МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное бюджетное образовательное учреждение высшего образования

«ПЕНЗЕНСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»

Факультет вычислительной техники

Кафедра «Вычислительная техника»

Направление подготовки: 09.04.01 «Информатика и вычислительная техника»

Магистерская программа: «Прикладной искусственный интеллект»

МАГИСТЕРСКАЯ ДИССЕРТАЦИЯ

на тему

ИССЛЕДОВАНИЕ МЕТОДОВ СБОРА И ОБРАБОТКИ ДАННЫХ ДЛЯ АЛГОРИТМОВ МАШИННОГО ОБУЧЕНИЯ

Студент 14.06.24 Майорова Алла Романовна
(подпись, дата) (ФИО полностью)

Руководитель 04.06.24 Зинкин С.А.
(подпись, дата) (фамилия, инициалы)

Нормоконтролёр 04.06.24 Бычков А.С.
(подпись, дата) (фамилия, инициалы)

Рецензент нач. отдела 17.06.24 Севастьянов А.В.
(должность, место работы) (подпись, дата) (фамилия, инициалы)

Работа допущена к защите (протокол зас. кафедры от 27.05.2024 г. № 11)

Заведующий кафедрой _____ Митрохин М.А.
(подпись) (фамилия, инициалы)

Работа защищена с отметкой отлично (протокол заседания ГЭК от 21.06.2024 г. № 34)

Секретарь ГЭК _____ Гурин Е.И.
(подпись) (фамилия, инициалы)

Пенза 2024

Федеральное государственное бюджетное образовательное учреждение
высшего образования
«ПЕНЗЕНСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»

Факультет
вычислительной техники

Кафедра
«Вычислительная техника»

«Утверждаю»
Заведующий кафедрой
_____М.А. Митрохин
«13» сентября 2022 г.

ЗАДАНИЕ
НА МАГИСТЕРСКУЮ ДИССЕРТАЦИЮ

1. Студент Майорова Алла Романовна гр. 22ВВИм2 факультета вычислительной техники

2.Тема работы «Исследование методов сбора и обработки данных для алгоритмов машинного обучения»

Тема утверждена приказом ПГУ № 765/о от "13" сентября 2022 г.

3. Руководитель магистерской диссертации д.т.н., профессор Зинкин С.А.

4. Задание на работу (назначение разработки, исходные данные и т.п.)

Провести исследование основных особенностей методов сбора и обработки данных.

Собрать и обработать данные для практической задачи. Выявить закономерности в данных. Провести обучение на разных наборах данных. Сравнить результаты, выбрать лучшее решение.

5. Перечень подлежащих разработке вопросов:

1. Анализ существующих методов сбора данных, выявление лучших и худших из них.
2. Выявление признаков, на основе которых будет осуществляться подбор метода сбора данных.
3. Сбор и анализ данных для формирования базы, на основе которой будет проходить обучение модели.
4. Обучение ML- моделей на основе собранной базы.
5. Анализ результатов обучения.

6. Календарный график выполнения работы

№ п/п	Наименование этапов работы	Объем работы	Срок выполнения	Подпись руководителя
1	Анализ предметной области сбора данных для ML алгоритмов.	10%	13.09.2022 – 14.11.2022	
2	Анализ предметной области обработки данных для ML алгоритмов.	10%	15.11.2022 – 16.01.2023	
3	Сбор и анализ данных для обучения	15%	17.01.2023 – 20.04.2023	
4	Обработка данных, выделение признаков.	25%	21.04.2023 – 23.09.2023	
5	Построение моделей и оценка качества	20%	24.09.2023 – 26.01.2024	
6	Сравнение результатов на нескольких наборах данных	10%	27.01.2024 – 29.03.2024	
7	Оформление выпускной работы	10%	29.03.2024 – 31.05.2024	

Дата выдачи задания "13" сентября 2022 г.

Руководитель магистерской диссертации _____ 13.09.22 Винкин С.А.
(подпись, дата)

Задание к исполнению принял студент _____ 13.09.22 Майорова А.Р.
(подпись, дата)

Работу к защите допустить
Декан факультета _____ 19.06.2024 Фионова Л.Р.
(подпись, дата)

РЕФЕРАТ

Магистерская работа содержит 84 листа, 36 рисунков, 6 таблиц, 23 источника, 2 приложения

МАШИННОЕ ОБУЧЕНИЕ, СБОР ДАННЫХ, ОБРАБОТКА ДАННЫХ, СЛИЯНИЕ ДАННЫХ, АЛГОРИТМЫ МАШИННОГО ОБУЧЕНИЯ

Основной целью работы является исследование эффективности, скорости сбора и обработки данных для решения конкретных задач машинного обучения.

Обьектом исследования в магистерской диссертации выступают методы сбора и обработки данных.

Предметом исследования являются особенности различных методов сбора и обработки данных с последующим их использованием в машинном обучении.

В рамках данной работы проведен обзор алгоритмов машинного обучения, выделены основные критерии сбора данных и основные источники для практической задачи.

В результате магистерской работы был проведен анализ критериев качества моделей машинного обучения для 5 наборов данных. Выделены их достоинства и недостатки. Улучшено качество моделей, путем обработки исходных данных.

Результаты исследования могут быть применены в области машинного обучения, а также в области научных исследований в рамках data-ориентированного подхода к машинному обучению.

СОДЕРЖАНИЕ

ПЕРЕЧЕНЬ СОКРАЩЕНИЙ И ОБОЗНАЧЕНИЙ 4
ВВЕДЕНИЕ 5
1 Анализ предметной области 7
1.1 Машинное обучение в AI 7
1.2 Общая характеристика задач машинного обучения 9
1.3 Этапы машинного обучения 10
2 Обзор методов обработки данных для ML алгоритмов 13
2.1 Источники сбора данных 13
2.2 Критерии выбора способа сбора данных 17
2.3 Обзор составных частей набора данных 18
2.4 Этап анализа данных 19
2.5 Подготовка данных для машинного обучения 20
3 Используемые инструменты 27
3.1 Выбор сред 27
3.2 Выбор моделей 27
3. 3 Выбор библиотек 28
4 Эксперименты и результаты исследования 30
4.1 Постановка задачи 30
4.2 Выбор способа сбора данных 30
4.3 Оценка качества наборов данных 34
4.3.1 Оценка качества первого набора данных 34
4.3.2 Оценка качества второго набора данных 43
4.4 Балансировка классов 49
4.5 Обучение моделей 51
5 Характеристика результатов обучения 55
ЗАКЛЮЧЕНИЕ 58
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ 59
ПРИЛОЖЕНИЕ А Листинг программы 62
ПРИЛОЖЕНИЕ Б Презентация 76

ПЕРЕЧЕНЬ СОКРАЩЕНИЙ И ОБОЗНАЧЕНИЙ

В настоящей работе применяют следующие сокращения и обозначения

БД – База данных

ЕГЭ – Единый государственный экзамен

ИИ – Искусственный интеллект

ЭИОС – Электронная информационно-образовательная среда

AI (Artificial intelligence) – Искусственный интеллект

API (Application programming interface) – Программный интерфейс приложения

EDA (Exploratory Data Analysis) – Разведочный анализ данных

ML (Machine Learning) – Машинное обучение

SQL (Structured Query Language) – язык структурированных запросов

ВВЕДЕНИЕ

Изначально разработанные для обработки больших объемов данных, технологии машинного обучения и искусственного интеллекта находят все большее распространение в современном мире, охватывая новые сферы жизни человека: от развлечений и социального взаимодействия до образования и экономики.

В настоящее время существует два подхода к увеличению точности машинного обучения. Это ориентированный на данные и ориентированный на улучшение параметров модели.

Темой магистерской диссертации является исследование методов сбора и обработки данных для алгоритмов машинного обучения.

Основной целью работы является исследование эффективности и скорости сбора и обработки данных для решения конкретных задач машинного обучения.

Гипотезой исследования является предположение о том, что эффективность различных методов сбора и обработки данных влияет на производительность алгоритмов машинного обучения, где более качественные и понятные данные приводят к созданию более точных и надежных моделей.

Обьектом исследования в магистерской диссертации выступают методы сбора и обработки данных.

Предметом исследования является применение различных методов сбора и обработки данных с последующим их использованием в машинном обучении.

Основными задачами работы является:

– анализ существующих методов сбора данных, выявление лучших и худших из них,

– выявление признаков, на основе которых будет осуществляться подбор метода сбора данных,

– сбор и анализ данных для формирования базы, на основе которой

будет проходить обучение модели,

– обучение ML- моделей на основе собранной базы,

– анализ результатов обучения.

Магистреская диссертация направлена на повышение эффективности применения сбора и обработки данных для повышения точности машинного обучения.

1 Анализ предметной области

Начальный этап задач, предполагающий работу в рамках информационных технологий, представляет собой теоретический анализ многих аспектов: технологической, предметной и информационной основы задачи, инструментов, с помощью которых можно прийти к наиболее эффективному решению, разделения задачи на последовательные функциональные блоки, реализации из которых приведет к ее решению, возможные архитектурные решения, которые могут быть использованы для получения приложения, отвечающего всем требованиям.

Первым шагом к выполнению необходимого анализа будет исследование общих характеристик методов сбора и обработки данных, а также существующих алгоритмов машинного обучения.

1.1 Машинное обучение в AI

Искусственный интеллект основан на принципе, согласно которому человеческий интеллект может быть определен таким образом, чтобы машина могла имитировать его и выполнять необходимые задачи. Место ML в AI представленно на рисунке 1.

Рисунок 1 – Место ML в AI

Машинное обучение определяют, как “Область исследований, которая дает компьютерам возможность учиться без явного программирования”.

Это исследование того, как сделать машины более похожими на человека в их поведении и решениях, предоставив им возможность учиться и разрабатывать свои собственные программы [1].

На машины подаются данные хорошего качества, и для построения моделей ML используются различные алгоритмы для обучения машин на основе этих данных. Выбор алгоритма зависит от типа имеющихся данных и вида деятельности, которую необходимо автоматизировать.

Машинное обучение отличается от традиционного программирования, тем что в традиционном программировании мы бы загрузили входные данные и хорошо написанную и протестированную программу в машину для генерации выходных данных, а при машинном обучении входные данные вместе с выходными подаются в машину на этапе обучения, и она сама разрабатывает программу [1]. Схема различий между машинным обучением и классическим программированием представленна на рисунке 2.

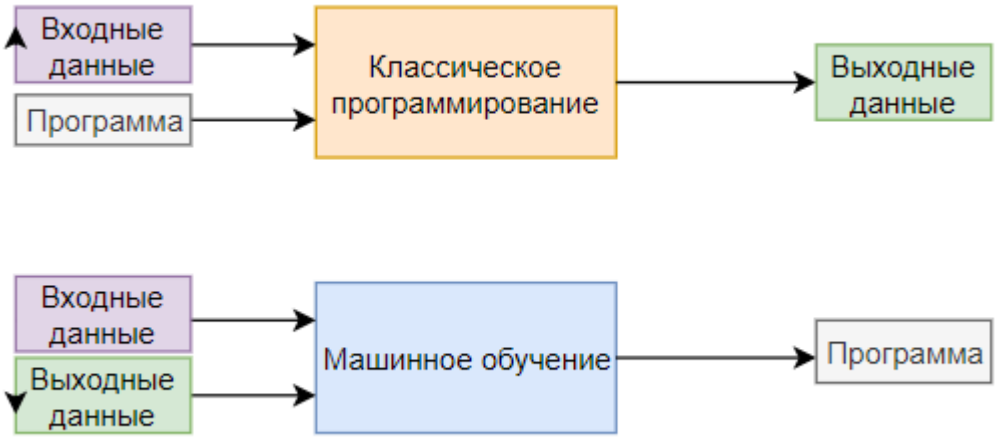


Рисунок 2 – Отличие классического программирования от ML

Машинному обучению уделяется все необходимое внимание. Это помогает автоматизировать многие задачи, особенно те, которые могут быть выполнены только людьми. Тиражирование таких задач на машины может быть достигнуто только с помощью машинного обучения.

Предприятия могут автоматизировать рутинные задачи и быстро создавать модели для анализа данных.

Различные отрасли промышленности зависят от огромных объемов данных, которые они применяют для оптимизации своей деятельности и принятия разумных решений. Эти данные также помогают в создании моделей, обрабатывающих и анализирующих большие объемы сложных данных для получения точных результатов. Такие модели точны и масштабируемы, и функционируют с меньшими затратами времени на выполнение работ [2]. Создавая такие точные модели машинного обучения, предприятия могут использовать всю выгоду автоматизации.

1.2 Общая характеристика задач машинного обучения

Задачи машинного обучения с учителем можно разделить на четыре вида:

1) алгоритм классификации выполняет либо бинарную, то есть отвечает на вопрос «да»/ «нет», либо многоклассовую классификацию, то есть определяет класс объекта, попавшего на вход модели,

2) алгоритм регрессии выдает какое-то числовое значение,

3) ранжирование – это алгоритм, при котором ответы надо получить сразу на множестве объектов, после чего отсортировать их по значениям ответов,

4) прогнозирование – это алгоритм, в котором объектами являются отрезки временных рядов, обрывающиеся в тот момент, когда требуется сделать прогноз на будущее [2].

Задачи машинного обучения без учителя можно разделить на три вида:

1) алгоритм кластеризации находит правила классификации и количество классов. Основное отличие от задач классификации заключается в том, что группы и принципы их разделения неизвестны,

2) при поиске ассоциативных правил исходные данные представляются в виде признаковых описаний. Требуется найти такие наборы признаков, и такие значения этих признаков, которые особенно часто встречаются в признаковых описаниях объектов [2],

3) алгоритм снижения размерности заключается в том, чтобы по исходным признакам с помощью некоторых функций преобразования перейти к наименьшему числу новых признаков, не потеряв при этом никакой существенной информации об объектах выборки. В классе линейных преобразований наиболее известным примером является метод главных компонент [2].

Деление машинного обучения на виды представленно на рисунке 3.

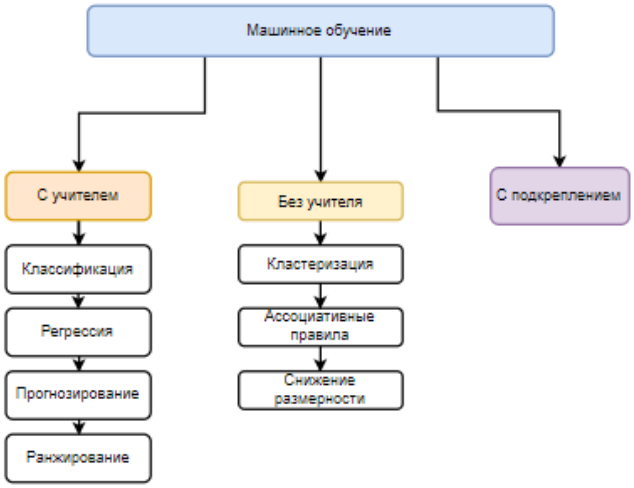


Рисунок 3 – Виды машинного обучения

1.3 Этапы машинного обучения

Этапы проекта с использованием машинного обучения представлены на рисунке 4.

Рисунок 4 – Этапы проекта с использованием машинного обучения

Разработка ML имеет 11 этапов:

1) постановка задачи. На данном этапе определяется решаемая задача, цели модели, идеальный результат модели,

2) выбор подходящих метрик. На этом этапе определяются метрики, по которым будет оценена полученная модель,

3) сбор данных. На данном этапе собираются данные для модели, они берутся из разных источников или же используются готовые датасеты. Этот шаг очень важен, потому что от качества и количества данных, которые будут собраны, напрямую определяется, насколько хорошей может быть прогностическая модель,

4) анализ данных. На этом этапе подробно изучаются все собранный данные. Насколько они адекватны и полезны,

5) предобработка данных. На этом этапе данные загружаются в подходящее место и готовятся для их использования в машинном обучении. На этом этапе в зависимости от предстоящей задачи машинного обучения обрабатываются данные. Например, обрабатываются пропуски и выбросы и масштабируются данные,

6) выбор модели. Существует множество моделей, которые исследователи и специалисты по обработке данных создали за эти годы. Некоторые из них очень хорошо подходят для графических данных, другие - для последовательностей (например, текста или музыки), некоторые - для числовых данных, остальные - для текстовых данных. Выбор модели осуществляется на основе полученных данных и поставленной задачи,

7) обучение. Часто этот этап считается основной частью машинного обучения. На этом этапе данные используются для постепенного улучшения способности модели,

8) оценка. Оценка позволяет протестировать модель на данных, которые никогда не использовались для обучения,

9) настройка параметров. После того, как была проведена оценка, нужно посмотреть, можно ли еще каким-либо образом улучшить обучение. Можно сделать это, изменив параметры,

10) внедрение модели в продуктовую среду. Создание подходящего интерфейса пользователя или внедрение в существующий интерфейс,

11) поддержка/улучшение модели. На этом этапе необходима поддержка модели в продуктовой среде.

Вложение слишком больших усилий только в построение модели вместо того, чтобы сосредоточиться на данных, является распространенной ошибкой, которой следует избегать в успешных проектах ML.

Вывод: в рамках этого раздела были достигнуты следующие результаты: произведён исчерпывающий анализ области, в рамках которой планируется вести исследоввание. Были описаны основные задачи, решаемые с помощью машинного обучения, выделены этапы разработки моделей ML.

2 Обзор методов обработки данных для ML алгоритмов

2.1 Источники сбора данных

Рассмотрим подробнее этапы работы с данными.

Сбор данных – самый важный шаг в решении любой задачи машинного обучения. Ваша модель может быть настолько хороша, насколько хорош набор данных, из которого она создана. Особенно это касается решения задач машинного обучения с учителем.

На самом деле, большинство проблем моделей машинного обучения вызваны не моделями, а проблемами в наборе данных. И все же процесс создания набора данных является недооцененной темой. Это связано с тем, что создание и улучшение наборов данных – это человеческая задача, которая, как правило, отнимает очень много времени. В мире искусственного интеллекта задачи, требующие человеческого труда, не считаются захватывающими.

В любом случае, прежде чем мы обучим модель, нам нужен набор данных [3].

Существует шесть основных методов создания этих наборов:

1) выбор готового набора данных. Такой набор можно найти на платформах для обучения и соревнований по Data Science и ML. На таких сайтах датасеты представляют собой набор данных для решения конкретной, поставленной задачи. Интернет-порталы и сообщества по DS и ML, где содержатся датасеты для индивидуального и группового обучения, а также научных исследований. Открытые данные – информация машиночитаемого формата, размещенная в Интернете и доступная для всеобщего свободного использования и дальнейшей републикации без ограничений авторского права и других способов контроля [4],

2) парсинг и скраппинг сторонних ресурсов. Парсинг – это процесс анализа содержимого веб-страниц с помощью роботов-парсеров – специальных программ или скриптов. Далеко не все сайты предоставляют API-интерфейс для автоматического сбора с них нужной информации. В этом случае на помощь приходят технологии парсинга [3]. Самые известные

парсеры в сети – это роботы поисковых систем, которые анализируют сайты, сохраняют результаты анализа в своих базах, чтобы выдать при поиске наиболее релевантные и актуальные документы. Можно написать свой собственный веб-парсер на Python или PHP для автоматического сбора данных из внешних ресурсов по заданным характеристикам, а можно воспользоваться уже готовыми решениями [3],

3) сбор данных на основе данных, имеющихся у заказчика.

Некоторые организации пред тем, как использовать машинное обучение несколько лет собирают данные. Организации получают данные используя, имеющиеся у них данные для задач прогнозирования и классификации. Некоторые же собирают данные с пользователей, как только им понадобилась определенная информация, например, с помощью опросов [3]. Системы сбора информации сразу заполняют заранее подготовленную таблицу структурированными данными. Например, так можно собрать датасет о демографии клиентов на основе опроса, который они заполняют на сайте. Наиболее успешными проектами являются те, которые интегрируют стратегию сбора данных во время жизненного цикла продукта. Сбор данных не может быть одноразовым, он должен быть встроен в основной продукт. По сути, каждый раз, когда пользователь использует продукт необходимо собирать данные о взаимодействии. Данные от заказчика могут быть получены в виде готовых таблиц. Также в виде предоставленного доступа к БД, или же к хранилищу данных,

4) сбор данных вручную. Люди лично, без какой-либо автоматизации, осматривают объекты и описывают их признаки. Именно так создаются обучающие наборы данных из данных, которые изначально не структурированы. Например, так создаются наборы данных для распознавания изображений – люди смотрят на фотографии и пишут, что именно на них изображено. Но автоматический сбор данных обладает явными преимуществами, такими как скорость, сокращение расходов и т. д. [2],

5) краудсорсинг – это сбор данных с помощью привлечения группы людей. В настоящее время существует множество компаний, занимающихся краудсорсингом,

6) генерация синтетических данных. Генерация синтетических данных – это безопасное, быстрое и масштабируемое решение по сравнению с традиционными инструментами. Синтетическая генерация используется, когда невозможно по временным или человеческим ресурсам обеспечить сбор. Например, необходимо собрать датасет с документами, содержащими персональные данные. Основные инструменты для генерации данных представлены в таблице 1.

Таблица 1 – Инструменты для генерации данных

Способ	Таблицы	Изображения и видео	Текст	Аудио
Модели	GANS CTGAN WGAN WGAN-GP VAES	GANS VAES	GPT-3 GPT-4 PaLM LLAMA Bard	WaveNet Jukebox Voicebox
Инструменты	SDV Copulas	Unity Blender ThreeDWorld	Jasper Copy.ai	Ableton Live iZotope Audacity
Сервисы	MOSTLY AI YData Hazy DataRobot Tonic	NVIDIA Omniverse CVEDIA	Rytr.me Notion Al	ElevenLabs Murf Lovo Mubert

После анализа основных методов было выполнено их сравнение, выделены основные недостатуи п реимещества каждого.

Сравнительная характеристика методов сбора данных представлена в таблице 2.

Таблица 2 – Сравнительная характеристика методов сбора данных

Способ	Преимущества	Недостатки
Выбор готового набора данных	Малые временные затраты на сбор данных. Уменьшение трудовых затрат на очистку и разметку данных (могут быть использованы уже очищенные данные). Объем датасета.	Данные могут быть не актуальны для некоторого типа задач. Способ не всегда подходит к задачам прогнозирования. Недостоверность данных.
Данные, полученные парсингом сторонних ресурсов	Возможно использование очистки данных, встроенной в парсер. Встроенная оптимизация полей в готовых парсерах.	Большие временные затраты на очистку и разметку данных. Недостоверность некоторых источников для сбора, необходимость использования проверенных ресурсов. Недостатки алгоритма парсера.
Данные, предоставленные заказчиком	Актуальность предоставленных данных. Идеальное сопряжение с задачами прогнозирования.	Избыточность данных. Временные затраты на очистку данных. Возможно предоставление неструктурированных данных
Ручной сбор данных	Актуальность предоставленных данных.	Человеческий фактор. Скорость сбора данных. Стоимость человеческих ресурсов. Ограничения доступов. Сложность масштабирования.
Краудсорсинг	Масштабируемость. Отсутствие смещений данных. Снижение нагрузки на кадровый резерв организации.	Неустойчивое качество данных, Отсутствие стандартизации данных. Человеческий фактор.

Продолжение таблицы 2

Способ	Преимущества	Недостатки
Синтетическая генерация данных	Снижение нагрузки на кадровый резерв организации. Меньше ограничений на использование конфиденциальных данных. Данные для обучения без пробелов и выбросов. Экономия денег и времени. Получение редких данных.	Требуются большие вычислительные мощности. Требуется программа для генерации данных, если нужные данные не соответствуют стандартным принципам [1]. Модели, обученные на синтетических данных, хорошо работают при поступлении синтетических данных, с реальными могут возникнуть проблемы.

В некоторых проектах используется гибридный способ сбора, например, основные данные предоставляются заказчиком, а недостающие собираются с помощью парсинга официальных ресурсов (в основном для задач прогнозирования) или использования готового набора данных (в основном для задач классификации). Также часто гибритизируют синтетическую генерацию и ручной сбор, например, для задач классификации или распознавания документов, содержащих персональные данные [3].

2.2 Критерии выбора способа сбора данных

Выбор способа сбора представляет собой очень сложную задачу. Достаточно сложно оценить возможности как тот или иной метод отразится на результате модели до ее реализации [3].

Выбор способа сбора данных определяется рядом критериев. К ним относятся:

– отношение стоимость-качество. Необходимо наличие баланса между стоимостными параметрами и качеством полученных данных. Проблема качества данных представляет собой достаточно серьезную тему,

– актуальность и соответствие задаче. Это, пожалуй, самая важная характеристика. Учитывая постоянные изменения в мире, данные имеют достаточно короткие сроки актуальности,

– масштабируемость. Эта характеристика означает, что добавление новых данных должно давать прогнозируемое увеличение производительности модели при приемлемых затратах,

– временные характеристики. Скорость того, насколько быстро можно собрать определенный датасет для конкретной задачи также является критически важным фактором при обучении модели. В особенности если срок решения задачи рекордно низкий [4].

2.3 Обзор составных частей набора данных

Составные части датасета:

– объект: фото, фрагмент аудио, заболевание и другие,

– характеристики объекта: конкретные признаки, связи с другими объектами.

Характеристики объекта задаются в числовом виде. Если же в датасете могут содержаться признаки в виде словесных обозначений, необходимо учесть этот момент при обработке данных перед обучением.

Виды датасетов:

– наборы числовых данных,

– двумерные наборы данных,

– многомерные наборы данных,

– категориальные наборы данных,

– наборы корреляционных данных.

Набор числовых данных – это набор данных, в котором данные выражены в цифрах, а не на естественном языке. Числовые данные иногда называют количественными данными. Набор всех количественных данных /числовых данных называется набором числовых данных.

Набор данных, содержащий две переменные, называется двумерным набором данных. В нем рассматривается взаимосвязь между двумя переменными. Двумерный набор данных обычно содержит два типа связанных данных.

Если набор данных содержит три или более трех типов данных, то набор данных называется многомерным набором данных. Другими словами, многомерный набор данных состоит из отдельных измерений, которые получены как функция трех или более чем трех переменных. Явной связи между данными нет.

Категориальные наборы данных представляют особенности или характеристики человека или объекта.

Графовые данные. Данные о связях между объектами, которые могут быть представлены визуально в виде схемы из объектов, соединённых стрелками. А могут быть в виде таблицы, где в строках и колонках указаны объекты, а в пересечениях — связи между ними.

Для разных задач машинного обучения нужна подготовка разного набора данных. Например, набор, созданный специально для задачи регрессии может не подойти для задачи классификации. Он может не обладать классификационными данными.

2.4 Этап анализа данных

Исследовательский анализ данных – это задача, выполняемая специалистами по обработке данных для ознакомления с данными.

Исследовательский анализ данных – это метод, который специалисты по обработке данных и другие специалисты используют для понимания наборов данных перед их моделированием.

EDA помогает определить наилучшие возможные способы манипулирования информационными ресурсами, облегчая изучение данных и обнаружение скрытых тенденций, проверку гипотез и предположений. Более того, метод тщательно анализирует данные, чтобы обеспечить следующие критерии:

– оптимальную интерпретацию в наборе данных,

– обнаруживать перспективные структуры,

– выявлять выбросы и аномалии,

– определять оптимальные настройки факторов,

– обнаруживать значимые переменные данных и многое другое.

Целью проведения EDA является определение характеристик набора данных.

2.5 Подготовка данных для машинного обучения

Невозможно загрузить все накопленные данные разного формата в алгоритм машинного обучения и получить на выходе правильные результаты. Кроме того, исходные данные часто искажены и ненадежны: они могут содержать значения, выходящие за пределы допустимых диапазонов, аномальные значения, а также пропуски [5].

Сам процесс подготовки данных для модели строится следующим образом:

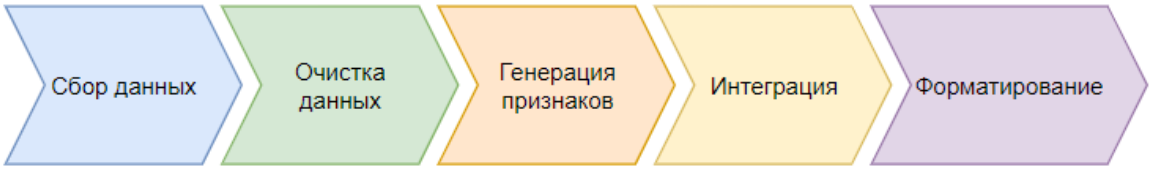


Рисунок 5 – Процесс подготовки данных

Процесс подготовки данных включает в себя 5 этапов:

1) выборка данных,

2) очистка данных,

3) генерациия признаков,

4) интеграция,

5) форматирование.

На первом этапе, сборе данных происходит отбор признаков и обьектов с учетом с учетом их релевантности для целей интеллектуального анализа данных. Данные могут располагаться в разных исходных системах, хранилище

данных или озере данных, которое становится все более распространенным хранилищем в средах больших данных, содержащих смесь структурированных и неструктурированных данных. Также могут быть использованы внешние источники данных. Откуда бы ни поступали данные, необходимо переместить их в хранилище данных для выполнения оставшихся этапов процесса.

Второй этап включает в себя набор шагов по подготовке данных. Начиная с исследования данных, профилирования и предварительной обработки до работы по очистке данных для исправления ошибок и других проблем с качеством данных. Происходит удаление опечаток, некорректных значений, пустых значений, исключение дублей, восстановление уникальности, целостности и логических связей. Преобразование данных также выполняется для обеспечения согласованности наборов данных [5].

На третьем этапе создание производных признаков и их преобразование в векторы для модели, а также трансформация для повышения точности алгоритмов машинного обучения.

На четвертом этапе происходит объединение данных из различных источников (информационных систем, таблиц, протоколов и т. д.), в том числе их агрегация, когда новые значения вычисляются путем суммирования информации из набора существующих записей [5].

На пятом этапе происходят синтаксические изменения, не меняющие смысла данных, но необходимые для средств моделирования, например, сортировка в определенном порядке или удаление ненужных знаков препинания в текстовых полях, обрезка «длинных» слов, округление до целого [5] и т. д.

Также необходимо помнить о данных проверках:

– проверка на соответствие типов данных. Здесь решается вопрос соответствуют ли типы данных столбцов их содержимому,

– выбор допустимых значений. Выбираются ли только допустимые значения для категориальных или порядковых полей,

– перекрестная проверка полей. Некоторые поля могут влиять на действительность других полей.

На этапе подготовки данных получаются данные в наилучшем формате для машинного обучения. Наличие качественных данных важнее, чем использование сложных алгоритмов, поэтому этот этап важен.

Далее будут рассмотрены критерии при подготовке данных.

Ключевыми критериями при подготовке данных являются отсутствующие значения, выбросы и группировка разреженных классов, категориальное кодирование, работа с искаженными данными, масштабирование и устранение смещений.

Отсутствующие значения один из важнейших критериев. Ведь большинство моделей машинного обучения требуют, чтобы все функции были полными. Следовательно, необходимо устранить недостающие значения. Самое простое решение – удалить все строки, в которых отсутствует значение, но таким способом можно потерять важную информацию или внести смещение. В качестве альтернативы можно присвоить значение и предоставить соответствующую замену отсутствующим данным. Обычные вычисления заключаются в использовании среднего значения. Для категориальных данных необходимо создать такую категорию, как “Неизвестно”.

Выбросы являются следующим критерием. Если во время анализа данных исследование выбросов обнаруживает только низкочастотные выбросы, которые, вероятно, являются ошибочными, можно рассматривать их как отсутствующие данные и либо удалять, либо заменять их. Однако, если предполагается, что можно потерять важные данные, удалив или изменив выбросы, необходимо использовать алгоритм машинного обучения и метрику оптимизации, устойчивую к выбросам.

Процесс удаления выбросов из набора данных называется обрезкой или усечением. Можете заменить выбросы методами, похожими на изменение недостающих данных, такими как среднее значение. Более сложные методы

включают упорядочивание, которое заменяет экстремальные значения минимальными и максимальными и дискретизацию, которая делит непрерывную переменную на дискретные группы.

Группировка разреженных классов также является одним из критериев.

На этапе преобразования данных рассматривается, как категориальные данные преобразуются в формат, который может считывать модель машинного обучения. Это усилие часто включает в себя создание функции для каждого отдельного значения в этой категории. Если каждый категориальный признак имеет много различных значений, это преобразование приводит к появлению еще многих признаков. Многие алгоритмы машинного обучения могут бороться со слишком большим количеством функций, называемых проклятием размерности.

Категориальное кодирование является очень важным критерием при обработке. Кодирование меток преобразует категориальные переменные в числовые представления, которые являются машиночитаемыми. Первое, что необходимо определить, – это являются ли ваши категории порядковыми. Например, уровень образования имеет порядок, при котором степень магистра оценивается выше, чем степень бакалавра. Необходимо сохранить эту взаимосвязь, используя метод порядковой кодировки. Кроме порядковой кодировки существуют также:

– кодирование Гельмерта,

– двоичное кодирование,

– фиктивное кодирование,

– кодирование меток,

– одноразовое кодирование,

– кодирование Джеймса Стейна,

– обратная разница,

– частотное кодирование.

Производительность многих алгоритмов зависит от кодировки признаков. Необходим выбор алгоритма кодирование исходя из полученных данных и поставленной задачи.

Работа с искаженными данными является критерием, поддерживающим нормальность. Нормальность часто предполагается с помощью статистических методов. При использовании алгоритмов регрессии, можно увидеть значительные улучшения, если преобразовать переменные с искаженными распределениями. Чтобы приблизиться к нормальному распределению, необходимо использование, логарифмов (основание e или основание 10), обратных чисел или методов Бокса-Кокса.

Масштабирование – это метод преобразования данных в определенный диапазон. Регрессионные алгоритмы чувствительны к изменению величины и диапазона между объектами. Цель масштабирования состоит в том, чтобы изменить значения каждого числового признака в наборе данных на общий масштаб. Таким образом, изменения в различных функциях становятся более сопоставимыми.

Масштабирование может быть выполнено с помощью нормализации.

Смягчение предвзятости также необходимо для корректной работы. Если вы обнаружите предвзятость в своих данных, вы можете смягчить ее с помощью различных методов предварительной обработки. Эти методы часто заменяют текущие значения или метки значениями, которые приводят к более справедливой модели. Примерами алгоритмов смягчения смещения предварительной обработки являются повторное взвешивание, оптимизированная предварительная обработка, обучение справедливым представлениям и устранение несопоставимых воздействий.

Преобразование данных также может включать в себя четыре следующих действия:

1) фильтрация,

2) условное квантование временных рядов. Временные ряды часто содержат потоки с измерениями в разные временные метки. Можно

квантовать выборку из распределений исходных временных рядов на соответствующих квантованных временных шагах,

3) дискретизация. Непрерывные поля могут запутать модель. Например, дискретный набор диапазонов иногда работает лучше, чем непрерывные значения, особенно при меньших объемах данных и с более простыми моделями,

4) разметка данных. Способы разметки данных в основном похожи на способы сбора данных. Это внутренние ресурсы компаний, краудсорсинг, открытые и закрытие инструменты программной разметки.

Для различных задач машинного обучения задача разметки данных имеет существенные различия.

Задачи разметки данных:

– аннотирование изображений и видео,

– разметка текста для задач NLP,

– аннотирование аудио.

В настоящее время в основном используется автоматическая разметка данных. Например, для сегментации часто используется модель SAM [6]. Сравнение метрик качества данной модели представлено на рисунке 6.

Dataset	DSC		mIoU	
	Finetune Dec	Finetune Enc-Dec	Finetune Dec	Finetune Enc-Dec
Kvasir	0.882	0.902	0.835	0.863
CVC-ClinicDB	0.890	0.921	0.837	0.877
CVC-ColonDB	0.891	0.894	0.835	0.843
CVC-300	0.924	0.924	0.882	0.883
ETIS	0.885	0.903	0.834	0.852

Рисунок 6 – Сравнение метрик DSC и mIoU на нескольких датасетах

Вывод: в рамках данного раздела были достигнуты следующие результаты: Произведён исчерпывающий анализ источников данных, методов сбора данных. Выделены основные критерии выбора бора, подходящих к

разным проектам ML. Представлен обзор составных частей набора данных. Было выполнено сравнение способов сбора данных, было проведено исследование инструментов для гернерации данных, выделены основные инструменны для получения разных наборов данных.

3 Используемые инструменты

3.1 Выбор сред

Для анлиза и визуализации данных были использованы такие интсрументы:

– Python,

– Google Colab.

Python – популярный язык программирования, используемый для web-разработки, разработки программного обеспечения, анализе данных, машинном обучении.

В качестве альтернативы при выборе языка были рассмотрены такие языки как R, Julia, Scala и SQL.

Язык Python был выбран в связи с тем, что является самым популярным для анализа данных, имеет большое количество встроенных компонентов и библиотек, предназначенных для реализации, универсален и прост в использовании.

В качестве альтернативы при выборе языка были рассмотрены такие языки как R и Julia.

Google Colab – интерактивная среда, которая используется для работы с кодом на языке Python в браузере. Так как сервер является облачным, он позволяет использовать большие мощности для ускорения вычислений

В качестве альтернативы был рассмотрены такие инструменты как Jupyter, Databricks, Deepnote, Yandex DataSphere.

Google Colab в связи с универсальностью создания ноутбуков, понятным интерфейсом, возможностью полной визуализации. Минусами данной платформы является версионирование и небольшое время жизни окружения, но она является самой используемой для анализа данных.

3.2 Выбор моделей

Для обучения были выбраны следующие алгоритмы:

– логистическая регрессия,

– метод опорных векторов (SVM),

– случайный лес,

– метод K ближайших соседей,

– многослойный персептрон (MLP).

Логистическая регрессия была выбрана для анализа, так как данный алгоритм показывает хорошие результаты на бинарной классификации. Для реализации многоклассовой классификации были выбраны ClassifierChain и MultiOutputClassifier, который может быть добавлен к любому классификатору. Они удобны и показывают хорошие результаты.

Метод SVM может быть хорош при нелинейном распределении данных и в задачах с большим количеством признаков.

Случайный лес хорошо работает на больших обьемах данных, устойчив к обучению и способено обрабатывать категориальные переменные.

Метод k ближайших соседей эффективен в задачах с разреженными данными.

MLP представляет собой искусственную нейронную сеть. Алгоритм также способен обрабатывать нелинейные данные и эффективно работает с большим количеством признаков.

3. 3 Выбор библиотек

Для создания моделей были исользованы следующие библиотеки Python:

– pandas,

– urbangrammar_graphics,

– matplotlib,

– statistics,

– statistics,

– scipy,

– sklearn.

Библиотека pandas была использована для работы со структурами данных. Matplotlib.pyplot – для визуализации плотов. Statistics и scipy.stats –

для работы со статистическими функциями. Seaborn – для создание интерактивных статистических графиков. Библиотека sklearn – использовалось для обработки и нормализации данных, и построения моделей машинного обучени, а также метрик.

Вывод: в данном разделе были выбраны среды и язык для разработки, выделены преимущества используемых средств, а также описано использование основных библиотек.

4 Эксперименты и результаты исследования

4.1 Постановка задачи

Одним из основополагающих критериев для сбора данных является решаемая проблема. При корректно сформулированной проблеме возможно выделение основных аспектов, необходимых для обучения модели.

Для примера возьмем проблему низкой успеваемости студентов, которая приводит к отчислению, предлагается использовать.

Решение включает в себя три составляющие:

1) прогнозирование успеваемости студентов на основе исторических данных об успеваемости, посещаемости и так далее,

2) классификацию студентов на категории «успевающих» и «неуспевающих» по результатам прогнозирования,

3) использование этих результатов для оказания индивидуальной поддержки студентам из группы риска.

4.2 Выбор способа сбора данных

Какие для этого нужны данные? На первый взгляд это оценки в предыдущем семестре/семестрах, а также вид аттестации. Но будут ли эти данные информативными? И каково будет их количество.

Для набора получения подобного набора можно выделить несколько способов сбора данных:

– прямая выгрузка из БД ЭИОС,

– проведение опроса среди обучающихся по их прошлому семестру,

– технологии веб-скраппинга,

– при возможности подключения через API, сбор данных через API.

Рассмотрим плюсы и минусы способов сбора данных.

Для прямой выгрузки данных из БД ЭИОС положительными можно выделить следующие критерии:

– достоверность данных, хранящихся в системах электронно-образовательных сред, обладают высокой достоверностью, ввиду их постоянной проверки,

– получить выгрузку из базы данных довольно просто и быстро, воспользовавшись запросом к БД или специальным инструментом для выгрузки данных (при наличии).

К минусам прямой выгрузки данных из БД ЭИОС можно отнести:

– наличие пустых данных, так как данные в информационных системах могут быть не до конца заполнены. Например, если не все преподаватели внесли записи в электронный профиль студента,

– недостаточное количество данных, так как не всегда есть возможность сразу оценить количество данных для машинного обучения. Количество данных при выгрузке может сильно отличаться от предполагаемого, содержать пропуски, а также дублирующие записи,

– наличие персональных данных. В БД информационных сред хранится огромное количество персональных данных. Доступ к ним может быть ограничен правами пользователя, как и доступ к БД в целом. Возможен доступ только на чтение данных.

Также при выгрузке данных из БД существует проблема версионности. В зависимости от того несколько актуальны данные о студенте в электронной образовательной системе можно говорить о возможности выгрузок. Для большинства организаций, имеющих дело с большими объемами разнообразных наборов данных, общим является то, что многие наборы данных многократно анализируются несколько различными способами, а промежуточные результаты сохраняются для дальнейшего использования. Для хранения и повторного использования данных учитываются стоимость хранения, аренды ресурсов для хранения и восстановления данных [4].

При отсутствии доступа к БД сред, но наличии доступа к веб интерфейсу можно воспользоваться инструментами вэб скраппинга. В данный момент существует огромное количество утилит для осуществления скраппинга веб страниц. Например, можно воспользоваться одним из готовых решений с сайта https://www.pythonscraping.com/code/[7].

Следующий способ – это проведение опросов среди студентов об их жизни и текущей успеваемости.

Данный способ также имеет место быть. В современном мире многие компании собирают данные для обучения моделей с помощью опросов.

Но так же как и других способов у данного есть свои плюсы и минусы.

К плюсам способа относится наличии возможности выяснения необходимой информацию у первоисточника.

Минусами при данном способе является:

– малая достоверность, так как в силу психологических особенностей человек склонен преувеличивать/преуменьшать свои достижения,

– низкая скорость, ведь для проведения опросов необходимо огромное количество времени. При выборе данного способа необходимо ориентироваться на временные рамки.

Можно также воспользоваться готовым набором данных. Но в открытом доступе в основном находятся данные об успеваемости студентов других стран, а при анализе данных необходимо учитывать демографические особенности.

Изначально необходимо определить на основе каких данных будет происходить прогнозирование.

Данные могут быть различны. Выгрузки с образовательных порталов, отзывы преподавателей, активность в социальных сетях, дополнительные занятия студентов.

Для каждого из этих вариантов необходимо использование своего метода сбора данных.

Большим удобством будет обладать сбор данных о студентах в электронной образовательной среде. От преподавателей может понадобиться выставить рейтинг и своевременно заполнить все оценки, от студентов пройти опросы для выявления внеучебных занятий, а также полностью заполнить карточку для осуществления полноты данных.

В текущих условиях для того, чтобы собрать все данные на одной платформе и поддерживать их версионность потребуются большие профессиональные и финансовые ресурсы для доработки системы.

При условии наличия большего объёма данных можно не волноваться об их избыточности. Ведь одним из этапов подготовки данных к машинному обучению является их очистка.

В нашем случае данные не избыточны, их нет совсем, поэтому на первом этапе необходимо определиться с основными критериями, которые будут использоваться для прогнозирования.

Для составления признаков для набора данных необходимо изучение предметной области. Необходимо выделить объект и предмет обучения. В нашем случае объектом является информация о студенте, а предметом прогнозирование успеваемости.

Также при прогнозировании необходимо определение целевой переменной. То есть в каком формате будет измеряться успеваемость. В конкретной оценке по предмету, среднем балле на конец следующего курса, в процентах пропущенных занятий или в общей успешности студента на момент выпуска.

Для выявления признаков необходимо разобрать факторы, влияющие на объект обучения.

Так как студент это в первую очередь человек, необходимо учесть физиологические качества, влияющие на успеваемость. Такие как пол, возраст и другие.

Также необходим сбор факторов окружающей среды. Например, данные о семье студента, какое образование имеют родители. Согласно исследованиям, существует прямая взаимосвязь между образованием родителей и образованием детей. Также не будет лишней информация о семейном положении родителей и количестве детей в семье. Стабильная структура семьи может создать благоприятные условия для обучения ребенка. Изменения в семейном положении могут повлиять на успеваемость ребенка

Затем необходимо указать признаки, напрямую коррелирующие с учебным процессом. Такие как факультет студента, основа обучения, средний балл за предыдущий семестр (возможно использование баллов за каждый предмет в отдельности. Но при наличии большого числа предметов данная концепция приведет к избыточности данных). Необходимы данные о посещаемости. Можно собрать данные о посещаемости в виде количества пропущенных или процента посещенных/пропущенных занятий.

Также необходима оценка личностных особенностей человека: наличие дополнительных занятий, наличие работы совместно с обучением, количество времени, выделяемого не учебу.

В данном случае невозможно исключить ни один из факторов, влияющих на успеваемость. При его исключении возможно недостоверное моделирование. А какие факторы являются основополагающими в том или ином случае определяется при анализе данных.

Возьмем несколько наборов данных и проведем первичный анализ, построив корреляционные матрицы.

Для сравнения возьмем два набора данных, содержащих разные признаки и полученных разными способами:

– набор данных получен с помощью выгрузки из ЭИОС,

– другой набор данных.

4.3 Оценка качества наборов данных

4.3.1 Оценка качества первого набора данных

Модель данных первого небора представлена на рисунке 3.

Таблица 3 – Модель данных первого набора

Атрибут	Описание атрибута
ID учащегося	Содержит ID учащегося в системе
Семестр обучения	Семестр обучения, к которому относятся оценки
Курс	Курс, на котором обучается студент
Название дисциплин	Содержит наименование дисциплины

Продолжение таблицы 3

Атрибут	Описание атрибута
Оценка	Оценка по дисциплине в определенном курсе и семестре
Дата выставления по факту	Содержит дату выставления оценки в формате ДД.ММ.ГГГГ
Дата выставления по ведомости	Содержит дату выставления оценки в формате ДД.ММ.ГГГГ

Представим на рисунке 7 визуализацию первого набора данных.

	ID учащегося	Название дисциплины	Тип отчетности	Курс	Семестр	Оценка	Дата выставления (по ведомости)	Дата выставления (по факту выставления)
0	2128152	Интеллектуальные системы	Экзамен	1	1	отл.(5)	27.01.2022	27.01.2022
1	2128152	Информационное общество и проблемы прикладной ...	Зачет	1	1	зачет	29.12.2021	19.01.2022
2	2128152	Информационное общество и проблемы прикладной ...	Экзамен	1	1	отл.(5)	19.01.2022	19.01.2022
3	2128152	Информационное общество и проблемы прикладной ...	Курсовой проект	1	1	отл.(5)	29.12.2021	19.01.2022
4	2128152	Математическое моделирование	Зачет	1	1	зачет	10.01.2022	01.02.2022
...	...	...	...	...	...	...	...	...
68977	517727579	Иностранный язык	Зачет	1	1	зачет	03.04.2023	03.04.2023
68978	517727579	История (история России, всеобщая история)	Экзамен	1	1	хор.(4)	03.04.2023	03.04.2023
68979	517727579	Математика	Экзамен	1	1	хор.(4)	03.04.2023	03.04.2023
68980	517727579	Русский язык и деловые коммуникации	Зачет	1	1	зачет	03.04.2023	03.04.2023
68981	517727579	Физическая культура и спорт	Зачет	1	1	зачет	03.04.2023	03.04.2023

Рисунок 7 – Визуализация набора данных №1

Также данный набор данных был представлен на рускунке 8 в другом формате.

Данный формат набор принял после ряда выполненных над ним преобразований.

```
<ipython-input-4-6a704b2f75d9>:1: DtypeWarning: Columns (3,4,5,6,7,8,9,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28,32,37,40,41,43,44,
  df=pd.read_csv('/content/Weblabs/output.csv',encoding='cp1251', sep=';')
```

	ID учащегося	Семестр	Зачет	Информационное общество и проблемы прикладной информатики	Математическое моделирование	Художественные основы интерфейсов информационных систем	Английский язык	Методология и технология проектирования информационных систем	Планирование и организация научных исследований	Программные среды компьютерного дизайна	...	Вычислительные системы и параллельная обработка данных.1	Основы WEI программирования.:
0	2128152.0	1	NaN	зачет	зачет	отл.(5)	NaN	NaN	NaN	NaN	...	NaN	NaN
1	NaN	2	NaN	NaN	NaN	NaN	зачет	зачет	зачет	зачет	...	NaN	NaN
2	NaN	3	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	...	NaN	NaN
3	2361510.0	1	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	...	NaN	NaN
4	NaN	2	NaN	NaN	NaN	NaN	зачет	NaN	зачет	NaN	...	NaN	NaN
...	...	...	...	...	...	...	...	...	...	...	...	...	...
7356	517726888.0	3	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	...	NaN	NaN
7357	517726917.0	1	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	...	NaN	NaN
7358	517727374.0	1	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	...	NaN	NaN
7359	517727431.0	1	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	...	NaN	NaN
7360	517727579.0	1	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	...	NaN	NaN

7361 rows × 1081 columns

Рисунок 8 – Набор данных после преобразований №1

Если увидеть набор данных, где данные распределены по предметам можно подумать, что данный набор безнадежен. И при проведении статистических методов на данном наборе будут отрицательные результаты.

Для первого набора данных статистические показатели корректно посчитать не удалось в связи с большим количеством пропущенных значений.

В качестве статистических показателей взяты: медиана, мода и среднее значения для каждого показателя. Результаты представлены на рисунке 9.

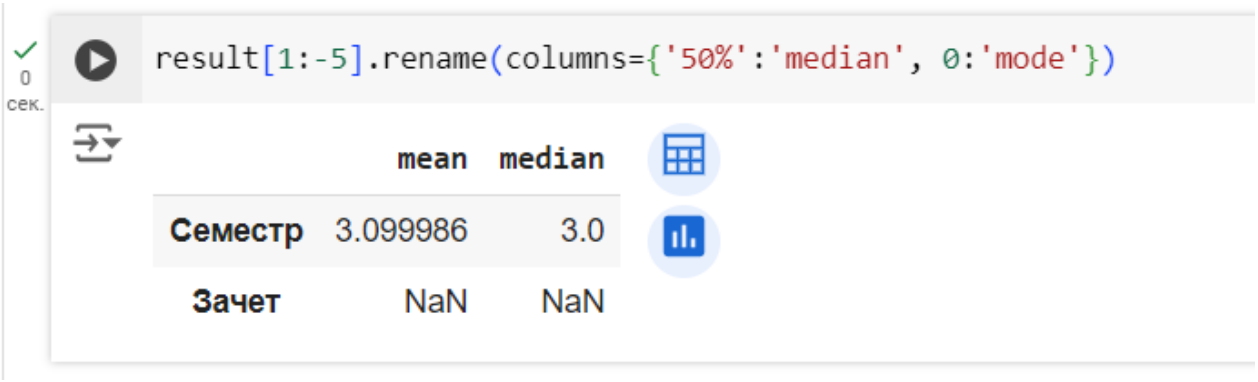


Рисунок 9 – Статистические показатели набора №1

Отсутствие возможности вывода статистических показателей приводит к невозможности работы с данными, так как определение распределения данных затруднено.

Возможно восстановление пропущенных данных в первом наборе. Посмотрим на процент пропусков, представленный на рисунке 10.

```
for col in df.columns:
 pct_missing = np.mean(df[col].isnull())
 k=round(pct_missing*100)
 print('{} - {}%'.format(col, k))
```

```
ID учащегося - 71%
Семестр - 0%
Зачет - 100%
Информационное общество и проблемы прикладной информатики - 100%
Математическое моделирование - 99%
Художественные основы интерфейсов информационных систем - 100%
Английский язык - 98%
Методология и технология проектирования информационных систем - 100%
Планирование и организация научных исследований - 99%
Программные среды компьютерного дизайна - 100%
Производственная практика (научно-исследовательская работа) - 94%
Разработка фирменного стиля - 100%
Учебная  (технологическая (проектно-технологическая)) практика - 100%
Философия и методология научных исследований - 99%
Оценка и обеспечение информационной безопасности - 100%
Управление разработкой программных средств компьютерного дизайна - 100%
Вычислительные системы - 100%
Мировые тенденции развития вычислительной техники и информационных технологий - 99%
Твердотельное моделирование  в системах автоматизированного проектирования - 100%
Учебная (технологическая (проектно-технологическая)) практика - 99%
Цифровая обработка сигналов - 100%
Мультимедийные технологии в проектировании информационных ресурсов и интерфейсов - 100%
Современные технологии программирования - 100%
Технические и программные средства информационных систем - 100%
Проектирование устройств цифровой обработки сигналов - 100%
Технология и языки параллельного программирования - 100%
Управление параллельными процессами и ресурсами в вычислительных системах - 100%
Непрерывные и дискретные математические модели - 100%
Инженерная графика - 98%
Иностранный язык (английский) - 95%
Современные информационные технологии - 83%
Физическая культура и спорт - 82%
Конструирование компонентов и узлов электронных средств - 100%
История радиоэлектронных средств - 98%
Введение в информационные технологии проектирования  электронных средств - 99%
Правоведение - 88%
```

Рисунок 10 – Процент пропусков первого набора данных

Затем были удалены дубликаты данных и незначащий признак. Результат представлен на рисунке 11.

```
[17] df = df.drop('ID учащегося', axis=1).drop_duplicates()
```

```
df.drop_duplicates()
```

	Семестр	Зачет	Информационное общество и проблемы прикладной информатики	Математическое моделирование	Художественные основы интерфейсов информационных систем	Английский язык	Методология и технология проектирования информационных систем	Планирование и организация научных исследований	Программные среды компьютерного дизайна	Производственная практика (научно-исследовательская работа)	...	Вычислительные системы и параллельная обработка данных.1	Основы WEB программирования.1	Структуры и алгоритмы компьютерной обработки данных.2	Информационное обеспечение автоматизированных систем.1	Компьютерное моделирование.2	Технология разработки программного обеспечения.2
0	1	NaN	зачет	зачет	отл.(5)	NaN	NaN	NaN	NaN	NaN	...	NaN	NaN	NaN	NaN	NaN	NaN
1	2	NaN	NaN	NaN	NaN	зачет	зачет	зачет	зачет	отл.(5)	...	NaN	NaN	NaN	NaN	NaN	NaN
2	3	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	отл.(5)	...	NaN	NaN	NaN	NaN	NaN	NaN
3	1	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	отл.(5)	...	NaN	NaN	NaN	NaN	NaN	NaN
4	2	NaN	NaN	NaN	NaN	зачет	NaN	зачет	NaN	отл.(5)	...	NaN	NaN	NaN	NaN	NaN	NaN
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
7356	3	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	...	NaN	NaN	NaN	NaN	NaN	NaN
7357	1	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	отл.(5)	...	NaN	NaN	NaN	NaN	NaN	NaN
7358	1	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	...	NaN	NaN	NaN	NaN	NaN	NaN
7359	1	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	...	NaN	NaN	NaN	NaN	NaN	NaN
7360	1	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	...	NaN	NaN	NaN	NaN	NaN	NaN

4152 rows × 1080 columns

Рисунок 11 – Датасет после модификации

Количество неопределенных значений датасета приближено практически по всем атрибутам к 100%. При дополнении значений набора данных можно использовать замену недостающих значений на моду или

среднее значение. Или использовать более выгодное восстановление с помощью прогнозирования признаков, импутации данных с помощью K ближайших соседей. Но при при вычислении из-за малого количества данных значения нельзя использовать для дополнения набора ввиду их недостоверности.

Генерация синтетических данных на основе имеющихся также невозможно из-за малого количества исходных данных.

При получении такого набора данных вернуть их к первоначальному варианту довольно сложно и не имеет смысла, так как обработать такие данные будет проблематично и займет не малое количество времени.

Если же исходный набор данных близок к изображенному на рисунке 6 возможно обработать данные таким образом, чтобы даже они были пригодны для ML- моделей.

Первым этапом необходимо определить какую полезную информацию можно извлечь из этих данных для оценки успеваемости студента. Выделены несколько аспектов:

– процент отличных оценок,

– процент зачетов,

– процент неудач студента,

– процент неявок на экзамены,

– сумма оценок за семестр.

Процентое соотношение взято как самый простой способ получения оценки показателей успешности студента, возможно использование других методов оценки. Процентное соотношение убирает влияние времени выгрузки данных. Например, если оставить количество отличчных, хороших, удовлетворительных оценок, то оказать влияние наа данные может время сбор. Так набор, собранный в середине года и в конце будет разниться из-за отстутствие у части студентов оценок за весь период, хотя у студентов курсов выше это значение будет полным. Тогде необходимо будет выделить семестр, курс и другие параметры, для минимизации влияния данного фактора.

Для получения показателей необходимо модифицировать данные.

Приведем набор данных к виду, представленному на рисунке 12, в котором не используются отдельные предметы, так как у каждой специальности и каждой ФГОС эти предметы имеют разные названия, что и привело одну из версий набора данных огромному количеству неопределенных значений.

ID учащегося	Семестр	Оценка	зачет	н/я	не атт.	не доп.	незачет	неуд.(2)	отл.(5)	уд.(3)	хор.(4)	Количество оценок
2128152	1	2	0	0	0	0	0	6	0	0		8
	2	6	0	0	0	0	0	6	0	0		12
	3	2	0	0	0	0	0	6	0	0		8
2361510	1	0	0	0	0	0	0	7	0	0		7
	2	5	0	0	0	0	0	5	0	0		10
...	...	...	...	...	...	...	...	...	...	...	...	...
517726888	3	2	2	0	0	3	0	0	1	1		9
517726917	1	1	0	0	0	0	0	3	2	0		6
517727374	1	1	3	0	0	0	0	2	0	1		7
517727431	1	3	0	0	0	0	0	1	0	1		5
517727579	1	3	0	0	0	0	0	0	0	2		5

Рисунок 12 – Модифицированный набор данных

Затем посчитаем для набора процентное соотношение количества оценок для определения успешности студента. Результаты представлены не рисунке 13.

df

ID учащегося	Семестр	Оценка	зачет	н/я	не атт.	не доп.	незачет	неуд.(2)	отл.(5)	уд.(3)	хор.(4)	Количество оценок	Процент отличных оценок	Процент неудач	зачетов	Процент нелвок	Процент хороших оценок	Процент удовлетворительных
2128152	1	2	0	0	0	0	0	0	6	0	0	8	75.000000	0.0	25.000000	0.000000	0.000000	0.000000
	2	6	0	0	0	0	0	0	6	0	0	12	50.000000	0.0	50.000000	0.000000	0.000000	0.000000
	3	2	0	0	0	0	0	0	6	0	0	8	75.000000	0.0	25.000000	0.000000	0.000000	0.000000
2361510	1	0	0	0	0	0	0	0	7	0	0	7	100.000000	0.0	0.000000	0.000000	0.000000	0.000000
	2	5	0	0	0	0	0	0	5	0	0	10	50.000000	0.0	50.000000	0.000000	0.000000	0.000000
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
517726888	3	2	2	0	0	3	0	0	1	1		9	0.000000	0.0	22.222222	22.222222	11.111111	11.111111
517726917	1	1	0	0	0	0	0	3	2	0		6	50.000000	0.0	16.666667	0.000000	0.000000	33.333333
517727374	1	1	3	0	0	0	0	2	0	1		7	28.571429	0.0	14.285714	42.857143	14.285714	0.000000
517727431	1	3	0	0	0	0	0	1	0	1		5	20.000000	0.0	60.000000	0.000000	20.000000	0.000000
517727579	1	3	0	0	0	0	0	0	0	2		5	0.000000	0.0	60.000000	0.000000	40.000000	0.000000

7328 rows × 16 columns

Рисунок 13 – Модифицированный набор данных

Удалим ненужные атрибуты данных из набора и кластеризуем на три класса с помощью метода K- ближайних соседей. Результаты представлены не рисунке 14.

```
df_clean.drop_duplicates()
```

ID учащегося	Семестр	Оценка	Количество оценок	Процент отличных оценок	Процент неудач	зачетов	Процент неявок	Процент удовлетворительных	Процент хороших оценок	Cluster
2128152	1	8	75.000000	0.000000	25.000000	0.000000	0.000000	0.000000	2	
	2	12	50.000000	0.000000	50.000000	0.000000	0.000000	0.000000	2	
2361510	1	7	100.000000	0.000000	0.000000	0.000000	0.000000	0.000000	2	
	2	10	50.000000	0.000000	50.000000	0.000000	0.000000	0.000000	2	
	3	6	100.000000	0.000000	0.000000	0.000000	0.000000	0.000000	2	
...	...	...	...	...	...	...	...	...	...	...
517726641	3	12	0.000000	66.666667	16.666667	0.000000	8.333333	0.000000	0	
517726917	1	6	50.000000	0.000000	16.666667	0.000000	33.333333	0.000000	1	
517727374	1	7	28.571429	0.000000	14.285714	42.857143	0.000000	14.285714	1	
517727431	1	5	20.000000	0.000000	60.000000	0.000000	0.000000	20.000000	1	
517727579	1	5	0.000000	0.000000	60.000000	0.000000	0.000000	40.000000	1	

1894 rows × 8 columns

Рисунок 14 – Модифицированный набор данных

Оценим качество кластеризации по визуализациям данных, представленной на рисунке 15. Набор данных был разлен на три и на четыре кластера. Визуализации разделения на три кластера выглядит лучше, чем разделение на четыре кластера, но межкластерное расстояние все равно мало.

Кластреры имеют довольно некачественный вид, что усложнить классификацию в дальнейшем.

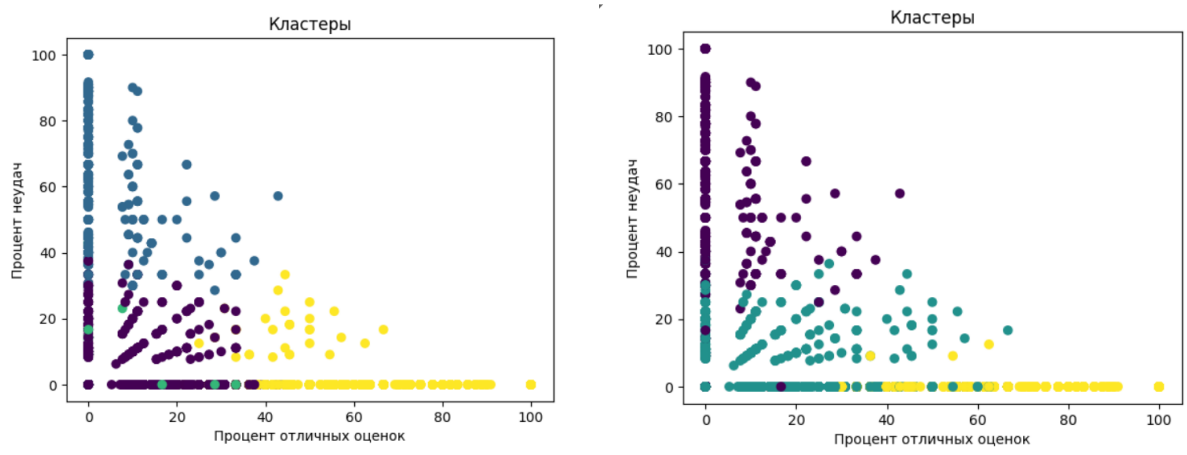


Рисунок 15 – Визуализация кластеризации

Из графиков видно, что качество кластеризации на среднем уровне, и лучшие результаты показало деление на три кластера.

При выводе описательной статистики мы получаем корректные данные, которые можно использовать для определения распределения. Результат представлен на рисунке 16.

```
[146] mean
```

```
Оценка
Количество оценок              9.312091
Процент отличных оценок       19.408551
Процент неудач                 5.626957
зачетов                       36.621079
Процент неявок                 5.936630
Процент удовлетворительных    15.766501
Процент хороших оценок        16.139091
Predict                        1.413483
Name: mean, dtype: float64
```

```
[147] median
```

```
Оценка
Количество оценок             10.000000
Процент отличных оценок       10.000000
Процент неудач                 0.000000
зачетов                       41.666667
Процент неявок                 0.000000
Процент удовлетворительных     0.000000
Процент хороших оценок        11.111111
Predict                        2.000000
Name: 50%, dtype: float64
```

Рисунок 16 – Статистические признака набора

Процент неопределенных данных также сокращен, что облегчит работу с набором при дальнейшей очистке. Результат представлен на рисунке 17.

```
for col in df.columns:
 pct_missing = np.mean(df[col].isnull())
 k=round(pct_missing*100)
 print('{} - {}%'.format(col, k))
```

```
зачет - 0%
н/я - 0%
не атт. - 0%
не доп. - 0%
незачет - 0%
неуд.(2) - 0%
отл.(5) - 0%
уд.(3) - 0%
хор.(4) - 0%
Сданные - 0%
Несданные - 0%
Количество оценок - 0%
Процент отличных оценок - 0%
Процент неудач - 0%
зачетов - 0%
Процент неявок - 0%
Процент удовлетворительных - 0%
Процент хороших оценок - 0%
```

Рисунок 17 – Пропущенные значения набора

Тепловая карта данного набора содержит небольшие корреляции данных. Но это нормально, так как целевая переменная была получена

исскуственным способом – кластеризацией. Тепловая карта представлена на рисунке 18.

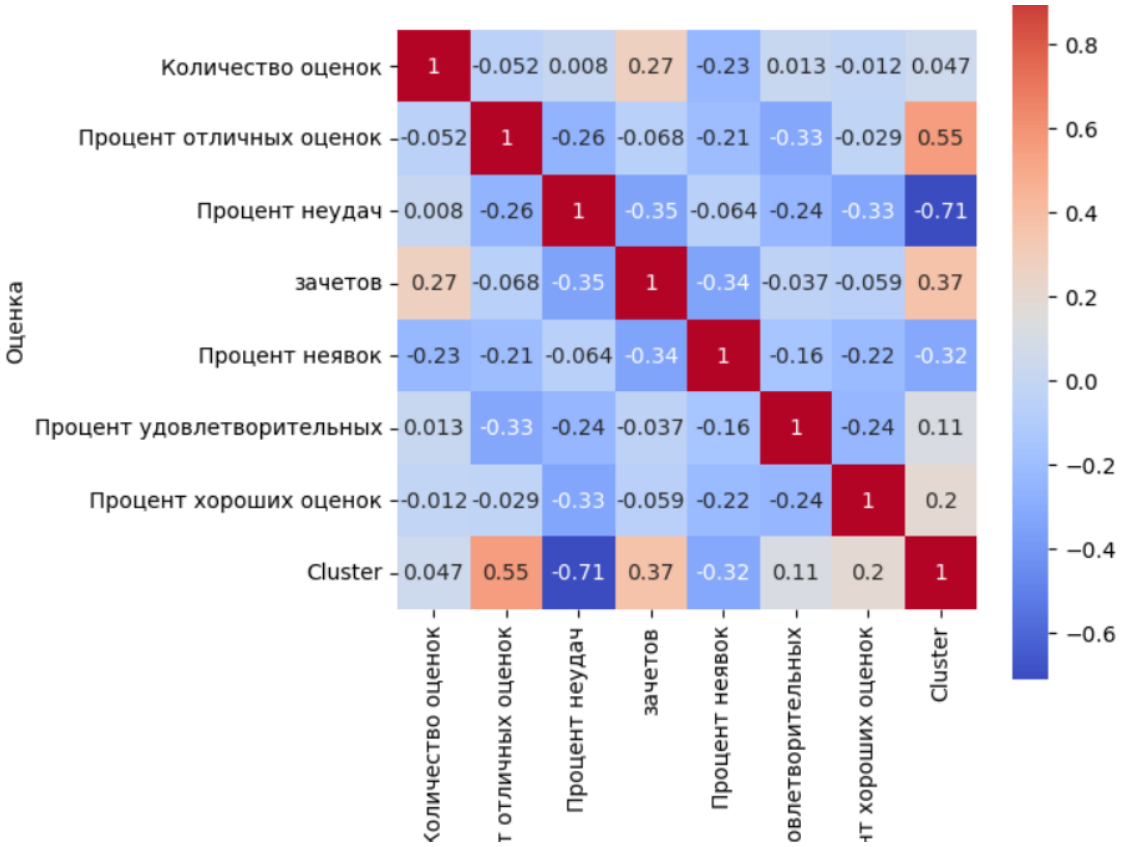


Рисунок 18 – Тепловая карта набора

На тепловой карте видно, что количество отличных оценок и зачетов являются положительно коррилирующими признаками по отношению к целевой переменной, а процент неудач отрицательно коррелирующим. В общем случае таки показатели тепловой карты обоснованны тем, что целевая переменная создавалась на основе кластеризации, где выделение целевых групп было необднозначно.

Вывод набору данных: Исходные данные пригодны для ML. Точность модели будет низкой, так как данные содержат малое количество показателей, влияющих на успеваемость студентов, и имеют корреляции. При дополнении данного набора данными о жизни студента: возраст, наличие работы, и других. Правильность предсказаний увеличится.

Альтернатива набора: В БД ЭИОС-Moodle существует возможность выгрузки большего количества данных. Но выгрузка данного набора ограничена законом о персональных данных и защите информации.

Необходимые данные: пол, возраст, баллы ЕГЭ/ВУЗовские испытания, зачисление по особым правим, данные об обучении за все курсы, курс обучения, дополнительные индивидуальные достижения. В ВУЗовской системе данные находятся в разных таблицах и получить их можно с помощью SQL запроса к БД.

4.3.2 Оценка качества второго набора данных

Модель данных второго набора представлена в таблице 4.

Таблица 4 – Модель данных второго набора

Атрибут	Описание атрибута
sex	Пол студента
age	Возраст студента
weekly_study_hours	Время, которое студент выделяет на внеурочную деятельность
fails	Количество долгов на момент сбора данных
job	Признак трудоустроенности студента. Имеет ли студент работу на момент сбора данных.
familyEDU	Признак, показывающий имеют ли высшее образование оба родителя в семье
activities	Признак, отвечающий за дополнительные активности студента
Free_time	Количество свободного времени
health	Рейтинг здоровья
child_in_family	Количество детей в семье
nationality	Нашиональность студента
placeofBirth	Место рождения
score	Средний балл по профильным предметам (учитывая отсутствие оценки, как отсутствие предмета)
predict	Целевая переменная

При просмотре данные второго набора выглядят намного лучше первого. Данные представлены на рисунке 19.

```
df=pd.read_csv('/content/DataForColab/NEW_DF2.csv',encoding='cp1251', sep=',')
df
```

	sex	job	child_in_family	activities	weekly_study_hours	Nationality	Score	Health	FamilyEDU	age	Predict	Fails	Free_time
0	M	no	3	no	27	Russian	2.93	1	0	19	F	6	4
1	F	no	2	no	47	Russian	2.98	4	1	21	F	15	19
2	F	no	9	yes	13	Russian	2.98	2	0	22	F	12	22
3	F	no	5	no	3	Russian	3.01	5	0	20	F	9	11
4	M	no	5	no	10	Russian	3.04	3	1	19	F	10	4
...	...	...	...	...	...	...	...	...	...	...	...	...	...
1992	M	no	2	yes	12	Russian	4.79	2	0	18	A	1	18
1993	M	no	3	no	46	Russian	4.80	2	1	17	A	1	8
1994	M	no	1	no	19	Russian	4.80	4	1	21	A	1	9
1995	M	no	2	no	30	Russian	4.80	4	0	17	A	1	12
1996	M	no	2	no	20	Russian	4.80	1	1	21	A	0	23

1997 rows × 13 columns

Рисунок 19 – Набор данных №2

Для первичного анализа данных необходимо выяснить распределение. Для этого воспользуемся методами описательной статистики.

Результаты вывода некоторых статистических показателей представлены на рисунке 20.

	mean	mean	0
sex	0.500250	0.500250	1.0
child_in_family	3.660991	3.660991	2.0
weekly_study_hours	17.759139	17.759139	3
Score	4.041112	4.041112	4.01
Health	2.958938	2.958938	1.0
FamilyEDU	0.504256	0.504256	1.0
age	19.475714	19.475714	17.0
Fails	2.648473	2.648473	0.0
Free_time	12.228343	12.228343	7.0
job	NaN	NaN	no
activities	NaN	NaN	no
Nationality	NaN	NaN	Russian
Predict	NaN	NaN	B

Рисунок 20 – Статистические показатели набора №2

Для большей части параметров, значения моды, медианы и среднего очень близки, что может говорить о том, что распределения этих параметров близко к нормальному.

Проверим процент неопределенных данных второго набора. Результат проверки представлен на рисунке 21.

```
for col in df.columns:
 pct_missing = np.mean(df[col].isnull())
 k=round(pct_missing*100)
 print('{} - {}%'.format(col, k))
```

```
sex - 0%
job - 0%
child_in_family - 0%
activities - 0%
weekly_study_hours - 0%
Nationality - 0%
Score - 0%
Health - 0%
FamilyEDU - 0%
age - 0%
Predict - 0%
Fails - 0%
Free_time - 0%
```

Рисунок 21 – Проверка неопределенных данных второго набора.

Из анализа видно, что данные не содержат пропусков. В дальнейшем с этим набором можно использовать методы анализа и подготовки данных. Проверим корреляцию данных с целевой переменной для определения возможности классификации.

Возьмем тепловую матрицу данного набора, представленную на рисунке 22.

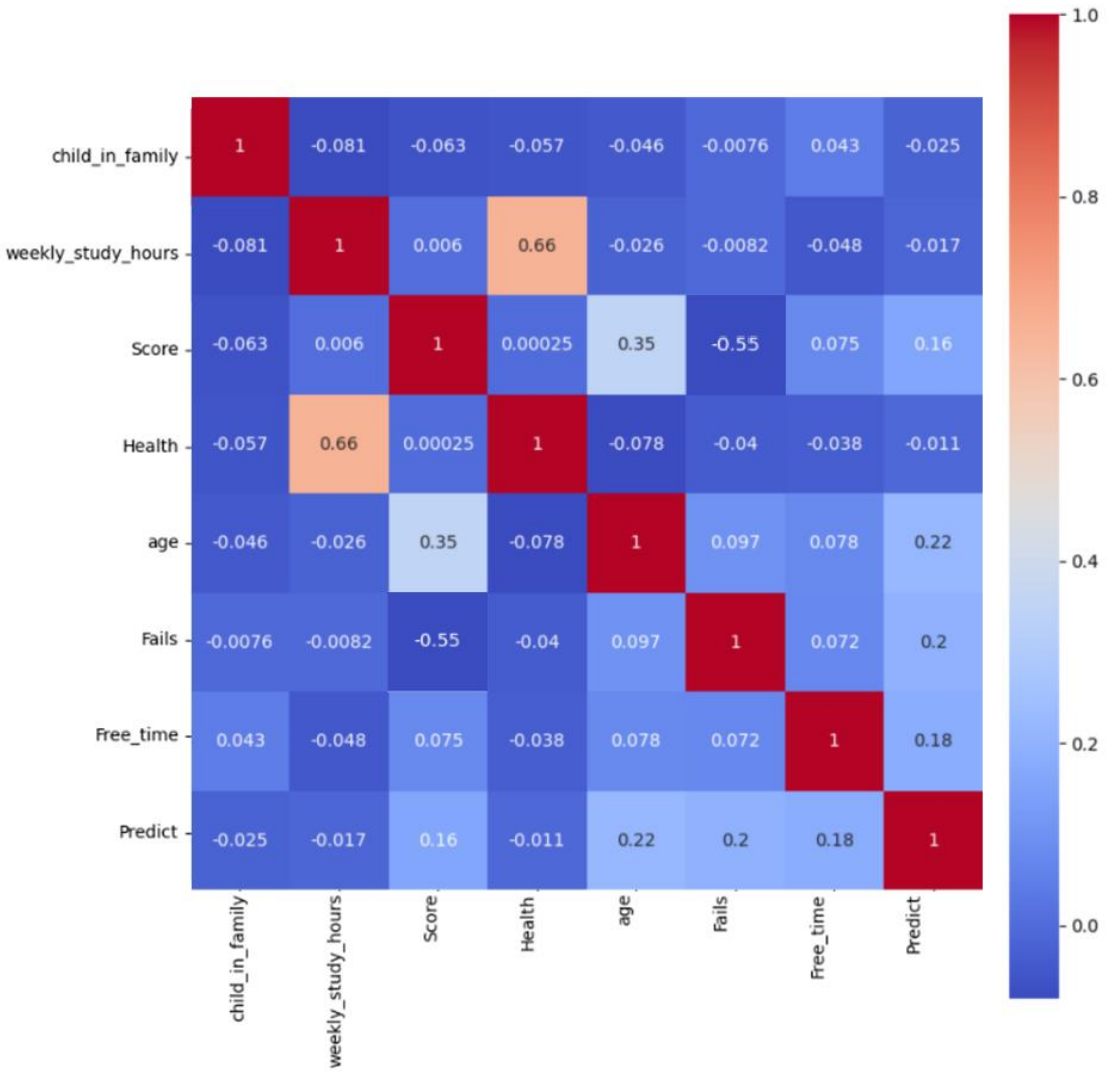


Рисунок 22 – Тепловая карта второго набора.

На тепловой корреляционной матрице видно, что атрибуты в малой степени коррелируют между собой, и с целевой переменной.

В общем случае такие данные уже можно использовать для обучения. Но при грамотной предварительной обработке возможно увеличение качества модели.

Вывод по набору: Данный набор можно использовать для обучения модели без дополнений. Но при наличии данных, которые также могут влиять на успешность студентов, их добавление в набор будет только повышать уровень прогнозироваания, если данные признаки окажутся значимыми.

Так как второй набор содержит намного больше признаков, которые могут оказаться значимыми, продолжим его обработку.

Посторим на «ящики с усами» данного набора. Результаты представлены на рисунке 23.

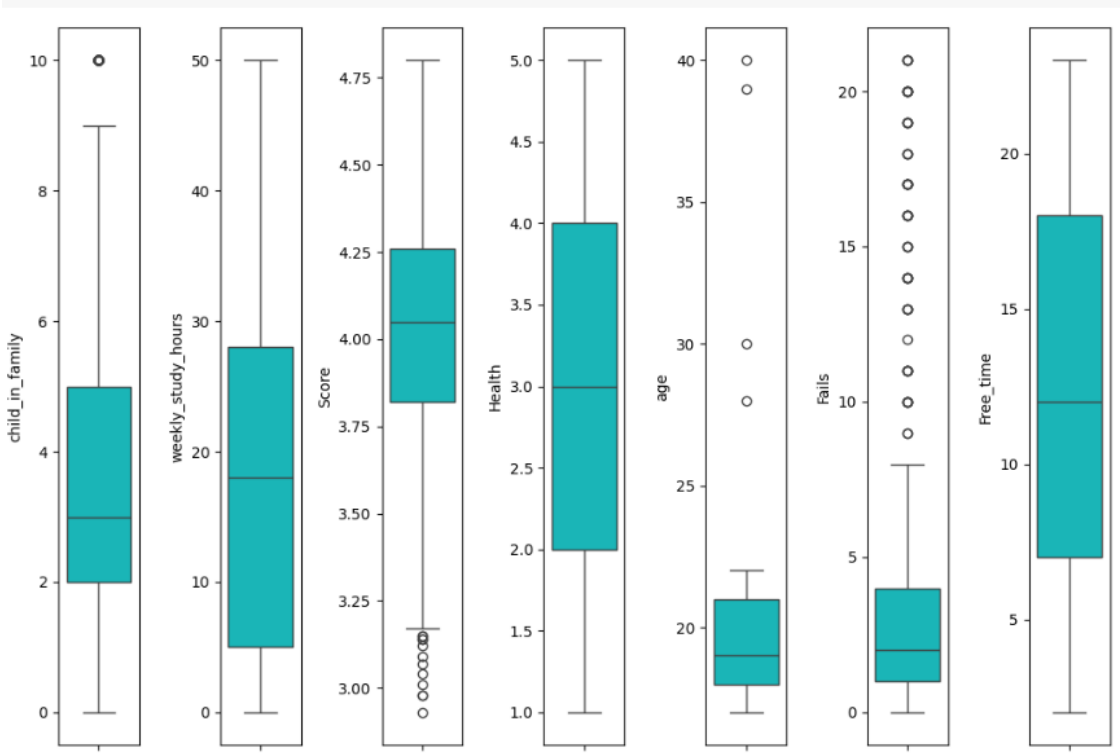


Рисунок 23 – Ящик с усами до очистки

Видим, что несколько атрибутов содержат выбросы, уберем их. Результаты представлен на рисунке 24.

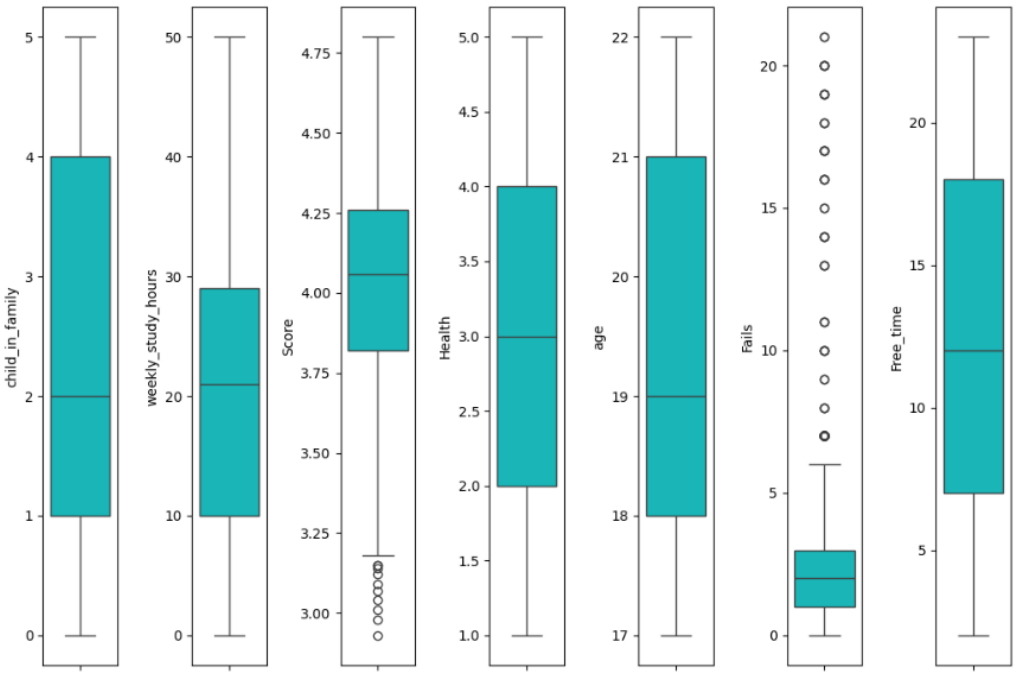


Рисунок 24 – Ящик с усами после очистки

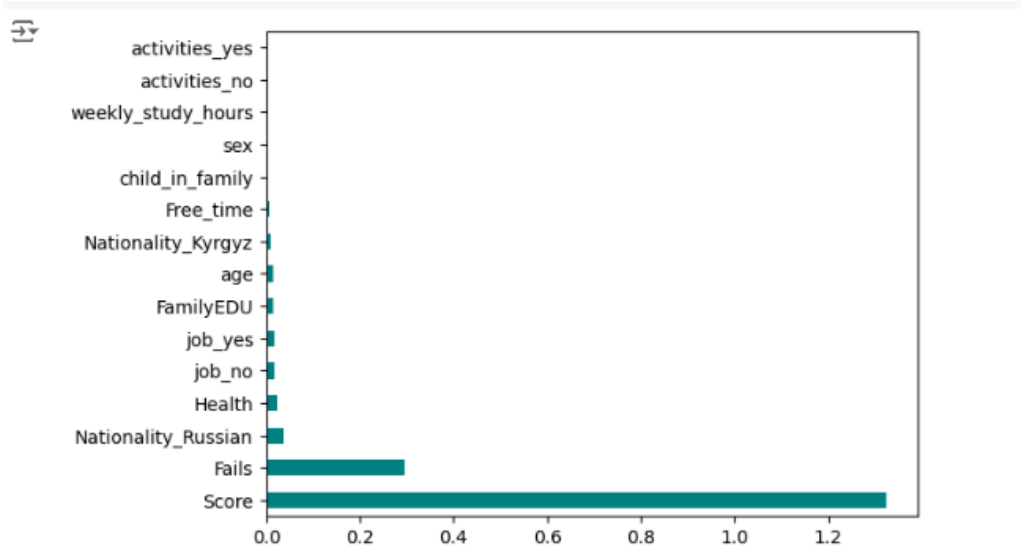
Очевидно, что рейтинг и количество незачетов не являются выбросами, так как данные показатели могут быть равны значениям, показанным на «ящиках с усами».

Построим графики для отбора значимых признаков.

Задача-минимум – сократить признаковое пространство без потери качества.

Задача-максимум – сократить признаковое пространство и при этом улучшить качество модели.

Для первого графика используем меру взаимной информации. График меры представлен на рисунке 25.




```
Score                1.324085
Fails                0.295422
Nationality_Russian  0.038252
Health               0.023730
job_no               0.017378
job_yes              0.016763
FamilyEDU            0.015724
age                  0.015528
Nationality_Kyrgyz   0.010258
Free_time            0.005921
child_in_family      0.004820
sex                  0.000000
weekly_study_hours   0.000000
activities_no        0.000000
activities_yes       0.000000
dtype: float64
```

Рисунок 25 – График отбора значимых признаков (взаимная информация)

Для второго возьмет критерий Хи-квадрат. Распределение признаков по данному критерию представлено на рисунке 26.

```
11          job_yes   2.183381
7           health    1.950173
6       ChildCount    1.672061
```

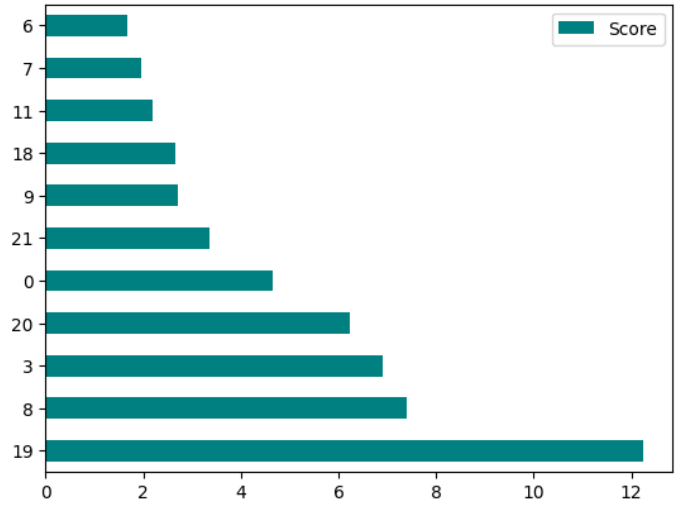


Рисунок 26 – График отбора значимых признаков(хи-квадрат)

Также отберем значимые признаки методом Lasso. Отбор с помощью даннрого метода представлен на рисунке 30.

```
[ ] from sklearn.linear_model import LogisticRegression
    from sklearn.feature_selection import SelectFromModel

    # Устанавливаем наш параметр регуляризации C=1
    logistic = LogisticRegression(C=1, penalty="l1", solver='liblinear', random_state=7).fit(X, y)
    # Где X, y - входные и выходные данные соответственно.
    model = SelectFromModel(logistic, prefit=True)

    X_new = model.transform(X)

    # выводим финальную оценку прогнозирования.
    selector_ind = model.get_support()
    df_rfe1 = X.iloc[:, selector_ind]
    print(df_rfe1.columns)
```

```
Index(['sex', 'child_in_family', 'weekly_study_hours', 'Score', 'Health',
       'FamilyEDU', 'age', 'Fails', 'Free_time', 'job_no', 'job_yes',
       'activities_no', 'activities_yes', 'Nationality_Russian'],
      dtype='object')
/usr/local/lib/python3.10/dist-packages/sklearn/base.py:432: UserWarning: X has feature names, but SelectFromModel was fitted without feature names
  warnings.warn(
```

Рисунок 27 – Значимые признаки

Проанализируем полученные данные и выберем значитые признаки для постоения каждой из моделей на основе графиков.

В данном разделе для кодирования категориальных переменных были использованы:

– Label Encoding,

– One-Hot Encoding.

4.4 Балансировка классов

Определим дисбаланс классов. Некоторые из выбранных моделей теоритически слабо чуствительны к дисбалансу из-за математического

аппарата, используемого в модели и распределения данных. Баланс набора представлен на рисунке 28.

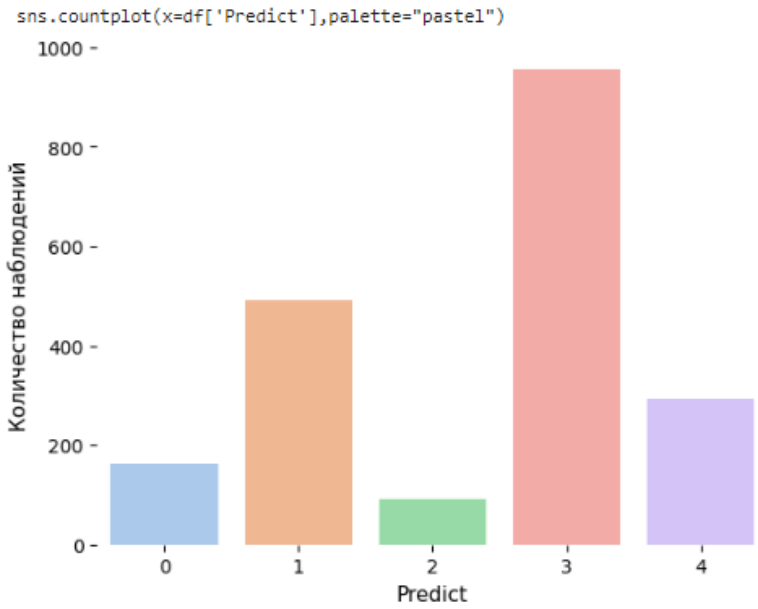


Рисунок 28 – Баланс набора

Очевидно, что дин из классов сильно преобладает над другими. После оценки методов балансировки был выбран для балансировки SMOTE данный алгоритм генерирует синтетические данные минорного класса. А затем испотзован простой андерсемплинг. Результат показан на рисунке 29.

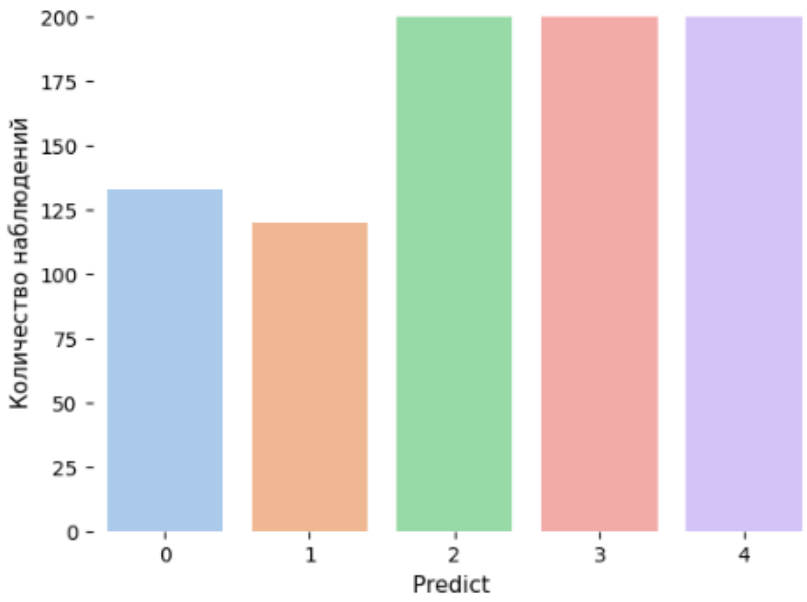


Рисунок 29 – Баланс набора

4.5 Обучение моделей

Обучение проводилось 5 методами обучения. Далее опишем обучение каждым методом для первого набора данных. Настройки параметров моделей не изменялись в зависимости от набора. Разделение на тестовую и тренировочную выборку также не корретировалось.

Для метода k-ближайших соседей определяем оптимальнык k параметр. Для каждоко k-параметра определяем точность и по данным показателям строим график. Определение оптимального k-параметра представленно на рисунке 30.

```
from sklearn.metrics import accuracy_score
acc = {}
for k in range(3, 30, 2):
    knn = KNeighborsClassifier(n_neighbors=k)
    knn.fit(X_train, y_train)
    y_pred = knn.predict(X_test)
    acc[k] = accuracy_score(y_test, y_pred)

plt.plot(range(3,30,2), acc.values())
plt.xlabel('K')
plt.ylabel('Accuracy')
plt.show()
```

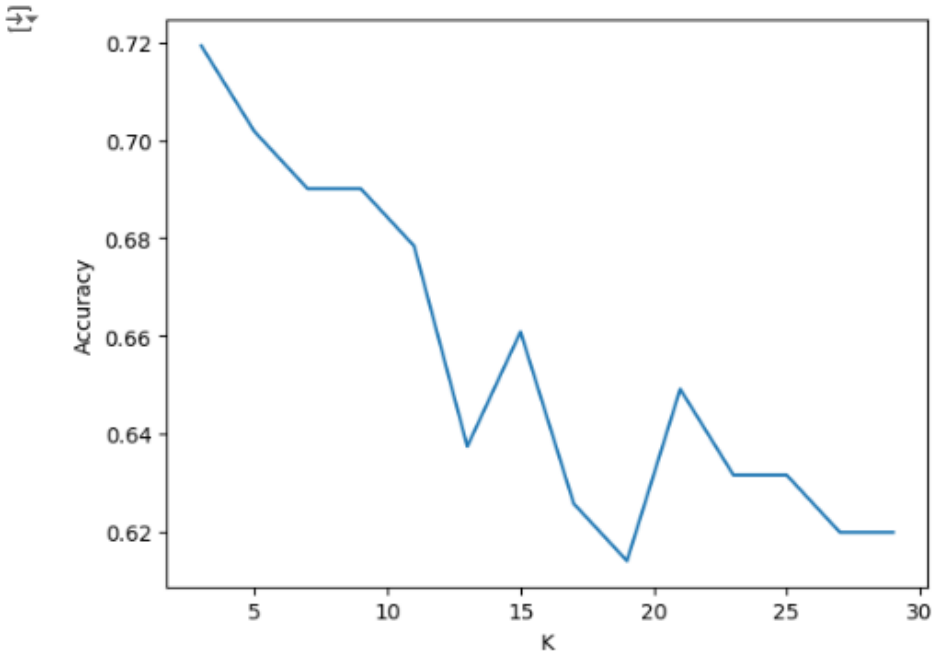


Рисунок 30 – Выбор параметра для обучения

Построим и обучим модель с оптимальным параметром. Результат представлен на рисунке 31.

```
[ ] knn = KNeighborsClassifier(n_neighbors=3)
    knn.fit(X_train, y_train)
    y_pred = knn.predict(X_test)
    print(accuracy_score(y_test, y_pred))
    print(classification_report(y_test, y_pred))
```

```
0.7192982456140351
              precision    recall  f1-score   support

           0       0.79      0.82      0.81        28
           1       0.38      0.38      0.38        21
           2       0.91      0.93      0.92        46
           3       0.63      0.75      0.68        36
           4       0.71      0.55      0.62        40

    accuracy                           0.72       171
   macro avg       0.69      0.69      0.68       171
weighted avg       0.72      0.72      0.72       171
```

Рисунок 31 – Обучение и точность модели

Далее строим модель логистической регрессии. Результаты показаны на рисунке 32.

```
from sklearn.multioutput import ClassifierChain
from sklearn.linear_model import LogisticRegression
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import label_binarize

y = label_binarize(y, classes=[1, 4, 2, 0, 3])
n_classes = y.shape[1]
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.
base_lr = LogisticRegression(solver='lbfgs', random_state=0)
chain = ClassifierChain(base_lr, order='random', random_state=0)
y_score = chain.fit(X_train, y_train).predict(X_test)
accuracy = accuracy_score(y_test, y_score)
print(f'Accuracy: {accuracy}')
print(classification_report(y_test, y_score))
```

```
Accuracy: 0.8056206088992974
```

Рисунок 32 – Обучение и точность модели

Использован метод для много классовой классификации ClassifierChain, так как модель линейной регрессии является бинарным алгоритмом. Использование алгоритма MultiOutputClassifier также дает неплохие результаты, что показано на рисунке 33. Результаты модели берутся по наивысшему показателю метрик качества.

```
[ ]  y = label_binarize(y, classes=[0, 1, 2, 3, 4])
     n_classes = y.shape[1]
     X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.5, random_state=0)

[ ]  from sklearn.multioutput import ClassifierChain

     base_lr = LogisticRegression(solver='lbfgs', random_state=0)
     chain = ClassifierChain(base_lr, order='random', random_state=0)
     pred=chain.fit(X_train, y_train).predict(X_test)

[ ]  from sklearn.multioutput import MultiOutputClassifier
     classifier = MultiOutputClassifier(LogisticRegression()).fit(X_train, y_train)
     pred=classifier.predict(X_test)

     #точность
     from sklearn.metrics import accuracy_score
     acc = accuracy_score(y_test.argmax(axis=1), pred.argmax(axis=1))
     print(f"Точность: {round(acc,3)*100}%")

     Точность: 82.0%
```

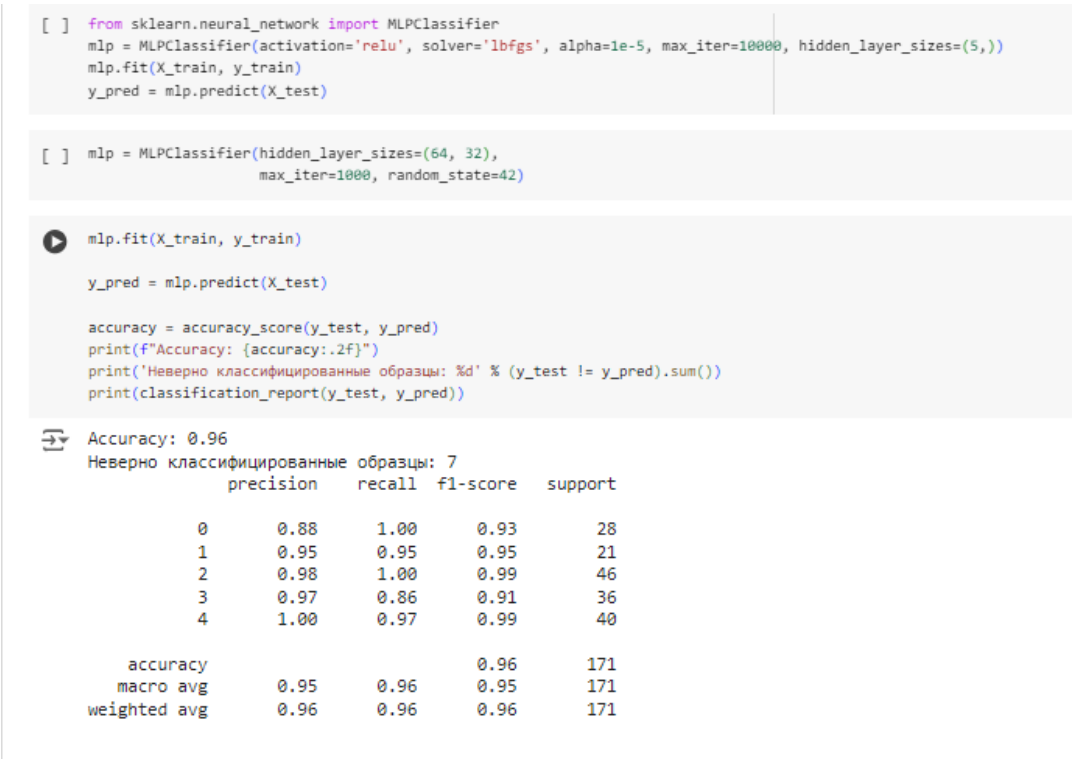
Рисунок 33 – Обучение и точность модели

Затем строим классификатор SVC, предварительно снова определяя X и y, так как для логистической регрессии проводилась бинаризация. Результат представлен на рисунке 34.

```
[ ]  model = SVC()
     model.fit(X_train, y_train)

     y_pred = model.predict(X_test)

     accuracy = accuracy_score(y_test, y_pred)
     print(f'Accuracy: {accuracy}')
     print(classification_report(y_test, y_pred))

     Accuracy: 0.8771929824561403
                   precision    recall  f1-score   support

                0       0.82      0.96      0.89        28
                1       0.75      0.57      0.65        21
                2       0.98      1.00      0.99        46
                3       0.90      0.78      0.84        36
                4       0.84      0.93      0.88        40

         accuracy                           0.88       171
        macro avg       0.86      0.85      0.85       171
     weighted avg       0.88      0.88      0.87       171
```

Рисунок 34 – Обучение и точность модели

Аналогично строим модели рандомного леса и многослойного персептрона. Результат представлен на рисунке 35.



```
[ ] from sklearn.neural_network import MLPClassifier
    mlp = MLPClassifier(activation='relu', solver='lbfgs', alpha=1e-5, max_iter=10000, hidden_layer_sizes=(5,))
    mlp.fit(X_train, y_train)
    y_pred = mlp.predict(X_test)

[ ] mlp = MLPClassifier(hidden_layer_sizes=(64, 32),
                        max_iter=1000, random_state=42)

    mlp.fit(X_train, y_train)

    y_pred = mlp.predict(X_test)

    accuracy = accuracy_score(y_test, y_pred)
    print(f"Accuracy: {accuracy:.2f}")
    print('Неверно классифицированные образцы: %d' % (y_test != y_pred).sum())
    print(classification_report(y_test, y_pred))

    Accuracy: 0.96
    Неверно классифицированные образцы: 7
                  precision    recall  f1-score   support

               0       0.88      1.00      0.93        28
               1       0.95      0.95      0.95        21
               2       0.98      1.00      0.99        46
               3       0.97      0.86      0.91        36
               4       1.00      0.97      0.99        40

        accuracy                           0.96       171
       macro avg       0.95      0.96      0.95       171
    weighted avg       0.96      0.96      0.96       171
```

Рисунок 35 – Обучение и точность модели

В случае с многослойным персептроном были взяты следующие параметры:

– activation = 'relu' – функция активации для скрытых слоев для ускорения обучения,

– solver = 'lbfgs' – оптимизатор для обучения нейронной сети,

– alpha = 1e-5 – параметр регуляризации для уменьшения переобучения,

– max_iter = 10000 – максимальное количество обучения,

– hidden_layer_sizes = (5,) – количество нейронов в скрытых слоях (1 слой, 5 нейронов).

Вывод: в данном разделе были описаны наборы данных и их описательная статистика. Проведена работа с выбросами, масштабированием, категориальными признаками. Описаны методики обучение моделей на данных, подобраны параметры обучения. Для метода k-ближайших соседей определен способ выбора наилучшего k.

5 Характеристика результатов обучения

Проведем сравнительную характеристику результатов обучения на обработанных данных с выделением значимых признаков и без их выделения. Сравнение точностей моделей на разных наборах данных представлено в таблице 6. При классификации MLP указан не только процент точности, но и количество неверно классифицированных обьектов тестового набора.

Таблица 6 – Результаты обучения на наборе данных

Модель	Обучение на данных без обработки	Обучение после балансировки классов	Обучение на данных после обработки и выделения значимых признаков
Логистическая регрессия	0.6896	0.8198	0.8056
Метод опорных векторов	0.88	0.8772	0.8771
Случайный лес	0.985	0.9886	0.9941
K ближайших соседей	0.6425	0.7090	0.7192
MLP	0.95(20)	0.96(16)	0.96(7)

Очевидно, что после балансировки классов у некоторых моделей повысились показатели точности. Это связано с тем, что разные модели по-разному реагируют на баланс классов. К тому же для балансировки был использован SMOTE. Данный метод дабавляет к разбалансированным классам синтетические данные, что не всегда хорошо.

Увеличения точности показали практически все варианты моделей. В логистической регрессии балансировка оказывает влияние на свободный член, что и приводит к увеличению точности. В случае с методом опорных векторов изменение точности было незначительным на всех этапах обучения. Но данный метод показал один из лучших результатов.

Результаты обучения на данных без балансировки имеют показатели, близкие к данным с балансировкой, но при ее отсутствии некоторые модели склонны давать предсказания с опорой на класс большинства генеральной совокупности, что некорректно.

При выборе признаков были выбраны признаки указынные на рисунке 36.

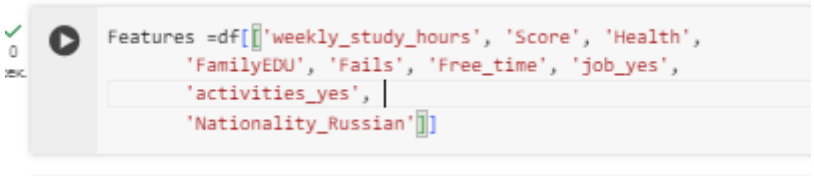


Рисунок 36– Выбранные признаки

При выборе данных признаков практически все модели показали небольшие улучшения, кроме логистической регрессии. В случае с ней точность сильно уменьшилась, что свидетельствует о том, что алгоритму не подошел данный набор признаков. В методе опорных векторов результаты остались на прошлом уровне, что соответствует задаче минимум. Но это ускорило работу модели, а также может сказаться на дальнейшей реализации.

Для увеличения показателей некоторых моделей разделим нашу выборку на три класса, путем деления на три новые группы.

Проведем классификацию и анализ точности при делении на три класса. Результаты представлены в таблице 6.

Таблица 6 – Результаты обучения нерезультативных аглоритмов

Модель	Обучение на данных со сбалансированными классами	Обучение на данных после обработки и выделения значимых признаков
Логистическая регрессия	0.9504	0.915
Метод опорных векторов	0.952	0.925
K ближайших соседей	0.856	0.8375

Из приведенных данных видно, что результаты моделей значительно улучшились при корректировке данных. Особенно это заметно в случае K ближайших соседей и логистической регрессии.

В случае с логистической регрессией на результат повлияло то, что данный алгоритм является бинарным и при использовании многоклассовой классоикации обобщается на несколько классов. Так, при делении на три класса в логистической регрессии будет создано три классификатора. При мсногоклассовой классификации были использованы ClassifierChain и

MultiOutputClassifier. Метод ClassifierChain показал результаты лучше, чем MultiOutputClassifier.

Метод k-ближайших соседей не имеет ограничений по количеству классов. Но при уменьшении их количества можно исправить проблему несоглассованности межклассовых расстояний.

Вывод: в данном разделе были рассмотрены результаты 5 алгоритмов обучения, но основе данных, полученных в 4 разделе. Проанализировав результаты обучений, можно сделать вывод, что улучшить качество моделей изменением данных возможно. Улучшения при изменении довольно значительны.

Настройки моделей при экспериментах не изменялись, за исключением выбора k-значения в методе k-ближайших соседей. Для каждого набора выбиралось k-значение, показывающее лучший результат.

ЗАКЛЮЧЕНИЕ

Во время исследования и анализа предметной области были рассмотрены различные варианты сбора и обработки, в том числе краудсорсинг, SQL-выгрузки, синтетическая генерация данных. Приведена таблица ресурсов для синтетической генерации. Также рассмотрены способы подготовки данных перед машинным обучением, используемый для адекватных моделей. Проведены эксперименты по сбору и обработке данных.

Были подробно рассмотрены алгоритмы обучения, логистическая регрессия и ее преобразования для многоклассовой классификации, метод k-ближайших соседей, метод случайного леса, метод опорных векторов, MLP. Для решения задач магистерской диссертации были проанализированы различные варианты сбора и обработки, которые позволяют эффективно решать задачи датаориентированного подхода машинного обучения.

В рамках данной выпускной квалификационной работы было проанализированно влияние различных методов на модель обучения для получения адекватных результатов критериев точности. В качестве инструмента для анализа и визуализаций использовался Python. В основу моделей легли основные встроенные модели языка.

Также были выделены критерии сбора данных для общих задач. А также проанализирована доступность метода для прикладной задачи.

Полученные результаты обладают научной ценностью, которая заключается в перспективе дальнейших исследований в области анализа данных, развитие которой позволит улучшить качество моделей методом корректировки данных, ведь настройки ML-моделей конечны и их использование не всегда показывает эффективные результаты.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Газимагамадова Д. М. Машинное обучение // Инновационный дискурс развития современной науки и технологий Сборник статей III Международной научно-практической конференции. Петрозаводск, 2021, С.45-50

2. Бондаренко А. Н. Деревья принятия решений, история развития моделей бустинга история математики // Сборник работ студентов-магистрантов факультета ВМК МГУ имени М. В. Ломоносова. Москва, 2021, С. 373–385

3. Майорова А. Р. Выбор способа сбора данных для ml: проблемы и критерии // Вызовы цифровой экономики: технологический суверенитет и экономическая безопасность Сборник статей VI Всероссийской научно-практической конференции с международным участием. – 2023. – С. 365–369

4. Для чего аналитику данных датасет и где его взять – URL: https://practicum.yandex.ru/blog/dataset-dlya-mashinnogo-obucheniya-i-analiza/ (дата обращения: 04.01.2024)

5. Сидорова Д. Н. Подготовка данных для кластеризации событий в журналах информационной безопасности //Инженерный вестник дона С. 207–215

6. SAM и тут, и сям: Segment Anything Model в задачах компьютерного зрения – URL: https://habr.com/ru/companies/sberdevices/articles/759328/ (дата обращения: 04.03.2024).

7. Новикова Д. В. Сравнительный анализ эффективности методов кодирования категориальных переменных в задаче прогнозирования // Заметки по информатике и математике. Сборник научных статей. Ярославль, 2022. С. 61–69.

8. Дьяконов А. Г. Методы решения задач классификации с категориальными признаками // Прикладная математика и информатика. Труды факультета Вычислительной математики и кибернетики МГУ имени М. В. Ломоносова. 2014. № 46. С. 103—127.

9. Меньшиков Я. С. Преимущества автоматического сбора данных в сети интернет над ручным сбором данных. // Universum: технические науки. 2022. № 10–1 (103). С. 33–36.

10. А. Болур, Т. Л. Андерсон, Л. Дж. Дикейзер и Х. К. Т.Вонг. Роль времени в обработке информации: обзор. SIGMOD Rec., 1982

11. Principles of dataset versioning: exploring the recreation/storage tradeoff – URL: https://www.researchgate.net/publication/277023254_Principles_of_Dataset_Versioning_Exploring_the_RecreationStorage_Tradeoff (дата обращения: 15.07.2023).

12. Райан Митчелл. Современный скрапинг веб-сайтов с помощью Python. 2-е межд. Издание

13. Боровская, Е. В. Основы искусственного интеллекта / Е. В. Боровская, Н. А. Давыдова. – 4-е изд. – М.: Лаборатория знаний, 2020. – 130 с.

14. Джоши, Патрик Искусственный интеллект с примерами на Python / Патрик Джоши. – СПб.: ООО «Диалектика», 2019. – 448 с.

15. Хенрик Б., Джозеф Р., Марк Ф. Машинное обучение. – Издательский дом «Питер», 2017. – 336 с.

16. Ведяхин А. Сильный искусственный интеллект: на подступах к сверхразуму / А. Ведяхин [и др.]. – М.: Интеллектуальная Литература, 2021. – 323 с.

17. Ermisch John & Pronzato Chiara. Causal Effects of Parents' Education on Children's Education // CHILD Working Papers wp05_10, CHILD - Centre for Household, Income, Labour and Demographic economics. Italy, 2010.

18. Губин Е.И. Методика подготовки больших данных для прогнозного анализа //Наука и бизнес: Пути развития, № 3(105) 2020, С. 27-31

19. Основные виды и сферы применения машинного обучения – URL: https://na-journal.ru/3-2022-informacionnye-tekhnologii/3487-osnovnye-vidy-i-sfery-primeneniya-mashinnogo-obucheniya. (дата обращения: 15.07.2023).

20. Handling modules – URL: https://docs.python.org/3/library/urllib.html. Дата обращения: (дата обращения: 01.07.2023).

21. Pandas – URL: https://pandas.pydata.org/ (дата обращения: 15.07.2023).

22. Machine Learning in Python – URL: https://scikit-learn.org/stable/ (дата обращения: 15.07.2023).

23. NumPy – URL: https://numpy.org/ (дата обращения: 15.07.2023).

ПРИЛОЖЕНИЕ А

(обязательное)

Исследование методов сбора и обработки данных для алгоритмов машинного обучения

Листинг программы

Листинг А.1 – Подготовка данных набора №1

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import scipy.stats as sts
%matplotlib inline
import statistics as st
import seaborn as sns
from sklearn.preprocessing import StandardScaler
from sklearn.model_selection import train_test_split
from sklearn.neighbors import KNeighborsClassifier
from sklearn.linear_model import LogisticRegression
from sklearn.ensemble import RandomForestClassifier
from sklearn.tree import DecisionTreeClassifier
from sklearn.svm import SVC
from sklearn.metrics import classification_report
from sklearn.metrics import confusion_matrix
import urbangrammar_graphics as ugg
rom sklearn.cluster import AgglomerativeClustering
import matplotlib.pyplot as plt
import pandas as pd

df=pd.read_csv('/content/DataForColab/ABCDE.csv',encoding='cp125
1', sep=';')

df['sex'].replace('M', 0,inplace=True)
df['sex'].replace('F', 1,inplace=True)

df = pd.get_dummies(df)

unique_values = df['Оценка'].unique()
print(unique_values)

counts = df.groupby(['ID учащегося',
'Семестр'])['Оценка'].value_counts()

counts = df.groupby(['ID учащегося',
'Семестр'])['Оценка'].value_counts()
pass_fail_counts = counts.unstack(fill_value=0)
print(pass_fail_counts)

# Сумма всех оценок за семестр по каждому id
total_grades = sum([pass_fail_counts[key] for key in
pass_fail_counts if key not in ["ID учащегося", "Семестр"]])
print(f"Сумма всех оценок за семестр: {total_grades}")

df = pd.DataFrame(pass_fail_counts)
df['Количество оценок']=total_grades
df['Процент отличных оценок'] = ((df['отл.(5)']+df['зачет']) /
df['Количество оценок']) * 100
df['Процент неудач'] = ((df['неуд.(2)'] + df['не атт.'] + df['не
доп.']) / df['Количество оценок']) * 100
```

```
df['зачетов'] = (df['зачет'] / df['Количество оценок']) * 100
df['Процент отличных оценок'] = (df['отл.(5)'] / df['Количество
оценок']) * 100
df['Процент неявок']= (df['н/я'] / df['Количество оценок']) *
100
df['Процент хороших оценок'] = (df['хор.(4)'] / df['Количество
оценок']) * 100

df['Процент удовлетворительных'] = (df['уд.(3)']/ df['Количество
оценок']) * 100

g = sns.pairplot(df_clean, hue="Predict",
palette=ugg.COLORS[3:])
g.fig.suptitle("Academic", y=1.01)

plt.figure(figsize=(8, 4))

plt.subplot(1, 2, 1)
plt.scatter(df_clean['Количество оценок'], df_clean['Процент
отличных оценок'], c=df_clean['Predict'], cmap='viridis')
plt.xlabel('Количество оценок')
plt.ylabel('Процент отличных оценок')
plt.title('K-means Clustering - Процент отличных оценок')

plt.subplot(1, 2, 2)
plt.scatter(df_clean['Количество оценок'], df_clean['Процент
неудач'], c=df_clean['Predict'], cmap='viridis')
plt.xlabel('Количество оценок')
plt.ylabel('Процент неудач')
plt.title('K-means Clustering - Процент неудач')

plt.tight_layout()
plt.show()

from sklearn.metrics import silhouette_score
silhouette_score(X, df_clean['Predict'])

model = AgglomerativeClustering(n_clusters=3)
X = df_clean[['Процент отличных оценок', 'Процент неудач',
'зачетов', 'Процент неявок', 'Процент хороших оценок']]
clusters = model.fit_predict(X)

df_clean['Cluster'] = clusters
plt.scatter(X.iloc[:,0], X.iloc[:,1], c=clusters,
cmap='viridis')
plt.xlabel('Процент отличных оценок')
plt.ylabel('Процент неудач')
plt.title('Кластеры')
plt.show()

pass_fail_counts.info()

mean=df_clean.describe().iloc[1]
```

```
mean

for col in df.columns:
 pct_missing = np.mean(df[col].isnull())
 k=round(pct_missing*100)
 print('{} - {}%'.format(col, k))

df_clean = df_clean.drop_duplicates()

corr = df_clean.corr()
plt.figure(figsize=(6,6))
sns.heatmap(corr, annot=True, cmap='coolwarm', square=True)
plt.show()
```

Листинг А.2 – Обработка данных набора №2

```
# Commented out IPython magic to ensure Python compatibility.
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import scipy.stats as sts
# %matplotlib inline
import statistics as st
import seaborn as sns
from sklearn.preprocessing import StandardScaler
from sklearn.model_selection import train_test_split
from sklearn.neighbors import KNeighborsClassifier
from sklearn.linear_model import LogisticRegression
from sklearn.ensemble import RandomForestClassifier
from sklearn.tree import DecisionTreeClassifier
from sklearn.svm import SVC

from sklearn.metrics import classification_report
from sklearn.metrics import confusion_matrix

df=pd.read_csv('/content/DataForColab/NEW_DF2.csv',encoding='cp1
251', sep=',')
df

df.loc[df['Predict'] == 'B', 'Predict'] = 'C'
df.loc[df['Predict'] == 'D', 'Predict'] = 'F'

for col in df.columns:
 pct_missing = np.mean(df[col].isnull())
 k=round(pct_missing*100)
 print('{} - {}%'.format(col, k))

df.describe()

df.describe(include='O')

df.info()

df['sex'].replace('M', 0,inplace=True)
df['sex'].replace('F', 1,inplace=True)
df['Predict'].replace('A', 0,inplace=True)
df['Predict'].replace('C', 1,inplace=True)
df['Predict'].replace('F', 2,inplace=True)
df['Predict'].replace('B', 3,inplace=True)
df['Predict'].replace('D', 4,inplace=True)

df.columns

features = [
    'sex', 'child_in_family', 'weekly_study_hours',
        'Score', 'Health',  'age',
       'Fails', 'Free_time'
```

```
]

df

features = [
     'child_in_family', 'weekly_study_hours',
        'Score', 'Health',  'age',
       'Fails', 'Free_time'
]

corr = df[features].corr()
plt.figure(figsize=(10,10))
sns.heatmap(corr, annot=True, cmap='coolwarm', square=True)
plt.show()

plt.figure(figsize=(15, 7))
for i in range(0, len(features)):
    plt.subplot(1, 10, i+1)

sns.boxplot(y=df[features[i]],color='darkturquoise',orient='v')
    plt.tight_layout()

"""Вывод: с помощью ящика с усами для многих показателей мы
можем определить 1 и 3 квартили, медиану, межквартильный
диапазон. Но в данном случае нас больше интересуют выбросы:

для age выбросы будут > 20;

для stadytime выбросы больше 3;

для failures >3;

и так далее

Убраем выбросы
"""

df_clean = df

df_clean=df_clean[df_clean['child_in_family']< 6]

df_clean=df_clean[df_clean['age']< 25]

df_clean

plt.figure(figsize=(15, 7))
for i in range(0, len(features)):
    plt.subplot(1, 10, i+1)

sns.boxplot(y=df_clean[features[i]],color='darkturquoise',orient
='v')
    plt.tight_layout()
```

```
df_clean = pd.get_dummies(df_clean)
df_clean

Predict=df_clean['Predict']
df_clean = df_clean.drop('Predict', axis=1)

df

# Затем вставляем его на последнее место
df_clean.insert(15, 'Predict', Predict)

X=df_clean.iloc[:,:15]
y=df_clean.iloc[:,15]

import matplotlib.pyplot as plt
from sklearn.feature_selection import mutual_info_classif

importances = mutual_info_classif(X, y)
feature_importances = pd.Series(importances,X.columns)
sorted_features =
feature_importances.sort_values(ascending=False)
sorted_features.plot(kind='barh', color='teal')
plt.show()
sorted_features

"""Наибольшую взаимосвязь с целевым показателем имеют признаки
failures, ChildCount, job_yes"""

# примените класс SelectKBest, чтобы извлечь лучшие показатели
from sklearn.feature_selection import SelectKBest
from sklearn.feature_selection import chi2

bestfeatures = SelectKBest(score_func=chi2, k='all')
fit = bestfeatures.fit(X,y)
dfscores = pd.DataFrame(fit.scores_)
dfcolumns = pd.DataFrame(X.columns)

#объединим два фрейма данных для лучшей визуализации
featureScores = pd.concat([dfcolumns,dfscores],axis=1)
featureScores.columns = ['Specs','Score']  #название показателей
print(featureScores.nlargest(11,'Score'))

sortedFeatures = featureScores.nlargest(11,'Score')
# feature_importances = pd.Series(featureScores['Score'])
sortedFeatures.plot(kind='barh', color='teal')
plt.show()

"""Методом lasso выделим основные признаки
Результат похож на полученные ранее. Оставим эти признаки.


"""
```

```
from sklearn.linear_model import LogisticRegression
from sklearn.feature_selection import SelectFromModel

# Устанавливаем наш параметр регуляризации C=1
logistic = LogisticRegression(C=1, penalty="l1",
solver='liblinear', random_state=7).fit(X, y)
# Где X, y - входные и выходные данные соответственно.
model = SelectFromModel(logistic, prefit=True)

X_new = model.transform(X)

# выводим финальную оценку прогнозирования.
selector_ind = model.get_support()
df_rfe1 = X.iloc[:, selector_ind]
print(df_rfe1.columns)

Label = df_clean['Predict']
Features =df_clean[['sex', 'child_in_family',
'weekly_study_hours', 'Score', 'Health',
       'FamilyEDU', 'age', 'Fails', 'Free_time', 'job_no',
'job_yes',
       'activities_no', 'Nationality_Russian']]

Features

from sklearn.preprocessing import MinMaxScaler
min_max_scaler = MinMaxScaler()
df_clean['child_in_family'] =
min_max_scaler.fit_transform(df_clean[['child_in_family']])
df_clean['weekly_study_hours'] =
min_max_scaler.fit_transform(df_clean[['weekly_study_hours']])
df_clean['Score'] =
min_max_scaler.fit_transform(df_clean[['Score']])
df_clean['Health'] =
min_max_scaler.fit_transform(df_clean[['Health']])
df_clean['FamilyEDU'] =
min_max_scaler.fit_transform(df_clean[['FamilyEDU']])
df_clean['age'] =
min_max_scaler.fit_transform(df_clean[['age']])
df_clean['Fails'] =
min_max_scaler.fit_transform(df_clean[['Fails']])
df_clean['Free_time'] =
min_max_scaler.fit_transform(df_clean[['Free_time']])
df_clean['sex'] =
min_max_scaler.fit_transform(df_clean[['sex']])

df_clean

unique_values = df_clean['Predict'].unique()

unique_values

counts = np.bincount(df_clean['Predict'], minlength=4)
```

```
for i, count in enumerate(counts):
    print(f"Class {i}: {count}")

"""Видим дисбаланс классов."""

import plotly.express as px
sns.countplot(x=df['Predict'],palette="pastel")
plt.box(False)
plt.xlabel('Predict',fontsize=11)
plt.ylabel('Количество наблюдений',fontsize=11)
plt.show()


fraud_or_not = df['Predict'].value_counts().tolist()
values = [fraud_or_not[0], fraud_or_not[1]]

X = df_clean.drop('Predict', axis=1)
y=df_clean['Predict']

from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y,
stratify=y, test_size=0.25)

from imblearn.over_sampling import SMOTE
smote = SMOTE(sampling_strategy='minority')
X_sm, y_sm = smote.fit_resample(X, y)
print(X_sm.shape)
print(X.shape)
df_new_sm=pd.concat([X_sm, y_sm], axis=1)
df_new_sm.shape

sns.countplot(x=df_new_sm['Predict'], palette="pastel")
plt.box(False)
plt.xlabel('Predict ',fontsize=11)
plt.ylabel('Количество наблюдений',fontsize=11)
plt.show()

"""Улучшения очевидны. Сбалансируем выборку по среднему
классу."""

df_new_sm

# перемешаем датафрейм
shuffled_df = df_new_sm.sample(frac=1,random_state=4)
CHD_df = shuffled_df.loc[shuffled_df['Predict'] == 2]

non_CHD_df = shuffled_df.loc[shuffled_df['Predict'] ==
1].sample(n=250,random_state=42)
second_CHD_df = shuffled_df.loc[shuffled_df['Predict'] ==
0].sample(n=250,random_state=42)
```

```
# Снова объединим оба фрейма данных
normalized_df = pd.concat([CHD_df, non_CHD_df,second_CHD_df])

sns.countplot(x=normalized_df['Predict'], palette="pastel")
plt.box(False)
plt.xlabel('Predict ',fontsize=11)
plt.ylabel('Количество наблюдений',fontsize=11)
plt.show()


# Сохраним датафреймы в формате CSV
normalized_df.to_csv(file_path, index=False)

file_path = '/content/df_new_sm3.csv'
```

Листинг А.3 – Обучение моделей

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import scipy.stats as sts
# %matplotlib inline
import statistics as st
import seaborn as sns
from sklearn.preprocessing import StandardScaler
from sklearn.model_selection import train_test_split
from sklearn.neighbors import KNeighborsClassifier
from sklearn.linear_model import LogisticRegression
from sklearn.ensemble import RandomForestClassifier
from sklearn.tree import DecisionTreeClassifier
from sklearn.svm import SVC

from sklearn.metrics import classification_report
from sklearn.metrics import confusion_matrix

df=pd.read_csv('/content/DataForColab/normalized_df5.csv',encodi
ng='cp1251', sep=',')
df

sns.countplot(df['Predict'])

df['Predict'].replace('A', 0,inplace=True)
df['Predict'].replace('C', 1,inplace=True)
df['Predict'].replace('F', 2,inplace=True)
df['Predict'].replace('B', 3,inplace=True)
df['Predict'].replace('D', 4,inplace=True)

label = df['Predict']

df.columns

Features =df[['weekly_study_hours', 'Score', 'Health',
       'FamilyEDU', 'Fails', 'Free_time', 'job_yes',
       'activities_yes',
       'Nationality_Russian']]

scaler = StandardScaler()
scaler.fit(Features)
scaled = scaler.transform(Features)

duplicateRows = df[df.duplicated ()]
duplicateRows

from sklearn.preprocessing import MinMaxScaler
min_max_scaler = MinMaxScaler()
df['age'] = min_max_scaler.fit_transform(df[['age']])

Label = df['Predict']
```

```
from sklearn.preprocessing import StandardScaler
scaler = StandardScaler()
scaler.fit(Features)
scaled = scaler.transform(Features)

from sklearn.model_selection import train_test_split
X = scaled
y = Label
X_train, X_test, y_train, y_test = train_test_split(X, y,
test_size=0.20, random_state=42)

"""Определение оптимального K"""

from sklearn.metrics import accuracy_score
acc = {}
for k in range(3, 30, 2):
    knn = KNeighborsClassifier(n_neighbors=k)
    knn.fit(X_train, y_train)
    y_pred = knn.predict(X_test)
    acc[k] = accuracy_score(y_test, y_pred)

plt.plot(range(3,30,2), acc.values())
plt.xlabel('K')
plt.ylabel('Accuracy')
plt.show()

knn = KNeighborsClassifier(n_neighbors=3)
knn.fit(X_train, y_train)
y_pred = knn.predict(X_test)
print(accuracy_score(y_test, y_pred))
print(classification_report(y_test, y_pred))

"""# Логистическая регрессия"""

from sklearn.preprocessing import label_binarize
y = label_binarize(y, classes=[0, 1, 2, 3, 4])
n_classes = y.shape[1]
X_train, X_test, y_train, y_test = train_test_split(X, y,
test_size=0.5, random_state=0)

from sklearn.multioutput import ClassifierChain

base_lr = LogisticRegression(solver='lbfgs', random_state=0)
chain = ClassifierChain(base_lr, order='random', random_state=0)
pred=chain.fit(X_train, y_train).predict(X_test)

from sklearn.multioutput import MultiOutputClassifier
classifier =
MultiOutputClassifier(LogisticRegression()).fit(X_train,
y_train)
pred=classifier.predict(X_test)
```

```
#точность
from sklearn.metrics import accuracy_score
acc = accuracy_score(y_test.argmax(axis=1), pred.argmax(axis=1))
print(f"Точность: {round(acc,3)*100}%")

from sklearn.metrics import precision_recall_curve
from sklearn.metrics import roc_curve, auc
precision, recall, thresholds = roc_curve(y_test.ravel(),
pred.ravel())

fpr = dict()
tpr = dict()
roc_auc = dict()
for i in range(n_classes):
    fpr[i], tpr[i], _ = roc_curve(y_test[:, i], pred[:, i])
    roc_auc[i] = auc(fpr[i], tpr[i])

fpr["micro"], tpr["micro"], _ = roc_curve(y_test.ravel(),
pred.ravel())
roc_auc["micro"] = auc(fpr["micro"], tpr["micro"])
fpr["macro"], tpr["macro"], _ = roc_curve(y_test.ravel(),
pred.ravel())
roc_auc["macro"] = auc(fpr["macro"], tpr["macro"])

plt.figure()
lw = 2
plt.plot(fpr['micro'],tpr['micro'],color="darkorange",lw=lw,labe
l="ROC curve (area = %0.2f)" % roc_auc[2],)
plt.plot([0, 1], [0, 1], color="navy", lw=lw, linestyle="--")
plt.xlim([0.0, 1.0])
plt.ylim([0.0, 1.05])
plt.legend(loc="lower right")
plt.show()

from sklearn.multioutput import ClassifierChain
from sklearn.linear_model import LogisticRegression
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import label_binarize

y = label_binarize(y, classes=[1, 4, 2, 0, 3])
n_classes = y.shape[1]
X_train, X_test, y_train, y_test = train_test_split(X, y,
test_size=0.5, random_state=0)
base_lr = LogisticRegression(solver='lbfgs', random_state=0)
chain = ClassifierChain(base_lr, order='random', random_state=0)
y_score = chain.fit(X_train, y_train).predict(X_test)
accuracy = accuracy_score(y_test, y_score)
print(f'Accuracy: {accuracy}')
print(classification_report(y_test, y_score))

"""Использование другого классификатора SVC"""

from sklearn.model_selection import train_test_split
```

```
X = scaled
y = Label
# split the data to 20% test,80% train with random state=42
X_train, X_test, y_train, y_test = train_test_split(X, y,
test_size=0.20, random_state=42)

model = SVC()
model.fit(X_train, y_train)

y_pred = model.predict(X_test)

accuracy = accuracy_score(y_test, y_pred)
print(f'Accuracy: {accuracy}')
print(classification_report(y_test, y_pred))

"""Рандомный лес"""

clf = RandomForestClassifier(n_estimators = 100)

clf.fit(X_train, y_train)
y_pred = clf.predict(X_test)

from sklearn import metrics

accuracy = accuracy_score(y_test, y_pred)
print(f'Accuracy: {accuracy}')
print(classification_report(y_test, y_pred))

"""Перцептрон"""

from sklearn.neural_network import MLPClassifier
mlp = MLPClassifier(activation='relu', solver='lbfgs', alpha=1e-
5, max_iter=10000, hidden_layer_sizes=(5,))
mlp.fit(X_train, y_train)
y_pred = mlp.predict(X_test)

mlp = MLPClassifier(hidden_layer_sizes=(64, 32),
                    max_iter=1000, random_state=42)

mlp.fit(X_train, y_train)

y_pred = mlp.predict(X_test)

accuracy = accuracy_score(y_test, y_pred)
print(f"Accuracy: {accuracy:.2f}")
print('Неверно классифицированные образцы: %d' % (y_test !=
y_pred).sum())
print(classification_report(y_test, y_pred))
```

ПРИЛОЖЕНИЕ Б

(обязательное)

Исследование методов сбора и обработки данных для алгоритмов машинного обучения

Презентация

Исследование методов сбора и обработки данных для алгоритмов машинного обучения

Выполнил: студент гр.22ВВИм2
Майорова А.Р.
Руководитель: к.т.н., профессор
Зинкин С.А.

Пенза, 2024

Цель, объект и предмет работы

Цель работы:

Исследование эффективности и скорости сбора и обработки данных для решения задач машинного обучения.

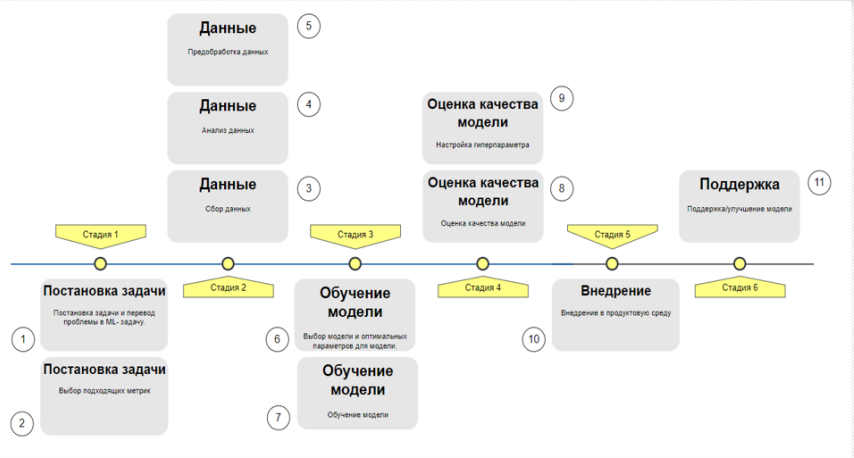
Объект исследования:

Методы сбора и обработки данных.

Предмет исследования:

Применение различных методов сбора и обработки данных с последующим их использованием в машинном обучении.

Этапы машинного обучения



Источники и критерии сбора данных

Источники:

- Выбор готового набора данных
- Парсинг и скраппинг сторонних ресурсов
- Сбор данных на основе имеющихся данных
- Сбор данных вручную
- Краудсорсинг
- Генерация синтетических данных

Критерии:

- Отношение стоимость-качество
- Актуальность и соответствие задаче
- Масштабируемость
- Временные характеристики

Способ	Таблицы	Изображения и видео	Текст	Аудио
Модели	GANS CTGAN WGAN WGAN-GP VAES	GANS VAES	GPT-3 GPT-4 PaLM LLAMA Bard	WaveNet Jukebox Voicebox
Инструменты	SDV Copulas	Unity Blender ThreeDWorld	Jasper Copy.ai	Ableton Live iZotope Audacity
Сервисы	MOSTLY AI YData Hazy DataRobot Tonic	NVIDIA Omniverse CVEDIA	Rytr.me Notion Al	ElevenLabs Murf Lovo Mubert

Процесс подготовки данных

Инструменты анализа

Средства разработки:

- **Python**

Инструменты:

- **Google colab**
- **Git**

Исследуемые наборы данных

	ID учащегося	Название дисциплины	Тип отчетности	Курс	Семестр	Оценка	Дата выставления (по ведомости)	Дата выставления (по факту выставления)
0	2128152	Интеллектуальные системы	Экзамен	1	1	отл.(5)	27.01.2022	27.01.2022
1	2128152	Информационное общество и проблемы прикладной ...	Зачет	1	1	зачет	29.12.2021	19.01.2022
2	2128152	Информационное общество и проблемы прикладной ...	Экзамен	1	1	отл.(5)	19.01.2022	19.01.2022
3	2128152	Информационное общество и проблемы прикладной ...	Курсовой проект	1	1	отл.(5)	29.12.2021	19.01.2022
4	2128152	Математическое моделирование	Зачет	1	1	зачет	10.01.2022	01.02.2022
...	...	...	...	...	...	...	...	...
68977	517727579	Иностранный язык	Зачет	1	1	зачет	03.04.2023	03.04.2023
68978	517727579	История (история России, всеобщая история)	Экзамен	1	1	хор.(4)	03.04.2023	03.04.2023
68979	517727579	Математика	Экзамен	1	1	хор.(4)	03.04.2023	
68980	517727579	Русский язык и деловые коммуникации	Зачет	1	1	зачет	03.04.2023	
68981	517727579	Физическая культура и спорт	Зачет	1	1	зачет	03.04.2023	

Выгрузка, полученная из ЭИОС

```
df=pd.read_csv('/content/DataForColab/NEW_DF2.csv',encoding='cp1251', sep=',')
df
```

	sex	job	child_in_family	activities	weekly_study_hours	Nationality	Score	Health	FamilyEDU	age	Predict	Fails	Free_time
0	M	no	3	no	27	Russian	2.93	1	0	19	F	6	4
1	F	no	2	no	47	Russian	2.98	4	1	21	F	15	19
2	F	no	9	yes	13	Russian	2.98	2	0	22	F	12	22
3	F	no	5	no	3	Russian	3.01	5	0	20	F	9	11
4	M	no	5	no	10	Russian	3.04	3	1	19	F	10	4
...	...	...	...	...	...	...	...	...	...	...	...	...	...
1992	M	no	2	yes	12	Russian	4.79	2	0	18	A	1	18
1993	M	no	3	no	46	Russian	4.80	2	1	17	A	1	8
1994	M	no	1	no	19	Russian	4.80	4	1	21	A	1	9
1995	M	no	2	no	30	Russian	4.80	4	0	17	A	1	12
1996	M	no	2	no	20	Russian	4.80	1	1	21	A	0	23

1997 rows × 13 columns

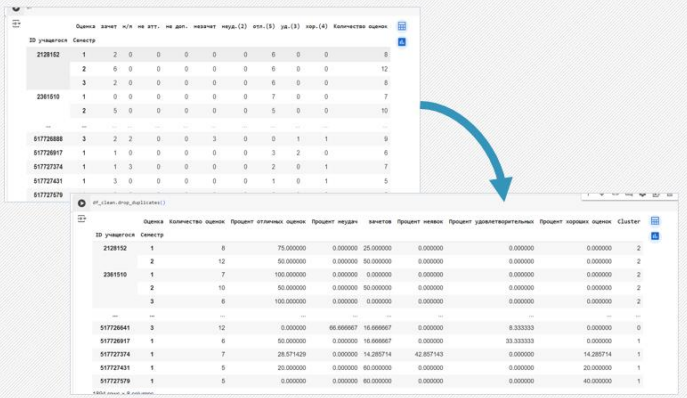
Набор, созданный с помощью опроса и генерации данных на основе нейронной сети

Модели данных

Атрибут	Описание атрибута
ID учащегося	Содержит ID учащегося в системе
Семестр обучения	Семестр обучения, к которому относятся оценки
Курс	Курс, на котором обучается студент
Название дисциплин	Содержит наименование дисциплины
Оценка	Оценка по дисциплине в определенном курсе и семестре
Дата выставления по факту	Содержит дату выставления оценки в формате ДД.ММ.ГГГГ
Дата выставления по ведомости	Содержит дату выставления оценки в формате ДД.ММ.ГГГГ

Атрибут	Описание атрибута
sex	Пол студента
age	Возраст студента
weekly_study_hours	Время, которое студент выделяет на внеурочную деятельность
fails	Количество долгов на момент сбора данных
job	Признак трудоустроенности студента. Имеет ли студент работу на момент сбора данных.
familyEDU	Признак, показывающий имеют ли высшее образование оба родителя в семье
activities	Признак, отвечающий за дополнительные активности студента
Free_time	Количество свободного времени
health	Рейтинг здоровья
child_in_family	Количество детей в семье
nationality	Национальность студента
score	Средний балл по профильным предметам (учитывая отсутствие оценки, как отсутствие предмета)
predict	Целевая переменная

Анализ и подготовка набора №1

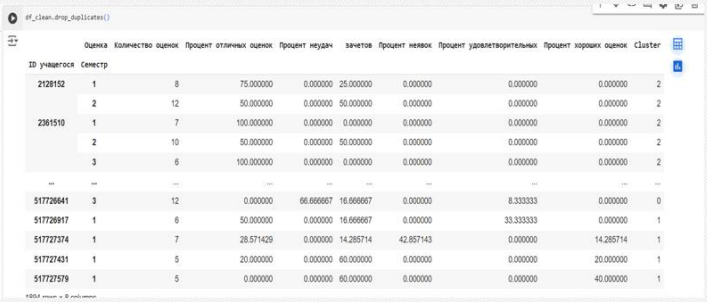


```
[146] mean

Оценка
Количество оценок                 9.312091
Процент отличных оценок          19.408551
Процент неудач                    5.626957
Зачетов                          36.621079
Процент неявок                    5.936630
Процент удовлетворительных       15.766501
Процент хороших оценок           16.139091
Predict                           1.413483
Name: mean, dtype: float64

[147] median

Оценка
Количество оценок                 10.000000
Процент отличных оценок           10.000000
Процент неудач                     0.000000
Зачетов                           41.666667
Процент неявок                     0.000000
Процент удовлетворительных         0.000000
Процент хороших оценок            11.111111
Predict                            2.000000
Name: 50%, dtype: float64
```

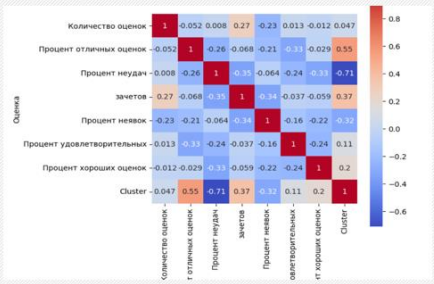
```
for col in df.columns:
 pct_missing = np.mean(df[col].isnull())
 k=round(pct_missing*100)
 print('{} - {}%'.format(col, k))
```

```
зачет - 0%
н/я - 0%
не атт. - 0%
не доп. - 0%
незачет - 0%
неуд.(2) - 0%
отл.(5) - 0%
уд.(3) - 0%
хор.(4) - 0%
Сданные - 0%
Несданные - 0%
Количество оценок - 0%
Процент отличных оценок - 0%
Процент неудач - 0%
зачетов - 0%
Процент неявок - 0%
Процент удовлетворительных - 0%
Процент хороших оценок - 0%
```

Первый набор после преобразований и его тепловая карта



ID учащегося	Семестр	Оценка	Количество оценок	Процент отличных оценок	Процент неудач	зачетов	Процент неявок	Процент удовлетворительных	Процент хороших оценок	Cluster
2128152	1	8	75.000000	0.000000	25.000000	0.000000	0.000000	0.000000	2
	2	12	50.000000	0.000000	50.000000	0.000000	0.000000	0.000000	2
2361510	1	7	100.000000	0.000000	0.000000	0.000000	0.000000	0.000000	2
	2	10	50.000000	0.000000	50.000000	0.000000	0.000000	0.000000	2
	3	6	100.000000	0.000000	0.000000	0.000000	0.000000	0.000000	2
...	...	...	...	...	...	...	...	...	...
517726641	3	12	0.000000	66.666667	16.666667	0.000000	8.333333	0.000000	0
517726917	1	6	50.000000	0.000000	16.666667	0.000000	33.333333	0.000000	1
517727374	1	7	28.571429	0.000000	14.285714	42.857143	0.000000	14.285714	1
517727431	1	5	20.000000	0.000000	60.000000	0.000000	0.000000	20.000000	1
517727579	1	5	0.000000	0.000000	60.000000	0.000000	0.000000	40.000000	1

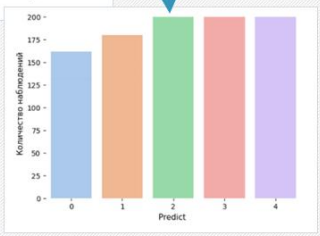


Анализ и подготовка набора №2

	mean	mean	0
sex	0.500250	0.500250	1.0
child_in_family	3.660991	3.660991	2.0
weekly_study_hours	17.759139	17.759139	3
Score	4.041112	4.041112	4.01
Health	2.958938	2.958938	1.0
FamilyEDU	0.504256	0.504256	1.0
age	19.475714	19.475714	17.0
Fails	2.648473	2.648473	0.0
Free_time	12.228343	12.228343	7.0
job	NaN	NaN	no
activities	NaN	NaN	no
Nationality	NaN	NaN	Russian
Predict	NaN	NaN	B

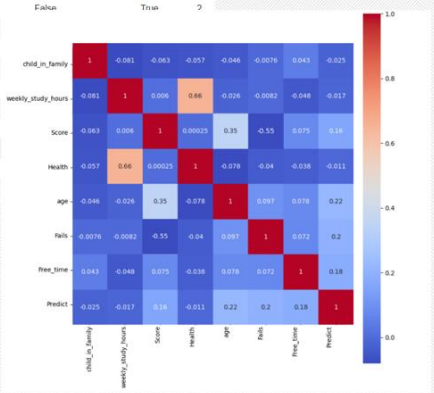
```
for col in df.columns:
    pct_missing = np.mean(df[col].isnull())
    k=round(pct_missing*100)
    print('{} - {}%'.format(col, k))
```

```
sex - 0%
job - 0%
child_in_family - 0%
activities - 0%
weekly_study_hours - 0%
Nationality - 0%
Score - 0%
Health - 0%
FamilyEDU - 0%
age - 0%
Predict - 0%
Fails - 0%
Free_time - 0%
```



Второй набор после преобразований и его тепловая карта

	sex	child_in_family	weekly_study_hours	Score	Health	FamilyEDU	age	Fails	Free_time	job_no	job_yes	activities_no	activities_yes	Nationality_Kyrgyz	Nationality_Russian	Predict
3	0.0	0.800000	0.400000	0.427807	0.250000	1.0	0.000000	0.142857	0.428571	True	False	True	False	False	True	2
2	0.0	0.600000	0.120000	0.390374	0.750000	0.0	0.200000	0.190476	0.714286	True	False	True	False	False	True	2
9	1.0	0.200000	0.280000	0.379679	0.250000	1.0	0.800000	0.000000	0.000000	True	False	False	True			
	0.0	1.000000	0.620000	0.283422	1.000000	0.0	0.800000	0.857143	0.571429	True	False	True	False			
9	0.0	1.000000	0.060000	0.449198	1.000000	1.0	0.400000	0.285714	0.476190	True	False	True	False			
	...	...	...	...	...	...	...	...	...	...	...	...	...			
6	0.0	0.200000	0.280000	0.855615	0.000000	0.0	0.000000	0.000000	0.238095	True	False	True	False			
0	0.0	0.631742	0.060000	0.972413	1.000000	0.0	0.884129	0.027588	0.837112	True	False	True	False			
1	1.0	0.248983	0.044898	0.877989	0.811229	0.0	1.000000	0.047619	0.143826	True	False	True	False			
4	0.0	0.359334	0.687967	0.871658	0.250000	0.0	0.000000	0.000000	0.714682	True	False	True	False			
0	0.0	0.244539	0.095631	0.885903	0.500000	1.0	0.222695	0.000000	0.772509	True	False	False	True			



Модели ML

- Логистическая регрессия
- Метод k-ближайших соседей
- Случайный лес
- Метод опорных векторов
- MLP

```
[ ] knn = KNeighborsClassifier(n_neighbors=3)
    knn.fit(X_train, y_train)
    y_pred = knn.predict(X_test)
    print(accuracy_score(y_test, y_pred))
    print(classification_report(y_test, y_pred))

0.7192982456140351
              precision    recall  f1-score   support

           0       0.79
           1       0.38
           2       0.91
           3       0.63
           4       0.71

    accuracy
   macro avg       0.69
weighted avg       0.72
```

```
[ ] from sklearn.neural_network import MLPClassifier
    mlp = MLPClassifier(activation='relu', solver='lbfgs', alpha=1e-5, max_iter=10000, hidden_layer_sizes=(5,))
    mlp.fit(X_train, y_train)
    y_pred = mlp.predict(X_test)

[ ] mlp = MLPClassifier(hidden_layer_sizes=(64, 32),
                        max_iter=1000, random_state=42)

    mlp.fit(X_train, y_train)

    y_pred = mlp.predict(X_test)

    accuracy = accuracy_score(y_test, y_pred)
    print(f"Accuracy: {accuracy:.2f}")
    print('Неверно классифицированные образцы: %d' % (y_test != y_pred).sum())
    print(classification_report(y_test, y_pred))

Accuracy: 0.96
Неверно классифицированные образцы: 7
              precision    recall  f1-score   support

           0       0.88      1.00      0.93        28
           1       0.95      0.95      0.95        21
           2       0.98      1.00      0.99        46
           3       0.97      0.86      0.91        36
           4       1.00      0.97      0.99        40

    accuracy                           0.96       171
   macro avg       0.95      0.96      0.95       171
weighted avg       0.96      0.96      0.96       171
```

Результаты обучения

Модель	Обучение на данных без обработки	Обучение после балансировки классов	Обучение на данных после обработки и выделения значимых признаков
Логистическая регрессия	0.6896	0.8198	0.8056
Метод опорных векторов	0.88	0.8772	0.8771
Случайный лес	0.985	0.9886	0.9941
К ближайших соседей	0.6425	0.7090	0.7192
MLP	0.95(20)	0.96(16)	0.96(7)

Результаты обучения

Модель	Обучение на данных со сбалансированными классами	Обучение на данных после обработки и выделения значимых признаков
Логистическая регрессия	0.9504	0.915
Метод опорных векторов	0.952	0.925
К ближайших соседей	0.856	0.8375

Заключение

В результате выполнения работы:

- Рассмотрены различные варианты сбора и обработки данных. Приведена таблица ресурсов для синтетической генерации.
- Обучены 5 моделей на 5 наборах данных.
- Проанализировано влияние различных методов на модель обучения для получения адекватных результатов и критериев точности.

Полученные результаты обладают научной ценностью, которая заключается в перспективе дальнейших исследований в области анализа данных, развитие которой позволит улучшить качество моделей методом корректировки данных, ведь настройки ML-моделей конечны и их использование не всегда показывает эффективные результаты.

Также данные модели можно применить для прогнозирования успеваемости, но для данной задачи лучше использовать потоковые данные для обучения.