ПРАВИТЕЛЬСТВО РОССИЙСКОЙ ФЕДЕРАЦИИ
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ
«ВЫСШАЯ ШКОЛА ЭКОНОМИКИ»

Факультет компьютерных наук
Образовательная программа «Программная инженерия»

СОГЛАСОВАНО
PhD, доцент базовой кафедры ПАО
Сбербанк “Финансовые технологии и
анализ данных”


Масютин
Алексей Александрович
«19» сентября 2023 г.

УТВЕРЖДАЮ
Академический руководитель
образовательной программы
«Программная инженерия»
профессор департамента программной
инженерии, канд. техн. наук

_____ В. В. Шилов
«___» _____ 2023 г.

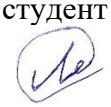
РАЗРАБОТКА РЕКОМЕНДАТЕЛЬНОГО СЕРВИСА ДЛЯ СТУДЕНТОВ МАГИСТРАТУРЫ И БАКАЛАВРИАТА НИУ ВШЭ

Пояснительная записка

ЛИСТ УТВЕРЖДЕНИЯ

RU.17701729.11.04-01 81 01-1

Исполнитель
студент группы БПИ206


/ Л.А.Поляков /
«19» сентября 2023 г.

Подп. и дата	
Инв. № дубл.	
Взам. инв. №	
Подп. и дата	
Инв. № подл	

Москва 2023

УТВЕРЖДЕН
RU.17701729.11.04-01 81 01-1

РАЗРАБОТКА РЕКОМЕНДАТЕЛЬНОГО СЕРВИСА ДЛЯ СТУДЕНТОВ МАГИСТРАТУРЫ И БАКАЛАВРИАТА НИУ ВШЭ

Пояснительная записка

RU.17701729.11.04-01 81 01-1

Листов 10

Подп. и дата	
Инв. № дубл.	
Взам. инв. №	
Подп. и дата	
Инв. № подл	

Москва 2023

СОДЕРЖАНИЕ

1. ВВЕДЕНИЕ

1.1.Название программы

Название программы – “Рекомендательный сервис для студентов магистратуры и бакалавриата НИУ ВШЭ”.

1.2.Документы, на основании которых ведется разработка

Разработка велась в рамках задания на курсовую работу в соответствии с учебным планом подготовки бакалавров (НИУ ВШЭ, факультет компьютерных наук) по направлению «Программная инженерия».

2. НАЗНАЧЕНИЕ И ОБЛАСТЬ ПРИМЕНЕНИЯ

2.1.Назначение программы

2.1.1. Функциональное назначение

Сервис является веб-сервисом позволяющим с помощью моделей машинного обучения анализировать данные об успеваемости студентов и давать разного рода прогнозы.

2.1.2. Эксплуатационное назначение

Сервис может быть использован студентами для прогнозирования вероятности их отчисления и для рекомендаций им подходящих курсов для изучения на основании их успеваемости.

2.2.Краткая характеристика области применения

Сервис может быть применен в образовательной сфере.

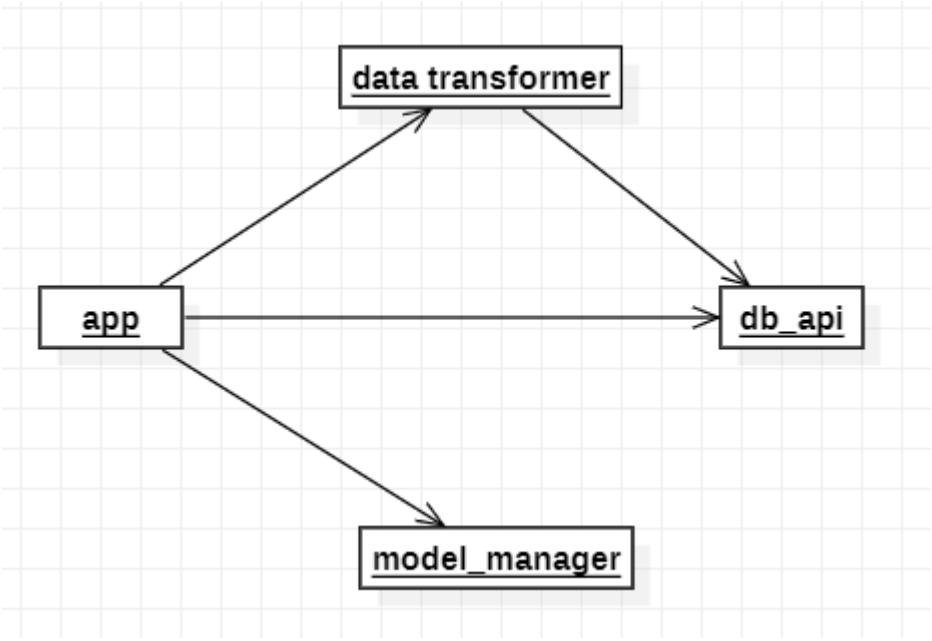
3. ТЕХНИЧЕСКИЕ ПОКАЗАТЕЛИ

3.1.Постановка задачи на разработку программы

Модуль должен обеспечивать функционал оценки вероятности оттока обучающегося с образовательной программы по причине риска неуспеваемости, а также функционал выявления значимых факторов, которые приводят к событию оттока на основе образовательной траектории обучающегося. Модуль должен иметь возможность выявления паттернов для потенциально проблемных обучающихся в виде комбинаций оценок предыдущих курсов. На основе работы модуля пользователь может получить предупреждение о том, что нужно дополнительно пройти тот или иной курс.

3.2.Описание алгоритма и функционирования программы

3.2.1. Архитектура сервиса



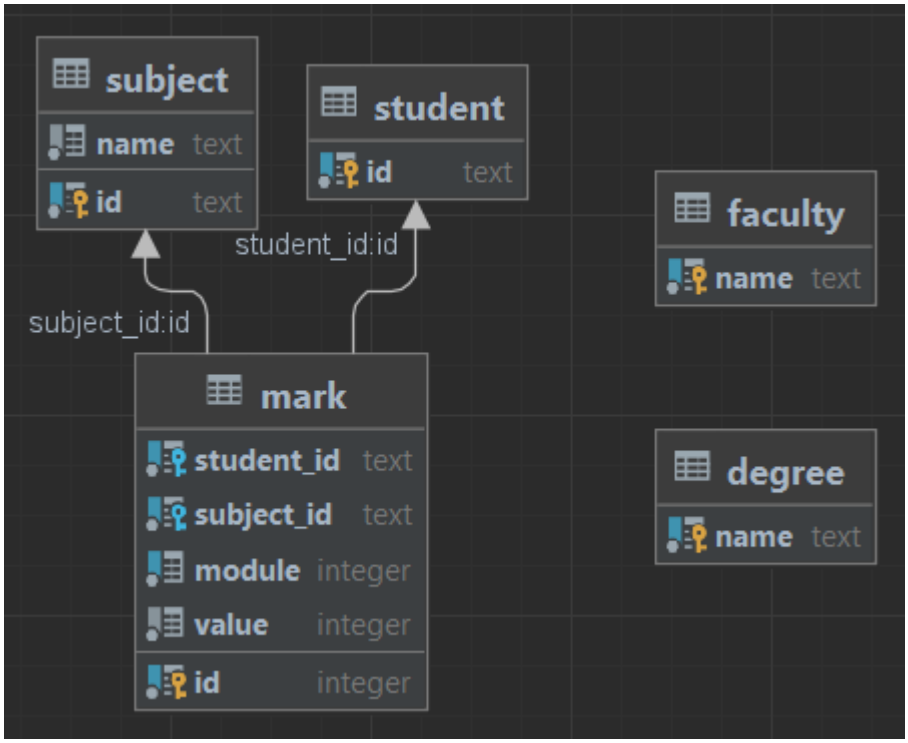
Сервис написан на языке Python и состоит из четырех главных компонент:

1) app – входная точка в программу, через app можно загрузить данные и получить прогнозы моделей
2) data_transformer – трансформер парсит csv-данные и сохраняет их, а также строит датафрейм по сохраненным данным
3) db_api – апишка для базы данных PostgreSQL
4) model_manager – менеджер моделей

Когда приходит запрос на загрузку данных по студентам, данные парсятся с помощью data_transformer и сохраняются в базу. Когда приходит запрос на получение прогноза по студенту, сначала с помощью data_transformer строится датафрейм, затем через model_manager берется модель под факультет и степень образования студента, и она отдает

прогноз. Каждая модель обучена на данных по конкретному факультету и степени образования и все модели наследуют общий класс Model.

3.2.2. База данных



База данных выбрана PostgreSQL. Данные хранятся в таблицах:

1) faculty – факультеты
2) degree – степени образования
3) subject – предметы, предмет привязан к факультету и степени
4) student – студенты, студент привязан к факультету и степени
5) mark – оценки, оценка ставится в конкретном модуле и привязана к студенту и предмету

3.2.3. API сервиса

API сервиса обрабатывает следующие ручки:

1) POST /subjects – загрузка предметов в сервис. Обязательные аргументы – faculty, degree. Предметы загружаются в виде csv-файла в формате:

ID,Subject

...

В ответе возвращается количество добавленных предметов.

2) POST /data – загрузка оценок в сервис. Обязательные аргументы – faculty, degree. Данные загружаются в виде csv-файла. Структура файла:

ID,Факультет,Образовательная программа,Уровень обучения,Курс,Модуль,Предмет_1, Предмет 2,...

...

В ответе возвращается количество добавленных или обновленных оценок

3) GET /prediction – получение вероятности отчисления студента. Обязательные аргументы – faculty, degree, id. Возвращает вероятность.
4) GET /recommend – получение списка рекомендаций для студента. Обязательные аргументы – faculty, degree, id, необязательный – n – количество рекомендаций (по умолчанию = 5).

3.2.4. Модели машинного обучения

Были использованы следующие модели машинного обучения:

1) Decision Tree Classifier - модель машинного обучения, которая представляет собой дерево с узлами и листьями. Узлы дерева представляют собой условия, применяемые к признакам, а листья — конечные определенные для наблюдений классы
2) Logistic Regression - модель машинного обучения, линейный классификатор, позволяющий оценивать апостериорные вероятности принадлежности объектов классам
3) Random Forest Classifier - модель машинного обучения, в основе которой лежит ансамбль решающих деревьев. Каждое из деревьев строится на случайном подмножестве признаков (столбцов) и случайном подмножестве наблюдений из обучающей выборке (строк)
4) CatBoost Classifier - это библиотека градиентного бустинга, разработанная компанией Яндекс. Она представляет собой эффективную реализацию алгоритма градиентного бустинга и использует особый тип деревьев решений, называемых "небрежными" (oblivious) деревьями, для построения сбалансированных деревьев

3.2.5. Предобработка данных

Сырые данные, которые достаются из базы и поступают в модель машинного обучения для получения прогноза, содержат лишь модуль и оценки по предметам за этот модуль. Поэтому над данными проводится предобработка в следующем порядке:

1) кумулятивное суммирование признаковых строк с функцией ffill(), таким образом для последнего модуля для каждого предмета хранится сумма оценок или NaN, если оценок нет

2) ввод dummy-переменных, принимающих значение 0 или 1 в зависимости от того, стоит ли NaN в кумулятивной сумме или нет
3) замена оставшихся NaN нулями; за счет ввода dummy-переменных можно не бояться потерять различие 0 и NaN в оценках

3.2.6. Использование моделей

Как уже было сказано в разделе “Архитектура сервиса” по студенту строится датафрейм и он уже отдается в модель для получения прогноза. Для получения вероятности отчисления студента используется обычная функция predict_proba() модели классификатора, которая возвращает вероятность получения каждого класса (класс 1 – студент будет отчислен). Для получения рекомендаций для студента используется атрибут feature_importances_ модели, который возвращает важность признаков. Эти признаки в порядке уменьшения важности добавляются в ответ при условии, что признак – предмет, и у студента по этому предмету не стоит оценка 8+.

3.2.7. Замечания

Стоит отметить, что модель не должна до/переобучаться с добавлением новых данных, как может показаться на первый взгляд - так как эти данные поступают за последний модуль, еще нет таргета – неизвестно, отчислен ли студент. Предполагается, что с завершением семестра модели до/переобучаются на новых данных вне контекста данного сервиса.

3.3.Описание и обоснование выбора метода организации входных и выходных данных

Входные данные с предметами и оценками студентов для ручек POST были предоставлены в таком формате после подписания NDA соглашения.

4. ОЖИДАЕМЫЕ ТЕХНИКО-ЭКОНОМИЧЕСКИЕ ПОКАЗАТЕЛИ

4.1.Ориентировочная экономическая эффективность

Улучшение образовательного процесса положительно сказывается как на вузе, так и на его учащихся и выпускниках, что является фундаментом экономики страны.

4.2.Предполагаемая потребность

Предполагаемая потребность обуславливается тем, что в вузе студенты учатся ежегодно.

5. ИСТОЧНИКИ, ИСПОЛЬЗОВАННЫЕ ПРИ РАЗРАБОТКЕ

1) ГОСТ 19.101-77 Виды программ и программных документов. // Единая система программной документации. – М.: ИПК Издательство стандартов, 2001.

2) ГОСТ 19.102-77 Стадии разработки. // Единая система программной документации. – М.: ИПК Издательство стандартов, 2001.

3) ГОСТ 19.103-77 Обозначения программ и программных документов. // Единая система программной документации. – М.: ИПК Издательство стандартов, 2001.

4) ГОСТ 19.104-78 Основные надписи. // Единая система программной документации. – М.: ИПК Издательство стандартов, 2001.

5) ГОСТ 19.105-78 Общие требования к программным документам. // Единая система программной документации. – М.: ИПК Издательство стандартов, 2001.

6) ГОСТ 19.106-78 Требования к программным документам, выполненным печатным способом. // Единая система программной документации. – М.: ИПК Издательство стандартов, 2001.

7) ГОСТ 19.404-79 Пояснительная записка. Требования к содержанию и оформлению. // Единая система программной документации. – М.: ИПК Издательство стандартов, 2001.

8) ГОСТ 19.603-78 Общие правила внесения изменений. // Единая система программной документации. – М.: ИПК Издательство стандартов, 2001.

9) ГОСТ 19.604-78 Правила внесения изменений в программные документы, выполненные печатным способом. // Единая система программной документации. – М.: ИПК Издательство стандартов, 2001.

10) https://github.com/anamarina/RecSys_course

11) формализация задачи и метрики качества ранжирования https://neerc.ifmo.ru/wiki/index.php?title=%D0%A0%D0%B0%D0%BD%D0%B6%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5

12) 4 видеозаписи лекций Жени Соколова, а то у меня только презы и ноуты выложены https://www.lektorium.tv/node/33563