

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ  
федеральное государственное бюджетное образовательное учреждение высшего образования  
«Тольяттинский государственный университет»

Кафедра \_\_\_\_\_ «Прикладная математика и информатика» \_\_\_\_\_  
(наименование)

01.03.02 Прикладная математика и информатика  
(код и наименование направления подготовки / специальности)

Компьютерные технологии и математическое моделирование  
(направленность (профиль) / специализация)

## ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА (БАКАЛАВРСКАЯ РАБОТА)

на тему «Разработка алгоритма оптимизации на основе машинного обучения для задачи планирования ресурсов»

Обучающийся

С. С. Лавин

(Инициалы Фамилия)

(личная подпись)

Руководитель

канд.пед.наук, доцент, Т.А. Агошкова

(ученая степень (при наличии), ученое звание (при наличии), Инициалы Фамилия)

Консультант

канд.пед.наук, доцент, А. В. Егорова

(ученая степень (при наличии), ученое звание (при наличии), Инициалы Фамилия)

Тольятти 2024

## Аннотация

Тема бакалаврской работы: «Разработка алгоритма оптимизации на основе машинного обучения для задачи планирования ресурсов».

Бакалаврская работа посвящена процессам разработки алгоритма, который работает на основе машинного обучения и решает задачу планирования ресурсов.

В ходе выполнения исследований по бакалаврской работе были описаны ресурсы, подходящие для планирования и понятие динамики выручки, а также какие существуют алгоритмы оптимизации на основе машинного обучения для задачи планирования ресурсов и описана математическая модель метода случайного леса.

Во введении прописывается актуальность темы, написаны цель и задачи.

В первом разделе рассматривается предметная область исследования и проводится анализ ресурсов для планирования и разбор понятия динамики выручки.

Во втором разделе произведен обзор существующих алгоритмов оптимизации на основе машинного обучения для задачи планирования ресурсов и описана математическая модель метода случайного леса, также мы описали компоненты системы и спроектировали систему для разработки ПО.

В третьем разделе продемонстрирована реализация ПО и проведено его функциональное тестирование, а также произведен анализ полученных результатов.

В заключении представлены результаты выполнения бакалаврской работы.

Бакалаврская работа состоит из введения, трёх разделов, заключения и списка использованной литературы.

Бакалаврская работа состоит из 45 страниц, 14 рисунков, 3 таблиц, 36 источников, 10 формул и 1 листинга.

## **Abstract**

Topic of the bachelor's thesis: “Development of an optimization algorithm based on machine learning for the problem of resource planning.”

The bachelor's thesis is devoted to the development of an algorithm that works on the basis of machine learning and solves the problem of resource scheduling.

During the bachelor's work, resources suitable for planning and the concept of revenue dynamics were described, as well as what optimization algorithms exist based on machine learning for the problem of resource planning and a mathematical model of the random forest method was described.

The introduction states the relevance of the topic, the purpose and objectives are written.

The first section examines the subject area of the study and analyzes resources for planning and analyzes the concept of revenue dynamics.

In the second section, we review existing optimization algorithms based on machine learning for the resource scheduling problem and describe the mathematical model of the random forest method, we also described the components of the system and designed a system for software development.

The third section demonstrates the implementation of the software and its functional testing, as well as an analysis of the results obtained.

In conclusion, the results of the bachelor's work are presented.

The bachelor's thesis consists of an introduction, three sections, a conclusion and a list of references.

Bachelor's work consists of 45 pages, 14 figures, 3 tables, 36 sources, 10 formulas and 1 listing.

## Содержание

Введение.....	5
1 Описание задачи планирования ресурсов .....	8
1.1 Постановка задачи и анализ предметной области .....	8
1.2 Обзор и выбор ресурсов для прогнозирования.....	9
1.3 Анализ финансовых ресурсов для планирования.....	11
1.4 Практическое применение планирования динамики выручки в рамках поставленной задачи .....	13
2 Описание алгоритмов оптимизации на основе машинного обучения.....	15
2.1 Обзор и выбор алгоритма оптимизации на основе машинного обучения .....	15
2.2 Описание математической модели случайного леса.....	17
2.3 Описание архитектуры и проектирование системы для дальнейшего написания программного кода.....	21
3 Реализация и тестирование программного обеспечения. ....	27
3.1 Реализация программного обеспечения .....	27
3.2 Тестирование программного обеспечения .....	31
Заключение .....	36
Список используемой литературы и используемых источников.....	38
Приложение А Листинг 1 (код программного обеспечения по расчёту динамики выручки) .....	42

## Введение

С постоянным ростом сложности и объемов производства, а также увеличением конкуренции, эффективное управление ресурсами становится критически важным для бизнеса. Многие компании стремятся автоматизировать и оптимизировать процессы планирования, чтобы улучшить производительность, снизить издержки и повысить качество продукции или услуг [1].

В настоящее время исследования в этой области активно ведутся. Многие исследователи и компании работают над применением методов машинного обучения, таких как нейронные сети, генетические алгоритмы, и алгоритмы обучения с подкреплением, для решения задач планирования ресурсов. Они стремятся создать алгоритмы, способные адаптироваться к изменяющимся условиям производства, учитывать различные ограничения и оптимизировать распределение ресурсов с учетом поставленных целей [2].

Одним из основных направлений исследований является разработка гибридных алгоритмов, объединяющих в себе преимущества различных методов оптимизации. Также активно изучается применение технологий обработки больших данных и анализа данных для улучшения качества прогнозирования и принятия решений в планировании ресурсов [3].

В целом, текущее состояние исследований в данной области свидетельствует о значительном интересе к применению машинного обучения для решения задач оптимизации планирования ресурсов. Ожидается, что развитие новых алгоритмов и подходов в этой области приведет к созданию более эффективных и адаптивных систем управления ресурсами, способствующих повышению конкурентоспособности компаний [4].

Однако, несмотря на достигнутые результаты, можно сделать вывод, что данная тема еще не полностью раскрыта и нуждается в дальнейшей разработке. Исследователи в области разработки алгоритмов оптимизации на

основе машинного обучения для задач планирования ресурсов сталкиваются с рядом сложностей и проблем.

Во-первых, эффективность алгоритмов машинного обучения сильно зависит от качества и доступности данных. В задачах планирования ресурсов может быть нехватка данных или данные могут быть неточными, что затрудняет обучение моделей [5].

Во-вторых, модели, основанные на машинном обучении, могут быть достаточно сложными и трудными для интерпретации. Это может создавать проблемы при попытке объяснить принимаемые решения или при внедрении разработанных алгоритмов в реальные системы [6].

Стоит также отметить, что некоторые методы машинного обучения требуют больших вычислительных ресурсов, особенно при работе с большими объемами данных. Это может быть ограничивающим фактором для применения этих методов в реальных производственных средах.

Таким образом, следует отметить, что разработка алгоритма оптимизации на основе машинного обучения для задачи планирования ресурсов является активной и перспективной областью исследований, но требует дальнейшего развития и углубленного изучения для достижения более полного понимания и эффективного применения в практике.

Объектом исследования бакалаврской работы является разработка алгоритма оптимизации на основе машинного обучения для задачи планирования ресурсов.

Предметом исследования – математическая модель, алгоритмы, библиотеки, встроенные библиотеки и графический интерфейс для задачи планирования ресурсов.

Цель работы - разработка алгоритма оптимизации на основе математической модели машинного обучения для задачи планирования ресурсов.

Задачи бакалаврской работы включают в себя:

- Анализ и обзор существующих алгоритмов оптимизации для задачи планирования ресурсов.
- Изучение практического применения алгоритма оптимизации для задачи планирования ресурсов в рамках поставленной задачи.
- Обзор и выбор метода машинного обучения для задачи планирования ресурсов.
- Описание математической модели выбранного метода машинного обучения.
- Описание архитектуры и системы для разработки ПО.
- Реализация и описание алгоритма программного обеспечения.
- Практическое тестирование разработанного алгоритма и анализ полученных результатов.

В первом разделе рассматривается постановка задачи на работу и предметная область, а также какие ресурсы подходят для планирования. Далее происходит выбор финансового типа ресурса и среди этого типа выбирается для оценки динамика выручки.

Во втором разделе, описана математическая модель метода случайного леса. Также была спроектирована система для разработки и компоненты ПО.

В третьем разделе, описан процесс реализации и тестирования программного обеспечения, а также анализ полученных результатов.

# **1 Описание задачи планирования ресурсов**

## **1.1 Постановка задачи и анализ предметной области**

Основной задачей данной бакалаврской работы является разработка алгоритма оптимизации с использованием машинного обучения в рамках задачи планирования ресурсов.

Цель бакалаврской работы – оптимизировать рабочий процесс в рамках планирования ресурсов в кампании, в которой я прохожу преддипломную практику, с помощью алгоритма машинного обучения.

Алгоритм оптимизации - это процедура или набор правил, которые направлены на нахождение наилучшего решения в рамках определенной задачи. Целью алгоритмов оптимизации является минимизация или максимизация некоторой целевой функции, учитывая ограничения, которые могут быть наложены на решение [7].

Алгоритмы оптимизации могут быть применены к широкому спектру задач в различных областях, включая математику, инженерию, экономику, бизнес и другие. Они могут использоваться для решения задач планирования, распределения ресурсов, управления, машинного обучения и т. Д [8].

Типы алгоритмов оптимизации могут варьироваться в зависимости от характеристик задачи и доступных ресурсов. Некоторые из распространенных методов оптимизации включают в себя:

- Методы локальной оптимизации.
- Методы глобальной оптимизации.
- Методы комбинаторной оптимизации.
- Методы мультикритериальной оптимизации.
- Методы оптимизации на основе правил.

Алгоритм оптимизации в рамках задачи планирования ресурсов представляет собой процедуру или набор правил, направленных на нахождение оптимального решения для распределения ресурсов с целью



максимизации производительности, минимизации издержек или достижения других целей, определенных для конкретной задачи планирования [9].

## 1.2 Обзор и выбор ресурсов для прогнозирования

Правильное планирование ресурсов позволяет оптимизировать их использование, минимизировать риски и повысить общую эффективность организации. [10]

Для ИТ-компании, специализирующейся на решениях для расчета и учета платы за жилищно-коммунальные услуги, оптимизация планирования ресурсов играет решающую роль в обеспечении бесперебойного функционирования сервисов. [11]

Поэтому при начале работы необходимо ознакомиться с ресурсами, которые можно применить в задаче планирования ресурсов и предсказать примерные их значения. Для этого рассмотрим сравнительный анализ ресурсов, к которым можно применить задачу планирования ресурсов (таблица 1). [12]

Таблица 1 – Сравнительный анализ ресурсов, к которым можно применить задачу планирования ресурсов

Тип ресурса	Описание ресурса	Преимущества планирования ресурса	Недостатки планирования ресурса
Материальные	Сырье, материалы, оборудование и инфраструктура, необходимые для выполнения задач [13].	Планирование легче из-за возможности точного измерения и оценки; Правильное планирование ведет к улучшению производства и качества продукции; Планирование позволяет рационально использовать активы для увеличения стоимости компании.	Сложно предсказать точные сроки замены и обслуживания; Высокие начальные вложения могут требовать сложного долгосрочного планирования; Необходимо учитывать возможные проблемы с транспортировкой и хранением.

Продолжение таблицы 1

Трудовые	Персонал, их навыки, знания и опыт, необходимые для выполнения задач [14].	Возможность планирования адаптивных графиков работы и обучения. Планирование может способствовать созданию инновационной среды. Возможность разработки программ повышения квалификации и обучения.	Трудно предсказать производительность и возможные ошибки; Трудно планировать работу из-за ограниченности рабочих часов; Необходимо учитывать дополнительные усилия для поддержания мотивации сотрудников.
Финансовые	Деньги и кредитные средства, используемые для финансирования деятельности организации [15].	Финансовые ресурсы могут быть запланированы для приобретения любых других ресурсов; Легко планировать с учетом возможности быстрого перемещения средств; Планирование позволяет эффективно привлекать кредиты и инвестиции.	Трудно предсказать экономические изменения и их влияние на ресурсы; Планирование должно учитывать возврат заемных средств и процентные платежи; Необходимо учитывать инфляцию и другие экономические факторы, влияющие на стоимость ресурсов.
Информационные	Данные, знания, патенты, технологии и информационные системы, поддерживающие принятие решений и управление [16].	Планирование позволяет оптимизировать использование информации для улучшения процессов. Планирование информационных ресурсов может обеспечить уникальные преимущества на рынке. Планирование способствует быстрой обработке и передаче данных, что ускоряет принятие решений	Необходимость учета затрат на обновление и поддержание информационных систем; Планирование должно включать меры по защите от кибератак и утечек данных; Трудности в планировании из-за необходимости интеграции с существующими системами и процессами.

Стоит учитывать, что каждый тип ресурса имеет свои особенности, которые необходимо учитывать в процессе планирования. Материальные ресурсы требуют тщательного учета износа и затрат на логистику. Трудовые ресурсы требуют внимания к человеческому фактору и мотивации. Финансовые ресурсы зависят от внешней экономической ситуации и рисков

долговой нагрузки. Информационные ресурсы требуют постоянного обновления и защиты от угроз безопасности. Учитывая все эти аспекты, можно разработать сбалансированный план, обеспечивающий устойчивое развитие и успех организации. [17]

Из всех ресурсов, которые могут быть подвергнуты прогнозированию, финансовые ресурсы являются одними из ключевых показателей, влияющих на финансовую устойчивость и успех предприятия. Их довольно легко планировать и в условиях рыночной экономики финансы являются основополагающим ресурсом, от которого зависят все остальные ресурсы. Разберём подробнее какие финансовые ресурсы компании наиболее важные для планирования.

### 1.3 Анализ финансовых ресурсов для планирования

Для того, чтобы начать планирование финансовых ресурсов компании необходимо понять, какой тип финансовых ресурсов является самым важным в рамках деятельности компании. Для этого разберем несколько типов финансовых ресурсов с примерами и определением того, почему тот или иной ресурс важен. Анализ финансовых ресурсов компании приведен ниже (таблица 2).

Таблица 2 – Анализ финансовых ресурсов компании

Название ресурса	Описание ресурса	Примеры ресурса	Важность ресурса
Собственный капитал	Средства, вложенные владельцами компании, а также прибыль, не распределенная между акционерами и оставленная для реинвестирования.	Уставный капитал; Накопленная прибыль.	Высокая, так как собственный капитал определяет финансовую устойчивость компании и её независимость от внешних источников финансирования.

Продолжение таблицы 2

Кредиты и займы	Средства, привлеченные от банков и других финансовых учреждений, которые подлежат возврату с процентами.	Банковские кредиты; Облигационные займы.	Важны для оперативного финансирования крупных проектов и покрытия временных дефицитов денежных средств.
Инвестиции	Средства, полученные от внешних инвесторов, таких как венчурные капиталисты, частные инвесторы или фонды.	Венчурные, частные и стратегические инвестиции.	Критически важны для стартапов и быстрорастущих компаний, нуждающихся в значительных вливаниях капитала.
Денежные потоки от операционной деятельности	Денежные средства, генерируемые основной деятельностью компании, такие как выручка от продаж товаров и услуг.	Доходы от продаж; Арендные платежи; Комиссионные доходы.	Очень важны для поддержания текущей ликвидности и способности компании выполнять свои обязательства.
Доходы от финансовых операций	Средства, полученные от финансовых инвестиций и других операций, не связанных напрямую с основной деятельностью компании.	Проценты по депозитам; Дивиденды от акций; Доходы от продажи ценных бумаг.	Важны для диверсификации источников дохода и увеличения финансовой стабильности.
Государственные субсидии и гранты	Средства, предоставляемые государственными органами для поддержки определённых видов деятельности.	Гранты на исследования и разработки; Субсидии на экологические проекты.	Очень важны для предприятий, работающих в приоритетных для государства отраслях.

На основе анализа, приведенного в таблице 2, можно сделать вывод, что самыми важными ресурсом для планирования являются собственные средства компании, кредиты займы и инвестиции. Однако это не совсем так, потому что все эти ресурсы можно привлечь, если компания хорошо себя чувствует на рынке и у нее предполагается потенциальный рост выручки в ближайшее

время. Соответственно, самым важным ресурсом все-таки являются денежные потоки от основной операционной деятельности компании и чем они больше, тем большие средства можно вложить, занять и привлечь. Однако и это тоже является комплексным ресурсом, но самый важный параметр при рассмотрении денежных потоков от основной операционной деятельности компании является динамика выручки. Рассмотрим этот параметр более подробно.

#### **1.4 Практическое применение планирования динамики выручки в рамках поставленной задачи**

Динамика выручки представляет собой изменение выручки компании за определённый период времени. Этот показатель важен для анализа роста или сокращения доходов, что позволяет оценить эффективность бизнеса, стратегию продаж и маркетинга. [18]

Выручка компании состоит из нескольких ключевых элементов:

- Доходов, полученных от реализации продукции или предоставления услуг.
- Суммы, полученные от сдачи в аренду имущества.
- Проценты, дивиденды и другие доходы от финансовых вложений.
- Все остальные доходы, не относящиеся к основным видам деятельности.

Динамика выручки рассчитывается как процентное изменение выручки за два сравниваемых периода. Формула для расчёта динамики выручки представлена ниже (формула 1).

$$D_B = \frac{V_{ТП} - V_{ПП}}{V_{ПП}} * 100\% \quad (1)$$

где

$V_{ТП}$  – выручка за текущий период

$V_{ПП}$  – выручка за предыдущий период

С использованием данных о предыдущих продажах модели машинного обучения могут быть обучены для прогнозирования будущей выручки компании. В нашем случае будут использоваться отчёты о еженедельной чистой прибыли компании, в которой я прохожу преддипломную практику.

Чистая прибыль – это все доходы минус все расходы за выбранный период, что является самым важным показателем в компании, который наиболее привлекательный для инвесторов [19].

Прогнозирование динамики выручки позволяет компании:

- Анализировать текущее состояние финансов и прогнозировать будущие результаты. Это в свою очередь может помочь выявлять тренды и понимать, насколько эффективно используются ресурсы компании.
- Принимать обоснованные решения о распределении финансовых ресурсов, выборе инвестиционных проектов и разработке стратегий развития.
- Помогает лучше понять спрос на их продукцию или услуги и адаптировать маркетинговые стратегии для увеличения продаж. Это может включать изменение ценовой политики, разработку новых продуктов или услуг, а также оптимизацию каналов продаж.

Выводы по разделу 1:

В первом разделе была поставлена задача на исследование, определена цель ВКР и проанализирована предметная область.

Также были проанализированы виды ресурсов для планирования и выбраны финансовые ресурсы, среди которых выявлен наиболее важный показатель – динамика выручка. Этот параметр был рассмотрен и изучен в заключительном пункте первого раздела.

## **2 Описание алгоритмов оптимизации на основе машинного обучения**

### **2.1 Обзор и выбор алгоритма оптимизации на основе машинного обучения**

Алгоритмы оптимизации на основе машинного обучения — это методы и техники, которые применяют модели и принципы машинного обучения для нахождения оптимальных решений в задачах, требующих эффективного распределения и использования ресурсов. Эти алгоритмы позволяют улучшать производительность моделей, оптимизировать параметры и минимизировать ошибки или потери [20].

Применение алгоритмов оптимизации на основе машинного обучения для планирования ресурсов имеет ряд преимуществ:

- Позволяет автоматизировать процесс планирования, что особенно полезно в больших и сложных системах, где вручную обрабатывать данные и принимать решения становится трудно.
- Методы оптимизации, такие как градиентный спуск и байесовская оптимизация, могут быть использованы для нахождения оптимальных параметров моделей и решений, минимизируя затраты и максимизируя эффективность использования ресурсов.
- Алгоритмы машинного обучения, такие как случайный лес, могут использоваться для прогнозирования будущих потребностей в ресурсах на основе исторических данных.
- Алгоритмы машинного обучения могут адаптироваться к изменениям в данных и условиях (изменения в спросе на продукцию, колебания в доступности ресурсов и другие динамичные факторы).

Для планирования еженедельной динамики выручки рассмотрим различные методы прогнозирования и выявим наиболее подходящий алгоритм для этой задачи (таблица 3).

Таблица 3 – Сравнительный анализ алгоритмов оптимизации на основе машинного обучения

Алгоритм оптимизации	Описание алгоритма	Недостатки алгоритма	Преимущества алгоритма
Случайный лес	Метод ансамблевого обучения, создающий множество деревьев решений и усредняющий их результаты для повышения точности [21].	Высокая вычислительная стоимость; Сложность интерпретации модели; Большие требования к памяти.	Высокая точность; Устойчивость к переобучению; Способность работать с большими данными и большим количеством признаков.
Градиентный спуск	Итеративный метод оптимизации, минимизирующий функцию потерь путём корректировки параметров модели в направлении градиента [22].	Локальные минимумы; Чувствительность к шагу обучения.	Простота; Эффективность для больших данных.
Рандомизированный поиск	Метод гиперпараметрической оптимизации, случайным образом выбирающий комбинации гиперпараметров [23].	Не гарантирует глобальный оптимум; Требует большого числа итераций.	Простота; Возможность поиска в широком диапазоне параметров.
Байесовская оптимизация	Использует вероятностные модели для поиска глобального оптимума функции, строя апостериорное распределение [24].	Трудность в настройке и реализации; Ограниченная применимость для высоко размерных задач.	Эффективность для сложных и дорогостоящих функций; Быстрая сходимость при ограниченном числе итераций.
Поиск на основе сетки	Метод систематического перебора всех возможных комбинаций гиперпараметров из заданного набора значений [25].	Высокая вычислительная стоимость; Неэффективность для большого числа гиперпараметров.	Полный поиск всех комбинаций; Гарантирует нахождение лучшего решения среди заданных параметров.
Эволюционные алгоритмы	Метаэвристические методы, вдохновлённые естественным отбором, используют популяцию кандидатов и операторы эволюции [26].	Высокая вычислительная сложность; Требуется тщательная настройка параметров.	Глобальные оптимумы; Хорошая производительность в сложных пространствах поиска.



На основе статистического анализа в таблице 3 можно сделать вывод, что модель случайного леса является наиболее подходящей для решения поставленной задачи в виду того, что алгоритм имеет высокую точность, устойчивость к переобучению и способность работать с большими данными и большим количеством признаков.

## 2.2 Описание математической модели случайного леса

Случайный лес — это ансамблевый метод машинного обучения, который состоит из множества деревьев решений. Основная идея заключается в построении большого числа деревьев решений на разных выборках данных и усреднении их результатов для получения финального предсказания. Ниже представлено более детальное описание математической модели случайного леса [27].

Модель случайного леса включает в себя следующие шаги решения и компоненты:

- Деревья решений - базовые модели, используемые в случайном лесу.
- Бутстраппинг - метод выборки данных с возвращением для создания различных обучающих наборов данных.
- При каждом разделении узла дерева рассматривается случайное подмножество признаков для выбора наилучшего разделения.

Каждое дерево решений строится с использованием алгоритма CART (Classification and Regression Trees). Разберем этот алгоритм более подробно. Он включает в себя следующие шаги:

Для каждого узла выбирается случайное подмножество признаков  $F_x$ , где  $k$  – количество признаков, которое меньше общего числа признаков  $d$  [28].

Для каждого признака  $x \in F_x$  и каждого порога  $u$  определяется критерий разделения по формуле (формула 2).

$$G(f, \theta) = |S_{left}| * Imp(S_{left}) + |S_{right}| * Imp(S_{right}) \quad (2)$$

где

$S_{left}$  и  $S_{right}$  - подмножества данных, полученные при разделении по  $x$  и по  $y$ .

$Imp()$  - мера неоднородности узла в дереве решений.

Чем более неоднороден узел, тем выше его мера неоднородности. В задачах классификации обычно используется индекс Джини или энтропия, а в задачах регрессии — среднеквадратичная ошибка (MSE) [29].

В нашем случае мера неоднородности узла в дереве решений будет среднеквадратичной ошибкой MSE.

Затем выбирается пара  $(x^*, y^*)$ , которая минимизирует критерий разделения  $G(x, y)$ .

Рекурсивное разделение повторяется для дочерних узлов до тех пор, пока не достигнут условия остановки:

- Максимальная глубина дерева.
- Минимальное количество образцов в узле.
- Минимальное уменьшение загрязненности при разделении.

Бутстраппинг является методом создания множественных подвыборок из исходного набора данных с целью построения ансамбля моделей. Этот метод особенно полезен для снижения вариативности и увеличения устойчивости модели, что в итоге улучшает её производительность. [30]

Из исходного набора данных  $D$ , содержащего  $n$  образцов, создаются  $B$  бутстрап-подвыборок путём случайной выборки с возвращением. Каждая бутстрап-подвыборка  $D^{(b)}$  имеет размер  $n$  и состоит из образцов, случайно выбранных с возвращением из  $D$ . Для каждой бутстрап-подвыборки  $D^{(b)}$  обучается отдельная модель  $h_b(x)$ . [31]

Предсказания всех моделей объединяются для получения финального предсказания ансамбля. Обозначим исходный набор данных в виде формулы 3, а создание бутстрап-подвыборок в виде формулы 4 (формула 3 и 4).

$$D = \{(x_i, y_i)\}_{i=1}^n, \quad (3)$$

$$D^{(b)} = \{(x_i^{(b)}, y_i^{(b)})\}_{i=1}^n, \quad (4)$$

где

$x_i$  – вектор признаков.

$y_i$  – целевая переменная.

$(x_i^{(b)}, y_i^{(b)})$  - случайно выбранные образцы с возвращением из  $D$ .

При каждом разделении узла дерева рассматривается случайное подмножество признаков  $F_k$ . Каждый классификатор в лесу выдает предсказание по следующей формуле (формула 5).

$$\widehat{y}_b = h_b(x), b = 1, \dots, B \quad (5)$$

где

$h_b(x)$  – предсказание  $b$ -ого дерева.

Дальнейшие расчёты финального предсказания и меры неоднородности узла отличаются для задач классификации и регрессии. Рассмотрим каждый вариант по отдельности [32].

Для задачи классификации финальное предсказание определяется голосованием большинства (формула 6).

$$\widehat{y} = mode(\{\widehat{y}_1, \widehat{y}_2, \dots, \widehat{y}_B\}) \quad (6)$$

Меры неоднородности узла в задаче классификации могут определяться по формуле Джини  $G(S)$  (формула 7) и по формуле энтропии  $E(S)$  (формула 8) [33].

$$G(S) = 1 - \sum_{i=1}^C p_i^2 \quad (7)$$

$$E(S) = - \sum_{i=1}^C p_i \log(p_i) \quad (8)$$

где

$p_i$  - доля объектов класса  $i$  в подмножестве  $S$ .

$C$  – количество классов.

В своей ВКР я буду использовать задачу регрессии и для неё применяются следующие формулы.

Финальное предсказание определяется как усреднённое значение предсказаний всех деревьев (формула 9) [34].

$$\hat{y} = \frac{1}{n} \sum_{i=1}^n \hat{y}_i \quad (9)$$

где:

$n$  – количество деревьев.

$\hat{y}_i$  – прогнозируемое значение  $i$ -ого дерева.

Меры неоднородности узла в задаче регрессии могут определяться по формуле среднеквадратичной ошибки MSE (формула 10).

$$MSE = \frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2 \quad (10)$$

где

$\hat{y}_i$  – прогнозируемые моделью значения.

$y_i$  – истинные значения чистой прибыли.

Случайный лес является мощным инструментом для задач классификации и регрессии, обеспечивающий высокую точность и устойчивость к переобучению [35]. Благодаря использованию бутстраппинга и случайного подмножества признаков, этот метод эффективно справляется с задачами оптимизации и планирования ресурсов, предоставляя надёжные и масштабируемые решения [36].

В рамках поставленной задачи на предприятии, в котором я прохожу преддипломную практику, будет использоваться задача регрессии для прогнозирования динамики выручки. Для оценки качества модели будет рассчитываться среднеквадратичная ошибка (MSE).

MSE является одной из ключевых метрик оценки качества регрессионных моделей. Она позволяет количественно оценить среднее отклонение прогнозных значений от фактических, тем самым предоставляя объективную оценку точности модели. Минимизация MSE является одной из основных целей при обучении регрессионных моделей.

Таким образом, применение алгоритма случайного леса, ориентированного на задачу регрессии с расчетом MSE, позволит разработать надежную и масштабируемую систему прогнозирования динамики выручки на предприятии. Это заложит основу для эффективного планирования ресурсов и оптимизации бизнес-процессов.

### **2.3 Описание архитектуры и проектирование системы для дальнейшего написания программного кода**

Для того, чтобы начать разрабатывать программное обеспечение необходимо разбить весь проект по модульно. Всего будет три компонента: компонент сбора и обработки данных, компонент обработки и компонент визуализации.

Компонент сбора и обработки данных получает на вход 4 файла отчёта о чистой прибыли отдельного офиса в формате `xlsx`. Отчёты включают в себя текущую неделю, предыдущую неделю, такую же неделю в прошлом году и следующую неделю в прошлом году.

Компонент обучения и прогнозирования анализирует полученные файлы отчёта о чистой прибыли офиса и на основе полученных результатов прогнозирует динамику выручки для следующей недели и результирующую динамику выручки за текущую неделю.

Компонент визуализации строит и выводит графики в интерфейсе, загружая их в формат html.

После описания всех компонентов работы следует приступить к проектированию системы для разработки программного обеспечения. Программный код будет использовать модель случайного леса для построения результирующей линии динамики выручки. Построение будет осуществляться на основе двух недель: текущей и такой же недели в прошлом году. Алгоритм также будет рассчитывать текущую еженедельную динамику выручки, ежегодную динамику выручки и предсказывать динамику выручки на следующую неделю. Сначала рассмотрим общую блок-схему системы разработки для реализации программного кода (рисунок 1), а затем более углубленно рассмотрим декомпозицию.

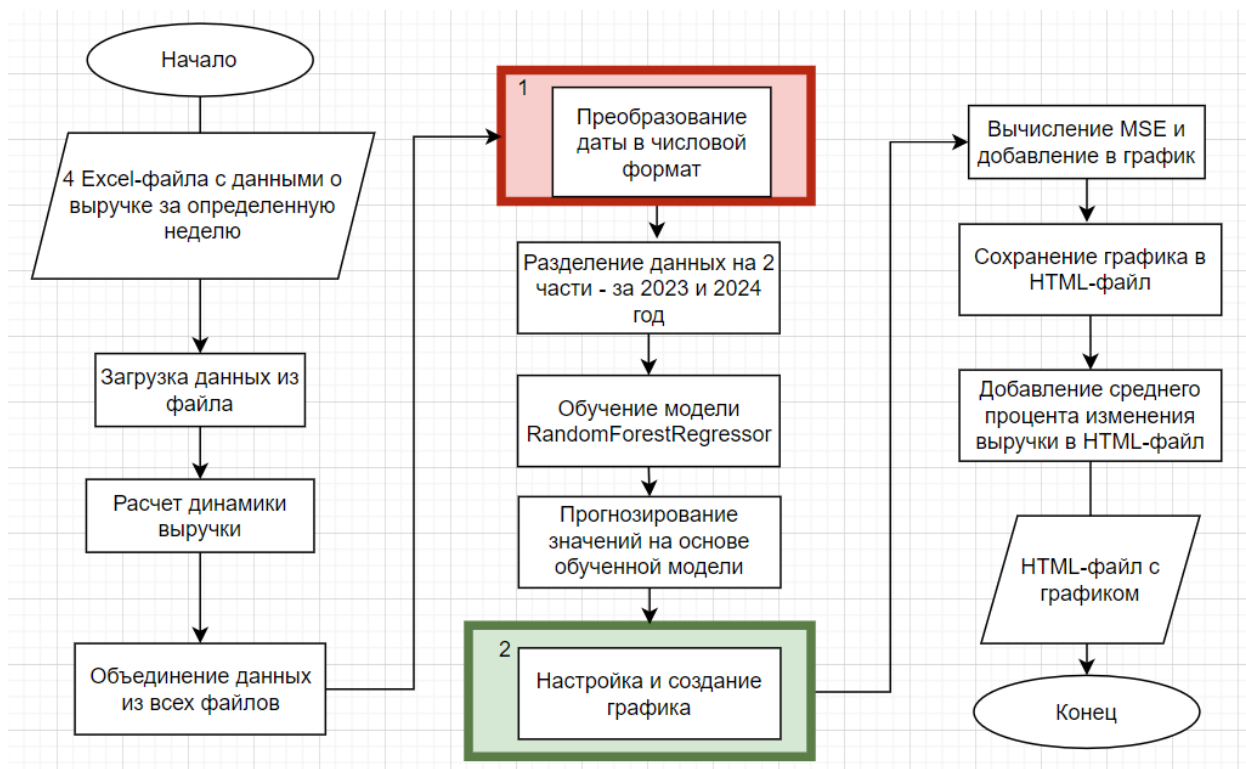


Рисунок 1 - Общая блок-схема системы для разработки ПО

Из диаграммы, представленной на рисунке 1, мы видим, что в начале программы будут загружены 4 файла Excel с данными о чистой прибыли за

текущую и предыдущую неделю, а также за такую же неделю, но год назад и за следующую неделю год назад. Из этих файлов загружаются данные в датасет. Затем происходит расчёт динамики выручки. После чего происходит объединение данных из всех файлов.

После происходит преобразование даты в числовой формат. На этом моменте хотелось бы остановиться поподробнее и рассмотреть декомпозицию второго уровня для первого блока, который обозначен красным цветом на рисунке 1 (рисунок 2).

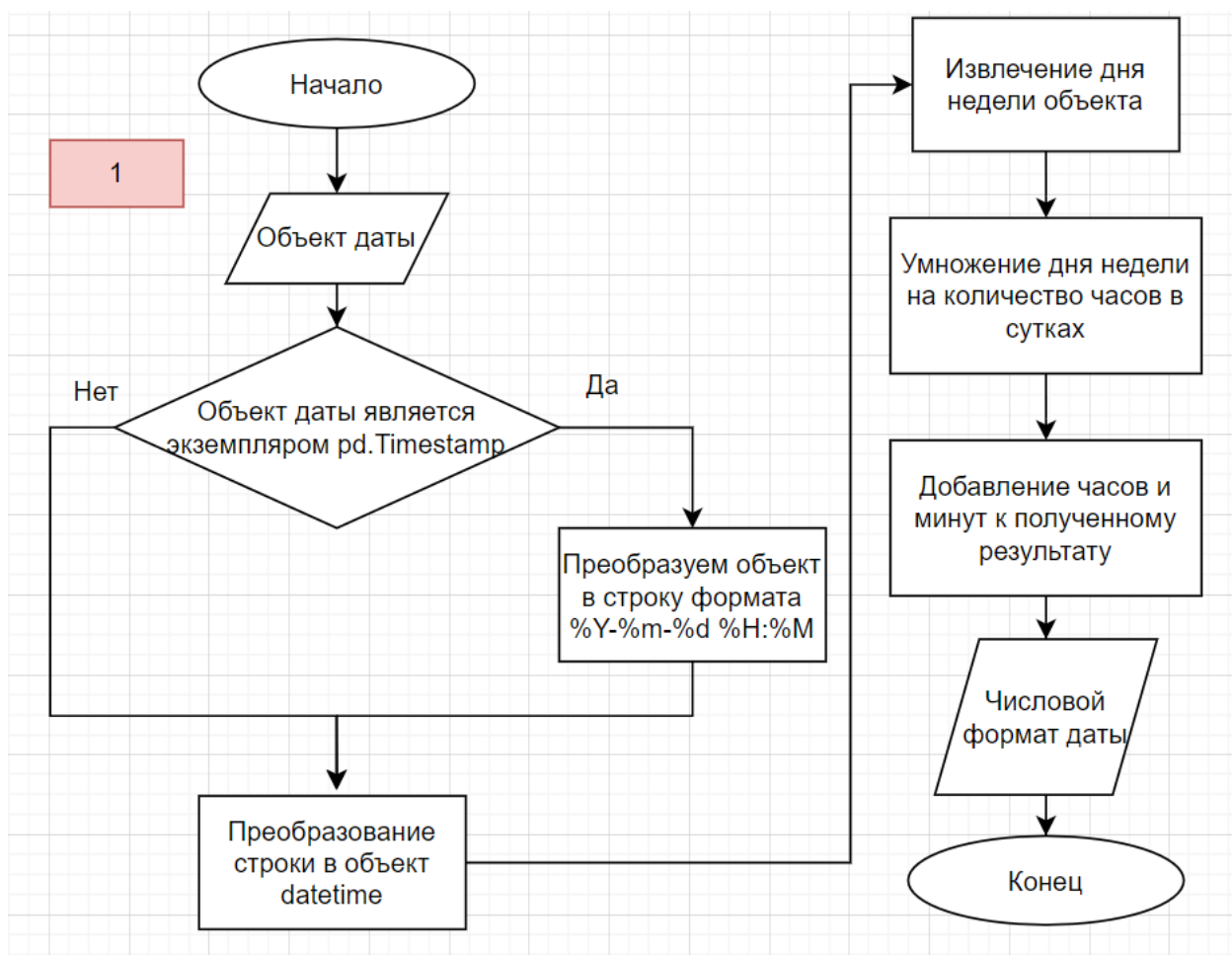


Рисунок 2 - Декомпозиция второго уровня для первого блока

Из блок-схемы на рисунке 2 видно, что при преобразовании даты в числовой формат происходит несколько важных этапов:

- Проверка исходного формата даты в отчете и при необходимости преобразование к единому подходящему формату.
- Конвертация строкового представления даты в объект `datetime`.
- Извлечение дня недели из даты и умножение его на количество часов в сутках.
- Добавление часов и минут к полученному значению для корректного отображения на оси X.
- Возврат преобразованного числового формата даты.

Далее, как показано на рисунке 1, происходит разделение данных на две временные группы - 2023 и 2024 год. Это необходимо для сравнения фактических данных и прогнозных значений.

После этого выполняется обучение модели случайного леса на основе данных 2023 года и прогнозирование значений на 2024 год.

Как описано в декомпозиции второго уровня (рисунок 3), на следующем этапе происходит настройка и создание графика:

- Формирование пустого графика.
- Добавление линии прогноза модели и маркеров для данных 2023 и 2024 годов.
- Установка заголовков осей и общего заголовка.
- Настройка легенды графика.
- Добавление аннотации с динамикой выручки и рамки для визуального выделения.
- Настройка отображения меток на оси X в соответствии с днями недели.

Завершающим этапом является вычисление среднеквадратичной ошибки (MSE), добавление ее значения на график, сохранение графика в HTML-формате и включение среднего процента измерения выручки в итоговый HTML-файл.



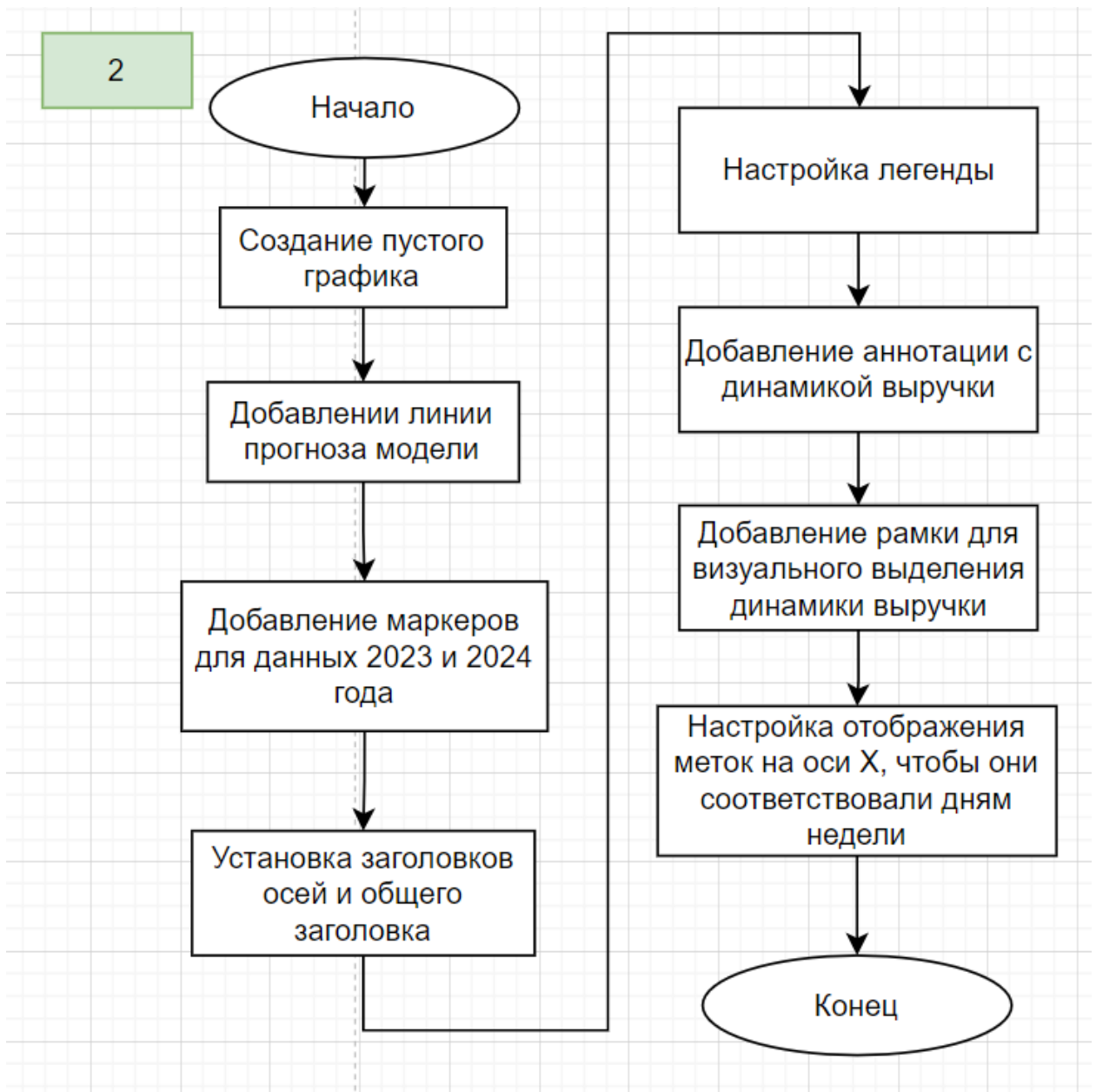


Рисунок 3 - Декомпозиция второго уровня для второго блока

Таким образом, представленная блок-схема демонстрирует комплексный подход к преобразованию, анализу и визуализации данных с использованием модели случайного леса для прогнозирования и оценки качества результатов.

Итогом работы алгоритма станет файл в формате HTML с динамическим графиком динамики выручки.

Выводы по разделу 2:

Во втором разделе был проведен тщательный сравнительный анализ различных алгоритмов машинного обучения для задачи оптимизации. По результатам анализа был выбран алгоритм случайного леса как наиболее подходящий для решения поставленной задачи.

После этого была подробно описана математическая модель случайного леса, включая ее ключевые компоненты и принципы работы. Это позволило глубже понять внутреннюю логику и механизмы функционирования выбранного алгоритма оптимизации.

Кроме того, в этом разделе были детально проработаны компоненты необходимого программного обеспечения. Была спроектирована комплексная система, которая обеспечивает весь жизненный цикл от обработки исходных данных до визуализации результатов.

Такой всесторонний подход, включающий сравнение альтернативных методов, глубокое изучение выбранной модели и тщательное проектирование программной системы, является важным фундаментом для дальнейшей успешной разработки и внедрения решения.

Детальная проработка теоретических и проектных аспектов на этом этапе позволит в дальнейшем более эффективно реализовать и внедрить итоговое программное обеспечение, обеспечивая его надежность, масштабируемость и соответствие поставленным требованиям.

## 3 Реализация и тестирование программного обеспечения.

### 3.1 Реализация программного обеспечения

Описание реализации программного обеспечения всегда начинается с импорта всех необходимых библиотек. Рассмотрим библиотеки, которые использовались в процессе написания программного кода:

- pandas работает с данными.
- numpy производит численные операции.
- plotly создает и сохраняет графики.
- sklearn.ensemble.RandomForestRegressor необходима для создания модели случайного леса.
- datetime работает с датами и временем.

После импорта библиотек объявляются константы, которые будут содержать данные из отчётов о чистой прибыли за разные недели, а также создаётся константа для выходного файла HTML, куда будет сохранен график.

Затем была описана функция `load_data`, которая обрабатывает данные. Она получает на вход 4 файла в формате `xlsx`. В ней для начала функция `pd.read_excel` загружает данные из файлов Excel в датафрейм `data_1`, `data_2`, `data_3`, `data_4`. Потом вычисляются динамика выручки в процентах между первой и второй неделями, между третьей и четвертой неделями и между второй и четвертой неделями.

Также в переменную `average_change` записывается среднее значение динамики выручки, округленное до двух знаков после запятой. В конце работы функции объединяются все четыре датафрейма в один, игнорируя индексы оригинальных датафреймов. Результат работы представлен ниже (рисунок 4).

После этого, в программе была объявлена функция для преобразования даты из файла отчёта в число. Сделано это при помощи функции `data_to_number`, которая объявлена ниже, для того чтобы на графике всё выводилось корректно. Объявление функции представлено ниже (рисунок 5).

```

9 # Константы для файлов данных и выходного файла
10 DATA_1 = '1_week.xlsx'
11 DATA_2 = '2_week.xlsx'
12 DATA_3 = '3_week.xlsx'
13 DATA_4 = '4_week.xlsx'
14 OUTPUT_FILE = 'payments_plot.html'
15
16 # Функция для загрузки и предварительной обработки данных из Excel файлов
17 usage
18 def load_data(file_name_1, file_name_2, file_name_3, file_name_4):
19     data_1 = pd.read_excel(file_name_1)
20     data_2 = pd.read_excel(file_name_2)
21     data_3 = pd.read_excel(file_name_3)
22     data_4 = pd.read_excel(file_name_4)
23
24     # Расчет динамики выручки
25     s1 = ((data_2['amount'].sum() - data_1['amount'].sum()) / data_1['amount'].sum()) * 100
26     s2 = ((data_4['amount'].sum() - data_3['amount'].sum()) / data_3['amount'].sum()) * 100
27     s3 = ((data_4['amount'].sum() - data_2['amount'].sum()) / data_2['amount'].sum()) * 100
28     average_change = ((s1 + s2 + s3) / 3).round(2)
29
30     # Объединение данных из всех файлов и возврат результатов
31     return pd.concat([data_1, data_2, data_3, data_4], ignore_index=True), average_change, s1.round(2), s2.round(2), s3.round(2)

```

Рисунок 4 - Объявление констант и реализация функции для обработки данных

```

32 # Функция для предварительной обработки данных: преобразование даты в число
33 usage
34 def preprocess_data(data):
35     data['date_number'] = data['date'].apply(date_to_number)
36     return data

```

Рисунок 5 - Реализация функции для преобразования данных в число

Затем, были реализованы ещё две функции. В первой функции происходит процесс разделения данных из датафрейма на 2023 и 2024 год. Во второй функции происходит обучение модели случайного леса. Инициализация модели происходит с помощью встроенной функции в Python со следующими параметрами: количество деревьев в лесу – 10, максимальная глубина каждого дерева - 100. Модель обучается на данных X и y. Функция возвращает обученную модель. Объявление функций представлено ниже (рисунок 6).

Далее вызывается функция, которая создаёт предсказания для обученной модели. Также, в функции происходит построение графика и

сохранение его в формате html.

```
37 # Функция для разделения данных на 2023 и 2024 годы
    1 usage
38 def split_data(data):
39     data_2023 = data[data['date'].dt.year == 2023]
40     data_2024 = data[data['date'].dt.year == 2024]
41     return data_2023, data_2024
42
43 # Функция для обучения модели RandomForestRegressor
    1 usage
44 def train_model(data):
45     X = data['date_number'].values.reshape(-1, 1)
46     y = data['amount'].values
47     model = RandomForestRegressor(n_estimators=10, max_depth=100)
48     model.fit(X, y)
49     return model
```

Рисунок 6 - Реализация функций разделения данных по годам и обучения модели случайного леса

В конце реализации функции происходит расчёт среднеквадратичной ошибки и верстка для её вывода.

Результат представлен ниже (рисунок 7).

```
51 # Функция для прогнозирования и создания графика
    1 usage
52 def predict_and_plot(model, data_2023, data_2024, s1, s2, s3, average_change):
53     X_pred = np.linspace(0, 24 * 7, 24 * 7 * 10).reshape(-1, 1)
54     y_pred = model.predict(X_pred)
55     fig = configure_plot(X_pred, y_pred, data_2023, data_2024, s1, s2, s3, average_change)
56     # Вычисление MSE для всех данных
57     mse_total = mean_squared_error(data['amount'], model.predict(data['date_number'].values.reshape(-1, 1)))
58
59     # Добавление MSE в график
60     fig.add_annotation(x=0.5, y=1.05, xref='paper', yref='paper', text=f"Среднеквадратическая ошибка (MSE): {mse_total:.2f}", showarrow=False,
61                       font=dict(size=12))
62     write_html(fig, file=OUTPUT_FILE, auto_open=False)
63
```

Рисунок 7 - Реализация функций прогнозирования и построения графика

Следующая функция `configure_plot` настраивает график. Вначале

настраивается список по дням недели, который используется в качестве меток по оси x. Затем на график выводятся предсказанные ранее данные с помощью модели случайного леса, а также выводятся данные за 2023 год и за 2024 год. После чего на график выводятся предсказанные значения динамики выручки и среднеквадратичной ошибки. Для каждого блока была описана верстка. Результат представлен ниже (рисунок 8).

```
64 # Функция для настройки графика
65 usage
66 def configure_plot(X_pred, y_pred, data_2023, data_2024, s1, s2, s3, average_change):
67     days_of_week = ['Понедельник', 'Вторник', 'Среда', 'Четверг', 'Пятница', 'Суббота', 'Воскресенье']
68     fig = go.Figure()
69     fig.add_trace(go.Scatter(x=X_pred.flatten(), y=y_pred, mode='lines', name='Результирующая модель случайного леса', line=dict(color='red')))
70     fig.add_trace(go.Scatter(x=data_2023['date_number'], y=data_2023['amount'], mode='markers', name='2023', marker=dict(color='blue')))
71     fig.add_trace(go.Scatter(x=data_2024['date_number'], y=data_2024['amount'], mode='markers', name='2024', marker=dict(color='green')))
72     # Настройка макета графика и добавление аннотаций
73     fig.update_layout(
74         xaxis_title='День недели',
75         yaxis_title='Сумма выплаты (руб.)',
76         title='Выплаты по дням недели и часам',
77         legend=dict(orientation="h", yanchor="bottom", y=1.02, xanchor="right", x=1),
78         annotations=[
79             dict(
80                 x=0,
81                 y=max(data_2023['amount'].max(), data_2024['amount'].max()),
82                 text=f"Текущая динамика выручки: {s1}%",
83                 xref='x',
84                 yref='y',
85                 showarrow=False
86             ),
87             dict(
88                 x=0,
89                 y=max(data_2023['amount'].max(), data_2024['amount'].max()) - 700,
90                 text=f"Годовая динамика выручки: {s2}%",
91                 xref='x',
92                 yref='y',
93                 showarrow=False
94             )
95         ]
96     )
```

Рисунок 8 - Реализация функций настройки графика

В конце была реализована функция, которая преобразовывает даты в числовой формат и используется в функции preprocess\_data. В основном блоке кода происходит обработка полученных отчётов, обучение модели случайного леса, прогнозирование значений, построение и сохранение графика в формате html. Результат представлен ниже (рисунок 9).

В ходе реализации программы были решены многие проблемы, например преобразование полученных данных о дате и времени произведения финансовых операций из отчёта в формате xlsx в переменные для записи значений по горизонтальной оси X и визуализация дней недели на оси X для

упрощенного восприятия информации пользователем.

```
131 # Функция для преобразования даты в числовой формат
132 1 usage
132 def date_to_number(date_obj):
133     if isinstance(date_obj, pd.Timestamp):
134         date_str = date_obj.strftime('%Y-%m-%d %H:%M')
135     else:
136         date_str = date_obj
137     date_time_obj = datetime.strptime(date_str, _format: '%Y-%m-%d %H:%M')
138     # Преобразование даты в число, где день недели умножается на 24 (часы в сутках) и добавляется час и минуты
139     return date_time_obj.weekday() * 24 + date_time_obj.hour + date_time_obj.minute / 60.0
140
141 if __name__ == '__main__':
142     data, average_change, s1, s2, s3 = load_data(DATA_1, DATA_2, DATA_3, DATA_4)
143     data = preprocess_data(data)
144     data_2023, data_2024 = split_data(data)
145     model = train_model(data)
146     predict_and_plot(model, data_2023, data_2024, s1, s2, s3, average_change)
```

Рисунок 9 - Реализация основной функции

В итоге разработано программное обеспечение, которое выполняет все необходимые функции по требованиям заказчика в рамках поставленной задачи. Приступим к тестированию ПО.

### 3.2 Тестирование программного обеспечения

В качестве проведения тестирования был выбран метод функционального тестирования ПО, который заключается в том, чтобы проверить как приложение себя ведет и взаимодействует с пользователем, чтобы понять, насколько ожидания заказчика могут быть оправданы.

Благодаря такому тестированию можно выделить слабые места в программном обеспечении и устранить их, расширить функционал и улучшить пользовательский опыт.

Запустим программу и посмотрим, как поведет себя разработанное программное обеспечение и насколько оно готово к взаимодействию. Результат запуска представлен ниже (рисунок 10).



Рисунок 10 - Общий план вывода ПО

На рисунке 10 мы видим построение модели случайного леса. Все зеленые точки на графике – это чистая прибыль за неделю текущего года, а все синие точки – это данные о чистой прибыли компании за предыдущий год. На оси Y изображена сумма чистой прибыли в рублях, а по оси X изображены дни недели, разбитые на временные промежутки.

Информация, для которого берется из четырех файлов отчётов о чистой прибыли компании, была представлена компанией, в которой я прохожу преддипломную практику. В первом столбце представлена информация о времени и дате совершения финансовой операции, а во втором столбце представлена информация о сумме чистой прибыли в рублях, которая необходима для расчёта динамики выручки.

Примеры таких отчётов, которые используются для построения графика представлены ниже (рисунки 11–12). Из рисунков 11–12 видно, что левый столбец преобразуется в число, а правый остается без изменений.



	A	B
1	date	amount
2	23.01.23 18:46	20000
3	23.01.23 19:00	25000
4	24.01.23 10:30	18000
5	24.01.23 11:00	22000
6	25.01.23 15:20	21000
7	25.01.23 16:00	15000
8	26.01.23 12:15	20000
9	26.01.23 13:00	17000
10	27.01.23 9:45	23000

Рисунок 11 - Отчёт о чистой прибыли за четвертую неделю января

	A	B
1	date	amount
2	30.01.23 16:45	23000
3	30.01.23 17:00	15000
4	31.01.23 9:30	20000
5	31.01.23 10:00	17000
6	01.02.23 13:00	23000
7	01.02.23 14:00	20000
8	02.02.23 10:45	19000
9	02.02.23 11:00	22000
10	03.02.23 15:30	25000
11	03.02.23 16:00	21000
12	04.02.23 12:20	24000
13	04.02.23 13:00	28000
14	05.02.23 9:55	22000
15	05.02.23 10:00	20000

Рисунок 12 - Отчёт о чистой прибыли за первую неделю февраля

Рассмотрим более подробно вывод прогноза о динамике выручки (рисунок 13).

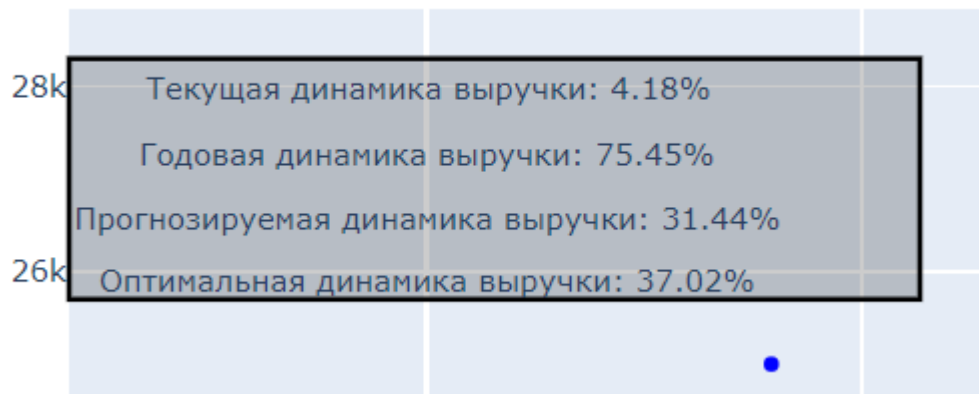


Рисунок 13 - Расчёт динамики выручки

Из рисунка 13 видно, что выводятся четыре показателя динамики выручки. Первое – это текущая еженедельная динамика выручки, которая рассчитывается исходя из текущей и предыдущей недели в этом году. Вторым показателем – это годовая динамика выручки, которая складывается из чистой прибыли на текущей неделе и из такой же недели в прошлом году. Прогнозируемая динамика выручки – это параметр, который складывается из такой же недели в прошлом году и прогнозируемой недели в прошлом году. Наконец, последний параметр – это оптимальная динамика выручки на следующую неделю, которая складывается как среднее значение для всех трех предыдущих параметров.

В график были добавлены функции интерактивности. При наведении на отдельную точку можно определить выручку за этот период и то, из какого года взяты эти данные (рисунок 14).

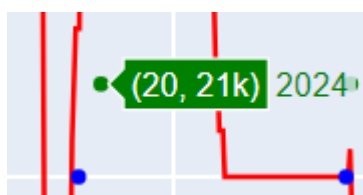


Рисунок 14 - Пример интерактивности графика

Также при нажатии на легенду можно скрывать выбранный элемент из графика и можно приближать график, выделяя область мышью. Также над графиком выводится значение среднеквадратичной ошибки (рисунок 15).



Среднеквадратическая ошибка (MSE): 0.324

Рисунок 15 - Вывод среднеквадратичной ошибки

В результате работы программы не было выявлено никаких ошибок и неточностей, все значения выводятся и рассчитываются корректно. На графике происходит построение средней результирующей динамики выручки методом случайного леса на основе двух отчётов о чистой прибыли кампании за текущую и предыдущую неделю. Это необходимо для того, чтобы проанализировать и предсказать еженедельные пики продаж и динамику выручки в определенные дни и часы. Также график является интерактивным, на нём просчитывается значение среднеквадратичной ошибки, а также динамика выручки еженедельная, годовая и прогнозируемая. Полученное в результате программное обеспечение полностью отвечает всем требованиям заказчика.

Выводы по разделу 3:

В третьем разделе ВКР была описана реализация программного кода и протестировано программное обеспечение методом функционального тестирования.

Также был произведен анализ, полученных в ходе разработки, результатов.

## Заключение

Выполненная бакалаврская работа посвящена разработке алгоритма оптимизации на основе методов машинного обучения для задачи планирования ресурсов. Алгоритм оптимизации на основе методов машинного обучения для задачи планирования ресурсов - это программное решение, которое использует технологии искусственного интеллекта для повышения эффективности управления финансовыми и материальными ресурсами организации. Основная цель таких алгоритмов - повысить эффективность управления ресурсами за счет более точного прогнозирования динамики ключевых показателей и оптимизации процессов планирования. Это позволяет компаниям сокращать издержки, улучшать финансовые результаты и повышать конкурентоспособность.

В ходе исследования были поставлены и успешно решены следующие задачи:

- проведен обзор предметной области и описаны ресурсы, которые могут быть использованы в задаче планирования ресурсов. Среди финансовых ресурсов компании был выделен наиболее важный параметр для оценки - динамика выручки. Подробно рассмотрен термин динамики выручки и его практическое применение;
- выполнен сравнительный анализ различных алгоритмов оптимизации на основе машинного обучения. По результатам анализа был выбран алгоритм случайного леса, для которого подробно описана математическая модель;
- разработана модульная архитектура программной системы, проведено ее проектирование для дальнейшей реализации;
- реализовано программное обеспечение, осуществлено его функциональное тестирование. Тестирование проводилось на основе отчетов о чистой прибыли компании, где проходила преддипломная практика.

В результате проведенной работы было разработано программное обеспечение, которое позволяет компаниям прогнозировать и планировать динамику выручки с использованием методов машинного обучения. Данное решение предназначено для оптимизации процессов планирования финансовых ресурсов в организациях.

Основные возможности разработанного программного обеспечения:

- прогнозирование динамики выручки на основе методов машинного обучения;
- планирование финансовых ресурсов с учетом прогнозируемых показателей выручки;
- применение современных методов прогнозирования для повышения точности и эффективности финансового планирования.

Данное решение может быть внедрено в практическую деятельность различных организаций для оптимизации процессов управления финансовыми ресурсами. Дальнейшее развитие проекта может включать расширение функциональности, интеграцию с другими системами управления ресурсами, а также адаптацию к специфике различных отраслей.

## Список используемой литературы и используемых источников

1. Авербух А. И. Компьютерное моделирование: система ModSim. — М.: Интернет-Университет Информационных Технологий (ИУИТ), 2005. — 352 с.
2. Александров В.П. Алгоритмы и методы оптимизации в задачах планирования ресурсов. [Электронный ресурс]: учебное пособие / Александров В.П. — Электрон. текстовые данные. — Санкт-Петербург: Санкт-Петербургский государственный университет, 2019. — 50 с.
3. Баранов А.Н. Методы машинного обучения в задачах оптимизации. [Электронный ресурс]: монография / Баранов А.Н. — Электрон. текстовые данные. — Москва: МГУ имени М.В. Ломоносова, 2020. — 120 с.
4. Борисов С.В. Динамика выручки в условиях неопределенности. [Электронный ресурс]: монография / Борисов С.В. — Электрон. текстовые данные. — Екатеринбург: УрФУ имени первого Президента России Б.Н. Ельцина, 2018. — 95 с.
5. Васильев А.К. Применение моделей случайного леса в экономике. [Электронный ресурс]: учебное пособие / Васильев А.К. — Электрон. текстовые данные. — Новосибирск: НГУ, 2017. — 70 с.
6. Голубев И.В. Оптимизация планирования производственных ресурсов на основе машинного обучения. [Электронный ресурс]: монография / Голубев И.В. — Электрон. текстовые данные. — Санкт-Петербург: ИТМО, 2019. — 130 с.
7. Дмитриев П.Н. Машинное обучение и оптимизация: теория и приложения. [Электронный ресурс]: учебное пособие / Дмитриев П.Н. — Электрон. текстовые данные. — Томск: ТПУ, 2021. — 80 с.
8. Егоров М.А. Алгоритмы планирования ресурсов в условиях неопределенности. [Электронный ресурс]: монография / Егоров М.А. — Электрон. текстовые данные. — Москва: ВШЭ, 2016. — 110 с.
9. Жуков Д.С. Анализ динамики выручки с использованием моделей

машинного обучения. [Электронный ресурс]: учебное пособие / Жуков Д.С. — Электрон. текстовые данные. — Казань: КФУ, 2019. — 90 с.

10. Зайцев В.Н. Модели и методы оптимизации для задач планирования ресурсов. [Электронный ресурс]: монография / Зайцев В.Н. — Электрон. текстовые данные. — Санкт-Петербург: СПбГУ, 2017. — 100 с.

11. Иванов А.А. Применение модели случайного леса в прогнозировании выручки. [Электронный ресурс]: учебное пособие / Иванов А.А. — Электрон. текстовые данные. — Новосибирск: НГУЭУ, 2018. — 60 с.

12. Казаков В.И. Машинное обучение в задачах планирования и управления ресурсами. [Электронный ресурс]: учебное пособие / Казаков В.И. — Электрон. текстовые данные. — Москва: МГУ, 2020. — 85 с.

13. Кириллов Д.М. Анализ временных рядов и динамика выручки. [Электронный ресурс]: монография / Кириллов Д.М. — Электрон. текстовые данные. — Екатеринбург: УрФУ, 2017. — 105 с.

14. Козлов П.Т. Методы машинного обучения в экономике. [Электронный ресурс]: учебное пособие / Козлов П.Т. — Электрон. текстовые данные. — Новосибирск: НГУ, 2019. — 75 с.

15. Лебедев Н.С. Прогнозирование выручки с использованием моделей машинного обучения. [Электронный ресурс]: монография / Лебедев Н.С. — Электрон. текстовые данные. — Санкт-Петербург: ИТМО, 2018. — 90 с.

16. Макаров И.В. Оптимизация ресурсного планирования на предприятиях. [Электронный ресурс]: учебное пособие / Макаров И.В. — Электрон. текстовые данные. — Томск: ТПУ, 2017. — 80 с.

17. Никитин О.А. Модели случайного леса в задачах прогнозирования. [Электронный ресурс]: учебное пособие / Никитин О.А. — Электрон. текстовые данные. — Москва: МФТИ, 2020. — 70 с.

18. Орлов К.С. Динамика выручки в условиях рыночной экономики. [Электронный ресурс]: монография / Орлов К.С. — Электрон. текстовые данные. — Екатеринбург: УрФУ, 2018. — 85 с.

19. Павлов А.В. Применение алгоритмов машинного обучения для

задач планирования. [Электронный ресурс]: монография / Павлов А.В. — Электрон. текстовые данные. — Санкт-Петербург: ИТМО, 2019. — 95 с.

20. Романов М.Л. Оптимизация планирования ресурсов с использованием машинного обучения. [Электронный ресурс]: учебное пособие / Романов М.Л. — Электрон. текстовые данные. — Казань: КФУ, 2021. — 90 с.

21. Савельев П.Н. Прогнозирование временных рядов и динамика выручки. [Электронный ресурс]: монография / Савельев П.Н. — Электрон. текстовые данные. — Москва: МГУ, 2017. — 100 с.

22. Тихонов В.В. Методы машинного обучения для задач планирования ресурсов. [Электронный ресурс]: учебное пособие / Тихонов В.В. — Электрон. текстовые данные. — Новосибирск: НГУ, 2020. — 80 с.

23. Уваров Е.Ю. Динамика выручки и машинное обучение. [Электронный ресурс]: монография / Уваров Е.Ю. — Электрон. текстовые данные. — Санкт-Петербург: СПбГУ, 2019. — 110 с.

24. Федоров Д.С. Модель случайного леса и её применение в экономике. [Электронный ресурс]: учебное пособие / Федоров Д.С. — Электрон. текстовые данные. — Москва: ВШЭ, 2018. — 70 с.

25. Харитонов В.П. Алгоритмы оптимизации в задачах планирования ресурсов. [Электронный ресурс]: монография / Харитонов В.П. — Электрон. текстовые данные. — Екатеринбург: УрФУ, 2020. — 95 с.

26. Breiman L. Random Forests. *Machine Learning*, 2001. — 45(1): 5-32.

27. Bertsimas D., Kallus N. From Predictive to Prescriptive Analytics. *Management Science*, 2020. — 66(3): 1025-1044.

28. Chandola V., Banerjee A., Kumar V. Anomaly Detection: A Survey. *ACM Computing Surveys*, 2009. — 41(3): 1-58.

29. Cortes C., Vapnik V. Support-vector networks. *Machine Learning*, 1995. — 20(3): 273-297.

30. Domingos P. A few useful things to know about machine learning. *Communications of the ACM*, 2012. — 55(10): 78-87.



31. Hastie T., Tibshirani R., Friedman J. The Elements of Statistical Learning: Data Mining, Inference, and Prediction. 2nd ed. Springer, 2009. — 745 p.
32. Ho T.K. Random decision forests. Proceedings of the 3rd International Conference on Document Analysis and Recognition, 1995. — 278-282.
33. Krizhevsky A., Sutskever I., Hinton G.E. ImageNet classification with deep convolutional neural networks. Advances in Neural Information Processing Systems, 2012. — 1097-1105.
34. Lecun Y., Bengio Y., Hinton G. Deep learning. Nature, 2015. — 521: 436-444.
35. Nabney I.T. Netlab: Algorithms for Pattern Recognition. Springer, 2002. — 432 p.
36. Witten I.H., Frank E., Hall M.A., Pal C.J. Data Mining: Practical Machine Learning Tools and Techniques. 4th ed. Morgan Kaufmann, 2016. — 654 p.

## Приложение А

### Листинг 1 (код программного обеспечения по расчёту динамики выручки)

Листинг 1 - Код программного обеспечения по расчёту динамики выручки

```
import pandas as pd
import numpy as np
import plotly.graph_objs as go
from plotly.io import write_html
from sklearn.ensemble import RandomForestRegressor
from sklearn.metrics import mean_squared_error
from datetime import datetime

# Константы для файлов данных и выходного файла
DATA_1 = '1_week.xlsx'
DATA_2 = '2_week.xlsx'
DATA_3 = '3_week.xlsx'
DATA_4 = '4_week.xlsx'
OUTPUT_FILE = 'payments_plot.html'

# Функция для загрузки и предварительной обработки данных из Excel файлов
def load_data(file_name_1, file_name_2, file_name_3, file_name_4):
    data_1 = pd.read_excel(file_name_1)
    data_2 = pd.read_excel(file_name_2)
    data_3 = pd.read_excel(file_name_3)
    data_4 = pd.read_excel(file_name_4)
```

## Продолжение Приложения А

```
# Расчет динамики выручки
s1 = ((data_2['amount'].sum() - data_1['amount'].sum()) / data_1['amount'].sum())
* 100
s2 = ((data_4['amount'].sum() - data_3['amount'].sum()) / data_3['amount'].sum())
* 100
s3 = ((data_4['amount'].sum() - data_2['amount'].sum()) / data_2['amount'].sum())
* 100
average_change = ((s1 + s2 + s3) / 3).round(2)

# Объединение данных из всех файлов и возврат результатов
return pd.concat([data_1, data_2, data_3, data_4], ignore_index=True),
average_change, s1.round(2), s2.round(2), s3.round(2)

# Функция для предварительной обработки данных: преобразование даты в
число
def preprocess_data(data):
    data['date_number'] = data['date'].apply(date_to_number)
    return data

# Функция для разделения данных на 2023 и 2024 годы
def split_data(data):
    data_2023 = data[data['date'].dt.year == 2023]
    data_2024 = data[data['date'].dt.year == 2024]
    return data_2023, data_2024

# Функция для обучения модели RandomForestRegressor
def train_model(data):
    X = data['date_number'].values.reshape(-1, 1)
```

## Продолжение Приложения А

```
y = data['amount'].values
model = RandomForestRegressor(n_estimators=10, max_depth=100)
model.fit(X, y)
return model

# Функция для прогнозирования и создания графика
def predict_and_plot(model, data_2023, data_2024, s1, s2, s3, average_change):
    X_pred = np.linspace(0, 24 * 7, 24 * 7 * 10).reshape(-1, 1)
    y_pred = model.predict(X_pred)
    fig = configure_plot(X_pred, y_pred, data_2023, data_2024, s1, s2, s3,
average_change)
    # Вычисление MSE для всех данных
    mse_total = mean_squared_error(data['amount'],
model.predict(data['date_number'].values.reshape(-1, 1)))

    # Добавление MSE в график
    fig.add_annotation(x=0.5, y=1.05, xref='paper', yref='paper',
text=f"Среднеквадратическая ошибка (MSE): {mse_total:.2f}",
showarrow=False,
font=dict(size=12))
    write_html(fig, file=OUTPUT_FILE, auto_open=False)

# Функция для настройки графика
def configure_plot(X_pred, y_pred, data_2023, data_2024, s1, s2, s3,
average_change):
    days_of_week = ['Понедельник', 'Вторник', 'Среда', 'Четверг', 'Пятница',
'Суббота', 'Воскресенье']
    fig = go.Figure()
```

## Продолжение Приложения А

```
fig.add_trace(go.Scatter(x=X_pred.flatten(), y=y_pred, mode='lines',
name='Результирующая модель случайного леса', line=dict(color='red')))
fig.add_trace(go.Scatter(x=data_2023['date_number'], y=data_2023['amount'],
mode='markers', name='2023', marker=dict(color='blue')))
fig.add_trace(go.Scatter(x=data_2024['date_number'], y=data_2024['amount'],
mode='markers', name='2024', marker=dict(color='green')))
# Настройка макета графика и добавление аннотаций
fig.update_layout(
    xaxis_title='День недели',
    yaxis_title='Сумма выплаты (руб.)',
    title='Выплаты по дням недели и часам',
    legend=dict(orientation="h", yanchor="bottom", y=1.02, xanchor="right",
x=1),
    annotations=[
        dict(
            x=0,
            y=max(data_2023['amount'].max(), data_2024['amount'].max()),
            text=f"Текущая динамика выручки: {s1}%",
            xref='x',
            yref='y',
            showarrow=False
        ),
        dict(
            x=0,
            y=max(data_2023['amount'].max(), data_2024['amount'].max()) - 700,
            text=f"Годовая динамика выручки: {s2}%",
            xref='x',
            yref='y',
```

## Продолжение Приложения А

```
    showarrow=False
),
dict(
    x=0,
    y=max(data_2023['amount'].max(), data_2024['amount'].max()) - 1400,
    text=f"Прогнозируемая динамика выручки: {s3}%",
    xref='x',
    yref='y',
    showarrow=False
),
dict(
    x=0,
    y=max(data_2023['amount'].max(), data_2024['amount'].max()) - 2100,
    text=f"Оптимальная динамика выручки: {average_change}%",
    xref='x',
    yref='y',
    showarrow=False
),
],
shapes=[
    dict(
        type='rect',
        xref='paper',
        yref='y',
        x0=0,
        y0=max(data_2023['amount'].max(), data_2024['amount'].max()) - 2300,
        x1=0.25,
        y1=max(data_2023['amount'].max(), data_2024['amount'].max()) + 300,
```

## Продолжение Приложения А

```
        line=dict(
            color='black',
            width=2,
        ),
        fillcolor='rgba(0,0,0,0.2)'
    )
]
)
fig.update_xaxes(tickvals=list(range(24, 192, 24)), ticktext=days_of_week)
return fig

# Функция для преобразования даты в числовой формат
def date_to_number(date_obj):
    if isinstance(date_obj, pd.Timestamp):
        date_str = date_obj.strftime('%Y-%m-%d %H:%M')
    else:
        date_str = date_obj
    date_time_obj = datetime.strptime(date_str, '%Y-%m-%d %H:%M')
    # Преобразование даты в число, где день недели умножается на 24 (часы в
    сутках) и добавляется час и минуты
    return date_time_obj.weekday() * 24 + date_time_obj.hour +
date_time_obj.minute / 60.0
if __name__ == '__main__':
    data, average_change, s1, s2, s3 = load_data(DATA_1, DATA_2, DATA_3,
DATA_4)
    data = preprocess_data(data)
    data_2023, data_2024 = split_data(data)
    model = train_model(data)
    predict_and_plot(model, data_2023, data_2024, s1, s2, s3, average_change)
```