

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ
ВЫСШЕГО ОБРАЗОВАНИЯ
НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ
«ВЫСШАЯ ШКОЛА ЭКОНОМИКИ»
Санкт-Петербургская школа физико-математических и компьютерных
наук

Шелухина Екатерина Андреевна

Разработка клиентской части веб-сервиса для платформы машинного обучения без программирования Robbi

Бакалаврская работа

Научный руководитель:
д. ф.-м. н., профессор Новиков Б. А.

Рецензент:
ООО "ВК", Лидер группы Радыгин К. В.

Санкт-Петербург
2023

Оглавление

Реферат	3
Abstract	4
Определения, обозначения и сокращения	5
Введение	7
Глава 1. Анализ существующих решений	10
1. Обзор существующих решений	10
1.1. Партнерство с консультантами по машинному обучению	10
1.2. Автоматизация процесса использования моделей	11
1.3. Использование готовых решений	12
2. Сравнение Robbi с аналогами	13
2.1. Удобство использования и эффективность	14
3. Выводы по главе	17
Глава 2. Функции и особенности Robbi	18
1. Список основных функций веб-сервиса	18
2. Главный сценарий использования веб-сервиса	18
3. Настройка параметров модели и обучения	19
4. Управление состояниями обучения	23
5. Выводы по главе	24
Глава 3. Технологии, архитектура и реализация клиентской части веб-сервиса	25
1. Языки программирования	25
2. Архитектура веб-сервиса	26
3. Взаимодействие пользовательского интерфейса и бекенда	27
3.1. Запросы от фронтенда к бекенду	27
3.2. Обновление состояний в пользовательском интерфейсе через бекенд	28
4. Описание пользовательского интерфейса	29
5. Выводы по главе	34
Заключение	35
Список литературы	36

Реферат

Этот дипломный проект посвящен разработке клиентского веб-сервиса для платформы Robbi, платформы машинного обучения без программирования. Цель данной работы заключается в создании веб-сервиса с пользовательским интерфейсом, взаимодействующим с бэкэндом без необходимости владения программированием. Для достижения этой цели были поставлены следующие задачи: (1) разработка пользовательского интерфейса, соответствующего основным сценариям использования сервиса; (2) разработка бэкэнд-компонента, взаимодействующего с пользовательским интерфейсом для обработки запросов пользователей; (3) реализация функциональности создания и конфигурирования моделей машинного обучения без программирования в рамках веб-сервиса.

Результатом данной работы является веб-сервис, построенный на платформе Robbi, включающий клиентскую часть с пользовательским интерфейсом, обеспечивающим удобное взаимодействие с моделями машинного обучения. Этот сервис позволяет пользователям без навыков программирования создавать, настраивать и использовать модели машинного обучения, упрощая и ускоряя процесс и повышая доступность и применимость машинного обучения.

Будущие исследования и разработки могут быть сосредоточены на расширении функциональности веб-сервиса, включении новых типов моделей машинного обучения и обеспечении безопасности данных. Кроме того, следует учесть улучшение пользовательского интерфейса и проведение тщательного тестирования, чтобы обеспечить надежность и эффективность системы.

Данная работа состоит из 40 страниц, 3 глав, 15 рисунков, 2 таблиц. Использовано 34 источника.

Ключевые слова: веб-сервис; клиентская часть; пользовательский интерфейс; платформа машинного обучения без программирования;

Abstract

This diploma project is dedicated to the development of a client web service for the Robbi platform, a machine learning platform without programming. The purpose of this work is to create a web service with a user interface that interacts with the backend without the need for programming skills. To achieve this goal, the following tasks were set: (1) development of a user interface corresponding to the main scenarios of using the service; (2) development of a backend component that interacts with the user interface to process user requests; (3) implementation of the functionality of creating and configuring machine learning models without programming within a web service.

The result of this work is a web service built on the Robbi platform, including a client part with a user interface that provides convenient interaction with machine learning models. This service allows users without programming skills to create, configure and use machine learning models, simplifying and speeding up the process and increasing the availability and applicability of machine learning.

Future research and development may focus on expanding the functionality of the web service, including new types of machine learning models and ensuring data security. In addition, user interface improvements and thorough testing should be considered to ensure the reliability and efficiency of the system.

This work consists of 40 pages, 3 chapters, 15 figures, 2 tables. 34 sources were used.

keywords: web service; client part; user interface; no-code machine learning platform;

Определения, обозначения и сокращения

1. Машинное обучение (МО)[29] - это подраздел искусственного интеллекта, который изучает и разрабатывает алгоритмы и модели, позволяющие компьютерным системам обучаться и делать прогнозы или принимать решения на основе данных без явного программирования.
2. Платформа машинного обучения[10] - это среда или инструментарий, предназначенный для разработки, обучения и использования моделей машинного обучения. Он обеспечивает функциональность, инструменты и ресурсы, необходимые для работы с данными, создания моделей и выполнения задач машинного обучения.
3. Искусственный интеллект (ИИ)[26] - это область компьютерных наук, которая изучает создание интеллектуальных систем, способных выполнять задачи, требующие человеческого интеллекта. Это включает в себя способность к обучению, принятию решений, распознаванию образов, обработке естественного языка и многому другому.
4. Веб-сервис[32] - это программное обеспечение, предоставляющее функциональность и возможности через сеть Интернет. Он позволяет клиентским приложениям взаимодействовать с удаленными серверами, обмениваться данными и выполнять различные операции. Веб-сервисы часто используют стандарты, такие как HTTP, XML и JSON, для коммуникации и предоставления данных.
5. Клиентская часть[28] - это компонент программного приложения, который работает на стороне клиента, то есть на устройстве пользователя. Это часть приложения, с которой пользователь взаимодействует напрямую. Она обеспечивает пользовательский интерфейс и передает запросы на серверную часть приложения для обработки.
6. Бэкэнд[27] - это компонент системы, отвечающий за обработку данных и бизнес-логику, работающий на серверной стороне. Он обеспечивает хранение, обработку и передачу данных между клиентской частью и другими системами. Бэкэнд часто включает в себя базы данных, серверы приложений и другие компоненты, необходимые для обработки запросов и выполнения операций.
7. Модели машинного обучения[30] - это алгоритмы и структуры данных, созданные с помощью машинного обучения для решения конкретных задач. Они основаны на обучении на основе данных и позволяют системе автоматически обнаруживать шаблоны, делать прогнозы и принимать решения без явного програм-

мирования. Модели машинного обучения могут быть использованы для классификации, регрессии, кластеризации и других задач.

8. HTTP (Hypertext Transfer Protocol)[13] - это протокол передачи данных, используемый для обмена информацией веб-серверами и клиентскими приложениями. Он определяет формат запросов и ответов, которые могут быть отправлены между сервером и клиентом, и обеспечивает коммуникацию и передачу данных в Интернете.
9. Пользовательский интерфейс (UI)[31] - это часть программного приложения, предназначенная для взаимодействия с пользователем. Он обеспечивает способ ввода данных, отображения информации и взаимодействия с функциональностью приложения. Цель пользовательского интерфейса - обеспечить удобство, понятность и эффективность взаимодействия между пользователем и приложением.
10. Конфигурирование моделей[20] - процесс настройки параметров моделей машинного обучения для достижения оптимальных результатов. Это включает выбор алгоритмов, определение гиперпараметров, установку правил обучения и прочие настройки, которые влияют на поведение и эффективность моделей. Конфигурирование моделей позволяет адаптировать их под конкретные задачи и данные.
11. Robbi[17] - платформа машинного обучения компании "Одноклассники", предназначенная для упрощения процесса создания, настройки и использования моделей машинного обучения.
12. YAML (Yet Another Markup Language)[33] - формат сериализации данных, основанный на читаемости для людей. Он используется для представления структурированных данных, таких как конфигурационные файлы, и обладает простым синтаксисом, основанном на отступах. YAML позволяет представлять данные в виде списков, словарей и простых типов, делая их легкими для чтения и редактирования.
13. JSON (JavaScript Object Notation)[11] - формат обмена данными, основанный на тексте и простом для чтения и записи для людей. Он широко используется в веб-разработке для передачи структурированных данных между клиентской и серверной сторонами. JSON позволяет представлять данные в виде пар ключ-значение, списков и вложенных структур, что делает его удобным для обработки и интерпретации программами.

Введение

Машинное обучение – это область искусственного интеллекта, которая позволяет компьютерным системам изучать и анализировать данные, автоматически улучшать свою производительность и делать прогнозы на основе полученной информации. Бизнесу машинное обучение предоставляет широкие возможности для оптимизации и улучшения различных процессов и задач, включая:

- **Анализ больших объемов данных:** МО позволяет анализировать большие объемы данных, которые могут содержать ценную информацию для принятия решений. Это может включать анализ данных о продажах и потребительском поведении клиентов, чтобы определить, какие товары и услуги наиболее популярны.
- **Автоматическая обработка текстовой информации:** МО может использоваться для автоматической обработки и анализа текстовых данных, таких как отзывы и комментарии пользователей о продуктах и услугах компании.
- **Создание и улучшение алгоритмов и моделей:** МО может быть использовано для создания и улучшения различных алгоритмов и моделей, таких как прогнозирование спроса на товары и услуги, оптимизация производства, управление рисками и т.д.
- **Автоматическая обработка и классификация изображений:** МО может использоваться для распознавания лиц клиентов, контроля качества продукции на производстве и т.д.

В целом, МО является мощным инструментом для бизнеса, который позволяет автоматизировать и улучшать множество различных задач и процессов, повышать эффективность операций и создавать новые продукты и услуги.

Рассмотрим сценарий, которые необходимо пройти бизнесу, что решить задачу с помощью машинного обучения:

- **Определение бизнес-задачи и постановка задачи для машинного обучения:** компания определяет конкретную бизнес-задачу, которую нужно решить с помощью машинного обучения. Для этого формулируется задача для анализа данных и выбирается подходящий алгоритм машинного обучения.
- **Сбор и предобработка данных:** для решения задачи машинного обучения необходимо собрать данные, которые будут использоваться для обучения модели. Это может быть как структурированные данные из базы данных компании, так и данные из внешних источников.

- Сбор данных: на этом этапе необходимо определить, какие данные необходимы для решения задачи. Это может потребовать усилий от бизнеса, чтобы определить, какие данные нужны, и как их собрать. Возможно, потребуются специалисты по сбору данных, чтобы помочь в этом процессе. Перед использованием эти данные должны быть предварительно обработаны и очищены от ошибок и выбросов
- Подготовка данных: после сбора данных их необходимо очистить и подготовить для анализа. Этот процесс может потребовать значительных усилий и времени, особенно если данные неструктурированные или содержат много шума. В этом случае могут потребоваться специалисты по обработке данных и статистическому анализу.
- Выбор и настройка модели машинного обучения: для решения задачи выбирается подходящая модель машинного обучения, которая будет обучена на собранных данных. Для настройки модели могут использоваться различные методы, например, кросс-валидация или подбор оптимальных параметров.
 - Это требует специалистов по машинному обучению и анализу данных, которые имеют опыт в выборе и настройке алгоритмов машинного обучения.
- Обучение модели: после выбора модели и предварительной настройки процесс обучения начинается.
- Оценка качества модели: после обучения модели оценивается ее качество с помощью различных метрик. Если качество модели не удовлетворительное, то процесс настройки и обучения модели повторяется.
 - Этот процесс может потребовать усилий от команды разработчиков и тестировщиков, чтобы провести тесты на разных наборах данных.
- Использование модели в бизнес-процессах: после получения удовлетворительных результатов модель может быть использована для решения бизнес-задачи. Например, модель может быть интегрирована в бизнес-приложения или использоваться для автоматического принятия решений.
 - На этом этапе требуются усилия от команды разработчиков и системных администраторов, чтобы обеспечить правильную работу модели и ее интеграцию с другими системами.
- Мониторинг и обновление модели: модель машинного обучения не является статичной и может требовать обновления при изменении условий бизнеса или при появлении новых данных. Поэтому необходимо постоянно мониторить ее работу и вносить изменения при необходимости.

- На данном этапе требуется команда мониторинга и обслуживания, которые будут отслеживать работу модели и вносить изменения при необходимости



Рис. 1: Сценарий внедрения машинного обучения в бизнес

Глава 1. Анализ существующих решений

1. Обзор существующих решений

Для упрощения использования машинного обучения в бизнесе можно рассмотреть следующие подходы:

1.1. Партнерство с консультантами по машинному обучению

Первый подход заключается в сотрудничестве с консультантами по машинному обучению. Такой подход может быть полезен для компаний, которые хотят получить экспертную помощь и рекомендации по использованию машинного обучения. Консультанты по машинному обучению имеют опыт работы с различными проектами и могут помочь бизнесу в выборе правильных инструментов, настройке и обучении моделей машинного обучения, а также внедрении их в рабочую среду.

Консультанты могут провести анализ бизнес-процессов и собрать необходимые данные для обучения моделей машинного обучения. Они также могут помочь определить, какие задачи можно автоматизировать с помощью машинного обучения и какие модели лучше всего подходят для решения конкретных задач.

Кроме того, консультанты могут обучать сотрудников компании, чтобы они могли использовать машинное обучение самостоятельно в будущем. Это может быть особенно полезно для компаний, которые хотят создать внутреннюю команду специалистов по машинному обучению и усилить свои внутренние компетенции в этой области.

Однако, стоимость услуг консультантов по машинному обучению может быть высокой и может быть неэффективной для небольших компаний с ограниченным бюджетом. Кроме того, сотрудничество с консультантами может потребовать большого количества времени и ресурсов для взаимодействия и внедрения решений в рабочую среду.

Для первого подхода, связанного с партнерством с консультантами по машинному обучению, можно привести несколько примеров компаний:

- Deloitte[7]: одна из крупнейших мировых компаний-консультантов по вопросам управления, финансов и технологий. Deloitte имеет подразделения, специализирующиеся на машинном обучении и искусственном интеллекте, которые могут помочь бизнесу в выборе правильных инструментов и решений, настройке и обучении моделей машинного обучения, а также внедрении их в рабочую среду.
- Accenture[2]: глобальная компания-консультант, специализирующаяся на услугах цифровой трансформации, технологиях и машинном обучении. Accenture предлагает услуги консультирования, разработки и внедрения решений в обла-

сти машинного обучения, а также помощь в формировании команды специалистов по машинному обучению внутри компании.

- PwC[14]: еще одна крупная мировая компания-консультант по вопросам управления, аудита и налогообложения. PwC имеет подразделения, специализирующиеся на машинном обучении и анализе данных, которые могут помочь бизнесу в выборе оптимальных решений, разработке и обучении моделей машинного обучения, а также внедрении их в рабочую среду.

1.2. Автоматизация процесса использования моделей

Второй подход, который может быть использован для упрощения использования машинного обучения в бизнесе - это автоматизация процесса использования моделей.

Разработка удобного пользовательского интерфейса и автоматической системы внедрения моделей в рабочую среду может значительно упростить процесс использования машинного обучения для бизнеса. Такая система может позволить сократить время, затрачиваемое на обучение и использование моделей, а также снизить количество ошибок, связанных с человеческим фактором.

Для автоматизации процесса использования моделей машинного обучения в бизнесе используются различные инструменты и технологии. Например, могут быть использованы платформы для автоматизации процессов машинного обучения, которые позволяют создавать, обучать и развертывать модели машинного обучения в автоматическом режиме.

Также могут быть разработаны специальные программы или библиотеки для использования моделей машинного обучения в рамках конкретных бизнес-процессов. Например, для обработки и анализа больших объемов данных в режиме реального времени могут быть использованы специальные программные платформы, которые позволяют интегрировать модели машинного обучения в рабочие процессы бизнеса.

Важным аспектом автоматизации процесса использования моделей машинного обучения является разработка удобного пользовательского интерфейса. Такой интерфейс должен быть интуитивно понятным и легко настраиваемым, чтобы позволить пользователям без специальных знаний по машинному обучению работать с моделями машинного обучения и использовать их результаты в своих бизнес-процессах.

Таким образом, автоматизация процесса использования моделей машинного обучения может значительно упростить использование машинного обучения в бизнесе и повысить его эффективность. Однако для этого необходимо правильно выбрать инструменты и технологии, а также разработать удобный пользовательский интерфейс.

- Amazon SageMaker - это полноценный сервис для машинного обучения в облаке, который предоставляет возможность создания, обучения и развертывания моделей машинного обучения.

- TensorFlow[24] - это библиотека машинного обучения с открытым исходным кодом, которая предоставляет широкий спектр инструментов для создания и обучения моделей машинного обучения.
- PyTorch[15] - это еще одна библиотека машинного обучения с открытым исходным кодом, которая предоставляет разработчикам удобный интерфейс для создания и обучения моделей машинного обучения.

1.3. Использование готовых решений

Третий подход заключается в использовании готовых решений и сервисов на рынке, которые позволяют использовать машинное обучение без необходимости самостоятельно настраивать и обучать модель.

Существует множество компаний, предоставляющих такие сервисы, такие как Amazon Web Services (AWS)[1], Google Cloud Platform (GCP)[8], Microsoft Azure[5] и другие. Они предлагают различные инструменты и решения, например, AutoML[3], которые позволяют создавать и настраивать модели машинного обучения с помощью графических интерфейсов и не программирования. Это позволяет бизнесу начать использовать машинное обучение без необходимости иметь внутреннюю команду специалистов по машинному обучению.

Кроме того, на рынке также существуют готовые решения и библиотеки, которые предоставляют готовые модели, например, TensorFlow, PyTorch, Scikit-learn[21] и другие. Эти решения обычно имеют открытый исходный код и используются как базовые инструменты для создания более сложных систем машинного обучения.

Преимуществом использования готовых решений является быстрый и простой старт использования машинного обучения, а также возможность экономии времени и денег на настройке и обучении модели. Кроме того, такие решения могут иметь готовые интеграции с другими системами и платформами, что позволяет интегрировать машинное обучение в рабочий процесс бизнеса.

Однако, недостатком таких решений может быть ограниченная настраиваемость и определенные ограничения в использовании моделей. Кроме того, такие решения могут быть дорогими, особенно для малых и средних бизнесов.

Для третьего подхода, основанного на использовании готовых решений, можно привести следующие примеры компаний:

- Amazon Web Services (AWS) предоставляет готовые сервисы машинного обучения, такие как Amazon SageMaker, Amazon Rekognition, Amazon Comprehend и другие. Компании могут использовать эти сервисы для анализа данных, распознавания изображений, обработки естественного языка и многого другого.
- Google Cloud Platform (GCP) также предоставляет готовые сервисы машинного

обучения, такие как Google Cloud Vision, Google Cloud Speech-to-Text, Google Cloud Natural Language и другие. Компании могут использовать эти сервисы для распознавания изображений и речи, анализа текста и многого другого.

- IBM Watson[25] предоставляет готовые сервисы машинного обучения и искусственного интеллекта, такие как IBM Watson Assistant, IBM Watson Discovery, IBM Watson Studio и другие. Компании могут использовать эти сервисы для создания виртуальных ассистентов, анализа данных, обучения моделей машинного обучения и многого другого.

Kaggle[12] - это онлайн-платформа для машинного обучения, которая предоставляет доступ к наборам данных, облачным вычислениям, инструментам разработки и соревнованиям по машинному обучению. Kaggle можно использовать как для третьего подхода, выбирая готовые наборы данных и модели для решения задач, так и для второго подхода, создавая собственные модели и решения, используя инструменты разработки и облачные вычисления на платформе.

H2O.ai[9] - это платформа для машинного обучения с открытым исходным кодом, которая предоставляет инструменты для создания моделей машинного обучения, обучения и развертывания моделей, анализа данных и визуализации. H2O.ai можно использовать как для третьего подхода, выбирая готовые модели и алгоритмы для решения задач, так и для второго подхода, разрабатывая собственные модели и решения, используя доступные инструменты и библиотеки.

2. Сравнение Robbi с аналогами

Сервис Robbi[17] от социальной сети Одноклассники является платформой для разработки и использования машинного обучения в типовых задачах. Сервис предоставляет следующие наработки:

- Предварительно обученные модели для типовых задач: сервис Robbi имеет набор предварительно обученных моделей машинного обучения, которые могут быть использованы для решения типовых задач, таких как классификация текстов, анализ тональности, распознавание объектов на изображениях и т.д. Эти модели уже обучены на больших объемах данных и могут быть применены для решения задач пользователей без необходимости проведения собственного обучения моделей.
- Инфраструктура для хранения и обработки данных: сервис Robbi предоставляет инфраструктуру для хранения и обработки данных, которые могут быть использованы для обучения моделей. Сервис позволяет пользователям загружать и хранить данные в больших объемах, а также обрабатывать эти данные для подготовки их к обучению моделей.

- Сервис для публикаций модели в виде API: сервис Robbii позволяет пользователям экспортировать обученные модели в формате API, что позволяет другим приложениям и сервисам использовать эти модели для решения различных задач. Это упрощает интеграцию машинного обучения в другие приложения и позволяет использовать обученные модели в более широком диапазоне задач.
- Краудсорсинговое приложение: сервис Robbii также включает в себя краудсорсинговое приложение, которое позволяет пользователям создавать и размещать задачи для обучения моделей. Пользователи могут загружать данные, создавать задачи для разметки данных и запускать обучение моделей на этих данных. Краудсорсинговое приложение позволяет пользователю масштабировать процесс обучения моделей, получить больше данных для обучения и ускорить процесс обучения.

2.1. Удобство использования и эффективность

Готовые решения для машинного обучения, такие как Amazon SageMaker, Microsoft Azure Machine Learning и Google Cloud AutoML, обладают рядом преимуществ, которые обеспечивают удобство использования.

Для начала, эти платформы предоставляют доступ к большим наборам данных, как внутренним, так и импортированным из внешних источников. Для обработки данных предоставляются инструменты нормализации, масштабирования и извлечения признаков, что облегчает и ускоряет работу с данными.

Кроме того, в каталогах этих платформ доступны предварительно обученные модели, которые можно использовать в своих проектах, что позволяет сократить время разработки и улучшить качество модели. Для более точной настройки моделей предоставляются возможности настройки гиперпараметров.

После создания модели, ее можно легко развернуть с помощью веб-сервиса или API, предоставленных платформой. Также обеспечивается возможность оценки производительности модели, что позволяет производить дальнейшую настройку и улучшение модели.

Сервис Robbii является более ограниченным по сравнению с Amazon SageMaker, Microsoft Azure Machine Learning и Google Cloud AutoML. Сервис Robbii предоставляет набор предварительно обученных моделей и инфраструктуру для хранения и обработки данных, которые могут быть использованы для обучения моделей, а также сервис для экспорта обученных моделей в виде API. В то же время, Amazon SageMaker[19], Microsoft Azure Machine Learning и Google Cloud AutoML предоставляют больше возможностей для создания и настройки собственных моделей машинного обучения.

Например, Amazon SageMaker предоставляет полноценную инфраструктуру для создания и настройки моделей машинного обучения, включая инструменты для визу-

альной настройки гиперпараметров, оптимизации моделей, масштабирования моделей и т.д. Microsoft Azure Machine Learning также предоставляет мощные инструменты для создания и настройки моделей, включая возможность использования визуального интерфейса для настройки моделей, использование языка Python для создания моделей и т.д. Google Cloud AutoML предоставляет инструменты для создания и настройки моделей машинного обучения, которые могут быть использованы без необходимости знания языка программирования.

Таким образом, сервис Robbi предоставляет более узкую специализацию, чем Amazon SageMaker, Microsoft Azure Machine Learning и Google Cloud AutoML, и может быть полезен для решения типовых задач машинного обучения без необходимости создания собственной модели. Однако, если пользователь требуется создание собственной модели, то Amazon SageMaker, Microsoft Azure Machine Learning и Google Cloud AutoML предоставляют больше возможностей для настройки и создания собственных моделей.

Однако, для создания собственной модели машинного обучения, требуется программирование и определенные знания в области алгоритмов машинного обучения и глубокого обучения. Эти сервисы предоставляют возможность разработки собственных моделей, но для этого требуется написание кода конфигурации модели. Это может быть сложным процессом для тех, кто не имеет достаточных знаний в области программирования и машинного обучения. Однако, при наличии соответствующих навыков, использование этих сервисов может значительно ускорить и упростить процесс создания и развертывания моделей машинного обучения.

Пример части конфигурации для запуска обучения модели в Google Cloud AI:

```
traininginput :
  scaleTier : CUSTOM
  masterType : complex_model_m
  workerType : complex_model_m
  parameterServerType : large_model
  workerCount : 9
  parameterServerCount : 3
  hyperparameters :
    enableTrialEarlyStopping : TRUE
    maxTrials : 30
    maxParallelTrials : 1
    resumePreviousJobId : [PREVIOUS_JOB_IDENTIFIER]
```

Listing 1: Конфигурация обучения в Google Cloud AI

Хотим создать сервис, который поможет пользователям быстро и эффективно решать различные задачи машинного обучения без необходимости иметь большой опыт

в программировании и настройке моделей.

Целью дипломной работы является разработка веб-сервиса, основанного на платформе машинного обучения Robbi, с пользовательским интерфейсом и взаимодействующей с ним частью бэкенда без необходимости программирования.

Для достижения поставленной цели были сформулированы следующие задачи. Во-первых, необходимо разработать пользовательский интерфейс в соответствии с основными сценариями сервиса. Во-вторых, требуется разработать взаимодействующую с пользовательским интерфейсом часть бэкенда, которая будет обеспечивать прием и обработку запросов от пользователей. В-третьих, необходимо реализовать функциональность создания и настройки моделей машинного обучения без программирования в рамках веб-сервиса.

Для реализации первой задачи необходимо создать удобный и интуитивно понятный интерфейс, который будет позволять пользователям легко освоить функционал сервиса. Для этого можно использовать современные технологии веб-разработки и дизайна интерфейсов.

Для решения второй задачи необходимо создать бэкенд, который будет обрабатывать запросы от пользователей и передавать их в модели машинного обучения.

Для реализации третьей задачи необходимо создать функционал, который позволит пользователям создавать и настраивать модели машинного обучения без необходимости программирования. Необходимо разработать специальный интерфейс для настройки параметров модели.

На данный момент в сервисе Robbi создание модели происходит с помощью написания конфигураций на языке YAML такого вида:

```
transforms :
  train :
    { type: Resize , height: *target_shape ,
      width: *target_shape , always_apply: !!bool True }
    { type: HorizontalFlip , p: 0.5 }
    { type: RandomRotate90 , p: 0.25 }
    { type: RandomBrightnessContrast , p: 0.2 }
    { type: ChannelShuffle , p: 0.2 }
    { type: OneOf, p: 0.2, transforms: [
      { type: IAAGaussianNoise , p: 0.5 },
      { type: GaussNoise , p: 0.5 }
    ]}
    { type: OneOf, p: 0.2, transforms: [
      { type: MotionBlur , p: 0.5 , blur_limit: 3 },
      { type: MedianBlur , p: 0.5 , blur_limit: 3 }
    ]}
  ]}
```



```

type: Normalize
type: ToTensorV2
val:
  { type: Resize , height: *target_shape ,
    width: *target_shape , always_apply: !!bool True }
type: Normalize
type: ToTensorV2

```

Listing 2: Конфигурация обучения в Robbi

3. Выводы по главе

Реализация поставленных задач позволит создать удобный и эффективный веб-сервис, основанный на платформе машинного обучения Robbi, который будет обеспечивать создание и настройку моделей машинного обучения без необходимости программирования.

Решения	Партнерство	Собственная автоматизация процесса	Готовые решения
Цена	Высокая	Средняя	Низкая
Время реализации	Длительный срок	Средний срок	Краткий срок
Качество решения	Высокое	Среднее	Низкое
Гибкость	Высокая	Средняя	Низкая
Сложность внедрения	Высокая	Средняя	Низкая
Необходимость наличия внутренней команды специалистов	Низкая	Высокая	Низкая

Таблица 1: Сравнение аналогов упрощения использования МО

Глава 2. Функции и особенности Robbi

1. Список основных функций веб-сервиса

1. Регистрация и авторизация пользователей;
2. Просмотр списка доступных пользователю датасетов;
3. Просмотр списка доступных пользователю архитектур;
4. Просмотр списка доступных моделей и управление их состояниями обучения;
5. Создание и загрузка собственного датасета;
6. Создание архитектуры модели и настройка параметров;

Диаграмма прецедентов на данном рисунке:

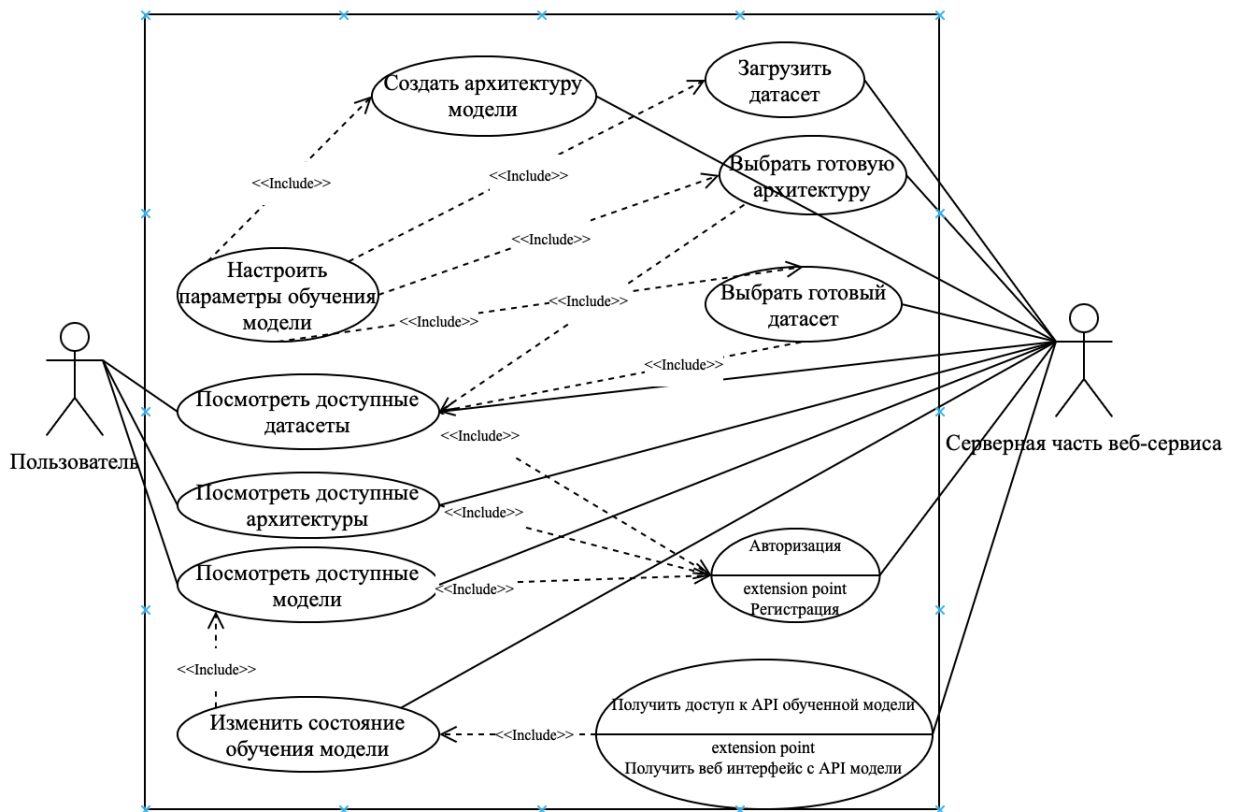


Рис. 2: Диаграмма прецедентов

2. Главный сценарий использования веб-сервиса

Основной пользовательский сценарий для Robbi заключается в том, чтобы предоставить пользователю возможность создать, обучить и использовать модели машин-

ного обучения без необходимости знания программирования и использования сложных инструментов.

1. Создание проекта: пользователь выбирает тип задачи, с которой он хочет работать, например, классификация данных, кластеризация или прогнозирование. Затем он загружает свои данные и выбирает методы обработки и предварительной обработки данных.
2. Обучение модели: пользователь выбирает тип модели машинного обучения и параметры обучения, а также определяет метрики оценки качества модели. Затем он запускает обучение модели на своих данных и оценивает качество работы модели.
3. Использование модели: пользователь использует обученную модель для решения своих задач. Например, он может использовать ее для классификации новых данных или для прогнозирования будущих результатов.
4. Мониторинг и обновление состояния модели: пользователь может следить за работой своей модели и регулярно обновлять ее, если это необходимо.
5. Совместная работа: пользователь может работать с другими пользователями и делиться своими данными и моделями, чтобы получить совместную пользу и результат.

3. Настройка параметров модели и обучения

Пользователи имеют несколько вариантов для создания обучения:

- Во-первых, пользователи могут выбрать готовую архитектуру из предложенного набора. Это позволяет быстро начать процесс обучения, используя заранее подготовленные модели, которые были специально разработаны для различных типов задач машинного обучения. Этот подход особенно полезен для пользователей, которые не имеют опыта в настройке моделей и предпочитают использовать проверенные решения.
- Во-вторых, пользователи могут самостоятельно настраивать параметры модели и обучения. Наш сервис предоставляет интуитивно понятный интерфейс, позволяющий пользователям экспериментировать с различными параметрами, чтобы достичь наилучших результатов обучения. Это дает пользователям полный контроль над процессом обучения и возможность адаптировать его под свои потребности и требования.

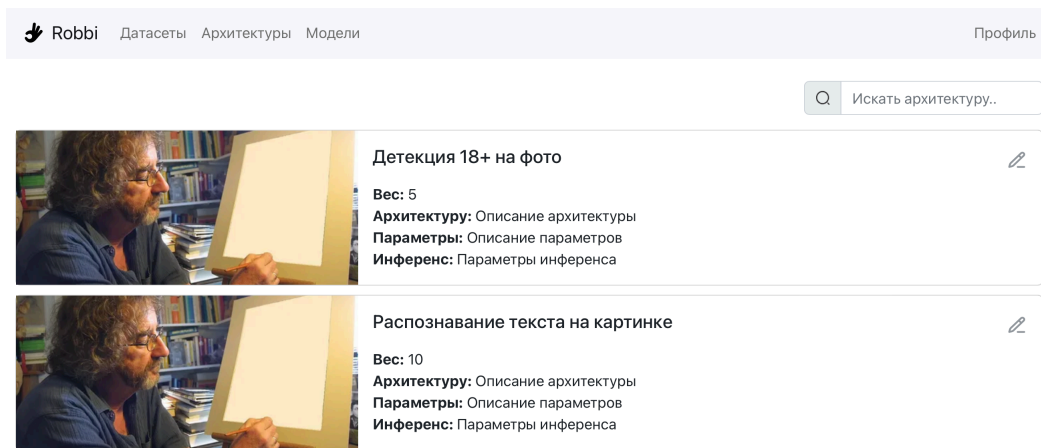


Рис. 3: Выбор готовой архитектуры

В этом разделе мы рассмотрим функции, разработанные для обеспечения простоты и гибкости при настройке параметров модели, что позволит пользователям достичь оптимальных результатов обучения без технических сложностей.

Описание процесса создания конфигурации для обучения модели машинного обучения из параметров, полученных в виде JSON-объекта с бэкенда.

1. Получение JSON-объекта с параметрами

- Первый этап заключается в получении JSON-объекта с параметрами от бэкенда. Эти параметры должны быть представлены в определенном формате, чтобы их можно было легко обрабатывать в дальнейшем. Пример того как некоторые операции приходят с бэкенда и какие параметры у них есть:

```

"transforms": {
  "Resize": {
    "height": "int",
    "width": "int",
    "interpolation": "OpenCV flag",
    "always_apply": "bool",
    "p": "float"
  },
  "HorizontalFlip": {
    "p": "float"
  }
}

```

Listing 3: JSON с операциями

2. Создание структуры React[16] из JSON-объекта

- Следующий этап состоит в создании структуры React из полученного JSON-объекта. Это может быть достигнуто с помощью библиотеки React, которая позволяет создавать веб-интерфейсы на основе компонентов.

Пример того как выглядит структура для дальнейшего построения React-объектов:

```
value: 'Resize ',
params: [
  {
    name: 'height ',
    value: 'int '
  },
  {
    name: 'width ',
    value: 'int '
  },
  {
    name: 'interpolation ',
    value: 'OpenCV flag '
  },
  {
    name: 'always_apply ',
    value: 'bool '
  },
  {
    name: 'p',
    value: 'float '
  }
]
```

Listing 4: Структура в React

3. Создание Node-компонентов интерфейса

- Когда структура React создана, мы можем использовать ее для создания Node-компонентов интерфейса. Эти компоненты представляют отдельные элементы веб-интерфейса, такие как текстовые поля, кнопки и т.д. Каждый Node-компонент должен быть настроен на основе соответствующих параметров, полученных из JSON-объекта.

4. Создание дерева параметров

- Создание Node-компонентов позволяет нам получить отдельные параметры для каждого элемента интерфейса. Затем мы можем создать дерево параметров, которое представляет собой иерархическую структуру всех параметров веб-интерфейса.

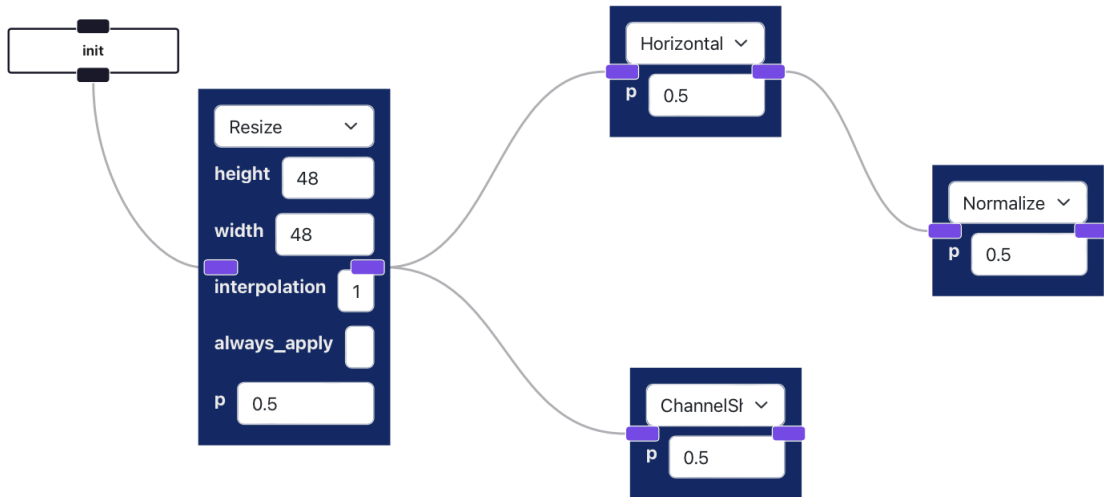


Рис. 4: Создание дерева операций для обучения

5. Формирование Config для обучения модели

- Когда мы имеем дерево параметров, мы можем создать JSON-структуру, которая содержит все параметры, необходимые для обучения модели машинного обучения. Эта структура называется Config и должна быть передана в платформу машинного обучения для запуска модели.

Таким образом, процесс создания конфигурации для обучения модели машинного обучения начинается с получения JSON-объекта с параметрами, затем создается структура React, из которой формируются Node-компоненты интерфейса. Каждый Node-компонент содержит параметры, которые могут быть использованы для создания дерева параметров. Дерево параметров, в свою очередь, формирует Config для обучения модели машинного обучения.

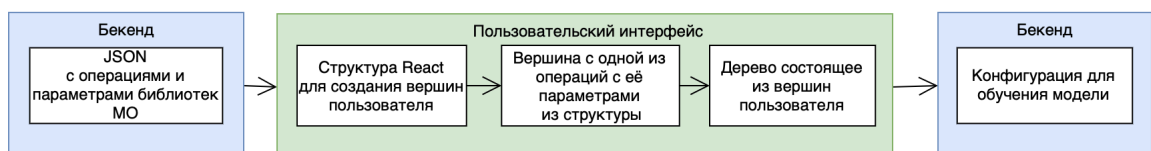


Рис. 5: Схема формирования конфигурации для обучения модели

4. Управление состояниями обучения

Эта функциональность отвечает за отслеживание, визуализацию и взаимодействие с различными этапами и состояниями обучения моделей, обеспечивая эффективное взаимодействие пользователя с платформой.

1. Отслеживание состояний обучения:

Здесь можно выделить два типа состояний обучения:

- Первый тип состояний связан с созданием процесса обучения. Здесь включены этапы, такие как загрузка или выбор данных для обучения, создание конфигурации или выбор готовой архитектуры модели из предоставленного сервиса, а также последующий запуск обучения. Это помогает пользователям иметь ясное представление о ходе подготовки процесса обучения и вовлекаться в принятие решений на каждом этапе.
- Второй тип состояний связан с уже созданными моделями, которые могут находиться в различных состояниях. Это может включать состояния, такие как обучение в процессе, ожидание, приостановка, завершение обучения или ошибки обучения. Пользовательский интерфейс предоставляет возможность отслеживать текущее состояние каждой модели, а также выводить соответствующую информацию и обновлять ее по мере изменения состояния.

2. Визуализация состояний обучения:

- Визуализации состояний обучения заключается в отображении различных статусов обучения на основной странице со всеми доступными моделями для пользователя.
- На этой странице каждая модель представлена с помощью информационного блока, который содержит основные сведения о модели, включая ее текущий статус обучения. Это может быть выражено в виде бейджа, расположенного на карточке модели. Например, бейдж "Обучается" может указывать на то, что модель находится в процессе обучения, а бейдж "Обучена" может указывать на успешное завершение обучения.

3. Взаимодействие с состояниями обучения:

- Пользовательский интерфейс должен предоставлять возможности взаимодействия с состояниями обучения. Например, пользователь может приостановить, возобновить или отменить процесс обучения.
- Кроме визуализации статуса обучения в пользовательском интерфейсе предоставляется уникальный набор кнопок или элементов управления, связанных

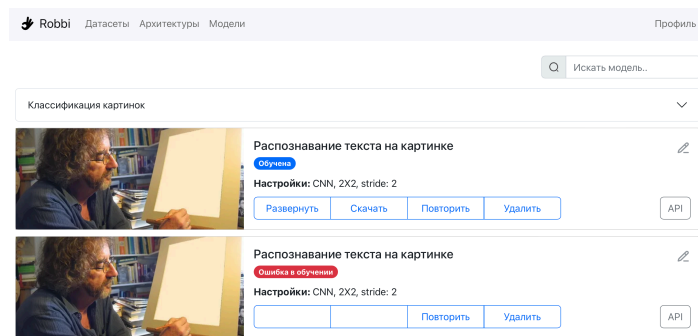


Рис. 6: Просмотр статусов моделей

с каждым статусом. Эти кнопки позволяют пользователю взаимодействовать с состоянием обучения и влиять на него. Например, для завершенной модели пользователю могут быть предложены кнопки "Развернуть" или "Повторить".

4. Обработка ошибок:

- При управлении состояниями обучения важно предусмотреть обработку возможных ошибок. Например, если загрузка данных не удалась или обучение завершилось с ошибкой, пользователю должно быть предоставляется сообщение об ошибке с соответствующими деталями и возможностью повторной попытки запуска или изменения конфигурации модели.

5. Обновление интерфейса:

- При изменении состояний обучения, пользовательский интерфейс должен быть обновлен, чтобы отразить текущее состояние. Например, элементы интерфейса могут становиться неактивными или меняться доступные пользователю действия в зависимости от статуса модели.

5. Выводы по главе

Веб-сервис предлагает широкий набор функций, позволяющих пользователям участвовать в процессе машинного обучения без необходимости программирования. От загрузки данных до настройки параметров модели и выполнения обучения, платформа обеспечивает удобство и гибкость в использовании.

Особый акцент стоит на настройке параметров обучения без программирования и управлении состояниями обучения, что позволяет пользователям контролировать каждый этап процесса обучения и принимать решения на основе доступной информации. Отслеживание состояний обучения, визуализация и возможность влиять на состояние обучения через набор кнопок и элементов управления повышают эффективность взаимодействия с платформой.

Глава 3. Технологии, архитектура и реализация клиентской части веб-сервиса

1. Языки программирования

Для реализации пользовательского интерфейса веб-сервиса были выбраны следующие технологии:

1. HTML/CSS: HTML (HyperText Markup Language) используется для создания структуры веб-страниц, а CSS (Cascading Style Sheets) - для оформления и стилизации элементов на странице. HTML и CSS позволят создать пользовательский интерфейс веб-сервиса и задать его внешний вид.
2. TypeScript[23]: является строго типизированным языком программирования, являющимся надмножеством JavaScript. Он добавляет статическую типизацию и другие возможности, улучшающие разработку больших проектов. TypeScript поможет создать более надежный и поддерживаемый код для клиентской части веб-сервиса.
3. React: это JavaScript-библиотека для создания пользовательских интерфейсов. Она позволяет разработчикам строить компоненты с переиспользуемой логикой и легко управлять состоянием приложения. React является популярным выбором для создания динамических веб-приложений.

Для использования React были выбраны следующие вспомогательные инструменты:

1. Bootstrap[6]: популярный фреймворк для разработки пользовательского интерфейса. Он предоставляет набор готовых компонентов, стилей и сетки, которые помогают быстро создавать привлекательные и отзывчивые веб-страницы. В проекте Bootstrap используется для стилизации и реализации готовых компонентов пользовательского интерфейса, упрощая процесс разработки и обеспечивая единообразный внешний вид элементов.
2. react-flow-renderer[34]: библиотека, которая предоставляет возможность визуализации и управления графическими элементами и диаграммами в React-приложениях. В проекте она используется для реализации и отображения графических компонентов, связанных с созданием процесса машинного обучения.
3. React Router[18]: библиотека для управления маршрутизацией (навигацией) в React-приложениях. Она позволяет создавать динамические маршруты и связывать их с соответствующими компонентами. В проекте React Router используется для определения и управления различными страницами и переходами между ними, обеспечивая более удобную навигацию по веб-сервису.

4. Axios[4]: библиотека для выполнения HTTP-запросов. Она предоставляет простой и удобный интерфейс для отправки запросов на сервер и получения ответов. В проекте Axios используется для взаимодействия с серверной частью веб-сервиса Robbi, отправки и получения данных для выполнения операций машинного обучения без программирования.

Для реализации бэкенда веб-сервиса, изначально написанного на Java, был выбран фреймворк Spring[22]. Это мощный и широко используемый фреймворк для разработки Java-приложений. Он предоставляет набор инструментов, библиотек и функциональности, которые значительно упрощают и ускоряют процесс разработки, а также способствуют созданию надежного и масштабируемого бэкенда.

2. Архитектура веб-сервиса

Архитектура веб-сервиса Robbi организована на основе клиент-серверной модели. В данном случае, компоненты пользовательского интерфейса, обработки пользовательских запросов и инфраструктура, могут быть разделены следующим образом:

1. Компонент пользовательского интерфейса:

- Фронтенд: Отвечает за отображение пользовательского интерфейса и взаимодействие с пользователями. Фронтенд веб-сервиса Robbi будет отображать интерфейс для пользователей, позволяющий им взаимодействовать с функциональностью машинного обучения без программирования, предоставляемой сервисом.

2. Компонент обработки пользовательских запросов:

- Бэкенд: Этот компонент является основным и отвечает за обработку пользовательских запросов, бизнес-логику и взаимодействие с клиентом “Одноклассников”. Он принимает и обрабатывает запросы от фронтенда, выполняет валидацию и получает метаданные из загруженных пользователем данных.

3. Клиент “Одноклассников”:

- Этот компонент представляет собой внешний клиент или интеграцию с сервисом, который уже обладает функциональностью управления обучением и инфраструктурой. Взаимодействие с этим компонентом происходит через API, позволяя управлять процессом обучения и использовать уже существующую инфраструктуру, включая базы данных.

В целом, данная архитектура разделяет компоненты пользовательского интерфейса и обработки запросов, чтобы обеспечить гибкость и разделение ответственности. Клиент “Одноклассников” предоставляет возможность управления обучением и инфраструктурой внешней системы, интегрируясь с веб-сервисом Robbi через уже готовое API.

3. Взаимодействие пользовательского интерфейса и бэкенда

3.1. Запросы от фронтенда к бэкенду

Реализации взаимодействия пользовательского интерфейса и бэкенда с использованием React и Axios включают следующие этапы:

- **Отправка запросов:** В React используется функциональность Axios для отправки HTTP-запросов с клиента к бэкенду. Например, для выполнения GET-запроса используется `axios.get(url)`, а для POST-запроса - `axios.post(url, data)`.
- **Обработка ответов:** Axios позволяет легко обрабатывать ответы от бэкенда. Для обработки полученных данных используются `promise` (позволяет асинхронно обрабатывать данные). Например, `axios.get(url).then(response => ...)` позволяет выполнить действия с полученными данными после успешного выполнения запроса.
- **Обработка ошибок:** Важно обрабатывать возможные ошибки при взаимодействии с бэкендом. Axios предоставляет механизм для обработки ошибок, который позволяет легко обнаруживать и обрабатывать различные сценарии ошибок, такие как ошибки сети, ошибки сервера и другие.
- **Обновление состояния интерфейса:** После получения ответа от бэкенда, используются компоненты React для обновления пользовательского интерфейса с учетом полученных данных. React предоставляет возможности для управления состоянием компонентов и перерисовки интерфейса при необходимости.

Итак, взаимодействие пользовательского интерфейса, разработанного с использованием React, и бэкенда осуществляется с помощью Axios, который обеспечивает отправку запросов, обработку ответов и обработку ошибок. Эти технологии позволяют создать эффективный и отзывчивый пользовательский интерфейс, который взаимодействует с бэкендом для получения и обработки данных.

Для реализации взаимодействия с фронтендом и передачи данных между фронтендом и бэкендом были разработаны API-интерфейсы. API-интерфейсы представляют собой определенный набор точек входа, по которым фронтенд может отправлять запросы к бэкенду и получать соответствующие ответы.

Описание	Тип	URL
Доступные модели	GET	/models
Доступ к API	GET	/models/id/api
Создать конфигурацию для обучения	POST	/models
Изменить состояние модели	POST	/models/id
Доступные архитектуры	GET	/architectures
Выбрать готовую архитектуру	GET	/architectures/id
Выбрать датасет	GET	/datasets/id
Загрузить датасет	POST	/datasets
Доступные датасеты	GET	/datasets

Таблица 2: Методы серверной части для обработки запросов клиентской части

3.2. Обновление состояний в пользовательском интерфейсе через бекенд

Использование технологии Server-Sent Events (SSE) для обновления состояния обучения модели представляет собой эффективный механизм обмена информацией в режиме реального времени между клиентской и серверной сторонами веб-приложения. SSE позволяет серверу инициировать отправку данных клиенту без необходимости повторных запросов от клиента. Данные передаются по протоколу HTTP через долгоживущее соединение, которое обеспечивает непрерывный поток данных.

В контексте веб-приложений, реализация SSE между React и Java Spring включает несколько важных шагов:

1. На стороне Java Spring:

- Реализация SSE начинается с создания специализированного SSE-контроллера на серверной стороне с использованием Java Spring. Для этого был создан класс-контроллер, который обрабатывает запросы SSE от клиентов и отправляет обновления состояния обучения модели.
- В методе SSE-контроллера инициализируется объект `SseEmitter`, который представляет долгоживущее соединение с клиентом. Затем сервер определяет логику обновления состояния обучения и периодически отправляет данные клиенту через метод `sendEvent()`.
- При получении обновлений состояния обучения модели, сервер вызывает соответствующий метод `send()` или `sendEvent()` у объекта `SseEmitter` для отправки данных клиенту.

2. На стороне React:

- В React-приложении создан компонент, который будет устанавливать SSE-соединение с сервером и обрабатывать полученные обновления состояния обучения модели.

- В методе `componentDidMount()` компонента используется объект `EventSource` для установки SSE-соединения с сервером Java Spring, указывая URL SSE-контроллера.
- Установлен обработчик событий `onmessage` объекта `EventSource`, который будет вызываться при получении обновлений состояния обучения от сервера. Внутри обработчика производится обновление состояния компонента и соответствующее обновление пользовательского интерфейса.
- Предусмотрено корректное закрытие SSE-соединения в методе `componentWillUnmount()`, где вызывается метод `close()` объекта `EventSource`.

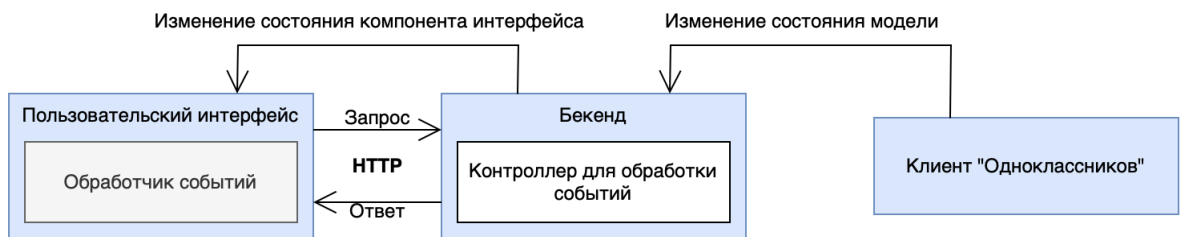


Рис. 7: Реализация SSE

4. Описание пользовательского интерфейса

Классовые компоненты являются классами JavaScript, которые наследуют функциональности от базового класса `React.Component`. Они содержат методы жизненного цикла, такие как `render()`, `componentDidMount()`, `componentDidUpdate()` и другие. Классовые компоненты могут иметь состояние (`state`) и обрабатывать события.

Компоненты:

- Переиспользуемые компоненты:

1. Карточка для отображения различных объектов, таких как описание моделей, архитектур и датасетов

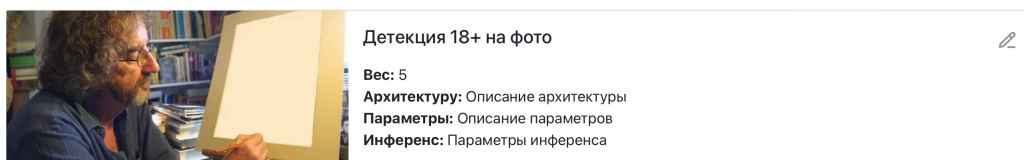


Рис. 8: Карточка готовой архитектуры

2. Поля для поиска на каждой из основных страниц пользовательского сценария, чекбоксы для выбора архитектур и датасетов

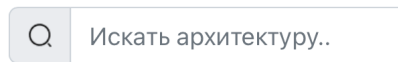


Рис. 9: Компонент для поиска по архитектурам

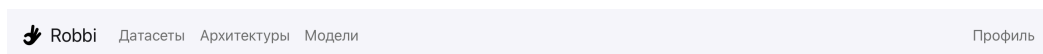


Рис. 10: Навигационная панель

3. Главная панель навигации для перемещения по основным страницам
- Непереиспользуемые компоненты:
 1. **ArchitectureView** (Отображение архитектур): Этот компонент отвечает за отображение списка доступных архитектур моделей. Он показывает информацию о каждой архитектуре, такую как название, описание и изображение.
 2. **DatasetLoadView** (Загрузка датасета): Этот компонент предоставляет интерфейс для загрузки датасета в приложение. Он содержит элементы, такие как кнопка загрузки, поле для выбора файла и прогресс-индикатор загрузки.
 3. **DatasetNavbar** (Навигационная панель датасета): Этот компонент предоставляет навигационную панель для перехода к различным разделам, связанным с датасетом. Например, он содержит ссылки на загрузку датасета, просмотр доступных датасетов и другие связанные функции.
 4. **DatasetView** (Отображение датасетов): Этот компонент отображает список доступных датасетов в приложении. Он показывает информацию о каждом датасете, такую как название, размер, описание и примеры данных.
 5. **UploadForm** (Форма загрузки файлов): Этот компонент отвечает за интерфейс загрузки файлов в веб-сервис. Он содержит элементы, такие как поле для выбора файла, кнопка загрузки и индикатор прогресса загрузки.
 6. **ModelControlButtons** (Кнопки управления моделями): Этот компонент отображает различные кнопки управления моделями в зависимости от их состояния. Например, он содержит кнопки для обучения модели, запуска/остановки процесса обучения, сохранения модели и другие связанные функции.
 7. **ModelView** (Отображение моделей): Этот компонент отображает список всех доступных моделей в приложении. Он показывает информацию о каждой модели, такую как название, описание, точность и другие метрики.
 8. **ParametersFlow** (Компонент для посторонних конфигураций): Этот компонент отвечает за отображение и управление посторонними конфигурациями

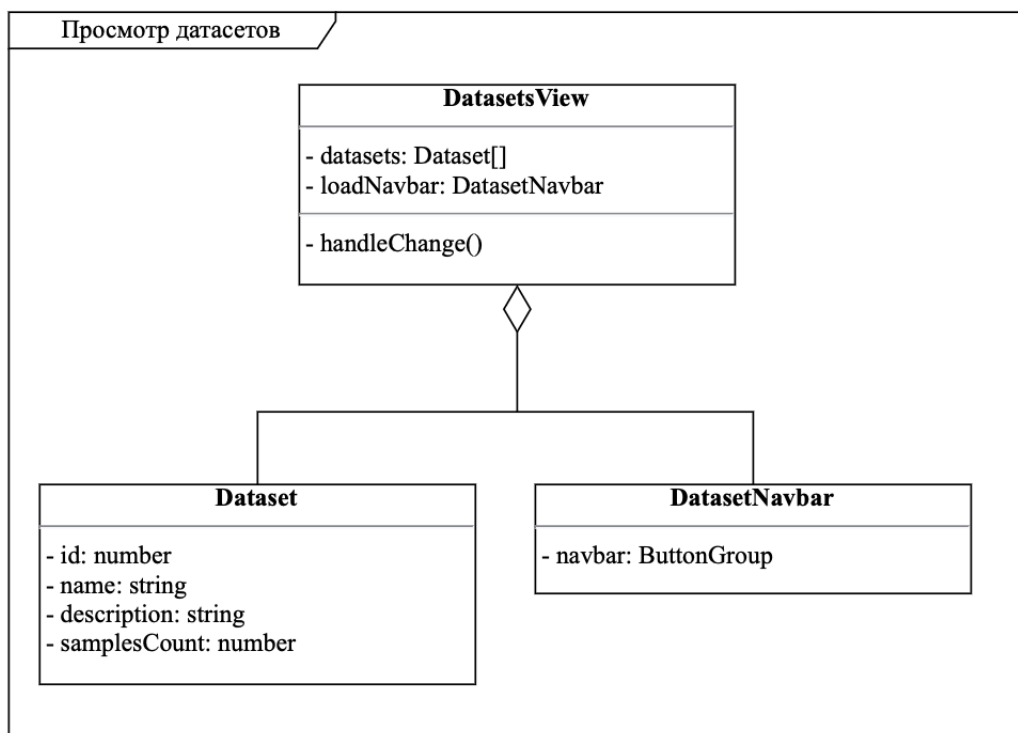


Рис. 11: UML классов: датасеты

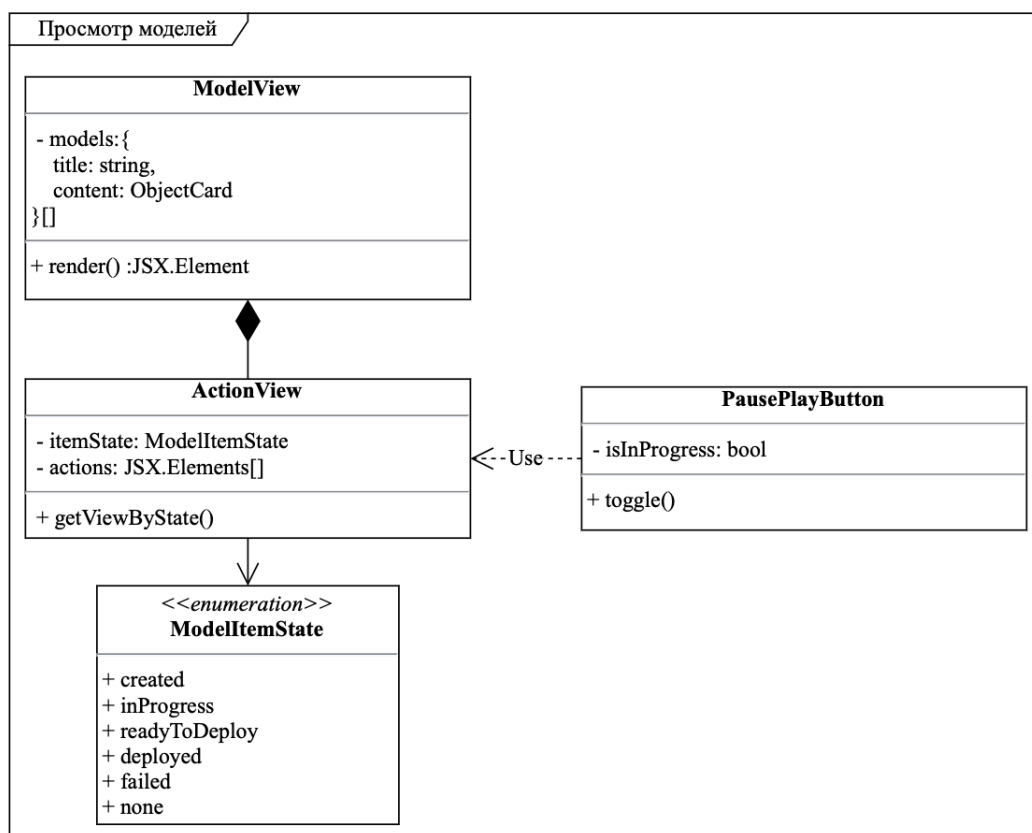


Рис. 12: UML классов: модели

для обучения новой модели. Он содержит поля для ввода параметров, выбор опций и другие элементы, связанные с настройками обучения.

9. **SelectNode** (Выбор операции и настройка параметров): Этот компонент используется для отображения интерактивного узла, позволяющего выбрать операцию (например, обучение модели) и настроить ее параметры. Он содержит выпадающие списки, ползунки, поля для ввода и другие элементы для управления параметрами операции.

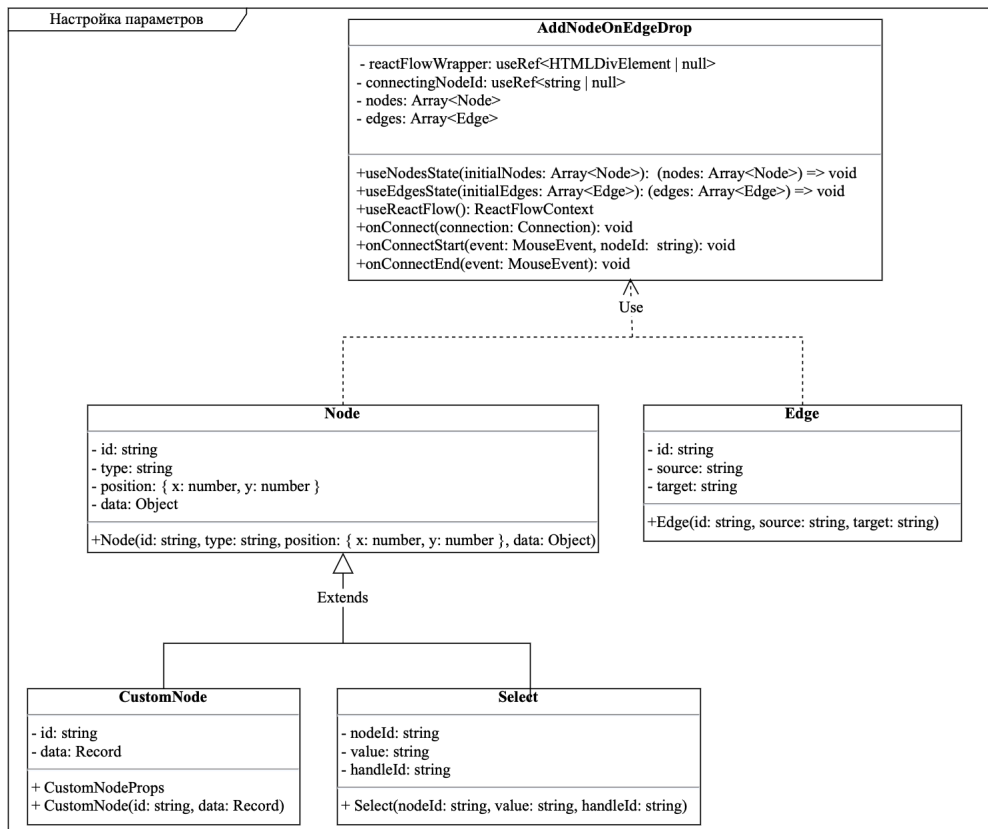


Рис. 13: UML классов: настройка параметров модели

Каждый из этих компонентов выполняет определенную функцию в пользовательском интерфейсе и используется в сочетании с другими компонентами для создания более сложных и функциональных интерфейсов.

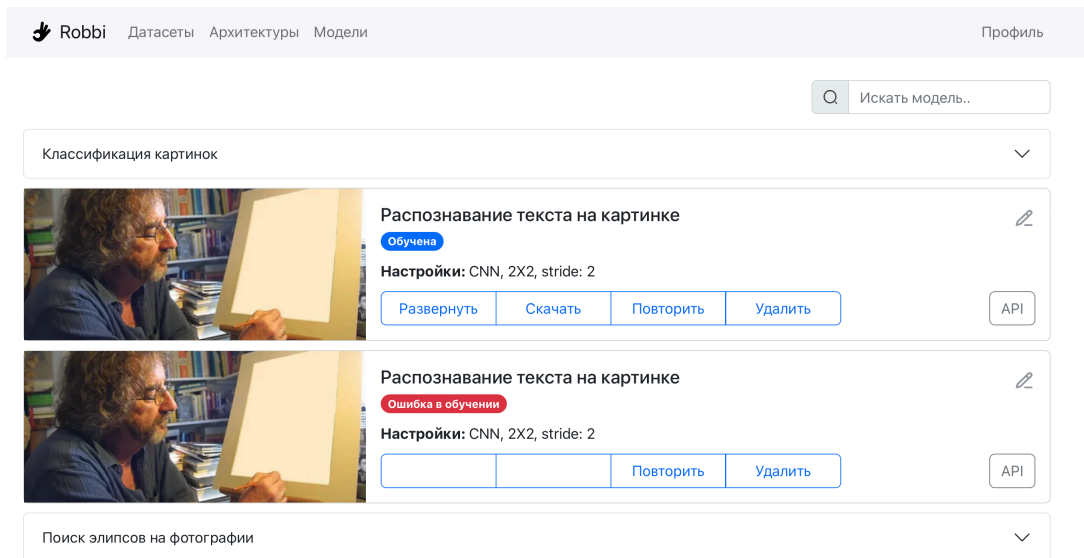


Рис. 14: Модели

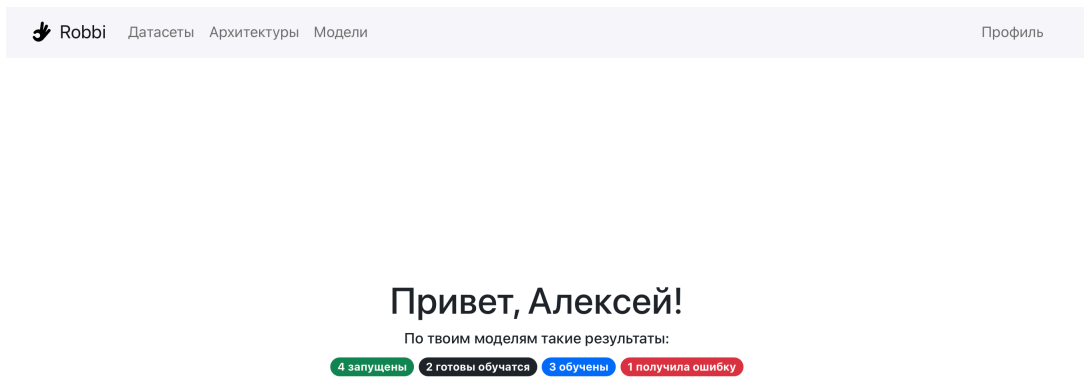


Рис. 15: Главная страница

5. Выводы по главе

В данной главе были рассмотрены различные технологии, архитектура и реализация клиентской части веб-сервиса. Веб-сервис был построен на основе архитектуры клиент-сервер, где клиентская часть обращается к бекенду для получения данных и выполнения операций. Была использована модель запроса-ответа, где фронтенд отправляет запросы к бекенду, который в свою очередь обрабатывает эти запросы и возвращает соответствующий ответ.

Один из важных аспектов взаимодействия между пользовательским интерфейсом и бекендом заключается в обновлении состояний в пользовательском интерфейсе через бекенд. Это позволяет обеспечить синхронизацию данных и обновление интерфейса при изменении состояния на сервере. Для этого были применены техники, такие как Server-Sent Events.

Было проведено подробное описание пользовательского интерфейса, включая его структуру, компоненты и взаимодействие с бекендом. Интерфейс был разработан с учетом основных сценариев использования сервиса и с использованием современных технологий веб-разработки и дизайна интерфейсов.

Заключение

В рамках данной дипломной работы была успешно выполнена разработка клиентской части веб-сервиса для платформы машинного обучения без программирования Robbi. Целью работы было создание удобного и интуитивно понятного пользовательского интерфейса, взаимодействующего с бэкендом, а также функциональности создания и настройки моделей машинного обучения без необходимости программирования.

В ходе работы были достигнуты следующие результаты:

- Разработан пользовательский интерфейс, соответствующий основным сценариям использования сервиса. Был создан удобный и интуитивно понятный интерфейс, используя современные технологии веб-разработки и дизайна интерфейсов.
- Реализована взаимодействующая с пользовательским интерфейсом часть бэкенда, которая обрабатывает запросы от пользователей и передает их в модели машинного обучения. Был создан бэкенд, обеспечивающий плавное взаимодействие между пользователем и системой.
- Разработана функциональность создания и настройки моделей машинного обучения без необходимости программирования. Был создан специальный интерфейс, позволяющий пользователям создавать и настраивать модели машинного обучения, а также настраивать их параметры.

Результатом выполнения данной дипломной работы является разработанный веб-сервис на платформе Robbi с клиентской частью, предоставляющей удобный и интуитивно понятный интерфейс для работы с моделями машинного обучения. Разработанный сервис позволяет пользователям без программирования создавать, настраивать и использовать модели машинного обучения, что значительно упрощает и ускоряет процесс работы с данными и повышает доступность и использование машинного обучения.

В ходе работы были учтены требования по эффективности, удобству использования и надежности системы. Однако, будущие исследования и разработки могут включать расширение функциональности веб-сервиса, добавление новых типов моделей машинного обучения, оптимизацию производительности и обеспечение безопасности данных. Также стоит уделить внимание улучшению пользовательского интерфейса и проведению тестирования системы для обеспечения ее надежной и эффективной работы.

В целом, данная дипломная работа является важным шагом в развитии платформы Robbi и предоставляет ценный инструмент для пользователей, интересующихся машинным обучением, но не обладающих программированием.

Список литературы

- [1] (AWS) Amazon Web Services. Amazon Web Services (AWS). — URL: <https://aws.amazon.com/> (online; accessed: 22.05.2023).
- [2] Accenture. Accenture. — URL: <https://www.accenture.com/> (online; accessed: 22.05.2023).
- [3] AutoML. AutoML. — URL: <https://cloud.google.com/automl> (online; accessed: 22.05.2023).
- [4] Axios. Axios. — URL: <https://axios-http.com/docs/intro> (online; accessed: 22.05.2023).
- [5] Azure Microsoft. Microsoft Azure. — URL: <https://azure.microsoft.com/> (online; accessed: 22.05.2023).
- [6] Bootstrap. Bootstrap. — URL: <https://getbootstrap.com/> (online; accessed: 22.05.2023).
- [7] Deloitte. Deloitte. — URL: <https://www2.deloitte.com/> (online; accessed: 22.05.2023).
- [8] (GCP) Google Cloud Platform. Google Cloud Platform (GCP). — URL: <https://cloud.google.com/> (online; accessed: 22.05.2023).
- [9] H2O.ai. H2O.ai. — URL: <https://www.h2o.ai/> (online; accessed: 22.05.2023).
- [10] IBM. Machine Learning Platform // IBM Cloud Learn. — 2021. — URL: <https://www.ibm.com/cloud/learn/machine-learning-platform> (online; accessed: 2023-05-22).
- [11] JSON. JSON // JSON. — 2021. — URL: <https://json.org> (online; accessed: 2023-05-22).
- [12] Kaggle. Kaggle. — URL: <https://www.kaggle.com/> (online; accessed: 22.05.2023).
- [13] Network Mozilla Developer. HTTP // Mozilla Developer Network. — 2021. — URL: <https://developer.mozilla.org/en-US/docs/Web/HTTP> (online; accessed: 2023-05-22).
- [14] PwC. PwC. — URL: <https://www.pwc.com/> (online; accessed: 22.05.2023).
- [15] PyTorch. PyTorch. — URL: <https://pytorch.org/> (online; accessed: 22.05.2023).
- [16] React. React. — URL: <https://reactjs.org/> (online; accessed: 22.05.2023).

- [17] Robbi. Robbi. — URL: <https://robbi.ai/> (online; accessed: 22.05.2023).
- [18] Router React. React Router. — URL: <https://reactrouter.com/> (online; accessed: 22.05.2023).
- [19] SageMaker Amazon. Amazon SageMaker. — URL: <https://aws.amazon.com/sagemaker/> (online; accessed: 22.05.2023).
- [20] Science Towards Data. How to Build a Machine Learning Model // Towards Data Science. — 2021. — URL: <https://towardsdatascience.com/how-to-build-a-machine-learning-model-439ab8fb3fb1> (online; accessed: 2023-05-22).
- [21] Scikit-learn. Scikit-learn. — URL: <https://scikit-learn.org/> (online; accessed: 22.05.2023).
- [22] Spring Java. Java Spring. — URL: <https://spring.io/> (online; accessed: 22.05.2023).
- [23] (TS) TypeScript. TypeScript (TS). — URL: <https://www.typescriptlang.org/> (online; accessed: 22.05.2023).
- [24] TensorFlow. TensorFlow. — URL: <https://www.tensorflow.org/> (online; accessed: 22.05.2023).
- [25] Watson IBM. IBM Watson. — URL: <https://www.ibm.com/watson> (online; accessed: 22.05.2023).
- [26] Wikipedia. Artificial intelligence // Wikipedia, The Free Encyclopedia. — 2021. — URL: https://en.wikipedia.org/wiki/Artificial_intelligence (online; accessed: 2023-05-22).
- [27] Wikipedia. Back-end web development // Wikipedia, The Free Encyclopedia. — 2021. — URL: https://en.wikipedia.org/wiki/Back-end_web_development (online; accessed: 2023-05-22).
- [28] Wikipedia. Front-end web development // Wikipedia, The Free Encyclopedia. — 2021. — URL: https://en.wikipedia.org/wiki/Front-end_web_development (online; accessed: 2023-05-22).
- [29] Wikipedia. Machine learning // Wikipedia, The Free Encyclopedia. — 2021. — URL: https://en.wikipedia.org/wiki/Machine_learning (online; accessed: 2023-05-22).

- [30] Wikipedia. Machine learning model // Wikipedia, The Free Encyclopedia.— 2021. — URL: https://en.wikipedia.org/wiki/Machine_learning_model (online; accessed: 2023-05-22).
- [31] Wikipedia. User interface // Wikipedia, The Free Encyclopedia.— 2021. — URL: https://en.wikipedia.org/wiki/User_interface (online; accessed: 2023-05-22).
- [32] Wikipedia. Web service // Wikipedia, The Free Encyclopedia.— 2021. — URL: https://en.wikipedia.org/wiki/Web_service (online; accessed: 2023-05-22).
- [33] YAML. YAML // YAML.— 2021. — URL: <https://yaml.org> (online; accessed: 2023-05-22).
- [34] react-flow renderer. react-flow renderer.— URL: <https://github.com/wbkd/react-flow> (online; accessed: 22.05.2023).