. В рамках кластерного и дисперсионного анализа авторами была получена модель для 10 кластеров, качественно отделяющая группы студентов по результатам их образовательной деятельности и выделяющая кластеры с аномально низкими и высокими значениями признаков.

Таким образом, были выделены студенты с высокими баллами и крайне низким затраченным временем на экзамен, что интерпретировалось как потенциальные попытки обмана системы.

Одной из самых часто встречающихся задач LA в литературе и на практике является прогнозный анализ. Чаще всего данный вид анализа направлен на раннее выявление “проблемных” студентов для последующего корректирования образовательной деятельности. В статье [10] авторами Akçapınar G., Altun A. и Aşkar P. задача раннего прогноза успеваемости студента сводится к задаче бинарной классификации – разбиение на классы “сдал” и “не сдал”. В описываемом исследовании используемыми данными для классификации являлись различные показатели взаимодействия студента с системой управления обучением.

Так как для данной задачи нет единственного лучшего подхода, выбор метода классификации проводился на основе сравнения результатов работы нескольких популярных методов относительно чувствительности к выбору “проблемных” студентов.

LA часто использует визуализацию данных для представления результатов анализа в доступной форме в рамках удобного приложения-инструмента, что позволяет пользователям легко интерпретировать данные и делать на их основе обоснованные решения.

Чаще всего такие инструменты для анализа и визуализации образовательных процессов реализуются в виде отдельных веб-дашбордов. Примером дашборда для визуализации образовательного процесса является Student Activity Meter (SAM) [11]. Данный инструмент может отображать данные взаимодействия студентов с виртуальным учебным пространством – как, когда и сколько времени конкретные студенты взаимодействовали с конкретными ресурсами (например, веб-страницами), представляя это в виде временных графиков интенсивности активности выбранного курса, а также в виде параллельных координат для анализа конкретных действий студентов. SAM разрабатывался как для использования преподавателями для анализа групп студентов, так и студентами для учета собственной активности.

Из более поздних можно выделить веб-дашборд LADA, представляющий собой инструмент для визуализации исторических данных о конкретном студенте – его оценках (также в сравнении с другими студентами), раннее пройденных (и проваленных) курсах, а также текущих выбранных текущих курсах [12]. Ключевой частью LADA является функционал по помощи в выборе курсов студента. Основой данного функционала LADA является прогнозирование успешности обучения студентов, реализованное с помощью многоуровневого кластерного анализа. Фреймворк для кластеризации автоматически подбирает признаки различного уровня “глубины” (от общих до более частных) исходя из их иерархии и проводит кластеризацию на их основе. Пример различной “глубины” выбранных признаков для программы обучения: только количество выбранных курсов, количество выбранных курсов в пределах определенных тематик, метки конкретных выбранных курсов. При формировании плана обучения нового студента определяется кластер с самыми

похожими студентами, закончившими обучения. Процент студентов, успешно окончивших курсы для выбранного кластера, интерпретируется как вероятность успеха рассматриваемого студента. Непосредственно используемым алгоритмом кластеризации является Fuzzy CMeans – алгоритм нечеткой кластеризации, позволяющий одной точке данных находиться в нескольких кластерах с разным уровнем достоверности. В дашборде результат кластерного анализа визуально отображен как вероятность удачно завершить выбранные курсы.

Из дашбордов, ориентированных на визуализацию оценок, можно выделить LISSA [13]. Визуальной основой данного дашборда являются удобно расположенные графики-гистограммы, отображающие распределения оценок в конкретные периоды (по завершении контрольных точек) как в целом, так и по конкретным курсам. Цель LISSA – дать возможность увидеть результаты образовательных результатов студентов первого курса для корректирования их последующей работы.

В 2020 году Тюменский государственный университет приступил к переходу на новую образовательную модель “2+2” [14]. Главной особенностью данной модели является её двухуровневость – бакалавриат (и специалитет) разбиваются на два этапа: первые два года обучения и последующие. В течение первых двух лет обучения студенты осваивают общую для поступающих в ВУЗ студентов ядерную программу, которая позволяет развивать универсальные компетенции и дает время на то, чтобы определиться с направлением подготовки (при переходе на следующий этап его можно сменить). В течение последующих лет обучения студент находится на профессиональном треке, предполагающий развитие профессиональных компетенций и глубокое освоение дисциплин окончательно выбранного направления подготовки.

Ядерная программа – перечень обязательных общеобразовательных дисциплин первых двух лет обучения в университете. Данные дисциплины являются общими для всех студентов ядра, независимо от первоначально выбранного направления подготовки. Группы студентов дисциплин ядра не

делятся по направлениям подготовки – в пределах одной группы могут находится студенты совершенно разных направлений и институтов в пределах Тюменского государственного университета. Это приводит к тому, что в пределах дисциплины в образовательном процессе участвует очень большое количество студентов и преподавателей. Данные особенности программы усложняют контроль за образовательным процессом кураторами дисциплин.

Появляется необходимость агрегации, визуализации и интерпретации данных мониторинга образовательных результатов студентов в разрезах направлений подготовки и преподавателей по дисциплинам.

ГЛАВА 2. МЕТОДЫ И ТЕХНОЛОГИИ

2.1. ИСПОЛЬЗУЕМЫЕ МЕТОДЫ АНАЛИЗА И ВИЗУАЛИЗАЦИИ ДАННЫХ

Для проведения анализа на зависимость признаков был использован коэффициент корреляции Спирмана. Выбор данной характеристики обусловлен тем, что она позволяет определять монотонную зависимость и устойчива к выбросам.

Для проведения кластеризации данных было использовано три метода – метод k-средних (KMeans), алгоритм DBSCAN и метод иерархической агломеративной кластеризации. Метод KMeans – наиболее простой и популярный метод для кластеризации, действия алгоритма основано на последовательной минимизации суммарного квадратичного отклонения точек данных от центров кластеров [15]. На каждой итерации происходит перерасчет координат центроидов и переприсваивание точек данным ближайшему центроиду, алгоритм завершается, когда точки остаются в том же кластере на следующей итерации.

Алгоритм DBSCAN основан на предположениях о плотности [16]. Большое количество близко расположенных точек данных (область высокой плотности) группируются в кластеры, а одинокие точки с далеко расположенными соседями (область малой плотности) помечаются как выбросы.

Иерархическая аггломеративаная кластеризация – общий подход к иерархической кластеризации, основанный на подходе сверху-вниз – каждая точка добавляется в свой отдельный кластер и с каждой итерацией выбирается точки/кластеры с минимальным расстоянием и затем объединяются в общий кластер [17]. На каждой итерации, расстояние от кластера до точки или до другого кластера считается в зависимости от выбранного критерия связи (использовался “ward” – минимизирующий дисперсию объединяемых кластеров).

KMeans был выбран, так как является самым простым и популярным

алгоритмом, который хорошо работает для типовых задач кластеризации. DBSCAN и иерархический метод были выбраны в дополнение к KMeans, так как позволяют работать с различными метриками расстояния (KMeans однозначно сходится только используя Евклидово расстояние). Также большой плюс всех этим методов – они хорошо масштабируются в контексте размера датасета.

Для визуализации данных с множествами признаков был использован метод стохастического вложения с t-распределением (TSNE) – метод для визуализации данных высокой размерности [18], а также метод главных компонент (PCA) – метод для понижения размерности данных [19]. TSNE моделирует объект высокой размерности двух- или трёхмерной точкой таким образом, что похожие объекты моделируются близлежащими точками, а непохожие моделируются с высокой вероятностью точками, далеко отстоящими друг от друга. PCA основан на вычислении собственных значений и векторов матрицы ковариации признаков для поиска главных компонент – линейных комбинаций признаков, для которых дисперсия максимальна. Для визуализации на двумерном графике выбираются компоненты с максимальной дисперсией.

Кроме TSNE и PCA для визуализации данных высокой размерности могут быть использован ещё один метод – линейный дискриминантный анализ (LDA). LDA основан на поиске линейных комбинаций признаков, таких что, расстояние между точками уже известных разных классов максимально, а расстояние между точками одного класса минимально [20].

Метод TSNE был выбран так как данный метод был специально разработан для визуализации объектов на графиках. PCA был выбран как дополнительный метод для визуализации, который удачно отображает данные на основе дисперсии признаков.

2.2. ТЕХНОЛОГИИ ДЛЯ РАЗРАБОТКИ ИНСТРУМЕНТА

В качестве инструмента для первоначального анализа и визуализации данных были использованы среды Google Colab и JupiterLab (в составе платформы anaconda). Оба этих инструмента позволяют выполнять

произвольный код на языке Python 3 и отображать графики внутри файла-блокнота.

Для работы с данными используются следующие библиотеки Python:

- pandas – для работы с таблицами данных [21];
- numpy – для работы с массивами данных (в сочетании с pandas);
- statsmodel – для выполнения статистических тестов над данными [22];
- sklearn – методы для визуадизаци TSNE и PCA, методы для кластеризации KMeans, DBSCAN и AgglomerativeClustering [23].

Для визуализации данных использовались библиотеки plotly, matplotlib, seaborn.

Для разработки инструмента использовался язык программирования Python 3 с применением текстового редактора Visual Studio Code и библиотеки Streamlit [24]. Библиотека Streamlit разработана специально для облегчения разработки приложений/веб-дашбордов, направленных на анализ и визуализацию данных.

Для визуализации данных в виде таблиц и графиков использовались ранее упомянутые библиотеки pandas и plotly соответственно. Данные библиотеки поддерживаются streamlit. Выбор plotly обуславливается и тем, что данная библиотека формирует интерактивные веб-графики (возможность приближать, отдалять выбранный диапазон, скрывать легенды, показываться значения на графике в позиции курсора и т.д.).

ГЛАВА 3. ПРЕДОБРАБОТКА И АНАЛИЗ ДАННЫХ

3.1. ОПИСАНИЕ ДАННЫХ

Первоначальные данные – информация об оценках студентов и группах бакалавриата и специалитета по курсам Hard Core дисциплин первого семестра 2022-2023 года обучения в Тюменском Государственном Университете.

Данные были получены в результате запроса в систему Modeus куратором дисциплин.

Формат файла данных – .xlsx. Файл имеет четыре листа: первые три представляют из себя данные о результатах образовательной деятельности студентов предоставленных дисциплин, последний лист – список всех команд курсов и соответствующих им преподавателям по лекциям и практикам.

Структура таблицы студентов (1–3 лист):

Столбцы строкового типа:

- Название РМУП – название курса.
- Ссылка на РМУП – ссылка на курс в системе МОДЕУС.
- ФИО студента.
- Команда – группа студента.
- Тип встречи.
- Название встречи.
- Предмет контроля.
- Итоговая оценка – итоговая оценка по курсу (“отл.”, “хор.”, “удовл.”, “неудовл.”).

Столбец объектного типа:

- Оценка за предмет контроля (баллы и отметка о посещении);

Столбец вещественного типа:

- Итог ТУ – итоговое количество баллов по курсу.

Структура таблицы команд курсов (4 лист, все столбцы имеют строковый тип):

- Название РМУП – название курса.

- Ссылка на РМУП – ссылка на курс в системе МОДЕУС.
- Команда – группа курса.
- Цикл – тип занятий.
- Преподаватель по практике.
- Преподаватель по лекциям.

Также использовалась информация из двух дополнительных файлов формата. xlsx. В первом дополнительном файле находится информация о выбранных студентами направлений подготовки. Во втором дополнительном файле находится информация о том, является ли тот или иной студент иностранным студентом.

Таблицы из всех листов первоначального и дополнительных файлов были объединены в одну таблицу перед тем, как начать предобработку имеющихся данных. Чтобы избежать возможного дублирования записей при конкатенации таблиц с одинаковыми дисциплинами перед объединением таблицы проверялись на наличие повторений в дисциплинах попарно между друг другом. В результате проверки не было выявлено перекреста по дисциплинам между таблицами из листов – информация об образовательных результатах студентов для каждой отдельной дисциплины находилась в пределах только одного листа. Этот факт дал возможность без проблем объединить таблицы из листов в одну большую таблицу с результатами для всех имеющихся дисциплин.

После описанного шага, к объединенной таблице были добавлены столбцы с информацией о направлении подготовки и о том, является ли студент иностранным студентом.

3.2. ПРЕДОБРАБОТКА И ПРЕДВАРИТЕЛЬНЫЙ АНАЛИЗ ДАННЫХ

Изначально предоставленные данные – данные о студентах и командах, описанные в предыдущем разделе. Для формирования полной таблица была проведена проверка на пересечение информации между листами информации о студентах, после чего они были объединены.

Полная таблица информации о студентах имеет следующие характеристики:

- 488049 записей, каждая запись предоставляет информацию о конкретной учебной встрече студента, сопровождающаяся также финальным значением “Итог Ту” и “ Итоговая оценка”;
- Имеет информацию об 961 студенте для 158 уникальных предметов и 868 команд;
- Имеет большое количество пропущенных значений в столбцах “Оценка за предмет контроля”, “Итог ТУ” и “Итоговая оценка” (198971, 88911 и 183325 ссоответственно).

В результате предобработки данных были удалены дубликаты строк для некоторых студентов. При этом не все повторяющиеся строки являются дубликатами для удаления.

Например, РМУП был составлен так, что несколько встреч относится к одной тематике из-за чего они имеет одинаковое описание, и при этом студент имел на них одинаковые оценки. В случае наличия реальных дубликатов, количество строк на студента в пределах одной дисциплины будет больше, чем у других. С помощью проверки длины вектора оценок предметов контроля выявлялись студенты с неправильным количеством записей. Дубликаты записей для таких студентов удалялись вручную.

На рисунке 1 представлен график, показывающий, сколько предметов имеют соответствующее число студентов. Большинство предметов имеют лежат в диапазоне от 0 до 19 студентов, в то время как есть определенной количество предметов, с числом студентов, превышающим 900. К таким дисциплинам относятся “Алгебра”, “Математический анализ”, “Программирование и основы алгоритмизации” и некоторые другие. Большое количество дисциплин с таким малым количеством учащихся студентов связано с тем, что предоставленный файл данных содержал также курсы “Мастерских” с ограниченным количеством студентов.

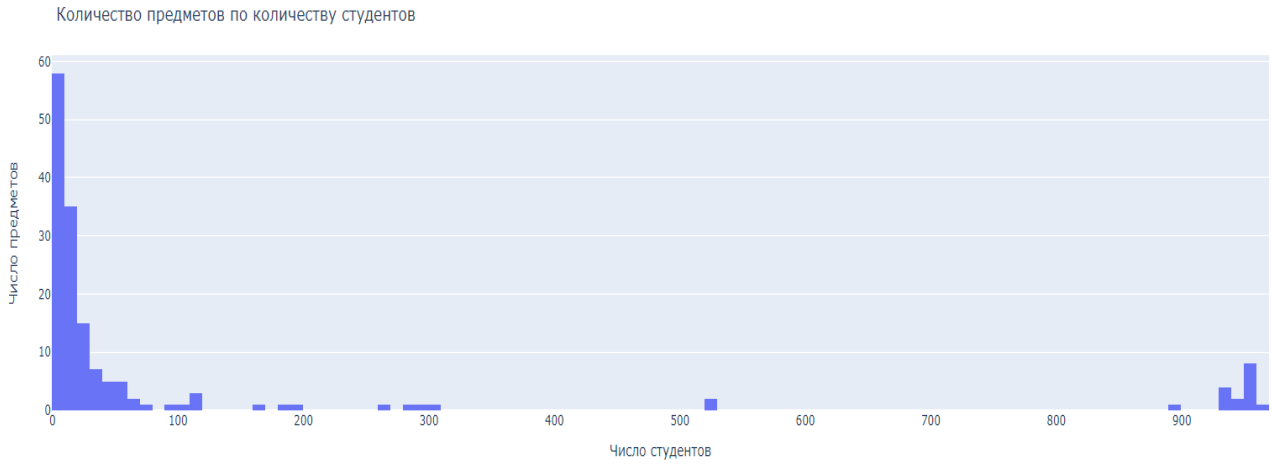
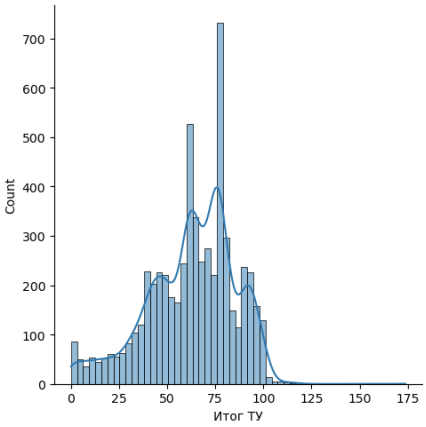


Рис. 1. Количество предметов по количеству студентов.

Для единственного числового столбца “Итог ТУ” на рисунке 2 показан график распределения и описательная статистика. Можно заметить, что на графике распределения виднеются бугры на значениях, близких к 61, 76 и 91. Данные значения соответствуют нижним границам оценки “удовлетворительно”, “хорошо” и “отлично” соответственно.



	Итог ТУ
count	399138.000000
mean	50.706709
std	31.541790
min	0.000000
25%	16.000000
50%	61.000000
75%	76.100000
max	173.800000

Рис. 2. Распределение значений и описательная статистика для столбца “Итог Ту”.

В результате обсуждений, было решено сконцентрировать внимание на трёх дисциплинах – “Алгебра”, “Математический анализ”, “Программирование и основы алгоритмизации 1”. Выбранные дисциплины входят в число ключевых математических дисциплин HardCore и соответственно имеют большее количество записанных студентов.

Для последующего анализа данных на зависимости были сформированы

дополнительные признаки:

- Посещаемость студентом курса (рассчитывается как кол-во “П” / (кол-во “П” + кол-во “Н”)).
- Доля полученных баллов по практическим занятиям (также используется вариант с абсолютными значениями).
- Доля полученных баллов по контрольным работам (также используется вариант с абсолютными значениями).
- Является ли математиком (студентом ИМиКН) (0 – нет, 1 – да).

Для выбранных дисциплин был проведен корреляционный анализ в разрезе преподавателей и отдельно их групп по парам признаков:

- Доля полученных баллов отдельно по контрольным и по практикам.
- Итоговый балл и посещаемость курса.

Для расчета корреляции использовался коэффициент Спирмана, отражающий наличие монотонной связи между переменными. Таблицы с результатами расчета корреляции представлены на рисунках 3 и 4.

Анализируя значения коэффициента корреляции между контрольными и практиками, представленные на рисунке 3, можно прийти к следующим выводам:

Для дисциплины “Алгебра”:

Имеется очень слабая (до 0.2) зависимость для следующих групп:

- Команда П-03.01;
- П-01.02 Спорт Прогрм. – 5 человек набрали 80+ баллов за практики, еще несколько ~60 баллов, при этом не набирая баллы за контрольные, из-за чего только для них формируется обратная тенденция;
- П-04.02 – похожая на Спорт Прогрм., но не настолько сильно;
- П-02.03 – лучшая по среднему баллу, не имеет неудов.

Для дисциплины "Программирование и основы алгоритмизации”:

Команда ПиОА П-01.01 Спорт Прогрм. имеет заметную обратную зависимость (от -0.7 до -0.5). Команда имеет большое количество студентов,

получивших большое количество баллов исключительно за практики. Судя по всему, они получали дополнительные баллы за участия на различных олимпиадах, которые были добавлены к практикам.

В целом:

Для большинства команд имеется заметная (от 0.5 до 0.7) и сильная (от 0.7 до 0.9) монотонная зависимость. Данная тенденция нарушается в тех случаях, когда студентам дается возможность набирать большое количество баллов на практиках. Это приводит к тому, что студент набирает необходимое количество баллов для ожидаемой оценки без необходимости хорошо писать контрольные работы.

Преподаватель по практике	Команда	Практики / Контрольные
АлгП-0	АЛГЕБРА П-09.03	0.803335
	АЛГЕБРА П-08.02	0.779166
	АЛГЕБРА П-01.01	0.778543
АлгП-1	АЛГЕБРА П-05.04	0.861208
	АЛГЕБРА П-06.02	0.680576
	АЛГЕБРА П-04.01	0.647039
	АЛГЕБРА П-03.01	0.109333
АлгП-10	АЛГЕБРА П-07.04	0.849800
	АЛГЕБРА П-09.04	0.839321
АлгП-11	АЛГЕБРА П-02.01	0.758023
	АЛГЕБРА П-05.01 Спорт Прогрм	0.607264
	АЛГЕБРА П-01.03	0.347965
АлгП-12	АЛГЕБРА П-01.02 Спорт Прогрм	-0.068988
АлгП-13	АЛГЕБРА П-01.04	0.853088
АлгП-2	АЛГЕБРА П-07.02	0.814904
	АЛГЕБРА П-03.04	0.774835
	АЛГЕБРА П-05.02	0.675813
	АЛГЕБРА П-06.03	0.517735
	АЛГЕБРА П-04.02	0.191139
АлгП-3	АЛГЕБРА П-07.01 Э ВШЭ	0.880226
	АЛГЕБРА П-08.01Э ВШЭ	0.844216
	АЛГЕБРА П-09.02 Э ВШЭ	0.752630
АлгП-4	АЛГЕБРА П-05.03	0.912742
	АЛГЕБРА П-06.04	0.905947
	АЛГЕБРА П-08.04	0.857503
	АЛГЕБРА П-02.04	0.823159
АлгП-5	АЛГЕБРА П-03.02	0.801551
АлгП-6	АЛГЕБРА П-07.03	0.928590
	АЛГЕБРА П-03.03	0.743529
	АЛГЕБРА П-04.03	0.469419
	АЛГЕБРА П-02.03	0.010579
АлгП-7	АЛГЕБРА П-09.01	0.879371
	АЛГЕБРА П-08.03	0.830820
АлгП-8	АЛГЕБРА П-06.01	0.716473
	АЛГЕБРА П-04.04	0.657939
АлгП-9	АЛГЕБРА П-02.02	0.765914

Преподаватель по практике	Команда	Практики / Контрольные
ПрогП-0	ПиОА П-09.03	0.870028
	ПиОА П-07.01	0.834495
	ПиОА П-08.02	0.739500
ПрогП-1	ПиОА П-04.03	0.739303
	ПиОА П-05.04	0.698482
ПрогП-10	ПиОА П-08.01	0.665301
ПрогП-11	ПиОА П-07.03	0.835599
	ПиОА П-03.03	0.828507
	ПиОА П-04.02 Спорт Прогрм	0.787006
ПрогП-2	ПиОА П-06.04	0.845857
	ПиОА П-04.01	0.717880
	ПиОА П-09.01	0.606399
	ПиОА П-01.01 Спорт Прогрм	-0.604591
ПрогП-3	ПиОА П-07.02	0.832958
	ПиОА П-02.03	0.687220
ПрогП-4	ПиОА П-06.01	0.791300
	ПиОА П-09.04	0.763830
ПрогП-5	ПиОА П-05.01	0.803857
	ПиОА П-08.04	0.753427
	ПиОА П-04.04	0.619570
ПрогП-6	ПиОА П-05.02	0.896641
	ПиОА П-06.03	0.794952
	ПиОА П-01.04	0.777497
	ПиОА П-02.01	0.673250
	ПиОА П-03.01	0.589060
ПрогП-7	ПиОА П-06.02	0.901732
	ПиОА П-05.03	0.862312
	ПиОА П-07.04	0.858410
	ПиОА П-02.04	0.831798
	ПиОА П-03.02	0.578106
ПрогП-8	ПиОА П-02.02	0.655550
	ПиОА П-01.02	0.644367
	ПиОА П-03.04	0.546470
ПрогП-9	ПиОА П-09.02	0.732318
	ПиОА П-01.03	0.542349
	ПиОА П-08.03	0.454269

Преподаватель по практике	Команда	Практики / Контрольные
МатП-0	МА П-02.04	0.877080
	МА П-07.02	0.844248
	МА П-05.02	0.774139
	МА П-08.02	0.510925
МатП-1	МА П-07.03	0.858523
	МА П-03.03	0.713074
	МА П-02.03	0.622745
	МА П-06.03	0.588379
	МА П-04.01	0.493313
МатП-10	МА П-01.04	0.568674
	МА П-09.04	0.463736
МатП-11	МА П-01.03	0.555029
МатП-2	МА П-06.02	0.909433
	МА П-05.04	0.773421
	МА П-01.02	0.453835
МатП-3	МА П-04.02	0.757081
	МА П-06.04	0.744602
	МА П-02.02	0.648353
МатП-4	МА П-07.01 Э ВШЭ	0.797149
	МА П-09.02 Э ВШЭ	0.673977
	МА П-08.01 Э ВШЭ	0.632804
МатП-5	МА П-07.04	0.846504
	МА П-08.04	0.736313
	МА П-06.01	0.593885
МатП-6	МА П-02.01	0.879750
	МА П-08.03	0.753997
МатП-7	МА П-05.01	0.678882
	МА П-05.03	0.626867
	МА П-04.03	0.563928
	МА П-04.04	0.485065
	МА П-03.02 СпортПрогр	0.477341
	МА П-03.04	0.429043
	МА П-01.01 Спорт Прогрм	0.235385
МатП-8	МА П-09.01	0.775758
	МА П-03.01	0.309372
МатП-9	МА П-09.03	0.768594

Рис. 3. Таблицы корреляций доли полученных баллов по практикам и контрольным работам в разрезе преподавателей и их команд.

Анализируя значения корреляций между итоговым баллом и посещаемостью, представленные на рисунке 4, можно прийти к следующим выводам:

Для дисциплины “Алгебры”:

Имеется очень слабая (до 0.2) зависимость для следующих групп:

- П-01.02 Спорт Прогрм. – у большинства идеальная посещаемость, а оценки изменяться от 25 до 100 баллов.

Для дисциплины “Программирование и основы алгоритмизации”:

Имеется очень слабая (до 0.22) зависимость для следующих групп:

- П-02.03 – просто очень слабая корреляция (оценка практически не зависит от посещения).
- П-01.01 Спорт Прогрм. – хорошая посещаемость (больше 80% посещений), оценка изменяется независимо от посещаемости.
- П-08.01 – Не число (nan), так как идеальная посещаемость абсолютно у всех (корреляция с вектором одинаковых значений), что приводит к такой ошибке.

Для дисциплины “Математический анализ”:

Имеется очень слабая (до 0.2) зависимость для следующих групп:

- П-08.02 – оценки и посещаемость действительно мало коррелируют, несмотря на разброс данных значений.
- МА П-01.01 Спорт Прогрм и МА П-01.02 – аналогичная ситуация с высокой посещаемостью, но крайне разными показателями итоговых баллов (большая часть больше 60).

В целом: для большинства команд имеется заметная (от 0.5 до 0.7) монотонная зависимость. Данная тенденция нарушается в тех случаях, когда у студентов в командах имеется посещаемость близкая к идеальной, но при этом сильно разнятся оценки. В таких случаях отсутствие на одной или двух парах не влияет на оценку – для таких студентов значение баллов может находится в диапазоне от 0 до 100, из-за чего при расчёте корреляции оказывается, что никакой зависимости не обнаруживается.

Преподаватель по практике	Команда	Итог ТУ Посещаемость
АлгП-0	АЛГЕБРА П-09.03	0.744885
	АЛГЕБРА П-08.02	0.519067
	АЛГЕБРА П-01.01	0.416148
АлгП-1	АЛГЕБРА П-05.04	0.846794
	АЛГЕБРА П-04.01	0.813515
	АЛГЕБРА П-06.02	0.803878
	АЛГЕБРА П-03.01	0.664031
АлгП-10	АЛГЕБРА П-09.04	0.729133
	АЛГЕБРА П-07.04	0.659895
	АЛГЕБРА П-02.01	0.417172
АлгП-11	АЛГЕБРА П-05.01 Спорт Програм	0.916070
	АЛГЕБРА П-01.03	0.674776
АлгП-12	АЛГЕБРА П-01.02 Спорт Програм	0.172053
АлгП-13	АЛГЕБРА П-01.04	0.685158
АлгП-2	АЛГЕБРА П-05.02	0.836921
	АЛГЕБРА П-06.03	0.706350
	АЛГЕБРА П-04.02	0.662569
	АЛГЕБРА П-03.04	0.645358
	АЛГЕБРА П-07.02	0.604637
АлгП-3	АЛГЕБРА П-09.02 Э ВШЭ	0.771107
	АЛГЕБРА П-08.01Э ВШЭ	0.765045
	АЛГЕБРА П-07.01 Э ВШЭ	0.715419
АлгП-4	АЛГЕБРА П-06.04	0.788055
	АЛГЕБРА П-02.04	0.672458
	АЛГЕБРА П-05.03	0.637159
	АЛГЕБРА П-08.04	0.582493
АлгП-5	АЛГЕБРА П-03.02	0.553104
АлгП-6	АЛГЕБРА П-03.03	0.921428
	АЛГЕБРА П-07.03	0.785862
	АЛГЕБРА П-04.03	0.703439
	АЛГЕБРА П-02.03	0.588584
АлгП-7	АЛГЕБРА П-09.01	0.772808
	АЛГЕБРА П-08.03	0.714780
АлгП-8	АЛГЕБРА П-04.04	0.775595
	АЛГЕБРА П-06.01	0.744035
АлгП-9	АЛГЕБРА П-02.02	0.831764

Преподаватель по практике	Команда	Итог ТУ Посещаемость
ПрогП-0	ПиОА П-09.03	0.775233
	ПиОА П-08.02	0.752955
	ПиОА П-07.01	0.592717
ПрогП-1	ПиОА П-04.03	0.720873
	ПиОА П-05.04	0.704090
ПрогП-10	ПиОА П-08.01	nan
	ПиОА П-04.02 Спорт Програм	0.763582
ПрогП-11	ПиОА П-07.03	0.663155
	ПиОА П-03.03	0.562414
ПрогП-2	ПиОА П-06.04	0.822321
	ПиОА П-04.01	0.717227
	ПиОА П-09.01	0.706915
	ПиОА П-01.01 Спорт Програм	0.146535
ПрогП-3	ПиОА П-07.02	0.660566
	ПиОА П-02.03	0.216394
ПрогП-4	ПиОА П-09.04	0.696010
	ПиОА П-06.01	0.625070
ПрогП-5	ПиОА П-05.01	0.838866
	ПиОА П-04.04	0.806169
	ПиОА П-08.04	0.786325
ПрогП-6	ПиОА П-05.02	0.826049
	ПиОА П-01.04	0.824731
	ПиОА П-06.03	0.633645
	ПиОА П-03.01	0.459664
	ПиОА П-02.01	0.358161
ПрогП-7	ПиОА П-02.04	0.830102
	ПиОА П-06.02	0.817143
	ПиОА П-07.04	0.738924
	ПиОА П-05.03	0.540245
	ПиОА П-03.02	0.507204
ПрогП-8	ПиОА П-02.02	0.726741
	ПиОА П-03.04	0.570250
	ПиОА П-01.02	0.420201
ПрогП-9	ПиОА П-01.03	0.774038
	ПиОА П-09.02	0.706417
	ПиОА П-08.03	0.470668

Преподаватель по практике	Команда	Итог ТУ Посещаемость
МатП-0	МА П-05.02	0.774473
	МА П-07.02	0.727605
	МА П-02.04	0.707427
	МА П-08.02	0.183362
МатП-1	МА П-04.01	0.742867
	МА П-03.03	0.715029
	МА П-07.03	0.685701
	МА П-06.03	0.568498
	МА П-02.03	0.460281
МатП-10	МА П-01.04	0.759867
	МА П-09.04	0.709916
МатП-11	МА П-01.03	0.715793
МатП-2	МА П-05.04	0.784845
	МА П-06.02	0.782358
	МА П-01.02	0.182360
МатП-3	МА П-02.02	0.774601
	МА П-06.04	0.753814
	МА П-04.02	0.670203
МатП-4	МА П-07.01 Э ВШЭ	0.746492
	МА П-08.01 Э ВШЭ	0.639153
	МА П-09.02 Э ВШЭ	0.638342
МатП-5	МА П-08.04	0.677990
	МА П-07.04	0.648460
	МА П-06.01	0.308609
МатП-6	МА П-02.01	0.732376
	МА П-08.03	0.724789
МатП-7	МА П-04.04	0.698703
	МА П-05.01	0.549863
	МА П-03.04	0.523894
	МА П-05.03	0.452655
	МА П-04.03	0.408154
	МА П-03.02 СпортПрогр	0.350884
	МА П-01.01 Спорт Програм	-0.261284
МатП-8	МА П-03.01	0.557978
	МА П-09.01	0.403449
МатП-9	МА П-09.03	0.432711

Рис. 4. Таблицы корреляций итоговых баллов и значениями посещаемости студента в разрезе преподавателей и их команд.

Для выбранных дисциплин были рассмотрены графики распределения итоговых баллов, представленные на рисунке 5. На графиках видно, что для всех дисциплин характерна следующая ситуация: заметны скачки в количестве студентов для диапазонов баллов 57-62, 72-77 и 87-92. Это связано с тем, что количество баллов равное 61, 76 и 91 дают возможность получить автоматом оценки, равные 3 (удовлетворительно), 4 (хорошо) и 5 (отлично) соответственно.

Также для всех дисциплин можно заметить то, что имеются студенты, набравшие больше 100 баллов. В случае с алгеброй имеется странная ситуация – ученик получил 172 балла, причем 95 баллов было выставлено за работу на экзамене.

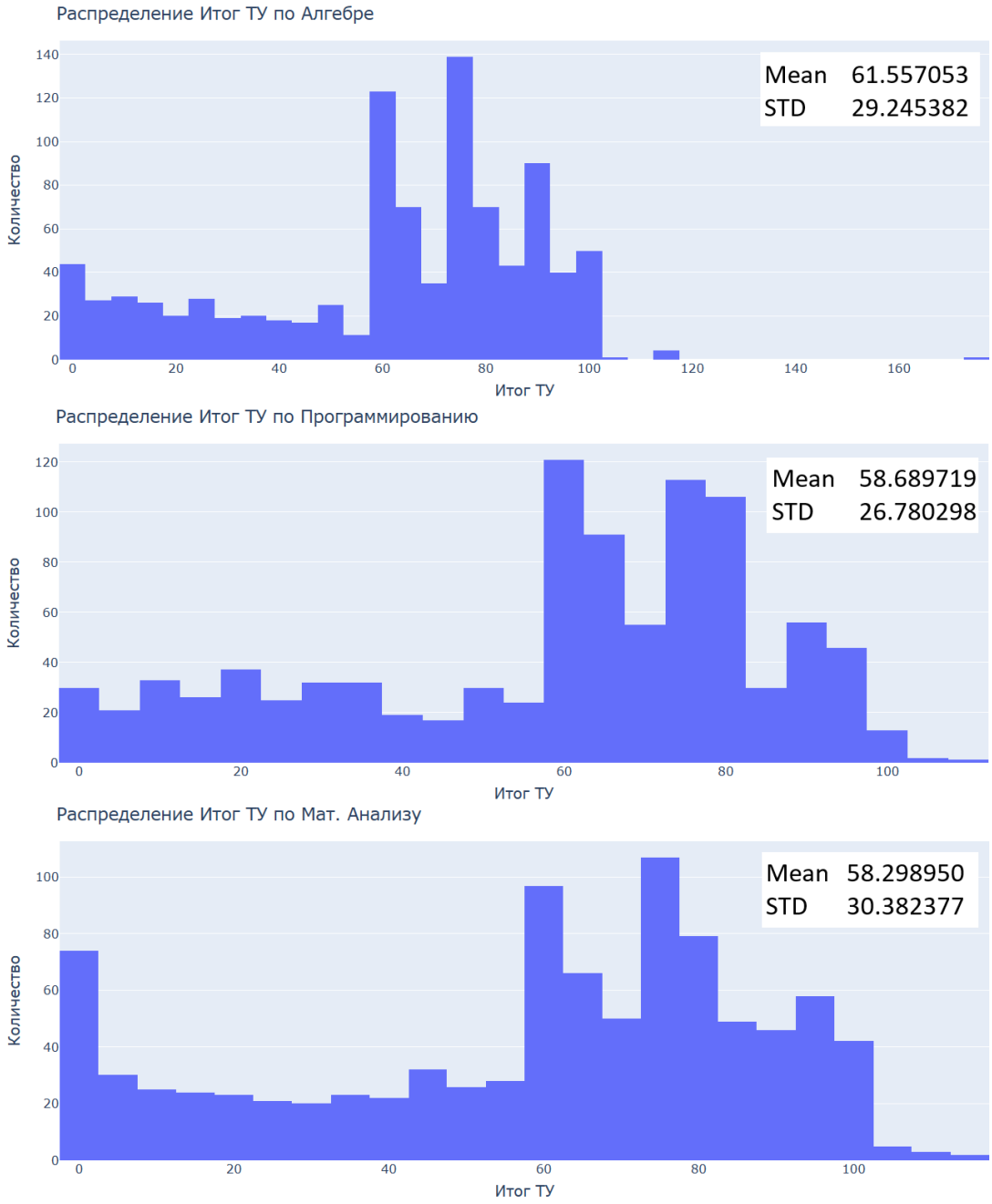


Рис. 5. Гистограмма частот итоговых баллов и график средних значений итоговых баллов в разрезе групп по выбранным дисциплинам.

На графике с дисциплиной “Математический анализ” можно заметить, что имеется очень большое количество студентов с крайне низкими баллами в диапазоне 0 – 2.4 балла – 74 студента. Для двух других дисциплин количество таких студентов составило 30 и 44, при том, что общее количество студентов для этих дисциплин примерно одинаково.

На рисунке 6 представлены графики средних значений итоговых баллов в разрезе групп для выбранных дисциплин.

Для дисциплины “Алгебра”:

Худший результат показала команда АЛГЕБРА П-07.04 (АлгП-10) – команда, собранная из студентов не математических направлений. Команда также имеет около половины оценок “неудовлетворительно”.

Лучший результат показала АЛГЕБРА П-02.03 (АлгП-6) – команда, собранная из студентов математических направлений (ИМиКН). Также, все ученики из этой команды получили положительную оценку на экзамене (не имеют оценок “неудовлетворительно” и непосещений экзамена).

Для дисциплины “Программирование и основы алгоритмизации”:

Худший результат показала ПиОА П-05.04 (ПрогП-1) – команда со всеми математиками, при этом все являются иностранными студентами, практически все сдали на “удовлетворительно”.

Лучший результат показала ПиОА П-01.01 Спорт Прогрм (ПрогП-2) – команда отличников математических направлений, практически все сдали на “хорошо” и “отлично” с высокими баллами.

Для дисциплины “Математический анализ”:

Худший результат показала команда МА П-04.04 (МатП-7) – команда из направлений ИМиКН, иностранные студенты. Большая часть оценок команды – “удовлетворительно”.

Лучший результат показала МА П-03.01 (МатП-8) – команда студентов математических направлений с высокой посещаемостью.

По всем дисциплинам:

Команды, набранные из студентов математических направлений ИМиКН, показывают лучшие результаты. Команды, составленные из иностранных студентов, чаще показывают наихудшие результаты.

Рис. 6. График средних значений итоговых баллов в разрезе групп по выбранным предметам.

Было решено рассмотреть отдельно группы студентов Высшей Школы Экономики (ВШЭ) и Спортивного программирования для дисциплин “Алгебра”, “Программирование и основы алгоритмизации”, “Математический анализ”.

Общая картина для каждой группы аналогична распределению баллов по

всей соответствующей дисциплине, где для отдельных групп свойственно то, что оценок больше всего около баллов, соответствующим нижней границе получения оценки “автоматом”. При этом большая часть финальных баллов студентов находятся в диапазоне для получения соответствующих им оценки “автоматом”. Это может говорить о том, что студенты чаще соглашаются со своей текущей оценкой и не стараются улучшить свою оценку на экзаменационных мероприятиях.

По алгебре команды ВШЭ показали себя хуже, чем команды по Спортивному Программированию (см. рис.7). Для команд ВШЭ больший процент оценок “удовлетворительно” и больший разброс в финальных баллах. Можно отметить, что студенты команды П-07.01 получили много оценок “хорошо”, заработав примерно 76 баллов, что соответствует нижней границе получения данной оценки автоматом.

Команды Спортивного программирования в основном имеют оценки “хорошо” и “отлично”. Можно отметить, что команда П-01.02 имеет высокую долю отличников “стобалльников”.

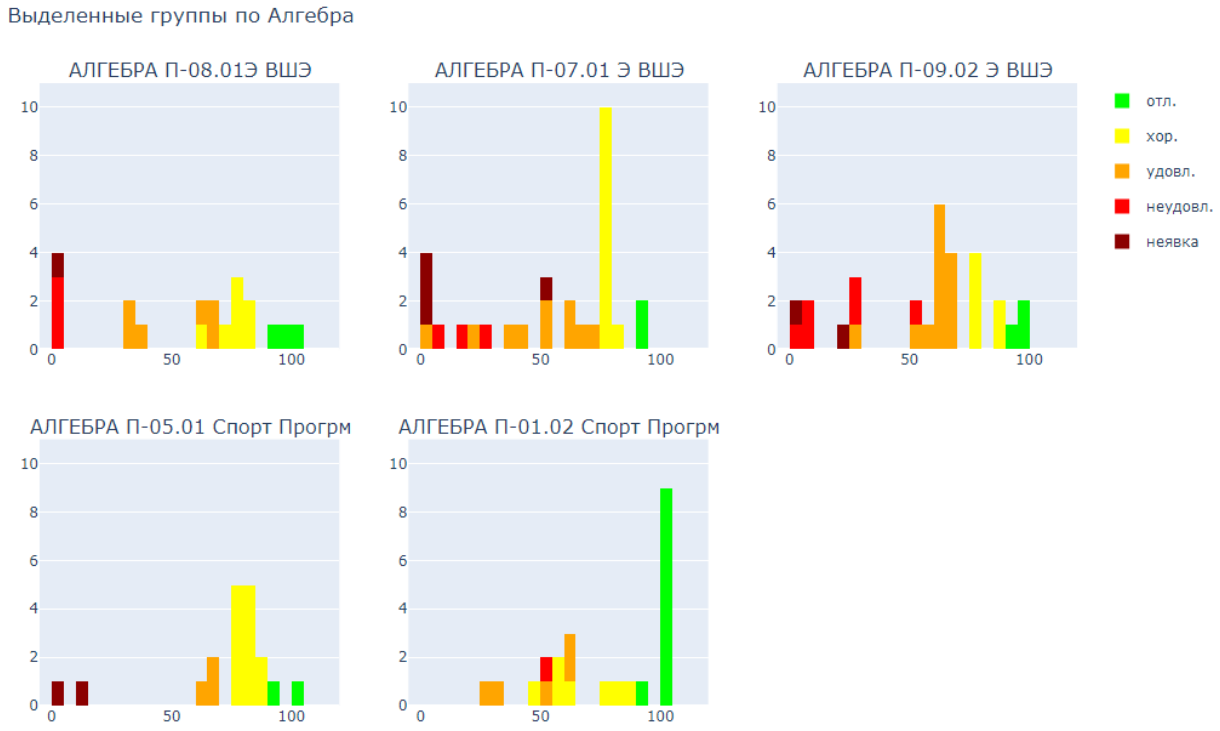


Рис. 7. Гистограмма распределения итоговых баллов для выделенных команд по дисциплине “Алгебра”.

Для программирования не имелось отдельно выделенных групп ВШЭ, студенты ВШЭ были распределены между несколькими разными командами (см. рис.8). Между группами Спортивного программирования имеется заметная разница – команда П-01.01 имеет меньший разброс по баллам и не имеет не сдавших экзамен студентов, при этом имея больше всего “хорошистов” и “отличников”. При этом среди сдавших для команды П-04.02 имеется больше “отличников” чем “хорошистов” в отличии от П-01.01, в которой студентов с оценками “хорошо” и ”отлично” примерно поровну.

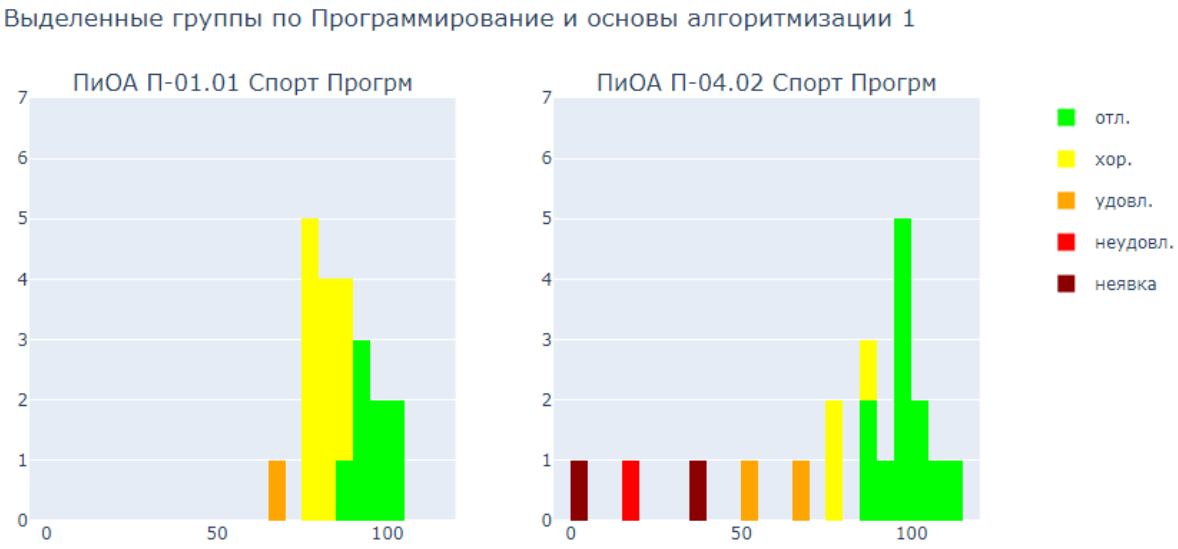


Рис. 8. Гистограмма распределения итоговых баллов для выделенных команд по дисциплине “Программирование и основы алгоритмизации”.

Группы Спортивного Программирования показали себя хуже всего на предмете математический анализ (см. рис.9). Количество не сдавших для этих групп заметно больше, чем для групп СП других дисциплин.

Для групп ВШЭ можно отметить, что в команде П-09.02 заметное число “хорошистов”, которые имеют баллы, близкие к нижней границе оценки “хорошо”.

В целом, видна та же ситуация, что и на более глобальных графиках распределения оценок – большинство студентов получает такие оценки, баллы которых позволяют. Очевидно, что исключением является диапазон оценки не сдачи, там собирается заметное количество студентов, которые не смогли

набрать минимальный проходной балл.

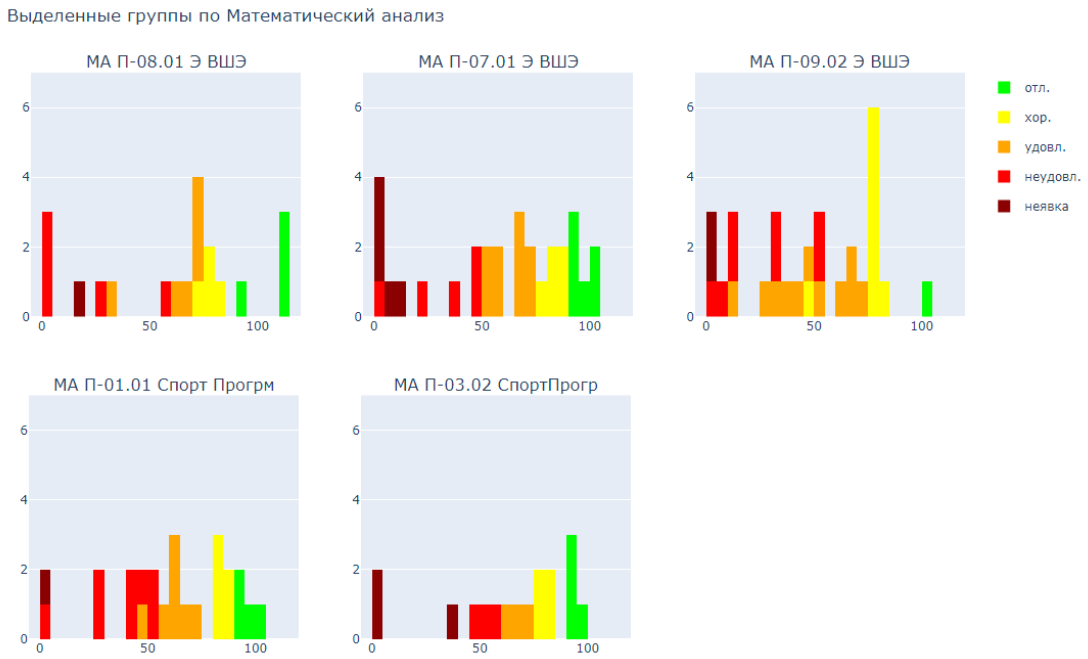


Рис. 9. Гистограмма распределения итоговых баллов для выделенных команд для дисциплины “Математический анализ”.

3.3. КЛАСТЕРНЫЙ АНАЛИЗ ДАННЫХ

Для каждого из выбранных предметов проводилась кластеризация с использованием методов KMeans, DBSCAN и агломертивной иерархической кластеризации.

Кластеризация проводилась для ответа на следующие вопросы (проверка гипотез):

1. Преподаватели выставляют оценки по одному принципу?
2. Студенты делятся на группы, исходя их посещаемости?
3. Студенты явно разделяются на плохих, средних и хороших, исходя из их образовательных результатов по дисциплине?

Для проведения кластерного анализа из столбца “Оценка за предмет контроля” полученной таблицы был сформирован вектор признаков для каждого студента. При этом данный столбец имеет пропуски в данных, как по баллам, там и по посещаемости. Для проведения кластерного анализа было необходимо заполнить данные про. Было сформировано два подхода для формирования данного вектора:

- Все значения вектора заменялись на 1 (если была выставлена оценка за предмет контроля) и 0 (если оценка не была выставлена);
- Числовые значения оригинального столбца оставались без изменения – пропущенные баллы заменялись на нули (в соответствии с итоговыми баллами), отметки по посещениям были заменены на значения номинальной шкалы, где 0 – пропуск, 1 – “Н”, 2 – “П”.

Кластеризация проводилась:

1. По всему вектору признаков (все оценки и отметки по посещаемости).
2. По вектору отметок по посещаемости.
3. Отдельно по контрольным точкам дисциплины.

Для выбора количества кластеров для алгоритма KMeans использовались методы локтя и силуэта. Выбор количества кластеров для иерархической кластеризации производился исходя из построенной дендрограммы.

Кластеризация по всему вектору признаков

Для того чтобы проверить, формируются ли кластеры по преподавателям, был использован метод TSNE для понижения размерности и визуализации. Хоть метод TSNE не является методом для кластеризации, он может помочь определить некоторые группы близких объектов, так как TSNE хорошо сохраняет локальную структуру. Дополнительно, был использован метод KMeans для проведения кластеризации и последующего кластерного анализа.

На рисунке 10 представлен результат визуализации вектора признаков для первого подхода и матрица ошибок кластеризации, сравнивая по преподавателю. Графики слева и по центру отображают точки-векторы признаков студентов, при этом в графиках слева цвет сопоставлен с преподавателем по практике данного студента, а в графиках по центру цвет сопоставлен кластеру, в который попал студент.

Матрица ошибок была приведена как визуализация наполнения кластеров в разрезе преподавателей – для удобного понимания в какой из кластеров попал тот или иной преподаватель.

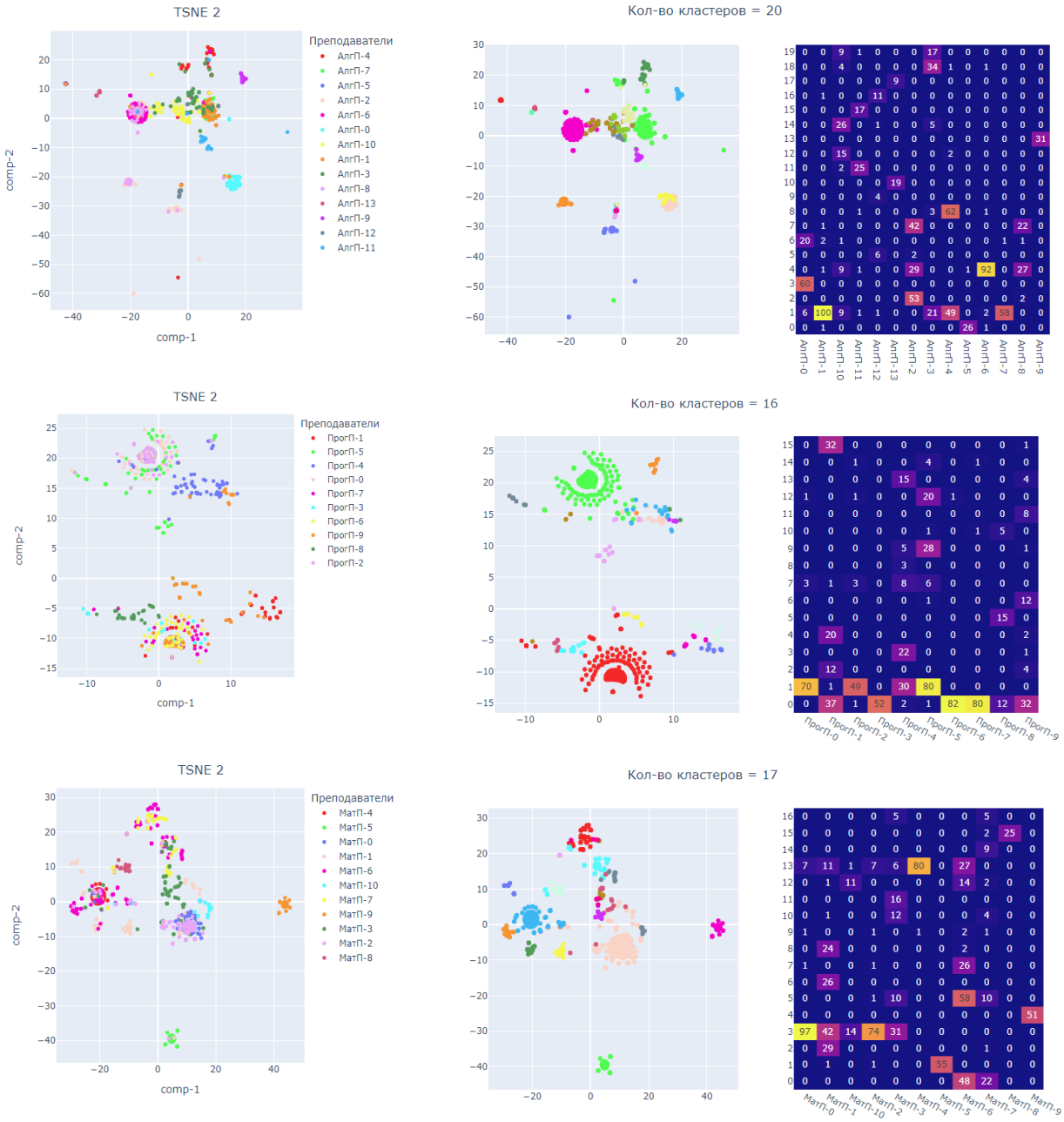


Рис. 10. Визуализация векторов и кластеризация для выбранных дисциплин – “Алгебра”, “Программирование и основы алгоритмизации”, “Математический анализ”.

Для “Алгебра” – рассматривая TSNE – отделились следующие: АлгП-0 (первые пять встреч, кроме одной не заполнены), АлгП-11 (не ставил оценки за первые практики), АлгП-12 (некоторые студенты переводные, редко отмечает посещаемость, группа “Спорт программирования”), АлгП-9 (выставляет все посещения кроме трех пар в середине семестра). Большие кластеры 1 – не выставляются оценки за практики, 4 – заполняются полностью, в том числе

нули.

Для “Программирование и основы алгоритмизации” – два больших кластера 0 и 1, обусловлены разностью заполнения практических занятий и контрольных работ. В отличии от алгебры, два основных кластера 0 и 1 студентами с большинством различных преподавателей. Это говорит о том, что большая часть из преподавателей выставляет значения сходным образом. Данные кластеры заполнены преподавателями, заполняющей полностью оценки и отметки по посещениям.

Для “Математический анализ” – Kmeans, как метод кластеризации показывал подобные результаты и визуально сближенные TSNE группы студентов также были собраны в одни кластеры методом Kmeans. Хорошо отделились (с точки зрения преподавателей) группы студентов с преподавателями МатП-5 (выставляет все посещения и только контрольные точки) и МатП-9 (выставляет все посещения и контрольные точки, и было два занятия, где выставлялись баллы за практическое задание).

На рисунке 11 представлен результат визуализации вектора признаков для второго подхода, в котором были использованы данные в первоначальном виде с раннее описанном способе заполнения значений с пропусками. В данном случае представлены только графики визуализации векторов студентов относительно преподавателей.

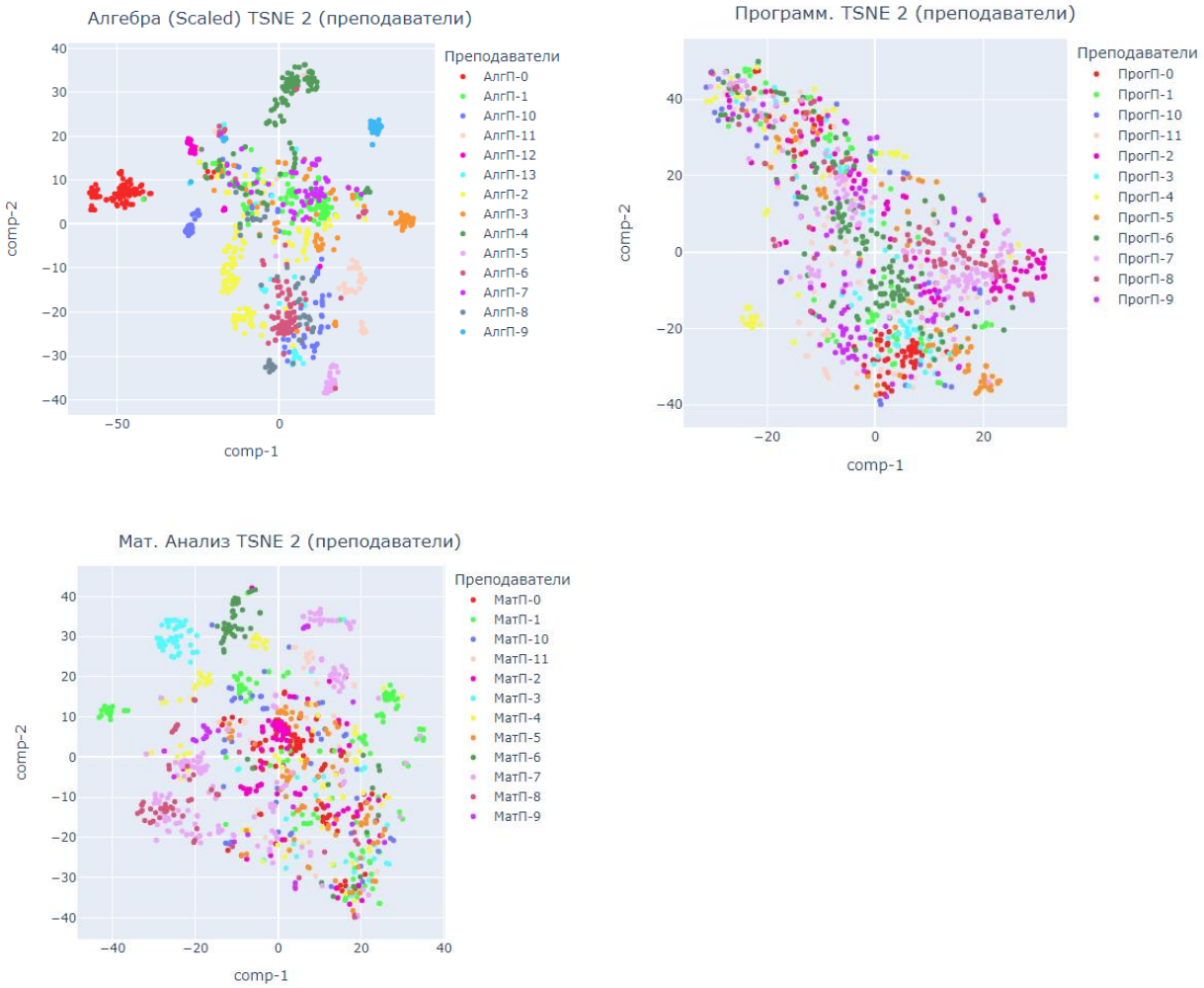


Рис. 11. Визуализация векторов после понижения размерности для выбранных дисциплин.

Визуально относительно удачно отделились по преподавателям:

1) АлгП-0 (полностью);
2) АлгП-2 (частично, команды размазаны так же, как и весь кластер, но есть две команды которые отобразились отдельно и не перемешались с другими командами – АЛГЕБРА П-03.04 и П-05.02);
3) АлгП-3 (препод для ВШЭ – частично, команды визуально перемешаны так же, как и весь кластер).

Также были изменения после Scale:

1) АлгП-4, АлгП-9, АлгП-10 (команда П-07.04 полностью отделилась), АлгП-11, АлгП-5 сильно отделились.

Иностранные студенты для дисциплины “Алгебра” – в данном случае студенты иностранцы хорошо не отделились, но для группы АлгП-8 Алгебра П-04-04 большее число иностранных студентов также хорошо отделились – как и для бинарного случая.

В целом, для дисциплин “Программирование и основы алгоритмизации” и “Математический анализ” изменений не обнаруживается – иностранные и студенты с малой посещаемостью пропадают, но на кластеры в целом это не влияет, они также разбросаны и не отделяются.

Для “Программирование и основы алгоритмизации” – деление идет на два кластера, не по преподавателям и группам – в оба кластера попадают и студенты из одних и тех же команд, также с иностранными студентами.

Кластеризация только по вектору посещений

На рисунке 12 показана визуализация результатов кластеризации для дисциплины “Алгебра” с использованием методов KMeans, DBSCAN и Agglomerative clustering (иерархическая кластеризация).

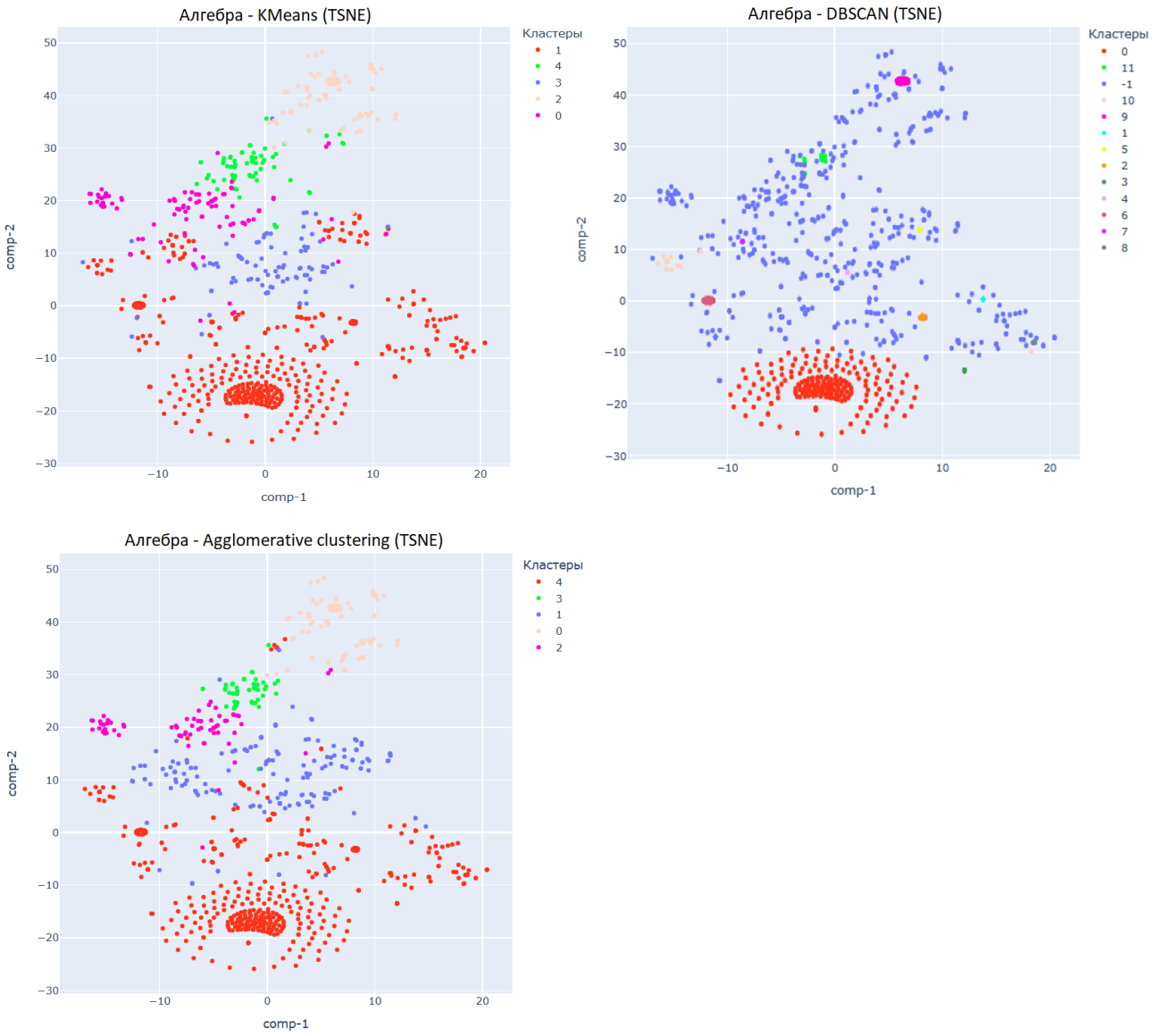


Рис. 12. Визуализация кластеров методов KMeans, DBSCAN и Agglomerative clustering для вектора посещений по дисциплине “Алгебра”.

Для KMeans по дисциплине “Алгебра” было выбрано количество кластеров равным 5 – “локоть” показывал большую смену уклона на количестве 3 и 5, было решено выбрать 5 кластеров для более детального разбиения, несмотря на то что 3 кластера показывали лучшую среднюю метрику силуэта. Визуализация точек кластеров показана на рисунке 12.

Анализ наполнения кластеров:

- Кластер 0 (103 точки), 2 (114 точек) и 3(101 точка) – среднее между всеми, промежуточная посещаемость;

- Кластер 1 преимущественно наполнен студентами с хорошей и идеальной посещаемостью (это очень большая группа студентов – 558 точек);
- Кластер 4 преимущественно наполнен студентами с плохой или отсутствующей (не проставленной) посещаемостью (74 точки).

KMeans показал себя довольно плохо на имеющихся данных – кластеры слишком сильно разрозненные и довольно сильно пересекаются между собой.

Для DBSCAN по дисциплине “Алгебра” были выбраны следующие гиперпараметры – eps=0.001, min_samples=10, metric=’cosine’. Параметры подбирались экспериментально. Визуализация точек кластеров показана на рисунке 12. Кластер “-1” является “кластером” выбросов (508 точек).

Анализ наполнения кластеров:

- Кластер 0 – идеальная посещаемость студентов (271 точек);
- Кластер 1 – один пропуск – тема “Симметричные многочлены” (10 точек);
- Кластер 2 – один пропуск – тема “Собственные вектора” (14 точек);
- Кластер 3 – один пропуск – тема “Комплексные числа” (10 точек);
- Кластер 4 – один пропуск – тема “Теорема Безу” (20 точек);
- Кластер 5 – два пропуска подряд – темы “Линейные пространства” и “СЛУ и СЛОУ…” (10 точек);
- Кластер 6 – два пропуска подряд – темы “Формулы Муавра” и “Многочлен и его корни” (10 точек);
- Кластер 7 – два пропуска подряд – темы “Процесс ортогонализации” и “Квадратичные формы” (10 точек);
- Кластер 8 – не указано посещение для второй контрольной работы – вероятно отсутствие студента (10 точек);
- Кластер 9 – два пропуска подряд – темы “Матрицы…” и “Определители…” (10 точек);

- Кластер 10 – две не указанных посещаемости – темы “Квадратичные формы…” и “Распадающиеся квадратичные формы” (18 точек);
- Кластер 11 – первые три посещения не указаны (14 точек);
- Кластер 12 – первые четыре посещения не указаны (24 точек);
- Кластер 13 – первое, второе и четвертое посещение не указаны (11 точек);

DBSCAN показал себя довольно плохо на имеющихся данных – как оказалось, большее число точек находится в областях низкой плотности, из-за чего сформировать хорошие кластеры не вышло. Большинство имеющихся кластеров сформировалось из-за того, что векторы внутри кластера идентичны и соответственно формируют область высокой плотности.

Для иерархической кластеризации по дисциплине “Алгебра” было выбрано 5 кластеров исходя из графика дендрограммы. Из-за таких же умозаключений, как и в случае с KMeans, было решено взять 5 кластеров вместо 3, на которые намекает график. Визуализация точек кластеров показана на рисунке 12.

Анализ наполнения кластеров:

Можно заметить, что метод AgglomerativeClustering показал очень похожий на KMeans результат, что отражает те же самые проблемы:

- Кластер 0 (115 точек), 2 (75 точек) и 1 (188 точек) – среднее между всеми, промежуточная посещаемость;
- Кластер 4 преимущественно наполнен студентами с хорошей и идеальной посещаемостью (это очень большая группа студентов – 518 точек);
- Кластер 3 преимущественно наполнен студентами с плохой или отсутствующей (не проставленной) посещаемостью (54 точки).

AgglomerativeClustering показал себя довольно плохо на имеющихся данных – по аналогии с KMeans, кластеры слишком сильно разрозненные и довольно сильно пересекаются между собой.

На рисунке 13 показана визуализация результатов кластеризации для дисциплины “Программирование и основы алгоритмизации” с использованием методов KMeans, DBSCAN и Agglomerative clustering (иерархическая кластеризация).

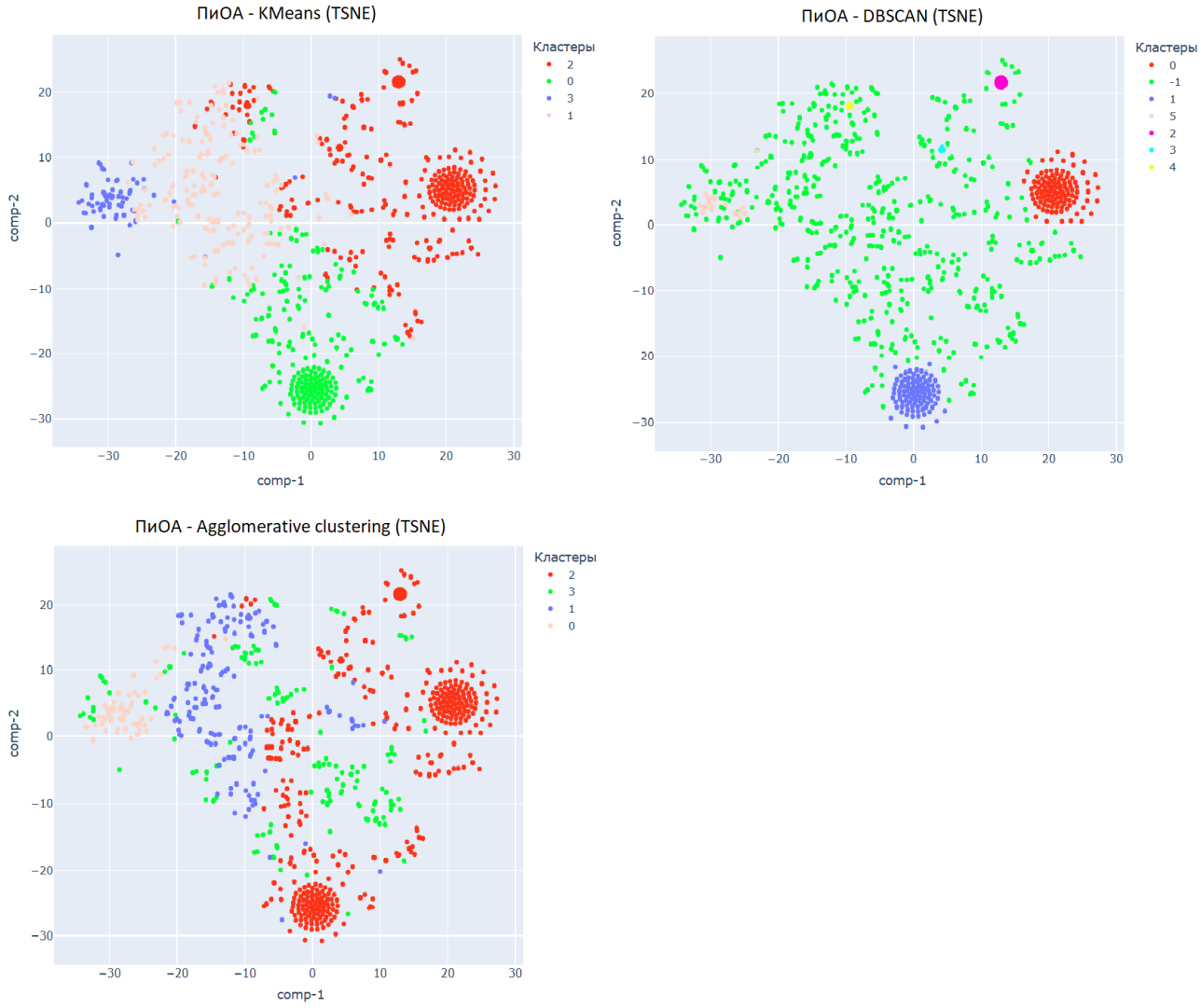


Рис. 13. Визуализация кластеров методов KMeans, DBSCAN и Agglomerative clustering для вектора посещений по дисциплине “Программирование и основы алгоритмизации”.

При кластеризации по дисциплине “Программирование и основы алгоритмизации” для KMeans было выбрано количество кластеров равным 4 – “локоть” был мало заметен и было решено выбрать количество кластеров исходя из графика TSNE и на среднего значения силуэта. Визуализация точек

кластеров показана на рисунке 13.

Анализ наполнения кластеров:

- Кластер 0 (277 точек) и 2 (405 точек) – хорошая или идеальная посещаемость, с разницей в том, что в кластере 0 находятся те, кто посещал экзаменационное мероприятие, а в кластере 2 – те, кто не посещал;
- Кластер 1 преимущественно наполнен студентами с средней посещаемостью (50%-60% посещенных занятий, 203 точки);
- Кластер 3 преимущественно наполнен студентами с плохой или отсутствующей (не проставленной) посещаемостью (75 точки).

Как и в случае с дисциплиной “Алгебра”, KMeans не смог хорошо отделить кластеры достаточно четко – хоть мы и можем примерно говорить о том, какие студенты могут попасть в кластер, имеется слишком много пограничных случаев.

Для дисциплины “Программирование и основы алгоритмизации” у метода кластеризации DBSCAN были выбраны следующие гиперпараметры – eps=0.001, min_samples=10, metric=’manhattan’. Параметры подбирались экспериментально. Визуализация точек кластеров показана на рисунке 13. Кластер “-1” является “кластером” выбросов (670 точек).

Анализ наполнения кластеров:

- Кластер 0 (118 точек) и 1 (99 точек) – идеальная посещаемость, с разницей в том, что в кластере 0 находятся те, кто не посещал экзаменационное мероприятие, а в кластере 1 – те, кто посещал;
- Кластер 2 – посещали экзаменационное мероприятие, но не имеют отметки о посещении предыдущего занятий (21 точка);
- Кластер 3 – последние 3 мероприятия – не посещали (10 точек);
- Кластер 4 – не посещали 3 мероприятия – 2 в середине семестра и экзаменационное мероприятия (11 точек);

- Кластер 5 – либо все мероприятия не посещали, либо большое количество (31 точек);

Аналогичная с дисциплиной “Алгебра” ситуация, только в этот раз выделились два кластера с идеальной посещаемостью.

Для иерархической кластеризации по дисциплине “Программирование и основы алгоритмизации” было выбрано 4 кластера основываясь на графике дендрограммы – была попытка сформировать более равномерные кластеры. Визуализация точек кластеров показана на рисунке 13.

Анализ наполнения кластеров:

- Кластер 0 – плохое посещение (73 точки);
- Кластер 1 – в основном плохое и среднее посещение (188 точек);
- Кластер 2 – идеальное и хорошее посещение (529 точек);
- Кластер 3 – сильно разбросанный кластер, в основном средняя посещаемость (170 точек);

В отличии от примера с “Алгеброй”, результат на том же количестве кластеров, как и у KMeans, дал радикально другой результат.

На рисунке 14 показана визуализация результатов кластеризации для дисциплины “Математический анализ” с использованием методов KMeans, DBSCAN и Agglomerative clustering (иерархическая кластеризация).

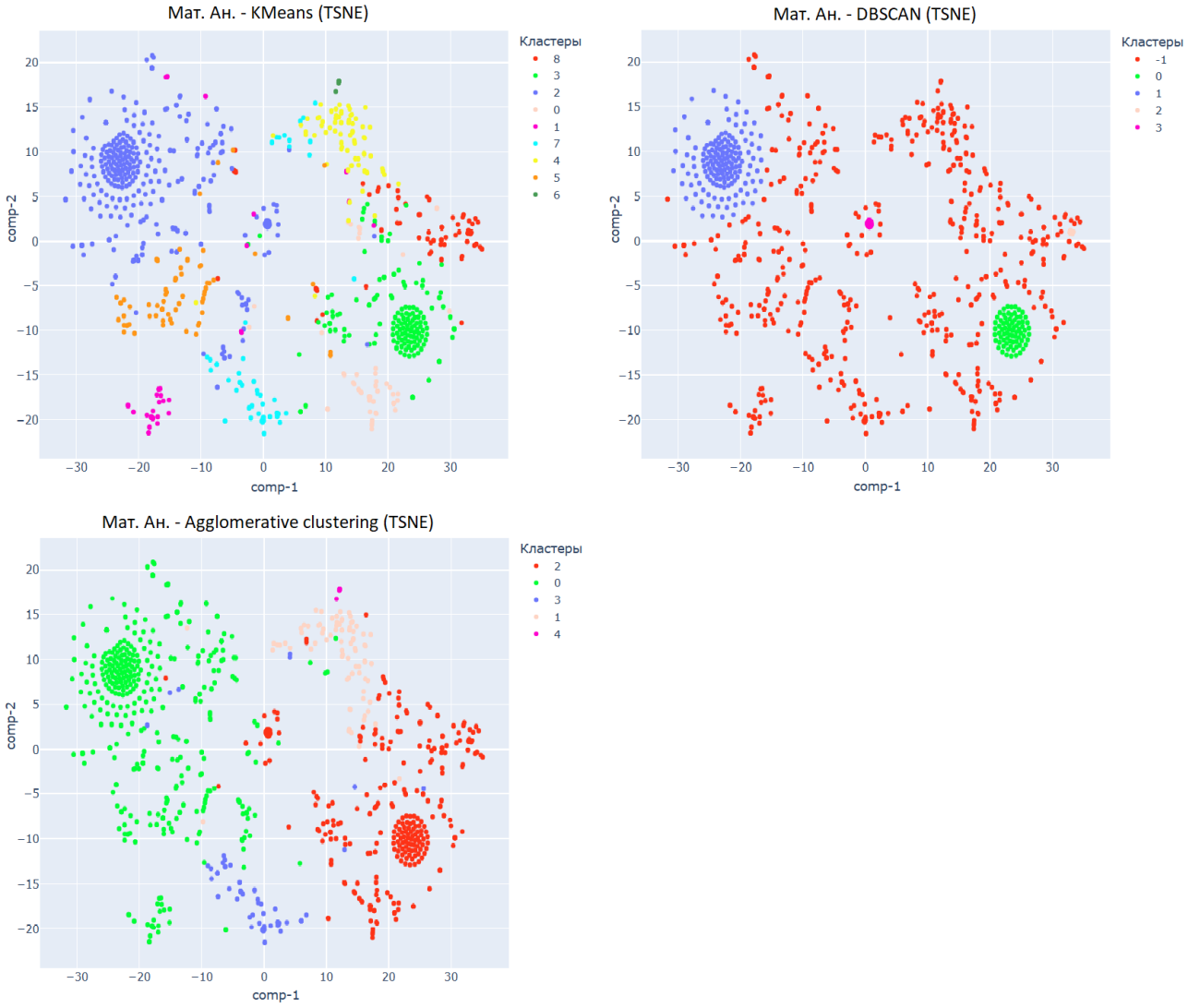


Рис. 14. Визуализация кластеров методов KMeans, DBSCAN и Agglomerative clustering для вектора посещений по дисциплине “Математический анализ”.

Для KMeans по дисциплине “Математический анализ” было выбрано количество кластеров равным 9 – “локоть” был довольно прямым также начиная от 3 кластеров, из-за чего основное решение падало на среднее силуэта – на 9 кластерах значение превысило довольно низкое значение для 3 кластера. Визуализация точек кластеров показана на рисунке 14.

Анализ наполнения кластеров:

- Кластер 0 – посещение выше среднего, пропуск одного мероприятия в середине семестра (55 точки);
- Кластер 1 – в основном плохое и среднее посещение (43 точек);

- Кластер 2 – идеальное и хорошее посещение, пропуск экзамена (369 точек);
- Кластер 3 – идеальное и хорошее посещение, посещение экзамена (181 точки);
- Кластер 4 – сильно разбросанный кластер, в основном средняя посещаемость
  (75 точек);
- Кластер 5 – средняя посещаемость, пропуск экзамена (80 точек);
- Кластер 6 – в кластере собрались студенты, у которых либо ничего не отмечено, либо не отмечено больше половины (6 точек);
- Кластер 7 – сильно разбросанный кластер, средняя посещаемость, пропускали второе и третье мероприятие (67 точек);
- Кластер 8 – хорошая посещаемость, посещение экзамена, все посещают первое мероприятие (76 точек);

Из всех рассмотренных предметов, Математический анализ лучше всех показал возможность к кластеризации KMeans, хотя сложность в кластеризации все равно имеется.

Для DBSCAN по дисциплине “Математический анализ” были выбраны следующие гиперпараметры – eps=0.5, min_samples=10, metric=’manhattan’. Параметры подбирались экспериментально – в худшем случае кластеры сильно перемешивались (при высоком eps). Визуализация точек кластеров показана на рисунке 14. Кластер “-1” является “кластером” выбросов (672 точки).

Анализ наполнения кластеров:

- Кластер 0 - идеальное посещение + посещение экзамена (76 точек);
- Кластер 1 - идеальное посещение + нет отметки о посещении экзамена (76 точек); (179 точек);
- Кластер 2 – идеальное посещение, за исключением отсутствия какой-либо отметки по теме “Формула Тейлора” (11 точек);
- Кластер 3 - идеальное посещение + не явился на экзамен (14 точек);

Явно разделились идеальные посещения по посещению экзамена – Н, П и пусто (null).

Для иерархической кластеризации по дисциплине “Математический анализ” было выбрано 5 кластеров основываясь на графике дендрограммы – была попытка сформировать более равномерные кластеры. Визуализация точек кластеров показана на рисунке 14.

Анализ наполнения кластеров:

- Кластер 0 – размытый кластер посещающих, включает тех, кто не имеет посещение экзамена (455 точек);
- Кластер 1 – (88 точек);
- Кластер 2 – объединяет большую часть хорошо и отлично посещающих, которые имеют посещение экзамена (329 точек);
- Кластер 3 – кластер с хорошим посещением второй половины семестра (74 точки);
- Кластер 4 – в кластере собрались студенты, у которых либо ничего не отмечено, либо не отмечено больше половины (6 точек);

Кластеризация отдельно по контрольным точкам дисциплины

Для выбранных дисциплин также была проведена кластеризация по их контрольным точкам:

- “Алгебра” – Контрольные работы №1-3 и коллоквиум;
- “Программирование и основы алгоритмизации” – Контрольные работы №1-3;
- “Математический анализ” – Контрольные работы №1-2, практическое задание и письменный ответ на зачете с оценкой.

Для кластеризации по контрольным точкам использовался метод KMeans, выбор количества кластера, как и в случае с векторами посещений, осуществлялся на основе комбинации результатов метода локтя и силуэта. Визуализация кластеризации для каждой дисциплины показана на рисунке 15.

Для формирования графиков слева на рисунке используется метод TSNE,

графики справа используют метод PCA - используются две главные компоненты, отвечающие примерно за описание 84% информации (дисперсии).

Формирование “кругов” на графике TSNE обусловлено тем, что в данных точках находятся идентичные векторы для контрольных точек (а именно – все нули, плохие студенты). Из-за наличия большого количества таких векторов данных во время формирования двумерной точки, идентичные значения формируются в подобной форме.

Формирование “углов” на графике PCA можно объяснить наличием большого количества “крайних” значений контрольных точек – максимальных и минимальных баллов.

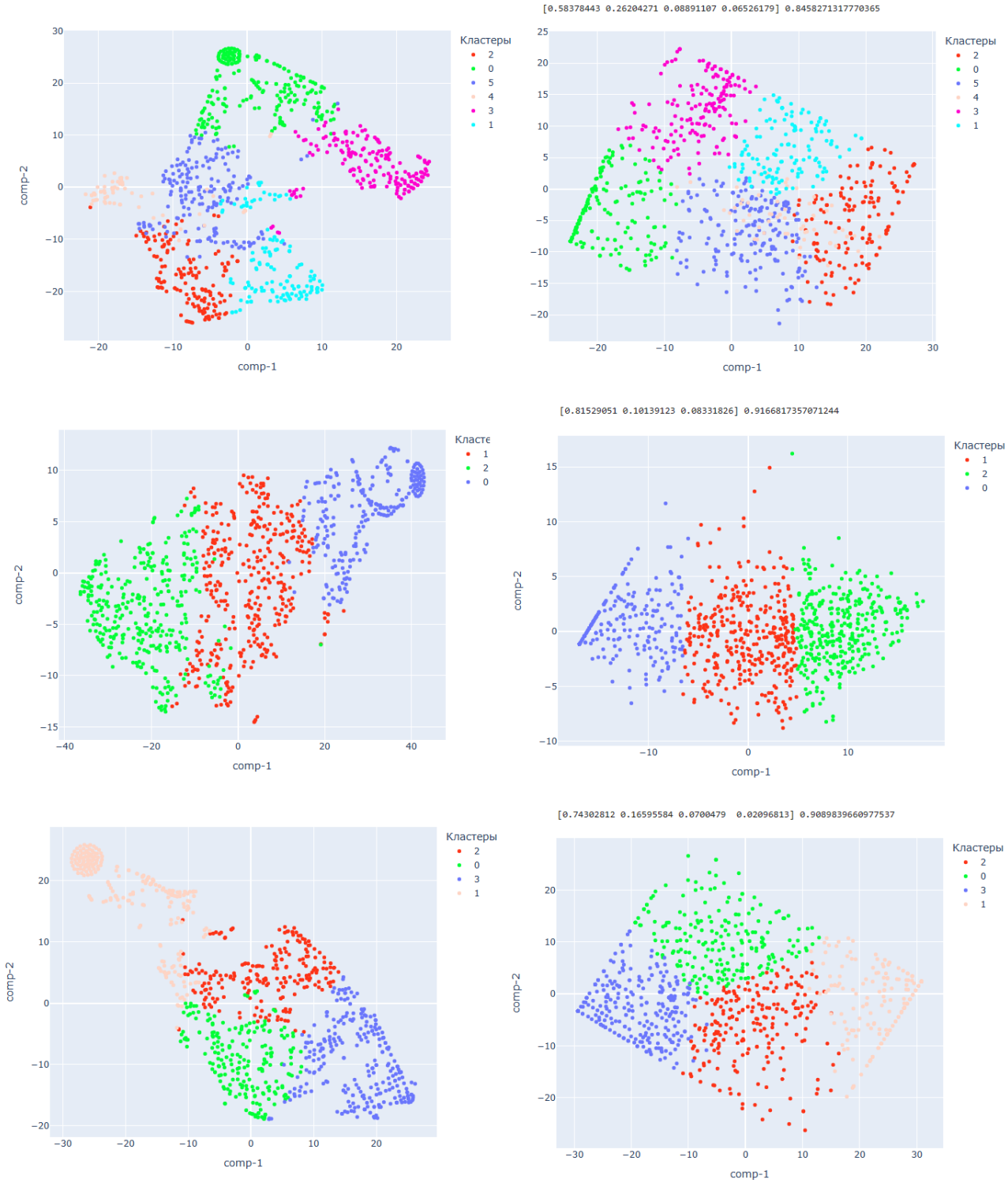


Рис. 15. Визуализация кластеризации методом TSNE и PCA для (сверху-вниз): “Алгебра”, “Программирование и основы алгоритмизации”, “Математический анализ”.

Анализ кластеров для дисциплины “Алгебра” (относительно TSNE):

- Кластер 0 – в центре нули по всем контрольным точкам (219 точек);

- Кластер 1 – низ центр, часто хорошие по контрольным, средний коллоквиум (149 точек);
- Кластер 2 – низ лево, разные по контрольным, хороший коллоквиум (152 точки);
- Кластер 3 – на краю справа хорошие по контрольным работам, низкий коллоквиум (180 точек);
- Кластер 4 – крайне лево, плохая первая контрольная, хорошая вторая, средняя третья, разный коллоквиум (выше среднего) (63 точки);
- Кластер 5 – очень разные по контрольным, средний коллоквиум (187 точек).

Анализ кластеров для дисциплины “Программирование и основы алгоритмизации” (относительно TSNE):

- Кластер 0 – верх право, разные по контрольным, но все в основном ниже 10 баллов, включает студентов со всеми нулями (251 точка);
- Кластер 1 – центр, средние по контрольным, большой разброс (369 точек);
- Кластер 2 – низ лево, разные по контрольным, но все в основном выше 10 баллов за каждую, в среднем 15 баллов (340 точек);

Анализ кластеров для дисциплины “Математический анализ” (относительно TSNE):

- Кластер 0 – середина низ, контрольная работа №1 ниже среднего, контрольная работа №2 выше среднего, очень много с хорошей оценкой за практическое задание (217 точек);
- Кластер 1 – верх лево, в основном ниже среднего, включает студентов со нулями за все выбранные контрольные точки (239 точек);
- Кластер 2 – середина верх, очень разные оценки за контрольную работу №1, средние по контрольной работе №2, высокие по практическому заданию, есть те, кто сдавал письменный ответ (217 точек);

- Кластер 3 – низ право, очень хорошие контрольные и практическое задание выше среднего (279 точки);

У большинства нули по письменному ответу на экзамене, но есть исключения.

ГЛАВА 4. ПРОГРАММНЫЕ РЕШЕНИЯ

Приложения разрабатывалось с помощью streamlit – мощной библиотеки для разработки веб-инструментов, заточенных под анализ и визуализацию данных, на языке Python. Одна из важных особенностей streamlit это совместимость со многими библиотеками, такими как pandas и plotly, что упрощает разработку приложения.

Веб-приложения, разрабатываемые с помощью данной библиотеки, имеют клиент-серверную архитектуру (см. рис.16). Непосредственно streamlit-приложение запускается на сервере Tornado – асинхронном веб-сервере Python. Код на языке Python запускается на данном сервере каждый раз, когда пользователь взаимодействует с интерфейсом веб-приложения. Взаимодействие между сервером и клиентом производится с помощью протокола связи WebSocket.

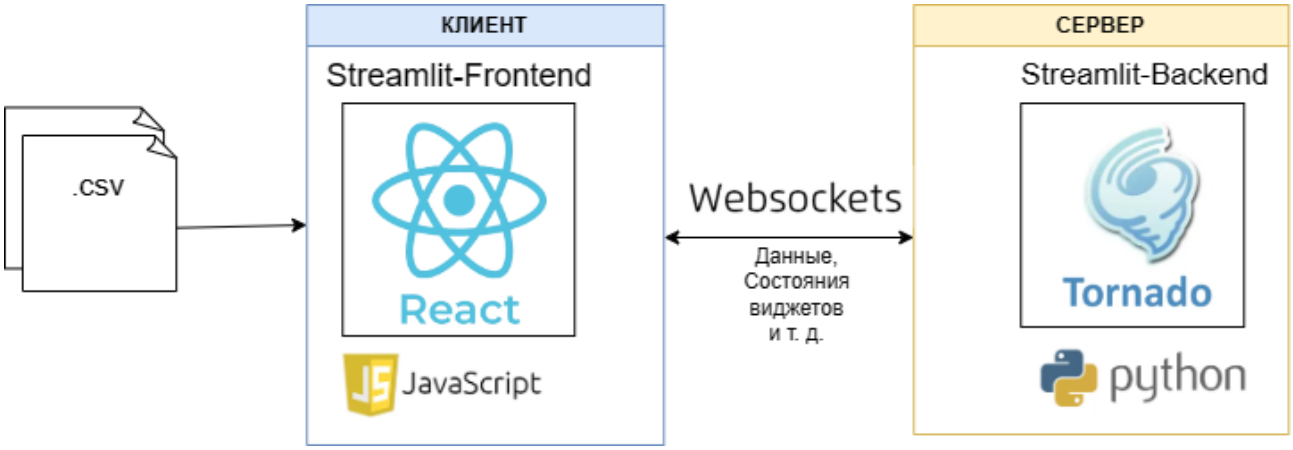


Рис. 16. Архитектура веб-приложения, разрабатываемого с помощью библиотеки streamlit.

При этом разработчик напрямую не взаимодействует с описанными элементами – все элементы приложения описываются в файлах на языке python. Каждая страница приложения – отдельный файл с расширением .py, определенный в подпапке “pages”. Содержание файла – скрипт, содержащий функции, которые отвечают за объявление и расположение на странице различных виджетов, предоставляемых streamlit.

Структура проекта выглядит следующим образом:

- Папка modules – содержит python модули с разработанными функциями. В отдельные модули были выделены функции, чей функционал используется на нескольких страницах приложения;
- Папка assets – содержит изображения, иконки и файл стиля css.
- Папка pages – содержит .py файлы страниц приложения. Каждый файл содержит код для отображения виджетов на странице и код разработанных функций для формирования графиков.
- Файл “Начальная_страница.py” – стартовый файл .py приложения. Представляет собой первую страницу приложения. Содержание аналогично файлам, расположенным в папке pages.

В таблице 1 описаны разработанные функции.

Таблица 1

Разработанные функции

Модуль	Функция	Назначение
df_transform	get_wide_df(df, key)	Переводит dataframe из “длинного” формата в “широкий”. Возвращает преобразованную копию dataframe.
	get_tuple_col_df(df, key)	Переводит столбец “длинного” формата в столбец объектов tuple. Возвращает преобразованную копию dataframe.
prepare_page	prepare_page()	Первоначальная настройка для страниц – формирование sidebar, логотипа, добавление таблицы стиля.
prepare_data	prepare_df(df, subject)	Производит обработку и преобразование загружаемых данных для дальнейшей работы. Возвращает преобразованную копию dataframe.

Продолжение таблицы 1

Файл страницы	Функция	Назначение
1_Визуадизация _распределений.py	get_multi_dist_plot(df, frame_columns, filter_col, selected_values_of_filter = None, n = 1, draw_info = False, with_barchart = False, height = 350)	Формирует график с подграфиками распределений выбранных столбцов dataframe в выбранном разрезе. Возвращает объект графика.
	show_info(fig, hist_data, row, col)	Добавляет квартили и среднее для подграфика переданного графика.
2_Посещения _студентов.py	get_multi_attendance_plot(df, filter_col, frame_columns, value_vars_cols, filter_col_values = None, num_cols = 5, height = 300)	Формирует график с подграфиками посещений студентов по дисциплине в выбранном разрезе. Возвращает объект графика.
3_Баллы_и _оценки.py	get_multi_mark_plot(df, filter_col, filter_col_values = None, num_cols = 5, name = '', height = 300)	Формирует график с подграфиками баллов и оценок студентов по дисциплине в выбранном разрезе. Возвращает объект графика.
4_Проблемные _студенты.py	get_piechart_subject_data(main_subj, selected_subjs)	Формирует словарь – данные для круговой диаграммы, отражающую успеваемость студентов по выбранным дисциплинам. Возвращает словарь и dataframe.
	get_piechart_att_data(filtered_df)	Формирует словарь – данные по выбранному фильтру, отражающие посещаемость отфильтрованных студентов. Возвращает словарь.
5_Кластеризация.py	get_cluster_2d_plot(cluster_data, labels, title=None)	Формирует график разброса для визуализации кластеризации. Возвращает объект графика.
	get_multi_bar_chart(data_with_labels)	Формирует график с подграфиками оценок студентов в выбранном разрезе. Возвращает объект графика.
	get_multi_pie_chart(data_with_labels, selected_filter_column)	Формирует график с подграфиками наполнения кластеров по выбранному разрезу. Возвращает объект графика.

Так как при взаимодействии с элементами интерфейса приложения, код в

файле соответствующей страницы выполняется полностью заново, для сохранения состояния приложения между перезапусками используются api состояния сессии. Для выполнения действий после изменения состояния виджета (реагирование на определенные события) используются callback функции, выполняемые первыми, перед остальными функциями приложения.

В приложении для интерпретации данных состояние сессии используется для хранения загруженных датафреймов, чтобы данные не исчезали во время переходов между страницами, а также для сохранения состояния кнопки выбора предмета, для тех же целей.

Одним из ограничений streamlit является отсутствие детальной настройки компонента st.sidebar. При использовании данного компонента при разработке многостраничных приложений верхняя часть компонента автоматически передается в пользование компоненту навигатора по страницам. Таким образом, с точки зрения разработки приложения с использованием предоставленных компонентов streamlit не имеется возможности напрямую добавлять виджеты и другие элементы в начало st.sidebar, например иконки или изображения с использованием компонента st.image. Для обхода данного ограничения используется прямое изменения css стиля элемента с атрибутом data-testid = stSideBarNav с использованием соответствующего селектора. Для этого используется свойство css background-image. Значением для данного свойства является url путь к статическому ресурсу на сервере приложения. Исходя их этого, возникает следующее ограничение streamlit – не имеется возможности прямой загрузки статических ресурсов на сервер приложения. Для обхода данного ограничения изображение, сохраненное в ресурсах проекта, вставляется в css код тега style как бинарные данные, преобразованные в Base64.

Теги html могут быть напрямую добавлены в любую часть кода страницы с помощью метода st.markdown. Таким же образом в код страницы добавляется файл со стилями css для других компонентов приложения. Для идентификации компонента для последующей модификации его css кода, использовались dev-

tools браузера (один из предлагаемых разработчиками streamlit способов настройки стилей имеющихся компонентов).

Так как код каждой страницы запускается при любом взаимодействии с виджетами соответствующей страницы, в streamlit предусмотрен специальный декораторы @st.cache_data. Данный декоратор кеширует результат выполнения функции и впоследствии возвращает сохраненный результат выполнения функции при использовании тех же входных параметров. Это удобно в тех случаях, когда выполнения кода определенной функции занимает довольно длительное время, а результат выполнения этой функции влияет на визуальную составляющую приложения и существует необходимость сохранять вид приложения даже в случае изменения состояний не связанных с данной функцией элементов интерфейса. В разрабатываемом приложении декоратор @st.cache_data использовался для сохранения сгенерированных графиков, а также результатов кластеризации по выбранным данным.

Входные данные для работы приложения – файл формата .csv, строки которого содержат информацию о студенте + данные об одном конкретном мероприятии (по аналогии с первоначальным файлом данных из МОДЕУС). Данные из файла были предобработаны следующим образом:

- Проверка и корректировка размера вектора оценок и посещений (были рассмотрены случаи с удалением дубликатов встреч и отсутствующем векторе посещений).
- Корректировка порядка вектора оценок (были рассмотрены случаи, когда в пределах встречи имелись несколько предметов контроля по баллам – отдельно “Оценка за работу на практике” и “Контрольная работа”, “Практическое задание”, “Письменный ответ”.
- При наличии – заполнение null значений для векторов баллов и посещений (заменяются нулями для обоих векторов).

Для визуализации имеющихся данных на графиках, загруженные таблицы приводятся к следующему виду:

- Столбцы с данными о студенте:

Столбцы объектного типа numpy (np.object_):

- Название РМУП – название курса.
- Идентификатор студента.
- ФИО студента.
- Команда – группа студента.
- Направление подготовки студента.
- Итоговая оценка – итоговая оценка по дисциплине.

Столбцы вещественного типа numpy (np.float32):

- Посещаемость студента (формируется как отношение посещений встреч студентом к общему количеству встреч)
- Итог ТУ – итоговое количество баллов по курсу.

- Дополнительные столбцы с баллами за каждое контрольное мероприятие отдельно.
- Столбцы-векторы с баллами за каждое мероприятие и отметками о каждом посещении.

Таким образом каждая запись в таблице описывает данные одного студента в рамках одного предмета. Для приведения таблицы в описанный вид, была написана специальная функция для преобразования.

Преобразование загруженной таблицы в итоговый pandas dataframe осуществляется следующим образом:

1) Сохраняется две копии оригинальной таблицы, в первой копии отбираются только столбцы с данными о студенте и удаляются дубликаты (остается одна строка на одного студент).
2) Вторая копия разбивается на две таблицы – для векторов посещений и для векторов оценок. Обе таблицы преобразуются из вида “строка – информация об одной встрече студента” к виду “строка – информация о всех встречах одного студента” с помощью метода pandas groupby по идентификатору студента с применяемой функцией tuple – таким образом формируется столбец, в котором находится tuple-вектор с информацией обо всех посещениях/оценках для каждого студента.

3) Значения из таблиц посещений/оценок сливаются с помощью метода merge в одну таблицу и, с помощью операции спискового включения по строкам данной таблицы, “разворачиваются” в столбцы таблицы.
4) Полученная таблица сливается (merge) с таблицей с информацией о студентах. Преобразование окончено.

Как и в случае с графиками, результат функции преобразования таблиц сохраняется в кэше для дальнейшего использования другими частями приложения.

Непосредственно для визуализации данных, в приложении используется библиотека plotly, а также в некоторых случаях, дополнительная библиотека streamlit-eChart. Обе библиотеки представляют собой инструменты для построения интерактивных графиков и могут быть без проблем отображены в приложении streamlit. Использование двух библиотек обосновывается тем, что plotly (используется в приложении чаще всего) не может возвращать данные о событиях (таких как клик мышки по графику и т. д.) в полном объема для графиков, не использующих стандартные оси x и y (так как имеет четкую структуру ответа) – например piechart. Графики же streamlit-eChart, специально разработанной “обертки” для рендеринга графиков Apache eChart для совместной работы с streamlit, могут возвращать данные о событии в полном объеме.

Для визуализации распределений значений столбцов передаваемого dataframe (таблицы студентов) используются функция генерации plotly ff.create_distplot, но не напрямую. Проблема заключалась в том, что ff, figure factory, генерирует целую фигуру-график, которую нельзя установить в клетку подграфовиков, формируемых методом make_subplots. Для того, чтобы это было возможно, данные трейсов (структуры, хранящие информацию об одном конкретном графике) были взяты напрямую через метод select_traces. В последствии данные трейсы были один за другим перенесены в полный граф с подграфами. Дополнительно, перед отрисовкой kde, в соответствующие подграфы могут быть добавлены графики гистограмм с помощью добавления

трейсов go.Histogram() (на подграф их может быть несколько и отображаются наложениям друг на друга настройкой layout - “barmode=overlay” с выбранным параметром прозрачности). На рисунке 17 представлен алгоритм для построения графика с под-графиками распределения выбранных показателей образовательной работы студентов.

Описание алгоритма для построения графика:

- Формирование фигуры с количеством пустых подграфиков, определенным исходя из выбранных признаков для визуализации, значений фильтра и выбранного количества трейсов на один подграфик.
- Прохождения циклов по строкам и столбцам фигуры для формирования каждого подграфика. В зависимости от переданных параметров выбирается тип графика (функция распределения и гистограмма или только функция распределения) и добавляются ли на подграфик перцентили.
- Проведение финальной настройки фигуры – настройка размеров и отступов.

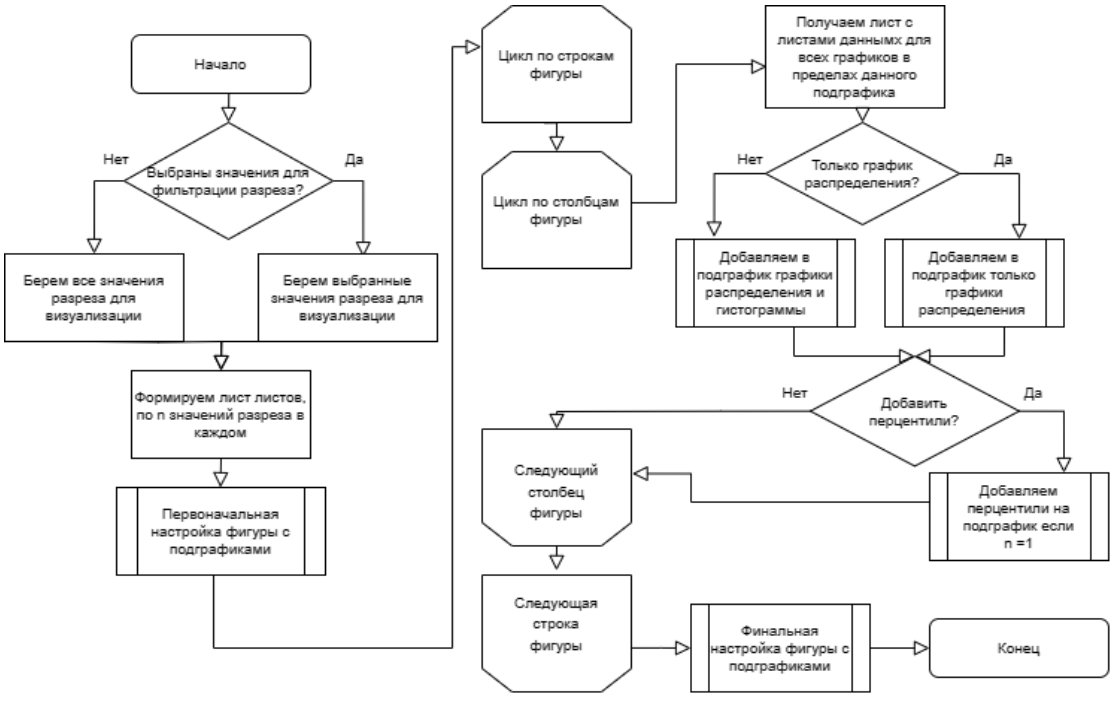


Рис. 17. Алгоритм построения графика для страницы “Визуализация распределений”.

Код функции для формирования графиков распределения образовательных показателей студентов в выбранном разрезе предоставлен в Приложении 1.

Для визуализации посещаемости студентов в виде гистограмм по каждому посещению отдельно используется метод pandas melt для dataframe’ов, позволяющий перевести dataframe в более удобный вид для передачи данных в столбчатый график. Столбцы-векторы переводятся в формат столбцов переменная-значение – каждая строка в таблице становится записью об одном посещении с порядковым номером.

Таким образом, после этого данные имеют вид, подходящий для передачи на вход методу визуализации столбчатого графика. Количество неявок и незаполненных пропусков добавлялись вместе в один график с настройкой layout - “barmode=stack”, позволяющей столбцам находится друг над другом, а не на заднем плане или сбоку. На рисунке 18 представлен алгоритм для построения рисунка-графика с подграфиками общего посещения предмета студентами по значениям выбранного разреза.

Описание алгоритма для построения графика:

- Формирование фигуры с количеством пустых подграфиков, определенное исходя из значений фильтра и выбора о формировании фигуры по командам преподавателей. Если фигура формируется по командам преподавателей, то каждая строка будет отражать команды одного преподавателя
- Прохождения циклов по строкам и столбцам фигуры для формирования каждого подграфика – столбчатого графика. В зависимости от того, выбрано ли формирование графика по командам преподавателей, каждый график будет либо отражать посещение всех студентов конкретного преподавателя, либо одной конкретной команды.
- Проведение финальной настройки фигуры – настройка размеров и отступов.

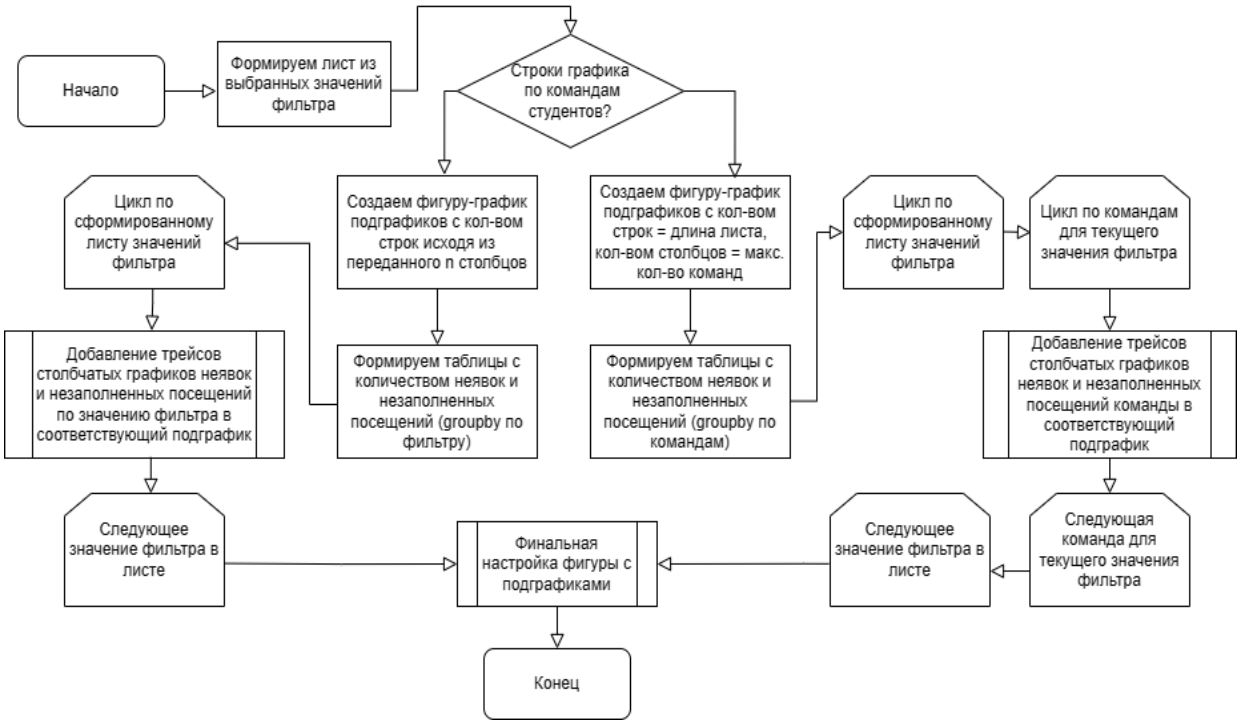


Рис. 18. Алгоритм построения графика для страницы “ Посещения студентов”.

Код функции для формирования графиков посещаемости студентов в выбранном разрезе предоставлен в Приложении 2.

Для визуализации распределения финальных баллов студентов в совокупности с их итоговой оценкой по предмету для каждой оценки (“неявка”, “неуд.”, “удовл.”, ”хор.” и “отл.”) формируется свой отдельный трейс go.Histogram() с установленным названием и группой легенды для соответствующей оценки, начиная от “неявка” и заканчивая “отл.”. Таким образом, установив настройку layout “barmode=stack”, бины гистограмм для каждой оценки будут расположены друг над другом, в указанном порядке снизу вверх. Данное действие проводится для каждого подграфика. На рисунке 19 представлен алгоритм для построения рисунка-грифика с подграфиками распределения финальных баллов и оценок студентов по значениям выбранного разреза.

Описание алгоритма для построения графика:

- Формирование фигуры с количеством пустых подграфиков, определенное исходя из значений фильтра и выбора о формировании фигуры по командам преподавателей. Если фигура формируется по

командам преподавателей, то каждая строка будет отражать команды одного преподавателя

- Прохождения циклов по строкам и столбцам фигуры для формирования каждого подграфика – общих столбчатых графиков по оценке. В зависимости от того, выбрано ли формирование графика по командам преподавателей, каждый график будет либо отражать посещение всех студентов конкретного преподавателя, либо одной конкретной команды.
- Проведение финальной настройки фигуры – настройка размеров и отступов.

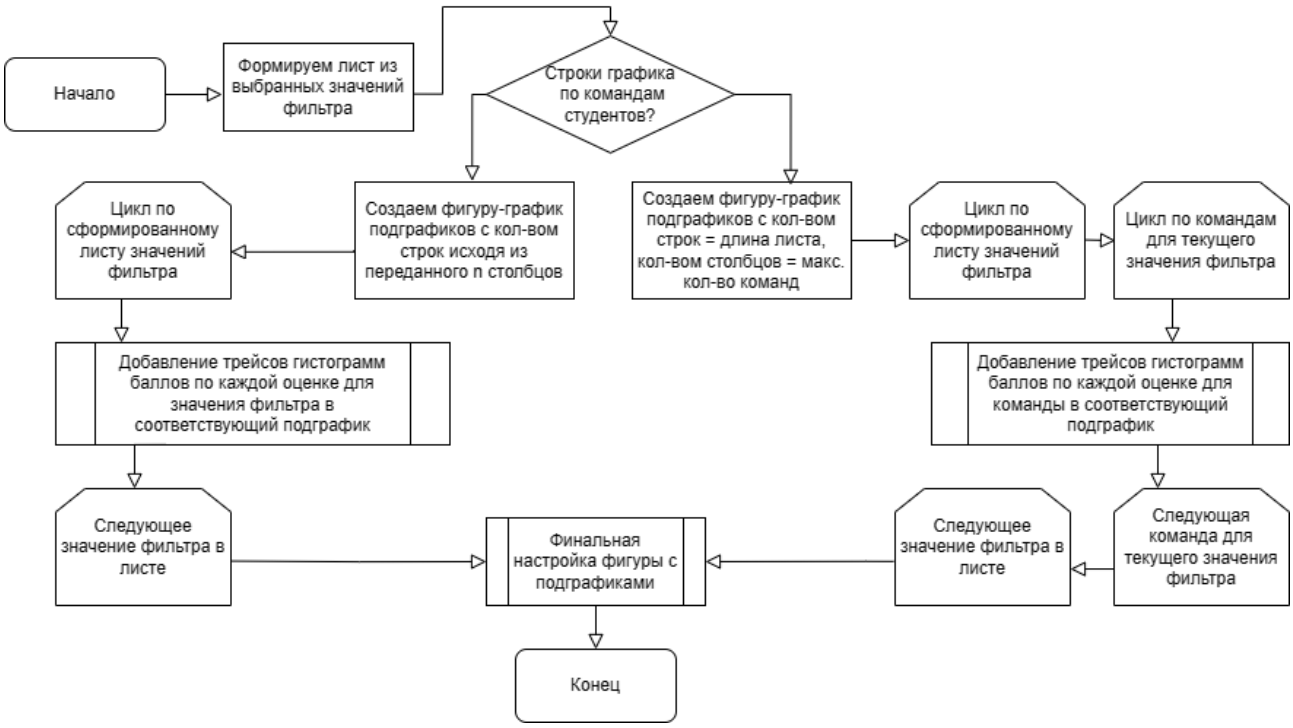


Рис. 19. Алгоритм построения графика для страницы “ Баллы и оценки”.

Код функции для формирования графиков распределения баллов и оценок студентов в выбранном разрезе предоставлен в Приложении 3.

Для отображения информации о проблемных студентах использовались специально разработанные для streamlit доп. библиотеки – streamlit-echart (интерактивные графики) и streamlit-aggrid (интерактивные таблицы). Данные библиотеки – пользовательские обертки-компоненты для рендеринга графиков Apache Echarts и таблиц JS AgGrid.

ГЛАВА 5. ОПИСАНИЕ ИНСТРУМЕНТА

Приложение представляет собой онлайн инструмент в виде веб-дашборда со следующими страницами:

- Страницей приветствия и загрузки данных.
- Страницей, содержащей графики распределений числовых признаков – образовательных результатов студентов.
- Страницей, содержащей графики посещений студентов (в виде гистограмм).
- Страницей, содержащей комбинированные графики финальных баллов и оценок студентов (и в виде кросс-таблицы).
- Страницей, содержащей информацию о студентах, не сдавших дисциплину – получившие оценку “Неудовлетворительно” в результате работы на финальном контрольном мероприятии, либо его не посетившие (не претендующие на хорошую оценку автоматом).
- Страницей, содержащей инструмент для проведения кластерного анализа по имеющимся данным студентов.

На странице приветствия (“Начальная страница”) описывается основная информация о приложении (см. рис.20). Здесь осуществляется загрузка файлов с информацией о студентах разных дисциплин. Можно загружать как один, так и несколько файлов одновременно. Названия загруженных файлов будут отображаться на данной странице. При необходимости можно удалить загруженные файлы и начать работу с приложением заново, нажав на кнопку “Удалить файлы”.

На рисунке 20 также показана боковая панель, с помощью которой производится навигация между страницами приложения и выбор основного предмета, относительно которого происходит построение графиков и таблиц в пределах каждой страницы.

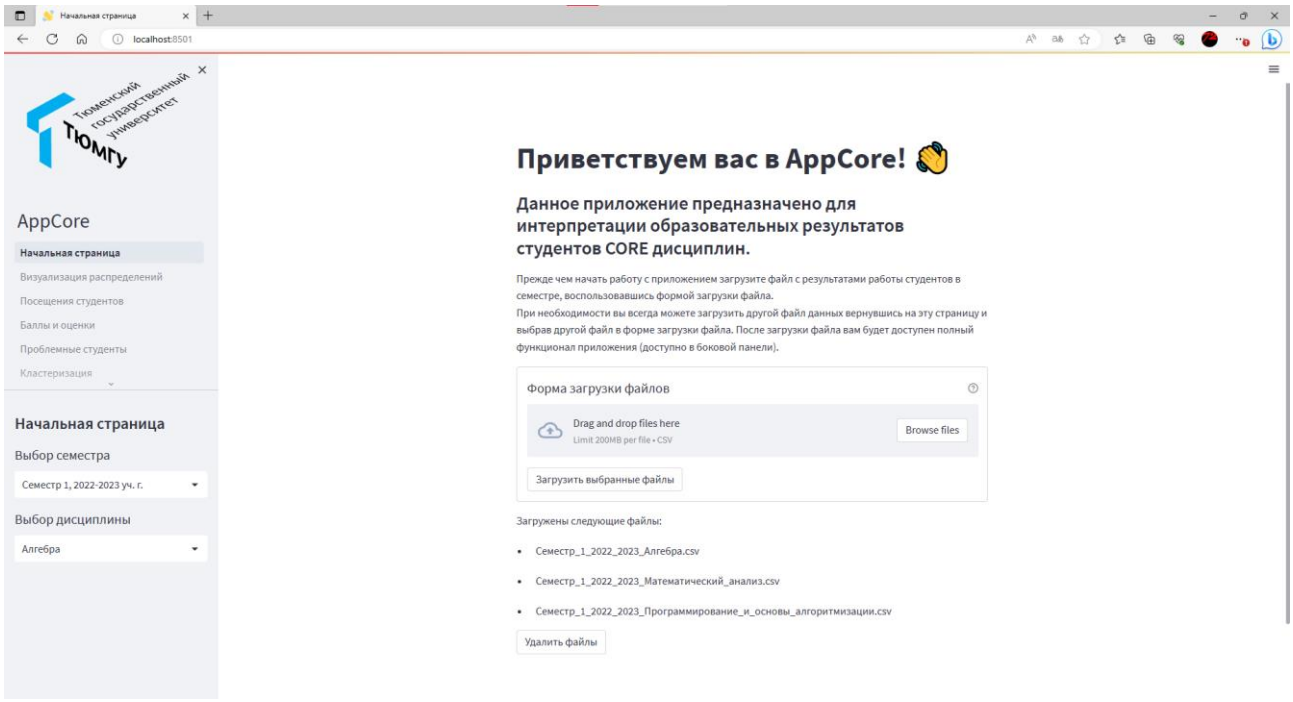


Рис. 20. Начальная страница приложения.

Навигация между страницами осуществляется через левую боковую панель, здесь же осуществляется переключение между дисциплинами.

На странице “Визуализация распределений” осуществляется формирование графиков распределений значений различных показателей студентов, например, контрольных точек, в выбранном разрезе – по направлениям обучения студентов или по преподавателям практики (см. рис.21). Если в фильтре будет выбрано “Преподаватель практики”, то будет также доступна возможность показать распределение значений отдельно по группам – в таком случае не будет доступен выбор количества графиков в пределах одного подграфика, так как все графики группы для каждого преподавателя будут находится в отдельной строке для преподавателя. Выбрав столбец фильтра, можно дополнительно выбрать только нужные значения из этого фильтра. Данный описанный функционал реализован и на дальнейших страницах. На рисунке 22 показаны сформированные графики распределений и дополнительная информация по выбранному графику.

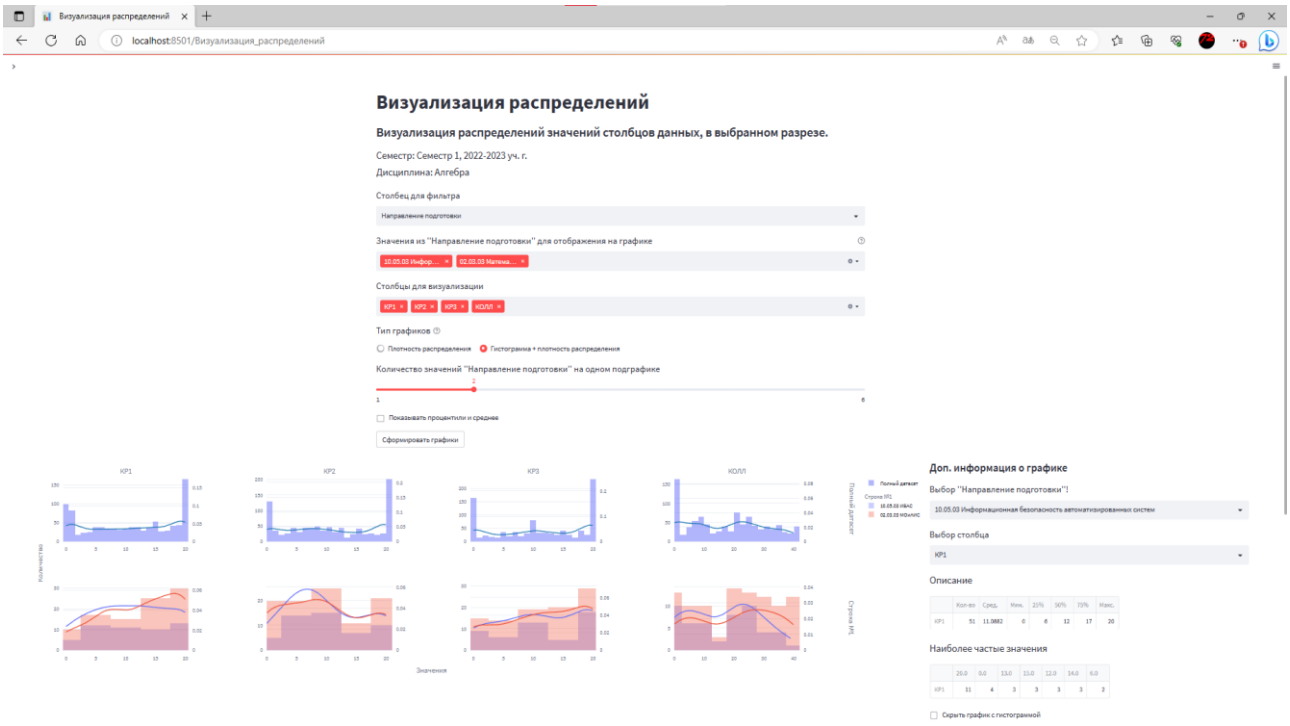


Рис. 21. Страница “Визуализация распределений”.

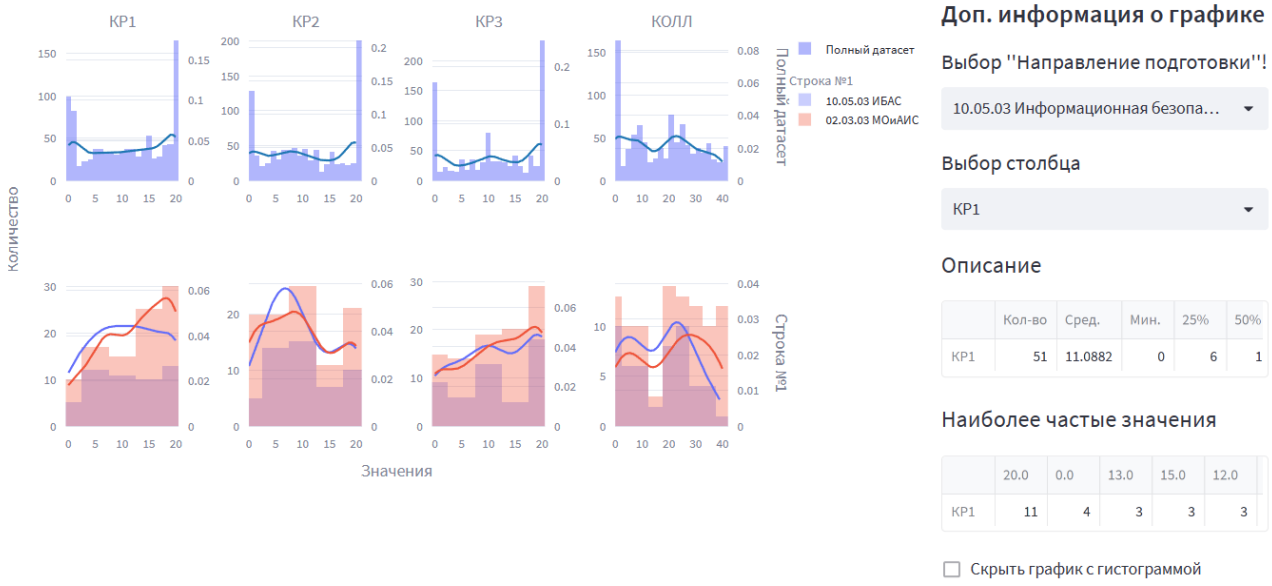


Рис. 22. Построенные графики распределений и дополнительная информация по выбранному графику.

Из уникального для этой страницы:

- Возможность настроить количество графиков в одном подграфике с помощью слайдера.
- Возможность показывать графики как в виде плотности распределения KDE (ядерная оценка плотности – оценка плотности случайной

величины на основе конечной выборки) с помощью соответствующего переключателя.

- Можно выбрать пункт “Показать перцентили и средние”, чтобы отобразить их на графиках (для того чтобы можно было увидеть их не только в полном датасете, нужно выбрать на слайдере один график на подграф).
- Справа от графика отображена дополнительная панель для дополнительной информации, на ней можно увидеть текстовое описание статистических характеристик для графика каждого отдельного графика в виде комбинации графика плотности распределения KDE и гистограммы с настройкой размеров “контейнеров”. Это может быть полезно, в случае, когда значения по умолчанию для размеров “контейнеров” могут быть слишком большими и нужна большая детализация.

На странице “Посещения студентов” отображаются графики посещения студентов в виде гистограмм, где каждая колонка – количество посещений/пропусков для каждого занятия в выбранном разрезе (см. рис.23).

Если в фильтре будет выбрано “Преподаватель практики”, то будет также доступна возможность показать распределение значений отдельно по группам – в таком случае не будет доступен выбор количества графиков в пределах одной строки, так как каждая строка будет отображать графики для одного отдельного преподавателя.

С помощью легенды графика их можно включать только необходимую информацию. Полное количественное значение приведено для понимания количества студентов, рассматриваемое на каждом графике.

На рисунке 24 показаны построенные графики по посещению дисциплины.

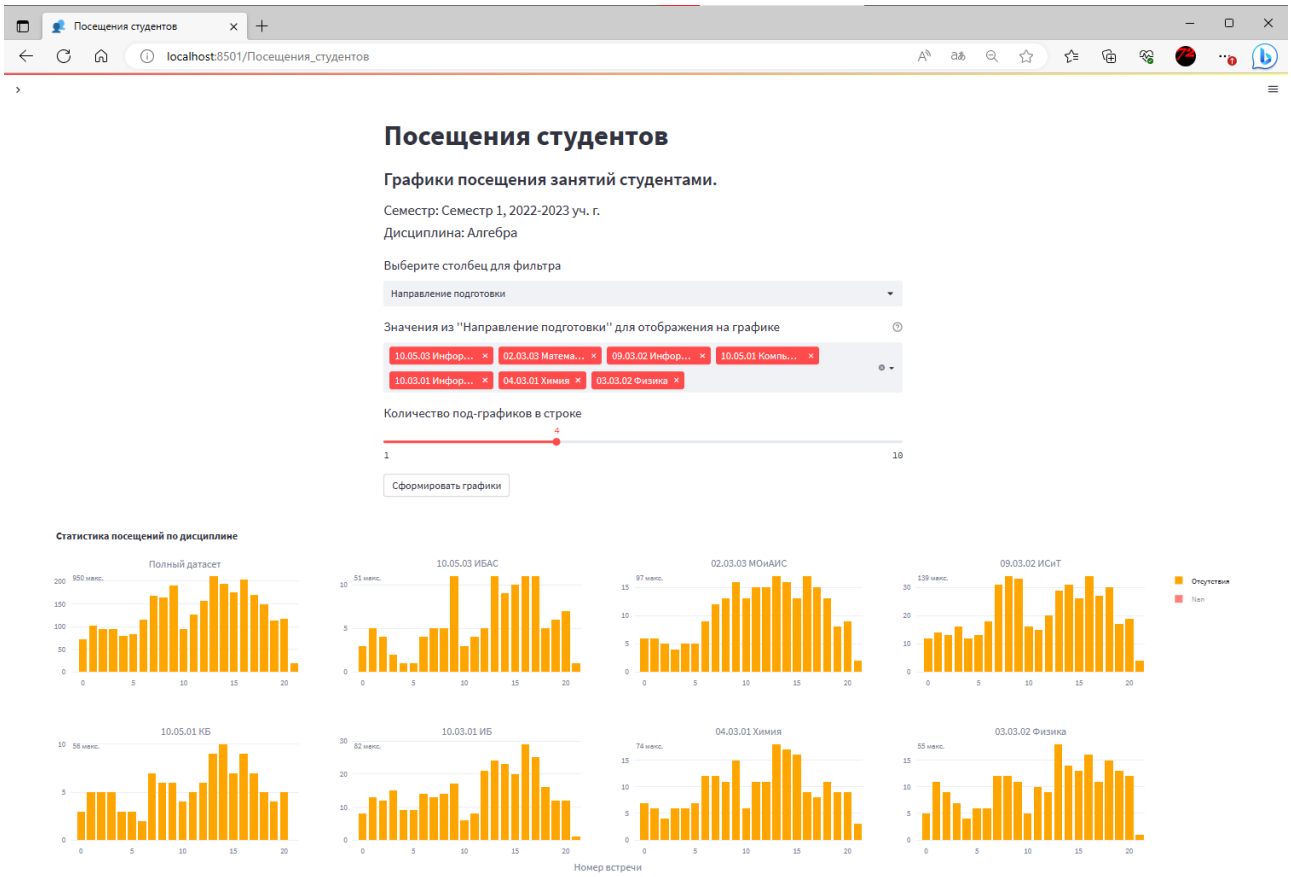


Рис. 23. Страница “Посещения студентов”.

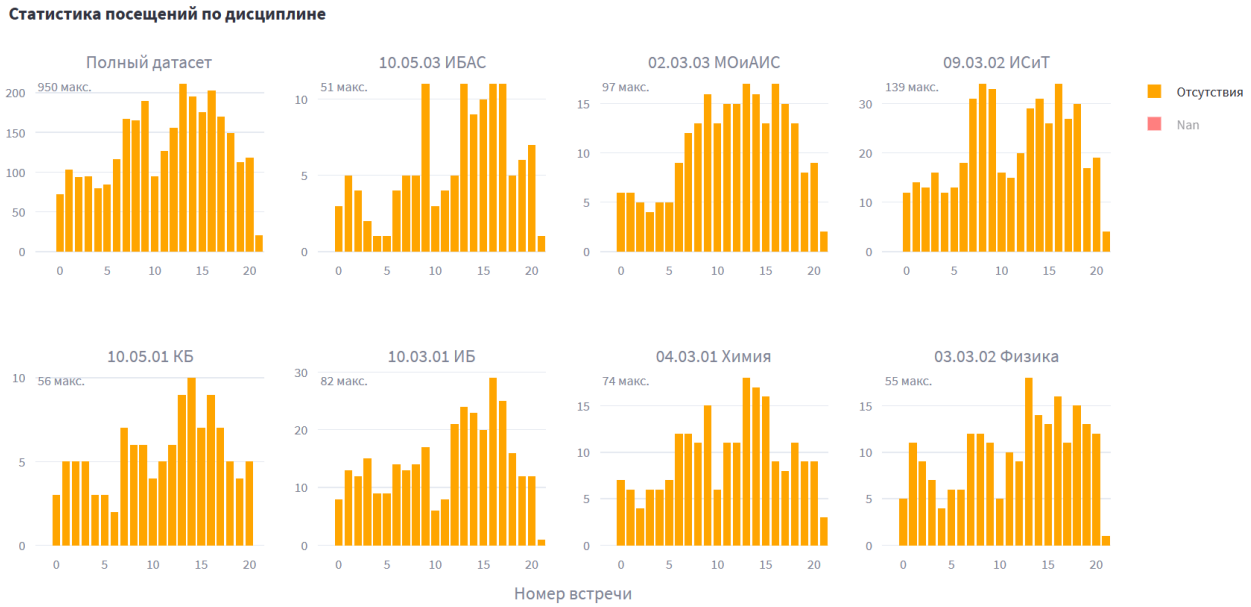


Рис. 24. Построенные графики по посещению дисциплины.

На странице “Баллы и оценки” отображаются комбинированные графики распределения итоговых баллов и оценок студентов в выбранном разрезе (см. рис. 25). Настройка графиков работает аналогично настройке графиков на странице “Посещения студентов”. На графиках отображается гистограмма

оценок, столбцы, расположенные друг над другом, показывают количество итоговых оценок из этого диапазона баллов. Также справа отображается кросс-таблица для выбранного разреза. На рисунке 26 показаны построенные графики баллов и кросс-таблица.

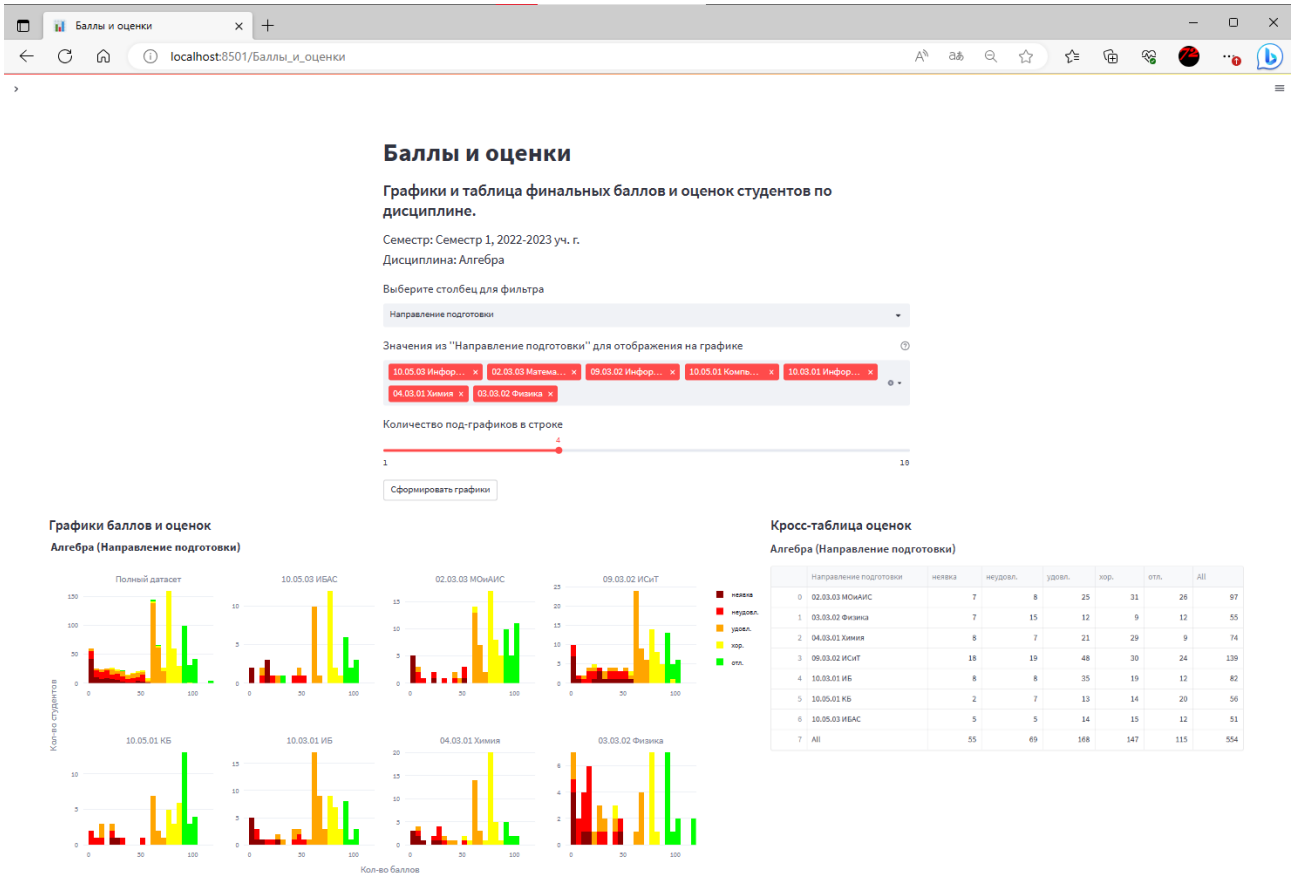
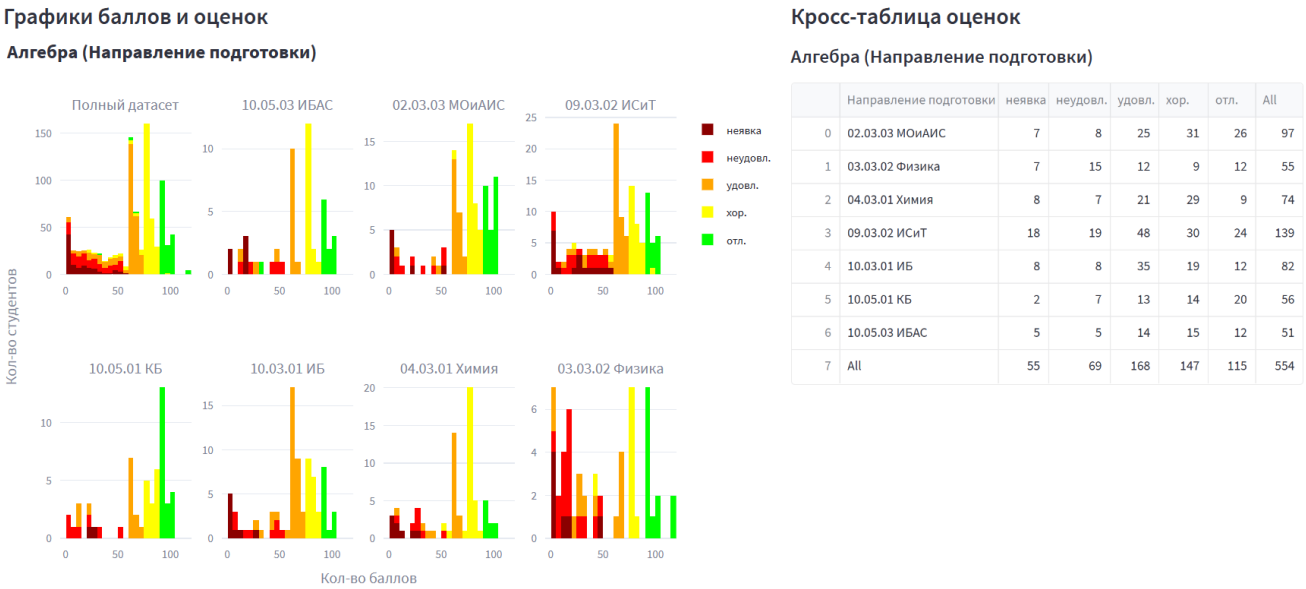


Рис. 25. Страница “Баллы и оценки”.



	Направление подготовки	неявка	неудовл.	удовл.	хор.	отл.	All
0	02.03.03 МОиАИС	7	8	25	31	26	97
1	03.03.02 Физика	7	15	12	9	12	55
2	04.03.01 Химия	8	7	21	29	9	74
3	09.03.02 ИСиТ	18	19	48	30	24	139
4	10.03.01 ИБ	8	8	35	19	12	82
5	10.05.01 КБ	2	7	13	14	20	56
6	10.05.03 ИБАС	5	5	14	15	12	51
7	All	55	69	168	147	115	554

Рис. 26. Построенные графики баллов и кросс-таблица.

Страница “Проблемный студенты” содержит таблицы с студентами, набравшими менее 61 балла и не получившие хорошую оценку (неудовлетворительно и неявка) (см. рис.27). С помощью правой панели можно настроить фильтры для отображения графиков – круговых диаграмм, на которых отображается доля студентов, не сдавших основной выбранный предмет, и дополнительных, в разрезе выбранных фильтров (преподавателя по практике или направления подготовки). При нажатии на сегмент круговой диаграммы оценок, появится вторая диаграмма, отражающая долю людей с посещаемостью в менее или равно 50% и более 50% для выбранного сегмента, а также появится таблица отфильтрованных значений первой таблицы на основе сделанного выбора. При нажатии на сегмент второй круговой диаграммы добавится второй фильтр, соответствующий выбранному сегменту. Обе таблицы представляют собой интерактивные элементы с возможностью дополнительной фильтрации по значениям столбцов.

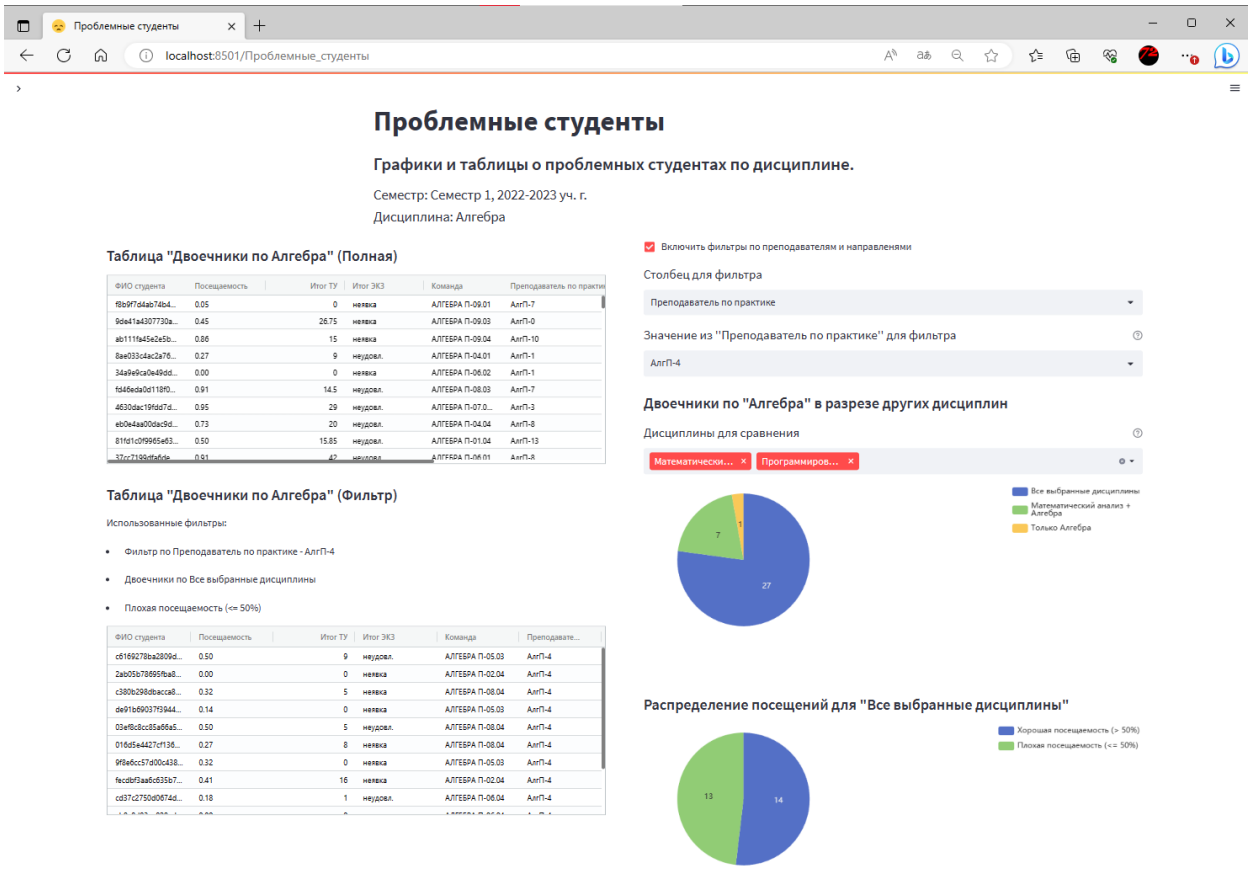


Рис. 27. Страница “Проблемные студенты”.

Страница “Кластерный анализ результатов мониторинга” содержит инструмент для проведения кластерного анализа по имеющимся данным студентов (см. рис.28). Перед началом работы можно выбрать данные, которые будут использоваться как входные данные для кластеризации, а также значения для отображения дополнительной информации о результирующих кластерах. Дополнительные данные не участвуют в процессе кластеризации и отображаются на графики параллельных координат, а также в таблице описательной статистики и таблице наполнения кластера.

В зависимости от количества выбранных признаков для кластеризации изменяется график разброса:

- Для одного признака – график не показывается, достаточно графика параллельных координат.
- Для двух признаков – используется двухмерный график разброса по выбранным признакам.
- Для более двух признаков – используется двухмерный график разброса, размерность оригинальных данных снижается до двух компонент (с помощью метода TSNE), график формируются по данным компонентам.

График параллельных координат во многих случаях может оказаться достаточно сложным для прочтения. Интерактивный график plotly позволяет выбирать, какие значения отображать (например, только выбранные кластеры).

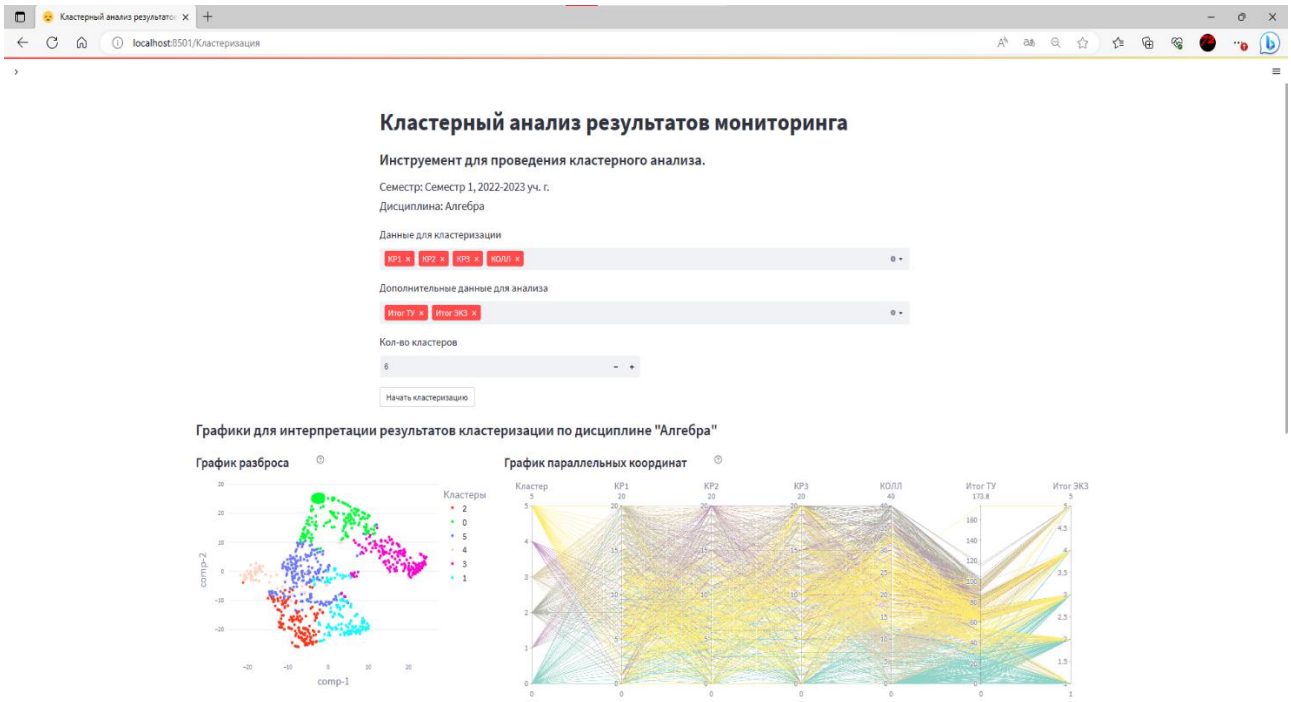


Рис. 28. Страница “Кластерный анализ результатов мониторинга” – настройка кластеризации, графики разброса и параллельных координат.

На вкладке “По одному кластеру” находится информация для выбираемого кластера (см. рис.29). Имеются круговые диаграммы в разрезе направлений и преподавателей, график распределения баллов и оценок, описательная статистика и таблица наполнения кластера. Зеленым цветом в таблице описательной статистики обозначены средние значения выбранных признаков для кластеризации, соответствующие центроиду кластера.

Как и в случае с проблемными студентами, для отображения таблицы наполнения кластеров используется специальный вид интерактивной таблицы, используемый для дополнительной фильтрации результатов.

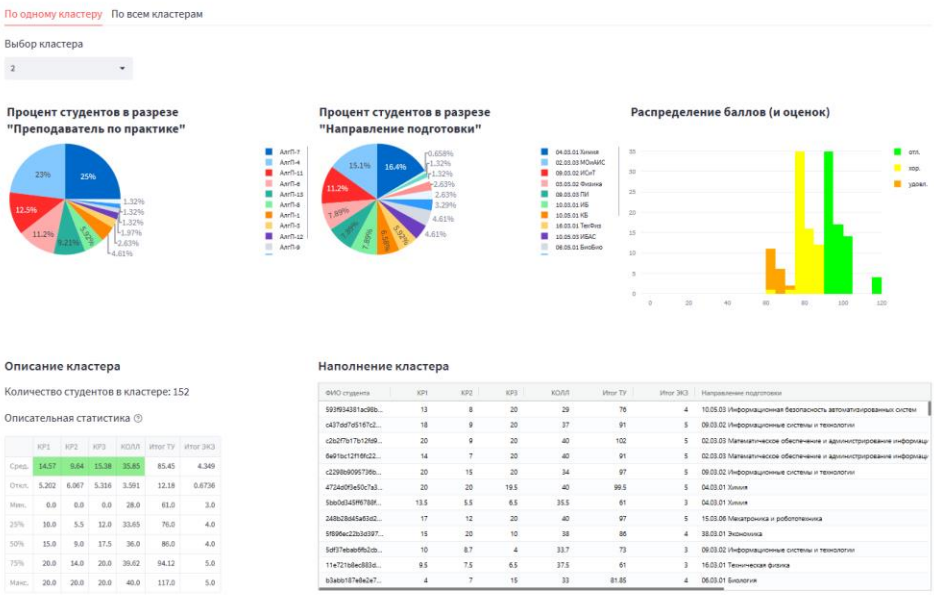


Рис. 29. Страница “Кластерный анализ результатов мониторинга” – вкладка с информацией для одного выбранного кластера.

На вкладке “По всем кластерам” в зависимости от выбранного столбца для фильтра показывается график для каждого кластера – уже упомянутые круговые диаграммы и график распределения значений баллов студентов (см. рис.30). Данная вкладка позволяет более детально рассмотреть особенности наполнения кластеров относительно друг друга.

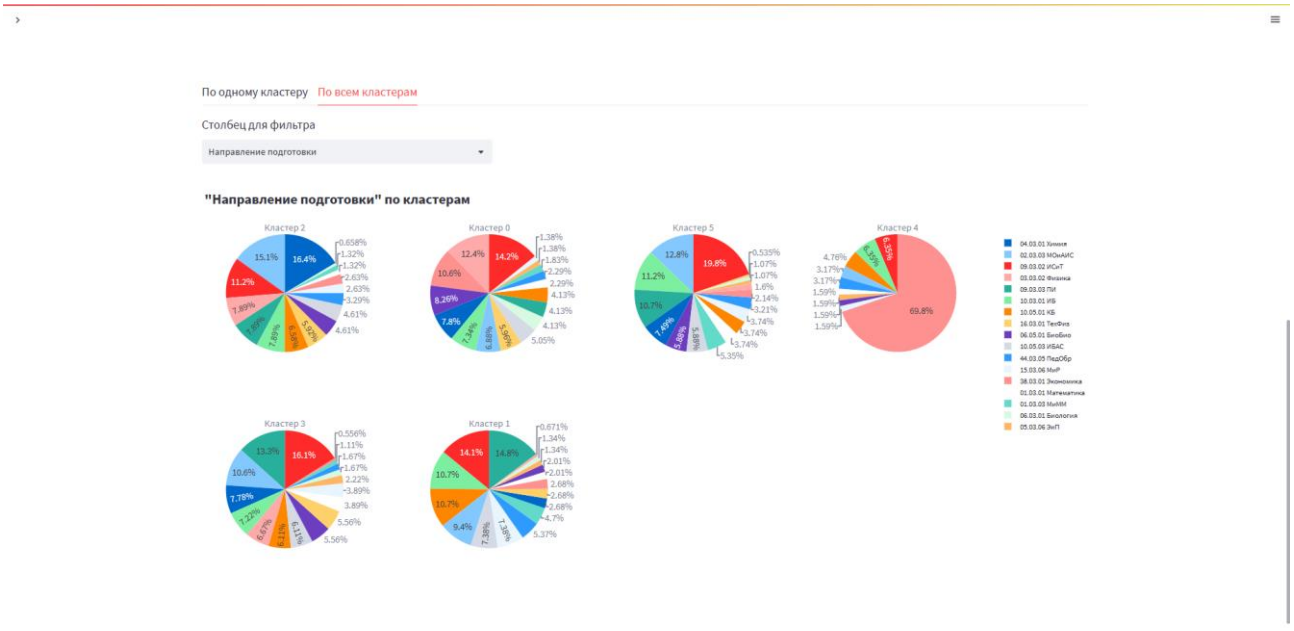


Рис. 30. Страница “Кластерный анализ результатов мониторинга” – вкладка с информацией о всех кластерах.

ЗАКЛЮЧЕНИЕ

В ходе написания выпускной квалификационной работы были выполнены следующие задачи:

- В результате изучения предметной области были рассмотрены существующие программы для визуализации и анализа результатов образовательной деятельности студентов.
- Были получены данные студентов по дисциплинам CORE;
- Проведена предобработка и предварительный анализ имеющихся данных студентов.
- Дополнительно проведен кластерный анализ данных студентов.
- Определен функционал целевого приложения.
- Определены средства разработки для реализации приложения.
- Разработано и протестировано приложение.

Разработанный программный продукт обладает следующим функционалом:

- Возможность загружать и удалять файлы с данными о результатах учебной деятельности студента по дисциплинам формата .csv.
- Возможность визуализировать распределения значений признаков для студентов (финальные баллы, общие баллы за практические занятия и контрольные мероприятия, баллы отдельно по каждому контрольному мероприятию, кол-во посещений) в виде графиков для предмета, в разрезе направлений подготовки, преподавателей и их команд студентов.
- Визуализация финальных образовательных результатов (посещений и комбинации баллов и оценок за экзамен/зачет) в виде графиков и таблиц для предмета, в разрезе направлений подготовки, преподавателей и их команд студентов;
- Визуализация в виде графиков и таблиц количества и уровня посещаемости проблемных студентов для предмета, в разрезе

направлений подготовки, преподавателей и в совокупности с результатами по другим предметам.

- Возможность проводить кластеризацию с использованием данных студентов и визуализировать результаты в виде графиков и таблиц, показывающих особенности и наполнение кластеров для проведения кластерного анализа, интерпретации результатов кластеризации.

СПИСОК ЛИТЕРАТУРЫ

1. Баранова Е. В., Гизатуллина Г. С. Модель веб-ресурса «Деканат» как компонента интегрированной системы управления учебным процессом // Сборник научных статей по материалам международной научной конференции 1–12 апреля 2019 года. СПб.: Изд-во РГПУ им. А. И. Герцена, 2019. С. 144.
2. Modeus (Модеус) – платформа управления ИОТ: [Электронный ресурс]. URL: https://modeus.custis.ru. (Дата обращения: 16.06.2023).
3. Moodle – Open-source learning platform: [Электронный ресурс]. URL: https://moodle.org. (Дата обращения: 16.06.2023).
4. Баранова Е. В. Цифровые инструменты для анализа учебной деятельности студентов / Е. В. Баранова, Н. О. Верещагина, Г. В. Швецов // Известия Российского государственного педагогического университета им. А.И. Герцена. – 2020. – № 198. – P. 56-65.
5. Moodle plugins directory: Analytics graphs: [Электронный ресурс] // Moodle – Open-source learning platform. URL: https://moodle.org/plugins/block_analytics_graphs. (Дата обращения: 16.06.2023).
6. Moodle plugins directory: Overview statistics: [Электронный ресурс] // Moodle – Open-source learning platform. URL: https://moodle.org/plugins/report_overviewstats. (Дата обращения: 16.06.2023).
7. Ferguson R. Learning analytics: drivers, developments and challenges // International Journal of Technology Enhanced Learning. – 2012. – Vol. 4. – №. 5-6. – P. 304-317.
8. Сокольников, А. Н. Математические методы в анализе учебной деятельности студентов / А. Н. Сокольников // Педагогическое образование и наука. – 2022. – № 2. – С. 135-138.
9. Арефьев, В. П. Кластерный анализ результатов оценивания знаний в системе заочного обучения с использованием дистанционных

образовательных технологий / В. П. Арефьев, А. А. Михальчук, Н. М. Филипенко // Современные проблемы науки и образования. – 2013. – № 3. – С. 428.

10. Akçapınar G., Altun A., Aşkar P. Using learning analytics to develop early-warning system for at-risk students //International Journal of Educational Technology in Higher Education. – 2019. – Vol. 16. – №. 1. – P. 1-20.
11. Govaerts S. et al. The student activity meter for awareness and self-reflection // CHI'12 Extended Abstracts on Human Factors in Computing Systems. – 2012. – P. 869-884.
12. Gutiérrez F. et al. LADA: A learning analytics dashboard for academic advising // Computers in Human Behavior. – 2020. – Vol. 107. – P. 105826.
13. Charleer S. et al. Learning analytics dashboards to support adviser-student dialogue // IEEE Transactions on Learning Technologies. – 2017. – Vol. 11. – №. 3. – P. 389-399.
14. Создай индивидуальный образовательный трек | ТюмГу: [Электронный ресурс] // Тюменский государственный университет. URL: https://www.utmn.ru/obrazovanie/priority-education/. (Дата обращения: 16.06.2023).
15. Ahmed M., Seraj R., Islam S. M. S. The k-means algorithm: A comprehensive survey and performance evaluation // Electronics. – 2020. – Vol. 9. – №. 8. – P. 1295.
16. Schubert E. et al. DBSCAN revisited, revisited: why and how you should (still) use DBSCAN //ACM Transactions on Database Systems (TODS). – 2017. – Vol. 42. – №. 3. – P. 1-21.
17. Murtagh F., Contreras P. Algorithms for hierarchical clustering: an overview // Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery. – 2012. – Vol. 2. – №. 1. – P. 86-97.
18. Van der Maaten, L., & Hinton, G. Visualizing data using t-SNE – Journal of machine learning research, 9(11), 2008. – P. 2580-2605

19. Holland, Steven M. Principal components analysis (PCA). – Department of Geology, University of Georgia, Athens, GA., 2008. – P. 1-11.
20. Tharwat, A., Gaber, T., Ibrahim, A., & Hassanien, A. E. Linear discriminant analysis: A detailed tutorial. – AI communications, 30(2)., 2017. – P. 169-190.
21. Pandas documentation – pandas 1.3.5 documentation: [Электронный ресурс] // Pandas – Python Data Analysis Library. URL: https://pandas.pydata.org/docs/. (Дата обращения: 10.05.2023).
22. Statsmodels documentation – Statsmodels v0.13.1 User Guide: [Электронный ресурс] // Statsmodels. URL: https://www.statsmodels.org/stable/user-guide.html. (Дата обращения: 10.05.2023).
23. Sckit-learn documentation – Sckit-learn 1.2.2 User Guide: [Электронный ресурс] // Sckit-learn. URL: https://scikit-learn.org/stable/user_guide.html. (Дата обращения: 10.05.2023).
24. Streamlit documentation – Streamlit Documentation: [Электронный ресурс] // Streamlit. URL https://docs.streamlit.io. (Дата обращения: 10.05.2023).

Приложение 1

Код функции для формирования графиков распределения образовательных показателей студентов в выбранном разрезе

```
def get_multi_dist_plot (df, frame_columns, filter_col = None,
selected_values_of_filter = None,  n = 1, draw_info = False, height = 350):
    rows = 1
    if filter_col is not None:
        lst = df[filter_col].unique().tolist().copy() if
selected_values_of_filter is not None and selected_values_of_filter else
selected_values_of_filter
        lst = [x for x in lst if pd.isnull(x) == False]
        jobs = [lst[i:i + n] for i in range(0, len(lst), n)]
        rows = len(jobs) + 1
    row_titles = ["Полный датасет"] + ["Строка №" + str(i-1) for i in range(2,
rows + 1)]
    fig = make_subplots(rows=rows, cols=len(frame_columns),
subplot_titles=frame_columns, row_titles=row_titles, x_title='Значения',
y_title='Плотность')
    col_xaxis_range = {}
    for i, col in enumerate(frame_columns):
        if draw_info:
            fig.layout.annotations[i].update(y=1 + (45/(250*rows)))
        hist_data = [df[col].to_list()]
        dist_fig = ff.create_distplot(hist_data=hist_data, group_labels=['Полный
датасет'], bin_size=2, show_rug=False, show_hist=False)
        for trace in dist_fig.select_traces():
            trace['showlegend'] = (True if i == 0 else False)
            col_xaxis_range[col] =
{'xmin':min(trace['x']),'xmax':max(trace['x'])}
            fig.add_trace(trace, row=1, col=i+1)
            if draw_info:
                show_info(fig=fig, hist_data=hist_data[0], row=1, col=i+1)
    if filter_col is not None:
        for i in range(2, rows + 1):
            for j, col in enumerate(frame_columns):
                hist_data = [df[df[filter_col] == job][col].to_list() for job in
jobs[i-2]]
                dist_fig = ff.create_distplot(hist_data=hist_data,
group_labels=jobs[i-2], bin_size=2, show_rug=False, show_hist=False)
                for trace in dist_fig.select_traces():
                    trace['showlegend'] = (True if j == 0 else False)
                    trace['legendgroup'] = "group" + str(i-1)
                    trace['legendgrouptitle_text'] = "Строка №" + str(i-1)
                    fig.add_trace(trace, row=i, col=j+1)
                    if n==1 and draw_info:
                        show_info(fig=fig, hist_data=hist_data[0], row=i,
col=j+1)
                fig.update_xaxes(range=[col_xaxis_range[col]['xmin'],
col_xaxis_range[col]['xmax']], row=i, col=j+1)
    fig.update_layout(margin=dict(t=(100 if draw_info else 45)))
    fig.update_layout(height=rows*height)
    return fig
```

Приложение 2

Код функции для формирования графиков посещаемости студентов в выбранном разрезе

```
def get_multi_attendance_plot(df, filter_col, frame_cols, value_vars_cols, filter_col_values = None,  num_cols = 5, height = 300):
    full_df = df[frame_cols].melt(id_vars=['ФИО студента',filter_col], value_vars=df[value_vars_cols].columns[1:])
    full_df['variable'] = full_df['variable'].str[1:].astype(int)
    full_df['value'] = full_df['value'].astype(str)
    gp_att_full = full_df.groupby(['variable'])['value'].apply(lambda x: ((x == '1.0')).sum()).reset_index(name='count')
    gp_att = (full_df[full_df[filter_col].isin(filter_col_values)] if filter_col_values is not None and filter_col_values else full_df).groupby([filter_col,'variable'])['value'].apply(lambda x: ((x == '1.0')).sum()).reset_index(name='count')
    gp_none_full = full_df.groupby(['variable'])['value'].apply(lambda x: ((x == '0.0')).sum()).reset_index(name='count')
    gp_non = (full_df[full_df[filter_col].isin(filter_col_values)] if filter_col_values is not None and filter_col_values else full_df).groupby([filter_col,'variable'])['value'].apply(lambda x: ((x == '0.0')).sum()).reset_index(name='count')
    lst = full_df[full_df[filter_col].isin(filter_col_values)][filter_col].unique().tolist().copy() if filter_col_values is not None and filter_col_values else full_df[filter_col].unique().tolist().copy()
    lst = ['Полный датасет'] + [x for x in lst if pd.isnull(x) == False]
    n = num_cols
    rows = (len(lst) - 1) // n + 1
    fig = make_subplots(rows=rows, cols=n, subplot_titles=lst, x_title='Номер встречи', y_title='Кол-во студентов')
    for i, cn in enumerate(lst):
        fig.add_trace(go.Bar(x=gp_att[gp_att[filter_col] == cn]['variable'] if cn != 'Полный датасет' else gp_att_full['variable'],
                             y=gp_att[gp_att[filter_col] == cn]['count'] if cn != 'Полный датасет' else gp_att_full['count'],
                             name='Отсутствия', showlegend=True if i == 0 else False, legendgroup='group1', marker = dict(color = 'orange')), row = i // n + 1, col = i % n + 1)
        fig.add_trace(go.Bar(x=gp_non[gp_non[filter_col] == cn]['variable'] if cn != 'Полный датасет' else gp_none_full['variable'],
                             y=gp_non[gp_non[filter_col] == cn]['count'] if cn != 'Полный датасет' else gp_none_full['count'],
                             name='Nan', showlegend=True if i == 0 else False, legendgroup='group2', marker = dict(color = 'red')), row = i // n + 1, col = i % n + 1)
        if cn != 'Полный датасет':
            max_stud = gp_full_df[gp_full_df[filter_col] == cn]['count'].tolist()[0] + gp_att[gp_att[filter_col] == cn]['count'].tolist()[0] + gp_non[gp_non[filter_col] == cn]['count'].tolist()[0]
        else:
            max_stud = gp_full_df_full['count'].tolist()[0] + gp_att_full['count'].tolist()[0] + gp_none_full['count'].tolist()[0]
        fig.add_annotation(text=f"{max_stud} макс.", row = i // n + 1, col = i % n + 1, xref="x", yref="y domain", x=0.5, y=1.0, showarrow=False)
    fig.update_layout(height=height*rows. title_text="Статистика посещений по предмету", barmode='stack')
    return fig
```

Приложение 3

Код функции для формирования графиков распределения баллов и оценок студентов в выбранном разрезе

```
def get_multi_mark_plot(df, filter_col, filter_col_values = None, num_cols = 5,
name = '', hieght = 300):
    if filter_col_values is not None and filter_col_values:
        lst =
df[df[filter_col].isin(filter_col_values)][filter_col].unique().tolist().copy()
    else:
        lst = df[filter_col].unique().tolist().copy()
    lst = ['Полный датасет'] + [x for x in lst if pd.isnull(x) == False]
    cols = num_cols
    rows = (len(lst) - 1) // cols + 1
    fig = make_subplots(rows=rows, cols=cols, subplot_titles=lst, x_title='Кол-
во баллов', y_title='Кол-во студентов')
    bin_size = dict(start=0.0, end=120.0, size=5)
    for i, grp in enumerate(lst):
        check = (grp == 'Полный датасет')
        fig.add_trace(
            go.Histogram(x=(df[df['Итог ЭКЗ']==1] if check else
df[(df[filter_col]==grp) & (df['Итог ЭКЗ']==1)])['Итог ТУ'], marker = dict(color
= 'darkred'),  name='неявка', legendgroup='group5',
            showlegend=True if check else False), row=(i // cols)+1, col=(i %
cols)+1)
        fig.add_trace(
            go.Histogram(x=(df[df['Итог ЭКЗ']==1] if check else
df[(df[filter_col]==grp) & (df['Итог ЭКЗ']==2)])['Итог ТУ'], marker = dict(color
= 'red'),  name='неудовл.', legendgroup='group4',
            showlegend=True if check else False), row=(i // cols)+1, col=(i %
cols)+1)
        fig.add_trace(
            go.Histogram(x=(df[df['Итог ЭКЗ']==1] if check else
df[(df[filter_col]==grp) & (df['Итог ЭКЗ']==3)])['Итог ТУ'], marker = dict(color
= 'orange'),  name='удовл.', legendgroup='group3',
            showlegend=True if check else False), row=(i // cols)+1, col=(i %
cols)+1)
        fig.add_trace(
            go.Histogram(x=(df[df['Итог ЭКЗ']==1] if check else
df[(df[filter_col]==grp) & (df['Итог ЭКЗ']==4)])['Итог ТУ'], marker = dict(color
= 'yellow'),  name='хор.', legendgroup='group2',
            showlegend=True if check else False), row=(i // cols)+1, col=(i %
cols)+1)
        fig.add_trace(
            go.Histogram(x=(df[df['Итог ЭКЗ']==1] if check else
df[(df[filter_col]==grp) & (df['Итог ЭКЗ']==5)])['Итог ТУ'], marker = dict(color
= 'lime'),  name='отл.', legendgroup='group1',
            showlegend=True if check else False), row=(i // cols)+1, col=(i %
cols)+1)
    fig.update_layout(title_text=f"{name} ({filter_col})", barmode='stack',
height=rows*hieght)
    fig.update_traces(xbins=bin_size)
    fig.update_xaxes(range=[-5,120])
    return fig
```