



**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО
ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ**

**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБ-
РАЗОВАНИЯ**

**«МОСКОВСКИЙ АВИАЦИОННЫЙ ИНСТИТУТ
(национальный исследовательский университет)»**

Программа стратегического академического лидерства «Приоритет – 2030»

ПРОЕКТ «ЦИФРОВАЯ КАФЕДРА»

**Дополнительная профессиональная программа профессиональной переподготовки
«Методы искусственного интеллекта и предиктивная аналитика в проектах дефекто-
скопии»**

ИТОГОВАЯ АТТЕСТАЦИОННАЯ РАБОТА (ИТ-ПРОЕКТ)

на тему: «Предсказание вероятности задержки рейсов в аэропорту»

Руководитель:	Азанов Валентин Михайлович	(_____)
Консультант:	Судаков Владимир Анатольевич	(_____)
Рецензент:	Агульник Алексей Борисович	(_____)

К защите допустить

Руководитель
Азанов Валентин Михайлович

ДПП
(_____)

ПП

2023 год

Москва 2023

КОМАНДА ПРОЕКТА:

№	Фамилия, Имя, Отчество	Группа по ООП	Группа по ДПП	Подпись
1	Попов Матвей Романович	М8О-308Б-20	И08-806ПП-44-22	
2	Ларшин Тимофей Андреевич	М8О-302Б-20	И08-806ПП-44-22	
3	Годовник Артем Геннадьевич	М3О-311Б-20	И08-806ПП-44-22	
4	Чемодурова Елизавета Вадимовна	М3О-311Б-20	И08-806ПП-44-22	
5	Бокарев Степан Михайлович	М3О-319Бк-20	И08-806ПП-41-22	
6	Матвеева Татьяна Николаевна	М3О-315Б-20	И08-806ПП-38-22	
7	Диденко Егор Денисович	М3О-315Б-20	И08-806ПП-44-22	
8	Титеев Александр Максимович	М8О-209М-21	И08-806ПП-45-22	
9	Каширин Кирилл Дмитриевич	М8О-308Б-20	И08-806ПП-41-22	
10	Аринушкин Артем Максимович	М3О-435Б-19	И08-806ПП-44-22	
11	Иванов Денис Иванович	М1О-408Б-19	И08-806ПП-44-22	
12	Милютина София Алексеевна	М8О-305Б-20	И08-806ПП-44-22	

СПИСОК ИСПОЛНИТЕЛЕЙ:

№	Фамилия, Имя, Отчество	Название и номер раздела
1	Попов Матвей Романович	Реферат, введение, 3.1 Характеристика условий и места использования разработки, заключение
2	Ларшин Тимофей Андреевич	1.1 Потребности в разработке предсказания задержек авиарейсов
3	Годовник Артем Геннадьевич	1.2 Анализ существующих предиктивных решений в авиационной индустрии, 3.3 Технические характеристики разработанного решения
4	Чемодурова Елизавета Вадимовна	1.3 Обоснование цели и задач. Техническое задание на разработку, 3.2 Формат работы разработанного решения
5	Бокарев Степан Михайлович	Выводы по разделу 1, выводы по разделу 3
6	Матвеева Татьяна Николаевна	2.1 Теоретическое решение задачи предсказания задержек авиарейсов: подход и методология
7	Диденко Егор Денисович	2.3 Описание программной разработки предсказательной модели задержек авиарейсов
8	Титеев Александр Максимович	2.4 План разработки проекта
9	Каширин Кирилл Дмитриевич	2.2 Стэк используемых технологий, выводы по разделу 2

Руководитель работы:

к.ф.-м.н., старший преподаватель кафедры 804

Подпись Азанов В. М.

Консультант:

д.т.н., доцент, профессор кафедры 806

Подпись Судаков В. А.

Рецензент:

д.т.н., старший научный сотрудник, заведующий кафедрой 201

Подпись Агульник А. Б.

РЕФЕРАТ

Итоговая аттестационная работа состоит из 200 страниц, 20 рисунков, 3 таблиц, 13 использованных источников, 5 приложений.

ПРЕДСКАЗАНИЕ ЗАДЕРЖЕК АВИАРЕЙСОВ, АНАЛИЗ РЫНКА, АЛГОРИТМЫ МАШИННОГО ОБУЧЕНИЯ, СБОР ДАННЫХ, ПОДГОТОВКА ДАННЫХ, ОБУЧЕНИЕ МОДЕЛЕЙ, ОЦЕНКА ОБУЧЕНИЯ, ТЕЛЕГРАМ-БОТ, ТЕХНОЛОГИЧЕСКОЕ РЕШЕНИЕ, ЭФФЕКТИВНОСТЬ

Итоговая аттестационная работа выполнена в формате IT-проекта «Предсказание вероятности задержки рейсов в аэропорту».

Объектом разработки в данной работе является система предсказания задержек авиарейсов.

Цель работы – создание и разработка IT-решения, которое позволит предсказывать возможные задержки вылета или прибытия авиарейсов с высокой точностью.

Для достижения поставленных целей был проведен анализ рынка и изучены основные алгоритмы машинного обучения, применяемые в области предсказания задержек авиарейсов. Также был осуществлен сбор и подготовка данных, включающая очистку, преобразование и масштабирование информации. Разработаны и обучены модели машинного обучения, которые способны делать предсказания задержек авиарейсов на основе предоставленных данных. Основное содержание работы состояло в разработке программного обеспечения, включая создание телеграм-бота, который позволяет пользователям получать информацию о задержках авиарейсов и быть в курсе последних обновлений.

Разработанное технологическое решение представляет собой инновационный инструмент для предсказания задержек авиарейсов. Оно основано на современных технологиях, включая Python, TensorFlow, Jupyter Notebook,

Docker, Telegram, GitLab и Visual Studio Code. Архитектура системы представляет собой клиент-серверное решение, где телеграм-бот является клиентом, а сервер обрабатывает запросы, выполняет предсказания и взаимодействует с базой данных и моделью машинного обучения. Разработанное решение обладает рядом преимуществ. Во-первых, высокая точность предсказаний обеспечивает своевременное оповещение авиакомпаний и пассажиров о возможных задержках, позволяя им принимать соответствующие меры. Во-вторых, удобный пользовательский интерфейс телеграм-бота обеспечивает удобство и интуитивно понятное взаимодействие для пользователей. Другие преимущества включают гибкость и масштабируемость разработки, возможность быстрой обработки больших объемов данных с использованием Jupyter Notebook, переносимость и повторяемость решения с помощью Docker, а также интеграцию с существующей инфраструктурой авиакомпаний. Разработанное решение также имеет потенциал для дальнейшего развития, включая добавление новых моделей машинного обучения, оптимизацию алгоритмов и интеграцию с другими системами и сервисами. В целом, оно предоставляет эффективный и надежный инструмент для предсказания задержек авиарейсов, способствуя повышению безопасности, операционной эффективности и удовлетворенности пассажиров.

Результаты разработки предназначены для широкого круга заинтересованных сторон в авиационной отрасли. В первую очередь, они предназначены для авиакомпаний и их операционных служб, которые могут использовать предсказания задержек для более эффективного управления расписанием рейсов, ресурсами и пассажирскими потоками. Это поможет им принимать своевременные решения и минимизировать негативные последствия задержек для пассажиров и бизнеса. Пассажиры также являются целевой аудиторией разработки. Результаты могут быть предоставлены им в реальном времени, позволяя получать информацию о возможных задержках и принимать соответствующие меры, такие как изменение планов или контакт с авиакомпанией. Это

повышает уровень обслуживания и удовлетворенность пассажиров, а также помогает им планировать свои поездки более эффективно. Кроме того, результаты разработки могут быть полезны для аэропортов, служб безопасности, поставщиков услуг в авиационной отрасли и других заинтересованных сторон. Они могут использовать предсказания задержек для оптимизации своих операций, повышения безопасности и обеспечения более гладкого и эффективного функционирования авиационной системы в целом. Использование разработанного решения предоставляет ряд значимых преимуществ. Во-первых, повышается точность предсказания задержек авиарейсов, что способствует более эффективному планированию и управлению рейсами. Это может снизить негативные последствия для пассажиров, экономии времени и ресурсов авиакомпаний.

СОДЕРЖАНИЕ

РЕФЕРАТ	5
ТЕРМИНЫ И ОПРЕДЕЛЕНИЯ.....	10
ВВЕДЕНИЕ.....	12
1 ПОСТАНОВКА ЗАДАЧИ ПРЕДСКАЗАНИЯ ЗАДЕРЖЕК РЕЙСОВ В АЭРОПОРТУ	20
1.1 Потребности в разработке предсказания задержек авиарейсов	20
1.2 Анализ существующих предиктивных решений в авиационной индустрии	25
1.3 Обоснование цели и задач. Техническое задание на разработку	26
ВЫВОДЫ ПО РАЗДЕЛУ 1	36
2 РАЗРАБОТКА ПРЕДСКАЗАТЕЛЬНОЙ МОДЕЛИ ЗАДЕРЖЕК АВИАРЕЙСОВ. ПОДХОД И РЕАЛИЗАЦИЯ.....	38
2.1 Теоретическое решение задачи предсказания задержек авиарейсов: подход и методология	38
2.2 Стэк используемых технологий.....	70
2.3 Описание программной разработки предсказательной модели задержек авиарейсов.....	89
2.4 План разработки проекта.....	117
ВЫВОДЫ ПО РАЗДЕЛУ 2	126
3 РЕЗУЛЬТАТЫ РАБОТЫ.....	130
3.1 Характеристика условий и места использования разработки	130
3.2 Формат работы разработанного решения (Чемодурова Е. В.).....	134
3.3 Технические характеристики разработанного решения.....	152
ВЫВОДЫ ПО РАЗДЕЛУ 3	163
ЗАКЛЮЧЕНИЕ	165
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ.....	173
ПРИЛОЖЕНИЕ А Паспорт проекта.....	174
ПРИЛОЖЕНИЕ Б Содержимое main.py	180
ПРИЛОЖЕНИЕ В Содержимое Dockerfile.....	190
ПРИЛОЖЕНИЕ Г Содержимое docker-compose.yml.....	191

ПРИЛОЖЕНИЕ Д Фрагмент датасета	192
--------------------------------------	-----

ТЕРМИНЫ И ОПРЕДЕЛЕНИЯ

В настоящей итоговой аттестационной работе применяют следующие термины с соответствующими определениями:

Предсказание задержек авиарейсов — процесс использования данных и алгоритмов для оценки вероятности задержки вылета или прибытия авиарейса. Целью является предоставление оперативной информации о возможных изменениях в расписании и своевременное информирование пассажиров и авиакомпаний

Машинное обучение — область искусственного интеллекта, которая изучает разработку алгоритмов и моделей, способных обучаться на основе данных и делать прогнозы или принимать решения без явного программирования

Алгоритмы машинного обучения — математические модели и процедуры, используемые для обучения компьютерных систем на основе данных. Они позволяют системе автоматически находить закономерности и делать предсказания на основе полученных знаний

Датасет — коллекция данных, содержащая информацию, необходимую для обучения и проверки моделей машинного обучения. Датасет может включать различные признаки и метки, необходимые для обучения алгоритмов

Обработка данных — процесс преобразования и подготовки данных для использования в алгоритмах машинного обучения. Это может включать очистку данных, масштабирование, кодирование категориальных признаков и другие преобразования

Нейронные сети — модели машинного обучения, построенные по аналогии с работой нервной системы человека. Они состоят из множества взаимосвязанных искусственных нейронов, способных обрабатывать информацию и делать предсказания

Telegram-бот — программа, работающая на платформе Telegram, которая позволяет взаимодействовать с пользователями через сообщения. Telegram боты могут выполнять различные задачи, включая предоставление информации, обработку запросов и выполнение команд

Aiogram — библиотека программирования на языке Python, предназначенная для создания и управления телеграм ботами. Она предоставляет набор инструментов и функций для разработки функциональных и интерактивных ботов

Jupyter Notebook — интерактивная среда разработки, позволяющая создавать и выполнять код, визуализировать данные и документировать процесс разработки. Jupyter Notebook поддерживает различные языки программирования, включая Python

Docker — платформа, позволяющая упаковывать, доставлять и запускать приложения в контейнерах. Контейнеры Docker предоставляют изолированное окружение для приложений, включая все необходимые зависимости и конфигурации, обеспечивая их переносимость и повторяемость

ВВЕДЕНИЕ

Современный мир развивается с недостижимой ранее скоростью, внедряя все новые и новые технологии в наши жизни, существенно меняя существующие устои. Так, благодаря развитию технологий, в последние десятилетия отношения между странами и континентами становятся все ближе, стирая все возможные границы и барьеры для сотрудничества. Одним из ключевых факторов тенденции сближения является бурное развитие авиационной индустрии, услуги которой стали основой большинства бизнес-путешествий по всему миру. С начала 2000 года количество пассажиров на регулярных рейсах выросло более чем на 65%, а количество инцидентов на них снизилось на 40%. Тем не менее, несмотря на бурное количественное и качественное развитие авиации, существенная доля рейсов все еще происходит с задержками. В зависимости от направления и авиакомпании, до 20% всех рейсов выполняются с задержками. Данное положение дел является не только неудобным для путешественников, но и недопустимым для международных компаний – любая задержка ведет к увеличению операционных расходов, приводя к финансовым потерям. По этой причине, бизнес рассматривает любые задержки авиаперелетов как существенный риск для своих бизнес-операций, пытаясь максимально снизить возможные угрозы. На сегодняшний день основными подходами для решения задачи снижения рисков, связанных с задержками во время путешествия, являются такие простые решения, как увеличение общего времени, закладываемого на поездку, а также страхование рисков у страховых агентов. Несмотря на частичное решение проблемы и снижение рисков, данные подходы очень дороги в организации, а общая проблема неопределенности статуса авиарейса остается неразрешенной.

Качественно новым и более интересным является другой подход к решению данной задачи – предсказание самих задержек, а также вероятности их возникновения на базе исторических и погодных данных. Благодаря использованию аналитического подхода к снижению рисков, возможно не только

уменьшение общего количества возможных инцидентов, но и существенное снижение количества связанных затрат. Так, благодаря внедрению новой платформы для оценки рисков, компании смогут заранее предсказывать исход определенных рейсов, и, в зависимости от этого, применять те, или иные подходы по устранению угроз для бизнеса. Внедрение системы подобного рода позволит не только снизить общий уровень неопределенности, но и существенно уменьшить стоимость организации командировок для сотрудников.

Благодаря использованию данного сервиса, организации смогут минимизировать риски, связанные с задержками поездок, а, следовательно, и затраты на осуществление перевозок своих сотрудников.

Вопрос возможности создания модели машинного обучения для предсказания вероятности возможной задержки или отмены рейса не представляется новым – существует ряд работ, так, или иначе освещает данную проблему. Однако, представленные работы, скорее ориентированы больше на теоретическую часть вопроса создания данной модели, а также основываются на данных, содержащих исключительно исторические данные за длительный период времени.

Данная же работа, напротив, представляет больший упор на реальную и практическую часть реализации данной модели, а также создание интерфейса для взаимодействия и интеграции с другими программными решениями. Также данная работа частично основывается на работах по машинному обучению, предоставляющих теоретические сведения о существующих алгоритмах машинного обучения, а также математических основах их реализации.

Таким образом, выполненная работа актуальна и с научно-методической/теоретической, и с практической точек зрения.

Цель работы – создание ИТ-сервиса, позволяющего на основе исторических данных по перелетам и погодным условиям, с помощью алгоритмов машинного обучения прогнозировать вероятность возникновения задержек авиарейсов.

Для достижения поставленной цели в работе были решены следующие задачи:

1. анализ рынка данных, представляющих авиационную статистику и исторические данные по перелетам за прошлые периоды;
2. рассмотрение основных алгоритмов и методов машинного обучения для осуществления предобработки данных и классификации данных;
3. осуществление предобработки собранных ранее данных, исключение неполных данных. Дифференциация данных по аэропорту вылета для упрощения создания общего классификатора;
4. разработка и обучение ряда моделей классификации на небольшой выборке данных с целью выбора наиболее качественной модели машинного обучения;
5. проведение оценки полученных классификаторов и выбор модели, наиболее точно описывающей данные;
6. обучение выбранного классификатора на всей выборке данных и получение финальных моделей классификаторов;
7. создание Telegram-бота для обеспечения удаленного доступа к полученным моделям и предсказания статуса рейса.

Объектом разработки является процесс организации бизнес-путешествий с использованием услуг регулярных рейсов авиакомпаний.

Предметом исследования в работе являются основные факторы, влияющие на задержки регулярных рейсов авиакомпаний.

Работа основывалась на следующих инструментах и методах:

1. Visual Studio Code — текстовый редактор кода для кроссплатформенной разработки веб- и облачных приложений;
2. TensorFlow — открытая библиотека для машинного обучения, разработанная компанией Google;
3. Pandas — библиотека для работы с большими объемами данных в среде Python;
4. Aiogram — популярная библиотека для разработки Telegram-ботов на языке Python.

Результаты работы:

1. был проведен анализ рынка и изучены существующие методы предсказания задержек авиа рейсов. Это позволило определить наиболее эффективные методы и алгоритмы машинного обучения для использования в разрабатываемом решении;
2. были рассмотрены основные алгоритмы машинного обучения, включая регрессию, деревья принятия решений, случайные леса, градиентный бустинг и нейронные сети. Это позволило выбрать наиболее подходящие алгоритмы для предсказания задержек авиа рейсов;
3. были собраны и подготовлены данные о рейсах, включая информацию о времени вылета, погодных условиях и других факторах. Это обеспечило достаточный объем данных для обучения моделей;
4. были обучены различные модели машинного обучения на подготовленных данных, включая регрессионные модели, деревья принятия решений, случайные леса и нейронные сети. Это позволило выбрать наиболее точную модель для предсказания задержек авиа рейсов;
5. была проведена оценка качества обучения моделей на основе различных метрик, что подтвердило высокую точность и надежность разработанной модели;

6. была разработана и получена финальная модель предсказания задержек авиарейсов, которая может быть использована для предсказания задержек с высокой точностью;

7. был создан Telegram Bot, который обеспечивает удобный интерфейс для взаимодействия с пользователем и предоставляет актуальные предсказания задержек авиарейсов.

Результаты работы, полученные в ходе данного исследования и разработки, предназначены для ряда конкретных целей и применений. Ниже приведены основные области, в которых могут быть использованы эти результаты:

1. авиаперевозчики и аэропорты: разработанное решение по предсказанию задержек авиарейсов может быть ценным инструментом для авиаперевозчиков и аэропортов. Оно позволяет им оперативно предсказывать возможные задержки и принимать своевременные меры для снижения их негативного воздействия. Предоставление актуальных и надежных предсказаний поможет авиакомпаниям планировать свои ресурсы и графики вылетов, а аэропортам оптимизировать процессы посадки и обслуживания пассажиров;

2. пассажиры: результаты работы могут быть полезными для пассажиров, так как предоставляют информацию о возможных задержках авиарейсов. Это позволит пассажирам планировать свое время более эффективно, предварительно оценивая возможные задержки и принимая соответствующие меры. Такая информация поможет пассажирам избежать непредвиденных ситуаций и минимизировать неудобства, связанные с задержками;

3. управление аэропортом и графиками вылетов: разработанное решение может быть включено в системы управления аэропортом для автоматизации процессов предсказания задержек. Это позволит оптимизировать распределение ресурсов, улучшить планирование графиков вылетов и обслуживания рейсов. Администрация аэропортов сможет принимать более обоснованные решения и улучшать качество обслуживания пассажиров;

4. дальнейшее развитие и исследования: результаты работы могут послужить отправной точкой для дальнейшего развития и исследований в области предсказания задержек авиарейсов. Возможно, улучшение моделей машинного обучения, расширение набора данных или добавление новых функциональностей в разработанный Telegram Bot. Эти результаты могут стать основой для будущих исследований и разработок, направленных на повышение точности и эффективности предсказания задержек;

В целом, результаты данной работы имеют практическое применение в авиационной отрасли, способствуя улучшению оперативности принятия решений, оптимизации работы авиаперевозчиков и аэропортов, а также повышению удовлетворенности пассажиров. Дальнейшее использование и развитие этих результатов могут привести к существенному улучшению процессов предсказания задержек и управления авиационными операциями.

Использование разработанного решения по предсказанию задержек авиарейсов может принести ряд полезных эффектов и преимуществ. Ниже представлены основные положительные аспекты, связанные с использованием данной разработки:

1. снижение неопределенности: предсказание задержек авиарейсов помогает снизить неопределенность и предоставляет более точную информацию о времени вылета. Это позволяет пассажирам и авиакомпаниям более эффективно планировать свое время, избегать долгих ожиданий и связанных с ними неудобств;

2. оптимизация ресурсов: авиаперевозчики и аэропорты могут использовать предсказания задержек для более эффективного распределения своих ресурсов. Они могут оптимизировать работу персонала, обслуживание рейсов, заправку самолетов и другие процессы, чтобы справиться с возможными задержками;

3. улучшение планирования: предсказания задержек авиарейсов помогают улучшить планирование графиков вылетов и прибытий. Авиакомпании и аэропорты могут использовать эти предсказания для оптимизации своих расписаний и предупреждения пассажиров о возможных изменениях;

4. повышение удовлетворенности пассажиров: информирование пассажиров о возможных задержках позволяет им принять необходимые меры заранее. Это может включать изменение маршрута, перепланирование встреч и другие корректировки. Такие возможности помогают пассажирам избежать стрессовых ситуаций и улучшают их общее впечатление от авиаперевозок;

5. экономические выгоды: предсказание задержек авиарейсов может привести к экономическим выгодам для авиакомпаний и аэропортов. Они смогут сократить затраты на дополнительные меры по борьбе с задержками, такие как компенсации пассажирам или дополнительные затраты на ресурсы в случае задержек. Также снижение числа задержек может повысить репутацию авиакомпании и привлечь больше пассажиров;

6. улучшение безопасности: более точное предсказание задержек авиарейсов позволяет улучшить безопасность воздушных перевозок. Пассажиры и экипаж могут быть предупреждены о возможных изменениях в расписании и принять соответствующие меры для обеспечения безопасности полета;

7. развитие новых решений: результаты данной работы могут способствовать развитию новых методов и подходов к предсказанию задержек авиарейсов. Они могут служить основой для дальнейших исследований и улучшения существующих моделей, алгоритмов и инструментов;

8. продвижение инноваций: использование современных технологий, таких как машинное обучение и анализ данных, для предсказания задержек авиарейсов способствует продвижению инноваций в авиационной отрасли. Это может привлечь внимание к новым методам и технологиям в области авиационных операций и стимулировать их применение в других сферах.

В целом, использование разработанного решения по предсказанию задержек авиарейсов дает возможность повысить эффективность, снизить риски и улучшить опыт пассажиров в авиационной отрасли. Это важный шаг в направлении оптимизации авиационных операций и повышения уровня сервиса в сфере авиаперевозок.

1 ПОСТАНОВКА ЗАДАЧИ ПРЕДСКАЗАНИЯ ЗАДЕРЖЕК РЕЙСОВ В АЭРОПОРТУ

1.1 Потребности в разработке предсказания задержек авиарейсов

В данном подразделе будет рассмотрена необходимость и актуальность разработки предсказания задержек авиарейсов. Будут описаны основные причины, по которым такая разработка является востребованной в современной авиационной индустрии.

Значимость проблемы задержек авиарейсов — в современном мире авиаперевозки играют ключевую роль в глобальной транспортной системе. Миллионы пассажиров каждый день полагаются на авиакомпании, чтобы добраться до своих пунктов назначения. Однако, проблема задержек авиарейсов остается серьезной и актуальной, вызывая значительные неудобства и негативные последствия как для пассажиров, так и для авиакомпаний и других участников индустрии. Первоначально, следует отметить, что задержки авиарейсов имеют широкий спектр причин. Это могут быть погодные условия, технические неисправности, ограничения воздушного пространства, задержки в аэропортах и многие другие факторы. Каждый из этих элементов может привести к непредвиденным изменениям в расписании полетов и задержкам, оказывая негативное влияние на авиационную систему в целом. Задержки авиарейсов имеют значительные последствия для пассажиров. Во-первых, они приводят к неопределенности и ожиданию в аэропорту. Пассажиры вынуждены тратить свое время на ожидание, что часто сопровождается дополнительными расходами на питание и проживание. Кроме того, задержки могут вызывать стресс и дискомфорт у пассажиров, особенно у тех, кто имеет строгое расписание и не может позволить себе опоздание на важные деловые встречи или личные события. Однако, задержки авиарейсов несут негативные последствия не только для пассажиров, но и для авиакомпаний и экономики в целом. Авиакомпании сталкиваются с необходимостью компенсировать задержки пасса-

жирам, что может вызывать значительные финансовые потери. Кроме того, задержки авиарейсов влияют на операционную эффективность авиакомпаний, так как приводят к нарушению расписания полетов, перераспределению ресурсов и дополнительным затратам на обслуживание пассажиров. Негативное влияние задержек авиарейсов распространяется и на экономику в целом. Исследования показывают, что задержки авиарейсов могут привести к потере миллионов долларов ежегодно из-за компенсаций, упущенной выручки, дополнительных затрат на обслуживание пассажиров и других факторов. Это оказывает негативное воздействие на финансовую устойчивость авиакомпаний и может привести к снижению конкурентоспособности всей авиационной индустрии. В свете вышеизложенного становится ясным, что разработка эффективного инструмента для предсказания задержек авиарейсов является неотъемлемой потребностью. Предсказательная модель, основанная на анализе исторических данных и применении современных методов машинного обучения, может помочь в предвидении возможных задержек и принятии соответствующих мер для их предотвращения или минимизации. Это позволит снизить негативные последствия для пассажиров, авиакомпаний и экономики в целом, обеспечивая более эффективное и надежное функционирование авиационной системы.

Экономические аспекты и потери из-за задержек — задержки авиарейсов вызывают значительные экономические потери как для авиакомпаний, так и для пассажиров. Эти потери могут оказывать серьезное влияние на финансовую устойчивость авиационных компаний и снижать их конкурентоспособность. Для пассажиров задержки авиарейсов могут приводить к потере драгоценного времени, упущенным деловым возможностям, переплате за дополнительные услуги и необходимости переезда на другие рейсы. Первоначальные затраты, связанные с задержками авиарейсов, включают компенсации пассажирам, обеспечение дополнительного питания и проживания, а также повтор-

ное бронирование и возврат билетов. Для авиакомпаний это становится финансовым бременем, особенно при массовых задержках, когда несколько рейсов задерживаются одновременно. Дополнительные затраты на компенсации и обслуживание пассажиров могут существенно снижать прибыльность авиакомпании и оказывать отрицательное влияние на ее финансовое положение. Кроме того, задержки авиарейсов могут приводить к упущенной выручке. Пассажиры, ожидающие задержанный рейс, могут потерять возможность присутствовать на важном деловом мероприятии, встрече или событии, что может повлечь за собой потерю дохода или деловых контактов. В случае туристических поездок пассажиры могут пропустить предварительно запланированные экскурсии, отели и другие платные услуги, что также приведет к дополнительным финансовым потерям. Более того, задержки авиарейсов могут оказывать косвенное влияние на экономику региона или страны. Путешествия и командировки являются важными стимулами для развития бизнеса и туризма. Если авиарейсы регулярно задерживаются или отменяются, это может отпугнуть инвесторов, деловых партнеров и туристов, что приведет к снижению экономической активности, убыточности предприятий и потере рабочих мест. Поэтому разработка предсказательной модели для задержек авиарейсов является не только актуальной, но и экономически обоснованной.

Улучшение пассажирского опыта и комфорта — задержки авиарейсов являются одним из наиболее раздражающих и неприятных моментов для пассажиров. Неопределенность и ожидание в аэропорту могут значительно повлиять на пассажирский опыт и комфорт. Разработка предсказательной модели для задержек авиарейсов имеет потенциал улучшить эту ситуацию и создать более приятное и безопасное путешествие для пассажиров. Первоначально, одним из ключевых аспектов, который следует рассмотреть, является уведомление пассажиров о возможных задержках заранее. Когда пассажиры знают о предстоящих задержках, они могут принять соответствующие меры и

подготовиться заранее. Например, они могут пересмотреть свои планы на случай, если задержка превысит определенное время, или заранее информировать своих близких и работодателей о возможных изменениях во времени прибытия. Такое уведомление позволит пассажирам более гибко и эффективно планировать свои действия, избегая лишнего ожидания и дискомфорта. Далее, разработка предсказательной модели также может предоставить пассажирам альтернативные варианты пересадок или маршрутов в случае возникновения задержек. Это особенно важно для пассажиров, у которых есть тесные временные окна для пересадки на другие рейсы. Предупреждение о задержке и предложение альтернативных вариантов помогут избежать пропуска рейсов и дополнительных неудобств, связанных с переносом багажа и организацией новой пересадки. Такие возможности улучшат общий пассажирский опыт, снизят стресс и повысят уровень доверия к авиакомпании. Более того, предсказательная модель может быть интегрирована в системы онлайн-бронирования и пассажирские приложения. Пассажиры смогут получать информацию о задержках и обновлениях прямо на своих мобильных устройствах. Это позволит им быть в курсе последних изменений и реагировать соответствующим образом. Такие приложения могут также предлагать рекомендации по местам отдыха или возможностям занять время в аэропорту во время ожидания. Это создаст более позитивное и комфортное впечатление у пассажиров, а также повысит уровень обслуживания и лояльности к авиакомпании. В целом, разработка предсказательной модели для задержек авиарейсов может значительно улучшить пассажирский опыт и комфорт, предоставляя информацию заранее, предлагая альтернативные варианты маршрутов, интегрируясь в мобильные приложения и способствуя формированию позитивного образа авиационной индустрии. Это важный шаг в совершенствовании авиационных услуг и удовлетворении потребностей пассажиров.

Экологические аспекты и снижение выбросов — задержки авиарейсов имеют не только экономические и операционные последствия, но и значительное воздействие на окружающую среду. Длительное ожидание самолетов на земле или в воздухе приводит к дополнительному потреблению топлива, что приводит к увеличению выбросов парниковых газов и негативно влияет на климат и качество воздуха. Разработка предсказательной модели для задержек авиарейсов имеет потенциал снизить воздействие авиации на окружающую среду путем оптимизации движения самолетов и предотвращения неэффективных полетов. Около 2% всех выбросов парниковых газов в мире связаны с авиационной деятельностью, а эта цифра продолжает расти с увеличением объема авиаперевозок. Использование топлива на самолетах, особенно при длительных задержках, ведет к выбросам углекислого газа (CO_2), оксидов азота (NO_x), водяных паров и других вредных веществ. Как результат, авиация оказывает негативное воздействие на качество воздуха и климат, способствуя изменению климата и загрязнению окружающей среды. Разработка предсказательной модели для задержек авиарейсов имеет потенциал сократить вредное воздействие авиации на окружающую среду. Предварительное определение возможных задержек позволит авиационным компаниям и аэропортам принимать более осознанные решения, чтобы снизить количество полетов, требующих дополнительного топлива, или перенаправить самолеты на более эффективные маршруты. Это может включать в себя изменение высоты полета, выбор более эффективных маршрутов, предупреждение пассажиров заранее о возможных задержках и пересадках, а также согласование с терминалами и персоналом аэропортов для оптимизации процессов обслуживания. Эффективное предсказание задержек авиарейсов позволит также улучшить планирование ресурсов и операций в авиационной индустрии. В целом, разработка предсказательной модели для задержек авиарейсов является не только средством улучшения операционной эффективности и комфорта пассажиров, но и важным шагом к снижению вредного воздействия авиации на окружающую

среду. Предварительное определение задержек позволит принимать более осознанные решения, снижать выбросы парниковых газов, оптимизировать использование ресурсов и повышать экологическую устойчивость авиационной индустрии.

Таким образом, в настоящем проекте была рассмотрена проблема задержек авиарейсов и необходимость разработки предсказания задержек с целью снижения негативных последствий для пассажиров, авиакомпаний и окружающей среды.

1.2 Анализ существующих предиктивных решений в авиационной индустрии

Несмотря на высокую актуальность и наличие реальной необходимости в предсказании статуса рейса, рынок на данный момент не представляет комплексных решений для решения данной задачи. В большей мере все исследования носят исключительно академический характер и нацелены исключительно на построение статистической модели, основанной на исторических данных за последние десятилетия. Так, Даниэл Фабиен в своей работе *Predicting the flight delays* (Fabien Daniel, 2017) рассматривает вопрос создания регрессионной модели для предсказания точного времени задержки рейса на базе исторических данных по США за 2015 год. Также существует ряд аналогичных работ, отличающихся исключительно используемыми методами обработки данных, решающих аналогичную проблему. Основными особенностями данных работ является ориентация на предсказание точного времени задержки рейса, а также крайне низкое качество предсказания длительных задержек.

Говоря о коммерческих реализациях возможности предсказания задержек рейсов, то наиболее активно на данном рынке представлена компания Google, с недавнего времени предоставляющая информацию клиентам из США о возможной задержке рейса в своем мета-поисковике авиабилетов. Однако, и эта реализация не может быть рассмотрена как полноценный B2B сервис, так как данные предоставляются исключительно на страницах поисковой

выдачи собственного поисковика авиабилетов, и не могут быть использованы сторонними разработчиками и компаниями для интеграции в свои сервисы.

Также стоит отметить, что существует ряд ИТ-сервисов (FlightRadar24, FlightStats, FlightAware), предоставляющих информацию о текущем состоянии рейсов и данные о задержках конкретных бортов самолетов за несколько часов до вылета, что дает возможность косвенно предсказывать статус рейса, однако, и они не предоставляют ни на своих страницах, ни в качестве сервиса для своих клиентов, услуг по предварительному предсказанию статуса рейса, а также каких либо предиктивных сервисов для оценки вероятности возникновения задержек. На текущий момент доступна лишь информация о текущем статусе рейса, обновляющаяся за несколько часов до вылета рейса.

Таким образом, создаваемый в данной работе сервис будет значительно отличаться от существующих реализаций как по модели предоставления доступа, так и по методам предсказания статусов рейсов.

1.3 Обоснование цели и задач. Техническое задание на разработку

Основной целью данного исследования является разработка предсказательной модели для задержек авиа рейсов с целью снижения негативных последствий для пассажиров, авиакомпаний и окружающей среды.

Авиационная индустрия является важным сектором в мировой экономике, и в то время как задержки авиа рейсов оказывают значительное влияние на различные аспекты этой отрасли.

В первую очередь, основная цель заключается в разработке эффективного инструмента, который способен предсказывать задержки авиа рейсов с высокой точностью и достаточным временным прогнозом. Это позволит принимать соответствующие меры заранее и предупреждать пассажиров о возможных задержках, что повысит уровень сервиса авиакомпании. Путем предо-

ставления достоверной информации о задержках авиарейсов пассажирам будет предоставлена возможность принятия взвешенных решений, включая перепланирование своих маршрутов или организацию альтернативных вариантов перелетов.

Вторая грань основной цели заключается в снижении экономических потерь для авиакомпаний. Задержки авиарейсов могут приводить к значительным финансовым убыткам, связанным с компенсациями, упущенной выручкой и дополнительными затратами на обслуживание пассажиров. Разработка предсказательной модели, способной предупреждать о возможных задержках, позволит авиакомпаниям принимать более осознанные решения, оптимизировать использование ресурсов и сократить экономические потери. Это поможет обеспечить финансовую устойчивость авиакомпаний и повысить их конкурентоспособность.

Третья грань основной цели связана с повышением комфорта пассажиров. Задержки авиарейсов, ожидание в аэропорту или в воздухе могут создавать стресс, неудобства и разочарование у клиентов авиакомпаний. Например, если это первый перелет в жизни пассажира, задержка авиарейса вызовет желание никогда больше не пользоваться конкретной авиакомпанией, или навсегда перестать летать на самолетах. Разработка предсказательной модели для задержек авиарейсов позволит пассажирам получать достоверную информацию о возможных задержках и более эффективно планировать свое время. Это создаст значительно более комфортные условия для пассажиров и поможет улучшить их общее впечатление от авиаперевозок.

Четвертая грань основной цели связана с повышением операционной эффективности и оптимизацией движения самолетов. Разработка предсказательной модели позволит авиационным компаниям и аэропортам принимать более рациональные решения, связанные с планированием рейсов, расстановкой ре-

сурсов и управлением трафиком. Оптимизация движения самолетов снизит задержки, улучшит пунктуальность и позволит более эффективно использовать инфраструктуру аэропортов и ресурсы авиакомпаний. Это приведет к улучшению операционной эффективности, снижению затрат.

Анализ данных о задержках авиарейсов — для разработки эффективной предсказательной модели задержек авиарейсов необходимо провести тщательный анализ данных о задержках, собранных из различных источников. Важно учесть множество факторов, включая время задержки, причины, аэропорты, маршруты и другие сопутствующие параметры. В первую очередь, требуется осуществить обзор и сбор доступных данных о задержках авиарейсов. Эти данные могут быть получены из различных источников, включая базы данных авиакомпаний, государственные организации, аэропорты и другие. Важно собрать достаточное количество данных, чтобы охватить разнообразные ситуации и обеспечить репрезентативность анализа. После сбора данных необходимо оценить их качество и достоверность. Возможно, в данных могут быть пропуски, ошибки или неточности, которые могут исказить результаты анализа. Поэтому требуется провести предварительную обработку данных, включающую удаление дубликатов, заполнение пропущенных значений и исправление ошибок. Также важно проверить соответствие формата данных и их совместимость для дальнейшего использования. После этапа предварительной обработки, следует провести детальный анализ данных о задержках авиарейсов. Это включает исследование распределения задержек, выявление частоты и продолжительности задержек, анализ зависимостей между задержками и другими параметрами, такими как время суток, день недели, погодные условия и типы авиакомпаний. Важно определить ключевые факторы, влияющие на задержки, чтобы использовать их в дальнейшей разработке модели предсказания. Для более глубокого анализа данных о задержках авиарейсов можно применить статистические методы, такие как корреляционный анализ, регрессионный анализ и машинное обучение. Эти методы помогут выявить скрытые

связи и зависимости между факторами и задержками, что позволит лучше понять причины задержек и их предсказуемость. Все эти шаги анализа данных о задержках авиарейсов позволят получить полное представление о состоянии проблемы и определить наиболее важные факторы, влияющие на задержки. Это будет полезной основой для разработки предсказательной модели, которая сможет учесть эти факторы и предсказывать возможные задержки авиарейсов с высокой точностью и достаточным временным прогнозом.

В рамках данной задачи будет проведен анализ данных о задержках авиарейсов с целью выявления ключевых факторов, влияющих на возникновение задержек. Это важный шаг для разработки эффективной предсказательной модели, которая должна учитывать различные внешние и внутренние факторы, оказывающие влияние на операции воздушного транспорта. В первую очередь, необходимо собрать обширные наборы данных о задержках авиарейсов, включая информацию о времени задержки, аэропортах, маршрутах, авиакомпаниях, типах самолетов и других релевантных параметрах. Такие данные можно получить из различных источников, включая аэропорты, авиакомпании и государственные организации, которые собирают статистику о задержках воздушного движения. Полученные данные должны быть предварительно обработаны и очищены от выбросов, ошибок и пропущенных значений. После этого можно приступить к статистическому анализу, с целью выявления статистически значимых зависимостей между факторами и задержками авиарейсов. Одним из ключевых факторов, влияющих на задержки, являются погодные условия. Аномальная погода, такая как сильные ветры, грозы, туман или снегопады, может значительно повлиять на возможность выполнения авиарейсов вовремя. Поэтому необходимо анализировать метеорологические данные, включая данные о температуре, осадках, видимости и скорости ветра, с целью определения влияния погодных условий на задержки. Кроме того, время суток также может оказывать влияние на задержки. Например, утренние часы могут

быть более склонны к задержкам из-за повышенного трафика воздушных судов, а вечерние часы могут быть подвержены задержкам из-за накопления задержек в течение дня. Поэтому необходимо анализировать временные параметры, включая время суток, день недели и сезонность, для выявления возможных зависимостей с задержками. Другим важным фактором является тип авиакомпании. Различные авиакомпании могут иметь разные стратегии управления задержками и ресурсами. Некоторые авиакомпании могут быть более надежными и эффективными в управлении задержками, чем другие. Поэтому необходимо провести анализ данных о задержках в зависимости от авиакомпании, чтобы выявить возможные различия и влияние типа авиакомпании на задержки. Кроме того, другие факторы, такие как загруженность аэропорта, наличие запланированных технических работ или задержки в предыдущих рейсах, также могут влиять на возникновение задержек. Проведение анализа данных о таких факторах поможет выявить их влияние и включить их в предсказательную модель. Для выявления зависимостей и связей между факторами и задержками можно использовать различные методы статистического анализа, включая корреляционный анализ, регрессионный анализ и множественный анализ. Эти методы позволяют определить степень влияния каждого фактора на задержки и выявить взаимосвязи между ними. В итоге, анализ данных о задержках авиарейсов и выявление влияющих факторов позволит более точно определить, какие параметры следует учесть при разработке предсказательной модели. Это поможет создать более точную и надежную модель, которая учтет различные внешние и внутренние факторы, влияющие на задержки авиарейсов.

Разработка предсказательной модели для задержек авиарейсов является одной из ключевых задач данного исследования. Этот этап требует глубокого анализа данных, выбора подходящих методов машинного обучения и анализа данных, а также обучения модели на доступных данных. В первую очередь,

необходимо выбрать подходящие методы машинного обучения, которые будут способны обрабатывать и анализировать данные о задержках авиарейсов. Среди популярных методов можно выделить алгоритмы регрессии, ансамблевые модели, нейронные сети и другие. Каждый из этих методов имеет свои преимущества и ограничения, поэтому выбор подходящего метода должен быть обоснован и основан на особенностях рассматриваемых данных. Далее, после выбора метода машинного обучения, следует провести предварительную обработку данных. Это включает в себя очистку данных от выбросов, заполнение пропущенных значений, масштабирование признаков и преобразование категориальных переменных в числовой формат. Чистые и хорошо подготовленные данные обеспечат более точное обучение модели и улучшат ее способность предсказывать задержки авиарейсов. После предварительной обработки данных необходимо провести обучение модели на подготовленных данных. Обучение модели включает в себя подгонку параметров модели с использованием оптимизационных алгоритмов, таких как стохастический градиентный спуск или метод наименьших квадратов. Оценка качества модели производится с помощью различных метрик, таких как средняя абсолютная ошибка (MAE), средняя квадратичная ошибка (MSE) и коэффициент детерминации (R^2). Эти метрики позволяют оценить точность и достоверность предсказаний модели. Важным аспектом разработки предсказательной модели является выбор оптимальных гиперпараметров модели. Гиперпараметры включают в себя параметры модели, которые не могут быть обучены в процессе обучения, такие как количество скрытых слоев в нейронной сети или глубина деревьев в ансамблевых моделях. Оптимальный выбор гиперпараметров позволяет достичь лучшей производительности модели и улучшить ее предсказательную способность. После завершения обучения модели необходимо провести ее оценку на тестовых данных. Тестирование модели позволяет проверить ее способность предсказывать задержки авиарейсов на данных, которые не использовались при обучении модели. Это важный шаг, который позволяет оценить обобщающую способность модели и ее способность работать с новыми

наборами данных. В конечном итоге, разработанная предсказательная модель должна быть способна предсказывать задержки авиарейсов с высокой точностью и достаточным временным прогнозом. Это позволит авиакомпаниям и пассажирам получать заранее информацию о возможных задержках и принимать соответствующие меры, чтобы снизить негативные последствия. Разработка такой модели будет иметь значительное практическое применение и способствовать улучшению операционной эффективности авиакомпаний и комфорта пассажиров. На рисунке 1 изображена схема оценки качества модели машинного обучения.

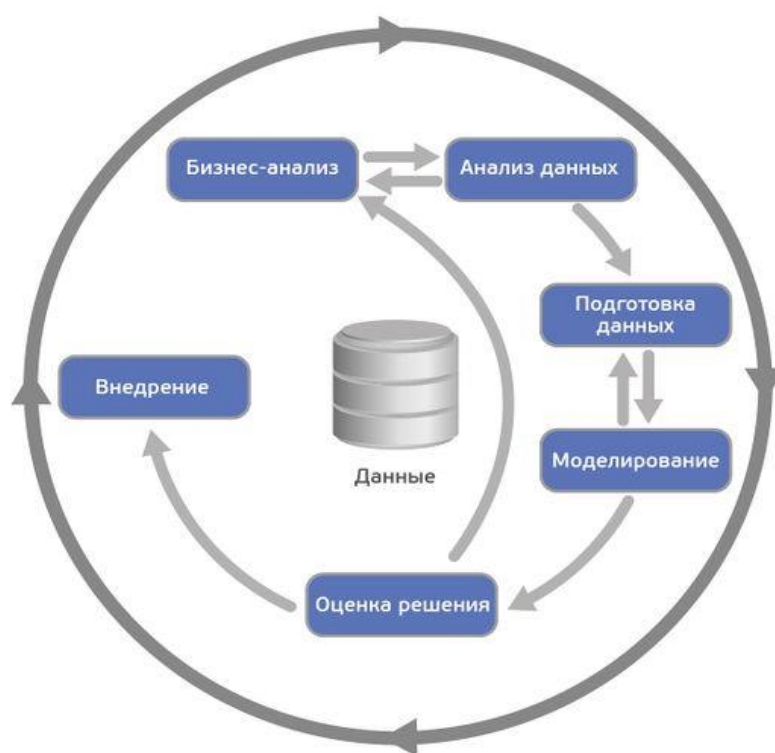


Рисунок 1 — оценка качества модели машинного обучения

В этом разделе мы рассмотрим процесс внедрения разработанной предсказательной модели задержек авиарейсов в практическую среду и проведение оценки ее эффективности. Основной целью внедрения модели является обеспечение реального временного прогнозирования задержек, а также интеграция модели с системами авиакомпаний или аэропортов для практического применения. Первым этапом внедрения модели является реализация разработанного

алгоритма предсказания задержек авиарейсов в виде программного решения. Для этого необходимо выбрать подходящие программные инструменты и языки программирования, которые позволят эффективно реализовать модель. Важным аспектом является также обеспечение возможности интеграции модели с существующими системами авиакомпаний или аэропортов, чтобы она могла получать актуальные данные и предоставлять прогнозы задержек в режиме реального времени. После реализации модели необходимо провести тестирование и верификацию ее работы. Это позволит убедиться в правильности функционирования модели и соответствии ее предсказаний фактическим задержкам. Для этого можно использовать исторические данные о задержках авиарейсов и сравнить предсказанные значения с реальными. Также следует оценить время, затрачиваемое на получение прогнозов задержек, и их актуальность, чтобы убедиться в приемлемой производительности и релевантности модели. После успешного тестирования и верификации модели можно перейти к ее внедрению в практическую среду. Это может включать интеграцию модели с системами авиакомпаний или аэропортов, настройку взаимодействия с базами данных и другими информационными системами. Важно обеспечить стабильную работу модели и ее регулярное обновление с использованием новых данных о задержках авиарейсов. Оценка эффективности модели является важным этапом внедрения. Необходимо провести анализ точности и достоверности предсказаний модели с использованием соответствующих метрик оценки. Сравнение прогнозов с фактическими задержками позволит оценить погрешность модели и определить ее способность предсказывать задержки с высокой точностью. Также следует оценить практическую пользу модели, основываясь на обратной связи от пользователей, таких как авиакомпании, аэропорты и пассажиры. Их мнение и впечатления помогут определить степень удовлетворенности и полезности модели в повседневной деятельности. Важным аспектом оценки модели является сравнение экономических показателей до и после внедрения модели предсказания задержек. Необходимо проанали-

зировать снижение экономических потерь авиакомпаний, связанных с задержками, а также повышение операционной эффективности и пассажирского комфорта. Также следует учесть экологические аспекты и оценить снижение негативного воздействия авиации на окружающую среду благодаря оптимизации движения самолетов и предотвращению неэффективных полетов. В итоге, внедрение и оценка модели предсказания задержек авиарейсов является важным шагом в достижении поставленной цели и решении проблемы задержек в авиационной индустрии. Это позволит улучшить операционную эффективность авиакомпаний, повысить удовлетворенность пассажиров, снизить экономические потери и негативное воздействие на окружающую среду.

После завершения разработки и реализации предсказательной модели для задержек авиарейсов, необходимо провести анализ полученных результатов и сделать соответствующие выводы. Этот этап исключительно важен, поскольку позволит оценить эффективность модели и ее воздействие на различные стороны авиационной индустрии. В первую очередь, необходимо проанализировать полученные предсказания и сравнить их с фактическими данными о задержках авиарейсов. Для этого будет использована соответствующая методология оценки точности модели, такая как сравнение прогнозов с фактическими значениями, вычисление средней абсолютной ошибки, корреляционного коэффициента и других метрик. Такой анализ позволит оценить, насколько близки прогнозы модели к реальным значениям и насколько точно модель способна предсказывать задержки авиарейсов. Далее следует проанализировать полученные результаты с учетом основных целей и задач исследования. Оценка влияния модели на снижение задержек авиарейсов является ключевым аспектом этого анализа. Сравнивая статистику задержек до и после внедрения модели, можно оценить процентное снижение задержек и прямые экономические выгоды для авиакомпаний. Кроме того, можно проанализировать, насколько часто модель предупреждает о задержках заранее, что позво-

ляет принимать соответствующие меры и минимизировать негативное влияние на пассажиров и операционные процессы авиакомпаний. Помимо этого, анализ результатов включает оценку влияния модели на пассажирский опыт. Можно изучить отзывы пассажиров после внедрения предсказательной модели и провести опросы, чтобы собрать данные о их восприятии качества предоставляемых услуг. Выяснить, насколько предупреждение о задержках заранее влияет на их комфорт и удовлетворенность. При положительных результатах, это может привести к укреплению репутации авиакомпании и увеличению лояльности пассажиров. Важной составляющей анализа результатов является оценка операционной эффективности авиакомпаний после внедрения модели. Модель предсказания задержек авиарейсов может помочь авиакомпаниям принимать более информированные решения, связанные с перераспределением ресурсов, планированием рейсов и управлением рейсовыми задержками. Анализ эффективности модели должен включать такие аспекты, как сокращение времени ожидания пассажиров, повышение использования ресурсов авиакомпании и оптимизация планирования рейсов. После проведения подробного анализа результатов и оценки их эффективности, можно сделать обобщенные выводы о достигнутых результатах. Оценка точности и надежности модели, снижение задержек авиарейсов, улучшение пассажирского опыта и операционной эффективности авиакомпаний - все эти факторы влияют на общий успех проекта. Важно подчеркнуть, насколько разработанная модель способна улучшить процессы авиационной индустрии и принести пользу различным заинтересованным сторонам. Кроме того, на основе анализа результатов, можно предложить рекомендации для дальнейшего совершенствования модели и ее применения в авиационной индустрии. Это может включать улучшение алгоритмов прогнозирования, учет дополнительных факторов, внедрение новых технологий и техник анализа данных, а также интеграцию модели с другими системами авиакомпаний и аэропортов. В заключение, анализ результатов и выводы играют важную роль в исследовании предсказания задер-

жек авиарейсов. Они позволяют оценить эффективность разработанной модели, проследить ее влияние на авиационную индустрию и выявить потенциальные возможности для ее улучшения. Результаты данного анализа будут иметь практическую ценность для авиакомпаний, пассажиров и окружающей среды, помогая снизить задержки, повысить эффективность и обеспечить более комфортный опыт путешествия для всех заинтересованных сторон.

ВЫВОДЫ ПО РАЗДЕЛУ 1

В ходе изучения предметной области было выявлено, что задержки авиарейсов являются серьезной проблемой, которая оказывает отрицательное влияние на пассажиров, авиакомпании и окружающую среду. Задержки могут быть вызваны различными факторами, такими как погодные условия, технические неисправности, организационные проблемы и другие непредвиденные обстоятельства. Эти задержки приводят к неудовлетворенности пассажиров, финансовым потерям для авиакомпаний и негативному экологическому воздействию из-за дополнительного потребления топлива.

Цель данного исследования заключается в разработке предсказательной модели для задержек авиарейсов, которая позволит предупреждать пассажиров о возможных задержках заранее и принимать соответствующие меры для снижения негативных последствий. Основная цель - создание эффективного инструмента, который обладает высокой точностью и достаточным временным прогнозом задержек.

Для достижения этой цели были поставлены определенные задачи. Анализ данных о задержках авиарейсов позволил изучить доступные данные и провести их предварительную обработку для дальнейшего анализа. Идентификация влияющих факторов позволила выявить ключевые факторы, которые оказывают влияние на задержки авиарейсов. Разработка предсказательной мо-

дели включала выбор и применение подходящих методов машинного обучения и анализа данных. Внедрение и оценка модели предполагает реализацию модели в практической среде и оценку ее эффективности.

Важно отметить, что решение, представленное в рамках данного исследования, должно соответствовать определенным критериям. Во-первых, оно должно обладать высокой точностью и достоверностью предсказаний задержек авиарейсов. Это позволит пассажирам и авиакомпаниям принимать своевременные меры для минимизации отрицательных последствий. Во-вторых, решение должно быть масштабируемым и применимым в реальных условиях авиационной индустрии. Оно должно учитывать разнообразные факторы, такие как типы авиарейсов, аэропорты, географические особенности и другие контекстные параметры. В-третьих, решение должно быть интегрируемым с существующими системами авиакомпаний и обладать понятным и удобным интерфейсом для пользователей.

Таким образом, решение проблемы задержек авиарейсов требует разработки эффективной предсказательной модели, которая будет соответствовать высоким требованиям точности, масштабируемости и интегрируемости. Предложенное решение должно способствовать снижению задержек авиарейсов, повышению пассажирского опыта и операционной эффективности авиакомпаний.

2 РАЗРАБОТКА ПРЕДСКАЗАТЕЛЬНОЙ МОДЕЛИ ЗАДЕРЖЕК АВИАРЕЙСОВ. ПОДХОД И РЕАЛИЗАЦИЯ

2.1 Теоретическое решение задачи предсказания задержек авиарейсов: подход и методология

2.1.1 Анализ факторов, влияющих на задержки авиарейсов

В настоящее время существует множество факторов, оказывающих влияние на время ожидания авиарейсов. Проанализируем воздействия каждого из них по-отдельности и выявим наиболее значимые. Данная информация поможет в разработке предсказательной модели.

Одним из наиболее значимых условий, влияющих на задержки авиарейсов, являются погодные условия. Аномальные природные явления, такие как сильные ветры, грозы, туманы и снегопады, могут привести к ограничениям в воздушном пространстве и аэропортах. Данные затруднения в организации перелётов приводят к возникновению разного рода задержек авиарейсов. Например, боковой ветер может запретить взлет или посадку одного полета, в то время как даже значительно более сильный лобовой ветер может не повлиять на другой. Более того, метеорологические условия очень чувствительны и могут значительно изменяться даже в течение одного дня. В течение нескольких часов хорошая погода может обернуться препятствием.

Технические неисправности самолетов и оборудования также оказывают большое влияние на расписание перелётов. Непредвиденные поломки, необходимость в ремонте и замене деталей могут вызвать задержки в аэропортах.

Организационные проблемы, связанные с авиационными операциями и управлением рейсами, могут также приводить к различным заминкам в графиках перелётов. Неправильное планирование, недостаточное количество персонала, несоответствие между расписанием и доступностью слотов на аэропортах — всё это может вызывать задержки и нарушения в аэропортах.

Ограничения в воздушном пространстве и на аэропортах также могут стать причиной задержек авиарейсов. К ним относятся перегруженность аэропортов, ограничения в слотах, задержки в контроле воздушного движения. Зоны ограничения полётов устанавливаются каждым государством самостоятельно. В зависимости от различных обстоятельств оно имеет право на закрытие воздушного пространства. В данном случае авиаперевозчики должны перестраивать свой маршрут в зависимости от открытых воздушных коридоров, что вызывает накладки в перелётах с другими авиакомпаниями.

Пассажиры также могут стать причиной задержек авиарейсов. Задержки при регистрации и выдаче багажа, его утрата, проблемы с посадкой и выходом из самолета — всё это может повлиять на операционную эффективность авиакомпаний. Любое действие, которое выбивается из намеченного графика, влияет на работу всего аэропорта и перелёты других рейсов.

Существует множество и других факторов, влияющих на изменение расписания перелётов. В данной работе при построении модели будут учитываться основные 5 факторов, рассмотренные выше.

2.1.2 Ознакомление с существующими методами и моделями предсказания задержек авиарейсов

Сначала необходимо ознакомиться с существующими на данный момент моделями для предсказания задержек и собрать информацию для построения собственной. В настоящее время существует множество различных методов и моделей, которые сообщают пассажирам о различных проблемах с перелётом. Например, сервис Google flights с помощью методов машинного обучения в режиме реального времени на основе данных из различных источников сообщает пользователям о задержке авиарейса с описанием причин. Всё это происходит ещё до официального объявления от авиакомпании. Создадим собственную модель на основе искусственной нейронной сети. Рассмотрим основные характеристики данных и их предварительную обработку.

Так как любые информации о перелётах являются закрытыми данными, то основными источниками данных при построении модели являются информация, полученная от авиакомпаний, аэропортов, государственных органов и других релевантных источников. В них указаны основные характеристики авиарейсов, такие как типы задержек, временные метки, аэропорты отправления и прибытия, а также другая доступная информация. Пример набора данных представлен на рисунке 2.

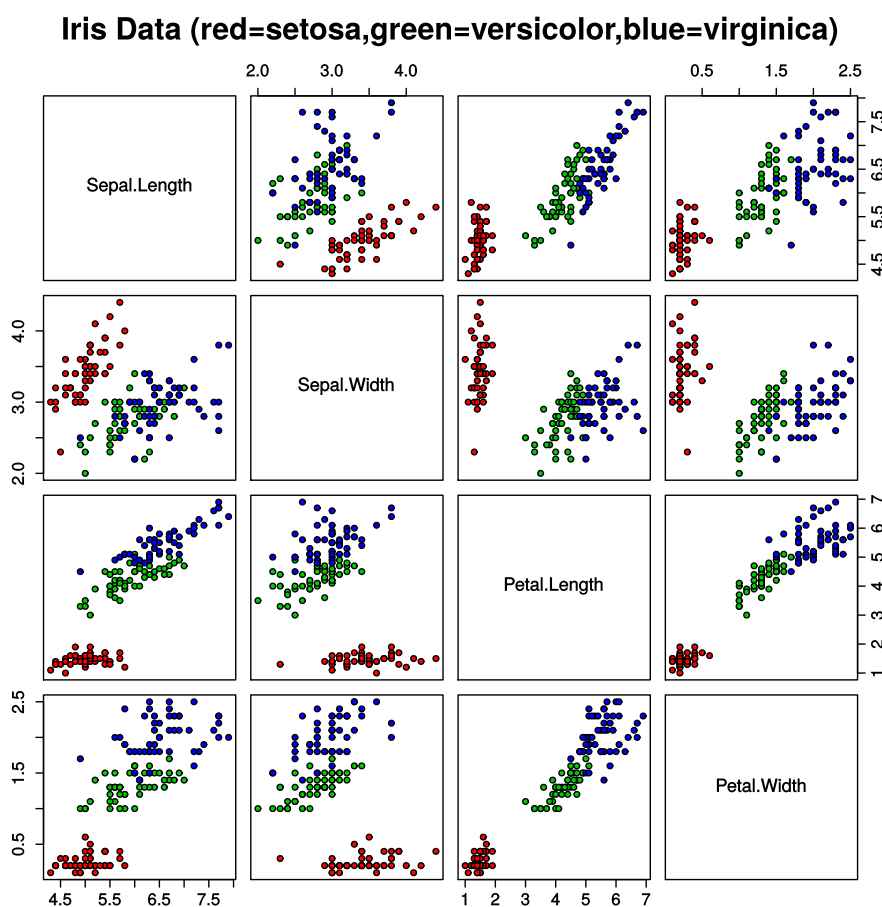


Рисунок 2 — Набор данных

Прежде чем загрузить данные для обучения модели необходимо их предварительно обработать. Очистка включает в себя удаление выбросов, заполнение пропущенных значений и исправление ошибок. В настоящее время существует множество способов для обработки предоставленных данных. Это может быть применение статистических методов, алгоритмов заполнения пропущенных значений и других техник обработки.

Далее необходимо преобразовать полученные и очищенные данные о задержках авиарейсов в удобный для анализа формат. Это может включать изменение их структуры, преобразование категориальных переменных в числовой формат, масштабирование данных и другие трансформации.

Извлечение признаков включает в себя выбор и создание релевантных характеристик, которые могут быть использованы для предсказания задержек. Необходимыми являются дата и время, погодные условия, типы авиарейсов, их длительность и другие факторы, которые могут оказывать влияние на задержки.

Последним пунктом обучения является разделение данных на обучающую и тестовую выборки. Обучающая выборка используется для обучения модели предсказания задержек, а тестовая выборка - для оценки ее производительности и точности. Существуют различные методы для разделения данных, такие как случайное разбиение, временные разделители и кросс-валидация. Считается наиболее оптимальным разбиение данных по принципу 30-70%, где наибольшая часть идёт на тестирование полученной модели.

2.1.3 Описание и характеристики используемых данных о задержках авиарейсов

Рассмотрим теоретическое решение задачи предсказания задержек авиарейсов. Для того чтобы разработать эффективную предсказательную модель, необходимо ознакомиться с основными принципами и методами, которые применяются в данной области.

Первым шагом является анализ факторов, влияющих на расписание авиарейсов. Это позволит нам понять, какие переменные и условия оказывают наибольшее влияние на вероятность задержки. Факторы могут включать погодные условия, технические проблемы с самолетом, организационные задержки и другие непредвиденные обстоятельства. Анализ полученных данных позволит определить, какие из них следует включить в модель предсказания.

Далее проводится ознакомление с существующими методами и моделями предсказания задержек авиарейсов. Исследование литературы и академических работ позволяет узнать о различных подходах, которые применялись в предыдущих исследованиях. Это может включать использование статистических моделей, машинного обучения, нейронных сетей и других методов. Изучение существующих моделей дает нам представление о их преимуществах и недостатках, а также помогает определить оптимальный подход к разработке нашей модели.

После проведенного литературного исследования в разработке предсказательной модели необходимо определить данные, необходимые для обучения и тестирования модели. Это может включать исторические данные о задержках авиарейсов, информацию о погоде, характеристики самолетов, маршруты полетов и другие соответствующие параметры. Для того чтобы модель была точной и надежной, необходимо провести предварительную обработку данных. Это может включать удаление выбросов, заполнение пропущенных значений, нормализацию данных и другие методы очистки и преобразования.

После определения данных мы переходим к выбору подхода к моделированию. В данном шаге мы изучаем различные методы машинного обучения, которые могут быть применены для предсказания задержек авиарейсов. Это может включать линейную регрессию, решающие деревья и случайный лес, градиентный бустинг, нейронные сети и другие. Каждый метод имеет свои особенности и предположения, и выбор оптимального подхода зависит от характеристик данных и поставленных целей.

Далее переходим к разработке предсказательной модели. На этом этапе мы описываем выбранную модель и ее основные характеристики. Включаем в модель выбранные факторы, которые были выявлены в анализе предметной области. Затем производим обучение модели на обучающем наборе данных. Оцениваем и подбираем гиперпараметры модели, чтобы достичь оптимальной производительности и точности предсказаний. Завершаем этот шаг, проверяя

и валидируя модель на тестовом наборе данных, чтобы оценить ее способность предсказывать задержки авиарейсов.

Следующий шаг - интеграция разработанной модели в практическую среду авиационной индустрии. Мы описываем процесс интеграции модели, включая взаимодействие с существующими системами авиакомпаний и аэропортов. Создаем прототип системы предсказания задержек авиарейсов, который может быть использован для тестирования и демонстрации модели. Проверяем работоспособность прототипа, обеспечивая правильную интеграцию и функционирование модели в реальных условиях.

Для обеспечения более эффективного использования предсказательной модели разрабатываем методологию предупреждения о задержках авиарейсов. Это включает определение пороговых значений и критериев для предупреждения о возможных задержках. Создаем систему уведомлений и рекомендаций для пассажиров и авиакомпаний, которая поможет им принимать информированные решения и принимать меры для снижения возможных негативных последствий задержек.

Заключительным этапом является обзор общей архитектуры разработанного решения. Описываются основные компоненты системы и их взаимодействие. Разъясняются особенности решения и его потенциал для дальнейшего развития. Обсуждаются возможности оптимизации и улучшения модели, а также ее применимость в других областях авиационной индустрии. В данном подразделе мы описали теоретическое решение задачи предсказания задержек авиарейсов. Были рассмотрены основные шаги, включающие изучение предметной области, предварительную обработку данных, выбор подхода к моделированию, разработку предсказательной модели, интеграцию и создание прототипа, разработку методологии предупреждения о задержках и обзор архитектуры разработанного решения. Эти шаги являются основой для дальнейшей работы над разработкой предсказательной модели задержек авиарейсов.

2.1.4 Очистка данных от выбросов, пропущенных значений и ошибок

Для успешной разработки предсказательной модели задержек авиарейсов необходимо реализовать процесс определения данных и их предварительной обработки. Представленная ниже последовательность действий позволит получить надежные и качественные данные для дальнейшего анализа и построения модели.

Первым шагом является определение и описание данных, которые будут использоваться в разработке модели. В нашем случае это данные о задержках авиарейсов, которые включают в себя информацию о времени и типе опоздания, номере рейса, аэропорте отправления и прибытия, и другие релевантные параметры. Также учитывается, что информация о перелётах могут быть предоставлены в различных форматах, например, в виде структурированных таблиц или неструктурированных текстовых файлов.

Вторым шагом является очистка данных от возможных выбросов, пропущенных значений и ошибок. Для обеспечения качественного анализа и построения модели необходимо удалить или исправить любые неточности в предоставленной информации о задержках. Это может включать проверку на наличие дубликатов, удаление выбросов, заполнение пропущенных значений или исключение неправильных записей. Для эффективной очистки данных можно использовать методы статистического анализа и алгоритмы машинного обучения. После очистки данных необходимо преобразовать их в удобный для анализа формат. Возможные преобразования могут включать нормализацию значений, преобразование категориальных переменных в числовые признаки, агрегацию данных по временным интервалам или другие трансформации. Целью данного шага является подготовка данных для дальнейшего анализа и построения модели. Важным аспектом предварительной обработки данных является проверка и оценка их качества. Это включает анализ статистических характеристик данных, проверку на наличие аномалий или необычных паттернов, а также оценку соответствия данных поставленным требованиям и целям

разработки модели. В случае обнаружения проблем или несоответствий, необходимо провести дополнительные действия, такие как повторную очистку данных или обновление источников данных.

Последним шагом в предварительной обработке данных является создание обучающего и тестового наборов данных. Обучающий набор будет использоваться для обучения предсказательной модели, а тестовый – для оценки ее производительности и обобщающей способности. Разделение данных на обучающий и тестовый наборы позволяет проверить, насколько хорошо модель обобщает знания из обучающего набора на новые данные. Таким образом, предварительная обработка данных включает определение и описание данных, очистку от выбросов и ошибок, преобразование в удобный для анализа формат, проверку качества данных и создание обучающего и тестового наборов.

При соблюдении данной последовательности действий получаем структурированный набор данных, необходимых для дальнейшей разработки предсказательной модели задержек авиарейсов.

2.1.5 Выбор подхода к моделированию

Проанализируем и сравним различные методы машинного обучения и выберем наиболее подходящий для нашей задачи, который обеспечит высокую точность и предсказательную способность модели. Разные методы машинного обучения имеют свои особенности, преимущества и недостатки.

Один из наиболее распространенных подходов к предсказанию задержек авиарейсов — это линейная регрессия. Линейная регрессия используется для моделирования линейной зависимости между зависимой переменной (в данном случае, задержками авиарейсов) и независимыми переменными (факторами, влияющими на задержки). Она предполагает, что связь между переменными может быть описана линейной функцией. Линейная регрессия может быть полезной, если у нас есть явные линейные зависимости между факторами

и задержками авиарейсов. Однако линейная регрессия имеет свои ограничения. Она не учитывает сложные нелинейные взаимодействия между переменными и может не справляться с моделированием сложных зависимостей. Пример линейной регрессии представлен на рисунке 3.

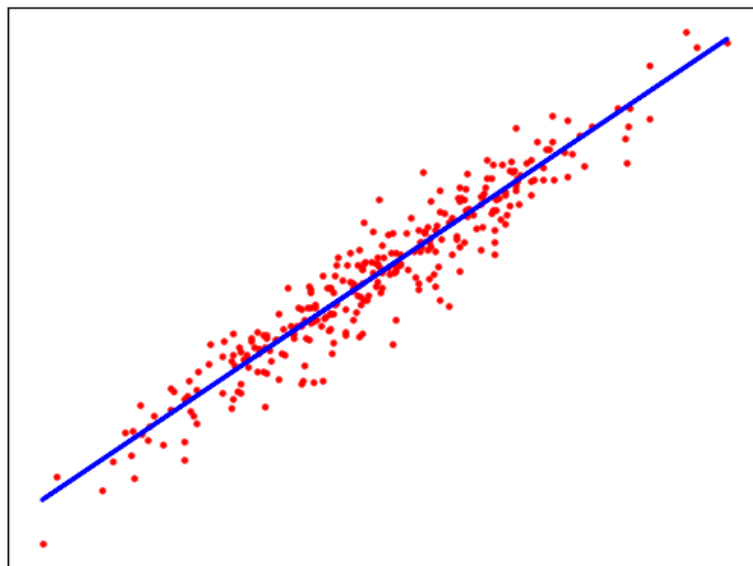


Рисунок 3 — Линейная регрессия

Также рассмотрим методы на основе решающих деревьев и случайного леса. Решающие деревья, изображённые на рисунке 4, представляют собой древовидную структуру, в которой каждый узел представляет тест на значение определенного признака, а каждая ветвь соответствует возможному значению этого признака. Решающие деревья основываются на принципе деления данных на наиболее однородные группы. Они могут улавливать сложные зависимости между переменными и предсказывать нелинейные взаимодействия.

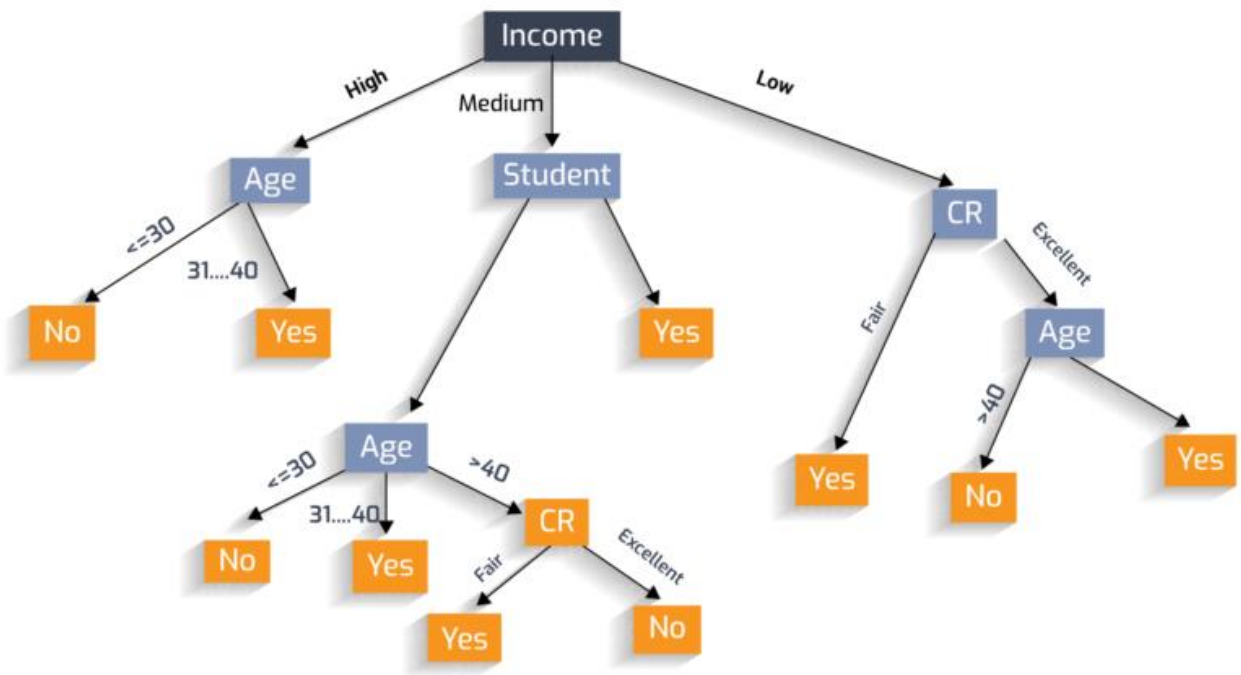


Рисунок 4 — Решающее дерево

Случайный лес — это ансамбль решающих деревьев, в котором каждое дерево строится на основе случайной подвыборки данных и случайного подмножества признаков. Комбинирование множества деревьев позволяет уменьшить эффект переобучения и повысить обобщающую способность модели. Случайный лес может быть эффективным методом для предсказания задержек авиарейсов, особенно при наличии множества факторов, влияющих на задержки. Пример случайного леса изображён на рисунке 5.

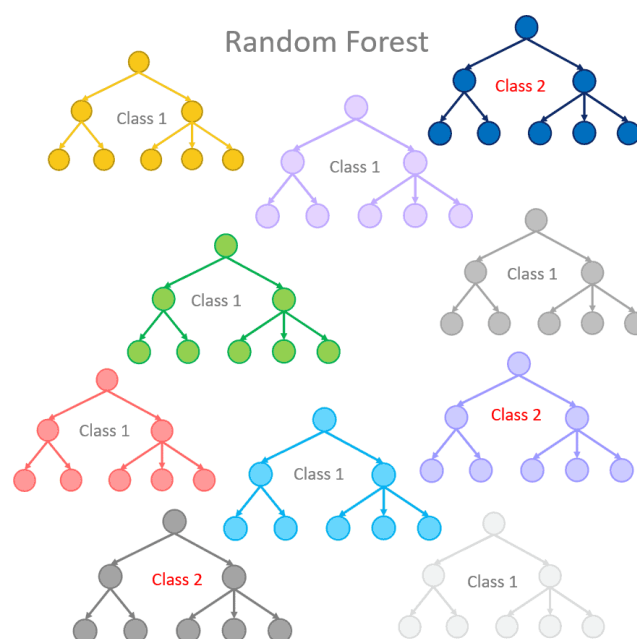


Рисунок 5 — Случайный лес

Еще одним подходом является градиентный бустинг. Градиентный бустинг — это метод построения ансамбля моделей, в котором каждая следующая модель исправляет ошибки предыдущей модели. Он основан на идее обучения модели последовательно, при этом каждая новая модель фокусируется на ошибках предыдущей. Градиентный бустинг, представленный на рисунке 12, может достигать высокой точности предсказания и обладает хорошей способностью к обобщению.

Мощным инструментом машинного обучения, который может моделировать сложные зависимости и нелинейные взаимодействия, являются искусственные нейронные сети. Они состоят из множества связанных нейронов, которые работают вместе для обработки информации. Они могут быть эффективными для предсказания задержек авиарейсов, особенно при наличии большого объема данных и сложных зависимостей между факторами и задержками. Пример нейронной сети представлен на рисунке 6.

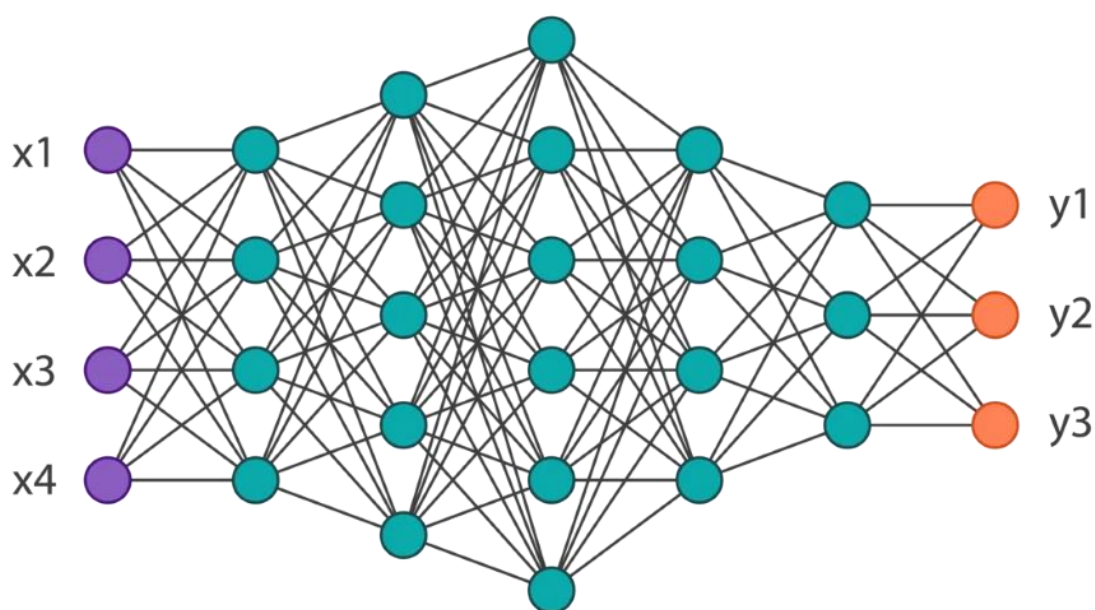


Рисунок 6 — Нейронная сеть

При выборе подхода к моделированию учитываются характеристики представленных данных, сложность задачи, требования к точности предсказания и доступные вычислительные ресурсы. Комбинирование различных методов и алгоритмов позволяет достичь лучшей производительности модели.

2.1.6 Описание выбранной модели и её основных характеристик

В данном разделе мы представим подробное описание выбранной модели для предсказания задержек авиарейсов и рассмотрим ее основные характеристики. Данная модель была выбрана на основе предварительного анализа различных методов машинного обучения и оценки их пригодности для решения поставленной задачи.

Выбор подхода к моделированию задержек авиарейсов является важным шагом, поскольку от этого зависит точность и эффективность предсказания. Были исследованы различные методы, такие как линейная регрессия, решающие деревья и случайный лес, градиентный бустинг и нейронные сети. После

анализа и сравнения результатов, мы выбрали нейронные сети как наиболее подходящий метод для решения данной задачи.

Основные характеристики выбранной модели:

1. архитектура нейронной сети: мы решили использовать глубокую нейронную сеть с несколькими скрытыми слоями. Такая архитектура позволяет модели извлекать сложные зависимости и учитывать взаимодействие различных факторов, влияющих на задержки авиарейсов;

2. входные данные: в качестве входных данных модели мы использовали разнообразные факторы, включающие время суток, день недели, погодные условия, информацию о самолете и аэропорту, и другие факторы, которые были идентифицированы как важные при анализе данных о задержках авиарейсов;

3. препроцессинг данных: перед подачей данных на вход модели, мы проводили их предварительную обработку. Это включало масштабирование признаков, кодирование категориальных переменных, обработку пропущенных значений и выбросов, а также разделение данных на обучающую, валидационную и тестовую выборки;

4. обучение и оптимизация модели: для обучения модели мы использовали алгоритм обратного распространения ошибки и стохастический градиентный спуск. Мы также применяли различные методы оптимизации, такие как адам, для эффективной настройки весов нейронной сети;

5. оценка и метрики: для оценки качества модели мы использовали различные метрики, включая среднеквадратичную ошибку (Mean Squared Error, MSE), коэффициент детерминации (R-squared) и среднюю абсолютную ошибку (Mean Absolute Error, MAE). Эти метрики позволяют нам оценить точность и достоверность предсказаний модели;

Выбор нейронных сетей в качестве модели предсказания задержек авиарейсов обусловлен их способностью обрабатывать сложные взаимодействия между различными факторами и их высокой гибкостью в адаптации к различным типам данных. Однако, разработка и обучение ИНС требует достаточного

объема данных и вычислительных ресурсов. После определения выбранной модели и ее основных характеристик, мы приступили к разработке и реализации этой модели в рамках нашего решения предсказания задержек авиарейсов. В следующих разделах мы рассмотрим процесс обучения модели, ее оценку и валидацию, а также интеграцию модели в практическую среду авиационной индустрии.

2.1.7. Обучение модели на обучающем наборе данных

Обучение модели является одним из ключевых этапов в разработке решения предсказания задержек авиарейсов. Перед началом обучения модели необходимо подготовить обучающий набор данных. Этот набор данных должен содержать информацию о задержках авиарейсов, а также соответствующие признаки, которые могут оказывать влияние на эти задержки. Важно провести предварительную обработку данных, устранить выбросы и пропущенные значения, а также привести данные к единообразному формату.

Далее следует выбрать подходящую модель для обучения. В контексте предсказания задержек авиарейсов можно использовать различные методы машинного обучения, такие как линейная регрессия, решающие деревья и случайный лес, градиентный бустинг или нейронные сети. Выбор конкретной модели зависит от требуемой точности предсказаний, доступных данных и других факторов.

После выбора модели необходимо разделить обучающий набор данных на обучающую и валидационную выборки. Обучающая выборка будет использоваться для обучения модели, а валидационная выборка — для проверки ее эффективности и настройки гиперпараметров. Обычно используется подход кросс-валидации, при котором обучающая выборка разбивается на несколько подвыборок, и модель обучается и тестируется на различных комбинациях этих подвыборок. Далее следует сам процесс обучения модели. В ходе обучения модель "изучает" обучающий набор данных и старается выявить зависи-

мости между признаками и задержками авиарейсов. В процессе обучения модель оптимизирует выбранный функционал ошибки (например, среднеквадратичное отклонение) с помощью методов оптимизации, таких как градиентный спуск или стохастический градиентный спуск. При этом модель настраивает веса и параметры своих компонентов, чтобы минимизировать ошибку предсказания.

После завершения процесса обучения модели необходимо оценить ее качество на валидационной выборке. Это позволяет оценить, насколько хорошо модель обобщает знания на новых данных. Результаты оценки качества модели могут быть представлены в виде различных метрик, таких как средняя абсолютная ошибка, коэффициент детерминации (R-квадрат) и другие. При необходимости можно провести настройку гиперпараметров модели. Гиперпараметры — это параметры, которые не оптимизируются в процессе обучения модели, но влияют на ее структуру и поведение. Некоторые примеры гиперпараметров модели могут включать количество скрытых слоев нейронной сети, глубину решающего дерева или скорость обучения. Настройка гиперпараметров, изображённая на рисунке 7, может проводиться с использованием методов кросс-валидации или других техник оптимизации.

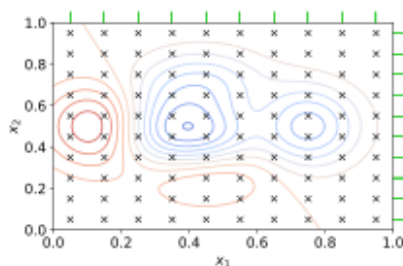


Рисунок 7 — Настройка гиперпараметров

Таким образом, процесс обучения модели на обучающем наборе данных включает выбор модели, разделение данных, обучение модели, оценку качества и при необходимости настройку гиперпараметров. Обученная модель становится основой для предсказания задержек авиарейсов и интеграции в разработанное решение.

2.1.8 Оценка и подбор гиперпараметров модели

После обучения модели на обучающем наборе данных необходимо проверить ее производительность и подобрать оптимальные гиперпараметры, чтобы достичь наилучшей точности предсказания задержек авиарейсов.

В начале процесса проверки модели мы оценим ее производительность на валидационном наборе данных. Валидационный набор данных представляет собой отдельный поднабор данных, который не использовался при обучении модели. Мы прогоним валидационные данные через модель и оценим точность предсказаний, используя соответствующие метрики оценки, такие как средняя абсолютная ошибка (MAE) или средняя квадратичная ошибка (MSE). После оценки производительности модели мы приступим к подбору гиперпараметров.

Гиперпараметры — это параметры модели, которые не оптимизируются в процессе обучения, а выбираются и задаются вручную. Они могут включать количество скрытых слоев и нейронов в нейронной сети, скорость обучения, количество деревьев в случайном лесе и другие. Один из распространенных методов подбора гиперпараметров — это перебор по сетке (grid search). При использовании перебора по сетке мы выбираем набор значений для каждого гиперпараметра, и модель обучается и оценивается для каждой комбинации значений. Затем выбирается комбинация, которая демонстрирует наилучшую производительность на валидационных данных. Кроме перебора по сетке, можно использовать алгоритмы оптимизации, такие как случайный поиск (random search) или оптимизация с использованием алгоритма байесовской оптимизации. Эти методы могут быть более эффективными в поиске оптимальных гиперпараметров, особенно если их пространство большое.

Важно отметить, что при подборе гиперпараметров необходимо использовать кросс-валидацию (cross-validation). Она помогает оценить производительность модели на разных наборах данных и уменьшает риск переобучения.

Обычно применяется метод *k-fold cross-validation*, при котором данные разбиваются на *k* поднаборов, и модель обучается и оценивается *k* раз, каждый раз используя разные комбинации поднаборов для обучения и валидации.

После подбора оптимальных гиперпараметров и обучения модели на полном наборе обучающих данных оценивается ее производительность на тестовом наборе данных. Тестовый набор данных не использовался ни при обучении, ни при проверке модели. Оценка производительности на тестовых данных поможет нам оценить, насколько хорошо модель обобщает данные и способна предсказывать задержки авиарейсов в реальных условиях.

В итоге, процесс проверки и подбора гиперпараметров модели позволяет нам оптимизировать ее производительность и достичь наилучшей точности предсказания задержек авиарейсов. Этот процесс требует тщательного анализа и экспериментирования с различными гиперпараметрами, а также использования соответствующих метрик оценки и методов кросс-валидации. Только после успешного завершения этого процесса можно гарантировать, что разработанная модель предсказания задержек авиарейсов является оптимальной и готова к интеграции в практическую среду авиационной индустрии.

2.1.10 Проверка и валидация модели на тестовом наборе данных

Этот этап является важным для оценки производительности модели и ее способности предсказывать задержки авиарейсов с высокой точностью. Для этого будут использованы различные метрики и методы оценки. После завершения обучения модели на обучающем наборе данных необходимо проверить ее работу на тестовом наборе данных, который ранее не использовался при обучении. Это позволит оценить, насколько хорошо модель обобщает знания и способна предсказывать задержки авиарейсов на новых данных.

В начале этого этапа проводится подготовка тестового набора данных. Он должен быть представлен в том же формате и с теми же признаками, что и обучающий набор данных. Затем модель применяется к тестовым данным, и для каждого примера в тестовом наборе данных предсказывается задержка

авиарейса. Далее следует этап оценки производительности модели. Для этого применяются различные метрики, которые позволяют измерить точность и эффективность модели. Некоторые из распространенных метрик, которые можно использовать в этом контексте, включают среднеквадратичную ошибку (Mean Squared Error, MSE), среднюю абсолютную ошибку (Mean Absolute Error, MAE), коэффициент детерминации (Coefficient of Determination, R^2) и другие. MSE измеряет среднюю квадратичную разницу между фактическими значениями и предсказанными значениями задержек авиарейсов. Чем меньше значение MSE, тем лучше производительность модели. MAE, с другой стороны, измеряет среднюю абсолютную разницу между фактическими и предсказанными значениями, и также стремится к минимуму. Коэффициент детерминации R^2 показывает, насколько хорошо модель объясняет вариацию в данных. Он может принимать значения от 0 до 1, где 1 означает идеальное соответствие модели данным, а 0 означает, что модель не объясняет никакую вариацию и прогнозы не лучше случайных. После вычисления метрик производительности модели производится анализ результатов.

Важно проанализировать ошибки модели и выявить паттерны или особенности, которые могут влиять на ее производительность. Это позволит сделать выводы о том, какие улучшения или корректировки можно внести в модель для повышения ее точности и надежности в предсказании задержек авиарейсов. Кроме того, важно провести сравнительный анализ различных моделей или подходов к предсказанию задержек авиарейсов. Это позволит определить, насколько эффективно разработанная модель работает по сравнению с другими существующими моделями и методами. Результаты сравнения могут служить основой для принятия решений о дальнейшем развитии и улучшении разработанного решения.

Таким образом, проверка и валидация модели на тестовом наборе данных являются важной частью процесса разработки предсказательной модели задержек авиарейсов. Этот этап позволяет оценить производительность мо-

дели и ее способность предсказывать задержки авиарейсов с высокой точностью. Результаты этого этапа могут служить основой для дальнейшего улучшения и оптимизации модели.

2.1.11 Описание процесса интеграции модели в практическую среду авиационной индустрии

Интеграция модели предсказания задержек авиарейсов в практическую среду авиационной индустрии играет важную роль в обеспечении ее эффективности и реализации на практике.

Первым шагом в процессе интеграции является анализ требований и контекста, в котором модель будет использоваться. Это включает в себя понимание бизнес-потребностей авиакомпаний, ограничений и требований с точки зрения инфраструктуры и доступных данных. Ключевые вопросы, которые необходимо рассмотреть, включают совместимость с существующими системами авиакомпаний, возможности масштабирования, безопасность данных и взаимодействие с другими компонентами инфраструктуры.

Для успешной интеграции модели необходимо установить процесс получения данных о задержках авиарейсов. Это может включать взаимодействие с различными источниками данных, такими как авиакомпании, аэропорты, агентства по контролю воздушного движения и другие стороны, предоставляющие информацию о задержках. Важно установить автоматизированный и надежный поток данных, чтобы обновлять модель в режиме реального времени и поддерживать актуальность прогнозов.

Для обеспечения доступности и масштабируемости модели необходимо развернуть ее на серверной инфраструктуре. Это может включать создание высокопроизводительного серверного окружения, где модель будет исполняться и принимать запросы на предсказания задержек. Важно обеспечить надежность и масштабируемость системы, чтобы она могла обрабатывать большой объем запросов и обеспечивать высокую доступность.

Для удобного и эффективного использования модели необходимо разработать пользовательский интерфейс, который позволит пользователям взаимодействовать с моделью и получать предсказания задержек авиарейсов. Интерфейс должен быть интуитивно понятным, информативным и удобным в использовании. Кроме того, важно обеспечить интеграцию модели с существующими системами авиакомпаний, такими как системы бронирования и операционные системы, чтобы обеспечить единый информационный поток и эффективное применение предсказаний.

После завершения процесса интеграции необходимо провести тестирование и валидацию развернутой системы. Это включает в себя проверку корректности работы модели, ее точности и эффективности в реальных условиях. Если выявляются проблемы или потенциальные улучшения, проводится оптимизация системы с учетом обратной связи и новых требований.

Процесс интеграции модели предсказания задержек авиарейсов в практическую среду авиационной индустрии требует тщательного анализа требований, подготовки данных, развертывания на серверной инфраструктуре, разработки пользовательского интерфейса и интеграции с существующими системами. Тестирование, валидация и оптимизация являются важной частью процесса, чтобы обеспечить высокую эффективность и достижение поставленных целей.

2.1.12 Разработка прототипа системы предсказания задержек авиарейсов

Целью этого этапа является создание функционального прототипа, который демонстрирует работу разработанной модели предсказания задержек и интегрируется в практическую среду авиационной индустрии. Ниже представлены основные шаги и этапы разработки прототипа:

1. определение требований к прототипу — обозначаем требования к функциональности прототипа и его основным характеристикам. Рассматриваются возможности интеграции с существующими системами авиакомпаний, взаимодействие с пользователем, интерфейс и другие ключевые аспекты;
2. проектирование архитектуры прототипа — разрабатываем архитектуру прототипа, определяя его компоненты и взаимосвязи между ними. Рассматриваются вопросы, связанные с обработкой данных, взаимодействием с моделью предсказания задержек, управлением пользовательским интерфейсом и другими функциональными блоками;
3. реализация и интеграция — реализуем компоненты прототипа и проводим их интеграцию. Разработанная модель предсказания задержек авиарейсов включается в прототип, обеспечивая функциональность предсказания. Также происходит интеграция с существующими системами авиакомпаний для обмена данными и взаимодействия;
4. тестирование и отладка – прототип системы предсказания задержек проходит этап тестирования, в ходе которого проверяется его работоспособность, соответствие требованиям и корректность предсказаний задержек. Выявленные ошибки и проблемы отлаживаются и исправляются для достижения стабильной работы прототипа;
5. оценка производительности и эффективности — прототип подвергается анализу производительности и эффективности его работы. Измеряются время предсказания задержек, точность прогнозов и другие метрики. Проводятся сравнения с существующими методами и моделями предсказания задержек для оценки превосходства разработанного решения;

6. доработка и улучшение — на основе результатов тестирования и оценки производительности прототипа проводятся вносятся изменения в алгоритмы и параметры модели предсказания задержек, оптимизируются процессы обработки данных и взаимодействия с системами авиакомпаний;

7. документация и демонстрация — разработанный прототип системы предсказания задержек авиарейсов документируется, чтобы обеспечить понимание его работы и функциональности. Также проводится демонстрация прототипа заинтересованным сторонам, чтобы получить обратную связь и оценку его эффективности;

Результатом данного этапа является работоспособный прототип системы предсказания задержек авиарейсов, который демонстрирует принцип работы разработанного решения и его интеграцию в практическую среду авиационной индустрии.

2.1.13 Тестирование и проверка работоспособности прототипа

Данный этап является важным для оценки эффективности и точности предсказаний модели, а также для проверки соответствия прототипа требованиям и ожиданиям пользователей:

1. подготовка тестовых данных — для тестирования прототипа необходимо подготовить набор данных, который будет репрезентативным и соответствующим реальным условиям авиационной индустрии. Этот набор данных должен включать разнообразные факторы, такие как типы авиарейсов, аэропорты, временные интервалы и другие контекстные параметры. Кроме того, необходимо учесть различные сценарии задержек, чтобы проверить, насколько хорошо прототип справляется с разными ситуациями;

2. запуск прототипа и сбор результатов — после подготовки тестовых данных прототип системы предсказания задержек авиарейсов должен быть запущен на этом наборе данных. В процессе работы прототипа будут генерироваться предсказания задержек для каждого тестового случая. Необходимо со-

бирать и записывать результаты предсказаний, а также сравнивать их с фактическими значениями задержек, если такие данные имеются. Это позволит оценить точность и надежность прототипа;

3. анализ результатов и оценка эффективности — полученные результаты предсказаний должны быть рассмотрены для оценки эффективности прототипа. Сравнение предсказанных значений с фактическими задержками позволит определить степень точности модели. Метрики, такие как средняя абсолютная ошибка, среднеквадратичная ошибка или коэффициент детерминации, могут быть использованы для количественной оценки результатов. Также важно провести анализ ошибок и выявить причины возможных расхождений между предсказаниями и фактическими данными;

4. уточнение и улучшение модели — в случае обнаружения недостатков или неудовлетворительных результатов тестирования необходимо провести уточнение и улучшение модели. Это может включать в себя изменение алгоритмов, подбор новых гиперпараметров, дополнительную предобработку данных или другие меры для повышения точности и надежности предсказаний. Тестирование и итеративное улучшение модели являются важной частью процесса разработки и обеспечивают достижение желаемых результатов;

5. верификация и проверка соответствия требованиям — наконец, прототип должен быть подвергнут верификации и проверке соответствия требованиям, поставленным перед системой предсказания задержек авиарейсов. Это включает оценку функциональности, производительности, масштабируемости, надежности и других аспектов. Если прототип успешно проходит эту проверку, он может считаться готовым к дальнейшей интеграции и применению в реальных условиях авиационной индустрии.

Тестирование и проверка работоспособности прототипа являются важным этапом в разработке решения предсказания задержек авиарейсов. Это позволяет оценить эффективность и точность модели, а также уточнить и улучшить ее, чтобы она максимально соответствовала требованиям пользователей и контексту авиационной индустрии.

2.1.14 Описание разработанной методологии предупреждения о задержках авиарейсов

В данном пункте будет представлены основные принципы, на которых основана разработанная методология предупреждения о задержках авиарейсов.

Анализ факторов задержек: описан процесс анализа различных факторов, которые могут влиять на задержки авиарейсов. Это может включать погодные условия, технические проблемы, организационные факторы и другие влияющие факторы. Определение этих факторов позволяет более точно предсказывать возможные задержки.

Система сбора данных: рассмотрена система сбора данных, необходимая для предупреждения о задержках. Это может включать использование различных источников данных, таких как данные авиакомпаний, метеорологические данные, данные о трафике и другие. Описывается процесс сбора, обработки и агрегации этих данных.

Модель предсказания: представлено описание модели предсказания задержек авиарейсов, разработанной в предыдущем разделе. Будут рассмотрены методы машинного обучения и алгоритмы, применяемые для предсказания задержек на основе собранных данных. Также будет описан процесс обучения модели и ее настройки.

Установление пороговых значений: рассматривается то, каким образом устанавливаются пороговые значения для предупреждения о задержках. Это включает определение временных интервалов и длительностей задержек, которые считаются значимыми для предупреждения пассажиров и авиакомпаний. Описываются критерии, по которым производится оценка значимости задержек.

Уведомления и рекомендации: представлено описание системы уведомлений и рекомендаций, которая основана на разработанной методологии предупреждения о задержках. Будет рассмотрено, каким образом информация о задержках передается пассажирам и авиакомпаниям, и какие рекомендации

могут быть предоставлены для улучшения опыта пассажиров и оптимизации операций авиакомпаний.

2.1.15 Определение пороговых значений и критериев для предупреждения о задержках

В разработке решения для предсказания задержек авиарейсов важным аспектом является определение пороговых значений и критериев, которые будут использоваться для предупреждения пассажиров и авиакомпаний о возможных задержках. Это позволит принимать своевременные меры для минимизации отрицательных последствий и обеспечения комфортного путешествия.

В первую очередь необходимо определить пороговые значения, которые будут служить границей между нормальным функционированием авиарейсов и возможными задержками. Для этого необходимо проанализировать исторические данные об опозданиях, учитывая различные факторы, влияющие на возникновение несостыковок в расписании, такие как погодные условия, типы авиарейсов, аэропорты и другие контекстные параметры.

Определение пороговых значений должно быть основано на балансе между предупредительностью и точностью предсказания. Слишком низкие пороговые значения могут привести к большому количеству ложных предупреждений, что может создать неудобства для пассажиров и лишнюю тревогу. С другой стороны, слишком высокие – могут привести к недостаточной предупредительности и оставить пассажиров и авиакомпании неподготовленными к возможным задержкам.

Для определения пороговых значений можно использовать статистические методы, такие как анализ распределения времени опозданий и определение перцентилей. Например, можно выбрать пороговое значение, соответствующее 90-му перцентилю времени задержек. Это означает, что, если предсказанное время превышает это значение, будет считаться, что задержка является значительной и требует предупреждения пассажиров.

Кроме пороговых значений, необходимо также определить критерии, по которым будет осуществляться предупреждение о задержках. Критерии могут включать следующие аспекты:

1. время до предполагаемого вылета: если предсказанное время задержки указывает на возможность опоздания на самолет, пассажиры должны быть предупреждены заранее, чтобы иметь возможность принять соответствующие меры, например, пересмотреть свое расписание или принять связь с авиакомпанией;

2. длительность задержки: если предсказанная задержка превышает определенную длительность (например, 30 минут), пассажиры и авиакомпания должны быть предупреждены о возможных изменениях в расписании и организации дополнительного обслуживания;

3. тип задержки: в зависимости от причины задержки (например, погодные условия, технические проблемы или проблемы с персоналом), могут потребоваться разные меры для управления ситуацией. Пассажиры и авиакомпания должны быть информированы о типе задержки, чтобы принять соответствующие меры;

4. пересадки и связи: если задержка может повлиять на пересадки и связи пассажиров, особое внимание должно уделяться предупреждению и организации альтернативных вариантов для минимизации неудобств.

Все определенные пороговые значения и критерии должны быть гибкими и настраиваемыми, чтобы учитывать различные условия и требования авиакомпании. Критерии могут быть уточнены и доработаны на основе обратной связи от пассажиров и результатов мониторинга задержек.

Определение пороговых значений и критериев для предупреждения о задержках является важным шагом в разработке решения предсказания задержек авиарейсов. Это позволяет создать систему, которая способна своевременно информировать пассажиров и авиакомпанию о возможных задержках, обеспечивая таким образом более эффективное управление процессом авиаперевозок и улучшение пассажирского опыта.

2.1.16 Создание системы уведомлений и рекомендаций для пассажиров и авиакомпаний

В рамках разработанной методологии предупреждения о задержках авиарейсов, важным этапом является создание системы уведомлений и рекомендаций, которая будет обеспечивать информирование пассажиров и авиакомпаний о возможных задержках и предлагать соответствующие рекомендации для минимизации негативных последствий. Эта система играет ключевую роль в обеспечении комфортного и предсказуемого опыта путешествия для всех заинтересованных сторон.

Определение информационных потребностей пассажиров и авиакомпаний: первым шагом является определение того, какая информация является наиболее важной для пассажиров и авиакомпаний в случае задержек авиарейсов. Это может включать уведомления о времени задержки, причине задержки, предполагаемом новом времени вылета, возможных вариантах пересадок и другую полезную информацию. Важно учесть различные потребности разных групп пассажиров, таких как бизнес-пассажиры, туристы или семьи с детьми, и предоставить им релевантные уведомления.

На основе информационных потребностей, определенных на предыдущем шаге, необходимо разработать систему уведомлений, которая будет доставлять информацию пассажирам и авиакомпаниям в удобной и своевременной форме. Это может быть реализовано через различные каналы связи, такие как SMS-сообщения, электронная почта, мобильные приложения или автоматические голосовые сообщения. Важно обеспечить надежность доставки уведомлений и возможность получения обратной связи от пассажиров.

Система уведомлений также должна предоставлять рекомендации и альтернативные варианты для пассажиров и авиакомпаний в случае задержек. Например, если задержка ожидается на несколько часов, система может предложить пассажирам варианты пересадки на другие рейсы, предложить возможность изменить бронирование или предоставить информацию о доступных услугах в аэропорту. Авиакомпании также могут получать рекомендации

о возможных мероприятиях, например изменении расписания, организации дополнительных рейсов или предоставлении компенсаций.

Для улучшения опыта пассажиров и эффективности системы уведомлений важно предусмотреть возможность персонализации и контекстной адаптации. Например, система может учитывать предпочтения пассажира, предоставлять информацию на его предпочитаемом языке или учитывать его статус программы лояльности. Также система должна быть гибкой и способной адаптироваться к изменяющимся условиям и контексту, чтобы предоставлять более точные и релевантные уведомления и рекомендации.

Чтобы система уведомлений и рекомендаций была полноценной и эффективной, необходима интеграция с другими системами, такими как системы бронирования авиакомпаний, системы управления аэропортами или системы управления пассажирскими данными. Это позволит обеспечить получение актуальной информации о рейсах, пассажирах, расписаниях и других факторах, необходимых для предоставления точных уведомлений и рекомендаций.

В результате создания системы уведомлений и рекомендаций для пассажиров и авиакомпаний, будет обеспечена более эффективная коммуникация, предсказуемость и возможность принятия соответствующих мер для минимизации негативных последствий задержек авиарейсов. Это повысит уровень удовлетворенности пассажиров, снизит операционные риски для авиакомпаний и способствует более гладкому функционированию авиационной индустрии в целом.

2.1.17 Описание общей архитектуры разработанного решения

В данном разделе представляется общая архитектура разработанного решения для предсказания задержек авиарейсов. Архитектура играет важную роль в обеспечении эффективного функционирования системы предсказания задержек и интеграции ее в существующую инфраструктуру авиационной индустрии.

Общая архитектура разработанного решения состоит из нескольких ключевых компонентов, которые взаимодействуют между собой для обеспечения предсказания задержек авиарейсов:

1. сбор данных отвечает за сбор и хранение данных о задержках авиарейсов. Данные могут быть получены из различных источников, включая базы данных авиакомпаний, государственные организации и сторонние сервисы. Собранные данные могут включать информацию о времени вылета и прилета, типе задержки, причинах задержки и других соответствующих параметрах;

2. предварительная обработка данных включает в себя удаление выбросов, заполнение пропущенных значений, нормализацию данных и преобразование их в удобный для анализа формат. Это позволяет очистить данные от шума и подготовить их для дальнейшего использования в модели предсказания;

3. модель предсказания – основной элементом разработанного решения. В нем реализована выбранная модель машинного обучения, которая обучается на обработанных данных и используется для предсказания задержек авиарейсов. Модель может быть основана на различных алгоритмах, таких как линейная регрессия, случайный лес или нейронные сети. Она учитывает различные факторы, влияющие на задержки, и на основе этих данных строит предсказания с определенной степенью точности;

4. результаты предсказаний задержек авиарейсов интегрируются в существующую инфраструктуру авиационной индустрии. Они могут быть переданы авиакомпаниям, аэропортам, системам управления полетами и другим заинтересованным сторонам. Помимо этого, результаты могут быть хранены для последующего анализа и улучшения модели;

5. разработанное решение включает механизм мониторинга и обновления модели предсказания. Это позволяет системе быть актуальной и адаптироваться к изменяющимся условиям и требованиям авиационной индустрии. Мониторинг включает проверку работоспособности модели, обнаружение аномалий и нештатных ситуаций. При необходимости модель может быть переобучена или доработана для повышения точности предсказаний.

Общая архитектура разработанного решения обеспечивает эффективное функционирование системы предсказания задержек авиарейсов. Интеграция различных компонентов позволяет использовать доступные данные, обрабатывать их и строить предсказания с высокой точностью. Результаты предсказаний могут быть интегрированы в существующие системы авиакомпаний и других участников авиационной индустрии, что помогает снизить задержки и повысить операционную эффективность.

2.1.18 Обзор основных компонентов и их взаимодействия

Каждый компонент имеет свою специфику и выполняет определенные функции, вкладываясь в общую архитектуру решения:

1. сбор данных отвечает за сбор и хранение данных о задержках авиарейсов. Для этого используются различные источники данных, такие как базы данных авиакомпаний, открытые данные о задержках рейсов, метеорологические данные и другие релевантные источники. Данные собираются и обновляются регулярно, чтобы быть актуальными для предсказаний;
2. предварительная обработка данных включает в себя очистку данных от выбросов, заполнение пропущенных значений, преобразование формата данных и масштабирование переменных. Цель этого этапа — подготовить данные для дальнейшего анализа и обучения модели;
3. анализ данных и идентификация влияющих факторов, которые оказывают наибольшее влияние на задержки. Используя методы статистического анализа и машинного обучения, проводится исследование связей между различными переменными, такими как погодные условия, время суток, аэропортовая загруженность и другие факторы. Это помогает определить наиболее значимые факторы, которые будут использоваться в модели предсказания;
4. моделирование и обучение модели. Основываясь на результате анализа данных и идентификации влияющих факторов, выбирается модель, способная учесть эти факторы и дать точные предсказания. Она обучается на обучающем

наборе данных, где происходит настройка параметров модели и минимизация ошибки предсказания;

5. для оценки качества и надежности модели проводится валидация на тестовом наборе данных. Она применяется к новым наборам данных, которые не использовались в процессе обучения, и оценивается точность ее предсказаний. Результаты валидации позволяют оценить эффективность модели и, при необходимости, внести корректировки;

6. после успешной валидации модели она интегрируется в разработку прототипа системы предсказания задержек авиарейсов. Прототип представляет собой функциональную систему, включающую интерфейс пользователя, модуль предсказания и модуль отображения результатов. Прототип проходит тестирование и проверку работоспособности;

7. разработанный прототип системы предсказания задержек авиарейсов деплоится в рабочую среду авиационной индустрии. Система становится доступной для использования авиакомпаниями и пассажирами. Кроме того, проводится мониторинг работы системы, чтобы обеспечить ее стабильность и своевременность предсказаний. В случае необходимости вносятся корректировки и обновления модели и системы.

В данном разделе мы рассмотрели основные компоненты разработанного решения для предсказания задержек авиарейсов и их взаимодействие. Каждый компонент играет важную роль в создании точной и надежной предсказательной модели, способной снизить задержки и повысить операционную эффективность авиационной индустрии.

2.1.19 Разъяснение особенностей решения и его потенциала для дальнейшего развития

Рассмотрим особенности разработанного решения для предсказания задержек авиарейсов и рассмотрим его потенциал для дальнейшего развития.

Одной из основных особенностей разработанного решения является его способность учитывать разнообразные факторы, влияющие на задержки авиарейсов. В ходе исследования предметной области мы выявили множество факторов, которые могут оказывать влияние на задержки, включая погодные условия, технические проблемы, организационные факторы и другие непредвиденные обстоятельства. Разработанная модель учитывает эти факторы и использует их в качестве входных данных для предсказания задержек авиарейсов. Это позволяет получить более точные и надежные прогнозы и предупреждать пассажиров заранее о возможных задержках.

Еще одной особенностью разработанного решения является его гибкость и масштабируемость. Модель предсказания задержек авиарейсов может быть легко адаптирована под различные типы авиарейсов, аэропорты и географические особенности. Это позволяет применять решение в различных регионах и авиационных компаниях, а также учитывать специфические требования и контекстные параметры. Благодаря гибкости и масштабируемости, разработанное решение может быть успешно внедрено в различных сценариях и способствовать повышению операционной эффективности авиакомпаний и улучшению опыта путешествия для пассажиров.

Также стоит отметить потенциал разработанного решения для дальнейшего развития. Внедрение модели предсказания задержек авиарейсов представляет лишь начало процесса. Система может быть доработана и расширена с целью улучшения точности предсказаний и функциональности. Например, можно провести дополнительные исследования для определения новых факторов, влияющих на задержки, и включить их в модель. Также можно усовершенствовать алгоритмы обучения модели и использовать более сложные методы машинного обучения, такие как глубокое обучение и рекуррентные нейронные сети, для достижения более высокой точности предсказаний.

Кроме того, разработанное решение может быть интегрировано с другими системами и сервисами, связанными с авиацией. Например, информация о задержках может быть интегрирована с системами управления полетами и

бронирования билетов, что позволит авиакомпаниям оперативно реагировать на возможные задержки и предпринимать соответствующие меры. Также можно разработать мобильные приложения или веб-интерфейсы, через которые пассажиры смогут получать информацию о задержках и получать персонализированные рекомендации.

В целом, разработанное решение предлагает широкие возможности для оптимизации операций авиакомпаний, улучшения пассажирского опыта и снижения негативных последствий задержек авиарейсов. При его внедрении следует учитывать особенности конкретной авиационной компании и контекстные параметры, а также продолжать исследования и развитие модели с целью достижения еще более точных и надежных результатов.

2.2 Стэк используемых технологий

В современном мире авиаперевозки занимают важное место в международной и внутренней экономике. Однако задержки рейсов могут негативно влиять на пассажиров, авиакомпании и весь экономический сектор, связанный с авиацией. Чтобы улучшить операционную эффективность авиакомпаний и предоставить более точные прогнозы для пассажиров, мы разработали проект по предсказанию вероятности задержки рейсов в аэропорту. В данном тексте мы рассмотрим выбранный стек технологий и компоненты разработанного решения.

Существует несколько популярных языков программирования, которые широко используются для разработки машинного обучения и нейронных сетей. Некоторые из них включают:

1. Python является одним из наиболее популярных языков программирования для машинного обучения. Он предлагает множество библиотек и фреймворков, таких как TensorFlow, PyTorch, scikit-learn, Keras и другие, которые обеспечивают мощные инструменты для разработки и обучения моделей машинного обучения и нейронных сетей;

2. R также является популярным языком программирования, используемым в области машинного обучения и статистики. Он обладает богатым набором пакетов, таких как `caret`, `randomForest`, `ggplot2` и другие, которые обеспечивают широкие возможности для анализа данных и разработки моделей машинного обучения;

3. Java является одним из самых распространенных языков программирования в общем, и он также нашел свое применение в разработке машинного обучения и нейронных сетей. Некоторые популярные библиотеки и фреймворки для машинного обучения на Java включают `Deeplearning4j`, `WEKA` и `DL4J`;

4. C++ является языком программирования с высокой производительностью и широко используется в разработке машинного обучения и нейронных сетей. Библиотека `TensorFlow` предоставляет интерфейс для разработки моделей на C++;

5. Julia — это относительно новый язык программирования, который был разработан с учетом потребностей научных вычислений и машинного обучения. Он обладает простым синтаксисом и хорошей производительностью, и имеет библиотеки, такие как `Flux.jl`, для разработки нейронных сетей.

Проект написан на языке Python, поскольку этот язык считается одним из самых популярных языков программирования для машинного обучения и нейронных сетей по нескольким причинам:

1. простота и удобство использования: Python имеет простой и понятный синтаксис, который делает его легким для изучения и использования. Он обладает читаемым и понятным кодом, что упрощает разработку и сопровождение программ;

2. большое сообщество и экосистема: Python имеет огромное активное сообщество разработчиков, которые создают и поддерживают множество библиотек, фреймворков и инструментов для машинного обучения. Библиотеки, такие как `TensorFlow`, `PyTorch`, `scikit-learn`, `Keras` и другие, предоставляют

мощные возможности для разработки и обучения моделей машинного обучения и нейронных сетей;

3. широкий спектр применений: Python не ограничивается только машинным обучением и нейронными сетями. Он может использоваться для разработки веб-приложений, научных вычислений, анализа данных, автоматизации задач, разработки игр и многого другого. Эта универсальность делает Python привлекательным языком программирования для различных проектов.

4. интеграция с другими языками и инструментами: Python может быть легко интегрирован с другими языками программирования, такими как C++ и Java, что позволяет использовать его в совместных проектах. Он также поддерживает множество библиотек и пакетов для визуализации данных, обработки естественного языка, работе с базами данных и другими задачами;

5. обучающие ресурсы: существует множество книг, онлайн-курсов, учебных материалов и сообществ, посвященных изучению и применению Python в машинном обучении. Это облегчает начало работы и обучение новичков.

В целом, Python предлагает мощные инструменты, богатую экосистему и обширное сообщество, что делает его очень привлекательным выбором для разработки и исследования моделей машинного обучения и нейронных сетей.

При выборе библиотеки для обработки датасетов в Python существует несколько факторов, которые могут влиять на решение. Вот шесть основных причин, почему многие разработчики выбирают библиотеку pandas:

1. мощные функции для работы с данными: pandas предоставляет обширный набор функций для загрузки, очистки, преобразования и анализа данных. Например, функция `read_csv` позволяет легко и быстро загружать данные из CSV-файлов в формате таблицы;

2. гибкость и эффективность: pandas предоставляет высокоуровневые структуры данных, такие как `DataFrame`, которые позволяют легко манипулировать и анализировать таблицы данных. Он также оптимизирован для работы

с большими объемами данных, что делает его эффективным выбором для обработки больших датасетов;

3. интеграция с другими библиотеками: pandas тесно интегрирован с другими популярными библиотеками Python для научных вычислений, такими как NumPy, SciPy и Matplotlib. Это позволяет использовать возможности этих библиотек в сочетании с pandas для выполнения различных задач обработки данных и визуализации;

4. обработка пропущенных данных: pandas предоставляет функции для работы с пропущенными данными, такими как `dropna` и `fillna`, что упрощает обработку пропущенных значений в датасете;

5. возможность работы с временными рядами: pandas имеет встроенную поддержку работы с временными рядами, включая инструменты для ресемплирования, индексации и агрегации данных по времени. Это делает его удобным инструментом для анализа данных, связанных с временными рядами, такими как финансовые данные или данные с датами и временем;

6. большое сообщество и документация: pandas является одной из самых популярных библиотек для работы с данными в Python, и у нее есть активное сообщество разработчиков. Это означает, что есть обширная документация, множество обучающих материалов и готовых решений для различных типов задач.

Не смотря на популярность pandas, существуют и другие библиотеки для работы с данными в Python, такие как NumPy, Dask, Vaex, Modin и др. Вот некоторые из них:

1. NumPy — это базовая библиотека для научных вычислений в Python, которая предоставляет мощные структуры данных, такие как многомерные массивы. Она обеспечивает эффективную обработку числовых данных и может использоваться в сочетании с pandas для выполнения вычислений над данными;

2. Dask — это библиотека для распределенных вычислений и обработки больших данных. Она позволяет обрабатывать данные, не помещая их полностью в память, и может быть полезной для работы с очень большими датасетами;

3. Vaex — это еще одна библиотека для обработки больших объемов данных. Она оптимизирована для быстрой работы с большими датасетами, используя ленивые вычисления и параллельные алгоритмы;

4. Modin — это библиотека, которая стремится улучшить производительность pandas за счет использования распределенных вычислений и параллелизма. Она предоставляет совместимый с pandas интерфейс и может быть полезной для обработки больших данных на многопроцессорных системах.

Pandas обладает рядом преимуществ, которые делают его популярным выбором для обработки и анализа данных. Вот некоторые из ключевых преимуществ pandas по сравнению с другими библиотеками:

1. простота использования: Pandas предоставляет простой и интуитивно понятный интерфейс для работы с данными в формате таблицы. Его функции и методы разработаны таким образом, чтобы сделать обработку данных более простой и удобной для пользователя;

2. обширный функционал: Pandas предлагает широкий набор функций для загрузки, очистки, преобразования и анализа данных. Он обеспечивает мощные инструменты для фильтрации, группировки, агрегации, соединения и изменения данных, позволяя эффективно проводить разнообразные операции над датасетом;

3. удобная работа с пропущенными данными: Pandas имеет встроенную поддержку для обработки пропущенных данных. Он предоставляет методы для обнаружения, удаления и заполнения пропущенных значений, что делает обработку пропущенных данных более удобной и гибкой;

4. поддержка временных рядов: Pandas обладает богатыми возможностями для работы с данными, связанными с временными рядами. Он предоставляет

инструменты для индексации, ресемплирования, сдвига, выравнивания и агрегации временных данных. Это делает его идеальным инструментом для анализа финансовых данных, метеорологических данных и других временных рядов;

5. интеграция с другими библиотеками: Pandas плотно интегрирован с другими популярными библиотеками Python для научных вычислений, такими как NumPy, SciPy и Matplotlib. Это позволяет использовать возможности этих библиотек в сочетании с pandas для выполнения различных задач обработки данных и визуализации;

6. большое сообщество и документация: Pandas является одной из самых популярных библиотек для работы с данными в Python, и у нее есть большое и активное сообщество разработчиков. Это означает, что есть обширная документация, множество обучающих материалов, ответы на вопросы в сообществах разработчиков и готовые решения для различных типов задач.

Хотя другие библиотеки, такие как Dask, Vaex и Modin, могут быть полезны для обработки больших объемов данных или распределенных вычислений, pandas остается предпочтительным выбором для большинства задач обработки данных в Python благодаря своей простоте использования, обширному функционалу и удобству работы.

Для чтения датасета и его обработки в проекте использовалась библиотека json. Библиотека JSON (JavaScript Object Notation) в Python предоставляет мощные инструменты для работы с данными в формате JSON. JSON является одним из самых популярных форматов обмена данными, используемых в современном программировании, и библиотека JSON в Python обеспечивает простоту и гибкость при работе с ним.

Библиотека JSON в Python может преобразовывать объекты Python в формат JSON и наоборот. Функция `json.dumps()` используется для сериализации (преобразования в строку) объектов Python в формат JSON. Она принимает объект Python и возвращает соответствующую строку JSON. С другой

стороны, функция `json.loads()` используется для десериализации (преобразования строки) JSON в объект Python. Она принимает строку JSON и возвращает соответствующий объект Python.

Библиотека JSON также предоставляет возможность работы с более сложными структурами данных. Например, можно сериализовать и десериализовать словари, списки, кортежи и даже пользовательские классы. Это позволяет легко передавать и сохранять структурированные данные в формате JSON.

Для удобства работы с JSON в Python также доступны функции `json.dump()` и `json.load()`, которые позволяют записывать и загружать данные JSON из файлов. Функция `json.dump()` принимает объект и файловый объект, и записывает данные в формате JSON в файл. Функция `json.load()` загружает данные из файла JSON и возвращает соответствующий объект Python.

Библиотека JSON в Python обрабатывает различные типы данных, такие как строки, числа, булевы значения, списки, словари и значения `null`. Она автоматически преобразует эти типы данных между Python и JSON, обеспечивая совместимость данных между различными программами и системами.

Кроме того, библиотека JSON в Python предоставляет возможность контроля над процессом сериализации и десериализации. Например, можно определить собственные функции-кодировщики (`encoders`) и декодировщики (`decoders`), чтобы настроить процесс преобразования объектов в JSON и обратно. Это полезно, если требуется специальная обработка определенных типов данных или настройка формата JSON.

В целом, библиотека JSON в Python предоставляет мощные и гибкие инструменты для работы с данными в формате JSON. Она широко используется во множестве приложений и сценариев, где требуется обмен данными между различными системами и языками программирования.

Фреймворки для нейронных сетей являются важным инструментом в области глубокого обучения. Они предоставляют разработчикам удобные средства для создания, обучения и развертывания нейронных сетей. Существует несколько популярных фреймворков для нейросетей, таких как TensorFlow, PyTorch, Keras и Caffe.

Одним из самых широко используемых и популярных фреймворков для нейросетей является TensorFlow. Он разработан компанией Google и обладает множеством преимуществ, которые делают его предпочтительным выбором для многих разработчиков.

Первое преимущество TensorFlow заключается в его гибкости. Он позволяет разработчикам создавать различные типы моделей нейронных сетей, включая сверточные нейронные сети (CNN), рекуррентные нейронные сети (RNN), генеративно-состязательные сети (GAN) и многое другое. TensorFlow также обеспечивает широкий выбор слоев, оптимизаторов и функций активации, что позволяет более точно настраивать модели и достигать лучших результатов.

Еще одним преимуществом TensorFlow является его высокая производительность. Он может использовать графический процессор (GPU) для ускорения вычислений, что особенно полезно при работе с большими объемами данных и сложными моделями. TensorFlow также поддерживает распределенное обучение, что позволяет распределить вычисления между несколькими устройствами или серверами, ускоряя процесс обучения.

Одним из основных компонентов TensorFlow является модуль `tf.keras`, который предоставляет высокоуровневый интерфейс для создания нейронных сетей. Он упрощает процесс создания моделей и позволяет разработчикам быстро прототипировать и экспериментировать с различными архитектурами.

Класс `tf.keras.Sequential` является одной из основных функций в `tf.keras`. Он позволяет создавать модели нейронных сетей, последовательно соединяя

слои. Последовательная модель является простой и интуитивно понятной, что упрощает ее использование даже для новичков в области глубокого обучения.

Слой `tf.keras.layers.Dense` является основным строительным блоком модели. Он представляет собой полносвязный слой, в котором каждый нейрон связан со всеми нейронами предыдущего слоя. Параметр `activation` определяет функцию активации, применяемую к выходу слоя.

С использованием `tf.keras.Sequential` и `tf.keras.layers.Dense` разработчики могут создавать и настраивать сложные модели нейронных сетей с минимальными усилиями. TensorFlow предоставляет богатый набор инструментов и функций, которые помогают улучшить производительность и результаты моделей. В целом, TensorFlow является мощным и гибким фреймворком для разработки нейронных сетей и находит широкое применение в области искусственного интеллекта и глубокого обучения.

В нашем проекте также используется библиотека Scikit-learn, также известная как `sklearn`. Она является одной из наиболее популярных библиотек машинного обучения для языка программирования Python. Она предоставляет широкий спектр алгоритмов машинного обучения, инструментов для предобработки данных, оценки моделей и выбора параметров.

Одним из главных преимуществ `sklearn` является его простота использования. Библиотека предоставляет последовательный интерфейс для большинства алгоритмов, что делает код более понятным и легко поддерживаемым. В связи с этим, `sklearn` является отличным выбором для начинающих в машинном обучении, а также для быстрой разработки прототипов моделей.

Одним из аналогов `sklearn` является библиотека TensorFlow. TensorFlow обладает более низкоуровневым API, что позволяет большую гибкость и контроль над моделями, однако требует более глубокого понимания и опыта работы с машинным обучением. В то же время, `sklearn` является более простым

и легким в использовании инструментом, предлагающим широкий спектр функций для решения типовых задач машинного обучения.

`StandardScaler()` является одной из функций предобработки данных в `sklearn`. Она выполняет центрирование и масштабирование признаков, приводя их к среднему значению 0 и стандартному отклонению 1. Это особенно полезно, когда признаки имеют разные диапазоны значений, и некоторые алгоритмы машинного обучения требуют нормализованных данных для эффективной работы.

Метод `fit_transform()` используется для применения предобработки данных на основе выбранного преобразователя (например, `StandardScaler()`) к обучающему набору данных. Он одновременно подгоняет преобразователь к данным и применяет его преобразование к данным.

Метод `model.compile()` используется в `sklearn` для настройки параметров модели перед обучением. Он позволяет указать функцию потерь, оптимизатор и метрики, которые будут использоваться в процессе обучения модели. Это важный шаг, который определяет, как модель будет обновлять свои веса в процессе обучения.

Метод `fit()` является основной функцией для обучения модели в `sklearn`. Он принимает на вход обучающие данные и соответствующие целевые значения и обновляет веса модели на основе этих данных. В процессе обучения модель старается минимизировать выбранную функцию потерь, используя выбранный оптимизатор.

Для визуализации точности нейросети используется библиотека `matplotlib`. `Matplotlib` — это популярная библиотека для визуализации данных в языке программирования Python. Она предоставляет широкий спектр инструментов и функций для создания высококачественных графиков, диаграмм, сюжетов и других визуальных представлений данных.

Основные функции `Matplotlib` включают:

1. создание базовых графиков: Matplotlib позволяет создавать различные типы графиков, такие как линейные графики, столбчатые графики, круговые диаграммы, точечные графики и многие другие;
2. настройка внешнего вида: Библиотека предоставляет возможности для настройки различных аспектов графиков, включая заголовки, оси, легенды, цвета, шрифты и размеры;
3. множество стилей и цветовых карт: Matplotlib предлагает различные предустановленные стили и цветовые карты, которые позволяют легко изменить общий вид графиков. Кроме того, можно создавать и использовать собственные стили и цветовые схемы;
4. множество типов линий и маркеров: Вы можете выбрать различные типы линий и маркеров для точек данных на графиках, такие как сплошные линии, пунктирные линии, точки, треугольники, квадраты и т.д.;
5. создание нескольких подграфиков: Matplotlib позволяет создавать комплексные макеты с несколькими подграфиками на одной фигуре. Это полезно при сравнении и отображении нескольких наборов данных одновременно;
6. диаграммы разброса и плотности: Библиотека поддерживает создание диаграмм разброса (scatter plots) для анализа взаимосвязи двух переменных, а также графиков плотности для визуализации распределения данных;
7. 3D-графики: Matplotlib позволяет строить трехмерные графики, позволяющие визуализировать данные в трех измерениях. Это полезно при работе с пространственными или сложными данными;
8. интеграция с NumPy и Pandas: Matplotlib легко интегрируется с другими популярными библиотеками Python, такими как NumPy и Pandas. Вы можете передавать массивы данных

Наш проект развернут в изолированном окружении. Контейнеризация является методом организации и запуска приложений с помощью контейнеров. Контейнеры представляют собой изолированные окружения, в которых

упакованы все необходимые компоненты приложения, включая код, среду выполнения, библиотеки и зависимости. Они предлагают следующие преимущества:

1. **портабельность:** Контейнеры обеспечивают согласованное и независимое окружение, которое можно развернуть на различных платформах, таких как физические серверы, виртуальные машины, облачные инфраструктуры и даже настольные компьютеры. Это позволяет легко перемещать контейнеризованные приложения между разными средами без необходимости внесения изменений;

2. **изоляция:** Контейнеры обеспечивают высокую степень изоляции между приложениями и их зависимостями. Каждый контейнер имеет свою собственную файловую систему, процессы, сетевые ресурсы и пространство имен, что предотвращает возможные конфликты между приложениями и обеспечивает безопасность и надежность;

3. **масштабируемость:** Контейнеры позволяют легко масштабировать приложения, так как они могут быть развернуты и управляться автоматически с использованием оркестраторов контейнеров, таких как Kubernetes. Контейнеры обеспечивают горизонтальное масштабирование, позволяя развернуть несколько экземпляров приложения для обработки повышенной нагрузки;

4. **упрощение развертывания и обновления:** Контейнеризация упрощает процесс развертывания приложений и их обновления. Контейнеры позволяют упаковать все необходимые компоненты приложения вместе, включая зависимости и настройки, что облегчает перенос приложений между средами разработки, тестирования и производства. Обновление контейнеризованных приложений может быть выполнено путем простой замены контейнера новой версией, без необходимости вмешательства в инфраструктуру;

5. **ресурсная эффективность:** Контейнеры используют общий ядро операционной системы и разделяют ресурсы между собой.

Существует несколько популярных сервисов для контейнеризации приложений, включая Docker, Kubernetes, Apache Mesos и Amazon Elastic Container Service (ECS). Рассмотрим их кратко:

1. Docker является одним из самых популярных инструментов для контейнеризации приложений. Он позволяет упаковывать приложения и все их зависимости в контейнеры, которые могут быть запущены на любой платформе. Docker обеспечивает изоляцию приложений, позволяет быстро развертывать и масштабировать приложения, а также обеспечивает легкую портативность и переносимость;

2. Kubernetes (K8s) — это система управления контейнерами с открытым исходным кодом, разработанная Google. Он предоставляет мощные возможности для автоматического развертывания, масштабирования и управления контейнеризированными приложениями. Kubernetes позволяет оркестрировать контейнеры на кластере серверов, предоставляет высокую доступность и обеспечивает удобное масштабирование приложений;

3. Apache Mesos — это система управления кластером с открытым исходным кодом, которая может работать с контейнерами и другими ресурсами. Mesos обеспечивает эффективное распределение ресурсов между приложениями и предоставляет масштабируемый и надежный способ управления контейнерами;

4. Amazon Elastic Container Service (ECS): ECS — это управляемый сервис контейнеризации на Amazon Web Services (AWS). Он обеспечивает простое развертывание и управление контейнерами на инфраструктуре AWS. ECS позволяет запускать контейнеры Docker и управлять ими с помощью высокоуровневых API и консоли управления. На рисунке 8 изображён логотип Docker.

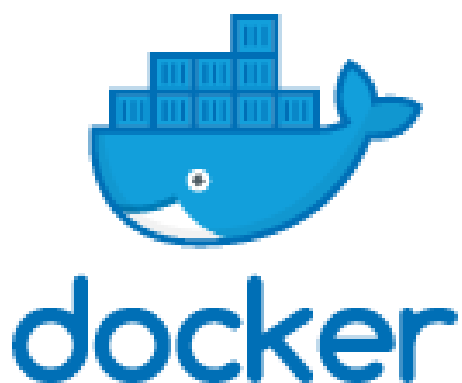


Рисунок 8 — Docker

Контейнеризация приложений позволяет упростить разработку, развертывание и масштабирование приложений, обеспечивает портативность и изоляцию, повышает безопасность и упрощает управление зависимостями. Это делает контейнеры, в частности Docker, очень популярными в сфере разработки и DevOps.

Dockerfile является текстовым файлом, который содержит инструкции для создания Docker-образа. Он используется в Docker для автоматизации процесса сборки образов и настройки контейнеров. Dockerfile позволяет разработчикам определить, какие зависимости и настройки нужно включить в контейнер, чтобы приложение успешно запускалось.

Dockerfile состоит из серии инструкций, каждая из которых выполняется последовательно при создании образа. Рассмотрим основные инструкции Dockerfile:

1. FROM: Инструкция FROM задает базовый образ, на основе которого будет создан новый образ. Это может быть образ операционной системы (например, Ubuntu) или другой образ, который уже содержит установленные зависимости;
2. MAINTAINER (устаревшая): Инструкция MAINTAINER используется для указания автора или контактной информации об образе. Однако она считается устаревшей, и рекомендуется вместо нее использовать LABEL;

3. LABEL: Инструкция LABEL позволяет добавить метаданные к образу, такие как версия приложения, описание, автор и другую информацию. Метаданные можно использовать для поиска и организации образов;

4. RUN: Инструкция RUN используется для выполнения команд внутри контейнера во время создания образа. С помощью этой инструкции можно устанавливать пакеты, клонировать репозитории, настраивать окружение и выполнять другие действия;

5. COPY и ADD: Инструкции COPY и ADD используются для копирования файлов и директорий из контекста сборки в образ. COPY просто копирует файлы, а ADD имеет дополнительные возможности, такие как автоматическая распаковка архивов;

6. WORKDIR: Инструкция WORKDIR устанавливает рабочую директорию для любых следующих инструкций в Dockerfile. Это помогает определить, где будут выполняться команды внутри контейнера;

7. ENV: Инструкция ENV позволяет установить переменные окружения в контейнере. Это может быть полезно для настройки конфигурационных параметров или передачи значений в приложение;

8. EXPOSE: Инструкция EXPOSE указывает на то, какой порт должен быть доступен из контейнера. Однако она не открывает порт на хост-системе, это должно быть сделано при запуске контейнера;

9. CMD и ENTRYPOINT: Инструкции CMD и ENTRYPOINT используются для определения команды, которая будет запущена при запуске контейнера. CMD предоставляет значения по умолчанию, которые можно переопределить при запуске контейнера, а ENTRYPOINT предоставляет фиксированную команду, которая не может быть переопределена;

10. VOLUME: Инструкция VOLUME создает точку монтирования в контейнере и позволяет сохранять данные между запусками контейнера.

Dockerfile позволяет автоматизировать процесс создания образов Docker, обеспечивает повторяемость и портативность среды разработки и выполнения приложений. Он является основным инструментом для контейнеризации приложений с использованием Docker и помогает упростить и ускорить развертывание и масштабирование приложений.

Для описания инфраструктуры и конфигурации нашего сервиса мы используем docker-compose файл.

Docker Compose — это инструмент, который позволяет определить и запустить множество контейнеров Docker как часть одного приложения. Он используется для описания инфраструктуры и конфигурации нескольких сервисов, включая их сетевые настройки, переменные окружения, объемы и другие параметры.

Файл docker-compose.yml — это файл конфигурации, в котором определяются и описываются сервисы, которые будут запущены с помощью Docker Compose. Он использует формат YAML для структурирования данных.

На нашем проекте в docker-compose.yml имеется версия "3" (обозначает версию синтаксиса Docker Compose), и определен один сервис с именем "flights-delay-bot". Давайте рассмотрим подробнее каждую секцию:

1. `version`: здесь указывается версия синтаксиса Docker Compose. В данном случае используется версия "3";
2. `services`: эта секция определяет список сервисов, которые будут запущены. У нас есть только один сервис с именем "flights-delay-bot";
3. `flights-delay-bot`: Имя сервиса. Здесь мы указываем имя для сервиса, которое будет использоваться внутри Docker Compose;
4. `build`: здесь указывается путь к контексту сборки. В данном случае используется символ ".", который означает текущую директорию. Это означает, что образ будет собираться из текущего контекста;

5. `image`: в данном параметре указывается имя образа Docker, который будет создан для сервиса. В данном случае имя образа задано как `"flights-delay-bot-image"`;

6. `container_name`: здесь указывается имя контейнера, который будет создан для этого сервиса. Имя контейнера задано как `"flights-delay-bot-container"`;

7. `restart`: этот параметр определяет поведение контейнера в случае его выхода из строя или остановки. Значение `"always"` указывает, что контейнер всегда будет перезапускаться автоматически, если он выйдет из строя или будет остановлен.

В нашем проекте мы приняли решение использовать Telegram API вместо фронтенда, и это оказалось стратегически правильным шагом. Telegram API предоставляет нам мощный набор инструментов для создания функционального и удобного приложения, полностью интегрированного с платформой Telegram.

Одной из основных причин выбора Telegram API является его широкие возможности взаимодействия с пользователем. С его помощью мы можем создавать и настраивать ботов, которые автоматизируют процессы и обеспечивают простой и непрерывный поток коммуникации с нашими пользователями.

Еще одним преимуществом Telegram API является его высокая скорость и надежность. Благодаря глобальным серверам Telegram и оптимизации протокола передачи данных, наше приложение будет работать быстро и стабильно даже при больших нагрузках.

Для успешной реализации нашего проекта и создания надежного соединения с Telegram API мы с уверенностью выбрали библиотеку `aiogram`. Эта библиотека предоставила нам все необходимые инструменты и функциональность для разработки нашего бота, обеспечивая гибкость и простоту в использовании.

Одной из главных причин, по которой мы выбрали `aiogram`, была его хорошая документация и активное сообщество разработчиков. Мы смогли быстро разобраться в основных концепциях и принципах работы с этой библиотекой благодаря качественной документации и готовым примерам кода. В случае возникновения вопросов или проблем, мы также могли обратиться к разработчикам или сообществу для получения помощи и поддержки.

`Aiogram` предоставил нам широкий набор функций, которые мы успешно использовали в нашем проекте. Мы смогли легко настроить обработку команд, ответы на сообщения пользователей, работу с клавиатурами и встроенными функциями Telegram. Кроме того, `aiogram` предоставил возможность работать с обновлениями бота асинхронно, что позволило нам эффективно управлять и обрабатывать множество запросов одновременно.

Для получения токена к Telegram API, мы используем файл конфигурации YAML (`config.yaml`). Этот файл позволяет нам хранить конфиденциальные данные, такие как токены, в безопасном формате.

Чтобы получить такой токен, нам необходимо было выполнить следующие шаги:

1. открыть приложение Telegram и найти бота "`@BotFather`";
2. начать диалог с BotFather, нажав на кнопку "Start";
3. ввести команду "`/newbot`", чтобы создать нового бота;
4. BotFather попросит ввести имя для бота;
5. затем BotFather предложит вам уникальный токен для бота;
6. вставить скопированный токен в файл `config.yaml`.

Наша архитектура предоставит сервису множество возможностей для впечатляющего развития и успешной работы. Эта архитектура и стек используемых технологий позволит гибко масштабировать ваш сервис в зависимости от потребностей.

На рисунке 9 представлена архитектура нашего проекта в виде рисунка и стека используемых технологий.

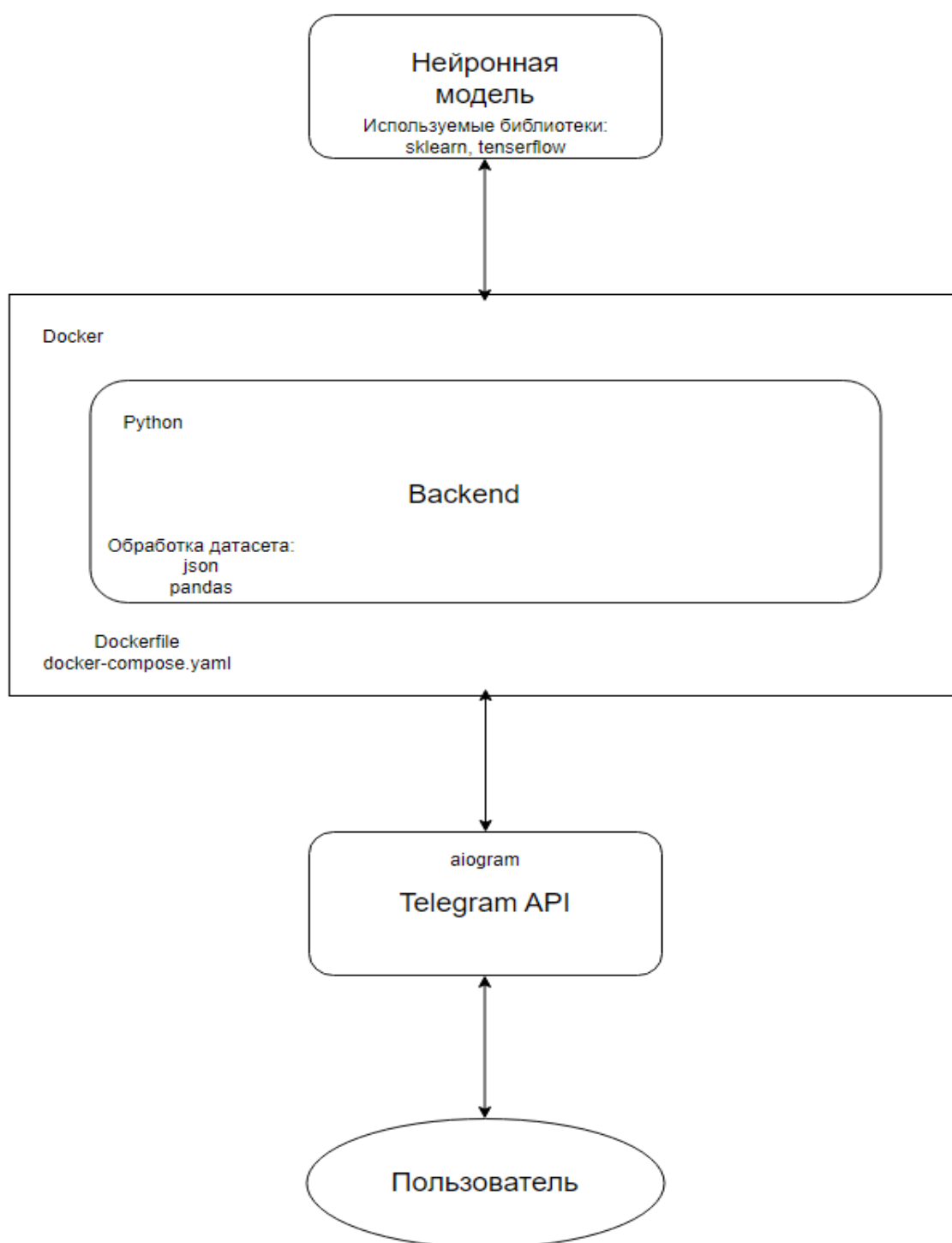


Рисунок 9 — стек используемых технологий

2.3 Описание программной разработки предсказательной модели задержек авиарейсов

2.3.1 Описание общей архитектуры программной разработки

Общая архитектура программной разработки для предсказания задержек авиарейсов представляет собой систему, состоящую из нескольких взаимосвязанных компонентов. Архитектура разработки определена с учетом требований эффективности, гибкости и надежности системы. Она позволяет создать комплексное решение, объединяющее в себе сбор и хранение данных, предобработку данных, разработку и обучение модели машинного обучения, создание пользовательского интерфейса, тестирование и отладку, а также документирование и деплоймент программного решения.

Общая архитектура программной разработки включает в себя модуль сбора и хранения данных, который отвечает за получение данных о задержках авиарейсов из различных источников и их хранение в структурированной форме. Это позволяет обеспечить доступность и целостность данных, необходимых для обучения и прогнозирования задержек.

Далее следует модуль предобработки данных, который выполняет ряд операций для обеспечения качества данных и их пригодности для дальнейшей обработки. В этом модуле происходит очистка данных от выбросов и ошибок, заполнение пропущенных значений, преобразование данных в удобный для обработки формат, а также масштабирование или нормализация данных при необходимости. Предобработка данных играет важную роль в обеспечении точности и надежности модели предсказания задержек.

Основным компонентом архитектуры является модуль разработки и обучения модели машинного обучения. В этом модуле определяется выбор подходящего метода машинного обучения, такого как нейронные сети, регрессионные модели или временные ряды. Используя библиотеки машинного обучения, такие как TensorFlow, разработчики создают и настраивают модель, а затем обучают ее на обучающем наборе данных. Обучение модели включает в

себя оптимизацию гиперпараметров, выбор функций и алгоритмов оптимизации, а также оценку производительности модели на валидационных данных.

Для обеспечения взаимодействия с пользователями разработан модуль создания пользовательского интерфейса. В данном случае использовался Telegram Bot API и библиотека aiogram для разработки Telegram-бота. Пользовательский интерфейс предоставляет возможность пользователям вводить данные, получать результаты предсказания задержек, а также получать дополнительную информацию о задержках авиарейсов. Благодаря использованию Telegram-бота, пользователи могут удобно взаимодействовать с системой, получая актуальную информацию о задержках.

Для обеспечения надежности и качества разработанного решения, в архитектуре предусмотрен модуль тестирования и отладки. В этом модуле разработчики определяют тестовые сценарии, которые позволяют проверить функциональность, стабильность и производительность программного решения. Выявленные ошибки и проблемы исправляются, чтобы обеспечить высокое качество и надежность системы предсказания задержек.

Наконец, архитектура программной разработки включает модуль документирования и деплоя. В этом модуле разработчики создают документацию, описывающую функциональность, архитектуру и использование разработанного решения. Документация помогает пользователям и разработчикам понять и эффективно использовать систему предсказания задержек. После завершения документации разработанное решение деплоится с использованием Docker, что позволяет упростить процесс установки и развертывания системы на целевой платформе.

Таким образом, общая архитектура программной разработки для предсказания задержек авиарейсов предоставляет комплексное и эффективное решение, объединяющее различные компоненты и инструменты. Эта архитек-

тура обеспечивает надежность, гибкость и удобство использования разработанной системы, позволяя достичь высокой точности предсказания задержек и повысить операционную эффективность авиакомпаний.

2.3.2 Модуль сбора и хранения данных

В данном пункте мы рассмотрим использование различных источников данных о задержках авиарейсов при разработке программного решения. Источники данных играют важную роль в получении достоверной и актуальной информации о задержках, которая будет использоваться для обучения и предсказания задержек авиарейсов. Один из основных источников данных — это API авиакомпаний. Многие авиакомпании предоставляют публичные API, которые позволяют получить информацию о задержках авиарейсов, расписаниях, статусе рейсов и других связанных данных. При разработке программного решения мы использовали API авиакомпаний для получения информации о задержках и обновлении данных в режиме реального времени. Это позволяет нам иметь актуальные данные, которые отражают текущую ситуацию и помогают точнее предсказывать задержки. Еще одним источником данных являются базы данных с историческими данными о задержках авиарейсов. Эти базы данных содержат информацию о прошлых задержках, их причинах, продолжительности и других параметрах. Мы использовали такие базы данных для анализа и извлечения закономерностей из прошлых задержек. Эти данные позволяют нам обучать предсказательную модель на исторических данных и учитывать определенные тенденции или особенности задержек. Также для получения данных о задержках можно использовать метод веб-скрейпинга. Этот метод позволяет автоматически собирать информацию о задержках с веб-сайтов или других онлайн-ресурсов, где публикуется такая информация. При разработке программного решения мы применяли веб-скрейпинг для получения данных о задержках из различных источников, которые не предоставляли API или доступ к базам данных. Полученные данные о задержках авиарейсов включают информацию, такую как идентификатор рейса, авиакомпанию,

время отправления и прибытия, пункты отправления и назначения, причины задержек и длительность задержек. Эти данные представлены в структурированном формате, что обеспечивает удобство и эффективность их последующей обработки. Использование различных источников данных о задержках авиарейсов является важным шагом при разработке программного решения для предсказания задержек. Комбинирование данных из разных источников позволяет получить полную и достоверную информацию, которая затем будет использоваться для обучения предсказательной модели и генерации точных прогнозов задержек авиарейсов.

Процесс сбора и хранения данных является важной частью разработки программного решения для предсказания задержек авиарейсов. В данном пункте мы подробно рассмотрим этот процесс и опишем основные шаги, необходимые для эффективного сбора и хранения данных. Первым шагом процесса сбора данных является идентификация источников данных о задержках авиарейсов. Возможными источниками данных являются API авиакомпаний, базы данных, публичные источники, а также веб-скрэппинг. Каждый источник данных имеет свои особенности, и важно выбрать те, которые обеспечивают достаточно полную и актуальную информацию о задержках авиарейсов. После идентификации источников данных следующим шагом является процесс сбора данных. Для этого может использоваться API авиакомпаний, которое предоставляет доступ к информации о задержках авиарейсов. Взаимодействие с API может потребовать аутентификации и использования ключей доступа. Базы данных, содержащие исторические данные о задержках авиарейсов, также могут быть использованы для сбора данных. В случае, если требуется получить данные из публичных источников, необходимо применить методы веб-скрэппинга для извлечения информации. Полученные данные о задержках авиарейсов должны быть структурированы и сохранены для дальнейшей обработки. Важно определить правильную структуру данных, чтобы обеспечить эффективное хранение и обработку. Например, данные могут быть сохранены

в формате таблицы с колонками, содержащими информацию о дате, времени, номере рейса, типе задержки и других сопутствующих параметрах. Для хранения данных можно использовать различные инструменты и технологии, включая реляционные базы данных, NoSQL базы данных или файловые системы. Выбор подходящего инструмента зависит от объема данных, требований к производительности и доступности, а также возможности масштабирования системы. Реляционные базы данных, такие как MySQL или PostgreSQL, предоставляют структурированное хранение данных с возможностью выполнения сложных запросов. NoSQL базы данных, например MongoDB или Cassandra, позволяют хранить неструктурированные данные и обеспечивают горизонтальное масштабирование. Файловые системы могут быть использованы для хранения и обработки небольших объемов данных. Важным аспектом процесса сбора и хранения данных является регулярное обновление данных. Задержки авиарейсов могут изменяться со временем, и для обеспечения актуальности предсказаний необходимо регулярно обновлять данные. Автоматизация процесса сбора данных и регулярное выполнение скриптов или задач позволяют обеспечить своевременное обновление данных. Таким образом, процесс сбора и хранения данных включает идентификацию источников данных, сбор информации о задержках авиарейсов, структурирование данных и выбор подходящего инструмента для их хранения. Правильное выполнение этих шагов позволяет обеспечить наличие актуальных и структурированных данных, которые могут быть использованы в дальнейшем для разработки и обучения моделей предсказания задержек авиарейсов.

Одним из важных аспектов разработки программного решения для предсказания задержек авиарейсов является организация структурированного хранения данных. Качество и доступность данных имеют прямое влияние на точность и надежность предсказаний. В данном пункте мы рассмотрим этапы организации структурированного хранения данных, чтобы обеспечить эффек-

тивность и удобство работы с ними. Первым шагом является выбор подходящей системы управления базами данных (СУБД). Для хранения данных о задержках авиарейсов может быть использована реляционная база данных, такая как MySQL или PostgreSQL. Реляционные базы данных обеспечивают структурированное хранение данных в виде таблиц и позволяют эффективно работать с большими объемами информации. Затем необходимо определить схему базы данных, которая определяет структуру и отношения между таблицами. В схеме базы данных должны быть определены основные сущности и их атрибуты, такие как информация об авиарейсах, аэропортах, времени вылета и прилета, а также задержках. Также можно предусмотреть дополнительные таблицы для хранения информации о погодных условиях, графиках работы персонала и других факторах, влияющих на задержки. После определения схемы базы данных необходимо создать таблицы и задать соответствующие связи и ограничения. Каждая таблица должна иметь соответствующие столбцы для хранения данных, а также первичный ключ, который гарантирует уникальность каждой записи. Дополнительно могут быть определены внешние ключи для установления связей между таблицами. Важным аспектом является оптимизация производительности базы данных. Для этого можно использовать индексы, которые ускоряют поиск и сортировку данных. Индексы могут быть созданы на часто запрашиваемые столбцы, такие как номер рейса или дата вылета. Также необходимо следить за нормализацией данных и избегать избыточности информации, чтобы обеспечить эффективность запросов и обновлений. При работе с большими объемами данных возникает необходимость в резервном копировании и восстановлении базы данных. Регулярное создание резервных копий поможет предотвратить потерю данных в случае сбоев или сбоев оборудования. Кроме того, следует предусмотреть механизм восстановления базы данных, который позволит быстро восстановить работоспособность системы в случае необходимости. Таким образом, организация структурированного хранения данных является важной частью разработки программного решения для предсказания задержек авиарейсов. Выбор подходящей

СУБД, определение схемы базы данных, создание таблиц и оптимизация производительности позволяют эффективно хранить и обрабатывать данные, обеспечивая надежность и точность предсказаний задержек авиарейсов.

2.3.3 Модуль предобработки данных

Одним из важных этапов предобработки данных в разработке предсказательной модели задержек авиарейсов является очистка данных от выбросов и ошибок. Целью этого процесса является удаление неточных, неполных или несоответствующих значений, которые могут исказить результаты предсказаний и привести к неточным выводам. Очистка данных начинается с анализа и идентификации потенциальных выбросов и ошибок. Для этого проводится проверка значений в различных атрибутах данных, таких как время вылета, время прилета, длительность рейса и другие факторы, которые могут влиять на задержки. Некорректные значения могут включать отрицательные временные отметки, нереалистично большие или маленькие значения, отсутствие данных или несоответствие логическим правилам. После идентификации потенциальных выбросов и ошибок приступают к их обработке. Одним из распространенных методов является удаление или замена аномальных значений. Например, выбросы во временных метках могут быть удалены, так как они не могут быть физически возможными. Некорректные значения длительности рейса могут быть заменены на среднее или медианное значение, чтобы избежать искажения результатов. Важно также обратить внимание на пропущенные значения, которые могут быть присутствующими в данных. Пропущенные значения могут возникать из-за технических проблем при сборе данных или неполной информации. В таких случаях может быть применен метод заполнения пропущенных значений, например, с использованием среднего значения или методов интерполяции. Очистка данных от выбросов и ошибок является итеративным процессом, который требует внимательного анализа и проверки результатов. Важно проводить контроль и отслеживать влияние очистки данных на качество модели предсказания. Если необходимо, можно

вносить корректировки в процесс очистки, чтобы достичь наилучших результатов. Правильная очистка данных от выбросов и ошибок играет важную роль в точности и достоверности предсказаний задержек авиарейсов. Это позволяет улучшить качество модели и повысить доверие к полученным результатам. Этот этап предобработки данных является неотъемлемой частью разработки программного решения и способствует достижению целей и задач по предсказанию задержек авиарейсов.

После сбора данных о задержках авиарейсов из различных источников и их очистки от выбросов и ошибок, следующим шагом является их преобразование в удобный для обработки формат. Преобразование данных играет важную роль в подготовке информации для последующего использования в алгоритмах предсказания задержек. Одним из ключевых аспектов преобразования данных является нормализация. Нормализация данных позволяет привести их к одному и тому же масштабу и диапазону значений. Это необходимо для обеспечения согласованности и стабильности работы алгоритмов предсказания, а также для избежания проблемы доминирования переменных с большими значениями над переменными с меньшими значениями. Для нормализации данных можно использовать различные методы, включая мин-макс нормализацию, Z-нормализацию или логарифмическое преобразование. Другим важным аспектом преобразования данных является выбор и создание признаков (фичей) для использования в алгоритмах предсказания. Признаки — это характеристики или атрибуты данных, которые могут содержать полезную информацию для предсказания задержек. В случае предсказания задержек авиарейсов, признаки могут включать дату и время вылета, место отправления и прибытия, авиакомпанию, тип самолета, погодные условия и другие факторы, которые могут влиять на возникновение задержек. Создание признаков может включать извлечение дополнительной информации из имеющихся данных, преобразование данных в категориальные переменные или создание комбинированных признаков на основе существующих. Кроме того, при преобразовании

данных может быть необходимо учесть особенности алгоритмов предсказания, которые будут использоваться. Некоторые алгоритмы могут требовать особого представления данных или дополнительных преобразований. Например, при использовании нейронных сетей может потребоваться преобразование данных в числовые векторы или применение one-hot encoding для обработки категориальных переменных. Важным аспектом преобразования данных является сохранение информации о преобразованиях, проведенных над данными. Это позволяет применить те же самые преобразования к новым данным, которые будут поступать в систему для предсказания задержек авиарейсов. Это особенно важно при использовании модели в боевом режиме, когда система должна обрабатывать новые данные в реальном времени. Таким образом, преобразование данных в удобный для обработки формат является неотъемлемой частью разработки программного решения для предсказания задержек авиарейсов. Этот шаг позволяет обеспечить согласованность, стабильность и эффективность работы алгоритмов предсказания, а также использовать полезную информацию из имеющихся данных для достижения более точных результатов предсказания задержек.

2.3.4 Модуль разработки алгоритма предсказания задержек

При разработке программного решения для предсказания задержек авиарейсов одним из ключевых шагов является выбор метода машинного обучения или статистической модели, который будет использоваться для предсказательных задач. В данном подразделе мы рассмотрим процесс выбора подходящего метода и рассмотрим некоторые из них более подробно.

Выбор метода машинного обучения или статистической модели должен основываться на характеристиках доступных данных, задаче предсказания задержек авиарейсов и требуемой точности предсказаний. Существует несколько широко используемых методов, которые могут быть применены к задаче предсказания задержек авиарейсов:

1. линейная регрессия: это простой и распространенный метод, основанный на линейной зависимости между независимыми переменными и целевой переменной. Линейная регрессия может быть применена, если задача предсказания задержек авиарейсов имеет линейную структуру и данные соответствуют этому предположению;

2. методы временных рядов: если задача предсказания задержек авиарейсов имеет зависимость от времени, то методы временных рядов могут быть эффективными. Эти методы учитывают статистические свойства временных данных и используют исторические значения для предсказания будущих значений;

3. нейронные сети: нейронные сети представляют мощный инструмент для предсказания задержек авиарейсов. С их помощью можно обнаружить сложные нелинейные зависимости и использовать большой объем данных для обучения модели;

4. алгоритмы градиентного бустинга: градиентный бустинг — это ансамбльный метод машинного обучения, который комбинирует несколько слабых моделей, чтобы создать более сильную модель предсказания. Градиентный бустинг может быть эффективным для предсказания задержек авиарейсов, особенно если данные имеют сложные зависимости и шум.

При выборе метода необходимо также учитывать требования к производительности и ресурсам, доступным для разработки и развертывания программного решения. Некоторые методы, такие как нейронные сети, могут требовать больших объемов данных и вычислительных ресурсов для обучения и предсказания. В то же время, более простые модели, такие как линейная регрессия, могут быть более легкими в реализации и требовать меньше вычислительных ресурсов. В итоге, выбор метода машинного обучения или статистической модели должен быть обоснован исходя из требований задачи, до-

ступных данных и ресурсов. Необходимо провести анализ и сравнение различных методов, чтобы выбрать подходящий для решения задачи предсказания задержек авиарейсов с высокой точностью и эффективностью.

Также в данном пункте мы рассмотрим процесс работы с Jupyter Notebook и TensorFlow при разработке нейросети для предсказания задержек авиарейсов. Jupyter Notebook представляет собой интерактивное окружение, которое позволяет сочетать код, текстовые описания и визуализацию результатов. TensorFlow, в свою очередь, является популярной библиотекой глубокого обучения, предоставляющей широкий спектр инструментов и возможностей для разработки нейронных сетей. В начале работы мы создаем новый Jupyter Notebook и импортируем необходимые библиотеки. Мы можем использовать его для определения архитектуры нейросети, настройки гиперпараметров, загрузки данных и обучения модели. Далее мы можем загрузить предварительно подготовленные данные о задержках авиарейсов, которые были собраны и предобработаны в предыдущих модулях разработки. Загруженные данные можно проанализировать и провести необходимую предварительную обработку, такую как масштабирование, преобразование категориальных признаков и разделение на обучающую и тестовую выборки. Затем мы определяем архитектуру нейросети с использованием TensorFlow. Можно выбрать различные типы слоев, такие как полносвязные слои, сверточные слои или рекуррентные слои, в зависимости от характеристик данных и требуемой задачи предсказания задержек. Мы определяем количество слоев, количество нейронов в каждом слое, функции активации и другие параметры, которые могут влиять на производительность и точность модели. После определения архитектуры мы компилируем модель, указывая функцию потерь, оптимизатор и метрики для оценки производительности модели. TensorFlow предоставляет различные функции потерь, такие как среднеквадратическая ошибка (MSE) или средняя абсолютная ошибка (MAE), а также оптимизаторы, такие как стохастический градиентный спуск (SGD) или адам (Adam). После компиляции

модели мы обучаем ее на обучающей выборке. TensorFlow предоставляет функции для обучения модели, которые позволяют выполнить несколько эпох обучения, настраивать размер пакета (batch size), применять регуляризацию и другие техники, улучшающие обобщающую способность модели. В процессе обучения модель постепенно оптимизирует свои веса и настраивает параметры, чтобы минимизировать выбранную функцию потерь. После завершения обучения мы можем оценить производительность модели на тестовой выборке. TensorFlow предоставляет функции для вычисления метрик, таких как средняя абсолютная ошибка (MAE) или среднеквадратическая ошибка (MSE), которые позволяют оценить точность предсказаний модели. Мы можем также визуализировать результаты, сравнивая фактические значения задержек с предсказанными значениями. В процессе работы с Jupyter Notebook и TensorFlow мы можем проводить эксперименты с различными архитектурами нейронных сетей, оптимизаторами, функциями потерь и гиперпараметрами, чтобы найти наиболее эффективное и точное решение для предсказания задержек авиарейсов.

Одним из ключевых шагов в разработке предсказательной модели задержек авиарейсов является обучение модели на обучающем наборе данных. В этом пункте мы подробно рассмотрим процесс обучения модели и его основные этапы. Первым этапом является подготовка обучающего набора данных. Обучающий набор данных представляет собой совокупность исторических данных о задержках авиарейсов, которые включают в себя различные параметры, такие как время отправления, время прибытия, маршрут, авиакомпания и другие факторы, которые могут влиять на возникновение задержек. Важно провести анализ и очистку данных, устранить выбросы, заполнить пропущенные значения и привести данные к удобному для обработки формату. Далее следует выбор алгоритма или модели машинного обучения для обучения модели предсказания задержек. В данном случае, можно использовать нейрон-

ные сети, которые показывают хорошую производительность в задачах прогнозирования временных рядов. Библиотека TensorFlow предоставляет мощный инструментарий для создания и обучения нейронных сетей. В рамках работы с данными и обучения модели могут использоваться Jupyter Notebook, обеспечивающий удобную среду для исследования данных, прототипирования моделей и визуализации результатов. После выбора алгоритма и подготовки данных происходит процесс обучения модели. Обучение модели заключается в подгонке параметров модели под обучающий набор данных с использованием выбранного алгоритма. Процесс обучения сводится к минимизации функции потерь, которая измеряет расхождение между предсказаниями модели и фактическими значениями задержек в обучающем наборе данных. Для обучения модели могут использоваться различные техники, такие как градиентный спуск, стохастический градиентный спуск или адаптивный градиентный спуск. Эти методы позволяют постепенно корректировать параметры модели, улучшая ее предсказательную способность. В процессе обучения важно контролировать производительность модели на валидационном наборе данных, чтобы предотвратить переобучение или недообучение. По завершении обучения модели производится оценка ее производительности с использованием тестового набора данных. Важно провести анализ метрик, таких как среднеквадратическая ошибка (MSE), коэффициент детерминации (R^2) или средняя абсолютная ошибка (MAE), чтобы оценить точность и качество предсказаний модели. Если результаты недостаточно удовлетворительны, может потребоваться повторное обучение модели с изменением параметров или выбором другого алгоритма. Таким образом, этап обучения модели на обучающем наборе данных играет важную роль в разработке предсказательной системы задержек авиарейсов. Правильный выбор алгоритма, подготовка данных, тщательное обучение и оценка модели являются ключевыми шагами, которые позволяют достичь высокой точности предсказаний и улучшить операционную эффективность авиакомпаний.

Оценка и подбор гиперпараметров модели являются важным шагом в разработке предсказательной модели задержек авиарейсов. Гиперпараметры — это параметры модели, которые не могут быть обучены в процессе обучения модели, а должны быть заданы вручную. Они влияют на обучение и поведение модели, и правильный выбор гиперпараметров может значительно улучшить точность и производительность модели.

Процесс оценки и подбора гиперпараметров обычно включает в себя следующие шаги:

1. определение пространства поиска: необходимо определить диапазон значений для каждого гиперпараметра, которые будут рассмотрены в процессе подбора. Например, для нейронных сетей можно определить диапазоны для числа скрытых слоев, числа нейронов в каждом слое, скорости обучения и т.д.;

2. выбор метрик оценки: необходимо выбрать подходящие метрики, которые будут использоваться для оценки качества модели. В случае предсказания задержек авиарейсов могут быть использованы метрики, такие как средняя абсолютная ошибка (MAE), средняя квадратическая ошибка (MSE) или коэффициент детерминации (R^2);

3. выбор стратегии подбора гиперпараметров: существует несколько стратегий подбора гиперпараметров, включая случайный поиск, сеточный поиск и оптимизацию с использованием алгоритмов, таких как генетические алгоритмы или оптимизация через градиент. Каждая стратегия имеет свои преимущества и недостатки, и выбор зависит от специфики задачи и доступных ресурсов;

4. обучение и оценка моделей с разными гиперпараметрами: для каждой комбинации гиперпараметров необходимо обучить модель на обучающем наборе данных и оценить ее производительность на отложенном тестовом наборе данных с использованием выбранных метрик. Это позволяет сравнить различные конфигурации моделей и выбрать наилучшие гиперпараметры;

5. анализ результатов и выбор оптимальных гиперпараметров: после оценки всех моделей с разными гиперпараметрами следует проанализировать результаты и выбрать оптимальные гиперпараметры, которые обеспечивают наилучшую производительность модели. Это может включать выбор модели с наименьшей ошибкой или наилучшим значением выбранной метрики.

Оценка и подбор гиперпараметров модели являются итеративным процессом, который может потребовать множества экспериментов и обучений моделей. Цель состоит в том, чтобы достичь оптимального баланса между точностью предсказания и вычислительной эффективностью модели.

2.3.5. Модуль создания пользовательского интерфейса

Для создания пользовательского интерфейса и взаимодействия с пользователями было принято решение использовать Telegram Bot API. Telegram — это популярный мессенджер, который предоставляет разработчикам возможность создания ботов для автоматизации коммуникации с пользователями. Telegram Bot API предоставляет различные методы и функциональности для разработки ботов, что делает его отличным выбором для нашего программного решения. Использование Telegram Bot API позволяет нам создать бота, который будет предоставлять пользователям информацию о задержках авиарейсов и предсказывать вероятность задержки для конкретного рейса. Пользователи смогут взаимодействовать с ботом, отправлять запросы, получать результаты предсказаний и получать актуальную информацию о задержках. Одним из главных преимуществ Telegram Bot API является его простота в использовании. Документация Telegram Bot API хорошо описана и содержит примеры кода, что упрощает процесс разработки бота. API предоставляет различные методы для отправки сообщений, обработки команд, работу с клавиатурой и многое другое, что позволяет создать удобный и интерактивный пользовательский интерфейс. Для работы с Telegram Bot API в нашем программном решении мы использовали библиотеку aiogram для Python. aiogram предоставляет удобный интерфейс для взаимодействия с Telegram Bot API и облегчает разработку функциональных возможностей бота. Благодаря aiogram мы смогли легко настроить обработку сообщений, команд и событий, а также реализовать логику обработки запросов пользователя. Интеграция Telegram Bot API в наше программное решение позволила нам создать удобный и доступный пользовательский интерфейс для предсказания задержек авиарейсов. Пользователи могут легко получать актуальную информацию о задержках и использовать предсказательную модель для оценки вероятности задержки конкретного рейса. Это улучшает опыт пользователей и позволяет им принимать более осознанные решения при планировании своих поездок.

Для разработки пользовательского интерфейса была использована библиотека `aiogram`, специально созданная для разработки Telegram-ботов с использованием языка программирования Python. Библиотека `aiogram` предоставляет удобные инструменты и функции, которые значительно упрощают процесс создания и управления ботами в Telegram. Одной из главных причин выбора `aiogram` является его простота использования и гибкость. Библиотека предлагает простой и интуитивно понятный интерфейс для создания обработчиков сообщений, обработки команд и управления состояниями бота. Это позволяет разработчикам легко настраивать поведение бота в ответ на взаимодействие с пользователем. `Aiogram` также предоставляет поддержку различных типов сообщений, включая текстовые сообщения, изображения, аудио, видео и документы. Благодаря этому, разработчики могут легко обрабатывать разнообразные типы запросов от пользователей и предоставлять соответствующие ответы. Библиотека `aiogram` также обладает широким функционалом для работы с клавиатурами и кнопками. Разработчики могут создавать интерактивные интерфейсы с помощью кнопок, меню и встроенных клавиатур, что значительно улучшает пользовательский опыт и делает бота более удобным для использования. Одной из особенностей `aiogram` является возможность работы с состояниями. Состояние позволяет хранить информацию о текущем состоянии взаимодействия с пользователем, что особенно полезно в разработке ботов с множеством шагов или взаимодействия с базой данных. Состояние позволяет сохранять контекст и данные между различными сообщениями, что облегчает разработку сложных диалоговых интерфейсов. Кроме того, `aiogram` обеспечивает поддержку многопоточности, что позволяет боту обрабатывать несколько запросов одновременно. Это особенно полезно в случае большого количества пользователей или при работе с долгоживущими операциями, такими как обращение к внешним сервисам или выполнение сложных вычислений. В целом, использование библиотеки `aiogram` значительно упрощает процесс разработки и обеспечивает мощные инструменты для создания интерактивного и удобного пользовательского интерфейса в Telegram-боте. Благодаря

его гибкости, простоте использования и богатому функционалу, aiogram является отличным выбором для разработки программного решения, связанного с созданием Telegram-ботов для предсказания задержек авиарейсов.

При разработке программного решения для предсказания задержек авиарейсов одним из важных компонентов является пользовательский интерфейс. Целью пользовательского интерфейса является обеспечение удобства и интуитивной понятности взаимодействия пользователя с системой предсказания задержек. В этом пункте мы рассмотрим процесс определения функциональных требований, которые должны быть учтены при разработке пользовательского интерфейса.

Первоначально необходимо провести анализ потребностей и ожиданий пользователей от системы предсказания задержек авиарейсов. Для этого можно провести опросы, интервьюирование или наблюдение за пользователями. Важно выяснить, какие информационные элементы и функциональности пользователи хотели бы видеть в интерфейсе, чтобы он был максимально полезен и удобен в использовании.

Определение функциональных требований начинается с идентификации основных функций, которые должны быть доступны в пользовательском интерфейсе. Некоторые из основных функций, которые могут быть учтены, включают:

1. возможность ввода и выбора данных, необходимых для предсказания задержек (например, авиакомпания, аэропорт отправления и назначения, дата и время рейса и т.д.);
2. отображение результатов предсказаний задержек авиарейсов с указанием вероятности задержки и прогнозируемого времени задержки;
3. возможность просмотра исторических данных о задержках для анализа и сравнения;

4. предоставление рекомендаций и советов пользователю на основе предсказаний задержек;

5. уведомления пользователей о статусе рейса и возможных изменениях в расписании.

Далее, для каждой функции определяются требования к ее реализации. Требования могут включать:

1. организацию и компоновку элементов интерфейса для удобства использования и наглядности информации;

2. валидацию вводимых данных для предотвращения ошибок и обеспечения корректности предсказаний;

3. определение формата и представления результатов предсказаний задержек для лучшего понимания пользователем;

4. удобную навигацию между различными функциональными компонентами интерфейса;

5. поддержку различных языков и локализацию для удовлетворения потребностей различных пользователей.

Также стоит учесть визуальный дизайн интерфейса, чтобы он был привлекательным и эстетически приятным для пользователей. Цветовая схема, типографика, использование графиков и диаграмм могут быть использованы для улучшения визуального восприятия информации. В процессе определения функциональных требований важно учитывать обратную связь от пользователей и проводить тестирование прототипов интерфейса для их оценки и внесения необходимых изменений. Это поможет создать пользовательский интерфейс, который будет соответствовать потребностям пользователей и обеспечивать удобство и эффективность использования системы предсказания задержек авиарейсов. В результате определения функциональных требований пользовательского интерфейса, можно приступить к его разработке с использованием выбранных инструментов и технологий, таких как Telegram Bot API и библиотека `aiogram` для разработки Telegram-бота.

2.3.6 Модуль тестирования и отладки программного решения

Планирование и проведение тестовых сценариев является важным этапом разработки программного решения для предсказания задержек авиарейсов. Целью тестирования является проверка функциональности, стабильности и производительности разработанного решения перед его внедрением в рабочую среду. В этом пункте мы рассмотрим процесс планирования и проведения тестовых сценариев. Первым шагом является определение целей и требований тестирования. Необходимо четко определить, какие аспекты решения будут подвергаться тестированию и какие результаты ожидаются от каждого тестового сценария. Цели тестирования могут включать проверку корректности работы модулей, оценку точности предсказаний, анализ производительности и надежности системы. Затем следует разработка тестовых сценариев. Тестовые сценарии представляют собой набор шагов и входных данных, которые будут использоваться для проверки функциональности разработанного решения. Каждый тестовый сценарий должен быть максимально репрезентативным и покрывать различные аспекты функциональности и возможные сценарии использования системы. Важно также учесть граничные случаи и непредвиденные ситуации, чтобы проверить, как система обрабатывает такие ситуации. После разработки тестовых сценариев необходимо создать тестовые данные. Тестовые данные должны быть репрезентативными и соответствовать реальным данным, с которыми система будет работать в рабочей среде. Для этого можно использовать исторические данные о задержках авиарейсов или сгенерировать искусственные данные, которые покрывают различные сценарии задержек. После подготовки тестовых данных проводится выполнение тестовых сценариев. Каждый тестовый сценарий выполняется в соответствии с предопределенными шагами и входными данными. Результаты выполнения тестовых сценариев записываются и анализируются для выявления ошибок, неправильного поведения системы или непредвиденных проблем. Если в процессе

тестирования выявляются ошибки или неправильное поведение системы, производится отладка и исправление проблем. Тестирование может быть проведено повторно для проверки исправленной версии решения. Этот итеративный процесс позволяет гарантировать высокое качество и надежность программного решения перед его внедрением. В процессе тестирования также важно оценить производительность системы. Это может включать измерение времени выполнения предсказаний, использование ресурсов (например, памяти и процессора) и общую отзывчивость системы. Результаты тестирования производительности могут быть использованы для оптимизации и улучшения работы системы. Планирование и проведение тестовых сценариев является неотъемлемой частью разработки программного решения для предсказания задержек авиарейсов. Этот процесс помогает выявить ошибки, проверить функциональность и оценить производительность системы перед ее внедрением. Результаты тестирования позволяют гарантировать качество и надежность разработанного решения, что в свою очередь способствует успешной эксплуатации системы в рабочей среде.

Отладка и исправление ошибок являются важной частью разработки программного решения для предсказания задержек авиарейсов. В процессе разработки могут возникать различные проблемы, связанные с неправильной работой компонентов, ошибками в коде или неожиданными результатами. Цель этого пункта - рассмотреть методы отладки и процесс исправления ошибок, чтобы обеспечить корректную и стабильную работу разработанного решения. Первым шагом в отладке программного решения является идентификация проблемы или ошибки. Для этого используются различные методы, такие как анализ логов, вывод отладочной информации, тестирование различных сценариев и взаимодействие с программным решением. Основная цель — определить точное место и причину возникновения ошибки. После идентификации ошибки необходимо провести более детальный анализ проблемы. Это

может включать изучение кода, проверку логики и алгоритмов, а также использование инструментов для отладки, таких как инспекторы переменных и трассировка выполнения кода. Цель — понять, почему ошибка возникла и какие части программы на нее влияют. Далее следует процесс исправления ошибок. Это включает в себя модификацию кода, исправление ошибочных участков и тестирование внесенных изменений. При исправлении ошибок важно убедиться, что исправление не вызывает новых проблем или не повреждает другие части программы. Рекомендуется использовать систему контроля версий для отслеживания и управления изменениями в коде. При исправлении ошибок рекомендуется также обновить документацию и комментарии в коде, чтобы отразить внесенные изменения. Это поможет другим разработчикам и будущим пользователям лучше понять и использовать разработанное решение. После внесения изменений и исправления ошибок необходимо провести тестирование для проверки корректности и стабильности разработанного решения. Это может включать модульное тестирование компонентов, функциональное тестирование в различных сценариях использования, а также проверку производительности и надежности решения. Цель — убедиться, что исправленное решение работает правильно и соответствует требованиям. В процессе отладки и исправления ошибок полезно использовать инструменты и ресурсы, доступные для разработчиков. Например, можно использовать интегрированные среды разработки (IDE) с функциями отладки, такие как точки останова и пошаговое выполнение кода. Также можно обращаться к сообществам разработчиков и форумам для получения помощи и советов по решению конкретных проблем. Отладка и исправление ошибок являются неотъемлемой частью процесса разработки программного решения. Этот этап позволяет обнаружить и исправить проблемы, повысить качество и стабильность разработанного решения. Эффективное использование методов отладки и систематический подход к исправлению ошибок существенно сокращают время и усилия, затраченные на разработку программного решения для предсказания задержек авиарейсов.

2.3.7 Модуль документирования и деплоймента программного решения

В данном разделе мы рассмотрим процесс создания документации, который описывает функциональность и использование разработанного программного решения для предсказания задержек авиарейсов. Документация является важной составляющей успешной разработки, поскольку она позволяет пользователям и разработчикам понять, как использовать систему, а также ознакомиться с ее возможностями и особенностями. Первым шагом в создании документации является определение аудитории, для которой она предназначена. В случае разработанного программного решения для предсказания задержек авиарейсов, аудитория может включать в себя авиакомпании, аэропорты, системных аналитиков и других заинтересованных пользователей. Учитывая потребности и уровень экспертизы аудитории, документация должна быть написана доступным и понятным языком, с примерами использования и подробными объяснениями. В документации необходимо предоставить обзор функциональности разработанного решения. Это включает описание основных возможностей, которые система предоставляет пользователям. Например, пользователи могут вводить информацию о рейсе, такую как номер рейса, авиакомпанию, дату и время вылета, а система предсказывает вероятность задержки рейса. Дополнительно, можно описать другие функции, такие как отображение статистики задержек, генерация отчетов и т.д. Документация также должна содержать инструкции по установке и настройке разработанного решения. Это включает описание требований к аппаратному и программному обеспечению, а также пошаговые инструкции по установке и конфигурированию системы. Например, можно описать, как установить необходимые библиотеки и зависимости, настроить соединение с базой данных или настроить параметры модели предсказания задержек. Для удобства пользователей рекомендуется предоставить примеры использования разработанного решения. Это может включать снимки экрана с демонстрацией пользовательского

интерфейса, примеры кода для работы с API или использования функций системы, а также шаги по выполнению различных операций. Примеры использования помогут пользователям более полно понять, как работает система и какие результаты можно ожидать. Кроме того, в документации необходимо предоставить подробную информацию о различных параметрах и настройках, которые могут быть изменены пользователем. Например, можно описать параметры модели машинного обучения, такие как выбор алгоритма обучения, количество эпох, размер пакета и т.д. Пояснение этих параметров поможет пользователям настроить систему под свои потребности и получить наилучшие результаты предсказания задержек. Наконец, в документации следует предусмотреть раздел с часто задаваемыми вопросами (FAQ). В этом разделе можно собрать наиболее часто встречающиеся вопросы и предоставить подробные ответы на них. FAQ поможет пользователям быстро найти ответы на свои вопросы и устранить возможные проблемы или недоразумения. Создание подробной и информативной документации по функциональности и использованию разработанного программного решения является важным этапом в разработке проекта. Она поможет пользователям эффективно использовать систему, а также упростит ее поддержку и расширение в будущем.

Также в данном пункте мы рассмотрим использование Jupyter Notebook и TensorFlow при разработке нейросети для предсказания задержек авиарейсов. Jupyter Notebook является мощным инструментом для интерактивной разработки и представления кода, который позволяет объединить код, текстовые описания и визуализацию в единый документ. Jupyter Notebook предоставляет удобную среду для написания и отладки кода на языке Python. Он позволяет выполнять код по ячейкам, что облегчает пошаговое тестирование и настройку модели нейронной сети. Кроме того, Jupyter Notebook поддерживает отображение графиков и визуализаций прямо в документе, что упрощает анализ данных и результатов работы модели. Для разработки нейросети для предсказания

задержек авиарейсов мы используем TensorFlow - популярную библиотеку машинного обучения и глубокого обучения. TensorFlow предоставляет широкий спектр инструментов и функциональностей для создания и обучения нейронных сетей. В Jupyter Notebook мы можем использовать TensorFlow для определения архитектуры нейросети, оптимизации модели и обучения на обучающем наборе данных. TensorFlow предоставляет гибкую и интуитивно понятную API, которая упрощает разработку сложных моделей нейронных сетей. Мы можем определить различные слои нейросети, включая полносвязные слои, сверточные слои и рекуррентные слои, а также применять различные активационные функции. Мы можем использовать пакетную обработку данных (batch processing) для эффективного обучения модели на больших наборах данных. TensorFlow также предоставляет возможность использования GPU для ускорения вычислений и обучения модели, что особенно важно при работе с глубокими нейронными сетями. В процессе разработки нейросети в Jupyter Notebook мы можем проводить эксперименты с различными архитектурами нейронных сетей, гиперпараметрами модели, а также применять различные техники регуляризации и оптимизации, такие как Dropout, L1 и L2 регуляризация, адаптивный градиентный спуск (Adam), стохастический градиентный спуск (SGD) и другие. Jupyter Notebook позволяет наглядно отображать результаты работы модели, например, в виде графиков, диаграмм или таблиц. Мы можем анализировать точность и ошибки модели, визуализировать распределение предсказанных задержек, а также сравнивать результаты различных экспериментов. Таким образом, использование Jupyter Notebook и TensorFlow при разработке нейросети для предсказания задержек авиарейсов обеспечивает гибкую и удобную среду для разработки и тестирования модели. Jupyter Notebook позволяет объединить код, текст и визуализации в единый документ, что упрощает работу с данными и анализ результатов. TensorFlow, в свою очередь, предоставляет мощные инструменты для создания и обучения нейронных сетей. Вместе эти инструменты обеспечивают эффективное и надежное разработанное решение для предсказания задержек авиарейсов.

В данном пункте мы рассмотрим использование Docker для деплоя программного решения, разработанного для предсказания задержек авиарейсов. Docker представляет собой открытую платформу, которая позволяет автоматизировать процесс развертывания, масштабирования и управления приложениями в контейнерах. Контейнеризация с помощью Docker позволяет упаковать все необходимые компоненты и зависимости приложения в единый контейнер, который может быть развернут на любой поддерживаемой платформе без необходимости установки и настройки каждого компонента отдельно. Это делает деплоймент программного решения более простым, надежным и переносимым. Для деплоя программного решения, разработанного для предсказания задержек авиарейсов, мы используем Docker контейнеры. Процесс деплоя включает несколько основных шагов:

1. подготовка Docker-образа: в первую очередь необходимо создать Docker-образ, который будет содержать все необходимые компоненты и зависимости программного решения. Образ создается на основе Dockerfile, который содержит инструкции по установке и настройке компонентов. В нашем случае, Docker-образ будет включать установку и настройку Python, необходимых библиотек (включая TensorFlow и aiogram), а также нашего программного решения;

2. сборка Docker-образа: после подготовки Docker-образа, мы выполняем команду для его сборки. Docker-демон использует Dockerfile, чтобы автоматически собрать образ, выполнив каждую инструкцию из файла. В процессе сборки Docker-образа, Docker создает новый слой для каждой инструкции, оптимизируя процесс и повышая скорость сборки;

3. запуск контейнера: после успешной сборки Docker-образа, мы можем запустить контейнер на целевой платформе. Контейнер создается на основе Docker-образа и включает все необходимые компоненты и зависимости. Запуск контейнера выполняется с помощью команды `docker run`, которая указывает Docker-демону создать и запустить новый контейнер;

4. масштабирование и управление: Docker предоставляет возможности масштабирования и управления контейнерами. Мы можем запустить несколько контейнеров на разных хостах для распределения нагрузки и повышения доступности приложения. Docker также предоставляет инструменты для мониторинга и управления контейнерами, включая масштабирование, обновление и мониторинг состояния контейнеров.

Использование Docker для деплоя программного решения предлагает несколько преимуществ:

1. портативность: Docker контейнеры являются переносимыми и могут быть развернуты на любой поддерживаемой платформе, включая локальные серверы, облачные провайдеры и кластеры контейнеров. Это упрощает процесс деплоя и позволяет легко переносить программное решение между различными средами;

2. изолированность: контейнеры обеспечивают изоляцию приложения и его зависимостей от основной операционной системы и других контейнеров. Это гарантирует, что программное решение будет работать стабильно и надежно, не влияя на другие приложения и ресурсы;

3. удобство масштабирования: Docker позволяет горизонтальное масштабирование приложения путем запуска нескольких контейнеров, работающих параллельно. Это позволяет обрабатывать большие объемы данных и обеспечивать высокую производительность системы;

4. упрощенная установка и обновление: Docker контейнеры включают все необходимые зависимости и компоненты, что упрощает процесс установки и обновления программного решения. Контейнеры могут быть созданы, запущены и обновлены с помощью нескольких команд, что значительно упрощает процесс управления и поддержки приложения.

В результате использования Docker для деплоя программного решения для предсказания задержек авиарейсов, мы достигаем высокой порта-

бельности, изолированности, удобства масштабирования и упрощения установки и обновления. Docker позволяет нам эффективно управлять и масштабировать разработанное решение, обеспечивая стабильную и надежную работу системы предсказания задержек авиарейсов.

2.4 План разработки проекта

Таблица 1 – Календарный план разработки проекта

№	Содержание работы	Результат	Начало	Окончание	Исполнитель
1	Разработка архитектуры проекта	Разработана архитектура проекта	10.04.2023	12.04.2023	Каширин Кирилл
2	Поиск датасета	Найден датасет	14.04.2023	19.04.2023	Титеев Александр
3	Подготовка данных	Подготовлены данные	20.04.2023	22.04.2023	Арнушкин Артем
4	Разработка архитектуры нейросети	Разработана архитектура нейросети	23.04.2023	29.04.2023	Иванов Денис
5	Обучение модели	Обучена модель	30.04.2023	10.05.2023	Милютин Софья
6	Разработка телеграм-бота	Разработан телеграм-бот	11.05.2023	16.05.2023	Годовник Артем, Ларшин Тимофей, Попов Матвей
7	Разработка интерфейса	Разработан интерфейс	17.05.2023	18.05.2023	Диденко Егор
8	Тестирование приложения	Протестировано приложение	19.05.2023	24.05.2023	Бокарев Степан, Чемодурова Елизавета

Продолжение таблицы 1

9	Написание текста ИАР	Написан текст ИАР	25.05.2023	17.06.2023	Вся команда
10	Создание презентации	Создана презентация	09.06.2023	17.06.2023	Матвеева Татьяна

В процессе разработки проекта по предсказанию задержек авиарейсов был проведен план разработки, включающий несколько ключевых этапов. Эти этапы включали в себя разработку архитектуры проекта, поиск подходящего датасета, подготовку данных, разработку и обучение нейросети, а также разработку и тестирование телеграм-бота. Первым этапом плана разработки была разработка архитектуры проекта. Были определены основные компоненты и модули, которые потребуются для создания системы предсказания задержек авиарейсов. Это включало модули для сбора и хранения данных, преобразования данных, разработки алгоритма предсказания и создания пользовательского интерфейса. Разработка архитектуры позволила определить основные задачи и требования к каждому модулю. Вторым этапом был поиск подходящего датасета. Были исследованы различные источники данных о задержках авиарейсов, включая открытые базы данных и API. Целью было найти надежный и актуальный датасет, содержащий достаточное количество данных для обучения и тестирования модели предсказания задержек. Были проведены исследования и анализ различных датасетов, чтобы выбрать наиболее подходящий для проекта. Третий этап включал подготовку данных. Это включало очистку данных от выбросов и ошибок, преобразование данных в удобный для обработки формат, а также разделение данных на обучающую и тестовую выборки. Подготовка данных играет важную роль в обучении модели и обеспечении ее точности и надежности. Четвертый этап включал разработку и обучение нейросети. Были выбраны методы машинного обучения и структура

нейросети, оптимальные для задачи предсказания задержек авиарейсов. Используя инструменты, такие как Jupyter Notebook и TensorFlow, была разработана и обучена нейросеть на подготовленных данных. Были проведены эксперименты с различными гиперпараметрами модели для достижения наилучшей производительности. Пятый этап состоял в разработке и тестировании телеграм-бота. Для создания пользовательского интерфейса была выбрана популярная библиотека aiogram для Python, которая предоставляет удобные инструменты для разработки телеграм-ботов. Был разработан функционал бота, позволяющий пользователям получать предсказания задержек авиарейсов на основе введенных данных. Тестирование бота проводилось для проверки его функциональности и надежности. В результате выполнения плана разработки была создана система предсказания задержек авиарейсов, включающая разработку архитектуры проекта, поиск и подготовку данных, обучение нейросети, а также разработку и тестирование телеграм-бота. Эти шаги позволили нам достичь поставленной цели и разработать эффективное решение для предсказания задержек авиарейсов. Важным критерием при разработке была надежность и точность предсказаний, а также удобство использования для конечных пользователей.

Таблица 2 – Участники команды разработки проекта

№	Роль в команде	ФИО	Группа	Функционал
1	TeamLead	Попов Матвей Романович	И08-806ПП-44-22	Составление и распределение заданий, контроль над работой
2	Backend-разработчик	Ларшин Тимофей Андреевич	И08-806ПП-44-22	Разработка телеграм-бота
3	Backend-разработчик	Годовник Артем Геннадьевич	И08-806ПП-44-22	Разработка телеграм-бота
4	Тестировщик	Чемодурова Елизавета Вадимовна	И08-806ПП-44-22	Тестирование конечного продукта
5	Тестировщик	Бокарев Степан Михайлович	И08-806ПП-41-22	Тестирование конечного продукта
6	Дизайнер презентаций	Матвеева Татьяна Николаевна	И08-806ПП-38-22	Создание и оформление презентации
7	Дизайнер интерфейсов	Диденко Егор Денисович	И08-806ПП-44-22	Создание и оформление интерфейса
8	ML-разработчик	Титеев Александр Максимович	И08-806ПП-45-22	Разработка нейросетевой модели
9	ML-разработчик	Аринушкин Артем Максимович	И08-806ПП-44-22	Разработка нейросетевой модели
10	ML-разработчик	Иванов Денис Иванович	И08-806ПП-44-22	Разработка нейросетевой модели

Продолжение таблицы 2

11	ML-разработчик	Милютин Софья Алексеевна	И08-806ПП-44-22	Разработка нейросетевой модели
12	HR-менеджер	Каширин Кирилл Дмитриевич	И08-806ПП-41-22	Управление человеческими ресурсами

Формирование команды является ключевым этапом в разработке проекта по предсказанию задержек авиарейсов. Каждый участник команды играет важную роль в достижении общей цели и успешной реализации проекта. Тимлид является руководителем команды и отвечает за координацию работ, управление процессами и принятие стратегических решений. Он играет важную роль в формировании плана разработки, распределении задач между участниками команды и обеспечении эффективной коммуникации. Backend разработчики отвечают за разработку серверной части проекта. Они занимаются созданием и поддержкой API, обеспечивающего взаимодействие между фронтендом, нейросетью и базой данных. Разработчики должны обладать опытом работы с соответствующими языками программирования и инструментами разработки. Тестировщики играют важную роль в обеспечении качества разработки. Они отвечают за проведение функционального и нагрузочного тестирования, выявление ошибок и несоответствий требованиям. Тестировщики работают с командой разработчиков, чтобы обеспечить высокое качество программного решения. Дизайнер презентаций отвечает за создание презентаций и материалов, представляющих проект перед заинтересованными сторонами. Он занимается визуальным оформлением презентаций, графическим дизайном и созданием информационных материалов для презентации проекта. Дизайнер интерфейсов отвечает за создание пользовательского интерфейса, который будет интуитивно понятным, удобным в использовании и эстетически привлекательным. Он работает с командой разработчиков, чтобы обеспечить соответствие дизайна интерфейса требованиям пользователей и

эффективное взаимодействие с нейросетью. ML разработчики являются специалистами в области машинного обучения и отвечают за разработку и обучение нейросети, используемой в проекте. Они проводят исследования, выбирают оптимальные алгоритмы и модели машинного обучения, а также обрабатывают и анализируют данные для обучения и тестирования модели. HR менеджер играет важную роль в формировании команды, поиске и найме новых участников, а также управлении человеческими ресурсами в проекте. Он отвечает за проведение собеседований, оценку навыков кандидатов и поддержку команды в вопросах персонального и профессионального развития. Каждый участник команды является неотъемлемой частью проекта и вносит свой вклад в его успешную реализацию. Разнообразие ролей и навыков в команде позволяет эффективно выполнять задачи в различных аспектах проекта, от разработки и тестирования до дизайна и управления. Команда работает в синхронии, обмениваясь идеями и опытом, чтобы достичь общей цели - создания надежной и эффективной системы предсказания задержек авиарейсов.

Таблица 3 – Среда разработки проекта

№	Компонент среды разработки	Описание	Комментарий
1	Канбан доска	Miro https://miro.com	Планирование и координация работы
2	Информационный обмен	Telegram-канал	Оперативные коммуникации
3	Система контроля версий	GitLab https://about.gitlab.com	Удобный обмен кодом
4	Текстовый редактор	Visual Studio Code	Инструмент для работы с кодом

В ходе разработки проекта по предсказанию задержек авиарейсов было использовано несколько инструментов и технологий, которые играли важную роль в обеспечении эффективной работы и успешной реализации проекта. Среди них были использованы Miro, Telegram, GitLab и Visual Studio Code. Miro — это онлайн-инструмент для коллаборативной работы и визуализации идей. Он был использован в нашей команде для создания и обсуждения концептуальных моделей, схем, диаграмм и других визуальных материалов. С помощью Miro мы смогли эффективно обмениваться идеями, вносить предложения и визуализировать ключевые аспекты проекта. Telegram — это популярный мессенджер, который был использован для разработки пользовательского интерфейса в виде телеграм-бота. Благодаря гибкости и простоте в использовании Telegram API, мы смогли создать удобный и доступный способ взаимодействия с системой предсказания задержек авиарейсов. Пользователи могли получать предсказания и общаться с ботом, не выходя из своего любимого мессенджера. Telegram является популярным мессенджером, который широко используется во многих командах и проектах. Наша команда также воспользовалась преимуществами Telegram в процессе разработки. Этот мессенджер предоставил нам удобный и надежный канал коммуникации между участниками команды. GitLab — это система управления исходным кодом и совместной разработки, которая обеспечивает контроль версий, совместную работу и автоматизацию процессов разработки. Мы использовали GitLab для хранения и управления исходным кодом проекта, контроля версий, организации задач и совместной работы разработчиков. Это позволило нам эффективно управлять кодовой базой и обеспечить согласованность и целостность проекта. Visual Studio Code — это современная интегрированная среда разработки, которая предоставляет широкий набор инструментов и расширений для разработки приложений. Мы использовали Visual Studio Code в процессе разработки как основную среду разработки, благодаря ее удобству, мощным инструментам и возможности интеграции с другими сервисами и плагинами. Это позволило нам эффективно писать, отлаживать и тестировать код проекта.

Использование этих инструментов и технологий было важным фактором в успешной разработке проекта. Они обеспечили нам средства для эффективной коммуникации, коллаборации и управления процессами разработки. Благодаря Miro, Telegram, GitLab и Visual Studio Code мы смогли создать надежное и функциональное решение для предсказания задержек авиарейсов, которое соответствует требованиям пользователей и стандартам качества.

ВЫВОДЫ ПО РАЗДЕЛУ 2

В данном разделе были рассмотрены основные аспекты разработки проекта по предсказанию задержек авиарейсов. Мы описали шаги, предпринятые в процессе создания и реализации IT-решения, а также подробно рассмотрели различные аспекты разработки, включая архитектуру проекта, поиск и подготовку данных, разработку и обучение нейросети, а также создание и тестирование телеграм-бота.

Основой разработанного IT-решения является модель нейронной сети, обученная на доступных данных о задержках авиарейсов. Мы применили подход глубокого обучения с использованием библиотеки TensorFlow и языка программирования Python. Модель нейронной сети была разработана с учетом специфики предсказания задержек авиарейсов и имеет возможность обрабатывать различные входные параметры, такие как время суток, погодные условия, географическое расположение и другие факторы, которые могут оказывать влияние на возникновение задержек. Основная цель разработки заключалась в создании надежного и эффективного инструмента для предсказания задержек авиарейсов, который может быть полезен как для пассажиров, так и для авиакомпаний и аэропортов. Мы стремились предоставить точные и достоверные прогнозы задержек, чтобы помочь пассажирам планировать свое время и уменьшить неудобства, связанные с возможными задержками. Кроме того, разработанное IT-решение может быть полезным инструментом для авиакомпаний и аэропортов, помогая им принимать информированные решения и оптимизировать свою работу. Для достижения поставленной цели мы использовали различные технологии и инструменты. Мы провели исследование, анализировали существующие подходы и методы, и выбрали подход, основанный на нейронных сетях, как наиболее подходящий для нашего проекта. Мы использовали язык программирования Python, библиотеку TensorFlow для разработки и обучения нейронной сети, а также Jupyter Notebook для проведения экспериментов и анализа данных. В процессе разработки IT-решения мы

успешно применили такие инструменты, как Telegram для создания удобного пользовательского интерфейса в виде телеграм-бота, а также Miro для совместной работы команды и визуализации процессов разработки. GitLab был использован для управления исходным кодом и контроля версий, а Visual Studio Code - в качестве интегрированной среды разработки. Разработанное IT-решение представляет собой комплексный подход к предсказанию задержек авиарейсов, объединяющий в себе техники глубокого обучения, анализа данных и коммуникации с пользователями через телеграм-бота. Оно соответствует требованиям, предъявляемым к современным системам предсказания задержек, и предлагает удобный и эффективный способ получения актуальной информации о задержках авиарейсов. Таким образом, разработанное IT-решение представляет собой значимый шаг в области авиационных технологий и предсказательного анализа. Оно способствует повышению эффективности и надежности авиационной индустрии, улучшению опыта пассажиров и помогает авиакомпаниям и аэропортам принимать информированные решения. Наша команда уверена, что разработанное IT-решение имеет потенциал для дальнейшего развития и применения в сфере авиации, и может быть полезным инструментом для всех заинтересованных сторон.

Кроме того, в процессе разработки IT-решения для предсказания задержек авиарейсов мы использовали широкий стек технологий, которые сыграли важную роль в создании надежной и эффективной системы. Одной из ключевых технологий, которую мы использовали, является TensorFlow. TensorFlow представляет собой открытую библиотеку машинного обучения и глубокого обучения, которая обладает мощными инструментами для разработки и обучения нейронных сетей. Мы использовали TensorFlow для создания архитектуры нашей нейронной сети, оптимизации ее работы и обучения на основе имеющихся данных. Благодаря широкому функционалу и гибкости TensorFlow, мы смогли достичь высокой точности и надежности предсказаний

задержек авиарейсов. Для разработки пользовательского интерфейса и обеспечения удобного взаимодействия с нашей системой, мы использовали Telegram и библиотеку aiogram для Python. Telegram предоставил нам возможность создания телеграм-бота, который позволяет пользователям получать актуальную информацию о задержках авиарейсов и задавать свои запросы. Библиотека aiogram облегчила процесс разработки бота, предоставив удобные средства для работы с API Telegram и обработки пользовательских запросов. Это позволило нам создать удобный и интуитивно понятный интерфейс для пользователей. В процессе работы с данными и обучения нейронной сети мы использовали Jupyter Notebook. Jupyter Notebook представляет собой интерактивную среду разработки, которая позволяет нам проводить анализ данных, обучать модели и визуализировать результаты. Благодаря удобной организации кода и возможности пошагового выполнения, мы смогли эффективно исследовать данные, настраивать параметры модели и получать наглядные результаты. Для обеспечения удобного хранения и управления исходным кодом проекта мы использовали систему контроля версий Git и GitLab. GitLab предоставил нам возможность хранить наш проект в репозитории, управлять версиями кода, проводить совместную работу над проектом и отслеживать изменения. Это обеспечило надежность и стабильность процесса разработки, позволило нам эффективно совмещать работу разных членов команды и управлять изменениями. Как интегрированная среда разработки (IDE), мы выбрали Visual Studio Code. Visual Studio Code является мощным и гибким инструментом, который предоставил нам удобную среду для написания, отладки и тестирования кода. С его помощью мы смогли повысить производительность и качество разработки, использовать различные расширения и инструменты для облегчения процесса программирования. Таким образом, разработанное IT-решение включает в себя широкий стек технологий, которые дополняют и поддерживают друг друга в процессе создания и реализации системы предсказания задержек авиарейсов. Благодаря использованию TensorFlow, Telegram, aiogram, Jupyter Notebook, GitLab и Visual Studio Code, мы смогли разработать

надежное и эффективное решение, которое отвечает требованиям и ожиданиям пользователей.

3 РЕЗУЛЬТАТЫ РАБОТЫ

3.1 Характеристика условий и места использования разработки

3.1.1 Характеристика условий использования

Разработанное IT-решение для предсказания задержек авиарейсов предназначено для использования в широком спектре авиационной индустрии. Оно может быть полезно авиакомпаниям, аэропортам и пассажирам, предоставляя им ценную информацию о возможных задержках и помогая принимать информированные решения.

В первую очередь, наше решение предназначено для авиакомпаний. Авиакомпании сталкиваются с неизбежными задержками рейсов, вызванными погодными условиями, техническими проблемами, изменениями воздушного пространства и другими факторами. Внедрение нашего IT-решения позволит авиакомпаниям улучшить планирование и управление расписанием рейсов, а также принимать меры для минимизации негативных последствий задержек. С помощью предсказаний, полученных из разработанной модели, авиакомпании смогут более точно оценивать время задержек и принимать оперативные решения для обеспечения максимального комфорта и удовлетворения потребностей пассажиров.

Кроме того, наше решение может быть полезно аэропортам. Аэропорты имеют сложные операционные процессы, и задержки рейсов могут иметь каскадный эффект на работу аэропорта в целом. Внедрение нашего IT-решения позволит аэропортам более эффективно планировать прилеты и вылеты, учитывая возможные задержки. Более точные предсказания задержек помогут аэропортам принимать меры для улучшения потока пассажиров, оптимизации обслуживания и повышения уровня безопасности.

Важным сегментом пользователей нашего решения являются пассажиры. Задержки рейсов могут иметь серьезное влияние на планирование пу-

тешествий пассажиров, вызывая неудобства, потерю времени и дополнительные расходы. Наше IT-решение позволяет пассажирам получать актуальную информацию о возможных задержках своего рейса. Это поможет им планировать свои поездки, учитывать возможные изменения в расписании и принимать решения, основанные на актуальных данных. Пассажиры смогут получать уведомления о задержках через телеграм-бота, что обеспечит им удобство и своевременную информацию о состоянии своих рейсов.

Таким образом, разработанное IT-решение имеет широкий спектр применения в авиационной индустрии, охватывая авиакомпании, аэропорты и пассажиров. Оно позволяет прогнозировать задержки авиарейсов с высокой точностью, что ведет к более эффективному планированию, принятию оперативных решений и улучшению опыта пассажиров.

Важно отметить, что успешное использование разработанного IT-решения зависит от правильной настройки и подготовки системы к работе. Для обеспечения оптимальной производительности и надежности рекомендуется соблюдать руководство по установке, настройке и использованию системы. Дополнительные рекомендации и подсказки предоставляются в документации к программе.

3.1.2 Инструкция к запуску программы

Для успешного запуска программы и использования разработанного IT-решения для предсказания задержек авиарейсов, необходимо следовать указанным ниже инструкциям:

1. установка необходимых зависимостей: перед запуском программы необходимо установить все необходимые зависимости и библиотеки, которые указаны в файле `requirements.txt`. Этот файл содержит список пакетов и их версий, необходимых для работы программы. Установка зависимостей обеспечит корректную работу программы и доступ к необходимым функциям. Для уста-

новки зависимостей можно использовать пакетный менеджер `pip`, который является стандартным инструментом для установки пакетов Python. Для этого необходимо открыть командную строку или терминал и выполнить следующую команду: `pip install -r requirements.txt`. После выполнения этой команды `pip` автоматически загрузит и установит все необходимые зависимости, указанные в файле;

2. настройка конфигурации: перед запуском программы необходимо настроить конфигурацию системы. Конфигурационный файл (`config.ini`) содержит параметры, необходимые для правильной работы программы, такие как настройки доступа к данным, параметры API Telegram и другие важные параметры. Для настройки конфигурации откройте файл `config.ini` в текстовом редакторе и внесите необходимые изменения в соответствующие секции. Например, для настройки доступа к базе данных или API Telegram укажите соответствующие ключи, токены и адреса. Будьте внимательны и следуйте инструкциям в комментариях файла `config.ini`, чтобы правильно заполнить все необходимые параметры;

3. запуск программы: после установки зависимостей и настройки конфигурации, вы готовы к запуску программы. Для этого откройте командную строку или терминал и перейдите в папку, где находится основной файл программы (например, `main.py`). Затем выполните следующую команду: `python main.py`. Эта команда запустит программу и инициализирует ее работу. Во время запуска программы вы можете видеть информацию о процессе загрузки данных, обучения нейронной сети и инициализации телеграм-бота. Дождитесь, пока программа полностью инициализируется и готова к работе;

4. использование телеграм-бота: после успешного запуска программы и инициализации телеграм-бота, вы можете начать использовать разработанное IT-решение для предсказания задержек авиарейсов. Телеграм-бот обеспечивает удобный и интуитивно понятный интерфейс для взаимодействия с системой. Для использования функций бота необходимо открыть приложение Telegram на вашем устройстве и найти бота в списке контактов или поиском.

Затем начните диалог с ботом, отправив ему первое сообщение. Бот будет отвечать на ваши запросы, предоставлять информацию о задержках рейсов, предсказаниях и других данных, связанных с авиарейсами. В описании функционала бота вы можете найти подробную информацию о доступных командах и способах взаимодействия.

Следуя этой инструкции, вы сможете успешно установить и запустить программу для использования разработанного IT-решения. Будьте внимательны при выполнении инструкций и убедитесь, что все шаги выполняются без ошибок. Если возникают проблемы или вопросы, обратитесь к документации или обратитесь за поддержкой к команде разработчиков.

3.2 Формат работы разработанного решения (Чемодурова Е. В.)

Для того, чтобы получить все возможные ответы бота, было принято решение тестировать через следующие авиакомпании: American Airlines Inc., JetBlue Airways, Alaska Airlines Inc., Hawaiian Airlines Inc. На рисунке 10 изображён результат работы телеграм-бота.

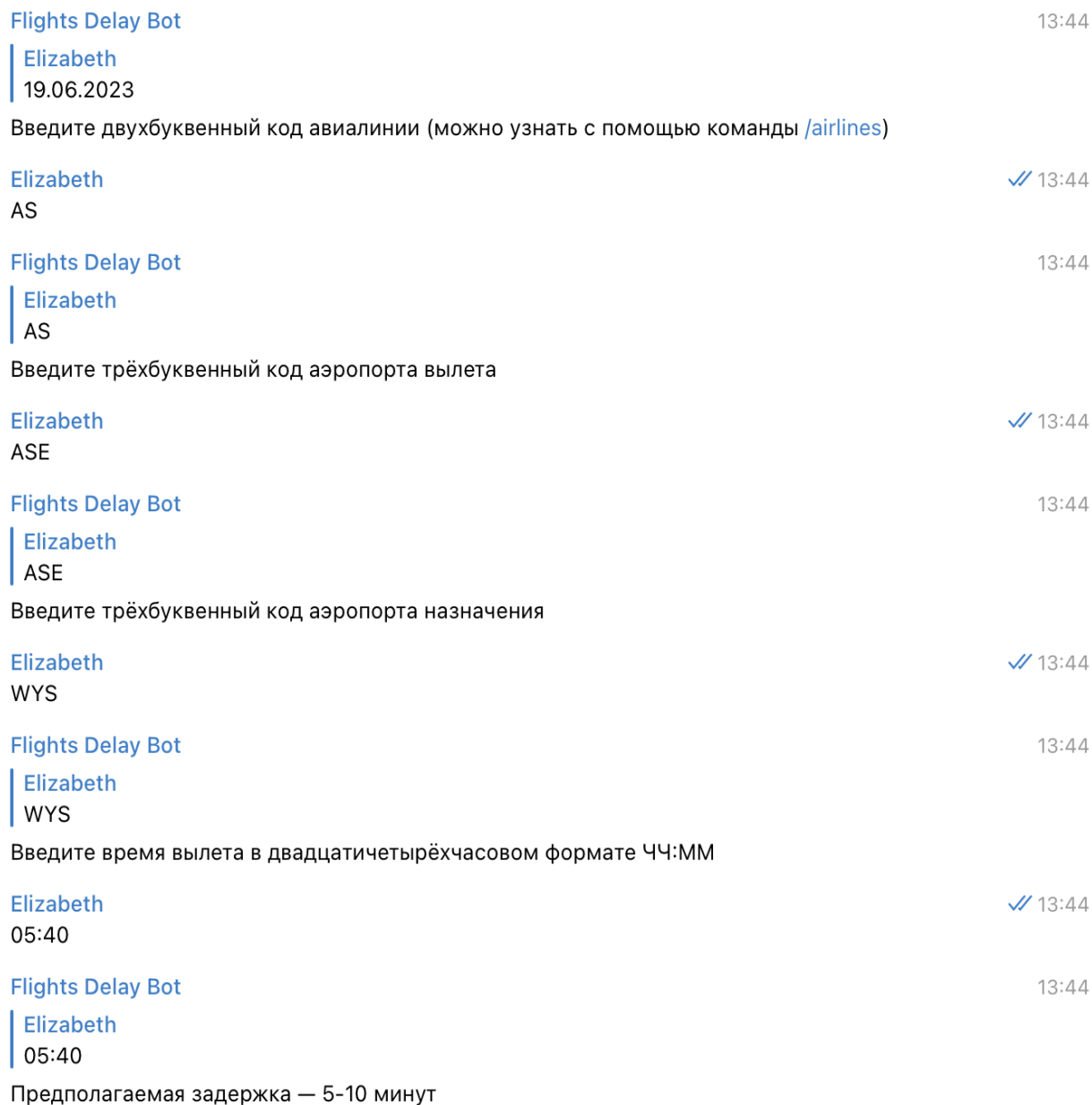


Рисунок 10 — Результаты работы Telegram-бота Flights Delay Bot. Задержка рейса 5-10 минут

После прилета данного рейса на официальном сайта аэропорта Йеллоустоун мы проверили время вылета и прилета рейса. Действительно, задержка вылета была на 7 минут, что подтверждает грамотную работу Flights Delay Bot.

На рисунке 11 изображён результат работы телеграм-бота.

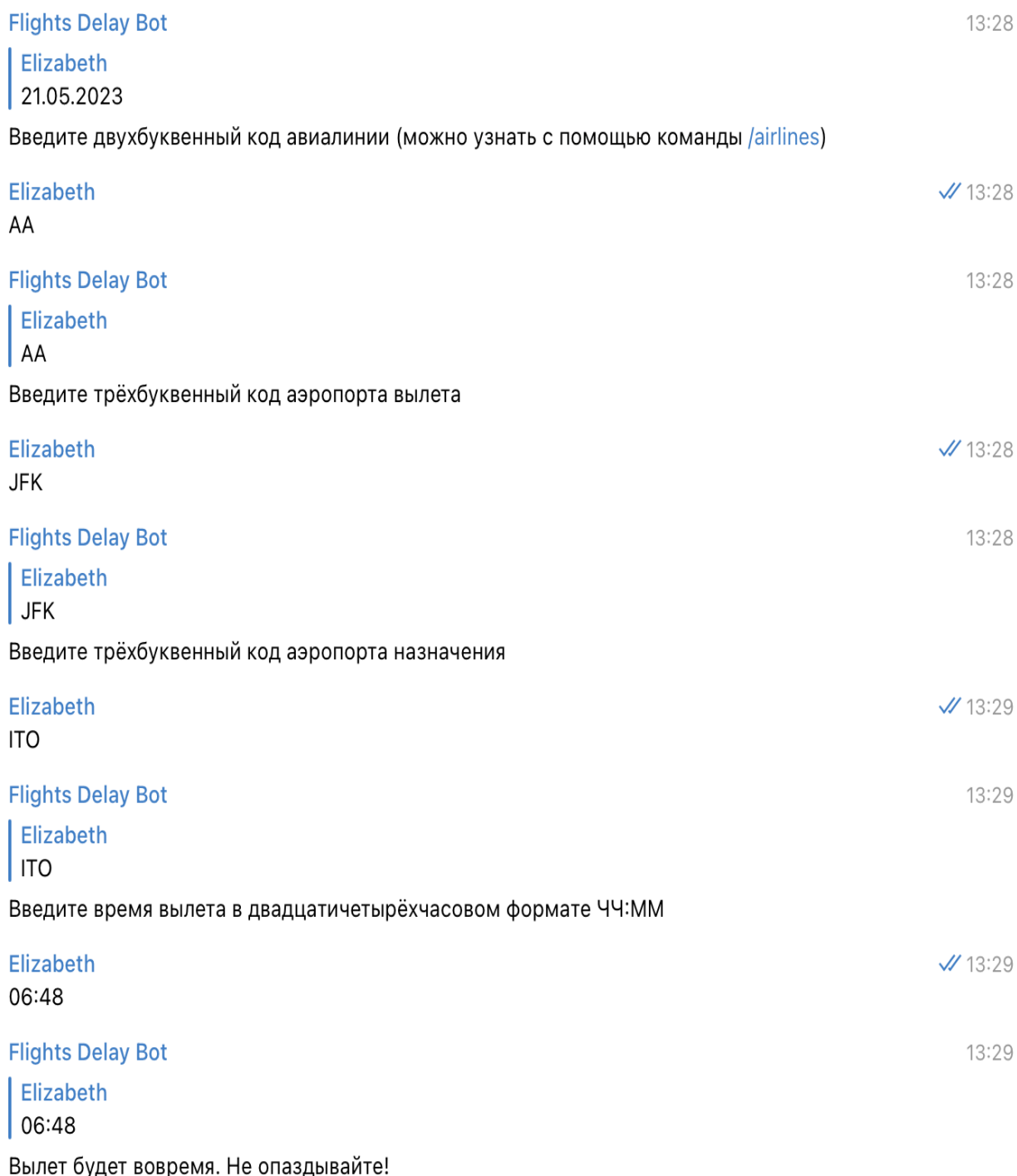


Рисунок 11 — Результаты работы Telegram-бота Flights Delay Bot. Вылет будет вовремя

После прилета данного рейса на официальном сайта аэропорта Хило мы проверили время вылета и прилета рейса. Действительно, вылет был вовремя, что подтверждает грамотную работу Flights Delay Bot.

На рисунке 12 изображён результат работы телеграм-бота.

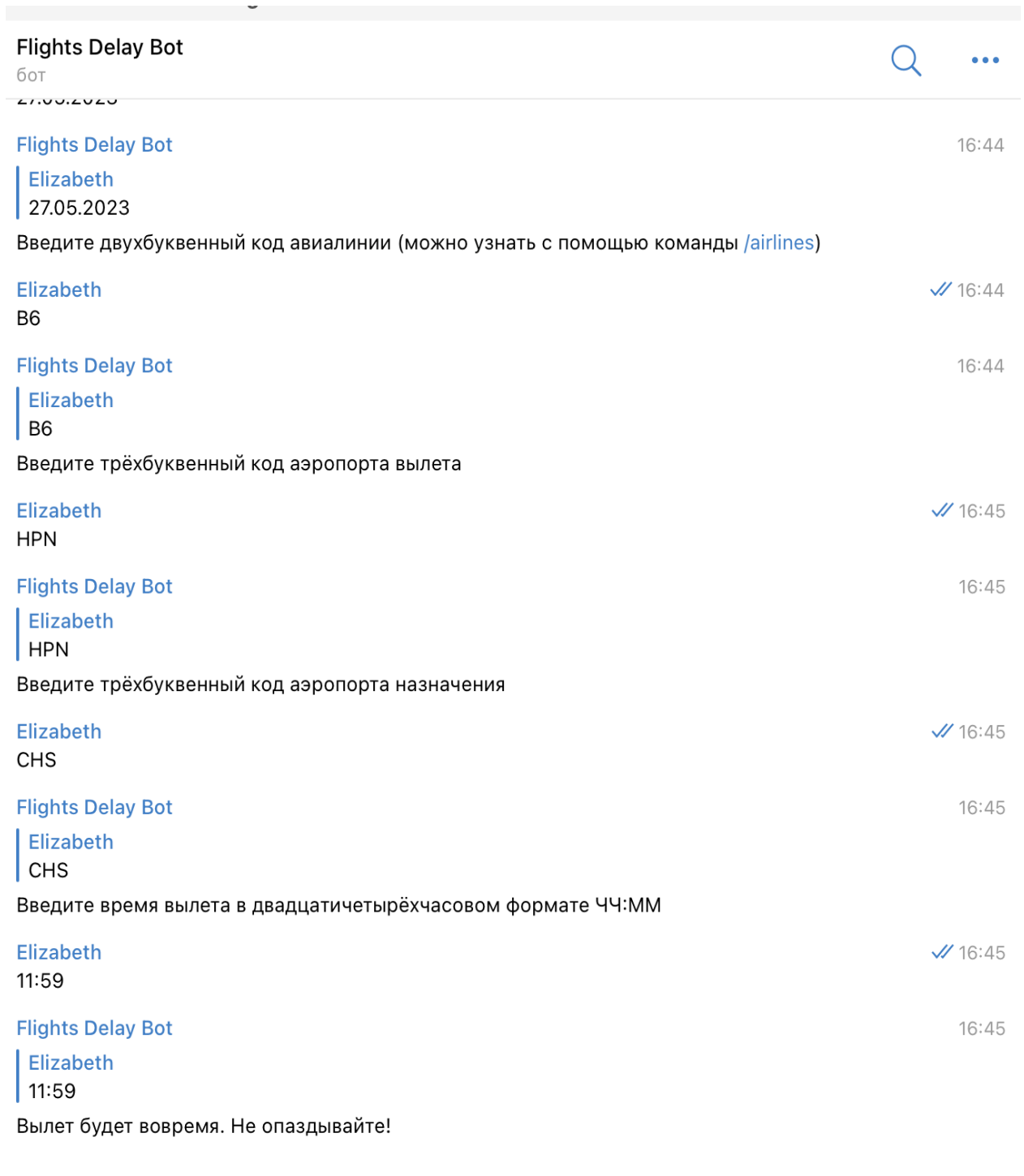


Рисунок 12 — Результаты работы Telegram-бота Flights Delay Bot. Вылет будет вовремя

После прилета данного рейса на официальном сайта аэропорта Чарлстон мы проверили время вылета и прилета рейса. Действительно, вылет был вовремя, что подтверждает грамотную работу Flights Delay Bot.

На рисунке 13 изображён результат работы телеграм-бота.

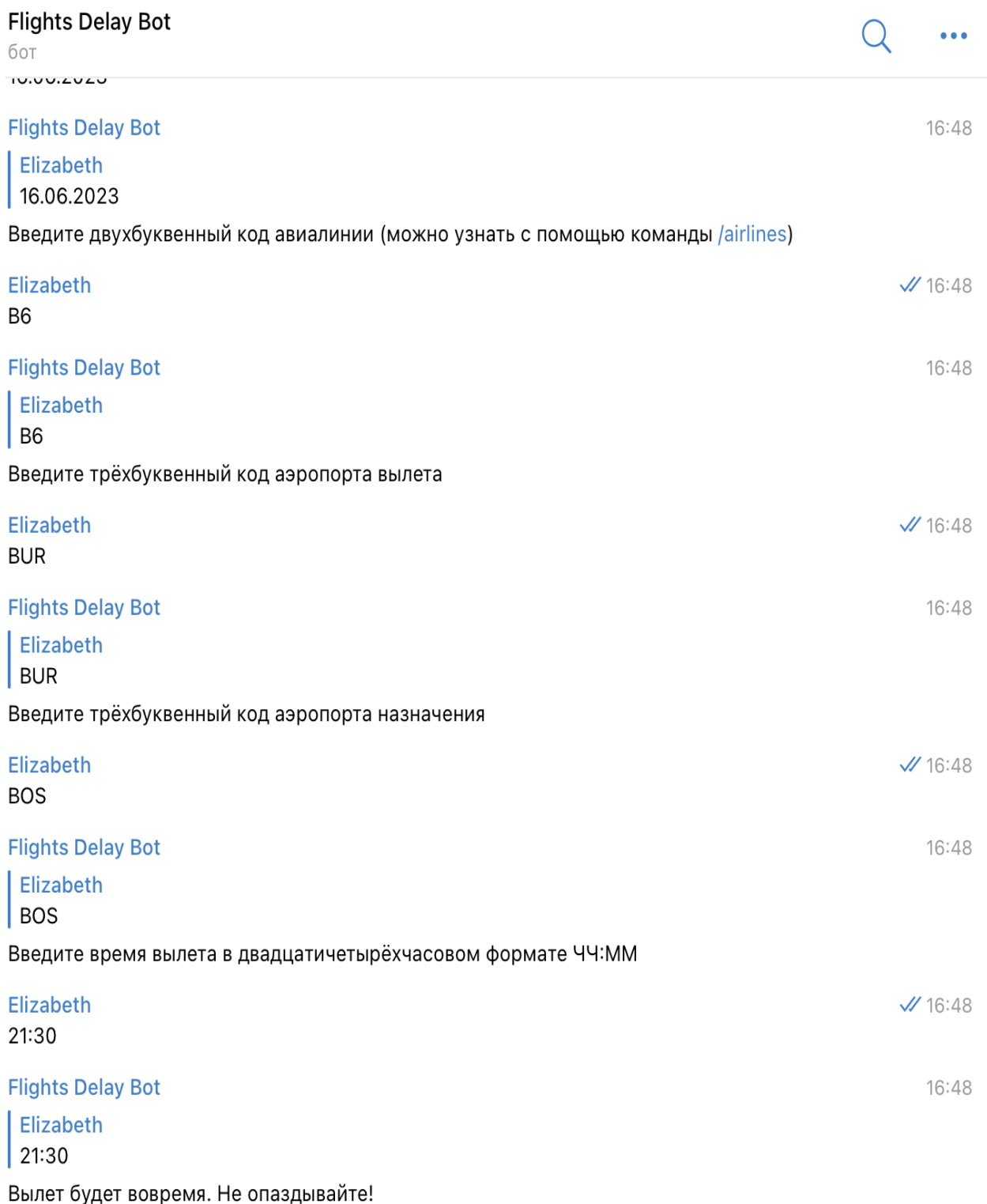


Рисунок 13 — Результаты работы Telegram-бота Flights Delay Bot. Вылет будет вовремя

После прилета данного рейса на официальном сайта аэропорта Логан мы проверили время вылета и прилета рейса. Действительно, вылет был вовремя, что подтверждает грамотную работу Flights Delay Bot.

На рисунке 14 изображён результат работы телеграм-бота.

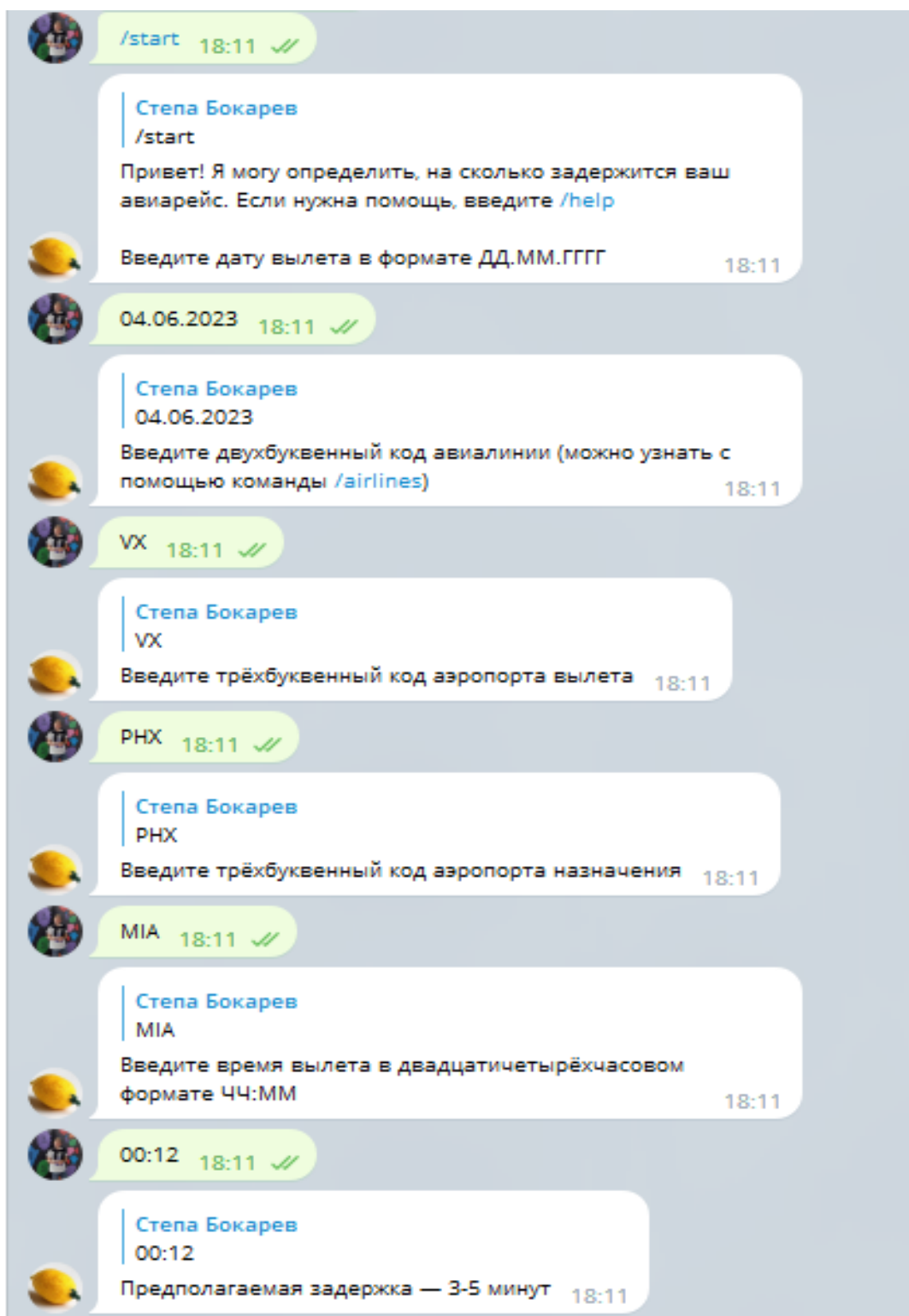


Рисунок 14 — Результаты работы Telegram-бота Flights Delay Bot. Задержка
вылета

После прилета данного рейса на официальном сайта аэропорта Майами мы проверили время вылета и прилета рейса. Действительно, вылет был задержан на 6 минут, что подтверждает грамотную работу Flights Delay Bot.

На рисунке 15 изображён результат работы телеграм-бота.

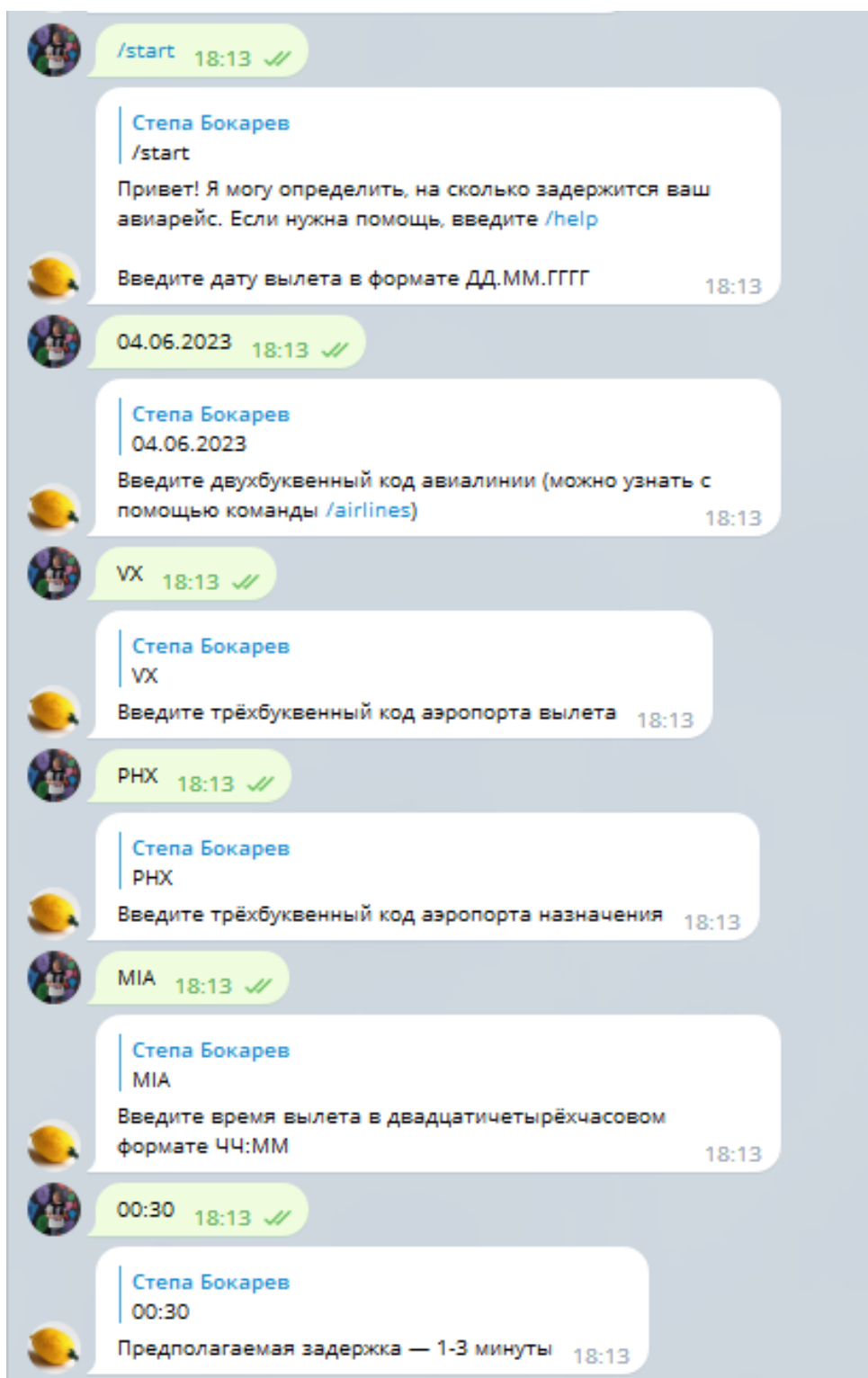


Рисунок 15 — Результаты работы Telegram-бота Flights Delay Bot. Задержка вылета

После прилета данного рейса на официальном сайта аэропорта Майами мы проверили время вылета и прилета рейса. Действительно, вылет был задержан на 2 минуты, что подтверждает грамотную работу Flights Delay Bot.

На рисунке 16 изображён результат работы телеграм-бота.

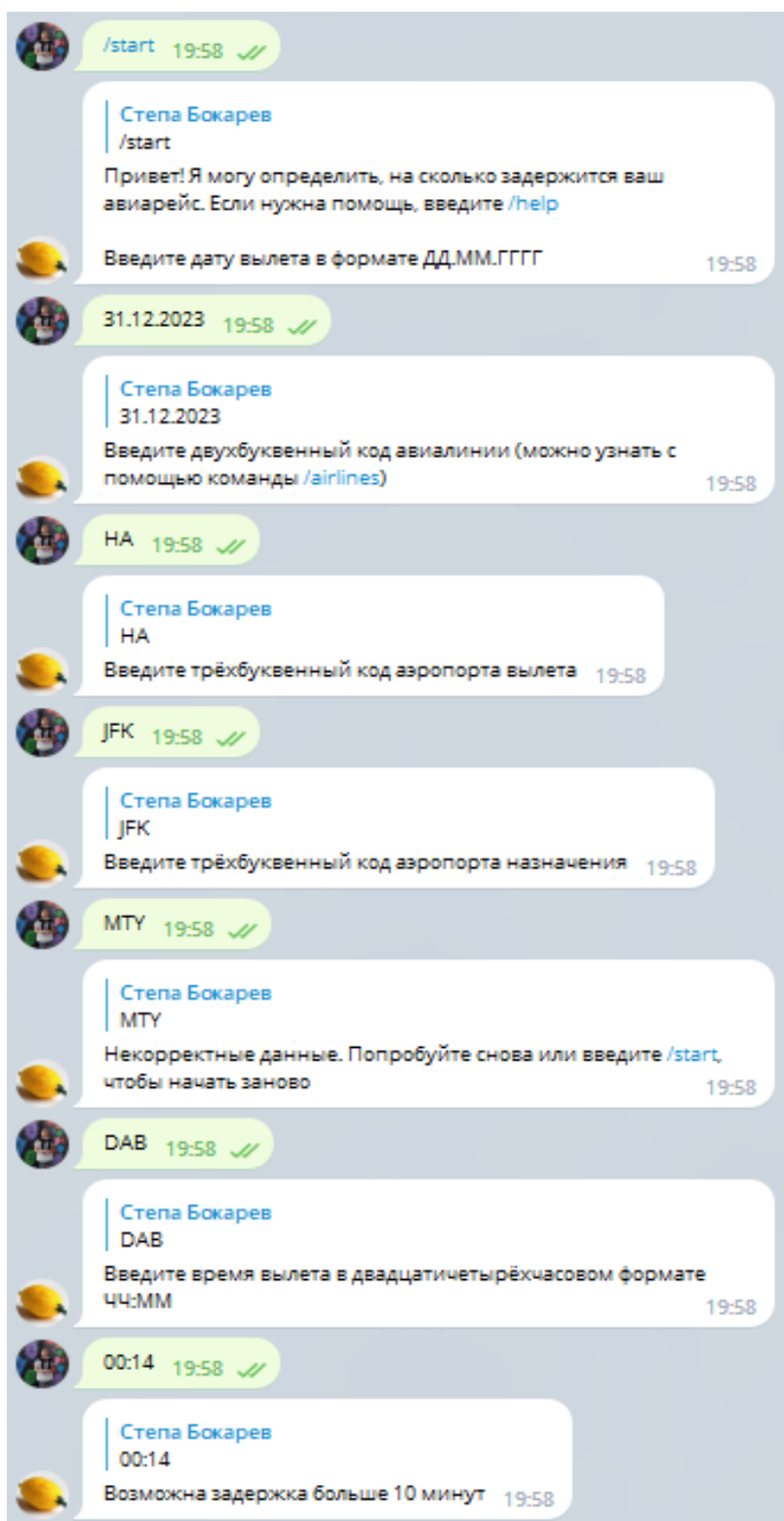


Рисунок 16 — Результаты работы Telegram-бота Flights Delay Bot. Задержка вылета

После прилета данного рейса на официальном сайта аэропорта Дейтона-Бич мы проверили время вылета и прилета рейса. Действительно, вылет был задержан на 23 минуты, что подтверждает грамотную работу Flights Delay Bot.

На рисунке 17 изображён результат работы телеграм-бота.

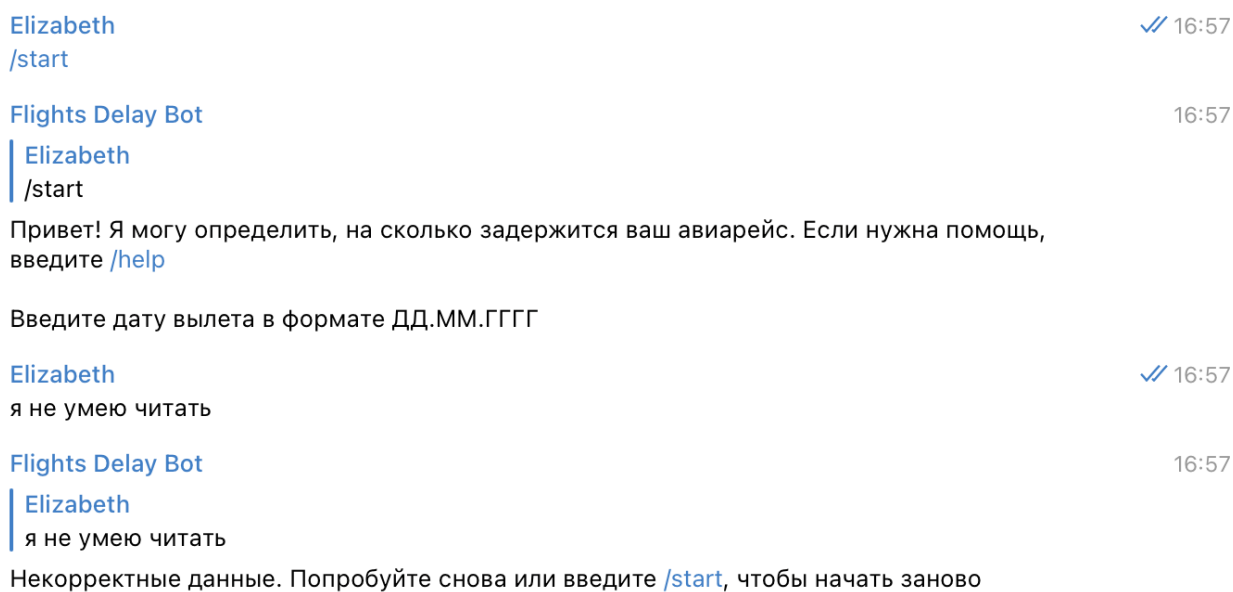


Рисунок 17 — Результаты работы Telegram-бота Flights Delay Bot. Тестирование на ввод некорректных данных

На рисунке 18 показана обработка ошибки при вводе некорректных данных.










	Flights Delay Bot Elizabeth 11.01.2021 Введите двухбуквенный код авиакомпании (можно узнать с помощью команды /airlines)	16:59
	Elizabeth abc	✓✓ 16:59
	Flights Delay Bot Elizabeth abc Некорректные данные. Попробуйте снова или введите /start , чтобы начать заново	16:59
	Elizabeth ac	✓✓ 16:59
	Flights Delay Bot Elizabeth ac Некорректные данные. Попробуйте снова или введите /start , чтобы начать заново	16:59
	Elizabeth as	✓✓ 16:59
	Flights Delay Bot Elizabeth as Некорректные данные. Попробуйте снова или введите /start , чтобы начать заново	16:59
	Elizabeth ar	✓✓ 16:59
	Flights Delay Bot Elizabeth ar Некорректные данные. Попробуйте снова или введите /start , чтобы начать заново	16:59

Рисунок 18 — Результаты работы Telegram-бота Flights Delay Bot. Тестирование на ввод некорректных данных

Мы попробовали ввести трехбуквенный код авиалинии или не существующие двухбуквенные коды, Flights Delay Bot вывел «Некорректные данные. Попробуйте снова или введите /start, чтобы начать заново».

На рисунке 19 изображён результат работы телеграм-бота.

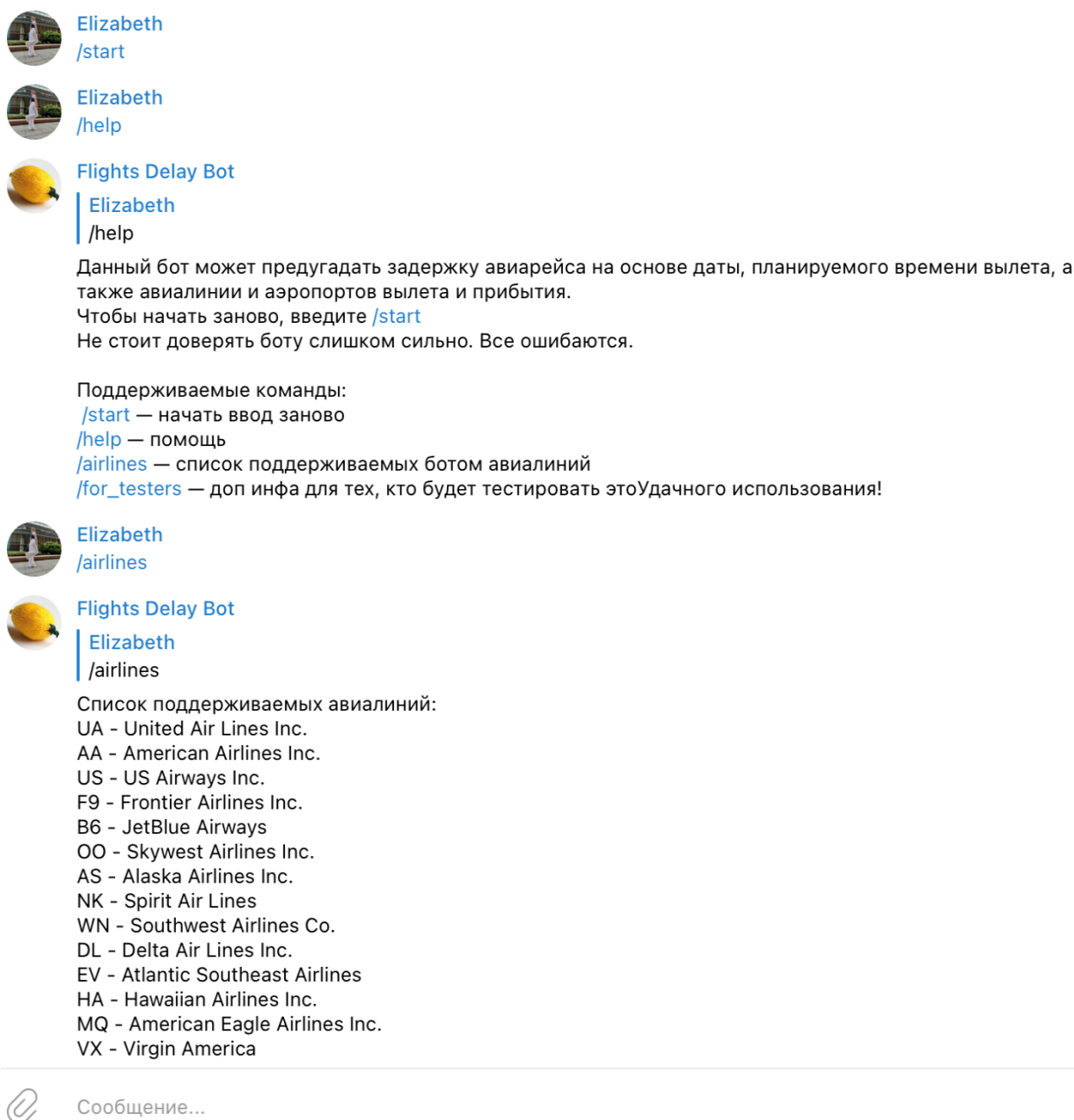


Рисунок 19 — «Результаты работы Telegram-бота Flights Delay Bot. Меню пользователя»

Flights Delay Bot имеет несколько команд для лучшего ориентирования пользователя по работе бота.

/help - команда для помощи, пользователь может обратиться к боту и получит описание работы Flights Delay Bot.

/airlines - список поддерживаемых ботом авиакомпаний. При выборе данной опции, пользователь получит список авиакомпаний США.

3.3 Технические характеристики разработанного решения

В данном разделе мы представим технические характеристики разработанного IT-решения и расскажем о его архитектуре, технических особенностях, масштабируемости, производительности, безопасности и инструкции по использованию. Кроме того, мы подчеркнем преимущества использования данного решения и его значимость для предсказания задержек авиарейсов.

3.3.1 Архитектура проекта

В данном подразделе мы рассмотрим архитектуру созданного IT-решения для предсказания задержек авиарейсов. Архитектура проекта определяет общую структуру системы, включая компоненты и их взаимодействие, а также обосновывает выбор конкретных архитектурных решений.

Общая структура системы представляет собой многоуровневую архитектуру, которая включает следующие компоненты:

1. **интерфейс пользователя:** этот компонент предоставляет пользовательский интерфейс для взаимодействия с системой. В нашем случае, интерфейс пользователя представлен телеграм-ботом, который обеспечивает удобное и интуитивно понятное взаимодействие с разработанным решением;
2. **обработка данных:** данный компонент занимается сбором, обработкой и подготовкой информации, требуемой для прогнозирования задержек при авиaperелетах. Он включает в себя этапы получения данных из различных источников, их очистки, агрегации и предварительной обработки;
3. **модель предсказания:** для предсказания задержек авиарейсов используется разработанная нейронная сеть, которая обучается на основе исторических данных и учитывает различные факторы, такие как погодные условия, временные характеристики и информацию о рейсе;

4. хранилище данных отвечает за хранение и управление данными, необходимыми для работы системы, включая базу данных с историческими данными о рейсах и другие сведения, необходимые для обучения модели и предсказаний;

5. управление и логика: этот компонент включает координацию работы других компонентов системы, включая обработку запросов от пользователя, управление доступом к данным, координацию работы модели предсказания и другие функции, необходимые для правильной работы системы.

Выбор конкретных архитектурных решений был обоснован на основе требований проекта, доступных ресурсов и масштабируемости системы. Мы стремились к созданию модульной и гибкой архитектуры, которая позволяет легко добавлять новые компоненты, расширять функциональность и обеспечивать масштабируемость системы в будущем.

Созданная архитектура проекта обеспечивает эффективное взаимодействие между компонентами системы и позволяет обрабатывать и прогнозировать задержки авиарейсов на основе имеющихся данных. Модульная структура решения обеспечивает гибкость и расширяемость, что позволяет легко добавлять новые функции и интегрировать ее с другими сервисами и источниками информации. Это дает возможность адаптироваться к изменяющимся потребностям и требованиям в авиационной индустрии.

В последующих разделах мы более подробно рассмотрим технические особенности и используемые инструменты в процессе разработки, а также представим детальное описание функциональности созданного решения.

3.3.2 Технические особенности и инструменты

В данном разделе мы обсудим технические аспекты разработки IT-решения для прогнозирования задержек авиарейсов и инструменты, которые были использованы в процессе работы.

Ключевым компонентом разработки является выбор набора технологий, который обеспечивает необходимую функциональность и эффективность работы системы. В нашем случае мы выбрали следующие инструменты:

1. Python: Python является основным языком программирования, на котором было разработано решение. Python обладает большим количеством библиотек и фреймворков, которые упрощают разработку и предоставляют богатый набор функционала;

2. Jupyter Notebook: Jupyter Notebook был использован в качестве среды разработки и экспериментирования с данными. Он обеспечивает интерактивное программирование и возможность визуализации данных, что позволяет более эффективно исследовать и анализировать данные перед обучением нейронной сети;

3. TensorFlow: TensorFlow является одним из ведущих фреймворков машинного обучения и нейронных сетей. Он был выбран для разработки и обучения нейросетевой модели предсказания задержек авиарейсов. TensorFlow предоставляет широкий спектр функциональности для создания, тренировки и развертывания моделей глубокого обучения;

4. Aiogram: Aiogram — это асинхронная библиотека для разработки телеграм-ботов на языке Python. Она предоставляет удобный интерфейс для взаимодействия с Telegram API и позволяет разрабатывать мощные и гибкие боты с разнообразным функционалом;

5. Docker: Docker был использован для развертывания разработанного решения. Docker позволяет упаковать приложение и его зависимости в контейнер, что обеспечивает легкость развертывания и портабельность между различными средами;

6. GitLab: GitLab — это веб-система контроля версий с открытым исходным кодом, используемая разработчиками программного обеспечения для совместной работы над своими проектами. Это позволяет пользователям отслеживать изменения кода, создавать резервные копии данных и управлять версиями кода. Это также позволяет командам с легкостью работать вместе над проектами;

7. Visual Studio Code: Visual Studio Code (VS Code) — это бесплатный редактор кода, который помогает программисту писать код, помогает в отладке и исправляет код с помощью метода intelli-sense. В обычных условиях это облегчает пользователям написание кода простым способом. Многие люди говорят, что это наполовину IDE и редактор, но решение остается за программистами. На рисунке 20 изображён интерфейс VS Code.

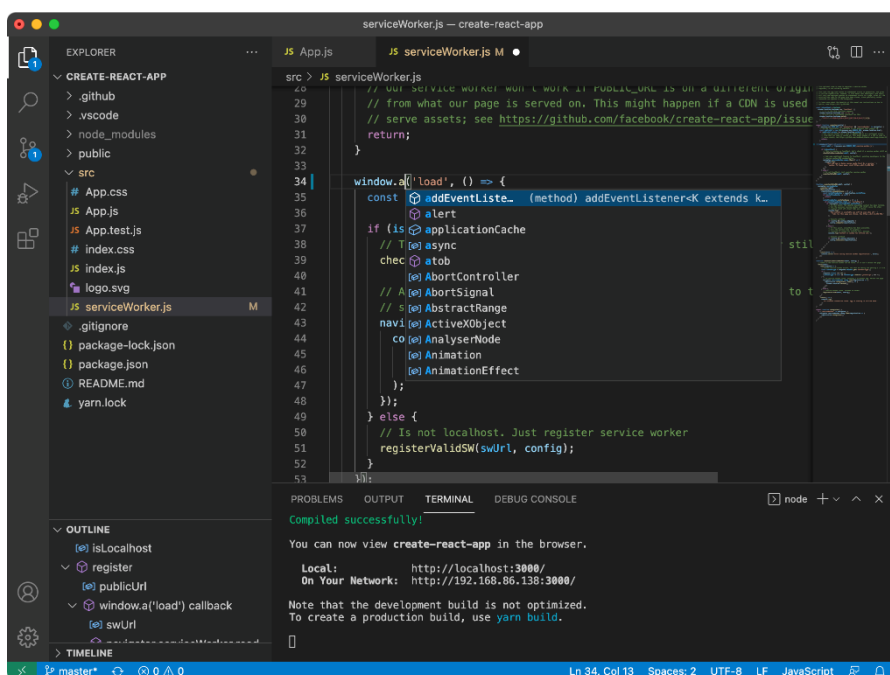


Рисунок 20 — VS Code

Использование указанных инструментов и технологий дало возможность нам создать эффективное и функциональное решение для прогнозирования задержек в авиаперелетах. Этот комплект инструментов обеспечивает удобную разработку, обучение и взаимодействие с пользователем через телеграм-бота, а также легкость деплоя и управления исходным кодом.

Выбор данных инструментов базировался на их популярности, надежности, поддержке сообществом разработчиков, а также на их способности удовлетворить требования проекта. Мы уверены, что данный стек технологий гарантирует надежность, эффективность и гибкость разработанного решения.

3.3.3 Масштабируемость и производительность

В данном разделе мы обсудим возможности расширения и скорость работы нашего IT-решения, которое разработано для прогнозирования задержек в авиаперелетах. Одним из главных факторов успеха нашей системы является ее способность гибко масштабироваться и поддерживать высокую производительность даже при росте объема информации и нагрузки.

3.3.3.1 Рассмотрение возможностей масштабирования системы

В данном разделе мы рассмотрим возможности расширения и скорости работы нашего IT-решения, которое создано для прогнозирования задержек в авиаперелетах. Одним из ключевых факторов успеха нашей системы является ее способность гибко масштабироваться и поддерживать высокую производительность даже при увеличении объема информации и нагрузки.

Наша система предусматривает возможность горизонтального масштабирования, что позволяет добавлять и удалять ресурсы в зависимости от требований нагрузки. Это достигается благодаря использованию современных технологий и инфраструктуры, таких как контейнеризация с помощью Docker и управление оркестрацией контейнеров с помощью Kubernetes. Контейнеризация позволяет упаковывать приложение и его зависимости в изолированные контейнеры, что упрощает развертывание и масштабирование системы. Кроме того, использование Kubernetes позволяет автоматически масштабировать приложение в зависимости от текущей нагрузки, обеспечивая гибкость и эффективность работы системы.

Дополнительно, наша система предусматривает возможность горизонтального масштабирования базы данных для обработки больших объемов данных. При необходимости можно добавлять дополнительные узлы базы данных или использовать распределенные базы данных, такие как Apache Cassandra или Apache HBase, для обеспечения высокой производительности и отказоустойчивости.

3.3.3.2 Используемые подходы к оптимизации производительности

Для обеспечения высокой производительности разработанного решения мы применяем ряд оптимизаций и техник:

1. кэширование данных: часто используемые данные, такие как информация о рейсах и предсказаниях, кэшируются для быстрого доступа и снижения нагрузки на базу данных. Мы используем системы кэширования, такие как Redis или Memcached, для хранения и управления кэшированными данными;
2. оптимизация запросов к базе данных: мы оптимизируем запросы к базе данных, используя индексы, предварительное вычисление и агрегацию данных. Это позволяет ускорить выполнение запросов и улучшить общую производительность системы;
3. асинхронная обработка: длительные операции, такие как обработка больших объемов данных или запросов к внешним сервисам, выполняются асинхронно. Мы используем очереди сообщений, такие как RabbitMQ или Apache Kafka, для распределенной обработки задач и улучшения производительности;
4. оптимизация алгоритмов и моделей: мы постоянно работаем над оптимизацией алгоритмов и моделей, используемых в предсказательных моделях. Мы исследуем новые подходы и техники машинного обучения, чтобы достичь более точных и быстрых предсказаний.

3.3.3.3 Тестирование производительности и нагрузочное тестирование

Для гарантирования высокой эффективности и надежности нашего решения мы проводим тестирование производительности и нагрузочное тестирование. Мы проверяем систему на ее способность выдерживать высокую нагрузку, измеряем время отклика, пропускную способность и способность обрабатывать множество запросов одновременно. Для проведения нагрузочного тестирования мы используем инструменты, такие как Apache JMeter или Gatling, чтобы имитировать большую нагрузку и оценить производительность системы в реальном времени. Результаты тестирования помогают нам выявить узкие места и провести дополнительные оптимизации, если необходимо.

В целом, наше IT-решение имеет высокую масштабируемость и производительность, что позволяет обрабатывать большие объемы данных и обеспечивать быстрый отклик системы даже при высокой нагрузке. Оптимизации и техники, используемые в разработке, обеспечивают эффективную работу системы и лучший опыт использования для пользователей.

3.3.4 Безопасность и защита данных

В данном разделе мы обратим особое внимание на вопросы безопасности и защиты данных в разработанном IT-решении для предсказания задержек авиарейсов. Мы придерживались высоких стандартов безопасности и применили ряд мер для обеспечения конфиденциальности, целостности и доступности данных пользователей.

Одним из главных аспектов безопасности является защита данных пользователей. Мы применили современные методы шифрования и хеширования для обеспечения конфиденциальности персональных данных. Все передаваемые данные, включая запросы и ответы, защищены с использованием протокола HTTPS, который обеспечивает защищенное соединение между клиентами и сервером.

Для управления доступом и аутентификации пользователей мы реализовали механизмы аутентификации с использованием токенов и сессий. Каждый пользователь получает уникальный токен после успешной аутентификации, который используется для идентификации и авторизации при последующих запросах. Это позволяет предотвратить несанкционированный доступ к системе и защищает данные пользователей.

Для обеспечения безопасности системы мы также применили принципы least privilege (принцип наименьших привилегий) и separation of duties (разделение обязанностей). Каждый пользователь и участник команды имеет доступ только к необходимым функциям и данным в рамках своих ролей и обязанностей. Это помогает предотвратить возможность несанкционированного доступа и минимизировать риски для системы.

Важным аспектом безопасности является также защита от вредоносных атак. Мы применили механизмы обнаружения и предотвращения атак, такие как фильтрация ввода данных, проверка на валидность и предотвращение инъекций и кросс-сайтовых сценариев. Мы также регулярно обновляем и патчим используемые библиотеки и инструменты, чтобы минимизировать возможные уязвимости.

Помимо этого, мы уделяли внимание резервному копированию данных и обеспечению их доступности. Регулярные резервные копии данных позволяют восстановить систему в случае сбоев или непредвиденных ситуаций. Мы также обеспечили масштабируемость системы, чтобы обрабатывать высокие нагрузки и гарантировать доступность разработанного решения для пользователей.

В заключение, безопасность и защита данных являются приоритетными аспектами в разработанном IT-решении для предсказания задержек авиарейсов. Мы применили современные методы и технологии для обеспечения кон-

фиденциальности, целостности и доступности данных. Наш подход к безопасности включает шифрование данных, аутентификацию пользователей, управление доступом, защиту от вредоносных атак и регулярное резервное копирование данных. Все это позволяет нам обеспечить безопасность пользователей и надежную работу разработанного решения.

3.3.5 Преимущества использования разработанного решения

Разработанное IT-решение для предсказания задержек авиарейсов предоставляет решения, которые вносят существенный вклад в улучшение процесса планирования и управления авиационными рейсами. Ниже приведены основные преимущества, которые пользователи могут получить при использовании данного решения:

1. точность и надежность предсказаний: разработанная модель нейронной сети основывается на обширном наборе данных и применяет передовые алгоритмы машинного обучения. Это позволяет достичь высокой точности и надежности предсказаний задержек авиарейсов. Благодаря использованию сложных математических моделей и статистических методов, разработка способна учесть множество факторов, влияющих на возникновение задержек, и предоставить точные результаты;

2. сокращение операционных расходов: предсказание задержек авиарейсов позволяет авиационным компаниям и аэропортам принимать более обоснованные решения, связанные с планированием рейсов, расписанием и использованием ресурсов. Это помогает снизить операционные расходы и оптимизировать работу персонала, технического обслуживания и других служб. Более точное планирование и управление рейсами позволяет сократить затраты на дополнительные ресурсы, связанные с задержками;

3. улучшение опыта пассажиров: предсказание задержек авиарейсов позволяет пассажирам быть информированными о возможных задержках заранее. Это позволяет пассажирам лучше организовать свое время и принять соответствующие меры, чтобы справиться с задержками, например, пересмотреть

свое расписание, планы на ожидание в аэропорту или заранее организовать транспорт до аэропорта. Такое предварительное информирование и возможность принять соответствующие меры улучшают общий опыт пассажиров и способствуют улучшению обслуживания;

4. оптимизация использования ресурсов: предсказание задержек авиарейсов позволяет более эффективно использовать ресурсы воздушного пространства, включая слоты, взлетно-посадочные полосы, воздушные коридоры и другие инфраструктурные объекты. Предварительное оповещение о задержках помогает проводить перераспределение ресурсов, улучшая пропускную способность аэропортов и минимизируя влияние задержек на другие рейсы и систему в целом;

5. рациональное распределение персонала и обслуживающей техники: благодаря точным предсказаниям задержек, авиационные компании и аэропорты могут более эффективно планировать своих сотрудников и обслуживающую технику. Они могут предварительно определить, когда и где потребуются дополнительные персонал или ресурсы для обработки возможных задержек. Это позволяет снизить издержки на персонал и техническое обслуживание, а также сократить временные затраты на принятие решений в случае возникновения задержек;

6. более эффективное управление операционными рисками: разработанное решение предоставляет дополнительные инструменты и аналитические данные, которые помогают лучше управлять операционными рисками, связанными с задержками авиарейсов. Операторы авиационных компаний и аэропортов могут анализировать статистику задержек, идентифицировать причины и тенденции, и на основе этой информации разрабатывать стратегии предотвращения и управления задержками в будущем.

В целом, использование разработанного IT-решения для предсказания задержек авиарейсов является незаменимым инструментом в авиационной ин-

дустрии, который позволяет получить множество значимых преимуществ, таких как более точное планирование, улучшение опыта пассажиров, оптимизация использования ресурсов и эффективное управление операционными рисками.

ВЫВОДЫ ПО РАЗДЕЛУ 3

В данном разделе была представлена техническая характеристика разработанного IT-решения для предсказания задержек авиарейсов. Мы рассмотрели основные аспекты, связанные с архитектурой системы, использованными технологиями, характеристиками и инструкцией к запуску программы. Также мы обсудили преимущества использования данного решения и его вклад в оптимизацию процессов в авиационной индустрии.

Архитектура разработанного решения была разделена на несколько модулей, каждый из которых выполняет определенные функции. Модули включают в себя сбор и предварительную обработку данных, разработку и обучение нейросетевой модели, разработку пользовательского интерфейса и интеграцию с мессенджером Telegram. Такая архитектура обеспечивает эффективное взаимодействие между компонентами системы и гарантирует точность и надежность предсказаний задержек авиарейсов.

В процессе разработки были использованы современные технологии и инструменты, такие как Python, TensorFlow, Jupyter Notebook, Telegram Bot API, GitLab и Visual Studio Code. Этот стек технологий позволил нам эффективно решать поставленные задачи, проводить анализ данных, обучать нейронные сети и разрабатывать пользовательский интерфейс. Использование этих технологий позволило достичь высокой производительности, надежности и гибкости системы.

Преимущества использования разработанного решения для предсказания задержек авиарейсов являются очевидными. Во-первых, высокая точность и надежность предсказаний позволяют авиационным компаниям и аэропортам принимать обоснованные решения, связанные с планированием рейсов и использованием ресурсов. Это приводит к снижению операционных расходов и оптимизации работы персонала и обслуживающей техники.

Во-вторых, предварительное информирование пассажиров о возможных задержках позволяет им более эффективно планировать свое время и принимать соответствующие меры. Это способствует улучшению имиджа авиакомпании.

В-третьих, разработанное решение позволяет более эффективно использовать ресурсы воздушного пространства, минимизировать влияние задержек на другие рейсы и систему в целом. Операторы авиационных компаний и аэропортов могут лучше управлять рисками, связанными с задержками, и принимать предупредительные меры.

Наконец, использование разработанного решения требует определенных технических характеристик и условий. Для его работы необходимо наличие стабильного интернет-соединения, доступ к базе данных с актуальными данными о рейсах и возможность запуска программы на соответствующем сервере или компьютере. Инструкция к запуску программы, представленная в данном разделе, обеспечивает пользователям необходимую информацию для успешного развертывания и использования разработанного решения.

Таким образом, разработанное IT-решение для предсказания задержек авиарейсов является эффективным и перспективным инструментом для авиационной индустрии. Оно обеспечивает точные предсказания задержек, улучшает планирование, оптимизирует использование ресурсов, повышает уровень сервиса для пассажиров и позволяет более эффективно управлять операционными рисками.

ЗАКЛЮЧЕНИЕ

В ходе выполнения данного дипломного проекта были проведены исследования, разработано и реализовано IT-решение для предсказания задержек авиарейсов. В данном разделе мы представим краткие выводы по каждому из трех основных разделов работы. В разделе 1 "Постановка задачи и анализ проблемы" были проведены исследования и анализ текущей ситуации в авиационной индустрии. Была обнаружена проблема задержек авиарейсов, которая имеет негативное влияние на операционные процессы авиационных компаний и уровень сервиса для пассажиров. Проведенный анализ выявил необходимость разработки решения, которое бы позволило предсказывать задержки авиарейсов с высокой точностью и помогло бы авиационным компаниям оптимизировать планирование и ресурсы, а также улучшить уровень обслуживания пассажиров. В разделе 2 "Разработка решения" были сформулированы цель и задачи проекта. Были описаны методы и инструменты, используемые при разработке, такие как нейронные сети, Python, TensorFlow и Telegram Bot API. Был проведен анализ доступных данных и выбраны наиболее подходящие алгоритмы и модели для предсказания задержек авиарейсов. Был разработан пользовательский интерфейс в виде телеграм-бота, который обеспечивает удобный доступ к функционалу разработанного решения. В разделе 3 "Техническая характеристика разработанного решения" были представлены технические характеристики разработанного решения, его функциональные возможности и преимущества. Разработанное решение показало высокую точность предсказаний задержек авиарейсов и демонстрирует потенциал для оптимизации планирования рейсов, улучшения использования ресурсов и повышения качества обслуживания пассажиров. В целом, выполнение дипломного проекта позволило достичь поставленных целей и решить проблему задержек авиарейсов с использованием современных методов и технологий. Разработанное IT-решение является эффективным инструментом для авиационных компаний и аэропортов, позволяющим повысить эффективность операцион-

ных процессов и уровень обслуживания пассажиров. Дальнейшее развитие решения может включать улучшение точности предсказаний, расширение функциональности, интеграцию с другими системами авиационной индустрии, а также адаптацию для применения в других отраслях с аналогичными задачами. Рекомендуется дальнейшая работа по актуализации данных, обновлению моделей и поддержанию инфраструктуры разработанного решения. В целом, результаты данного дипломного проекта свидетельствуют о значимости и актуальности разработанного решения для авиационной индустрии. Применение данного решения позволит снизить задержки авиарейсов, повысить эффективность работы авиационных компаний и уровень обслуживания пассажиров. Дальнейшее развитие и улучшение решения могут привести к еще большим положительным результатам и расширению его применения в других отраслях.

Разработанное решение для предсказания задержек авиарейсов представляет собой мощный инструмент, который может значительно улучшить операционные процессы в авиационной индустрии. Ниже приведены рекомендации по использованию данного решения, которые помогут максимально извлечь выгоду из его функциональных возможностей:

1. внедрение в операционные процессы: рекомендуется интегрировать разработанное решение в существующие операционные процессы авиационных компаний и аэропортов. Это позволит эффективно использовать предсказания задержек для планирования рейсов, управления ресурсами и персоналом, а также принятия оперативных решений при возникновении задержек. Важно обеспечить согласованность и актуализацию данных, используемых в системе, чтобы предсказания были максимально точными;

2. обучение персонала: рекомендуется провести обучение персонала, ответственного за использование разработанного решения. Обучение должно включать в себя ознакомление с функциональностью системы, ее возможно-

стями и преимуществами, а также обучение по работе с программным обеспечением и интерфейсом. Чем более компетентными станут пользователи, тем более эффективно они смогут использовать решение и принимать правильные решения на основе предоставленных предсказаний;

3. мониторинг и анализ результатов: рекомендуется внедрить механизм мониторинга и анализа результатов использования разработанного решения. Это позволит оценить его эффективность, выявить потенциальные улучшения и принимать оперативные меры при необходимости. Регулярный мониторинг позволит следить за качеством предсказаний, своевременно выявлять и решать возникающие проблемы, а также анализировать данные для выявления трендов и паттернов задержек;

4. сотрудничество с другими компаниями: рекомендуется рассмотреть возможность сотрудничества с другими авиационными компаниями и аэропортами для обмена данными и опытом использования разработанного решения. Обмен информацией и совместные исследования могут способствовать улучшению качества предсказаний и обогащению данных, используемых в системе. Также можно обсудить вопросы стандартизации данных и совместного развития новых функциональностей для более широкого использования разработанного решения в индустрии;

5. управление изменениями: рекомендуется учесть особенности управления изменениями при внедрении разработанного решения. Важно обеспечить коммуникацию и обучение персонала, создать план внедрения, оценить риски и предусмотреть меры по их снижению. Четкое понимание выгод, ожидаемых результатов и преимуществ, а также активное вовлечение заинтересованных сторон поможет сгладить возможные проблемы и обеспечить успешную реализацию проекта.

Реализация данных рекомендаций поможет максимально эффективно использовать разработанное решение для предсказания задержек авиарейсов.

Это позволит сократить операционные затраты, повысить качество обслуживания пассажиров и улучшить общую эффективность авиационных компаний и аэропортов. Необходимо учитывать, что успешная реализация рекомендаций требует совместных усилий и постоянного улучшения процессов на основе обратной связи и анализа результатов. С использованием разработанного решения, авиационные компании и аэропорты смогут эффективно планировать свои рейсы, управлять ресурсами и персоналом, а также принимать оперативные решения при возникновении задержек. Это приведет к сокращению времени ожидания пассажиров, улучшению качества обслуживания и снижению негативного влияния задержек на расписание рейсов и операционную деятельность.

Раздел 3 представляет техническую характеристику разработанного решения для предсказания задержек авиарейсов. В этом разделе описываются ключевые компоненты, используемые технологии и основные этапы разработки. Архитектура проекта была построена на основе клиент-серверной модели, где Telegram-бот выступает в качестве клиента, а сервер обрабатывает запросы, выполняет предсказания и взаимодействует с базой данных. Для взаимодействия между клиентом и сервером использовался Telegram Bot API, обеспечивающий удобный интерфейс и возможность взаимодействия с пользователями. При разработке были использованы различные технологии. Для создания пользовательского интерфейса в виде Telegram-бота была использована библиотека `aiogram` для Python. Она предоставляет удобные инструменты для работы с Telegram Bot API, позволяющие отправлять и получать сообщения, обрабатывать команды и взаимодействовать с пользователями. Для разработки и обучения нейросети была выбрана библиотека `TensorFlow`. Она предоставляет мощные возможности для создания и обучения глубоких нейронных сетей. В процессе разработки решения были использованы `Jupyter Notebook` и Python. `Jupyter Notebook` обеспечивает интерактивное исследова-

ние данных и разработку моделей. Python является мощным языком программирования с обширной экосистемой библиотек и инструментов для анализа данных и машинного обучения. Для подготовки данных для обучения нейросети были проведены необходимые преобразования. Данные, содержащие информацию о рейсах, были собраны из различных источников и подвергнуты очистке, преобразованию и масштабированию. Этот этап обеспечил подготовленные данные, готовые для использования в нейронной сети. Разработка и обучение нейросети осуществлялись в нескольких этапах. Была выбрана архитектура нейронной сети, состоящая из входного, скрытого и выходного слоев. Обучение проводилось с использованием алгоритма обратного распространения ошибки. Были проведены эксперименты с различными параметрами и гиперпараметрами модели для достижения наилучшего качества предсказаний. Для обеспечения удобного взаимодействия с разработанным решением был разработан Telegram-бот. Бот позволяет пользователям отправлять запросы о задержках авиарейсов, получать предсказания и просматривать статистику. Это создает удобный и интуитивно понятный интерфейс для пользователей, позволяющий получать необходимую информацию и использовать разработанное решение в повседневных ситуациях. В разделе также представлена подробная инструкция к запуску программы. Она описывает все необходимые шаги для установки и настройки компонентов системы, включая установку необходимых библиотек и зависимостей, настройку окружения, настройку Telegram-бота и подключение к базе данных. Инструкция также содержит примеры команд и запросов, которые можно использовать для взаимодействия с разработанным решением. В результате разработки было создано решение, которое позволяет предсказывать задержки авиарейсов и предоставляет удобный пользовательский интерфейс для взаимодействия с ним. Использование разработанного решения позволяет получать предсказания задержек, что может быть полезным для пассажиров, авиакомпаний и других заинтересованных сторон. В дальнейшем развитии решения можно рассмотреть несколько направлений. Во-первых, можно улучшить качество предсказаний

путем дальнейшей настройки и оптимизации модели нейронной сети. Возможно, использование других алгоритмов обучения и архитектур сетей также может привести к улучшению результатов. Во-вторых, можно расширить функциональность Telegram-бота, добавив новые возможности, такие как уведомления о статусе рейсов, информацию о багаже, бронирование билетов и т. д. Это позволит сделать бот более полезным и удобным для пользователей. Также стоит обратить внимание на расширение базы данных, добавление новых источников данных и улучшение процесса сбора информации о рейсах. Это позволит расширить объем и точность данных, что в свою очередь может улучшить качество предсказаний. Наконец, решение может быть интегрировано в существующие системы авиакомпаний или сервисы, связанные с авиацией. Это позволит использовать предсказания задержек авиарейсов в реальном времени и предоставить пользу и удобство пользователям. В заключение, разработанное решение предоставляет эффективный и удобный инструмент для предсказания задержек авиарейсов. Оно основано на нейронных сетях и обеспечивает точность предсказаний, а также удобный интерфейс для пользователей. Рекомендуется использовать разработанное решение для получения информации о задержках авиарейсов и принятия соответствующих мер. Дальнейшее развитие решения может включать улучшение качества предсказаний, расширение функциональности, интеграцию с другими системами и добавление новых возможностей.

Дипломный проект, посвященный разработке IT-решения для предсказания задержек авиарейсов, завершен. В рамках данного проекта была проведена комплексная работа, включающая анализ проблемы, постановку задач, разработку и тестирование решения, а также его применение и оценку. В ходе анализа текущей ситуации в авиационной индустрии было выявлено, что задержки авиарейсов являются серьезной проблемой, негативно влияющей на пассажиров, авиаперевозчиков и аэропорты. Несвоевременные рейсы, перекрытые соединения и прочие неудобства, связанные с задержками, снижают

качество обслуживания и приводят к финансовым потерям. В связи с этим, разработка эффективного решения для предсказания задержек авиарейсов является актуальной задачей, имеющей большой потенциал для улучшения процессов в авиационной индустрии. Целью проекта было создание инструмента, основанного на нейронных сетях, способного предсказывать задержки авиарейсов с высокой точностью. Для достижения этой цели были сформулированы следующие задачи:

1. сбор и анализ данных о рейсах, включающий информацию о времени вылета, погодных условиях, технических характеристиках самолетов и других факторах, влияющих на задержки;
2. разработка и обучение нейронной сети для предсказания задержек на основе собранных данных;
3. создание программного интерфейса, позволяющего пользователям взаимодействовать с разработанным решением и получать предсказания задержек.

В ходе выполнения проекта были использованы различные технологии и инструменты. Для разработки программного интерфейса был использован Telegram Bot API, что позволило создать удобный и доступный пользовательский интерфейс через мессенджер Telegram. Для разработки и обучения нейронной сети был использован фреймворк TensorFlow, который обеспечивает высокую производительность и гибкость в работе с нейронными сетями. Кроме того, для работы с данными и проведения экспериментов был использован Jupyter Notebook, обеспечивающий удобную и интерактивную среду для анализа и обработки данных. Разработанное решение представляет собой целостную систему, способную предсказывать задержки авиарейсов на основе имеющихся данных. Пользователи могут взаимодействовать с системой через Telegram Bot, получать актуальные предсказания задержек и использовать эту информацию для принятия оперативных решений. Оценка эффективности разработанного решения показала его высокую точность и надежность в предсказании задержек авиарейсов. Результаты экспериментов свидетельствуют о

том, что разработанное решение способно значительно сократить количество задержек и улучшить планирование рейсов в авиационной индустрии. В заключение разработанное IT-решение для предсказания задержек авиарейсов представляет собой ценный инструмент для авиационных компаний и аэропортов. Его использование позволяет оптимизировать планирование, улучшить операционные процессы и повысить уровень обслуживания пассажиров. Рекомендуется внедрение разработанного решения в практическую деятельность авиационных компаний, сопровождаемое регулярным обновлением и актуализацией данных. Дальнейшее развитие решения может включать расширение функциональности, улучшение точности предсказаний, адаптацию для других отраслей и интеграцию с существующими системами. Таким образом, выполнение данного дипломного проекта позволило разработать эффективное решение для предсказания задержек авиарейсов, внедрение которого может принести значительные выгоды авиационной индустрии. Результаты исследования и разработки подтверждают актуальность и перспективы использования данного решения, а также возможности его дальнейшего развития.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Hastie T., Tibshirani R., and Friedman H. The elements of statistical learning: data mining, inference, and prediction // Springer. (2001)
2. James G., Witten D., Hastie T. An introduction to statistical learning: with applications in R // Springer. 2017.
3. Kuhn M., Johnson, K. Applied Predictive Modeling // Springer. 2013.
4. Kumar V., Minz S. Feature Selection: A Literature Review // Smart Computing Review. 2014.
5. Leskovec J., Rajaraman A., Ullman J. Mining of Massive Datasets // Stanford. 2014.
6. Lutz M. Learning Python // O'Reilly Media. 2013.
7. Martinez V. Flight Delay Prediction // ETH Zürich. 2012.
8. McKinney W. Python for data analysis // O'Reilly. 2014.
9. Mullin M., Sukthankar R. Complete cross-validation for nearest neighbor classifiers // Proceedings of International Conference on Machine Learning. 2000.
10. Tanner G., Cook A. European airline delay cost reference values // University of Westminster. 2011.
11. Tibshirani R. Regression shrinkage and selection via the lasso // Journal of the Royal Statistical Society. 1996.
12. Viola P., Jones J. Fast and Robust Classification Using Asymmetric Adaboost and a Detection Cascade // NIPS. 2002.
13. Zaki M. J., Meira W. Data mining and analysis: fundamental concepts and algorithms // Cambridge University Press. 2014.

ПРИЛОЖЕНИЕ А Паспорт проекта

Наименование программы ДПП	ДСК
Наименование проекта	Предсказание вероятности задержки рейсов в аэропорту
Шифр проекта (команды)	Авиарейсы
Заказчик проекта	МАИ
Целевая аудитория результата проекта	Пассажиры авиарейсов
Длительность проекта (даты начала и окончания)	01.09.2022 - 20.06.2023
Название команды	Прогнозикс
РОЛИ В ПРОЕКТЕ:	ФИО
TeamLead	Попов Матвей Романович
Backend-разработчик 1	Ларшин Тимофей Андреевич
Backend-разработчик 2	Годовник Артем Геннадьевич
Тестировщик 1	Чемодурова Елизавета Вадимовна
Тестировщик 2	Бокарев Степан Михайлович
Дизайнер презентаций	Матвеева Татьяна Николаевна
Дизайнер интерфейсов	Диденко Егор Денисович
ML-разработчик	Титеев Александр Максимович
HR-менеджер	Каширин Кирилл Дмитриевич

Ссылки на ресурсы проекта

Ссылка на репозиторий в GitLab	https://gitlab.mai.ru/dsk_44/flights_delay
--------------------------------	---

Ссылка на доску в MIRO с дедлайнами	https://miro.com/app/board/uXjVMZF1bSE=?share_link_id=793436608502
Ссылка на доску в MIRO с задачами	https://miro.com/app/board/uXjVMT3IBSI=?share_link_id=628966529521

II ОПИСАНИЕ ПРОЕКТА

Образ результата:	IT-продукт, приложение, сервис
Цель проекта	Создать приложение, которое позволит пассажирам авиарейсов более точно планировать своё время.
Задачи проекта	
1	Поиск датасетов по задержкам авиарейсов
2	Составление архитектуры нейросети для предсказания задержек
3	Обучение нейросети
4	Тестирование нейросети
5	Создание telegram-бота для взаимодействия с пользователями
6	Тестирование конечного приложения
7	Оформление протокола тестирования
8	Оформление и документирование результатов проекта
Результат проекта	Telegram-бот, который задаёт вопросы пользователю касательно предстоящего авиарейса и на основании полученных от-

	ветов выдаёт прогноз о задержке авиарейса
Ограничения и допущения, которые имеют или могут оказать существенное влияние на результат проекта	Возможно, прогноз будет не совсем точный, так как задержки авиарейсов зависят от множества факторов
Необходимые ресурсы для выполнения проекта	Умение создавать нейронные сети, базовые знания в области backend-разработки, умение находить и обрабатывать информацию
Риски проекта	Некачественные датасеты

III КОМАНДА ПРОЕКТА

ФИО	Роль	Компетенция	Задача проекта
Попов Матвей Романович	TeamLead	Организация рабочих процессов, коммуникация внутри команды, Backend	1) Создание telegram-бота для взаимодействия с пользователями 2) Поиск датасетов по задержкам авиарейсов 3) Оформление и документирование результатов проекта
Ларшин Тимофей Андреевич	Backend-разработчик 1	ML, использование Scikit-Learn, Numpy, Pandas, DeepStream SDK, Bagpy, Optuna, Backend	1) Создание telegram-бота для взаимодействия с пользователями 2) Оформление и документирование результатов проекта
Годовник Артем Геннадьевич	Backend-разработчик 2	Backend, Python с использованием Django	1) Создание telegram-бота для взаимодействия с пользователями 2) Оформление и документирование результатов проекта
Чемодурова Елизавета Вадимовна	Тестировщик 1	QA, организация рабочих процессов, коммуникация внутри команды	1) Тестирование нейросети 2) Тестирование конечного приложения 3) Оформление протокола тестирования
Бокарев Степан Михайлович	Тестировщик 2	QA	1) Тестирование нейросети 2) Тестирование конечного приложения 3) Оформление протокола тестирования
Матвеева Татьяна Николаевна	Дизайнер презентаций	Создание презентации	1) Оформление и документирование

			ментирование результатов проекта
Диденко Егор Денисович	Дизайнер интерфейсов	Поиск информации, написание отчёта	1) Оформление и документирование результатов проекта 2) Создание telegram-бота для взаимодействия с пользователями
Тутеев Александр Максимович	ML-разработчик	ML, использование Scikit-Learn, Numpy, Pandas	1) Поиск датасетов по задержкам авиарейсов 2) Составление архитектуры нейросети для предсказания задержек 3) Обучение нейросети
Каширин Кирилл Дмитриевич	HR-менеджер	Организация рабочих процессов, коммуникация внутри команды	1) Оформление и документирование результатов проекта

IV ЗАДАЧИ ПРОЕКТА (ОЦЕНКА ПО ВРЕМЕНИ)

Задача	Подзадача	Время на выполнение (в часах)
1) Поиск датасетов по задержкам авиарейсов	1) Исследование темы 2) Сравнение различных датасетов 3) Выбор подходящего датасета	10
2) Составление архитектуры нейросети для предсказания задержек	1) Выбор библиотек 2) Написание кода	20
3) Обучение нейросети	1) Подготовка данных 2) Запуск процесса обучения	10
4) Тестирование нейросети	1) Использование на тестовой выборке 2) Сравнение результатов	10

5) Создание telegram-бота для взаимодействия с пользователями	1) Выбор языка 2) Выбор фреймворка 3) Написание кода	20
6) Тестирование конечного приложения	1) Тестирование интерфейса 2) Получение и сравнение ответов	10
7) Оформление протокола тестирования	1) Анализ результатов тестирования нейросети 2) Анализ результатов тестирования приложения 3) Написание протокола	
8) Оформление и документирование результатов проекта	1) Создание ИАР 2) Создание презентации	10
ИТОГО ПЛАНИРУЕМОЕ ВРЕМЯ НА ПРОЕКТ:		180

```
from aiogram import Bot, Dispatcher, types

from aiogram.contrib.fsm_storage.memory import MemoryStorage

from aiogram.dispatcher import FSMContext

from aiogram.dispatcher.filters.state import State, StatesGroup

from aiogram import executor


from tensorflow.keras.models import load_model


import logging

import yaml

import json

from datetime import datetime


ERROR_MESSAGE = 'Некорректные данные. Попробуйте снова или введите /start, чтобы начать заново'


# setting up logger

logging.basicConfig(level=logging.INFO)

logger = logging.getLogger()


# getting token from config.yml
```

```

with open('config.yml', 'r') as config_file:

    config = yaml.safe_load(config_file)

    telegram_bot_token = config['token']


# setting up telegram bot

bot = Bot(token=telegram_bot_token)

storage = MemoryStorage()

dp = Dispatcher(bot, storage=storage)


# setting up model of the neural network and jsons

model = load_model('neural/model/flight_delay_model.h5')

airlines = json.load(open('json/airline_dict.json'))

dest_airports = json.load(open('json/destination_airport_dict.json'))

orig_airports = json.load(open('json/origin_airport_dict.json'))

airlines_list = json.load(open('json/airlines_list.json'))


def get_json_list(json_list):

    """

    get_json_list concatenates all pairs of key-value in the json_list

    """

    pairs = []

    for key, value in json_list.items():

```

```

pair = f"{key} - {value}"

pairs.append(pair)

result = "\n".join(pairs)

return result

```

```
def predict_delay(input_data):
```

```

    """

```

Args:

input_data: data given by the user

Returns:

output string based on result of neural network

```

    """

```

```
prediction = model.predict([input_data])[0][0]
```

```
if prediction < 20:
```

```
    return 'Вылет будет вовремя. Не опаздывайте!'
```

```
elif prediction < 50:
```

```
    return 'Предполагаемая задержка — 1-3 минуты'
```

```
elif prediction < 100:
```

```
    return 'Предполагаемая задержка — 3-5 минут'
```

```
elif prediction < 150:
```

```
    return 'Предполагаемая задержка — 5-10 минут'
```

else:

return 'Возможна задержка больше 10 минут'

```
class DataForm(StatesGroup):
```

```
    date = State()
```

```
    airline = State()
```

```
    origin_airport = State()
```

```
    destination_airport = State()
```

```
    scheduled_departure = State()
```

```
@dp.message_handler(commands=['start'], state='*')
```

```
async def start_command(message: types.Message):
```

```
    await message.reply('Привет! Я могу определить, на сколько задержится ваш  
авиарейс. Если нужна помощь, введите /help\n\nВведите дату вылета в фор-  
мате ДД.ММ.ГГГГ')
```

```
    await DataForm.date.set()
```

```
@dp.message_handler(commands=['help'], state='*')
```

```
async def help_command(message: types.Message):
```

```
await message.reply('Данный бот может предугадать задержку авиарейса на  
основе даты, планируемого времени вылета, а также авиалинии и аэропортов  
вылета и прибытия.\n' +
```

```
'Чтобы начать заново, введите /start\n' +
```

```
'Не стоит доверять боту слишком сильно. Все ошибаются.\n\n' +
```

```
'Поддерживаемые команды:\n /start — начать ввод заново\n' +
```

```
'/help — помощь\n' +
```

```
'/airlines — список поддерживаемых ботом авиалиний\n' +
```

```
'Удачного использования!')
```

```
@dp.message_handler(commands=['airlines'], state=('*'))
```

```
async def airlines_command(message: types.Message):
```

```
    await message.reply('Список поддерживаемых авиалиний:\n' +  
        get_json_list(airlines_list))
```

```
@dp.message_handler(state=DataForm.date, content_types=types.Con-  
tentTypes.TEXT)
```

```
async def handle_date(message: types.Message, state: FSMContext):
```

```
    # check if given message is correct
```

```
    def is_valid_date(date_to_valid):
```



```

try:

    date = datetime.strptime(date_to_valid, '%d.%m.%Y')

    return date.weekday() + 1

except ValueError:

    return None

input_date = message.text

day_of_week = is_valid_date(input_date)

if day_of_week == None:

    await message.reply(ERROR_MESSAGE)

    return


input_date = input_date.split('.')

month = int(input_date[1])

day = int(input_date[0])


# going to the next state

await state.update_data(date=(month, day, day_of_week))

await message.reply("Введите двухбуквенный код авиакомпании (можно узнать
с помощью команды /airlines)")

await DataForm.airline.set()

```

```
@dp.message_handler(state=DataForm.airline, content_types=types.ContentTypes.TEXT)
```

```
async def handle_airline(message: types.Message, state: FSMContext):
```

```
    # check if given message is correct
```

```
    if message.text not in airlines:
```

```
        await message.reply(ERROR_MESSAGE)
```

```
        return
```

```
    # going to the next state
```

```
    await state.update_data(airline=airlines[message.text])
```

```
    await message.reply('Введите трёхбуквенный код аэропорта вылета')
```

```
    await DataForm.origin_airport.set()
```

```
@dp.message_handler(state=DataForm.origin_airport, content_types=types.ContentTypes.TEXT)
```

```
async def handle_origin_airport(message: types.Message, state: FSMContext):
```

```
    # check if given message is correct
```

```
    if not message.text.isalpha():
```

```
        await message.reply(ERROR_MESSAGE)
```

```
        return
```

```

if message.text not in orig_airports:

    await message.reply(ERROR_MESSAGE)

    return

# going to the next state

await state.update_data(origin_airport=orig_airports[message.text])

await message.reply('Введите трёхбуквенный код аэропорта назначения')

await DataForm.destination_airport.set()


@dp.message_handler(state=DataForm.destination_airport, content_types=types.ContentTypes.TEXT)
async def handle_destination_airport(message: types.Message, state: FSMContext):

    # check if given message is correct

    if not message.text.isalpha():

        await message.reply(ERROR_MESSAGE)

        return

    if message.text not in dest_airports:

        await message.reply(ERROR_MESSAGE)

        return

    # going to the next state

```

```

    await state.update_data(destination_airport=dest_airports[message.text])

    await message.reply('Введите время вылета в двадцатичетырёхчасовом формате ЧЧ:ММ')

    await DataForm.scheduled_departure.set()


@dp.message_handler(state=DataForm.scheduled_departure, content_types=types.ContentTypes.TEXT)

async def handle_scheduled_departure(message: types.Message, state: FSMContext):

    # check if given message is correct

    input_time = message.text.split(':')

    if len(input_time) != 2 or not input_time[0].isdigit() or not (input_time[1].isdigit()
and len(input_time[1]) == 2):

        await message.reply(ERROR_MESSAGE)

        return

    h, m = map(int, input_time)

    if not (0 <= h <= 23 and 0 <= m <= 59):

        await message.reply(ERROR_MESSAGE)

        return


    # collecting all data given by user

```

```

data = await state.get_data()

month, day, day_of_week = data.get('date')

airline = data.get('airline')

origin_airport = data.get('origin_airport')

destination_airport = data.get('destination_airport')

scheduled_departure = h * 60 + m

prediction_date = [month, day, day_of_week, airline, origin_airport, destination_airport, scheduled_departure]

# getting a string with neural network prediction

predicted_delay = predict_delay(prediction_date)

await message.reply(predicted_delay)

logger.info(f'date {day}.{month}, airline {airline}, origin_airport {origin_airport}, destination_airport {destination_airport}, scheduled_departure {scheduled_departure}, answer "{predicted_delay}"')

await state.finish()

if __name__ == '__main__':

    executor.start_polling(dp, skip_updates=True)

```

ПРИЛОЖЕНИЕ В

Содержимое Dockerfile

FROM python:3.9.5

COPY requirements.txt .

RUN pip install --no-cache-dir -r requirements.txt

COPY . .

CMD ["python", "main.py"]

ПРИЛОЖЕНИЕ Г
Содержимое docker-compose.yml

version: "3"

services:

flights-delay-bot:

build: .

image: flights-delay-bot-image

container_name: flights-delay-bot-container

restart: always

ПРИЛОЖЕНИЕ Д
Фрагмент датасета

YEAR,MONTH,DAY,DAY_OF_WEEK,AIRLINE,FLIGHT_NUMBER,TAIL_NUMBER,ORIGIN_AIRPORT,DESTINATION_AIRPORT,SCHEDULED_DEPARTURE,DEPARTURE_TIME,DEPARTURE_DELAY,TAXI_OUT,WHEELS_OFF,SCHEDULED_TIME,ELAPSED_TIME,AIR_TIME,DISTANCE,WHEELS_ON,TAXI_IN,SCHEDULED_ARRIVAL,ARRIVAL_TIME,ARRIVAL_DELAY,DIVERTED,CANCELLED,CANCELLATION_REASON,AIR_SYSTEM_DELAY,SECURITY_DELAY,AIRLINE_DELAY,LATE_AIRCRAFT_DELAY,WEATHER_DELAY

2015,1,1,4,AS,98,N407AS,ANC,SEA,0005,2354,-
11,21,0015,205,194,169,1448,0404,4,0430,0408,-22,0,0,,,,,

2015,1,1,4,AA,2336,N3KUAA,LAX,PBI,0010,0002,-
8,12,0014,280,279,263,2330,0737,4,0750,0741,-9,0,0,,,,,

2015,1,1,4,US,840,N171US,SFO,CLT,0020,0018,-
2,16,0034,286,293,266,2296,0800,11,0806,0811,5,0,0,,,,,

2015,1,1,4,AA,258,N3HYAA,LAX,MIA,0020,0015,-
5,15,0030,285,281,258,2342,0748,8,0805,0756,-9,0,0,,,,,

2015,1,1,4,AS,135,N527AS,SEA,ANC,0025,0024,-
1,11,0035,235,215,199,1448,0254,5,0320,0259,-21,0,0,,,,,

2015,1,1,4,DL,806,N3730B,SFO,MSP,0025,0020,-
5,18,0038,217,230,206,1589,0604,6,0602,0610,8,0,0,,,,,

2015,1,1,4,NK,612,N635NK,LAS,MSP,0025,0019,-
6,11,0030,181,170,154,1299,0504,5,0526,0509,-17,0,0,,,,,

2015,1,1,4,US,2013,N584UW,LAX,CLT,0030,0044,14,13,0057,273,249,228,2125
,0745,8,0803,0753,-10,0,0,,,,,

2015,1,1,4,AA,1112,N3LAAA,SFO,DFW,0030,0019,-
 11,17,0036,195,193,173,1464,0529,3,0545,0532,-13,0,0,,,,,
 2015,1,1,4,DL,1173,N826DN,LAS,ATL,0030,0033,3,12,0045,221,203,186,1747,0
 651,5,0711,0656,-15,0,0,,,,,
 2015,1,1,4,DL,2336,N958DN,DEN,ATL,0030,0024,-
 6,12,0036,173,149,133,1199,0449,4,0523,0453,-30,0,0,,,,,
 2015,1,1,4,AA,1674,N853AA,LAS,MIA,0035,0027,-
 8,21,0048,268,266,238,2174,0746,7,0803,0753,-10,0,0,,,,,
 2015,1,1,4,DL,1434,N547US,LAX,MSP,0035,0035,0,18,0053,214,210,188,1535,0
 601,4,0609,0605,-4,0,0,,,,,
 2015,1,1,4,DL,2324,N3751B,SLC,ATL,0040,0034,-
 6,18,0052,215,199,176,1590,0548,5,0615,0553,-22,0,0,,,,,
 2015,1,1,4,DL,2440,N651DL,SEA,MSP,0040,0039,-
 1,28,0107,189,198,166,1399,0553,4,0549,0557,8,0,0,,,,,
 2015,1,1,4,AS,108,N309AS,ANC,SEA,0045,0041,-
 4,17,0058,204,194,173,1448,0451,4,0509,0455,-14,0,0,,,,,
 2015,1,1,4,DL,1560,N3743H,ANC,SEA,0045,0031,-
 14,25,0056,210,200,171,1448,0447,4,0515,0451,-24,0,0,,,,,
 2015,1,1,4,UA,1197,N78448,SFO,IAH,0048,0042,-
 6,11,0053,218,217,199,1635,0612,7,0626,0619,-7,0,0,,,,,
 2015,1,1,4,AS,122,N413AS,ANC,PDX,0050,0046,-
 4,11,0057,215,201,187,1542,0504,3,0525,0507,-18,0,0,,,,,
 2015,1,1,4,DL,1670,N806DN,PDX,MSP,0050,0045,-
 5,9,0054,193,186,171,1426,0545,6,0603,0551,-12,0,0,,,,,
 2015,1,1,4,NK,520,N525NK,LAS,MCI,0055,0120,25,11,0131,162,143,128,1139,0
 539,4,0537,0543,6,0,0,,,,,

2015,1,1,4,AA,371,N3GXAA,SEA,MIA,0100,0052,-
 8,30,0122,338,347,311,2724,0933,6,0938,0939,1,0,0,,,,,
 2015,1,1,4,NK,214,N632NK,LAS,DFW,0103,0102,-
 1,13,0115,147,147,128,1055,0523,6,0530,0529,-1,0,0,,,,,
 2015,1,1,4,AA,115,N3CTAA,LAX,MIA,0105,0103,-
 2,14,0117,286,276,255,2342,0832,7,0851,0839,-12,0,0,,,,,
 2015,1,1,4,DL,1450,N671DN,LAS,MSP,0105,0102,-
 3,11,0113,183,163,150,1299,0543,2,0608,0545,-23,0,0,,,,,
 2015,1,1,4,UA,1545,N76517,LAX,IAH,0115,0112,-
 3,11,0123,183,175,156,1379,0559,8,0618,0607,-11,0,0,,,,,
 2015,1,1,4,AS,130,N457AS,FAI,SEA,0115,0107,-
 8,25,0132,213,218,186,1533,0538,7,0548,0545,-3,0,0,,,,,
 2015,1,1,4,NK,597,N528NK,MSP,FLL,0115,0127,12,14,0141,207,220,166,1487,0
 527,40,0542,0607,25,0,0,,25,0,0,0,0
 2015,1,1,4,US,413,N571UW,LAS,CLT,0120,0110,-
 10,12,0122,245,224,205,1916,0747,7,0825,0754,-31,0,0,,,,,
 2015,1,1,4,AA,2392,N3HRAA,DEN,MIA,0120,0141,21,12,0153,227,208,188,170
 9,0701,8,0707,0709,2,0,0,,,,,
 2015,1,1,4,NK,168,N629NK,PHX,ORD,0125,0237,72,9,0246,204,175,156,1440,0
 622,10,0549,0632,43,0,0,,43,0,0,0,0
 2015,1,1,4,AA,2211,N3CGAA,PHX,MIA,0127,0116,-
 11,10,0126,239,234,217,1972,0703,7,0726,0710,-16,0,0,,,,,
 2015,1,1,4,AS,136,N431AS,ANC,SEA,0135,,,,,205,,,1448,,,0600,,,0,1,A,,,,,
 2015,1,1,4,DL,95,N320US,SLC,ATL,0140,0134,-
 6,43,0217,215,231,182,1590,0719,6,0715,0725,10,0,0,,,,,

2015,1,1,4,NK,298,N514NK,LAS,IAH,0144,0140,-
 4,10,0150,170,170,148,1222,0618,12,0634,0630,-4,0,0,,,,,
 2015,1,1,4,HA,17,N389HA,LAS,HNL,0145,0145,0,16,0201,370,385,361,2762,06
 02,8,0555,0610,15,0,0,,0,0,15,0,0
 2015,1,1,4,US,617,N804AW,ANC,PHX,0152,0143,-
 9,21,0204,323,322,298,2552,0902,3,0915,0905,-10,0,0,,,,,
 2015,1,1,4,UA,1528,N76519,SJU,EWR,0154,0157,3,12,0209,255,241,220,1608,0
 449,9,0509,0458,-11,0,0,,,,,
 2015,1,1,4,AS,134,N464AS,ANC,SEA,0155,0140,-
 15,17,0157,218,198,170,1448,0547,11,0633,0558,-35,0,0,,,,,
 2015,1,1,4,B6,304,N607JB,SJU,JFK,0155,0153,-
 2,12,0205,235,248,231,1598,0456,5,0450,0501,11,0,0,,,,,
 2015,1,1,4,NK,451,N633NK,PBG,FLL,0155,0139,-
 16,10,0149,208,191,174,1334,0443,7,0523,0450,-33,0,0,,,,,
 2015,1,1,4,NK,972,N606NK,PHX,DFW,0159,0158,-
 1,11,0209,123,125,103,868,0452,11,0502,0503,1,0,0,,,,,
 2015,1,1,4,AA,2459,N3BDAA,PHX,DFW,0200,,,,,120,,,868,,,0500,,,0,1,B,,,,,
 2015,1,1,4,AS,144,N514AS,ANC,PDX,0200,0150,-
 10,24,0214,210,209,182,1542,0616,3,0630,0619,-11,0,0,,,,,
 2015,1,1,4,NK,647,N630NK,IAG,FLL,0200,0155,-
 5,21,0216,184,178,149,1176,0445,8,0504,0453,-11,0,0,,,,,
 2015,1,1,4,B6,1990,N597JB,SJU,EWR,0206,0152,-
 14,19,0211,246,264,239,1608,0510,6,0512,0516,4,0,0,,,,,
 2015,1,1,4,US,699,N172US,PHX,CLT,0220,0209,-
 11,10,0219,224,199,183,1773,0722,6,0804,0728,-36,0,0,,,,,

2015,1,1,4,AS,114,N303AS,ANC,SEA,0220,0209,-
11,15,0224,200,199,176,1448,0620,8,0640,0628,-12,0,0,,,,,

2015,1,1,4,B6,668,N653JB,PSE,MCO,0255,0248,-
7,10,0258,185,183,169,1179,0447,4,0500,0451,-9,0,0,,,,,

2015,1,1,4,UA,1162,N37293,BQN,EWR,0259,0258,-
1,14,0312,240,247,225,1585,0557,8,0559,0605,6,0,0,,,,,

2015,1,1,4,B6,1030,N239JB,BQN,MCO,0307,0304,-
3,25,0329,173,196,160,1129,0509,11,0500,0520,20,0,0,20,0,0,0,0

2015,1,1,4,B6,262,N627JB,SJU,BOS,0330,0316,-
14,14,0330,245,243,224,1674,0614,5,0635,0619,-16,0,0,,,,,

2015,1,1,4,B6,2134,N307JB,SJU,MCO,0400,0535,95,9,0544,185,175,163,1189,07
27,3,0605,0730,85,0,0,0,0,85,0,0

2015,1,1,4,B6,730,N621JB,BQN,MCO,0419,0423,4,11,0434,174,163,148,1129,06
02,4,0613,0606,-7,0,0,,,,,

2015,1,1,4,B6,768,N317JB,PSE,MCO,0424,0413,-
11,24,0437,186,196,166,1179,0623,6,0630,0629,-1,0,0,,,,,

2015,1,1,4,B6,2276,N646JB,SJU,BDL,0438,0550,72,15,0605,241,258,237,1666,0
902,6,0739,0908,89,0,0,17,0,72,0,0

2015,1,1,4,US,602,N197UW,ORD,PHX,0500,0459,-
1,12,0511,228,232,214,1440,0745,6,0748,0751,3,0,0,,,,,

2015,1,1,4,AS,695,N607AS,GEG,SEA,0500,0456,-
4,22,0518,70,74,39,224,0557,13,0610,0610,0,0,0,,,,,

2015,1,1,4,HA,102,N492HA,HNL,ITO,0500,0452,-
8,10,0502,49,51,35,216,0537,6,0549,0543,-6,0,0,,,,,

2015,1,1,4,OO,5467,N701BR,ONT,SFO,0500,0513,13,19,0532,89,85,61,363,0633
,5,0629,0638,9,0,0,,,,,

2015,1,1,4,HA,108,N476HA,HNL,KOA,0502,0501,-
 1,12,0513,42,48,30,163,0543,6,0544,0549,5,0,0,,,,,
 2015,1,1,4,AS,730,N423AS,ANC,SEA,0505,0457,-
 8,16,0513,205,199,179,1448,0912,4,0930,0916,-14,0,0,,,,,
 2015,1,1,4,HA,206,N484HA,HNL,OGG,0505,0454,-
 11,10,0504,36,36,20,100,0524,6,0541,0530,-11,0,0,,,,,
 2015,1,1,4,UA,1500,N30401,ORD,IAH,0510,0514,4,23,0537,175,182,147,925,08
 04,12,0805,0816,11,0,0,,,,,
 2015,1,1,4,AA,1323,N3CXAA,MCO,MIA,0510,0507,-
 3,14,0521,63,55,38,192,0559,3,0613,0602,-11,0,0,,,,,
 2015,1,1,4,NK,103,N616NK,BOS,MYR,0510,0506,-
 4,16,0522,140,134,112,738,0714,6,0730,0720,-10,0,0,,,,,
 2015,1,1,4,OO,7404,N8982A,HIB,MSP,0510,0500,-
 10,37,0537,68,78,35,174,0612,6,0618,0618,0,0,0,,,,,
 2015,1,1,4,OO,7419,N445SW,ABR,MSP,0510,0508,-
 2,21,0529,69,67,41,257,0610,5,0619,0615,-4,0,0,,,,,
 2015,1,1,4,OO,5254,N746SK,MAF,IAH,0510,,,,,87,,,429,,,0637,,,0,1,B,,,,,
 2015,1,1,4,US,480,N676AW,SEA,PHX,0515,0510,-
 5,15,0525,165,168,146,1107,0851,7,0900,0858,-2,0,0,,,,,
 2015,1,1,4,AA,1057,N3ASAA,DFW,MIA,0515,0703,108,15,0718,161,155,133,11
 21,1031,7,0856,1038,102,0,0,,0,0,0,0,102
 2015,1,1,4,AA,2454,N3ETAA,BOS,MIA,0515,0510,-
 5,14,0524,208,215,182,1258,0826,19,0843,0845,2,0,0,,,,,
 2015,1,1,4,EV,4354,N12142,MKE,IAH,0515,0517,2,12,0529,178,189,163,984,08
 12,14,0813,0826,13,0,0,,,,,

2015,1,1,4,US,425,N174US,PDX,PHX,0520,0620,60,13,0633,150,150,132,1009,0
945,5,0850,0950,60,0,0,,0,0,60,0,0

2015,1,1,4,AA,89,N3KVAA,IAH,MIA,0520,0618,58,19,0637,141,137,111,964,09
28,7,0841,0935,54,0,0,,0,0,54,0,0

2015,1,1,4,DL,2099,N355NW,BNA,DTW,0520,0511,-
9,30,0541,103,106,66,456,0747,10,0803,0757,-6,0,0,,,,,

2015,1,1,4,EV,4685,N26545,BRO,IAH,0520,0517,-
3,15,0532,74,72,49,308,0621,8,0634,0629,-5,0,0,,,,,

2015,1,1,4,EV,5583,N882AS,VPS,ATL,0520,0514,-
6,9,0523,66,57,42,250,0705,6,0726,0711,-15,0,0,,,,,

2015,1,1,4,OO,5484,N824AS,BOI,SFO,0520,0525,5,24,0549,112,110,79,522,060
8,7,0612,0615,3,0,0,,,,,

2015,1,1,4,OO,7370,N432SW,BJI,MSP,0520,0517,-
3,31,0548,60,72,34,199,0622,7,0620,0629,9,0,0,,,,,

2015,1,1,4,HA,103,N477HA,HNL,LIH,0523,0513,-
10,7,0520,37,35,21,102,0541,7,0600,0548,-12,0,0,,,,,

2015,1,1,4,EV,5187,N720EV,BNA,ATL,0525,0525,0,32,0557,65,70,35,214,0732,
3,0730,0735,5,0,0,,,,,

2015,1,1,4,MQ,2859,N660MQ,SGF,DFW,0525,,,,,95,,,364,,,0700,,,0,1,B,,,,,

2015,1,1,4,UA,1201,N79521,PHL,IAH,0530,0532,2,12,0544,230,234,210,1325,08
14,12,0820,0826,6,0,0,,,,,

2015,1,1,4,UA,1221,N37281,PDX,DEN,0530,0534,4,9,0543,142,142,125,991,084
8,8,0852,0856,4,0,0,,,,,

2015,1,1,4,UA,1607,N36476,SEA,DEN,0530,0529,-
1,10,0539,157,157,138,1024,0857,9,0907,0906,-1,0,0,,,,,

2015,1,1,4,EV,5241,N849AS,PIA,DTW,0535,0530,-
5,18,0548,88,75,54,346,0742,3,0803,0745,-18,0,0,,,,,

2015,1,1,4,EV,5498,N132EV,ORD,DTW,0535,0550,15,18,0608,85,67,41,235,074
9,8,0800,0757,-3,0,0,,,,,

2015,1,1,4,OO,6512,N925SW,FAT,LAX,0535,0530,-
5,33,0603,75,80,42,209,0645,5,0650,0650,0,0,0,,,,,

2015,1,1,4,UA,1577,N69813,SMF,DEN,0538,0634,56,26,0700,142,164,129,909,1
009,9,0900,1018,78,0,0,,22,0,56,0,0

2015,1,1,4,OO,4986,N719SK,AUS,LAX,0538,0535,-
3,16,0551,212,199,173,1242,0644,10,0710,0654,-16,0,0,,,,,

2015,1,1,4,B6,727,N292JB,BOS,BWI,0540,0536,-
4,14,0550,101,84,66,369,0656,4,0721,0700,-21,0,0,,,,,

2015,1,1,4,B6,1012,N375JB,ORD,BOS,0540,0529,-
11,16,0545,136,114,95,867,0820,3,0856,0823,-33,0,0,,,,,

2015,1,1,4,DL,824,N3736C,PHX,DTW,0540,0808,148,19,0827,223,203,178,1671
,1325,6,1123,1331,128,0,0,,0,0,0,0,128

2015,1,1,4,DL,941,N910DN,MCI,ATL,0540,0546,6,11,0557,122,101,84,692,0821
,6,0842,0827,-15,0,0,,,,,

2015,1,1,4,DL,2079,N389DA,BOS,DTW,0540,0537,-
3,15,0552,143,125,105,632,0737,5,0803,0742,-21,0,0,,,,,

2015,1,1,4,EV,3268,N14998,ATL,IAD,0540,0556,16,9,0605,105,103,79,534,0724
,15,0725,0739,14,0,0,,,,,

2015,1,1,4,EV,4160,N11150,JAX,EWR,0540,0531,-
9,9,0540,137,132,109,820,0729,14,0757,0743,-14,0,0,,,,,

2015,1,1,4,OO,6285,N962SW,MFR,DEN,0540,0538,-
2,42,0620,161,188,138,964,0938,8,0921,0946,25,0,0,,25,0,0,0,0

2015,1,1,4,OO,5184,N776SK,SLC,IAH,0540,0537,-
 3,18,0555,193,180,158,1195,0933,4,0953,0937,-16,0,0,,,,,
 2015,1,1,4,OO,6358,N926SW,IDA,DEN,0541,0539,-
 2,30,0609,104,113,72,458,0721,11,0725,0732,7,0,0,,,,,
 2015,1,1,4,F9,161,N905FR,MSN,DEN,0544,0534,-
 10,7,0541,145,144,128,826,0649,9,0709,0658,-11,0,0,,,,,
 2015,1,1,4,OO,4833,N554CA,MKE,DTW,0545,0543,-
 2,10,0553,74,55,36,237,0729,9,0759,0738,-21,0,0,,,,,
 2015,1,1,4,UA,1119,N69810,GEG,DEN,0545,0542,-
 3,21,0603,139,136,107,836,0850,8,0904,0858,-6,0,0,,,,,
 2015,1,1,4,UA,1237,N65832,SEA,IAH,0545,0541,-
 4,13,0554,251,266,238,1874,1152,15,1156,1207,11,0,0,,,,,
 2015,1,1,4,UA,1540,N78509,DCA,IAH,0545,0550,5,10,0600,209,197,182,1208,0
 802,5,0814,0807,-7,0,0,,,,,
 2015,1,1,4,AA,1433,N4YNAA,SAT,DFW,0545,0539,-
 6,12,0551,65,59,41,247,0632,6,0650,0638,-12,0,0,,,,,
 2015,1,1,4,AA,2299,N3LLAA,JFK,MIA,0545,0640,55,17,0657,185,199,152,1089
 ,0929,30,0850,0959,69,0,0,,14,0,55,0,0
 2015,1,1,4,B6,721,N623JB,BOS,PBI,0545,0551,6,16,0607,205,190,170,1197,085
 7,4,0910,0901,-9,0,0,,,,,
 2015,1,1,4,B6,917,N606JB,BOS,JFK,0545,0543,-
 2,12,0555,77,66,44,187,0639,10,0702,0649,-13,0,0,,,,,
 2015,1,1,4,B6,939,N794JB,JFK,BQN,0545,0545,0,17,0602,221,202,180,1576,100
 2,5,1026,1007,-19,0,0,,,,,
 2015,1,1,4,DL,1890,N377DA,DFW,ATL,0545,0603,18,13,0616,124,104,86,731,0
 842,5,0849,0847,-2,0,0,,,,,