МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
Федеральное государственное автономное образовательное учреждение высшего образования
«Дальневосточный федеральный университет»
(ДВФУ)
Институт математики и компьютерных технологий
Департамент программной инженерии и искусственного интеллекта

Ягольницкий Сергей Юрьевич

ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА

бакалаврская работа
вид ВКР

ПРОГРАММНОЕ СРЕДСТВО ДЛЯ ПРОВЕДЕНИЯ ЭКСПЕРИМЕНТОВ ПО МАШИННОМУ ОБУЧЕНИЮ. ПОДСИСТЕМА ПРЕДОБРАБОТКИ ДАННЫХ

по направлению подготовки (специальности) 09.03.04 «Программная инженерия»
профиль «Программная инженерия»

Владивосток
2023

В материалах данной выпускной квалификационной работы не содержатся сведения, составляющие государственную тайну, и сведения, подлежащие экспортному контролю	Автор работы _____ подпись группа Б9119-09.03.04прогин « ____ » _____ 2023 г.
Уполномоченный по экспортному контролю И.Л. Артемьева подпись И.О. Фамилия « ____ » _____ 2023 г.	Руководитель ВКР профессор, к.т.н. должность, ученое звание С.В. Смагин подпись И.О. Фамилия « ____ » _____ 2023 г.
Защищена в ГЭК с оценкой Секретарь ГЭК О.А. Крестникова подпись И.О. Фамилия « ____ » _____ 2023 г.	**«Допустить к защите»** и.о. директора департамента ученая степень, ученое звание О.А. Крестникова подпись И.О. Фамилия « ____ » _____ 2023 г.

Аннотация

В работе представлено описание подсистемы предобработки данных программного средства для проведения экспериментов по машинному обучению. Данное средство предназначено для специалистов в области машинного обучения, а также для обучения студентов. В первой главе работы представлен обзор литературы на тему «Программное средство для проведения экспериментов по машинному обучению. Подсистема предобработки данных», обоснована актуальность работы. Во второй главе приведена модель предметной области и осуществлена формальная постановка задачи предобработки выборки. В третьей главе представлена техническая документация, описывающая требования и архитектуру разрабатываемого программного средства. В четвертой главе описаны методы реализации, стек используемых технологий, представлены результаты тестирования и экспериментального исследования.

Оглавление

Введение

Машинное обучение – одна из наиболее актуальных и быстро развивающихся областей в сфере информационных технологий. В настоящее время мы сталкиваемся с огромным количеством данных, из которых при помощи машинного обучения можно извлечь ценную информацию и создать предсказательные модели. Это создает огромный потенциал для улучшения бизнес-процессов и качества жизни людей.

Областей применения машинного обучения очень много, вот некоторые из них: медицина, финансы, производство, транспорт, маркетинг, наука. Автоматизация бизнес-процессов, повышение эффективности производства, улучшение качества медицинских услуг и предоставление более точных прогнозов – все это стало возможно благодаря машинному обучению.

Эксперименты играют очень важную роль в машинном обучении. Они позволяют:

- Улучшить качество модели: эксперименты помогают определить оптимальные параметры модели, выбрать наиболее эффективные алгоритмы и настроить модель для оптимального решения конкретной задачи.
- Оценить производительность модели: эксперименты позволяют оценить скорость работы модели, объем памяти, необходимый для ее функционирования, а также ресурсы, необходимые для ее обучения и тестирования.
- Обеспечить надежность модели: эксперименты помогают проверить модель на различных данных, убедиться в ее способности к обобщению и уменьшить вероятность переобучения.
- Повысить интерпретируемость модели: эксперименты могут помочь понять, какие признаки влияют на предсказания модели, и какие изменения приводят к улучшению или ухудшению ее качества.

Как правило, эксперименты состоят из нескольких этапов: предобработка данных, выбор модели, настройка параметров, обучение модели, оценка модели.

Каждый из этих этапов может включать в себя множество различных подзадач и действий.

Целью выпускной квалификационной работы бакалавра является разработка подсистемы предобработки данных программного средства для проведения экспериментов по машинному обучению.

Задачи выпускной квалификационной работы бакалавра:

1. Обзор литературы на тему «Предобработка данных», обоснование актуальности работы.

2. Анализ и построение модели предметной области, формализация понятия эксперимент, осуществление формальной постановки задачи предобработки данных

3. Разработка технического проекта подсистемы программного средства, включающего в себя требования и архитектуру всех подсистем.

4. Создание прототипа и реализация подсистемы программного средства, используя заданный стек технологий, проведение тестирования и экспериментального исследования разработанной подсистемы.

1 Анализ существующих программных средств, методов и решений задачи предобработки данных

В данной главе рассмотрены и проанализированы существующие программные средства и методы предобработки данных для машинного обучения.

1.1 Методы для предобработки данных

В машинном обучении предварительная обработка данных является важным этапом, который включает в себя различные методы для подготовки данных перед их использованием в обучении моделей. Рассмотрены следующие методы для предварительной обработки данных:

1) обработка пропущенных значений;
2) нормализация признаков,
3) кодирование категориальных признаков,
4) выбор признаков,
5) обработка выбросов,
6) обработка дисбаланса классов,
7) Разбиение на тренировочную и тестовую выборки.

1.1.1 Обработка пропущенных значений

Пропущенные значения – это значения, которые отсутствуют в признаках входных данных машинного обучения. Они могут возникать по разным причинам, например, из-за ошибок в сборе данных, отсутствия значений в данных, потери данных при передаче и т.д.

Пропущенные значения могут оказывать негативное влияние на качество модели машинного обучения, а некоторые модели и вовсе не работают с наличием пропусков, поэтому необходимо разработать стратегии обработки пропущенных значений[10].

Существуют различные методы обработки пропущенных значений, которые рассмотрим далее.

1.1.1.1 Удаление строк или столбцов с пропущенными значениями

Метод удаления строк или столбцов с пропущенными значениями – это один из методов обработки пропущенных значений в данных, который заключается в удалении строк или столбцов, содержащих пропущенные значения.

Данный метод может быть эффективным, если в данных относительно небольшое количество пропущенных значений и удаление строк или столбцов с пропущенными значениями не повлияет на общую репрезентативность исходных данных.

Для применения данного метода необходимо проанализировать данные и определить, какие строки или столбцы содержат пропущенные значения. Затем, при наличии небольшого количества таких строк или столбцов, их можно удалить, чтобы получить набор данных без пропущенных значений. Однако, необходимо понимать, что при удалении строк или столбцов с пропущенными значениями может произойти потеря информации и важных данных, что может повлиять на результаты анализа[15].

Кроме того, при использовании данного метода необходимо учитывать, что удаление строк или столбцов может изменить статистические характеристики данных, такие как среднее значение, медиана, стандартное отклонение и т.д. Поэтому перед удалением строк или столбцов необходимо оценить, как это повлияет на результаты анализа данных.

Данный метод может быть эффективным в тех случаях, когда пропущенные значения случайны и их количество невелико по сравнению с размером набора данных. Однако, если пропущенные значения имеют систематический характер, например, если они отсутствуют только в определенных категориях или группах данных, то использование данного метода может привести к искажению результатов и ошибочным выводам.

В целом, метод удаления строк или столбцов с пропущенными значениями является одним из наиболее простых и быстрых методов обработки

пропущенных значений, который может быть эффективен в некоторых случаях, но не всегда является оптимальным решением.

1.1.1.2 Замена пропущенных значений

Ещё один из способов обработки пропущенных значений в данных – это замена или заполнение пропущенных значений значениями, которые должны быть представлены в данных. Замена пропущенных значений может быть основана на различных стратегиях.

Самый простой способ замены пропущенных значений – это заполнить пропущенные значения средним, медианой или модой. Например, для количественных данных можно вычислить среднее значение и заменить все пропущенные значения на это значение. Для категориальных данных можно заменить все пропущенные значения на моду.

Другой метод замены пропущенных значений – замена значений на основе статистических моделей. Например, можно использовать методы регрессии или классификации, чтобы заполнить пропущенные значения на основе других свойств объектов.

Кроме того, можно использовать интерполяцию для заполнения пропущенных значений. Интерполяция – это метод, который используется для оценки пропущенных значений на основе значений других объектов. Например, можно использовать линейную интерполяцию, чтобы заполнить пропущенные значения между двумя близкими точками данных.

Важно помнить, что замена пропущенных значений может привести к искажению данных и возможно снижению точности модели машинного обучения. Поэтому необходимо тщательно выбирать метод и стратегию замены пропущенных значений в зависимости от специфики данных и задачи машинного обучения.

1.1.2 Нормализация признаков

Нормализация – это процесс приведения значений признаков к одному масштабу. Это может потребоваться, если признаки измерены в разных единицах измерения или имеют разные диапазоны значений[10].

1) Десятичная нормализация производится путем перемещения десятичной точки на число разрядов:

$$x_i = \frac{x_i}{10^n}, \quad (1)$$

где n – число разрядов наибольшего наблюдаемого значения.

2) Минмакс – линейное преобразование данных в диапазоне от 0 до 1, по следующие формуле:

$$X_{norm} = \frac{X - X_{min}}{X_{max} - X_{min}} \quad (2)$$

В отличии от прошлого метода результирующие значения будут располагаться на всем диапазоне.

3) Z-стандартизация. Недостатком MinMax нормализации является наличие аномальных значений данных, которые растягивают диапазон, что приводит к тому, что нормализованные значения концентрируются в некотором узком диапазоне. Чтобы избежать этого, следует определять диапазон с помощью значений среднего и дисперсии:

$$x_i = \frac{(x_i - \bar{X})}{\sigma_x}, \quad (3)$$

где $\bar{X}$ – среднее арифметическое признака X, σ_x – стандартное отклонение значений признака X.

1.1.3 Кодирование категориальных признаков

Кодирование категориальных признаков – процедура, которая представляет собой преобразование категориальных признаков в численное представление по некоторым оговоренным ранее правилам. Множество моделей машинного обучения способны работать только с числовыми данными, поэтому данная обработка может оказаться необходимой. Далее разберем способы кодирования признаков или кодировщики(encoder)[5].

Label Encoder

Самый часто используемый метод. Преобразование представляет из себя присваивание и использование уникального ключа-числа для каждой категории. Однако при кодировании данным способом некоторые алгоритмы могут обнаружить зависимости, которые не подразумевались.

One-hot Encoder

Данный тип кодирования, основывается на создании бинарных признаков, которые показывают принадлежность к уникальному значению. Т.е. на каждое значение признака создаться новый признак и принадлежность к нему будет выражаться через единицу, а отсутствие принадлежности через ноль.

Главным недостатком One-hot Encoder является существенное увеличение объема данных, так как признаки с большим количеством уникальных значений кодируются большим количеством бинарных признаков.

Binary Encoder

Идея данного метода заключается в кодировании номера значения признака в виде двоичной записи. Например, если у нас всего 7 уникальных значений, то первое будет закодировано как 001, а последнее – 111. Но в данном подходе отсутствует интерпретируемость данных как в One-hot.

Target Encoder

Основная цель данного метода заключается в использовании целевой метки, для кодирования категориальных признаков.

Для задачи регрессии Target Encoder использует среднее значение целевой метки по данному значению категориального признака.

Для задачи бинарной классификации использует вероятность единичного класса для данного значения категориального признака.

Leave-One-Out Encoder является расширением Target Encoder в котором кодирование конкретного объекта обучающей выборки не учитывает значение данного объекта при подсчете среднего/вероятности.

James-Stien является некоторым средневзвешенным между значениями для объектного значения категориального признака и значением для всей выборки.

Однако данный encoder определен только для задач регрессии и хорошо работает только в случае нормального распределения целевой метки.

1.1.4 Выбор признаков

Выбор признаков – это процесс выбора наиболее значимых признаков (факторов, переменных) из исходных данных для использования в модели машинного обучения. Целью выбора признаков является уменьшение размерности данных, улучшение качества модели и ускорение обучения.

Далее рассмотрим методы оценки важности признака.

1.1.4.1 Фильтрация признаков

Основная идея фильтрации признаков заключается в том, чтобы оценить важность каждого признака на основе некоторой метрики и выбрать только те, которые наиболее полезны для построения модели. Методы фильтрации, как правило, достаточно быстрые и имеют низкую стоимость вычислений.

Корреляция признаков

Существует множество метрик для оценки важности признаков. Одна из самых распространенных метрик – это корреляция Пирсона. Высчитывает коэффициент Пирсона следующим образом[16]:

$$r_x = \frac{n * \Sigma(x_i * y_i) - \Sigma x_i * \Sigma y_i}{\sqrt{(n * \Sigma x_i^2 - (\Sigma x_i)^2) - (n * \Sigma y_i^2 - (\Sigma y_i)^2)}}, \quad (4)$$

где n – количество наблюдений, xi – значения, принимаемые первой переменной, yi – значения, принимаемые второй переменной.

Корреляция измеряет линейную зависимость между двумя признаками. Если два признака сильно коррелируют, то один из них может быть удален, так

как он не добавляет дополнительной информации или, если большая корреляция между признаком и целевой переменной, то возможно он более информативен.

Chi-square

Тест, показывающий зависимость между переменными[7]:

$$x^2 = \sum \frac{(observed - expected)^2}{expected}, \quad (5)$$

где observed – число наблюдаемых событий, а expected – число ожидаемых.

Для независимых событий значение теста будет равно нулю.

1.1.4.2 Обертывающий метод

Обертывающий метод: этот метод основывается на обучении модели на каждом поднаборе признаков и выборе наилучшего поднабора, основываясь на результате работы модели. Популярные методы обертки[18]:

1) Прямой отбор. Этот метод начинается с самого эффективного признака и последовательно добавляет другие, которые дают наилучший результат в сочетании с предыдущими.

2) Выбор перебором. Целью перебора является последовательное рассмотрение влияние каждого признака на результат решения модели машинного обучения.

3) Исчерпывающий выбор. Суть метода в оценке каждого подмножества признаков. Метод просматривает все возможные комбинации признаков и возвращает наиболее эффективное подмножество.

Встроенный метод

Встроенный метод – метод отбора признаков, при котором процесс выбора признаков является частью процесса обучения модели. В отличие от обертывающих методов, в которых выбор признаков выполняется после обучения модели, во встроенных методах признаки отбираются в процессе обучения модели. Это позволяет снизить риск переобучения, поскольку модель выбирает только те признаки, которые максимально влияют на качество предсказаний.

Встроенные методы часто используют регуляризацию, которая штрафует модель за использование избыточных или неинформативных признаков. Это позволяет модели выбрать только наиболее значимые признаки для построения модели и уменьшить риск переобучения. Одним из наиболее популярных методов регуляризации является L1-регуляризация, которая приводит к отбору признаков путем установки нулевых весов для неинформативных признаков.

Еще одним популярным встроенным методом является метод главных компонент. Основная идея метода заключается в том, чтобы найти новое пространство признаков, в котором данные будут максимально разделимы. Это достигается путем нахождения линейных комбинаций исходных признаков, называемых главными компонентами, которые обладают максимальной дисперсией. Первая главная компонента находится таким образом, чтобы она объясняла максимальную долю дисперсии данных, вторая главная компонента находится таким образом, чтобы она была некоррелирована с первой главной компонентой и объясняла максимальную долю оставшейся дисперсии, и так далее.

Преимуществом метода главных компонент является то, что он позволяет снизить размерность данных, уменьшив количество признаков, не теряя при этом существенной информации. Это может быть особенно полезно в случае большого количества признаков или при наличии коррелированных признаков, которые могут затруднять анализ данных.

Недостатком метода может являться сложность интерпретации полученных главных компонент, особенно если они являются линейными комбинациями большого числа исходных признаков.

Другие встроенные методы: линейная регрессия с L1-регуляризацией (лассо), линейная регрессия с L2-регуляризацией (гребневая регрессия), метод опорных векторов (SVM) с L1-регуляризацией, решающие деревья и случайный лес.

1.1.5 Обработка выбросов

Выбросы — это значения, которые существенно отличаются от остальных значений в наборе данных. Выбросы могут появляться в данных вследствие ошибок измерения, ошибок при вводе данных, пропусков данных, и других причин. Наличие выбросов может искажать статистические метрики, такие как среднее значение и дисперсия, что может приводить к неверным выводам при анализе данных и обучении моделей машинного обучения[9].

Обработка выбросов включает в себя выявление и удаление выбросов, а также замену выбросов на более подходящие значения, например, на медиану или среднее значение без учета выбросов.

Для поиска выбросов в данных существует несколько методов. Один из них — это использование правила трёх сигм, которое основано на распределении данных. Согласно этому правилу, если значение признака находится дальше, чем на три стандартных отклонения от среднего, то оно может быть выбросом.

Еще один метод — это использование межквартильного расстояния (IQR). IQR определяется как разница между 75-м и 25-м процентилем выборки. При этом любое значение, которое находится за пределами границы, определяемой как $Q1 – 1.5 \times IQR$ и $Q3 + 1.5 \times IQR$, считается выбросом.

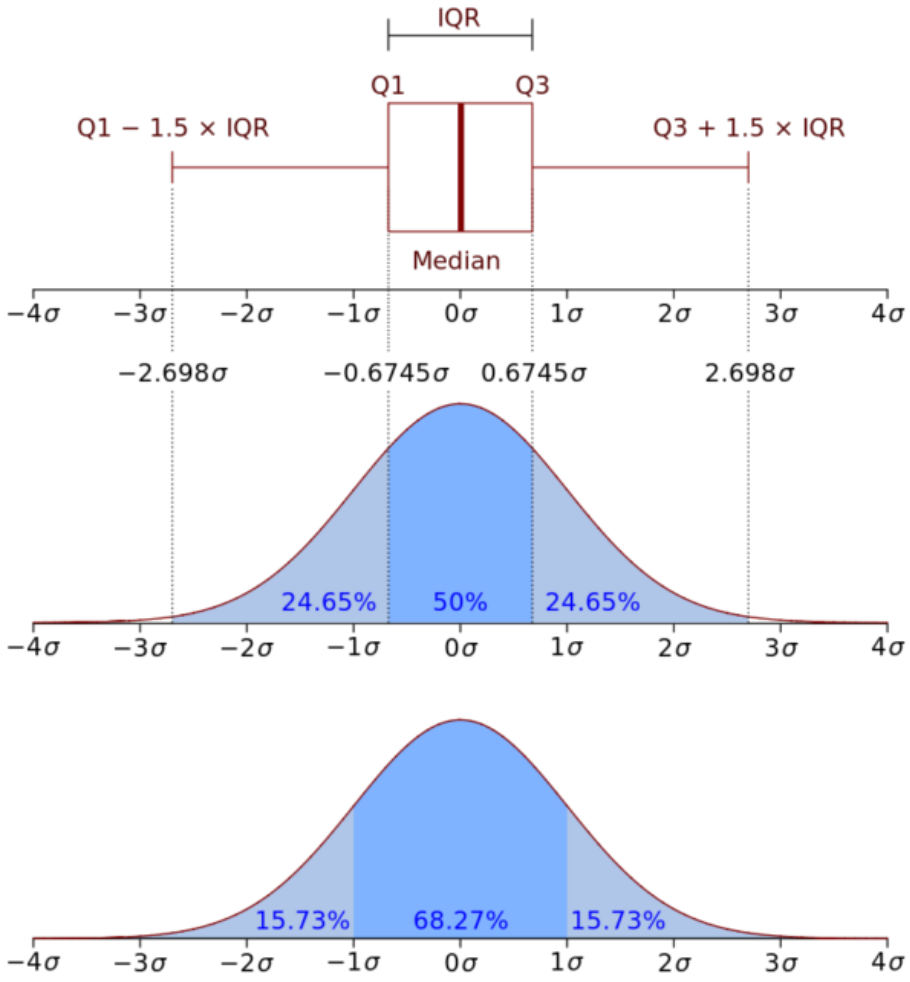

Рисунок 1 – IQR и нормальное распределение.

Также для поиска выбросов могут использоваться различные визуализации данных, например, box-plot, scatter-plot или histogram.

1.1.6 Обработка дисбаланса классов

Дисбаланс классов в задачах машинного обучения возникает, когда один класс представлен значительно меньшим количеством примеров, чем другой класс. Это может привести к смещению модели в сторону более представительного класса и плохим предсказаниям на классы с меньшим количеством примеров[1].

Существует несколько подходов к обработке дисбаланса классов:

1) Увеличение размера меньшего класса (oversampling): в этом методе используются техники генерации дополнительных примеров меньшего класса,

такие как SMOTE (Synthetic Minority Over-sampling Technique), ADASYN (Adaptive Synthetic Sampling) и другие.

2) Уменьшение размера большего класса (undersampling): в этом методе используются техники удаления случайных примеров из большего класса, такие как Random Under-sampling, Tomek Links и другие.

3) Использование взвешивания классов (class weighting): в этом методе используется модификация функции потерь, которая учитывает важность каждого класса в модели. Веса классов могут быть вычислены автоматически или заданы вручную.

4) Использование ансамблевых методов (ensemble methods): в этом методе используются комбинации моделей с разными параметрами и/или алгоритмами обучения для более точного предсказания меньшего класса.

1.1.7 Разбиение выборки

Финальным этапом предобработки данных является разбиение выборки на обучающую и тестовую. Разбиение выборки позволяет проверять модель на данных, которые не использовались при ее обучении, и оценивать, насколько хорошо модель может обобщать.

Обычно выборку могут разбивать на следующие подмножества[13]:

1) обучающий набор – подмножество данных, для обучения модели;

2) валидационная выборка – используется для контроля процесса обучения;

3) тестовый набор – подмножество данных для оценки производительности модели.

Далее рассмотрим методы разбиения[17]:

1) Простое случайное разбиение: выборка разбивается случайным образом на две части — обучающую и тестовую. Этот метод не гарантирует, что в обучающей и тестовой выборках будут присутствовать представители всех классов, поэтому часто используется более сложные методы разбиения.

2) Стратифицированное случайное разбиение: выборка разбивается на обучающую и тестовую таким образом, чтобы соотношение классов в обоих выборках было таким же, как и в исходной.

3) K-fold кросс-валидация: выборка разбивается на K частей, называемых фолдами. Затем модель обучается на K-1 фолдах, а тестирование производится на оставшемся фолде. Процедура повторяется K раз, каждый раз использование разные фолды для тестирования и обучения, и в результате получаются K оценок качества модели[8].

4) Stratified K-fold кросс-валидация: комбинация стратификации и k-fold кросс-валидации.

5) Leave-One-Out кросс-валидация: каждый пример выделяется в тестовую выборку по очереди, а обучение производится на всех остальных примерах. Таким образом, если в выборке n примеров, то производится n итераций, каждый раз используя n-1 примеров для обучения.

6) Leave-P-Out кросс-валидация: похожа на Leave-One-Out, но каждый раз выделяется не один, а p примеров в тестовую выборку.

1.2 Обзор программных средств для предобработки данных

Существует множество инструментов для предварительной обработки данных, в том числе и программ, которые могут автоматизировать процесс и облегчить работу исследователю данных. В данном обзоре мы рассмотрим несколько известных программ для предварительной обработки данных и оценим их возможности и преимущества.

1.2.1 RapidMiner

RapidMiner – это интегрированная среда обработки и анализа данных, которая позволяет пользователям извлекать, загружать, преобразовывать, визуализировать и анализировать данные из различных источников. Она позволяет проводить всю цепочку обработки данных от их подготовки до построения моделей машинного обучения и визуализации результатов.

RapidMiner предоставляет пользователю интерфейс визуального программирования без необходимости самостоятельно программировать. Весь функционал программного средства разбит на блоки, которые пользователь использует по необходимости, также программа имеет множество библиотек с разнообразным функционалом[14].

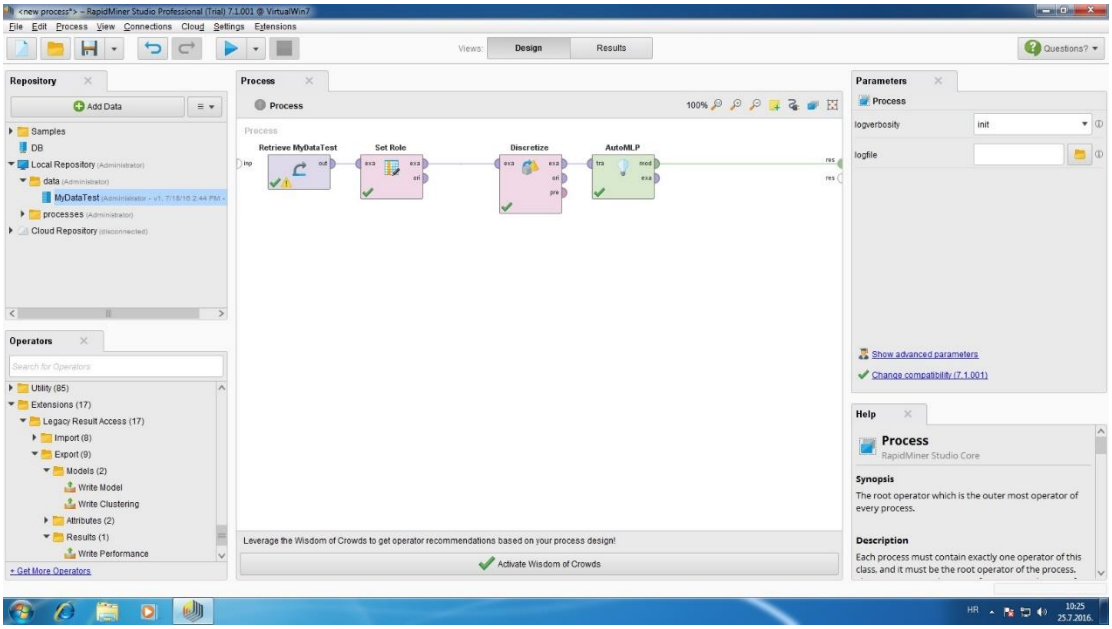

Рисунок 2 – Интерфейс RapidMiner.

1.2.2 Orange

Orange – это бесплатная среда визуального программирования с открытым исходным кодом, предназначенная для анализа данных и машинного обучения. Она разработана с помощью языка программирования Python и может быть использована как в виде графического интерфейса, так и в виде библиотеки Python.

Orange является мощным инструментом для анализа данных и машинного обучения, который предоставляет широкий набор функций и простой интерфейс для работы с данными. Он может быть использован как новичками в анализе данных, так и профессионалами в области машинного обучения[12].

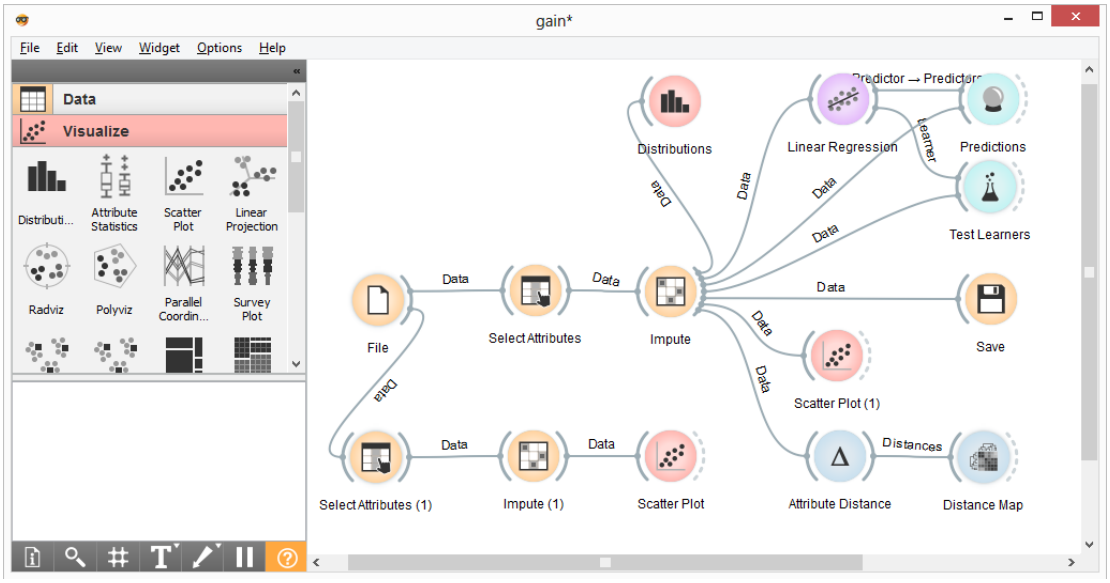

Рисунок 3 – Интерфейс Orange.

1.2.3 Alteryx

Alteryx – это программа для подготовки и анализа данных, которая предоставляет мощный инструментарий для обработки данных и создания рабочих процессов через графический интерфейс, включает подключение к источникам данных, очистку и преобразование данных, выполнение анализа данных и машинного обучения, а также визуализацию результатов[2].

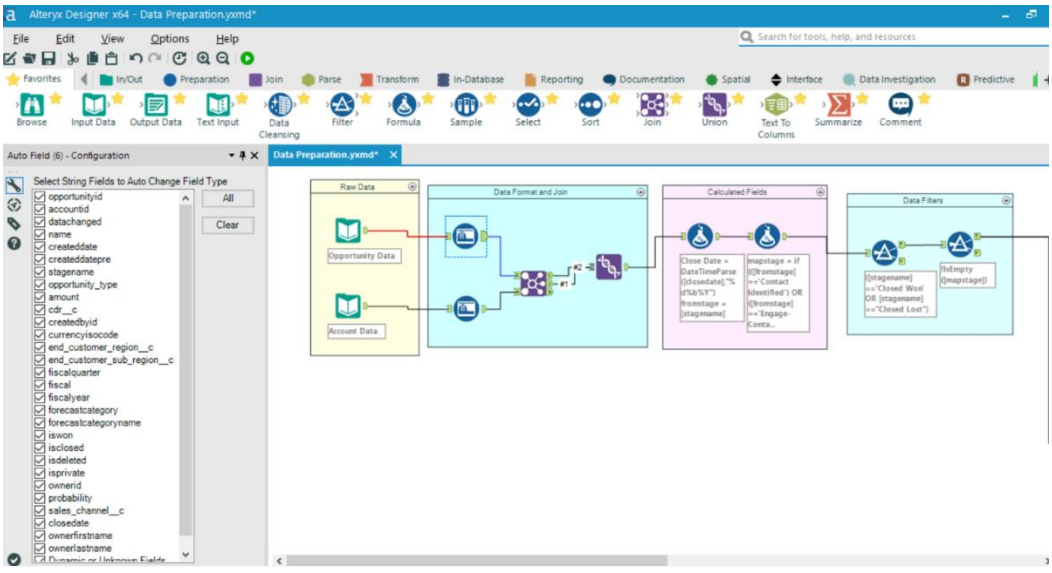

Рисунок 4 – Интерфейс Alteryx.

1.2.4 Плюсы и минусы программных средств

Alteryx, RapidMiner, и Orange – все это платформы для предварительной обработки и анализа данных. Плюсы и минусы этих программ приведены в таблице 1:

Таблица 1 – плюсы и минусы программ для предварительной обработки данных.

	RapidMiner	**Orange**	**Alteryx**
Функциональность	+	+/–	+
Визуализация	+	+	+
Бесплатность	–	–	–
Сложность интерфейса	–	–	–
Поддержка русского языка	–	–	–

В целом, каждая из этих платформ имеет свои преимущества и недостатки, и лучший выбор зависит от потребностей и требований конкретного проекта. Однако все они очень требовательны к знаниям в области обработки данных,

машинного обучения, интерфейса и возможностей самой программы и нет поддержки русского языка

1.3 Выводы по главе

В настоящий момент разработано множество программных средств для предварительной обработки данных, в них реализовано большинство методов и алгоритмов предобработки. Однако они слишком сложны для начинающего пользователя и нету полной автоматизации процесса. Следовательно, разработка системы для экспериментов по машинному обучению и, в частности, подсистемы предварительного анализа и предобработки данных является актуальной.

2 Анализ и построение модели предметной области

В данной главе описаны основные термины и понятия предметной области «Эксперименты в области машинного обучения». Представлено описание предметной области, выявлены объекты и построена модель, приведена формальная постановка задачи работы.

2.1 Формальные определения терминов

Выборка данных – данные, которые характеризуют некоторый классифицируемый объект, представленные в виде матрицы размерности NxM, где каждый столбец – это признак, а строка – объект(экземпляр), пересечения столбец-строка содержат значения признака для объекта. Значение может содержать пропуски, а совокупность значений признака принадлежит определенному типу данных.

Целевой признак (класс) – признак выборки данных, предсказание которого является целью эксперимента.

Пропущенные значения – пустые значения признака, либо содержащие Nan, Null.

Тип данных – данные, которые можно присвоить одну из следующих групп:

числовые – данные, состоящие только из вещественных чисел или пропущенных значений;

категориальные – остальные данные.

2.2 Глоссарий терминов

Выброс – Объект, значения признаков которого находятся далеко за пределами других объектов того же класса что и сам объект.

Обработка – изменение выборки данных, с целью указанной в критерии.

Статистика – данные характеризующие объекты или признаки.

Графики – визуализация статистик.

Корреляция – взаимосвязь двух признаков.

Дисбаланс классов – разница между количеством классов превышает определенное значение.

2.3 Методы обработки выбросов

Профессиональная деятельность, рассматриваемая в данной выпускной квалификационной работе – проведение экспериментов в области машинного обучения.

Целевая аудитория программной системы разделяется на следующие три группы пользователей: начинающие специалисты по машинному обучению, преподаватели по направлению машинного обучения и студенты. Начинающие специалисты могут использовать данное программное средство с целью быстрого получения самого эффективного алгоритма. Преподаватели по направлению машинного обучения могут использовать данное программное средство как инструмент для обучения студентов, позволяющий давать им домашнее задание и проверять результаты экспериментов. Студенты могут использовать данное программное средство с целью обучения через множественные эксперименты с алгоритмами и получения знаний о том, как влияет предобработка данных и различный гиперпараметры алгоритмов на качество обучения.

Для проведения экспериментов в машинном обучении первоначально загружается выборка данных, затем происходит предобработка этой выборки, каждый столбец обрабатывается определенным, выбранным пользователем образом. После предобработки выборка разбивается несколькими способами на обучающую и тестовую. Далее появляется возможно применить некоторые методы машинного обучения. Какие методы доступны определяется на основании типа данных полей выборки. После методов машинного обучения определяются границы для числовых гиперпараметров и выбираются не числовые гиперпараметры из списка допустимых для данного алгоритма, метрики качества для сравнения моделей между собой. После обучения и тестирования методов для каждого набора гиперпараметров определяется

результат работы алгоритма, затем выбирается лучший по соотношению времени обучения к результату. Результат решения задачи отдается пользователю.

2.4 Формальная поставка задачи предобработки выборки

Входные данные:

– выборка данных

– список методов предобработки выборки с их параметрами

– параметры разбиения выборки

Выходные данные:

– Выборка данных, обработанная в соответствии со списком методов предобработки и разбитая на обучающуюся и тестовую

Связь:

Данные выборки обрабатываются в соответствии с указанным списком методов предобработки выборки, который может включать в себя следующие методы: обработки пропусков, нормализации данных, кодировки категориальных признаков, фильтрации признаков, обработки выбросов, удаление признака, удаление значений, замена значений на null. После чего выборка разбивается на обучающуюся и тестовую в соответствии с параметрами разбиения выборки.

2.4.1 Формальная постановка задачи предварительного анализа данных

Входные данные:

– выборка данных

– список методов анализа данных

Выходные данные:

– статистика

– графики

Связь:

Данные выборки обрабатываются и на выходе предстают в виде статистической информации и графиков для анализа пользователем. Это могут быть следующая информация: количество каждого класса, диаграмма распределения классов, тепловая карта корреляции признаков, вывод всех типов данных выборки, количество пропущенных значений каждого признака и график распределения значений признака.

2.4.1.1 Посчитать классы

Входные данные:

– выборка данных

Выходные данные:

– количество классов и их названия

Связь:

– считаются разные значения в последнем столбце

2.4.1.2 Анализ распределения классов

Входные данные:

– выборка данных

Выходные данные:

– количество экземпляров каждого класса

– столбчатая диаграмма распределения классов

Связь:

– считаются строки матрицы с одинаковыми значениями в столбце целевых признаков

На Рисунок 5 представлен пример столбчатой диаграммы распределения классов. На оси x расположены названия классов, на оси y – количество строк(экземпляров) класса.

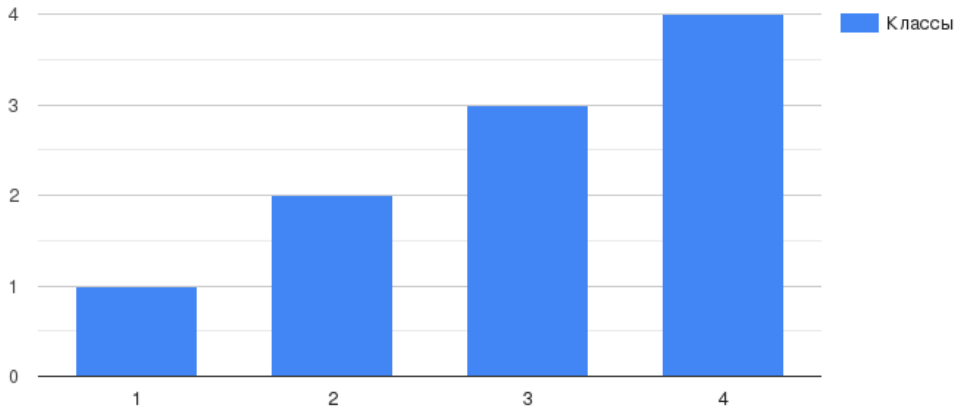

Рисунок 5 – пример столбчатой диаграммы распределения классов.

2.4.1.3 Корреляционный анализ признаков

Входные данные:

– выборка данных

Выходные данные:

– значение корреляции между каждыми двумя столбцами

– тепловая карта корреляции столбцов

Связь:

– считается корреляция между каждыми двумя столбцами

На Рисунок 6 представлен пример тепловой карты корреляции, по оси x и y расположены названия классов, на их пересечении – значение корреляции между ними.

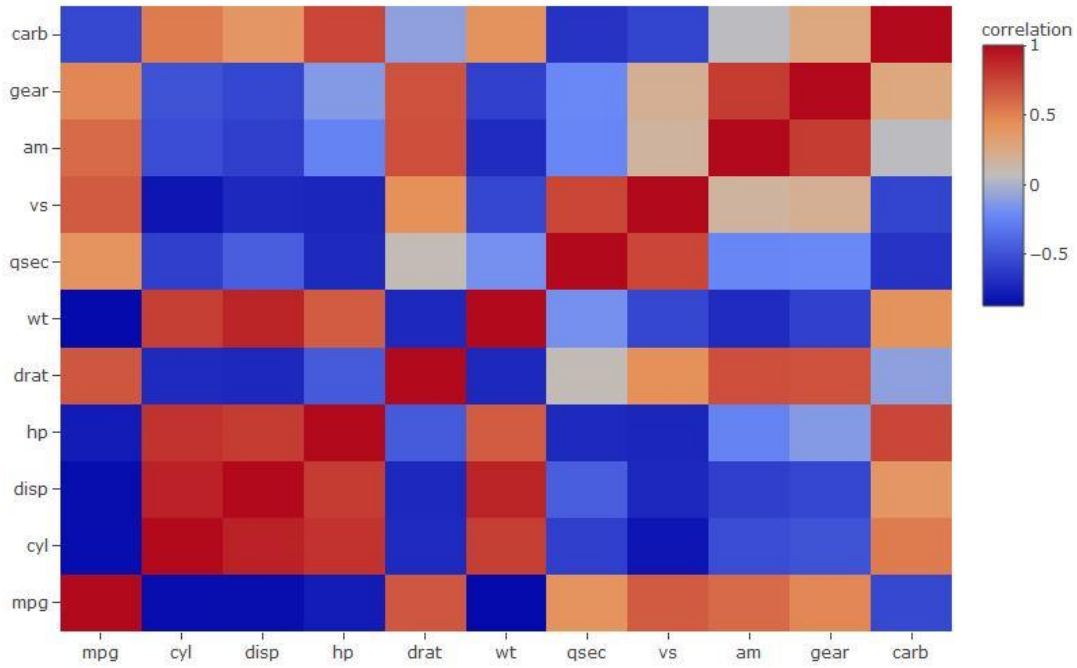

Рисунок 6 – Пример тепловой карты корреляции.

2.4.1.4 Анализ типов данных

Входные данные:

– выборка данных

Выходные данные:

– тип данных каждого столбца

Связь:

– типы данных делятся на: числовые, интервальные, бинарные, категориальные, текстовые.

2.4.1.5 Посчитать пропущенные значения для каждого признака

Входные данные:

– выборка данных

Выходные данные:

– количество пропущенных значений для каждого признака

Связь:

– считается пропущенные значения для каждого столбца

2.4.1.6 Анализ распределения признака

Входные данные:

– выборка данных

Выходные данные:

– столбчатая диаграмма распределения признака

Связь:

– фиксируется столбец и считается количество строк с одинаковым значением столбца с последующим выводом на график

На Рисунок 7 представлен пример распределения признака. По оси x расположены значения признака, а на оси y – количество объектов.

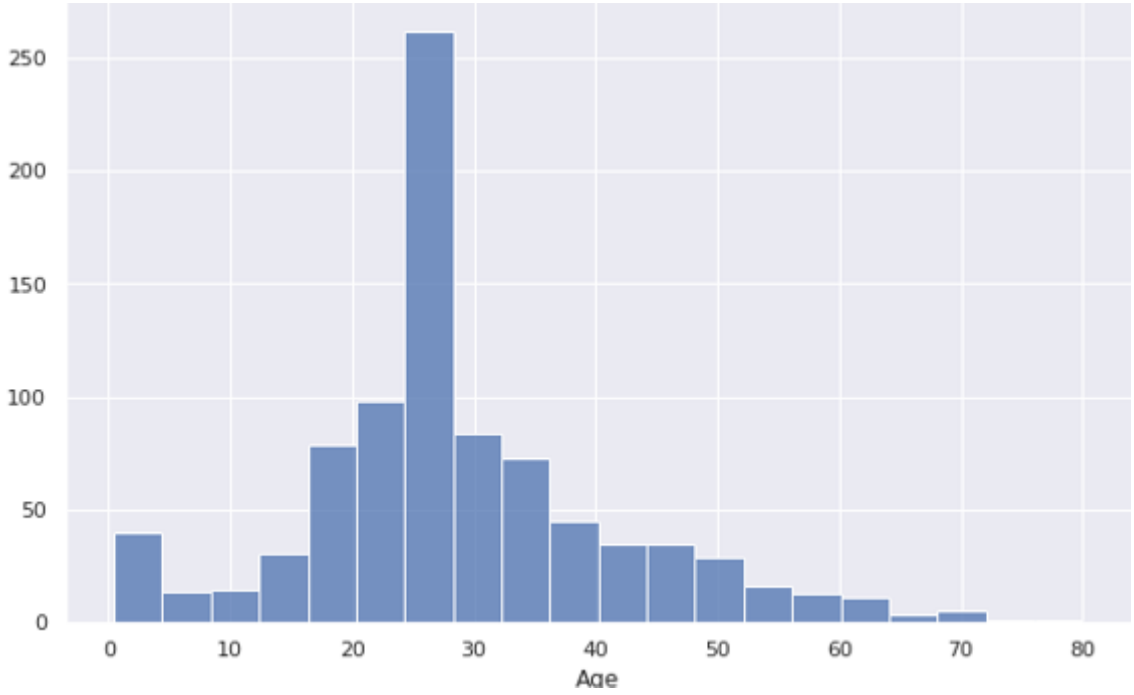

Рисунок 7 – Пример распределения признака.

2.4.2 Формальная постановка задачи предобработки данных

Входные данные:

– выборка данных

– методы предобработки данных

Выходные данные:

– выборка данных, обработанная в соответствии с выбранными методами предобработки.

Связь:

– данные выборки обрабатываются в соответствии со списком методов предобработки.

2.4.2.1 Удаление пропущенных значений

Входные данные:

– выборка данных

– критерий: объекты с пропущенными значениями удаляются

Выходные данные:

– выборка данных

Связь:

– строки либо столбцы выборки, в которых есть пропущенные значения, удаляются из выборки

2.4.2.2 Замена пропущенных значений

Входные данные:

– выборка данных

– критерий: распределение признаков после заполнения пропущенных значений должно стать ближе к нормальному

Выходные данные:

– выборка данных

Связь:

– строки, в столбцах которых есть пропущенные значения, заменяются используя методы замены пропущенных значений, которые включают замену на: среднее, моду, медиану

2.4.2.3 Обработка выбросов

Входные данные:

– выборка данных

– критерий: в выборке не должно остаться выбросов

Выходные данные:

– выборка данных

Связь:

– объекты-выбросы обрабатываются методами обработки выбросов, которые включают в себя: IQR, стандартное отклонение

2.4.2.4 Отбор признаков

Входные данные:

– выборка данных

– число N

– критерий: в выборке должно остаться только N признаков

Выходные данные:

– выборка данных

Связь:

– признаки анализируются с помощью методов оценки важности признаков и выбирается N наиболее значимых. Методы оценки: корреляция, кси-квадрат

2.4.2.5 Нормализация признаков

Входные данные:

– выборка данных

– критерий: значения между признаками должны быть в едином масштабе

Выходные данные:

– выборка данных

Связь:

– столбцы нормализуются методами нормализации, которые включают в себя: десятичную нормализацию, min-max, максимальное абсолютное отклонение

2.4.2.6 Кодировка категориальных признаков

Входные данные:

– выборка данных

– критерий: замена значений признаков на числовые

Выходные данные:

– выборка данных

Связь:

– столбцы категориальных признаков кодируются методами кодирования, которые в себя включают следующие методы: Label encoding, One-hot encoding, Binary encoding

2.4.2.7 Обработка дисбалансных классов

Входные данные:

– выборка данных

– критерий: разница между количеством объектов классов в заданных пределах

Выходные данные:

– выборка данных

Связь:

– классы обрабатываются методами обработки дисбалансных классов, путем оставление равного количества классов

2.4.2.8 Разбиение выборки

Входные данные:

– выборка данных

– параметры разбиения выборки

Выходные данные:

– обучающая выборка

– тестовая выборка

Связь:

– выборка разбивается, согласно параметрам разбиения выборки

2.5 Выводы по главе

Таким образом, в главе были описаны формальные постановки задач для предварительного анализа данных и предобработки данных.

3 Технический проект программного средства

Программная система для проведения экспериментов по машинному обучению состоит из трех основных подсистем: предобработки данных, организации экспериментов и работы с алгоритмами. В данной главе приведено описание всей системы и подсистемы предобработки данных.

3.1 Спецификация требований к программной системе

Спецификация требований к программной системе содержит основные функциональные, пользовательские и системные требования. В системе предполагается участие двух типов пользователей: администратора и пользователя.

3.1.1 Функциональные требования

Требования к подсистеме организации экспериментов:

Ф.01– хранение профилей пользователей;

Ф.02 – хранение выборок данных;

Ф.03 – хранение алгоритмов машинного обучения;

Ф.04 – хранение результатов экспериментов;

Ф.05 – загружать выборку данных;

Ф.06 – должна позволять пользователю регистрироваться в системе;

Ф.07 – должна позволять пользователю входить в систему под своим именем;

Требования к подсистеме предобработки данных:

Ф.08 – проводить предварительный анализ данных в выборке;

Ф.09 – проводить предобработку выборки данных;

Ф.10 – анализировать доступность алгоритмов выборки данных;

Ф.11 – выполнять разбиение выборки данных на обучающую и контрольную;

Требования к подсистеме работы с алгоритмами:

Ф.12 – загружать новые алгоритмы машинного обучения в систему;

Ф.13 – запускать алгоритмы машинного обучения с фиксированными гиперпараметрами

Ф.14 – запускать алгоритмы машинного обучения используя подбор гиперпараметров;

Ф.15 – получать результаты работы алгоритмов машинного обучения;

Ф.16 – демонстрировать лучший из алгоритмов по заданным метрикам.

3.1.2 Пользовательские требования

П.01 – Программное средство должно вести диалог с пользователем в терминах ПО;

П.02 – Программное средство должно быть на русском языке, кроме случаев, если название метода из машинного обучения не имеет перевода;

П.03 – Программное средство должно быть выполнено в материальном дизайне;

П.04 – Программное средство не должно потреблять большое количество интернет-трафика;

3.1.3 Системные требования

С.01 – серверная часть ПС должна работать на компьютере с поддержкой виртуализации и установленной программой Docker версии 22.

С.02 – клиентская часть должна работать в современных браузерах с включенной поддержкой java script.

3.2 Архитектурно-контекстная диаграмма

Архитектурно-контекстная диаграмма отражает интерфейсы системы с внешним миром, а именно, информационные потоки между системой и внешними сущностями, с которыми она должна быть связана. Она идентифицирует эти внешние сущности, а также, процессы, отражающие главные цели или природу системы насколько это возможно.

На рисунке 8 представлена архитектурно-контекстная диаграмма разрабатываемой программной системы, она состоит из трех подсистем. Пунктиром выделена часть, которая рассматривается в данной выпускной квалификационной работе.

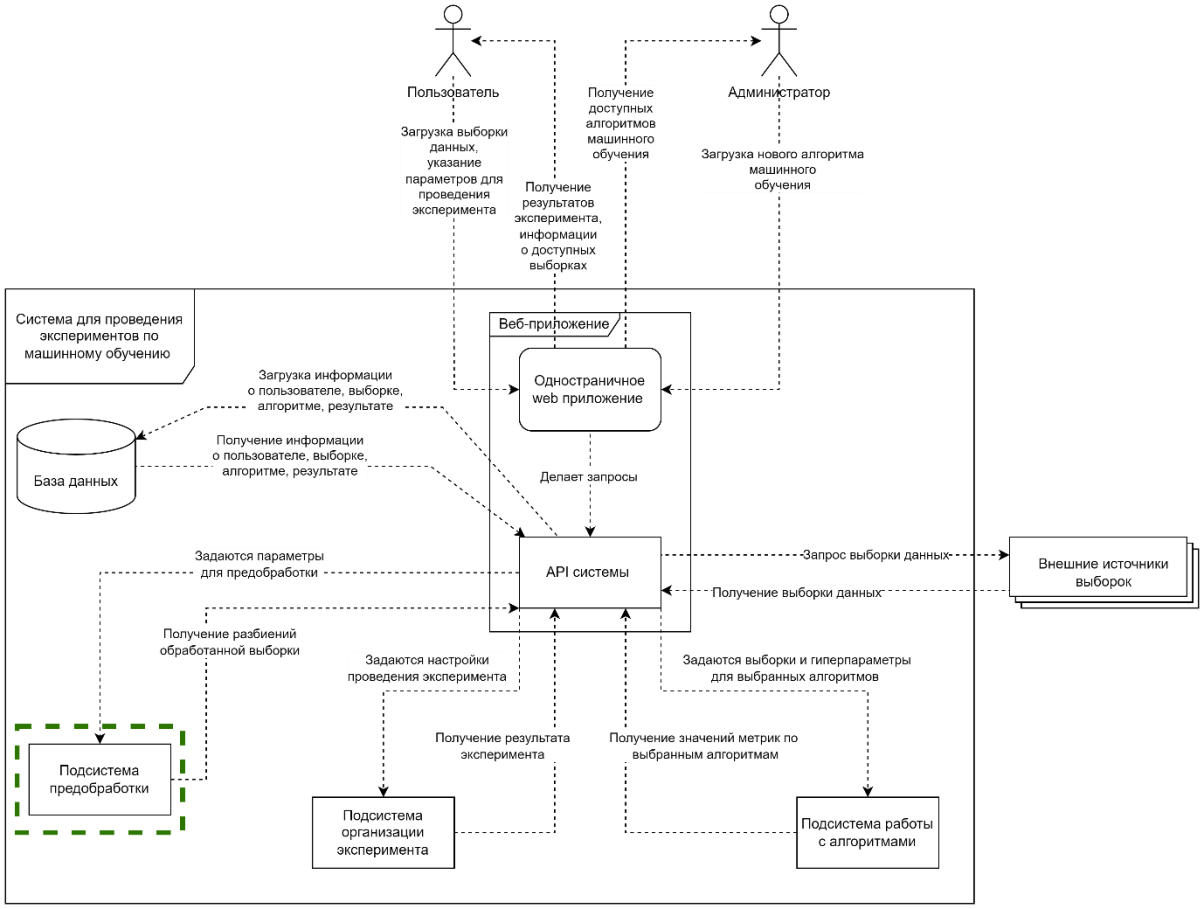

Рисунок 8 – Архитектурно-контекстная диаграмма системы.

3.3 Детализация подсистемы

На рисунке 9 представлена архитектурно-контекстная диаграмма подсистемы предобработки данных.

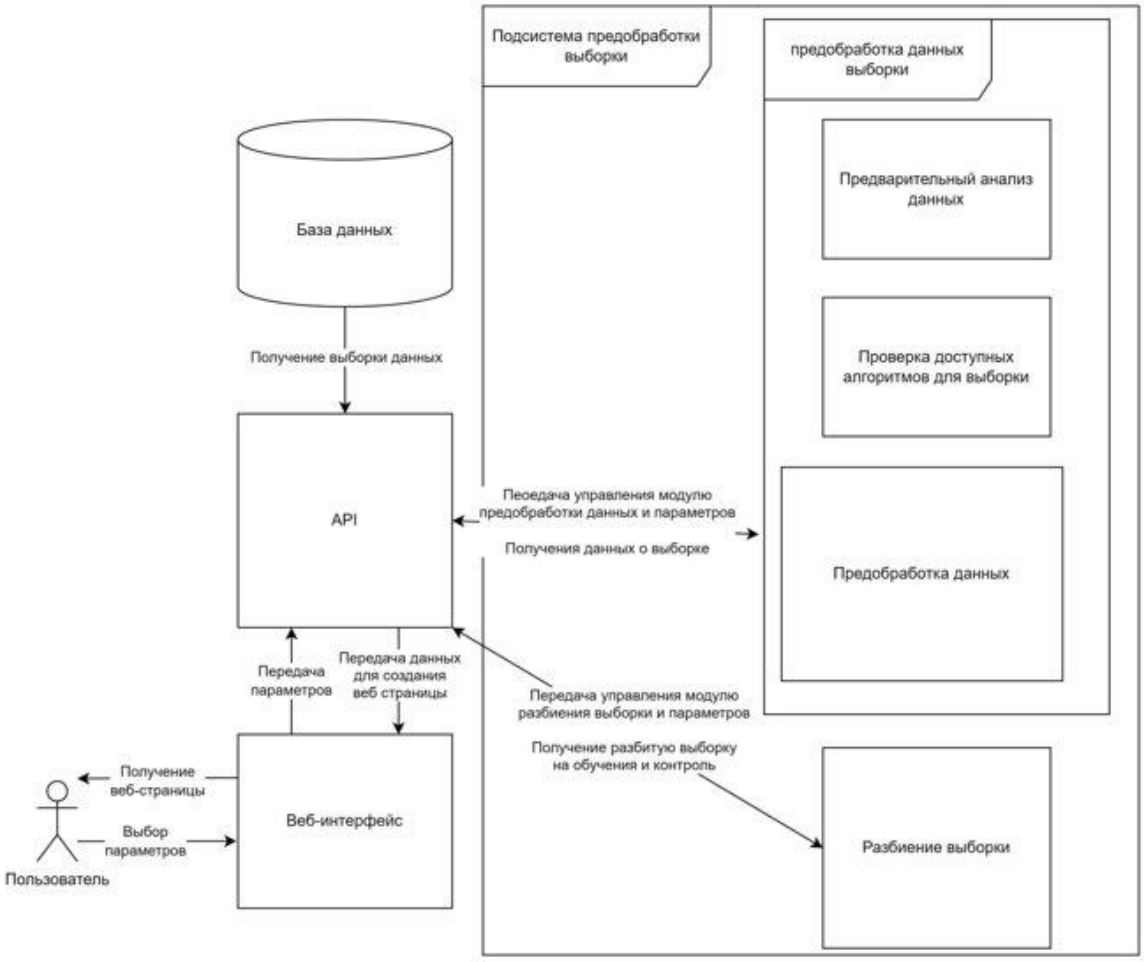

Рисунок 9 – Архитектурно-контекстная диаграмма подсистемы предобработки данных

3.4 Использования системы

Систему используют следующие группы людей:

1. Пользователь – человек, имеющий возможность регистрироваться, авторизироваться в системе, загружать выборки данных в систему, проводить эксперименты по машинному обучению, а также просматривать результаты проведенных экспериментов и статистику по ним.

2. Администратор – человек, обладающий всеми возможностями пользователя, помимо регистрации(регистрация администраторов проводится вручную), а также способный добавлять и удалять алгоритмы в системе.

В данной работе разрабатывается подсистема предобработки данных. В ней пользователь может: выбрать целевой признак(определяемый класс), определить спецсимволы, которыми обозначаются пропущенные значения,

удалить признак, выбрать метод для следующих типов обработки: обработка пропущенных значений, кодирование категориальных признаков, нормализация данных, обработка выбросов, фильтрация признаков. После выбора предобработок и подтверждения, пользователь переходить на вкладку разбиения выборки, где может выбрать: количество разбиений для выборки, коэффициент случайности и возможность соблюдения баланса классов для разбиений, после чего данные и пользователь переходят дальше в API

На рисунке 10 представлена use-case диаграмма использования системы пользователем.

На рисунке 11 представлена use-case диаграмма использования системы администратором.

На рисунке 12 представлена use-case диаграмма использования подсистемы предобработки данных.

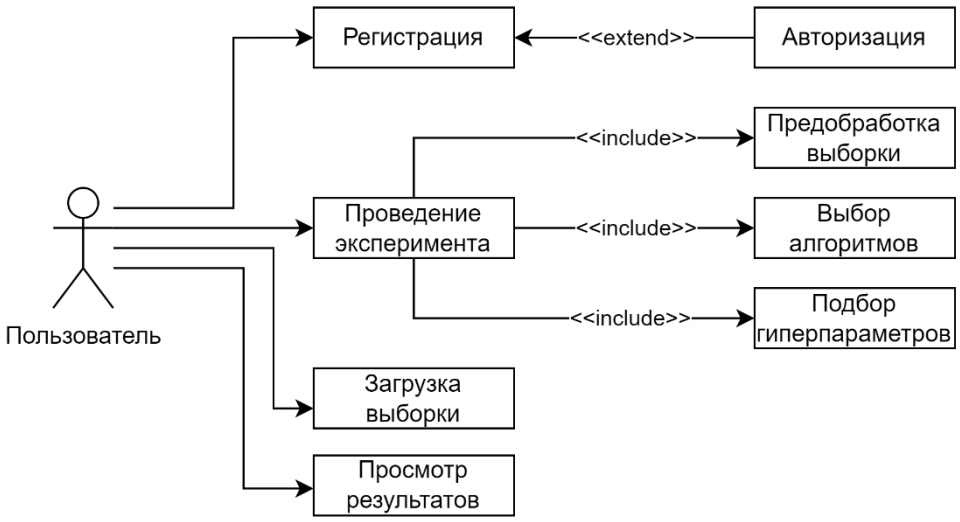

Рисунок 10 – Use-case диаграмма использования пользователем системы

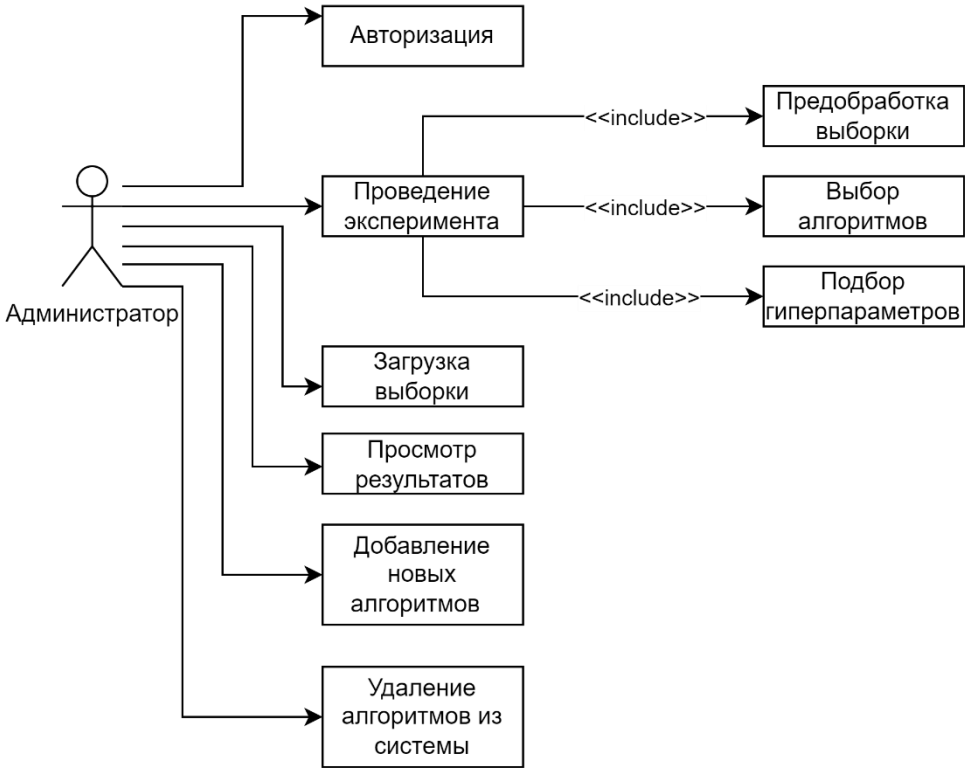

Рисунок 11 – Use-case диаграмма использования администратором системы

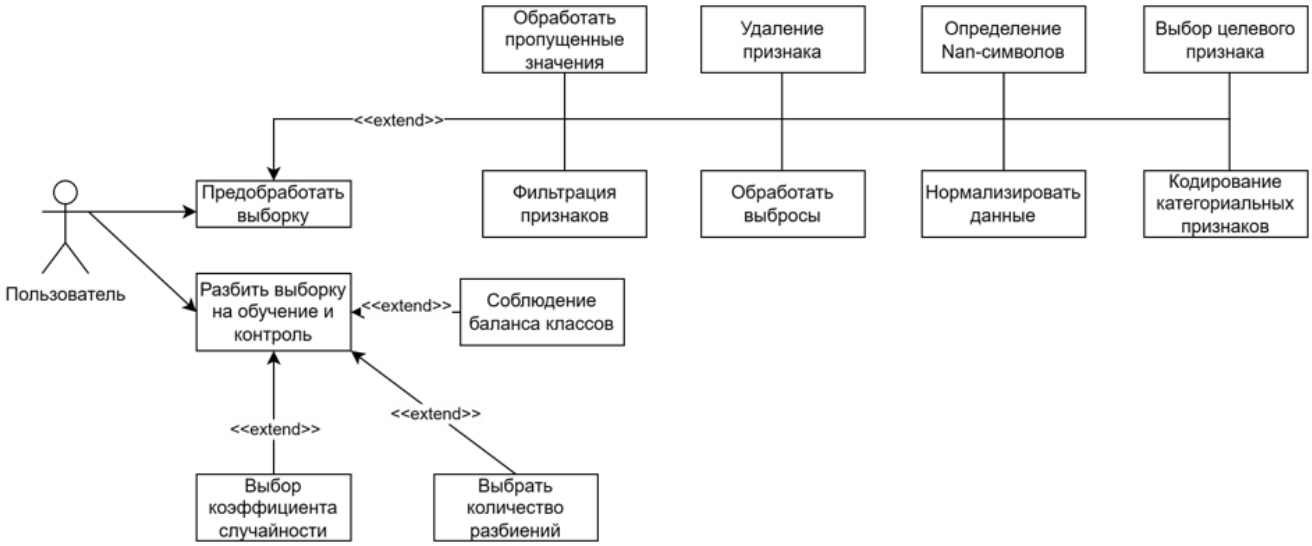

Рисунок 12 – Use-case диаграмма использования подсистемы предобработки данных

3.5 Диаграмма потоков данных

На рисунке 13 представлена диаграмма потоков данных подсистемы предобработки данных.

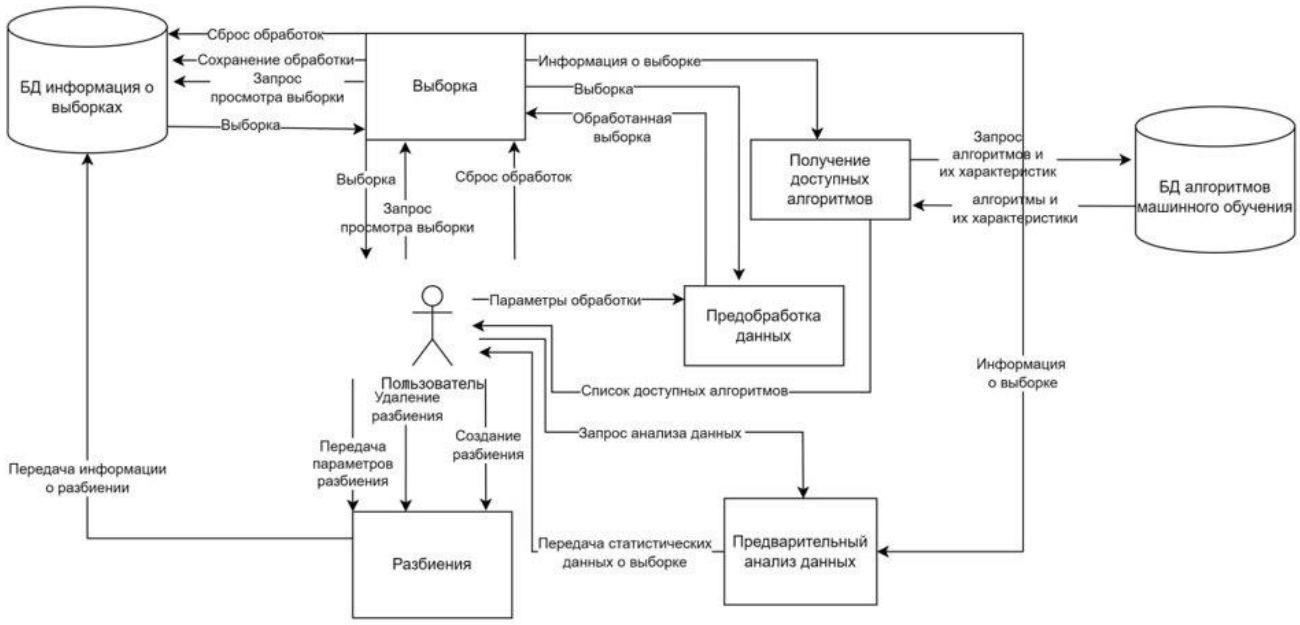

Рисунок 13 – диаграмма потоков данных подсистемы предобработки данных.

3.6 Проект верхнего уровня

Далее будут рассмотрены структура базы данных и проект интерфейса программного средства.

3.6.1 Структура базы данных

В качестве базы данных системы была выбрана реляционная модель данных. На рисунке 14 представлена схема структуры базы данных. В результате анализа ПО в подсистеме были использованы следующие таблицы:

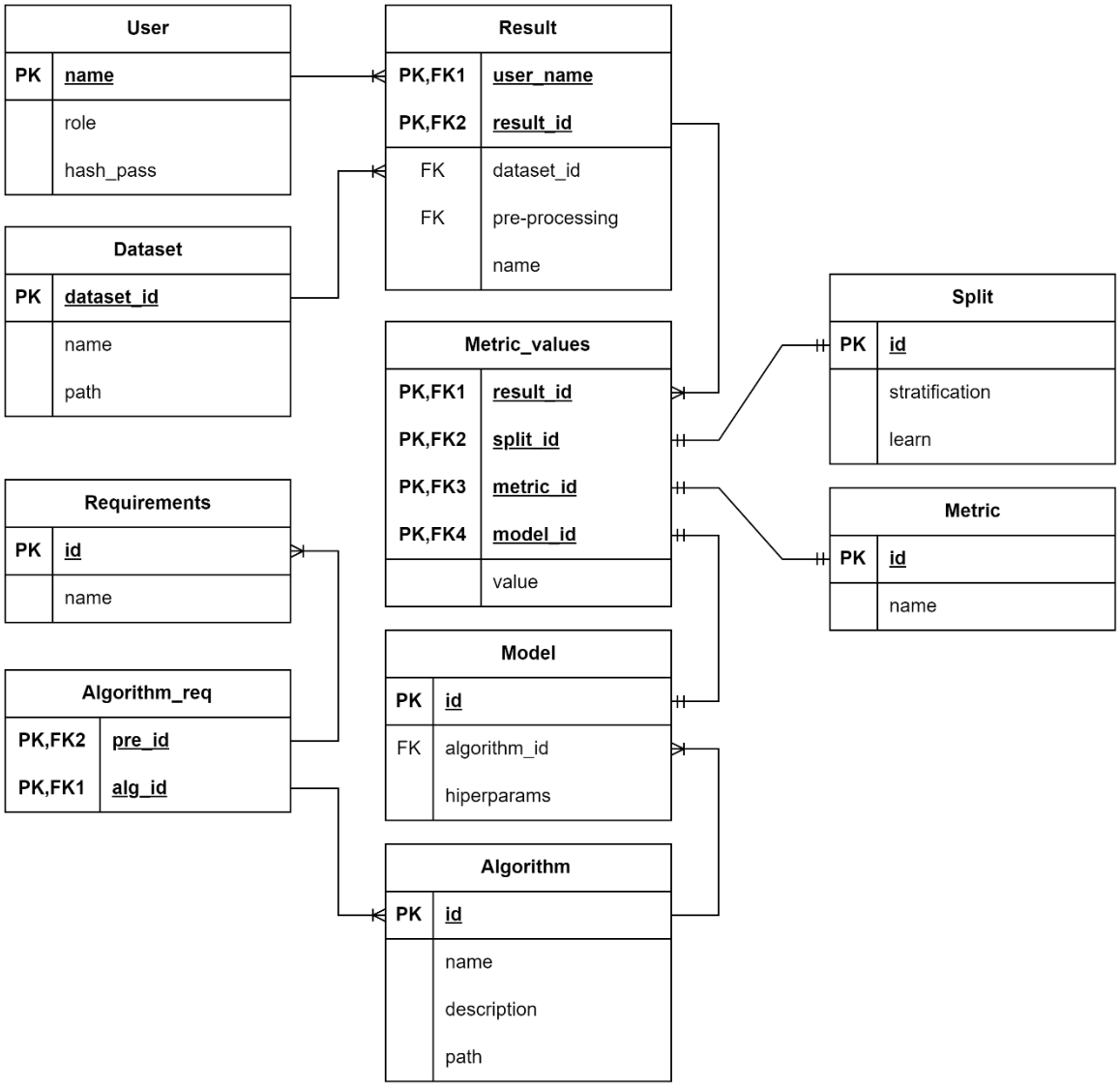

Рисунок 14 – схема структуры базы данных

3.6.1.1 Таблица «Выборка данных»

Хранит информацию о выборках данных. Имеет следующие столбцы:

1. id (обязательный) – идентификатор выборки данных, используемый для однозначной идентификации выборки в системе:
   a. тип значения – целое число;
   b. диапазон допустимых значений – от 1 до 2^{64}-1;
2. name (обязательный) – название выборки данных используемое для предоставления пользователю:
   a. тип значения – текст;
   b. диапазон допустимых значений – от 4 до 20 символов;

3. path (обязательный) – путь для сохранения выборки данных на сервере:

a. тип значения – текст;

b. диапазон допустимых значений – от 1 до 100 символов

3.6.1.2 Таблица «разбиение»

Хранит информацию о разбиении выборки данных на обучающую и контрольную. Имеет следующие столбцы:

1. id (обязательный) – идентификатор разбиения, используемый для однозначной идентификации разбиения в системе:

a. тип значения – целое число;

b. диапазон допустимых значений – от 1 до 2^{64}-1;

2. learn (обязательный) – процент выборки на которой обучается алгоритм:

a. тип значения – целое число;

b. диапазон допустимых значений – от 1 до 99;

3. stratification (обязательный) – индикатор того, что для разбиения использовалась стратификация:

a. тип значения – целое число;

b. диапазон допустимых значений – {1, если стратификация использовалась, 0, в ином случае};

3.6.1.3 Таблица «требования к выборке»

Хранит требования для запуска алгоритма машинного обучения. Имеет следующие столбцы:

1. id (обязательный) – идентификатор требования, используемый для однозначной идентификации требования в системе:

a. тип значения – целое число;

b. диапазон допустимых значений – от 1 до 2^{64}-1;

2. name (обязательный) – человекочитаемое название требования для обозначения в системе:

a. тип значения – текст;

b. диапазон допустимых значений – от 4 до 20 символов;

3.6.1.4 Таблица «результат эксперимента»

Хранит информацию о результатах экспериментов, которые провел пользователь. Имеет следующие столбцы:

1. result_id (обязательный) – часть составного ключа идентификатор, используемый для однозначной идентификации эксперимента в системе:

a. тип значения – целое число;

b. диапазон допустимых значений – от 1 до 2^{64}-1;

2. user_name (обязательный) – часть составного ключа, идентификатор пользователя, который проводил эксперимент.

a. тип значения – текст;

b. диапазон допустимых значений – от 4 до 20 символов;

3. dataset_id (обязательный) – идентификатор выборки данных, каждый эксперимент привязан к выборке данных на которой он проводился:

a. тип значения – целое число;

b. диапазон допустимых значений – от 1 до 2^{64}-1;

4. name (обязательный) – человекочитаемое название эксперимента для обозначения в системе:

a. тип значения – текст;

b. диапазон допустимых значений – от 4 до 20 символов;

5. pre-processing (обязательный) – информация о том, какие методы предобработки были применены к выборке данных:

a. тип значения – JSON;

3.6.1.5 Таблица «требования алгоритма»

Хранит связи между требованиями к выборке и алгоритмам машинного обучения. Имеет следующие столбцы:

1. pre_id (обязательный) – часть составного ключа, идентификатор требования, которое необходимо для запуска алгоритма:
   a. тип значения – целое число;
   b. диапазон допустимых значений – от 1 до 2^{64}-1;
2. alg_id (обязательный) – часть составного ключа, идентификатор алгоритма, для которого описано требование:
   a. тип значения – целое число;
   b. диапазон допустимых значений – от 1 до 2^{64}-1;

3.6.2 Проект интерфейса

Система использует веб-интерфейс для взаимодействия с пользователем. Чтобы попасть на страничку предобработки данных необходимо начать эксперимент и выбрать или загрузить набор данных, с которыми будет производиться работа. Перейти на начало эксперимента можно нажатием на область, выделенную зеленой рамкой на рисунке 15.

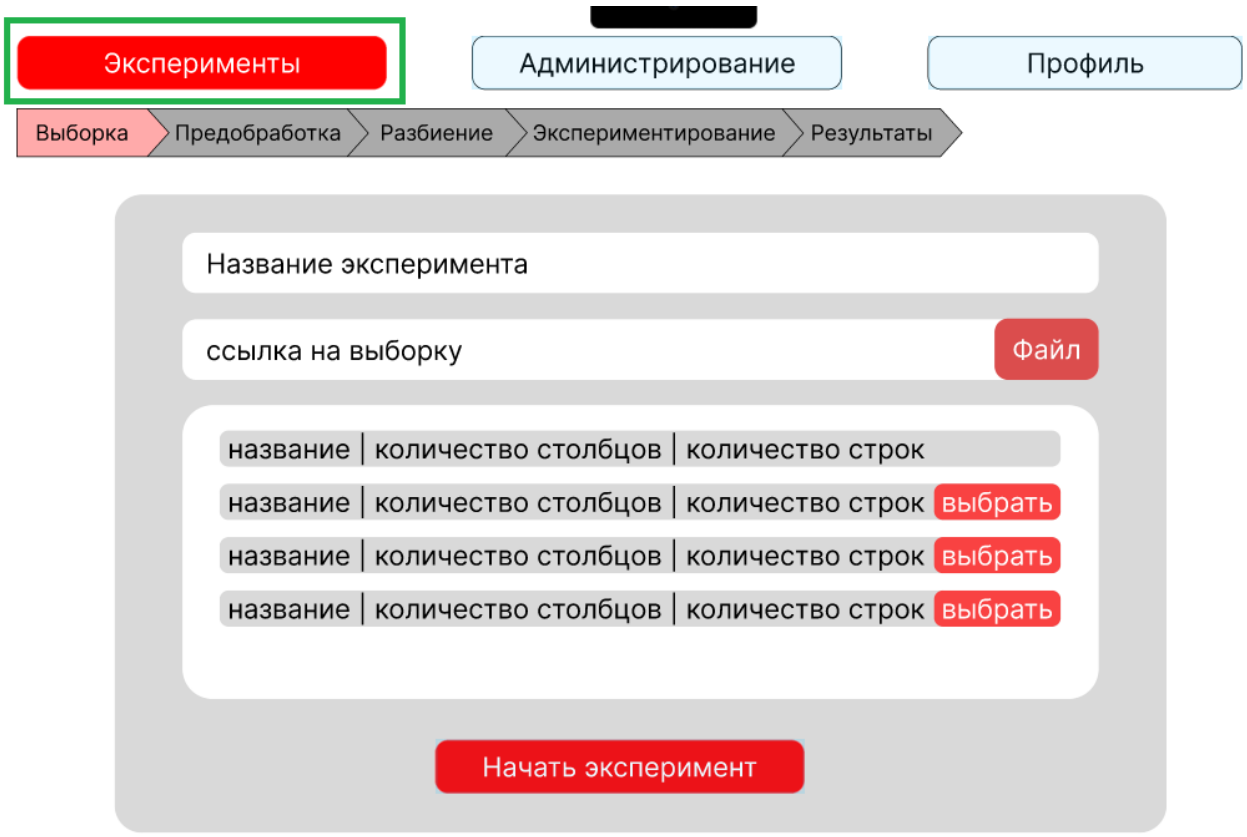

Рисунок 15 – страница выбора набора данных

Далее выбираем выборку данных, с которой будем работать либо из списка этих выборок, либо подгружаем свою нажимая на кнопку “Файл”, после чего переходим на страничку предобработки данных путем нажатия на кнопку “Начать эксперимент”. На рисунке 16 показана страница предобработки данных.

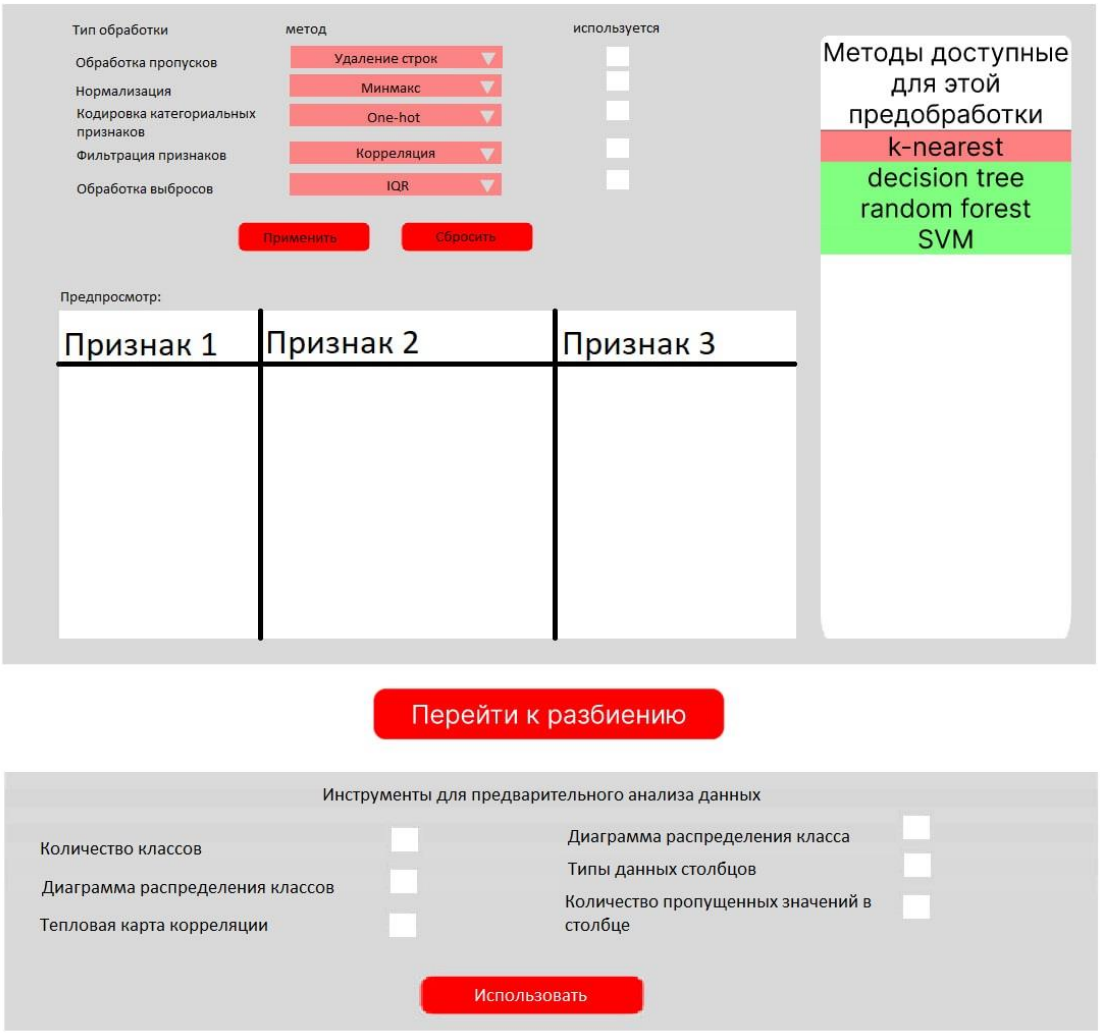

Рисунок 16 – страница предобработки данных

На этой странице ниже надписи предпросмотр расположена таблица с данными, которые используются в текущем эксперименте. В верхней части страницы расположены типы и методы предобработок, которые можно выбрать и поставить галочку для использования их. После чего можно нажать кнопку “Применить” для изменения набора данных и обновления таблицы предпросмотра или нажать кнопку “Сбросить”, для возвращения данных в состояния после загрузки. В правой части страницы зеленым цветом выделены алгоритмы, которые можно использовать при текущих данных. В нижней части страницы расположены инструменты для анализа данных, которые можно включить, выбрав соответствующие галочки и нажав кнопку использовать.

После нажатия кнопки “Перейти к разбиения” мы перемещаемся на страницу разбиений выборки, которая проиллюстрирована на рисунке 17.

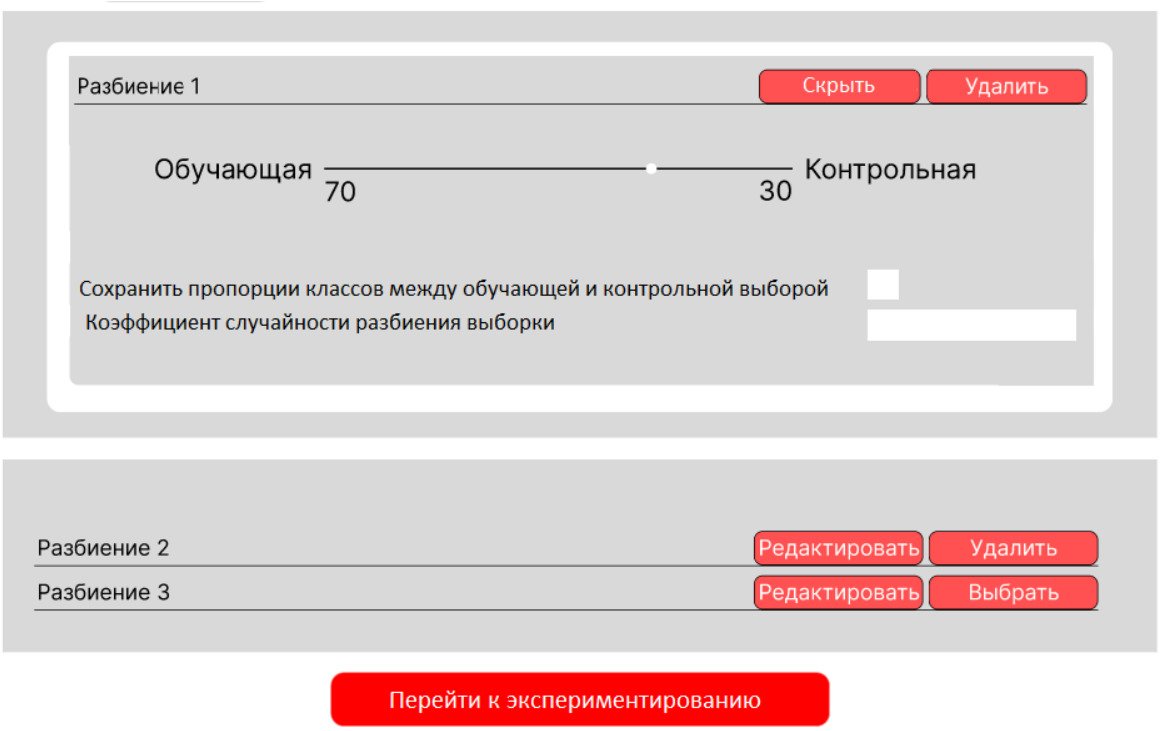

Рисунок 17 – страница разбиения выборки

На странице разбиения выборки можно создать от 1 до 3 разбиений, разбиение создается или редактируется путем нажатия кнопки “Редактировать” и удаляется нажатием кнопки “Удалить”. В каждом разбиении можно выбрать соотношения объектов в обучающей и контрольной выборки путем перетягивания ползунка, включить сохранение пропорций классов, и выбора коэффициента случайности в соответствующих полях. После настройки разбиений для перехода на следующий этап эксперимента нажимается кнопка “Перейти к экспериментированию”.

3.7 Выводы по главе

В данной главе описано устройство программного средства и, в частности, подсистемы предобработки данных на разных уровнях

4 Разработка и тестирование программного средства

В данной главе описаны инструменты и технология реализации подсистемы предобработки данных. Также представлены результаты ее тестирования и экспериментального исследования, а также сделаны выводы о качестве ее работы, в соответствии требованиями.

4.1 Инструменты разработки

4.1.1 Язык программирования

Для разработки был выбран язык программирования Python, из-за следующих преимуществ, которые он предоставляет:

- динамическая типизация данных;
- огромное количество библиотек и ресурсов для работы с данными;
- простота синтаксиса;
- интерпретируемость;
- много учебных материалов.

4.1.2 Среда разработки

В качестве среды разработки был выбран IDE PyCharm из-за опыта работы с ним и большого функционала:

- авто форматирование структуры кода;
- подсветка синтаксиса;
- удобная загрузка зависимостей;
- интегрированная работа с гитом;
- возможность интегрировать docker.

4.1.3 Разработка интерфейса

Для разработки интерфейса был использован сервис Figma. Figma — кроссплатформенный графический онлайн-редактор для совместной работы. Программа позволяет создавать wireframe, UI, прототипы, презентации и с

лёгкостью передавать материалы в разработку. В онлайн-режиме можно наблюдать рабочий процесс, оставлять комментарии и обсуждать макет[19].

4.1.4 Библиотеки

Для разработки подсистемы предобработки данных использовались следующие библиотеки:

- Pandas – библиотека, предоставляющая удобные структуры данных, имеющие большое количество операций. Стандартная библиотека для работы с структурированными данными, коей является выборка.
- Numpy – мощный инструмент для работы с многомерными данными, основа для Pandas.
- Scikit-learn – популярная библиотека для машинного обучения, библиотека содержит различные алгоритмы и методы анализа и обработки данных.
- Seaborn – библиотека для создания статистических графиков на python, также имеет немало методов предобработки данных.
- Matplotlib – библиотека для визуализации данных, имеющая обширный функционал.

4.2 Тестирование программного средства

В таблице 2 представлен набор тестов подсистемы предобработки выборки.

Таблица 2 – набор тестов для подсистемы предобработки выборки

№	Тестовая ситуация	Начальное состояние системы	Действие	Ожидаемый результат
1	Показ выборки данных	Страница выбора набора данных для эксперимента	Пользователь переходит на страницу предобработки данных	Отображаются все столбцы и первые 5 строк из выборки данных

Продолжение таблицы 2

№	Тестовая ситуация	Начальное состояние системы	Действие	Ожидаемый результат
2	Показ доступных алгоритмов машинного обучения для выборки	Страница выбора набора данных для эксперимента или страница предобработки данных	Пользователь переходит на страницу предобработки данных или обновляет ее	Отображается список доступных и недоступных алгоритмов машинного обучения для текущего состояния выборки
3	Отдельное тестирование методов каждого метода предобработки	Страница предобработки данных	Пользователь выбирает по очереди методы предобработки для выборки и нажимает кнопку “Применить”	Выборка данных обрабатывается выбранным методом обработки. Отображаемая выборка заменятся обработанной. Меняется список подходящих алгоритмов машинного обучения.
4	Тестирование нескольких и всех методов предобработки сразу	Страница предобработки данных	Пользователь выбирает несколько методов предобработки и нажимает кнопку “Применить”	Выборка данных обрабатывается выбранными методами обработки. Отображаемая выборка заменятся обработанной. Все алгоритмы доступны в списке.
5	Тестирование отмены предобработок	Страница предобработки данных	Пользователь нажимает кнопу “Сбросить”	Выборка данных заменяется изначально загруженной в систему
6	Тестирование методов предварительного анализа данных	Страница предобработки данных	Пользователь выбирает методы анализа данных и нажимает кнопку “Анализировать”	Выводится информация и графики, соответствующие выбранным методам анализа
7	Тестирование перехода к разбиению выборки	Страница предобработки данных	Пользователь подтверждает выбранные обработки нажатием кнопку “Перейти к разбиению”	Обработанная выборка загружается в систему. Переход на страницу предобработки данных

Окончание таблицы 2

№	Тестовая ситуация	Начальное состояние системы	Действие	Ожидаемый результат
8	Тестирование одного разбиение выборки с различными параметрами	Страница разбиения выборки	Пользователь выбирает параметры для разбиения выборки и нажимает кнопку “Перейти к экспериментирова-нию”	Массив из одного элемента в котором находятся параметры разбиения передается в систему пользователь переходит на страницу экспериментов
9	Тестирование нескольких разбиений выборки с различными параметрами	Страница разбиения выборки	Пользователь выбирает параметры для нескольких разбиений и нажимает кнопку “Перейти к экспериментирова-нию”	Массив параметров разбиения передается в систему, пользователь переходит на страницу экспериментов

4.3 Выводы по главе

Таким образом, в главе были описаны инструменты разработки и проведено тестирование подсистемы предобработки данных программного средства для проведения экспериментов.

Заключение

В процессе работы над выпускной квалификационной работой бакалавра были решены следующие задачи:

1. Проведен обзор литературы на тему «Предобработка данных», обоснована актуальность работы.

2. Проведен анализ и построена модель предметной области, формализовано понятие эксперимента, осуществлена формальная постановка задачи предобработки выборки

3. Разработан технический проект подсистемы программного средства, включающий в себя требования и архитектуру всех подсистем.

4. Создан прототип и реализована подсистема программного средства, используя заданный стек технологий, проведено тестирование и экспериментальное исследование разработанной подсистемы.

Таким образом, можно сделать вывод о том, что все задачи были решены, а цель работы была достигнута.

Список литературы

1. Dealing with Imbalanced Dataset in Machine Learning: Techniques and Best Practices. [Электронный ресурс] – Режим доступа – https://www.blog.trainindata.com/machine-learning-with-imbalanced-data/ (дата обращения: 11.03.2022)

2.Hare Equty. Alteryx – компания, которая убьет Excel от Microsoft. 2021 [Электронный ресурс] – Режим доступа – https://dzen.ru/a/YIFVNWf0ci5JQt_e (дата обращения: 05.06.2023)

3.Hastie, T., Tibshirani, R., & Friedman, J. (2009). The Elements of Statistical Learning: Data Mining, Inference, and Prediction. Springer.

4.Japkowicz, N., & Stephen, S. (2002). The class imbalance problem: A systematic study. Intelligent Data Analysis, 6(5), 429-449. https://doi.org/10.3233/IDA-2002-6504 (дата обращения: 19.06.2023)

5.Jason Brownlee. "Data Cleaning: How to Clean, Prepare, and Standardize Data for Machine Learning" – Machine Learning Mastery – January 1, 2020

6.Jason Brownlee. Feature Selection for Machine Learning. Machine Learning Mastery. [Электронный ресурс] – Режим доступа – https://machinelearningmastery.com/feature-selection-machine-learning-python/ (дата обращения: 18.04.2022)

7. Jaylla. Методы отбора фич. [Электронный ресурс] – Режим доступа – https://habr.com/ru/articles/264915/ (дата обращения: 14.10.2022)

8. K-Fold кросс-Валидация. [Электронный ресурс] – Режим доступа – https://www.codecamp.ru/blog/cross-validation-k-fold/?ysclid=ljqsndt1k6186196022 (дата обращения: 12.03.2022)

9. Loginom. Silver Kit. Обнаружение и коррекция одномерных выбросов в данных. [Электронный ресурс] – Режим доступа – https://loginom.ru/blog/outliers?ysclid=ljqs145hx066429810 (дата обращения: 14.11.2022)

10. Mark Tabladillo. STeam Data Scince Process. 01.07.2023. [Электронный ресурс] – https://learn.microsoft.com/ru-ru/azure/architecture/data-science-process/prepare-data (дата обращения: 03.11.2022)

11. Moez Ali. Handling Machine Learning Categorical Data with Python Tutorial. 2023. [Электронный ресурс] – Режим доступа – https://www.datacamp.com/tutorial/categorical-data (дата обращения: 05.06.2023)

12. NTA. Апельсиновый Data Mining. 2021. [Электронный ресурс] – Режим доступа – https://vc.ru/dev/198641-apelsinovyy-data-mining (дата обращения: 02.06.2023)

13. NTA. Как разделять набор данных. [Электронный ресурс] – Режим доступа – https://vc.ru/newtechaudit/485313-kak-razdelyat-nabor-dannyh?ysclid=ljqsszbvh3355380220 (дата обращения: 05.03.2022)

14. vchampion. Введение в RapidMiner. 2015. [Электронный ресурс] – Режим доступа – https://habr.com/ru/articles/269427/ (дата обращения: 05.06.2023)

15. Дмитрий Макаров. Курс “Анализ и обработка данных” [Электронный ресурс]: онлайн курс. – Режим доступа – https://www.dmitrymakarov.ru/data-analysis/ (дата обращения: 05.06.2023)

16. Кодкамп. Коэффициент корреляции Пирсона. [Электронный ресурс] – Режим доступа – https://www.codecamp.ru/blog/pearson-correlation-coefficient/ (дата обращения: 13.09.2022)

17. Кросс валидация. [Электронный ресурс] – Режим доступа – https://academy.yandex.ru/handbook/ml/article/kross-validaciya (дата обращения: 06.03.2023)

18. Отбор признаков в задачах машинного обучения. 2021 [Электронный ресурс] – Режим доступа – https://habr.com/ru/articles/550978/ (дата обращения: 11.08.2022)

19. Тимофей Королёв. Что такое Figma и для чего она нужна. [Электронный ресурс] – Режим доступа – https://gb.ru/posts/chto-takoe-figma-i-dlya-chego-ona-nuzhna (дата обращения: 10.06.2023)