

МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ
БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
ФАКУЛЬТЕТ ПРИКЛАДНОЙ МАТЕМАТИКИ И
ИНФОРМАТИКИ

Кафедра компьютерных технологий и систем

ФИЛЬКОВА
Ульяна Григорьевна

РАЗРАБОТКА КЛИЕНТ-СЕРВЕРНОГО ПРИЛОЖЕНИЯ ДЛЯ
АНАЛИЗА И ПРОГНОЗИРОВАНИЯ ДАННЫХ

Дипломная работа

Научные руководители:
Кандидат физико-
математических наук,
доцент
Е.С. Чеб,
Ассистент кафедры КТС
А.Г. Каркоцкий

Допущена к защите

«__» _____ 20__ г.

Заведующий кафедрой компьютерных технологий и систем
доктор педагогических наук, профессор В.В. Казаченок

Минск, 2025

ОГЛАВЛЕНИЕ

ВВЕДЕНИЕ.....	8
ГЛАВА 1.....	9
МЕТОДЫ ПРОГНОЗИРОВАНИЯ.....	9
1.1 Качественные методы.....	9
1.2 Количественные методы.....	9
1.2.1 Временные ряды.....	10
1.2.2 Причинно-следственные методы.....	11
1.2.3 Машинное обучение.....	12
1.2.4 Комбинированные методы.....	12
1.3 Критерии выбора метода прогнозирования.....	13
1.4 Проблемы и ограничения методов прогнозирования.....	14
1.4.1 Нехватка или низкое качество данных.....	14
1.4.2 Переобучение (overfitting).....	15
1.4.3 Сложности интерпретации моделей.....	15
1.4.4 Ограничения качественных методов.....	16
1.4.5 Ошибки в условиях нестабильности и кризисов.....	16
1.4.6 Проблема устойчивости.....	17
1.5 Оценка качества прогноза.....	17
1.5.1 Основные метрики точности прогнозов.....	17
1.5.2 Кросс-валидация.....	19
1.5.3 Разделение на тренировочную и тестовую выборки.....	19
1.5.4 Бэк-тестирование (Backtesting).....	20
1.5.5 Выбор метрик и подходов в зависимости от задачи.....	20
1.6 Основные выводы по главе.....	21
ГЛАВА 2.....	22
ВЫБОР МЕТОДОВ ПРОГНОЗИРОВАНИЯ.....	22
2.1 Линейная регрессия.....	22
2.1.1 Основная идея линейной регрессии.....	22
2.1.2 Типы линейной регрессии.....	22

2.1.3 Особенности линейной регрессии	24
2.1.4 Примеры применения линейной регрессии	24
2.2 Модель ARIMA	24
2.2.1 AR (Авторегрессия)	25
2.2.2 I (Интеграция)	25
2.2.3 MA (Скользящая средняя)	26
2.2.4 Математическая формулировка ARIMA	26
2.2.5 Алгоритм работы с ARIMA	27
2.2.6 Примеры применения ARIMA.....	27
2.3 Метод случайных лесов.....	28
2.3.1 Основная идея метода случайных лесов	28
2.3.2 Построение одного дерева в случайном лесу	29
2.3.3 Алгоритм случайного леса.....	29
2.3.4 Примеры применения случайных лесов.....	30
2.4 Метод К ближайших соседей	30
2.4.1 Алгоритм метода k-ближайших соседей.....	30
2.4.2 Основные параметры метода.....	31
2.4.3 Особенности метода KNN.....	31
2.4.4 Примеры применения метода KNN	31
2.5 Основные выводы по главе	32
ГЛАВА 3.....	34
РАЗРАБОТКА КЛИЕНТ-СЕРВЕРНОГО ПРИЛОЖЕНИЯ	34
3.1 Постановка цели приложения.....	34
3.2 Функционал приложения	34
3.3 Выбор инструментов	34
3.3.1 Клиентская часть.....	34
3.3.2 Серверная часть.....	35
3.4 Описание взаимодействия API в приложении.....	35
3.5 Интерфейс приложения.....	35
3.6 Тестирование приложения	42
3.7 Сравнение с аналогичными решениями	44

3.7.1 Обзор аналогичных технологий и приложений	45
3.7.2 Отличительные особенности и преимущества разработанного приложения.....	46
3.8 Основные выводы по главе	46
ЗАКЛЮЧЕНИЕ	48
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ	49
ПРИЛОЖЕНИЕ А	51
ПРИЛОЖЕНИЕ Б.....	55
ПРИЛОЖЕНИЕ В	59

РЕФЕРАТ

Дипломная работа – 61 с., 9 рис., 14 формул, 3 таблицы, 18 источников, 3 приложения.

Ключевые слова: ПРОГНОЗИРОВАНИЕ; PYTHON; VUE; МАШИННОЕ ОБУЧЕНИЕ; ВРЕМЕННЫЕ РЯДЫ; ЛИНЕЙНАЯ РЕГРЕССИЯ; ARIMA; СЛУЧАЙНЫЙ ЛЕС; К БЛИЖАЙШИХ СОСЕДЕЙ; ВИЗУАЛИЗАЦИЯ.

Объект исследования – методы прогнозирования числовых показателей по временным и категориальным данным, а также реализация этих методов в программном обеспечении с помощью языка программирования Python.

Цель работы – разработка приложения, реализующего различные подходы к прогнозированию значений на основе анализа входных данных с использованием методов машинного обучения и статистики.

Инструменты разработки – для реализации фронтенд-части использовался фреймворк Vue.js с библиотеками Pinia (для управления состоянием), Vue Router (для маршрутизации), Vuelidate (для валидации форм), Chart.js и vue-chartjs (для построения графиков), ExcelJS и xlsx (для работы с Excel-файлами), jsPDF и jsPDF-autotable (для генерации PDF-отчетов), а также MathJax (для отображения формул). Для разработки и сборки проекта применялись инструменты Vite, Prettier и ESLint. Бэкенд-часть была реализована на языке Python с использованием библиотек Pandas (для обработки табличных данных), Scikit-learn (для построения и обучения моделей), Statsmodels (для временных рядов), Matplotlib и Seaborn (для построения визуализаций), а также Flask (для создания серверной части приложения и API).

Результаты – приложение, которое позволяет загружать пользовательские данные, выбирать метод прогнозирования, настраивать параметры моделей, визуализировать полученные прогнозы в виде графиков и таблиц и сохранять полученные данные, а также ознакомиться с теорией прогнозирования и изучить конкретные примеры прогнозирования данных с помощью различных методов.

Область применения – учебные учреждения, образовательные платформы, исследовательские проекты и др.

РЭФЕРАТ

Дыпломная праца – 61 с., 9 мал., 14 формул, 3 табліцы, 18 крыніц, 3 прыкладанні.

Ключавыя словы: ПРАГНОЗАВАННЕ; PYTHON; VUE; МАШЫННАЕ НАВУЧАННЕ; ЧАСОВЫЯ ШЭРАГІ; ЛІНІЙНАЯ РЭГРЭСІЯ; ARIMA; ВЫПАДКОВЫ ЛЕС; К БЛІЖАЙШЫХ СУСЕДЗЯЎ; ВІЗУАЛІЗАЦЫЯ.

Аб'ект даследавання – метады прагназавання лікавых паказчыкаў паводле часовых і катэгарыяльных даных, а таксама рэалізацыя гэтых метадаў у праграмным забеспячэнні з дапамогай мовы праграмавання Python.

Мэта працы – распрацоўка прыкладання, якое рэалізуе розныя падыходы да прагназавання значэнняў на аснове аналізу ўводных даных з выкарыстаннем метадаў машыннага навучання і статыстыкі.

Інструменты распрацоўкі – для рэалізацыі фронтэнду выкарыстоўваўся фрэймворк Vue.js з бібліятэкамі Pinia (для кіравання станам), Vue Router (для маршрутызацыі), Vuelidate (для валідацыі форм), Chart.js і vue-chartjs (для пабудовы графікаў), ExcelJS і xlsx (для працы з Excel-файламі), jsPDF і jsPDF-autotable (для генерацыі PDF-справаздач), а таксама MathJax (для адлюстравання формул). Для распрацоўкі і зборкі праекта ўжываліся інструменты Vite, Prettier і ESLint. Бэкэнд-частка была рэалізавана на мове Python з выкарыстаннем бібліятэк Pandas (для апрацоўкі таблічных даных), Scikit-learn (для пабудовы і навучання мадэляў), Statsmodels (для часовай аналітыкі), Matplotlib і Seaborn (для стварэння візуалізацый), а таксама Flask (для стварэння сервернай часткі і API).

Вынікі – прыкладанне, якое дазваляе загружаць карыстальніцкія даныя, выбіраць метады прагназавання, наладжваць параметры мадэляў, візуалізаваць атрыманыя прагнозы ў выглядзе графікаў і табліц, захоўваць вынікі, а таксама знаёміцца з тэорыяй прагназавання і вывучаць канкрэтныя прыклады прагназавання з выкарыстаннем розных метадаў.

Вобласць прымянення – навучальныя ўстановы, адукацыйныя платформы, даследчыя праекты і інш.

ABSTRACT

Graduate work – 61 p., 9 illustrations, 14 formulas, 3 tables, 18 sources, 3 appendices.

Keywords: FORECASTING; PYTHON; VUE; MACHINE LEARNING; TIME SERIES; LINEAR REGRESSION; ARIMA; RANDOM FOREST; K-NEAREST NEIGHBORS; VISUALIZATION.

The object of research is methods for forecasting numerical indicators based on time series and categorical data, as well as the implementation of these methods in software using the Python programming language.

The goal of the work is to develop an application that implements various approaches to forecasting values based on input data analysis using machine learning and statistical methods.

Development tools: the frontend part was implemented using the Vue.js framework with libraries such as Pinia (for state management), Vue Router (for routing), Vuelidate (for form validation), Chart.js and vue-chartjs (for chart rendering), ExcelJS and xlsx (for working with Excel files), jsPDF and jsPDF-autotable (for generating PDF reports), and MathJax (for displaying formulas). Project development and building were performed using Vite, Prettier, and ESLint. The backend part was implemented in Python using libraries such as Pandas (for data manipulation), Scikit-learn (for model building and training), Statsmodels (for time series analysis), Matplotlib and Seaborn (for data visualization), and Flask (for creating the server and API).

Results: an application that allows users to upload their own data, select a forecasting method, configure model parameters, visualize the results in the form of charts and tables, save the output, and also explore the theory of forecasting and study specific examples of data forecasting using various methods.

Scope of application: educational institutions, learning platforms, research projects, etc.

ВВЕДЕНИЕ

Прогнозирование данных играет важную роль при принятии решений в условиях неопределенности. Спектр его применения крайне широк. Это и экономика, и бизнес, и здравоохранение, и, конечно же, информационные технологии. При прогнозировании используются различные методы: экспертные оценки, временные ряды, а также причинные модели. Кроме того, активно внедряется машинное обучение.

Объемы данных непрерывно растут, это приводит к повышению спроса на услуги по автоматизации прогнозирования. При помощи алгоритмов выявляются скрытые зависимости и связи внутри данных, это приводит к значительному улучшению точности и адаптивности прогнозов. Результаты зависят от многих факторов, но качество данных и доступность вычислительных ресурсов играют в данных процессах ключевую роль.

Цель данной работы заключается в проектировании и разработке клиент-серверного приложения, которое позволит пользователю ознакомиться с различными методами прогнозирования, а также применить их на практике. В разработанном приложении присутствует возможность загрузить собственные данные, а также выбрать метод прогнозирования. По итогу работы приложения результаты прогнозирования наглядно представляются в виде графиков и таблиц.

В данный проект интегрированы различные методы прогнозирования: линейная регрессия, модели ARIMA, случайный лес, а также алгоритм k-ближайших соседей. Данные методы применяются в разнообразных сферах прогнозирования с различными целями. Пользователь приложения имеет возможность сравнить указанные методы и понять, какой метод ему необходим с учётом конкретных целей.

Клиентская часть данного приложения разработана с применением фреймворка Vue.js. Серверная часть – на языке Python с использованием Flask. Для визуализации итоговых данных применена библиотека Chart.js.

В первой главе текущей работы описываются теоретические аспекты методов прогнозирования. Во второй главе список методов ограничивается четырьмя различными подходами к автоматическому прогнозированию, данные подходы рассматриваются более детально и подробно. В третьей главе описывается процесс проектирования и реализации приложения, кроме того, в ней приводится анализ функционала и результаты тестирования данного приложения.

ГЛАВА 1

МЕТОДЫ ПРОГНОЗИРОВАНИЯ

Существуют разнообразные методы прогнозирования, которые можно условно разделить на две большие категории: **качественные** и **количественные**. Каждая из этих категорий имеет собственные принципы, преимущества и области применения. Количественные методы основаны на математических расчетах, в то время как качественные методы включают экспертные оценки различного рода [1, с. 16]. Также существуют гибридные и интеллектуальные подходы, такие как **машинное обучение** и **комбинированные методы**. Они сочетают в себе лучшие стороны традиционных подходов.

1.1 Качественные методы

Качественные методы прогнозирования применяются, если данные ограничены или при высокой неопределенности. Они не опираются на числовые расчеты, а используют мнение экспертов, интуицию и логический анализ. Опытные аналитики могут учесть в своих прогнозах множество определяющих факторов. Такие факторы зачастую просто невозможно выразить математическими уравнениями.

Ключевая особенность качественных методов – это то, что они основаны на субъективных оценках и опыте. Они позволяют учесть факторы, которые нельзя формализовать. Например, политические риски, поведение конкурентов, социальные тренды и т.д. Такие методы применяются в основном при стратегическом планировании и разработке новых продуктов.

Наиболее распространенные качественные подходы:

- **Метод экспертных оценок.** Эксперты обсуждают проблему, делятся индивидуальными прогнозами и на основе консенсуса формируют общий прогноз, который считается более надежным, чем индивидуальные.
- **Метод Дельфи.** Представляет собой итеративный процесс опроса анонимных экспертов, при котором происходит несколько раундов ответов и уточнений [1, с. 32]. Позволяет минимизировать влияние авторитета и достичь взвешенного коллективного решения. Метод популярен в исследованиях, связанных с развитием технологий, социологических и политических прогнозах.

1.2 Количественные методы

Количественные методы представляют собой формализованные подходы, основанные на статистических и математических моделях. Их основное преимущество заключается в возможности объективного измерения, верификации результатов и воспроизводимости прогнозов [2, с. 23]. Данные методы широко применяются в научных исследованиях, экономике, инженерии, а также в информационных системах и цифровой аналитике.

Количественные методы классифицируются на три большие группы: методы анализа временных рядов, причинно-следственные методы и методы, основанные на машинном обучении. Все чаще в последние годы они комбинируются между собой, а также с качественными методами, образуя так называемые гибридные модели.

1.2.1 Временные ряды

Методы временных рядов позволяют анализировать данные, представленные в виде последовательности значений, отсортированных по времени. Основная идея — выявление устойчивых закономерностей, таких как тренды, сезонные и циклические колебания, автокорреляции.

Основные методы временных рядов:

- **Скользящее среднее (Moving Average, MA):** простой способ сглаживания шума в данных. Используется для получения усредненной тенденции, особенно в продажах и потреблении ресурсов.
- **Взвешенное скользящее среднее** — придает разный вес недавним и старым значениям, улучшая чувствительность к изменениям.
- **Экспоненциальное сглаживание (Exponential Smoothing)** — улучшает реакцию модели на последние события. Используется как в однофакторных, так и в многокомпонентных (тренд, сезонность) задачах.
- **Holt's Linear Trend Model** — расширение экспоненциального сглаживания с учетом линейного тренда.
- **Holt-Winters (seasonal exponential smoothing)** — применяется для сезонных временных рядов (например, трафик, спрос).
- **ARIMA (Autoregressive Integrated Moving Average)** — один из самых мощных классических методов прогнозирования. Требует стационарности данных и включает авторегрессию (AR), интеграцию (I), скользящее среднее (MA).
- **SARIMA** — сезонная версия ARIMA, учитывающая годовые и квартальные циклы.

- **Prophet** — модель от Meta (Facebook), ориентирована на бизнес-аналитику: легко интерпретируемая, учитывает праздники и тренды.
- **LSTM (Long Short-Term Memory)** — нейросетевой подход к анализу временных рядов, подходит для задач с нелинейной зависимостью и большим объемом данных.

Примеры применения:

- Финансовые рынки (прогноз курса акций).
- Электроэнергетика (планирование нагрузки).
- Ритейл (предсказание спроса и товарных остатков).
- Прогнозирование погоды и трафика.

1.2.2 Причинно-следственные методы

В отличие от анализа временных рядов, причинно-следственные подходы (или структурные модели) предполагают наличие независимых переменных, оказывающих влияние на прогнозируемый показатель. Эти модели отвечают не только на вопрос «что произойдет», но и на «почему это произойдет».

Основные методы причинно-следственного анализа:

- **Простая линейная регрессия** — базовая модель зависимости между двумя переменными. Пример: зависимость объема продаж от рекламного бюджета.
- **Множественная регрессия** — расширяет модель, позволяя учитывать сразу несколько факторов (например, цена, сезон, конкуренция).
- **Нелинейные регрессионные модели** — используются, когда зависимость между переменными не подчиняется линейному закону.
- **Логистическая регрессия** — применяется для прогнозирования вероятностей событий (например, вероятность оттока клиента).
- **VAR (Vector Autoregression)** — многомерная модель, используемая в эконометрике.
- **Модели структурных уравнений (SEM)** — объединяют регрессию и факторный анализ, применимы в социологии, психологии и маркетинге.
- **Коэффициент эластичности** — позволяет оценить чувствительность спроса к изменениям цены, дохода и т. д.

Области применения:

- Маркетинг (оценка влияния рекламных кампаний).
- Макроэкономика (влияние процентной ставки на ВВП).

- Здравоохранение (оценка факторов риска заболеваний).

1.2.3 Машинное обучение

Машинное обучение представляет собой набор алгоритмов, которые обучаются на данных для выявления сложных закономерностей. Эти методы способны работать с большими объемами данных и обеспечивают высокую точность [4, с. 480].

Популярные методы машинного обучения:

- **Линейные модели с регуляризацией** (Ridge, Lasso, ElasticNet) — эффективны при большом количестве признаков.
- **Деревья решений и ансамбли** (Random Forest, Gradient Boosting, XGBoost, LightGBM, CatBoost) — обладают высокой точностью, устойчивостью к выбросам, автоматически определяют важность признаков.
- **Нейронные сети** — глубокие архитектуры (DNN), сверточные сети (CNN), рекуррентные сети (RNN), LSTM. Используются при больших данных, особенно для временных и текстовых данных.
- **К-ближайших соседей (KNN)** — работает по принципу сходства объектов.
- **Методы кластеризации (k-Means, DBSCAN)** — они используются для сегментации какого-то множества перед прогнозированием.
- **AutoML** — автоматизация процесса построения моделей.

Их преимущества — это адаптивность и способность обучаться на новых данных. Они применимы к табличным данным, изображениям, текстам и аудиосигналам.

Примеры применения AutoML — рекомендательные системы, предсказание кредитных рисков и прогнозирование технических сбоев на производстве.

1.2.4 Комбинированные методы

Комбинированные методы используют сразу несколько подходов. Это может быть сочетание качественных и количественных методов. Также может быть комбинирование различных математических моделей.

Основные подходы:

- **Взвешенное объединение прогнозов**
- **Stacking**
- **Blending**

- **Hybrid ARIMA-LSTM**
- **Экспертно-алгоритмическое объединение**

Применение: сложные экономические модели, прогнозирование в условиях неопределенности.

1.3 Критерии выбора метода прогнозирования

Определение наиболее подходящего метода прогнозирования зависит от ряда критических факторов (см. таблицу 1.1):

1. Доступность данных:

- Если данных нет или мало, то используют качественные методы.
- Если данных достаточно, то применяют количественные подходы.

2. Цели прогнозирования:

- Если задача краткосрочная, то используют временные ряды.
- Если задача долгосрочная, то используют сценарные и причинно-следственные методы.

3. Сложность системы:

- Если задача простая, то подходят методы скользящего среднего.
- Если задача сложная, то нужны алгоритмы машинного обучения.

4. Ресурсы:

- Если ресурсы ограничены, то используют качественные или простые количественные методы.
- Если ресурсов достаточно, то берут сложные алгоритмы и модели.

Все методы широко применяются в различных областях:

- **Бизнес:** прогнозирование продаж и денежных потоков.
- **Экономика:** оценка уровня инфляции и динамики валютных курсов.
- **Логистика:** оптимизация цепочек поставок и складских запасов.
- **Медицина:** прогнозирование эпидемий и медицинских ресурсов.
- **Климатология:** моделирование изменения климата и погодных условий.
- **IT:** анализ пользовательского поведения и системных сбоев.

Таблица 1.1 — Сравнительные характеристики методов прогнозирования

Метод	Данные	Сложность	Применение
Качественные	Минимальные или отсутствуют	Низкая	Стратегическое планирование, долгосрочные прогнозы
Временные ряды	Исторические данные	Средняя	Анализ продаж, прогноз спроса
Причинно-следственные	Исторические и внешние факторы	Высокая	Экономика, маркетинг
Машинное обучение	Большие объемы данных	Очень высокая	IT, аналитика
Комбинированные	Варьируются	Средняя-Высокая	Сложные системы, экономические модели

1.4 Проблемы и ограничения методов прогнозирования

Методы прогнозирования широко используются в различных областях. При этом ни один из них не лишен ограничений. Чтобы эффективно использовать какой-либо из методов, нужно изучить правила построения модели и интерпретации результата.

1.4.1 Нехватка или низкое качество данных

Большинство методов требуют большого объема данных. Однако на практике он не всегда доступен.

Одна из распространенных проблем — неполные данные или даже их отсутствие. Такое особенно часто случается в медицине и экономике.

Другая проблема — шум и выборы. Всегда есть риск совершить ошибку при сборе или вводе данных. Из-за этого могут появиться аномальные значения и исказить общую картину.

Третья проблема — нестабильность источников. Например, если ранее информация собиралась одним способом, а затем другим, то и анализировать эти данные нужно по-разному.

Все эти проблемы ухудшают качество модели и требуют предварительной подготовки данных — очистки, обработки пропущенных значений, нормализации и проверки на выбросы.

1.4.2 Переобучение (overfitting)

Одна из частых проблем в работе с моделями — переобучение. Это происходит, когда модель слишком точно подгоняется под обучающие данные и теряет способность делать качественные прогнозы на новых, еще не виденных данных. Особенно к этому склонны сложные алгоритмы, такие как нейронные сети или градиентный бустинг.

Как распознать переобучение:

- Слишком высокая точность на обучающей выборке, но при этом заметное снижение точности на тестовой;
- Излишняя сложность модели, которая не дает ощутимого прироста в качестве прогноза, но делает модель менее устойчивой к новым данным.

Методы борьбы:

- Кросс-валидация.
- Регуляризация.
- Упрощение модели.
- Увеличение объема данных.

1.4.3 Сложности интерпретации моделей

Некоторые современные методы, особенно в области машинного обучения, имеют характер "черного ящика". Например, нейросети с тысячами параметров обеспечивают высокую точность, но практически не поддаются интерпретации.

Это создает проблемы в следующих случаях:

- **Медицина и здравоохранение.** Врачи требуют объяснения, почему алгоритм вынес то или иное решение.
- **Финансы.** Регулирующие органы могут потребовать обоснований прогнозов, влияющих на ценообразование или кредитование.

Для повышения интерпретируемости используются методы вроде SHAP, LIME, Attention-механизмы, однако они увеличивают сложность анализа.

1.4.4 Ограничения качественных методов

Качественные методы — например, метод экспертных оценок или метод Дельфи — зависят от субъективного восприятия экспертов. Это влечет ряд проблем:

- **Предвзятость.** Эксперты могут неосознанно навязывать свое мнение группе.
- **Групповое мышление.** В условиях очного обсуждения может возникать конформизм, при котором участники соглашаются с доминирующим мнением.
- **Недостаточная репрезентативность.** Прогноз может не учитывать точки зрения меньшинств или внешних факторов, не замеченных экспертами.

Анонимные и многократные итерации, как в методе Дельфи, снижают эти риски, но не устраняют полностью.

1.4.5 Ошибки в условиях нестабильности и кризисов

Многие модели прогнозирования работают при допущении, что поведение системы в будущем будет похоже на прошлое. Но это предположение перестает работать, когда происходят резкие и неожиданные изменения — экономические кризисы, пандемии, военные конфликты и другие нестандартные ситуации [5, с. 67].

С какими проблемами мы сталкиваемся в таких условиях:

- **Нарушение трендов.** Данные, которые ранее давали устойчивые закономерности, теряют актуальность — например, потребительское поведение в пандемию кардинально изменилось.
- **Появление новых факторов.** Влияние переменных, которых не было в прошлом, становится значимым — как, например, карантинные меры в 2020 году.
- **Высокая неопределенность.** Чем нестабильнее ситуация, тем выше разброс возможных исходов — и тем менее надежны обычные линейные или экстраполяционные модели.

В таких случаях лучше использовать устойчивые или адаптивные модели, а также сценарный подход, где учитываются разные варианты развития событий и присваиваются вероятности.

1.4.6 Проблема устойчивости

Даже хорошо обученная модель может давать сбои. Особенно при работе с новыми данными. Или при применении в другой отрасли. Основные причины сбоев:

- Модель плохо обобщает информацию.
- Скрытые переменные.
- Изменения в распределении данных (data drift).

Для минимизации рисков регулярно переобучают модель. Постоянно мониторят ее качество. А также используют адаптивные алгоритмы. Они учитывают изменения в данных.

1.5 Оценка качества прогноза

Мало просто построить модель. Нужно оценить, насколько хорошо она работает. Точность прогнозов измеряют специальными метриками. Они помогают вычислить ошибки и сравнивать разные подходы. Неважно, какая модель используется. Это может быть простая скользящая средняя. Или сложная нейросеть. Валидация модели - обязательный этап. Модели часто проверяют на исторических данных. Например, обучают на данных до 2020 года. И проверяют, как она спрогнозировала на 2021 год.

1.5.1 Основные метрики точности прогнозов

Ниже приведены наиболее часто используемые количественные показатели, позволяющие сравнивать качество различных моделей:

- **MAE (Mean Absolute Error, средняя абсолютная ошибка)**

Показывает среднюю абсолютную разницу между предсказанными и фактическими значениями:

$$MAE = \frac{1}{n} \sum_{i=1}^n |y_i - \hat{y}_i|, \quad (1.1)$$

где

- y_i — фактическое (истинное) значение целевой переменной для i -го наблюдения,

- \hat{y}_i — предсказанное моделью значение для i -го наблюдения.

Простая и интерпретируемая метрика, устойчиво работает при наличии выбросов.

- **MAPE (Mean Absolute Percentage Error, средняя абсолютная процентная ошибка)**

Относительная ошибка, выраженная в процентах:

$$MAPE = \frac{100\%}{n} \sum_{i=1}^n \left| \frac{y_i - \hat{y}_i}{y_i} \right|, \quad (1.2)$$

где

- y_i — фактическое (истинное) значение целевой переменной для i -го наблюдения,

- \hat{y}_i — предсказанное моделью значение для i -го наблюдения.

Удобна для сравнения точности моделей на разных масштабах данных.

Однако может быть искажена, если $y_i \approx 0$.

- **RMSE (Root Mean Squared Error, корень из средней квадратичной ошибки)**

Более чувствителен к большим ошибкам:

$$RMSE = \sqrt{\frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2}, \quad (1.3)$$

где

- y_i — фактическое (истинное) значение целевой переменной для i -го наблюдения,

- \hat{y}_i — предсказанное моделью значение для i -го наблюдения.

Применяется там, где особенно важно избегать крупных промахов.

- **R^2 (Коэффициент детерминации)**

Показывает, какая доля дисперсии объясняется моделью:

$$R^2 = 1 - \frac{\sum (y_i - \hat{y}_i)^2}{\sum (y_i - \bar{y}_i)^2}. \quad (1.4)$$

где

- y_i — фактическое (истинное) значение целевой переменной для i -го наблюдения,
- \hat{y}_i — предсказанное моделью значение для i -го наблюдения,
- \bar{y}_i — среднее значение всех фактических y_i .

Принимает значения от 0 до 1. Чем ближе к 1, тем лучше модель объясняет поведение данных.

- **AIC (Akaike Information Criterion) и BIC (Bayesian Information Criterion)**

Эти критерии используются для сравнения моделей, а не для оценки точности в абсолютном выражении. Они учитывают как точность подгонки, так и сложность модели:

$$AIC = 2k - 2\ln(L), BIC = k\ln(n) - 2\ln(L), \quad (1.5)$$

где

- k — число параметров,
- L — функция правдоподобия,
- n — количество наблюдений.

Меньшее значение AIC/BIC указывает на лучшую модель.

1.5.2 Кросс-валидация

Кросс-валидация — один из основных подходов к проверке устойчивости модели. Наиболее распространенный способ — k -fold кросс-валидация, при котором данные делятся на k равных частей. Модель обучается на $k-1$ частях, а тестируется на оставшейся. Этот процесс повторяется k раз, и результат усредняется.

Преимущества:

- Обеспечивает более стабильную оценку.
- Предотвращает переобучение.
- Особенно полезна при небольшом объеме данных.

1.5.3 Разделение на тренировочную и тестовую выборки

Один из базовых, но критически важных подходов. Данные делятся на:

- Training set (обучающая выборка): используется для построения модели.
- Test set (тестовая выборка): используется для оценки ее эффективности.

Обычно в качестве тестовой части выделяется от 20% до 30% данных. Для временных рядов важно сохранять хронологический порядок: тренировочные данные должны предшествовать тестовым.

1.5.4 Бэк-тестирование (Backtesting)

Особенно важно для финансовых и экономических моделей. Оно проверяет, как будет себя вести модель, если ей будут доступны только исторические данные из прошлого. Суть заключается в следующем:

- Модель "обучается" только на исторических данных, известных на момент "тестирования".
- Затем оценивается, насколько точно модель предсказала будущие значения на прошлом (будущем относительно момента обучения). То есть нам уже известны новые данные, и мы сравниваем их с предсказанными моделью.

Бэк-тестинг позволяет оценить **реальную применимость** модели в условиях, близких к реальным.

1.5.5 Выбор метрик и подходов в зависимости от задачи

Выбор конкретной метрики зависит от характера данных и бизнес-целей, представленных в таблице 1.2.

Таблица 1.2 — Сравнительные характеристики методов прогнозирования

Ситуация	Рекомендуемые метрики
Прогнозирование продаж в денежном эквиваленте	MAE, RMSE
Оценка точности спроса (в штуках)	MAPE, MAE
Выбор между сложными моделями	AIC, BIC
Финансовые модели	R ² , Backtesting
Нестабильные временные ряды	RMSE, кросс-валидация

Оценка качества прогноза — неотъемлемый этап при построении моделей. Применение метрик MAE, MAPE, RMSE, R^2 , а также информационных критериев (AIC, BIC) позволяет объективно сравнивать модели и выявлять наиболее эффективные решения. Методы валидации, такие как кросс-валидация и бэк-тестинг, служат основой для повышения надежности и устойчивости прогнозов в реальных условиях. Без грамотной оценки модель может быть либо переобучена, либо не обладать предсказательной силой, несмотря на визуальную «хорошую» подгонку.

1.6 Основные выводы по главе

В первой главе рассмотрены основные подходы к прогнозированию, которые делятся на качественные и количественные методы. Качественные методы опираются на экспертные оценки. Их применяют в условиях нехватки данных или высокой неопределенности. Количественные методы же базируются на анализе числовой информации. Они включают в себя анализ временных рядов, причинно-следственные методы и алгоритмы машинного обучения. Также есть комбинированные подходы, они объединяют сильные стороны разных методов. Выбор метода зависит от доступности данных, целей исследования, сложности системы и доступных ресурсов.

ГЛАВА 2 ВЫБОР МЕТОДОВ ПРОГНОЗИРОВАНИЯ

Для практической части выберем несколько методов из разных категорий.

2.1 Линейная регрессия

Линейная регрессия – это один из базовых и наиболее широко применяемых методов машинного обучения, который основывается на статистическом подходе. Главная задача метода заключается в построении модели, способной описывать зависимость между входными переменными (факторами) и целевой переменной с помощью линейного уравнения [15, с. 74].

2.1.1 Основная идея линейной регрессии

Линейная регрессия предполагает, что зависимость между входными переменными x и целевой переменной y можно выразить линейным уравнением (2.1).

$$y = w_0 + w_1x_1 + w_2x_2 + \dots + w_nx_n + \epsilon, \quad (2.1)$$

где:

- y — целевая переменная (предсказываемое значение).
- x_1, x_2, \dots, x_n — объясняющие переменные (факторы или признаки).
- w_0 — свободный член (пересечение линии с осью y).
- w_1, w_2, \dots, w_n — коэффициенты (веса), которые показывают, как каждый признак влияет на y .
- ϵ — ошибка или шум (разница между реальным и предсказанным значениями).

Цель метода заключается в подборе коэффициентов (w_1, w_2, \dots, w_n), которые минимизируют расхождение между реальными и предсказанными значениями целевой переменной [3, с. 106].

2.1.2 Типы линейной регрессии

Простая линейная регрессия. Это частный случай, когда есть только один входной признак x . Определяется уравнением (2.2).

$$y = w_0 + w_1x. \quad (2.2)$$

На графике такая модель представляется в виде прямой линии. Задача состоит в том, чтобы найти ее оптимальный наклон w_1 и точку пересечения w_0 .

Множественная линейная регрессия. Если есть несколько признаков (x_1, x_2, \dots, x_n) , то модель становится многомерной и определяется уравнением (2.3).

$$y = w_0 + w_1x_1 + w_2x_2 + \dots + w_nx_n. \quad (2.3)$$

В этом случае задача состоит в нахождении гиперплоскости, которая минимизирует отклонение между предсказанными и реальными значениями [16, с. 142].

Подбор коэффициентов в линейной регрессии. Коэффициенты (w_0, w_1, \dots, w_n) подбираются так, чтобы минимизировать сумму квадратов ошибок (RSS), определяемую формулой (2.4) [17, с. 98].

$$RSS = \sum_{i=1}^m (y_i - (\hat{y}_i))^2, \quad (2.4)$$

где:

- m — количество наблюдений.
- y_i — реальное значение.
- $\hat{y}_i = w_0 + w_1x_{i1} + w_2x_{i2} + \dots + w_nx_{in}$ — предсказанное значение.

Для нахождения минимального значения RSS используются математические методы, такие как:

- **Метод градиентного спуска** — итеративный процесс поиска оптимальных коэффициентов.
- **Метод нормальных уравнений** — аналитическое решение системы уравнений.

2.1.3 Особенности линейной регрессии

Линейная регрессия предполагает линейную зависимость между признаками и целевой переменной [6, с. 312]. Ошибки модели должны быть независимыми. Дисперсия ошибок должна быть постоянной. Ошибки должны иметь нормальное распределение. Признаки не должны быть сильно коррелированы между собой.

Преимущества линейной регрессии:

Модель проста в реализации и интерпретации. Она работает быстро даже с большими данными. Коэффициенты регрессии показывают влияние каждого признака. Метод универсален [6, с. 313], он применяется в прогнозировании, анализе трендов, маркетинге и экономике.

Ограничения линейной регрессии:

Модель плохо работает с нелинейными зависимостями. Результаты чувствительны к выбросам. Качество предсказаний зависит от корректности данных. Метод не подходит для сложных нелинейных данных.

2.1.4 Примеры применения линейной регрессии

- **Экономика:** прогноз доходов населения на основе уровня образования, возраста, трудового стажа и сферы занятости. Линейная регрессия позволяет выявить, какие из этих факторов сильнее всего влияют на доход и на сколько увеличится зарплата при, например, повышении уровня образования.
- **Маркетинг:** оценка влияния рекламного бюджета, размера скидок, сезонных колебаний и цен конкурентов на **объем продаж**. Модель помогает понять, какая стратегия дает лучший отклик от покупателей, и оптимизировать бюджет.
- **Экология:** прогноз уровня загрязнения воздуха в зависимости от объемов выбросов, температуры воздуха, скорости ветра и влажности. Это позволяет определить, какие факторы больше всего способствуют загрязнению.

2.2 Модель ARIMA

Модель **ARIMA** (AutoRegressive Integrated Moving Average — авторегрессивная интегрированная скользящая средняя) широко используется

для анализа временных рядов и прогнозирования. Она объединяет три ключевых компонента:

- **AR (авторегрессия):** использует зависимости между текущим значением и его прошлыми значениями.
- **I (интеграция):** процедура для преобразования нестационарного ряда в стационарный путем разности.
- **MA (скользящая средняя):** корректирует прогнозы на основе ошибок предсказания предыдущих значений.

2.2.1 AR (Авторегрессия)

Авторегрессия предполагает, что текущее значение временного ряда зависит от нескольких его предыдущих значений (лагов) [8, с. 300]. Эта зависимость описывается уравнением, в котором используются коэффициенты для каждого лага.

Формула AR(p) определяется уравнением (2.5).

$$y = c + \phi_1 y_{t-1} + \phi_2 y_{t-2} + \dots + \phi_p y_{t-p} + \epsilon_t, \quad (2.5)$$

где:

- y — текущее значение временного ряда.
- $\phi_1, \phi_2, \dots, \phi_p$ — коэффициенты авторегрессии.
- p — количество учитываемых лагов.
- c — константа.
- ϵ_t — случайная ошибка (шум).

Если $\phi_1 > 0$, то предыдущее значение положительно влияет на текущее. Если $\phi_1 < 0$, наблюдается обратная зависимость.

2.2.2 I (Интеграция)

Интеграция используется для обработки нестационарных временных рядов. Нестационарные ряды характеризуются изменяющимися средними, дисперсией или трендами во времени [11, с. 798]. Чтобы сделать такой ряд стационарным, применяется операция разности.

Процедура интеграции определяется уравнением (2.6).

$$y' = y - y_{t-1}. \quad (2.6)$$

Если после одного дифференцирования ряд остается нестационарным, процесс повторяют по формуле (2.7).

$$y'' = y' - y'_{t-1} = y - 2y_{t-1} + y_{t-2}. \quad (2.7)$$

Параметр d указывает, сколько раз нужно выполнить эту операцию, чтобы сделать ряд стационарным.

2.2.3 МА (Скольльзящая средняя)

Модель скользящей средней основывается на анализе ошибки предсказания, допущенной в предыдущие моменты времени. Она предполагает, что текущее значение зависит от ошибки прогнозов для нескольких прошлых значений [8, с. 302].

Формула МА(q) определяется уравнением (2.8).

$$y = c + \epsilon_t + \theta_1 \epsilon_{t-1} + \theta_2 \epsilon_{t-2} + \dots + \theta_q \epsilon_{t-q}, \quad (2.8)$$

где:

- y — текущее значение временного ряда.
- $\epsilon_t, \epsilon_{t-1}, \dots, \epsilon_{t-q}$ — ошибки прогнозов.
- $\theta_1, \theta_2, \dots, \theta_q$ — коэффициенты, описывающие влияние ошибок.
- q — количество учитываемых ошибок.

2.2.4 Математическая формулировка ARIMA

Модель ARIMA обозначается как $ARIMA(p, d, q)$, где:

- p — порядок авторегрессии (число лагов в модели AR).
- d — порядок дифференцирования (число разностей для достижения стационарности).
- q — порядок скользящей средней (число лагов ошибок в модели МА).

Формула модели ARIMA представлена в уравнении (2.9).

$$y_t = c + \phi_1 y_{t-1} + \phi_2 y_{t-2} + \dots + \phi_p y_{t-p} + \epsilon_t + \theta_1 \epsilon_{t-1} + \dots + \theta_q \epsilon_{t-q}, \quad (2.9)$$

где:

- y_t — текущее значение временного ряда.
- c — константа.
- ϕ — коэффициенты авторегрессии.
- ϵ_t — ошибка в текущий момент времени.
- θ — коэффициенты скользящей средней.

2.2.5 Алгоритм работы с ARIMA

Основной алгоритм работы с ARIMA выглядит следующим образом [14, с. 519]:

1. **Анализ временного ряда:** определить, является ли ряд стационарным. Если ряд нестационарен, применить разности.
2. **Определение параметров p, d, q :** использовать графики автокорреляционной функции (ACF) и частичной автокорреляционной функции (PACF).
3. **Оценка модели:** подгонка модели ARIMA к данным с использованием методов максимального правдоподобия.
4. **Диагностика:** проверить остатки модели на соответствие белому шуму (отсутствие структуры в ошибках).
5. **Прогнозирование:** использовать модель для предсказания будущих значений временного ряда.

Ограничения модели ARIMA:

- Требование стационарности временного ряда.
- Сложности с моделированием данных с сезонностью (для этого используется SARIMA).
- Не подходит для временных рядов с сильно нелинейными зависимостями.

Расширения модели ARIMA — SARIMA и ARIMAX. SARIMA учитывает сезонность, а ARIMAX имеет дополнительные объясняющие переменные.

2.2.6 Примеры применения ARIMA

ARIMA применяется во множестве сфер. В финансовой сфере с помощью этого метода можно предсказывать поведение валютных курсов или цен на акции. Этой моделью учитываются и устойчивые тренды, и случайные колебания.

В транспортной сфере ARIMA позволяет планировать загрузку маршрутов и распределение ресурсов. Она это делает на основе данных о прошлых перевозках, например, количестве пассажиров по дням недели.

В сфере энергетики ARIMA позволяет спрогнозировать потребление электроэнергии. Она хорошо подходит для учитывания сезонности.

2.3 Метод случайных лесов

Случайный лес (Random Forest) является ансамблевым методом. Он не строит одну модель, а объединяет множество. В его основе — несколько деревьев решений, каждое из которых обучается на случайной части исходных данных. Благодаря такому подходу модель становится более устойчивой к переобучению и демонстрирует хорошие результаты на разнообразных задачах.

Дерево решений — это алгоритм, который пошагово делит данные на группы, основываясь на определенных признаках. Например, сначала по возрасту, потом по доходу, потом по другим критериям — пока не будет получено финальное предсказание. В случайном лесу таких деревьев много, и каждое «видит» только часть информации [11, с. 476].

Чтобы повысить разнообразие среди деревьев, применяется метод бэггинга (bagging). Это техника, при которой создаются подвыборки из оригинального набора данных. В них могут повторяться отдельные объекты, а какие-то — вовсе не попасть в выборку. Это помогает модели «видеть» ситуацию под разными углами [10, с. 831].

2.3.1 Основная идея метода случайных лесов

Случайный лес объединяет предсказания множества деревьев. При этом каждое дерево обучается независимо на своей случайной подвыборке. Когда нужно сделать прогноз, деревья «голосуют»: если задача — классификация, выбирается наиболее частый вариант, а если регрессия — берется среднее значение всех предсказаний [10, с. 848].

Преимущества случайного леса:

- Снижает риск переобучения по сравнению с отдельным деревом решений.
- Может работать как с задачами классификации, так и регрессии.
- Обработывает как числовые, так и категориальные данные.

2.3.2 Построение одного дерева в случайном лесу

Каждое дерево в случайном лесу строится с использованием метода решения дерева, но с дополнительным случайным выбором признаков для каждого разбиения [11, с. 435]. Процесс включает следующие этапы:

- Выбор случайного подмножества обучающих данных.
- Для каждого узла дерева случайным образом выбирается подмножество признаков для разбиения. Это означает, что не все признаки используются для принятия решения на каждом узле, что делает деревья менее зависимыми друг от друга [5, с. 132].
- Рекурсивное построение дерева до достижения определенного уровня глубины или минимального числа элементов в листе.

2.3.3 Алгоритм случайного леса

Основной алгоритм случайного леса выглядит следующим образом [13, с. 411]:

1. Создать несколько решающих деревьев на случайных подмножествах данных.
2. Для каждого дерева выбрать случайное подмножество признаков на каждом уровне разбиения.
3. На этапе предсказания каждое дерево предсказывает результат, а итоговый результат определяется путем голосования. Для задачи регрессии это будет среднее значение, для классификации — большинство голосов.

Преимущества случайных лесов:

- Высокая точность предсказаний, особенно при большом количестве деревьев.
- Устойчивость к переобучению благодаря случайности в обучении.
- Хорошо работает с неструктурированными данными, такими как изображения и текст.

Недостатки метода случайных лесов:

- Требуют значительных вычислительных ресурсов, особенно при большом количестве деревьев.

- Могут быть не так интерпретируемы, как отдельные деревья решений [7, с. 209]

2.3.4 Примеры применения случайных лесов

- **Банковское дело:** оценка **кредитного риска** клиента по множеству признаков: доход, возраст, наличие долгов, история платежей, регион проживания и т.д. Случайный лес определяет, какие из признаков наиболее важны для предсказания вероятности невозврата.
- **Здравоохранение:** прогноз риск развития заболевания на основе медицинских показателей, образа жизни, наследственности и истории болезни. Метод используется для выявления потенциальных пациентов группы риска.
- **Агробизнес:** прогноз урожайности сельхозкультур с учетом осадков, температуры, типа почвы, удобрений и плотности посева. Случайный лес помогает выявить нелинейные зависимости и взаимодействия факторов.

2.4 Метод К ближайших соседей

Метод К ближайших соседей (K-Nearest Neighbors, KNN) — это простой и эффективный алгоритм машинного обучения, основанный на поиске ближайших соседей по заданной метрике расстояния.

KNN не требует явного обучения: он сохраняет все данные и принимает решения на этапе предсказания, что делает его примером **ленивого обучения** (lazy learning) [8, с. 311].

2.4.1 Алгоритм метода k-ближайших соседей

Для нового объекта KNN выполняет следующие шаги [12, с. 386]:

1. Вычисляет расстояния до всех точек обучающей выборки.
2. Сортирует обучающие точки по возрастанию расстояния до нового объекта.
3. Выбирает K ближайших соседей (гиперпараметр метода).
4. Решает задачу:
 - Для классификации: выбирается класс, встречающийся чаще всего среди K соседей (метод голосования).
 - Для регрессии: вычисляется среднее значение целевых переменных K соседей.

2.4.2 Основные параметры метода

- **К — количество соседей:** чем меньше K , тем более детально метод будет учитывать локальные особенности данных, но тем сильнее он подвержен шуму. Увеличение K делает метод более устойчивым, но может привести к потере локальных деталей [9, с. 127].

- **Метрика расстояния:** используются различные метрики:
 - Евклидово расстояние — для данных с непрерывными значениями.
 - Манхэттенское расстояние — если данные представляют собой прямоугольную сетку.
 - Косинусное расстояние — для текстовых данных или векторов.

2.4.3 Особенности метода KNN

Преимущества метода KNN:

- Простота реализации и интерпретации.
- Подходит как для классификации, так и для регрессии.
- Не требует этапа обучения и сложной настройки.
- Возможность выбора различных метрик расстояния.

Недостатки метода KNN:

- Высокая вычислительная сложность: при большом объеме данных требуется вычисление расстояний до всех точек.
- Чувствителен к выбору гиперпараметра K .
- Требуется нормализация данных из-за влияния масштаба признаков.
- Неэффективен в задачах с высокой размерностью (проклятие размерности).

2.4.4 Примеры применения метода KNN

KNN применяется в разнообразных сферах. В рекомендательных системах он предсказывает интересы пользователя на основе поведения похожих пользователей. KNN анализирует оценки фильмов, покупки, клики, которые делают пользователи. Так он находит похожих пользователей и рекомендует им схожий контент.

В сфере образования KNN предсказывает успеваемость студента по его посещаемости, активности на платформе и результатам тестов. Так он находит похожих по характеристикам студентов и оценивает их посещаемость.

В сфере безопасности KNN обнаруживает аномалии в сетевом трафике. Он анализирует предыдущие атаки и сравнивает известные типы трафика. И так в будущем он может выявить новые потенциально опасные действия.

2.5 Основные выводы по главе

Во второй главе рассмотрено 4 метода прогнозирования. Именно они будут использованы при создании приложения. Эти методы относятся к разным направлениям в прогнозировании, поэтому пользователь на их примере сможет сравнить эти направления и понять, что ему подходит больше для его целей.

Линейная регрессия — это базовый метод, предполагающий линейную зависимость между переменными. Его довольно просто понять и использовать самостоятельно, однако он ограничен при нелинейных данных. Также в случае выбросов линейная регрессия сильно меняет свой прогноз, то есть она чувствительна к шуму и выбросам.

ARIMA — это очень мощный метод прогнозирования. Она гибкая, учитывает тренды и случайные колебания. Также она требует проверки на стационарность. Ее недостаток — ограниченность при сезонности и нелинейности.

Метод случайных лесов — это ансамблевый метод. Он точен, устойчив и универсален в применении. Но также он имеет и недостатки: высокая ресурсоемкость и сложность интерпретации.

Метод K ближайших соседей прост и универсален. Он хорошо работает при небольших объемах данных. Однако становится неэффективным при больших выборках и высокоразмерных пространствах.

Выбор именно этих четырех методов обусловлен необходимостью обеспечить разнообразие подходов как с теоретической, так и с практической стороны. Были отобраны методы, представляющие:

- **Классические статистические модели** (линейная регрессия, ARIMA) — за счет интерпретируемости и прочной математической основы;
- **Алгоритмы машинного обучения** (случайный лес, KNN) — благодаря их способности работать с нелинейными зависимостями и высокой точности прогнозов;
- **Разные типы данных** — ARIMA эффективно работает с временными рядами, тогда как остальные методы применимы к структурированным признаковым данным;

- **Разные аналитические цели** — от объяснения влияния факторов до построения точных предсказательных моделей.

Таким образом, выбранные модели дополняют друг друга и позволяют провести комплексный анализ данных с разных методологических позиций. Это обеспечивает объективность сравнения и практическую ценность полученных результатов.

ГЛАВА 3

РАЗРАБОТКА КЛИЕНТ-СЕРВЕРНОГО ПРИЛОЖЕНИЯ

Весь программный код сохранен в системе контроля версий GitHub [18]. В приложении содержатся только некоторые элементы программного кода.

3.1 Постановка цели приложения

Цель практической части дипломной работы — разработать клиент-серверное веб-приложение, которое может использоваться как для изучения, так и для практического освоения методов прогнозирования данных. Проект совмещает обучающий и прикладной аспекты. В теоретическом разделе приложения можно ознакомиться с основами методов. В практическом разделе можно ввести свои данные и применить методы на них. Можно выбрать один из методов прогнозирования либо использовать сразу все из них. Во втором случае можно будет сравнить результаты сразу всех методов. Также можно будет скачать полученный результат в виде файла.

3.2 Функционал приложения

В **интерфейс обучения** входят теоретические материалы, классификация методов и примеры прогнозирования. В этом разделе описываются преимущества, ограничения и сферы применения методов.

В **интерфейсе практической части** есть текстовое поле для ввода данных и кнопка для загрузки файла. Здесь можно выбрать параметры загрузки и считывания данных из файла. Далее пользователь указывает метод прогнозирования, и появляется окно с графиком прогноза. Здесь можно поменять представление данных и скачать его в виде файла.

3.3 Выбор инструментов

3.3.1 Клиентская часть

На стороне клиента использовался фреймворк Vue.js версии 3.5.13. Это современный фреймворк языка JavaScript. Он имеет модульную архитектуру и высокую производительность. Также он хорошо расширяется, и в нем удобно управлять состоянием приложения.

Для построения графиков использовались библиотеки Chart.js версии 4.4.7 и Vue Chart.js версии 5.3.2. Для работы с Excel — ExcelJS версии 4.4.0 и xlsx версии 0.18.5. Для создания PDF-отчетов jsPDF версии 2.5.2 и jsPDF-AutoTable версии 3.8.4. Для API-запросов Axios версии 1.7.8 и Vue Axios версии 3.5.2. Для математических формул MathJax версии 3.2.2.

3.3.2 Серверная часть

На стороне сервера использовался фреймворк Flask. Это легкий и быстрый фреймворк языка Python. Он идеально подходит для создания API и обработки CORS-запросов через Flask-CORS. Сервер обрабатывает данные пользователя, выполняет расчет прогнозов и общается с клиентом через API.

Для обработки и анализа данных использовались библиотеки NumPy и Pandas. Машинное обучение и регрессионный анализ выполнены с помощью библиотеки Scikit-learn, а временные ряды и ARIMA — с помощью Statsmodels.

3.4 Описание взаимодействия API в приложении

Клиент отправляет POST-запрос. Запрос идет на эндпоинт /forecast. В запросе передаются данные. Данные содержат три параметра. Первый параметр - исторические данные в JSON. Второй параметр - метод прогнозирования. Третий параметр - количество шагов прогноза.

Сервер получает запрос. Сервер проверяет данные. Проверяется наличие исторических данных. Проверяется указание метода. Если данные неверные - сервер возвращает ошибку. Код ошибки - 400. Ответ содержит JSON с описанием.

При успешной проверке сервер выбирает алгоритм. Алгоритм соответствует указанному методу. Сервер выполняет расчеты. Результат - массив прогнозных значений.

Если расчеты успешны - сервер возвращает ответ. Ответ содержит JSON с прогнозами. Код ответа - 200. При ошибке расчетов - код 500. Ответ содержит описание ошибки.

3.5 Интерфейс приложения

При запуске приложения пользователь видит интерфейс (рисунок 3.1).

Прогнозирование данных

Введите данные

Введите исторические данные

Выбрать файл

Количество дней: 5

Загрузить данные

Теория прогнозирования

Прогнозирование — это процесс оценки будущих значений на основе исторических данных.

Линейная регрессия

ARIMA

Случайные леса

KNN

Примеры

Метод линейной регрессии

Метод **линейной регрессии** — это один из самых простых и широко используемых методов машинного обучения, основанный на статистике. В его основе лежит идея построения **прямой линии**, которая наилучшим образом описывает связь между входными данными (факторами) и целевой переменной.

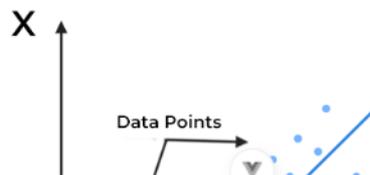


Рисунок 3.1 – Общий интерфейс приложения

Данные можно ввести двумя способами. Первый способ - текстовое поле. Второй способ - загрузка файла. Поддерживаются форматы .txt, .xls и .xlsx. При загрузке файла открывается модальное окно. Окно содержит настройки чтения данных (рисунок 3.2). Здесь пользователь может настроить параметры импорта.

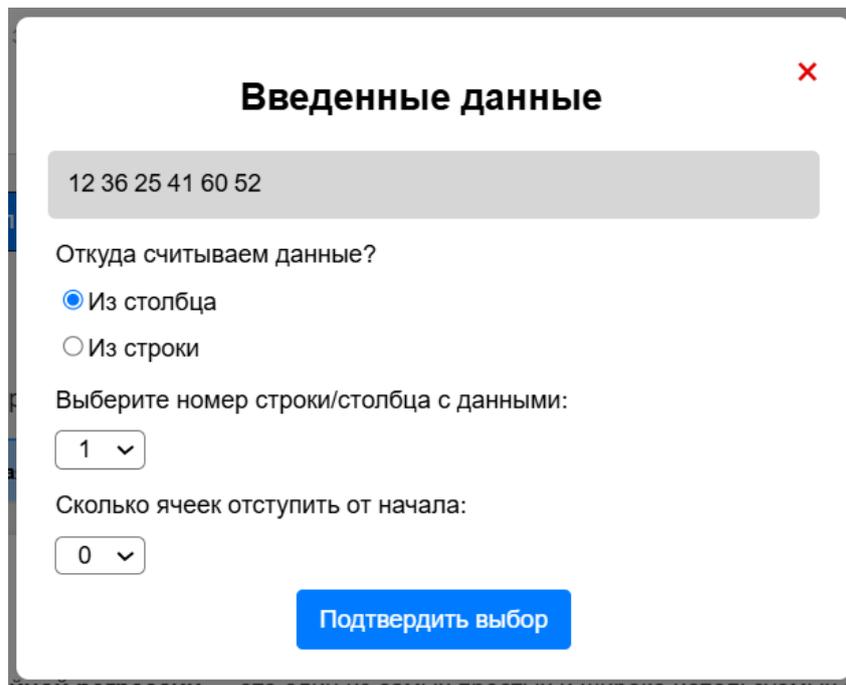


Рисунок 3.2 – Модальное окно с настройками чтения данных из файла

Пользователь подтверждает выбор. Система начинает загрузку данных (ПРИЛОЖЕНИЕ А). Данные отправляются на сервер. Сервер обрабатывает полученную информацию.

После обработки сервер возвращает готовый прогноз. В интерфейсе отображаются результаты (ПРИЛОЖЕНИЕ Б). Результаты показаны на рисунке 3.3. Блок содержит все расчетные значения. Пользователь может анализировать полученные данные.

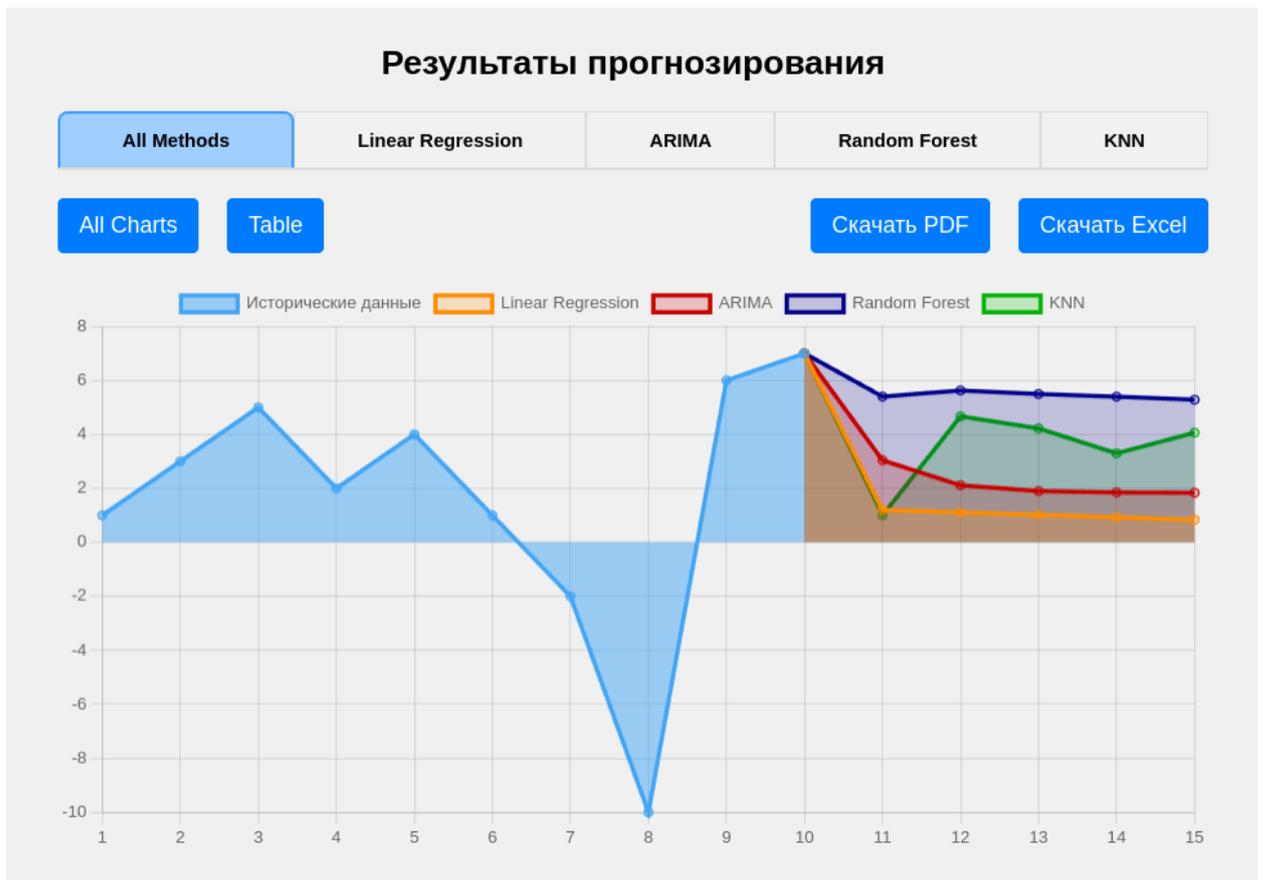


Рисунок 3.3 – Блок с результатами прогнозирования

Данные отображаются линейным графиком (Приложение В). Можно переключиться на столбчатую диаграмму (рис. 3.4) или таблицу (рис. 3.5).

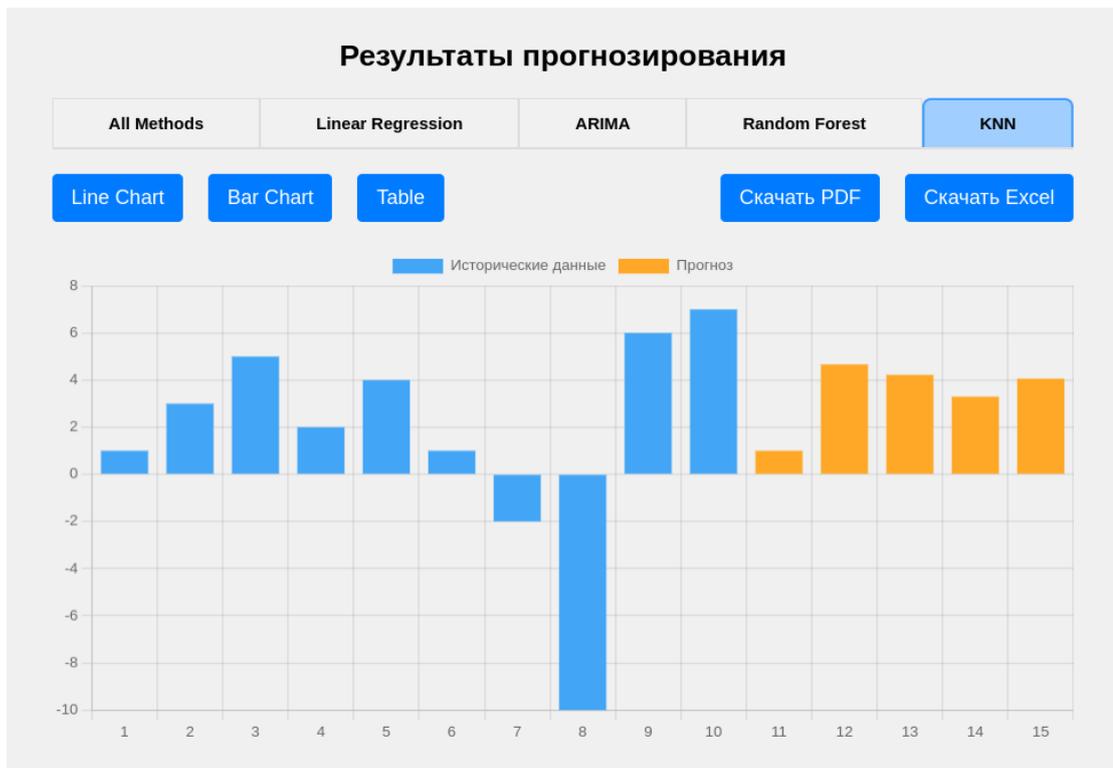


Рисунок 3.4 – Представление данных в виде столбчатой диаграммы

№	Данные	ARIMA	KNN	Linear Regression	Random Forest
1	1				
2	3				
3	5				
4	2				
5	4				
6	1				
7	-2				
8	-10				
9	6				
10	7				
11		3.0382999	1	1.2	5.4
12		2.1135897	4.6666667	1.1090909	5.626
13		1.8977508	4.2222222	1.0181818	5.49524
14		1.8473713	3.2962963	0.9272727	5.3965708
15		1.8356121	4.0617284	0.8363636	5.2833667

Рисунок 3.5 – Представление данных в виде таблицы

Если сменить метод, то график обновится автоматически (рис. 3.6).

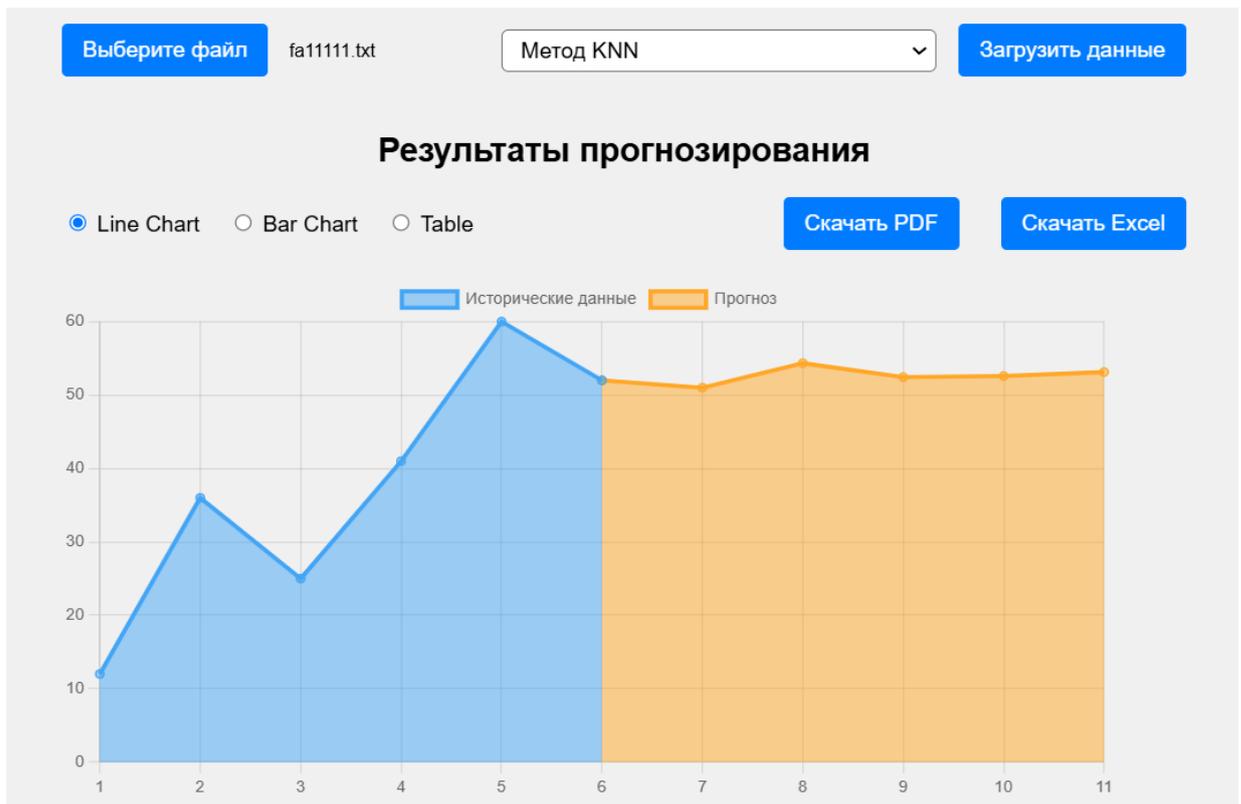


Рисунок 3.6 – Измененный график

Результаты доступны для скачивания в PDF или Excel (рис. 3.7).



Рисунок 3.7 – Кнопки для скачивания результатов

Приложение включает учебные материалы. Для каждого метода есть теория. Добавлены наглядные примеры. Присутствуют графики и диаграммы. См. рисунки 3.8 и 3.9.

Теория прогнозирования

Прогнозирование — это процесс оценки будущих значений на основе исторических данных.

Линейная регрессия

ARIMA

Случайные леса

KNN

Примеры

Метод линейной регрессии

Метод **линейной регрессии** — это один из самых простых и широко используемых методов машинного обучения, основанный на статистике. В его основе лежит идея построения **прямой линии**, которая наилучшим образом описывает связь между входными данными (факторами) и целевой переменной.

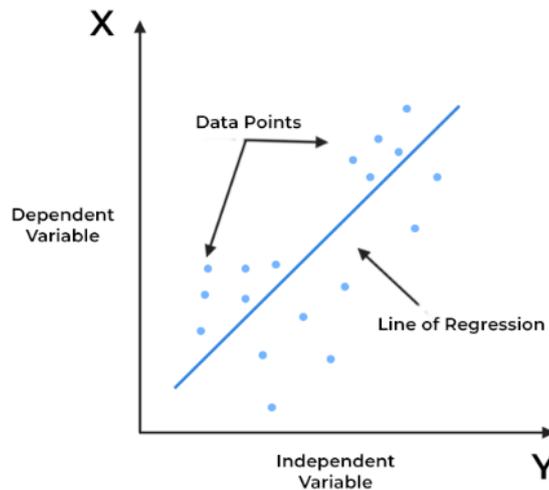


Рисунок 3.8 – Блок с теоретическими материалами и примерами

Пример работы KNN (на интуитивном уровне)

Классификация:

Есть обучающая выборка:

- (2, 3, "Класс А"),
- (5, 4, "Класс В"),
- (1, 2, "Класс А"),
- (6, 5, "Класс В"),

Получаем новый объект: (3, 3)

Находим расстояния до всех точек:

- До (2, 3): $(3 - 2)^2 + (3 - 3)^2 = 1$
- До (5, 4): $(5 - 3)^2 + (4 - 3)^2 = 2.24$
- До (1, 2): $(3 - 1)^2 + (3 - 2)^2 = 2.24$
- До (6, 5): $(6 - 3)^2 + (5 - 3)^2 = 3.61$

Сортируем: (2, 3), (5, 4), (1, 2), (6, 5).

Берем $K = 3$: ближайшие соседи — (2, 3, "Класс А"), (5, 4, "Класс В"), (1, 2, "Класс А").

Голосуем:

- Класс А: 2
- Класс В: 1

Ответ: "Класс А".

Рисунок 3.9 – Пример работы KNN из теоретического блока

3.6 Тестирование приложения

Для обеспечения корректной работы клиентской части приложения было проведено модульное тестирование с использованием инструментов Vitest и @vue/test-utils. Тесты позволяют проверить правильную работу визуальных компонентов, корректность отображения данных, правильное поведение при взаимодействии пользователя и соответствие визуализации прогнозным данным.

Цели тестирования:

- Проверка корректного отображения интерфейсных компонентов.
- Верификация правильности расчетов и визуализации.

- Проверка реактивности компонентов при изменении props.
- Проверка логики экспорта данных в PDF и Excel.
- Обеспечение отказоустойчивости при вводе данных.

Используемые инструменты представлены в таблице 3.1.

Таблица 3.1 — Инструменты тестирования приложения

Инструмент	Назначение
Vitest	Фреймворк для запуска тестов.
@vue/test-utils	Инструмент для монтирования и тестирования Vue-компонентов.
jsdom	Эмуляция DOM в среде Node.js для тестирования компонентов.
jsPDF / autotable	Экспорт таблиц в PDF в тестах.
xlsx	Экспорт данных в Excel.

Тестирование компонентов:

AllCharts.vue:

- Рендеринг линейного графика
- Корректность построения исторических и прогнозных данных
- Порядок методов прогнозирования
- Цвета для различных методов
- Обновление данных при изменении входных данных

AllTable.vue:

- Отображение исторических и прогнозных данных в виде таблицы
- Заголовки таблицы
- Работа кнопок для экспорта в PDF и Excel
- Генерация и сохранение файлов

BarChart.vue:

- Отображение столбчатой диаграммы

- Правильность отображения исторических и прогнозных данных
- Цвета и структура данных

TableChart.vue:

- Отображение таблицы с историческими и прогнозными данными
- Экспорт в Excel
- Генерация данных для экспорта

TableModal.vue:

- Отображение/скрытие модального окна
- Работа с формой выбора настроек
- Передача выбранных значений через событие confirm

Tabs.vue:

- Отображение всех вкладок
- Смена активной вкладки при клике
- Событие обновления текущей вкладки

TheorySources.vue:

- Отображение заголовка и описания
- Рендеринг компонента "Линейная регрессия" по умолчанию
- Интеграция с табами
- Отображение контента в зависимости от активной вкладки

Процесс тестирования:

Протестирована клиентская часть приложения. Выполнен ряд проверок. Подтверждена корректная работа всех компонентов.

Каждый компонент проверен отдельно. Выявлены возможные ошибки. Проведено сравнение с ожидаемым поведением.

Протестировано отображение компонентов. Проверена корректность данных. Подтверждена реактивность интерфейса.

Проверена логика экспорта данных. Протестирована обработка пользовательского ввода. Выполнено интеграционное тестирование.

Результаты тестирования:

Серьезные ошибки не обнаружены. Выявлены незначительные проблемы с отображением. Все недочеты устранены в процессе тестирования. Улучшено отображение графиков. Оптимизирован показ таблиц. Настроена реакция на изменение данных. Обеспечена стабильная работа приложения. Достигнуто повышение удобства интерфейса. Все компоненты успешно прошли тестирование.

3.7 Сравнение с аналогичными решениями

В процессе разработки приложения был проведен анализ существующих решений в области визуализации и применения методов прогнозирования. Рассмотрим наиболее близкие по функционалу и назначению системы и определим, чем отличается и превосходит их разработанное приложение.

3.7.1 Обзор аналогичных технологий и приложений

Microsoft Excel с надстройками (например, Analysis ToolPak, Forecast Sheet). Excel предоставляет базовые функции анализа временных рядов и построения прогнозов. Пользователь может применять экспоненциальное сглаживание, линейную регрессию и создавать графики. Однако:

- интерфейс перегружен и ориентирован на опытных пользователей;
- отсутствует интеграция с современными алгоритмами машинного обучения;
- нет возможности легко сравнивать различные методы прогнозирования.

IBM SPSS Forecasting. Мощный инструмент для анализа данных и прогнозирования, включающий ARIMA, ETS и другие статистические модели. Предназначен преимущественно для корпоративных пользователей:

- высокая стоимость лицензии;
- интерфейс перегружен профессиональной терминологией;
- требует предварительной подготовки данных и навыков в статистике.

Google Sheets + плагины (AutoML, BigQuery ML). Позволяет подключать внешние модели для анализа данных через плагины. Несмотря на гибкость:

- настройка требует технических знаний;
- сложная интеграция пользовательских моделей;
- слабая визуализация и отсутствие образовательной составляющей.

Образовательные платформы (Kaggle, DataCamp). Платформы предоставляют теоретическую информацию и интерактивные ноутбуки для обучения. Однако:

- нет полноценного пользовательского интерфейса для работы с собственными данными;
- визуализация ограничена средой Jupyter;

- отсутствует автоматизация построения прогнозов на стороне пользователя.

3.7.2 Отличительные особенности и преимущества разработанного приложения

По сравнению с перечисленными решениями, разработанное приложение обладает следующими преимуществами:

- **Интеграция обучения и практики:** Помимо практической работы с методами прогнозирования, пользователю предоставляется теоретический материал, что делает приложение подходящим для образовательных целей.
- **Универсальность:** Поддержка как классических методов (скользящее среднее, линейная регрессия), так и методов машинного обучения (деревья решений и др.), расширяет круг задач, которые можно решать с помощью приложения.
- **Интерактивность и визуализация:** Гибкая и интуитивно понятная визуализация результатов в виде графиков и таблиц делает анализ данных наглядным даже для пользователей без подготовки в области статистики.
- **Простота использования:** Пользовательский интерфейс построен с акцентом на доступность, а все сложные вычисления выполняются на серверной части, что снижает нагрузку на клиента.
- **Возможность сравнения методов:** Приложение позволяет применять к одним и тем же данным разные алгоритмы и сравнивать результаты визуально, чего нет в большинстве аналогов.
- **Поддержка экспорта данных:** Результаты прогнозов можно сохранить в удобных форматах (PDF, Excel), что делает приложение полезным в практической и прикладной деятельности.

3.8 Основные выводы по главе

В третьей главе описана разработка клиент-серверного приложения, цель которого — предоставить пользователю удобную платформу для обучения и практического применения методов прогнозирования данных. Приложение совмещает образовательный и аналитический функционал: оно позволяет ознакомиться с теоретической базой, загрузить собственные

данные, выбрать метод прогнозирования, получить и визуализировать результаты, а также сохранить их в различных форматах.

Функциональность реализована на современном технологическом стеке: клиентская часть построена на Vue.js с использованием специализированных библиотек для визуализации и экспорта данных, серверная — на Python с Flask, обеспечивающим обработку запросов и выполнение математических расчетов. API взаимодействие между клиентом и сервером организовано логично и прозрачно, с предусмотренной валидацией входных данных и обработкой ошибок.

Таким образом, приложение представляет собой не только удобный учебный инструмент, но и полноценную аналитическую платформу, адаптированную для пользователей с разным уровнем подготовки, способную эффективно демонстрировать работу различных методов прогнозирования на практике.

ЗАКЛЮЧЕНИЕ

В процессе работы над дипломной работой стояла задача создать клиент-серверное приложение, с помощью которого можно было бы изучать и применять на практике различные методы прогнозирования данных. Для этого следовало последовательно решить несколько ключевых задач:

1. Изучить и обобщить существующие подходы к прогнозированию: методы анализа временных рядов, причинно-следственные модели и алгоритмы машинного обучения.

2. Разработать структуру приложения, клиентскую и серверную части.

3. Реализовать серверную часть на Python с использованием Flask — она отвечает за обработку пользовательских запросов, выполнение алгоритмов прогнозирования и передачу результатов на клиент.

4. Создать клиентскую часть на основе Vue.js — фреймворк позволил реализовать удобный, интерактивный интерфейс.

Разработанное клиент-серверное приложение успешно решает поставленные задачи:

1. **Обучение и предоставление теоретической информации.** В приложении есть материал для изучения методов прогнозирования, примеров их использования и ограничений.

2. **Практическое применение методов прогнозирования.** В приложении можно загрузить свои данные, выбрать метод и построить прогноз.

3. **Визуализация и сохранение результатов.** Результаты прогнозирования отображаются в виде график разного формата и таблиц. Их можно скачать в формате Excel или PDF.

4. **Сравнение методов прогнозирования.** Пользователи могут легко сравнить прогнозы, полученные с помощью разных методов. А затем выбрать наиболее подходящий метод для своей задачи.

Главное достоинство приложения — универсальность. Его можно использовать и для образовательных, и для прикладных задач.

Разработанное приложение готово к применению на практике. Его можно использовать в образовательных целях и для анализа бизнес-данных. А также в исследованиях, связанных с прогнозированием.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Бокс, Дж. Э. П., Дженкинс, Г. М. Анализ временных рядов. Прогноз и управление – М.: Вильямс, 2016. – 912 с.
2. Бурков, А. Машинное обучение: подходы и алгоритмы – М.: Эксмо, 2020. – 512 с.
3. Введение в анализ временных рядов: учебное пособие для вузов / Артамонов Н.В. [и др.]; Московский государственный университет имени М.В. Ломоносова, Московская школа экономики, Кафедра эконометрики и математических методов экономики. – Вологда: ВолНЦ РАН, 2021. – 134 с.
4. Выгодчикова И.Ю. Алгоритм оценки параметров линейной множественной модели регрессии // Прикладная информатика. – 2015. – № 4 (58). – С. 105–116.
5. Гарретт, Г. Введение в машинное обучение с Python: практическое руководство по scikit-learn и TensorFlow – М.: БХВ-Петербург, 2019. – 480 с.
6. Гудфеллоу И., Бенджио Й., Курвил А. Глубокое обучение. – М.: Машиностроение, 2018.
7. Келлехер, Дж. Д., МакНэми, Б., Д’Арси, А. Основы машинного обучения для аналитического прогнозирования. – М.: Инфра-М, 2020. – 296 с.
8. Кузнецов, В. А. Основы регрессионного анализа: учебное пособие – М.: Инфра-М, 2018. – 256 с.
9. Лапин В. Прогнозирование и моделирование временных рядов. – Москва, 2020.
10. Маслов, А. В. Основы теории и практики машинного обучения. – М.: Научный мир, 2021. – 300 с.
11. Мерфи, К. Вероятностное машинное обучение. – М.: Вильямс, 2012. – 1104 с.
12. Растринин Л.А., Пономарев Ю.П. Экстраполяционные методы проектирования и управления. – М.: Машиностроение, 1986. – 120 с.
13. Рашка С., Мирджалили В. Python и машинное обучение: машинное и глубокое обучение с использованием Python, scikit-learn и TensorFlow 2 / под ред. С.Н. Тригуб. – СПб.: ООО "Диалектика", 2020. – 848 с.
14. Уилмот Д. Практическое прогнозирование с использованием Python. – 2019
15. Хайндман, Р., Афанасопулос, Г. Прогнозирование: принципы и практика. – OTexts, 2021. – 380 с.
16. Харитонов, А. В. Модели и методы анализа временных рядов – М.: Физматлит, 2017. – 312 с.

17. Шолле, Ф. Глубокое обучение на Python. – М.: ДМК Пресс, 201–384 с.
18. Forecast App [Электронный ресурс]. – Режим доступа: <https://github.com/UlyanaFilkova/forecast-app> – Дата доступа: 12.05.2025.

ПРИЛОЖЕНИЕ А

Загрузка и валидация входных данных

Данный программный код предназначен для загрузки данных из текстового файла или Excel-документа, валидации и предварительной обработки данных.

```
const handleFileUpload = (event) => {
  dataInput.value = ""
  file.value = null
  fileError.value = ""
  numberSelected.value = 1
  skipCells.value = 0
  readingDirection.value = ""
  const selectedFile = event.target.files[0]
  file.value = selectedFile
  v$.value.$touch()
  if (v$.invalid) {
    if (v$.required.invalid) {
      fileError.value = 'Загрузите файл или введите данные в текстовое
поле'
    } else if (v$.isFileType.invalid) {
      fileError.value = 'Неверный формат файла. Допустимые форматы:
.txt, .xlsx, .xls'
    } else if (v$.maxSize.invalid) {
      fileError.value = `Формат файла должен быть меньше
${MAX_FILE_SIZE_IN_BYTES / 1024 / 1024} MB.`
    }
  } else {
    fileError.value = ""
  }
}
```

```

if (selectedFile) {
  const fileExtension = selectedFile.name.split('.').pop().toLowerCase()
  if (fileExtension === 'txt') {
    readTextFile(selectedFile)
  } else if (fileExtension === 'xls' || fileExtension === 'xlsx') {
    readExcelFile(selectedFile)
  } else {
    fileError.value = 'Неверный формат файла. Допустимые форматы:
.txt, .xlsx, .xls'
  }
}
}
}
}

const readTextFile = (file) => {
  const reader = new FileReader()
  reader.onload = (e) => {
    const content = e.target.result.trim().replace(/\r/g, " ")
    const numberRegex = /^-?\d+(\.\d+)?(\s-?\d+(\.\d+)?)*$/
    if (!numberRegex.test(content)) {
      fileError.value = 'Файл должен содержать только числа, разделенные
пробелами'
      return
    }
    dataLines.value = [content.split(/\s+/).map(Number)]
    dataInput.value = content
  }
  reader.readAsText(file)
}

```

```

showModal.value = true
}
const readExcelFile = (file) => {
  const reader = new FileReader()
  reader.onload = async (e) => {
    const data = new Uint8Array(e.target.result)
    const workbook = new ExcelJS.Workbook()
    await workbook.xlsx.load(data)
    const worksheet = workbook.worksheets[0]
    dataLines.value = []
    let isValid = true
    worksheet.eachRow((row) => {
      const rowData = row.values.slice(1).map((value) => {
        if (typeof value !== 'number') {
          isValid = false
        }
        return value
      })
      dataLines.value.push(rowData)
    })
    if (!isValid) {
      fileError.value = 'Файл должен содержать только числа'
      return
    }
    dataInput.value = dataLines.value.map((row) => row.join(' ')).join('\n')
  }
  reader.readAsArrayBuffer(file)
}

```

```
showModal.value = true  
}
```

Отображение результатов прогнозирования

Данный программный код реализует интерфейс для отображения результатов прогнозирования с возможностью выбора метода прогноза (линейная регрессия, ARIMA, Random Forest и др.) и типа визуализации (линейный график, столбчатая диаграмма, таблица).

```
const store = useStore()

const forecastMethods = ref(['All Methods', 'Linear Regression', 'ARIMA',
'Random Forest', 'KNN'])

const selectedMethod = computed(() =>
forecastMethods.value[currentTab.value])

const currentTab = ref(0)

let chartTypes = [
  { label: 'All Charts', value: 'all' },
  { label: 'Line Chart', value: 'line' },
  { label: 'Bar Chart', value: 'bar' },
  { label: 'Table', value: 'table' },
]

const chartType = ref('all')

const updateChartType = () => {
  if (selectedMethod.value === 'All Methods') {
    chartType.value = 'all'
    chartTypes = [
      { label: 'All Charts', value: 'all' },
      { label: 'Table', value: 'allTable' },
    ]
  } else {
    chartType.value = 'line'
```

```

chartTypes = [
  { label: 'Line Chart', value: 'line' },
  { label: 'Bar Chart', value: 'bar' },
  { label: 'Table', value: 'table' },
]
}
}
watch(currentTab, updateChartType, {immediate: true})
const chartConfig = ref({
  responsive: true,
  scales: {
    x: { stacked: true, barPercentage: 0.5, categoryPercentage: 0.5 },
    y: { beginAtZero: true },
  },
})
const historicalData = computed(() => store.inputData)
const forecastData = computed(() => store.chartData)
const filteredForecastData = computed(() => {
  if (selectedMethod.value === 'All Methods') {
    return forecastData.value
  }
  return forecastData.value[selectedMethod.value] || []
})
const currentChartType = computed(() => {
  const chartMap = {
    line: LineChart,
    table: TableChart,

```

```

    all: AllCharts,
    allTable: AllTable,
    bar: BarChart,
  }
  return chartMap[chartType.value] || AllCharts
})
const getTableData = () => {
  const rows = []
  let forecast = filteredForecastData.value
  let methods = []
  if (Array.isArray(forecast)) {
    // If the forecast is an array (one method), convert it to an object
    methods = ['Forecast']
    forecast = { Forecast: forecast }
  } else if (typeof forecast === 'object' && forecast !== null) {
    // If forecast is an object with methods
    methods = Object.keys(forecast)
  } else {
    console.error("Unexpected forecast data structure:", forecast)
    return rows
  }
  rows.push(['', 'Data', ...methods])
  const historyLength = historicalData.value.length
  for (let i = 0; i < historyLength; i++) {
    const row = [`${i + 1}`, historicalData.value[i], ...methods.map(() => "")]
    rows.push(row)
  }
}

```

```
    const maxForecastLength = Math.max(...methods.map(m => (forecast[m] ?
forecast[m].length : 0)))
    for (let i = 0; i < maxForecastLength; i++) {
        const row = [`${historyLength + i + 1}`, "...methods.map(method =>
forecast[method]?.[i] ?? '-')]
        rows.push(row)
    }
    return rows
}
```

ПРИЛОЖЕНИЕ В

Отображение линейной диаграммы с результатами всех методов

Данный программный код предназначен для визуализации результатов прогнозирования в виде линейной диаграммы с автоматической генерацией подписей осей и назначением уникального цвета для каждого графика.

```
ChartJS.register(Title, Tooltip, Legend, LineElement, PointElement,  
CategoryScale, LinearScale, Filler);
```

```
const props = defineProps({  
  historicalData: {  
    type: Array,  
    required: true,  
  },  
  forecastData: {  
    type: Object,  
    required: true,  
  },  
  chartOptions: {  
    type: Object,  
    required: true,  
  },  
});  
  
const chartData = computed(() => {  
  const historicalLabels = props.historicalData.map((_, index) => `${index +  
1}`);  
  const datasets = [  
    {  
      label: 'Исторические данные',
```

```

    backgroundColor: 'rgba(66, 165, 245, 0.5)',
    borderColor: '#42A5F5',
    fill: true,
    data: props.historicalData,
    yAxisID: 'y',
  }
];

const methodOrder = ['Linear Regression', 'ARIMA', 'Random Forest',
'KNN'];

const sortedMethods = methodOrder.filter((method) => method in
props.forecastData);

const colors = {
  'Linear Regression': { backgroundColor: 'rgba(255, 140, 0, 0.2)',
borderColor: 'rgba(255, 140, 0, 1)' },
  'ARIMA': { backgroundColor: 'rgba(200, 0, 0, 0.2)', borderColor:
'rgba(200, 0, 0, 1)' },
  'Random Forest': { backgroundColor: 'rgba(0, 0, 139, 0.2)', borderColor:
'rgba(0, 0, 139, 1)' },
  'KNN': { backgroundColor: 'rgba(0, 180, 0, 0.2)', borderColor: 'rgba(0,
180, 0, 1)' }
};

sortedMethods.forEach((method) => {
  const color = colors[method] || { backgroundColor: 'rgba(200, 200, 200,
0.2)', borderColor: 'rgba(200, 200, 200, 1)' };
  datasets.push({
    label: method,
    backgroundColor: color.backgroundColor,
    borderColor: color.borderColor,
    fill: true,

```

```
    data: [...Array(props.historicalData.length - 1).fill(null),
props.historicalData.at(-1), ...props.forecastData[method]],
    yAxisID: 'y',
  });
});
return {
  labels: [...historicalLabels, ...forecastLabels],
  datasets,
};
});
```