МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное автономное образовательное учреждение  
высшего образования

**«Южно-Уральский государственный университет**

**(национальный исследовательский университет)»**

**Высшая школа электроники и компьютерных наук**

**Кафедра системного программирования**

**Разработка приложения для прогноза успеваемости студентов на основе данных портала Универис**

КУРСОВАЯ РАБОТА

по дисциплине «Программная инженерия»

ЮУрГУ – 09.03.04.20231.308-5692.КР

|  |  |
| --- | --- |
| Нормоконтролер3,  доктор физ.-мат. наук  \_\_\_\_\_\_\_\_\_\_\_\_\_\_М.Л. Цымблер  “\_\_\_”\_\_\_\_\_\_\_\_\_\_\_ 2023 г. | Научный руководитель:  доктор физ.-мат. наук  \_\_\_\_\_\_\_\_\_\_М.Л. Цымблер  Автор работы:  студент группы КЭ-303  \_\_\_\_\_\_\_\_\_\_Д.В. Старостенок  Работа защищена  с оценкой: \_\_\_\_\_\_\_\_\_\_\_  “\_\_\_”\_\_\_\_\_\_\_\_\_\_\_ 2023 г. |

Челябинск 2023

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное автономное образовательное   
учреждение высшего образования

**«Южно-Уральский государственный университет**

**(национальный исследовательский университет)»**

**Высшая школа электроники и компьютерных наук**

**Кафедра системного программирования**

УТВЕРЖДАЮ

Зав. кафедрой СП

\_\_\_\_\_\_\_\_\_\_ Л.Б. Соколинский

06.02.2023 г.

**ЗАДАНИЕ**

**на выполнение курсовой работы**

по дисциплине «Программная инженерия»

студенту группы КЭ-303

Старостенок Дмитрию Владимировичу,

обучающемуся по направлению 09.03.04 «Программная инженерия»

1. **Тема работы**

Разработка приложения для прогноза успеваемости студентов на основе данных портала Универис.

1. **Срок сдачи студентом законченной работы:** 31.05.2023 г.
2. **Исходные данные к работе**
   1. Документация API Универис DigitalTrace
   2. Scikit learn documentation. [Электронный ресурс] URL: <https://scikit-learn.org/stable/user_guide.html>
   3. Pandas documentation. [Электронный ресурс] URL https://pandas.pydata.org/docs/reference/index.html
   4. VanderPlas Jake Python Data Science Handbook: Essential Tools for Working with Data // O'Reilly Media 2nd edition, 2023
3. **Перечень подлежащих разработке вопросов**
   1. Провести анализ и выполнить спецификацию предметной области.
   2. Изучить API Универис.
   3. Разработать прогнозные модели успеваемости получаемые из API Универис.
   4. Выполнить проектирование интерфейса пользователя и модульной структуры приложения.
   5. Выполнить реализацию, отладку и тестирование приложения.
   6. Провести вычислительные эксперименты по исследованию точности прогноза.
4. **Дата выдачи задания:** 9 февраля 2023 г.

Научный руководитель М.Л. Цымблер

доктор физ.-мат. наук

Задание принял к исполнению Д.В. Старостенок

**Оглавление**

[ВВЕДЕНИЕ 5](#_Toc136457865)

[1. ОБЗОР РАБОТ ПО ТЕМАТИКЕ ИССЛЕДОВАНИЯ 7](#_Toc136457866)

[2. АНАЛИЗ ПРЕДМЕТНОЙ ОБЛАСТИ 10](#_Toc136457867)

[2.1. Постановка задачи 10](#_Toc136457868)

[2.2. Описание данных 10](#_Toc136457869)

[3. ПРОЕКТИРОВАНИЕ 14](#_Toc136457870)

[3.1. Требования к системе 14](#_Toc136457871)

[3.2. Варианты использования системы 15](#_Toc136457872)

[3.3. Предобработка данных 16](#_Toc136457873)

[3.4. Прогнозные модели 16](#_Toc136457874)

[3.5. Модуль прогнозирования 20](#_Toc136457875)

[3.6. Графический интерфейс 21](#_Toc136457876)

[4. РЕАЛИЗАЦИЯ 25](#_Toc136457877)

[4.1. Программные средства реализации 26](#_Toc136457878)

[4.2. Модуль извлечения данных 28](#_Toc136457879)

[4.3. Модуль предобработки 29](#_Toc136457880)

[4.4. Модуль прогнозных моделей 30](#_Toc136457881)

[4.5. Модуль анализа данных 31](#_Toc136457882)

[4.6. Реализация интерфейса 32](#_Toc136457883)

[5. ТЕСТИРОВАНИЕ 36](#_Toc136457884)

[5.1. Тестирование извлечения данных из API и CSV 36](#_Toc136457885)

[5.2. Тестирование анализа данных 38](#_Toc136457886)

[ЗАКЛЮЧЕНИЕ 44](#_Toc136457887)

[ЛИТЕРАТУРА 45](#_Toc136457888)

[ПРИЛОЖЕНИЯ 47](#_Toc136457889)

[Приложение А. Модуль извлечения данных 47](#_Toc136457890)

[Приложение Б. Модуль предобработки 49](#_Toc136457891)

[Приложение В. Модуль прогнозных моделей 50](#_Toc136457892)

[Приложение Г. Модуль анализа данных 52](#_Toc136457893)

[Приложение Д. Модуль интерфейса 53](#_Toc136457894)

ВВЕДЕНИЕ

**Актуальность работы**

В связи с тем, в настоящее время образование стало незаменимой составляющей жизни современного общества. В связи с этим, повышение эффективности образовательного процесса и улучшение качества обучения являются одними из важнейших задач в данной области. Данные портала Универис, которые содержат информацию об успеваемости студентов, являются ценным ресурсом для анализа и прогнозирования успеха студентов.

Разработка приложения для прогноза успеваемости студентов на основе данных портала Универис позволит не только предсказывать успех студентов на основе их текущих оценок и статистики по их учебной деятельности, но и предоставлять студентам и преподавателям инструменты для повышения эффективности обучения.

Такое приложение также может быть полезно для администрации университета, так как оно позволит им отслеживать прогресс студентов и принимать меры для улучшения качества образования. Кроме того, такое приложение может быть использовано для определения факторов, которые влияют на успеваемость студентов, и для разработки соответствующих программ для улучшения образовательной среды.

**Цель и задачи**

Целью данной работы является разработка приложения для прогноза успеваемости студентов на основе данных портала Универис. В качестве исходных данных используются информация об абитуриентах и их успеваемости в виде журналов.

Для достижения поставленной цели необходимо решить следующие задачи:

1. Провести анализ предметной области и на его основе разработать прогнозные модели;
2. Разработать алгоритм получения и предобработки данных;
3. Разработать приложение, выполняющее прогноз на основе моделей с помощью различных методов анализа данных;
4. Провести эксперименты, исследующие разработанное приложение.

Результатом выполнения данной работы будет создание приложения, которое позволит преподавателям и студентам получать прогнозы успеваемости на основе имеющихся данных, что поможет улучшить качество обучения.

**Структура и содержание работы.**

Курсовая работа состоит из введения, пяти разделов, заключения и списка литературы. Объем работы составляет 53 страниц, объем библиографического списка составляет – 16 наименований.

В первом разделе, «Обзор работ по тематике исследования», содержится обзор на работы по тематике исследования.

Во втором разделе, «Анализ предметной области», описывается постановка задачи и описание данных, которые будут использоваться для анализа.

В третьем разделе, «Проектирование», определены требования к системе, описаны модели данных и структура приложения.

В четвертом разделе, «Реализация», приведены диаграммы системы и описана реализация компонентов системы.

В пятом разделе, «Тестирование», описано функциональное тестирование работы интерфейса и эксперименты с разработанными моделями данных.

В приложении А содержится документация модуля извлечения данных.

В приложении Б документация модуля предобработки.

В приложении В документация модуля прогнозных моделей.

В приложении Г документация модуля анализа данных и в приложении Д, документация модуля интерфейса.

# ОБЗОР РАБОТ ПО ТЕМАТИКЕ ИССЛЕДОВАНИЯ

С ростом количества студентов в вузах, администрации университета становится все сложнее контролировать успеваемость каждого студента индивидуально. В связи с этим возникают потребности в создании системы, которая позволит прогнозировать успеваемость студентов. Для этого необходимо произвести анализ данных о студентах, их предыдущих успехах и неудачах, а также данных об их прогрессе в течение текущего учебного года.

Чтобы осуществить анализ в работе [13] используются деревья решений для предсказания успеваемости студентов, используются различные параметры и настройки, и оцениваются их производительность, используя различные метрики, такие как точность, полнота и F-мера. Проводится анализ важности функций, чтобы определить наиболее значимые переменные для прогнозирования успеваемости студентов.

Также работа [11] показывает возможность использования ансамблевых моделей машинного обучения и метода Shapley Additive Explanations (SHAP) для предсказания и интерпретации успеваемости студентов. Используют собранные данные, которые содержат информацию о студентах, такую как возраст, образование, опыт работы и оценки за задания и экзамены, которые проходят предварительную обработку данных, включая удаление пропущенных значений, преобразование категориальных переменных в числовые и стандартизацию данных.

Далее используют несколько алгоритмов машинного обучения, такие как случайный лес, градиентный бустинг и логистическая регрессию, для создания ансамблевых моделей, которые могут предсказывать успеваемость студентов.

А для выбора наиболее значимых признаков используется метод SHAP, чтобы определить наиболее важные признаки, влияющие на успеваемость студентов. Этот метод позволяет оценить вклад каждого признака в конечный результат и показывает, как каждый признак взаимодействует с другими.

В работе [1] авторы используют искусственные нейронные сети (ИНС) для прогнозирования успеваемости студентов. Используется алгоритм обратного распространения ошибки для обучения ИНС и применяются методы статистической оценки для оценки точности модели.

Для предобработки данных и извлечения признаков используются стандартные методы машинного обучения, такие как масштабирование данных и кодирование категориальных признаков.

Анализ данных реализуется с использованием языка программирования Python и фреймворка TensorFlow. Данные предварительно обрабатываются и извлекаются признаки, после чего модель обучается и оценивается с помощью стандартных методов машинного обучения и статистических методов оценки.

Авторы работы [5] применяют алгоритм градиентного бустинга XGBoost для прогнозирования успеваемости.

Для анализа использовались демографические данные, социально-экономические данные и данные об успеваемости, к которым применяется алгоритм XGBoost для построения регрессионной модели и прогнозирования успеваемости. Для настройки гиперпараметров модели XGBoost используется сеточный поиск и анализ важности признаков, чтобы определить наиболее значимые признаки для прогнозирования успеваемости, а также аддитивные объяснения Шепли (SHAP) для интерпретации прогнозов модели и понимания влияния различных признаков на результаты модели.

Для работы [7] применяется алгоритм случайного леса. Авторы предварительно обрабатывают данные, кодируя категориальные признаки, заполняя недостающие значения и нормализуя данные. После предварительной обработки набор данных разделяется на обучающий и тестовый наборы.

Затем применяется алгоритм случайного леса к обучающему множеству для построения классификационной модели, позволяющей предсказать, будет ли студент хорошо или плохо успевать по предметам. Используется сеточный поиск для настройки гиперпараметров модели случайного леса, а также анализ важности признаков, чтобы определить наиболее значимые признаки для прогнозирования успеваемости.

В работе [12] применяется нейронная сеть с прямой передачей и улучшенный алгоритм оптимизации роя частиц (PSO) для прогнозирования академической успеваемости. Собираются данные студентов, которые предварительно обрабатывают данные, кодируя категориальные переменные, заполняя недостающие значения и нормализуя данные.

Далее применяется нейронная сеть с прямой передачей и улучшенным алгоритм PSO для оптимизации гиперпараметров нейронной сети, включая количество скрытых слоев и нейронов в каждом слое, после, используется отбор признаков, чтобы определить наиболее значимые признаки для прогнозирования успеваемости.

# АНАЛИЗ ПРЕДМЕТНОЙ ОБЛАСТИ

В разделе 2.1. описывается постановка задачи. В разделе 2.2. выполняется описание данных, используемых для прогнозирования.

* 1. Постановка задачи

Важной частью разрабатываемого приложения является анализ успеваемости студентов, который осуществляется на основе данных из журналов, которые хранятся в системе Универис. Журналы содержат информацию о студентах и их успеваемости. Важным этапом является выбор оптимальных методов анализа данных, которые учитывают особенности имеющихся данных, а также определение наиболее значимых факторов, влияющих на успеваемость студентов. Это позволит создать модели, которые точно прогнозируют успеваемость студентов и предоставляют полезную информацию для принятия решений.

Для разработки приложения будут использоваться различные методы классификации и регрессии, которые позволят выявить закономерности, которые могут быть полезны для прогнозирования будущей успеваемости студентов. Важным шагом в данном проекте будет выбор оптимальных методов анализа данных, учитывая особенности имеющихся данных, а также определение наиболее значимых факторов, влияющих на успеваемость студентов, и создание моделей, которые позволят прогнозировать успеваемость на их основе.

* 1. Описание данных

Далее описаны основные атрибуты запросов к API, используемые для получения информации об учебном процессе в университете. Каждый запрос имеет свой набор атрибутов, соответствующих конкретной информации. Подробное описание каждого атрибута позволяет лучше понимать, какую информацию можно получить через API, и как ее использовать для решения различных задач.

Рисунок 1 представляет диаграмму связей таблиц, используемых при запросах к API, она помогает лучше понять отношение между используемыми данными.

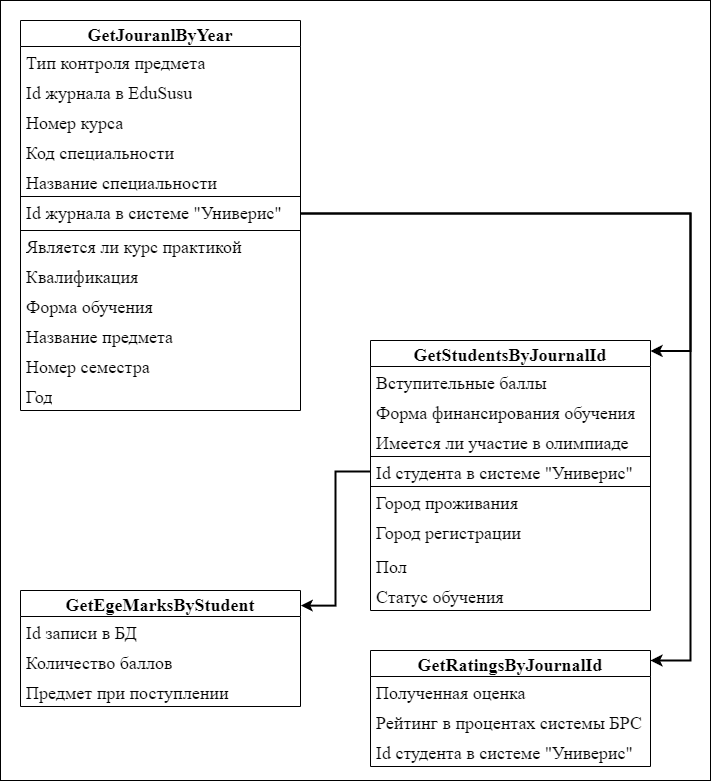


Рисунок – Диаграмма связей используемых таблиц

Первый запрос «GetJournalByYear» позволяет получить информацию о журнале по заданному году (Таблица 1).

Таблица – Описание атрибутов результата запроса GetJournalByYear

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Атрибут** | **Семантика** | | **Пример** | **Примечание** |
| CheckType | Тип контроля предмета | | Зачет, экзамен, аттестация, практика | - |
| CourseEduId | Id журнала в системе “EduSusu” | | 152872 | - |
| CourseNumber | Номер курса | | С 1 по 6 | - |
| DirectionCode | Код специальности | | 15.03.03 | - |
| DirectionName | Название специальности | Прикладная механика | | - |

Окончание таблицы 1

|  |  |  |  |
| --- | --- | --- | --- |
| **Атрибут** | **Семантика** | **Пример** | **Примечание** |
| Id | Id журнала в системе Универис | 5ef36ac9-67de-430c-a1dc-000d081c752c | Используется далее для получения информации о студентах и их успеваемости |
| IsPractice | Является ли курс практикой | true, false | - |
| Speciality | Квалификация | бакалавр, специалист | - |
| StudyForm | Форма обучения | очная, заочная | - |
| SubjectName | Название предмета | Электротехника и электроника | - |
| Term | Семестр, в котором проходит предмет | 2 | - |
| Year | Год | 2021 | - |

Второй запрос «GetStudentsByJournalId» предназначен для получения информации о студентах по журналу (Таблица 2).

Таблица – Описание атрибутов, результата запроса GetStudentsByJournalId

|  |  |  |  |
| --- | --- | --- | --- |
| **Атрибут** | **Семантика** | **Пример** | **Примечание** |
| EnrollScore | Вступительные баллы | 252 | - |
| FinancialForm | Форма финансирования обучения | бюджет | - |
| HasOlymp Participation | Имеется ли участие в олимпиаде | false | - |
| Id | Id студента в системе Универис | ad69ba03-36fb-4c4e-8c75-14b26da8ef38 | Используется для получения результатов ЕГЭ |
| LiveCity | Город проживания | г.Челябинск | - |
| RegisterCity | Город регистрации | г.Челябинск | - |
| Sex | Пол | Мужской, женский | - |
| Status | Статус обучения | учится, отчислен | - |

Третий запрос «GetRatingsByJournalId» предоставляет информацию о рейтинге студентов по журналу (Таблица 3).

Таблица – Описание атрибутов, результата запроса GetRatingsByJournalId

|  |  |  |  |
| --- | --- | --- | --- |
| **Атрибут** | **Семантика** | **Пример** | **Примечание** |
| Mark | Полученная оценка | 3 | - |
| Rating | Рейтинг в процентах по системе БРС | 60.77 | - |
| StudentId | Id студента в системе Универис | 9b6db037-5e67-4331-bdb5-35c5d75d5235 | - |

Четвертый запрос «GetEgeMarksByStudentId» предоставляет информацию о количестве баллов по ЕГЭ и предметах, с которыми поступал абитуриент (Таблица 4).

Таблица – Описание атрибутов, результата запроса GetEgeMarksByStudentId

|  |  |  |  |
| --- | --- | --- | --- |
| **Атрибут** | **Семантика** | **Пример** | **Примечание** |
| Id | Id записи предмета | 34306cea-7c32-4ad5-b923-eb3d123886c0 | - |
| Mark | Количество баллов | 64 | - |
| Subject | Предмет, с которым поступал абитуриент | Математика | - |

В целом, все атрибуты запросов к API направлены на получение различных характеристик студентов, их образования и результатов обучения. Они будут использованы для анализа и сравнения успеваемости студентов в различных сферах обучения.

# ПРОЕКТИРОВАНИЕ

В разделе 3.1. представлены функциональные и нефункциональные требования к системе. В разделе 3.2. описаны варианты использования системы. В разделе 3.3. описывается предобработка данных. В разделе 3.4. представлено описание прогнозных моделей. В разделе 3.5. представлен описан модуль прогнозирования, а в разделе 3.6. описан планируемый графический интерфейс.

* 1. Требования к системе

**Функциональные требования.**

В ходе анализа работы были определены следующие функциональные требования к разрабатываемой системе:

1. Система должна предоставлять возможность делать запросы к системе Универис для получения данных с последующим их сохранением;
2. Система должна предоставлять возможность настраивать получаемые данные;
3. Система должна предоставлять возможность просматривать получаемые данные;
4. Система должна иметь возможность выбрать алгоритм классификации или регрессии;
5. Система должна предоставлять возможность обучать аналитическую модель по получаемым данным.

**Нефункциональные требования.**

Нефункциональные требования представлены далее.

1. Реализация интерфейса, обработки и анализа данных должна быть выполнена на языке программирования Python;
2. Данные для анализа должны быть получены из системы Универис.
   1. Варианты использования системы

Для описания способов взаимодействия с системой была разработана диаграмма вариантов использования, которую показывает Рисунок 2.

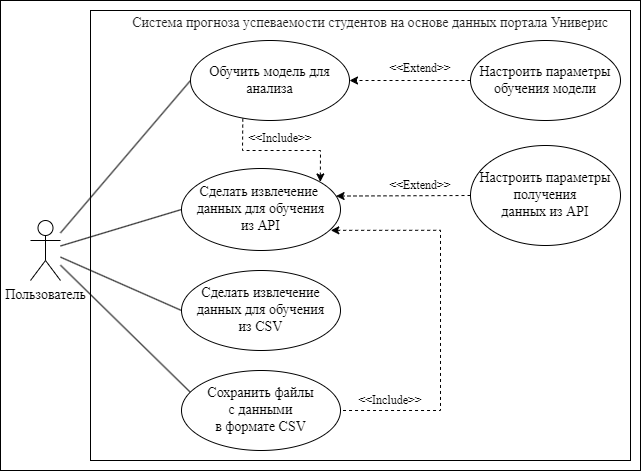


Рисунок – Диаграмма вариантов использования

Единственным актером, взаимодействующим с системой, является пользователь. Для него определены следующие варианты использования.

1. Сделать извлечение данных для обучения из API. Пользователь осуществляет загрузку данных с помощью запросов к API.
2. Настроить параметры получения данных из API. Пользователь задает определенные параметры выборки из системы Универис.
3. Сделать извлечение данных для обучения из CSV. Пользователь загружает CSV файлы с данными в программу.
4. Сохранить файлы с данными в формате CSV. Пользователь скачивает на свое устройство запрошенные с API данные, чтобы иметь возможность воспользоваться ими в будущем.
5. Обучить модель для анализа. Пользователь использует полученные данные для обучения определенных моделей для дальнейшего анализа.
6. Настроить параметры обучения модели. Пользователь задает параметры для дальнейшего обучения моделей.
   1. Предобработка данных

Необработанные данные из источников, таких как API, часто содержат множество ошибок и неточностей, которые могут привести к неправильным выводам и оценкам. Поэтому необходимо проводить предобработку данных, чтобы гарантировать точность и качество анализа.

После очистки данных необходимо провести предобработку. Предобработка данных – это процесс преобразования необработанных данных в формат, который можно использовать для анализа. Это может включать в себя удаление дубликатов, заполнение пропущенных значений, преобразование категориальных переменных в числовые значения и масштабирование данных.

Одним из инструментов, который может быть использован для предобработки данных, является библиотека Pandas в Python. Она позволяет загружать данные из различных источников и проводить различные операции с данными, включая удаление дубликатов, заполнение пропущенных значений, преобразование типов данных и масштабирование данных.

Важно отметить, что предобработка данных является итеративным процессом. Часто возникает необходимость повторно проводить очистку и предобработку данных в зависимости от конкретной задачи анализа и результатов первоначального анализа.

* 1. Прогнозные модели

Прогнозные модели являются неотъемлемой частью анализа данных. В условиях большого объема данных, необходимых для анализа, применение прогнозных моделей позволяет автоматизировать и ускорить процесс принятия решений на основе данных.

В задаче курсовой работы, модели предназначены для прогнозирования будущих результатов учебной деятельности студентов на основе данных, хранимых в системе Универис. Для того чтобы прогнозы были точными и достоверными, необходимо правильно сформировать модели для анализа.

Правильный выбор атрибутов, получаемых из API и использующихся в моделях, играют важную роль в точности прогнозов. Правильный выбор атрибутов позволяет увеличить качество и достоверность прогнозов, что, в свою очередь, имеет большое значение для прогноза. Далее описаны сформированные модели.

1. Модель прогноза оценки дисциплины за первый семестр. Позволяет сделать предсказание оценки, которую студент получит по определенному предмету в течение первого семестра обучения:

Входные поля для обучения:

* пол;
* дециль егэ1;
* дециль егэ2;
* дециль егэ3;
* город регистрации студента (местный/иногородний);
* форма финансирования обучения студента (бюджет/контракт);
* квантиль доли бюджетников в группе.

Выходное прогнозируемое поле:

* оценка по предмету.

1. Модель прогноза оценки дисциплины за второй семестр, который проходил в первомсеместре. Предназначена для прогнозирования оценки, которую студент получит по определенному предмету во втором семестре обучения, исходя из информации о первом семестре:

Входные поля для обучения:

* пол;
* дециль егэ1;
* дециль егэ2;
* дециль егэ3;
* город регистрации студента (местный/иногородний);
* форма финансирования обучения студента (бюджет/контракт);
* тип контроля предмета;
* квантиль доли бюджетников в группе;
* оценка по предмету за 1 семестр.

Выходное прогнозируемое поле:

* оценка по предмету за 2 семестр.

1. Модель прогноза вероятности отчисления по первому и второму семестру по экзаменам/зачетам представляет возможность определить вероятность, с которой студент может быть отчислен или продолжит обучение после второго семестра.

Входные поля для обучения:

* пол;
* дециль егэ1;
* дециль егэ2;
* дециль егэ3;
* город регистрации студента (местный/иногородний);
* форма финансирования обучения студента (бюджет/контракт);
* квантиль доли бюджетников в группе;
* дециль суммы рейтингов в группе студентов по всем дисциплинам 1 семестра;
* дециль суммы рейтингов в группе студентов по всем дисциплинам 2 семестра;
* дециль медианного рейтинга дисциплин экзаменов/зачетов 1 семестра;
* дециль медианного рейтинга дисциплин экзаменов/зачетов 2 семестра.

Выходное прогнозируемое поле:

* отчислен/учится.

1. Прогноз оценки экзамена/зачета по первому семестру представляет возможность предсказать оценку, которую студент получит по экзаменам или зачетам.

Входные поля для обучения:

* пол;
* дециль егэ1;
* дециль егэ2;
* дециль егэ3;
* город регистрации студента (местный/иногородний);
* форма финансирования обучения студента (бюджет/контракт);
* квантиль доли бюджетников в группе;
* дециль суммы рейтингов в группе студентов по всем дисциплинам 1 семестра.

Выходное прогнозируемое поле:

* дециль медианного рейтинга дисциплин экзаменов/зачетов 1 семестра.

1. Прогноз оценки экзамена/зачета по первому и второму семестру представляет возможность предсказать оценку, которую студент получит по экзаменам или зачетам.

Входные поля для обучения:

* пол;
* дециль егэ1;
* дециль егэ2;
* дециль егэ3;
* город регистрации студента (местный/иногородний);
* форма финансирования обучения студента (бюджет/контракт);
* квантиль доли бюджетников в группе;
* дециль суммы рейтингов в группе студентов по всем дисциплинам 1 семестра;
* дециль суммы рейтингов в группе студентов по всем дисциплинам 2 семестра;
* дециль медианного рейтинга дисциплин экзаменов/зачетов 1 семестра.

Выходное прогнозируемое поле:

* дециль медианного рейтинга дисциплин экзаменов/зачетов 2 семестра.

Таким образом, можно отметить, что выбор правильных атрибутов для прогноза является важным фактором, определяющий точность и эффективность разработанного приложения.

* 1. Модуль прогнозирования

Для решения задачи прогнозирования успеваемости студентов на основе полученных атрибутов и составленных моделей были выбраны два алгоритма машинного обучения: случайный лес и градиентный бустинг.

Алгоритм случайного леса использует ансамбль решающих деревьев для прогнозирования. В случайных лесах деревья в ансамбле строятся из выборки, взятой с заменой (то есть выборкой начальной загрузки) из обучающего набора. Каждое дерево обучается на подмножестве данных и делает прогноз. Затем прогнозы всех деревьев усредняются, чтобы получить окончательный прогноз. Этот метод хорошо работает для больших наборов данных и может обрабатывать большое количество признаков.

Градиентный бустинг также использует ансамбль решающих деревьев. Однако в отличие от случайного леса, градиентный бустинг строит деревья последовательно, каждое следующее дерево исправляет ошибки предыдущего. Этот метод хорошо работает для малых и средних наборов данных и может достигать высокой точности прогнозирования [16].

Выбор между алгоритмами случайного леса и градиентного бустинга зависит от специфики задачи и доступных данных. В обозначенной задаче оба алгоритма могут быть использованы для прогнозирования успеваемости студентов. Они помогут выявить взаимосвязи между различными факторами и их влиянием на успеваемость.

* 1. Графический интерфейс

Рисунок 3 показывает макет окна извлечения данных из API. На макете изображены основные элементы интерфейса.

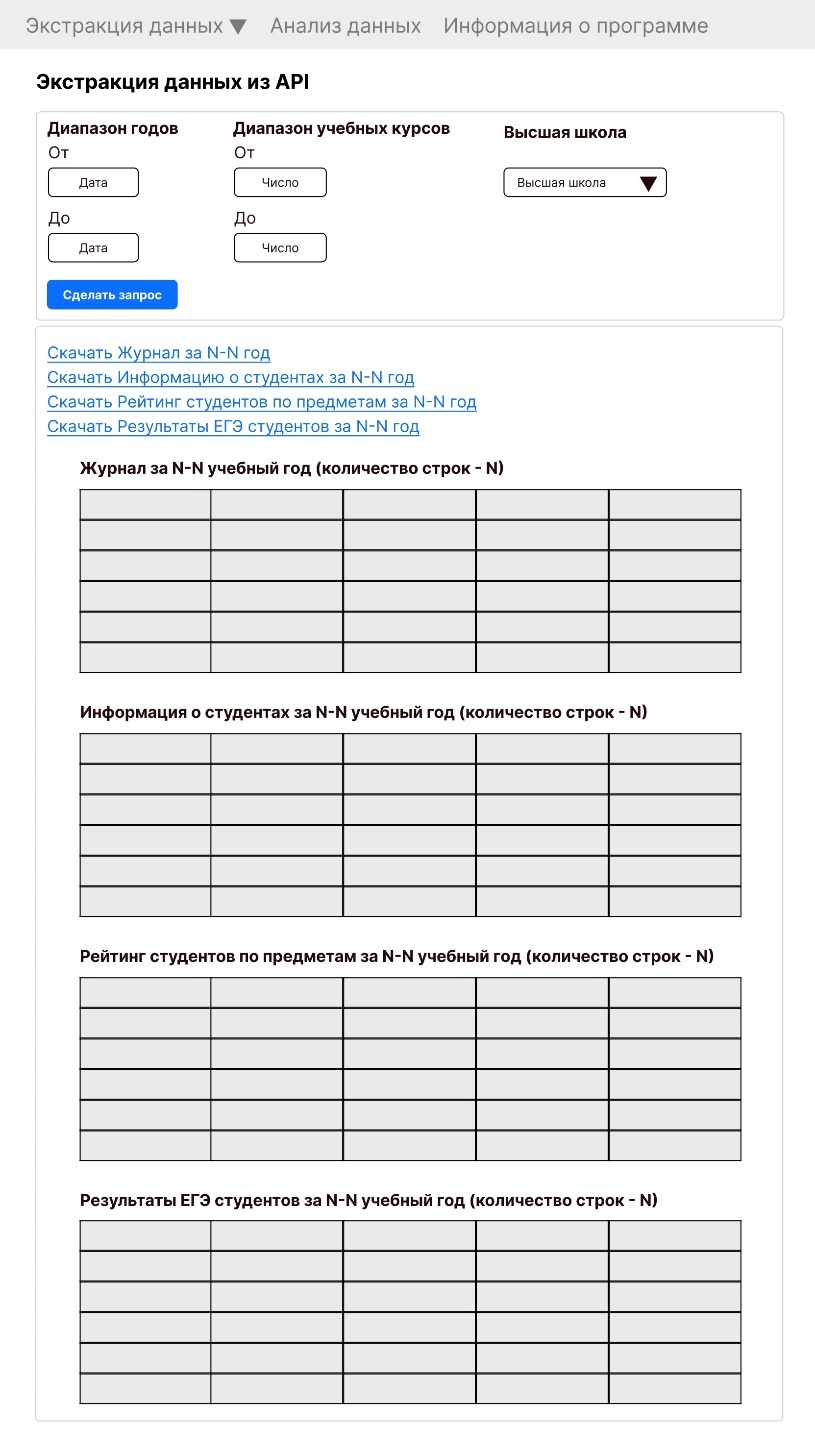


Рисунок – Макет извлечения данных из API

В верхней части находится меню переключения между страницами, далее расположены поля для настройки данных, получаемых из API, рядом с ними расположены кнопки для запроса по указанным настройкам. Ниже расположены ссылки для скачивания запрошенных файлов и таблицы, в которых отображаются данные, полученные после запроса.

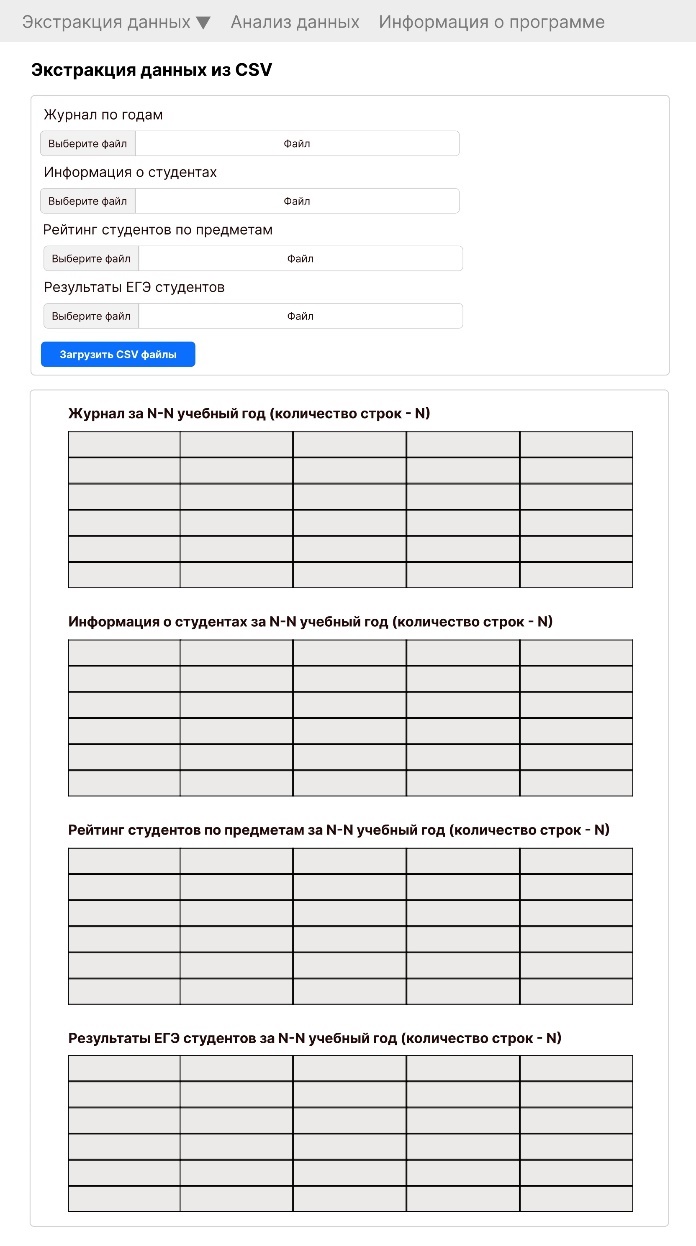


Рисунок – Макет извлечения данных из CSV файлов

Рисунок 4 показывает макет окна извлечения данных из CSV файлов. На странице также находится возможность переключения между окнами. Далее находится форма для загрузки CSV файлов. Ниже расположены таблицы загруженных файлов.

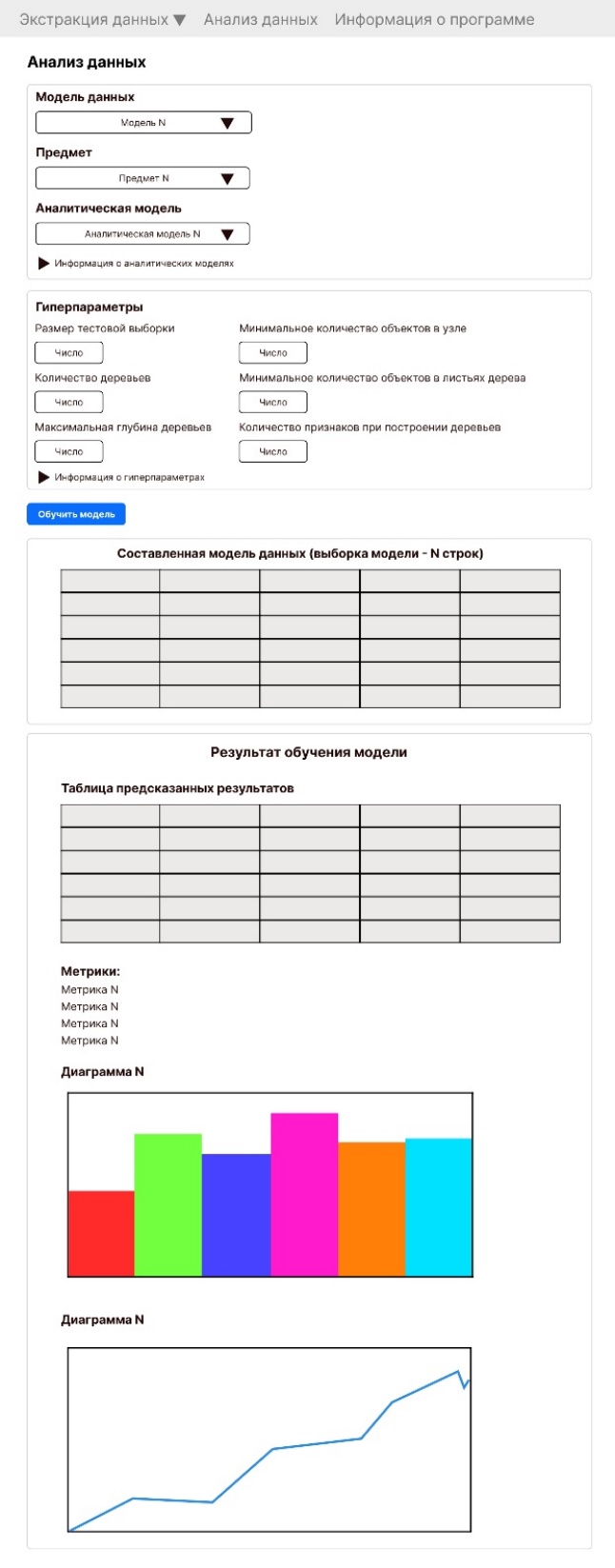


Рисунок 5 – Макет обучения модели

Рисунок 5 иллюстрирует макет окна обучения модели. На окне также расположено меню с переключения между окнами, ниже расположена форма с настройками, по которым будет обучаться модель, после нее таблица с составленной моделью данных. Далее расположены результаты обучения модели с таблицей предсказанных результатов, метриками и графиками.

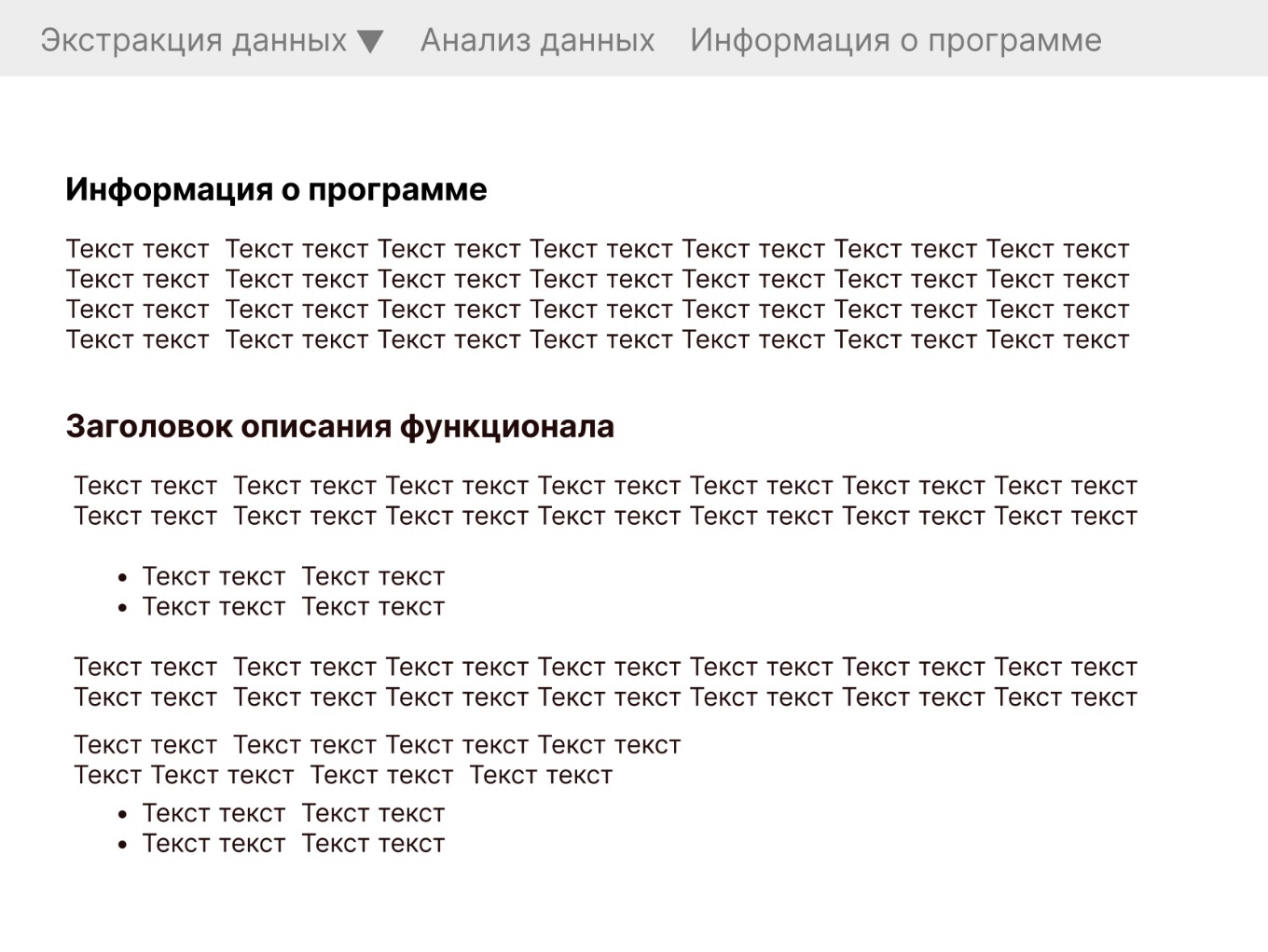


Рисунок – Макет информации о программе

Рисунок 6 показывает информацию о программе. На окне также имеется меню с переключения между окнами, далее располагается информация о программе, ее функционале и том, что пользователю нужно делать по шагам, для анализа данных в программе.

В целом, макеты окон являются интуитивно понятными и предоставляют пользователю удобный интерфейс для работы с данными из API, CSV файлов и обучения модели.

# РЕАЛИЗАЦИЯ

Диаграмма классов (Рисунок 7) позволяет визуализировать основные классы, их атрибуты и методы, а также отношения между ними.

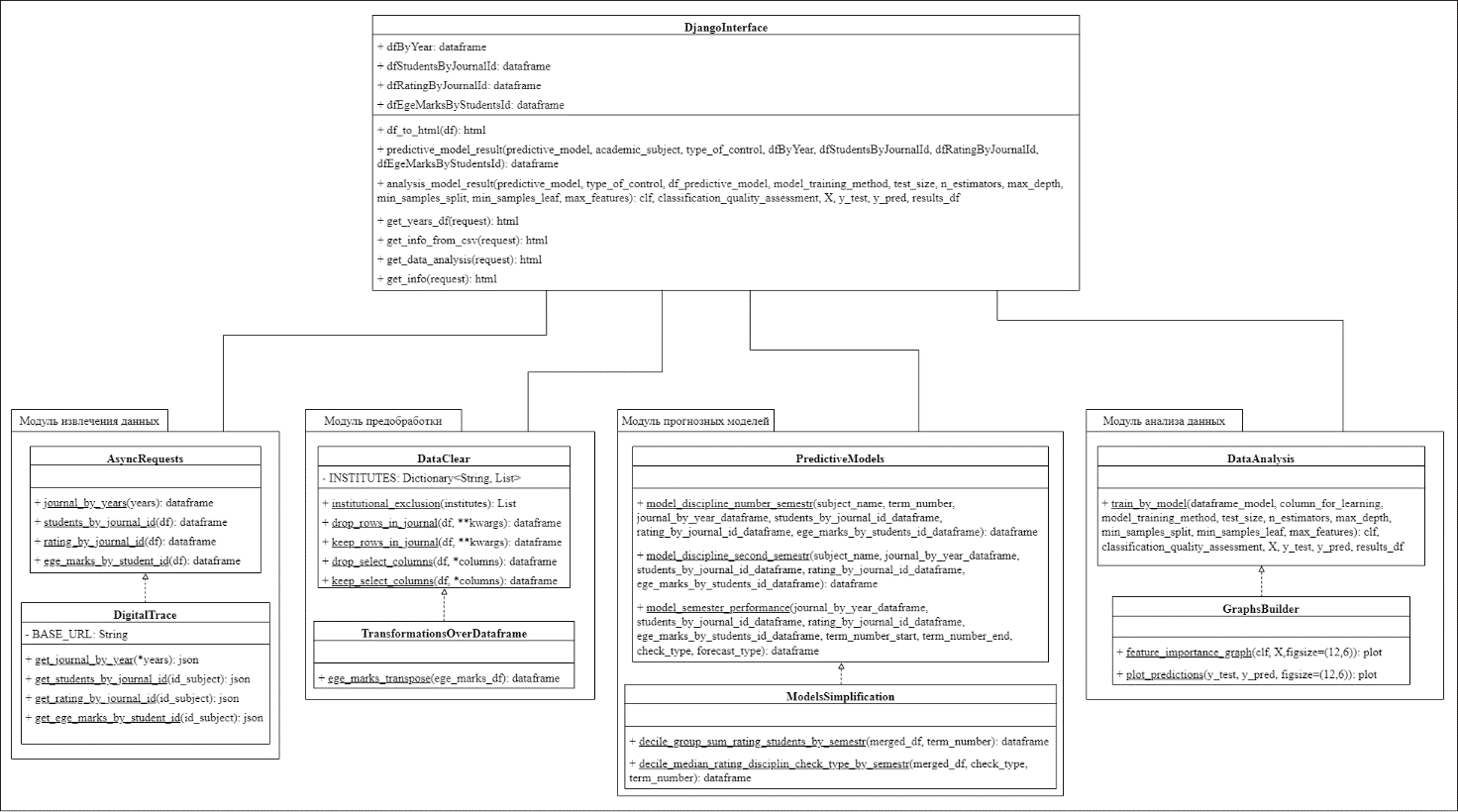


Рисунок – Диаграмма классов

Модуль извлечения данных позволяет получить данные из системы Универис для анализа, которые с помощью модуля предобработки возможно очистить и сохранить необходимые данные. Модуль прогнозных моделей формирует наборы данных для анализа, а модуль анализа данных позволяет обучить по сформированным прогнозным моделям аналитические модели и оценить их точность. Функционал модулей объединен в DjangoInterface, который представляет себя интерфейс взаимодействия с остальными модулями.

Для визуализации высокоуровневого представления, организации компонентов системы и зависимостей между ними, компоненты на диаграмме представляют собой модули разработанных классов, составлена диаграмма компонентов (Рисунок 8).

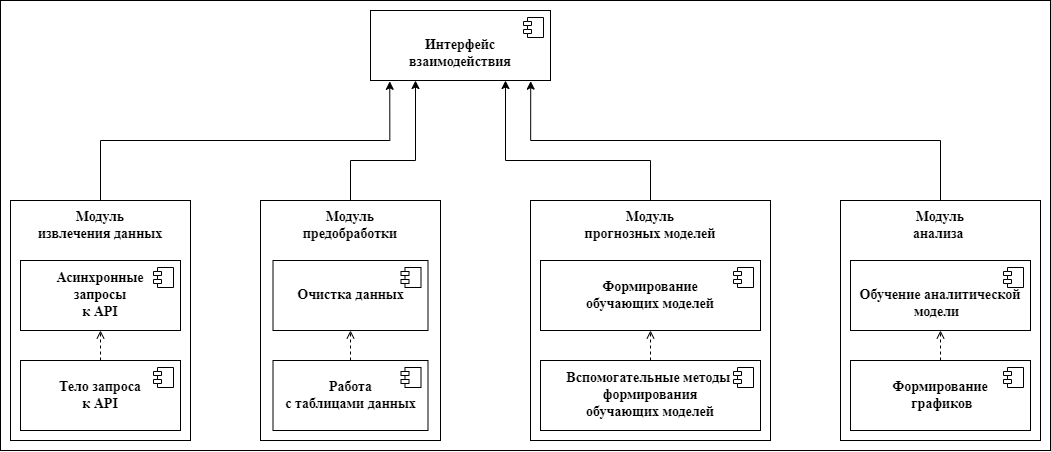


Рисунок – Диаграмма компонентов

Также составлена диаграмма деятельности (Рисунок 9), она позволяет более детально визуализировать случаи использования. Это поведенческая диаграмма, которая иллюстрирует поток деятельности через систему.

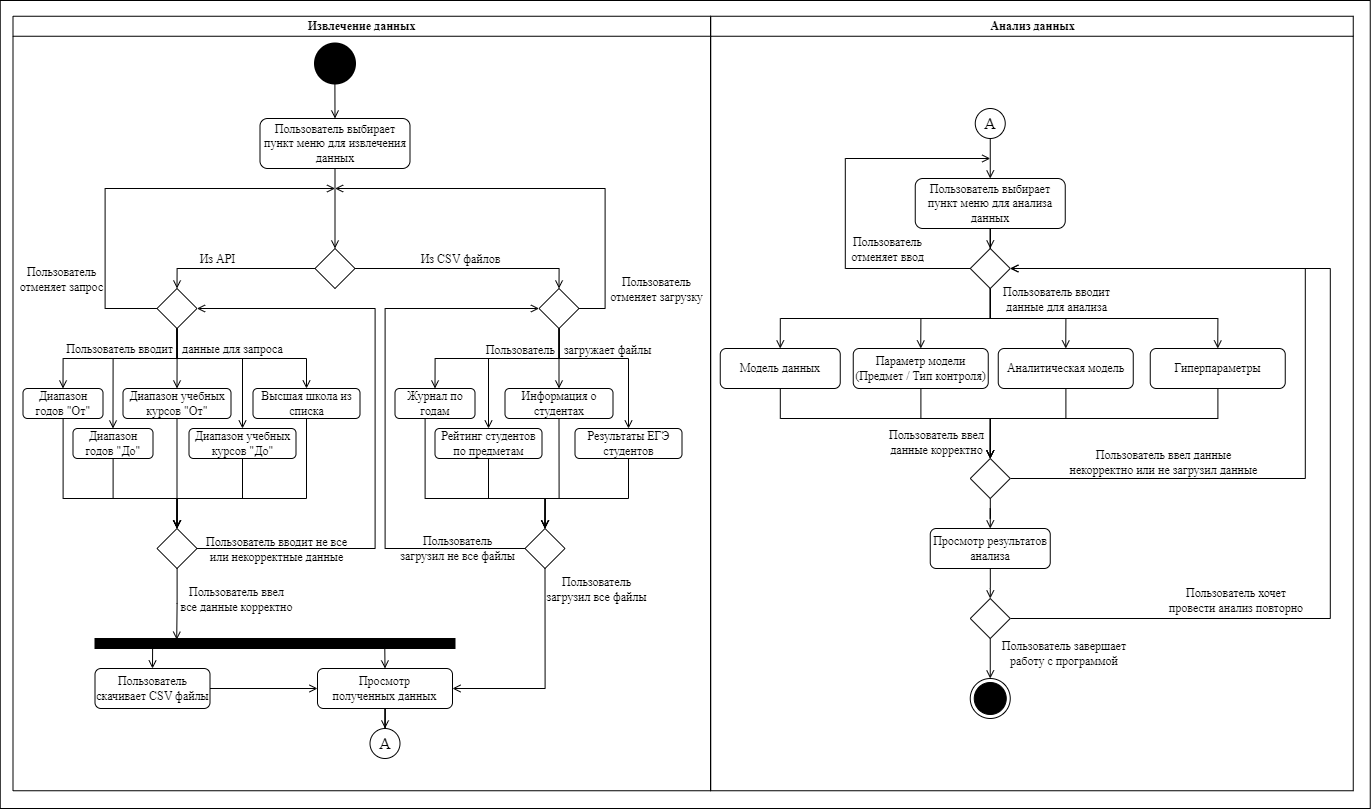


Рисунок – Диаграмма деятельности

* 1. Программные средства реализации

Разработка приложения для прогноза успеваемости студентов была выполнена с использованием языка программирования Python 3 [10] со следующим основным набором библиотек:

1. Pandas – это библиотека Python для анализа и обработки данных. Она предоставляет удобные и мощные инструменты для работы с таблицами, временными рядами и другими форматами данных [9].
2. Numpy – это библиотека Python для научных вычислений. Она предоставляет многомерные массивы и матрицы, а также функции для работы с ними [8].
3. Asyncio – это библиотека Python для асинхронного программирования. Она позволяет создавать асинхронные функции и сопрограммы, которые могут выполняться параллельно без использования потоков [3].
4. Aiohttp – это библиотека Python для создания асинхронных веб-приложений. Она предоставляет клиентские и серверные инструменты для работы с протоколами HTTP и WebSocket [2].
5. Scikit-Learn – это библиотека Python для машинного обучения. Она предоставляет инструменты для обучения моделей машинного обучения, включая классификацию, регрессию, кластеризацию, обработку текстов и другие задачи [14].
6. Matplotlib – это библиотека Python для визуализации данных. Она предоставляет инструменты для создания графиков, диаграмм, карт и других видов визуализации [6].
7. Tenacity – это библиотека Python для обработки ошибок и повторной попытки выполнения операций. Она предоставляет механизмы для обработки ошибок сетевых запросов, баз данных и других операций, которые могут временно недоступны или нестабильны [15].
8. Django – это фреймворк Python для создания веб-приложений. Она предоставляет инструменты для создания и управления базами данных, обработки запросов, авторизации пользователей, взаимодействия с другими веб-сервисами и многое другое [4].
   1. Модуль извлечения данных

Модуль извлечения состоит из классов DigitalTrace и AsyncRequests, дополнительная информация в Приложении А. Класс DigitalTrace формирует основу для запросов к API системы Универис, содержит поле BASE\_URL, которое задает основной путь для GET запросов к API системы Универис. Методы в классе DigitalTrace:

* get\_journal\_by\_year – принимает в аргумент года, по которым необходимо сделать запрос. Возвращает ответ в виде JSON файла с журналами по введенным годам;
* get\_students\_by\_journal\_id – принимает в аргумент id журнала в системе Универис, из ответа полученного в get\_journal\_by\_year. Возвращает JSON файл с информацией о студентах по журналу;
* get\_rating\_by\_journal\_id – принимает в аргумент id журнала в системе Универис, из ответа полученного в get\_journal\_by\_year. Возвращает JSON файл с информацией о рейтинге по журналу;
* get\_ege\_marks\_by\_student\_id – принимает в аргумент id студента в системе Универис, из ответа полученного в get\_students\_by\_journal\_id.

Методы класса AsyncRequests предназначены для осуществления асинхронных запросов, созданных в DigitalTrace.

Все методы, кроме journal\_by\_years, используют механизм семафоров для ограничения количества одновременно выполняемых запросов к API, чтобы избежать перегрузки удаленного сервера. Ограничение на количество задач позволяет эффективно использовать ресурсы и уменьшить нагрузку на сервер.

Также используется декоратор retry, который позволяет повторять выполнение асинхронной функции в случае возникновения ошибки при запросе к API. Это повышает устойчивость кода к сбоям, так как вместо сбоя при первой неудачной попытке выполнения задачи, код будет повторно пытаться выполнить задачу, что увеличит вероятность успешного выполнения операции в конечном итоге.

* 1. Модуль предобработки

Модуль предобработки включает в себя классы DataClear и TransformationsOverDataframe, дополнительная информация в Приложении Б.

Класс DataClear содержит в себе методы, необходимые для того, чтобы убрать лишние значения из выборки, имеет атрибут INSTITUTES, который представляет из себя словарь с ключами в виде названий высших школ ЮУрГУ и их значений в виде списка с кодом специальностей, который будут использованы для фильтрации по определенной высшей школе. Методы класса DataClear:

* institutional\_exclusion – Принимает названия высших школ, который необходимо будет убрать из полученного с помощью get\_journal\_by\_year файла. Возвращает список значений ключей указанных высших школ;
* drop\_rows\_in\_journal и keep\_rows\_in\_journal – Принимают dataframe и словарь именованных аргументов, drop\_rows\_in\_journal удаляет, а keep\_rows\_in\_journal сохраняет строки в dataframe по словарям аргументов;
* drop\_select\_columns и keep\_select\_columns – Принимают dataframe и словарь именованных аргументов, drop\_select\_columns удаляет, а keep\_select\_columns сохраняет столбцы в dataframe по словарям аргументов.

В классе TransformationsOverDataframe содержится метод ege\_marks\_transpose, который принимает метод результат возвращаемый ege\_marks\_by\_student\_id и возвращает dataframe с транспонированными строками результатов ЕГЭ студентов, формируя столбцы вступительных предметов.

* 1. Модуль прогнозных моделей

В модуле прогнозных моделей содержатся классы PredictiveModels и ModelsSimplification, дополнительная информация в Приложении В.

Класс PredictiveModels позволяет формировать прогнозные модели данных по полученной и предобработанной выборке из API, методы в классе:

* model\_discipline\_number\_semestr – принимает название предмета, номер семестра и набор предобработанных dataframe с журналом по году, студентам, их рейтингу по предмету и результатами ЕГЭ. На выходе формирует модель для прогноза оценки по дисциплине за указанный семестр;
* model\_discipline\_second\_semestr – принимает название предмета 2 семестра, который был в 1 семестре и набор предобработанных dataframe с журналом по году, студентам, их рейтингу по предмету и результатами ЕГЭ. На выходе формирует модель прогноза оценки дисциплины за 2 семестр, который был в 1 семестре;
* model\_semester\_performance – принимает набор предобработанных dataframe с журналом по году, студентам, их рейтингу по предмету и результатами ЕГЭ, числа от какого и до какого семестра сохранять выборку, тип контроля предмета в виде зачета или экзамена и параметр отвечающий за то, добавлять ли столбец Status в итоговую таблицу со значениями учится – 1, отчислен – 0. На выходе формирует модель вероятности отчисления по первому и второму семестру по экзаменам/зачетам, прогноз оценки экзамена/зачета по первому семестру или прогноз оценки экзамена/зачета по первому и второму семестру в зависимости от указанных параметров.

Класс ModelsSimplification содержит вспомогательные методы для формирования моделей в PredictiveModels, decile\_group\_sum\_rating\_students\_by\_semestr позволяет вычислить дециль в группе по сумме рейтингов студентов по всем дисциплинам переданного номера семестра и decile\_median\_rating\_disciplin\_check\_type\_by\_semestr, который вычисляет дециль медианного рейтинга дисциплины зачета или экзамена переданного номера семестра.

* 1. Модуль анализа данных

В модуле анализа данных содержатся классы DataAnalysis и GraphsBuilder, дополнительная информация в Приложении Г.

Класс DataAnalysis позволяет обучать определенные аналитические модели по сформированным прогнозным моделям данных. Содержит метод train\_by\_model, который принимает dataframe с сформированной моделью данных, столбцом, значения которого будут предсказываться, типом модели для обучения, размером выборки, которая будет использоваться для тестирования обученной модели и набором гиперпараметров значения которых используются для управления процессом обучения. На выходе возвращает обученную модель, оценки ее точности, данные для построения графиков и таблицу с предсказанными результатами.

Класс GraphsBuilder предназначен для построения графиков обученной модели, содержит методы:

* feature\_importance\_graph – на входе принимает обученную аналитическую модель, столбцы входных данных и размер графика, а на выходе формирует график важности признаков отображая гистограмму входных данных оказывающих наибольшее влияние на результат обучения;
* plot\_predictions – на входе принимает данные прогнозируемого столбца, предсказанные значения по обученной модели и размер графика, на выходе формирует график показывающий совпадение истинных и предсказанных значений обученной моделью на гистограмме.
  1. Реализация интерфейса

Интерфейс представляет собой веб-приложение, реализованное на основе фреймворка Django. Дополнительно для формирование визуальной составляющей используется HTML, CSS, JavaScript и Bootstrap. Дополнительная информация в Приложении Д.

Одной из ключевых страниц интерфейса является страница извлечения данных из API. На данной странице пользователю предоставляется возможность получить данные путем запроса к API системы Универис по заданным параметрам. С использованием разработанных модулей, данные получаются, затем осуществляется их очистка и обработка, далее пользователь имеет возможность просмотреть запрошенные данные и скачать их в формате CSV файлов. Этот функционал обеспечивает пользователю удобство при получении и сохранении данных из API. Рисунок 10 отображает выполненный интерфейс извлечения данных из API.

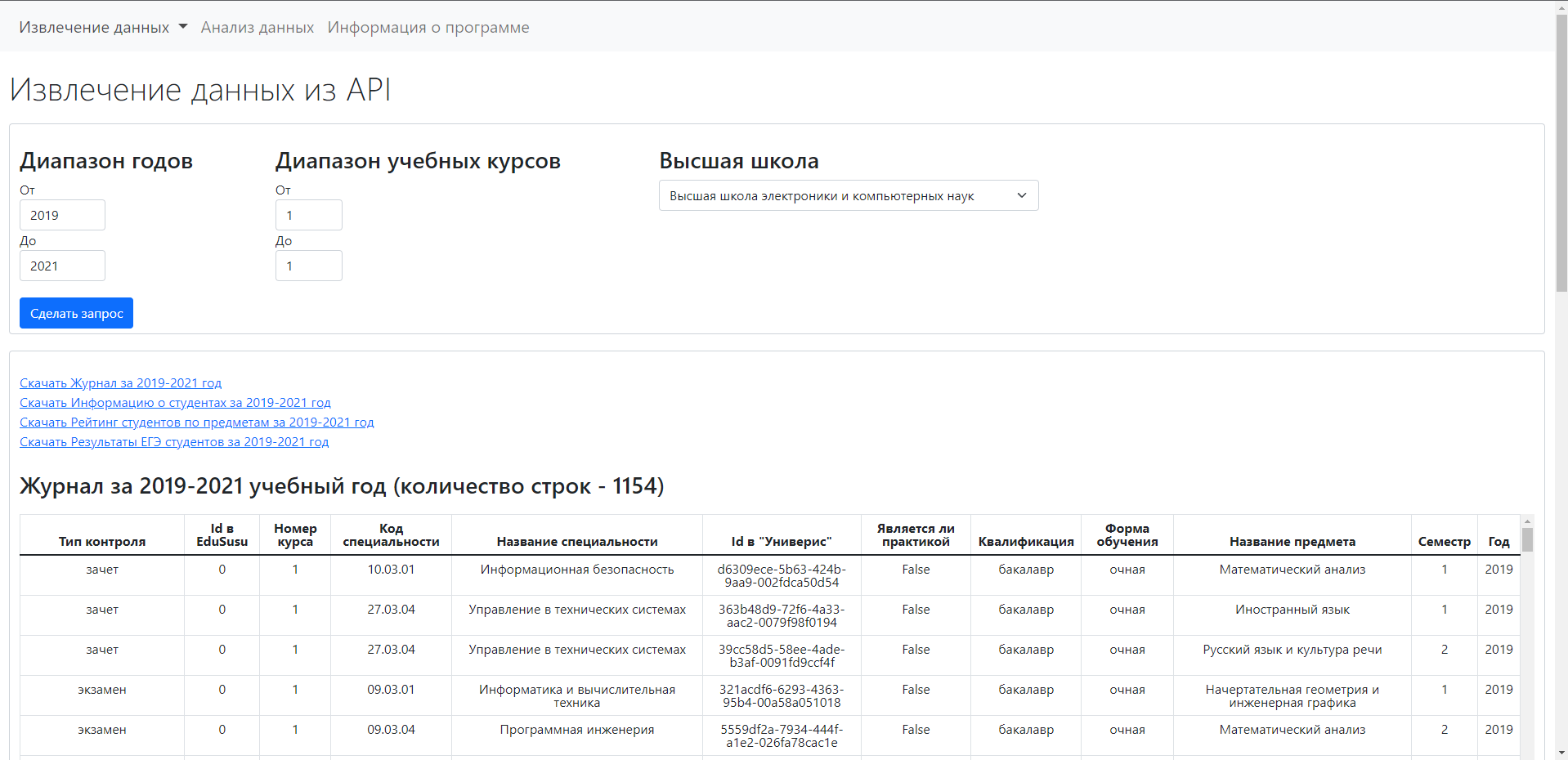


Рисунок – Интерфейс извлечения данных из API

Кроме того, была разработана страница извлечения данных из CSV файлов, которая предоставляет пользователю возможность загрузить предварительно сохраненные файлы в приложении. Рисунок 11 демонстрирует разработанный интерфейс извлечения из CSV файлов.

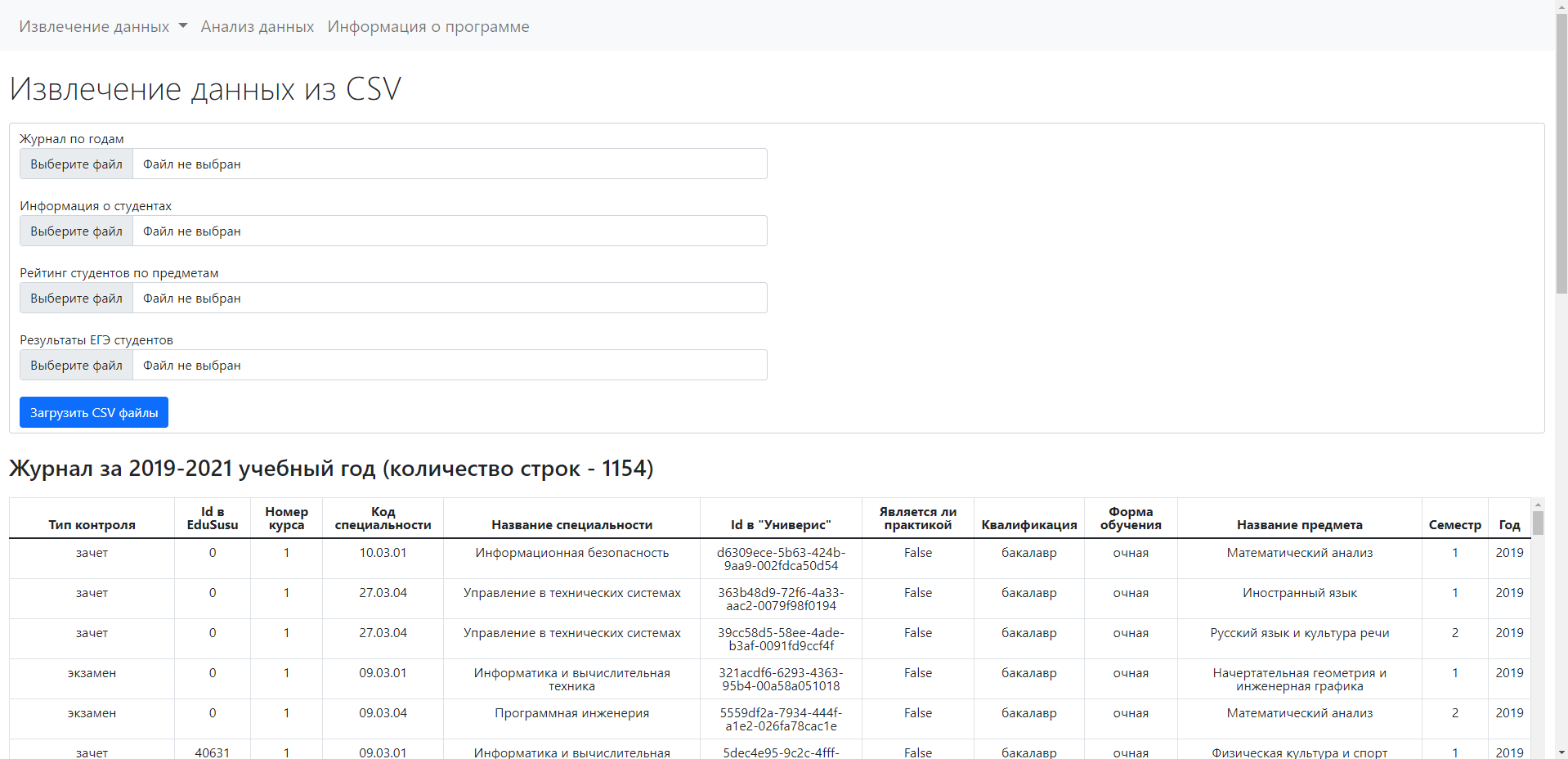


Рисунок – Интерфейс извлечения данных из CSV файлов

Разработанные модули позволяют осуществить извлечение и обработку данных из загруженных файлов, что позволяет пользователям использовать повторно ранее запрошенные данные.

Также в интерфейсе была реализована страница для анализа данных. Пользователям предоставляется возможность производить задавать модель данных, по которой будет сформирована выборка из полученных и обработанных данных, аналитическую модель, которая будет производить анализ по задаваемым пользователем гиперпараметрам. После обучения аналитической модели, пользователь имеет возможность увидеть таблицу результата обучения модели, а именно таблицу предсказанных результатов, метрики и диаграммы. Рисунок 12 и Рисунок 13 отображаю описанную страницу.

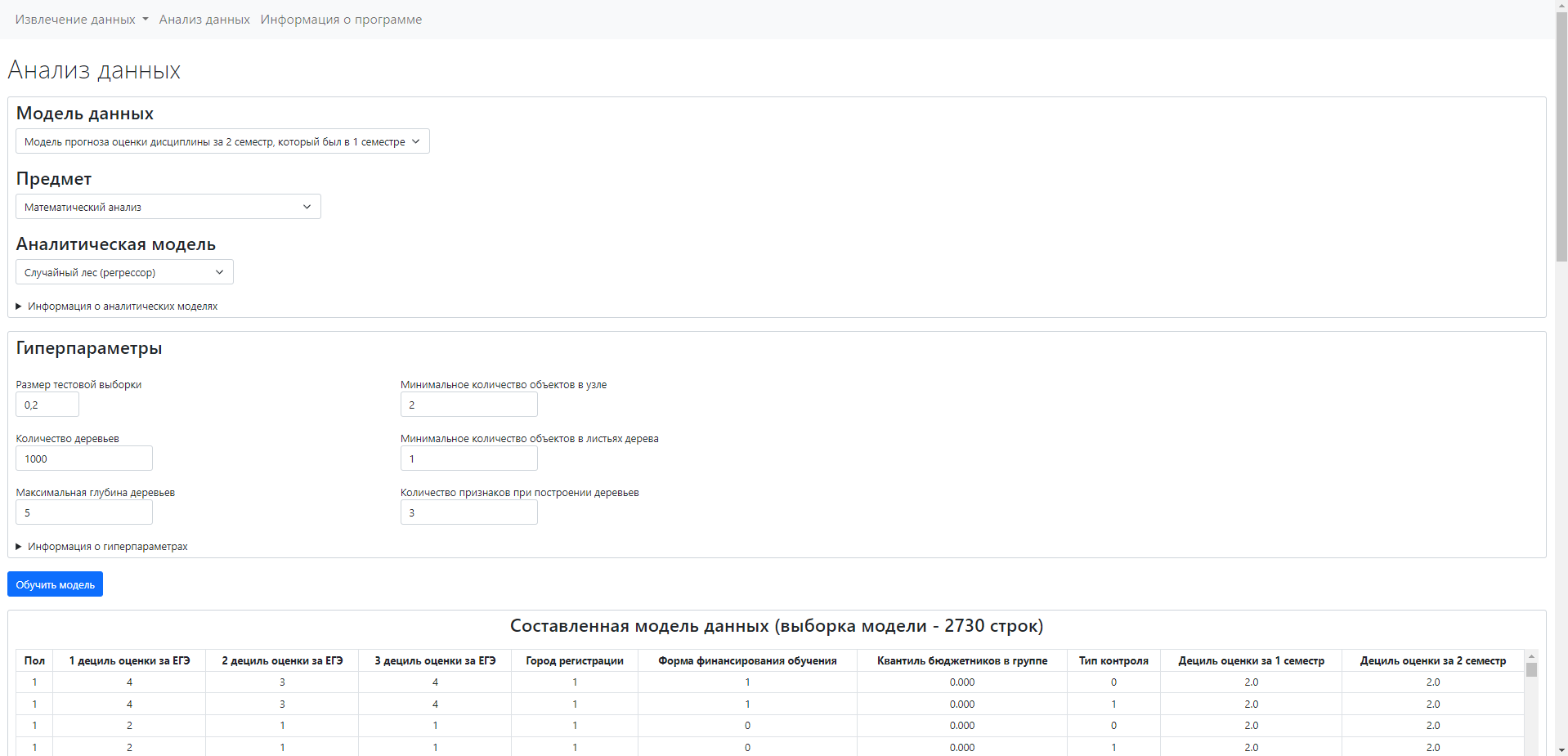


Рисунок – Интерфейс анализа данных

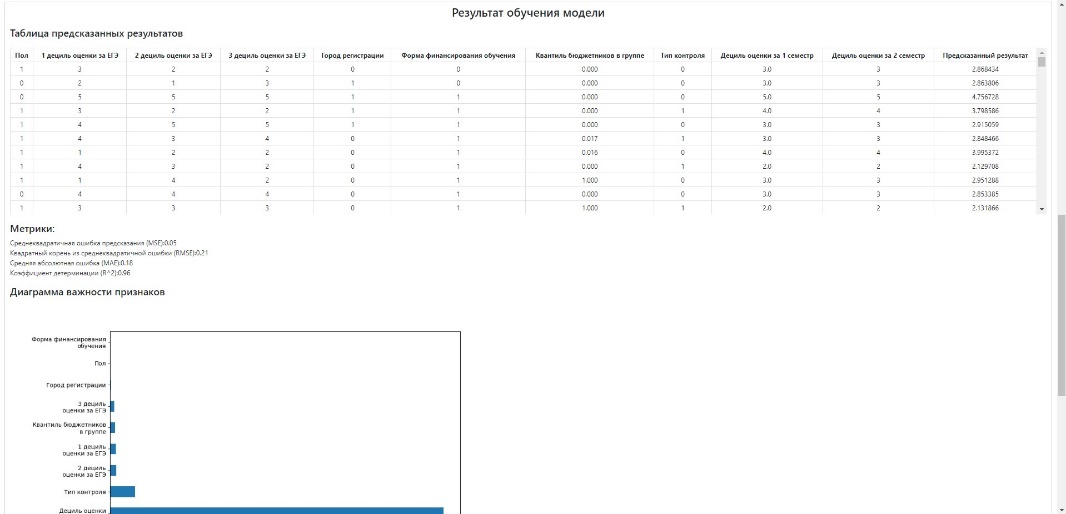


Рисунок – Интерфейс анализа данных, результаты

Дополнительно, в интерфейсе была включена страница с информацией о программе, где предоставляется пользователю подробное описание функциональности приложения и инструкцию по использованию. Рисунок 14 показывает созданную страницу с информацией.

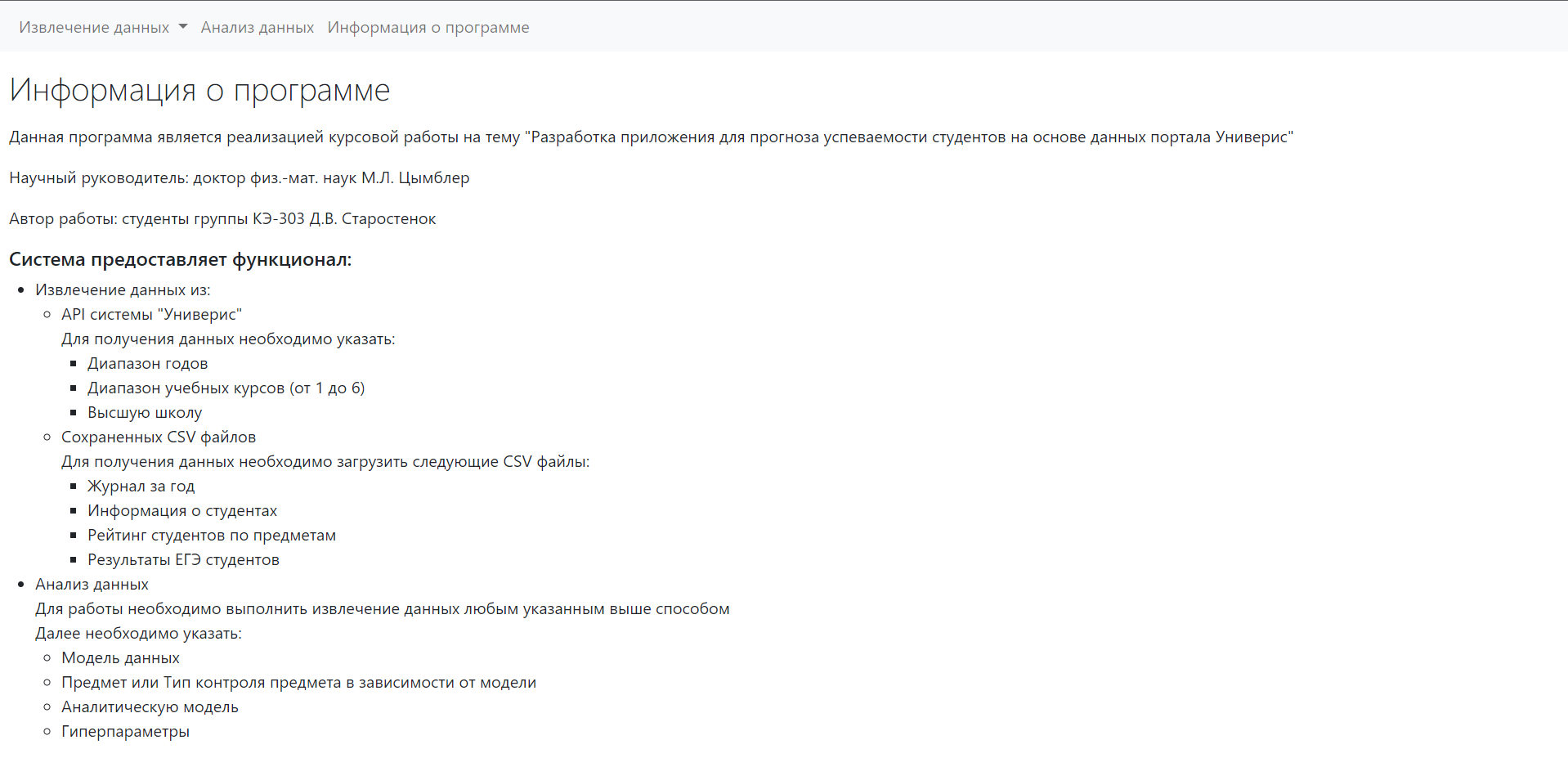


Рисунок – Интерфейс информации о программе

В целом, реализация программы включает в себя различные модули и классы, каждый из которых выполняет определенные функции, связанные с извлечением, предобработкой, прогнозированием и анализом данных. Эти модули и классы взаимодействуют между собой, обеспечивая функциональность и эффективность программы.

# ТЕСТИРОВАНИЕ

* 1. Тестирование извлечения данных из API и CSV

Проверка работы извлечения данных из API системы Универис заключается в проведение функционального тестирования (Таблица 5).

Таблица – Тестирование извлечения данных из API и CSV

| **№** | **Название теста** | **Шаги** | **Ожидаемый результат** | **Тест пройден?** |
| --- | --- | --- | --- | --- |
|  | Получение из API данных Высшей школы электроники и компьютерных наук за 2019–2021 год | 1. На странице извлечения данных из API указать диапазон годов от 2019 до 2021; 2. Диапазон учебных курсов указать от 1 до 1; 3. Высшую школу указать “Высшая школа электроники и компьютерных наук”; 4. Нажать кнопку “Сделать запрос”. | После завершения всех запрос таблицы полученных данных корректно отображаются, имеется возможность скачать их в формате CSV. | Да |
|  | Получение из API данных Политехнического института за 2019–2019 год | 1. На странице извлечения данных из API указать диапазон годов от 2019 до 2019; 2. Диапазон учебных курсов указать от 1 до 1; 3. Высшую школу указать “Политехнический институт”; 4. Нажать кнопку “Сделать запрос”. | После завершения всех запрос таблицы полученных данных корректно отображаются, имеется возможность скачать их в формате CSV. | Да |

Продолжение таблицы 5

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **№** | **Название теста** | **Шаги** | **Ожидаемый результат** | **Тест пройден?** |
|  | Скачивание полученных из API данных Политехнического института за 2019–2019 год | 1. На странице извлечения данных из API указать диапазон годов от 2019 до 2019. 2. Диапазон учебных курсов указать от 1 до 1; 3. Высшую школу указать; “Политехнический институт”; 4. Нажать кнопку “Сделать запрос”. 5. Нажать на гиперссылки:  * Скачать Журнал за 2019-2019 год; * Скачать Информацию о студентах за 2019-2019 год; * Скачать Рейтинг студентов по предметам за 2019-2019 год; * Скачать Результаты ЕГЭ студентов за 2019-2019 год. | Таблицы скачиваются в формате CSV. | Да |
|  | Проверка валидации получения данных из API | 1. На странице извлечения данных из API указать диапазон годов от 2021 до 2019. 2. Нажать кнопку “Сделать запрос”. | Система дает предупреждение о том, что год «От» должен быть меньше или равен году «До» | Да |

Окончание таблицы 5

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **№** | **Название теста** | **Шаги** | **Ожидаемый результат** | **Тест пройден?** |
|  | Получение данных из CSV файлов Высшей школы электроники и компьютерных наук за 2019–2021 год | 1. На странице извлечения данных из CSV добавить файлы по пунктам:  * Журнал по годам; * Информация о студентах; * Рейтинг студентов по предметам; * Результаты ЕГЭ студентов.  1. Нажать кнопку “Загрузить CSV файлы” | Добавленные CSV файлы загружаются в программу, данные возможно использовать | Да |
|  | Проверка валидации получения данных из CSV файлов | 1. На странице извлечения данных из CSV добавить файлы по пунктам:  * Журнал по годам; * Информация о студентах.  1. Нажать кнопку “Загрузить CSV файлы” | Система дает предупреждение, что необходимо загрузить остальные файлы | Да |

* 1. Тестирование анализа данных

Для достоверной и объективной оценки качества анализа классификационных моделей необходимо применять специализированные метрики.

Одной из распространенных метрик является точность (accuracy), которая измеряет долю правильно классифицированных наблюдений по отношению к общему числу наблюдений. Однако, точность может быть недостаточной для полного исследования модели, так как она не учитывает возможные неравенства важности различных классов и может оказаться искаженной в случае несбалансированных данных.

Для более полной и надежной оценки качества классификации используются также следующие метрики:

1. Полнота (recall), которая измеряет способность модели обнаруживать положительные классы из общего числа истинных положительных классов. Полнота является особенно важной метрикой в задачах, где ложноотрицательные прогнозы имеют серьезные последствия.
2. Точность (precision), которая измеряет способность модели предсказывать правильно положительные классы из общего числа положительных прогнозов. Точность позволяет оценить, насколько надежными являются положительные прогнозы модели.
3. F-мера (F1-score), которая является гармоническим средним между точностью и полнотой. F-мера позволяет совместно учитывать точность и полноту модели и является полезной метрикой в случаях, когда важны и точность, и полнота прогнозов.

В отличие от задач классификации, где прогнозируется категориальная переменная, в задачах регрессии прогнозируется непрерывная переменная. Для оценки точности прогнозов в регрессии применяются специфические метрики.

Одной из наиболее распространенных метрик в регрессии является средняя абсолютная ошибка (MAE), которая измеряет среднее абсолютное отклонение между прогнозами модели и фактическими значениями целевой переменной. MAE позволяет оценить среднюю разницу между прогнозами и реальными значениями без учета их направления.

Кроме того, для оценки точности прогнозирования в регрессии также используются следующие метрики:

1. Среднеквадратичная ошибка (MSE), которая измеряет среднюю квадратичную разницу между прогнозами модели и фактическими значениями. MSE чувствительна к большим отклонениям и является широко применяемой метрикой в задачах, где важно минимизировать большие ошибки.
2. Квадратный корень из среднеквадратичной ошибки (RMSE), который является более чувствительной метрикой к большим отклонениям. RMSE измеряет среднеквадратичное отклонение между прогнозами модели и фактическими значениями целевой переменной. Поскольку RMSE включает в себя квадратичное значение ошибки, он дополнительно подчеркивает важность точности прогнозов и стремится минимизировать большие отклонения.
3. Коэффициент детерминации (R2), который позволяет оценить долю вариации целевой переменной, объясненную моделью. R2 принимает значения от 0 до 1, где 1 означает идеальное соответствие модели данным.

После подбора параметров и обучения было проведено тестирование качества составленных моделей данных. Тестирование заключается в прогнозировании на реальных данных, полученных путем запроса к API системы Универис.

Таблица 6 представляет параметры и итоговые метрики модели данных прогноза оценки дисциплины за первый семестр.

Таблица – Модель прогноза оценки дисциплины за первый семестр

|  |  |  |  |
| --- | --- | --- | --- |
| Высшая школа | Политехнический институт | | |
| Диапазон годов | 2019-2021 | | |
| Предмет | Физическая культура и спорт | | |
| Аналитическая модель | Случайный лес (классификатор) | | |
| Гиперпараметры | | | |
| Размер тестовой выборки | 0,2 | Минимальное количество объектов в узле | 11 |
| Количество деревьев решений | 200 | Минимальное количество объектов в листьях дерева решений | 4 |
| Максимальная глубина деревьев решений | 10 | Количество признаков при построении деревьев решений | 2 |
| Метрики | | | |
| Accuracy | Precision | Recall | F1 |
| 0.72 | 0.73 | 0.73 | 0.73 |

Таблица 7 представляет параметры и итоговые метрики модели прогноза оценки дисциплины за второй семестр, который был в первом семестре.

Таблица – Модель прогноза оценки дисциплины за второй семестр, который был в первом семестре.

|  |  |  |  |
| --- | --- | --- | --- |
| Высшая школа | Высшая школа электроники и компьютерных наук | | |
| Диапазон годов | 2019-2021 | | |
| Предмет | Русский язык и культура речи | | |
| Аналитическая модель | Градиентный бустинг (Регрессор) | | |
| Гиперпараметры | | | |
| Размер тестовой выборки | 0,2 | Минимальное количество объектов в узле | 4 |
| Количество деревьев решений | 200 | Минимальное количество объектов в листьях дерева решений | 4 |
| Максимальная глубина деревьев решений | 10 | Количество признаков при построении деревьев решений | 4 |
| Метрики | | | |
| MSE | RMSE | MAE | R2 |
| 0.05 | 0.22 | 0.05 | 0.96 |

Таблица 8 представляет параметры и итоговые метрики модели вероятности отчисления по первому и второму семестру по экзаменам или зачетам.

Таблица – Модель вероятности отчисления по первому и второму семестру по экзаменам или зачетам.

|  |  |
| --- | --- |
| Высшая школа | Политехнический институт |
| Диапазон годов | 2019-2021 |
| Тип контроля | Зачет |
| Аналитическая модель | Градиентный бустинг (классификатор) |

Окончание таблицы 8

|  |  |  |  |
| --- | --- | --- | --- |
| Гиперпараметры | | | |
| Размер тестовой выборки | 0,2 | Минимальное количество объектов в узле | 3 |
| Количество деревьев решений | 1000 | Минимальное количество объектов в листьях дерева решений | 3 |
| Максимальная глубина деревьев решений | 10 | Количество признаков при построении деревьев решений | 5 |
| Метрики | | | |
| Accuracy | Precision | Recall | F1 |
| 0.91 | 0.90 | 0.90 | 0.90 |

Таблица 9 представляет параметры и итоговые метрики модели прогноза оценки по экзамену или зачету по первому семестру.

Таблица – Модель прогноза оценки по экзамену или зачету по первому семестру

|  |  |  |  |
| --- | --- | --- | --- |
| Высшая школа | Высшая школа электроники и компьютерных наук | | |
| Диапазон годов | 2019-2021 | | |
| Тип контроля | Экзамен | | |
| Аналитическая модель | Случайный лес (регрессор) | | |
| Гиперпараметры | | | |
| Размер тестовой выборки | 0,2 | Минимальное количество объектов в узле | 2 |
| Количество деревьев решений | 400 | Минимальное количество объектов в листьях дерева решений | 4 |
| Максимальная глубина деревьев решений | 10 | Количество признаков при построении деревьев решений | 5 |
| Метрики | | | |
| MSE | RMSE | MAE | R2 |
| 0.06 | 0.25 | 0.06 | 0.96 |

Таблица 10 представляет параметры и итоговые метрики модели прогноза оценки по экзамену или зачету по первому и второму семестру.

Таблица – Модель прогноза оценки по экзамену или зачету по первому и второму семестру

|  |  |  |  |
| --- | --- | --- | --- |
| Высшая школа | Высшая школа электроники и компьютерных наук | | |
| Диапазон годов | 2019-2021 | | |
| Тип контроля | Зачет | | |
| Аналитическая модель | Случайный лес (классификатор) | | |
| Гиперпараметры | | | |
| Размер тестовой выборки | 0,2 | Минимальное количество объектов в узле | 2 |
| Количество деревьев решений | 200 | Минимальное количество объектов в листьях дерева решений | 1 |
| Максимальная глубина деревьев решений | 10 | Количество признаков при построении деревьев решений | 3 |
| Метрики | | | |
| Accuracy | Precision | Recall | F1 |
| 0.96 | 0.96 | 0.96 | 0.96 |

Проведенные тесты показали корректную работу программы и адекватные результаты точности прогнозов.

ЗАКЛЮЧЕНИЕ

В рамках данной работы было разработано приложение для прогнозирования успеваемости студентов на основе данных портала Универис, оно позволяет предсказывать успех студентов на основе их текущих оценок и статистики учебной деятельности и предоставляет инструмент для повышения эффективности обучения студентов и преподавателей. При этой решены следующие цели и задачи:

1. Проведен анализ предметной области и на его основе разработаны прогнозные модели;
2. Разработан алгоритм получения и предобработки данных;
3. Разработано приложение, выполняющее прогноз на основе моделей с помощью различных методов анализа данных;
4. Проведены эксперименты, исследующие разработанное приложение.

Разработанное приложение имеет практическое применение и может быть полезно как студентам, так и преподавателям, позволяя им получать прогнозы успеваемости и использовать их для улучшения учебного процесса. Кроме того, администрация университета может воспользоваться приложением для отслеживания прогресса студентов и разработки программ для улучшения образовательной среды. В будущем планируется добавить возможность использовать обученную модель, больше прогнозных моделей, организовать возможность получения и объединения большего количества данных, а также добавить новые аналитические модели, такие как нейронные сети.

ЛИТЕРАТУРА

1. Ahajjam T., Moutaib M., Aissa H., Azrour M., Farhaoui Y., Fattah M. Predicting Students' Final Performance Using Artificial Neural Networks // Big Data Mining and Analytics. 2022. Vol. 5(4), p. 294-301
2. Aiohttp. [Электронный ресурс] URL: https://docs.aiohttp.org/en/stable/ (дата обращения: 02.05.2022 г.).
3. Asyncio. [Электронный ресурс] URL: https://docs.python.org/dev/library/asyncio.html (дата обращения: 02.05.2022 г.).
4. Django. [Электронный ресурс] URL: https://www.djangoproject.com/ (дата обращения: 02.05.2022 г.).
5. Jeganathan S., Lakshminarayanan A. R., Ramachandran N., Tunze G. B. Predicting Academic Performance of Immigrant Students Using XGBoost // International Journal of Information Technology and Web Engineering (IJITWE). 2022. Vol. 17(1). p. 1-19.
6. Matplotlib. [Электронный ресурс] URL: https://matplotlib.org/ (дата обращения: 02.05.2022 г.).
7. Nachouki M., Abou Naaj M. Predicting Student Performance to Improve Academic Advising Using the Random Forest Algorithm // International Journal of Distance Education Technologies (IJDET). 2022. Vol. 20(1). p. 1-17.
8. Numpy. [Электронный ресурс] URL: https://numpy.org/ (дата обращения: 02.05.2022 г.).
9. Pandas. [Электронный ресурс] URL: https://pandas.pydata.org/ (дата обращения: 02.05.2022 г.).
10. Python. [Электронный ресурс] URL: https://www.python.org (дата обращения: 02.05.2022 г.).
11. Sahlaoui H., Alaoui E.A.A., Nayyar A., Agoujil S., Jaber M.M. Predicting and Interpreting Student Performance Using Ensemble Models and Shapley Additive Explanations // IEEE Access. 2021. Vol. 9, p. 1-16.
12. Sathya Durga V., Thangakumar J. Predicting Academic Performance of Deaf Students Using Feed Forward Neural Network and An Improved PSO Algorithm // Special Issue on Artificial Intelligence in Cloud Computing. 2021. Vol. 18. p. 112-126.
13. Silvia B. Gonzalez-Brambila, Lourdes Sanchez-Guerrero, Irma Ardon-Pulido, Josue Figueroa-Gonzalez, Beatriz A. Gonzalez-Beltran. Predicting Academic Performance of Engineering Students after Approving a Mathematics Leveling Course using Decision Trees // Research in Computing Science. 2018. Vol. 147, p. 171-181.
14. Sklearn. [Электронный ресурс] URL: https://scikit-learn.org/stable/ (дата обращения: 02.05.2022 г.).
15. Tenacity. [Электронный ресурс] URL: https://tenacity.readthedocs.io/en/latest/ (дата обращения: 02.05.2022 г.).
16. Ансамблевые методы. [Электронный ресурс] URL: https://scikit-learn.ru/1-11-ensemble-methods/# (дата обращения: 02.05.2022 г.).

ПРИЛОЖЕНИЯ

Приложение А. Модуль извлечения данных

Листинг 1 – Документация класса DigitalTrace

async **get\_ege\_marks\_by\_student\_id**(id\_subject)

Оценки ЕГЭ студентов по id студентов  
   
Входные данные:  
    id\_subject (string): id из журнала по студентам  
   
Выходные данные:  
    json: результат запроса

async **get\_grades\_by\_journal\_id**(id\_subject)

Задания предметов студетов по id журнала  
   
Входные данные:  
    id\_subject (string): id из журнала по годам  
   
Выходные данные:  
    json: результат запроса

async **get\_journal\_by\_year**(year)

Получить журнал по году  
   
Входные данные:  
    year (int): год, по которому нужно получить журнал  
   
Выходные данные:  
    json: результат запроса

async **get\_rating\_by\_journal\_id**(id\_subject)

Оценки за предметы по id журнала  
   
Входные данные:  
    id\_subject (string): id из журнала по годам  
   
Выходные данные:  
    json: результат запроса

async **get\_students\_by\_journal\_id**(id\_subject)

Получить студентов по id из журнала по годам  
   
Входные данные:  
    id\_subject (string): id из журнала по годам  
   
Выходные данные:  
    json: результат запроса

async **get\_teachers\_by\_journal\_id**(id\_subject)

Получить преподавателей по id из журнала по годам  
   
Входные данные:  
    id\_subject (string): id из журнала по годам  
   
Выходные данные:  
    json: результат запроса

Листинг 2 – Документация класса AsyncRequests

async **ege\_marks\_by\_student\_id**(df)

Асинхронный запрос результатов ЕГЭ студентов по id в   
students\_by\_journal\_id  
   
Входные данные:  
    df (dataframe): dataframe с журналом за год полученный из   
students\_by\_journal\_id  
Выходные данные:  
    concat\_result: dataframe с результатом ЕГЭ студентов по id в students\_by\_journal\_id

async **journal\_by\_years**(years)

Асинхронный запрос журналов по годам  
   
Входные данные:  
    years (List): список годов для запроса  
Выходные данные:  
    results: [0] - объединенный dataframe по списку указанных   
годов;   
             [1] - [N] - dataframe с годами по отдельности;

async **rating\_by\_journal\_id**(df)

Асинхронный запрос рейтинга за предметы по id в журналах  
   
Входные данные:  
    df (dataframe): dataframe с журналом за год полученный из   
journal\_by\_years  
Выходные данные:  
    concat\_result: dataframe с рейтингом за предметы по id в   
журналах

async **students\_by\_journal\_id**(df)

Асинхронный запрос студентов по id в журналах  
   
Входные данные:  
    df (dataframe): dataframe с журналом за год полученный из   
journal\_by\_years  
Выходные данные:  
    concat\_result: dataframe с рейтингом за предметы по id в   
журналах

Приложение Б. Модуль предобработки

Листинг 3 – Документация класса DataClear

**drop\_rows\_in\_journal**(df, \*\*kwВходные данные)

Удаление строк в журнале dataframe по словарям  
   
Входные данные:  
    df (dataframe): dataframe  
    \*\*kwВходные данные: название поля из журнала = [значения для удаления]  
Выходные данные:  
    df: dataframe с удаленными строками

**drop\_select\_columns**(df, \*columns)

Удаляет указанные столбцы из DataFrame  
   
Входные данные:  
    df (dataframe): dataframe  
    \*columns: колонки для удаления  
Выходные данные:  
    dataframe: dataframe с удаленными колонками

**institutional\_exclusion**(institutes)

Список для исключения высших школ из выборки  
   
Входные данные:  
    institutes (List): высшие школы, которые будут исключены  
Выходные данные:  
    flat\_excluded\_institutes: список высших школ

**keep\_rows\_in\_journal**(df, \*\*kwВходные данные)

Сохранение выбранных строк в журнале dataframe по словарям  
   
Входные данные:  
    df (dataframe): dataframe  
    \*\*kwВходные данные: название поля из журнала=[значения,   
которые будут сохранены]  
Выходные данные:  
    df: dataframe с сохраненными строками

**keep\_select\_columns**(df, \*columns)

Сохраняет указанные столбцы из DataFrame  
   
Входные данные:  
    df (dataframe): dataframe  
    \*columns: колонки, которые будут сохранены  
Выходные данные:  
    dataframe: dataframe с сохраненными колонками

Листинг 4 – Документация класса TransformationsOverDataframe

**ege\_marks\_transpose**(ege\_marks\_df)

Преобразование строк предметов в столбцы для GetEgeMarksByStudent  
   
Входные данные:  
    ege\_marks\_df (dataframe): dataframe c результатами ЕГЭ  
Выходные данные:  
    ege\_marks\_df: dataframe c результатами ЕГЭ разбитыми на   
столбцы

Приложение В. Модуль прогнозных моделей

Листинг 5 – Документация класса PredictiveModels

**model\_discipline\_number\_semestr**(subject\_name: str, term\_number: int, journal\_by\_year\_dataframe, students\_by\_journal\_id\_dataframe, rating\_by\_journal\_id\_dataframe, ege\_marks\_by\_students\_id\_dataframe)

Дисциплина {subject\_name} за {term\_number} семестр  
  
Входные данные:  
    subjectName(str): название предмета   
    term\_number(int): номер семестра предмета  
    journal\_by\_year\_dataframe(dataframe): dataframe с журналом по году  
    students\_by\_journal\_id\_dataframe(dataframe): dataframe с журналом по студентам  
    rating\_by\_journal\_id\_dataframe(dataframe): dataframe с журналом по рейтингу по предмету  
    ege\_marks\_by\_students\_id\_dataframe(dataframe): dataframe с журналом оценок ЕГЭ студентов  
Выходные данные:  
    merged\_df: dataframe с новым столбцом

**model\_discipline\_second\_semestr**(subject\_name: str, journal\_by\_year\_dataframe, students\_by\_journal\_id\_dataframe, rating\_by\_journal\_id\_dataframe, ege\_marks\_by\_students\_id\_dataframe)

Дисциплина 2 семестр с таким же предметом в прошлом семестре  
   
Входные данные:  
    subjectName(str): название предмета 2 семестра, который был в 1 семестре  
    journal\_by\_year\_dataframe(dataframe): dataframe с журналом   
по году  
    students\_by\_journal\_id\_dataframe(dataframe): dataframe с   
журналом по студентам  
    rating\_by\_journal\_id\_dataframe(dataframe): dataframe с   
журналом по рейтингу по предмету  
    ege\_marks\_by\_students\_id\_dataframe(dataframe): dataframe с   
журналом оценок ЕГЭ студентов  
Выходные данные:  
    merged\_df: dataframe с новыми столбцами

**model\_semester\_performance**(journal\_by\_year\_dataframe, students\_by\_journal\_id\_dataframe, rating\_by\_journal\_id\_dataframe, ege\_marks\_by\_students\_id\_dataframe, term\_number\_start: int, term\_number\_end: int, check\_type: str, forecast\_type: bool)

Успеваемость по семестру  
   
 Входные данные:  
    journal\_by\_year\_dataframe(dataframe): dataframe с журналом   
по году  
    students\_by\_journal\_id\_dataframe(dataframe): dataframe с   
журналом по студентам  
    rating\_by\_journal\_id\_dataframe(dataframe): dataframe с   
журналом по рейтингу по предмету  
    ege\_marks\_by\_students\_id\_dataframe(dataframe): dataframe с   
журналом оценок ЕГЭ студентов  
    term\_number\_start(int): от какого семестра брать выборку  
    term\_number\_end(int): до какого семестра брать выборку  
    check\_type(int): зачет/экзамен

Окончание листинга 5 приложения В

    forecast\_type(bool): добавить столбец Status в итоговую таблицу   
со значениями Учится - 1 // Отчислен - 0  
   
Выходные данные:  
    merged\_df: dataframe с новым столбцом

Листинг 6 – Документация класса ModelsSimplification

**decile\_group\_sum\_rating\_students\_by\_semestr**(merged\_df, term\_number: int)

Дециль в группе сумма рейтингов студента по всем дисциплинам номера  
(term\_number) семестра  
   
Входные данные:  
    merged\_df (dataframe): объединенный dataframe   
    term\_number (int): номер курса  
   
Выходные данные:  
    merged\_df: dataframe с новым столбцом

**decile\_median\_rating\_disciplin\_check\_type\_by\_semestr**(merged\_df, check\_type: int, term\_number: int)

Дециль медианного рейтинга дисциплины зачета/экзамена(check\_type)   
номера(term\_number) семестра  
   
Входные данные:  
    merged\_df (dataframe): объединенный dataframe   
    check\_type (int): тип дисциплины 1 - экзамен / 0 - зачет  
    term\_number (int): номер семестра  
   
Выходные данные:  
    merged\_df: dataframe с новым столбцом

Приложение Г. Модуль анализа данных

Листинг 6 – Документация класса DataAnalysis

**train\_by\_model**(dataframe\_model, column\_for\_learning: str, model\_training\_method: str, test\_size: float, n\_estimators: int, max\_depth: int, min\_samples\_split: int, min\_samples\_leaf: int, max\_features: int)

Обучение по модели  
   
Входные данные:  
    dataframe\_model (dataframe): dataframe с обработанной   
моделью  
    column\_for\_learning (str): столбец, значения которого будут предсказываться  
    model\_training\_method (str): тип модели для обучения  
    test\_size (float): размер выборки, который будет   
предназначен для тестирования (0.1 - 0.9)  
    n\_estimators (int): сколько деревьев решений необходимо   
создать в процессе обучения модели  
    max\_depth (int): максимальная глубина дерева решений  
    min\_samples\_split (int): минимальное количество образцов (наблюдений), которые необходимо иметь в узле, чтобы он мог быть   
разделен на две ветви.   
    min\_samples\_leaf (int): минимальное количество объектов, которое должно быть в листовой вершине дерева решений.   
    max\_features (int): максимальное количество признаков,   
которые рассматриваются при поиске наилучшего разделения на   
каждом узле дерева решений.   
Выходные данные:  
    clf: обученная модель  
    classification\_quality\_assessment: оценка качества обученной аналитической модели  
    X: все столбцы кроме column\_for\_learning для входных данных  
    y\_test: данные прогнозируемого столбца для тестирования  
    y\_pred: предсказания по обученной модели  
    results\_df: dataframe для вывода результатов

Листинг 7 – Документация класса GraphsBuilder

**feature\_importance\_graph**(clf, X, figsize=(12, 6))

График важности признаков  
   
Входные данные:  
    clf: обученная модель  
    X (dataframe): все столбцы кроме column\_for\_learning для   
входных данных из DataAnalysis  
    figsize (int, int): размер графика. По умолчанию (12,6).  
Выходные данные:  
    feature\_importance\_graph: график важности признаков

**plot\_predictions**(y\_test, y\_pred, figsize=(12, 6))

График предсказаний и реальных значений  
   
Входные данные:  
    y\_test: данные прогнозируемого столбца для тестирования из   
DataAnalysis  
    y\_pred: предсказания по обученной модели из DataAnalysis  
    figsize (int, int): размер графика. По умолчанию (12,6).  
Выходные данные:  
    plot\_predictions\_graph: график предсказаний и реальных   
значений

Приложение Д. Модуль интерфейса

Листинг 8 – Документация DjangoInterface

**get\_data\_api**(request):

Генерация интерфейса взаимодействия со страницей извлечения данных из API  
  
Входные данные:  
    request: объект запроса  
Выходные данные:  
    html: интерфейс взаимодействия со страницей извлечения   
данных из API

**get\_info\_from\_csv**(request):

Генерация интерфейса взаимодействия со страницей извлечения данных из CSV файлов  
  
Входные данные:  
    request: объект запроса  
Выходные данные:  
    html: интерфейс взаимодействия со страницей извлечения   
данных из CSV файлов

**get\_data\_analysis**(request):

Генерация интерфейса взаимодействия со страницей Анализа данных  
  
Входные данные:  
    request: объект запроса  
Выходные данные:  
    html: интерфейс взаимодействия со страницей Анализа данных

**get\_info**(request):

Генерация интерфейса информации о программе  
  
Входные данные:  
    request: объект запроса  
Выходные данные:  
    html: интерфейс с информацией о программе