МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ
Федеральное государственное автономное образовательное учреждение
высшего образования
«ТЮМЕНСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»

ИНСТИТУТ МАТЕМАТИКИ И КОМПЬЮТЕРНЫХ НАУК
Кафедра программного обеспечения

РЕКОМЕНДОВАНО К ЗАЩИТЕ
В ГЭК И ПРОВЕРЕНО НА ОБЪЕМ
ЗАИМСТВОВАНИЯ
Заведующий кафедрой
к.т.н., доцент
_____М. С. Воробьева
24.06. 2019 г.

ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА
(магистерская диссертация)

РАЗРАБОТКА ПЛАТФОРМЫ ДЛЯ ПРОВЕДЕНИЯ ПРЕДВАРИТЕЛЬНОГО АНАЛИЗА БОЛЬШИХ ОБЪЕМОВ ДАННЫХ С ПРИМЕНЕНИЕМ АЛГОРИТМОВ МАШИННОГО ОБУЧЕНИЯ

02.04.03. Математическое обеспечение и администрирование информационных систем
Магистерская программа «Разработка, администрирование и защита вычислительных систем»

Выполнил работу
Студент 2 курса
очной формы обучения
Чемакин Тарас Александрович

Руководитель работы
к.т.н., доцент
Воробьева Марина Сергеевна

Рецензент
Генеральный директор
ООО «Инкомтехнологии Групп»
Воробьев Артем Максимович

г. Тюмень, 2019

СОДЕРЖАНИЕ

ВВЕДЕНИЕ

Машинное обучение на больших данных является одним из главных трендов в IT-индустрии в последние годы.

Накопленные объемы информации и вычислительные мощности позволяют по-новому исследовать мир, планировать работу и строить прогнозы.

Анализ больших данных позволяет увидеть скрытые закономерности, незаметные ограниченному человеческому восприятию.

Активнее всего большие данные используют в финансовой и медицинской отраслях, высокотехнологичных и интернет-компаниях, а также в государственном секторе. Однако, специальные сервисы позволяют успешно применять технологии больших данных и в маркетинге среднего и малого бизнеса без содержания специалистов и дорогостоящего оборудования.

Сегодня машинное обучение и глубокий анализ данных уже не является чем-то новым. Это обязательный пункт, без которого бизнес не сможет нормально конкурировать в современном мире. Анализ собираемой информации — ключ к улучшению показателей бизнеса.

Построение систем машинного обучения в настоящее время является одной из самых популярных, актуальных и современных областей человеческой деятельности на стыке информационных технологий, математического анализа и статистики. Главная проблема данных технологий – достаточно высокая трудоемкость. Построение систем машинного обучения без привлечения специализированных платформ требует огромного количества времени высокопрофессиональных специалистов как в сфере искусственного интеллекта, так и в той предметной области, к которой эта технология применяется.

IT-специалисты ТюмГУ разрабатывают систему для проведения анализа больших объемов данных с применением моделей машинного обучения. Система предназначена для автоматизации аналитической деятельности в бизнес-процессах и производстве с использованием алгоритмов машинного обучения.

Система представляет собой визуальную среду для управления данными и вычислениями над ними. Для работы в системе пользователю не требуется иметь навыки программирования. Проект призван обеспечить доступность использования алгоритмов машинного обучения в среднем и малом бизнесе для более качественной организации бизнес-процессов и увеличения прибыли.

Цель выпускной квалификационной работы — разработка и создание платформы для проведения предварительного анализа больших объемов данных с применением алгоритмов машинного обучения .

Задачи:

- обзор платформ, сервисов и библиотек, предназначенных для машинного обучения;
- обзор статей , посвященных инструментам по машинному обучению;
- разработка и реализация визуального редактора экспериментов;
- реализация вычислительных операций «Decision Tree», «Обучение», «Обучение и проверка», «Тестирование», «Предсказание»;
- проверка работы приложения на конкретном наборе данных.

1. ОБЗОР ПЛАТФОРМ МАШИННОГО ОБУЧЕНИЯ

Для решения задач машинного обучения применяются различные инструменты. К ним относятся платформы, предоставляющие все необходимые возможности для запуска проекта, и библиотеки, которые предоставляет только отдельные возможности.

Существующие онлайн-платформы могут предоставить все те возможности, которые доступны в инструментах машинного обучения и даже больше, так как разработчиками таких платформ являются лучшие специалисты в этой сфере. Такие платформы предоставляют возможности для ведения проекта машинного обучения от начала до конца. А именно, анализ данных, подготовка данных, моделирование, оценка и выбор алгоритма.

В настоящее время существует множество платформ и сервисов [5]:

- BigML (https://bigml.com/) — это платформа машинного обучения на основе облака с простым в использовании графическим интерфейсом. BigML также предоставляет простые механизмы для включения прогнозирующих моделей в приложениях через REST API. Платформа включает в себя наблюдаемое обучение (для построения прогнозных моделей), обучение без учителя (для понимания поведения), обнаружение аномалий (используется в выявлении случаев мошенничества), средства визуализации данных (разброс-диаграмм и графиков Sunburst) и множество механизмов для изучения данных. BigML является прагматичной, с низкой стоимостью, простой в использовании платформой для построения мощных прогностических моделей.
- Amazon Machine Learning (https://aws.amazon.com/ru/machine-learning/) — это сервис, который предоставляет возможности использовать технологии машинного обучения разработчикам всех уровней квалификации. Amazon

Machine Learning предоставляет инструменты визуализации и проводит разработчика через весь процесс создания модели машинного обучения без необходимости изучения сложных алгоритмов и технологий МО. На основе полученных моделей Amazon Machine Learning позволяет легко получить предсказания для приложения, используя простые API-интерфейсы, без необходимости реализации пользовательского кода.

- DataRobot (https://www.datarobot.com/) — это сервис машинного обучения на основе облачных вычислений, который выполняет большую часть мелких работ в процессе построения прогнозной модели. DataRobot автоматически ищет лучшие признаки, выбирает наиболее подходящие алгоритмы, тестирует модели и предоставляет API для развертывания модели. Сервис основан на алгоритмах, доступных в таких средах, как R, Python и Spark, и использует TextMining, обнаружение типа переменной, кодирование, масштабирование, преобразование и автоматическую генерацию признаков.

- FICO (https://www.fico.com/br/platform/fico-decision-management-platform) — один из наиболее опытных поставщиков статистических решений и технологий машинного обучения для бизнес-задач. Сервис FICO Analytic Cloud позволяет решать задачи машинного обучения, статистики, оптимизации и бизнес-правил управления, в контексте хорошо управляемой среде. FICO также предоставляет рынок для разработчиков аналитических решений и пользователей, которые имеют потребность в них.

- Google Cloud AI Platform (https://cloud.google.com/ai-platform/) может интегрироваться с приложениями, основанными на Google App Engine. API доступна через библиотеки для многих популярных языков, таких как Python, JavaScript и .NET. AI Platform обеспечивает возможности поиска по шаблону, сентиментального анализа клиентов, анализа оттока,

обнаружения спама, классификации документов, прогнозирования покупок, рекомендаций, интеллектуальной маршрутизации и многое другое. AI Platform может получать данные из BigQuery и Google Cloud Storage.

- Платформа Cognos Analytics (https://www.ibm.com/analytics/cognos-analytics) компании IBM предлагает возможности прогнозного анализа и визуализации данных в режиме диалогового интерфейса. Cognos Analytics автоматически применяет математические методы, чтобы показать наиболее значимые факты, закономерности и отношения. Доступна бесплатная версия с ограничениями на объемы данных.
- Платформа PurePredictive (http://www.purepredictive.com/) использует искусственный интеллект для автоматизации процесса машинного обучения. Платформа позволяет автоматизировать преобразование данных, поиск взаимосвязи между признаками и исправление искажений в данных. Облачная платформа масштабируется автоматически для рабочих нагрузок, что позволяет размещать наборы данных практически любого размера. Модели легко расширяются через веб-сервисы, и могут поддерживаться автоматически.
- Yottamine (https://yottamine.com/) включает в себя широкие возможности для импорта и применения моделей в реальных условиях. Сервис позволяет пользователям в полной мере воспользоваться возможностями масштабируемых по требованию облачных систем вычисления. Прогнозная служба Yottamine позволяет строить модели или делать предсказания в два простых шага. С помощью интеграции с масштабируемыми облачными системами обеспечивается высокая скорость и эффективность вычисления. Полученные модели могут быть экспортированы в PMML. Пользователи могут подключаться и управлять

Yottamine Predictive Web Services с использованием языка программирования R с помощью пакета YottamineR.

- Microsoft Azure Machine Learning Studio (https://azure.microsoft.com/ru-ru/services/machine-learning-studio/) — полностью управляемая облачная служба, позволяющая легко создавать и развертывать решения прогнозной аналитики, а также предоставлять общий доступ к ним. Служба машинного обучения Azure предназначена для прикладного машинного обучения. Машинное обучение Azure включает сотни встроенных пакетов Python и R и поддержку настраиваемого кода.

2. ОБЗОР СТАТЕЙ, ПОСВЯЩЕННЫХ ИНСТРУМЕНТАМ ПО МАШИННОМУ ОБУЧЕНИЮ

Созданием инструментов интерактивного машинного обучения занимаются многие известные университеты.

Свидетельством тому являются публикуемые в различных изданиях статьи, посвященные этой тематике.

Особый интерес представляет статья Yang Q. et al. «Проектирование инструментов интерактивного машинного обучения на основании того, как не-эксперты строят модели на самом деле» [4].

Авторами статьи являются Qian Yang из университета Карнеги — Меллона (Human-Computer Interaction Institute), Jina Suh и Gonzalo Ramos из Microsoft Research, Nan-Chen Chen из Вашингтонского университета (Human-Centered Design & Engineering).

В статье описаны результаты опросов и интервью с людьми, не являющимися экспертами в машинном обучении (machine learning, ML), но использовавшими эту технологию для решения производственных задач или реализации собственных проектов, и сделаны выводы об особенностях использования ими инструментов ML.

Также предложен способ организации пользовательского интерфейса к системам ML, основанный на выделении пользователем конкретных примеров для оценки получаемой модели, учитывающий эти особенности и позволяющий предотвратить наиболее частые ошибки пользователей.

Критериями участия в опросе являлись отсутствие у участников специального образования (по математике, статистике, ML) и использование ими моделей на основе ML для решения какой-либо задачи.

Опрос содержал вопросы о проекте, в котором участники использовали технологии ML, о входных и выходных данных модели, об ожиданиях и результатах. В опросе участвовало девяносто восемь человек. Кроме этого, были проведены интервью с еще двадцатью четырьмя людьми и десятью консультировавшими их экспертами ML (мнения экспертов учитывались отдельно).

По результатам опроса и интервью были сделаны следующие выводы:

- ML обычно используется для определения особенностей и закономерностей, выявляемых алгоритмами построения моделей, а не для применения обученной модели в дальнейшем. Большинство участников не искали максимальной точности в решении конкретной задачи, а только улучшали существующий неавтоматический способ решения.
- Большинство участников представляли себе алгоритмы ML в виде механизма преобразования данных и использовали пары входных и выходных значений для оценки точности модели. Участники выражали больше доверия к полученной модели, чем эксперты ML.
- Все участники являлись экспертами в своей предметной области и понимали особенности используемых данных. Большинство участников имели опыт программирования, что позволило им использовать доступные в интернете скрипты и примеры. Немногие участники использовали документацию используемых ими API. Также немногие использовали визуализации и описательные статистики для отладки моделей.
- В отличие от изначальных предположений, большинство участников не использовали графические инструменты ML, предпочитая текстовые инструменты из-за простоты использования скриптов и решений из интернета. Участники с навыками программирования также применяли

инструменты контроля версий, отслеживания ошибок и т. п. В отличие от них, эксперты-консультатны ML предпочитали графические инструменты из-за возможности повторного использования частей решений в будущих проектах.

К основным ошибкам участников относятся:

- Неполное понимание сложностей, возникающих при постановке задачи — приведении высокоуровневых требований к задачам и подзадачам, решение которых возможно с помощью ML. Участники не выполняли анализ сбалансированности данных, количества пропусков и т. п.
- Участники не выполняли отладку моделей. При недостаточной точности модели основным решением было увеличение тренировочной выборки. Участники не производили выделение и исключение признаков.
- Большинство участников использовали только процент правильных предсказаний модели в качестве метрики и не учитывали проблемы переобучения.

На основе этих результатов был предложен подход к построению пользовательских интерфейсов к системам ML — «Test-Driven Maching Teaching», основанный на существующих подходах «Machine Teaching» и «Test-Driven ML».

Основная идея подхода заключается в том, что пользователь должен выбрать небольшой набор примеров в качестве тестовых случаев, в которых пользователь наиболее заинтересован. Это заставляет пользователя просмотреть имеющиеся данные и конкретизировать задачу, а также позволяет инструменту своевременно сообщить о недостаточном размере выборки, несбалансированности и т. п.

На основе характеристик данных, типа задачи и выбранными пользователем тестовыми примерами инструмент может предложить подходящие алгоритмы или параметры.

После обучения вместо или совместно со стандартными метриками отображаются предсказания модели на тестовых примерах, из расчета на понимание и интерес пользователя.

Другой пример — статья Dudley J. J., Kristensson P. O. «Обзор дизайнов пользовательского интерфейса для интерактивного машинного обучения» [1].

Авторами статьи являются John J. Dudley и Per Ola Kristensson из Оксфордского университета.

В статье подробно рассмотрены подходы к интерактивному машинному обучению с точки зрения взаимодействия с пользователем. Описана обобщенная модель системы интерактивного машинного обучения и выявлены способы построения эффективных пользовательских интерфейсов для такой системы. Определены направления исследования пользовательского интерфейса, открывающие путь к более эффективным и продуктивным неэкспертным интерактивным приложениям для машинного обучения.

Еще одним примером является статья Feurer M. et al. «Эффективное и надежное автоматизированное машинное обучение» [2], посвященная автоматическому машинному обучению (automated machine learning, AutoML) — системам, выполняющим автоматический выбор признаков, алгоритмов и параметров для заданного пользователем набора данных и предназначенным для не-экспертов. В статье представлена разработанная система «Auto-sklearn», основанная на scikit-learn, показывающая лучшие результаты, чем более ранние аналогичные системы. Описанная в статье система заняла первое место на первом этапе конкурса ChaLearn AutoML Challenge в 2015 году.

Авторами статьи являются Matthias Feurer, Aaron Klein, Katharina Eggensperger, Jost Tobias Springenberg, Manuel Blum и Frank Hutter из Фрайбургского университета (Department of Computer Science).

Задача автоматического машинного обучения представляет интерес для сервисов вроде Amazon Machine Learning или Microsoft Azure Machine Learning, позволяющих не-экспертам использовать возможности ML на произвольных наборах данных. Система AutoML должна определить наиболее подходящие способы предобработки данных, алгоритм обучения и его гиперпараметры при ограниченных вычислительных ресурсах (времени выполнения, оперативной памяти).

Подход к решению задачи основан на используемом в предложенной ранее системе Auto-WEKA — Байесовской оптимизации в объединенном пространстве известных системе алгоритмов и их гиперпараметров. В статье рассмотрены две модификации, приводящие к повышению эффективности и надежности системы.

Первая модификация — метаобучение — заключается в использовании для инициализации оптимизационного процесса заранее обученных на известных наборах данных моделей. Для каждого такого набора рассчитываются метапризнаки — простые в вычислении статистические характеристики — которые используются для определения близости к новому набору данных. В качестве известных наборов взяты сто сорок наборов из ресурса OpenML.

Вторая модификация состоит в использовании не только наилучшей найденной модели, но и других рассмотренных в процессе оптимизации моделей путем построения ансамбля. В ансамбль включаются модели, максимизирующие результаты предсказаний на исключенной при обучении выборке данных.

Модификации реализованы в системе Auto-sklearn, использующей возможности библиотеки scikit-learn, в том числе пятнадцать алгоритмов классификации, четырнадцать методов обработки признаков и четыре способа предобработки данных. Было проведено сравнение работы разработанной системы с системой Auto-WEKA, а также анализ эффективности предложенных модификаций. На шести наборах данных из двадцати одного система Auto-sklearn показала значимо лучший результат, чем Auto-WEKA, еще на двенадцати — сравнимый с ней. При этом влияние модификаций значительно — версия системы с отключенными модификациями заметно проигрывает полной системе. Исходный код системы открыт и доступен в интернете.

К основным недостаткам Auto-sklearn относятся отсутствие возможности решения задач регрессии и кластеризации, а также ограничение на использование на небольших и средних наборах данных.

Авторы статьи предполагают, что рассмотренные модификации Байесовской оптимизации принесут результаты и при использовании с системами глубокого обучения на больших наборах данных.

3. ТЕХНИЧЕСКОЕ ОПИСАНИЕ ПЛАТФОРМЫ

3.1. Проект «Платформа ML»

Платформа ML — визуальная среда для проведения экспериментов по машинному обучению. Платформа позволяет создавать вычислительные эксперименты с использованием алгоритмов машинного обучения при помощи визуального редактора. Схема управления процессом машинного обучения показана на рисунке 1.

Платформа предназначена для автоматизации аналитической деятельности в бизнес-процессах, производстве и исследованиях.

Задачи проекта:

- создание инструмента для проведения предварительной обработки (подготовки) исходных данных к последующему анализу;
- создание платформы для проведения анализа больших объемов данных с использованием моделей машинного обучения с целью получения новых знаний и зависимостей в данных;
- создание инструмента для визуального управления процессом сбора, подготовки и анализа больших объемов данных;
- создание хранилища больших объемов исходных данных (датасетов) для их последующей обработки.

Базовые возможности системы:

- загрузка датасетов в систему;
- составление этапов эксперимента;
- визуальное управления этапами эксперимента;

- обучения и тестирования моделей классификации и регрессии.

С помощью системы можно решить, например, следующие задачи:

- определение категории риска на производстве по текущим показателям датчиков;
- определение наличия ископаемых по параметрам месторождения;
- предсказание продаж на основе данных за прошлые периоды;
- выделение групп потребителей о степени интереса к продукту на основе результатов проведенных маркетинговых исследований и др.

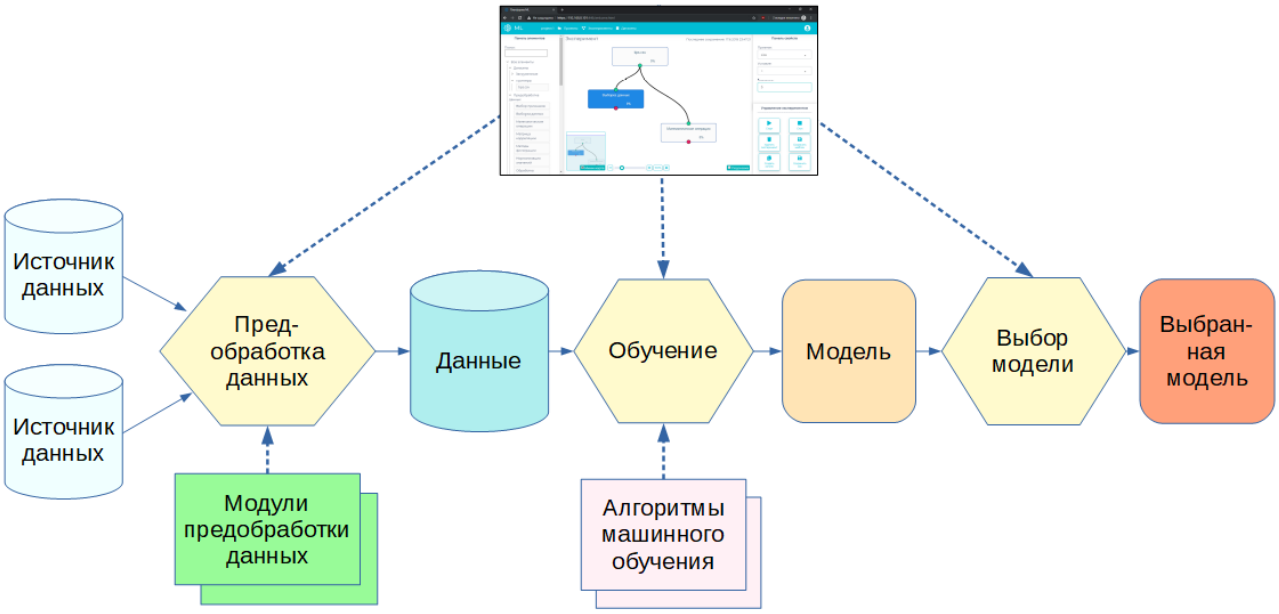

Рисунок 1. Схема управления процессом машинного обучения

3.2. Средства ПО и технологии

Python (https://www.python.org/) — интерпретируемый высокоуровневый язык программирования общего назначения, ориентированный на повышение производительности разработчика и читаемости кода [8]. Python поддерживает различные парадигмы программирования, в том числе процедурное, объектно-

ориентированное и функциональное программирование. Язык Python имеет динамическую типизацию и автоматическое управление памятью.

Flask (http://flask.pocoo.org/) — фреймворк для создания веб-приложений на языке программирования Python, использующий набор инструментов Werkzeug, а также шаблонизатор Jinja2 [6]. Flask относится к категории так называемых микрофреймворков — минималистичных каркасов веб-приложений, сознательно предоставляющих лишь самые базовые возможности. Дополнительные функции добавляются расширениями и интегрируются в фреймворк. Существуют расширения для ORM, валидации форм, загрузки файлов и различных технологий аутентификации.

NumPy (https://www.numpy.org/) — библиотека Python, предоставляющая возможности для эффективной работы с многомерными массивами, а также функции для решения задач линейной алгебры, генерации случайных чисел и статистические функции.

Pandas (https://pandas.pydata.org/) — библиотека Python, предназначенная для решения задач анализа данных и предоставляющая классы для работы с таблицами данных, функции для обработки данных, функции для импорта и экспорта данных в различных форматах.

Scikit-learn (https://scikit-learn.org/, [3]) — библиотека алгоритмов машинного обучения для языка Python, основанная на NumPy и SciPy. Реализует различные алгоритмы классификации, регрессии и кластеризации, в том числе метод опорных векторов, Random forest, градиентный бустинг, метод k-средних и DBSCAN.

TensorFlow (https://www.tensorflow.org/) — открытая программная библиотека для машинного обучения, разработанная компанией Google для решения задач построения и тренировки нейронной сети с целью автоматического нахождения и классификации образов [7]. Применяется как для исследований, так и для

разработки собственных продуктов Google. Основной API для работы с библиотекой реализован для Python, также существуют реализации для C++, Haskell, Java, Go и Swift.

PostgreSQL (https://www.postgresql.org/) — реляционная СУБД, придающая особое значение надежности, производительности, соответствию стандартам и расширяемостью.

Two.js (https://two.js.org/) — библиотека для работы с двухмерной графикой в современных веб-браузерах. Two.js предоставляет одинаковый API для построения изображений с помощью SVG, Canvas и WebGL.

3.3. Структура платформы

Платформа имеет клиент-серверную архитектуру (рисунок 2).

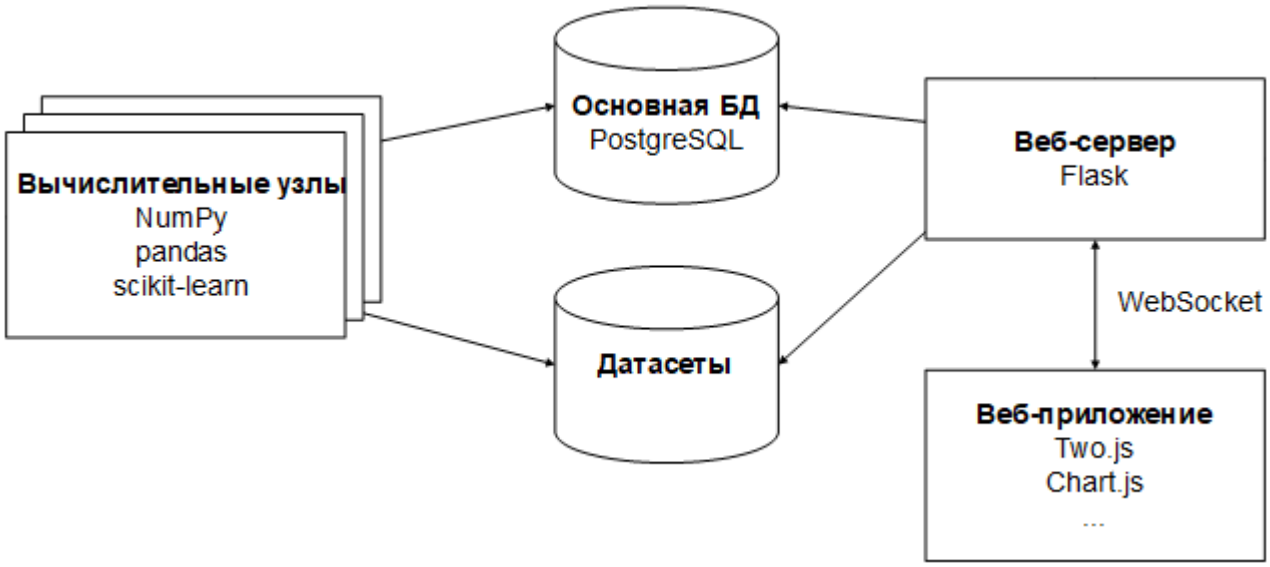

Рисунок 2. Схема системы

Серверная составляющая состоит из двух компонентов. Первый компонент — веб-сервер на базе веб-фреймворка Flask. Веб-сервер основан на архитектуре MVC.

Второй серверный компонент — вычислительные узлы, использующие для обработки данных библиотеки NumPy, Pandas, scikit-learn и TensorFlow.

Взаимодействие между веб-сервером и вычислительными узлами осуществляется через общую базу данных. Исходные датасеты, промежуточные и финальные результаты вычислений хранятся в файловом хранилище, также доступном как веб-серверу, так и вычислительным узлам.

Клиентская составляющая представляет собой веб-приложение. Редактор экспериментов использует графическую библиотеку Two.js. Для взаимодействия между клиентом и сервером используется технология WebSocket.

3.4. Организация данных

В системе все пользователи относятся к компаниям. Администратор компании может приглашать пользователей в свою компанию с разными правами доступа и доступом к разным проектам. У компании может быть несколько проектов. В одном проекте может быть несколько экспериментов и датасетов.

На рисунке 3 приведен фрагмент схемы базы данных, содержащий таблицы, описывающие проекты.

В таблице 1 приведены описания таблиц, описывающих проекты.

Датасеты загружаются в формате CSV. Возможен выбор кодировки, символа разделителя полей, разделителя дробной части. Каждый загруженный датасет доступен в виде блока в редакторе экспериментов. Датасеты хранятся в файлах в виде сериализованных объектов Python. Для каждого датасета отдельно доступна информация о его столбцах, используемая для подсказок пользователю.

На рисунке 4 приведен фрагмент схемы базы данных, содержащий таблицы, описывающие датасеты.

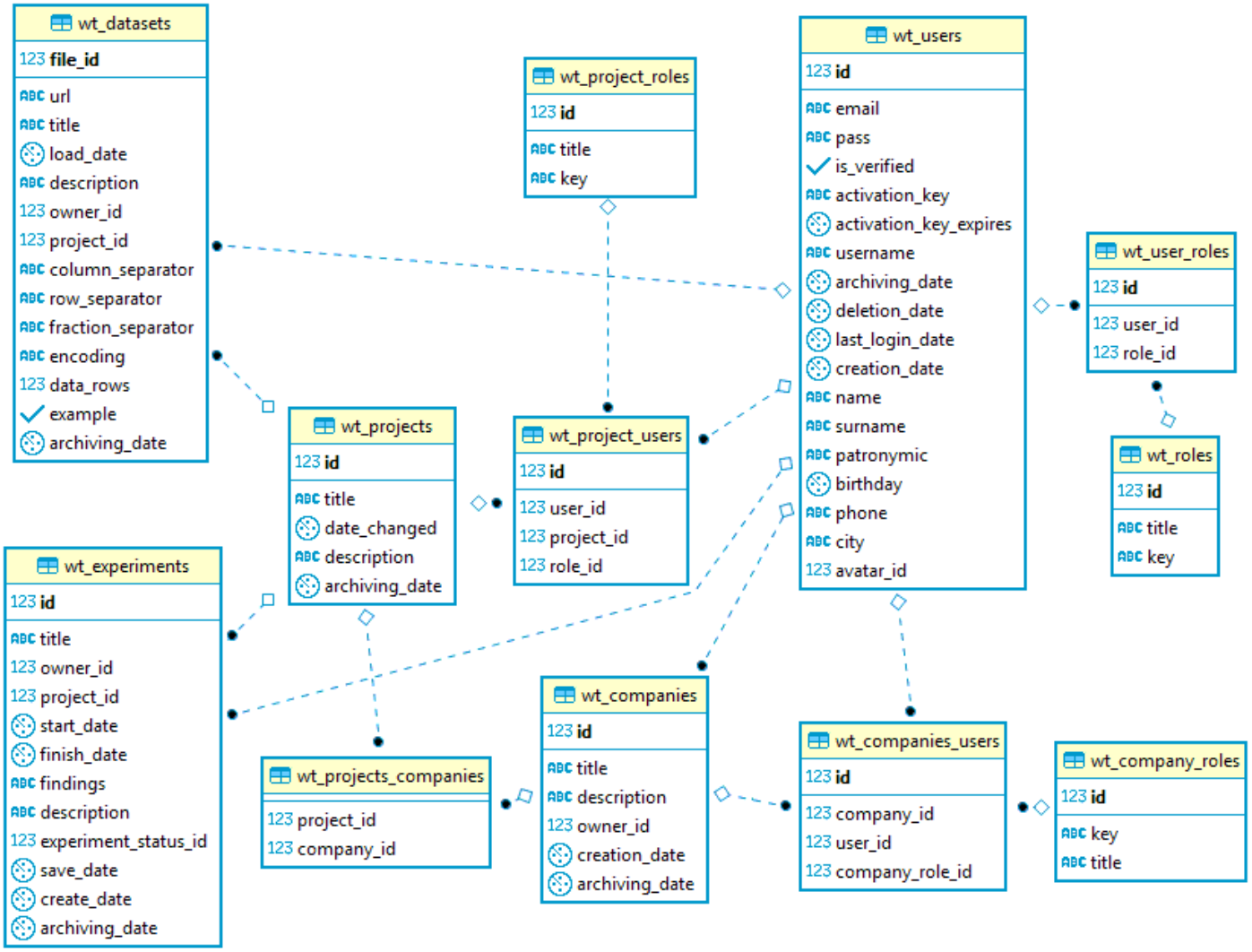

Рисунок 3. Фрагмент схемы базы данных, содержащий таблицы, описывающие проекты

Таблица 1. Таблицы, описывающие проекты

Таблица	Описание
wt_projects	Содержит основную информацию о проектах
wt_datasets	Содержит основную информацию о датасетах проектов
wt_expertiments	Содержит основную информацию об экспериментах проектов
wt_companies	Содержит информацию о компаниях
wt_users	Содержит информацию о пользователях
wt_project_users	Содержит информацию о ролях пользователей в проектах
wt_projects_companies	Содержит информацию о доступе проектов компаниям
wt_companies_users	Содержит информацию о ролях пользователей в компаниях
wt_user_roles	Содержит информацию о ролях пользователей в системе
wt_project_roles	Справочник ролей пользователей по отношению к проекту
wt_company_roles	Справочник ролей пользователей по отношению к компании
wt_roles	Справочник ролей пользователей в системе

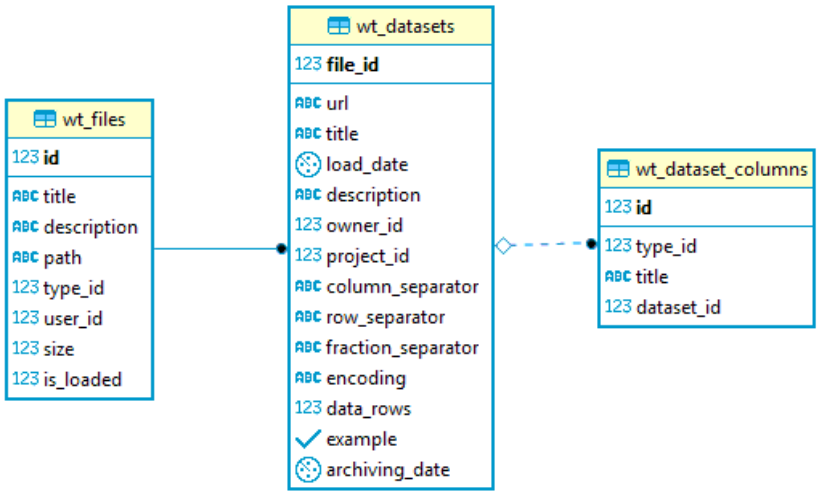

Рисунок 4. Фрагмент схемы базы данных, содержащий таблицы, описывающие датасеты

В таблице 2 приведены описания таблиц, описывающих датасеты.

Таблица 2. Таблицы, описывающие датасеты

Таблица	Описание
wt_datasets	Содержит основную информацию о датасетах
wt_dataset_columns	Содержит основную информацию о столбцах датасетов
wt_files	Содержит информацию о загруженных файлах

Датасеты могут быть использованы в экспериментах.

Эксперименты представляют собой последовательность шагов, каждый из которых заключается в выполнении одной из вычислительных операций над некоторыми датасетами и моделями. Результаты выполнения одной операции могут быть использованы в качестве входных параметров других операций. Для каждой операции также могут быть заданы ее свойства, набор которых зависит от типа операции.

На рисунке 5 приведен фрагмент схемы базы данных, содержащий таблицы, описывающие эксперименты.

В таблице 3 приведены описания таблиц, описывающих эксперименты.

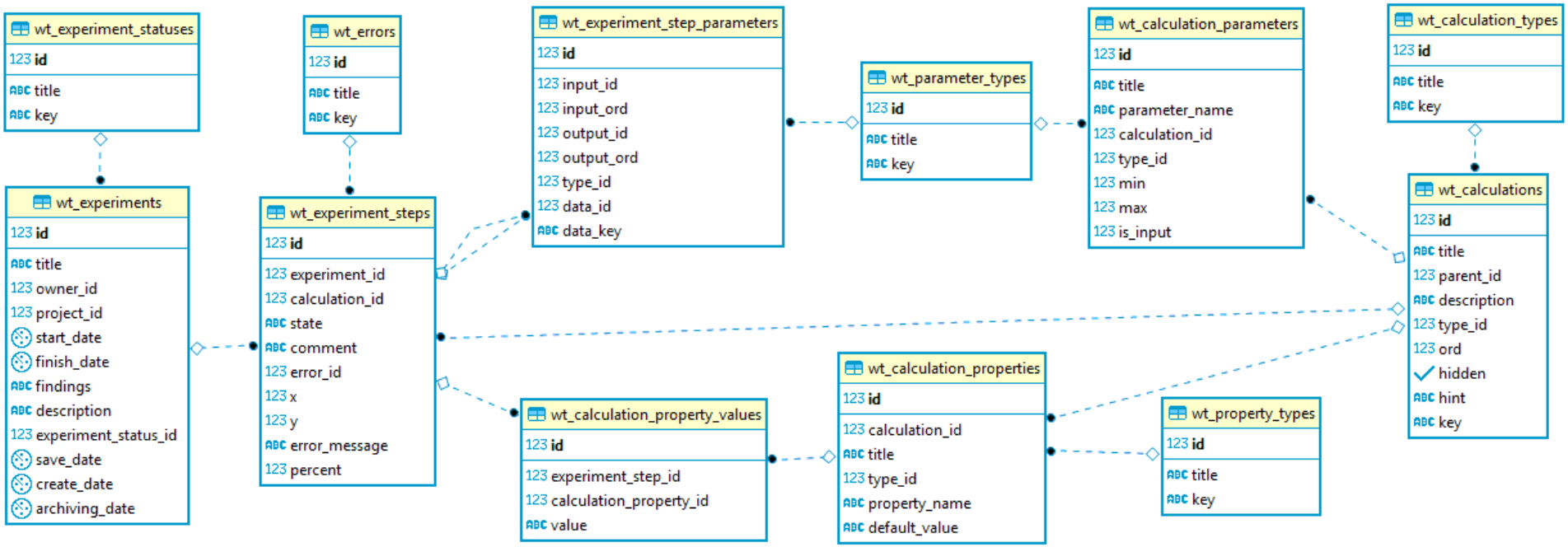

Рисунок 5. Фрагмент схемы базы данных, содержащий таблицы, описывающие эксперименты

Таблица 3. Таблицы, описывающие эксперименты

Таблица	Описание
wt_experiments	Содержит основную информацию о датасетах
wt_calculations	Содержит информацию о вычислительных операциях, то есть о типах шагов эксперимента
wt_calculation_types	Содержит информацию о типах вычислительных операций
wt_calculation_properties	Содержит информацию о свойствах вычислительных операций
wt_property_types	Содержит информацию о типах свойств вычислительных операций
wt_calculation_parameters	Содержит информацию о входных и выходных параметрах вычислительных операций
wt_parameter_types	Содержит информацию о типах входных и выходных параметров вычислительных операций
wt_experiment_steps	Содержит информацию о шагах экспериментов
wt_experiment_step_parameters	Содержит информацию о связях между входными и выходными параметрами шагов экспериментов
wt_calculation_property_values	Содержит информацию о значениях свойств шагов экспериментов
wt_errors	Справочник типов ошибок вычислительных операций
wt_experiment_statuses	Справочник состояний экспериментов

3.5. Раздел «Проекты»

В разделе «Проекты» можно просмотреть список проектов и сводную информацию о содержащихся в них экспериментах и датасетах. Для проекта есть возможность отредактировать название, описание и автора проекта, и удалить проект. Внешний вид страницы показан на рисунке 6.

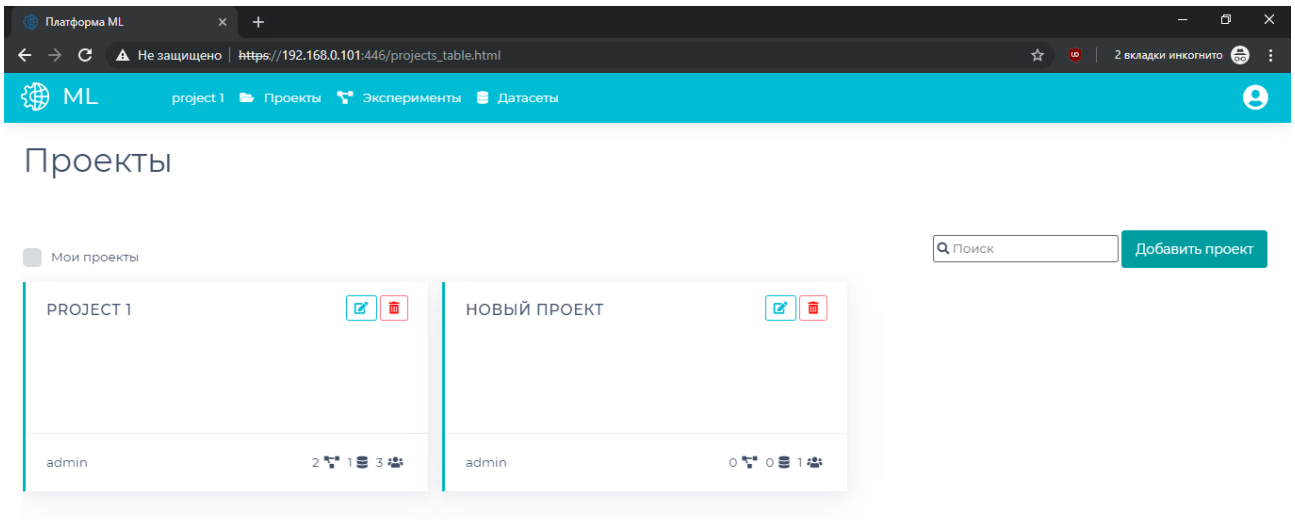

Рисунок 6. Страница проектов

Также возможно просмотреть всю информацию о конкретном проекте на одной странице, где отображаются списки экспериментов, датасетов и пользователей проекта. Внешний вид страницы показан на рисунке 7.

На вкладке датасетов можно увидеть информацию о датасетах текущего проекта и загрузить новый датасет.

На вкладке экспериментов отображен список экспериментов текущего проекта. Также можно создать новый эксперимент.

На вкладке пользователей в текущий проект можно добавить пользователя и сразу указать его роль в этом проекте. В системе определены три уровня доступа пользователей к проектам и их экспериментам: «Владелец», «Редактор» и «Просмотр».

Пользователь уровня «Владелец» имеет права на любые действия с экспериментами проекта. Пользователь уровня «Редактор» имеет права на редактирование схемы эксперимента. Пользователь уровня «Просмотр» имеет права на просмотр схемы эксперимента, но не имеет доступа к редактированию.

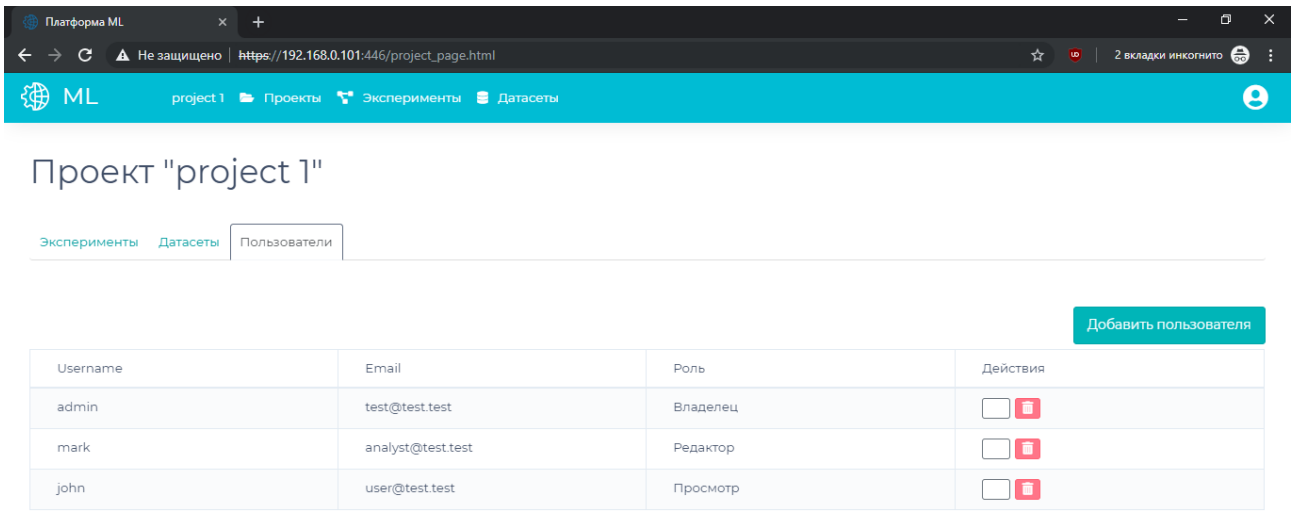

Рисунок 7. Пользователи проекта

За раздел «проекты» со стороны сервера отвечает контроллер `CWProjectController`. Поддерживаемые запросы приведены в таблице 4.

Таблица 4. Методы контроллера проектов

Запрос	Параметры	Метод контроллера	Описание
project_list	—	GetProjectList	Возвращает список проектов текущего пользователя
project_search	query	SearchProjects	Выполняет поиск проектов текущего пользователя по части названия
project_contents	project_id	GetProjectContents	Возвращает информацию о экспериментах, датасетах и пользователях в проекте
project_add	—	AddProject	Возвращает форму для добавления проекта
edit_project	project_id	EditProject	Возвращает форму для

Запрос	Параметры	Метод контроллера	Описание
			редактирования проекта
delete_project	project_id	DeleteProject	Помечает проект как удаленный
project_page	project_id	GetProjectPage	Возвращает страницу проекта
project_add_ user_form	project_id	AddUserToProjectForm	Возвращает форму для добавления пользователя в проект
edit_user_ in_project	project_id, user_id	EditUserInProject	Возвращает форму для изменения роли пользователя в проекте
delete_user_ in_project	project_id, user_id	DeleteUserInProject	Исключает пользователя из проекта

3.6. Визуальный редактор экспериментов

Редактор экспериментов предоставляет графический интерфейс для изменения последовательности шагов эксперимента.

Эксперимент представлен в виде схемы, каждый шаг имеет вид прямоугольника (блока), а сопоставление входных и выходных параметров представлено в виде кривых линий, соединяющих точки на блоках.

Интерфейс редактора показан на рисунке 8.

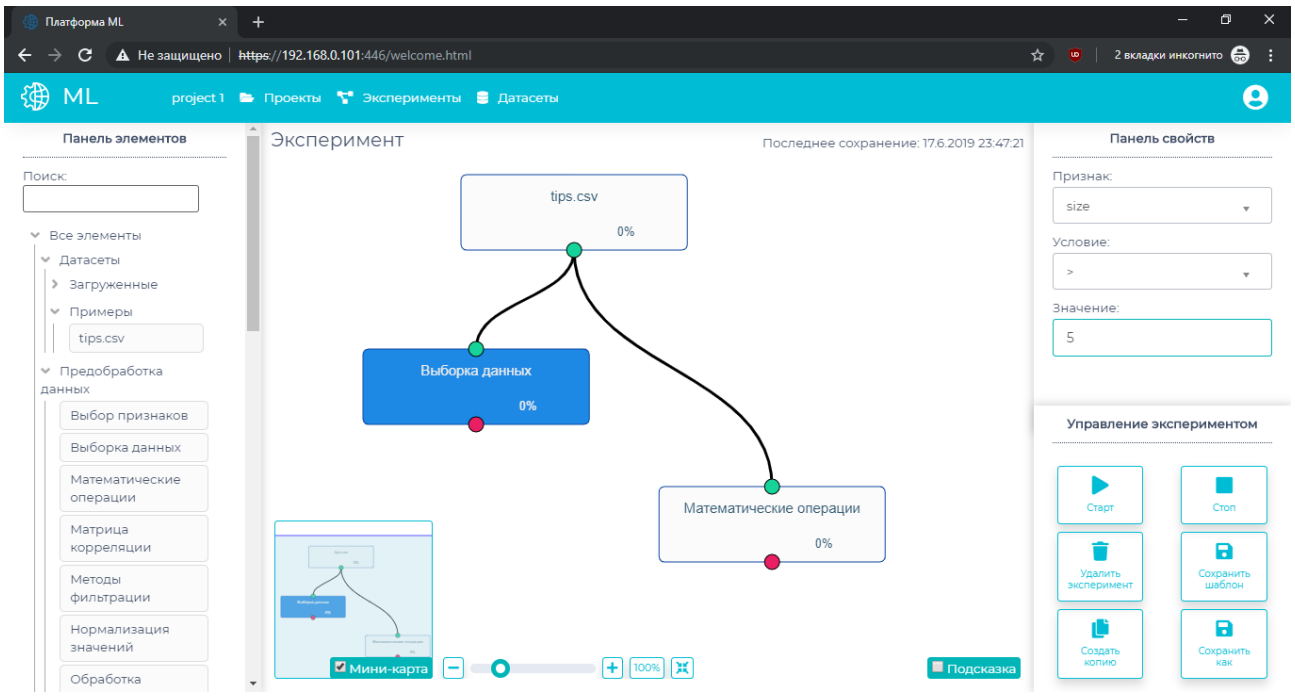

Рисунок 8. Интерфейс редактора экспериментов

В центральной части расположена рабочая область, на которой отображается схема эксперимента. Блоки можно произвольно перетаскивать по области. При перетаскивании пустого пространства происходит смещение видимой части схемы эксперимента.

Входные и выходные параметры шагов эксперимента отображаются в виде круглых точек соединения в верхней (для входных параметров) и нижней (для выходных параметров) частях блока.

Разные типы блоков имеют разное количество входных и выходных параметров, причем некоторые типы блоков вместо конкретного числа параметров допускают диапазон значений — в этом случае отображаются только использованные и одна пустая точка соединения.

Для соединения выходного параметра одного блока с входным параметром другого необходимо протянуть от одной их точек соединения к другой. Для отмены существующего соединения необходимо перетащить один из концов соединения в пустое пространство. Из каждой выходной точки может выходить

произвольное количество соединений, а в каждую входную точку может входить только одно соединение. При соединении двух точек различных блоков проверяется совпадение типов параметров и отсутствие циклов. Для запуска эксперимента необходимо, чтобы все входные параметры всех блоков были соединены с выходным параметром.

Поддерживаются функции удаления, копирования, вырезания и вставки блоков. При выделении нескольких блоков эти операции применяются ко всем выделенным блокам.

Под рабочей областью расположены элементы управления перемещением и масштабированием, а также мини-карта и короткое описание выделенного блока.

В левой панели находится дерево элементов, содержащее все датасеты, доступные в данном проекте или предоставленные системой, а также все вычислительные операции, сгруппированные по типу. Реализована функция поиска операции по названию. Для добавления датасета или операции в качестве шага эксперимента необходимо перетащить соответствующий блок на рабочую область.

В правой панели располагается редактор свойств выделенного блока и панель управления экспериментом. При отсутствии выделенного блока вместо свойств блока отображаются свойства эксперимента.

Все изменения сохраняются на сервер в режиме реального времени. Проверки на циклы повторяются на стороне сервера, а статусы измененных блоков и всех зависящих от них блоков сбрасываются.

За редактор экспериментов со стороны сервера отвечает контроллер `CWControllerWorkspace`. Поддерживаемые запросы приведены в таблице 5.

Таблица 5. Методы контроллера редактора экспериментов

Запрос	Параметры	Метод контроллера	Описание
workspace	project_id, experiment_id	GetWorkspace	Возвращает страницу редактора эксперимента
get_element_tree	project_id	GetElementTree	Возвращает содержимое дерева элементов в проекте
get_experiment	experiment_id	GetExperiment	Возвращает схему эксперимента
get_calculation_ properties	experiment_ step_id	GetCalculation Properties	Возвращает свойства шага эксперимента
get_experiment_ properties	experiment_id	GetExperiment Properties	Возвращает свойства эксперимента
get_experiment_ status	experiment_id	GetExperimentStatus	Возвращает состояние эксперимента
get_blocks_ statuses	experiment_id	GetBlocksStatuses	Возвращает состояния шагов эксперимента
copy_blocks	experiment_id , block_ids	CopyExperimentSteps	Копирует шаги эксперимента в буфер
paste_blocks	experiment_id	PasteExperimentSteps	Вставляет шаги эксперимента из буфера
save_experiment	experiment_id	SaveExperiment	Сохраняет изменения в схеме эксперимента
start_experiment	experiment_id	StartExperiment	Запускает эксперимент
start_block	experiment_ step_id	StartBlock	Запускает отдельный шаг эксперимента
stop_experiment	experiment_id	StopExperiment	Прерывает эксперимент
finish_experiment	experiment_id	FinishExperiment	Завершает эксперимент
update_ experiment_status	experiment_id	UpdateExperiment Status	Обновляет состояние эксперимента

Клиентская часть редактора реализована в классах `Workspace`, `WorkspaceContextMenu`, `ElementPanel`, `Block`, `BlockParameter`, `BlockLink`, `BlockWire`.

Класс `Workspace` отвечает за общие функции редактора. Методы класса описаны в таблице 6.

Таблица 6. Методы класса Workspace

Метод	Параметры	Описание
constructor	domElement, experiment_id, role	Инициализирует редактор
visualize	—	Открывает модальное окно с визуализацией выделенного блока
saveAs	—	Открывает модальное окно для сохранения копии эксперимента
deleteExperiment	—	Удаляет эксперимент
startExperiment	—	Запускает эксперимент
stopExperiment	—	Прерывает эксперимент
getBlocksStatuses	—	Получает и отображает состояния шагов эксперимента
appendTo	domElement	Размещает элементы управления редактора в элемент DOM
createBlocks	blocks, wires, offset	Создает блоки
loadExperiment	experiment_id	Загружает эксперимент для редактирования
saveExperiment	callback	Сохраняет изменения в схеме эксперимента
deleteSelection	—	Удаляет выделенные блоки
copySelection	callback	Копирует выделенные блоки в буфер
cutSelection	—	Копирует выделенные блоки в буфер и удаляет их
paste	—	Вставляет блоки из буфера
runSelection	—	Запускает выделенные шаги

Метод	Параметры	Описание
		эксперимента
commentSelection	—	Включает режим редактирования комментария выделенного блока
addBlock	block, setSelection	Добавляет блок в схему эксперимента
removeBlock	block	Удаляет блок из схемы эксперимента
addWire	wire	Добавляет соединение в схему эксперимента
removeWire	wire	Удаляет соединение из схемы эксперимента
computeBlockBounds	blocks	Вычисляет область, занимаемую блоками
computeFit	width, height	Вычисляет масштаб, при котором все блоки помещаются в указанный размер
scaleToFit	—	Изменяет масштаб так, что все блоки помещаются в область видимости
update	—	Обновляет графическое представление схемы

Класс `ElementTree` отвечает за работу дерева элементов. Методы класса описаны в таблице 7.

Таблица 7. Методы класса ElementTree

Метод	Параметры	Описание
constructor	workspace, domElement, role	Инициализирует дерево элементов
reload	—	Перезагружает дерево элементов для текущего эксперимента
load	project_id, callback	Загружает дерево элементов для указанного эксперимента
filter	search	Отображает только те элементы

Метод	Параметры	Описание
		дерева, название которых содержит указанную строку
register	path, blockKind, collapse	Добавляет элемент в дерево
collapseListener	catElem, ev	Обработчик события сворачивания и разворачивания ветвей дерева
hintListener	blockElem, blockKind, ev	Обработчик события отображения всплывающей подсказки
dragListener	blockElem, blockKind, ev	Обработчик события перетаскивания элемента

Класс `WorkspaceContextMenu` отвечает за контекстное меню блока. Методы класса описаны в таблице 8.

Таблица 8. Методы класса WorkspaceContextMenu

Метод	Параметры	Описание
constructor	options	Инициализирует контекстное меню
show	—	Отображает контекстное меню
hide	—	Скрывает контекстное меню

Класс `Block` отвечает за отдельный блок схемы. Методы класса описаны в таблице 9.

Таблица 9. Методы класса Block

Метод	Параметры	Описание
constructor	kind, x, y, comment, calc_type, status, error, percent	Инициализирует блок
dependsOn	other	Проверяет, требуется ли вычисление указанного блока до вычисления данного блока
getConnectedWires	—	Возвращает все соединения

Метод	Параметры	Описание
		данного блока
update	—	Обновляет графическое представление блока
updateStatus	newStatus, errorId	Обновляет состояние блока
updateLinks	kind, params, group	Обновляет входные или выходные точки соединения блока
updateWires	—	Обновляет соединения блока

Класс `BlockParameter` отвечает за группу точек соединения блока, соответствующую одному параметру. Методы класса описаны в таблице 10.

Таблица 10. Методы класса BlockParameter

Метод	Параметры	Описание
constructor	block, parameter	Инициализирует группу
getLink	slot	Возвращает точку соединения, соответствующую указанному номеру
getConnectedWires	—	Возвращает все соединения данного блока
update	—	Обновляет графическое представление

Класс `BlockLink` отвечает за отдельную точку соединения блока. Методы класса описаны в таблице 11.

Таблица 11. Методы класса BlockLink

Метод	Параметры	Описание
constructor	parameter, slot	Инициализирует точку соединения
isWired	—	Проверяет наличие соединений
canConnect	wire	Проверяет корректность соединения
addWire	wire	Добавляет соединение

Метод	Параметры	Описание
removeWire	wire	Удаляет соединение

Класс `BlockWire` отвечает за соединение между блоками. Методы класса описаны в таблице 12.

Таблица 12. Методы класса BlockWire

Метод	Параметры	Описание
constructor	—	Инициализирует соединение
isFullyWired	—	Проверяет наличие обоих концов соединения
unsetSource	—	Сбрасывает источник соединения (выходной параметр)
unsetTarget	—	Сбрасывает назначение соединения (входной параметр)
linkTo	link	Присоединяет к точке соединения
unlinkFrom	link	Отсоединяет от точки соединения
update	—	Обновляет графическое представление

3.7. Вычислительные узлы

Вычислительные узлы отслеживают появления в базе данных информации о блоках, запрошенных на вычисление. Возможен запуск целого эксперимента либо конкретного блока. Порядок вычислений определяется связями между блоками. Вычисления выполняются многопоточно.

Для каждого рассматриваемого блока загружаются его входные датасеты и модели. Для вычисления управление передается методу класса, соответствующего виду блока. Методы могут выдавать оповещения о текущем прогрессе в процентах, которые отображаются в интерфейсе. При ошибке тип

возникшего исключения сохраняется в БД, блок помечается соответствующим статусом, и вычисление эксперимента прекращается.

Датасеты хранятся в виде сериализованных объектов `pandas.DataFrame`.

Модели хранятся в виде сериализованных объектов различных классов библиотеки scikit-learn.

Каждому виду блоков соответствует отдельный класс. Одни блоки работают с датасетом целиком, а другие — с отдельными фрагментами датасета для того, чтобы была возможность использовать параллельную обработку.

Реализованы методы вычисления для различных блоков:

- предварительная обработка данных — математические операции над столбцами, обработка пропусков, фильтрация, преобразование столбцов и т. п.
- модели классификации и регрессии (Decision tree, Random forest, SVC, линейная регрессия и т. п.) — инициализация необученной модели с заданными гиперпараметрами;
- действия — обучение, тестирование, предсказание, кросс-валидация.

Список видов блоков с кратким описанием приведен в таблице 13.

Таблица 13. Список видов блоков

Название	**Описание**
Слияние	Слияние двух датасетов
Удаление дубликатов	Удаление дубликатов из датасета
Обработка пропусков	Заполнение пропусков
Обработка пропусков по регрессии	Заполнение пропусков с использованием регрессии
Выбор признаков	Выбор указанных признаков датасета
Удаление строк	Удаление строк с указанным количеством пропусков
Нормализация значений	Нормализация значений столбца

Название	Описание
Удаление столбцов	Удаление столбцов с указанным количеством пропусков
Конкатенация строк	Конкатенация строк наборов данных с одинаковыми полями
Выборка данных	Выбор строк исходного датасета по условию
Разделение данных	Разделение данных на обучающую и тестовую выборки
Округление	Округление значений в указанном столбце
Дискретизация данных	Преобразование числового признака в категориальный
Матрица корреляции	Рассчитывает коэффициенты корреляции между всеми парами выбранных признаков
Парсинг даты	Преобразование значений столбца из текста в дату
Описательная статистика	Описательная статистика датасета
Decision tree	Инициализация модели классификации на основе дерева решений
Нейронный классификатор	Инициализация модели классификации на основе полносвязной нейронной сети
SVC	Инициализация модели классификации на основе метода SVC
Логистическая регрессия	Инициализация модели классификации на основе логистической регрессии
K ближайших соседей	Инициализация модели классификации на основе алгоритма K ближайших соседей
Случайный лес	Инициализация модели классификации на основе метода «случайный лес»
SGD классификатор	Инициализация модели классификации на основе метода SGD
Градиентный бустинг	Инициализация модели классификации на основе градиентного бустинга
Наивный Байес	Инициализация модели классификации на основе гауссовского наивного байесовского классификатора
AdaBoost	Инициализация модели классификации на основе

Название	Описание
	метода AdaBoost
One-vs-Rest	Инициализация модели многоклассовой классификации на основе метода One-vs-Rest
One-vs-One	Инициализация модели многоклассовой классификации на основе метода One-vs-One
Классификатор голосованием	Объединяет несколько моделей классификации на основе голосования
GPC	Инициализация модели классификации на основе метода GPC
Кросс-валидация	Оценка обучения модели с помощью кросс-валидации
Elastic net	Инициализация модели регрессии на основе метода Elastic net
Линейная регрессия	Инициализация модели регрессии на основе линейной регрессии
K-средних	Инициализация модели кластеризации на основе метода K-средних
Предсказание	Предсказание целевых значений
Обучение	Обучение модели на датасете
Тестирование	Проверка точности модели
Визуализация классов	Визуализация классов датасета
Тест с визуализацией	Тестирование с визуализацией правильно классифицированных объектов

Блок «Тестирование» — один из основных блоков в системе. Этапы вычисления для этого блока состоят в следующем (см. приложение 1):

1. Столбец с целевым признаком отделяется от входного датасета.
2. На основе входной модели формируется столбец с предсказанными значениями целевого признака.

3. При решении задачи классификации из модели дополнительно достаются вероятности принадлежности объектов классам, если модель предоставляет эту информацию.
4. В зависимости от типа задачи (классификация или регрессия) рассчитываются метрики для модели.
    - Для моделей классификации рассчитываются точность, precision и recall, F1-мера, а также при возможности log loss и ROC-AUC.
    - Для моделей регрессии рассчитываются средняя абсолютная ошибка и коэффициент детерминации
5. Если решается задача бинарной классификация, то дополнительно рассчитывается кривая ROC.
6. В качестве выходных значений возвращаются дополненный столбцом с предсказанными значениями исходный датасет, датасет со значениями метрик, а также датасет с точками кривой ROC, если она была рассчитана.

3.8. Визуализация результатов

За визуализацию результатов вычисления отвечает отдельный контроллер `CWControllerVisualize`. При запросе визуализации выходного параметра блока на сервере из БД получается информация о виде блока и его входных и выходных параметрах. В зависимости от вида блока и номера выходного параметра выделяется нужная информация из датасетов и выбирается класс JavaScript, используемый для отображения информации на странице.

При визуализации всех выходных параметров результаты визуализации каждого параметра автоматически объединяются и отображаются пользователю в отдельных вкладках. Пример визуализации показан на рисунке 9.

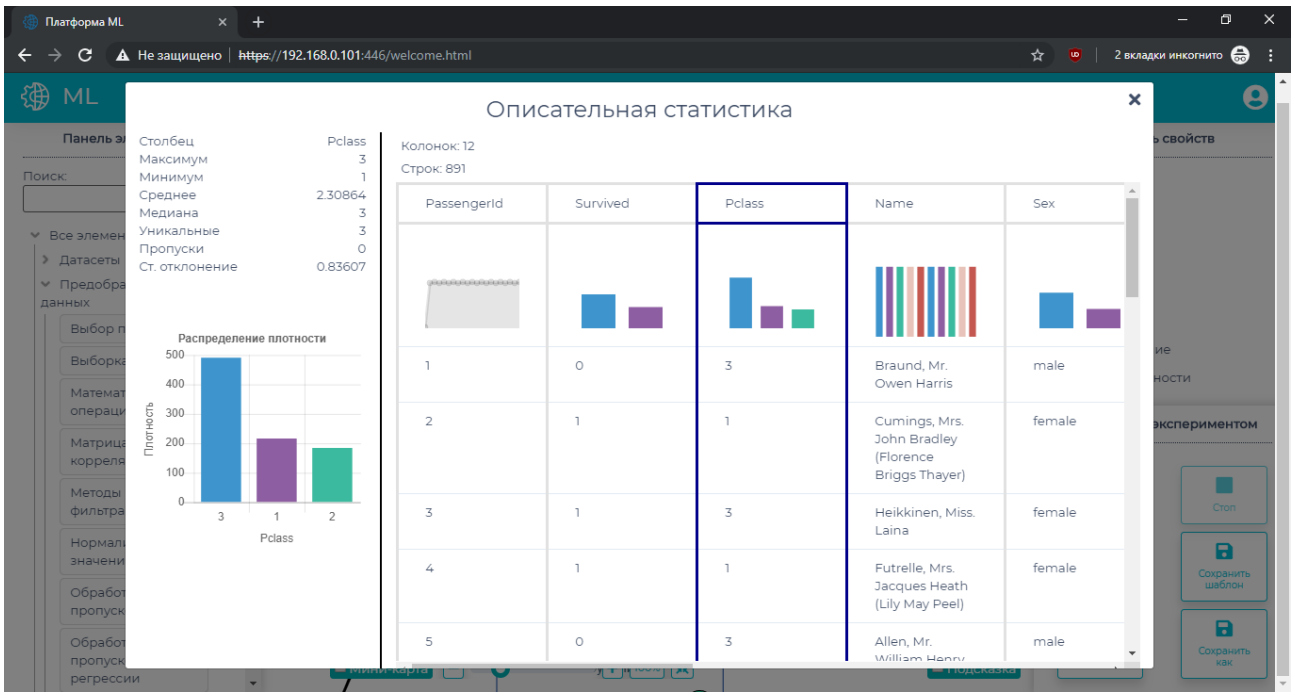

Рисунок 9. Визуализация блока «Описательная статистика»

Описание классов, используемых для визуализации, представлен в таблице 14.

Таблица 14. Классы, используемые для визуализации

Класс	Описание
CViewBase	Базовый класс для визуализации
CViewDataset	Отображает датасет в виде таблицы
CViewImage	Отображает изображение
CViewChart	Отображает график
CViewStatistics	Визуализирует описательную статистику датасета
CViewMetrics	Визуализирует результат тестирования модели
CViewClassVisualize	Визуализирует распределение классов
CViewMultiple	Отображает несколько визуализаций на отдельных вкладках

4. ТЕСТИРОВАНИЕ ПЛАТФОРМЫ

Для проверки работы системы был проведен эксперимент. В качестве набора данных был взят отчет ресторана быстрого питания «Tesla Burger» о доставке блюд ресторана и корпоративных обедов за период с 2017 г. по 2019 г.

4.1. Подготовка данных для платформы

Отчет содержит информацию о дате и времени заказа, клиенте, адресе доставки, содержании и общей сумме заказа.

Изначально отчет был представлен в формате XLS с группировкой по дате и клиенту. Первоначальный вид отчета показан на рисунке 10.

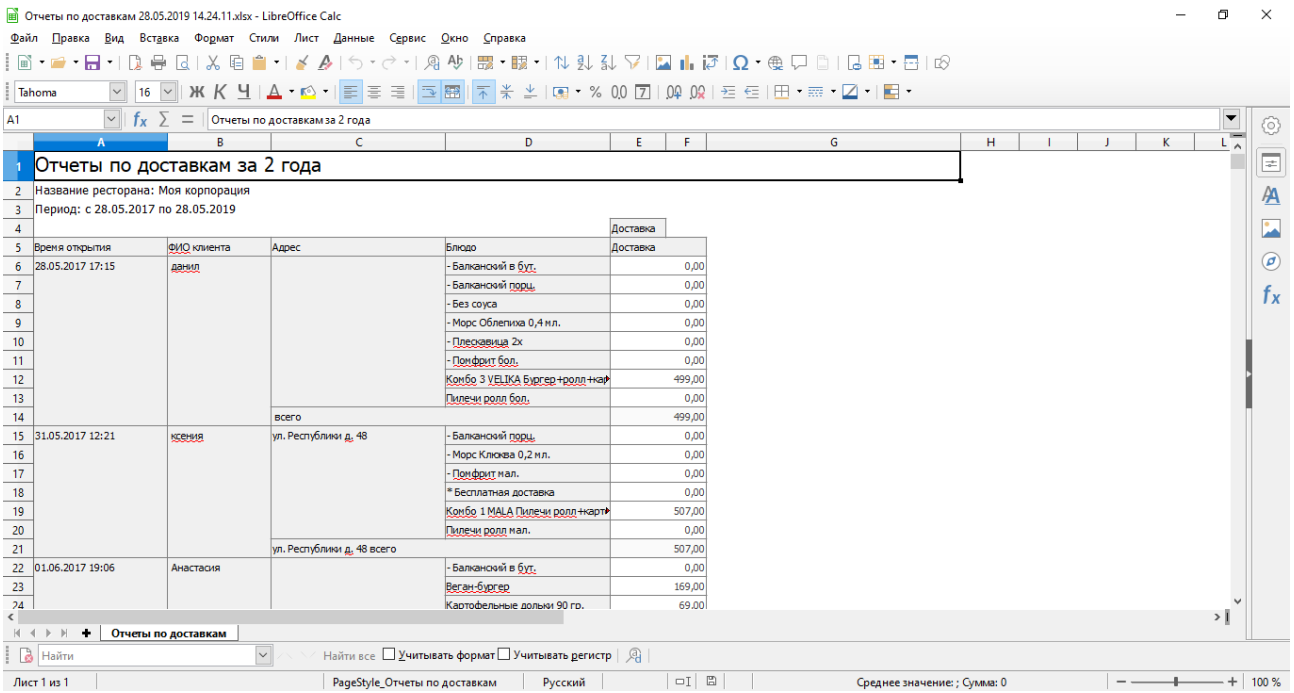

Рисунок 10. Исходный формат отчета

Так как платформа работает с датасетами в табличной форме, был написан отдельный скрипт для преобразования отчета в таблицу (см. приложение 2).

Скрипт выполняет следующую последовательность операций:

1. Считываются строки исходного отчета.

2. Восстанавливаются значения ячеек, подразумеваемые группировкой.
3. Строки группируются по дате заказа и клиенту.
4. Из каждой группы строк выделяются характеристики заказа и состав заказа в виде списка наименований блюд.
5. Формируется множество всех наименований блюд в отчете для определения набора столбцов в выходном файле.
6. Характеристики и составы заказов выводятся в выходной файл.

В результате выполнения скрипта получается датасет со столбцами:

- год
- месяц
- день
- час
- минута
- клиент
- адрес
- улица
- номер дома
- номер квартиры
- сумма заказа
- наименование 1, наименование 2, …, наименование N — отдельный столбец для каждого блюда в меню, показывающий наличие этого блюда в заказе.

Полученный датасет показан на рисунке 11.

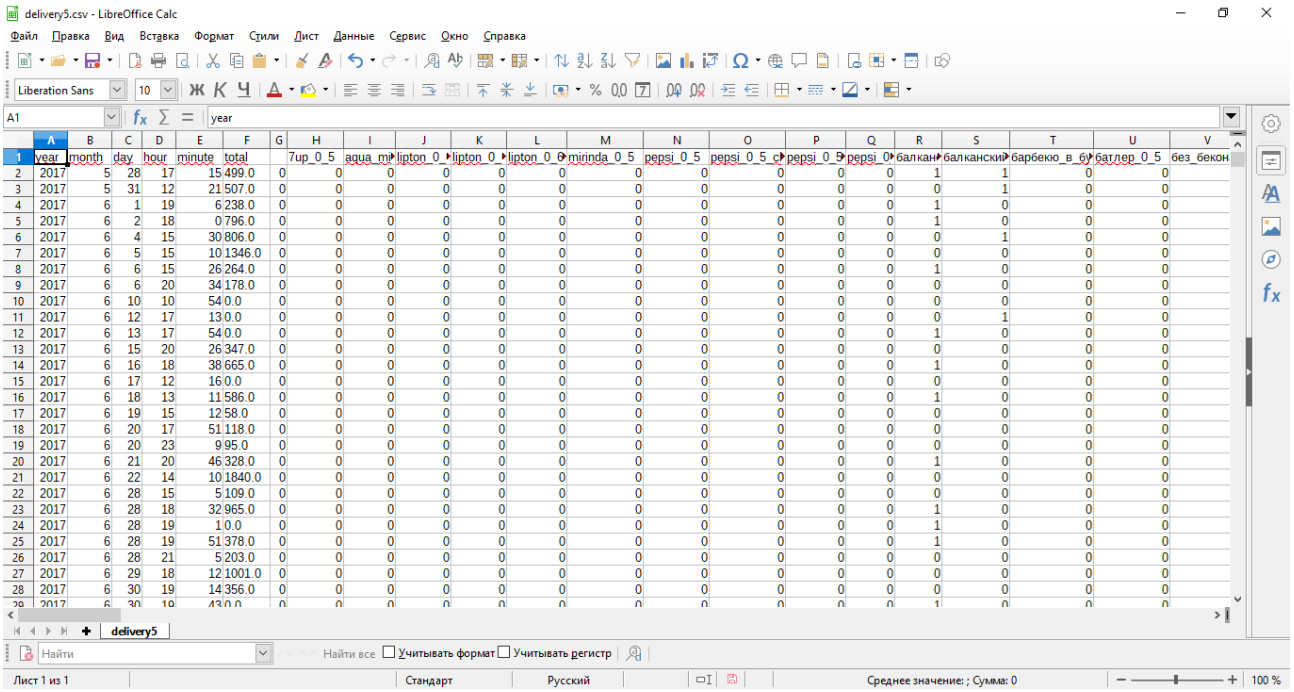

Рисунок 11. Полученный датасет

4.2. Описательная статистика

После загрузки датасета для визуального представления его основных характеристик был использован блок «Описательная статистика».

Блок «Описательная статистика» показывает базовые статистические характеристики и гистограмму для каждого столбца.

Результат показан на рисунке 12.

Из гистограмм видно, какие блюда чаще всего заказываются, а какие непопулярны. Также можно увидеть среднюю, минимальную и максимальную сумму заказа, количество различных клиентов и адресов. Здесь же можно увидеть количество пропущенных значений в каждом столбце.

В рассматриваемом датасете пропуски содержались только в столбце адреса.

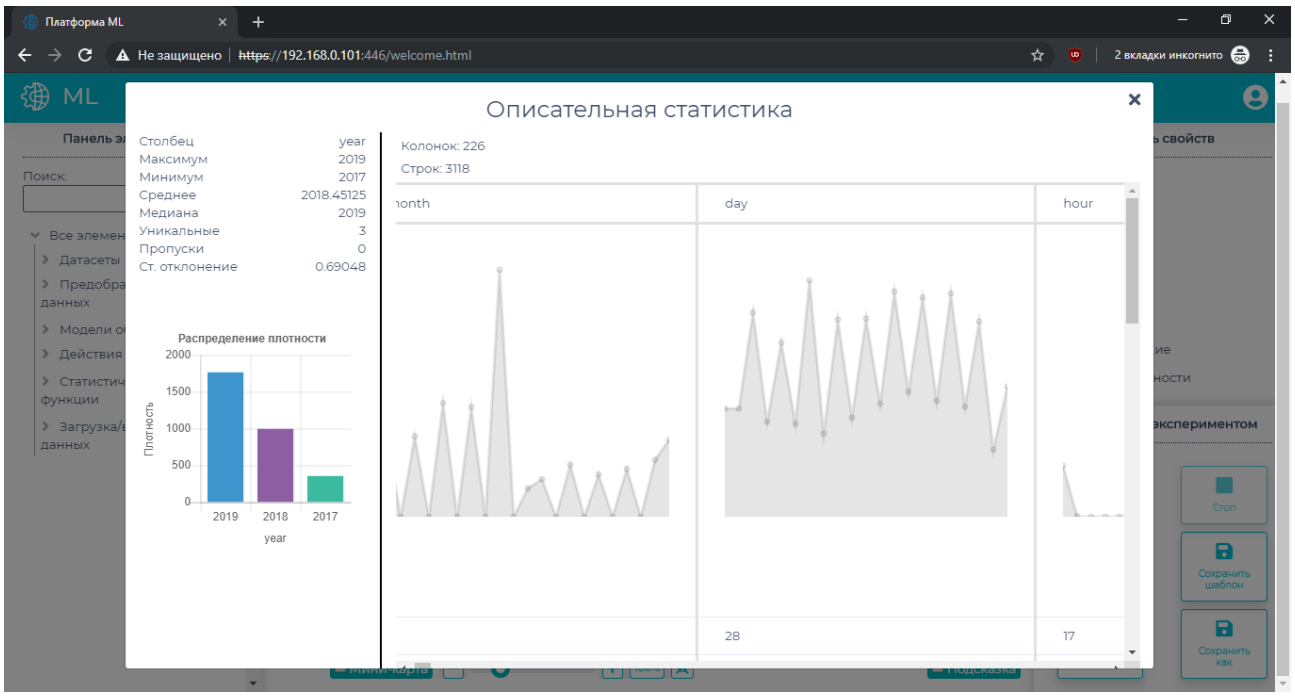

Рисунок 12. Описательная статистика

4.3. Обработка пропусков

Для заполнения пропущенных значений используется блок «Обработка пропусков».

Блок позволяет заполнить пропуски в указанных столбцах нулевым, средним, самым частым или указанным пользователем значением.

В датасете пропущенные адреса были заменены на адрес ресторана.

Результат показан на рисунке 13.

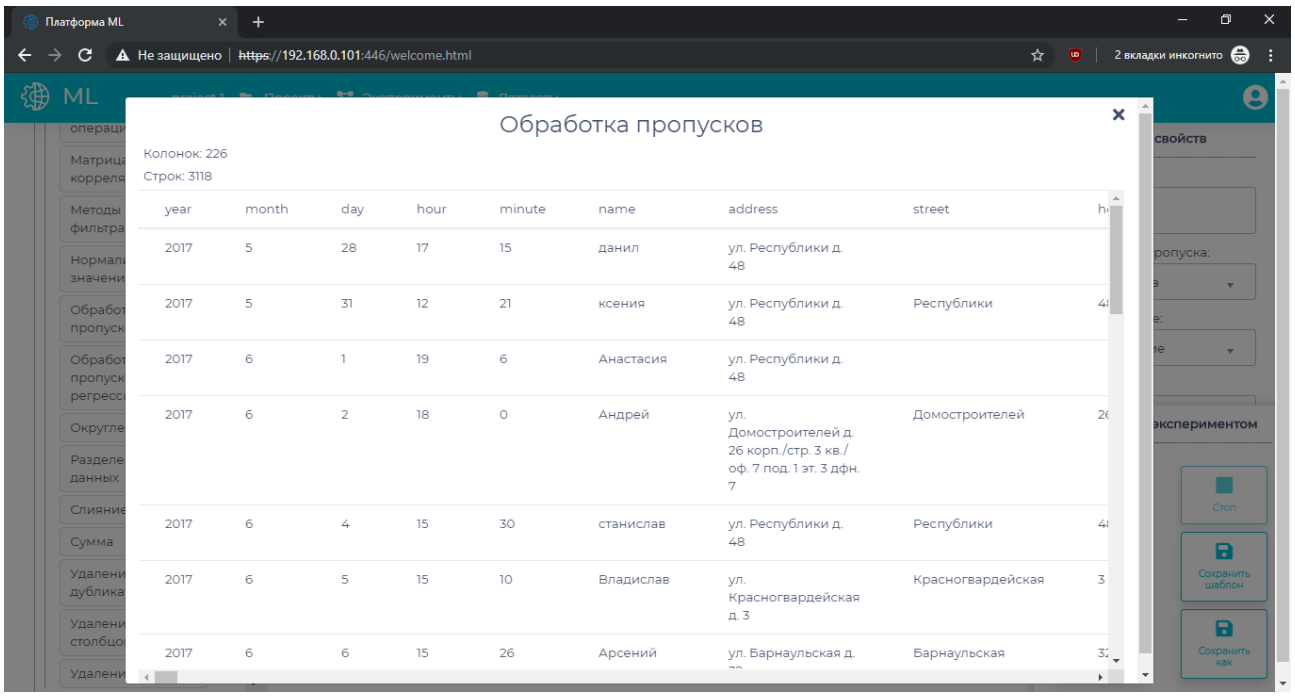

Рисунок 13. Обработка пропусков

4.4. Корреляция

Для анализа зависимостей между столбцами используется блок «Корреляция».

Блок отображает матрицу корреляции для выбранных пользователем столбцов.

Результат показан на рисунке 14.

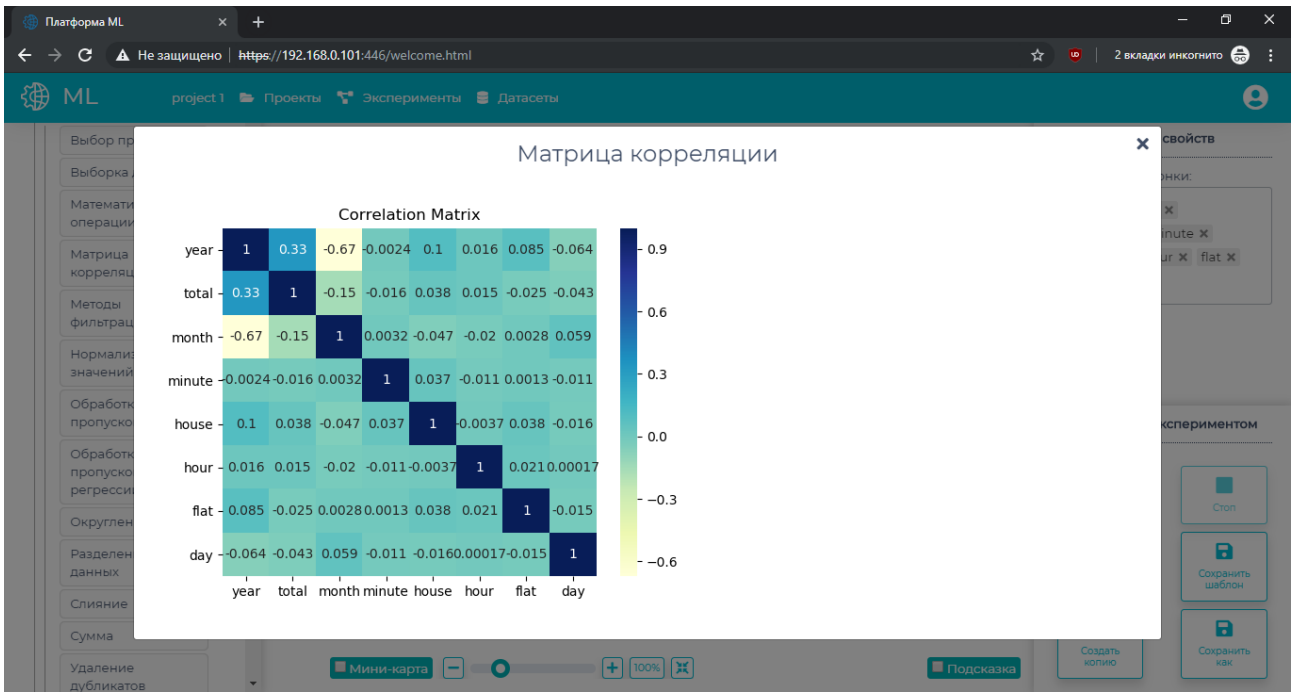

Рисунок 14. Матрица корреляции

4.5. Разделение данных

Перед тем как приступить к обучению модели, необходимо разделить датасет на обучающую и тестовую выборки. Для этого используется блок «Разделение данных».

Блок «Разделение данных» позволяет выбрать способ разделения — по порядку, случайный или с сохранением пропорции по классам, а также процент данных для обучающей выборки. При этом учитывается то, что в обучающей выборке должны присутствовать все возможные значения категориальных признаков.

В эксперименте был выбрано разделение случайным способом в соотношении 70:30.

Результат показан на рисунке 15.

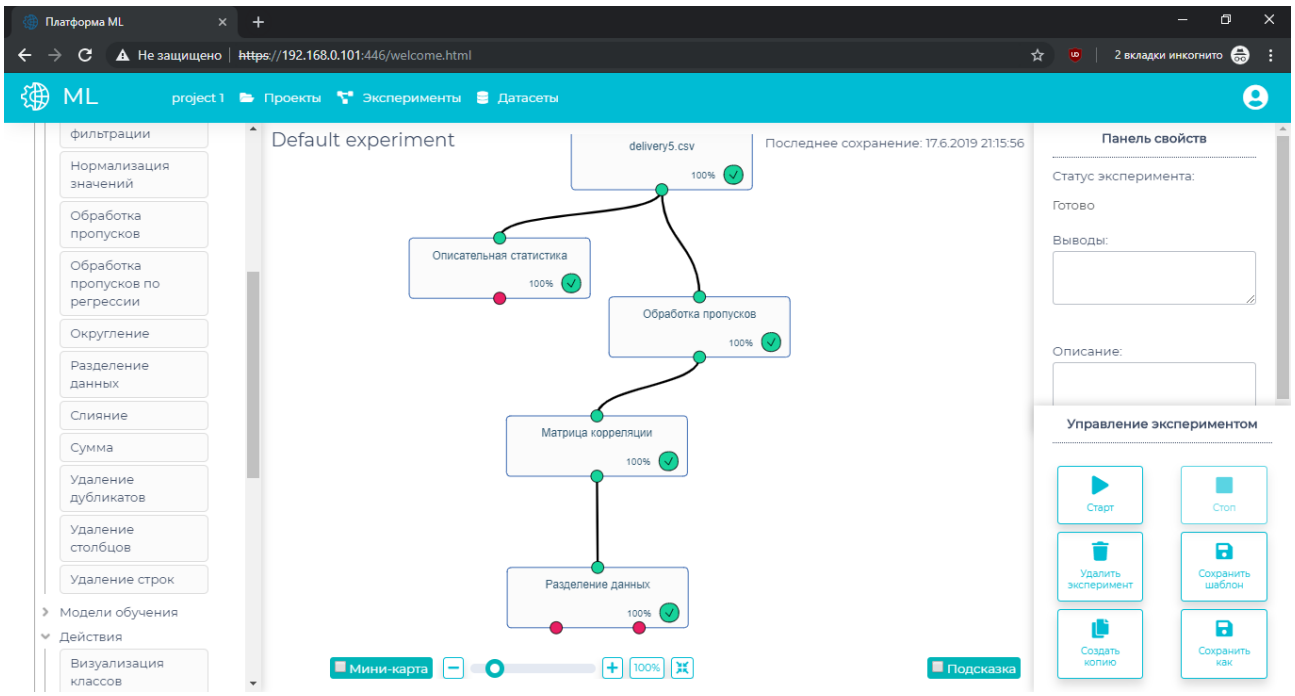

Рисунок 15. Разделение данных

4.6. Выбор модели

На основе рассматриваемого датасета не ожидалось получить эффективной модели, но для проверки системы было решено обучить модель для предсказания суммы заказа.

Для решения такой задачи в платформе предназначены блоки категории «Регрессия». Каждый из этих блоков соответствует некоторому алгоритму машинного обучения, например линейной регрессии или Elastic Net, и позволяет задавать гиперпараметры алгоритма.

Результатом работы блока является необученная модель. В эксперименте был выбран алгоритм Elastic Net. Полученная схема эксперимента показана на рисунке 16.

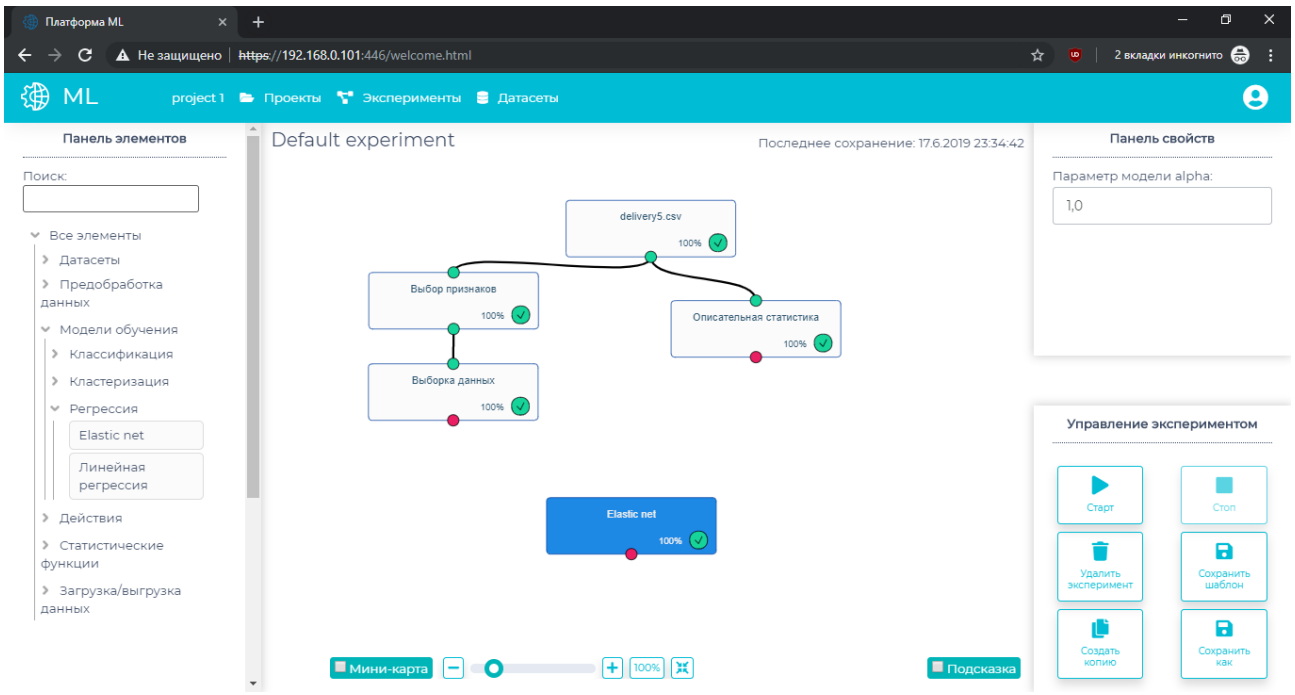

Рисунок 16. Блок модели

4.7. Обучение и тестирование модели

Для обучения модели используется блок «Обучение».

Входными данными блока являются необученная модель и обучающая выборка. Блок позволяет указать целевой столбец. Результатом работы блока является обученная модель

Для тестирования модели используется блок «Тестирование».

Входными данными блока являются обученная модель и тестовая выборка. Блок выполняет тестирование и рассчитывает стандартные метрики модели. Для классификации выводятся точность, precision, recall, F1-мера и ROC-AUC, для регрессии — средняя абсолютная ошибка и средняя квадратичная ошибка.

На рисунке 17 показаны результаты тестирования.

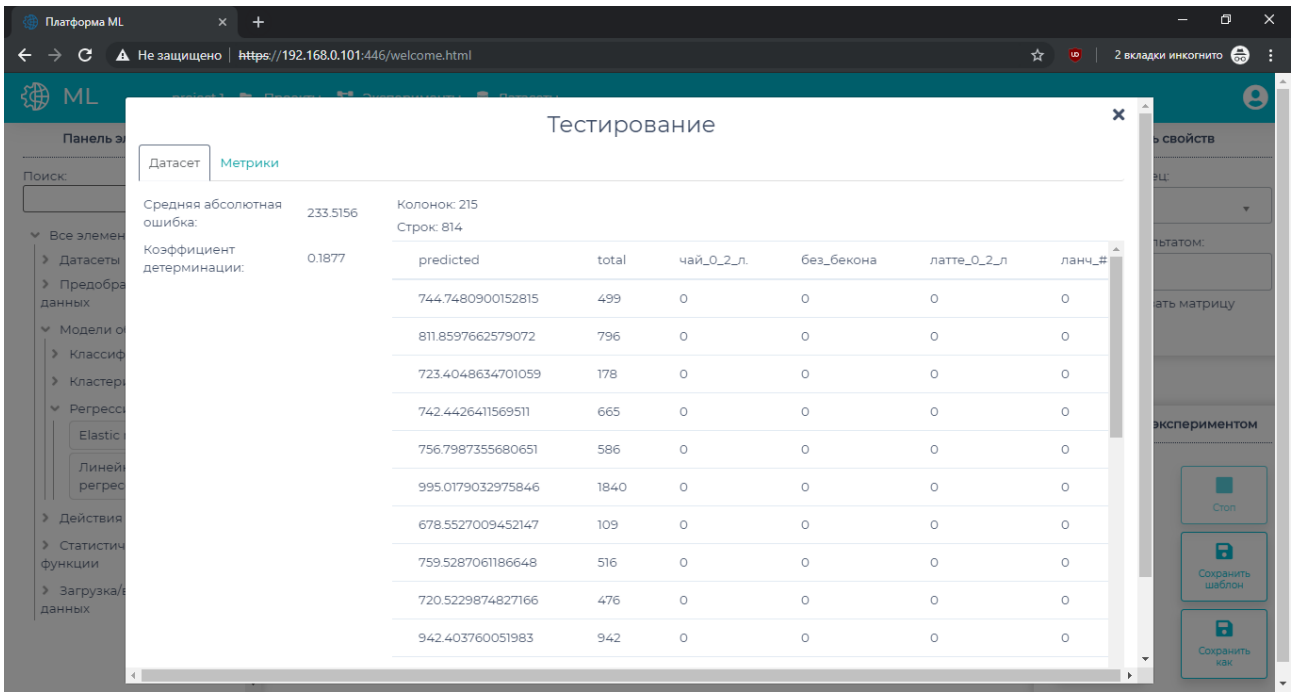

Рисунок 17. Результаты тестирования

4.8. Изменение набора признаков

После оценки результатов тестирования было сделано предположение, что высокая точность предсказания вызвана наличием состава заказа в данных.

Поскольку такое предсказание не имеет практического смысла, было решено повторить обучение без этой информации. Для этого перед блоком «Разделение данных» в эксперимент был добавлен блок «Выбор признаков».

Измененная схема эксперимента показана на рисунке 18.

Блок «Выбор признаков» позволяет оставить в датасете только выбранные столбцы.

После перезапуска эксперимента были получены результаты тестирования новой модели.

Результаты показаны на рисунке 19.

Как и ожидалось, ошибка предсказания заметно возросла.

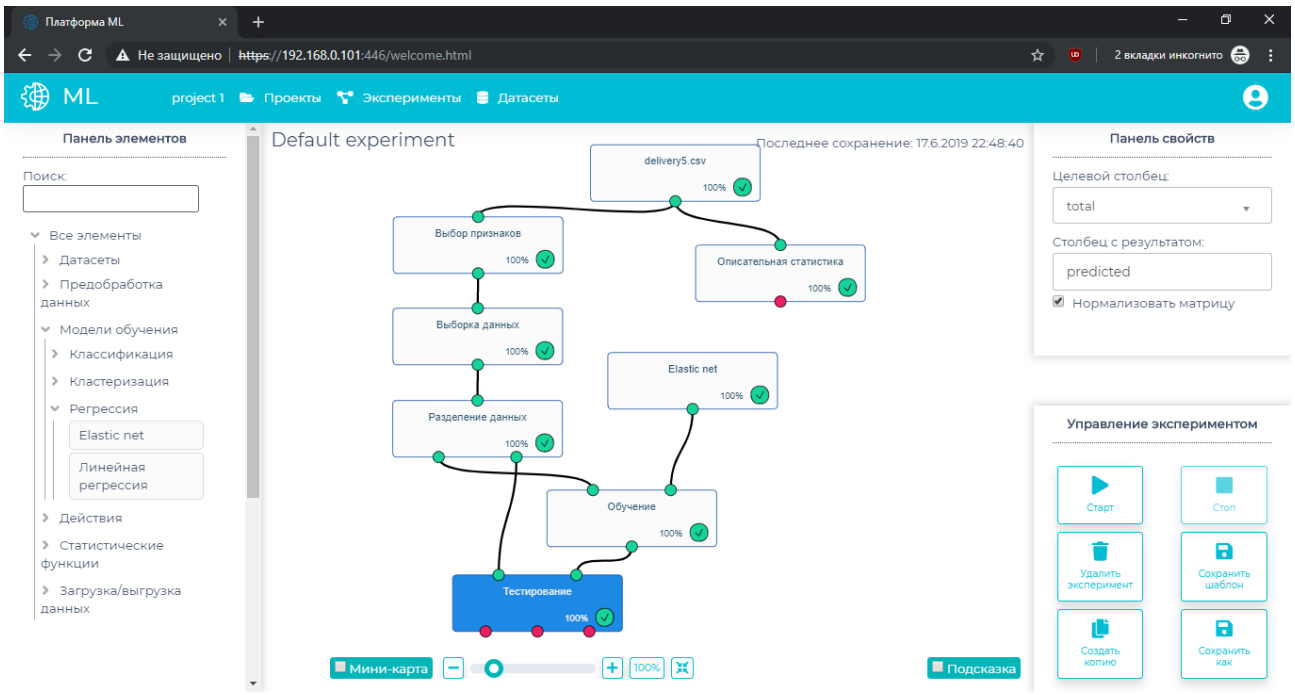

Рисунок 18. Выбор признаков

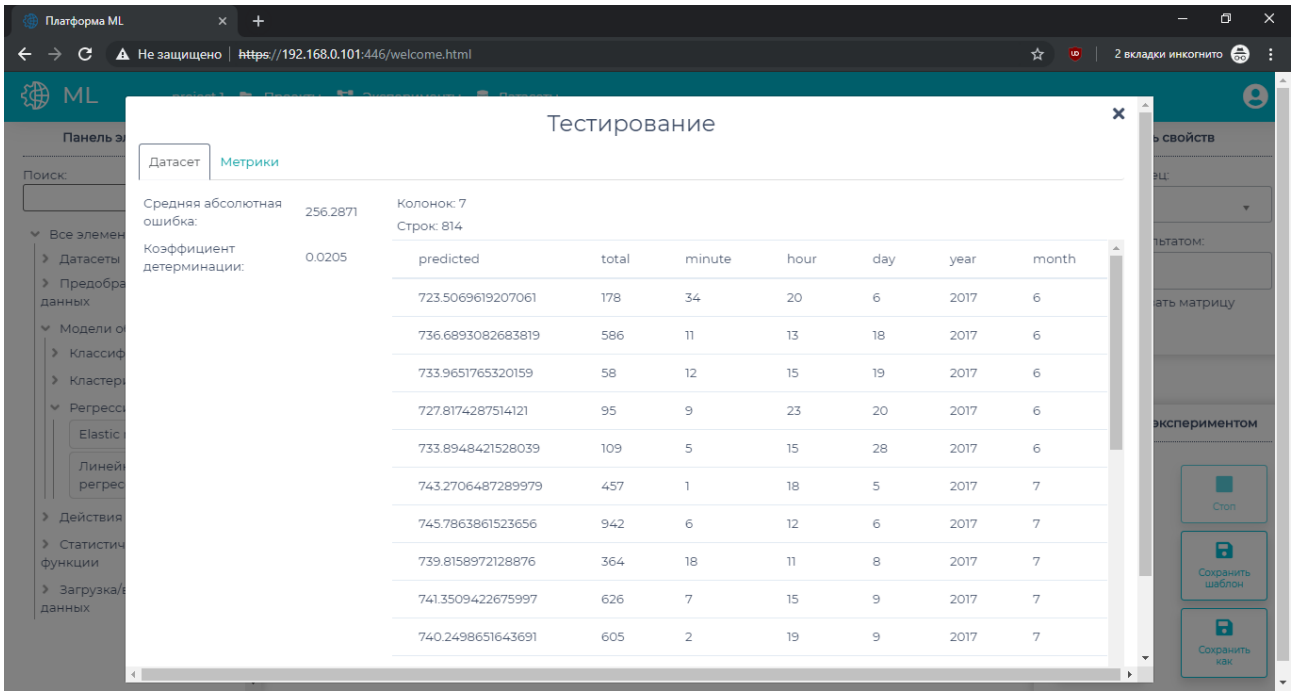

Рисунок 19. Результаты после изменения набора признаков

ЗАКЛЮЧЕНИЕ

В результате работы был выполнен обзор систем, упрощающих использование подходов машинного обучения для решения бизнес-задач.

Для платформы машинного обучения, разрабатываемой IT-специалистами ТюмГУ, был разработан визуальный редактор экспериментов с функциями добавления, перемещения, соединения, удаления, копирования и вставки шагов эксперимента. Интерфейс редактора также предоставляет возможности запуска эксперимента и визуализации промежуточных и финальных результатов.

Также были реализованы вычислительные операции для инициализации моделей, их обучения и тестирования, предсказания на их основе. Операции встроены в платформу и доступны из редактора экспериментов.

В будущем возможно расширение функционала платформы путем добавления новых вычислительных операций.

СПИСОК ЛИТЕРАТУРЫ И ИНТЕРНЕТ-РЕСУРСОВ

1. Dudley J. J., Kristensson P. O. A Review of User Interface Design for Interactive Machine Learning //ACM Transactions on Interactive Intelligent Systems (TiiS). – 2018. – Т. 8. – №. 2. – С. 8.
2. Feurer M. et al. Efficient and robust automated machine learning //Advances in Neural Information Processing Systems. – 2015. – С. 2962-2970.
3. Pedregosa F. et al. Scikit-learn: Machine learning in Python //Journal of machine learning research. – 2011. – Т. 12. – №. Oct. – С. 2825-2830.
4. Yang Q. et al. Grounding Interactive Machine Learning Tool Design in How Non-Experts Actually Build Models //Proceedings of the 2018 on Designing Interactive Systems Conference 2018. – ACM, 2018. – С. 573-584.
5. Артиков М. Э. Онлайн платформы машинного обучения // Молодой ученый. — 2016. — №12.4. — С. 11-13. — URL https://moluch.ru/archive/116/32168/ (дата обращения: 11.05.2019).
6. Гринберг М. Разработка веб-приложений с использованием Flask на языке Python. — Перевод с английского. — М.: ДМК Пресс, 2016.
7. Жерон О. Прикладное машинное обучение с помощью Scikit-Learn и TensorFlow. Концепции, инструменты и техники. — Перевод с английского. — М.: Вильямс, 2018 г..
8. Саммерфилд М. Python на практике. — Перевод с английского. — М.: ДМК Пресс, 2014.

Приложение 1

Фрагмент кода блока «Тестирование»

```
def calculate(self, args):
    target_column = self.GetPropertiesValue('selected_columns')
    result_column = self.GetPropertiesValue('result_col')

    model = self.GetModel()

    dataset = self.GetFullDataset(target_column)

    y = dataset[target_column]

    X = dataset
    X.drop(target_column, axis=1, inplace=True)

    yres = model.predict(X)

    yscore = None
    if hasattr(model, 'predict_proba'):
        yscore = model.predict_proba(X)

    result_df = pandas.concat([y, X], axis=1)
    result_df.insert(0, result_column, yres)

    self.report_progress(33)
    metrics = []

    if model._estimator_type == ClassifierMixin._estimator_type:
        metrics.append(['Точность', accuracy_score(y, yres)])
        metrics.append(['Precision', precision_score(y, yres, average='macro')])
        metrics.append(['Recall', recall_score(y, yres, average='macro')])
        metrics.append(['F1-мера', f1_score(y, yres, average='macro')])
        if yscore is not None:
            metrics.append(['Log loss', log_loss(y, yscore)])
        if yscore is not None and yscore.shape[1] == 2:
            metrics.append(['ROC-AUC', roc_auc_score(y, yscore[:,1], average='macro')])
    elif model._estimator_type == RegressorMixin._estimator_type:
        metrics.append(['Средняя абсолютная ошибка', mean_absolute_error(y, yres)])
        metrics.append(['Коэффициент детерминации', r2_score(y, yres)])

    metrics_df = pandas.DataFrame(metrics, columns = ['metric', 'score'])
    results = [result_df, metrics_df]
    self.report_progress(66)

    if yscore is not None and yscore.shape[1] == 2:
        fpr, tpr, thresholds = roc_curve(y, yscore[:,1])
        roc_df = pandas.DataFrame(
            {'fpr': fpr, 'tpr': tpr, 'thresholds': thresholds},
            columns = ['fpr', 'tpr', 'thresholds'])
        results.append(roc_df)
    else:
        results.append(None)
    self.report_progress(99)

    return results, ['dataset_and_metrics', 'metrics', 'roc_auc']
```

Приложение 2

Код скрипта для преобразования отчета о доставках в табличный вид

```
import argparse
import xlrd
from xlrd import xldate
import re
import csv
import sys
import itertools

class Converter:
    base_header = (
        'year', 'month', 'day', 'hour', 'minute', 'name',
        'address', 'street', 'house', 'flat', 'total',
    )

    _address_re = re.compile(r'ул\. (.*?) д\. +(\d+)(?:.*кв\./оф\. (\d+))?')
    _header_trans = str.maketrans(' ,', '__')

    def __init__(self, max_total=None, count_doubles=False,
                 normalize_items=False, default_address=None):
        self.max_total = max_total
        self.count_doubles = count_doubles
        self.normalize_items = normalize_items
        self.default_address = default_address

        self._datemode = None
        self.items = None

    def _expand_report(self, rows, header_cols):
        header = [''] * header_cols
        for row in rows:
            if not row[header_cols]:
                continue
            header_start = None
            for i in range(header_cols):
                if row[i]:
                    header_start = i
                    break
            if header_start is not None:
                header[header_start:] = row[header_start:header_cols]
            yield header + row[header_cols:]

    def _make_order(self, rows):
        year, month, day, hour, minute, nearest_second = \
            xldate.xldate_as_tuple(rows[0][0], self._datemode)
        client = rows[0][1].strip()
        address = next((row[2] for row in rows if row[2]), self.default_address or '')
        address_match = self._address_re.match(address)
        if address_match:
            street, house, flat = address_match.group(1, 2, 3)
        else:
            street, house, flat = '', '', ''
            if address:
                print('bad address: ' + repr(address), file=sys.stderr)
        total = round(sum(row[4] for row in rows), 2)
        item_set = {row[3].lstrip('-+* ').rstrip() for row in rows if row[3]}
        if self.normalize_items:
```

```
            item_set = {item.lower() for item in item_set}
        items = {item: 1 for item in item_set}
        if self.count_doubles:
            doubles = [(item[:-3], item) for item in items if item.endswith(' 2x')]
            for item, double in doubles:
                items[item] = items.get(item, 0) + items[double] * 2
                del items[double]
        return (year,month,day,hour,minute,client,address,street,house,flat,total), items

    def _get_result_rows(self, rows):
        orders = []
        all_items = set()
        for key, group in itertools.groupby(rows, lambda row: row[:2]):
            values, items = self._make_order(list(group))
            all_items.update(items)
            if self.max_total is None or values[-1] <= self.max_total:
                orders.append((values, items))
        self.items = tuple(sorted(all_items, key=lambda s: (s.lower(), s)))
        header = self.base_header + self.items
        header = tuple(col.translate(self._header_trans) for col in header)
        rows = [values + tuple(items.get(item, 0) for item in self.items)
                for values, items in orders]
        return header, rows

    def convert(self, input, output):
        with xlrd.open_workbook(input, on_demand=True) as wb:
            self._datemode = wb.datemode
            ws = wb.sheet_by_index(0)
            rows = (ws.row_values(i, 0, 5) for i in range(5, ws.nrows-1))
            expanded_rows = self._expand_report(rows, 3)
            header, result_rows = self._get_result_rows(expanded_rows)
            with open(output, 'w', encoding='utf-8', newline='') as csvfile:
                wr = csv.writer(csvfile)
                wr.writerow(header)
                for row in result_rows:
                    wr.writerow(row)

def convert(input, output, item_output=None, **kwargs):
    conv = Converter(**kwargs)
    conv.convert(input, output)
    if item_output is not None:
        with open(item_output, 'w', encoding='utf-8') as f:
            for item in conv.items:
                print(item, file=f)

def main():
    parser = argparse.ArgumentParser('delivery_report_convert')
    parser.add_argument('input')
    parser.add_argument('output')
    parser.add_argument('-m', '--max-total', type=float)
    parser.add_argument('-d', '--default-address', type=str)
    parser.add_argument('-x', '--count-doubles', action='store_true')
    parser.add_argument('-n', '--normalize-items', action='store_true')
    parser.add_argument('-t', '--item-output')
    args = parser.parse_args()
    convert(**vars(args))

if __name__ == '__main__':
    main()
```