TOMSK POLYTECHNIC UNIVERSITY ТОМСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ

Министерство науки и высшего образования Российской Федерации
федеральное государственное автономное
образовательное учреждение высшего образования
«Национальный исследовательский Томский политехнический университет» (ТПУ)

Инженерная школа информационных технологий и робототехники
Направление подготовки: 09.03.04 «Программная инженерия»
Отделение информационных технологий

БАКАЛАВРСКАЯ РАБОТА

Тема работы
Использование инструментов Data Mining для анализа успеваемости студентов университета

УДК 004.65:378.141.261-057.87

Студенты

Группа	ФИО	Подпись	Дата
8К71	Галлингер Владислав Андреевич		
8К71	Семенюта Антон Вадимович		

Руководитель ВКР

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Доцент	Губин Евгений Иванович	к.ф.-м.н.		

Научный консультант

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Ассистент	Кайда Анастасия Юрьевна	-		

КОНСУЛЬТАНТЫ ПО РАЗДЕЛАМ:

По разделу «Финансовый менеджмент, ресурсоэффективность и ресурсосбережение»

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Доцент	Маланина Вероника Анатольевна	к.э.н.		

По разделу «Социальная ответственность»

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Ассистент	Черемискина Мария Сергеевна	-		

ДОПУСТИТЬ К ЗАЩИТЕ:

Руководитель ООП	ФИО	Ученая степень, звание	Подпись	Дата
Доцент	Чердынцев Евгений Сергеевич	к.т.н.		

Томск – 2021 г.

Планируемые результаты обучения по ООП

Код результатов	Результат обучения (выпускник должен быть готов)
Р1	Применять базовые и специальные естественнонаучные и математические знания в области информатики и вычислительной техники, достаточные для комплексной инженерной деятельности.
Р2	Применять базовые и специальные знания в области современных информационных технологий для решения инженерных задач.
Р3	Ставить и решать задачи комплексного анализа, связанные с созданием аппаратно-программных средств информационных и автоматизированных систем, с использованием базовых и специальных знаний, современных аналитических методов и моделей.
Р4	Разрабатывать программные и аппаратные средства (системы, устройства, блоки, программы, базы данных и т. п.) в соответствии с техническим заданием и с использованием средств автоматизации проектирования.
Р5	Проводить теоретические и экспериментальные исследования, включающие поиск и изучение необходимой научно-технической информации, математическое моделирование, проведение эксперимента, анализ и интерпретация полученных данных, в области создания аппаратных и программных средств информационных и автоматизированных систем.
Р6	Внедрять, эксплуатировать и обслуживать современные программно-аппаратные комплексы, обеспечивать их высокую эффективность, соблюдать правила охраны здоровья, безопасность труда, выполнять требования по защите окружающей среды.
Р7	Универсальные компетенции
Р8	Использовать базовые и специальные знания в области проектного менеджмента для ведения комплексной инженерной деятельности.
Р9	Владеть иностранным языком на уровне, позволяющем работать в иноязычной среде, разрабатывать документацию, презентовать и защищать результаты комплексной инженерной деятельности.
Р10	Эффективно работать индивидуально и в качестве члена группы, состоящей из специалистов различных направлений и квалификаций, демонстрировать ответственность за результаты работы и готовность следовать корпоративной культуре организации.
Р11	Демонстрировать знания правовых, социальных, экономических и культурных аспектов комплексной инженерной деятельности.

Министерство образования и науки Российской Федерации
федеральное государственное автономное образовательное учреждение
высшего образования
**«НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ
ТОМСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»**

Инженерная школа информационных технологий и робототехники
Направление подготовки 09.03.04 «Программная инженерия»
Отделение информационных технологий

УТВЕРЖДАЮ:
Руководитель ООП

_____ _____ _____
(Подпись) (Дата) (Ф.И.О.)

ЗАДАНИЕ
на выполнение выпускной квалификационной работы

В форме:

Бакалаврской работы

(бакалаврской работы, дипломного проекта/работы, магистерской диссертации)

Студентам:

Группа	ФИО
8К71	Галлингеру Владиславу Андреевичу
8К71	Семенюте Антону Вадимовичу

Тема работы:

Использование инструментов Data Mining для анализа успеваемости студентов университета	
Утверждена приказом директора (дата, номер)	

Срок сдачи студентом выполненной работы:	

ТЕХНИЧЕСКОЕ ЗАДАНИЕ:

Исходные данные к работе	Объектом исследования являются данные об успеваемости студентов на весенний семестр 2019 года и данные о средней успеваемости групп по определенным дисциплинам у определенных преподавателей на осенний семестр 2019 года.
Перечень подлежащих исследованию, проектированию и разработке вопросов	1. Исследование предметной области 2. Интеллектуальных анализ данных об успеваемости студентов; 3. Проектирование программной системы; 4. Разработка программной системы;

	5. Финансовый менеджмент, ресурсоэффективность и ресурсосбережение; 6. Социальная ответственность.
Перечень графического материала	1. Пояснительные скриншоты и графики исследования; 2. Диаграммы UML; 3. Пояснительные скриншоты веб-приложения; 4. Матрица SWOT-анализа; 5. Диаграмма Ганта.

Консультанты по разделам выпускной квалификационной работы	
Раздел	**Консультант**
Финансовый менеджмент, ресурсоэффективность и ресурсосбережение	Маланина Вероника Анатольевна
Социальная ответственность	Черемискина Мария Сергеевна

Дата выдачи задания на выполнение выпускной квалификационной работы по линейному графику	

Задание выдал руководитель:

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Доцент	Губин Евгений Иванович	к.ф.-м.н.		

Задание приняли к исполнению студенты:

Группа	ФИО	Подпись	Дата
8К71	Галлингер Владислав Андреевич		
8К71	Семенюта Антон Вадимович		

Министерство образования и науки Российской Федерации
федеральное государственное автономное образовательное учреждение
высшего образования
**«НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ
ТОМСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»**

Инженерная школа информационных технологий и робототехники
Направление подготовки 09.03.04 «Программная инженерия»
Уровень образования бакалавриат
Отделение информационных технологий
Период выполнения осенний / весенний семестр 2020/2021 учебного года

Форма представления работы:

Бакалаврская работа

(бакалаврская работа, дипломный проект/работа, магистерская диссертация)

КАЛЕНДАРНЫЙ РЕЙТИНГ-ПЛАН
выполнения выпускной квалификационной работы

Срок сдачи студентом выполненной работы:	

Дата контроля	Название раздела (модуля) / вид работы (исследования)	Максимальный балл раздела (модуля)
01.02.2021	Раздел 1. Исследование предметной области	10
20.02.2021	Раздел 2. Интеллектуальный анализ данных об успеваемости студентов	20
15.03.2021	Раздел 3. Проектирование программной системы	10
29.03.2021	Раздел 4. Разработка программной системы	20
13.05.2021	Раздел 5. Финансовый менеджмент, ресурсоэффективность и ресурсосбережение	10
16.05.2021	Раздел 6. Социальная ответственность	10

Составил преподаватель:

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Доцент	Губин Евгений Иванович	к.ф.-м.н.		

СОГЛАСОВАНО:

Руководитель ООП	ФИО	Ученая степень, звание	Подпись	Дата
Доцент	Чердынцев Евгений Сергеевич	к.т.н.		

ЗАДАНИЕ ДЛЯ РАЗДЕЛА
«ФИНАНСОВЫЙ МЕНЕДЖМЕНТ, РЕСУРСОЭФФЕКТИВНОСТЬ И РЕСУРСОСБЕРЕЖЕНИЕ»

Студентам:

Группа	ФИО
8К71	Галлингер Владислав Андреевич
8К71	Семенюта Антон Вадимович

Школа	ИШИТР	Отделение школы (НОЦ)	ОИТ
Уровень образования	Бакалавриат	Направление/специальность	09.03.04 Программная инженерия

Тема ВКР:

Использование инструментов Data Mining для анализа успеваемости студентов университета	
Исходные данные к разделу «Финансовый менеджмент, ресурсоэффективность и ресурсосбережение»:	
1. *Стоимость ресурсов научного исследования (НИ): материально-технических, энергетических, финансовых, информационных и человеческих* 2. *Нормы и нормативы расходования ресурсов* 3. *Используемая система налогообложения, ставки налогов, отчислений, дисконтирования и кредитования*	*Материалы и покупные изделия 1750,00 рублей, амортизационные отчисления – 7800,00 рублей, затраты на основную заработную плату 190069,10 рублей, затраты на отчисление во внебюджетные фонды – 57020,74 рублей, накладные расходы – 41062,37 рублей.*
Перечень вопросов, подлежащих исследованию, проектированию и разработке:	
1. *Оценка коммерческого потенциала, перспективности и альтернатив проведения НИ с позиции ресурсоэффективности и ресурсосбережения*	*Потенциальные потребители результатов исследования. Анализ конкурентных технических решений. SWOT – анализ.*
2. *Планирование и формирование бюджета научных исследований*	*Структура работ в рамках научного исследования. Определение трудоемкости выполнения работ. Разработка графика проведения научного исследования. Бюджет научно-технического исследования*
3. *Определение ресурсной (ресурсосберегающей), финансовой, бюджетной, социальной и экономической эффективности исследования*	*Определение интегрального финансового показателя разработки. Определение интегрального показателя ресурсоэффективности разработки. Определение интегрального показателя эффективности.*
Перечень графического материала *(с точным указанием обязательных чертежей)***:**	
1. *Оценка конкурентоспособности технических решений* 2. *Матрица SWOT* 3. *Альтернативы проведения НИ* 4. *График проведения и бюджет НИ* 5. *Оценка ресурсной, финансовой и экономической эффективности НИ*	

Дата выдачи задания для раздела по линейному графику	01.03.2021

Задание выдал консультант:

Должность	ФИО	Ученая степень, звание	Подпись	Дата
доцент	Маланина Вероника Анатольевна	Кандидат экономических наук		

Задание приняли к исполнению студенты:

Группа	ФИО	Подпись	Дата
8К71	Галлингер Владислав Андреевич		
8К71	Семенюта Антон Вадимович		

ЗАДАНИЕ ДЛЯ РАЗДЕЛА
«СОЦИАЛЬНАЯ ОТВЕТСТВЕННОСТЬ»

Студентам:

Группа	ФИО
8К71	Галлингер Владислав Андреевич
8К71	Семенюта Антон Вадимович

Школа	Инженерная школа информационных технологий и робототехники	Отделение (НОЦ)	Отделение информационных технологий
Уровень образования	Бакалавриат	Направление/специальность	09.03.04 «Программная инженерия»

Тема ВКР:

Использование инструментов Data Mining для анализа успеваемости студентов университета	
Исходные данные к разделу «Социальная ответственность»:	
1. Характеристика объекта исследования (вещество, материал, прибор, алгоритм, методика, рабочая зона) и области его применения	Объект исследования: данные о характеристиках студентов университета. Область применения: системы контроля успеваемости студентов.
Перечень вопросов, подлежащих исследованию, проектированию и разработке:	
1. Правовые и организационные вопросы обеспечения безопасности: – специальные (характерные при эксплуатации объекта исследования, проектируемой рабочей зоны) правовые нормы трудового законодательства; – организационные мероприятия при компоновке рабочей зоны.	ГОСТ 12.2.032-78. Рабочее место при выполнении работ сидя. ГОСТ 22269-73. Рабочее место оператора. Взаимное расположение элементов рабочего места. ГОСТ 22269-76. Рабочее место оператора. Взаимное расположение элементов рабочего стола. ГОСТ 21889-76. Система “Человек-машина”. Кресло человека-оператора. ТК РФ. СанПиН 1.2.3685-21. Гигиенические нормативы и требования к обеспечению безопасности и (или) безвредности для человека факторов среды обитания.
2. Производственная безопасность: 2.1. Анализ выявленных вредных и опасных факторов 2.2. Обоснование мероприятий по снижению воздействия	Вредные факторы: -отсутствие или недостаток естественного и искусственного освещения; -зрительное напряжение; -повышенный уровень электромагнитного излучения; -отклонение показателей микроклимата;

	Опасные факторы: -повышенное значение напряжения в электрической цепи;
3. Экологическая безопасность:	Атмосфера: ртутное загрязнение и т.п. Гидросфера: бытовой мусор, ртутное загрязнение. Литосфера: бытовой мусор, ртутное загрязнение.
4. Безопасность в чрезвычайных ситуациях:	Возможные ЧС: пожар, наводнение, землетрясение, удар молнией, взрыв, террористический акт. Наиболее типичная ЧС: пожар.

Дата выдачи задания для раздела по линейному графику	28.02.2021

Задание выдал консультант:

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Ассистент	Черемискина Мария Сергеевна	-		

Задание приняли к исполнению студенты:

Группа	ФИО	Подпись	Дата
8К71	Галлингер Владислав Андреевич		
8К71	Семенюта Антон Вадимович		

Реферат

Выпускная квалификационная работа содержит 188 страниц, 175 рисунков, 23 таблицы, 13 формул, 1 приложение и 26 литературных источников.

Ключевые слова: анализ данных, предобработка данных, Data Mining, веб-приложение, проектирование, разработка, предсказательная модель

Цель работы: исследование методики обработки и анализа данных с помощью инструментов Data Mining, а также создание веб-приложения на основе проведенного исследования.

В первой главе представлено описание предметной области, обзор инструментов Data Mining, описание задачи исследования и задачи проектирования и разработки приложения.

Во второй главе представлен анализ данных о характеристиках студентов, построение и сравнение предсказательных моделей на его основе, а также анализ данных о группах, дисциплинах и преподавателях.

В третьей главе описано проектирование программной системы.

В четвертой главе описан процесс разработки программной системы.

Пятая глава представляет собой выполненное задание по разделу «Финансовый менеджмент, ресурсоэффективность и ресурсосбережение», при выполнении которого были использованы выводы, полученные в процессе анализа, в области проектного и финансового менеджмента, в том числе менеджмента рисков.

Шестая глава представляет собой выполненное задание по разделу «Социальная ответственность», где были рассмотрены аспекты производственной и экологической безопасности, безопасности в чрезвычайных ситуациях, а также правовые вопросы организации труда.

Перечень условных обозначений

API (Application Programming Interface) – описание способов, которыми одна компьютерная программа может взаимодействовать с другой программой.

Big Data – Инструменты и способы обработки больших объемов данных.

Bootstrap – набор инструментов для верстки сайтов и веб-приложений.

CSS (Cascading Style Sheets) – формальный язык описания внешнего представления документа, написанного с использованием языка разметки.

Data Mining – Интеллектуальный анализ данных, совокупность методов обнаружения нетривиальных и практически полезных знаний.

Django – фреймворк для веб-приложений на языке Python.

DRY (don’t repeat yourself) – принцип разработки программного обеспечения, нацеленный на снижение повторения информации различного рода, особенно в системах со множеством слоёв абстрагирования.

Jinja – это шаблонизатор для языка программирования Python. Он подобен шаблонизатору Django, но предоставляет Python-подобные выражения, обеспечивая исполнение шаблонов в песочнице. Это текстовый шаблонизатор, поэтому он может быть использован для создания любого вида разметки, а также исходного кода.

HTML (HyperText Markup Language) – стандартизированный язык разметки документов во Всемирной паутине.

HTTP (HyperText Transfer Protocol) – протокол прикладного уровня передачи данных по сети Интернет.

HTTP-метод GET – запрос, использующийся для получения содержимого указанного веб-ресурса.

HTTP-метод POST – запрос, предназначенный для передачи веб сервером данных, заключённых в тело сообщения, для хранения.

JS (JavaScript) – мультипарадигменный язык программирования, который поддерживает объектно-ориентированный, императивный и функциональный стили.

MVC (Model View Controller) – архитектурный паттерн, реализуемый для разделения данных приложения, пользовательского интерфейса и управляющей логики на три отдельных компонента: модель, представление и контроллер.

Python – высокоуровневый язык программирования общего назначения.

ReLU – Rectified Linear Unit, линейный выпрямитель.

SVM (англ. SVM, support vector machine) – Метод опорных векторов, набор схожих алгоритмов обучения с учителем, использующихся для задач классификации и регрессионного анализа.

URL (Uniform Resource Locator) – система унифицированных адресов электронных ресурсов, или единообразный определитель местонахождения ресурса.

Библиотека – сборник подпрограмм или объектов, используемых для разработки программного обеспечения.

Веб-приложение – клиент-серверное приложение, в котором клиент взаимодействует с сервером при помощи браузера.

Веб-сервер – сервер, принимающий HTTP-запросы от клиентов, обычно веб-браузеров, и выдающий им HTTP-ответы, как правило, вместе с HTML-страницей, изображением, файлом, медиа-потоком или другими данными.

Код состояния HTTP – часть первой строки ответа сервера при запросах по протоколу HTTP.

Фреймворк – программное обеспечение, позволяющее автоматизировать разработку и тестирование программного продукта.

Датасет – От англ. Dataset – набор данных.

Датафрейм (от англ. DataFrame) – объект библиотеки pandas, позволяющий работать с табличными данными.

Стейкхолдер – заинтересованное лицо или организация, долю, права, требования или интересы относительно системы или ее свойств.

L_2-регуляризация – Регуляризация Тихонова или ridge regression. Предназначена для предотвращения переобучения.

ИС – информационная система, предназначена для хранения, поиска и обработки информации.

Оглавление

Введение

Основной задачей, стоящей перед высшим учебным заведением, является подготовка квалифицированных специалистов. Для выполнения поставленной задачи применяются различные способы – это и контроль успеваемости, и мотивирование студентов в виде выплачиваемой стипендии в случае успешного обучения. Также используются методики для проверки качества преподавания. Так, например, в некоторых университетах студенты оценивают работу преподавателей. В качестве примера можно привести Томский политехнический университет.

Одним из способов контроля процесса преподавания и обучения является интеллектуальный анализ данных. С помощью инструментов Data Mining [1] можно выяснить, с чем связана успеваемость тех или иных студентов, какие параметры влияют на средний балл студентов, есть ли связь между средним баллом и определенным преподавателем и т.д. В данной работе рассматривается методика подготовки данных для анализа и построение предсказательных моделей, а также непосредственно анализ характеристик студентов, а именно поиск скрытых зависимостей и выделение признаков, наибольшим образом влияющих на целевую функцию.

Помимо этого, в работе применена методика получения «чистых» данных. Эффективность анализа во многом зависит от качества исходных данных, и поэтому их подготовка является важным шагом в различных областях. Игнорирование данного этапа может негативно сказаться на результатах анализа. Данные, полученные на этапе сбора – «сырые». Они могут содержать пропуски, дубликаты (повторяющиеся строки), значения, выбивающиеся по величине из основного ряда (выбросы), недопустимые значения. Признаки (факторы, объясняющие переменные) могут сильно коррелировать (мультиколлинеарность), или иметь распределение, отличное от нормального.

В ходе работы будут созданы модели, прогнозирующие успеваемость того или иного студента. Для построения предсказательной модели могут быть

использованы различные методы прогнозирования, такие как нейронные сети, деревья решений, алгоритмы градиентного бустинга, метод опорных векторов.

Результаты анализа и методики, исследуемые в этой работе, могут помочь руководству получить более полную картину о качестве преподавания в вузе и принять соответствующие решения. Этим объясняется актуальность данной работы и её практическая ценность.

Объектом исследования являются данные об успеваемости студентов на весенний семестр 2019 года и данные о средней успеваемости групп по определенным дисциплинам у определенных преподавателей на осенний семестр 2019 года.

Предметом исследования являются закономерности в вышеперечисленных данных, а также методика их анализа и обработки.

Кроме того, создание приложения для аналитики и визуализации данных определенной тематики и размерности значительно облегчает практическое использование стейкхолдерами [2].

Цель данной работы – исследование методики обработки и анализа данных с помощью инструментов Data Mining, а также создание веб-приложения на основе проведенного исследования.

Для достижения поставленной цели необходимо выполнить следующие задачи:

1. Подготовка данных для анализа;
2. Выделение наиболее важных параметров, влияющих на успеваемость студентов;
3. Построение портретов слабого и сильного студентов;
4. Построение предсказательных моделей и сравнение показателей метрик, используя очищенные и неочищенные данные;
5. Проектирование ИС по обработки и визуализации данных;
6. Разработка алгоритмического и программного обеспечения анализа и визуализации данных.

Глава 1. Исследование предметной области

1.1. Описание предметной области

Data Mining – собирательное название, используемое для обозначения совокупности методов обнаружения в данных ранее неизвестных, практически полезных знаний, необходимых для принятия решений в различных сферах человеческой деятельности. При передачи «Data Mining» на русском языке используются следующие словосочетания: просев информации, добыча данных, извлечение данных, а также интеллектуальный анализ данных.

Инструменты Data Mining применяются в различных сферах человеческой деятельности. Так, в медицине данные технологии применяются для раннего выявления сердечно-сосудистых, онкологических и нервных заболеваний, туберкулеза. В банковской сфере Data Mining нашел своё применение в системах кредитного скоринга, прогнозирующих вероятности возвращения заемщиком кредита в срок. В нефтегазовой отрасли активно используются технологии Big Data и машинное обучение. Так, благодаря системам прогнозирования и упреждения компании British Petroleum удалось сократить сроки строительства скважин на 30% и уменьшить общую стоимость скважины на 15% [6].

Инструменты Data Mining также используется и в образовании. Так, Университет Пердью в США запустил у себя систему предиктивной аналитики, которая собирает информацию об академической истории студентов, их активности в цифровой учебной среде и демографические данные. На основе этой информации рассчитывается уровень риска отсева для каждого студента. Благодаря такой интерактивной Big Data системе удалось улучшить результаты обучения и снизить показатели отсева [7].

Большинство образовательных учреждений и платформ используют свои модели для повышения качества образования. Они собирают различную информацию об обучающихся для обработки и прогнозирования различных

показателей: степени вовлеченности студента в учебный процесс, успеваемости, посещаемости, числа взятых книг из библиотеки и других. Благодаря сведениям, полученным из проанализированных данных, создаются рекомендательные системы, которые помогают студентам выбрать образовательные курсы и записаться на них [8].

Проводились и исследования в данной области. Так, ученые Пражского экономического университета провели аналитику данных о собственных студентах [9]. Удалось выявить факторы, влияющие наибольшим образом на успешное окончание учебы. Среди них оказались такие, как процент потерянных кредитных ваучеров в последнем семестре и временной разрыв между средним и высшим образованиями.

Исследователи Флоридского университета успешно использовали байесовскую сетевую модель в прогнозировании доли выполнения задач студентами [10]. В будущем авторы планируют внедрить инструмент в систему персонализированной связи преподавателей и студентов.

Институт Вейцмана в Израиле подчеркнул, что адекватная предварительная обработка образовательных данных чрезвычайно важна в процессе их анализа. Она может предотвратить серьезные неточности в результатах исследования и значительно повысить достоверность и надежность выводов [11].

Также исследователи Левенского университета обратили внимание на использование методов анализа данных для улучшения процесса обучения студентов и формирования адаптивного и дифференцированного обучения в онлайн-классах. Аналитику данных можно использовать для предсказания результата обучения студентов, обнаружения мошенничества с их стороны и измерения производительности преподавателей [12].

Таким образом, аналитика данных в сфере образования весьма актуальна на данный момент и является сферой интересов множества исследователей.

1.2. Описание методов исследования данных с помощью Data Mining

Основа методов Data Mining – всевозможные методы классификации, моделирования и прогнозирования, основанные на применении деревьев решений, искусственных нейронных сетей, эволюционного программирования, генетических алгоритмов, ассоциативной памяти, нечеткой логики. Также к методам Data Mining относят статистические методы (корреляционный и регрессионный анализ, факторный анализ, дисперсионный анализ и т.д.) [1].

В данной работе используются следующие методы предиктивной аналитики:

- Линейная регрессия;
- Логистическая регрессия;
- Метод k–ближайших соседей;
- Случайный лес;
- Метод опорных векторов;
- Полносвязная нейронная сеть.

Помимо предсказательных моделей, в данной работе также используются корреляционный анализ и некоторые методы дисперсионного анализа.

1.2.1. Линейная регрессия

Линейная регрессия – используемая в статистике регрессионная модель зависимости одной переменной y от другой или нескольких других переменных x с линейной функцией зависимости.

Регрессия – это способ выбрать из семейства функций ту, которая минимизирует функцию потерь. Последняя характеризует насколько сильно пробная функция отклоняется от значений в заданных точках.

Семейство функций, из которых производится выбор, представляет собой линейную комбинацию наперед заданных базисных функций f_i (формула 1).

$$f = \sum_i w_i f_i \quad (1)$$

где f_i – заданная базисная функция;

w_i – коэффициент перед базисной функцией.

1.2.2. Логистическая регрессия

Логистическая регрессия – статическая модель, используемая для прогнозирования вероятности возникновения некоторого события путем его сравнения с логистической кривой.

Данный классификатор позволяет оценивать апостериорные вероятности принадлежности объектов к классам. Делается предположение о том, что вероятность наступления события y = 1 равна (формулы 2-3):

$$P = \{y = 1|x\} = f(z) \quad (2)$$

$$f(z) = \frac{1}{1+e^{-z}} \quad (3)$$

где $z = \Theta^T x$, x – матрица значений независимых переменных, Θ – матрица значений коэффициентов регрессии.

1.2.3. Метод k–ближайших соседей

Метод k–ближайших соседей – метрический алгоритм для автоматической классификации объектов или регрессии.

В случае использования методы для классификации объект присваивается тому классу, который является наиболее распространенным среди k соседей данной элемента, классы которых уже известны. В случае использования метода для регрессии, объекту присваивается среднее значение по k ближайшим к нему объектам, значения которых уже известны.

На рисунке 1 представлена схема работы метода.

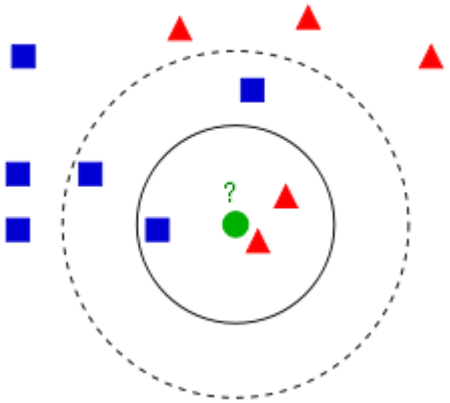


Рисунок 1. Схема работы алгоритма k–ближайших соседей

1.2.4. Случайный лес

Случайный лес – алгоритм машинного обучения, заключающийся в использовании ансамбля решающих деревьев. Алгоритм сочетает в себе две основные идеи: метод бэггинга Бреймана и метод случайных подпространств.

Алгоритм применяется для задач классификации, регрессии и кластеризации. Основная идея заключается в использовании большого ансамбля решающих деревьев, каждое из которых само по себе дает невысокое качество классификации, но итоговый результат получается удовлетворительным за счет их большого количества.

На рисунке 2 представлена схема работы метода.

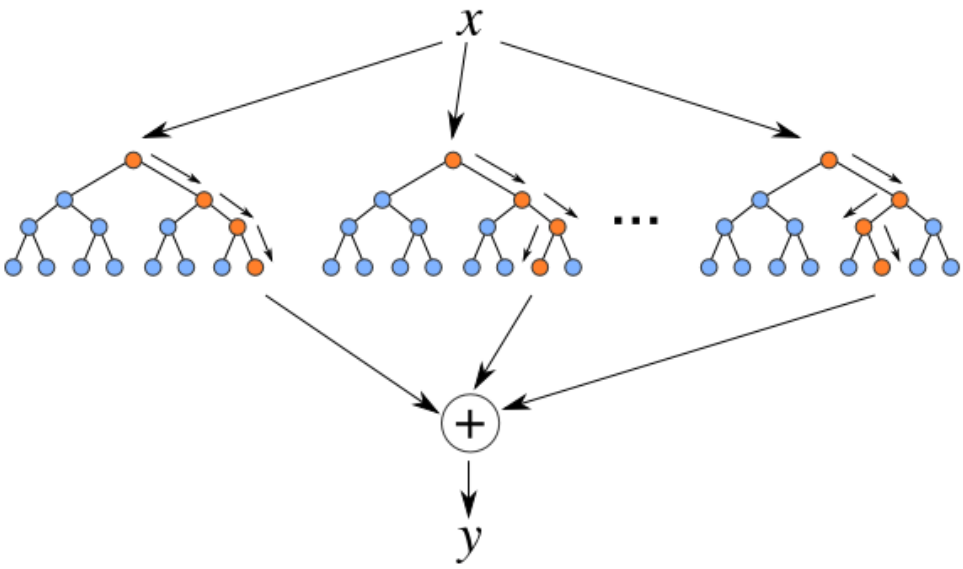


Рисунок 2. Схема работы случайного леса

1.2.5. Метод опорных векторов

Метод опорных векторов – семейство схожих алгоритмов обучения с учителем, использующихся для задач классификации и регрессионного анализа. Его особым свойством является непрерывное уменьшение эмпирической ошибки классификации и увеличение зазора.

Основная идея – перевод исходных векторов в пространство более высокой размерности и поиск разделяющей гиперплоскости с наибольшим зазором в этом пространстве. Две параллельных гиперплоскости строятся по обеим сторонам гиперплоскости, разделяющей классы. Разделяющей

гиперплоскостью будет гиперплоскость, создающая наибольшее расстояние до двух параллельных гиперплоскостей.

На рисунке 3 представлена схема работы метода.

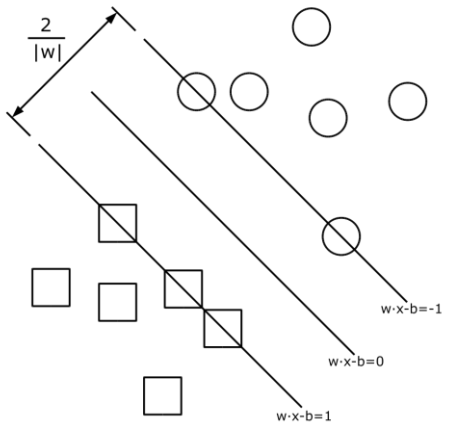


Рисунок 3. Схема работы метода опорных векторов

1.2.6. Полносвязная нейронная сеть

Нейронная сеть – математическая модель, построенная по принципу организации и функционирования сетей нервных клеток живого организма.

Искусственная нейронная сеть представляет собой систему соединенных и взаимодействующих между собой процессоров (нейронов). На вход нейрону поступает множество сигналов, каждый из которых является выходом другого нейрона. Каждый вход умножается на соответствующий вес, и все произведения суммируются, определяя уровень активации нейрона. Далее сигнал, как правило, преобразуется активационной функцией и дает выходной нейронный сигнал.

На рисунке 4 представлена схема работы искусственного нейрона.

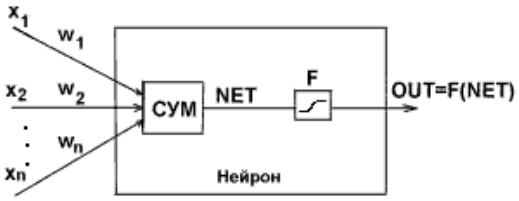


Рисунок 4. Схема работы искусственного нейрона

1.2.7. Корреляционный анализ

Корреляционный анализ – метод обработки статистических данных, заключающийся в изучении коэффициентов корреляции между переменными. При этом сравниваются коэффициенты корреляции между одной парой или множеством пар признаков для установления между ними статистических взаимосвязей. Корреляционный анализ – метод по изучению статистической зависимости между случайными величинами с необязательным наличием строгого функционального характера, при которой динамика одной случайной величины приводит к динамике математического ожидания другой.

Основными задачами описываемого метода являются:

- Получить информацию об одной из искомых переменных с помощью другой;
- Определить тесноту связи между исследуемыми переменными.

Также метод предполагает определение зависимости между изучаемыми признаками, в связи с чем задачи корреляционного анализа можно дополнить следующими:

- Выявление факторов, оказывающих наибольшее влияние на целевой признак;
- Выявление неизученных ранее причин связей;
- Построение корреляционной модели с ее параметрическим анализом;
- Исследование значимости параметров связи и их интервальная оценка.

Взаимозависимые факторы с коэффициентом парной корреляции более 0,75 в корреляционную модель предпочтительно не включать, как и такие, у которых связь с результативным параметром носит не прямолинейный или функциональный характер.

Традиционно данные корреляционного анализа представляются в виде корреляционной матрицы. Заголовками строк и столбцов являются

обрабатываемые переменные, а на их пересечении выводится коэффициент корреляции для соответствующей пары признаков.

1.2.8. Дисперсионный анализ

Дисперсионный анализ – метод в математической статистике, направленный на поиск зависимостей в экспериментальных данных путем исследования значимости различий в средних группой. Его суть сводится к изучению влияния одной или нескольких независимых переменных на зависимую.

Простейшим случаем дисперсионного анализа является одномерный однофакторный анализ для двух или нескольких независимых групп, когда все группы объединены по одному признаку. В ходе анализа проверяется нулевая гипотеза о равенстве средних. При анализе двух групп дисперсионный анализ тождественен двухвыборочному t-критерию Стьюдента для независимых выборок.

В то время, как для сравнения двух выборок, распределенных нормально, используются z- и t-критерии Стьюдента, при работе с выборками, распределенными ненормально используются критерий знаковых рангов (только для связанных выборок), критерий Манна-Уитни и перестановочный критерий.

1.3. Постановка задачи

1.3.1. Задача исследования

Требуется произвести подготовку данных для проведения разведочного анализа данных, нахождения скрытых зависимостей и решения задачи классификации студентов по их успеваемости. Исходные данные к работе будут взяты из двух датасетов:

1. Данные об успеваемости студентов Томского Политехнического Университета на весенний семестр 2019 года. Данные содержат 24 переменных и 8551 наблюдение, представляющие собой характеристики конкретного студента;

2. Данные о средней успеваемости групп по определенным дисциплинам у определенных преподавателей на осенний семестр 2019 года. Датасет содержит 22 признака и 5988 записей, представляющих собой характеристики пар группа–дисциплина.

В таблице 1 приведены переменные, характеризующие студентов.

Таблица 1. Список переменных первого датасета

Имя переменной	**Расшифровка**	**Тип переменной**
EDUCATION_FORM	Форма обучения	Строка
QUALIFICATION	Квалификация	Строка
EDUCATION_YEAR	Курс	Целый
SPECIALITY	Специальность	Строка
PROFILE	Профиль	Строка
GRADUATION_DEPARTMENT	Выпускное отделение	Строка
GRADUATION_SCHOOL	Выпускная школа	Строка
GROUP	Группа	Строка
EDUCATIONAL_DIVISION	Обучающее подразделение	Строка
FINANCIAL_FORM	Форма финансирования	Строка
COUNTRY	Страна	Строка
CITIZENSHIP	Гражданство	Строка
GENDER	Пол	Строка
BIRTHDAY	Дата рождения	Дата
ACADEMIC_LEAVE	Состоит ли в академическом отпуске	Строка
TOTAL_MARK_COUNT	Всего оценок в семестре	Целый
POSITIVE_MARK_COUNT	Положительных оценок	Целый
NEGATIVE_MARK_COUNT	Неудовлетворительных оценок	Целый
NEGATIVE_DISCIPLINES	Дисциплины по которым получены неудовлетворительные оценки	Строка

Продолжение таблицы 1.

Имя переменной	Расшифровка	Тип переменной
NEG_DISCIPLINE_MISSES	Пропусков по дисциплинам по которым получены неудовлетворительные оценки	Строка
NEG_DISCIPLINE_HOURS	Всего часов по дисциплинам по которым получены неудовлетворительные оценки часов пропусков в семестре	Целый
ALL_MISSES_HOURS	Всего часов пропусков в семестре	Целый
ALL_HOURS	Всего часов аудиторных занятий в семестре	Целый
INDEX	Индекс	Целый

В таблице 2 приведены переменные, характеризующие пары группа–дисциплина-преподаватель.

Таблица 2. Список переменных второго датасета

Имя переменной	Расшифровка	Тип переменной
EDUCATION_FORM	Форма обучения	Строка
QUALIFICATION	Квалификация	Строка
EDUCATION_YEAR	Курс	Целый
SPECIALITY	Специальность	Строка
GRADUATION_DIVISION	Выпускное подразделение	Строка
GRADUATION_SCHOOL	Выпускная школа	Строка
DISCIPLINE	Дисциплина	Строка
CERTIFICATION_TYPE	Вид аттестации	Строка
EDUCATIONAL_DIVISION	Обеспечивающее подразделение	Строка
GROUP	Группа	Строка
EDUCATIONAL_SCHOOL	Обеспечивающая школа	Строка

Продолжение таблицы 2.

Имя переменной	**Расшифровка**	**Тип переменной**
KT2_TEACHER_INDEX	Индексы преподавателей, выставляющих КТ2	Строка
TOTAL_TEACHER_INDEX	Индексы преподавателей, выставляющих итоговую оценку	Строка
LECTURERS_INDEX	Лекторы	Строка
TEACHERS_INDEX	Преподаватели	Строка
TEACHER_DIVISION	Подразделение преподавателя	Строка
TOTAL_MARK_COUNT	Всего выставлено оценок	Целый
EXCELLENT_MARK_COUNT	Всего отличных оценок	Целый
GOOD_MARK_COUNT	Всего хороших оценок	Целый
SATISFACTORY_MARK_COUNT	Всего удовлетворительных оценок	Целый
PASSED_STUDENT_COUNT	Количество сдавших дисциплину студентов	Целый
NOT_PASSED_STUDENT_COUNT	Количество не сдавших дисциплину студентов	Целый

Для построения предсказательной модели в качестве целевой переменной для задачи регрессии была выбрана «Успешность». Это искусственная переменная, построенная на параметрах POSITIVE_MARK_COUNT и TOTAL_MARK_COUNT. Зависимость описана формулой 4.

$$\text{Успешность} = \frac{\text{Количество сданных дисциплин}}{\text{Количество дисциплин всего}} \quad (4)$$

На основе параметра «Успешность» был сформирован признак «Класс» для задачи классификации студента. Класс «0» описывает студентов, чья успешность меньше или равна 0,25, класс «1» – от 0,25 до 0,75, не включая граничные значения, класс «2» – от 0,75 до 1.

1.3.2. Задача проектирования и разработки приложения

На базе проведенного исследования требуется спроектировать и разработать веб-приложение по аналитике и обработке данных о характеристиках студентов.

Данное веб-приложение будут использовать аналитики Томского политехнического университета. Следовательно, приложение должно предоставлять возможности для загрузки данных о студентах любого семестра в определенном формате, автоматически производить обработку и составлять отчет.

К итоговому продукту прилагаются требования, которые описывают его эксплуатационные свойства:

Функциональные требования (Ф):

Ф.1. В приложении должна быть предусмотрена загрузка датасета об успеваемости студентов в формате, приведенном в таблице 1;

Ф.2. В приложении должны быть реализованы следующие функции обработки датасета:

Ф.2.1. Замена столбцов, содержащих персональные данные студента, на индексы;

Ф.2.2. Заполнение миссингов на медиану;

Ф.2.3. Удаление строк (наблюдений), содержащих миссинги;

Ф.2.4. Замена значений выбросов на медиану;

Ф.2.5. Удаление строк (наблюдений), содержащих выбросы;

Ф.2.6. Устранение линейных зависимостей;

Ф.2.7. Составление целевых переменных;

Ф.2.8. Устранение факультативных дисциплин;

Ф.2.9. Удаление неинформативных признаков:

Ф.2.8.1. В ручном режиме;

Ф.2.8.2. В автоматическом режиме.

Ф.3. Визуализация статистики о данных;

Ф.4. Генерация отчета по предобработанному датасету:

Ф.4.1. Статистическая сводка (среднее, медиана, стандартное отклонение, максимальное и минимальное значения, Q1 и Q3 процентили);

Ф.4.2. Гистограммы распределения всех признаков;

Ф.4.3. Диаграмма размаха для числовых признаков;

Ф.4.4. Формирование и проверка гипотез:

Ф.4.4.1. Категориальный график;

Ф.4.4.2. Q-Q график;

Ф.4.3.3. Критерий Шапиро–Уилка;

Ф.4.3.4. В случае нормальности распределения должна выполняться проверка гипотез по критерию Стьюдента;

Ф.4.3.5. При ненормальности распределения проверка по критериям Манна–Уитни и перестановочному;

Ф.4.3.6. Категориальный график успешности.

Ф.5. Сохранение отчета в PDF–формате;

Ф.6. Использование предсказательной модели:

Ф.6.1. Обучение модели с нуля;

Ф.6.2. Предсказание успеваемости на основе введенных признаков.

Ф.6.3. Сохранение модели.

Глава 2. Интеллектуальный анализ данных об успеваемости студентов

2.1. Анализ данных о характеристиках студентов

2.1.1. Предобработка «сырых» данных

На вход был получен датасет, состоящий из 8551 студента (строк) и 24 колонок-характеристик обучающихся (рисунок 5).

```
RangeIndex: 8551 entries, 0 to 8550
Data columns (total 24 columns):
Форма обучения                                                          8551 non-null object
Квалификация                                                            8551 non-null object
Курс                                                                    8551 non-null int64
Специальность                                                           8551 non-null object
Профиль                                                                 6676 non-null object
Выпуск. отдел.                                                          8551 non-null object
Выпуск. школа                                                           7988 non-null object
Группа                                                                  8551 non-null object
Обуч. подразд.                                                          8551 non-null object
Форма финансирования                                                    8551 non-null object
Страна                                                                  8533 non-null object
Гражданство                                                             8551 non-null object
Пол                                                                     8551 non-null object
Дата рождения                                                           8551 non-null object
Академ отпуск (действующий) - да / нет                                  8551 non-null object
Всего                                                                   8551 non-null int64
Положительных                                                           8551 non-null int64
Неудовлетворительных                                                    8551 non-null int64
Дисциплины по которым получены неудовлетворительные оценки              6370 non-null object
Пропусков по дисциплинам по которым получены неудовлетворительные оценки    8551 non-null int64
Всего часов по дисциплинам по которым получены неудовлетворительные оценки  8539 non-null float64
Всего часов пропусков в семестре                                        8551 non-null int64
Всего часов аудиторных занятий в семестре                               8470 non-null float64
Индекс                                                                  8551 non-null int64
dtypes: float64(2), int64(7), object(15)
memory usage: 1.6+ MB
```

Рисунок 5. Краткая информационная сводка о взятом датасете

Ниже перечислены следующие характеристики:

1. «Форма обучения» – Форма обучения студента (Очная, заочная, очно-заочная);
2. «Квалификация» – Степень, которая будет присвоена студенту (бакалавр, магистр, специалист);
3. «Курс» – Курс, на котором студент обучается (1-5);
4. «Специальность» – Номер и название специальности студента;

5. «Профиль» – Название профиля студента;
6. «Выпуск. отдел.» – Выпускающее студента отделение;
7. «Выпуск. школа» – Выпускающая студента школа;
8. «Группа» – Группа, в которой обучается студент;
9. «Обуч. подразд.» – Обучающее подразделение студента;
10. «Форма финансирования» – Форма финансирования студента (на основе бюджетного финансирования, на договорной основе, по целевому приёму);
11. «Страна» – Страна постоянной прописки студента;
12. «Гражданство» – Гражданство студента;
13. «Пол» – Пол студента (мужской, женский);
14. «Дата рождения» – Дата рождения студента (формат DD/MM/YYYY);
15. «Академ отпуск (действующий) - да / нет» – Находится ли студент в академическом отпуске (Да, Нет);
16. «Всего» – Число дисциплин студента в прошлом семестре;
17. «Положительных» – Число сданных дисциплин у студента в прошлом семестре;
18. «Неудовлетворительных» – Число несданных дисциплин у студента в прошлом семестре;
19. «Дисциплины по которым получены неудовлетворительные оценки» – Перечень несданных дисциплин;
20. «Пропусков по дисциплинам по которым получены неудовлетворительные оценки» – Количество пропусков студента по несданным дисциплинам;
21. «Всего часов по дисциплинам, по которым получены неудовлетворительные оценки» – Количество часов по несданным дисциплина в учебном плане студента;
22. «Всего часов пропусков в семестре» – Количество часов пропусков студента в прошлом семестре;

23.«Всего часов аудиторных занятий в семестре» – Количество часов аудиторных занятий студента в прошлом семестре;

24.«Индекс» – Индекс студента.

Данные взяты на конец сессии весеннего семестра 2019 года.

На рисунке 6 представлены основные статистики числовых переменных датасета.

	Курс	Всего	Положительных	Неудовлетворительных	Пропусков по дисциплинам по которым получены неудовлетворительные оценки	Всего часов по дисциплинам по которым получены неудовлетворительные оценки	Всего часов пропусков в семестре	Всего часов аудиторных занятий в семестре
count	8551.000000	8551.000000	8551.000000	8551.000000	8551.000000	8539.000000	8551.000000	8470.000000
mean	2.335984	9.348263	5.745176	3.603087	12.468015	378.272924	26.132967	733.658796
std	1.199401	2.409408	3.715429	3.479882	37.211507	522.897072	43.844702	488.883365
min	1.000000	3.000000	0.000000	0.000000	0.000000	0.000000	0.000000	108.000000
25%	1.000000	8.000000	2.000000	0.000000	0.000000	0.000000	0.000000	400.000000
50%	2.000000	9.000000	7.000000	3.000000	0.000000	144.000000	8.000000	496.000000
75%	3.000000	10.000000	9.000000	6.000000	0.000000	476.000000	34.000000	1036.000000
max	5.000000	18.000000	15.000000	18.000000	446.000000	2976.000000	446.000000	2976.000000

Рисунок 6. Основные статистики числовых переменных датасета

На рисунках 7-9 представлены первые 5 студентов набора данных. По ним можно составить представление, как датасет выглядит в целом. Представленные ниже данные – результат функции Pandas.DataFrame.head().

	Форма обучения	Квалификация	Курс	Специальность	Профиль	Выпуск. отдел.	Выпуск. школа
0	Очная	Специалист	5	18.05.02 Химическая технология материалов совр...	Химическая технология материалов ядерно-топлив...	Отделение ядерно-топливного цикла	Инженерная школа ядерных технологий
1	Очная	Специалист	5	18.05.02 Химическая технология материалов совр...	Химическая технология материалов ядерно-топлив...	Отделение ядерно-топливного цикла	Инженерная школа ядерных технологий
2	Очная	Специалист	5	18.05.02 Химическая технология материалов совр...	Химическая технология материалов ядерно-топлив...	Отделение ядерно-топливного цикла	Инженерная школа ядерных технологий
3	Очная	Специалист	5	18.05.02 Химическая технология материалов совр...	Химическая технология материалов ядерно-топлив...	Отделение ядерно-топливного цикла	Инженерная школа ядерных технологий
4	Очная	Специалист	5	18.05.02 Химическая технология материалов совр...	Химическая технология материалов ядерно-топлив...	Отделение ядерно-топливного цикла	Инженерная школа ядерных технологий

Рисунок 7. Первые пять студентов набора данных (часть 1)

Группа	Обуч. подразд.	Форма финансирования	...	Академ отпуск (действующий) - да / нет	Всего	Положительных	Неудовлетворительных
0441	Инженерная школа ядерных технологий	на основе бюджетного финансирования	...	Нет	9	9	0
0441	Инженерная школа ядерных технологий	на договорной основе	...	Нет	9	0	9
0441	Инженерная школа ядерных технологий	на основе бюджетного финансирования	...	Нет	9	9	0
0441	Инженерная школа ядерных технологий	на основе бюджетного финансирования	...	Нет	9	9	0
0441	Инженерная школа ядерных технологий	на основе бюджетного финансирования	...	Нет	9	9	0

Рисунок 8. Первые пять студентов набора данных (часть 2)

Дисциплины по которым получены неудовлетворительные оценки	Пропусков по дисциплинам по которым получены неудовлетворительные оценки	Всего часов по дисциплинам по которым получены неудовлетворительные оценки	Всего часов пропусков в семестре	Всего часов аудиторных занятий в семестре	Индекс
NaN	0	0.0	14	408.0	0
Лабораторный практикум по гидрометаллургически...	110	408.0	110	408.0	1
NaN	0	0.0	12	408.0	2
NaN	0	0.0	14	408.0	3
NaN	0	0.0	4	408.0	4

Рисунок 9. Первые пять студентов набора данных (часть 3)

Рассмотрим распределение студентов по количеству неудовлетворительных оценок (рисунок 10). Видим, что данное распределение ненормально. Студентов, не имеющих неудовлетворительных оценок, большинство. С возрастанием количества не сданных дисциплин количество студентов убывает экспоненциально.

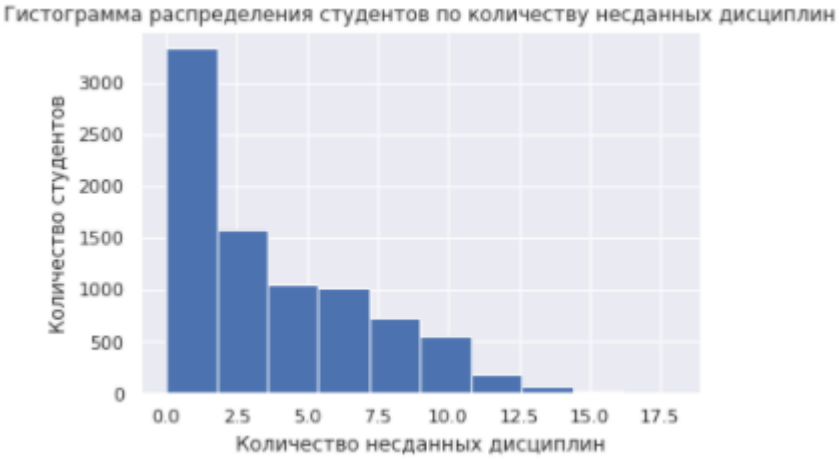


Рисунок 10. Распределение студентов по количеству неудовлетворительных оценок

Далее требуется работа с данными столбца «Дисциплины по которым получены неудовлетворительные оценки». При изучении датасета стало понятно, что некоторые из несданных дисциплин – факультативные предметы, которые отображаются у каждого члена группы, если хотя бы один из студентов на них записался (рисунок 11). При этом у оставшихся студентов такие предметы идут в данный столбец. Было решено, что если удалить все факультативные дисциплины из датасета, данные станут нагляднее и проще для анализа.

```
nan
Лабораторный практикум по гидрометаллургическим технологиям(Зач.), Основы проектирования химических производств(Зач.), Основы проектирования химических производств(КП), Радиохимическая переработк
Системы управления химико-технологическими процессами(КП)
Лабораторный практикум по гидрометаллургическим технологиям(Зач.), Основы проектирования химических производств(Зач.), Основы проектирования химических производств(КП), Системы управления химико-
Химическая технология редких и благородных металлов(Экз.)
Основы проектирования химических производств(КП)
Прикладная физическая культура(Зач.)
Введение в теорию ядерных реакторов(Зач.), Методы аналитического контроля в производстве материалов современной энергетики(Экз.), Оборудование производств редких элементов(Экз.), Профессиональная
Введение в теорию ядерных реакторов(Зач.), Методы аналитического контроля в производстве материалов современной энергетики(Экз.), Оборудование производств редких элементов(Экз.), Прикладная физич
Введение в теорию ядерных реакторов(Зач.), Методы аналитического контроля в производстве материалов современной энергетики(Экз.), Методы получения чистых веществ(Экз.), Оборудование производств р
Профессиональная подготовка на английском языке(Зач.)
Профессиональная подготовка на английском языке(Зач.), Учебно-исследовательская работа студентов(Зач.)
Введение в теорию ядерных реакторов(Зач.), Прикладная физическая культура(Зач.), Профессиональная подготовка на английском языке(Зач.)
Прикладная физическая культура(Зач.), Профессиональная подготовка на английском языке(Зач.)
Введение в теорию ядерных реакторов(Зач.), Профессиональная подготовка на английском языке(Зач.), Учебно-исследовательская работа студентов(Зач.)
Учебно-исследовательская работа студентов(Зач.)
Материаловедение(Зач.), Прикладная физическая культура(Зач.), Прикладная химическая термодинамика(Экз.), Профессиональная подготовка на английском языке(Зач.), Химические реакторы(Зач.), Химия ре
Материаловедение(Зач.), Учебно-исследовательская работа студентов(Зач.), Химия редких элементов(Экз.)
Материаловедение(Зач.), Химия редких элементов(Экз.)
Химия редких элементов(Экз.)
Материаловедение(Зач.), Прикладная физическая культура(Зач.), Прикладная химическая термодинамика(Экз.), Профессиональная подготовка на английском языке(Зач.), Учебно-исследовательская работа сту
Материаловедение(Зач.), Прикладная физическая культура(Зач.), Прикладная химическая термодинамика(Экз.), Профессиональная подготовка на английском языке(Зач.), Учебно-исследовательская работа сту
Прикладная физическая культура(Зач.), Прикладная химическая термодинамика(Экз.), Химия редких элементов(Экз.)
Прикладная химическая термодинамика(Экз.), Химия редких элементов(Экз.)
Материаловедение(Зач.), Общая химическая технология(Экз.), Прикладная физическая культура(Зач.), Прикладная химическая термодинамика(Экз.), Профессиональная подготовка на английском языке(Зач.),
```

Рисунок 11. Начало списка уникальных наборов несданных дисциплин

На рисунке 12 представлен список регулярных выражений, с помощью которых все факультативы удаляются из рассмотрения. Для поиска и замены факультативных дисциплин в списке используется библиотека re (библиотека для работы с регулярными выражениями в Python).

```
Второй иностранный язык \(немецкий\). A2.1\(Зач.\),[.]?
Второй иностранный язык \(немецкий\). A2.1\(Зач.\)[.]?
Второй иностранный язык \(немецкий\). A1.1\(Зач.\),[.]?
Второй иностранный язык \(немецкий\). A1.1\(Зач.\)[.]?
Второй иностранный язык \(китайский\). 1\(Зач.\),[.]?
Второй иностранный язык \(китайский\). 1\(Зач.\)[.]?
Второй иностранный язык \(французский\). A1.1\(Зач.\),[.]?
Второй иностранный язык \(французский\). A1.1\(Зач.\)[.]?
Иностранный язык для программ академической мобильности \(английский\). A2.2\(Зач.\),[.]?
Иностранный язык для программ академической мобильности \(английский\). A2.2\(Зач.\)[.]?
Управление проектами\(Зач.\),[.]?
Управление проектами\(Зач.\)[.]?
Факультативные дисциплины по выбору студента\(Зач.\),[.]?
Факультативные дисциплины по выбору студента\(Зач.\)[.]?
Креативность инженера\(Зач.\),[.]?
Креативность инженера\(Зач.\)[.]?
```

Рисунок 12. Список регулярных выражений для удаления факультативов

Результат применения регулярных выражений представлен на рисунке 13. Кроме того, была произведена замена значения nan на категориальную строку «Нет» и убраны лишние пробелы в начале и в конце каждой строки и дублирование пробелов между словами.

```
Нет
Лабораторный практикум по гидрометаллургическим технологиям(Зач.), Основы проектирования химических производств(Зач.), Основы проектирования химических производств(КП), Радиохимическая переработ
Системы управления химико-технологическими процессами(КП)
Лабораторный практикум по гидрометаллургическим технологиям(Зач.), Основы проектирования химических производств(Зач.), Основы проектирования химических производств(КП), Системы управления химико
Химическая технология редких и благородных металлов(Экз.)
Основы проектирования химических производств(КП)
Прикладная физическая культура(Зач.)
Введение в теорию ядерных реакторов(Зач.), Методы аналитического контроля в производстве материалов современной энергетики(Экз.), Оборудование производств редких элементов(Экз.), Профессиональна
Введение в теорию ядерных реакторов(Зач.), Методы аналитического контроля в производстве материалов современной энергетики(Экз.), Оборудование производств редких элементов(Экз.), Прикладная физи
Введение в теорию ядерных реакторов(Зач.), Методы аналитического контроля в производстве материалов современной энергетики(Экз.), Методы получения чистых веществ(Экз.), Оборудование производств
Профессиональная подготовка на английском языке(Зач.)
Профессиональная подготовка на английском языке(Зач.), Учебно-исследовательская работа студентов(Зач.)
Введение в теорию ядерных реакторов(Зач.), Прикладная физическая культура(Зач.), Профессиональная подготовка на английском языке(Зач.)
Прикладная физическая культура(Зач.), Профессиональная подготовка на английском языке(Зач.)
Введение в теорию ядерных реакторов(Зач.), Профессиональная подготовка на английском языке(Зач.), Учебно-исследовательская работа студентов(Зач.)
Учебно-исследовательская работа студентов(Зач.)
Материаловедение(Зач.), Прикладная физическая культура(Зач.), Прикладная химическая термодинамика(Экз.), Профессиональная подготовка на английском языке(Зач.), Химические реакторы(Зач.), Химия р
Материаловедение(Зач.), Учебно-исследовательская работа студентов(Зач.), Химия редких элементов(Экз.)
Материаловедение(Зач.), Химия редких элементов(Экз.)
Химия редких элементов(Экз.)
Материаловедение(Зач.), Прикладная физическая культура(Зач.), Прикладная химическая термодинамика(Экз.), Профессиональная подготовка на английском языке(Зач.), Учебно-исследовательская работа ст
Материаловедение(Зач.), Прикладная физическая культура(Зач.), Прикладная химическая термодинамика(Экз.), Профессиональная подготовка на английском языке(Зач.), Учебно-исследовательская работа ст
Прикладная физическая культура(Зач.), Прикладная химическая термодинамика(Экз.), Химия редких элементов(Экз.)
Прикладная химическая термодинамика(Экз.), Химия редких элементов(Экз.)
Материаловедение(Зач.), Общая химическая технология(Экз.), Прикладная физическая культура(Зач.), Прикладная химическая термодинамика(Экз.), Профессиональная подготовка на английском языке(Зач.),
Материаловедение(Зач.), Профессиональная подготовка на английском языке(Зач.), Учебно-исследовательская работа студентов(Зач.), Химия редких элементов(Экз.)
Иностранный язык (английский)(Экз.),
Аналитическая химия(Зач.),
```

Рисунок 13. Начало списка уникальных наборов несданных дисциплин (обработанного)

Также во всех категориальных переменных заменено значение nan на «Нет», а в числовых – на медиану всех значений признака.

После выполнения всех этапов предобработки данных необходимо выделить целевую переменную. Для данного датасета было принято решение

создать числовую переменную «Успешность», зависящую от столбцов «Всего» и «Положительных». Расчет успешности производится по формуле 4.

На рисунке 14 представлена вычисленная успешность для первых пяти студентов в списке.

ісего	Положительных	Неудовлетворительных	Дисциплины по которым получены неудовлетворительные оценки	Пропусков по дисциплинам по которым получены неудовлетворительные оценки	Всего часов по дисциплинам по которым получены неудовлетворительные оценки	Всего часов пропусков в семестре	Всего часов аудиторных занятий в семестре	Индекс студента	Год рождения	Успешность
9	9	0	Нет	0	0.0	14	408.0	0	1996	1.0
9	0	9	Лабораторный практикум по гидрометаллургически...	110	408.0	110	408.0	1	1995	0.0
9	9	0	Нет	0	0.0	12	408.0	2	1996	1.0
9	9	0	Нет	0	0.0	14	408.0	3	1996	1.0
9	9	0	Нет	0	0.0	4	408.0	4	1995	1.0

Рисунок 14. Вычисленный столбец «Успешность»

На рисунке 15 представлена гистограмма распределения студентов по их успешности. Видно, что распределение ненормально, имеется повышение плотности по краям и резкое снижение в центре распределения.

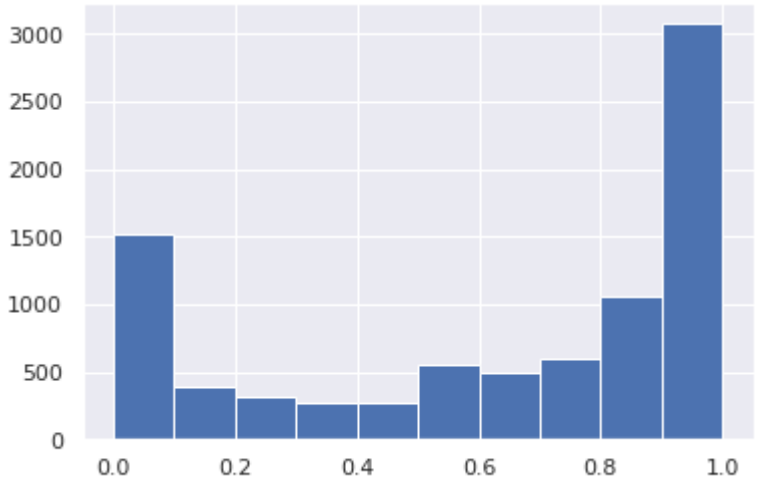


Рисунок 15. Гистограмма распределения студентов по их успешности

Далее все студенты разбиваются на три класса. Функция, классифицирующая студентов по их успеваемости, представлена на рисунке 16.

```
def classify(success):
  if success >= 0.75:
    return 2
  elif success > 0.25:
    return 1
  else:
    return 0
```

Рисунок 16. Функция расчета значений признака «Класс»

Следующим шагом производится «очистка» датасета от признаков, не представляющих интереса для дальнейшего его анализа, создания и обучения модели.

Следующие столбцы были удалены из рассмотрения:

- «Всего» – Избыточный столбец, использовавшийся для построения признака «Успешность»;
- «Положительных» – Избыточный столбец, использовавшийся для построения признака «Успешность»;
- «Неудовлетворительных» – Избыточный столбец, зависящий от столбцов «Всего» и «Положительных»;
- «Группа» – Малоинформативный признак, большой разброс успешности у студентов каждой группы, появляются новые категории признака каждый год;
- «Страна» – Признак, практически полностью дублирующий столбец «Гражданство»;
- «Дисциплина по которым получены неудовлетворительные оценки» – Признак, который нет возможности использовать для предсказания, так как на начало семестра его значения неизвестны;
- «Индекс студента» – Малоинформативный признак;
- «Год рождения» – Признак, сильно коррелирующий со столбцом «Курс»;
- «Выпуск. школа» – Признак, сильно коррелирующий со столбцом «Выпуск. отдел.»;

- «Всего часов по дисциплинам по которым получены неудовлетворительные оценки» – Признак, который нет возможности использовать для предсказания, так как на начало семестра его значения неизвестны;
- «Пропусков по дисциплинам по которым получены неудовлетворительные оценки» – Признак, который нет возможности использовать для предсказания, так как на начало семестра его значения неизвестны;
- «Всего часов пропусков в семестре» – Признак, который нет возможности использовать для предсказания, так как на начало семестра его значения неизвестны;
- «Всего аудиторных занятий в семестре» – Линейно коррелирует с одним из признаков. В дальнейшем рассмотрим детальнее.

Итоговый набор данных представлен на рисунке 17.

	Форма обучения	Квалификация	Курс	Профиль	Выпуск. отдел.	Обуч. подразд.	Форма финансирования	Гражданство	Пол	Академ отпуск (действующий) - да / нет	Успешность	Класс
0	Очная	Специалист	5	Химическая технология материалов ядерно-топлив...	Отделение ядерно-топливного цикла	Инженерная школа ядерных технологий	на основе бюджетного финансирования	Российская Федерация	Мужской	Нет	1.0	2
1	Очная	Специалист	5	Химическая технология материалов ядерно-топлив...	Отделение ядерно-топливного цикла	Инженерная школа ядерных технологий	на договорной основе	Российская Федерация	Мужской	Нет	0.0	0
2	Очная	Специалист	5	Химическая технология материалов ядерно-топлив...	Отделение ядерно-топливного цикла	Инженерная школа ядерных технологий	на основе бюджетного финансирования	Российская Федерация	Женский	Нет	1.0	2
3	Очная	Специалист	5	Химическая технология материалов ядерно-топлив...	Отделение ядерно-топливного цикла	Инженерная школа ядерных технологий	на основе бюджетного финансирования	Российская Федерация	Мужской	Нет	1.0	2
4	Очная	Специалист	5	Химическая технология материалов ядерно-топлив...	Отделение ядерно-топливного цикла	Инженерная школа ядерных технологий	на основе бюджетного финансирования	Республика Казахстан	Женский	Нет	1.0	2

Рисунок 17. Новый набор данных

На гистограмме распределения студентов по их классам (рисунок 18) видно, что большинство студентов относятся ко второму классу, а к первому относится чуть меньше, чем к нулевому.

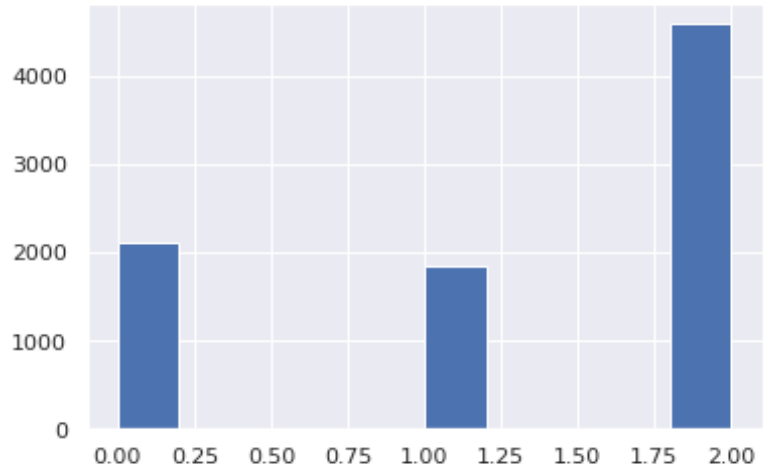


Рисунок 18. Гистограмма распределения студентов по классам

Последний этап предобработки – рассмотрение таблицы корреляции столбцов данного датасета (рисунок 19). Пусть признаки линейно зависимы, если коэффициент корреляции Спирмена больше или равен 0,75 по модулю. Для категориальных переменных использовалась функция get_dummies, которая строит на их основе новые бинарные признаки.

Прежде всего, имеются линейные зависимости между целевыми признаками «Класс» и «Успешность», что подчеркивает построение нами одного признака на основе другого. Следующая зависимость – между признаками «Форма Обучения_Очная» и «Всего аудиторных занятий в семестре». Следовательно, придется удалить признак «Всего аудиторных занятий в семестре». Как видно по таблице, других значений больших или равных 0,75 по модулю нет, следовательно, линейных зависимостей в исходном датасете не осталось.

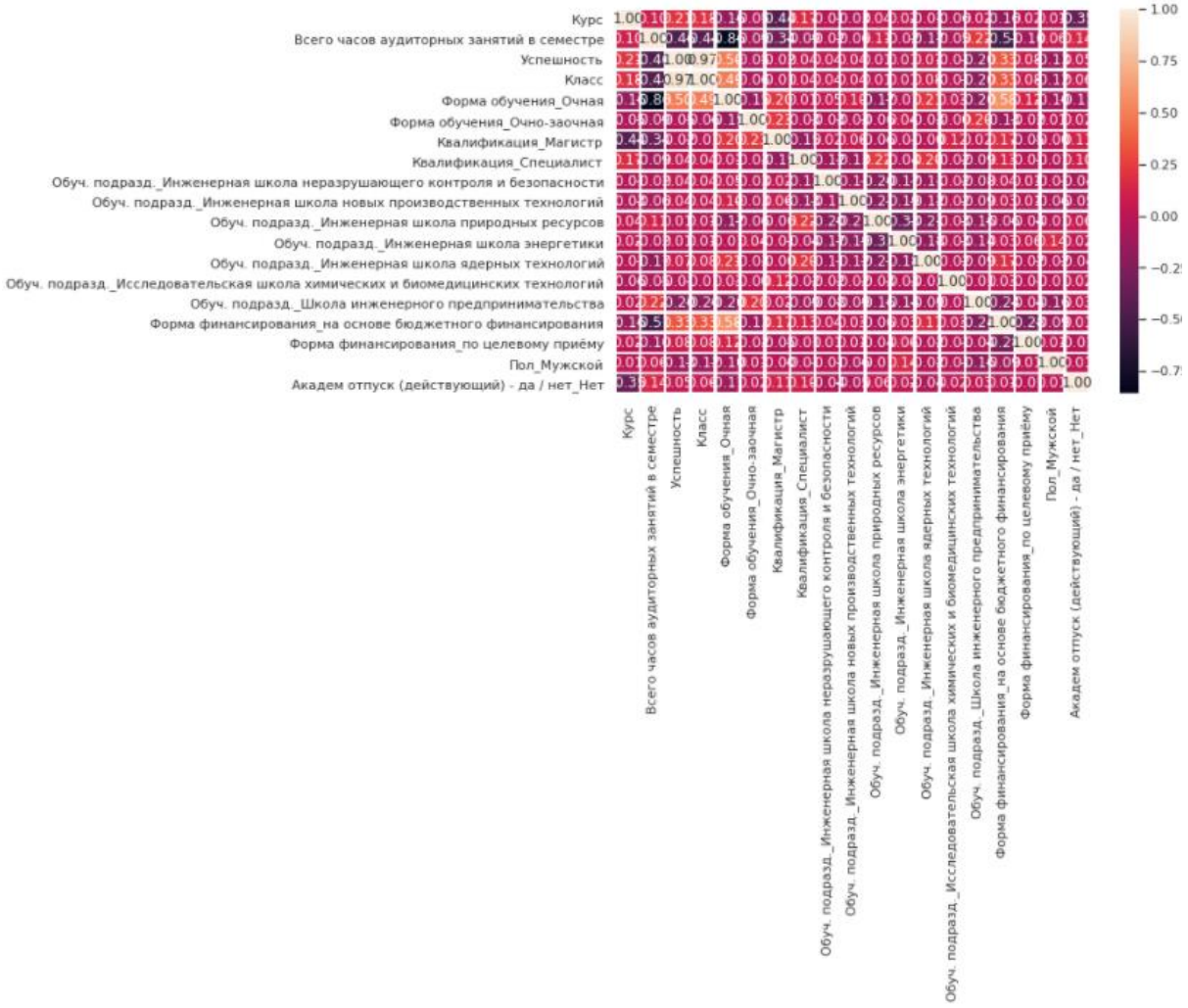


Рисунок 19. Таблица корреляции столбцов датасета

Таким образом, были убраны из рассмотрения факультативные дисциплины, удалены неинформативные признаки из датасета, устранены линейные зависимости и сформированы два новых целевых признака – «Успешность» и «Класс».

2.1.2. Разведочный анализ

На рисунке 20 представлен словарь частот не сданных дисциплин (академических задолженностей). Заметно, что студентов, не имеющих академических задолженностей – большинство.

```
{'Иностранный язык (английский)(Зач.)': 643,
 'Иностранный язык (английский)(Экз.)': 666,
 'Математика 2(ДЗ)': 488,
 'Нет': 2461,
 'Прикладная физическая культура(Зач.)': 950,
 'Профессиональная подготовка на английском языке(Зач.)': 668,
 'Профессиональный иностранный язык (английский)(Зач.)': 524,
 'Творческий проект(Зач.)': 585,
 'Учебно-исследовательская работа студентов(Зач.)': 1873,
 'Физика 1(ДЗ)': 515}
```

Рисунок 20. Словарь частот не сданных дисциплин

Самая часто встречающаяся не сданная дисциплина – «Учебно-исследовательская работа студентов». У данного предмета немного строгих сроков сдачи, возможно, поэтому студенты недооценивают этот предмет.

Также можно допустить, что студенты несерьезно относятся к предметам «Прикладная физическая культура» и «Иностранный язык (английский)», считая, что следует уделить внимание более сложным дисциплинам.

Также в десятку самых частых несданных дисциплин входят «Физика» и «Математика». Что неудивительно, они являются одними из самых сложных общих дисциплин в любом учебном плане.

На рисунке 21 представлена гистограмма распределения студентов по несданным дисциплинам.

Рисунок 21. Гистограмма распределения студентов по несданным дисциплинам

На рисунке 22 представлен словарь частот гражданств. На нем видно, что студентов из Российской Федерации большинство, но также присутствует много студентов из Республики Казахстан и Республики Узбекистан.

```
{'Арабская Республика Египет': 28,
 'Киргизская Республика': 75,
 'Китайская Народная Республика': 220,
 'Монголия': 29,
 'Республика Казахстан': 1379,
 'Республика Таджикистан': 40,
 'Республика Узбекистан': 454,
 'Российская Федерация': 6193,
 'Социалистическая Республика Вьетнам': 53,
 'Туркменистан': 22}
```

Рисунок 22. Словарь частот гражданств

На рисунке 23 представлена гистограмма распределения студентов по гражданству.

Рисунок 23. Гистограмма распределения студентов по несданным дисциплинам

На рисунке 24 представлен словарь частот значений признака «Год рождения». На конец весны 2019 года среди студентов большинство родилось с 1997-го по 2000-ый год.

```
1997: 1322
1998: 1217
1996: 1196
1999: 1069
2000: 1013
1995: 668
1994: 379
1993: 256
1992: 203
2001: 188
```

Рисунок 24. Словарь частот значений признака «Год рождения»

На рисунке 25 представлена гистограмма распределения студентов по годам рождения.

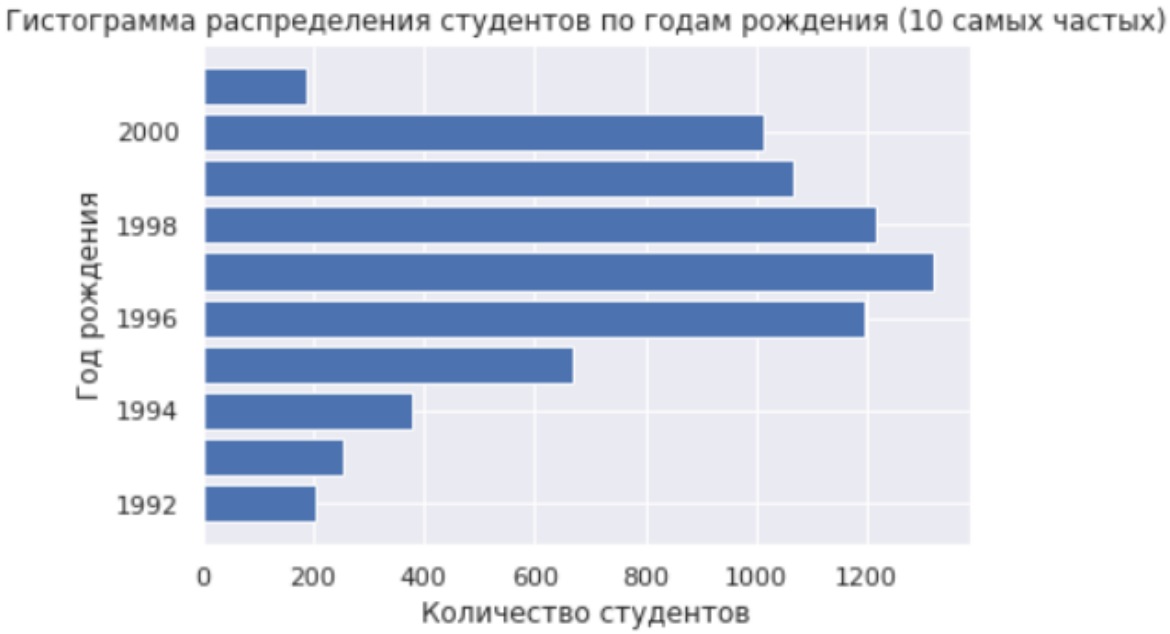


Рисунок 25. Гистограмма распределения студентов по годам рождения

На рисунке 26 изображен словарь частот различных форм обучения. Заметно, что большинство студентов обучаются очно. Очно-заочных студентов меньшинство.

```
Очная: 6119
Заочная: 2360
Очно-заочная: 72
```

Рисунок 26. Словарь частот форм обучения

На рисунке 27 представлена гистограмма распределения студентов по формам обучения.

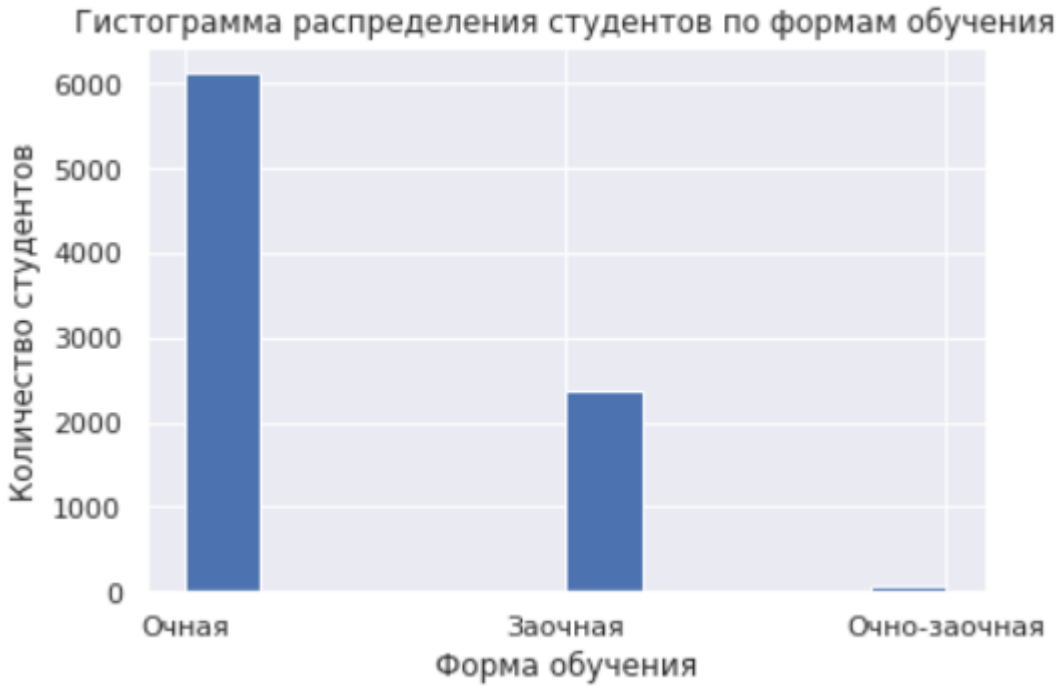


Рисунок 27. Гистограмма распределения студентов по формам обучения

На рисунке 28 представлен словарь частот различных квалификаций, присваиваемых студентам после окончания обучения. Обучающихся в бакалавриате большинство, обучающихся в магистратуре и специалитете – примерно равное количество.

```
Специалист: 1006
Бакалавр: 6328
Магистр: 1217
```

Рисунок 28. Словарь частот форм обучения

На рисунке 29 представлена гистограмма распределения студентов по квалификациям.

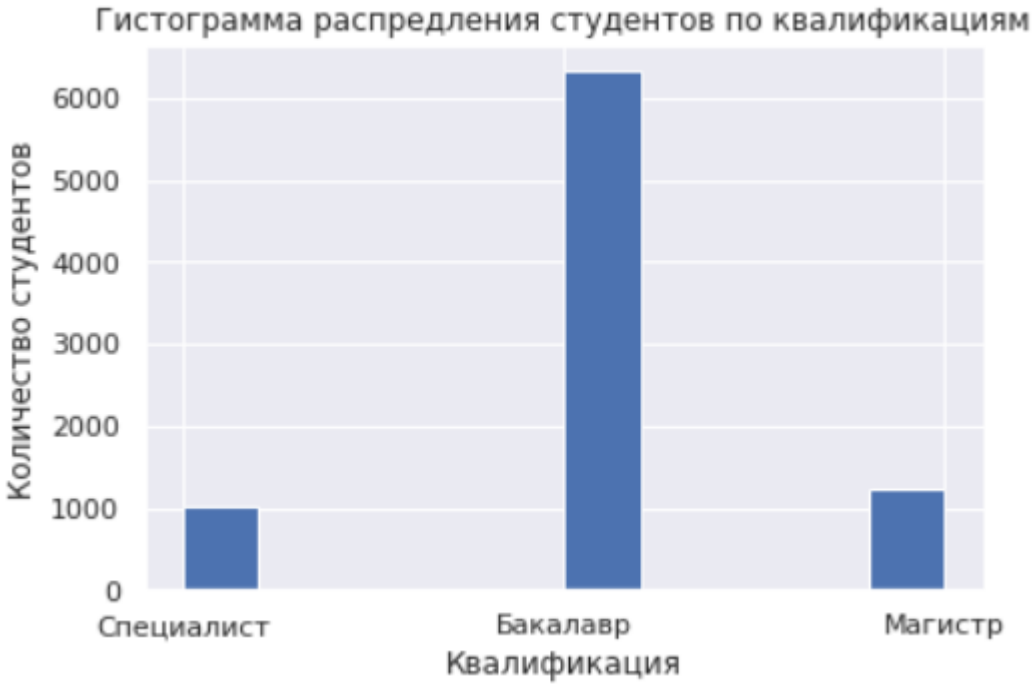


Рисунок 29. Гистограмма распределения студентов по квалификациям

Рассмотрим бакалавриат подробнее. На рисунке 30 представлен словарь частот значений поля «Курс» для студентов бакалаврита. Подавляющее меньшинство студентов обучаются на пятом курсе, на остальных курсах обучающихся примерно равное количество.

```
1: 1595
2: 1552
3: 1649
4: 1508
5: 24
```

Рисунок 30. Словарь частот форм обучения

На рисунке 31 представлена гистограмма распределения студентов бакалавриата по курсам обучения.

Рисунок 31. Гистограмма распределения студентов бакалавриата по курсам обучения

На рисунке 32 представлен словарь частот форм финансирования студентов. Заметно, что большинство обучается на основе бюджетного финансирования. Меньшинство – по целевому приему.

```
На основе бюджетного финансирования: 5752
На договорной основе: 2483
По целевому приему: 316
```

Рисунок 32. Словарь частот форм финансирования

На рисунке 33 представлена гистограмма распределения студентов по формам финансирования.

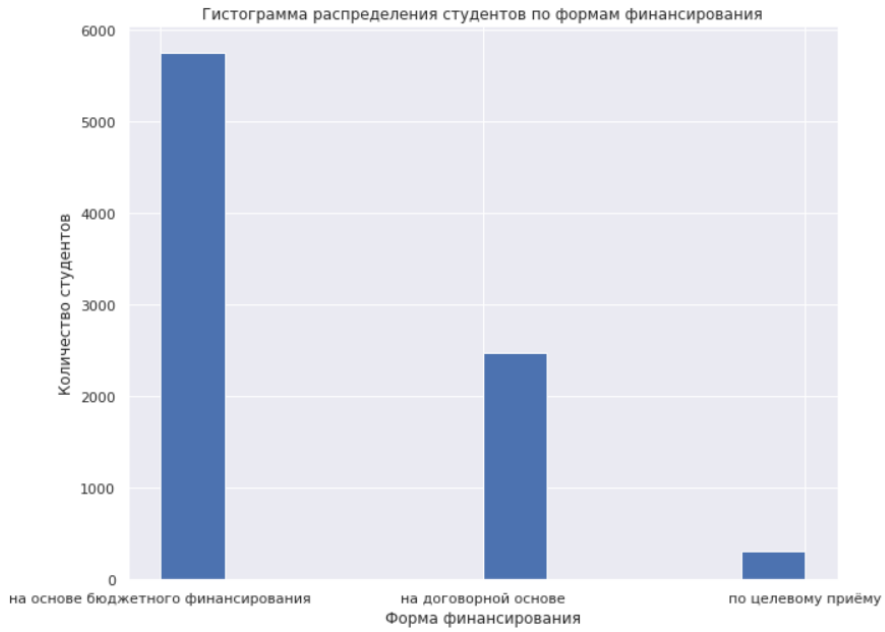


Рисунок 33. Гистограмма распределения студентов по формам финансирования

На рисунке 34 представлен словарь частот пола студентов. Большинство студентов являются мужчинами (студентов-мужчин примерно в 2,4 раза больше, чем студентов женщин).

```
Мужской: 6050
Женский: 2501
```

Рисунок 34. Словарь частот пола студентов

На рисунке 35 представлена гистограмма распределения студентов по полу.

Рисунок 35. Гистограмма распределения студентов по формам финансирования

На рисунке 36 представлен словарь частот инженерных школ, в которых обучаются студенты. Больше всего студентов обучаются в Инженерной школе природных ресурсов и Инженерной школе энергетики.

```
Инженерная школа ядерных технологий: 877
Инженерная школа неразрушающего контроля и безопасности: 799
Инженерная школа новых производственных технологий: 906
Инженерная школа информационных технологий и робототехники: 1120
Инженерная школа природных ресурсов: 2350
Школа инженерного предпринимательства: 542
Инженерная школа энергетики: 1936
Исследовательская школа химических и биомедицинских технологий: 21
```

Рисунок 36. Словарь частот инженерных школ

На рисунке 37 представлена гистограмма распределения студентов по инженерным школам.

Рисунок 37. Гистограмма распределения студентов по инженерным школам

На рисунке 38 представлен словарь частот значений признака «Академ отпуск (действующий) - да / нет». Большинство студентов на конец весны 2019 года в академическом отпуске не находились.

```
Нет: 7249
Да: 1302
```

Рисунок 38. Словарь частот значений признака «Академ отпуск (действующий) – да / нет»

На рисунке 39 представлена гистограмма распределения студентов по значениям признака «Академ отпуск (действующий) - да / нет».

Рисунок 39. Гистограмма распределения студентов по значениям признака «Академ отпуск (действующий) – да / нет»

На рисунке 40 представлена диаграмма размаха неудовлетворительных оценок по дисциплинам. Среднее арифметическое количество несданных

дисциплин равно 3,28, а медиана равна двум (рисунок 41). Среднеквадратичное отклонение же равно 3,36 несданных дисциплин.

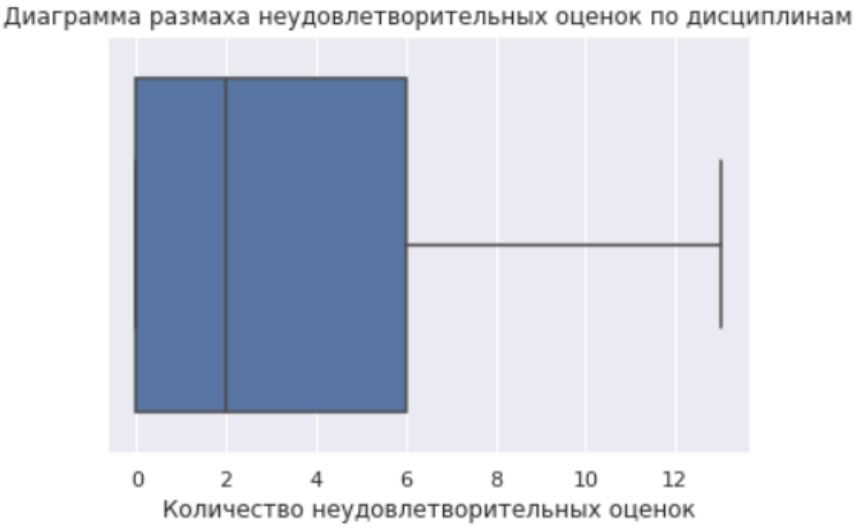


Рисунок 40. Диаграмма размаха неудовлетворительных оценок по дисциплинам

```
count    8551.000000
mean        3.281604
std         3.357490
min         0.000000
25%         0.000000
50%         2.000000
75%         6.000000
max        13.000000
Name: Неудовлетворительных, dtype: float64
```

Рисунок 41. Статистическая сводка о несданных дисциплинах

2.1.3. Формулирование и проверка статистических гипотез

Следующим этапом интеллектуального анализа данных является исследование датасета и формулирование статистических гипотез.

Сперва было изучено влияние пола на класс, к которому студент относится. На рисунке 42 представлена таблица сопряженности класса студента и его пола. Заметно, что в каждом из классов мужчин больше, чем женщин, однако во втором классе процентное отношение женщин резко возрастает. Возможно это повлияет на прогнозирование успешности студента.

Класс	0	1	2	All
Пол				
Женский	477	416	1608	2501
Мужской	1637	1429	2984	6050
All	2114	1845	4592	8551

Рисунок 42. Таблица сопряженности класса студента и его пола

На рисунке 43 представлены доли представителей второго класса среди мужчин и женщин соответственно. Видим, что доля представителей второго класса среди женщин больше.

```
% of successful in
Пол = Мужской :  0.4932231404958678
Пол = Женский :  0.6429428228708517
```

Рисунок 43. Доли представителей второго класса среди мужчин и женщин

На рисунках 44-45 приведены доверительные интервалы для показателей успешности среди мужчин и женщин соответственно. Интервалы не пересекаются, что дает основание сформулировать первую статистическую гипотезу: «Средняя успешность студентов-женщин значимо выше, чем успешность студентов-мужчин».

```
(0.5849802320578172, 0.6043283519201883)
```

Рисунок 44. Доверительный интервал для доли представителей второго класса среди мужчин

```
(0.6878211459167991, 0.7159942066190652)
```

Рисунок 45. Доверительный интервал для доли представителей второго класса среди мужчин

Далее было проверено, являются ли распределения успешности мужчин и женщин нормальными. На рисунке 46 приведены соответствующие Q-Q графики. На графиках видно, что распределения далеки от нормальных.

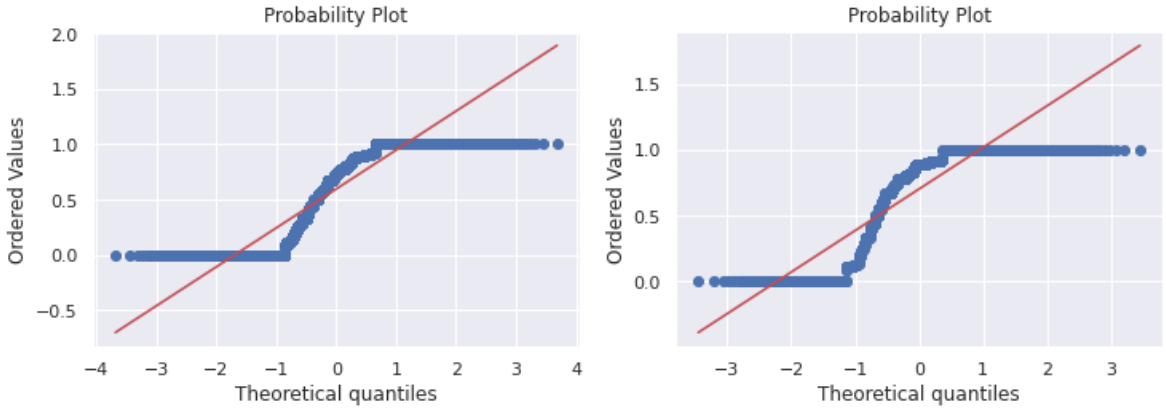


Рисунок 46. Q-Q графики для распределения успешности мужчин и женщин

Для большей точности нормальность распределений проверена критерием Шапиро-Уилка. При взятом уровне значимости α = 0,05 p-значение и в том и в другом случаях равны нулю, следовательно, нулевые гипотезы нормальности распределения отвергаются (рисунки 47-48).

```
Shapiro-Wilk normality test, W-statistic: 0.837279, p-value: 0.000000
```

Рисунок 47. Достигаемый уровень значимости по критерию Шапиро-Уилка для студентов мужского пола

```
Shapiro-Wilk normality test, W-statistic: 0.770120, p-value: 0.000000
```

Рисунок 48. Достигаемый уровень значимости по критерию Шапиро-Уилка для студентов женского пола

В таком случае невозможно определить вид распределения, поэтому были использованы непараметрические критерии. Для сравнения двух независимых выборок используется критерий Манна-Уитни и перестановочный критерий.

На рисунке 49 представлен достигаемый уровень значимости по критерию Манна Уитни. Он меньше α = 0,05, следовательно, нулевая гипотеза отклоняется в пользу двухсторонней альтернативной о том, что распределение одной выборки имеет сдвиг относительно другой.

```
MannwhitneyuResult(statistic=6273620.5, pvalue=7.268159604703943e-37)
```

Рисунок 49. Достигаемый уровень значимости по критерию Манна-Уитни

Для подсчета p-значения перестановочного критерия была создана функция permutation_test (рисунки 50-51).

```
def permutation_t_stat_ind(sample1, sample2):
    return np.mean(sample1) - np.mean(sample2)

def get_random_combinations(n1, n2, max_combinations):
    index = list(range(n1 + n2))
    indices = set([tuple(index)])
    for i in range(max_combinations - 1):
        np.random.shuffle(index)
        indices.add(tuple(index))
    return [(index[:n1], index[n1:]) for index in indices]

def permutation_zero_dist_ind(sample1, sample2, max_combinations = None):
    joined_sample = np.hstack((sample1, sample2))
    n1 = len(sample1)
    n = len(joined_sample)

    if max_combinations:
        indices = get_random_combinations(n1, len(sample2), max_combinations)
    else:
        indices = [(list(index), filter(lambda i: i not in index, range(n))) \
                    for index in itertools.combinations(range(n), n1)]

    distr = [joined_sample[list(i[0])].mean() - joined_sample[list(i[1])].mean() \
             for i in indices]
    return distr
```

Рисунок 50. Создание функции для подсчета p-значения перестановочного критерия (часть 1)

```
def permutation_test(sample, mean, max_permutations = None, alternative = 'two-sided'):
    if alternative not in ('two-sided', 'less', 'greater'):
        raise ValueError("alternative not recognized\n"
                         "should be 'two-sided', 'less' or 'greater'")

    t_stat = permutation_t_stat_ind(sample, mean)

    zero_distr = permutation_zero_dist_ind(sample, mean, max_permutations)

    if alternative == 'two-sided':
        return sum([1. if abs(x) >= abs(t_stat) else 0. for x in zero_distr]) / len(zero_distr)

    if alternative == 'less':
        return sum([1. if x <= t_stat else 0. for x in zero_distr]) / len(zero_distr)

    if alternative == 'greater':
        return sum([1. if x >= t_stat else 0. for x in zero_distr]) / len(zero_distr)
```

Рисунок 51. Создание функции для подсчета p-значения перестановочного критерия (часть 2)

На рисунке 52 представлен расчет p-значения перестановочного критерия. Как можно увидеть, для 10000 перестановок p-значение равно нулю, поэтому нулевая гипотеза отвергается (рисунок 52).

```
print ("p-value: %f" % permutation_test(students_df[students_df.Пол == 'Мужской'].Успешность, students_df[students_df.Пол == 'Женский'].Успешность, max_permutations = 10000))
```

```
p-value: 0.000000
```

Рисунок 52. Подсчет p-значения перестановочного критерия

При увеличении числа перестановок до 50000 p-значение также равно нулю (рисунок 53).

```
print ("p-value: %f" % permutation_test(students_df[students_df.Пол == 'Мужской'].Успешность, students_df[students_df.Пол == 'Женский'].Успешность, max_permutations = 50000))
```

```
p-value: 0.000000
```

Рисунок 53. Подсчет p-значения перестановочного критерия

Также на категориальном графике успешности студентов мужского и женского пола (рисунок 54) видно, что среднее значение успешности студентов женского пола выше, чем мужского.

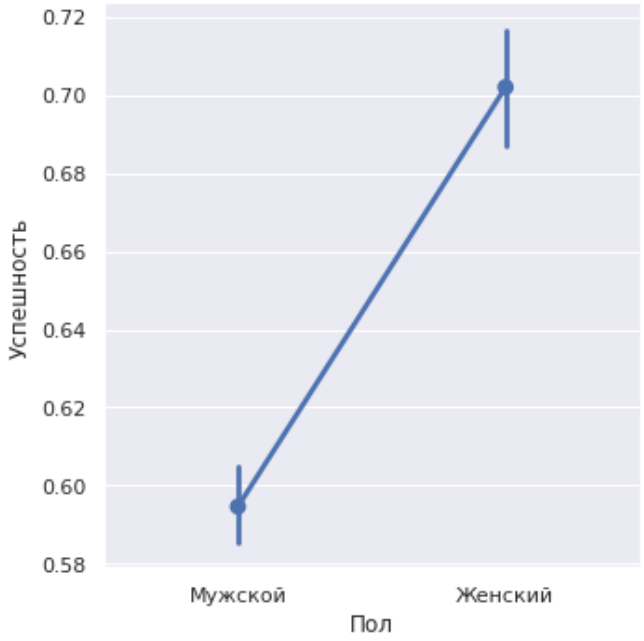


Рисунок 54. Категориальный график успешности студентов мужского и женского пола

Таким образом, с определенной долей уверенности можно сделать вывод, что студенты женского пола в среднем действительно имеют меньше несданных дисциплин.

На рисунке 55 представлена таблица сопряженности класса студента, его пола и обучающего подразделения. В каждой инженерной школе студентов-мужчин больше, чем женщин. Также было обнаружено, что больше всего

студентов обучается в Инженерной школе природных ресурсов и Инженерной школе энергетики.

Обуч. подразд.	Пол / Класс	Женский	Мужской	All
Инженерная школа информационных технологий и робототехники	0	35	240	275
	1	77	216	293
	2	164	388	552
Инженерная школа неразрушающего контроля и безопасности	0	28	159	187
	1	25	75	100
	2	194	318	512
Инженерная школа новых производственных технологий	0	45	134	179
	1	54	150	204
	2	237	286	523
Инженерная школа природных ресурсов	0	97	428	525
	1	129	466	595
	2	481	749	1230
Инженерная школа энергетики	0	55	439	494
	1	60	306	366
	2	217	859	1076
Инженерная школа ядерных технологий	0	25	102	127
	1	40	152	192
	2	217	341	558
Исследовательская школа химических и биомедицинских технологий	0	1	3	4
	1	4	8	12
	2	3	2	5
Школа инженерного предпринимательства	0	191	132	323
	1	27	56	83
	2	95	41	136
All		2501	6050	8551

Рисунок 55. Таблица сопряженности класса студента, его пола и обучающего подразделения

На категориальном графике успешности студентов разных полов и инженерных школ видно, что наиболее высокую успешность имеют студенты Инженерной школы ядерных технологий и Инженерной школы неразрушающего контроля и безопасности (рисунок 56). Исходя из вышеназванных наблюдений были сформулированы две гипотезы: «Средняя успешность студентов ИШНКБ выше, чем остальных (без учета студентов ИШЯТ)» и «Средняя успешность студентов ИШЯТ выше, чем остальных (без учета студентов ИШНКБ)».

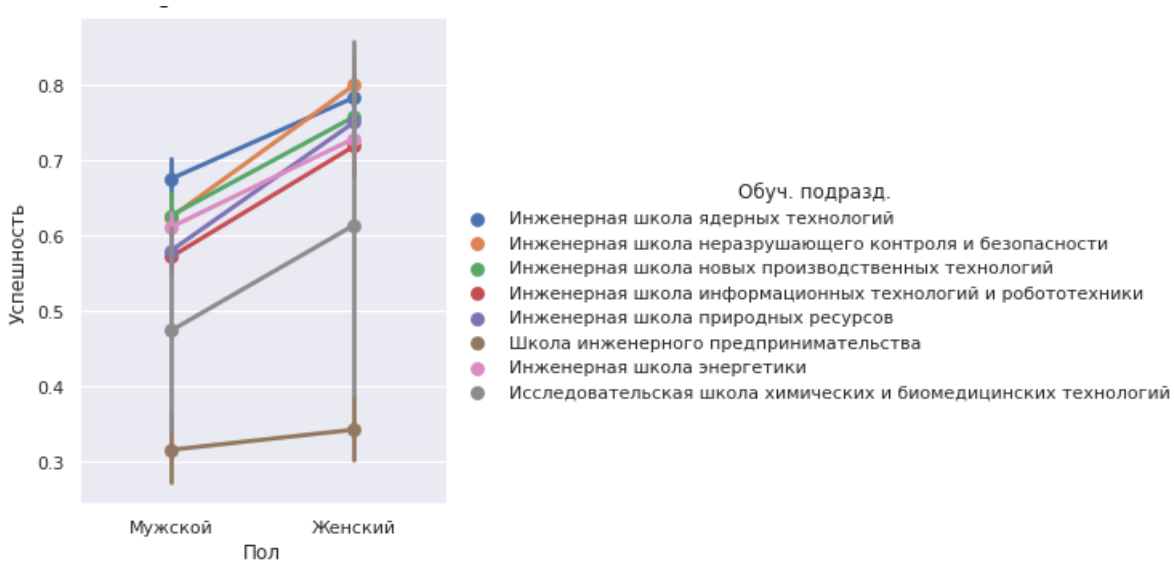


Рисунок 56. Категориальный график успешности студентов разных полов и инженерных школ

На рисунках 57-58 видны результаты применения критерия Манна-Уитни к обеим выборкам. P-значения в обоих случаях во много раз меньше α, поэтому нулевые гипотезы отклонены в пользу альтернативных (распределение одной выборки имеет сдвиг относительно другой).

```
MannwhitneyuResult(statistic=2686812.5, pvalue=4.977142733373969e-08)
```

Рисунок 57. P-значения для критерия Манна-Уитни для выборок студентов ИШЯТ и остальных (без студентов ИШНКБ)

```
MannwhitneyuResult(statistic=2436644.0, pvalue=5.428562559455192e-08)
```

Рисунок 58. P-значения для критерия Манна-Уитни для выборок студентов ИШНКБ и остальных (без студентов ИШЯТ)

Перестановочный критерий для обоих случаев показывает одинаковое значение p-значения, равное 0,0001, что гораздо меньше принятого уровня значимости (рисунки 59–60).

```
print ("p-value: %f" % permutation_test(students_df[students_df['Обуч. подразд.'] == 'Инженерная школа ядерных технологий'].Успешность, \
                                         students_df[(students_df['Обуч. подразд.'] != 'Инженерная школа ядерных технологий') & \
                                (students_df['Обуч. подразд.'] != 'Инженерная школа неразрушающего контроля и безопасности')].Успешность, max_permutations = 10000))
```

```
p-value: 0.000100
```

Рисунок 59. P-значения для перестановочного критерия для выборок студентов ИШНКБ и остальных (без студентов ИШЯТ)

```
print ("p-value: %f" % permutation_test(students_df[students_df['Обуч. подразд.'] == 'Инженерная школа неразрушающего контроля и безопасности'].Успешность, \
                                         students_df[(students_df['Обуч. подразд.'] != 'Инженерная школа ядерных технологий') & \
                                        (students_df['Обуч. подразд.'] != 'Инженерная школа неразрушающего контроля и безопасности')].Успешность, max_permutations = 10000))
```

```
p-value: 0.000100
```

Рисунок 60. P-значения для перестановочного критерия для выборок студентов ИШНКБ и остальных (без студентов ИШЯТ)

Следовательно, обе альтернативные гипотезы приняты.

Далее студенты сравниваются по форме финансирования. На рисунке 61 представлены диаграммы классов студентов для каждой формы финансирования. По трем диаграммам видно, что для студентов, обучающихся на основе бюджетного финансирования и по целевому приему, преобладает второй класс, а для студентов, обучающихся на договорной основе, – нулевой.

Рисунок 61. Диаграммы классов студентов для каждой формы финансирования

На категориальных графиках успешности студентов разных полов, инженерных школ и форм обучения видно, что наиболее высокую успешность имеют студенты, обучающиеся по целевому приему, что необычно (в основном лидируют обучающиеся на бюджетной основе) (рисунки 62-69). Гипотеза: «Средняя успешность студентов, обучающихся по целевому приему выше, чем остальных».

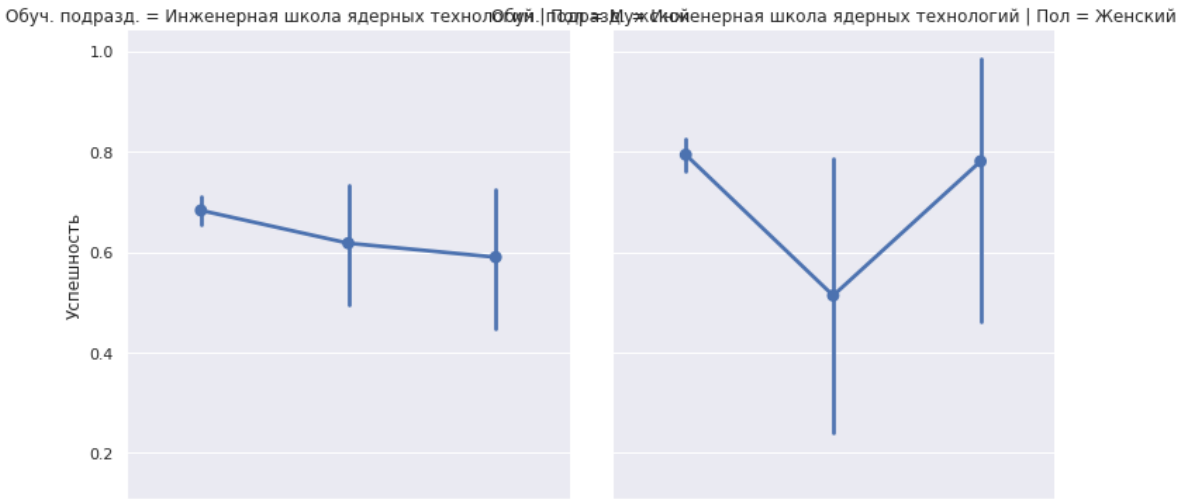


Рисунок 62. Категориальные графики успешности студентов разных полов, инженерных школ и форм обучения (часть 1)

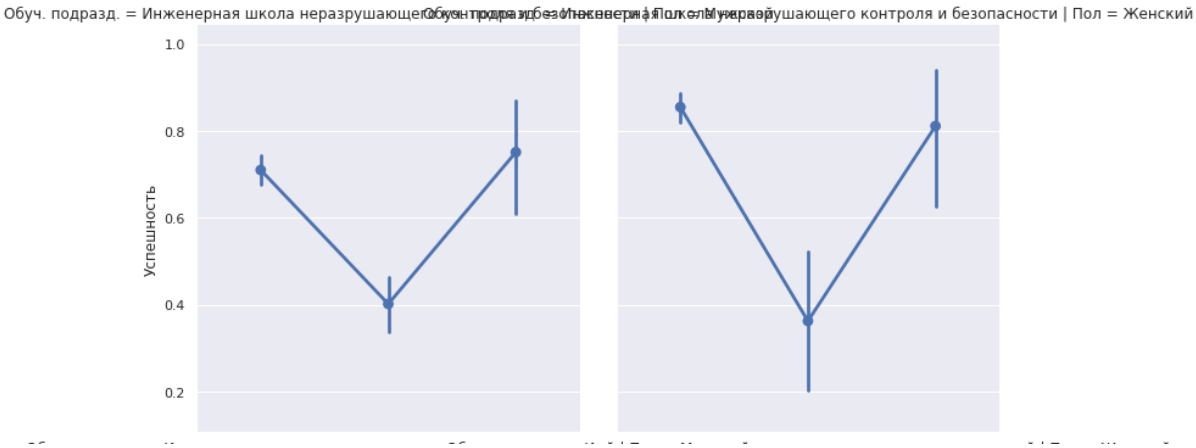


Рисунок 63. Категориальные графики успешности студентов разных полов, инженерных школ и форм обучения (часть 2)

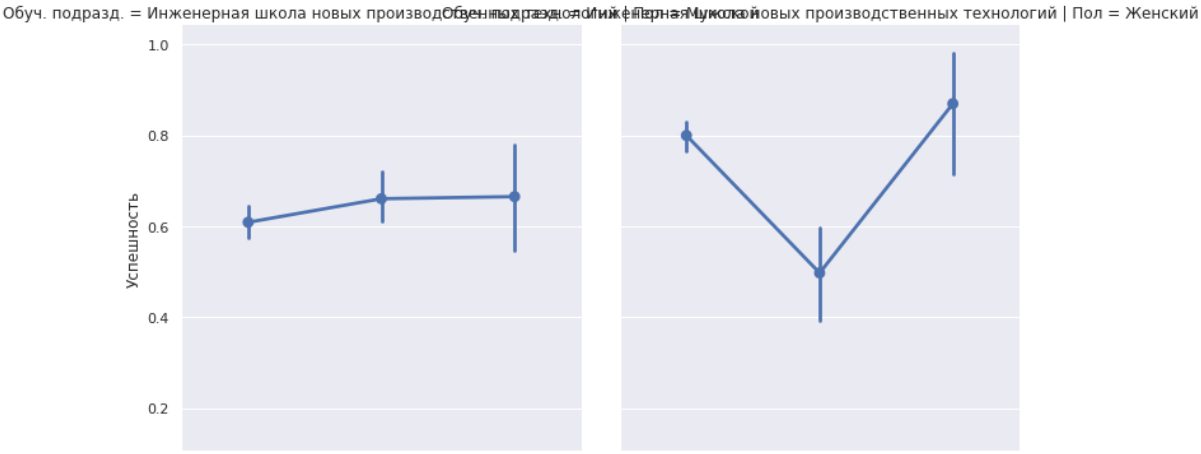


Рисунок 64. Категориальные графики успешности студентов разных полов, инженерных школ и форм обучения (часть 3)

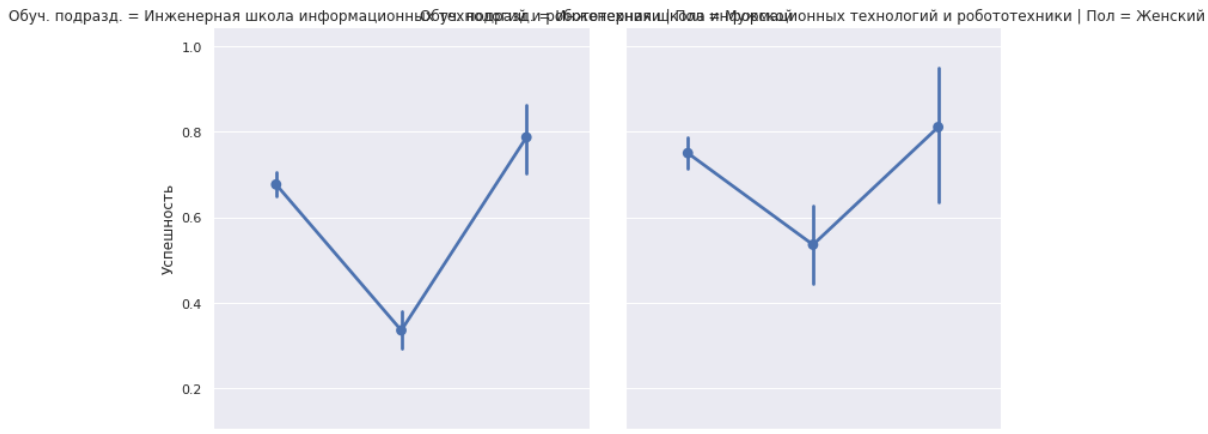


Рисунок 65. Категориальные графики успешности студентов разных полов, инженерных школ и форм обучения (часть 4)

Рисунок 66. Категориальные графики успешности студентов разных полов, инженерных школ и форм обучения (часть 5)

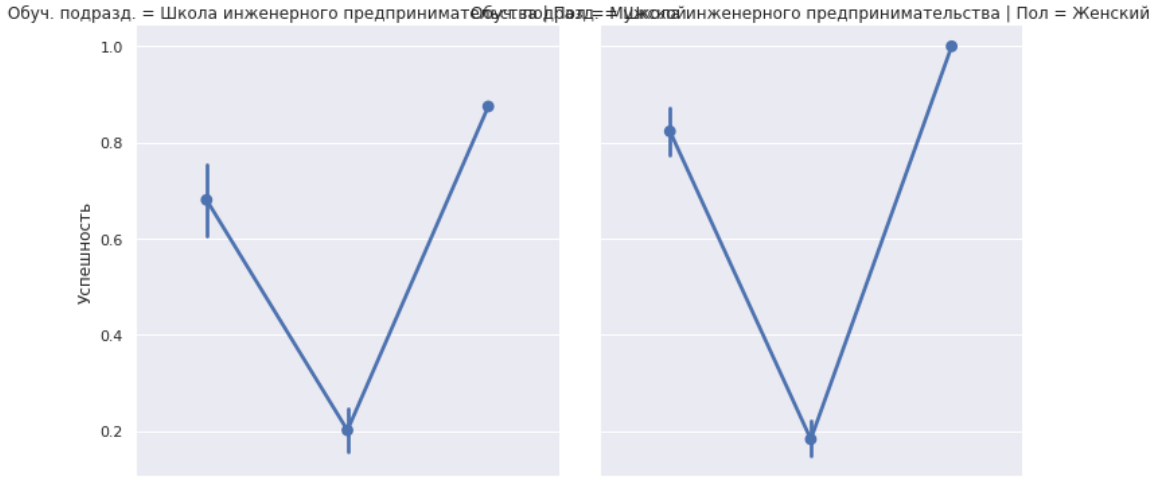


Рисунок 67. Категориальные графики успешности студентов разных полов, инженерных школ и форм обучения (часть 6)

Рисунок 68. Категориальные графики успешности студентов разных полов, инженерных школ и форм обучения (часть 7)

Рисунок 69. Категориальные графики успешности студентов разных полов, инженерных школ и форм обучения (часть 8)

Согласно статистическим критериям, средние значения успешности для студентов, обучающихся по целевому приему, и остальных равны 0,79 и 0,62 соответственно (рисунки 70-71).

```
1 students_df[students_df['Форма финансирования'] == 'по целевому приёму'].Успешность.mean()
```

```
0.7874167323534411
```

Рисунок 70. Среднее значение успешности для студентов, обучающихся по целевому приему

```
1 students_df[students_df['Форма финансирования'] != 'по целевому приёму'].Успешность.mean()
```

```
0.6198307076449135
```

Рисунок 71. Среднее значение успешности для студентов, не обучающихся по целевому приему

Следовательно, используются критерии Манна-Уитни и перестановочный. На рисунке 72 видно, что достигаемый уровень значимости по Манна-Уитни значительно меньше α=0,05.

```
MannwhitneyuResult(statistic=964159.0, pvalue=9.8393283434431e-16)
```

Рисунок 72. P-значение для критерия Манна-Уитни

На рисунке 73 видно, что достигаемый уровень значимости по перестановочному критерию значительно меньше α=0,05.

```
1 print ("p-value: %f" % permutation_test(students_df[students_df['Форма финансирования'] == 'по целевому приёму'].Успешность, \
2                                         students_df[students_df['Форма финансирования'] != 'по целевому приёму'].Успешность, \
3                                         max_permutations = 10000))
```

```
p-value: 0.000000
```

Рисунок 73. P-значение для перестановочного критерия

Скорее всего, студенты, обучающиеся по целевому приему, действительно имеют меньше долгов в процентном соотношении, чем остальные студенты.

На категориальном графике успешности студентов разных полов, инженерных школ и квалификаций, получаемых после обучения, не видна четкая картина преобладания успешности одних студентов над другими (рисунки 74-77).

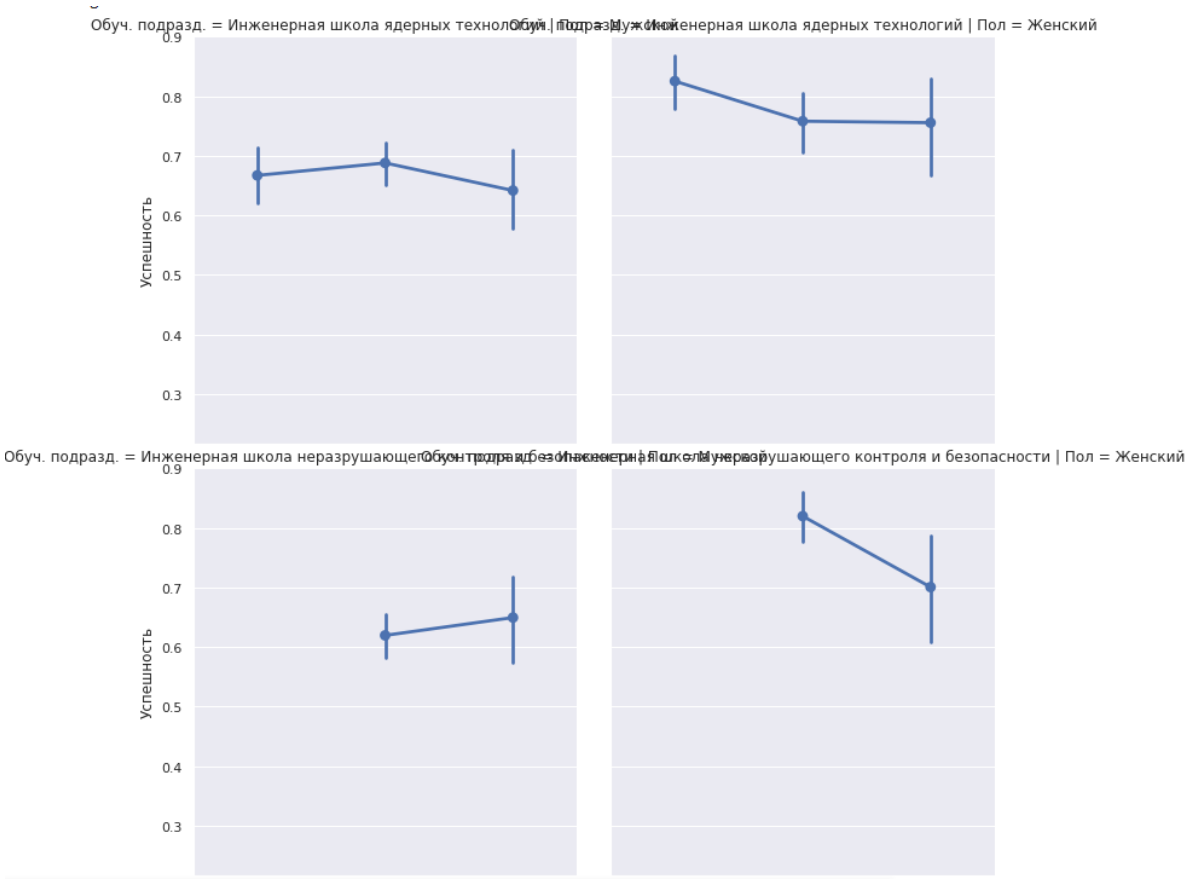


Рисунок 74. Категориальные графики успешности студентов разных полов, инженерных школ и квалификаций (часть 1)

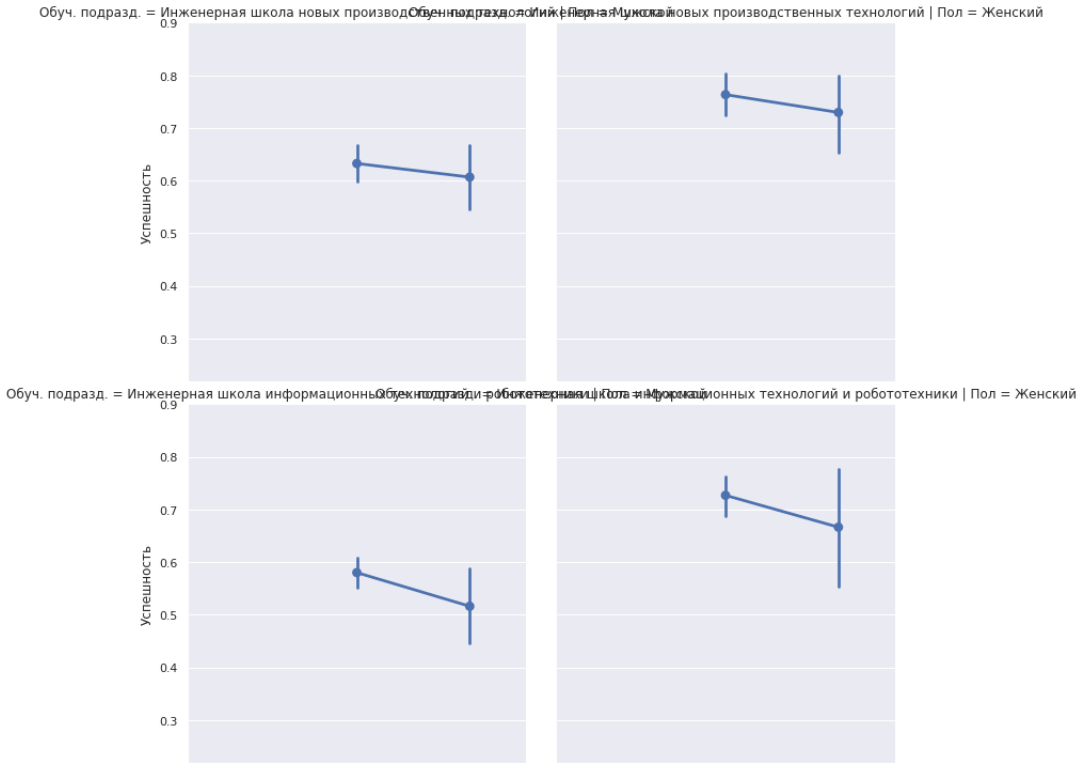


Рисунок 75. Категориальные графики успешности студентов разных полов, инженерных школ и квалификаций (часть 2)

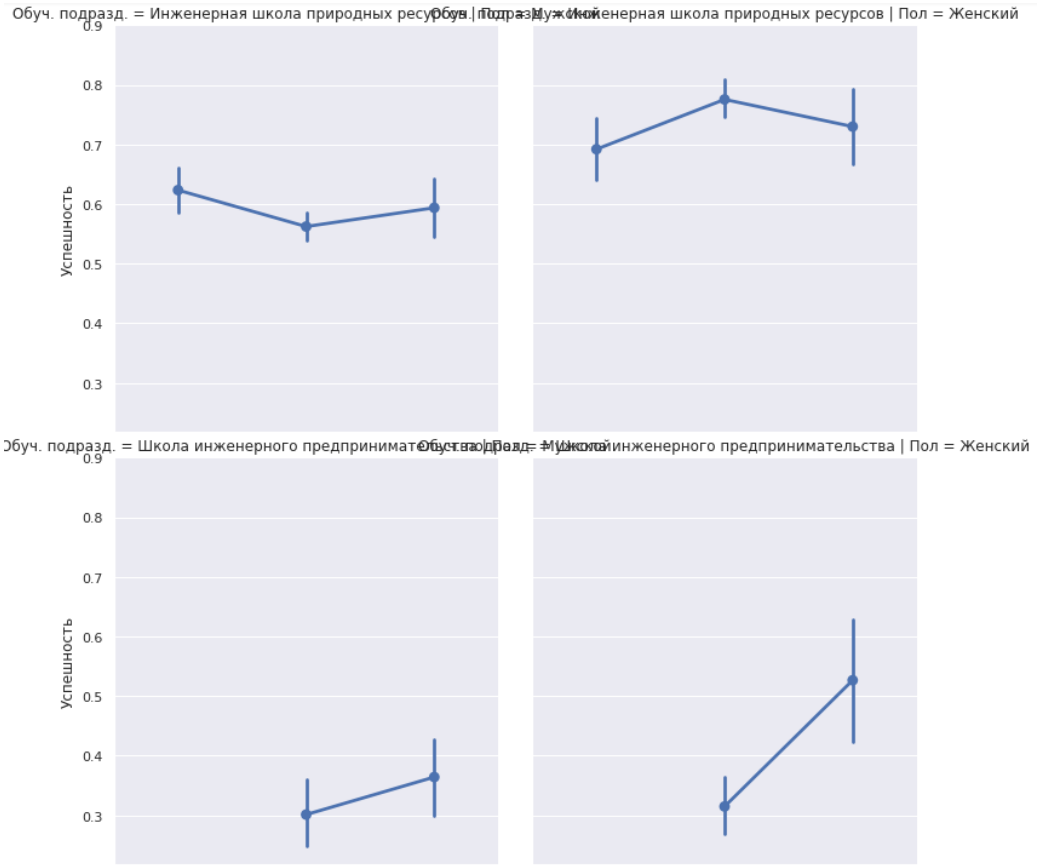


Рисунок 76. Категориальные графики успешности студентов разных полов, инженерных школ и квалификаций (часть 3)

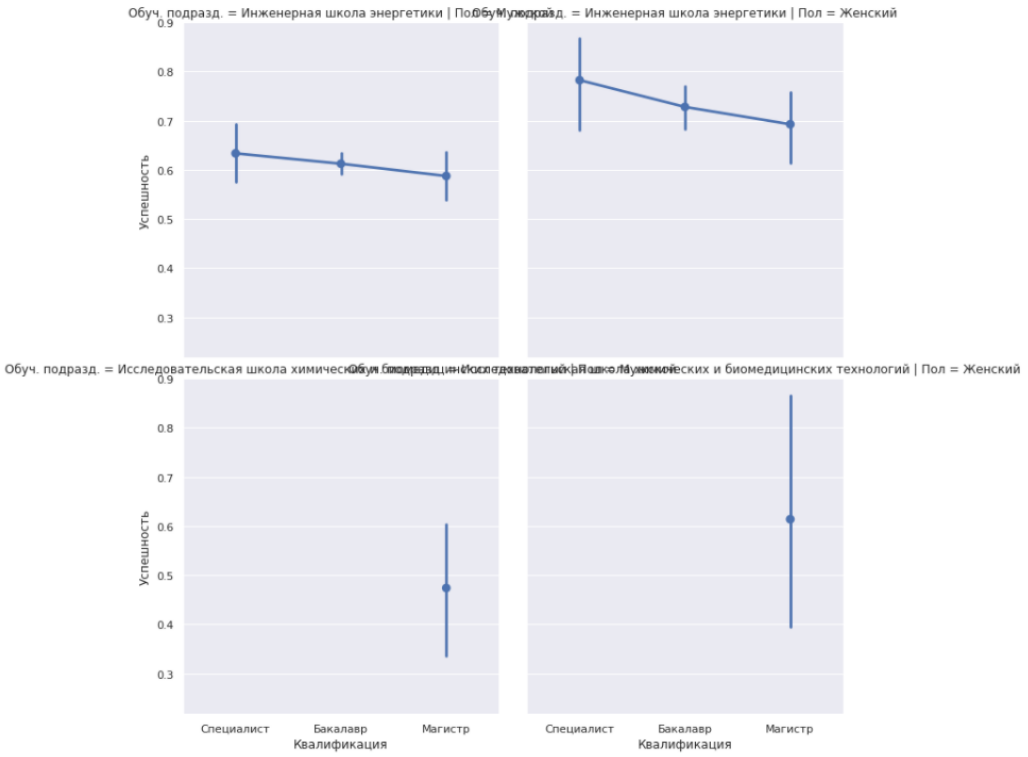


Рисунок 77. Категориальные графики успешности студентов разных полов, инженерных школ и квалификаций (часть 4)

Были рассмотрены диаграммы классов студентов различных квалификаций (рисунок 78). Во всех трех графиках большинство студентов относится ко второму классу, однако в графике, относящемся к бакалавриату, достаточно весомую часть всех студентов забирает нулевой класс.

Рисунок 78. Категориальные графики успешности студентов разных полов, инженерных школ и квалификаций (часть 4)

На рисунке 79 представлена таблица сопряженности класса студента и квалификации, которую он получит после обучения. Здесь уже становится видно, что специалитет обладает наибольшей долей успешных студентов.

Гипотеза: «Средняя успешность студентов, обучающихся на специалитете выше, чем остальных».

Класс	0	1	2	All
Квалификация				
Бакалавр	1676	1209	3443	6328
Магистр	256	386	575	1217
Специалист	182	250	574	1006
All	2114	1845	4592	8551

Рисунок 79. Таблица сопряженности класса студента и его будущей квалификации

На рисунке 80 представлены средние значения успешности студентов, обучающихся на специалитете, и остальных.

```
print(students_df[students_df['Квалификация'] == 'Специалист'].Успешность.mean())
print(students_df[students_df['Квалификация'] != 'Специалист'].Успешность.mean())
```

```
0.6706087015480643
0.6200791532302486
```

Рисунок 80. Средние значения успешности студентов, обучающихся на специалитете, и остальных

Использован критерий Манна-Уитни. На рисунке 81 видно, что достигаемый уровень значимости значительно меньше α=0,05.

```
MannwhitneyuResult(statistic=3590793.0, pvalue=0.0023979664048405986)
```

Рисунок 81. P-значение критерия Манна-Уитни

Использован перестановочный критерий. На рисунке 82 видно, что достигаемый уровень значимости значительно меньше α=0,05.

```
print ("p-value: %f" % permutation_test(students_df[students_df['Квалификация'] == 'Специалист'].Успешность, \
                                         students_df[students_df['Квалификация'] != 'Специалист'].Успешность, \
                                         max_permutations = 10000))
```

```
p-value: 0.000100
```

Рисунок 82. P-значение перестановочного критерия

Следовательно, альтернативная гипотеза принята.

Далее был рассмотрено гражданство. Для начала было изучено, как соотносятся студенты из Российской Федерации и остальные (рисунок 83).

```
students_df['РФ'] = students_df['Гражданство'].apply(lambda x: 1 if x == 'Российская Федерация' else 0)
```

```
students_df['РФ'].hist()
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f95bea89e48>
```

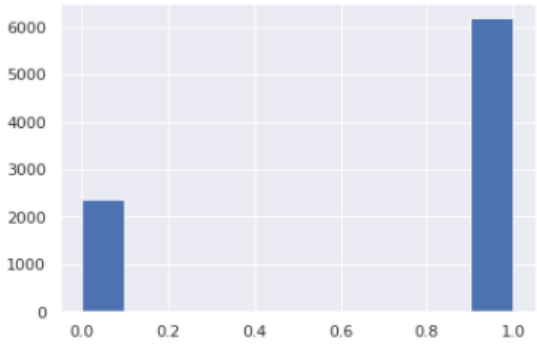


Рисунок 83. Диаграмма распределения студентов (из РФ / не из РФ)

На категориальном графике успешности студентов из РФ и остальных видно явное преобладание успешности студентов из Российской Федерации. (рисунок 84). Была сформулирована следующая гипотеза: «Среднее значение успешности студентов из Российской Федерации выше, чем остальных».

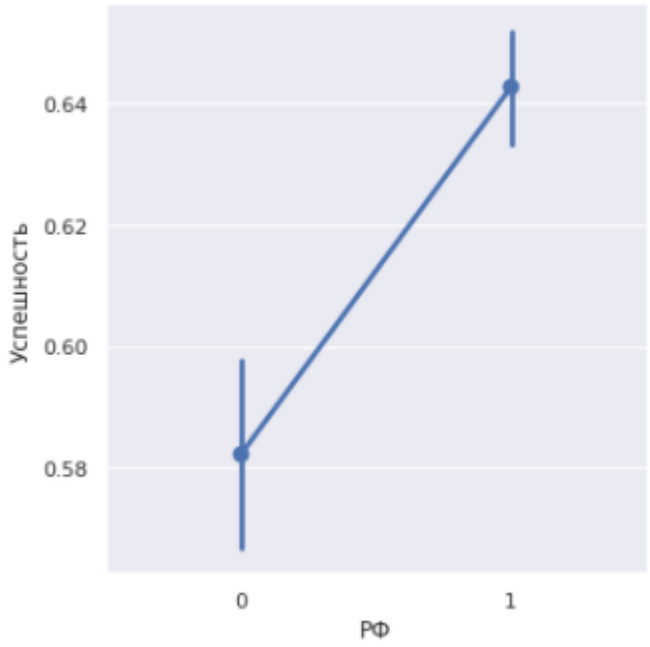


Рисунок 84. Категориальный график успешности студентов из РФ и остальных

На рисунке 85 представлены средние значения успешности студентов из Российской Федерации и остальных.

```
print(students_df[students_df['Гражданство'] == 'Российская Федерация'].Успешность.mean())
print(students_df[students_df['Гражданство'] != 'Российская Федерация'].Успешность.mean())
```

```
0.6427231603640475
0.5821649842006079
```

Рисунок 85. Средние значения успешности студентов из РФ и остальных

На рисунке 86 видно, что достигаемый уровень по критерию Манна-Уитни меньше 0,05.

```
MannwhitneyuResult(statistic=6632199.0, pvalue=1.3631014047108548e-11)
```

Рисунок 86. Достигаемый уровень значимости по критерию Манна-Уитни

На рисунке 87 видно, что достигаемый уровень значимости по перестановочному критерию меньше 0,05.

```
print ("p-value: %f" % permutation_test(students_df[students_df['Гражданство'] == 'Российская Федерация'].Успешность, \
                                         students_df[students_df['Гражданство'] != 'Российская Федерация'].Успешность, \
                                         max_permutations = 10000))
```

```
p-value: 0.000000
```

Рисунок 87. Достигаемый уровень значимости по перестановочному критерию

Следовательно, принята гипотеза о том, что среднее значение успешности студентов из Российской Федерации выше, чем остальных.

На категориальном графике успешности и курса студентов видно, что пик успешности происходит на четвертом курсе. (рисунок 88). Сформулирована следующая гипотеза: «Среднее значение успешности студентов четвертого курса выше, чем остальных».

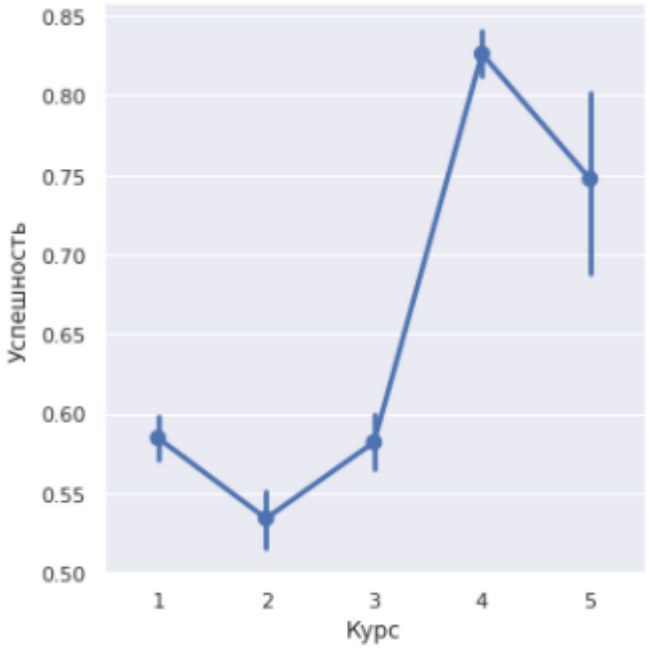


Рисунок 88. Категориальный график успешности и курса студентов

Рассмотрены диаграммы классов студентов различных курсов (рисунок 89). Видно, что большая часть четверокурсников принадлежит второму классу. При этом на всех курсах доли студентов, относящихся к нулевому и первому классу, остается одинаковой.

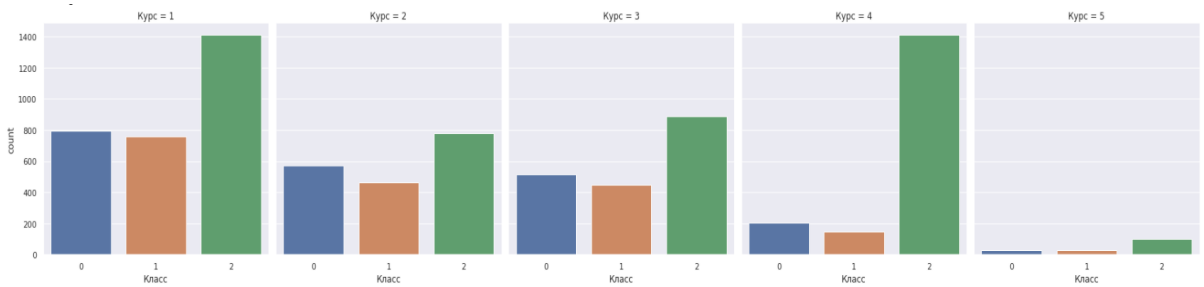


Рисунок 89. Диаграммы классов студентов различных курсов

На рисунке 90 представлены средние значения успешности студентов четвертого курса и остальных.

```
print(students_df[students_df['Курс'] == 4].Успешность.mean())
print(students_df[students_df['Курс'] != 4].Успешность.mean())
```

```
0.8261452648704796
0.5739733528416114
```

Рисунок 90. Средние значения успешности студентов четвертого курса и остальных

На рисунке 91 видно, что достигаемый уровень по критерию Манна-Уитни меньше 0,05. Следовательно, четверокурсники действительно имеют меньше долгов.

```
stats.mannwhitneyu(students_df[students_df['Курс'] == 4].Успешность, \
                   students_df[students_df['Курс'] != 4].Успешность)
```

```
MannwhitneyuResult(statistic=3276875.5, pvalue=2.188231311268633e-195)
```

Рисунок 91. Достигаемый уровень значимости по критерию Манна-Уитни

На рисунке 92 видно, что достигаемый уровень по перестановочному критерию также меньше 0,05.

```
print ("p-value: %f" % permutation_test(students_df[students_df['Курс'] == 4].Успешность, \
                                         students_df[students_df['Курс'] != 4].Успешность, \
                                         max_permutations = 10000))
```

```
p-value: 0.000000
```

Рисунок 92. Достигаемый уровень значимости по перестановочному критерию

Следовательно, гипотеза принята. Среднее значение успешности студентов четвертого курса действительно значимо больше, чем у остальных студентов.

Далее были рассмотрены диаграммы классов студентов, находящихся и не находящихся в академическом отпуске (рисунок 93). Видно, что на каждом графике студентов, относящихся ко второму классу, большинство.

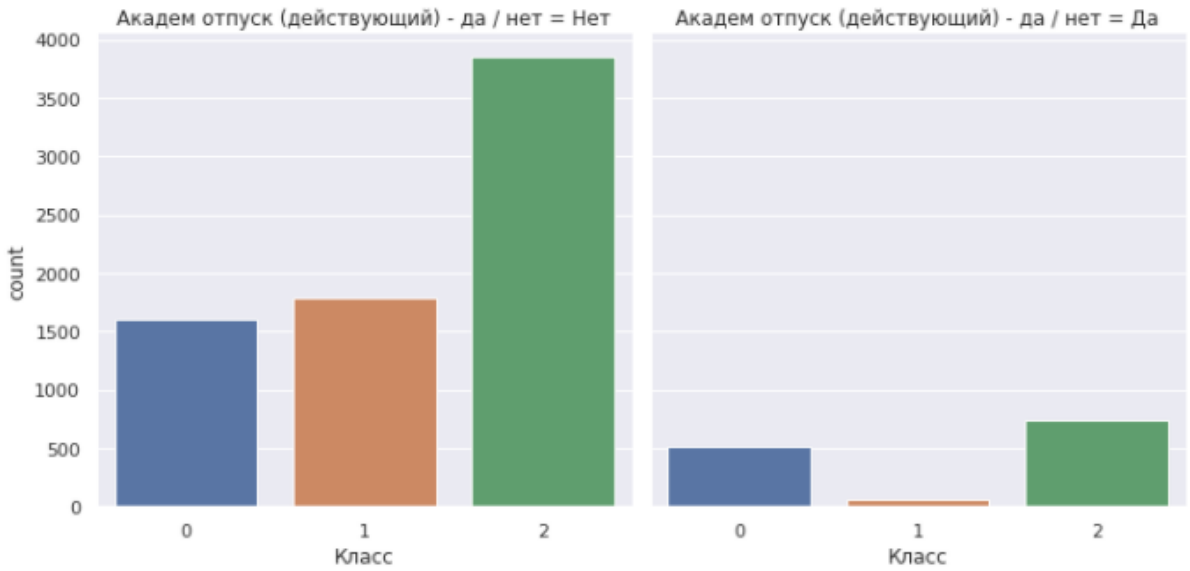


Рисунок 93. Диаграммы классов студентов, находящихся и не находящихся в академическом отпуске

На рисунке 94 представлены средние значения успешности студентов, находящихся и не находящихся в академическом отпуске. Заметим, что у вторых оно выше.

```
print(students_df[students_df['Академ отпуск (действующий) - да / нет'] == 'Да'].Успешность.mean())
print(students_df[students_df['Академ отпуск (действующий) - да / нет'] == 'Нет'].Успешность.mean())
```

```
0.5813009954715022
0.6340565138330374
```

Рисунок 94. Средние значения успешности студентов, находящихся и не находящихся в академическом отпуске

На рисунке 95 видно, что достигаемый уровень значимости по критерию Манна-Уитни меньше 0,05, хотя и относительно близок к нему. Тем не менее, можно считать, что студенты, не находящиеся в академическом отпуске, имеют меньше долгов.

```
MannwhitneyuResult(statistic=4572751.5, pvalue=0.035035514361236635)
```

Рисунок 95. Достигаемый уровень значимости по критерию Манна-Уитни

На рисунке 96 видно, что достигаемый уровень по перестановочному критерию также меньше 0,05.

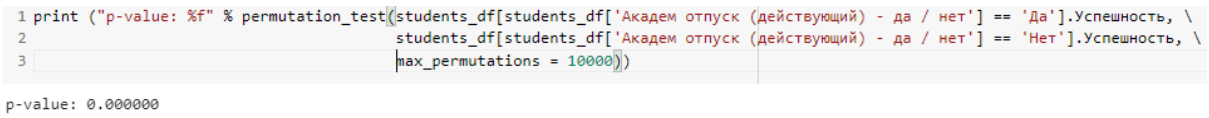


Рисунок 96. Достигаемый уровень значимости по перестановочному критерию

Таким образом, альтернативная гипотеза принята.

На этом формулирование и проверка статистических гипотез завершена. Всего выводов несколько:

- Студенты женского пола имеют меньше не сданных дисциплин, чем студенты мужского;
- Студенты ИШЯТ и ИШНКБ имеют меньше несданных дисциплин по сравнению с остальными;
- Студенты, обучающиеся по целевому приему, имеют меньше несданных дисциплин по сравнению со студентами с другими формами финансирования;
- Студенты, обучающиеся на специалитете, имеют меньше несданных дисциплин по сравнению с остальными;
- Студенты, имеющие гражданство Российской Федерации, имеют меньше несданных дисциплин по сравнению с остальными;
- Студенты-четверокурсники обучаются успешнее студентов других курсов;
- Студенты, не состоящие в академическом отпуске, обучаются успешнее.

2.1.4. Построение предсказательных моделей и их оценка

Ниже представлен список признаков, которые не понадобятся при обучении модели:

- «Всего»;
- «Положительных»;
- «Неудовлетворительных»;
- «Успешность»;
- «Специальность»;
- «Выпуск. школа»;
- «Группа»;
- «Страна»;
- «Дисциплины по которым получены неудовлетворительные оценки»;
- «Индекс студента»;
- «Год рождения»;
- «Всего часов по дисциплинам по которым получены неудовлетворительные оценки».

Данные уже нормализованы и приведены в бинарный вид, читаемый алгоритмами машинного обучения.

Далее были созданы наборы для тренировочной и тестовой выборки (разбиение 2 к 1). Разбиение производится для целевого вектора и для целевой матрицы (рисунок 97).

```
X_train, X_test, y_train_onehot, y_test_onehot = train_test_split(
    values, target_onehot, test_size=0.33, random_state=42)
X_train, X_test, y_train, y_test = train_test_split(
    values, target, test_size=0.33, random_state=42)
```

Рисунок 97. Достигаемый уровень значимости по перестановочному критерию

Сперва были применены модели логистической и линейной регрессии. Модель линейной регрессии была взята из библиотеки sklearn. Точность на тестовых данных оказалась равной 58.5%, что не соответствует ожиданиям. Модель линейной регрессии является достаточно слабой моделью, так как

способна лишь показать линейную зависимость между целевой и объясняющими переменными.

Была применена логистическая регрессия. Модель логистической регрессии была также взята из библиотеки sklearn. Итоговая точность на тестовых данных равна 73.4%. Качество предсказания увеличилась на 14.9%, что является хорошим приростом, но сама точность недостаточна.

Далее был опробован метод k-ближайших соседей из библиотеки sklearn. В данном методе нужно указать число k соседей, что является параметром экспериментальным. Для этого был создан цикл с перебором первых 20-ти значений параметра k и посмотрим, при каком значении точность модели будет наибольшей. На рисунке 98 представлен график зависимости точности предсказания от числа соседей.

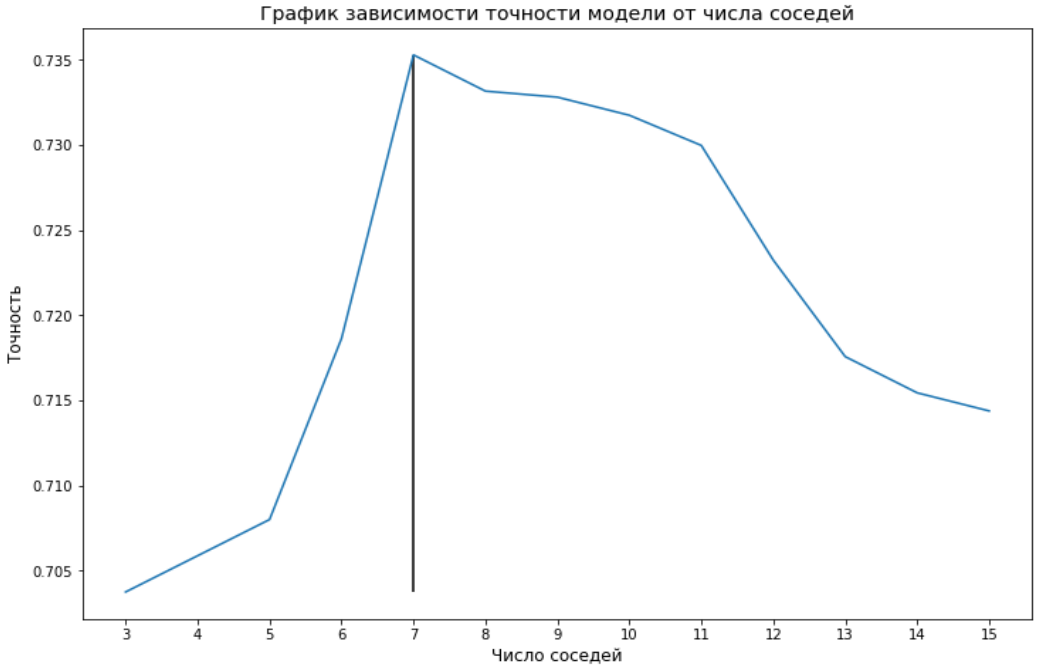


Рисунок 98. График зависимости точности прогноза от числа соседей

Наибольшая точность, которой удалось достигнуть с помощью данного алгоритма – 73,5%, что всего на 0.1% больше, чем у логистической регрессии.

Следующей моделью прогнозирования будет Random forest (он же случайный лес). Был также выполнен перебор параметров алгоритма, и на рисунке 99 можно увидеть зависимость точности предсказания от максимальной глубины дерева.

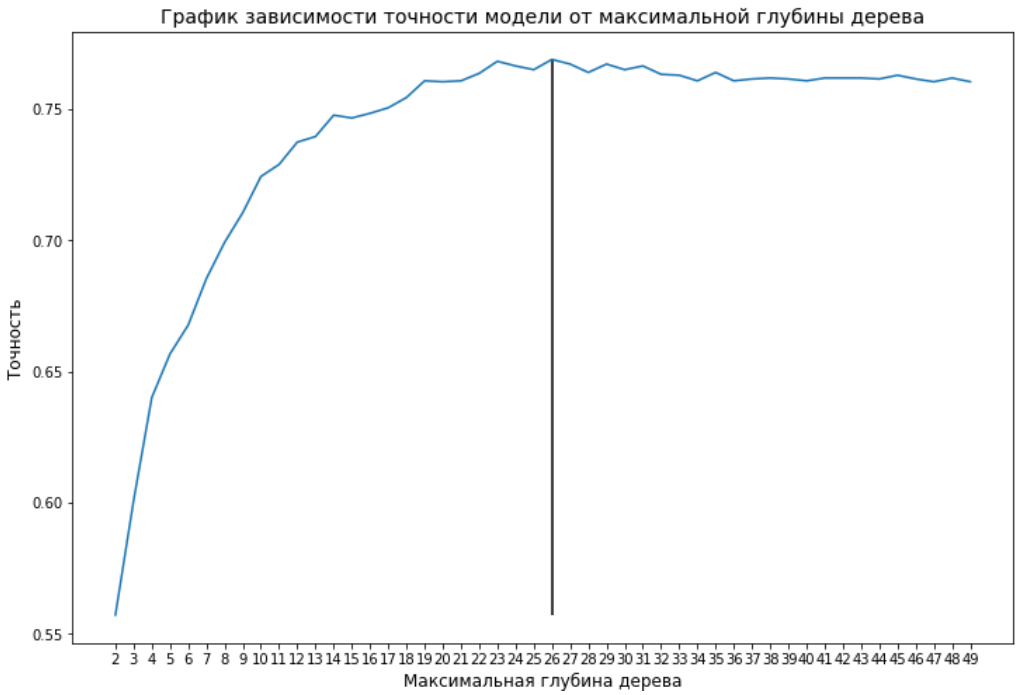


Рисунок 99. График зависимости точности прогноза от максимальной глубины дерева

По данному графику видно, что модель при дальнейшем увеличении максимальной глубины дерева (больше 20) начинает сходиться. Лучшая точность, которой удалось достичь, равна 77%.

Теперь рассмотрен метод опорных векторов. Метод опорных векторов имеет несколько различных модификаций, однако в данной работе будет использоваться метод из библиотеки sklearn с перебором параметра регуляризации C. Этот параметр обратно пропорционален силе L_2-регуляризатора, а штраф за ошибку модели равен штрафу L_2, возведенный в квадрат. На рисунке 100 представлен результат поиска наилучшей точности прогнозирования.

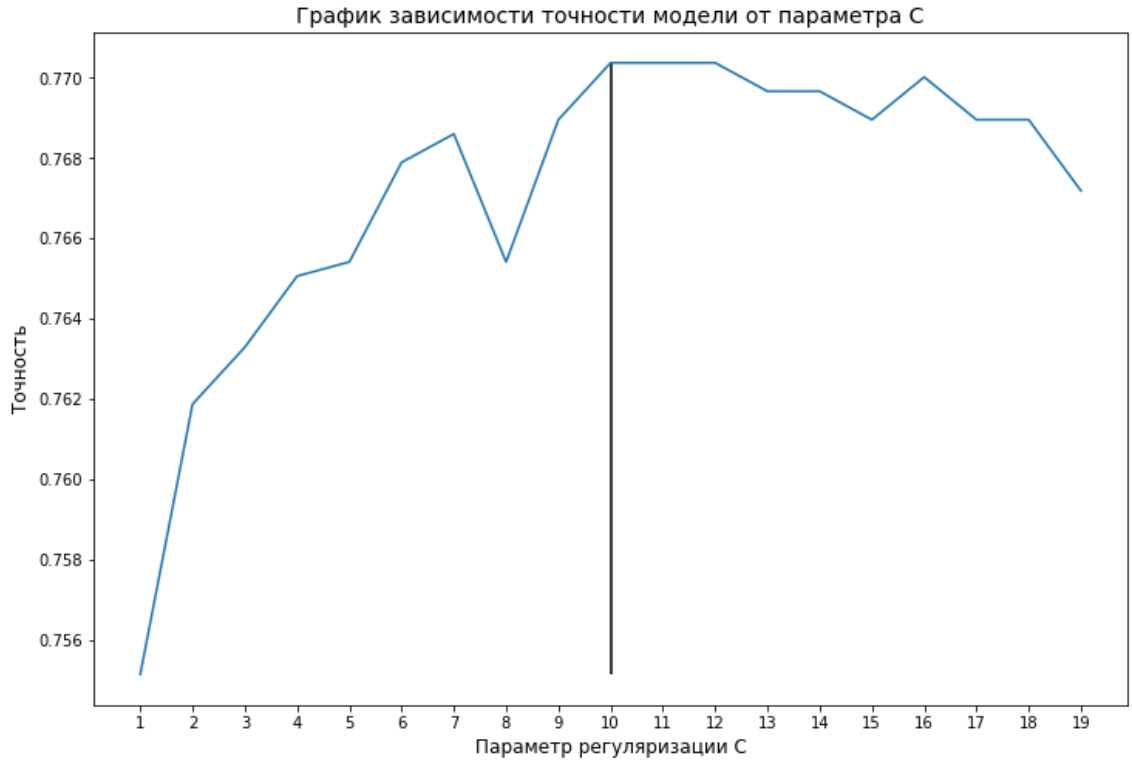


Рисунок 100. График зависимости точности прогноза от параметра C

Итоговая точность модели, построенной на основе метода опорных векторов, равна 77,4%.

Последней моделью является модель нейронной сети. Эта непростая, но в то же время гибкая модель машинного обучения способна перенастраивать свои веса в заранее заданной структуре нейронной сети, тем самым устремляя ошибку прогноза по наибыстрейшему пути спуска – антиградиенту.

Модель была построена с помощью библиотеки keras, которая представляет собой надстройку над фреймворками TensorFlow. Архитектура сети следующая:

- Размер входного вектора – 235;
- Число нейронов на входном слое – 128;
- Функция активации на входном слое – Tanh (гиперболический тангенс);
- Слой Dropout (слой для отключения нейронов с вероятностью 0.2);
- Число нейронов на скрытом слое – 256;
- Функция активации на скрытом слое – ReLU;
- Слой Dropout (слой для отключения нейронов с вероятностью 0.2);
- Число нейронов на выходном слое – 3;

- Функция активации на выходном слое – Softmax;
- Функция оптимизации – SGD (стохастический градиентный спуск);
- Функция потерь – категориальная кросс-энтропия;
- Метрика качества – точность (accuracy).

Несмотря на свою гибкость, нейронные сети для хорошего результата требуют определенных знаний и опыта. Итоговая точность, которой удалось достичь данным алгоритмом – 75.4%.

Таблица 3. Сравнение предсказательных моделей

Название алгоритма	Прогнозная точность, %	Время обучения, с
Линейная регрессия	0.585	0.212
Логистическая регрессия	0.734	0.874
К-ближайших соседей	0.735	0.492
Случайный лес	0.770	1.418
Метод опорных векторов	0.774	6.843
Модель нейронной сети	0.754	~120 за 100 эпох

Из таблицы можно выделить три лучших алгоритма по прогнозной точности и три по времени обучения.

По прогнозной точности:

- Метод опорных векторов
- Случайный лес
- Нейронная сеть

По времени обучения:

- Линейная регрессия
- К-ближайших соседей
- Логистическая регрессия

Таким образом, лучшей моделью оказалась модель на основе метода опорных векторов с точностью прогнозирования 77,4%. Метод опорных векторов будет использоваться в качестве прогнозной модели в веб-приложении.

2.2. Анализ данных о группах, дисциплинах и преподавателях

2.2.1. Предобработка «сырых» данных

В датасете о средней успеваемости групп по дисциплинам на осенний семестр 2019 года (он же исходный набор данных) всего 5988 строк и 22 столбца-параметра. Из 22 столбцов 16 принимают категориальные значения, а оставшиеся 6 – числовые. При этом столбцы с названиями «Индексы преподавателей КТ2», «Индексы преподавателей Итог», «Лекторы», «Преподаватели» в одной строке могут содержать от 0 до N значений, в то время как остальные столбцы в одной строке имеют лишь одно значение.

На рисунках 101-102 представлены фрагменты исходной таблицы.

	Форма обучения	Квалификация	Курс	Специальность	Подраздел. (вып.)	Школа (вып.)	Дисциплина	Вид аттестации	Подраздел. (обесп.)	Группа	...
0	Очная	Бакалавр	1	54.03.01 Дизайн	ОАР	ИШИТР	Цветоведение и колористика	Экзамен	ОАР	8Д81	...
1	Очная	Бакалавр	1	54.03.01 Дизайн	ОАР	ИШИТР	Введение в профессиональную деятельность	Зачет	ОАР	8Д81	...
2	Очная	Бакалавр	1	54.03.01 Дизайн	ОАР	ИШИТР	Академические живопись и рисунок	Экзамен	ОАР	8Д81	...
3	Очная	Бакалавр	1	54.03.01 Дизайн	ОАР	ИШИТР	История искусств и культура профессионального ...	Экзамен	ОАР	8Д81	...

Рисунок 101. Исходный датасет (часть 1)

Индекс преподавателей Итог	Лекторы	Преподаватели	Подраздел. (преп.)	Всего	На отлично	На хорошо	На удовлетворительно	Сдавших	Не сдавших
300	300	300, 449	ОАР	30	8	19	1	28	2
919	300		ОАР	30	0	0	0	27	3
190		190	ОАР	30	13	15	1	29	1
988	988	988	ОАР	30	8	15	4	27	3

Рисунок 102. Исходный датасет (часть 2)

На рисунках выше можно обратить внимание на наличие пропусков (они же миссинги), но так как это столбцы с категориальными значениями, очистка от миссингов не требуется. На рисунке 103 представлены гистограммы распределения категориальных параметров.

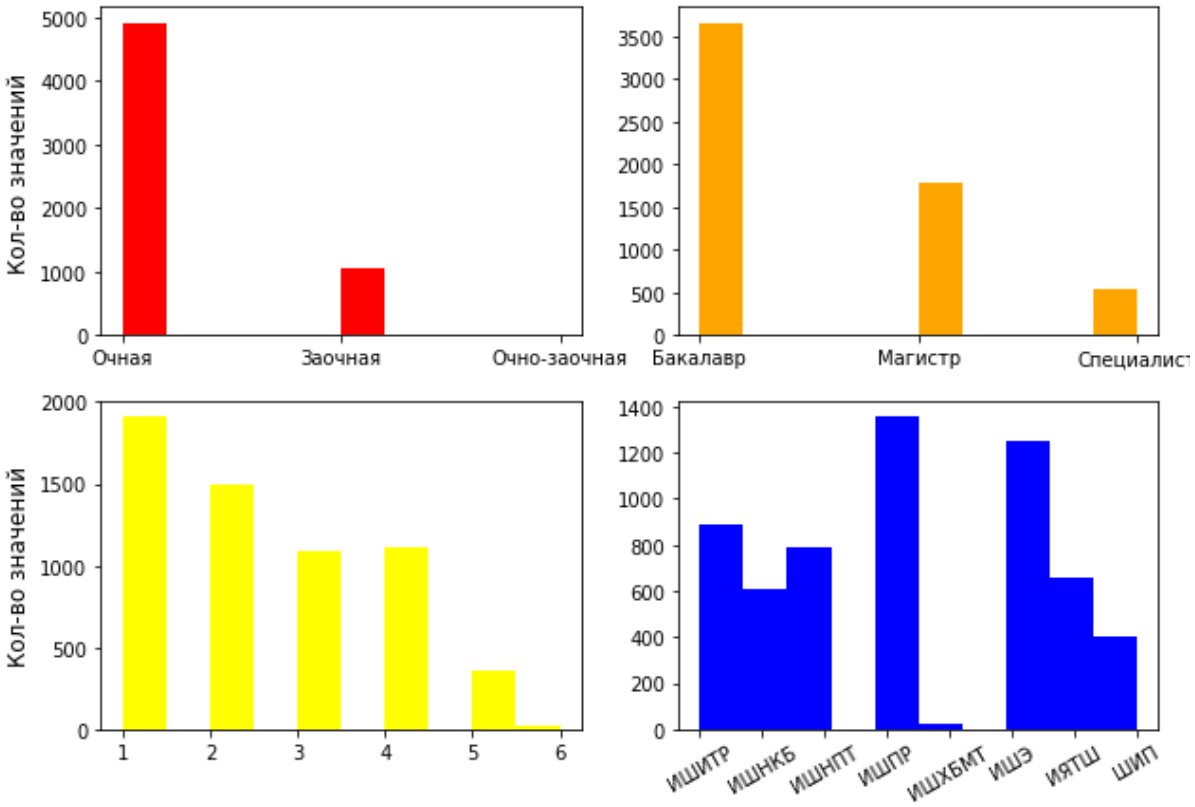


Рисунок 103. Гистограммы распределения категориальных параметров

Из числовых характеристик интересны только столбцы «Всего» и «Не сдавших», так как необходимо обнаружить признаки неуспешного студента для дальнейшего их [студентов] предсказания и упреждающего решения до того, как студент не сдаст дисциплины. Были рассмотрены диапазоны значений у этих двух столбцов. На рисунке 104 представлены гистограммы распределений числа студентов, которые сдавали определенную дисциплину, и количества несдач, а также диаграммы размаха. Все значения не выходят за рамки адекватного представления данных. Также по гистограмме справа можно понять, что большая часть (свыше 4 тыс.) уникальных пар «группа-дисциплина» в столбце «Не сдавших» имеет значение от 0 до 5. Иначе говоря, медиана несдач по группам равна 2 (среднее – 3.735). Подробнее со статистикой данного столбца можно ознакомиться на рисунке 105.

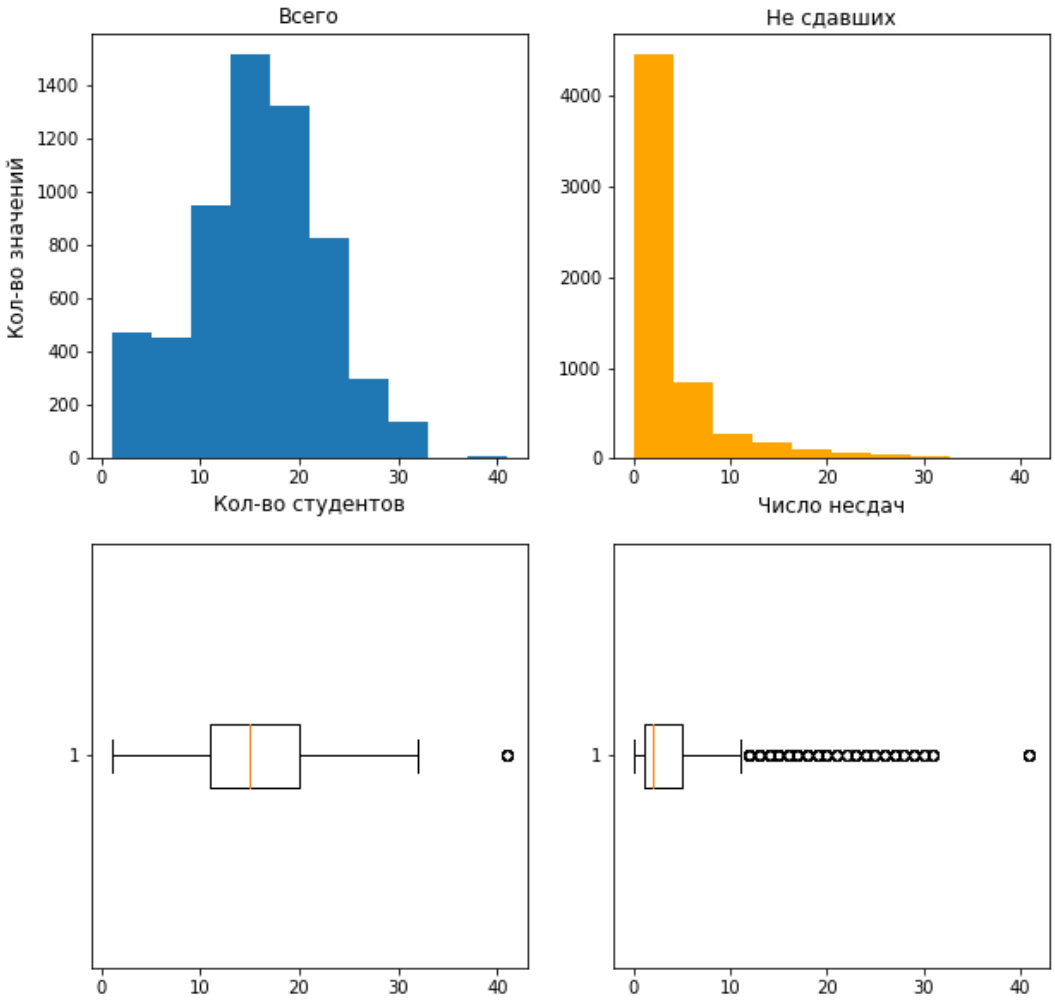


Рисунок 104. Гистограммы распределения столбцов и их диаграммы размаха

	Всего		Не сдавших
count	5988.000000	count	5988.000000
mean	15.297261	mean	3.735471
std	6.678106	std	5.240993
min	1.000000	min	0.000000
25%	11.000000	25%	1.000000
50%	15.000000	50%	2.000000
75%	20.000000	75%	5.000000
max	41.000000	max	41.000000

Рисунок 105. Статистика по столбцам «Всего» и «Не сдавших»

На основании разведочного анализа было установлено, что пороговым значением по несдачам будет число 5 (75 процентиль диаграммы размаха) не включительно. Также замечен выброс (число 41) и пары «группа-дисциплина», удовлетворяющие этому значению, будут рассмотрены отдельно от всех остальных групп значений (всего групп значений 3: первая – хорошие значения

от 0 до 5 несдач на пару «группа-дисциплина», вторая – неудовлетворительные значения более 5 несдач, и третья – 41 несдача).

2.2.2. Разведочный анализ данных

Данные, образовавшие выброс со значением несдач, равным 41, описывают группу Д-5А81 1 курса бакалавриата заочной формы обучения, в которой обучались 41 студент. Всего в исходной таблице 7 записей (пар «группа-дисциплина»), удовлетворяющих такому запросу.

Также следует рассмотреть вариант, когда в значении столбца «Всего» для пары «группа-дисциплина» меньше 6 студентов. Данный запрос выдал 550 записей. На рисунке 106 представлено распределение пары «группа-дисциплина» не сдавших. В целом, распределение похоже на генеральную совокупность.

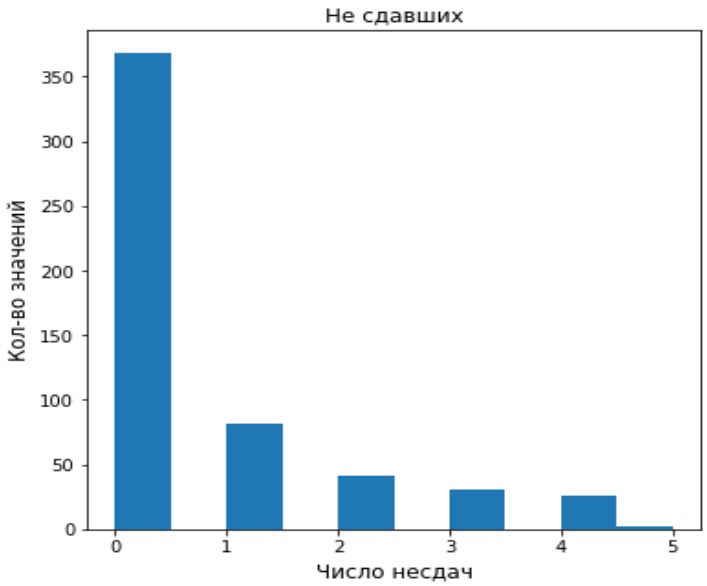


Рисунок 106. Гистограмма распределения «Не сдавших»

Далее рассмотрена вторая группа значений несдач. Всего было получено 1215 записей, в которых 331 группа и 391 дисциплина, что достаточно много. Также в эту группу входят пары «группа-дисциплина» как очной, так и заочной формы обучения всех курсов, квалификаций, инженерных школ и видов аттестации. На рисунках 107-109 представлены графики распределения этих пар по вышеперечисленным пяти параметрам как в исходном наборе данных, так и в отфильтрованном по порогу несдач, а также указано ниже в числовом виде

отношение долей значений столбцов для пар «группа-дисциплина» по фильтрованному датасету и исходному.

Значения больше 1 указывают, что данное свойство характерно для фильтрованной группы, и возможно, может быть определяющим фактором в причинах низкой успеваемости. Значения меньше 1 указывают на обратную зависимость: это свойство не характерно для фильтрованной группы. Значения, равные 1, говорят о том, что эти свойства не имеют значения, отфильтрована группа по числу несдач или нет.

Также был составлен более подробный список отношения долей, в который были внесены больше столбцов для более точечного вычленения выделяющиеся подгрупп. Список отсортирован от большего значения доли к меньшему с установленным порогом отсечения, равным 1.25. Парам «группа-дисциплина», которые имеют много несдач, присущи следующие свойства:

- Форма обучения – Заочная
- Курс – [3, 5, 6]
- Подразделение (выпускающее) – [ШИП, НОЦ И.Н. Бутакова]
- Подразделение (обеспечивающее) – [прочее, ОЭФ, ООД, ОМИ, ОАР, ОИТ, НОЦ И.Н. Бутакова, ОЕН, ОМ]
- Школа (выпускающая) – ШИП
- Школа (обеспечивающая) – ИШИТР
- Вид аттестации – [Курсовой проект, Курсовая работа]
- Специальность – [Менеджмент, Программная инженерия, Экономика, Теплоэнергетика и теплотехника, Информационные системы и технологии, Мехатроника и робототехника, Управление в технических системах, Биотехнические системы и технологии, Конструкторско-технологическое обеспечение машиностроительных производств, Техносферная безопасность, Прикладная математика, Информатика и вычислительная техника, Экология и природопользование, Дизайн, Автоматизация технологических процессов и производств, Информатика

и вычислительная техника, Нефтегазовое дело, Прикладная геология, Машиностроение, Физика, Теплоэнергетика и теплотехника].

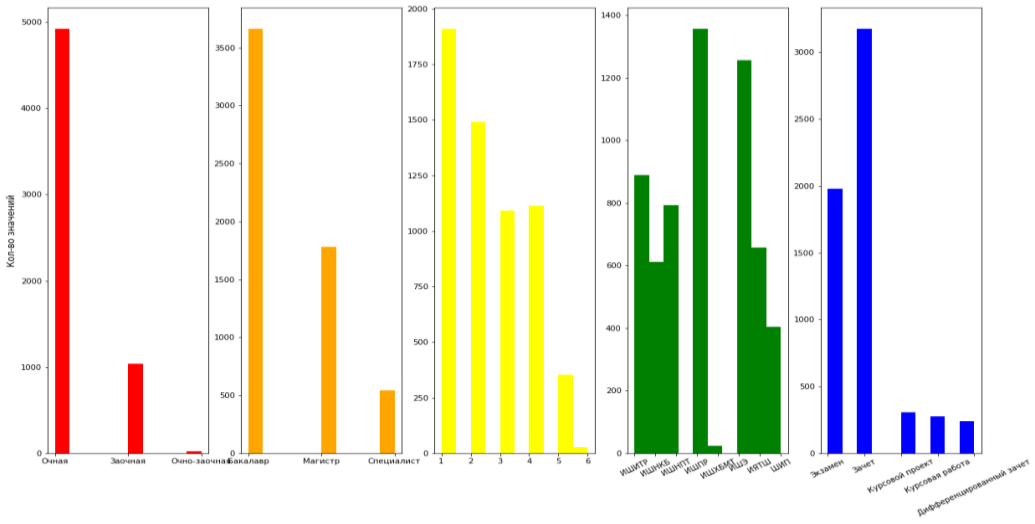


Рисунок 107. Гистограммы распределения по пяти параметрам в исходном датасете

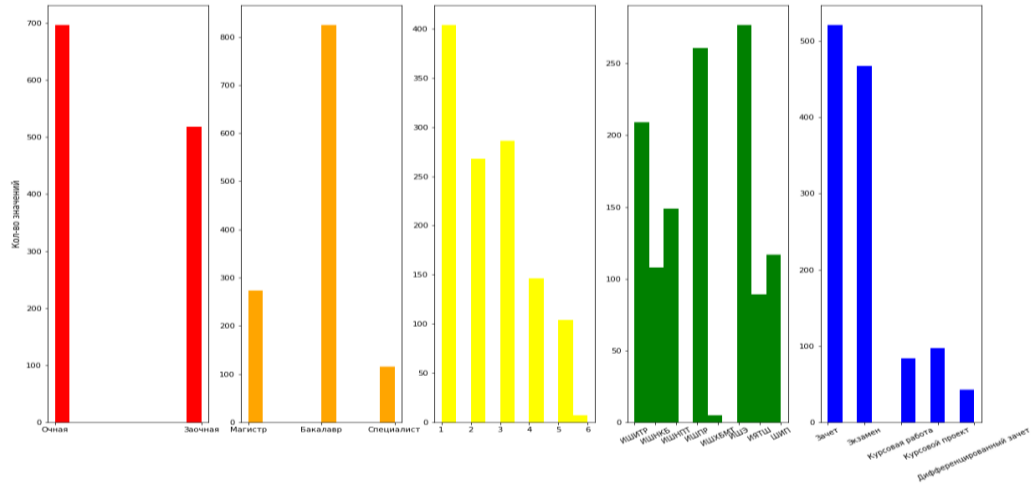


Рисунок 108. Гистограммы распределения по пяти параметрам в отфильтрованном датасете

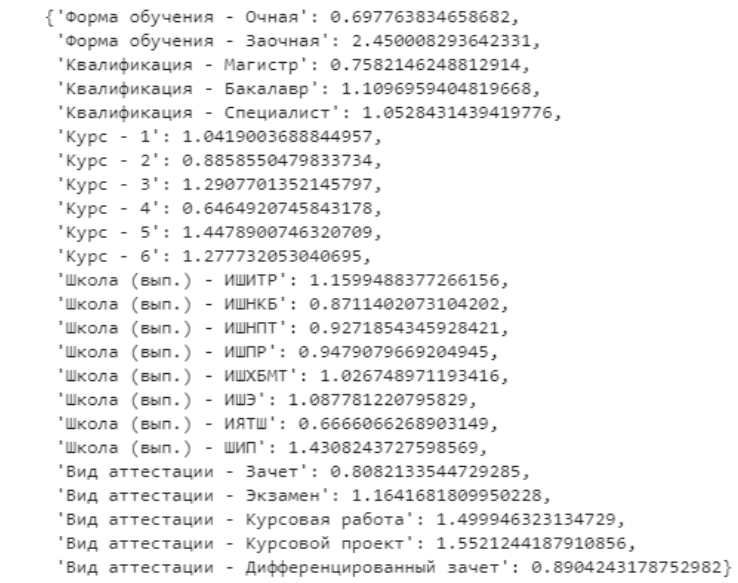


Рисунок 109. Соотношение доли каждого значения параметра в отфильтрованном датасете к исходному датасету

2.2.3. Поиск скрытых закономерностей

Данная глава посвящена ответу на вопрос: по какой причине у наблюдаемых групп низкая успеваемость? Это результат слабой подготовки группы, требовательного преподавателя, или вовсе сложности предмета?

Для начала была построена таблица, в которой преподаватель будет формировать строки. Для этого из исходной таблицы среди четырех столбцов были взяты индексы преподавателей и назначены им соответствующие позиции (если преподаватель был взят из столбца «Преподаватели», то его позиция будет называться «Преподаватель»). В новой таблице получилось 14775 строк и 8 столбцов. После проверок выяснилось, что у двух записей в исходной таблице не было ни в одном столбце индекс преподавателя, из-за чего дисциплина «Протоколы обмена медицинской информацией» не может быть вставлена в новую таблицу. Информация о медиане не сдавших не исказилась, однако 75 процентиль изменил свое значение с 5 на 4, но несмотря на это пороговое значение не было изменено. Для полученной таблицы были проделаны все те же действия, что и для исходной. Таким образом, получилось 1316 записей. На основе долей несдавших был разработан рейтинг и получена таблица

преподавателей и их суммы долей несдач. На рисунке 110 представлен фрагмент такой таблицы.

Преподаватель	Доля_несдавших
933	75.114037
384	61.121270
363	59.316297
1171	58.104252
610	57.863522
...	...
1073	0.000000
1047	0.000000
1018	0.000000
1015	0.000000
1012	0.000000

1316 rows × 1 columns

Рисунок 110. Фрагмент таблицы преподавателей и их рейтингов

Как затем оказалось, преподаватели, в значениях столбца которых нет позиций «Преподаватель» или «Лектор», имеют очень высокие рейтинги, а также множество записей. По этой причине группа таких преподавателей была отсеяна. Также был установлен порог числа несдавших, равный 5. На рисунке 111 представлен фрагмент отфильтрованной таблицы, в которой нет позиций «Индексы преподавателей КТ2», «Индексы преподавателей Итог» и их комбинации.

Преподаватель	Доля_несдавших
1145	12.000000
1271	9.565657
617	9.000000
245	8.681818
637	8.000000
...	...
986	0.230769
38	0.230769
270	0.230769
1013	0.230769
111	0.206897

636 rows × 1 columns

Рисунок 111. Фрагмент отфильтрованной таблицы преподавателей и их рейтингов

Пусть порог доли несдавших равен 3. Тогда 111 преподавателей окажутся с рейтингами выше установленного порога.

То же самое было проделано и для групп с оговоркой, что они будут получаться из фильтрованной исходной таблицы, где число не сдавших больше 5. На рисунке 112 представлен фрагмент таких групп.

Группа	Доля_несдавших
214Б	10.000000
214А	10.000000
Д-3А61	9.000000
4ТМ81	8.346154
Д-8Т81	8.000000
...	...
2Б71	0.272727
3-213А	0.269231
5А83	0.266667
8Е82	0.250000
2Б72	0.222222

339 rows × 1 columns

Рисунок 112. фрагмент отфильтрованной таблицы преподавателей и их рейтингов

Пусть значение порога, равно 4. Тогда 60 групп будут выделены и рассмотрены.

В итоге была получена таблица из декартового умножения с числом строк 288. В этой таблице соединены преподаватели и группы с высокой долей не сдач. На рисунке 113 представлен фрагмент этой таблицы.

	Преподаватель	Позиции_преподавателя	Дисциплина	Вид аттестации	Группа	Всего_студентов_в_группе	Сдавших	Несдавших	Доля_несдавших
0	1028	Преподаватели	Газотурбинные и парогазовые ТЭС	Экзамен	5БМ81	24.0	7.0	17.0	0.708333
1	1028	Преподаватели, Индексы преподавателей Итог, Ин...	Газотурбинные и парогазовые ТЭС	Курсовой проект	5БМ81	24.0	6.0	18.0	0.750000
2	1028	Преподаватели, Индексы преподавателей Итог, Ин...	Газотурбинные и парогазовые ТЭС	Курсовой проект	5БМ84	24.0	9.0	15.0	0.625000
3	1028	Преподаватели	Газотурбинные и парогазовые ТЭС	Экзамен	5БМ84	24.0	9.0	15.0	0.625000
4	1028	Преподаватели	Нагнетатели АЭС	Экзамен	504И	6.0	0.0	6.0	1.000000
...	...	...	...	...	...	...	...	...	...
283	993	Преподаватели, Лекторы	Маркетинг	Экзамен	Д-3Б72	24.0	0.0	24.0	1.000000
284	993	Преподаватели, Лекторы	Маркетинг	Курсовая работа	Д-3Б72	24.0	0.0	24.0	1.000000
285	999	Преподаватели	Учебно-исследовательская работа студентов	Зачет	Д-8Т71	24.0	0.0	24.0	1.000000
286	999	Преподаватели, Лекторы, Индексы преподавателей...	Программирование и алгоритмизация	Экзамен	Д-8Т61	18.0	0.0	18.0	1.000000
287	999	Преподаватели, Лекторы, Индексы преподавателей...	Программирование и алгоритмизация	Экзамен	Д-8Т62	7.0	0.0	7.0	1.000000

288 rows × 9 columns

Рисунок 113. Фрагмент полученной таблицы

По итогам проведенного анализа были выявлены свойства пар «группа-дисциплина», по которым можно делать выборку, а также построена таблица, в которой объединены преподаватели и группы с высоким рейтингом несдач. С помощью этой таблицы можно выполнять операции вычитания из новой таблицы с ведущим элементом «Преподаватель». Так, например, можно оставить только преподавателей с высоким рейтингом, а группы с низким и наоборот.

Итог проделанной работы: группы вносят больший вклад в высокий рейтинг не сдач, нежели преподаватели. Возможно, все дело в особой категории групп, которые не сдали ни одной дисциплины, в связи с чем общий рейтинг несдач завышается.

Глава 3. Проектирование информационной системы

Проектирование является важным этапом в жизненном цикле информационной системы [3]. Это – фундамент работ, описывающий цели, задачи, компоненты системы и взаимодействия между ними. Поэтому очень важно грамотно подойти к проектированию ИС, так как после этого идет ее разработка, внедрение и дальнейшее сопровождение, и цена исправления ошибок на этих этапах высока, и иногда приходится возвращаться на этап проектирования для устранения серьезных недочетов в архитектуре.

На этапе проектирования прежде всего должны быть получены ответы на такие вопросы: где и как хранить данные, какую модель данных использовать, как организовать передачу данных от одного модуля к другому и т.д. Построение диаграмм вариантов использования системы, потоков данных, классов анализа, компонентов являются основной частью проектирования ИС.

Процесс проектирования схемы хранения, обновления и передачи данных проходит следующим образом: в качестве исходной информации проектировщики получают результаты анализа предметной области, на основе которого принимается решение о используемой модели данных. Полученная в процессе анализа информационная модель сначала преобразуется в логическую, а затем в физическую модель данных.

Параллельно с проектированием схемы хранения, обновления и передачи данных выполняется проектирование процессов, чтобы получить спецификации (описания) всех модулей ИС. Главная цель проектирования процессов заключается в отображении функций, полученных на этапе анализа, в модули информационной системы.

3.1. Роли пользователей в системе и их возможности

В разрабатываемой информационной системе присутствует всего одна роль – аналитик. Аналитик извлекает из данных полезную информацию (находит закономерности, ранее не обнаруженные) с помощью инструментов Data Mining.

Данная система направлена на упрощение выполняемой работы путем сокращения тривиальных операций предобработки, визуализации данных, обучения и применения моделей машинного обучения, а также составления репрезентативных отчетов. В первую очередь, система позволяет аналитику загружать, открывать файл для предпросмотра и обрабатывать данные, которые затем сохраняются в файле таблицы Excel. Далее, аналитик может визуализировать обработанные данные с помощью графиков и ознакомиться с результатами автоматической аналитики (какие значения каких признаков положительно влияют на успеваемость студента). Также аналитику предоставляется возможность обучить модель машинного обучения на предварительно подготовленных данных, а затем ввести значения параметров, характеризующих некоторых студентов, и получить прогнозный ответ. Наконец, аналитик может составить отчет, содержащий в себе статистические данные по каждому признаку. На рисунке 114 представлена диаграмма возможностей пользователя.

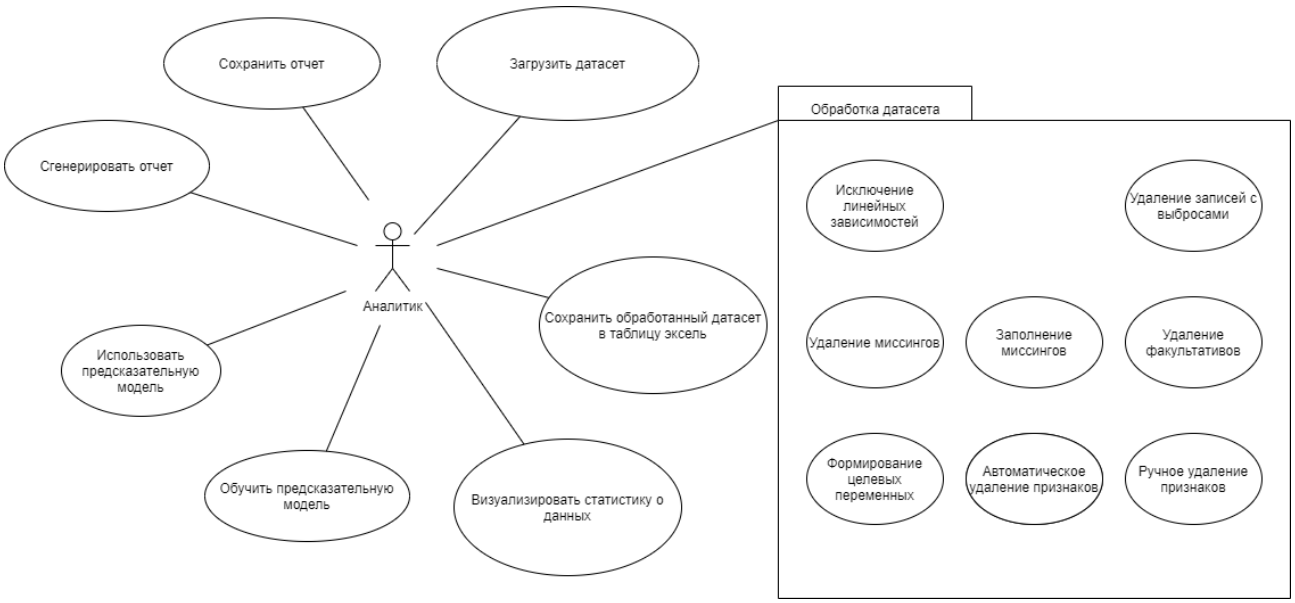


Рисунок 114. Диаграмма возможностей пользователя

В таблице 4 представлено ранжирование возможностей пользователя системы.

Таблица 4. Ранжирование возможностей пользователя

Ранг	Возможности пользователя
1	Обработать датасет
2	Визуализировать статистику данных
3	Загрузить датасет
4	Сгенерировать отчет
5	Сохранить отчет
6	Сохранить обработанный датасет в Excel таблице формате xlsx
7	Обучить предсказательную модель
8	Применить обученную модель для предсказания класса успешности обучения студента
8.1	Ввести значения признаков, на основе которых будет совершаться предсказание

3.2. Функциональное моделирование процесса

Задачей разрабатываемой программной системы является автоматизация выполнения некоторых работ аналитика. В список таких работ входит обработка данных, визуализация данных, составление отчетов, построение модели машинного обучения для дальнейшего прогнозирования ключевых характеристик исследуемого объекта. Ниже продемонстрированы диаграммы IDEF0 и IDEF3, демонстрирующие бизнес-процессы, протекающие в веб-приложении и взаимодействие аналитика с ним. На рисунке 115 представлена диаграмма первого уровня.

Рисунок 115. Диаграмма IDEF0 первого уровня

Вход – данные студентов.

Управление:

- Федеральной закон РФ от 27 июля 2006 года №152-ФЗ «О персональных данных»;
- Задание по обработке и анализу данных.

Механизмы:

- Аналитик;
- Веб-приложение как инструмент для решения задач.

Выходы:

- Отчет статистики данных студентов;
- Обученная на данных предсказательная модель для прогнозирования успеваемости студентов;
- Визуализированная статистика в веб-приложении.

На рисунке 116 представлена диаграмма второго уровня, декомпозирующая диаграмму первого уровня. На ней представлены пять основных этапов работы с данными, которые аналитик совершает при взаимодействии с веб-приложением.

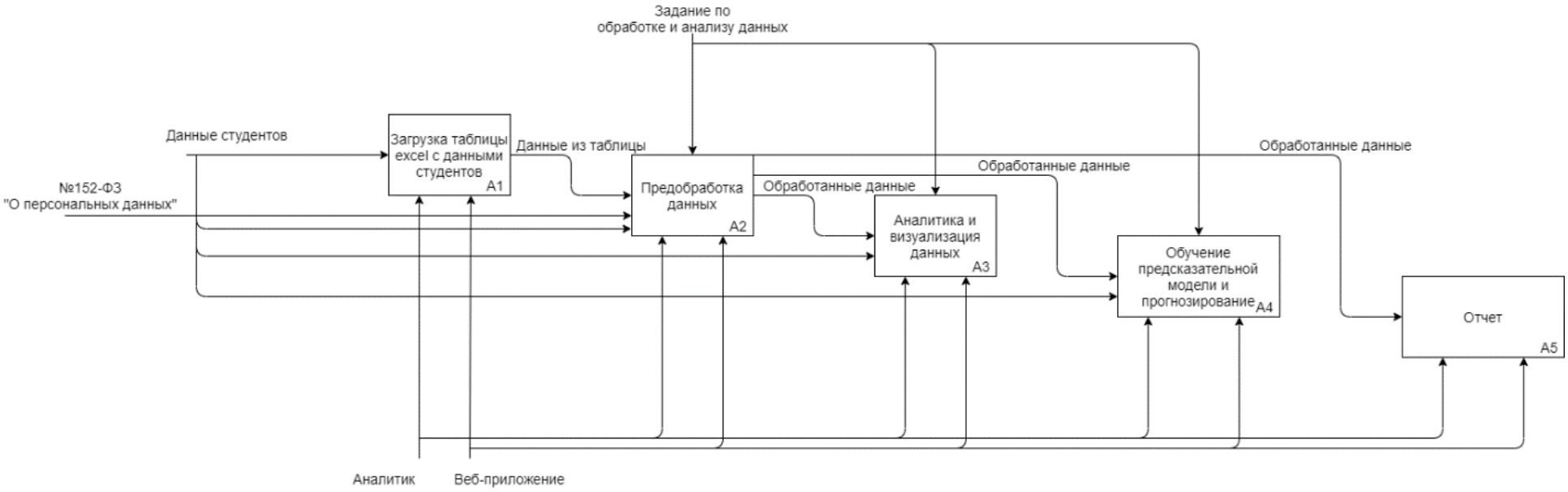


Рисунок 116. Диаграмма IDEF0 второго уровня

Рисунки 117-121 являются декомпозициями этапов A1-A5 соответственно с применением нотации IDEF3. Данная нотация была выбрана в связи с необходимостью использования логических блоков. Блок A1 отображает процесс загрузки данных на сервер. Блок A2 отображает процесс предобработки данных. На блоке A3 представлен процесс визуализации анализа данных и формирование гипотез. Блок A4 моделирует процесс обучения предсказательной модели на выбранных данных и позволяет выполнить прогноз введенных данных. Наконец, блок A5 отображает процесс генерации pdf-отчета, содержащего в себе статистику по выбранному датасету.

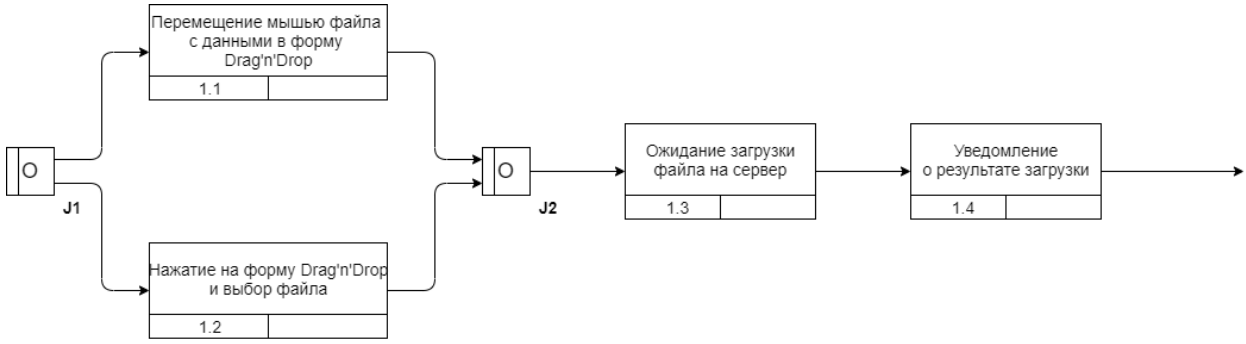


Рисунок 117. Диаграмма IDEF3 (блок A1)

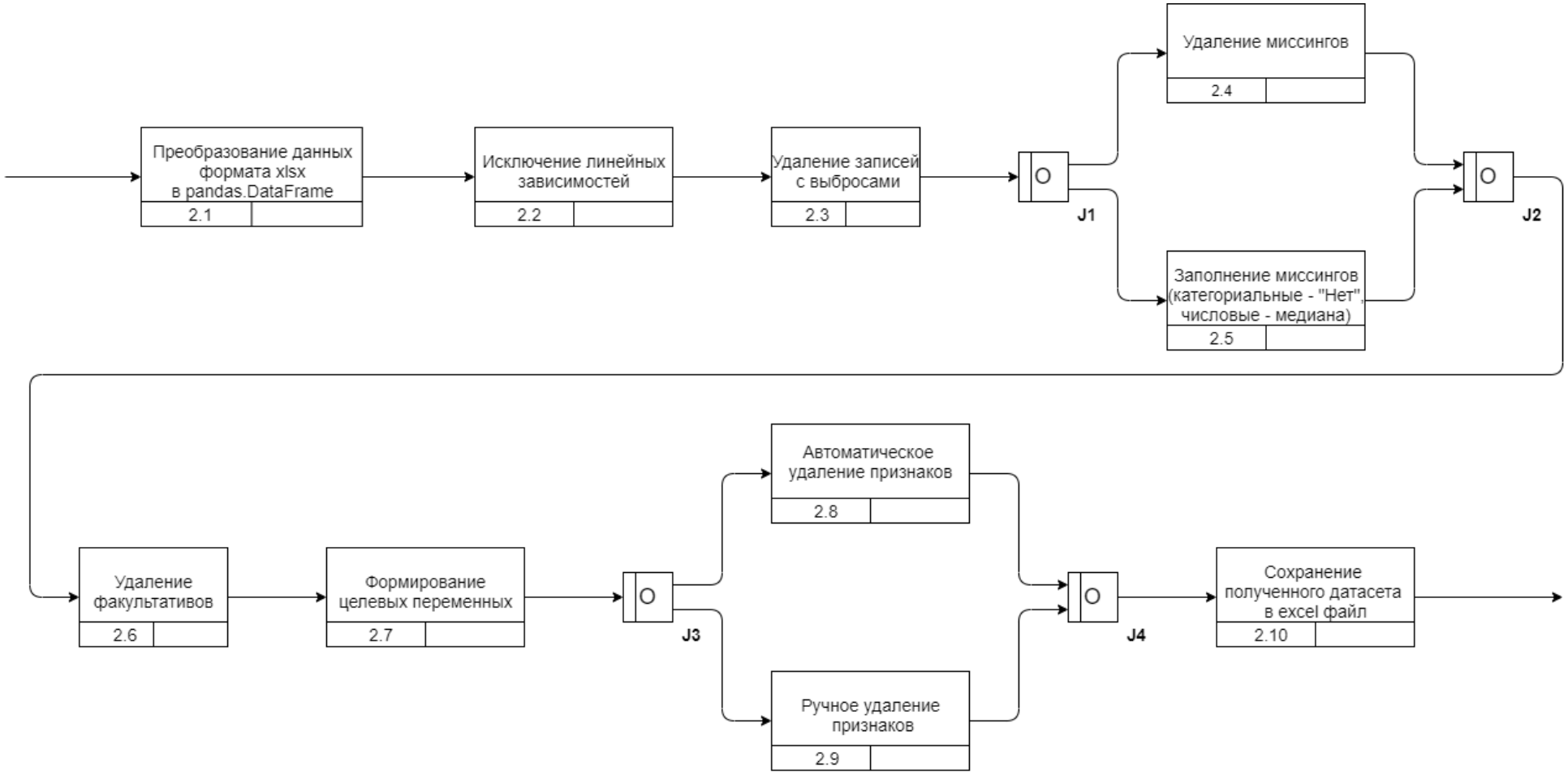


Рисунок 118. Диаграмма IDEF3 (блок A2)

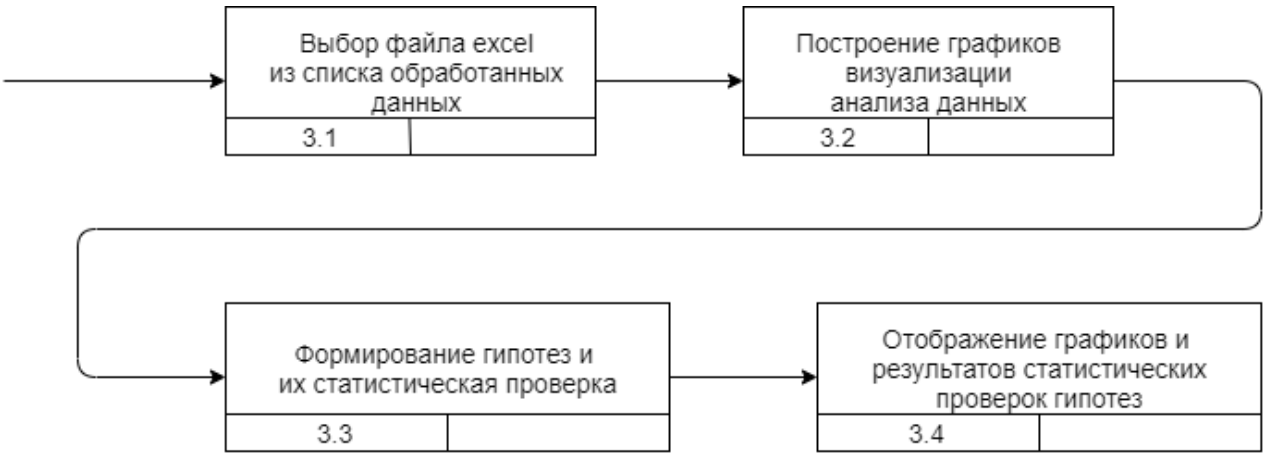


Рисунок 119. Диаграмма IDEF3 (блок A3)

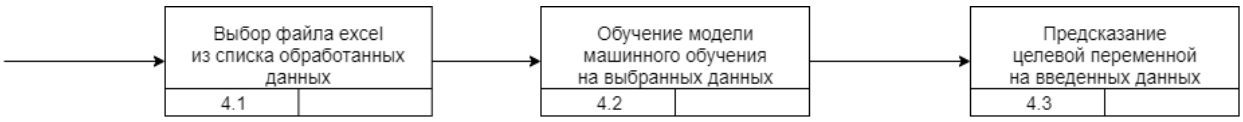


Рисунок 120. Диаграмма IDEF3 (блок A4)

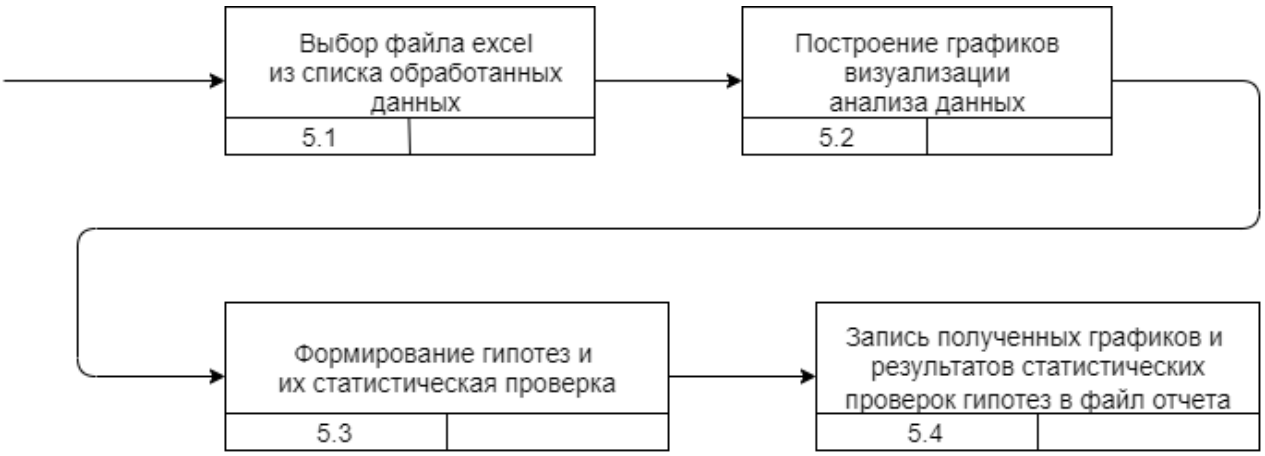


Рисунок 121. Диаграмма IDEF3 (блок A5)

Помимо диаграмм в нотации IDEF0 и IDEF3, были построены диаграммы последовательностей, наглядно представляющие поведение системы при заданных сценариях использования. На рисунках 122-126 представлены пять основных процессов, протекающих в веб-приложении при взаимодействии с пользователем.

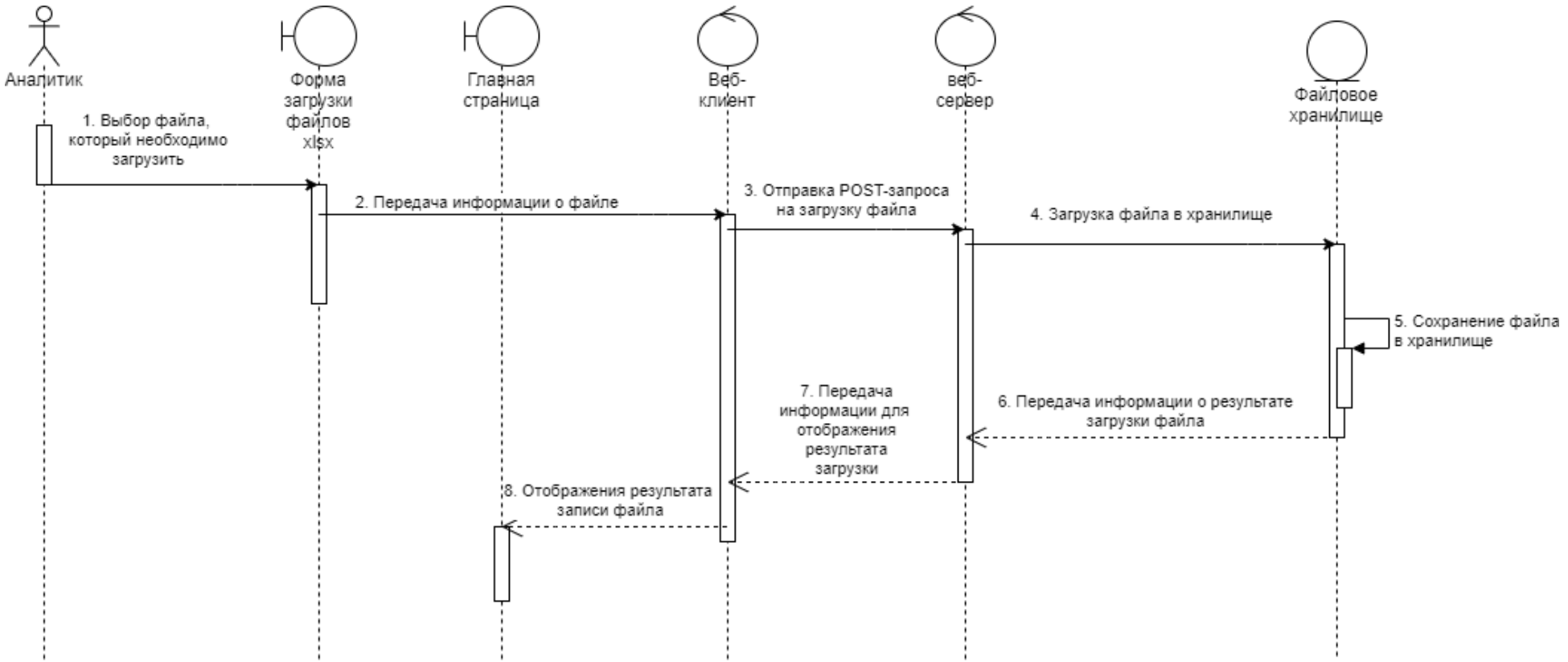


Рисунок 122. Процесс загрузки данных на сервер

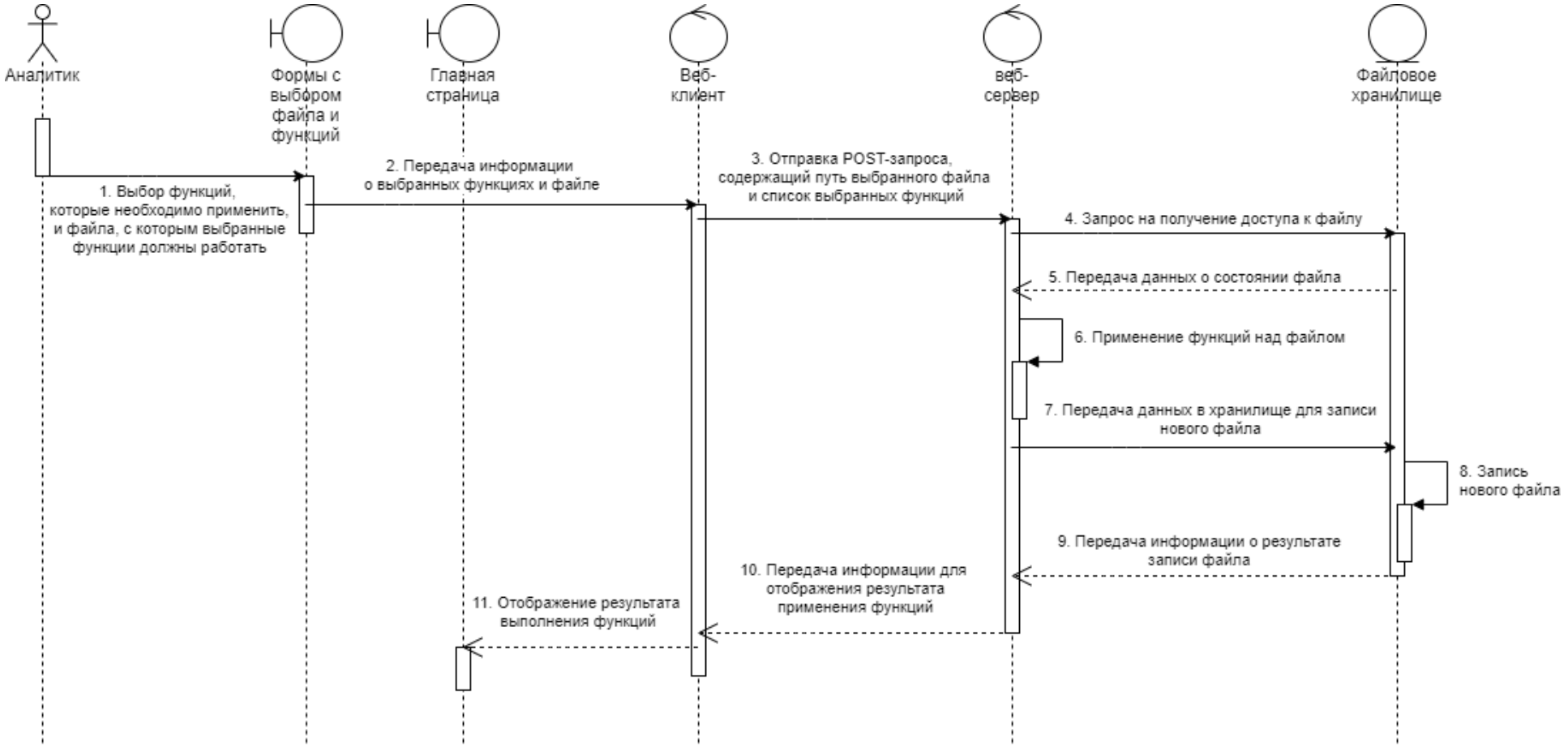


Рисунок 123. Процесс предобработки данных для дальнейшей работы с ними

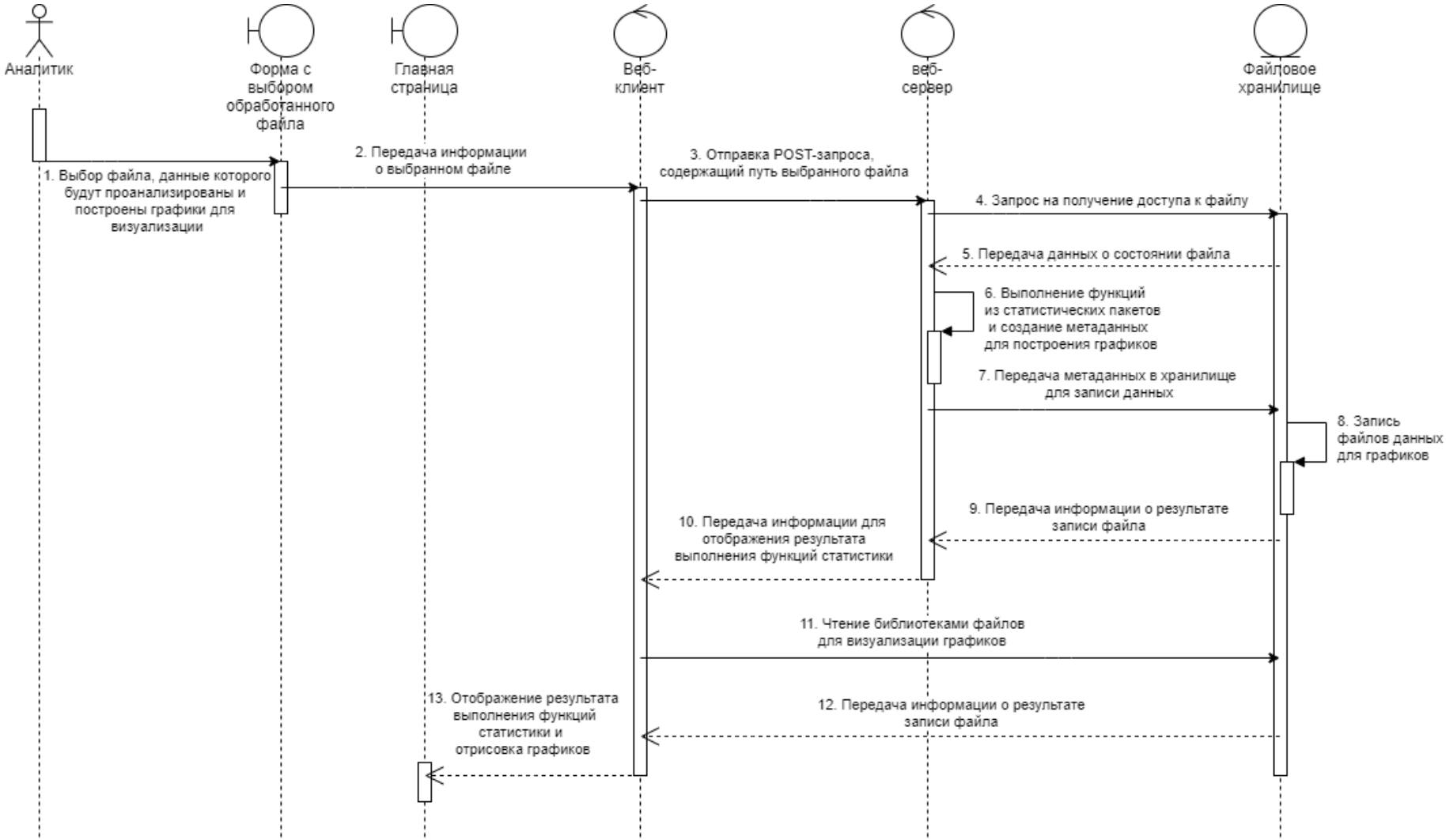


Рисунок 124. Процесс аналитики и визуализации статистики в веб-приложении

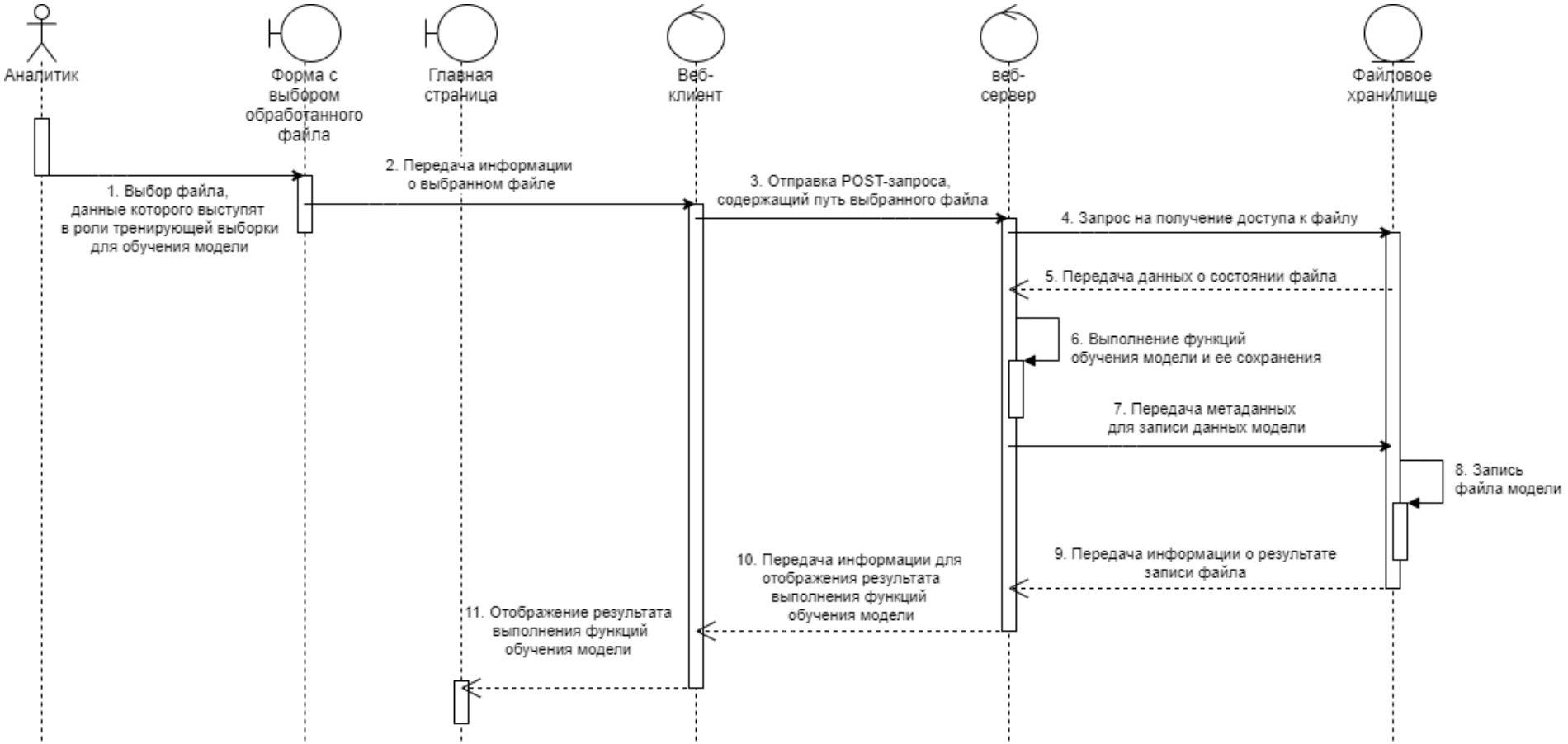


Рисунок 125. Процесс обучения предсказательной модели

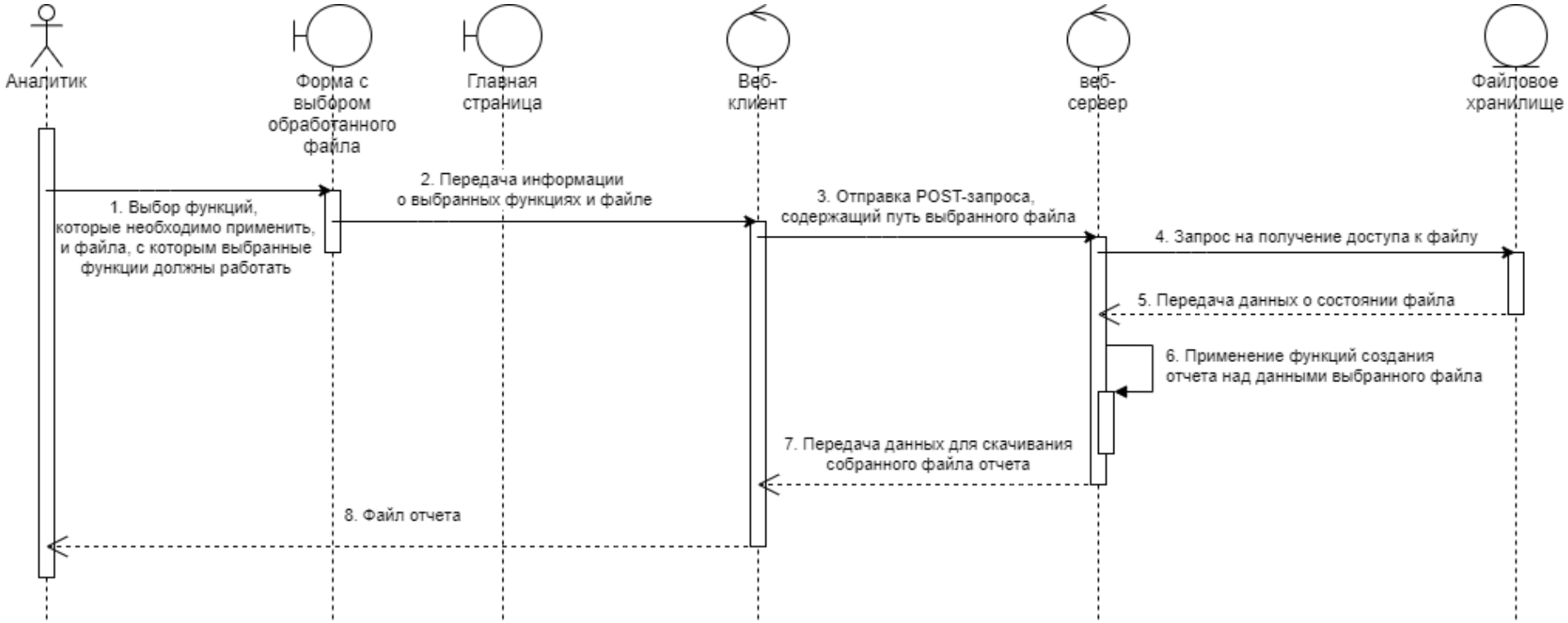


Рисунок 126. Процесс генерации отчета и его скачивание на ПК пользователя

3.3. Моделирование потоков данных программной системы

Для описания потоков данных проектируемой ИС разработаны диаграммы потоков данных с использованием нотации Data Flow Diagram (DFD). Данная нотация предназначена для моделирования ИС с использованием следующих элементов: внешняя сущность, процесс, хранилище данных и поток данных.

На рисунке 127 представлена общая диаграмма потоков данных, описывающая потоки данных между проектируемой системой и внешней сущностью – аналитиком.

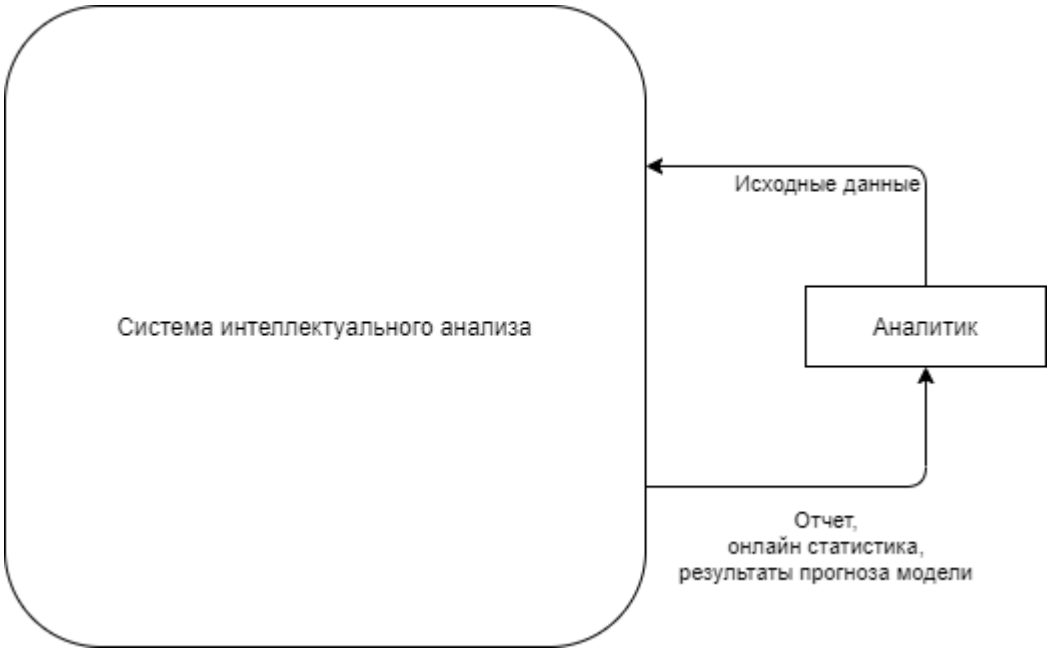


Рисунок 127. контекстуальная диаграмма потоков данных

На рисунке 128 представлена декомпозиция контекстуальной диаграммы. На ней раскрыта система интеллектуального анализа: протекающие в ней процессы, потоки данных и хранилища данных.

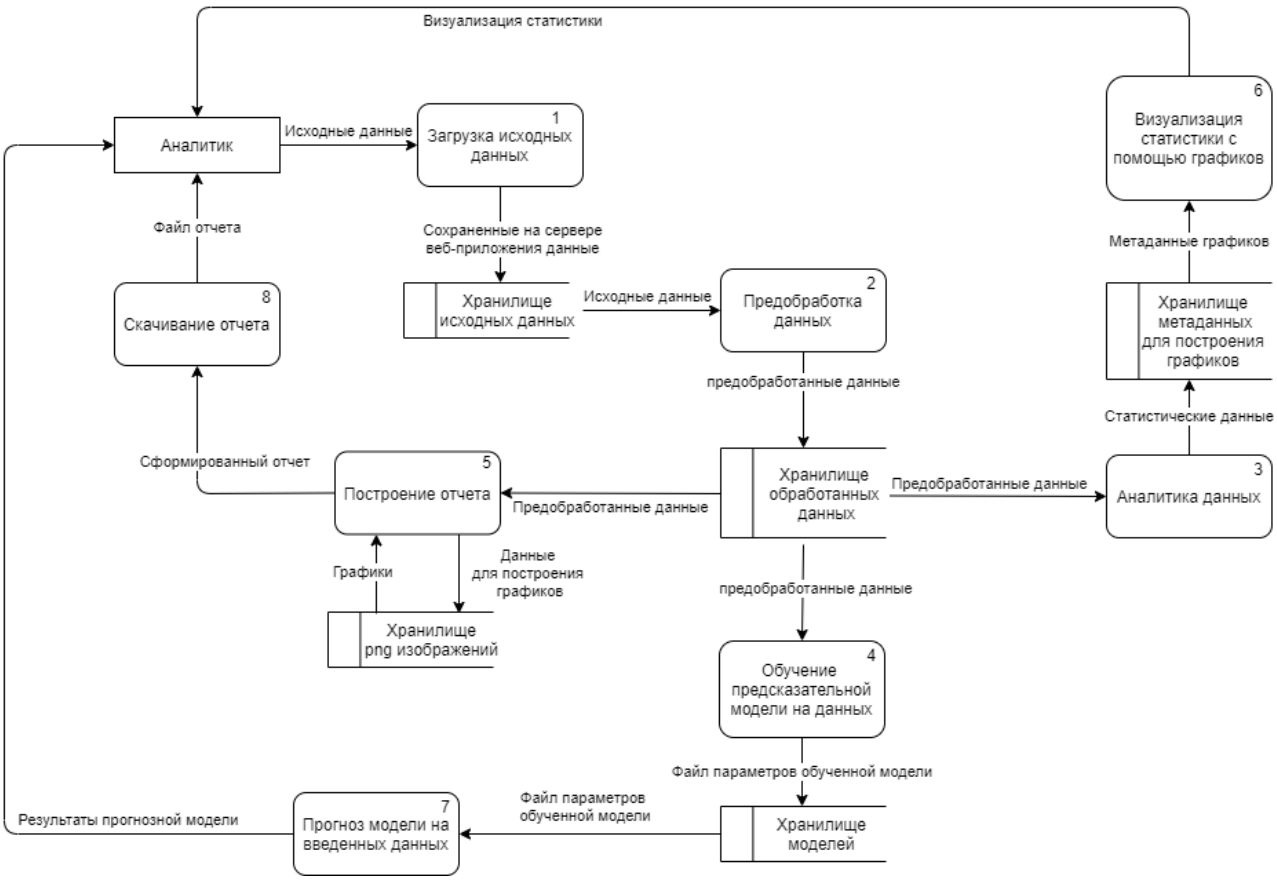


Рисунок 128. детализированная диаграмма потоков данных

На рисунках 129 и 130 ниже представлены декомпозиции второго и третьего процессов. На них более подробно описаны процессы обработки данных и аналитики.

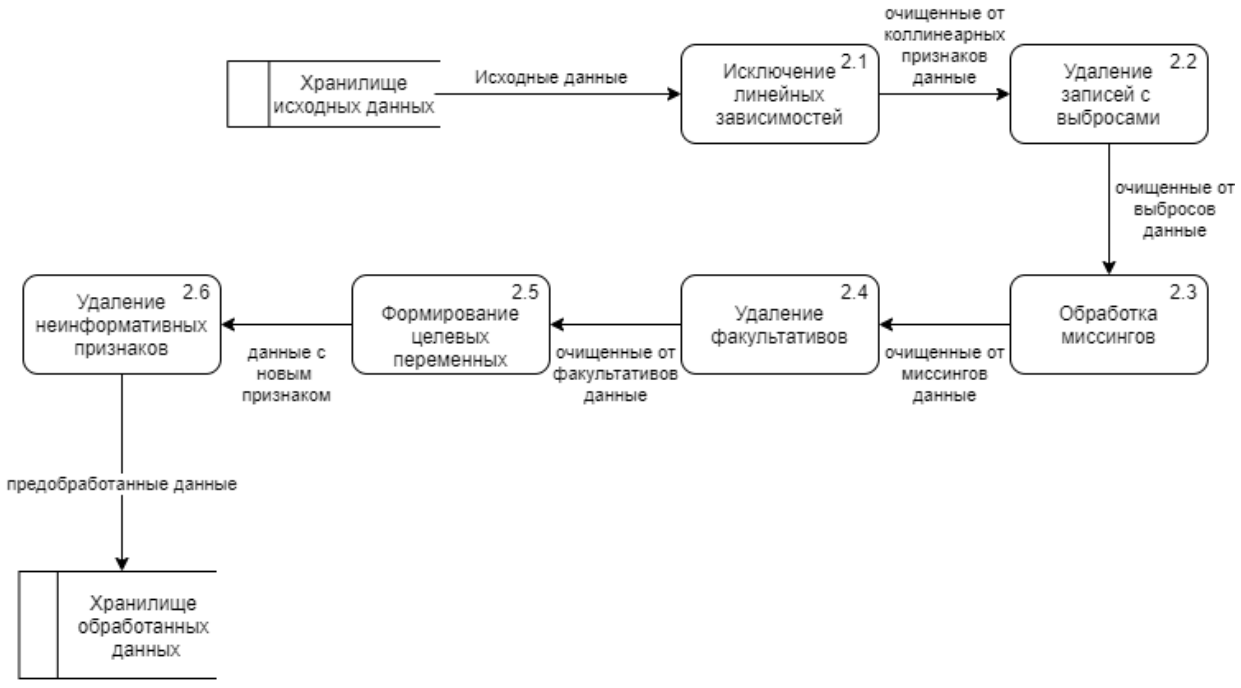


Рисунок 129. Декомпозиция второго процесса

Рисунок 130. Декомпозиция третьего процесса

3.4. Описание объектов системы

Диаграмма классов анализа применяется для описания структуры системы, реализует схему MVC и взаимосвязи между ними. Диаграмма классов анализа продемонстрирована на рисунке 131.

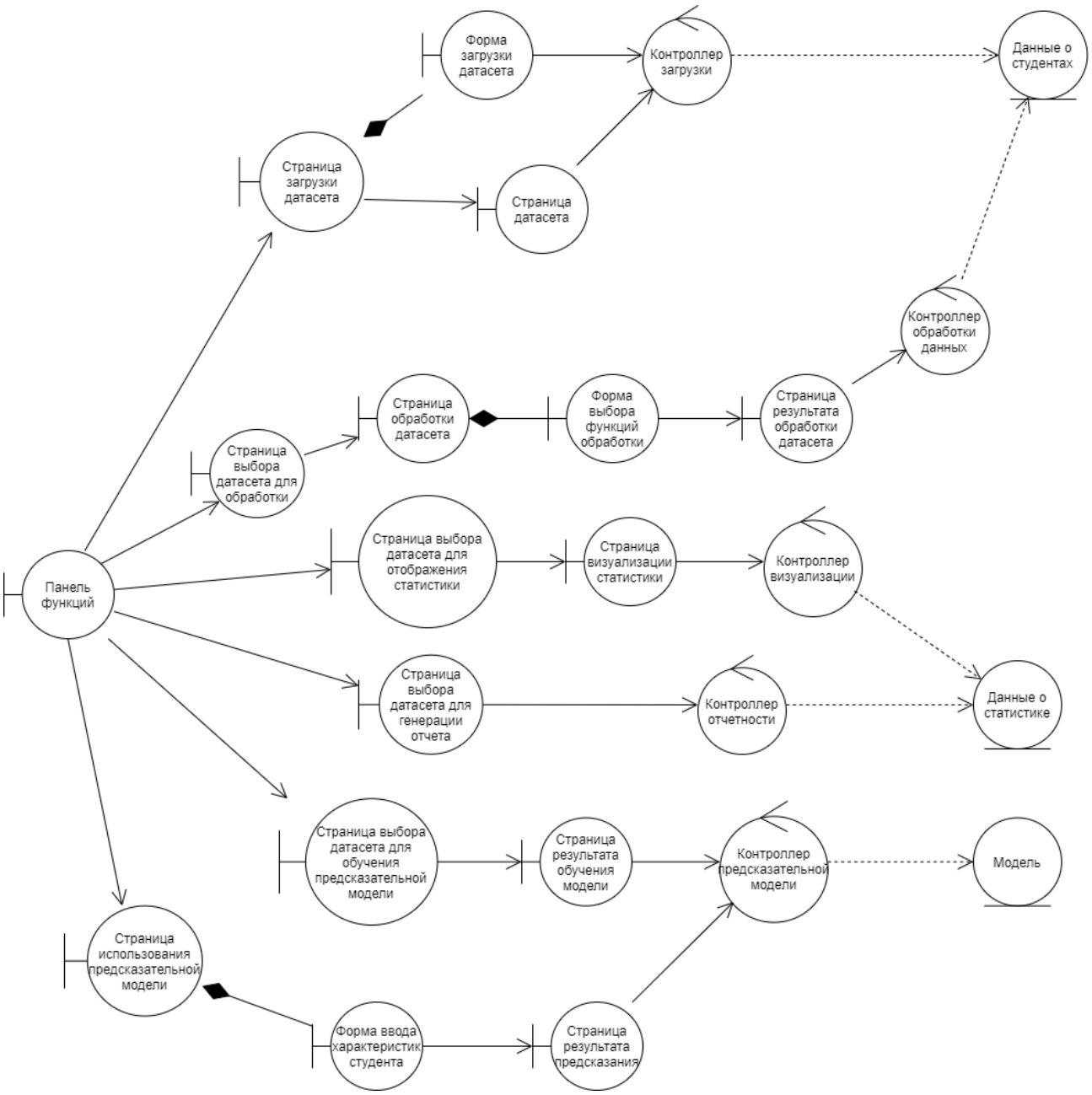


Рисунок 131. Диаграмма классов анализа

Диаграмма компонентов используется для визуализации организации компонентов системы и зависимостей между ними, позволяют получить высокоуровневое представление и компонентах системы. Диаграмма компонентов представлена на рисунке 132.

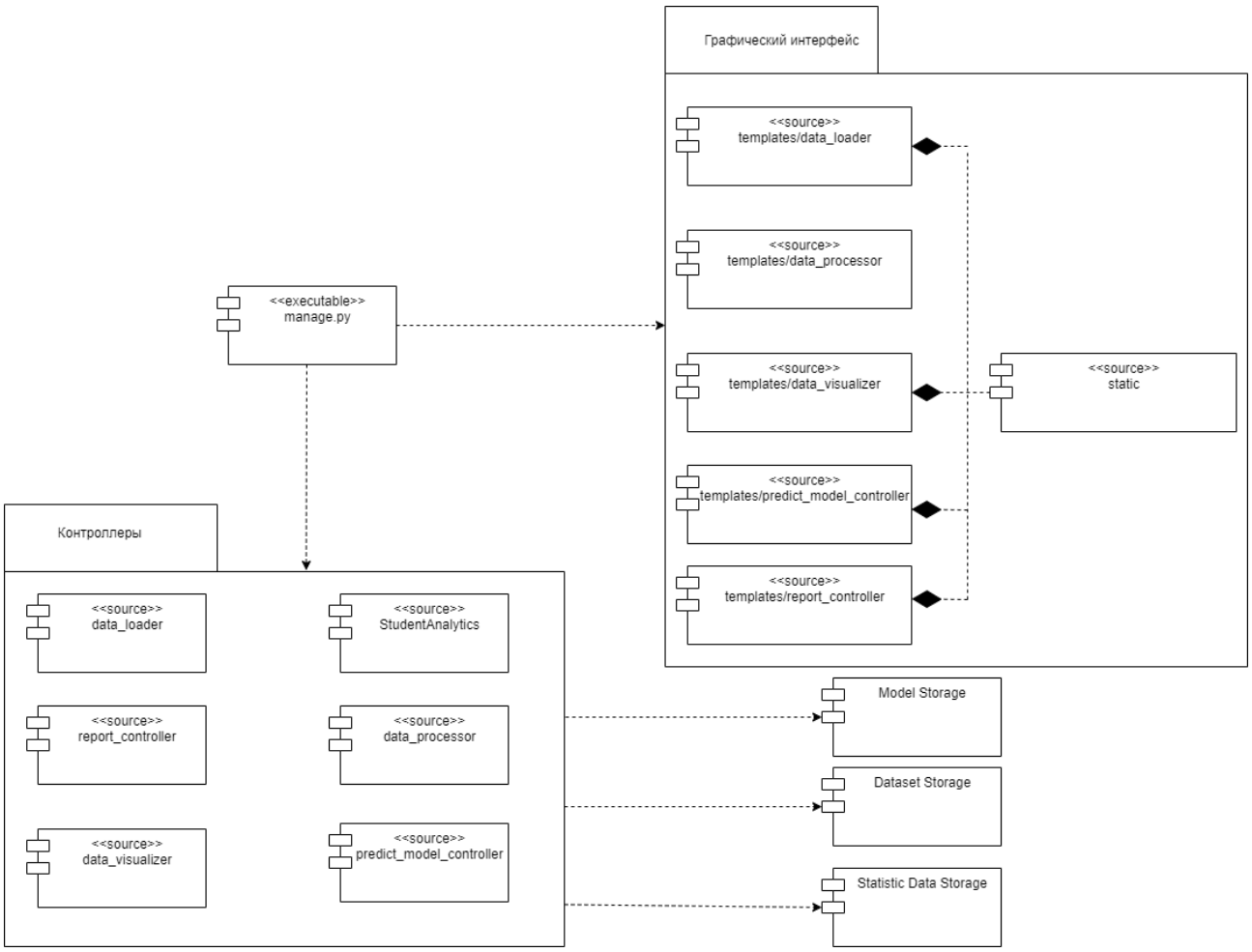


Рисунок 132. Диаграмма компонентов системы

Таким образом, были ИС была спроектирована. Были построены диаграммы возможностей пользователя, IDEF0 и IDEF3, диаграммы последовательностей пяти бизнес-процессов, диаграммы DFD, классов анализа и компонентов.

Глава 4. Разработка информационной системы

4.1. Обоснования выбора программных средств разработки

На первом этапе работы возник вопрос о технологиях, которые будут использоваться для разработки веб-приложения и визуализации статистики.

Соответственно, был проведен морфологический анализ программных платформ (Таблица 5). В качестве вариантов рассматривались фреймворки Django, Flask и Spring.

Таблица 5. Морфологический анализ программных платформ.

Метрика	**Вес метрики (макс. 10)**	**Django**	**Flask**	**Spring**
Встроенный функционал	0,2	9	5	7
Встраивание библиотек для обработки данных	0,4	10	10	6
Производительность	0,3	9	9	7
Документация	0,08	10	7	10
Безопасность	0,02	10	8	9
Итого	**1**	**9,5**	**8,42**	**6,88**

Таким образом, качестве программной платформы был выбран фреймворк Django. Данная технология обладает мощными защитой и встроенным функционалом (для проекта важны использование встроенных форм и собственный шаблонизатор), поддержкой библиотек для обработки данных на скриптовом языке Python. Кроме того, Django имеет развитое сообщество и подробную документацию.

Также был проведен морфологический анализ средств визуализации статистики (Таблица 6). Для сравнения были выбраны JavaScript-библиотеки ChartJS и Morris, а также Python-библиотека Matplotlib.

Таблица 6. Морфологический анализ средств визуализации.

Метрика	Вес метрики (макс. 10)	ChartJS	Morris	Matplotlib
Легкость освоение	0,2	8	8	7
Совместимость с современными браузерами	0,02	9	9	10
Документация	0,08	9	8	10
Интерактивность	0,4	10	9	2
Разнообразие графиков	0,3	9	7	9
Итого	**1**	**9,2**	**8,12**	**5,9**

Таким образом, качестве средства визуализации была выбрана JavaScript-библиотека ChartJS. Она достаточно легка в освоении, к ней приложена исчерпывающая документация, а также с помощью нее можно строить множество разнообразных интерактивных графиков.

Помимо выбранной с помощью морфологического анализа библиотеки, также дополнительно использовались некоторые Python-библиотеки (Таблица 7) и JavaScript-библиотеки (Таблица 8).

Таблица 7. Python-библиотеки.

Библиотека	Версия	Назначение
NumPy	1.20.3	Библиотека, предоставляющая реализации вычислительных алгоритмов, оптимизированные для работы с многомерными массивами.
Pandas	1.2.4	Библиотека, предоставляющая специальные структуры данных и операции для манипулирования числовыми таблицами и временными рядами.
Matplotlib	3.4.2	Библиотека, визуализирующая данные двухмерной и трехмерной графикой.
Xlrd	2.0.1	Библиотека для чтения данных и форматирования информации из файлов Excel в формате .xls.
Openpyxl	3.0.7	Библиотека для чтения данных и форматирования информации из файлов Excel в формате .xlsx/.xlsm/xltm.
Joblib	1.0.1	Библиотека, записывающая данные в формат .pkl.
Reportlab	3.5.67	Библиотека для создания PDF-файлов и графики.
Scikit-learn	0.24.2	Библиотека машинного обучения для программирования на Python.

Продолжение таблицы 7.

Библиотека	Версия	Назначение
Scipy	1.6.3	Библиотека, содержащая функции математики, естественных наук и инженерии.
Seaborn	0.11.1	Библиотека для создания статических графиков на Python.

Таблица 8. JavaScript-библиотеки.

Библиотека	Версия	Назначение
DropZone	5.7.0	Библиотека, реализующая Drag’n’Drop-функционал загрузки файла
ChartJS	3.2.1	Библиотека, отрисовывающая интерактивные графики
AnychartJS	8.9.0	Библиотека, отрисовывающая интерактивные графики (использовалась для построения диаграмм размаха)
Bootstrap	4.6.8	Библиотека, связывающая компоненты Bootstrap.css и jQuery
jQuery.dataTables	3.6.0	Библиотека, отрисовывающая интерактивные таблицы

4.2. Разработка серверной части приложения

Django – высокоуровневый Python веб-фреймворк, который позволяет быстро создавать безопасные и поддерживаемые веб-приложения [5].

Код Django написан в соответствии с использованием принципов и шаблонов проектирования, которые поощряют создание поддерживаемого и повторно используемого кода. В частности, используется принцип DRY, что сокращает избыточное дублирование и, соответственно, объем кода. Django группирует связанные функциональные возможности в повторно используемые «приложения» и связанный код в модули (на более низком уровне).

Работа фреймворка устроена следующим образом. Соответствующее веб-приложение ожидает HTTP-запросы от веб-клиента. Когда запрос получен, приложение выполняет бизнес-логику на основе URL-адреса и данных в POST и GET запросах. В зависимости от функционала приложения, может происходить

чтение или запись информации из базы данных. Затем приложение возвращает ответ клиенту, статически или динамически создавая HTML-страницу для отображения в браузере. Веб-приложения Django группируют код в отдельные файлы, представленные на рисунке 133.

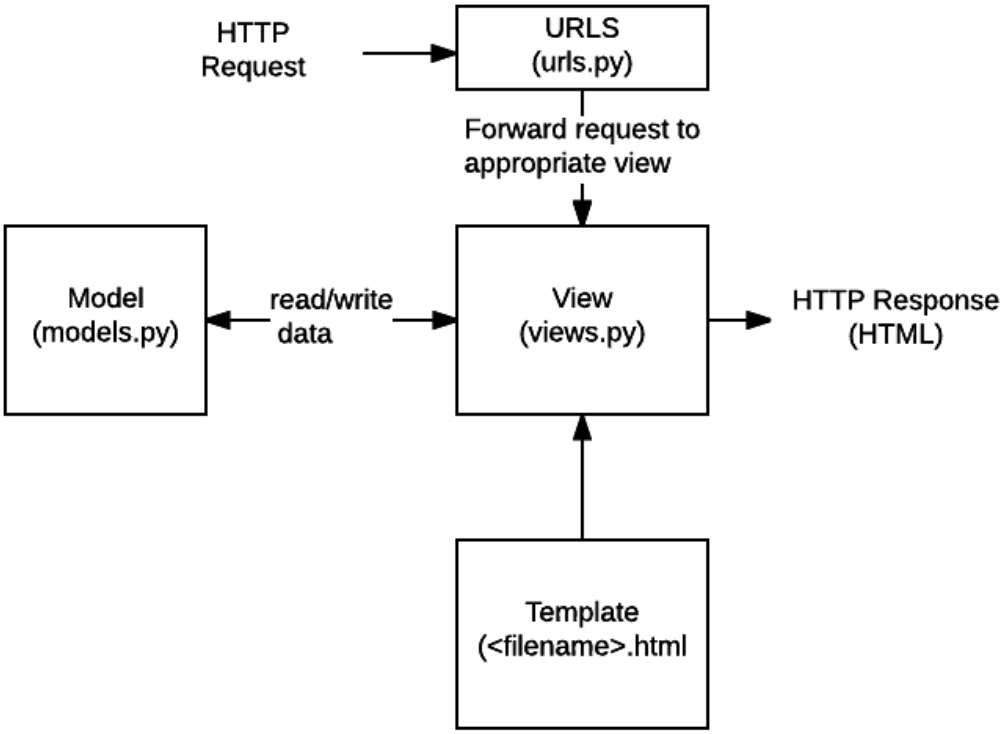


Рисунок 133. Схема приложения Django

Django реализует архитектуру «Model View Template», которая является разновидностью архитектуры «Model View Controller» и подразумевает разделение программы на три слабосвязанных компонента, каждый из которых отвечает за свою сферу деятельности.

- URLs – URL-маршрутизатор может извлекать данные из URL-адреса в соответствии с заданным шаблоном и передавать их в соответствующую функцию отображения в виде аргументов;
- Views – функция обработчика запросов, которая получает HTTP-запросы и возвращает ответы. Функция имеет доступ к данным, необходимым для удовлетворения запросов, и делегирует ответы в шаблоны через модели;
- Models – Модели представляют собой объекты Python, которые определяют структуру данных приложения и предоставляют механизмы для управления и выполнения запросов в базу данных;

- Templates – Шаблоны представляют собой текстовые файлы, определяющие структуру или разметку страницы, с полями для подстановки, которые используются для вывода актуального содержимого.

Структура проекта веб-приложения представлена на рисунке 134.

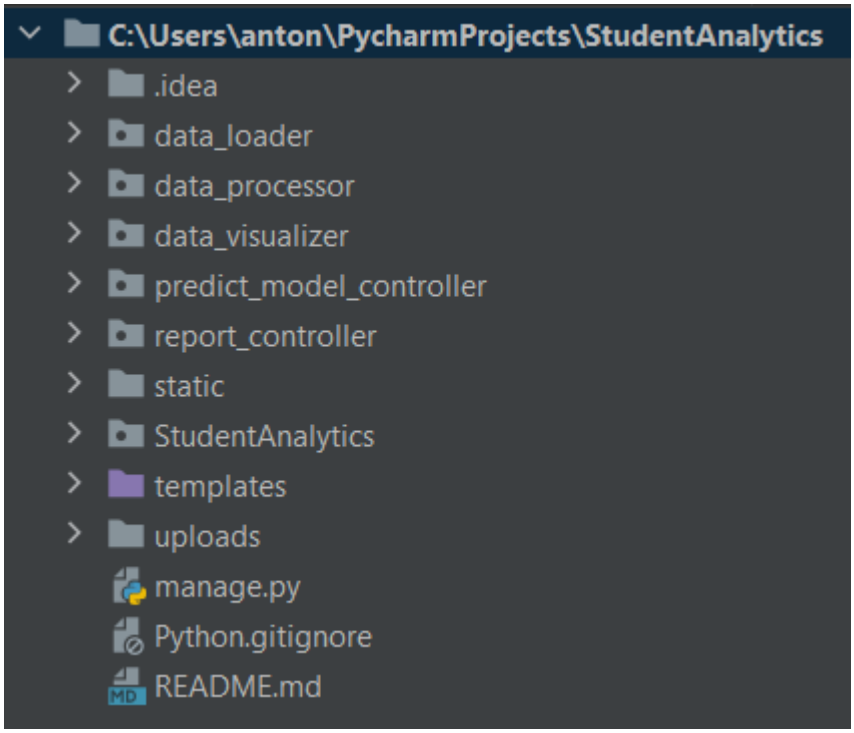


Рисунок 134. Структура проекта веб-приложения

Веб-приложение состоит из одноименного подкаталога и пять приложений Django, разделяющих между собой всю бизнес-логику веб-приложения:

- StudentAnalytics – подкаталог, содержащий настройки всего веб-приложения, а также связывающий отдельные приложения Django воедино. Структура подкаталога представлена на рисунке 135;
- data_loader – приложение, отвечающее за загрузку датасетов в файловую систему сервера;
- data_processor – приложение, отвечающее за обработку данных;
- data_visualizer – приложение, отвечающее за визуализацию статистики по датасету на странице веб-приложения;

- predict_model_controller – приложение, отвечающее за взаимодействие с предиктивной моделью;
- report_controller – отвечающее за генерацию отчета по выбранному набору данных.

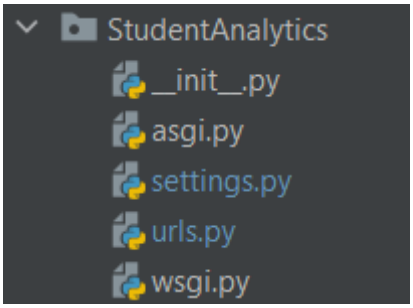


Рисунок 135. Структура подкаталога StudentAnalytics

В файле settings.py находятся настройки всего веб-приложения. Здесь были дополнительно внесены созданные Django-приложения в список INSTALLED_APPS (рисунок 136) и директория «static» в список STATICFILES_DIRS (рисунок 137).

```
# Application definition

INSTALLED_APPS = [
    'django.contrib.admin',
    'django.contrib.auth',
    'django.contrib.contenttypes',
    'django.contrib.sessions',
    'django.contrib.messages',
    'django.contrib.staticfiles',
    'data_loader',
    'data_processor',
    'data_visualizer',
    'predict_model_controller',
    'report_controller',
]
```

Рисунок 136. Список установленных приложений в файле настроек

```
STATICFILES_DIRS = [
    os.path.join(BASE_DIR, 'static'),
]
```

Рисунок 137. Список директорий со статическими файлами

В файле urls.py подкаталога потребовалось указать все файлы urls.py в созданных Django-приложениях. Содержание файла представлено на рисунке 138.

```python
from django.conf.urls import include, url

urlpatterns = [
    url(r'^', include('data_loader.urls', namespace='data_loader')),
    url(r'^', include('data_processor.urls', namespace='data_processor')),
    url(r'^', include('data_visualizer.urls', namespace='data_visualizer')),
    url(r'^', include('predict_model_controller.urls', namespace='predict_model_controller')),
    url(r'^', include('report_controller.urls', namespace='report_controller')),
]
```

Рисунок 138. Список urlpatterns в файле urls.py

Структура типичного Django-приложения представлена на рисунке 139. В файле forms.py реализованы классы форм, наследованные от Django.forms.Form. В файле urls.py – список URL-адресов, созданных с помощью регулярных выражений, и соответствующих им функций из views.py. Функции в views.py инкапсулируют бизнес-логику приложения и возвращают результат выполнения render(request). Функция render получает на вход параметр request, шаблон и аргументы шаблонизатора Django и генерирует и возвращает веб-страницу.

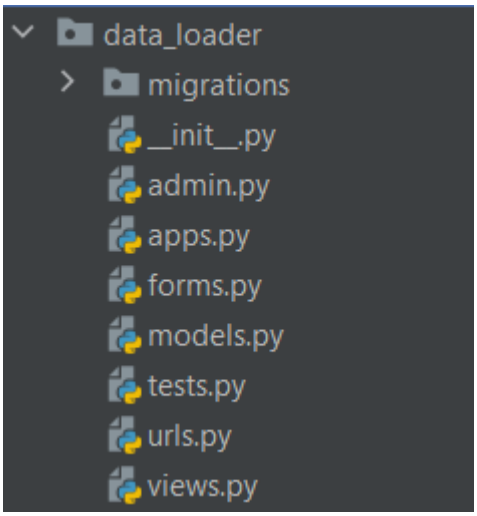


Рисунок 139. Структура Django-приложения

Приложение data_loader содержит форму для загрузки файла (рисунок 140). Данная форма связана с JS-библиотекой DropZone, которая позволяет

загружать файлы способом Drag’n’Drop и не перезагружать страницу после загрузки.

```python
from django import forms


class UploadFileForm(forms.Form):
    file = forms.FileField()
```

Рисунок 140. Класс формы для загрузки файла

Листинг файла views.py представлен в Приложении А. Процесс загрузки файла представлен на диаграмме IDEF3, блок А1 (рисунок 117).

Файл urls.py содержит адреса страниц загрузки датасета, собственной страницы датасета (взятых из списка загруженных или обработанных наборов данных), а также адрес для обработки формы (рисунок 141).

```python
from django.conf.urls import url
from . import views

app_name = 'data_loader'

urlpatterns = [
    url('upload', views.filelist, name='upload_filelist'),
    url(r'^$', views.filelist, name='filelist'),
    url(r'^dataset/(?P<file_path>.+)', views.dataset, name='dataset'),
    url(r'^processed_dataset/(?P<file_path>.+)', views.processed_dataset, name='processed_dataset'),
]
```

Рисунок 141. Список URL-адресов в приложении data_loader

Приложение data_processor содержит 2 класса форм: DatasetForm, отвечающий за выбор датасета из списка загруженных (рисунок 142), и DataProcessForm, позволяющий выбрать функции обработки данных.

```python
class DatasetForm(forms.Form):
    dataset_list = os.listdir(path='uploads/datasets')
    dataset_choices = zip(dataset_list, dataset_list)

    dataset = forms.ChoiceField(choices=dataset_choices, widget=forms.RadioSelect(
        attrs={"class": "radio dataprocess_radio todo-list"}))
```

Рисунок 142. Класс формы для загрузки файла

Листинг файла views.py с функциями обработки набора данных представлен в Приложении 1. Процесс обработки представлен на диаграмме IDEF3, блок А2 (рисунок 118).

Файл urls.py содержит адреса страниц выбора датасета, страницы выбора функций обработки, результата обработки, а также URL для форм выбора датасета и функций обработки. (рисунок 143).

```
from django.conf.urls import url
from . import views

app_name = 'data_processor'

urlpatterns = [
    url('^form_dataset_process$', views.dataset_process, name='upload_dataset_process'),
    url(r'^dataset_process$', views.dataset_process, name='dataset_process'),
    url('^form_dataprocess/(?P<file_name>.+)$', views.dataprocess, name='upload_dataprocess'),
    url(r'^dataprocess/(?P<file_name>.+)$', views.dataprocess, name='dataprocess'),
    url(r'^process_result$', views.process_result, name='process_result'),
]
```

Рисунок 143. Список URL-адресов в приложении data_processor

Приложение data_visualizer содержит класс формы DatasetForm для выбора набора данных для визуализации статистики (рисунок 144).

```
class DatasetForm(forms.Form):
    dataset_list = os.listdir(path='uploads/datasets')
    processed_dataset_list = os.listdir(path='uploads/processed_datasets')
    dataset_choices = zip(dataset_list, dataset_list)
    processed_dataset_choices = zip(processed_dataset_list, processed_dataset_list)

    dataset = forms.ChoiceField(choices=dataset_choices, widget=forms.RadioSelect(
        attrs={"class": "radio dataprocess_radio"}))
    processed_dataset = forms.ChoiceField(choices=processed_dataset_choices, widget=forms.RadioSelect(
        attrs={"class": "radio dataprocess_radio"}))
```

Рисунок 144. Список URL-адресов в приложении data_visualizer

В список urlpatterns приложения входят адреса для выбора набора данных и вывода статистики (на основе обработанных и загруженных данных). Список представлен на рисунке 145.

Листинг файла views.py с функциями визуализации статистики набора данных представлен в Приложении А. Процесс визуализации представлен на диаграмме IDEF3, блок А3 (рисунок 119).

```python
from django.conf.urls import url
from data_visualizer import views

app_name = 'data_visualizer'

urlpatterns = [
    url(r'^dataset_stat$', views.dataset_stat, name='dataset_stat'),
    url(r'^statistics/(?P<file_path>.+)', views.statistics, name='statistics'),
    url(r'^processed_statistics/(?P<file_path>.+)', views.processed_statistics, name='processed_statistics'),
]
```

Рисунок 145. Список URL-адресов в приложении data_visualizer

Приложение predict_model_controller содержит класс формы ModelPredictForm для ввода характеристик студента. Введенные значения используются для прогнозирования успеваемости студента в семестре. В качестве предсказательной модели используется алгоритм на основе метода опорных векторов.

В список URL-адресов приложения включены адреса для страницы выбора датасета для обучения модели, страницы результата обучения, страниц ввода характеристик студента и результата прогнозирования успеваемости. Кроме того, в списке присутствуют адреса для форм выбора набора данных и ввода значений признаков студента (рисунок 146).

Листинг файла views.py с функциями обучения модели и прогнозирования успеваемости студента представлен в Приложении А. Процесс обучения представлен на диаграмме диаграмме IDEF3, блок А4 (рисунок 120).

```python
from django.conf.urls import url
from . import views

app_name = 'predict_model_controller'

urlpatterns = [
    url(r'^svc_model/(?P<file_path>.+)', views.svc_model, name='svc_model'),
    url(r'^dataset_teaching_model$', views.dataset_teaching_model, name='dataset_teaching_model'),
    url(r'^predict_form_upload$', views.predict_form, name='predict_form_upload'),
    url(r'^predict_form$', views.predict_form, name='predict_form'),
    url(r'^model_data_upload$', views.predict_form, name='model_data_upload'),
    url(r'^predict_result$', views.predict_result, name='predict_result'),
]
```

Рисунок 146. Список URL-адресов в приложении predict_model_controller

Приложение report_controller не включает в себя классов форм. В файле urls.py перечислены адреса страниц выбора датасета и результата генерации отчета (рисунок 147).

Листинг файла views.py с функциями генерации представлен в Приложении А. Процесс генерации отчета представлен на диаграмме диаграмме IDEF3, блок А5 (рисунок 121).

```
from django.conf.urls import url
from . import views

app_name = 'report_controller'

urlpatterns = [
    url(r'^dataset_report$', views.dataset_report, name='dataset_report'),
    url(r'^report_result/(?P<file_path>.+)', views.report_result, name='report_result'),
    url(r'^processed_report_result/(?P<file_path>.+)', views.processed_report_result, name='processed_report_result')
]
```

Рисунок 147. Список URL-адресов в приложении report_controller

4.3. Разработка клиентской части приложения

Традиционно программисты стараются отделить бизнес-логику приложения от визуальной части. Для решения этой проблемы были созданы различные шаблонизаторы. В Django таким шаблонизатором является Jinja. Для вывода визуальной информации используется функция render, которая позволяет выводить HTML-шаблоны с вставками из шаблонизатора Jinja.

Все шаблоны хранятся в общей папке templates. Структура шаблонов представлена на рисунке 148.

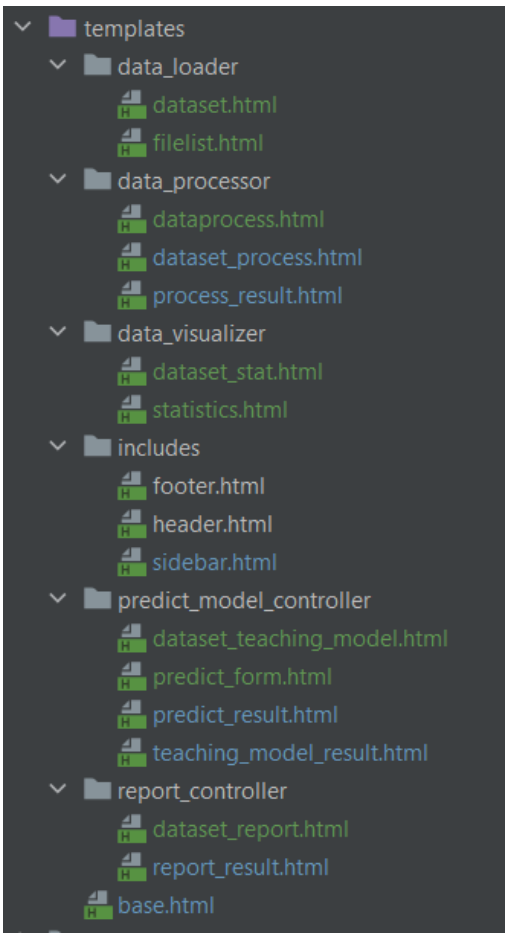


Рисунок 148. Структура шаблонов

- base.html – Базовый шаблон, содержащий тег <head></head>, подключающий общие для всех шаблонов CSS-стили и JavaScript-скрипты;
- includes – Директория, содержащая шаблоны для меню, подвала сайта и боковой панели;
- data_loader – Директория, содержащая шаблоны загрузки и просмотра датасетов;
- data_processor – Директория, содержащая шаблоны для выбора датасета, выбора функций его обработки и результата обработки;
- data_visualizer – Директория, содержащая шаблоны для выбора датасета и его визуализации;
- predict_model_controller – Директория, содержащая шаблоны для выбора датасета, обучения на нем алгоритма SVM [4], результата обучения, а

также шаблоны ввода признаков для прогнозирования и результата прогнозирования;

- report_controller – Директория, содержащая шаблоны для выбора датасета для составления отчёта.

При создании дизайна веб-приложения был использован свободно распространяемый шаблон AdminLTE-master, содержащий варианты исполнения множества элементов интерфейса, таких, как кнопки, поля форм и уведомления.

На рисунке 149 представлена карта веб-приложения. Подробно страницы приложения и переходы между ними описаны ниже.

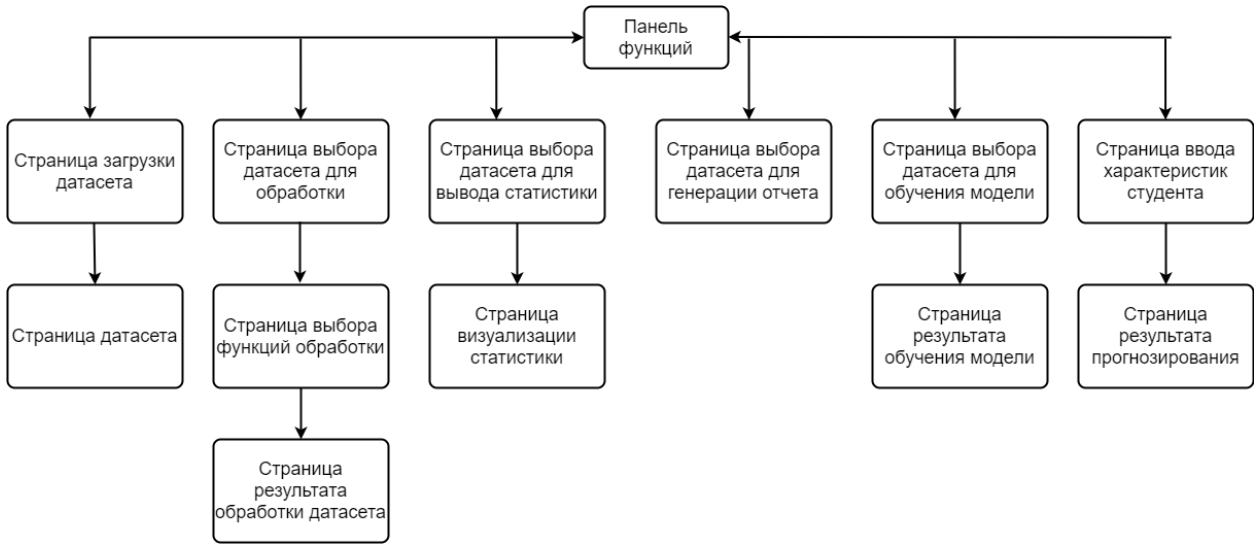


Рисунок 149. Карта веб-приложения

На каждой странице веб-приложения представлена шапка страницы, позволяющая скрыть или развернуть панель функций (рисунок 150) и панель настроек темы (рисунок 151). Панель функций позволяет выбрать страницы просмотра списка датасетов и их загрузки, страницы обработки датасета, страницы вывода статистики и страницы сохранения отчета. Также можно открыть страницы обучения модели и прогнозирования успеваемости конкретного студента на основе его характеристик.

В панели настроек темы можно зафиксировать интерфейс при пролистывании страницы, отобразить страницу в виде ленты, переключить панель функций в минималистичный режим, включить опцию «Открытие

панели функций при наведении мыши», переключения состояния меню настроек темы из «скрывать содержимое страницы» в «не скрывать содержимое страницы», переключения светлой и темной темы меню панели функций, а также выбор темы самого сайта из списка:

- «Синий»;
- «Чёрный»;
- «Фиолетовый»;
- «Зеленый»;
- «Красный»;
- «Жёлтый»;
- «Синий светлый»;
- «Чёрный светлый»;
- «Фиолетовый светлый»;
- «Зеленый светлый»;
- «Красный светлый»;
- «Жёлтый светлый».

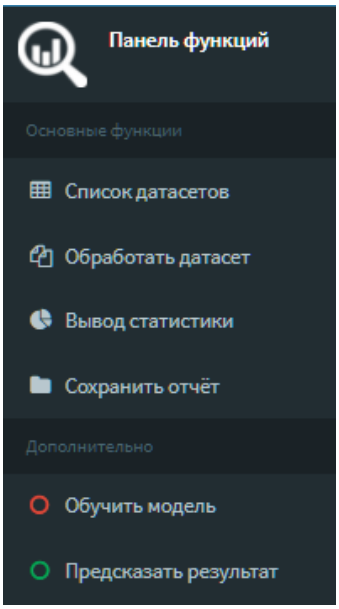


Рисунок 150. Панель функций

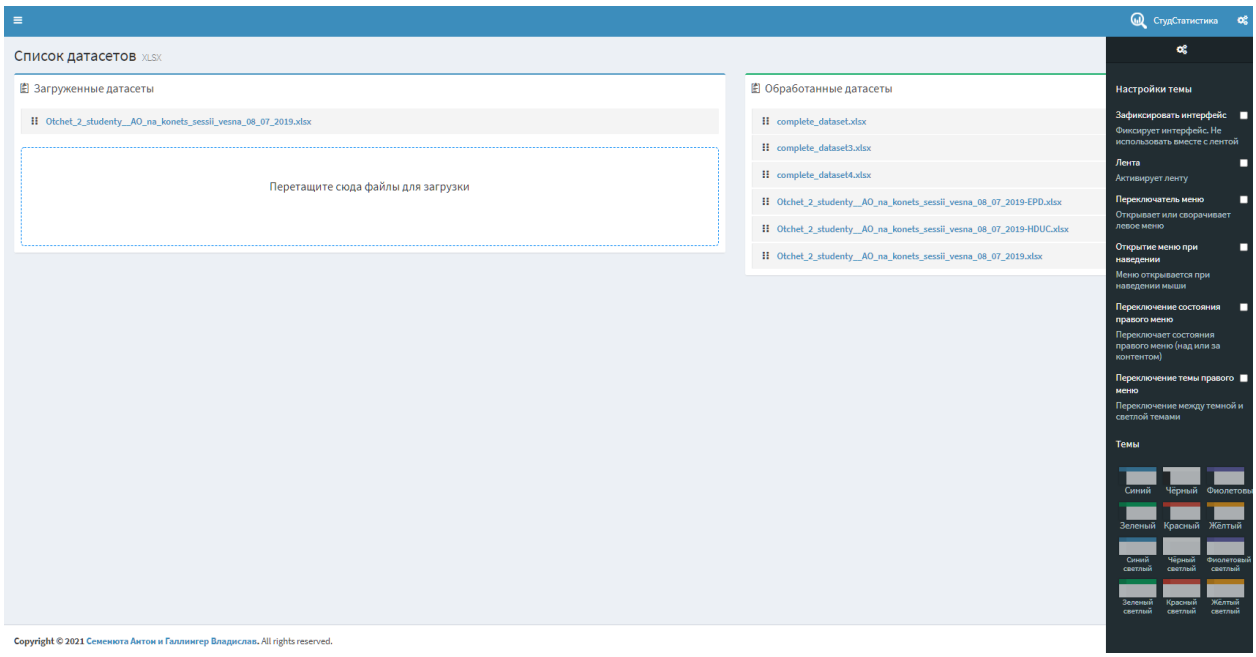


Рисунок 151. Панель настроек темы

В корне сайта находится страница загрузки датасета (рисунок 152). Также сюда ведет пункт «Список датасетов» в панели функций. Здесь можно просмотреть списки загруженных и обработанных датасетов. Также можно загрузить новый датасет (на рисунке 153 представлена форма после загрузки датасета). Данная форма создана с помощью JavaScript-библиотеки DropZone, которая позволяет загружать файлы без перезагрузки страницы. Для этого можно переместить файл мышью в выделенную область или нажать на область и выбрать файл из проводника.

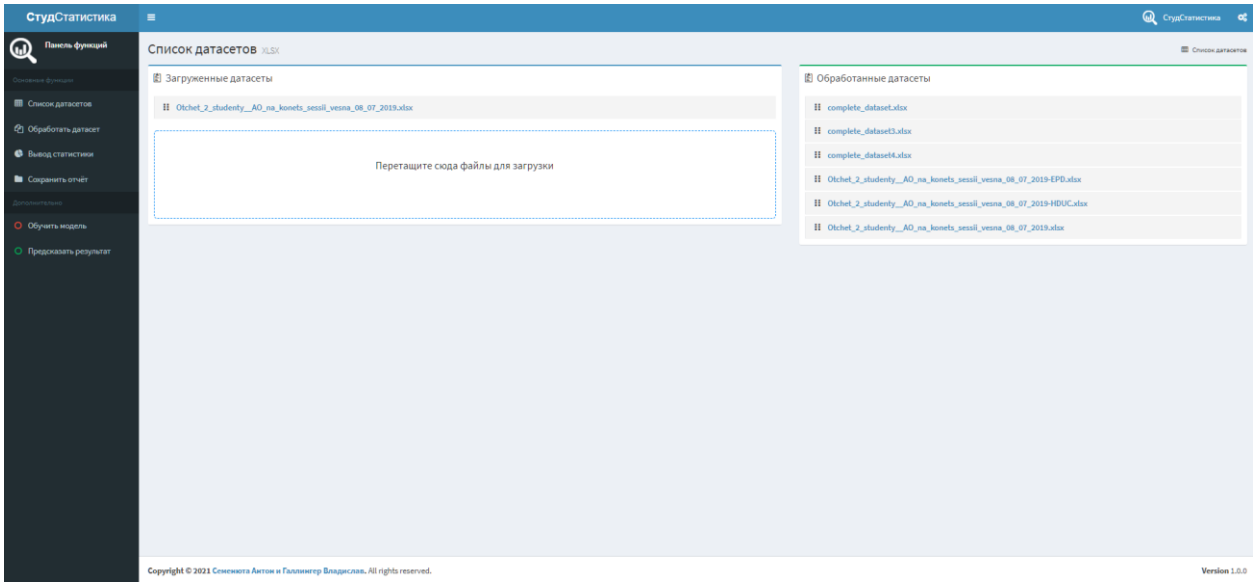


Рисунок 152. Страница загрузки датасета

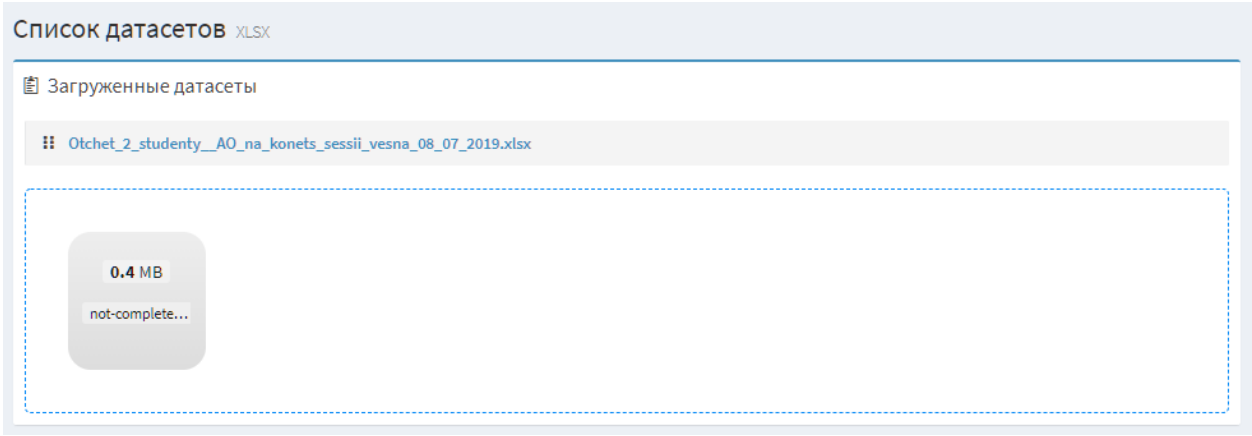


Рисунок 153. Форма после загрузки датасета

После нажатия на один из пунктов списков загруженных и обработанных датасетов, открывается страница датасета с интерактивной таблицей характеристик студента. Страница представлена на рисунке 154.

Данную таблицу можно сортировать, нажав на заголовок колонки с названием необходимого признака, при этом записи будут отсортированы в алфавитном порядке значений этого признака. Для создания таблиц использовалась JavaScript-библиотека jQuery.dataTables.

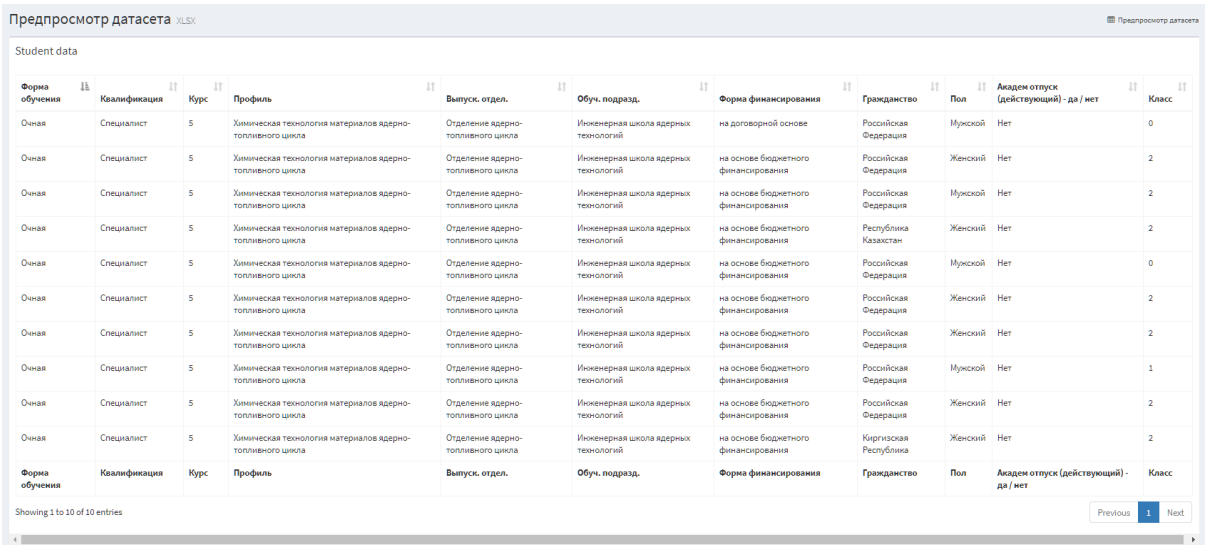


Рисунок 154. Страница датасета

После выбора в панели функций пункта «Обработать датасет», открывается страница выбора датасета для обработки. Список содержит только загруженные датасеты. Форма представлена на рисунке 155. При выборе датасета из формы и нажатия кнопки «Выбрать функции» открывается страница

выбора функций обработки. Выбор в форме реализован с помощью элементов RadioButton.

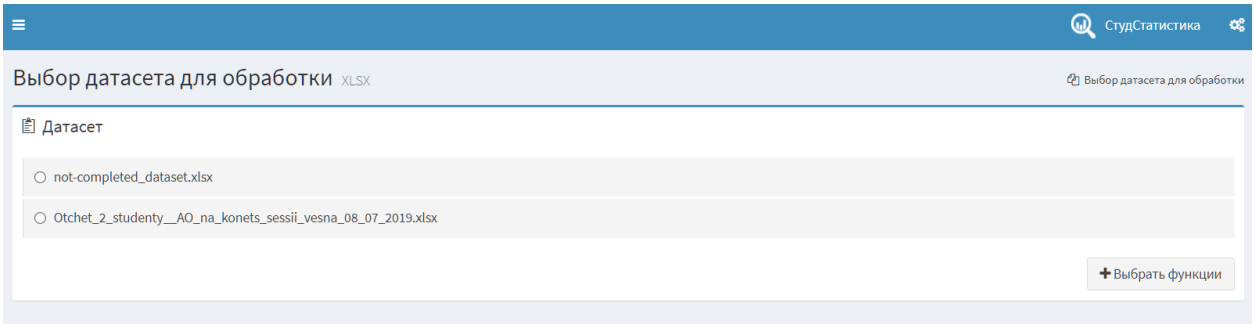


Рисунок 155. Страница выбора датасета для обработки

На рисунках 156-157 представлена форма выбора функций обработки датасетов. Выбор в форме реализован с помощью элементов RadioButton и CheckBox. При нажатии на кнопку «Обработать датасет» происходит обработка и сохранение датасета, а также переход на страницу результата обработки.

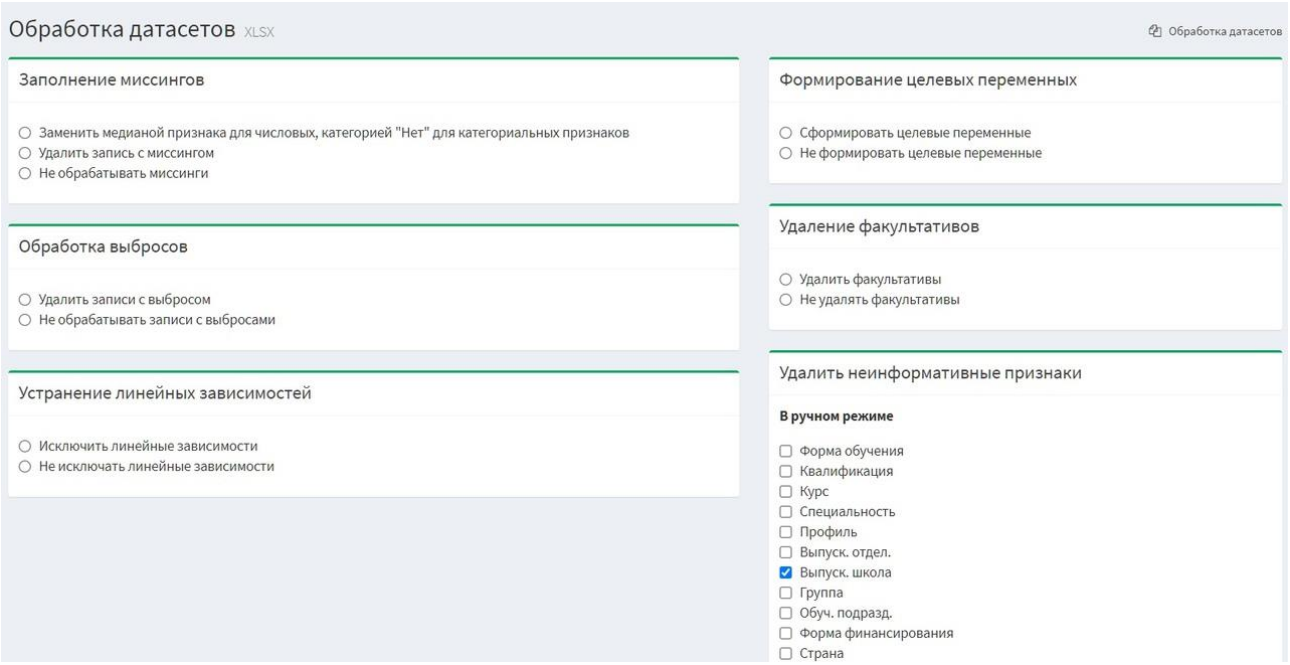


Рисунок 156. Страница выбора функций для обработки (часть 1)

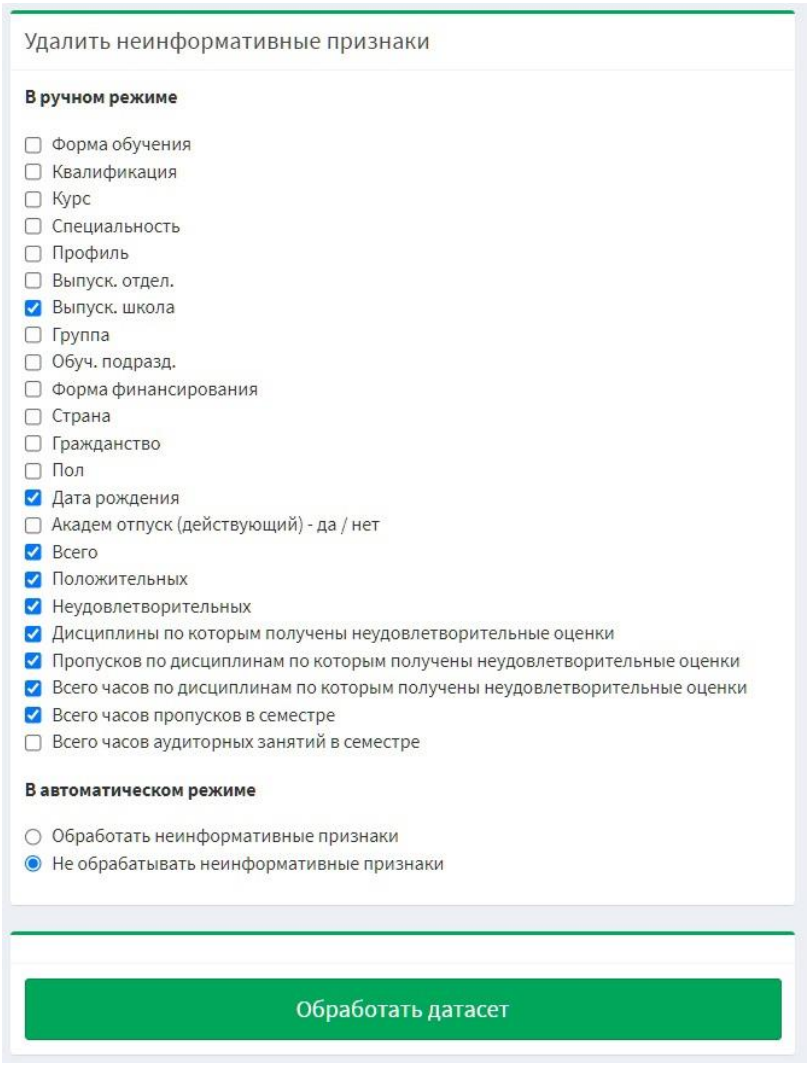


Рисунок 157. Страница выбора функций для обработки (часть 2)

На рисунке 158 представлена страница результата обработки датасета. Эта страница уведомляет пользователя об успешной обработке датасета, имени нового датасета и о странице, где можно посмотреть информацию о нем.

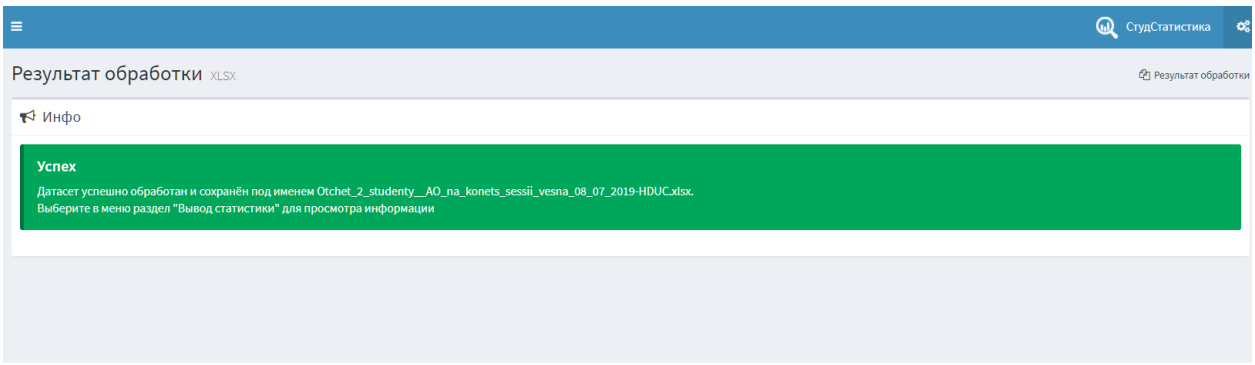


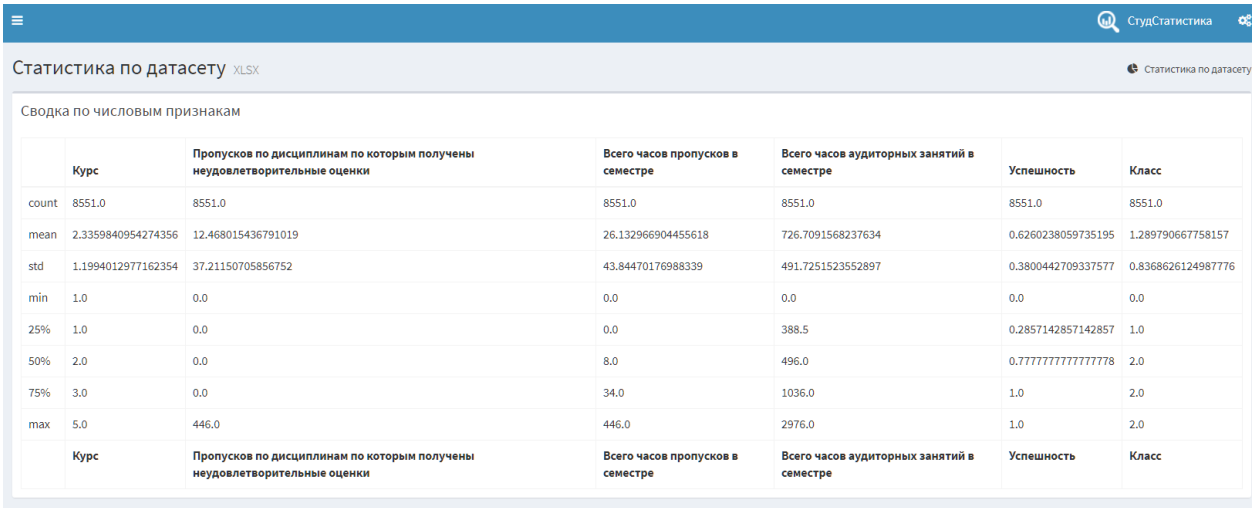
Рисунок 158. Страница результата обработки датасета

При нажатии на пункт «Вывод статистики» происходит переход на страницу выбора датасета для вывода статистики, представленной на рисунке

159. После нажатия на один из пунктов в списках загруженных и обработанных датасетов отображается страница визуализации статистики.

Рисунок 159. Страница выбора датасета для вывода статистики

Статистика на странице представлена с помощью интерактивных графиков и таблицы. На рисунке 160 представлена таблица, содержащая информационную сводку по числовым признакам. На рисунке 161 представлены интерактивные графики распределения, созданные с помощью JavaScript-библиотеки ChartJS. На рисунке 162 представлена одна из диаграмм размаха, которые создавались с помощью JavaScript-библиотеки AnyChartJS. На рисунках 163-164 представлены категориальные и Q-Q графики, созданные также с помощью библиотеки ChartJS и определенный вывод по датасету, основанный на соответствующих статистических критериях.



	Курс	Пропусков по дисциплинам по которым получены неудовлетворительные оценки	Всего часов пропусков в семестре	Всего часов аудиторных занятий в семестре	Успешность	Класс
count	8551.0	8551.0	8551.0	8551.0	8551.0	8551.0
mean	2.3359840954274356	12.468015436791019	26.132966904455618	726.7091568237634	0.6260238059735195	1.289790667758157
std	1.1994012977162354	37.21150705856752	43.84470176988339	491.7251523552897	0.3800442709337577	0.8368626124987776
min	1.0	0.0	0.0	0.0	0.0	0.0
25%	1.0	0.0	0.0	388.5	0.2857142857142857	1.0
50%	2.0	0.0	8.0	496.0	0.7777777777777778	2.0
75%	3.0	0.0	34.0	1036.0	1.0	2.0
max	5.0	446.0	446.0	2976.0	1.0	2.0
	Курс	Пропусков по дисциплинам по которым получены неудовлетворительные оценки	Всего часов пропусков в семестре	Всего часов аудиторных занятий в семестре	Успешность	Класс

Рисунок 160. Страница визуализации статистики (часть 1)

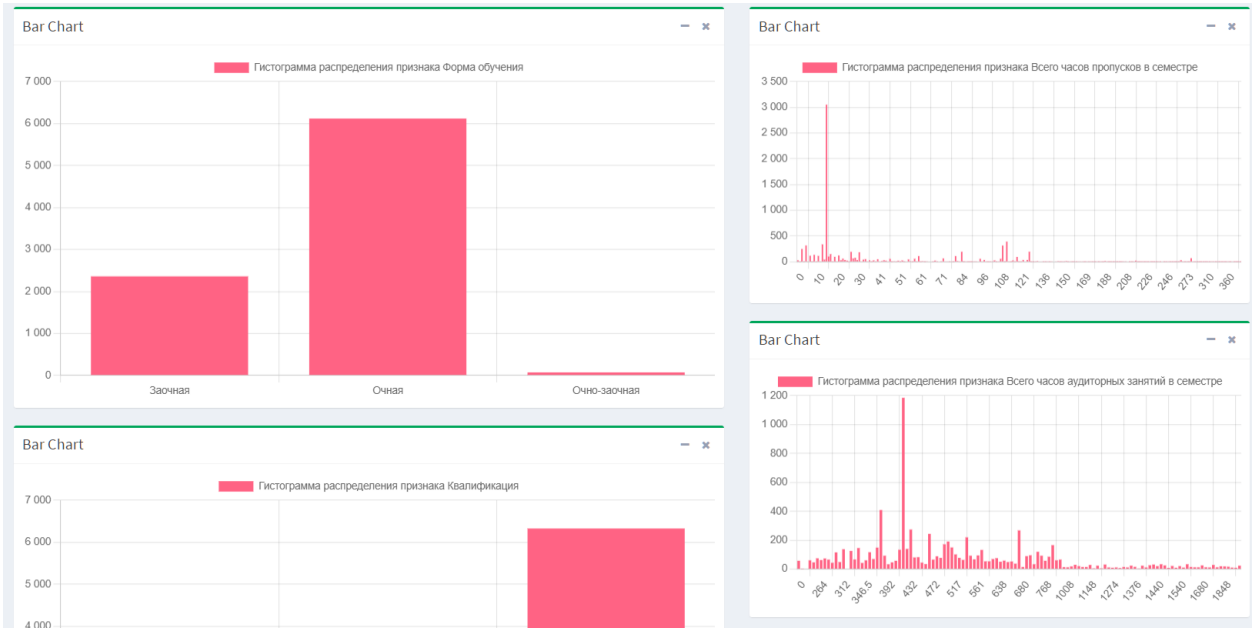


Рисунок 161. Страница визуализации статистики (часть 2)

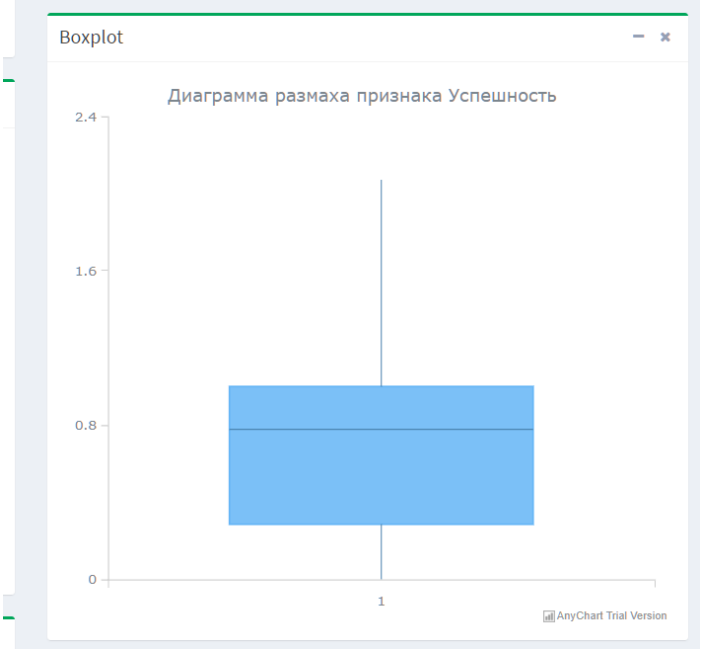


Рисунок 162. Страница визуализации статистики (часть 3)

Рисунок 163. Страница визуализации статистики (часть 4)

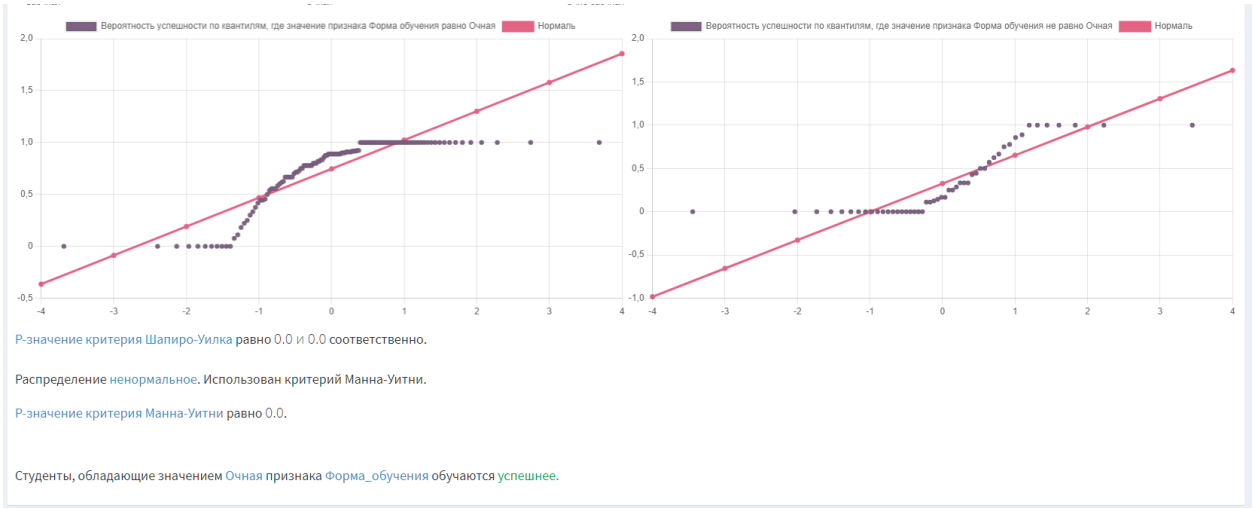


Рисунок 164. Страница визуализации статистики (часть 5)

При нажатии на пункт «Сохранить отчет» происходит переход на страницу с выбором датасета для генерации отчета, представленной на рисунке 165. После нажатия на один из пунктов в списках загруженных и обработанных датасетов генерируется и скачивается PDF-отчет выбранного датасета.

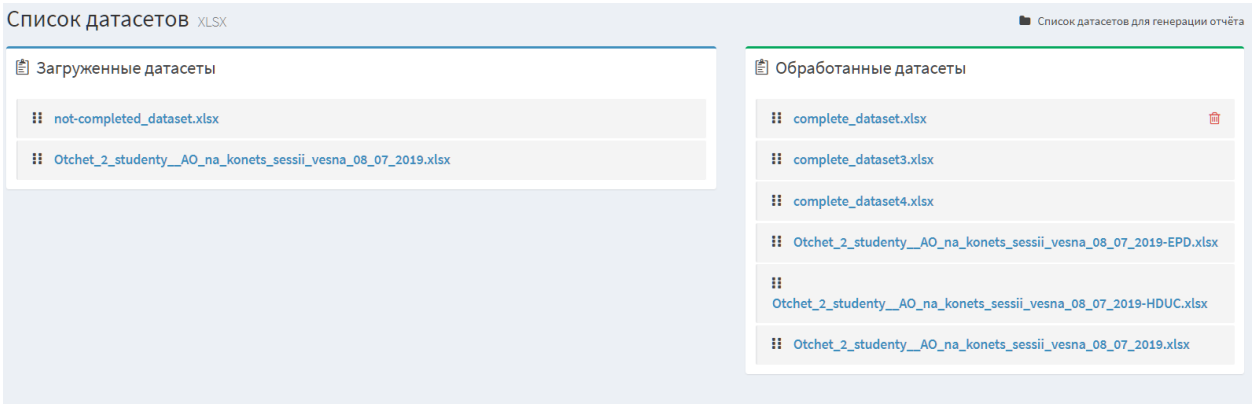


Рисунок 165. Страница выбора датасета для генерации отчета

Структура PDF-отчета сходна со страницей визуализации статистики по датасету. Первая часть отчета – эмблема университета и таблица со статистикой по числовым признакам студентов (рисунок 166). Вторая часть содержит диаграммы распределения и размаха признаков (рисунки 167-168). Третья часть представлена в виде категориальных, Q-Q графиков и выводов по датасету на основе соответствующих статистических критериев (рисунок 169).

ТОМСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ

Статистика по характеристикам студентов за x семестр y года

	Пропусков по дисциплинам по которым получены неудовлетворительные оценки	Всего часов пропусков в семестре	Всего часов аудиторных занятий в семестре	Успешность
count	8551.0	8551.0	8551.0	8551.0
mean	12.468015436791019	26.132966904455618	726.7091568237634	0.6260238059735195
std	37.21150705856752	43.84470176988339	491.7251523552897	0.3800442709337577
min	0.0	0.0	0.0	0.0
25%	0.0	0.0	388.5	0.2857142857142857
50%	0.0	8.0	496.0	0.7777777777777778
75%	0.0	34.0	1036.0	1.0
max	446.0	446.0	2976.0	1.0

Рисунок 166. Отчет по датасету (часть 1)

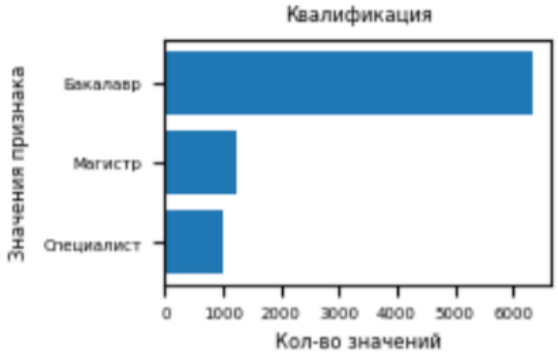


Рисунок 167. Отчет по датасету (часть 2)

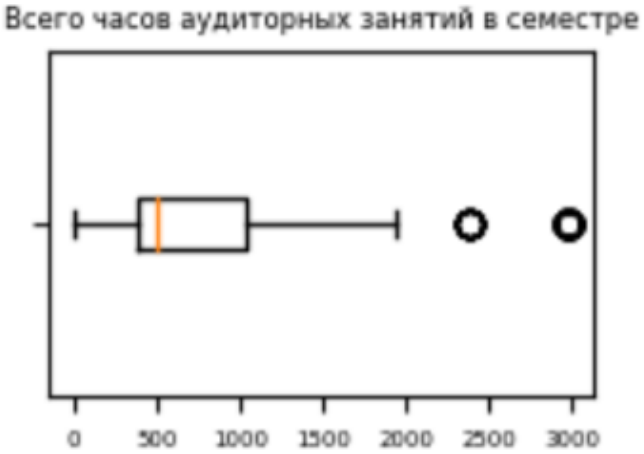
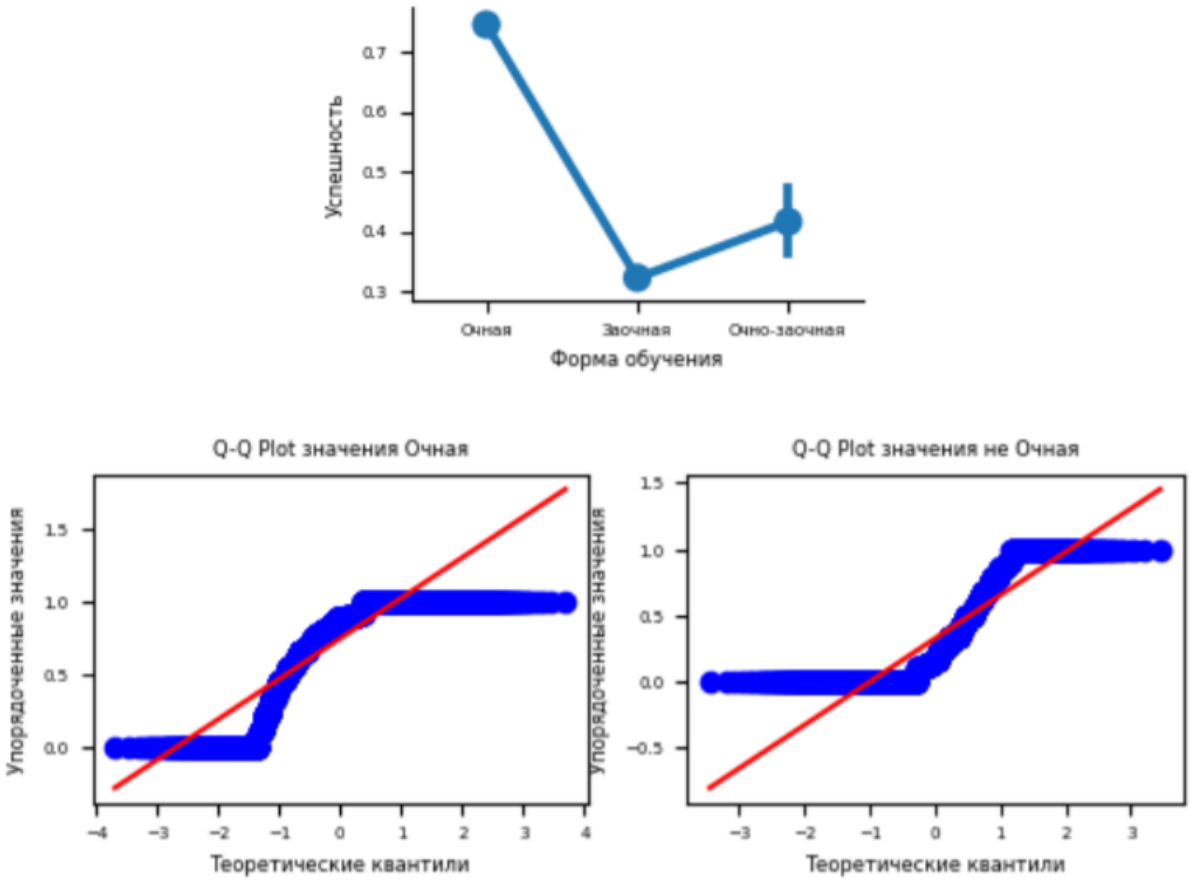


Рисунок 168. Отчет по датасету (часть 3)

Ниже, на рисунке 169 представлен вывод по данным на основе статистических критериев. Сначала распределение сравниваемых выборок проверяется на нормальность с помощью критерия Шапиро-Уилка. Если распределение нормальное, то сравнение выборок происходит с помощью t-критерия Стьюдента, иначе – с помощью критерия Манна-Уитни. В результате сравнения формируется вывод о статистически значимом или незначимом различии между средними выборок.

Поиск скрытых зависимостей

Форма обучения



P-значение критерия Шапиро-Уилка равно: 0.0 и 0.0 соответственно.
Распределение ненормальное. Использован критерий Манна-Уитни.
P-значение критерия Манна-Уитни равно 0.0
Студенты, обладающие значением Очная признака Форма обучения обучаются успешнее.

Рисунок 169. Отчет по датасету (часть 4)

При нажатии на пункт «Обучить модель» происходит переход на страницу выбора датасета для обучения SVM-модели, представленной на рисунке 170. После нажатия на один из пунктов в списках загруженных и обработанных датасетов происходит обучение модели и перенаправление на страницу результата обучения модели.

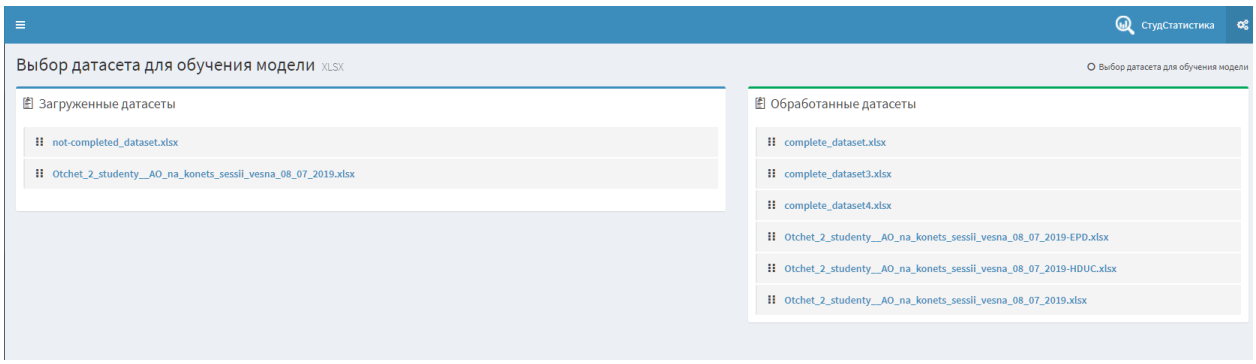


Рисунок 170.Страница выбора датасета для обучения модели

На странице результата обучения модели (рисунок 171) содержится уведомление о том, что процесс обучения завершился, и название датасета, на котором происходило обучение.

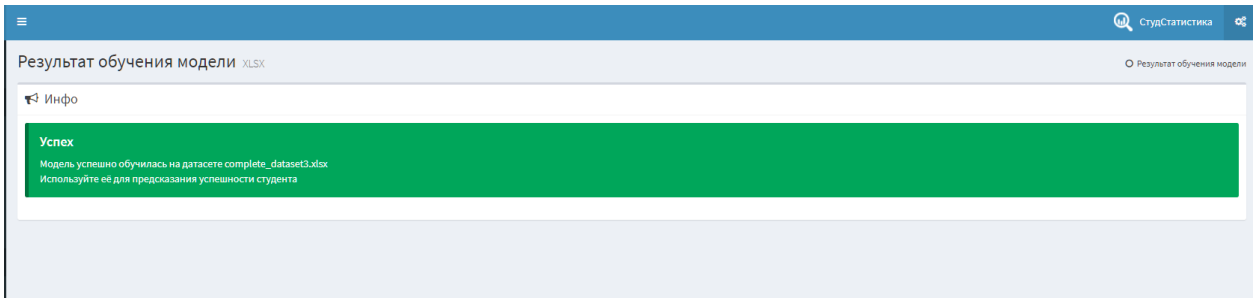


Рисунок 171 – Страница результата обучения модели

На рисунке 172 представлена страница ввода характеристик студента. При создании формы использовались базовые элементы форм Django – ChoiceField и CharField. При введении характеристик студента и нажатии кнопки «Спрогнозировать успеваемость» происходит предсказание с помощью модели и перенаправление на страницу результата прогнозирования. На рисунках 173-175 представлены варианты отображения такой страницы: сообщение о неуспешном студенте, об успешном, а также о студенте, который будет учится в пределах нормы. Помимо класса студента, в сообщении представлена уверенность модели в этом классе.

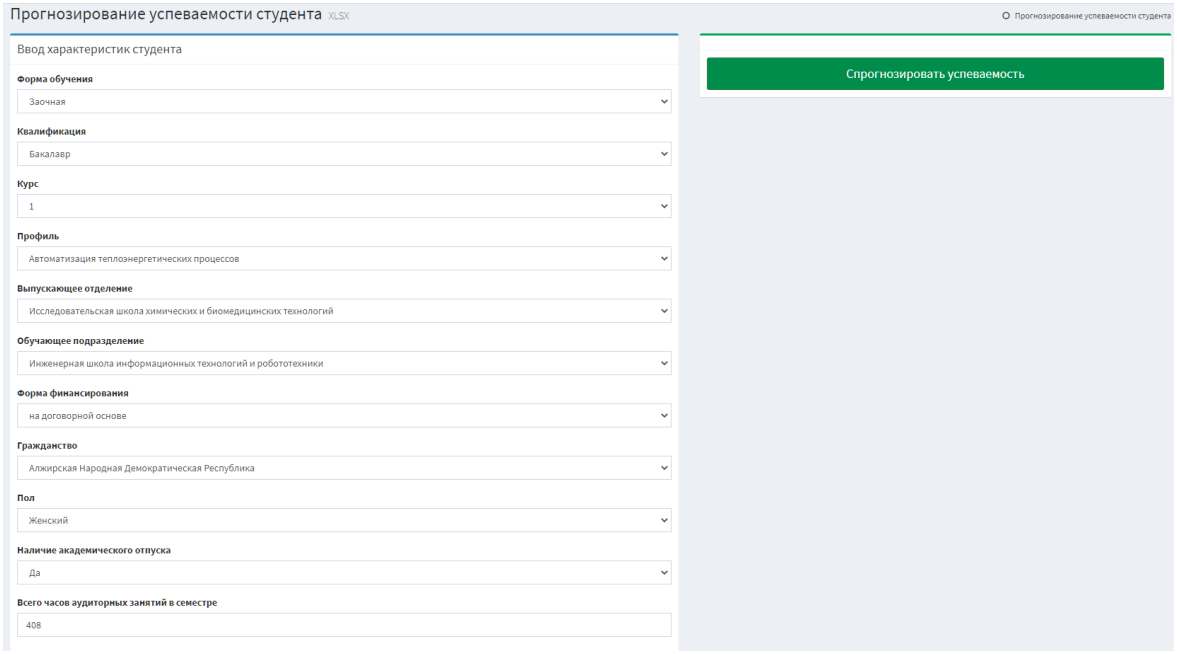


Рисунок 172. Страница ввода характеристик студента

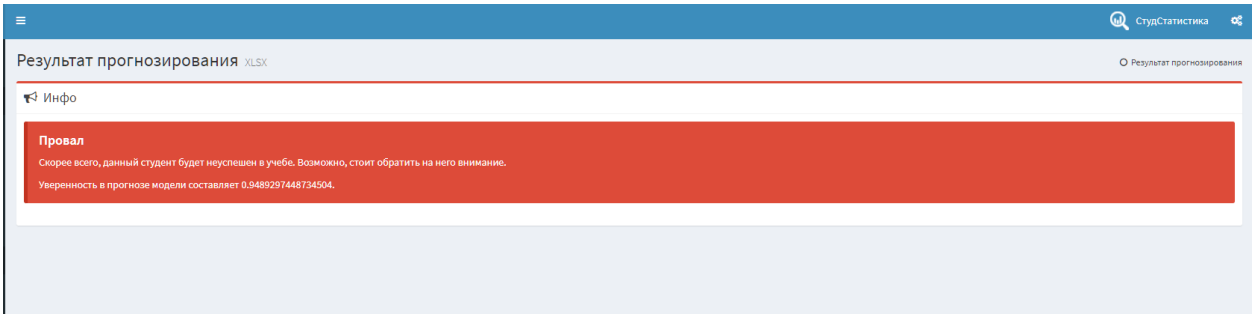


Рисунок 173. Страница результата прогнозирования (вариант 1)

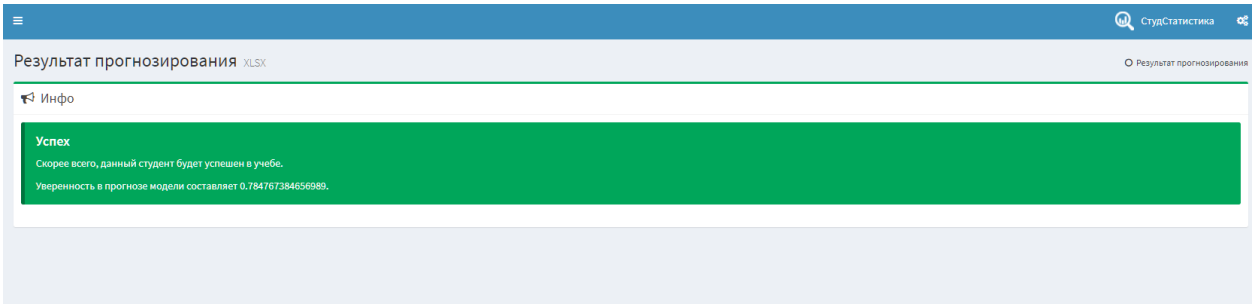


Рисунок 174. Страница результата прогнозирования (вариант 2)

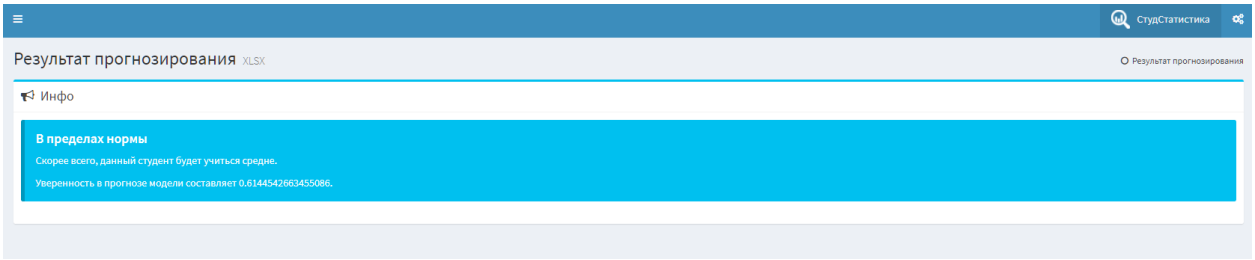


Рисунок 175. Страница результата прогнозирования (вариант 3)

Глава 5. Финансовый менеджмент, ресурсоэффективность и ресурсосбережение

Введение

В данной работе осуществляется исследование влияния характеристик студентов на их успеваемость, а также проектирование и разработка веб-приложения на его основе, которое в дальнейшем будет использоваться сотрудниками лаборатории ТПУ для визуализации закономерностей в данных и получения отчетов. Соответственно, задача раздела – экономически обосновать разработку, определить и рассчитать трудовые и денежные затраты на ее создание.

5.1. Оценка коммерческого потенциала и перспективности проведения научных исследований с позиции ресурсоэффективности и ресурсосбережения

5.1.1. Потенциальные потребители результатов исследования

Разрабатываемая программная система предназначена для использования сотрудниками научной лаборатории «Синтех» ТПУ, а также администрацией университета, которым необходимо получать актуальную информацию об успеваемости студентов каждый семестр. Данная система принимает на вход файл .xlsx с данными о характеристиках студентов и выдает .pdf отчет со статистикой и визуализацией данных. Предполагается, что каждый семестр сотрудники будут подавать новые данные определенной размерности и следить за изменениями в успеваемости и её причинами.

5.1.2. Анализ конкурентных технических решений

В ходе выполнения работы были выявлены и проанализированы конкуренты разрабатываемой программной системы. В качестве конкурента 1 была взята система, разработанная исследователями университета Пердью в

США, которая собирает информацию об академической истории студентов и их активности в цифровой учебной среде, а также демографические данные. На этой основе рассчитывается уровень риска отсева для каждого студента [13].

В качестве конкурента 2 была взята интерактивная система дескриптивной аналитики английского университета Ноттингем Трент. Она представляет собой панель монитора, которая показывает данные о вовлеченности студентов в учебный процесс. На дэшборде выводятся показатели вовлеченности каждого учащегося в сравнении с его одногруппниками, что позволяет понять, насколько он включен в учебный процесс и жизнь университета в целом [14].

По результатам анализа была составлена таблица 9 «Оценочная карта для сравнения конкурентных технических решений».

Таблица 9. Оценочная карта для сравнения конкурентных технических решений

Критерии оценки	**Вес критерия**	**Баллы**			**Конкурентоспособность**		
		$Б_ф$	**$Б_{к1}$**	**$Б_{к2}$**	**$К_ф$**	**$К_{к1}$**	**$К_{к2}$**
1	**2**	**3**	**4**	**5**	**6**	**7**	**8**
Технические критерии оценки ресурсоэффективности							
Интуитивно-понятный интерфейс	0,1	5	4	4	0,5	0,4	0,4
Кроссплатформенность	0,05	5	5	4	0,5	0,5	0,4
Сопроводительная документация	0,01	3	3	4	0,3	0,3	0,5
Простота ввода в эксплуатацию	0,1	5	5	3	0,5	0,5	0,3
Рациональные методы обработки данных	0,15	4	4	4	0,4	0,4	0,4
Корректная визуализация результатов	0,15	5	4	5	0,5	0,4	0,5
Безопасность	0,05	4	4	5	0,4	0,4	0,5
Экономические критерии оценки эффективности							
Цена разработки	0,15	5	4	3	0,5	0,4	0,3
Предполагаемый срок эксплуатации	0,14	5	5	5	0,5	0,5	0,5
Срок разработки	0,1	5	4	3	0,5	0,4	0,3
Итого	**1**	**46**	**42**	**40**	**4,6**	**3,6**	**4,1**

Из таблицы следует, что для выполнения задач по исследованию характеристик студентов ТПУ имеет смысл разработать собственную систему. Университеты Пердью и Ноттингем Трент используют свои собственные форматы и размерности данных, подходящие конкретно им. Кроме того, доступ к конкурентным решениям другим университетам не предоставляется (данные о системах взяты из статьи [13] и отчета [14]).

5.1.3. Технология QuaD

Технология QuaD (QUality ADvisor) представляет собой гибкий инструмент измерения характеристик, описывающих качество новой разработки и её перспективность, и позволяющий принимать решение целесообразности вложения денежных средств в НИР.

Таблица 10. Оценочная карта для сравнения конкурентных технических решений QuaD

Критерий оценки	**Вес крите-рия**	**Бал-лы**	**Макс. балл**	**Относи-тельное значение (3/4)**	**Средне-взвешенное значение (5×2)**
1	**2**	**3**	**4**	**5**	**6**
Показатели оценки качества разработки					
Интуитивно-понятный интерфейс	0,1	80	100	0,8	0,08
Кроссплатформенность	0,05	80	100	0,8	0,04
Сопроводительная документация	0,01	50	100	0,5	0,005
Простота ввода в эксплуатацию	0,1	70	100	0,7	0,07
Рациональные методы обработки данных	0,15	80	100	0,8	0,12
Корректная визуализация результатов	0,15	80	100	0,8	0,12
Безопасность	0,05	60	100	0,6	0,03
Показатели оценки коммерческого потенциала разработки					
Цена разработки	0,15	90	100	0,9	0,135
Предполагаемый срок эксплуатации	0,14	80	100	0,8	0,112
Срок разработки	0,1	80	100	0,8	0,08
Итого	**1**	**720**	**1000**	**7,5**	**0,792**

Средневзвешенное значение показателя качества и перспективности научной разработки составляет 79,2, что говорит о том, что перспективность выше среднего.

5.1.4. SWOT-анализ

SWOT-анализ – инструмент стратегического менеджмента, представляющий собой комплексный анализ научно-исследовательского проекта. SWOT-анализ применяют для исследования внешней и внутренней среды проекта. Метод помогает понять, какие действия необходимо предпринять для максимизации возможностей проекта и нейтрализации слабых сторон и угроз.

Целью использования SWOT-анализа является определение возможной эффективности и прогнозирование направлений будущего развития разрабатываемого решения.

Результаты проведения SWOT-анализа представлены в таблице 11.

Таблица 11. Сводная матрица SWOT-анализа

	Сильные стороны научно-исследовательского проекта: С1. Интуитивно-понятный интерфейс. С2. Пользователю не требуется знать языки программирования. С3. Доступная стоимость разработки. С4. Подробная визуализация зависимостей в данных.	**Слабые стороны научно-исследовательского проекта:** Сл1. Конечным продуктом будет пользоваться узкий круг специалистов. Сл2. Отсутствие в команде опыта проведения подобного рода исследований. Сл3. Необходимость поддержки и обновления ПО после внедрения. Сл4. Необходимость адаптации продукта под определенные данные определенных институтов.
Возможности: В1. Расширение команды разработки. В2. Легкая интеграция нового функционала. В3. Возможность запуска на большом количестве платформ. В4. Использование инновационной инфраструктуры ТПУ для упрощения разработок.	Расширение масштабов: разработкой могут заинтересоваться другие вузы, а также различные компании и государственные учреждения, которым подобная разработка помогла бы лучше изучить характеристики сотрудников и клиентов.	Легкая интеграция нового функционала и расширение команды разработки облегчат обновление ПО и его адаптацию под определенных пользователей.
Угрозы: У1. Появление свободно-распространяющихся аналогичных продуктов у других исследовательских команд. У2. Появление новых популярных платформ, на которых продукт не работает. У3. В случае расширения команды может быть нелегко нанять или заменить узких специалистов	Грамотное проведение презентации разработки. Акцентирование внимания пользователей на индивидуальных особенностях приложения. Проектирование и выпуск обновлений с учетом возможного появления новых платформ.	Основной угрозой является появление новых конкурентных продуктов. Заказчики могут предпочесть встроенные в свою систему инструменты обработки данных, если такие появятся. Мониторинг изменений в интересах заказчиков и выпуск соответствующих обновлений.

По результатам SWOT-анализа были выявлены сильные и слабые стороны научной разработки, а также её угрозы и возможности.

5.2. Планирование научно-исследовательских работ

Рациональное планирование занятости участников разработки и сроков проведения каждого из этапов работы позволяет успешно организовать процесс работы над конкретной задачей.

5.2.1. Структура работ в рамках научного исследования

На данном этапе составлен список работ, а также назначен исполнитель и выставлена временная продолжительность. Результатом планирования работ является линейных график реализации проекта. Перечень этапов работы и распределение исполнителей представлен в таблице 12.

Таблица 12. Перечень этапов работы и распределение исполнителей

№ п/п	Этапы работы	Исполнители
1	Постановка целей и задач	Научный руководитель
2	Разработка и утверждение ТЗ	Научный руководитель, студент
3	Подбор и изучение материалов по тематике	Научный руководитель, студент
4	Разработка календарного плана	Научный руководитель, студент
5	Обсуждение литературы	Научный руководитель, студент
6	Проведение анализа предметной области	Студент
7	Проведение анализа данных	Студент
8	Проектирование	Научный руководитель, студент
9	Разработка	Студент
10	Тестирование и отладка	Студент
11	Согласование выполненной работы с научным руководителем	Научный руководитель, студент
12	Оформление работы	Научный руководитель, студент

5.2.2. Определение трудоемкости выполнения работ и разработка графика проведения научного исследования

Трудовые затраты являются основными затратами на разработку, поэтому необходимо определить их для каждого из исполнителей. Ожидаемая продолжительность работ $t_{ож}$ с помощью экспертных оценок устанавливается согласно формуле 5:

$$t_{ож_i} = \frac{3t_{min_i} + 2t_{max_i}}{5}, \quad (5)$$

где t_{min} – минимальная продолжительность работ в днях;

t_{max} – максимальная продолжительность работ в днях.

Исходя из этого определяется продолжительность каждой работы в рабочих днях $Т_{РД}$, учитывающая выполнение работ несколькими исполнителями. Формула для вычисления длительности этапов в рабочих днях $Т_{РД}$ (формула 6):

$$T_{РД_i} = \frac{t_{ож_i}}{Ч_i}, \quad (6)$$

где $t_{ожi}$ – ожидаемая трудоемкость выполнения одной работы, чел.·дни;

$Ч_i$ – численность исполнителей, выполняющих одновременно одну и ту же работу на данном этапе, чел.

Теперь переведем рабочие дни в календарные, чтобы определить дату окончания работ. Продолжительность этапа в календарных днях $Т_{КД}$ рассчитывается по формуле 7:

$$Т_{КД} = Т_{РД} \cdot Т_К, \quad (7)$$

где $Т_{РД}$ – продолжительность выполнения этапа в рабочих днях;

$Т_К$ – коэффициент календарности, который вычисляется по формуле 8:

$$Т_К = \frac{Т_{КАЛ}}{Т_{КАЛ} - Т_{ВД} - Т_{ПД}}, \quad (8)$$

где $Т_{КАЛ}$ – календарные дни ($Т_{КАЛ}$ = 365);

$Т_{ВД}$ – выходные дни ($Т_{ВД}$ = 52);

$Т_{ПД}$ – праздничные дни ($Т_{ПД}$ = 19) [15].

Коэффициент календарности $Т_К$ равен 1,241. Все расчеты по трудозатратам представлены в таблице 13, результаты продолжительности

этапов работы являются общими трудоемкостями для каждого из исполнителей проекта. По величинам трудоемкости этапов по исполнителям ТКД (данные столбцов 11-13) был построен линейный график реализации проекта (диаграмма Ганта). Данный график приведен в таблице 14.

Таблица 13. Трудозатраты на выполнение проекта

Этап работы	**Доля выполняемой работы исполнителем, %**			**Продолжительность работ, чел.·дни**			**Длительность работ, дни**					
	НР*	**С1****	**С2*****	**t_{min}**	**t_{max}**	**$t_{ож}$**	**$T_{РД}$**			**$T_{КД}$**		
							НР	**С1**	**С2**	**НР**	**С1**	**С2**
1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**
Постановка целей и задач	100	0	0	2	4	2,8	3,36	0	0	4,17	0	0
Разработка и утверждение ТЗ	33	33	33	4	6	4,8	1,9	1,9	1,9	2,36	2,36	2,36
Подбор и излучение материалов по тематике	20	40	40	5	10	7	1,68	3,36	3,36	2,09	4,17	4,17
Разработка календарного плана	33	33	33	2	4	2,8	1,11	1,11	1,11	1,38	1,38	1,38
Обсуждение литературы	33	33	33	2	4	2,8	1,11	1,11	1,11	1,38	1,38	1,38
Проведение анализа предметной области	0	50	50	4	6	4,8	0	2,88	2,88	0	3,58	3,58
Проведение анализа данных	0	55	45	30	40	34	0	22,44	18,36	0	27,86	22,79
Проектирование	10	50	40	6	10	7,6	0,91	4,56	3,65	1,13	5,66	4,53
Разработка	0	40	60	30	40	34	0	16,32	24,48	0	20,26	30,39
Тестирование и отладка	0	45	55	8	12	9,6	0	5,18	6,34	0	6,44	7,87
Согласование выполненной работы с научным руководителем	33	33	33	6	10	7,6	3,01	3,01	3,01	3,74	3,74	3,74
Оформление работы	10	45	45	4	6	4,8	0,58	2,59	2,59	0,72	3,22	3,22
Итого:							**13,66**	**53,06**	**56,18**	**16,95**	**65,88**	**69,75**

* – Научный руководитель

** – Студент 1

*** – Студент 2

Таблица 14. Диаграмма Ганта

Этап	Ткд НР, календ. дн.	Ткд С1, календ. дн.	Ткд С2, календ. дн.	Продолжительность выполнения работ									
				февраль		март			апрель			май	
				10	20	30	40	50	60	70	80	90	100
1	4,17	0	0										
2	2,36	2,36	2,36										
3	2,09	4,17	4,17										
4	1,38	1,38	1,38										
5	1,38	1,38	1,38										
6	0	3,58	3,58										
7	0	27,86	22,79										
8	1,13	5,66	4,53										
9	0	20,26	30,39										
10	0	6,44	7,87										
11	3,74	3,74	3,74										
12	0,72	3,22	3,22										

Примечание: ■ – НР, ▨ – С1, ▤ – С2

5.2.3. Бюджет проекта

Для проекта по применению инструментов Data Mining для анализа успеваемости студентов, а также разработки информационной системы для реализации методологии применения этих инструментов, производится оценка затрат по следующим статьям:

- материалы;
- заработная плата;
- отчисления во внебюджетные фонды;
- амортизационные расходы;
- накладные расходы.

Работа по проекту выполнялась без участия иных организаций и без командировок и аренды имущества, следовательно, расходы по соответствующим статьям отсутствуют.

5.2.3.1. Расчет затрат на материалы

В затратах на материалы были учтены затраты на бумагу и картриджи для принтера, поскольку все необходимые материалы имелись в инвентаре исполнителей. Расчет затрат на материалы представлен в таблице 15.

Таблица 15. Затраты на материалы

Наименование материалов	**Цена за ед., руб.**	**Кол-во**	**Сумма, руб.**
Бумага для принтера, А4	250,00	1 уп.	250,00
Картридж, черная краска	1500,00	1 шт.	1500,00
Итого:			**1750,00**

5.2.3.2. Расчет заработной платы

Расчет основной заработной платы выполняется на основе трудоемкости выполнения каждого этапа и величины месячного оклада исполнителя.

Месячный оклад (МО) научного руководителя, занимающего должность доцента и имеющего степень кандидата физико-математических наук, составляет 47588 руб./мес., МО исполнителя составляет 21760 руб./мес. Исходя

из того, что в месяце в среднем 24,5 рабочих дня при шестидневной рабочей неделе, среднедневная тарифная заработная плата ($ЗП_{дн-т}$) рассчитывается по формуле 9:

$$ЗП_{дн-т} = \frac{МО}{24,5}, \quad (9)$$

Расчеты затрат на полную заработную плату приведены в таблице 16. Затраты времени по каждому исполнителю в рабочих днях с округлением до целого взяты из таблицы 13 (столбцы 8-10), в которой указаны трудозатраты исполнителей.

Таким образом, для перехода от тарифной суммы заработка исполнителя, связанной с участием в проекте, к соответствующему полному заработку, необходимо учесть районный коэффициент Кр = 1,3.

Таблица 16. Затраты на заработную плату

Исполнитель	**Оклад, руб./мес.**	**Среднедневная ставка, руб./раб. день**	**Затраты времени, раб.дни**	**$К_р$**	**Фонд заработной платы, руб.**
Научный руководитель	47588,00	1942,37	14	1,3	35351,09
Студент 1	21760,00	888,16	65	1,3	75049,80
Студент 2	21760,00	888,16	69	1,3	79668,24
Итого:					**190069,10**

5.2.3.3. Расчет затрат на отчисления во внебюджетные фонды

Отчисления во внебюджетные фонды включают в себя отчисления в пенсионный фонд, на социальное и медицинское страхование и составляют (22%+2,9%+5,1% = 30%) от заработной платы участников проекта [16].

$С_{фонд}$ определяется следующим образом (формула 10):

$$С_{фонд} = С_{зп} \cdot К_{фонд} = 190069,10 \cdot 0,3 = 57020,74 \text{ руб.} \quad (10)$$

5.2.3.4. Расчет амортизационных расходов

Амортизационные отчисления для рассматриваемого проекта включают в себя амортизацию используемого оборудования за время выполнения работы.

Продолжительность выполнения проекта – 3 месяца. Амортизационные отчисления рассчитываются по следующей формуле 11:

$$С_{аморт} = \frac{3}{12} \cdot Ц_{ОБ} \cdot Н_{А}, \quad (11)$$

где: $Н_А$ – годовая норма амортизации;

$Ц_{ОБ}$ – цена оборудования;

Годовая амортизация $Н_А$ – величина, обратная сроку амортизации оборудования n, который определяется согласно постановлению правительства РФ «О классификации основных средств, включенных в амортизационные группы». Компьютеры и печатающие устройства входят во вторую группу со сроком полезного использования от 2 до 3 лет включительно [17]. Для компьютера примем $С_А$ = 3 года, lля принтера – $С_А$ = 2 года. Расчет затрат на амортизационные отчисления представлен в таблице 17.

Таблица 17. Затраты на амортизационные отчисления

Наименование оборудования	**Норма аморт. Оборуд. $Н_А$**	**Стоимость оборуд. $Ц_{ОБ}$, руб.**	**Аморт. отчисления, $С_{аморт}$, руб.**
Ноутбук Aser	0,33	40000,00	3300,00
Ноутбук Dexp	0,33	40000,00	3300,00
Лазерный принтер	0,50	9600,00	1200,00
Итого:			**7800,00**

5.2.3.6. Расчет накладных расходов

В данном разделе были рассчитаны накладные расходы. Величина накладных расходов составляет 16% от суммы всех указанных ранее затрат и рассчитывается по формуле 12:

$$С_{проч} = 0{,}16 \cdot (С_{мат} + С_{зп} + С_{фонд} + С_{аморт}) \quad (12)$$

$$С_{проч} = 0{,}16 \cdot (1750{,}00 + 190069{,}10 + 57020{,}74 + 7800{,}00) = 41062{,}37$$

Таким образом, прочие накладные расходы составили 41062,37 руб.

5.2.3.7. Расчет общей себестоимости разработки

Общая себестоимость внедрения инструментов Data Mining для анализа успеваемости студентов в веб-приложение рассчитывается с помощью суммирования всех расходов. Она представлена в таблице 18.

Таблица 18. Общая стоимость разработки проекта

Статья затрат	**Условное обозначение**	**Сумма, руб.**
Материалы и покупные изделия	$C_{мат}$	1750,00
Заработная плата	$C_{зп}$	190069,10
Отчисления в фонды	$C_{фонд}$	57020,74
Амортизационные отчисления	$C_{ам}$	7800,00
Накладные расходы	$C_{накл}$	41062,37
Итого		**297702,20**

Общая себестоимость проекта получилась равной 297702,20 рублей. Наиболее сильно на общую стоимость влияют такие статьи затрат, как «Заработная плата», «Отчисления в социальные фонды» и «Накладные расходы».

Глава 6. Социальная ответственность

6.1. Введение

Выпускная квалификационная работа направлена на применение инструментов Data Mining для анализа успеваемости студентов ВУЗа, а также разработку веб-приложения для удобного пользования данной технологией. Эти технологии будут применены в ТПУ с целью выявления проблем с успеваемостью у студентов на ранних этапах обучения. Пользователем данной информационной системы являются сотрудники ТПУ. Необходимо отметить о важности такого раздела как социальная ответственность, где описываются вопросы обеспечения безопасности сотрудника, нормы производственной и экологической безопасности, а также безопасность в чрезвычайных ситуациях.

6.2. Правовые и организационные вопросы обеспечения безопасности

Трудовой кодекс РФ описывает основные положения отношений между организацией и сотрудниками, включая оплату и нормирование труда, выходных, отпуска, частичного или полного питания, предоставление права на подготовку и дополнительное профессиональное образование, вступление в профессиональный союз, страхование.

Работа в офисе относится ко второй категории тяжести труда – работы выполняются при оптимальных условиях внешней производственной среды и при оптимальной величине физической, умственной и нервно-эмоциональной нагрузки. Продолжительность рабочего дня работников не должна превышать 40 часов в неделю. Также возможно сокращение рабочего времени. Для работников, возраст которых меньше 16 лет – не более 24 часов в неделю, от 16 до 18 лет – не более 35 часов, как и для инвалидов I и II группы [18].

Статья 86 ТК РФ устанавливает следующие общие требования по защите персональных данных [18]:

1. Работники не должны отказываться от своих прав на сохранение и защиту тайны;
2. Работники и их представители должны быть ознакомлены под роспись с документами работодателя, устанавливающими порядок обработки персональных данных работников, а также об их правах и обязанностях в этой области;
3. Все персональные данные работника следует получать у него самого;
4. Защита персональных данных работника от неправомерного их использования или утраты должна быть обеспечена работодателем за счет его средств в порядке, установленном настоящим Кодексом и иными федеральными законами.

Оплата труда должна производиться не реже чем каждые полмесяца. Заработная плата работника не может быть ниже минимального размера оплаты труда, установленного в РФ. Доля заработной платы, выплачиваемой в неденежной форме, не может превышать 20 процентов от начисленной месячной заработной платы [18].

Работодатель обязан обеспечить нормальные условия для выполнения работниками норм выработки. К таким условиям относятся исправное состояние помещений, технологической оснастки и оборудования, своевременное обеспечение технической документацией [18].

В соответствии с ГОСТ 12.2.032-78 «Система стандартов безопасности труда. Рабочее место при выполнении работ сидя» рабочий стол может быть любой конструкции, отвечающей современным требованиям эргономики обеспечивающий оптимальное положение работника [19]. Согласно ГОСТ 22269-76 «Система "Человек-машина". Рабочее место оператора. Взаимное расположение элементов рабочего места», взаимное расположение элементов рабочего стола должно [20]:

- обеспечивать возможность осуществления всех необходимых движений и перемещений;

- обеспечивать необходимые зрительные и звуковые связи между оператором и оборудованием;
- способствовать оптимальному режиму труда и отдыха, снижению утомления, предупреждению появления ошибочных действий.

В соответствии с ГОСТ 21889-76 «Система "Человек-машина". Кресло человека-оператора» кресло оператора может быть с профилированными и непрофилированными элементами. Поверхность сиденья может быть плоской с наклоном 0-5°, или профилированной с углами наклона сиденья. Опорная плоскость сиденья также может быть плоской или профилированной с радиусом кривизны поясничной опоры, равным 460 мм, радиусом изгиба для грудного отдела позвоночника, равным 620 мм и другими точками изгиба [21].

6.3. Производственная безопасность

Условия труда, в которых разрабатывается веб-приложение для интеллектуальной работы с данными студентов, в том числе устройства, с помощью которых осуществляется деятельность компании, могут спровоцировать появление вредных и опасных факторов производства.

При выполнении работ на персональном компьютере (ПЭВМ) согласно «ГОСТ 12.0.003-2015 Система стандартов безопасности труда (ССБТ). Опасные и вредные производственные факторы. Классификация» могут иметь место следующие факторы, представленные в таблице 19 [22]:

Таблица 19. Возможные опасные и вредные факторы

Факторы (ГОСТ 12.0.003-2015)	**Этапы работ**	**Нормативные документы**
	Эксплу-атация	
Отсутствие или недостаток естественного и искусственного освещения	+	СанПиН 1.2.3685-21 «Гигиенические нормативы и требования к обеспечению безопасности и (или) безвредности для человека факторов среды обитания»
Зрительное напряжение	+	СанПиН 1.2.3685-21 «Гигиенические нормативы и требования к обеспечению безопасности и (или) безвредности для человека факторов среды обитания»

Продолжение таблицы 19.

Факторы (ГОСТ 12.0.003-2015)	**Этапы работ**	**Нормативные документы**
	Эксплу-атация	
Повышенный уровень электромагнитного излучения	+	СанПиН 1.2.3685-21 «Гигиенические нормативы и требования к обеспечению безопасности и (или) безвредности для человека факторов среды обитания»
Отклонение показателей микроклимата	+	СанПиН 2.2.4.548-96 «Гигиенические требования к микроклимату производственных помещений»
Повышенное значение напряжения в электрической цепи	+	ГОСТ 12.1.038-82 ССБТ. Электробезопасность. Предельно допустимые уровни напряжений прикосновения и токов.

1) Отсутствие или недостаток естественного и искусственного освещения

Недостаточная освещенность рабочей зоны помещения, оборудованной ПК, возникает из-за малого числа источников искусственного освещения и неверного расположения оконных проемов относительно рабочего места. Недостаточная освещенность оказывает большую нагрузку на зрение и является одной из причин нарушения зрительной функции. Искусственное освещение в помещениях для эксплуатации ПК должно осуществляться системой общего равномерного освещения. В случаях преимущественной работы с ПК следует применять системы комбинированного освещения (к общему освещению дополнительно устанавливаются светильники местного освещения, предназначенные для освещения зоны расположения экрана ПК). Окна в помещениях, где эксплуатируется вычислительная техника, преимущественно должны быть ориентированы на север и северо-восток. Нормативные показатели естественного, искусственного и совмещенного освещения в соответствии с СанПиН 1.2.3685-21 «Гигиенические нормативы и требования к обеспечению безопасности и (или) безвредности для человека факторов среды обитания» указаны в таблице 20 [23].

Таблица 20. Нормативные показатели естественного, искусственного и совмещенного освещения

<table>
<tr><td rowspan="3">Помещения</td><td rowspan="3">Рабочая поверхность и плоскость нормирования КЕО освещенности (Г – горизонтальная, В – вертикальная) и высота плоскости над полом, м</td><td colspan="2">Естественное освещение</td><td colspan="2">Совмещенное освещение</td></tr>
<tr><td colspan="2">КЕО e_n, %</td><td colspan="2">КЕО e_n, %</td></tr>
<tr><td>При верхнем или комбинированном освещении</td><td>При боковом освещении</td><td>При верхнем или комбинированном освещении</td><td>При боковом освещении</td></tr>
<tr><td>Кабинеты и рабочие комнаты, офисы, представительства</td><td colspan="2">Г-0,8</td><td>3,0</td><td>1,0</td><td>1,8</td><td>0,6</td></tr>
<tr><td rowspan="4">помещения</td><td colspan="5">Искусственное освещение</td></tr>
<tr><td colspan="3">Освещенность, лк</td><td></td><td></td></tr>
<tr><td colspan="2">При комбинированном освещении</td><td rowspan="2">При общем освещении</td><td rowspan="2">Показатель дискомфорта, М, не более</td><td rowspan="2">Коэффициент пульсации освещенности, $К_т$ %, не более</td></tr>
<tr><td>Всего</td><td>От общего</td></tr>
<tr><td>Кабинеты и рабочие комнаты, офисы, представительства</td><td>400</td><td>200</td><td>300</td><td>40</td><td>15</td></tr>
</table>

Если следовать приведенным мерам, то удастся минимизировать вероятность нарушения зрительной функции.

2) Зрительное напряжение

Работа на ПК сопровождается постоянным и значительным напряжением функций зрительного анализатора. Одной из основных особенностей является иной принцип чтения информации, чем при обычном чтении. Допустимые уровни ультрафиолетового излучения, создаваемые изделиями, в соответствии с СанПиН 1.2.3685-21 «Гигиенические нормативы и требования к обеспечению

безопасности и (или) безвредности для человека факторов среды обитания» указаны в таблице 21 [23].

Таблица 21. Допустимые уровни ультрафиолетового излучения

Вид изделий	**Спектральный диапазон длин волн, нм**	**Допустимая интенсивность облучения, Вт/м^2**
Экраны телевизоров, видеомониторов, осциллографов измерительных и других приборов, средств отображения информации с визуальным контролем	Свыше 315 до 400	Не более 0,1
	Свыше 280 до 315	Не более 0,0001
	От 200 до 280	Не допускается

Чтобы снизить зрительное напряжение, необходимо выбрать такой монитор, параметры которого удовлетворяли бы параметрам таблицы 3. Соблюдение этих норм позволит снизить зрительное напряжение.

3) Повышенный уровень электромагнитного излучения

При работе за компьютером человек подвергается воздействию электромагнитного излучения, исходящего от системного блока, монитора и передающих сигналы устройств. Эти воздействия негативно сказываются на здоровье человека: нарушаются нервная и сердечно-сосудистая система, возникают головные боли, ухудшается самочувствие.

Предельно допустимые уровни постоянного магнитного поля на рабочих местах, в соответствии с СанПиН 1.2.3685-21 «Гигиенические нормативы и требования к обеспечению безопасности и (или) безвредности для человека факторов среды обитания» указаны в таблице 22 [23].

Таблица 22. ПДУ постоянного и переменного магнитного поля на рабочих местах

ПДУ постоянного магнитного поля				
Время воздействия за рабочий день, мин	**Условия воздействия**			
	Общее		**Локальное**	
	ПДУ напряженности, кА/м	ПДУ магнитной индукции, мТл	ПДУ напряженности, кА/м	ПДУ магнитной индукции, мТл
≤ 10	24	30	40	50
11-60	16	20	24	30
61-480	8	10	12	15
ПДУ переменного магнитного поля частотой 50 Гц				
Время пребывания, ч	Допустимые уровни МП, Н [А/м] / В [мкТл] при воздействии:			
	общем		**локальном**	
≤ 1	1600/2000		6400/8000	
2	800/1000		3200/4000	
4	400/500		1600/2000	
8	80/100		800/1000	

Во избежание последствий негативного воздействия электромагнитного излучения, следует делать перерывы по 10-20 минут, чтобы снизить время его воздействия. Данные меры позволят снизить риск заболеваний, вызываемых электромагнитным излучением.

4) Отклонение показателей микроклимата

Микроклимат определяется действующими на организм человека показателями температуры, влажности и скорости движения воздуха. Длительное воздействие на человека неблагоприятных показателей микроклимата ухудшает его самочувствие, снижает производительность труда и приводит к заболеваниям, поэтому в организации должны обеспечиваться оптимальные параметры микроклимата, установленные СанПиН 1.2.3685-21, представленные в таблице 23 [23].

Таблица 23. оптимальные и допустимые величины показателей микроклимата

Оптимальные значения характеристик микроклимата				
Период года	**Температура воздуха, °С**	**Температура поверхностей, °С**	**Относительная влажность воздуха, %**	**Скорость движения воздуха, м/с**
Холодный	22-24	21-25	40-60	0,1
Теплый	23-25	22-26	40-60	0,1
Допустимые значения характеристик микроклимата				
Период года	**Температура воздуха, °С**	**Температура поверхностей, °С**	**Относительная влажность воздуха, %**	**Скорость движения воздуха, м/с**
Холодный	20-25	19-26	15-75	0,1
Теплый	21-28	20-29	15-75	0,1 – 0,2

Соблюдение по крайней мере допустимых параметров снижает риск заболевания сотрудников.

5) Повышенное значение напряжения в электрической цепи

В деятельности организации широко используется электричество для питания компьютерной техники, которая может являться источником опасности. Поражение электрическим током может произойти при прикосновении к токоведущим частям, находящимся под напряжением, на которых остался заряд или появилось напряжение

Электрический ток оказывает на человека термическое, электролитическое, биологическое и механическое воздействие. Действие электрического тока на человека приводит к травмам или гибели людей.

ГОСТ 12.1.038-82 ССБТ. «Электробезопасность. Предельно допустимые уровни напряжений прикосновения и токов» устанавливает ПДЗ напряжений прикосновения и токов, предназначенные для проектирования способов и средств защиты людей [24]. Так, для переменного тока частотой 50 Гц допустимое значение напряжения прикосновения составляет 2 В, а силы тока – 0,3 мА, для тока частотой 400 Гц, соответственно – 2 В и 0,4 мА, для постоянного тока – 8 В и 1 мА.

Мерами защиты от воздействия электрического тока являются оградительные устройства, устройства автоматического контроля и

сигнализации, изолирующие устройства и покрытия, устройства защитного заземления, устройства автоматического отключения, предохранительные устройства.

Пользование мерами защиты, приведенными выше, позволяют избежать поражения электрическим током.

6.4. Экологическая безопасность

Для проведения исследования данных о характеристиках студентов и разработки веб-приложения на его основе использовались персональные компьютеры и ноутбуки. Вышедшее из строя ПЭВМ и сопутствующая оргтехника относится к IV классу опасности и подлежит специальной утилизации. В жидкокристаллических мониторах с диагональю 26 дюймов содержится от 2 до 4 U–образных ламп массой по 13 г, которые содержат ртуть в количестве 62,14 мг/кг, что превышает норму допустимого содержания в почве в 30 раз. Неутилизированные компьютеры и оргтехника вызывают ртутное загрязнение литосферы и гидросферы, а при сжигании – и атмосферы.

Для оказания наименьшего влияния на окружающую среду, необходимо проводить специальную процедуру утилизации ПЭВМ и оргтехники, при которой более 90% отправится на вторичную переработку и менее 10% будут отправлены на свалки. При этом она должна соответствовать соответствующей процедуре утилизации [25].

В ходе деятельности также создавался бытовой мусор (канцелярские, пищевые отходы, искусственные источники освещения), который должен быть утилизирован в соответствии с определенным классом опасности или переработан, чтобы не оказывать негативное влияние на состояние литосферы и гидросферы.

6.5. Безопасность в чрезвычайных ситуациях

Анализ вероятных ЧС, которые могут возникнуть на рабочем месте при проведении исследования и разработке приложения

Работа по исследованию данных о характеристиках студента и создание веб-приложения на его основе происходила в офисе. На данном рабочем месте к возможным чрезвычайным ситуациям можно выделить: пожар, наводнение, землетрясение, удар молнией, взрыв, террористический акт.

С учетом наличия вычислительной техники в помещении наиболее вероятно возникновение пожара – неконтролируемого процесса горения, обусловленного возгоранием вычислительной техники и угрожающий жизни и здоровью работников.

Причинами пожара в помещении могут быть: токи короткого замыкания, неисправность устройства компьютера или электросетей, небрежность оператора при работе с компьютером, воспламенение ВЭВМ из-за перегрузки. Так же существуют причины неэлектрического характера, такие как курение, оставление без присмотра нагревательных приборов и неосторожное обращение с огнем.

Обоснование мероприятий по предотвращению ЧС и разработка порядка действия в случае возникновения ЧС

Согласно нормативному документу [26], при работе с компьютером требуется соблюдать следующие нормы:

- Для предохранение сети от перегрузок запрещается одновременно подключать к сети количество потребителей, превышающих допустимую нагрузку;
- Работы за компьютером следует проводить только при исправном состоянии оборудования и электропроводки;
- Иметь средства пожаротушения;

- Установить количество, размеры и соответствующее конструктивное исполнение эвакуационных путей и выходов;
- Обеспечить возможность беспрепятственного движения людей по эвакуационным путям.

При появлении пожара следует не поддаваться панике, организовать оповещение всех находящихся в здании людей и вызвать пожарную службу по номеру «01» или «112», четко назвав адрес учреждение, место возникновения пожара, а также свои должность, фамилию и номер телефона.

В случае возникновения пожара в здании автоматически должны срабатывать датчики пожаротушения, а звуковая система должна известить сотрудников о немедленной эвакуации из здания согласно плану эвакуации.

Вывод по разделу

В ходе работы над социальной частью ВКР были изучены правовые и организационные вопросы обеспечения безопасности. Рассмотрены производственная, экологическая безопасность, а также безопасность в чрезвычайных ситуациях, которые могут возникнуть в помещении при исследовании данных о студентах и разработке приложения.

Данная деятельность соответствовала всем заявленным нормам безопасности жизнедеятельности. Рабочее место во время исследования соответствовала указанным стандартам, а также санитарно-эпидемиологическим правилам и нормам. Следует отметить готовность исполнителей к чрезвычайным ситуациям.

Заключение

В результате выполнения бакалаврской работы был обработан и проанализирован датасет с данными студентов, сформулированы и проверены гипотезы, построена предсказательная модель, а также разработано веб-приложение, в котором реализованы функции загрузки данных, их обработки, анализа, визуализации, обучения прогнозной модели и возможность прогнозирования класса успешности студента посредством ввода новых данных, а также функция генерации отчета.

В ходе выполнения работы было проведено исследование предметной области, которое включает в себя ее описание, моделирование бизнес-процессов в нотации IDEF0, обзор и анализ существующих аналогов систем интеллектуального анализа. По результатам проведенного анализа было принято решение о разработке собственной программной системы.

Перед началом реализации был выполнен этап проектирования, в который входит определение ролей пользователей в системе и их возможности (диаграмма use-case), функциональное моделирование процессов (диаграммы в нотации IDEF0 и IDEF3, диаграммы последовательностей), моделирование потоков данных (диаграммы DFD, и наконец, описание объектов системы (диаграммы классов анализа и компонентов системы).

На этапе разработки веб-приложения были реализованы все необходимые модули: data_loader, выполняющий загрузку файлов на сервер; data_processor, выполняющий предобработку данных; data_visualizer, выполняющий построение графиков и выводов и их отображение на странице веб-приложения; predict_model_controller, выполняющий обучение прогнозной модели на указанном заранее предобработанном датасете, а также выполняющий предсказание класса успешности студента на основе введенных данных; report_controller, выполняющий генерацию отчета статистики по указанному датасету.

Выполнены задания по разделам «Финансовый менеджмент, ресурсоэффективность и ресурсосбережение» и «Социальная ответственность», показавшие возможность внедрения разрабатываемого приложения и его актуальность.

По итогам выполнения бакалаврской работы были закреплены и углублены теоретические знания, получены практические навыки обработки и анализа данных, проектирования и моделирования ИС и их разработки.

Полученные навыки удовлетворяют описанным ранее планируемым результатам обучения по профилю специальности «Программная инженерия».

Список публикаций студентов

Статьи:

1) Галлингер В.А. Построение предсказательной модели для определения успешности обучения студента // Наука. Технологии. Инновации: сборник научных трудов в 9 ч., Новосибирск, 30 Ноября-4 Декабря 2020. - Новосибирск: НГТУ, 2020 - Т. 2 - С. 170-173;
2) Галлингер В.А. Оценка влияния предобработки данных на работу предсказательных моделей // Молодежь и современные информационные технологии: сборник трудов XVIII Международной научно-практической конференции студентов, аспирантов и молодых ученых, Томск, 22–26 марта 2021 г. - Томск: ТПУ, 2021 (в печати).
3) Семенюта А. В. Предобработка данных о характеристиках студентов и проведение разведочного анализа с целью дальнейшего построения предсказательной модели // Наука. Технологии. Инновации: сборник научных трудов в 9 ч., Новосибирск, 30 Ноября-4 Декабря 2020. - Новосибирск: НГТУ, 2020 - Т. 2 - С. 221-225
4) Семенюта А.В. Предварительная обработка сырых данных о характеристиках студентов университета и разведочный анализ // Молодежь и современные информационные технологии: сборник трудов XVIII Международной научно-практической конференции студентов, аспирантов и молодых ученых, Томск, 22–26 марта 2021 г. - Томск: ТПУ, 2021 (в печати).

Список использованных источников

1. Data Mining [Электронный ресурс]. – Режим доступа: https://ru.wikipedia.org/wiki/Data_mining (дата обращения: 27.04.2021);
2. Стейкхолдер [Электронный ресурс]. – Режим доступа: https://ru.wikipedia.org/wiki/Стейкхолдер (дата обращения: 27.04.2021);
3. Информационная система [Электронный ресурс]. – Режим доступа: https://ru.wikipedia.org/wiki/Информационная_система (дата обращения: 27.04.2021);
4. SVM [Электронный ресурс]. – Режим доступа: https://ru.wikipedia.org/wiki/Метод_опорных_векторов (дата обращения: 27.04.2021);
5. Django [Электронный ресурс]. – Режим доступа: https://developer.mozilla.org/ru/docs/Learn/Server-side/Django/Introduction (дата обращения: 27.04.2021);
6. Machine Learining в нефтегазовой области [Электронный ресурс]. – Режим доступа: https://www.bigdataschool.ru/blog/machine-learning-в-нефтегазовой-отрасли.html (дата обращения: 08.04.2021);
7. Образовательная аналитика [Электронный ресурс]. – Режим доступа: http://www.edutainme.ru/post/learning-analytics/ (дата обращения: 08.04.2021);
8. Аналитика больших данных и Machine Learning в образовании: 5 кейсов из ВУЗов [Электронный ресурс]. – Режим доступа: https://www.bigdataschool.ru/blog/big-data-analytics-education-cases.html (дата обращения: 08.04.2021);
9. Bachelor's degree student dropouts: Who tend to stay and who tend to leave? [Электронный ресурс]. – Режим доступа: https://www.scopus.com/record/display.uri?eid=2-s2.0-85103298223&origin=resultslist&sort=plf-f&src=s&sid=ef4dc2586158ea7ef725d38e515ebd96&sot=b&sdt=b&sl=36&s=

TITLE-ABS-KEY%28Data+Mining+education%29&relpos=3&citeCnt=0&searchTerm= (дата обращения: 15.04.2021);

10. Automatic Assessment of Students' Engineering Design Performance Using a Bayesian Network Model [Электронный ресурс]. – Режим доступа: https://www.scopus.com/record/display.uri?eid=2-s2.0-85091452576&origin=resultslist&sort=plf-f&src=s&nlo=&nlr=&nls=&sid=ef4dc2586158ea7ef725d38e515ebd96&sot=b&sdt=b&sl=36&s=TITLE-ABS-KEY%28Data+Mining+education%29&relpos=20&citeCnt=0&searchTerm= (дата обращения: 15.04.2021);
11. Behind the scenes of educational data mining [Электронный ресурс]. – Режим доступа: https://www.scopus.com/record/display.uri?eid=2-s2.0-85090153832&origin=resultslist&sort=plf-f&src=s&nlo=&nlr=&nls=&sid=ef4dc2586158ea7ef725d38e515ebd96&sot=b&sdt=b&sl=36&s=TITLE-ABS-KEY%28Data+Mining+education%29&relpos=32&citeCnt=0&searchTerm= (дата обращения: 15.04.2021);
12. Data Analytics Applications in Education [Электронный ресурс]. – Режим доступа: https://www.researchgate.net/publication/320226279_Data_Analytics_Applications_in_Education (дата обращения: 15.04.2021);
13. Data Analytics Applications in Education [Электронный ресурс]. – Режим доступа: https://www.researchgate.net/publication/254462830_Course_signals_at_Purdue_Using_learning_analytics_to_increase_student_success (дата обращения: 15.04.2021);
14. Data Analytics Applications in Education [Электронный ресурс]. – Режим доступа: https://www.teachingandlearning.ie/publication/using-learning-

analytics-to-support-the-enhancement-of-teaching-and-learning-in-higher-education (дата обращения: 15.04.2021);

15. Производственный календарь на 2021 год [Электронный ресурс]. – Режим доступа: http://www.consultant.ru/law/ref/calendar/proizvodstvennye/2021/ (дата обращения: 13.04.2021);
16. Тарифы страховых взносов по ОПС, ОСС, ОМС [Электронный ресурс]. – Режим доступа: http://www.consultant.ru/document/cons_doc_LAW_93256/ff7c924d59eb608ca9b6f0a34739935e5eb0f0fe/ (дата обращения: 13.04.2021);
17. Постановление Правительства РФ от 1 января 2002 г. №1 «О Классификации основных средств, включаемых в амортизационные группы» [Электронный ресурс]. – Режим доступа: https://base.garant.ru/12125271/#:~:text=Постановление%20Правительства%20РФ%20от%201,г.%2C%2027%20декабря%202019%20г (дата обращения: 13.04.2021);
18. Трудовой кодекс Российской Федерации от 30.12.2001 N 197-ФЗ (ред. от 30.04.2021). – URL: https://docs.cntd.ru/document/901807664 (дата обращения: 03.05.2021). – Текст: электронный;
19. ГОСТ 12.2.032–78. Система стандартов безопасности труда (ССБТ). Рабочее место при выполнении работ сидя. Общие эргономические требования. – М.: ИПК Издательство стандартов, 2001. – URL: https://docs.cntd.ru/document/1200003913 (дата обращения: 03.05.2021). – Текст: электронный;
20. ГОСТ 22269-76. Система "Человек-машина". Рабочее место оператора. Взаимное расположение элементов рабочего места. Общие эргономические требования. – М.: Издательство стандартов, 1990. – URL: https://docs.cntd.ru/document/1200012834 (дата обращения: 03.05.2021). – Текст: электронный;
21. ГОСТ 21889-76. Система "Человек-машина". Кресло человека-оператора. Общие эргономические требования. – М.: Издательство стандартов, 1993.

– URL: https://docs.cntd.ru/document/1200012832 (дата обращения: 03.05.2021). – Текст: электронный;

22. ГОСТ 12.0.003-2015. Система стандартов безопасности труда (ССБТ). Опасные и вредные производственные факторы. Классификация. – М.: Стандартинформ, 2019. – URL: https://docs.cntd.ru/document/1200136071 (дата обращения: 03.05.2021). – Текст: электронный;
23. СанПиН 1.2.3685-21. Гигиенические нормативы и требования к обеспечению безопасности и (или) безвредности для человека факторов среды обитания. – М., 2021. – URL: https://docs.cntd.ru/document/573500115 (дата обращения: 03.05.2021). – Текст: электронный;
24. ГОСТ 12.1.038-82. Предельно допустимые значения напряжений прикосновения и токов. – М.: ИПК Издательство стандартов, 2001. – URL: https://docs.cntd.ru/document/5200313 (дата обращения: 03.05.2021). – Текст: электронный;
25. ГОСТ Р 53692-2009. Ресурсосбережение. Обращение с отходами. Этапы технологического цикла отходов. – М.: Стандартинформ, 2019. – URL: https://docs.cntd.ru/document/1200081740 (дата обращения: 03.05.2021). – Текст: электронный;
26. ГОСТ 12.1.004-91. Пожарная безопасность. Общие требования. – М.: Стандартинформ, 2006. – URL: https://docs.cntd.ru/document/9051953 (дата обращения: 03.05.2021). – Текст: электронный.

Приложение А

Листинг views.py приложения data_loader.

```
import os

from django.shortcuts import render
from .forms import UploadFileForm

import pandas as pd


def preview_df(df):
    return df.iloc[1:11].values, list(df.columns)


def dataset(request, file_path):
    df = pd.read_excel(f'uploads/datasets/{file_path}')
    first_rows, col_names = preview_df(df)
    return render(request, 'data_loader/dataset.html', {'first_rows':
first_rows, 'col_names': col_names})

def processed_dataset(request, file_path):
    df = pd.read_excel(f'uploads/processed_datasets/{file_path}')
    first_rows, col_names = preview_df(df)
    return render(request, 'data_loader/dataset.html', {'first_rows':
first_rows, 'col_names': col_names})

def filelist(request):
    filelist = os.listdir(path='uploads/datasets')
    processed_filelist = os.listdir(path='uploads/processed_datasets')
    if request.method == 'POST':
        form = UploadFileForm(request.POST, request.FILES)
        if form.is_valid():
            handle_uploaded_file(request.FILES['file'])
            return render(request, 'data_loader/filelist.html', {'form':
form, 'filelist': filelist,

'processed_filelist': processed_filelist})
    else:
        form = UploadFileForm
    return render(request, 'data_loader/filelist.html', {'form': form,
'filelist': filelist,
                                                'processed_filelist':
processed_filelist})


def handle_uploaded_file(f):
    title = f.name
    with open(f'uploads/datasets/{title}', 'wb+') as destination:
        for chunk in f.chunks():
            destination.write(chunk)
```

Листинг views.py приложения data_processor.

```
from django.shortcuts import redirect
from django.shortcuts import render

from .forms import DataProcessForm
from .forms import DatasetForm

import numpy as np
import pandas as pd
import re


def dataset_process(request):
    if request.method == 'POST':
        form = DatasetForm(request.POST)
        if form.is_valid():
            file_name = form.cleaned_data['dataset']
            return redirect('data_processor:dataprocess', file_name)
    else:
        form = DatasetForm
    return render(request, 'data_processor/dataset_process.html',
{'form': form})


def dataprocess(request, file_name):
    category_cols = ['Форма обучения', 'Квалификация', 'Курс',
'Специальность',
                     'Профиль', 'Выпуск. отдел.', 'Выпуск. школа',
'Группа',
                     'Обуч. подразд.', 'Форма финансирования', 'Страна',
'Гражданство',
                     'Пол', 'Дата рождения', 'Академ отпуск (действующий)
- да / нет',
                     'Дисциплины по которым получены неудовлетворительные
оценки']
    numeric_cols = ['Всего', 'Положительных', 'Неудовлетворительных',
                    'Пропусков по дисциплинам по которым получены
неудовлетворительные оценки',
                    'Всего часов по дисциплинам по которым получены
неудовлетворительные оценки',
                    'Всего часов пропусков в семестре', 'Всего часов
аудиторных занятий в семестре']
    if request.method == 'POST':
        form = DataProcessForm(file_name, request.POST)
        if form.is_valid():
            df = pd.read_excel(f"uploads/datasets/{file_name}")
            temp_list = file_name.split('.')[:-1]
            name_without_ext = ''
            for element in temp_list:
                name_without_ext += element

            # if statements
            if form.cleaned_data['linear_dep'] == "Исключить линейные
зависимости":
                delete_linear_dependencies(df)
                name_without_ext += f'-DLD'
```

```
            cols_to_del = []
            for numeric_col in numeric_cols:
                if numeric_col not in df.columns:
                    cols_to_del.append(numeric_col)
            numeric_cols = list(set(numeric_cols) - set(cols_to_del))

            if form.cleaned_data['outlayers'] == "Удалить записи с
выбросом":
                clean_outliers(df, numeric_cols)
                name_without_ext += f'-CO'

            if form.cleaned_data['missing'] == "Удалить запись с
миссингом":
                clean_missings_dropna(df)
                name_without_ext += f'-CMD'

            if form.cleaned_data['missing'] == "Заменить медианой
признака для числовых, категорией \"Нет\" " \
                                               "для категориальных
признаков":
                clean_missings_fillna(df, category_cols, numeric_cols)
                name_without_ext += f'-CMF'

            if form.cleaned_data['facult'] == "Удалить факультативы":
                delete_extra_disciplines(df)
                name_without_ext += f'-DED'

            # reformat_cols_type
            reformat_cols_type(df, category_cols, numeric_cols)

            if form.cleaned_data['target_val'] == "Сформировать целевые
переменные":
                create_target_vars(df)
                name_without_ext += f'-CTV'

            if form.cleaned_data['hand_not_inf']:
                drop_unnecessary_cols(df,
form.cleaned_data['hand_not_inf'])
                for col in form.cleaned_data['hand_not_inf']:
                    if col in category_cols:
                        category_cols.remove(col)
                    elif col in numeric_cols:
                        numeric_cols.remove(col)
                name_without_ext += f'-HDUC'

            # automatic not-informative columns processing
            if form.cleaned_data['not_inf'] == "Обработать
неинформативные признаки":
                unnecessary_cols_list = ['Дата рождения', 'Всего',
'Положительных', 'Неудовлетворительных',
                                         'Группа', 'Страна',
                                         'Дисциплина по которым получены
неудовлетворительные оценки',
                                         'Индекс студента', 'Выпуск.
школа',
                                         'Всего часов по дисциплинам по
которым получены неудовлетворительные оценки',
                                         'Пропусков по дисциплинам по
```

```
которым получены неудовлетворительные оценки',
                                        'Всего часов пропусков в
семестре', 'Всего аудиторных занятий в семестре']
                drop_unnecessary_cols(df, unnecessary_cols_list)
                for col in unnecessary_cols_list:
                    if col in category_cols:
                        category_cols.remove(col)
                    elif col in numeric_cols:
                        numeric_cols.remove(col)
                name_without_ext += f'-DUC'


df.to_excel(f"uploads/processed_datasets/{name_without_ext}.xlsx",
index=False)

            return render(request, 'data_processor/process_result.html',
{'file_name': f'{name_without_ext}.xlsx'})
    else:
        form = DataProcessForm(file_name)
    return render(request, 'data_processor/dataprocess.html', {'form':
form, 'file_name': file_name})


def drop_unnecessary_cols(df, unnecessary_cols_list):
    cols_to_del = []
    for col in unnecessary_cols_list:
        if col not in df.columns:
            cols_to_del.append(col)
    unnecessary_cols_list = list(set(unnecessary_cols_list) -
set(cols_to_del))
    df.drop(unnecessary_cols_list, axis=1, inplace=True)


def clean_missings_dropna(df):
    df.dropna(inplace=True)
    df.reset_index(inplace=True)


def clean_missings_fillna(df, category_cols, numeric_cols):
    for cat_col in category_cols:
        df[cat_col].fillna(value='Нет', inplace=True)
    for col in numeric_cols:
        df[col].fillna(value=df[col].median(), inplace=True)


def reformat_cols_type(df, category_cols, numeric_cols):
    df[category_cols].astype('category')
    df[numeric_cols].astype(float)


def clean_outliers(df, numeric_cols):
    # IQR (interquartile range) = Q3-Q1  # Q1 - 1.5*IQR  # Q3 + 1.5*IQR
    df_quarntiled = df[numeric_cols].quantile([.25, .75])
    for num_col in list(df_quarntiled.columns):
        median = df[num_col].median()
        quantiles = df[num_col].values
        iqr = quantiles[1] - quantiles[0]
        left_mustache = quantiles[0] - 1.5 * iqr
```

```
        right_mustache = quantiles[1] + 1.5 * iqr
        for i in range(len(df)):
            if left_mustache < df[num_col][i] < right_mustache:
                df[num_col][i] = median


def delete_extra_disciplines(df):
    facult_list = ['Второй иностранный язык \(немецкий\).
A2.1\(Зач.\),[.]?',
                   'Второй иностранный язык \(немецкий\).
A2.1\(Зач.\)[.]?',
                   'Второй иностранный язык \(немецкий\).
A1.1\(Зач.\),[.]?',
                   'Второй иностранный язык \(немецкий\).
A1.1\(Зач.\)[.]?',
                   'Второй иностранный язык \(китайский\).
1\(Зач.\),[.]?',
                   'Второй иностранный язык \(китайский\).
1\(Зач.\)[.]?',
                   'Второй иностранный язык \(французский\).
A1.1\(Зач.\),[.]?',
                   'Второй иностранный язык \(французский\).
A1.1\(Зач.\)[.]?',
                   'Иностранный язык для программ академической
мобильности \(английский\). A2.2\(Зач.\),[.]?',
                   'Иностранный язык для программ академической
мобильности \(английский\). A2.2\(Зач.\)[.]?',
                   'Управление проектами\(Зач.\),[.]?', 'Управление
проектами\(Зач.\)[.]?',
                   'Факультативные дисциплины по выбору
студента\(Зач.\),[.]?',
                   'Факультативные дисциплины по выбору
студента\(Зач.\)[.]?',
                   'Креативность инженера\(Зач.\),[.]?', 'Креативность
инженера\(Зач.\)[.]?']

    def discount(strings):
        if len(re.findall(strings[0], strings[1])) == 0:
            return 0
        else:
            return 1

    for facult in facult_list:
        bin_column = df['Дисциплины по которым получены
неудовлетворительные оценки'] \
            .apply(lambda s: discount([facult, s]))
        df[['Всего', 'Неудовлетворительных']] = df[['Всего',
'Неудовлетворительных']].sub(bin_column, axis=0)
        df.replace({facult: ''}, inplace=True, regex=True)

    df.replace(r'^\s*$', np.nan, inplace=True, regex=True)
    df['Дисциплины по которым получены неудовлетворительные оценки'] =
df[
        'Дисциплины по которым получены неудовлетворительные
оценки'].str.strip()
    df.replace({' +': ' '}, inplace=True, regex=True)
    df['Дисциплины по которым получены неудовлетворительные
оценки'].fillna('Нет', inplace=True)
```

```
def delete_linear_dependencies(df):
    corr = df.corr()
    column_list = list(corr.columns)
    for j in range(len(column_list)):
        for i in range(len(column_list)):
            if j == i:
                break
            if corr[column_list[j]][i] > 0.75:
                df.drop(column_list[j], axis=1, inplace=True)


def create_target_vars(df):
    df['Успешность'] = df['Положительных'] / df['Всего']

    def classify(success):
        if success >= 0.75:
            return 2
        elif success > 0.25:
            return 1
        else:
            return 0

    df['Класс'] = df['Успешность'].apply(classify)


def process_result(request):
    return render(request, 'data_processor/process_result.html')
```

Листинг views.py приложения data_visualizer.

```
import copy
import math
import json
import os
import sys
from pathlib import Path

from django.shortcuts import render

import pandas as pd
import scipy.stats

ROOT_DIR = Path(__file__).resolve().parent.parent
if str(ROOT_DIR) not in sys.path:
    sys.path.append(str(ROOT_DIR))

cat_cols = ['Форма обучения', 'Квалификация', 'Курс', 'Специальность',
'Профиль',
            'Выпуск. отдел.', 'Выпуск. школа', 'Группа', 'Обуч.
подразд.',
            'Форма финансирования', 'Страна',
            'Гражданство', 'Пол', 'Дата рождения',
            'Академ отпуск (действующий) - да / нет',
            'Дисциплины по которым получены неудовлетворительные оценки',
'Класс']
```

```
num_cols = ['Всего', 'Положительных', 'Неудовлетворительных',
            'Пропусков по дисциплинам по которым получены
неудовлетворительные оценки',
            'Всего часов по дисциплинам по которым получены
неудовлетворительные оценки',
            'Всего часов пропусков в семестре',
            'Всего часов аудиторных занятий в семестре', 'Успешность']


def statistics(request, file_path):
    df = pd.read_excel(f'uploads/datasets/{file_path}')
    df_cols = df.columns
    df_list = []
    for i, ind in enumerate(list(df.describe().index)):
        df_list.append([ind] + list(df.describe().values[i]))
    desc_cols = df.describe().columns

    df_cat_cols = []
    df_num_cols = []
    p_list = []

    for cat_col in cat_cols:
        if cat_col in df_cols:
            p_value1, p_value2, i_norm, pval, feature, feature_value =
build_hypothises(df, cat_col)
            if p_value1 is None:
                print('p value is None')
                continue
            df_cat_cols.append(
                cat_col.replace('(', '_').replace(')', '_').replace('/',
'').replace('.', '_').replace('-', '_').replace(' ', '_'))
            histogram_builder(df, cat_col)
            p_list.append({'p_value1': p_value1, 'p_value2': p_value2,
                           'i_norm': i_norm, 'pval': pval,
                           'feature': feature.replace('(',
'_').replace(')', '_').replace('/', '').replace(
                               '.', '_').replace('-', '_').replace(' ',
'_'),
                           'feature_value': feature_value})

    for num_col in num_cols:
        if num_col in df_cols:
            df_num_cols.append(
                num_col.replace('(', '_').replace(')', '_').replace('/',
'').replace('.', '_').replace('-', '_').replace(' ', '_'))
            histogram_builder(df, num_col)
            boxplot_builder(df, num_col)

    return render(request, 'data_visualizer/statistics.html', {'df_list':
df_list,
                                                       'df_cols': [''] +
list(df_cols),
                                                       'desc_cols': [''] +
list(desc_cols),
                                                       'df_cat_cols':
df_cat_cols,
                                                       'df_num_cols':
df_num_cols,
```

```
                                                'p_list': p_list})


def processed_statistics(request, file_path):
    df = pd.read_excel(f'uploads/processed_datasets/{file_path}')
    df_cols = df.columns
    df_list = []
    for i, ind in enumerate(list(df.describe().index)):
        df_list.append([ind] + list(df.describe().values[i]))
    desc_cols = df.describe().columns

    df_cat_cols = []
    df_num_cols = []
    p_list = []

    for cat_col in cat_cols:
        if cat_col in df_cols:
            p_value1, p_value2, i_norm, pval, feature, feature_value =
build_hypothises(df, cat_col)
            if p_value1 is None:
                print('p value is None')
                continue
            df_cat_cols.append(
                cat_col.replace('(', '_').replace(')', '_').replace('/',
'').replace('.', '_').replace('-', '_').replace(' ', '_'))
            histogram_builder(df, cat_col)
            p_list.append({'p_value1': p_value1, 'p_value2': p_value2,
                           'i_norm': i_norm, 'pval': pval,
                           'feature': feature.replace('(',
'_').replace(')', '_').replace('/', '').replace(
                               '.', '_').replace('-', '_').replace(' ',
'_'),
                           'feature_value': feature_value})

    for num_col in num_cols:
        if num_col in df_cols:
            df_num_cols.append(
                num_col.replace('(', '_').replace(')', '_').replace('/',
'').replace('.', '_').replace('-', '_').replace(' ', '_'))
            histogram_builder(df, num_col)
            boxplot_builder(df, num_col)

    return render(request, 'data_visualizer/statistics.html', {'df_list':
df_list,
                                                'df_cols': [''] +
list(df_cols),
                                                'desc_cols': [''] +
list(desc_cols),
                                                'df_cat_cols':
df_cat_cols,
                                                'df_num_cols':
df_num_cols,
                                                'p_list': p_list})


def dataset_stat(request):
    filelist = os.listdir(path='uploads/datasets')
    processed_filelist = os.listdir(path='uploads/processed_datasets')
```

```
    return render(request, 'data_visualizer/dataset_stat.html',
{'filelist': filelist,
                                                    'processed_filelist':
processed_filelist})


# content = [ list(labels), [(str(label), list(data)] ]
def linechart_to_json(content, index):
    labels = content[0]
    datasets = []
    for el in content[1]:
        datasets.append({
            "label": el[0],
            "data": el[1]
        })
    areaChartData = {"labels": labels, "datasets": datasets}
    write_to_file(f"{ROOT_DIR}\\chart-txt\\LineChartData-{index}.txt",
                  f"LineChartData{index} =
'[{json.dumps(areaChartData)}]'")


# uploads/json/LinearChartData.txt


# content = [ list(labels), [(str(label), list(data)] ]
def barchart_to_json(content, index):
    labels = content[0]
    datasets = []
    for el in content[1]:
        datasets.append({
            "label": el[0],
            "data": el[1]
        })
    areaChartData = {"labels": labels, "datasets": datasets}
    write_to_file(f"{ROOT_DIR}\\static\\chart-txt\\BarChartData-
{index}.txt",
                  f"BarChartData{index} =
'[{json.dumps(areaChartData)}]'")


# content = [ list(labels), list(line_data), str(label), list(data) ]
# data = {int(x), int(y)}
def q_q_plot_to_json(content, index):
    labels = content[0]
    line_data = content[1]
    label = content[2]
    data = content[3]
    areaChartData = {"labels": labels, "line_data": line_data, "label":
label, "scatter_data": data}
    write_to_file(f"{ROOT_DIR}\\static\\chart-txt\\QQChartData-
{index}.txt",
                  f"QQChartData{index} =
'[{json.dumps(areaChartData)}]'")


# content = [ str(x), int(low), int(q1), int(median), int(q3), int(high),
list(outliers)]
# if Q3 + 1.5*IQR < value < Q1 - 1.5*IQR:
```

```
# outliers.append(value)
def boxplot_to_json(content, index):
    data = []
    for el in content:
        data.append({"x": el[0], "low": el[1], "q1": el[2],
                     "median": el[3], "q3": el[4], "high": el[5],
"outliers": el[6]})
    areaChartData = {"data": data}
    write_to_file(f"{ROOT_DIR}\\static\\chart-txt\\BoxplotChartData-
{index}.txt",
                  f"BoxplotChartData{index} =
'[{json.dumps(areaChartData)}]'")


# content = [ list(labels), str(label), list(data) ]
def catplot_to_json(content, index):
    labels = content[0]
    label = content[1]
    datasets = content[2]
    areaChartData = {"labels": labels, "label": label, "datasets":
datasets}
    write_to_file(f"{ROOT_DIR}\\static\\chart-txt\\CatplotChartData-
{index}.txt",
                  f"CatplotChartData{index} =
'[{json.dumps(areaChartData)}]'")


def write_to_file(filepath, string):
    with open(filepath, 'w+', encoding="utf-8") as writer:
        writer.write(string)


# histogram

def histogram_builder(df, cat_feature):
    labels = list(df[cat_feature].unique())
    label = f'Гистограмма распределения признака {cat_feature}'
    bar_chart_data = list(df[cat_feature].value_counts().values)
    zip_list = zip(labels, bar_chart_data)
    zip_list = sorted(zip_list)
    res_list = [[i for i, j in zip_list],
                [j for i, j in zip_list]]
    labels = res_list[0]
    bar_chart_data = res_list[1]

    if str(type(labels[0])) == "<class 'numpy.int64'>":
        labels_np_dtype = copy.copy(labels)
        labels = []
        for l in labels_np_dtype:
            labels.append(int(l))

    if str(type(bar_chart_data[0])) == "<class 'numpy.int64'>":
        bar_chart_data_np_dtype = copy.copy(bar_chart_data)
        bar_chart_data = []
        for dat in bar_chart_data_np_dtype:
            bar_chart_data.append(int(dat))

    bar_chart_content = [labels, [(label, bar_chart_data)]]
```

```
    barchart_to_json(bar_chart_content,
                     cat_feature.replace('(', '_').replace(')',
'_').replace('/', '').replace(
                         '.', '_').replace('-', '_').replace(' ', '_'))


# boxplot
# content = [ str(x), int(low), int(q1), int(median), int(q3), int(high),
list(outliers)]
def boxplot_builder(df, num_feature):
    x = "1"
    q1 = df[num_feature].quantile(.25)
    median = df[num_feature].quantile(.5)
    q3 = df[num_feature].quantile(.75)
    iqr = q3 - q1
    low = q1 - 1.5 * iqr
    if low < 0:
        low = 0
    high = q3 + 1.5 * iqr
    outliers = []
    for value in df[num_feature]:
        if value < low or value > high:
            outliers.append(value)

    boxplot_content = [[x, low, q1, median, q3, high, outliers]]
    boxplot_to_json(boxplot_content,
                    num_feature.replace('(', '_').replace(')',
'_').replace('/', '').replace(
                        '.', '_').replace('-', '_').replace(' ', '_'))


# q-q plot


# slope, intercept, r
# = slope*x + intercept

def qq_plot_builder(df, cat_feature):
    df_grouped = df[[cat_feature,
'Успешность']].groupby(cat_feature).mean()
    feature_value = list(df_grouped[df_grouped['Успешность'] ==
max(df_grouped['Успешность'])].index)
    tup01, tup02 = scipy.stats.probplot(df[df[cat_feature] ==
feature_value[0]].Успешность, dist="norm")
    tup11, tup12 = scipy.stats.probplot(df[df[cat_feature] !=
feature_value[0]].Успешность, dist="norm")
    # for df[df[feature] == feature_value[0]:
    min_scale0 = math.floor(tup01[0][0])
    max_scale0 = math.ceil(tup01[0][-1])

    labels0 = list(range(min_scale0, max_scale0 + 1))
    line_data0 = []
    for i in range(len(labels0)):
        line_data0.append(tup02[0] * labels0[i] + tup02[1])
    label0 = f"Вероятность успешности по квантилям, где значение признака
{cat_feature} равно {feature_value[0]}"
    qq_plot_data0 = []
    for i in range(len(tup01[0])):
```

```
        if i % 50 == 0:
            qq_plot_data0.append({"x": tup01[0][i], "y": tup01[1][i]})
    qq_plot_data0.append({"x": tup01[0][-1], "y": tup01[1][-1]})

    qq_chart_content0 = [labels0, line_data0, label0, qq_plot_data0]

    q_q_plot_to_json(qq_chart_content0, cat_feature.replace('(',
'_').replace(')', '_').replace('/', '').replace(
        '.', '_').replace('-', '_').replace(' ', '_'))

    # for df[df[feature] != feature_value[0]:
    min_scale1 = math.floor(tup11[0][0])
    max_scale1 = math.ceil(tup11[0][-1])

    labels1 = list(range(min_scale1, max_scale1 + 1))
    line_data1 = []
    for i in range(len(labels1)):
        line_data1.append(tup12[0] * labels1[i] + tup12[1])
    label1 = f"Вероятность успешности по квантилям, где значение признака
{cat_feature} не равно {feature_value[0]}"
    qq_plot_data1 = []
    for i in range(len(tup11[0])):
        if i % 50 == 0:
            qq_plot_data1.append({"x": tup11[0][i], "y": tup11[1][i]})
    qq_plot_data1.append({"x": tup11[0][-1], "y": tup11[1][-1]})

    qq_chart_content1 = [labels1, line_data1, label1, qq_plot_data1]

    q_q_plot_to_json(qq_chart_content1,
                     f'Не_{cat_feature}'.replace('(', '_').replace(')',
'_').replace('/', '').replace(
                         '.', '_').replace('-', '_').replace(' ', '_'))


# catplot
def catplot_builder(df, cat_feature):
    data_grouped = df[[cat_feature,
'Успешность']].groupby(cat_feature).mean()
    labels = list(data_grouped.index)
    label = f"Catplot признака {cat_feature}"
    catplot_data = list(data_grouped.values)

    if str(type(catplot_data[0][0])) == "<class 'numpy.float64'>":
        catplot_data_np_dtype = copy.copy(catplot_data)
        catplot_data = []
        for dat in catplot_data_np_dtype:
            catplot_data.append(float(dat[0]))

    catplot_content = [labels, label, catplot_data]

    catplot_to_json(catplot_content, cat_feature.replace('(',
'_').replace(')', '_').replace('/', '').replace(
        '.', '_').replace('-', '_').replace(' ', '_'))


# hypothises
def build_hypothises(df, feature):
    df_grouped = df[[feature, 'Успешность']].groupby(feature).mean()
```

```
    feature_value = list(df_grouped[df_grouped['Успешность'] ==
max(df_grouped['Успешность'])].index)
    if len(df[df[feature] == feature_value[0]]) < 3:
        return None, None, None, None, None, None
    qq_plot_builder(df, feature)
    w_statistic, p_value1 = scipy.stats.shapiro(df[df[feature] ==
feature_value[0]].Успешность)
    w_statistic, p_value2 = scipy.stats.shapiro(df[df[feature] !=
feature_value[0]].Успешность)
    if p_value1 < 0.01 or p_value2 < 0.01:
        statistic, pvalue = scipy.stats.ttest_ind(df[df[feature] ==
feature_value[0]].Успешность,
                                                  df[df[feature] !=
feature_value[0]].Успешность)
        i_norm = 0
    else:
        statistic, pvalue = scipy.stats.mannwhitneyu(df[df[feature] ==
feature_value[0]].Успешность,
                                                     df[df[feature] !=
feature_value[0]].Успешность)
        i_norm = 1
    catplot_builder(df, feature)
    return p_value1, p_value1, i_norm, pvalue, feature, feature_value[0]
```

Листинг views.py приложения predict_model_controller.

```
import os
import sys
from pathlib import Path

from django.shortcuts import render
from .forms import ModelPredictForm
from sklearn.svm import SVC
from sklearn.model_selection import train_test_split
import numpy as np
import pandas as pd
import joblib

ROOT_DIR = Path(__file__).resolve().parent.parent
if str(ROOT_DIR) not in sys.path:
    sys.path.append(str(ROOT_DIR))


def svc_model(request, file_path):
    df = pd.read_excel(f"uploads/processed_datasets/{file_path}")
    df['Курс'] = df['Курс'].astype('category')
    if 'Успешность' in df.columns:
        df_prepared = df.drop(['Успешность'], axis=1)
    else:
        df_prepared = df
    df_dummy = pd.get_dummies(df_prepared)
    x_train, y_train, mean_list, std_list =
get_train_test_samples(df_dummy)
    clf = SVC(probability=True).fit(x_train, y_train)
    col_list = list(df_dummy.columns)
    save_model(clf, col_list, mean_list, std_list)
```

```
    return render(request,
'predict_model_controller/teaching_model_result.html', {'file_path':
file_path})


def predict_form(request):
    if request.method == 'POST':
        form = ModelPredictForm(data=request.POST)
        if form.is_valid():
            lock = False
            model, mean_std_list, feature_list = load_model('SVC0')
            params_dict = {'Форма обучения':
form.cleaned_data['educ_form'],
                           'Квалификация':
form.cleaned_data['qualification'],
                           'Курс': form.cleaned_data['course'],
                           'Профиль': form.cleaned_data['profile'],
                           'Выпуск. отдел.': form.cleaned_data['dep'],
                           'Обуч. подразд.': form.cleaned_data['subdep'],
                           'Форма финансирования':
form.cleaned_data['fin'],
                           'Гражданство':
form.cleaned_data['citizenship'],
                           'Пол': form.cleaned_data['gender'],
                           'Академ отпуск (действующий) - да / нет':
form.cleaned_data['vacation']}
            df = pd.DataFrame(params_dict, index=[0])
            data_dummy = pd.get_dummies(df)
            dummy_values = np.zeros(len(feature_list)).tolist()
            params_keys = list(data_dummy.columns)
            for param in params_keys:
                if param not in feature_list:
                    print("Входные данные не соответствуют требуемым")
                    lock = True
                    break
                else:
                    dummy_values[feature_list.index(param)] = 1
            if lock:
                print("Невозможно выполнить предсказание. Данные numpy
формата не сформированы")
                return render(request,
'predict_model_controller/predict_form.html', {'form': form})
            else:
                stud_prob = model.predict_proba([dummy_values])[0]
                print(stud_prob)
                ind_max = np.argmax(stud_prob)

                return render(request,
'predict_model_controller/predict_result.html', {'stud_class': ind_max,

'stud_prob': stud_prob[ind_max]})
    else:
        form = ModelPredictForm
    return render(request, 'predict_model_controller/predict_form.html',
{'form': form})


def norm_data(df_dummy):  # built-in function
```

```
    mean_list = []
    std_list = []
    for feature in (df_dummy.columns):
        if str(df_dummy[feature].dtype) == 'int64' or
str(df_dummy[feature].dtype) == 'float64':
            mean = np.mean(df_dummy[feature])
            std = np.std(df_dummy[feature])
            df_dummy[feature] = (df_dummy[feature] - mean) / std
            mean_list.append(mean)
            std_list.append(std)
    return mean_list, std_list


def get_train_test_samples(df_dummy):
    target = df_dummy['Класс']
    df_dummy.drop(['Класс'], axis=1, inplace=True)
    mean_list, std_list = norm_data(df_dummy)
    values = df_dummy.values
    x_train, x_test, y_train, y_test = train_test_split(values, target,
test_size=1, random_state=42)
    return x_train, y_train, mean_list, std_list


def save_model(model, col_list, mean_list, std_list):
    with open(f'uploads/models/SVC0_feat_1.txt', 'w') as writer:
        for col in col_list:
            writer.write(f"{col}\n")
    # сохраняем в файл веса mean std
    with open(f'uploads/models/SVC0_norm_w.txt', 'w') as writer:
        for i in range(len(mean_list)):
            writer.write(f"{mean_list[i]} {std_list[i]}\n")
    # сохраняем в файл веса модели
    _ = joblib.dump(model, f"uploads/models/SVC0.joblib.pkl")


def load_model(model_name):
    model = joblib.load(f"uploads/models/{model_name}.joblib.pkl")
    mean_std_list = []
    feature_list = []
    with open(f'uploads/models/{model_name}_norm_w.txt', 'r') as reader:
        str_list = reader.read().split('\n')
        for el in str_list:
            if str(el) != '':
                mean_std_list.append([float(el.split(' ')[0]),
float(el.split(' ')[1])])
    with open(f'uploads/models/{model_name}_feat_1.txt', 'r') as reader:
        str_list = reader.read().split('\n')
        for el in str_list:
            if str(el) != '':
                feature_list.append(el)
    return model, mean_std_list, feature_list


def dataset_teaching_model(request):
    filelist = os.listdir(path='uploads/datasets')
    processed_filelist = os.listdir(path='uploads/processed_datasets')

    return render(request,
```

```
'predict_model_controller/dataset_teaching_model.html', {'filelist':
filelist,

'processed_filelist': processed_filelist})


def teaching_model_info(request):
    return render(request, 'teaching_model_info.html')


def teaching_model_result(request, file_path):
    return render(request,
'predict_model_controller/teaching_model_result.html', {'file_path':
file_path})


def predict_result(request):
    return render(request,
'predict_model_controller/predict_result.html')
```

Листинг views.py приложения report_controller.

```
import copy
import itertools
import os

from django.http import HttpResponse
from django.shortcuts import render

import numpy as np
import pandas as pd
from matplotlib import pyplot as plt
from matplotlib import pylab as plb
import seaborn as sns
from scipy import stats
from reportlab.lib import colors
from reportlab.lib.pagesizes import A4
from reportlab.lib.styles import getSampleStyleSheet, ParagraphStyle
from reportlab.lib.units import mm
from reportlab.lib.enums import TA_JUSTIFY, TA_CENTER, TA_LEFT
from reportlab.lib.fonts import addMapping
from reportlab.pdfbase import pdfmetrics
from reportlab.pdfbase.ttfonts import TTFont
from reportlab.platypus import SimpleDocTemplate, Paragraph, Image,
Spacer, TableStyle, Table

CATEGORY_COLUMNS = ['Форма обучения', 'Квалификация', 'Курс',
'Специальность',
                    'Профиль', 'Выпуск. отдел.', 'Выпуск. школа',
'Группа',
                    'Обуч. подразд.', 'Форма финансирования', 'Страна',
'Гражданство',
                    'Пол', 'Дата рождения', 'Академ отпуск (действующий)
- да / нет',
                    'Дисциплины по которым получены неудовлетворительные
оценки', 'Курс', 'Класс']
```

```
NUMERIC_COLUMNS = ['Всего', 'Положительных', 'Неудовлетворительных',
                   'Пропусков по дисциплинам по которым получены
неудовлетворительные оценки',
                   'Всего часов по дисциплинам по которым получены
неудовлетворительные оценки',
                   'Всего часов пропусков в семестре', 'Всего часов
аудиторных занятий в семестре', 'Успешность']

title_label = 6
figsize_width = 2.2
figsize_height = 1.4
x_label = 6
y_label = 6
xy_tick = 5


def dataset_report(request):
    filelist = os.listdir(path='uploads/datasets')
    processed_filelist = os.listdir(path='uploads/processed_datasets')
    return render(request, 'report_controller/dataset_report.html',
{'filelist': filelist,
                                                    'processed_filelist':
processed_filelist})


def report_result(request, file_path):
    response = pdf_report(f'uploads/datasets/{file_path}')
    return response


def processed_report_result(request, file_path):
    response = pdf_report(f'uploads/processed_datasets/{file_path}')
    return response


def addPageNumber(canvas, doc):
    page_num = canvas.getPageNumber()
    canvas.drawRightString(200 * mm, 10 * mm, str(page_num))


def pdf_report(dataset_filepath: str):
    plt.switch_backend('agg')
    df = pd.read_excel(dataset_filepath)
    df_cols = list(df.columns)
    if 'Курс' in df_cols:
        df['Курс'] = df['Курс'].astype('object')
    if 'Класс' in df_cols:
        df['Класс'] = df['Класс'].astype('object')
    cat_cols = []
    num_cols = []
    for col in df_cols:
        if col in CATEGORY_COLUMNS:
            cat_cols.append(col)
        elif col in NUMERIC_COLUMNS:
            num_cols.append(col)

    df_filename = dataset_filepath.split('/')[-1]
    df_name = ''
```

```
    string = df_filename.split('.')[:-1]
    for s in string:
        df_name += s
    pdf_filename = df_name + '_results.pdf'

    response = HttpResponse(content_type='application/pdf')
    response['Content-Disposition'] = 'attachment; filename="%s"' %
pdf_filename

    pdfmetrics.registerFont(TTFont('Times', 'times.ttf', 'UTF-8'))
    pdfmetrics.registerFont(TTFont('Times-Bold', 'timesbd.ttf', 'UTF-8'))
    pdfmetrics.registerFont(TTFont('Times-Italic', 'timesi.ttf', 'UTF-
8'))
    pdfmetrics.registerFont(TTFont('Times-BoldItalic', 'timesbi.ttf',
'UTF-8'))

    addMapping('Times', 0, 0, 'Times')  # normal
    addMapping('Times', 0, 1, 'Times-Italic')  # italic
    addMapping('Times', 1, 0, 'Times-Bold')  # bold
    addMapping('Times', 1, 1, 'Times-BoldItalic')  # italic and bold

    doc = SimpleDocTemplate(response, pagesize=A4, rightMargin=40,
leftMargin=40, topMargin=20, bottomMargin=40,
                            title='Результаты')
    story = []
    styles = getSampleStyleSheet()
    styles.add(ParagraphStyle(name='Justify', alignment=TA_JUSTIFY,
fontName='Times', fontSize=11))
    styles.add(ParagraphStyle(name='Justify-Bold', alignment=TA_JUSTIFY,
fontName='Times-Bold'))
    normal_style = styles['Justify']
    doc_title = copy.copy(styles["Heading1"])
    doc_title.alignment = TA_LEFT
    doc_title.fontName = 'Times-Bold'
    doc_title.fontSize = 16
    logo = 'static/img/tpu-logo.png'
    im = Image(logo)
    story.append(im)
    story.append(Spacer(1, 10))
    title1 = "Статистика по характеристикам студентов за x семестр y
года"
    story.append(Paragraph(title1, doc_title))

    data = df_describe(df)
    table = Table(data)
    table.setStyle(TableStyle([
        ('FONT', (0, 0), (-1, -1), 'Times', 10),
        ('ALIGN', (0, 0), (0, -1), 'RIGHT'),
        ('GRID', (0, 0), (-1, -1), 0.25, colors.black),
    ]))
    story.append(table)
    story.append(Spacer(1, 10))

    for col in df_cols:
        hist = df_hist(df, col, df_name)
        him = Image(hist)  # , 70 * mm, 40 * mm
        story.append(him)
        story.append(Spacer(1, 10))
```

```
        if col in num_cols:
            boxplot = df_boxplot(df, col, df_name)
            bim = Image(boxplot)  # , 70 * mm, 40 * mm
            story.append(bim)
            story.append(Spacer(1, 10))


    title2 = "Поиск скрытых зависимостей"
    story.append(Paragraph(title2, doc_title))
    doc_title.fontSize = 12

    for cat_col in cat_cols:
        title3 = cat_col
        story.append(Paragraph(title3, doc_title))

        catplot = df_catplot(df, cat_col, df_name)
        cim = Image(catplot)
        story.append(cim)
        story.append(Spacer(1, 10))

        qq_plot = df_qqplot(df, cat_col, df_name)
        qim = Image(qq_plot)
        story.append(qim)
        story.append(Spacer(1, 10))

        stats_str, result = df_hypothises(df, cat_col)
        story.append(Paragraph(stats_str, normal_style))
        for res in result:
            story.append(Paragraph(res, normal_style))
        story.append(Spacer(1, 10))

    doc.build(story, onFirstPage=addPageNumber,
onLaterPages=addPageNumber)
    return response


def df_describe(df):
    df_desc = df.describe()
    df_desc_cols = ['']
    df_desc_cols += list(df_desc.columns)
    df_desc_idx = list(df_desc.index)
    data = []
    for i in range(len(df_desc_cols)):
        if len(df_desc_cols[i]) < 11:
            continue
        if len(df_desc_cols[i].split(' ')) == 1:
            continue
        c_inc = int(len(df_desc_cols[i]) / 2)
        c_decr = int(len(df_desc_cols[i]) / 2)
        while df_desc_cols[i][c_inc] != ' ' and df_desc_cols[i][c_decr]
!= ' ':
            if c_inc > len(df_desc_cols[i]):
                print(f"Переполнение c_inc: {c_inc}")
            if c_decr < 0:
                print(f"Переполнение c_decr: {c_decr}")
            c_inc += 1
            c_decr -= 1
        ch_list = []
```

```
            for ch in df_desc_cols[i]:
                ch_list.append(ch)
            if df_desc_cols[i][c_inc] == ' ':
                ch_list[c_inc] = '\n'
                df_desc_cols[i] = ''.join(ch_list)
            elif df_desc_cols[i][c_decr] == ' ':
                ch_list[c_decr] = '\n'
                df_desc_cols[i] = ''.join(ch_list)
        data.append(df_desc_cols)
        for i in range(len(df_desc)):
            tmp = [df_desc_idx[i]]
            for col in list(df_desc.columns):
                tmp.append(df_desc[col][i])
            data.append(tmp)
        return data


def df_hist(df, col, filename):
    path = f'static/img/hist'
    if not os.path.exists(path):
        os.mkdir(path)

    fig, ax = plt.subplots()
    if str(df[col].dtype) == 'object':
        val_cnt = df[col].value_counts()[:10]
        val_cnt_idx = list(val_cnt.index)
        val_cnt_vals = list(val_cnt.values)
        new_dict = {}
        for i in range(len(val_cnt)):
            if str(type(val_cnt_idx[i])) != "<class 'int'>" and
len(val_cnt_idx[i]) > 30:
                val_cnt_idx[i] = ' '.join(val_cnt_idx[i].split(' ')[:5])
+ '...'
            new_dict[val_cnt_idx[i]] = val_cnt_vals[i]
        ax.barh(list(reversed(list(new_dict.keys()))),
list(reversed(list(new_dict.values()))))
        if len(list(df[col].unique())) > 10:
            ax.set_xlabel('Кол-во значений', fontsize=x_label)
            ax.set_ylabel('Топ 10 значений признака', fontsize=y_label)
        else:
            ax.set_xlabel('Кол-во значений', fontsize=x_label)
            ax.set_ylabel('Значения признака', fontsize=y_label)
    else:
        ax.hist(df[col])
        ax.set_xlabel('Значения признака', fontsize=x_label)
        ax.set_ylabel('Кол-во значений', fontsize=y_label)
    ax.set_title(col, fontsize=title_label)
    fig.set_figwidth(figsize_width)
    fig.set_figheight(figsize_height)
    plt.tick_params(axis='both', labelsize=xy_tick)

    col_repaired = ','.join(col.split('/'))
    hist_filepath = f"{path}/hist_{filename}_{col_repaired}.png"
    plt.savefig(hist_filepath, bbox_inches='tight')
    return hist_filepath


def df_boxplot(df, numeric_col, filename):
```

```
    path = f'static/img/boxplot'
    if not os.path.exists(path):
        os.mkdir(path)

    fig, ax = plt.subplots()
    ax.boxplot(df[numeric_col], vert=False, labels=[''])
    ax.set_title(numeric_col, fontsize=title_label)
    fig.set_figwidth(figsize_width)
    fig.set_figheight(figsize_height)
    plt.tick_params(axis='both', labelsize=xy_tick)

    boxplot_filepath = f"{path}/boxplot_{filename}_{numeric_col}.png"
    plt.savefig(boxplot_filepath, bbox_inches='tight')
    return boxplot_filepath


def df_catplot(df, cat_col, filename):
    path = f'static/img/catplot'
    if not os.path.exists(path):
        os.mkdir(path)

    if len(list(df[cat_col].unique())) > 5:
        df_grouped = df[[cat_col, 'Успешность']].groupby(cat_col).mean()
\
            .sort_values(by='Успешность', ascending=False)
        feature_values = list(df_grouped[:4].index) + ['Все остальные']
        for i in range(len(feature_values)):
            if len(feature_values[i]) > 30:
                feature_values[i] = ' '.join(feature_values[i].split('
')[:5]) + '...'
        feature_data = []
        for i in range(4):
            feature_data.append(df_grouped.values[i][0])
        feature_data.append(df_grouped[4:].mean().values[0])
        df_catplot_dat = pd.DataFrame({cat_col: feature_values,
'Успешность': feature_data})
        if cat_col in ['Профиль', 'Выпуск. отдел.', 'Обуч. подразд.',
'Гражданство']:
            g = sns.catplot(y=cat_col, x='Успешность', kind='point',
data=df_catplot_dat, orient='h', height=3)
            g.ax.set_xlabel('Успешность', fontsize=x_label)
            g.ax.set_ylabel(cat_col, fontsize=y_label)
        else:
            g = sns.catplot(x=cat_col, y='Успешность', kind='point',
data=df_catplot_dat)
            g.set_xticklabels(rotation=30)
            g.ax.set_xlabel(cat_col, fontsize=x_label)
            g.ax.set_ylabel('Успешность', fontsize=y_label)
    elif cat_col == 'Форма финансирования':
        g = sns.catplot(y=cat_col, x='Успешность', kind='point', data=df,
orient='h', height=3)
        g.ax.set_xlabel('Успешность', fontsize=x_label)
        g.ax.set_ylabel(cat_col, fontsize=y_label)
    else:
        g = sns.catplot(x=cat_col, y='Успешность', kind='point', data=df)
        g.ax.set_xlabel(cat_col, fontsize=x_label)
        g.ax.set_ylabel('Успешность', fontsize=y_label)
    g.fig.set_figwidth(figsize_width)
```

```
    g.fig.set_figheight(figsize_height)
    plt.tick_params(axis='both', labelsize=xy_tick)

    col_repaired = ','.join(cat_col.split('/'))
    catplot_filepath = f"{path}/catplot_{filename}_{col_repaired}.png"
    g.savefig(catplot_filepath, bbox_inches='tight')
    return catplot_filepath


def df_qqplot(df, cat_col, filename):
    path = f'static/img/qq_plot'
    if not os.path.exists(path):
        os.mkdir(path)

    df_grouped = df[[cat_col, 'Успешность']].groupby(cat_col).mean()
    feature_value = list(df_grouped[df_grouped['Успешность'] ==
max(df_grouped['Успешность'])].index)[0]
    feat_name = feature_value
    if str(type(feature_value)) != "<class 'int'>" and len(feature_value)
> 20:
        feat_name = feature_value.split(' ')[1] + '...' +
feature_value.split(' ')[-1]
    plb.figure(figsize=(6, 4))
    ax1 = plb.subplot(2, 2, 1)
    stats.probplot(df[df[cat_col] == feature_value].Успешность,
dist="norm", plot=plb)
    ax2 = plb.subplot(2, 2, 2)
    stats.probplot(df[df[cat_col] != feature_value].Успешность,
dist="norm", plot=plb)

    ax1.set_title(f'Q-Q Plot значения {feat_name}', fontsize=title_label)
    ax1.set_xlabel('Теоретические квантили', fontsize=x_label)
    ax1.set_ylabel('Упорядоченные значения', fontsize=y_label)
    ax1.tick_params(axis='both', labelsize=xy_tick)

    ax2.set_title(f'Q-Q Plot значения не {feat_name}',
fontsize=title_label)
    ax2.set_xlabel('Теоретические квантили', fontsize=x_label)
    ax2.set_ylabel('Упорядоченные значения', fontsize=y_label)
    ax2.tick_params(axis='both', labelsize=xy_tick)

    col_repaired = ','.join(cat_col.split('/'))
    qqplot_filepath = f"{path}/qq_plot_{filename}_{col_repaired}.png"
    plb.savefig(qqplot_filepath, bbox_inches='tight')
    return qqplot_filepath


def df_hypothises(df, col):
    df_grouped = df[[col, 'Успешность']].groupby(col).mean()
    feature_value = list(df_grouped[df_grouped['Успешность'] ==
max(df_grouped['Успешность'])].index)[0]

    if len(df[df[col] == feature_value]) < 3:
        p_value1 = -1
    else:
        _, p_value1 = stats.shapiro(df[df[col] ==
feature_value].Успешность)
```

```
    if len(df[df[col] != feature_value]) < 3:
        p_value2 = -1
    else:
        _, p_value2 = stats.shapiro(df[df[col] !=
feature_value].Успешность)

    stats_str = f"P-значение критерия Шапиро-Уилка равно: {p_value1} и
{p_value2} соответственно."
    if p_value1 == -1 or p_value2 == -1:
        stats_str = "Недостаточно данных. Необходимо как минимум 3
значения"
        result = ['Для проведения статистических тестов не хватает
данных']
    elif (0 < p_value1 <= 0.01) or (0 < p_value2 <= 0.01):
        _, pval = stats.ttest_ind(df[df[col] ==
feature_value].Успешность,
                                  df[df[col] !=
feature_value].Успешность)
        result = [f"Распределение нормальное. Использован t-критерий
Стьюдента.",
                  f"P-значение t-критерия Стьюдента равно {pval}"]
        if pval > 0.01:
            result.append("Вывода по данному признаку сформировать не
удалось.")
        else:
            result.append(f"Студенты, обладающие значением
{feature_value} признака {col} обучаются успешнее.")
    else:
        _, pval1 = stats.mannwhitneyu(df[df[col] ==
feature_value].Успешность,
                                      df[df[col] !=
feature_value].Успешность)
        result = [f"Распределение ненормальное. Использован критерий
Манна-Уитни.",
                  f"P-значение критерия Манна-Уитни равно {pval1}"]
        if pval1 > 0.01:
            result.append("Вывода по данному признаку сформировать не
удалось.")
        else:
            result.append(f"Студенты, обладающие значением
{feature_value} признака {col} обучаются успешнее.")

    return stats_str, result


def permutation_t_stat_ind(sample1, sample2):
    return np.mean(sample1) - np.mean(sample2)


def get_random_combinations(n1, n2, max_combinations):
    index = list(range(n1 + n2))
    indices = {tuple(index)}
    for i in range(max_combinations - 1):
        np.random.shuffle(index)
        indices.add(tuple(index))
    return [(index[:n1], index[n1:]) for index in indices]
```

```
def permutation_zero_dist_ind(sample1, sample2, max_combinations=None):
    joined_sample = np.hstack((sample1, sample2))
    n1 = len(sample1)
    n = len(joined_sample)

    if max_combinations:
        indices = get_random_combinations(n1, len(sample2),
max_combinations)
    else:
        indices = [(list(index), filter(lambda i: i not in index,
range(n)))
                   for index in itertools.combinations(range(n), n1)]

    distr = [joined_sample[list(i[0])].mean() -
joined_sample[list(i[1])].mean() for i in indices]
    return distr


def permutation_test(sample, mean, max_permutations=None,
alternative='two-sided'):
    if alternative not in ('two-sided', 'less', 'greater'):
        raise ValueError("alternative not recognized\n"
                         "should be 'two-sided', 'less' or 'greater'")
    t_stat = permutation_t_stat_ind(sample, mean)
    zero_distr = permutation_zero_dist_ind(sample, mean,
max_permutations)
    if alternative == 'two-sided':
        return sum([1. if abs(x) >= abs(t_stat) else 0. for x in
zero_distr]) / len(zero_distr)
    if alternative == 'less':
        return sum([1. if x <= t_stat else 0. for x in zero_distr]) /
len(zero_distr)
    if alternative == 'greater':
        return sum([1. if x >= t_stat else 0. for x in zero_distr]) /
len(zero_distr)
```