

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

Кафедра: Прикладная математика и информатика
(наименование)

09.03.03 Прикладная информатика

(код и наименование направления подготовки, специальности)

Бизнес-информатика

(направленность (профиль)/специализация)

**ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА
(БАКАЛАВРСКАЯ РАБОТА)**

на тему Проектирование информационной системы анализа и
прогнозирования потребительского спроса на продукцию компании

Студент

А.В. Чайко

(И.О. Фамилия)

(личная подпись)

Руководитель

к.э.н., доцент Т.А. Раченко

(ученая степень, звание, И.О. Фамилия)

Тольятти 2024

Аннотация

Тема бакалаврской работы - «Проектирование информационной системы анализа и прогнозирования потребительского спроса на продукцию компании». Актуальность исследования обусловлена высокой конкуренцией на рынке и необходимостью компаний быстро и точно прогнозировать изменения спроса для поддержания устойчивого положения. Внедрение системы анализа данных позволяет организации оперативно адаптироваться к колебаниям рынка, что делает данное исследование востребованным.

Целью данной работы является разработка приложения для анализа и прогнозирования спроса на товары. Для её достижения были поставлены следующие задачи: анализ бизнес-процессов компании и определение ключевых требований к системе; выбор и внедрение подходящих методов анализа данных; разработка структуры системы и базы данных; создание и тестирование информационной системы для анализа потребительского спроса на продукцию компании; также тестирование готового продукта и оценка его эффективности.

Практическая значимость работы заключается в создании программного решения, которое может быть интегрировано в ИТ-инфраструктуру компании, предоставляя возможность точного и своевременного анализа данных и прогнозирования потребительского спроса. Это улучшает процесс принятия управленческих решений и повышает конкурентоспособность компании.

Работа состоит из введения, трёх глав, заключения и списка литературы, включающих 77 страниц печатного текста, 45 иллюстраций, 6 таблиц и 30 источников.

Оглавление

Введение.....	5
Глава 1 Функциональное моделирование предметной области	8
1.1 Характеристика ИП «Gujurly Zahmet»	8
1.2 Теоретические основы анализа и прогнозирования потребительского спроса	10
1.3 Разбор методов аналитики, используемых в компании и подбор технологии для концептуального моделирования	14
1.4 Определение и формирование требований к системе	24
Глава 2 Логическое проектирование системы анализа и прогнозирования потребительского спроса	29
2.1 Описание инструментов для анализа и прогнозирования на основе машинного обучения	29
2.2 Проектирование статической структуры приложения	34
2.3 Выбор системы управления базами данных (СУБД)	47
2.4 Логическая и физическая модели базы данных	53
Глава 3 Физическое проектирование и реализация приложения	62
3.1 Разработка информационной системы анализа и прогнозирования потребительского спроса на продукцию компании	62
3.2 Архитектура системы, методы анализа и прогнозирования	72
3.3 Алгоритм действия пользователей в системе анализа и прогнозирования потребительского спроса на продукцию компании	76
3.4 Разработка административной части системы анализа и прогнозирования потребительского спроса на продукцию компании	81
3.5 Разработка пользовательской части системы анализа и прогнозирования потребительского спроса на продукцию компании	89
3.6 Тестирование и отладка приложения	102
3.7 Оценка экономической эффективности проекта	105

Заключение	108
Список используемой литературы	110

Введение

Развитие информационных технологий сегодня открывает широкие возможности для компаний, стремящихся повысить конкурентоспособность за счет внедрения методов анализа данных и машинного обучения. С увеличением объема данных и ростом требований к скорости и точности принятия управленческих решений, использование современных аналитических инструментов становится необходимым для поддержания устойчивого положения на рынке. В условиях высокой конкуренции компании всё чаще обращаются к продвинутым технологиям для оптимизации бизнес-процессов, улучшения качества обслуживания клиентов и более эффективного прогнозирования спроса [19].

Одной из таких компаний является «Gujurly Zahmet», специализирующееся на предоставлении и обслуживании систем отопления, вентиляции, противопожарной защиты, водоснабжения и управления зданиями. Компания реализует крупные инфраструктурные проекты, включая правительственные контракты, и проводит предварительные геологические исследования для оптимальной прокладки коммуникаций.

С учетом динамики современных рыночных условий и необходимости быстрой адаптации к изменяющимся требованиям клиентов, внедрение методов машинного обучения в информационные системы компании становится не только актуальной, но и критически важной задачей. Современные методы машинного обучения и анализа данных позволяют выявлять скрытые закономерности в потребительских предпочтениях и прогнозировать спрос на продукцию, что способствует оптимизации ресурсопотребления и повышению удовлетворенности клиентов. Эти задачи делают разработку программного обеспечения для анализа и прогнозирования потребительского спроса как научно значимой, так и практически ценной.

Целью данной работы является разработка приложения для анализа и прогнозирования спроса на продукцию компании «Gujurly Zahmet».

Приложение должно предоставить компании инструмент для быстрого и точного анализа данных, что повысит ее способность адаптироваться к рыночным изменениям и обеспечит конкурентные преимущества на рынке.

Для достижения цели необходимо решить следующие задачи:

- изучить и охарактеризовать структуру компании «Gujurly Zahmet»
- провести анализ существующих методов анализа и прогнозирования потребительского спроса и определить подходящие для реализации;
- разработать прототип информационной системы анализа и прогнозирования потребительского спроса на продукцию компании, включая основные функции анализа данных;
- выбрать подходящие технологии и методы машинного обучения для реализации поставленной задачи, в том числе нейронные сети и алгоритмы обработки больших данных;
- разработать логическую модель и архитектуру приложения, способную обеспечить надежный и быстрый анализ данных;
- создать и протестировать информационную систему анализа и прогнозирования потребительского спроса на продукцию компании, оценить её точность и производительность в реальных условиях эксплуатации;
- провести оценку эффективности и определить возможности дальнейшего развития приложения для обеспечения более точного прогнозирования и удовлетворения потребностей клиентов.

Научная значимость работы заключается в разработке интегрированного подхода к анализу и прогнозированию потребительского спроса с использованием методов глубокого обучения и инструментов обработки больших данных. В работе исследуются возможности автоматизации бизнес-процессов на основе анализа данных и моделирования поведения потребителей [30].

Практическая значимость проекта состоит в создании приложения,

которое будет способствовать оптимизации принятия управленческих решений, ускорению бизнес-процессов и улучшению качества обслуживания клиентов. Приложение обеспечит компании «Gujurly Zahmet» возможность оперативного реагирования на изменения рыночной конъюнктуры и позволит ей занять более устойчивые позиции на конкурентном рынке.

Методы исследования, применяемые в рамках работы, включают моделирование бизнес-процессов, машинное обучение для анализа и прогнозирования и инструменты для обработки информации. Предполагается использование нейронных сетей и алгоритмов глубокого обучения для создания модели прогнозирования спроса на продукцию компании.

Объект исследования – алгоритмы и методы анализа данных, а также технологии и инструменты прогнозирования потребительского спроса на продукцию компании.

Предмет исследования – анализ и прогнозирование потребительского спроса на продукцию компании.

Цель исследования – разработка информационной системы анализа и прогнозирования потребительского спроса на продукцию компании.

Структура работы состоит из введения, трёх основных глав, заключения и списка использованной литературы. Первая глава посвящена анализу бизнес-процессов компании и требований к приложению. Вторая глава описывает логическое проектирование и выбор технологий для реализации проекта. Третья глава рассматривает процесс разработки, внедрения и тестирования приложения, оценивает его соответствие требованиям и выявляет перспективы для дальнейшего развития. В заключении представлены выводы по результатам работы, анализ эффективности разработанного приложения и рекомендации.

Глава 1 Функциональное моделирование предметной области

1.1 Характеристика ИП «Gujurly Zahmet»

Индивидуальное предприятие «Gujurly Zahmet» специализируется на поставке и обслуживании различных инженерных систем, включая отопление, вентиляцию, противопожарную защиту, водоснабжение и управление зданиями. Компания предоставляет широкий спектр услуг, таких как установка газовых, электрических и твердотопливных котлов, а также монтаж радиаторных систем и систем теплого пола. В области вентиляции «Gujurly Zahmet» предлагает решения как для приточных, так и вытяжных систем, обеспечивая оптимальный воздухообмен в помещениях.

Особое внимание уделяется системам противопожарной защиты, которые включают автоматические системы сигнализации и пожаротушения для различных типов зданий. Кроме того, компания предлагает услуги по установке водоснабжения, канализации и насосных станций, что обеспечивает надежное водоснабжение объектов. Важной частью деятельности является управление зданиями, включая техническое обслуживание, ремонт и управление системами безопасности.

«Геологические исследования» играют ключевую роль в проектировании, позволяя определить оптимальные места для прокладки водо- и газоснабжения, что существенно повышает эффективность эксплуатации систем. Предприятие также успешно реализует крупные проекты, включая правительственные контракты, демонстрируя высокие стандарты качества.

Среди новых направлений «Gujurly Zahmet» выделяется проектирование и внедрение Building Management Systems (BMS) - централизованных систем управления, которые интегрируют различные инженерные системы в единую платформу для мониторинга и управления. Это требует значительных знаний в области информационных технологий и привело к модификации структуры

компании. Организационная структура предприятия показана на рисунке 1.

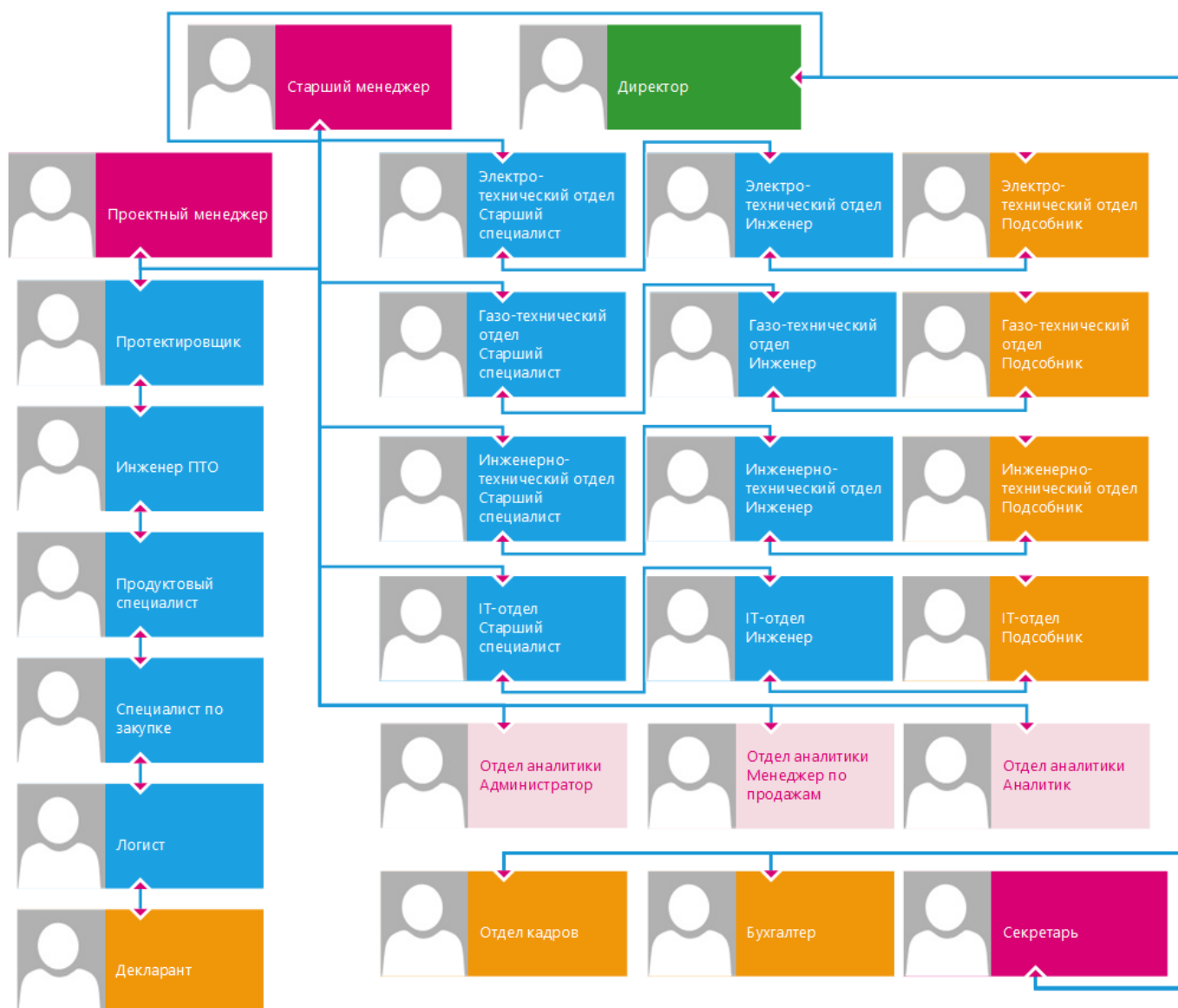


Рисунок 1 – Организационная структура ИП «Gujurly Zahmet»

В ближайшие годы «Gujurly Zahmet» планирует расширить свою деятельность в нефтегазовую отрасль, предлагая услуги по проектированию и обслуживанию моторной техники и гидравлического оборудования. Компания активно использует экологически чистые и энергоэффективные технологии, такие как солнечные батареи и ветрогенераторы, что способствует снижению затрат на электроэнергию и уменьшению воздействия на окружающую среду.

1.2 Теоретические основы анализа и прогнозирования потребительского спроса

Эффективное управление потребительским спросом является важной задачей для бизнеса, стремящегося оптимизировать свои операции и предлагать продукцию в соответствии с потребностями рынка. Анализ и прогнозирование спроса позволяют компании адаптировать свои стратегии, минимизировать издержки и лучше удовлетворять потребности потребителей. Рассмотрим основные методы анализа данных и методы прогнозирования, которые являются ключевыми в работе с данными о спросе.

Методы анализа данных:

- анализ временных рядов: представляют собой последовательность данных, упорядоченных по времени, что позволяет анализировать исторические данные о спросе и выявлять тренды и сезонные колебания. Этот метод включает такие техники, как сглаживание временных рядов (например, метод скользящего среднего) и анализ сезонных компонентов, которые помогают компаниям лучше понять, как спрос меняется с течением времени. На основе анализа временных рядов можно выявить периоды роста и спада спроса и спрогнозировать его изменения в будущем. Например, анализ исторических данных о продажах позволяет установить закономерности, которые отражают сезонные пики спроса, связанные с праздниками или выходом нового продукта на рынок [6];
- регрессионный анализ: является одним из основных методов статистического анализа данных и позволяет выявлять взаимосвязи между переменными, которые могут оказывать влияние на спрос. В контексте анализа потребительского спроса регрессионный анализ помогает определить, как изменения в цене, доходе потребителей, маркетинговых затратах и других внешних факторах могут повлиять на объем продаж. Линейная регрессия, например, используется для

выявления прямой зависимости между двумя переменными, тогда как множественная регрессия позволяет учитывать более сложные многомерные зависимости. Эти модели позволяют менеджерам прогнозировать, как изменение одного или нескольких факторов отразится на спросе [4];

- кластерный анализ: позволяет группировать данные на основе схожих характеристик, что особенно полезно для сегментации клиентов и продуктов. Этот метод помогает выявить группы товаров или клиентов, поведение которых схоже, что упрощает прогнозирование спроса и формирование маркетинговых стратегий. Например, кластерный анализ позволяет выделить сегменты потребителей с одинаковыми предпочтениями и предложить каждому сегменту продукты, которые наиболее вероятно будут востребованы. Это также помогает эффективно распределять ресурсы, фокусируясь на наиболее прибыльных сегментах;
- анализ ассоциаций: помогает выявлять взаимосвязи между товарами, которые покупатели часто приобретают вместе. Этот метод позволяет компании оптимизировать ассортимент, предложив совместно покупаемые товары в виде пакетов, тем самым увеличивая продажи. Например, анализ данных о покупках может выявить, что покупатели, приобретающие кофе, часто покупают сахар и сливки. Знание этих взаимосвязей позволяет предлагать товары совместно, что повышает шансы на успешную продажу;
- машинное обучение и алгоритмы классификации: включает такие методы, как деревья решений, случайный лес и логистическая регрессия, которые могут быть использованы для предсказания вероятности покупки товара определенным клиентом или сегментом клиентов. Эти алгоритмы позволяют анализировать исторические данные и выявлять факторы, которые влияют на поведение потребителей. Применение алгоритмов классификации помогает

точнее определять, какие товары или услуги будут востребованы в ближайшем будущем.

Методы прогнозирования:

- простое и экспоненциальное сглаживание: используются для прогнозирования спроса путем сглаживания колебаний в данных. Простое сглаживание подходит для данных без явных трендов и сезонности, тогда как экспоненциальное сглаживание учитывает эти факторы, придавая больший вес последним данным. Это позволяет реагировать на изменения спроса, сохраняя при этом стабильность прогноза. Методы сглаживания полезны для создания базовых прогнозов в условиях стабильного спроса, но их точность может уменьшаться в условиях высокой волатильности;
- модели ARIMA: являются мощным инструментом для прогнозирования временных рядов, так как они учитывают автокорреляцию и сезонность данных. Эти модели позволяют анализировать взаимосвязь между текущими и предыдущими значениями временного ряда, что делает их эффективными для прогнозирования в условиях, когда данные содержат сезонные и трендовые компоненты. ARIMA модели часто используются в прогнозировании продаж, так как они позволяют учитывать как краткосрочные, так и долгосрочные тренды в данных о потребительском спросе;
- рекуррентные нейронные сети (RNN): эффективно справляются с задачами прогнозирования в условиях сложных данных. RNN моделируют последовательные зависимости и могут предсказывать будущие значения на основе многослойных обученных связей. Данный метод особенно эффективен при анализе временных рядов и сезонных колебаний, так как он позволяет учитывать не только текущие, но и более отдаленные данные. RNN особенно полезны для долгосрочных прогнозов, когда требуется учитывать широкий спектр

факторов, влияющих на спрос;

- градиентный бустинг: создает мощные прогнозные модели путем объединения нескольких слабых моделей (обычно деревьев решений) в одну сильную. Градиентный бустинг позволяет учесть широкий спектр факторов, влияющих на спрос, и повышает точность прогноза за счет корректировки ошибок предыдущих моделей. В контексте анализа потребительского спроса градиентный бустинг может использоваться для выявления влияния различных факторов на спрос, таких как цена, сезонные скидки, мероприятия и т.д;
- сезонное декомпозиционное прогнозирование: позволяет разделить временные ряды на трендовые, сезонные и случайные компоненты. Декомпозиция позволяет увидеть, какие именно изменения являются временными, а какие - устойчивыми, что особенно полезно для прогнозирования сезонного спроса. Например, для товаров, которые пользуются популярностью в определенные сезоны, сезонная декомпозиция позволяет выделить постоянный уровень спроса и скорректировать прогноз, учитывая сезонные пики;
- сценарный анализ и моделирование «что-если»: позволяют компании формировать несколько сценариев развития событий в зависимости от изменения внешних факторов (цены, экономических условий). Это помогает понять, как различные условия могут повлиять на спрос и позволяет подготовить план действий на случай тех или иных изменений на рынке. Например, моделирование «что-если» позволяет оценить, как спрос может измениться при изменении ценовой политики, что помогает принимать более обоснованные решения.

1.3 Разбор методов аналитики, используемых в компании и подбор технологии для концептуального моделирования

В данном разделе проводится углубленный анализ современных технологий концептуального моделирования с акцентом на объектно-ориентированный анализ и проектирование с использованием унифицированного языка моделирования (UML) и IDEF0 [3]. UML (Unified Modeling Language) - универсальный язык визуального моделирования, используемый для спецификации, визуализации, документирования и проектирования различных аспектов сложных систем. Объектная модель, как фундаментальная часть UML, обеспечивает структуру, необходимую для представления, спецификации и проектирования информационных систем, что делает UML важным и незаменимым инструментом при разработке программного обеспечения. Использование UML способствует созданию ясной и структурированной модели системы, что уменьшает риски, связанные с разработкой, и повышает качество конечного продукта [1].

Для создания UML-моделей используются разнообразные программные средства, такие как Draw.io, Gliffy, Visual Paradigm и Microsoft Visio. В данном исследовании предпочтение отдано Visio благодаря его расширенным возможностям по созданию сложных диаграмм, интуитивно понятному интерфейсу и интеграции с другими продуктами Microsoft. Visio поддерживает множество типов диаграмм, таких как диаграммы последовательности и диаграммы процессов, и обладает высокой гибкостью в создании архитектурных решений различной сложности. Например, уникальной особенностью Visio является удобство в создании диаграмм развертывания, которые наглядно отображают физическое расположение компонентов и их взаимодействия, что выгодно отличает его от других инструментов. Например, диаграммы классов позволяют моделировать основные структуры данных и их взаимосвязи, диаграммы компонентов формализуют архитектурные элементы программного обеспечения, а

диаграммы развертывания помогают визуализировать физическое размещение компонентов системы и связи между ними.

UML также включает поведенческие диаграммы, такие как диаграммы последовательности и диаграммы состояний, которые обеспечивают моделирование динамического поведения системы. Диаграммы последовательности особенно полезны для иллюстрации взаимодействий между объектами во времени, помогая понять, как данные и управление перемещаются между компонентами в ходе выполнения операций. Диаграммы состояний предоставляют детализированное представление о возможных состояниях объектов и переходах между этими состояниями, что критически важно для анализа и проектирования реактивных систем. Таким образом, поведенческие диаграммы способствуют более глубокому пониманию функциональности системы и обеспечивают её соответствие требованиям пользователей и бизнес-процессам [7].

Методология IDEF0 представляет альтернативный подход к концептуальному моделированию, который позволяет описывать систему как интегрированную структуру, преобразующую входные данные в результаты. В отличие от других методов, IDEF0 фокусируется на функциональной декомпозиции процессов, что делает её более подходящей для анализа потоков и выявления ключевых функций системы.

IDEF0 была разработана с целью моделирования сложных систем, обеспечивая прозрачный анализ их функций и взаимосвязей. Основная концепция методологии состоит в представлении системы как набора функций, каждая из которых имеет свои входы, выходы, механизмы и элементы управления. Например, звонок клиента может быть представлен в модели как входная информация, в то время как процесс покупки товара - как конечный результат обработки этой информации. Такой подход позволяет наглядно и систематично описывать взаимодействие элементов системы [15].

Методология IDEF0 также может применяться для моделирования процессов в логистике, где входные данные представляют собой поставки

сырья, а результатом является распределение готовой продукции потребителям. Такой подход позволяет выделить ключевые функции логистических процессов и определить точки контроля и оптимизации. Системный подход IDEF0 к декомпозиции процессов позволяет разбирать систему на отдельные функциональные блоки, что существенно упрощает понимание и управление сложными процессами, а также облегчает выявление узких мест и проблемных областей [20]. Благодаря этому IDEF0 нашла широкое применение в бизнес-анализе, системной инженерии и разработке сложных управленческих систем [23].

Для создания моделей IDEF0 используются специализированные инструменты, такие как ERWIN, BPWIN и Ramus. В данном исследовании предпочтение отдано BPWIN, поскольку он является бесплатным и удобным в использовании, а также обеспечивает все необходимые функциональные возможности для создания и редактирования моделей. Основная цель использования методологии IDEF0 заключается в обеспечении универсального подхода к анализу и проектированию систем, представляя их как совокупность взаимосвязанных элементов: люди, информация, программное обеспечение, оборудование, ресурсы. Эти элементы взаимодействуют через потоки данных и ресурсов, обеспечивая выполнение функций системы.

Ключевые компоненты диаграммы IDEF0 включают

- функциональные блоки: обозначают основные функции системы, указывая, какие действия или процессы выполняются в рамках системы.
- стрелки: представляют потоки данных и материалов между функциональными блоками. Стрелки могут обозначать входные и выходные данные, а также управляющие воздействия и ресурсы, необходимые для выполнения функции.
- контролирующие узлы: обозначают условия и ограничения, накладываемые на выполнение функций. Они описывают факторы

или ограничения, которые влияют на выполнение задач, и обеспечивают соблюдение установленных стандартов и правил.

- текстовые аннотации: обеспечивают дополнительные пояснения для более глубокого понимания системы, включая детали, которые могут не быть очевидными из основной диаграммы, но важны для корректной интерпретации модели.

Функциональные блоки могут быть декомпозированы для получения более детализированного представления, что позволяет глубже анализировать каждый компонент системы и улучшает управление ею. Например, верхний уровень модели может представлять общий процесс, такой как управление производством, а дочерние диаграммы могут описывать более детализированные этапы, такие как закупка материалов, сборка продукции и контроль качества. Такой подход к декомпозиции позволяет лучше понять, как различные функции связаны между собой, и облегчает управление сложными проектами, разбивая их на более управляемые и измеримые части [24].

Методология IDEF0 обеспечивает бесценную основу для концептуального моделирования, позволяя аналитикам и проектировщикам создавать эффективные представления сложных систем, облегчать общение между заинтересованными сторонами и поддерживать принятие обоснованных решений при проектировании и внедрении систем.

Тем не менее, на основе проведенного анализа предпочтение отдается методологии UML. Этот выбор обусловлен следующими факторами:

UML предоставляет универсальный язык моделирования, поддерживающий различные виды диаграмм (диаграммы классов, последовательностей и состояний), что позволяет полноценно отобразить как статические, так и динамические аспекты системы.

Широкий набор инструментов UML, таких как Visual Paradigm и Microsoft Visio, обеспечивает удобство визуализации сложных взаимодействий и возможность совместной работы над диаграммами.

UML лучше подходит для моделирования системы, так как позволяет

учесть будущие изменения и модернизацию системы без необходимости полной переделки архитектуры.

Определившись с методологией, следует перейти к анализу текущих процессов компании направленных на анализ продаж и прогнозирование спроса. Текущие рабочие процессы по анализу и прогнозированию представлены на рисунке 2.

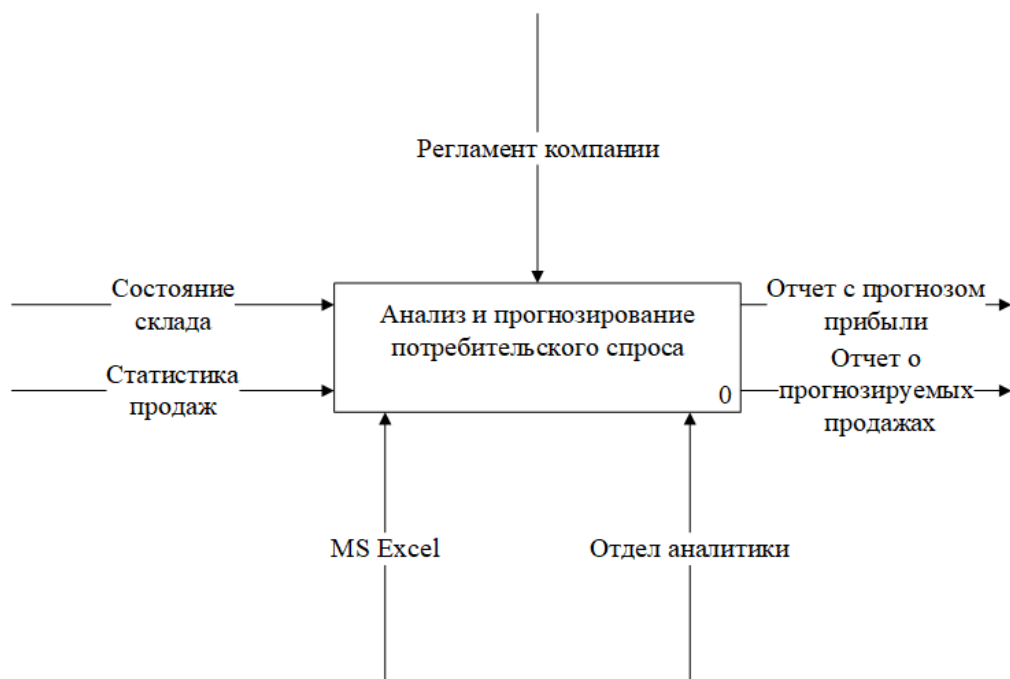


Рисунок 2 – Контекстная диаграмма «Анализ и прогнозирование потребительского спроса» AS-IS

После описания контекстной диаграммы следует разделить её на отдельные подсистемы и описывать каждую из них в деталях. Этот процесс называется декомпозицией и иллюстрируется на рисунке 3.

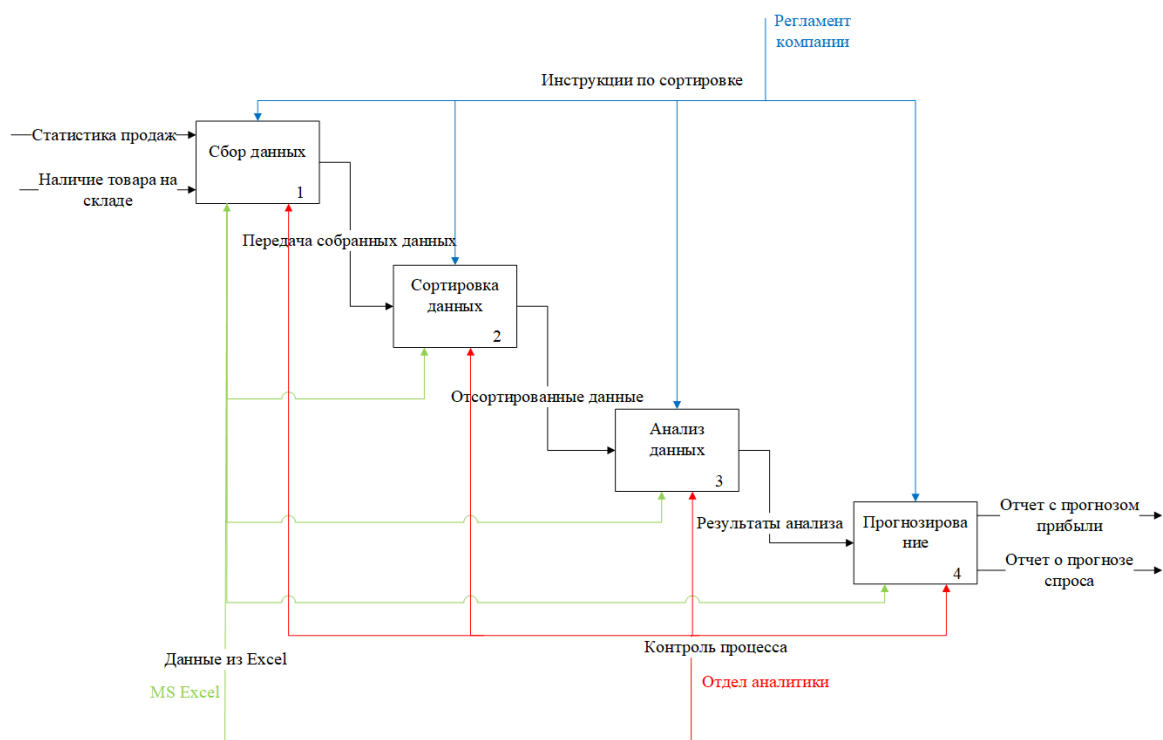


Рисунок 3 – Декомпозиция контекстной диаграммы «Анализ и прогнозирование потребительского спроса»

Деятельность аналитического отдела организации организована в рамках четырех основных этапов, каждый из которых играет ключевую роль в обеспечении качественного аналитического результата.

Сбор данных: на данном этапе осуществляется комплексный сбор всех необходимых данных, включая исторические сведения о продажах, текущий уровень складских запасов, внешние рыночные условия, а также отзывы клиентов и анализ конкурентов. Этот процесс создает информационную базу для дальнейшего анализа.

Предварительная обработка данных: данных предусматривает тщательную проверку, сортировку и очистку данных с целью обеспечения их точности и целостности. Цель этапа - устранение ошибок, пробелов и аномалий. Это позволяет создать высококачественный набор данных для дальнейшего анализа.

Аналитическая обработка данных: включает применение математических и статистических методов для анализа очищенных данных.

Этот этап является фундаментальным для получения достоверных выводов и инсайтов, и требует глубокого понимания применяемых моделей и инструментов анализа. Особое внимание уделяется выбору подходящих методов для извлечения значимых паттернов и закономерностей из данных, таких как регрессионный анализ или кластеризация.

Прогнозирование и отчетность: заключительный этап, включает построение прогнозных моделей на основе проведенного анализа, а также создание отчетов, представляющих результаты в удобной для восприятия форме. Визуализация данных с использованием графиков, диаграмм и текстовых интерпретаций с помощью таких инструментов, как Tableau, позволяет эффективно донести результаты до заинтересованных сторон и поддержать процесс принятия решений.

Анализ данных включает три основных этапа (Рисунок 4), которые являются важными для преобразования сырых данных в полезные аналитические инсайты [25].

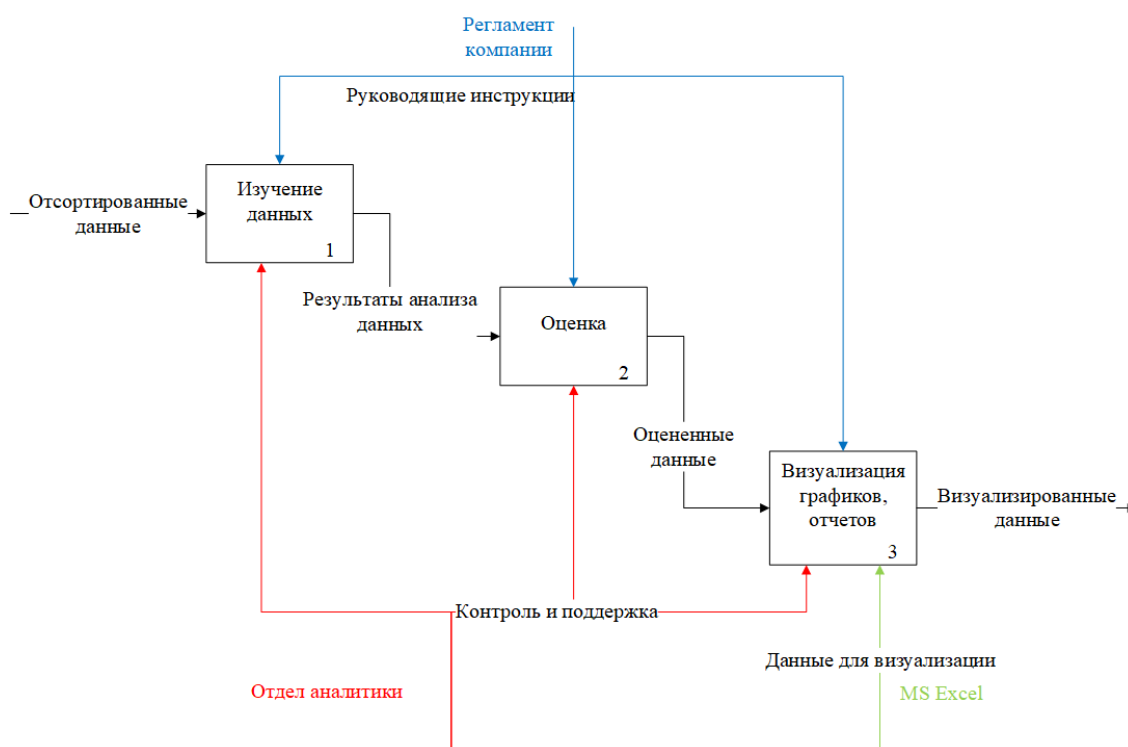


Рисунок 4 – Детализация контекстной диаграммы «Аналитика»

Сбор и предварительная обработка данных: начальный этап, включает тщательное изучение и предварительную обработку собранных данных. Основные действия включают выявление и исправление ошибок, устранение пропусков и отклонений от нормы, а также преобразование данных в цифровые форматы, пригодные для анализа. Такие методы, как нормализация и кодирование, применяются для подготовки наборов данных для дальнейших математических операций.

Оценка: на данном этапе осуществляется применение широкого спектра математических алгоритмов для анализа собранных данных и формирования прогнозов, основанных на текущих тенденциях. Используются статистические методы, такие как регрессионный анализ, кластеризация и анализ временных рядов, для выявления взаимосвязей между переменными и анализа структуры данных. Этот процесс не только позволяет глубоко изучить исторические тренды, но и способствует построению обоснованных сценариев для будущего. Это делает данный этап критически важным элементом прогнозного моделирования [2].

Визуализация: финальная стадия анализа данных заключается в представлении результатов в наглядной и доступной визуальной форме с использованием графиков, диаграмм и интерактивных информационных панелей с помощью таких инструментов, как Tableau и Power BI. Такой подход к визуализации обеспечивает ясность интерпретации данных, что упрощает восприятие ключевых выводов и способствует более эффективному процессу принятия управленческих решений.

Интеграция современных аналитических инструментов в бизнес-процессы позволяет не только оптимизировать рабочие процедуры, но и значительно сократить нагрузку на сотрудников, а также ускорить обработку больших объёмов информации. Например, использование инструмента RPA (Robotic Process Automation) для автоматизации повторяющихся задач показало высокую эффективность, что позволило значительно уменьшить

количество рутинной работы и освободить ресурсы для выполнения более стратегических задач. Это особенно актуально в условиях динамически изменяющегося рыночного контекста, где необходимость в оперативных и обоснованных решениях возрастает. Разработанная модель, сформированная на предыдущих этапах, также предоставляет возможность проведения сравнительного анализа различных методов обработки данных, таких как нормализация данных и стандартизация, что способствует повышению эффективности аналитических процедур и помогает определить наиболее подходящий подход для конкретного типа данных.

Для оценки эффективности предложенного подхода будет использована формула интенсивности труда (ИТ), которая может быть представлена формулой (1):

$$\text{ИТ} = \frac{\text{Человеко-часов}}{\text{Объем работы}}. \quad (1)$$

Эта формула позволяет количественно оценить эффективность использования рабочей силы в процессе анализа данных. Понимание того, сколько человеко-часов требуется для выполнения определенных задач с точки зрения рабочей нагрузки, поможет выявить узкие места в процессах и оптимизировать их. Оценка результативности предложенного подхода отражена в таблице 1.

Таблица 1 – Сравнение моделей процесса аналитики и прогнозирования данных

Модель	Человеко-часы для выполнения задачи	Объем работы	Интенсивность труда
Модель с традиционными методами анализа данных	35	100	0,35
Модель с современными методами анализа данных	10	100	0,10

Исходя из этих данных, можно сделать вывод, что модель, использующая современные методы анализа данных, оказывается более чем в

На основании полученных результатов рекомендуется внедрить новую аналитическую подсистему - «Модуль анализа данных с применением машинного обучения». Данный модуль, как показано на рисунке 5, обеспечит более глубокий и автоматизированный подход к анализу данных, что позволит повысить качество прогнозов за счет более точного выявления паттернов и скрытых зависимостей в данных, а также оптимизировать процесс принятия решений, например, путем более эффективного распределения ресурсов и автоматизированного определения приоритетных действий.

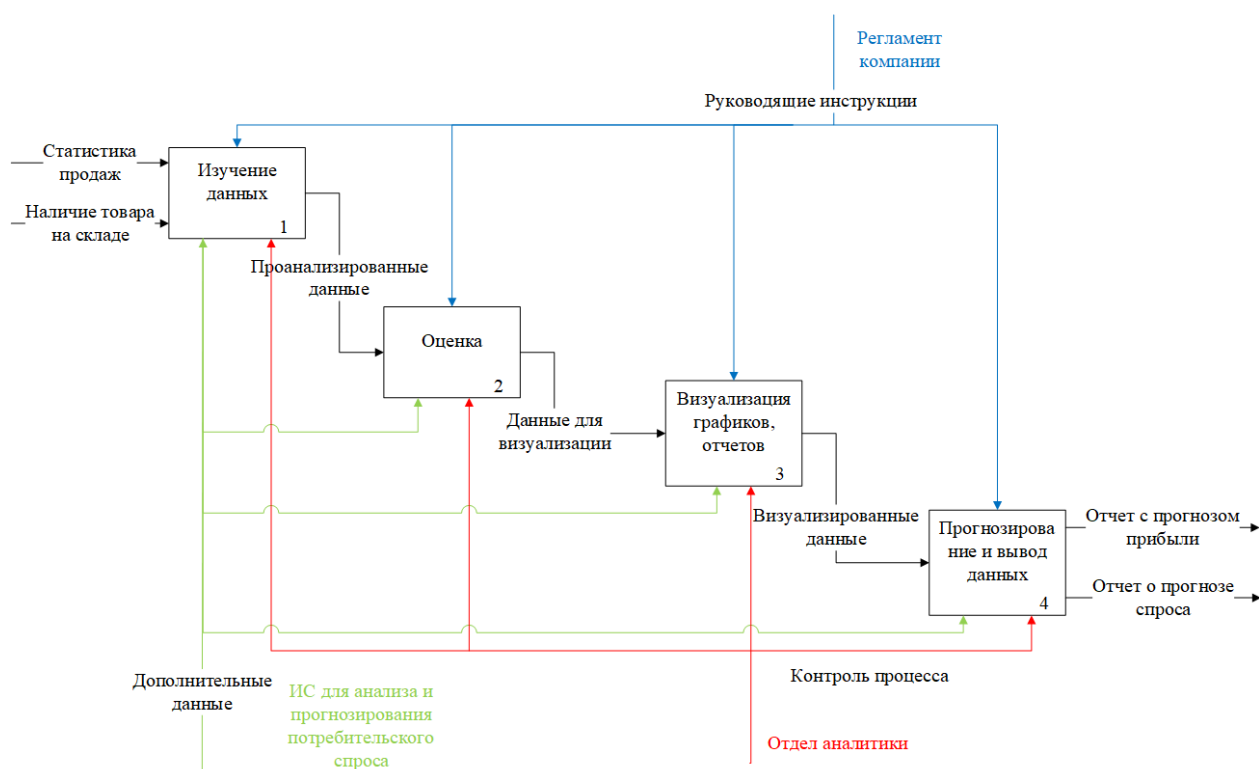


Рисунок 5 – Целевая модель бизнес-процесса «Анализ и прогнозирование»

потребительского спроса» ТО-ВЕ

Диаграмма демонстрирует, что использование приложения для анализа и прогнозирования продаж существенно улучшает бизнес-процессы. Благодаря этому инструменту, организация может уменьшить количество сотрудников, участвующих в выполнении задач, оптимизировать временные затраты и повысить точность результатов.

1.4 Определение и формирование требований к системе

Программное обеспечение для анализа и прогнозирования потребительского спроса, разработанное для использования в организации, должно удовлетворять ряду функциональных и нефункциональных требований, которые обеспечат точность, удобство, надежность и безопасность системы. Главной целью программы является поддержка аналитиков и менеджеров в процессе принятия решений, основанных на данных о спросе и продажах, вводимых вручную. Эта система позволяет формировать точные прогнозы, которые учитывают динамику продаж и сезонные колебания спроса, а также предоставлять визуальные отчеты, облегчающие анализ.

Функциональные требования к системе включают следующие основные компоненты:

Ввод и редактирование данных: в системе должна быть предусмотрена возможность ввода данных о продажах, которые включают такие параметры, как название и категория товара, дата реализации и количество проданных единиц. Пользователь должен иметь возможность создавать новые записи, редактировать существующие и удалять ненужные данные, чтобы актуализировать информацию в базе. Такой подход позволяет обеспечить гибкость в управлении данными и дает возможность пользователям оперативно корректировать сведения, учитывая актуальные изменения.

Обработка и анализ данных: после ввода данных система должна

выполнять анализ, группируя и сортируя данные по категориям товаров и временным интервалам. Это позволяет выявлять основные тенденции в продажах и изменения спроса на различные категории товаров. Программа также предоставляет пользователю информацию о средней динамике продаж, выделяя периоды роста или спада для более глубокого анализа потребительских предпочтений. Такие функции дают аналитикам и менеджерам возможность лучше понять поведение рынка и оперативно реагировать на изменения.

Функциональность прогнозирования: основной задачей системы является возможность предсказания будущего спроса на товары. Для этого в программное обеспечение должны быть внедрены алгоритмы анализа временных рядов, такие как экспоненциальное сглаживание, которые позволяют учитывать тенденции и сезонные колебания спроса. В результате программа должна предоставлять пользователю прогнозы, которые помогают формировать стратегии запасов и планировать поставки на основе текущей информации о продажах.

Визуализация результатов анализа и прогнозов: для удобства восприятия информации система должна формировать графики и таблицы, которые наглядно отображают результаты анализа и прогнозов. Пользователи должны иметь возможность настраивать период отображения данных, что позволяет фокусироваться на интересующих временных интервалах и получать отчеты в удобном для анализа формате. Визуальные представления помогают аналитикам и менеджерам быстро оценить результаты прогноза и принять необходимые решения.

Генерация отчетов: система должна создавать сводные отчеты, которые отображают ключевые показатели продаж и прогнозов. Отчеты могут включать информацию по категориям товаров, сравнение данных за разные периоды и визуальные элементы для облегчения восприятия информации. Эти отчеты могут использоваться руководством для анализа результатов и адаптации стратегии компании в зависимости от изменения потребительского

спроса.

Нефункциональные требования к системе включают следующие аспекты:

Производительность: программа должна обрабатывать данные и выполнять анализ с минимальными задержками, чтобы пользователь мог получить результаты в кратчайшие сроки. Это особенно важно в условиях, когда необходимо обработать большие объемы данных за длительные временные периоды. Оптимизация алгоритмов анализа и прогнозирования позволяет обеспечить высокую производительность, сохраняя точность расчетов и быстроту выполнения операций. Низкое время отклика системы способствует комфортной работе пользователей и повышает их удовлетворенность использованием программы.

Масштабируемость: хотя система ориентирована на определенный объем данных, она должна быть готова к масштабированию для поддержки увеличивающегося числа записей о продажах. Применение эффективных методов хранения и обработки данных позволяет системе адаптироваться к растущим объемам информации без значительных изменений в архитектуре. Это обеспечивает устойчивость системы к изменяющимся условиям и позволяет компании расширять использование программы по мере роста бизнеса.

Надежность: программа должна обеспечивать бесперебойную работу и быть устойчива к сбоям. Для этого предусматривается резервное копирование данных и возможности восстановления информации. Надежность системы особенно важна для организаций, которые полагаются на точные прогнозы в процессе планирования запасов и управления ресурсами. Внедрение механизмов регулярного резервного копирования и восстановления данных позволяет избежать потерь важной информации и минимизировать риски, связанные с возможными сбоями в работе программы.

Удобство использования: интерфейс системы должен быть интуитивно понятным, чтобы сотрудники с разным уровнем технической подготовки

могли эффективно работать с программой. Простой и лаконичный дизайн интерфейса позволяет пользователям быстро осваивать систему, а также настраивать отображение данных и выбирать временные периоды для анализа. Включение функций, которые поддерживают пользовательские настройки отображения информации, позволяет каждому сотруднику работать с интерфейсом, настроенным под его потребности, будь то аналитики, менеджеры или руководители компании.

Безопасность данных: система должна защищать данные от несанкционированного доступа, что особенно важно в условиях работы с коммерческой информацией. Для обеспечения безопасности данных внедрены механизмы аутентификации пользователей, а также контроль доступа к отдельным разделам системы на основе их роли. Кроме того, программа поддерживает шифрование данных и регулярное обновление систем защиты, что минимизирует вероятность утечек и повышает общий уровень информационной безопасности [9].

Технологическая база: для реализации данных требований программа может использовать языки программирования и технологии, подходящие для работы с аналитикой и визуализацией данных, такие как Python и библиотеки Pandas и Matplotlib. Это обеспечивает высокую производительность в обработке и визуализации данных, а также позволяет пользователю гибко управлять интерфейсом системы и отображаемыми данными. Программа также может поддерживать интеграцию с Microsoft Excel для упрощенного импорта и экспорта данных, что облегчает обмен информацией между сотрудниками компании.

Выводы к главе 1: в первой главе был проведен всесторонний анализ предметной области и описаны ключевые характеристики организации ИП «Gujurly Zahmet». Были рассмотрены основные направления деятельности компании и выявлены аспекты, требующие улучшения в контексте анализа и прогнозирования потребительского спроса. Важно понимать особенности текущих бизнес-процессов, чтобы предложить наилучшие решения для

оптимизации.

В ходе работы были определены и обоснованы выбор технологии для концептуального моделирования. Рассмотренные технологии позволили выбрать наиболее подходящие инструменты для построения модели, которая полностью соответствует требованиям компании и специфике её деятельности. Концептуальное моделирование стало основой для проектирования системы.

Также, были подробно описаны требования к разрабатываемой системе. Определение функциональных и нефункциональных требований позволило сформулировать полное видение системы, охватывающее как её основные функции, так и важные характеристики, такие как надежность, безопасность, производительность и масштабируемость. Эти требования послужат фундаментом для успешной реализации программного обеспечения.

Таким образом, мы заложили теоретическую и практическую основу для дальнейшего проектирования и разработки системы анализа и прогнозирования потребительского спроса.

Глава 2 Логическое проектирование системы анализа и прогнозирования потребительского спроса

2.1 Описание инструментов для анализа и прогнозирования на основе машинного обучения

В данном разделе описаны инструменты, необходимые для анализа и прогнозирования потребительского спроса. Выбор этих инструментов направлен на реализацию задач по созданию системы для обработки данных и формирования прогнозов, что соответствует цели разработки приложения для анализа спроса. Включение библиотек и технологий для машинного обучения позволяет создать модель, способную адаптироваться к изменениям спроса и учитывать сезонные колебания, тем самым обеспечивая выполнение задач ВКР.

Анализ исторических данных о продажах представляет собой критически важный этап в прогнозировании будущего спроса и эффективном управлении ассортиментом продукции. Глубокое понимание тенденций и колебаний в данных позволяет организациям не только быстро реагировать на текущие рыночные условия, но и стратегически планировать свои действия. Система, разработанная с использованием Django и поддерживаемая библиотеками Python, такими как Pandas, Matplotlib и Statsmodels, предоставляет аналитикам возможности для проведения всестороннего анализа данных и построения надежных прогнозных моделей [5]. Используя методы анализа временных рядов, включая экспоненциальное сглаживание и авторегрессионные модели интегрированной скользящей средней (ARIMA), организации могут выявлять важные тенденции и сезонные колебания, которые имеют критическое значение для оптимизации бизнес-процессов.

Одним из ключевых преимуществ аналитических методов является интеграция алгоритмов машинного обучения, способных учитывать многочисленные факторы, влияющие на спрос. Эти факторы включают

маркетинговые кампании, сезонные акции и макроэкономические условия, что позволяет компаниям более эффективно адаптировать свои стратегии к изменяющимся рыночным условиям [8]. Такая адаптивность не только повышает оперативность реагирования, но и значительно улучшает качество стратегических решений.

Еще одним важным аспектом управления бизнесом является оптимизация ассортимента продукции. Этот процесс включает выбор наиболее подходящего набора товаров для предложения клиентам, что напрямую влияет на объем продаж и прибыльность компании. Задачи оптимизации могут решаться с помощью методов линейного программирования и программирования на основе смешанных чисел с использованием таких библиотек, как PuLP. Эти инструменты облегчают поиск оптимальных комбинаций товаров, учитывая различные ограничения, такие как бюджетные рамки, наличие на складе и предпочтения клиентов. Например, с помощью оптимизации компании могут определить, на какие товары следует делать акцент в рекламных кампаниях, чтобы максимизировать прибыль, принимая во внимание сезонные колебания и изменения в предпочтениях потребителей.

Для эффективного планирования ассортимента необходимо проводить непрерывный мониторинг данных о запасах, объемах продаж и других ключевых факторах, таких как сезонные колебания, потребительские предпочтения и макроэкономические условия, влияющие на спрос. Такой мониторинг должен осуществляться в режиме реального времени, чтобы обеспечить своевременную реакцию на изменения. Интеграция программного обеспечения с системами управления складом и продажами играет важнейшую роль в повышении операционной эффективности. Использование современных API и стандартизированных протоколов, таких как REST API, обеспечивает надежную и непрерывную синхронизацию данных между системами. Это помогает поддерживать актуальность информации, автоматизировать процессы управления запасами и минимизировать ошибки.

В результате такая интеграция позволяет компании сосредоточиться на стратегических инициативах, таких как расширение рынка и оптимизация логистики.

Применение алгоритмов машинного обучения для прогнозирования и оптимизации: использование алгоритмов машинного обучения является мощным инструментом для построения более точных и адаптивных моделей прогнозирования. Например, методы классификации, такие как деревья решений и логистическая регрессия, позволяют выявлять скрытые паттерны в данных, которые сложно обнаружить с помощью традиционных методов анализа. Деревья решений особенно полезны для сегментации клиентов, позволяя выделять группы с различными предпочтениями и поведением, что способствует разработке персонализированных маркетинговых стратегий.

Алгоритмы кластеризации, такие как K-means, применяются для группировки данных на основе сходных характеристик, что помогает бизнесу лучше понять структуру спроса и адаптировать свои предложения [12]. Это особенно полезно в розничной торговле, где сегментация продуктов и клиентов способствует оптимальному размещению товаров и улучшению клиентского опыта.

Для прогнозирования спроса также применяются методы глубокого обучения, такие как рекуррентные нейронные сети (RNN), которые учитывают последовательные зависимости в данных. Эти модели особенно эффективны при анализе временных рядов, где необходимо учитывать долгосрочные зависимости и тренды. Применение RNN позволяет компании строить точные прогнозные модели, учитывающие исторические данные, текущие рыночные условия и ожидаемые изменения, делая планирование более точным и эффективным.

Кроме того, для оптимизации ассортимента и управления запасами применяются алгоритмы генетического программирования, которые позволяют находить оптимальные решения в условиях большого количества переменных и ограничений. Эти методы имитируют процесс естественного

отбора, что позволяет находить близкие к оптимальным решения даже в условиях высокой неопределенности и многокритериальности. Генетические алгоритмы полезны при решении задач, где традиционные методы оптимизации оказываются недостаточно эффективными из-за сложности или размерности проблемы.

Визуализация данных для поддержки принятия решений: визуализация данных является важным инструментом для интерпретации и представления аналитической информации, что позволяет компаниям принимать обоснованные решения на основе сложных наборов данных. Использование инструментов визуализации помогает управленцам и аналитикам легко анализировать тенденции, выявлять аномалии и оценивать результаты.

Библиотеки Python, такие как Matplotlib, Seaborn и Plotly, позволяют создавать графики, диаграммы и интерактивные панели, делая данные более доступными и понятными [28]. Например, графики временных рядов, построенные с использованием Matplotlib, могут наглядно демонстрировать изменения в спросе на продукцию с течением времени, а диаграммы распределения, созданные в Seaborn, помогают выявить закономерности и аномалии в продажах.

Интерактивные панели, созданные с использованием Plotly или Dash, позволяют пользователям взаимодействовать с данными, изменяя параметры визуализации и погружаясь в детали, что способствует более глубокому пониманию ключевых показателей эффективности (KPI) и ускоряет процесс принятия решений [29]. Это особенно важно в условиях быстро меняющегося рынка, где оперативная реакция на изменения может стать решающим фактором конкурентного преимущества.

Кроме того, визуализация данных упрощает коммуникацию внутри организации, позволяя различным отделам, включая маркетинг, продажи и логистику, работать с единым источником информации и принимать скоординированные решения. Четкие и понятные визуальные представления помогают избежать недопонимания и улучшают взаимодействие между

подразделениями, что способствует общей эффективности бизнеса [27].

В системе для анализа и прогнозирования потребительского спроса применяются несколько основных алгоритмов, которые обеспечивают точное предсказание на основе исторических данных. Эти алгоритмы позволяют учесть тенденции спроса и адаптироваться к изменениям на рынке, что помогает компании принимать более обоснованные решения.

Экспоненциальное сглаживание: используется для предсказания будущего спроса на основе прошлых данных, придавая больший вес более свежим данным. Такой подход позволяет учитывать недавние изменения спроса и корректировать прогнозы в зависимости от актуальных рыночных условий. Этот алгоритм применяется для создания краткосрочных прогнозов, когда требуется оперативная адаптация к изменениям на рынке.

Метрика оценки точности - Средняя абсолютная ошибка (MAE): используется для оценки точности прогнозов в системе. MAE измеряет точность прогнозирования, показывая среднее отклонение предсказанных значений от фактических. Этот показатель помогает оценить точность модели, выявляя среднюю величину ошибки прогноза и позволяя компании корректировать алгоритмы для повышения точности предсказаний.

В совокупности эти инструменты и методологии предоставляют организациям возможность более глубоко и эффективно анализировать данные, визуализировать результаты и использовать алгоритмы для адаптации к динамичным рыночным условиям. Это позволяет принимать обоснованные решения, улучшать качество прогнозов и оперативно реагировать на флуктуации рынка.

Описанные инструменты для анализа и прогнозирования обеспечивают основу для построения точных прогнозов, необходимых для поддержания конкурентоспособности компании. Использование методов машинного обучения и анализа временных рядов полностью соответствует задачам ВКР, направленным на разработку системы для анализа и предсказания спроса.

2.2 Проектирование статической структуры приложения

Этот раздел посвящен разработке статической структуры приложения, включающей проектирование классов и атрибутов, которые необходимы для обработки и анализа данных о спросе. Структура системы была выбрана с учетом задач по созданию гибкой и масштабируемой архитектуры, что отвечает требованиям к точности анализа и предсказания данных о потребительском спросе. Проектирование классов и атрибутов направлено на обеспечение высокой производительности и надежности приложения, что также соответствует задачам ВКР.

Диаграммы классов являются одним из важнейших средств для визуализации и проектирования программных систем в объектно-ориентированном программировании. Они предоставляют наглядное представление о структуре системы, определяя её основные компоненты и связи между ними. Это позволяет разработчикам, архитекторам и другим участникам процесса разработки эффективно сотрудничать, анализировать систему на различных уровнях абстракции и принимать обоснованные решения по её проектированию. В отличие от других типов диаграмм, например диаграмм последовательностей или диаграмм состояний, которые акцентируют внимание на динамических аспектах системы, диаграммы классов сфокусированы на статических взаимодействиях между объектами и их структурах, что позволяет увидеть каркас будущей системы [22].

Классы в объектно-ориентированном программировании играют ключевую роль, поскольку они представляют собой шаблоны или абстракции для объектов, которые будут использоваться в системе. Каждый объект создается на основе определенного класса и наследует его свойства и поведение. Класс можно представить как шаблон или модель, которая описывает, какие характеристики (атрибуты) и поведение (методы) будут присущи объектам, создаваемым на его основе. Другими словами, класс служит абстракцией, определяющей общие свойства и действия, которыми

будут обладать объекты этого типа.

Атрибуты - это переменные, которые хранят данные или состояние объекта. Каждая сущность в объектно-ориентированной системе может обладать своими уникальными характеристиками, которые и определяются через атрибуты. Например, в классе «Автомобиль» атрибуты могут включать «цвет», «модель», «год выпуска» и «двигатель». Эти атрибуты определяют конкретные свойства объекта «Автомобиль», которые могут быть разными для каждого экземпляра этого класса. Атрибуты могут иметь различные типы данных: строки, числа, логические значения и даже сложные структуры данных.

Важной особенностью атрибутов является их модификаторы доступа. Атрибуты могут быть:

- приватными (private) - доступ к таким атрибутам возможен только внутри самого класса. Это важный аспект инкапсуляции, когда внешние объекты или классы не могут напрямую изменять внутреннее состояние объекта;
- защищёнными (protected) - доступ к ним возможен как внутри класса, так и в его производных классах;
- публичными (public) - атрибуты доступны из любого места программы, что делает их уязвимыми для случайных изменений.

Примером может служить класс «Человек», у которого есть атрибут «возраст». Чтобы защитить его от прямого изменения извне, этот атрибут можно сделать приватным, а доступ к нему осуществлять через специальные методы доступа (геттеры и сеттеры), что также помогает реализовывать контроль валидации данных.

Методы (или операции) представляют собой функции, определяющие поведение объектов. Каждый класс может содержать один или несколько методов, которые позволяют объектам взаимодействовать с окружающей средой или изменять своё внутреннее состояние. Например, в классе «Автомобиль» метод может называться «запустить Двигатель», и он будет

изменять состояние объекта, переводя двигатель из неактивного состояния в активное.

Методы могут быть статическими и нестатическими:

Статические методы принадлежат самому классу и могут быть вызваны без создания экземпляра объекта. Они часто используются для выполнения операций, которые не зависят от состояния конкретного объекта, например, математические расчёты или вспомогательные функции.

Нестатические методы принадлежат объектам класса и могут изменять их состояние. Они вызываются через экземпляры класса и могут работать с атрибутами объекта.

Методы также могут иметь модификаторы доступа (`private`, `public`, `protected`), что позволяет регулировать, кто и каким образом может использовать или переопределять их. Например, метод, который отвечает за сложную внутреннюю логику класса, может быть закрыт для доступа внешних объектов.

Методы могут быть перегружены, то есть один и тот же метод может принимать разные наборы параметров. Это позволяет гибко настраивать поведение класса, делая его более универсальным.

Объекты являются конкретными экземплярами классов. Каждый объект обладает уникальным состоянием, хотя все объекты одного класса имеют одинаковый набор атрибутов и методов. Например, два объекта класса «Автомобиль» могут иметь разные значения атрибутов «цвет» или «модель», но оба они будут иметь одинаковые методы, такие как «запустить Двигатель» или «остановить Автомобиль».

Объекты могут быть физическими или абстрактными. Физические объекты, такие как автомобили или книги, представляют реальные сущности, в то время как абстрактные объекты, например, «Заказ» или «Транзакция», могут представлять концептуальные структуры в программных системах. Это делает объектно-ориентированный подход универсальным для моделирования как реальных, так и виртуальных систем.

Состояние объекта определяется значениями его атрибутов в конкретный момент времени. Состояние объекта может изменяться в результате выполнения методов. Например, объект «Дверь» может находиться в состоянии «открыта» или «закрыта», и метод «открыть» изменяет состояние объекта на «открыта».

Отслеживание состояния объекта имеет большое значение для управления его поведением в программе. Многие методы зависят от текущего состояния объекта и могут изменять его в зависимости от действий, которые необходимо выполнить. Примером может служить объект «Светофор», который изменяет своё состояние с «красного» на «зеленый» при выполнении метода «сменитьСигнал».

Поведение объекта определяется набором методов, которые описывают, как объект взаимодействует с другими объектами или с системой в целом. Каждый метод может менять внутреннее состояние объекта или вызывать методы других объектов для выполнения тех или иных операций. Например, в классе «Телевизор» метод «включить» может изменять состояние объекта с «выключен» на «включен», а метод «изменитьКанал» взаимодействует с другими объектами, связанными с передачей информации о канале.

Поведение объектов может быть сложным и зависеть от множества факторов, таких как текущие значения атрибутов, внешние события или результат выполнения других методов. Это делает объектно-ориентированные системы гибкими и мощными в управлении сложными взаимодействиями между компонентами программы.

Стереотипы в UML (Unified Modeling Language) - это специальные метки, которые помогают классифицировать классы и обозначать их специфические роли в системе. Они могут расширять стандартные понятия классов, вводя дополнительные типы классов или создавая новые шаблоны для использования в различных контекстах. Например, стереотип «entity» может быть использован для обозначения класса, который представляет данные, а стереотип «controller» - для класса, управляющего логикой

взаимодействия между объектами.

Контрольные классы управляют взаимодействиями между другими классами и обеспечивают выполнение бизнес-логики системы [10]. Они выполняют роль координаторов, которые управляют процессами, инициируют выполнение методов в других классах и обеспечивают соблюдение правил, заложенных в архитектуре системы.

Кроме определения самих классов, диаграммы классов фокусируются на отображении отношений между классами, что является важной частью моделирования систем. Одним из базовых типов отношений является ассоциация - это связь между двумя классами, указывающая на их потенциальное взаимодействие. Например, класс «Покупатель» может быть связан с классом «Заказ», что указывает на то, что каждый покупатель может делать заказы.

Более сложными типами отношений являются агрегация и композиция, которые определяют, как классы могут состоять из других классов. Например, класс «Автомобиль» может содержать несколько экземпляров класса «Колесо» через композицию, где колёса не могут существовать без автомобиля.

Диаграмма классов для системы анализа и прогнозирования потребительского спроса организации представляет собой детальное отображение структуры системы, включая множество классов и их взаимосвязей. Она служит основой для разработки и проектирования, помогая командам разработчиков и аналитиков глубже понять архитектуру системы и взаимодействие между ее компонентами. Кроме того, диаграмма классов является важным инструментом в процессе создания высококачественного программного обеспечения, способствуя ясности и точности в проектировании.

Диаграмма классов - это не только средство визуализации структуры системы, но и важный инструмент, который способствует пониманию взаимодействий между классами. Так разработчики могут более эффективно

работать над проектом, улучшая качество конечного продукта и обеспечивая его соответствие требованиям пользователей. В конечном итоге, правильное формирование и использование диаграмм классов позволяет значительно упростить процесс разработки, снизить вероятность ошибок и повысить продуктивность команды (таблица 2).

Диаграмма классов для системы анализа и прогнозирования потребительского спроса организации представлена на рисунке 6.

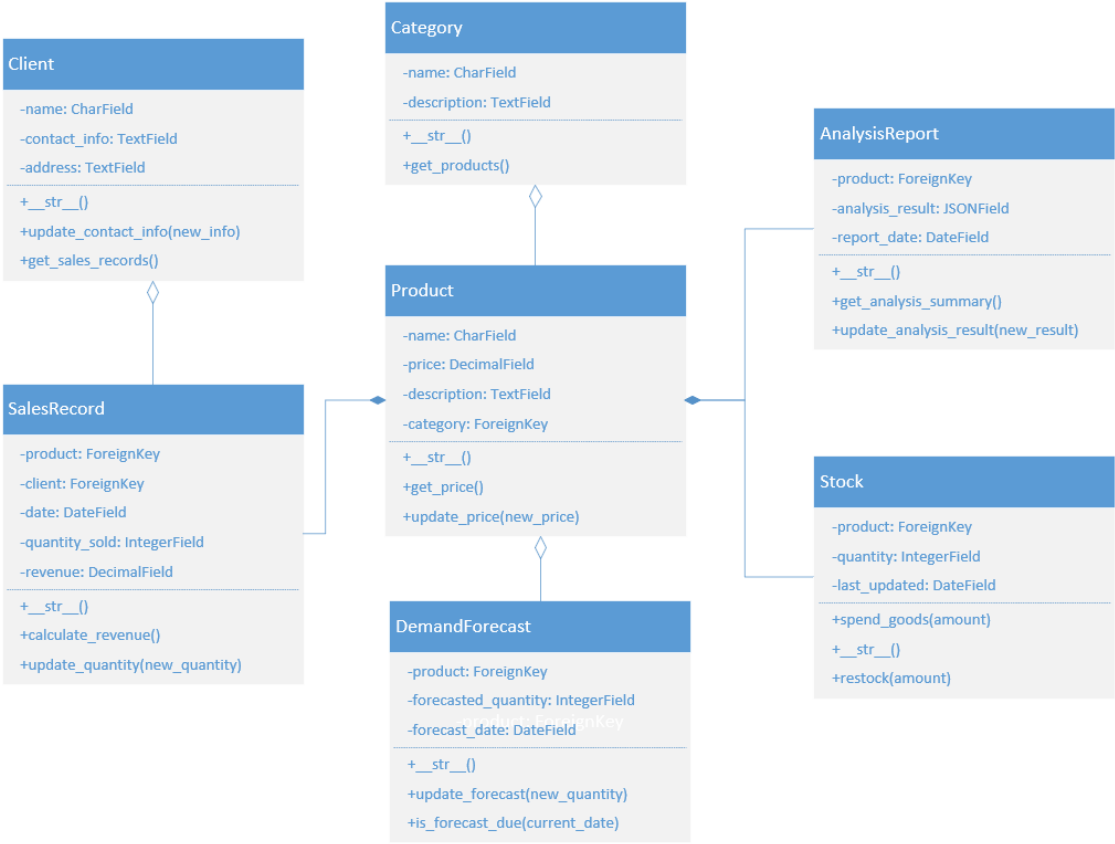


Рисунок 6 – Диаграмма классов

Таблица 2 – Описание полей диаграммы классов

Название поля	Описание	Тип данных
Client	Содержит информацию о клиенте, включая его имя, контактные данные и адрес.	Объект
SalesRecord	Хранит информацию о продаже, связывая данные о товарах, клиенте, дате и количестве проданного.	Объект

Продолжение таблицы 2

Название поля	Описание	Тип данных
Product	Описывает товар, включая его название, категорию и прочие характеристики.	Объект
Category	Указывает категорию продукта, помогая структурировать ассортимент товаров.	Строка
AnalysisReport	Генерируемый системой отчет, содержащий результаты анализа данных о продажах и спросе.	Текст
DemandForecast	Поле для хранения прогноза спроса на основе анализа исторических данных о продажах.	Числовое значение
Stock	Отражает количество товаров на складе, что помогает управлять запасами и отслеживать их наличие.	Целочисленный
Revenue	Сумма выручки от продажи конкретного товара или услуги.	Денежный
Date	Дата записи о продаже, используемая для анализа временных тенденций спроса.	Дата
Review	Отзывы клиентов о товаре, которые помогают оценить уровень удовлетворенности и улучшить продукт.	Текст
Quantity	Количество проданных единиц товара, важный показатель для анализа объемов продаж.	Целочисленный
AnalysisResult	Итоговый результат анализа данных, использующийся для формирования отчетов и прогноза.	Текст или число

Система анализа и прогнозирования потребительского спроса состоит из множества классов, каждый из которых выполняет определенные функции. На основе диаграммы, представленной ниже, мы можем более детально рассмотреть каждый из этих классов и их взаимодействия.

На рисунке 7 представлен класс Client, который моделирует сущность клиента, предоставляя ключевые атрибуты и методы для управления клиентскими данными. Данный класс содержит три атрибута: name, contact_info и address, а также два метода: update_contact_info(new_info) и get_sales_records().



Рисунок 7 – Класс «Client»

Атрибуты класса Client:

- name: имя клиента, представлено строковым типом данных;
- contact_info: контактная информация клиента, хранится в текстовом формате;
- address: адрес клиента, также представлен текстовым значением.

Методы класса Client:

- update_contact_info(new_info): обновляет контактную информацию клиента на основании переданного аргумента;
- get_sales_records(): возвращает все записи о продажах, связанных с данным клиентом.

Интерфейсная форма (окно управления клиентами):

- поля для ввода имени, контактной информации и адреса;
- кнопки: «Добавить клиента», «Редактировать клиента», «Удалить клиента».

На рисунке 8 изображен класс SalesRecord, который описывает запись о продаже, связывая информацию о продукте, клиенте и результатах финансовой транзакции. Класс включает атрибуты: product, client, date, quantity_sold и revenue, а также методы: calculate_revenue() и

update_quantity(new_quantity).

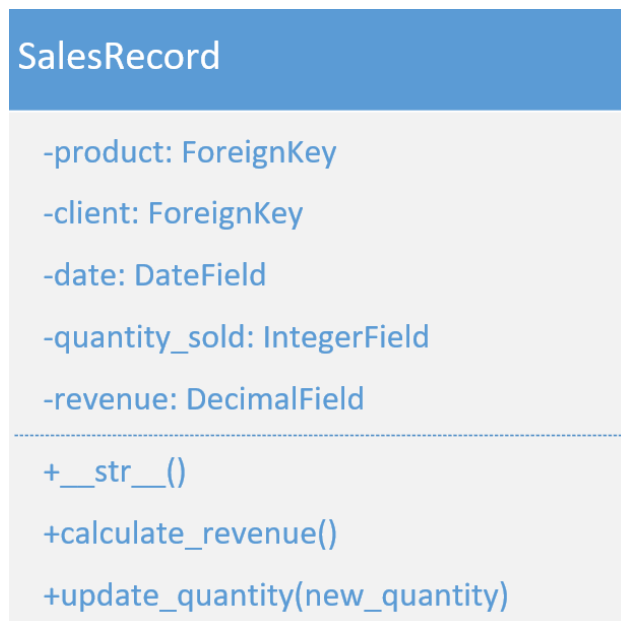


Рисунок 8 – Класс «SalesRecord»

Атрибуты класса SalesRecord:

- product: внешний ключ, ссылающийся на конкретный продукт, связывающий запись о продаже с соответствующим товаром;
- client: внешний ключ, указывающий на клиента, совершившего покупку;
- date: дата продажи, представлена типом данных дата;
- quantity_sold: количество проданных единиц товара, представлено целочисленным типом данных;
- revenue: общая сумма выручки от продажи, представлена числом с плавающей точкой.

Методы класса SalesRecord:

- calculate_revenue(): рассчитывает выручку на основе количества проданных единиц и цены товара;
- update_quantity(new_quantity): обновляет количество проданных

единиц до нового значения.

Интерфейсная форма (форма ввода данных о продаже):

- выпадающие списки для выбора продукта и клиента;
- поле выбора даты (календарь);
- поле для ввода количества проданных единиц;
- поле «Выручка» автоматически рассчитывается;
- кнопки: «Сохранить», «Редактировать запись».

На рисунке 9 представлен класс Product, который моделирует сущность продукта, включающую атрибуты: name, price, description и category. Класс также содержит методы get_price() и update_price(new_price).

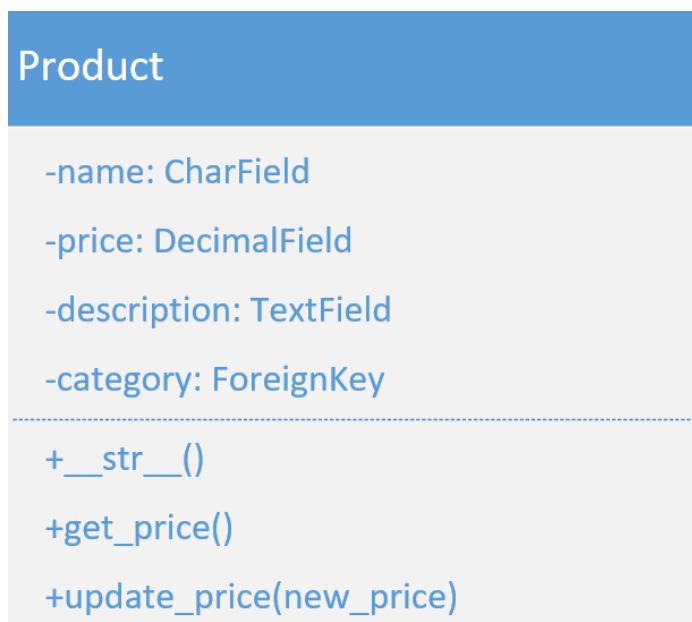


Рисунок 9 – Класс «Product»

Атрибуты класса Product:

- name: название продукта, представлено строковым типом данных;
- price: цена продукта, представлена числом с плавающей точкой;
- description: описание продукта, может быть пустым;
- category: внешний ключ, связывающий продукт с его категорией.

Методы класса Product:

- `get_price()`: возвращает текущую цену продукта;
- `update_price(new_price)`: изменяет цену продукта на указанное значение.

Интерфейсная форма (окно управления товарами):

- поля для ввода названия, цены, описания товара;
- выпадающий список для выбора категории;
- функции поиска и фильтрации товаров;
- кнопки: «Добавить товар», «Редактировать товар», «Удалить товар».

На рисунке 10 изображен класс DemandForecast, который описывает прогноз спроса на определенный продукт. Класс включает атрибуты: `product`, `forecasted_quantity` и `forecast_date`, а также методы: `update_forecast(new_quantity)` и `is_forecast_due(current_date)`.

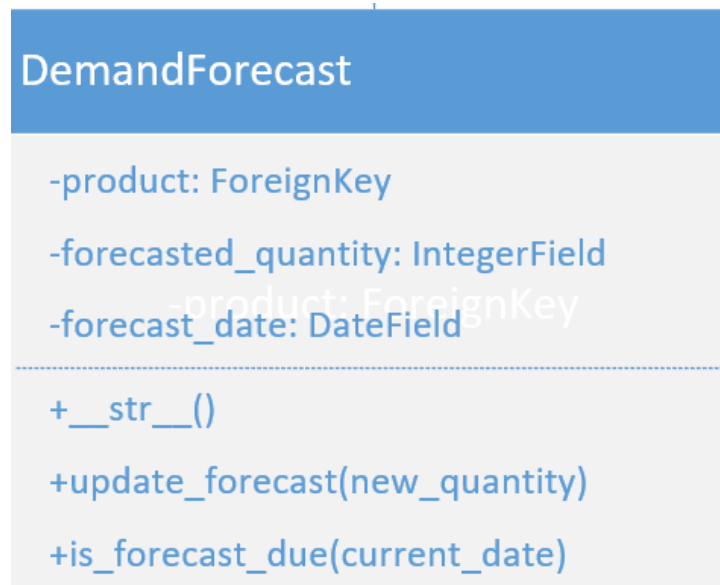


Рисунок 10 – Класс «DemandForecast»

Атрибуты класса DemandForecast:

- `product`: связывает прогноз с конкретным продуктом;
- `forecasted_quantity`: прогнозируемое количество единиц, представлено

целочисленным типом данных;

- `forecast_date`: дата, на которую составлен прогноз, представлена типом данных `date`.

Методы класса `DemandForecast`:

- `update_forecast(new_quantity)`: обновляет прогнозируемое количество товара;
- `is_forecast_due(current_date)`: проверяет, истек ли срок действия прогноза на основе текущей даты.

Интерфейсная форма (окно прогноза спроса):

- поля для ввода и просмотра прогнозируемого количества;
- календарь для выбора даты;
- визуализация данных прогноза с помощью графиков;
- кнопка «Обновить прогноз».

На рисунке 11 изображен класс `Stock`, который моделирует управление запасами продукции на складе. Класс включает атрибуты: `product`, `quantity` и `last_updated`, а также методы: `spend_goods(amount)` и `restock(amount)`.

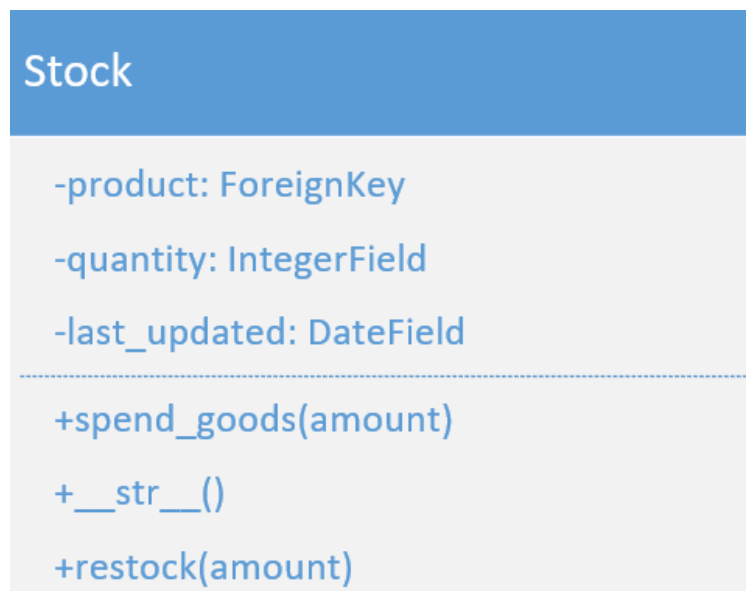


Рисунок 11 – Класс «Stock»

Атрибуты класса Stock:

- product: указывает на продукт, запасы которого хранятся на складе;
- quantity: количество доступных единиц данного продукта;
- last_updated: дата последнего обновления информации о запасах.

Методы класса Stock:

- spend_goods(amount): списывает указанное количество товара со склада, проверяя наличие достаточного количества;
- restock(amount): добавляет заданное количество товара на склад для пополнения запасов.

Интерфейсная форма (окно управления запасами):

- поля для выбора продукта и отображения количества на складе;
- автоматическое обновление даты последнего изменения;
- кнопки: «Списать товар», «Пополнить склад», «Обновить запасы».

На рисунке 12 представлен класс «AnalysisReport». Он включает в себя атрибуты: «product», «analysis_result» и «report_date», а также методы: get_analysis_summary() и update_analysis_result(new_result).

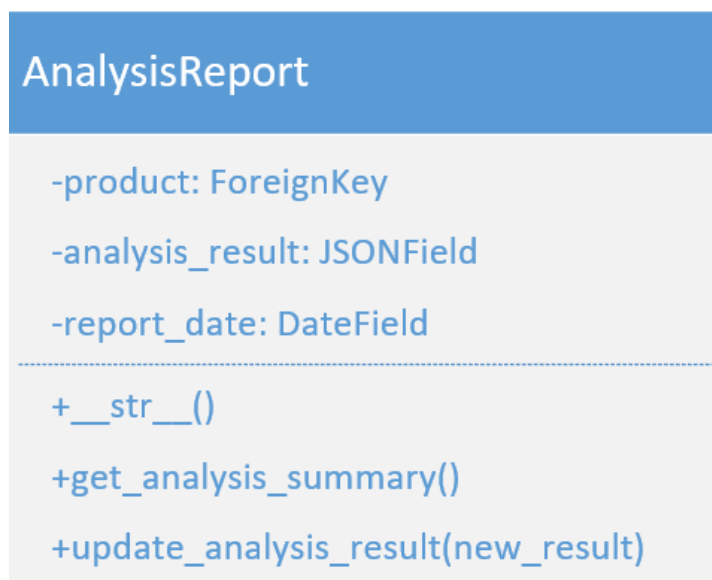


Рисунок 12 – Класс «AnalysisReport»

Атрибуты класса AnalysisReport:

- product: связывает отчет с конкретным продуктом;
- analysis_result: хранит результат анализа в формате JSON;
- report_date: указывает дату создания отчета.

Методы класса AnalysisReport:

- get_analysis_summary: возвращает краткое содержание результатов анализа;
- update_analysis_result: обновляет результат анализа.

Интерфейсная форма (окно создания отчетов):

- отображение данных анализа в виде текста или таблиц;
- кнопки: «Создать отчет», «Экспортировать в PDF/Excel»;
- фильтры для настройки данных.

Статическая структура приложения была спроектирована таким образом, чтобы поддерживать эффективную обработку и анализ данных. Каждый класс и его атрибуты играют ключевую роль в реализации функциональности системы, что позволяет достичь поставленных в ВКР целей, направленных на создание аналитического инструмента для прогнозирования спроса.

2.3 Выбор системы управления базами данных (СУБД)

В данном разделе рассматривается выбор СУБД для хранения данных, необходимых для анализа и прогнозирования спроса. Основные критерии выбора базы данных - это производительность, надежность и совместимость с системой аналитики. Выбор СУБД позволяет обеспечить эффективное управление данными, что необходимо для выполнения задач по созданию приложения для анализа спроса. Применение СУБД, поддерживающей обработку больших объемов данных, способствует точному анализу и прогнозированию.

Системы управления базами данных (СУБД) являются основополагающим элементом современной информационной инфраструктуры, обеспечивая надежное хранение, обработку и анализ данных [13]. Эти системы применяются в организациях всех масштабов - от малых предприятий до крупных транснациональных корпораций - и обеспечивают эффективное управление данными и доступ к ним. СУБД представляют собой специализированные программные платформы, которые обеспечивают структурирование, организацию, защиту и управление данными в пределах организации. Они выполняют важную роль посредника между пользователями и данными, предоставляя мощные инструменты, такие как SQL-запросы, индексация, а также средства для резервного копирования и восстановления, что делает их незаменимыми для поддержки как оперативного, так и стратегического управления.

СУБД выполняют ключевую функцию по обеспечению надежного и безопасного хранения данных, включая использование таких методов, как шифрование и механизмы контроля доступа. Кроме того, они предоставляют удобный и быстрый доступ к данным. Это включает возможности для поиска нужной информации, выполнения сложных аналитических вычислений, а также управление правами доступа на различных уровнях. Существенным аспектом является способность СУБД оптимизировать доступ к данным, минимизируя использование вычислительных ресурсов и сокращая время обработки. Это особенно важно для высоконагруженных систем и приложений, требующих высокой производительности. Оптимизация достигается за счет таких механизмов, как индексация, оптимизация запросов и кэширование данных.

Выбор подходящей системы управления базами данных - это стратегически значимое решение, которое оказывает прямое влияние на эффективность информационной системы организации. На рынке представлено множество различных СУБД, таких как MySQL, MongoDB, Redis и PostgreSQL, каждая из которых обладает уникальными

характеристиками и предназначена для решения конкретных задач. В зависимости от требований проекта, можно выбирать между различными типами СУБД: реляционными, NoSQL, In-Memory, распределёнными и другими. Каждая из этих систем имеет свои преимущества с точки зрения производительности, гибкости и масштабируемости, что делает её оптимальной для различных сценариев использования [14].

Реляционные СУБД являются одним из самых популярных типов систем управления базами данных, которые построены на основе реляционной модели. В этой модели данные хранятся в виде таблиц, состоящих из строк и столбцов, где каждая строка представляет собой запись, а столбец - отдельный атрибут данных. Реляционные СУБД обеспечивают целостность данных, что особенно важно при работе с большими массивами информации и при необходимости обеспечения строгих правил консистентности. Основными достоинствами реляционных СУБД являются поддержка языка SQL, возможность нормализации данных и широкие возможности для создания сложных запросов.

Другой класс СУБД, который стал популярным в последние годы, - это noSQL системы. Они разрабатывались для работы с неструктурированными данными, которые сложно разместить в традиционных таблицах. В отличие от реляционных СУБД, noSQL системы предлагают более гибкую структуру хранения информации, что особенно важно для приложений, работающих с большими объемами данных, таких как веб-сервисы, социальные сети и системы с высокими нагрузками. Эти СУБД поддерживают такие форматы данных, как JSON, XML, BSON, и могут использовать различные подходы к хранению: от документо-ориентированных баз данных до ключ-значение хранилищ и графовых баз данных [17].

In-Memory СУБД заслуживают отдельного внимания, так как они хранят данные в оперативной памяти, что значительно увеличивает скорость доступа и обработки. Эти СУБД особенно востребованы в случаях, когда критична скорость обработки данных в реальном времени, например, в финансовых

системах, где миллисекунды могут быть решающими для успешного выполнения операций. Хотя хранение данных в оперативной памяти делает их уязвимыми при сбоях, современные In-Memory решения часто используют гибридные подходы, сочетающие хранение в памяти и на диске для повышения надёжности [26].

В рамках нашего проекта наибольший интерес представляют реляционные СУБД, так как они обеспечивают высокую степень структурированности данных и их целостность, что особенно важно при построении сложных информационных систем с множеством взаимосвязанных элементов. Мы рассмотрим несколько ключевых кандидатов для использования в системе: Oracle, MySQL и PostgreSQL. Эти системы являются лидерами рынка и широко используются в различных сферах, включая банковский сектор, телекоммуникации, государственные службы и другие области, где требуются надёжные и масштабируемые решения для работы с большими массивами данных.

Для принятия обоснованного решения о выборе оптимальной системы управления базами данных (СУБД) для нашего проекта необходимо провести тщательный анализ требований и возможностей системы. Важным инструментом для такого анализа является диаграмма вариантов использования, которая входит в состав методологии UML (Unified Modeling Language).

Диаграмма вариантов использования наглядно демонстрирует все возможные сценарии взаимодействия пользователей с системой, что способствует более глубокому пониманию того, как функционируют и взаимодействуют различные компоненты системы. Это позволяет выявить потенциальные проблемы и улучшить архитектуру системы, обеспечивая ее соответствие требованиям проекта и упрощая процесс разработки.

Ключевыми компонентами диаграммы вариантов использования являются:

Варианты использования: это конкретные функции или действия,

которые система должна выполнять. Они демонстрируют, какие задачи можно решить с помощью системы и как пользователи смогут взаимодействовать с её функционалом.

Актёры: это пользователи системы или внешние элементы, такие как другие системы или приложения, которые взаимодействуют с нашей базой данных. Актёры могут быть как людьми (конечными пользователями), так и программными компонентами.

Связи: это отношения между актёрами и вариантами использования, которые описывают, как именно пользователи будут взаимодействовать с системой и какие функции будут им доступны.

Использование диаграммы вариантов использования (Рисунок 13) позволяет нам лучше понять все потенциальные сценарии использования системы и определить, какая СУБД наиболее полно удовлетворит потребности проекта. Это позволяет избежать ошибок при выборе технологии и обеспечить оптимальную работу системы на всех этапах её функционирования, от разработки до эксплуатации [18].



Рисунок 13 – Диаграмма вариантов использования приложения

Визуализация данных на диаграмме указывает на потенциальную перегрузку системы управления базами данных (СУБД), что подчеркивает важность высокой эффективности обработки запросов. Чтобы обеспечить быструю реакцию СУБД на запросы клиентов, необходимо тщательно выбирать подходящую систему управления базами данных.

При выборе оптимальной СУБД следует учитывать ряд ключевых факторов, в том числе: гибкость системы, ее производительность, возможность нормализации данных, простоту использования, совместимость с другими приложениями и платформами, поддержку стандартов, безопасность, информационную структуру, открытый исходный код и масштабируемость.

Организация данных в виде таблиц позволяет создать эффективную систему управления, обеспечивающую легкий доступ к информации и ее обработку. В этом отношении PostgreSQL выделяется как надежное и производительное решение, предлагающее широкий спектр функциональных возможностей, помогающих повысить надежность и скорость обработки данных.

Для наглядности, давайте сравним PostgreSQL с другими популярными системами управления базами данных, такими как MySQL и Oracle, в таблице 3, представленной ниже.

Таблица 3 - Сравнение СУБД

Требование	MySQL	Oracle	PostgreSQL
Гибкость	Да	Да	Да
Эффективность	Да	Да	Да
Возможность нормализации	Да	Да	Да
Простота использования	Нет	Нет	Да
Совместимость	Да	Нет	Да
Поддержка стандартов	Да	Нет	Да
Безопасность	Да	Да	Да
Структуризация	Да	Нет	Да
OpenSource	Да	Нет	Да
Масштабируемость	Да	Нет	Да

Итог	9	5	10
------	---	---	----

По результатам анализа видно, что PostgreSQL демонстрирует выдающиеся показатели по всем критериям. Эта система управления базами данных не только обладает высокой гибкостью и производительностью, но и предлагает обширные возможности для нормализации данных.

PostgreSQL является реляционной базой данных с открытым исходным кодом, которая позволяет эффективно управлять и обрабатывать большие объемы информации. Ее поддержка языка SQL, а также дополнительные функции, такие как расширенные типы данных и поддержка многопоточности, делают PostgreSQL идеальным выбором для сложных приложений.

Благодаря своей архитектуре, PostgreSQL обеспечивает надежность и безопасность данных, что особенно важно для организаций, работающих с конфиденциальной информацией. Кроме того, PostgreSQL хорошо совместим с различными операционными системами, что позволяет легко интегрировать его в существующие ИТ-структуры.

Выбранная система управления базами данных отвечает всем требованиям проекта и обеспечивает эффективное хранение и доступ к данным. Это решение является важным шагом в выполнении задач ВКР по разработке системы для анализа и прогнозирования потребительского спроса.

2.4 Логическая и физическая модели базы данных

Логическая модель базы данных описывает основные сущности и их связи, которые необходимы для управления данными в системе. На рисунке 14 представлена логическая модель базы данных системы анализа и прогнозирования потребительского спроса на продукцию компании, а в таблице 4 - представлено описание атрибутов ее сущностей.

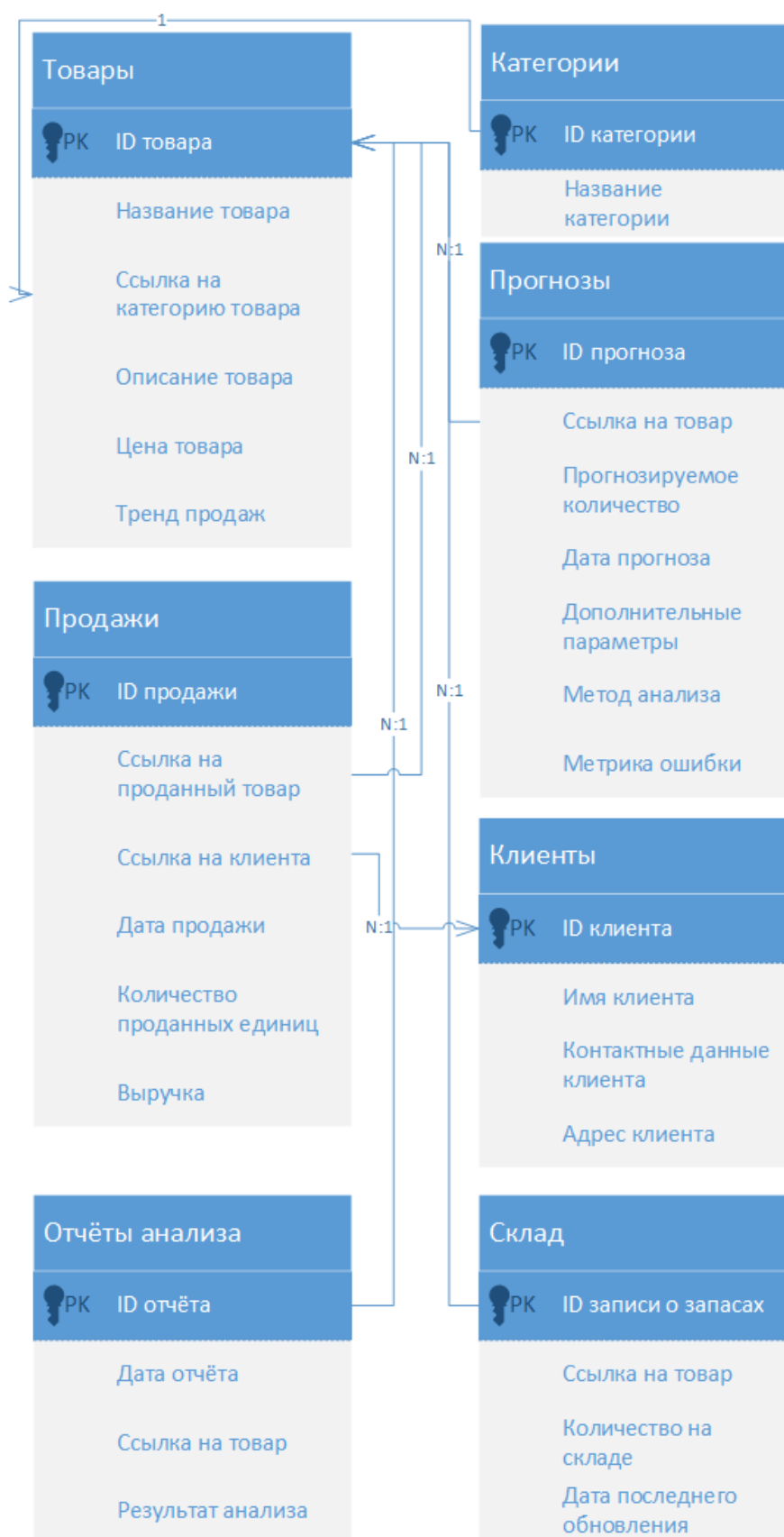


Рисунок 14 – Логическая модель базы данных

Таблица 4 - Атрибуты сущностей логической модели базы данных

Сущность	Атрибут	Описание, особенности использования
Товары	ID товара	Уникальный идентификатор товара, используется для связи с другими сущностями.
Товары	Название товара	Текстовое название товара, необходимое для идентификации продукта пользователями.
Товары	Ссылка на категорию товара	Указывает на категорию, к которой относится товар. Используется для классификации и фильтрации товаров.
Товары	Описание товара	Детальная информация о товаре, может содержать характеристики, преимущества и другие данные.
Товары	Цена товара	Указывает стоимость товара, используется для расчета общей выручки и предоставления информации клиентам.
Товары	Тренд продаж	Аналитический показатель, отражающий изменения спроса на товар (например, «растущий», «падающий»).
Категории	ID категории	Уникальный идентификатор категории, используется для связи с товарами.
Категории	Название категории	Название группы товаров, позволяет классифицировать товары (например, «Котлы», «Горелки»).
Продажи	ID продажи	Уникальный идентификатор транзакции, используется для учета продаж.
Продажи	Ссылка на проданный товар	Указывает на товар, который был продан, обеспечивает связь с сущностью «Товары».
Продажи	Ссылка на клиента	Указывает на покупателя, связана с сущностью «Клиенты».
Продажи	Дата продажи	Указывает дату и время продажи, используется для анализа временных данных.
Продажи	Количество проданных единиц	Показывает объем проданного товара, используется для расчета выручки и анализа спроса.
Продажи	Выручка	Общая сумма, полученная от продажи, рассчитывается как произведение количества и цены товара.
Клиенты	ID клиента	Уникальный идентификатор клиента, используется для связи с продажами.
Клиенты	Имя клиента	Имя покупателя, используется для персонализации и учета данных клиента.
Клиенты	Контактные данные клиента	Информация для связи с клиентом (например, телефон, email), используется для коммуникации и маркетинга.
Клиенты	Адрес клиента	Указывает местоположение клиента, может использоваться для доставки или статистического анализа.
Прогнозы	ID прогноза	Уникальный идентификатор записи прогноза, используется для учета и управления прогнозами.
Прогнозы	Ссылка на товар	Указывает на товар, для которого создан прогноз, обеспечивает связь с сущностью «Товары».
Прогнозы	Прогнозируемое количество	Ожидаемое количество продаж или потребления товара, используется для планирования.

Продолжение таблицы 4

Сущность	Атрибут	Описание, особенности использования
Прогнозы	Дата прогноза	Указывает дату, на которую составлен прогноз, используется для анализа временных данных.
Прогнозы	Дополнительные параметры	Дополнительная информация, связанная с прогнозом, например, предположения или ограничения.
Прогнозы	Метод анализа	Метод, использованный для составления прогноза, например, линейная регрессия или нейросетевой анализ.
Прогнозы	Метрика ошибки	Показатель точности прогноза (например, MAE, RMSE), используется для оценки качества прогнозной модели.
Отчёты анализа	ID отчёта	Уникальный идентификатор записи отчета, используется для управления и поиска данных.
Отчёты анализа	Дата отчёта	Указывает дату составления отчета, используется для временного анализа.
Отчёты анализа	Ссылка на товар	Указывает на товар, для которого составлен отчет, связана с сущностью «Товары».
Отчёты анализа	Результат анализа	Хранит информацию о результатах анализа (например, текст или структурированные данные).
Склад	ID записи о запасах	Уникальный идентификатор записи о запасах, используется для учета данных на складе.
Склад	Ссылка на товар	Указывает на товар, запас которого учитывается, обеспечивает связь с сущностью «Товары».
Склад	Количество на складе	Указывает текущее количество товара на складе, используется для инвентаризации и планирования поставок.
Склад	Дата последнего обновления	Указывает дату последнего изменения записи, используется для контроля актуальности данных.

Физическая модель базы данных, представленная на Рисунке 15 определяет, как данные будут храниться в СУБД PostgreSQL, оптимизируя структуру для быстрого доступа и обработки.

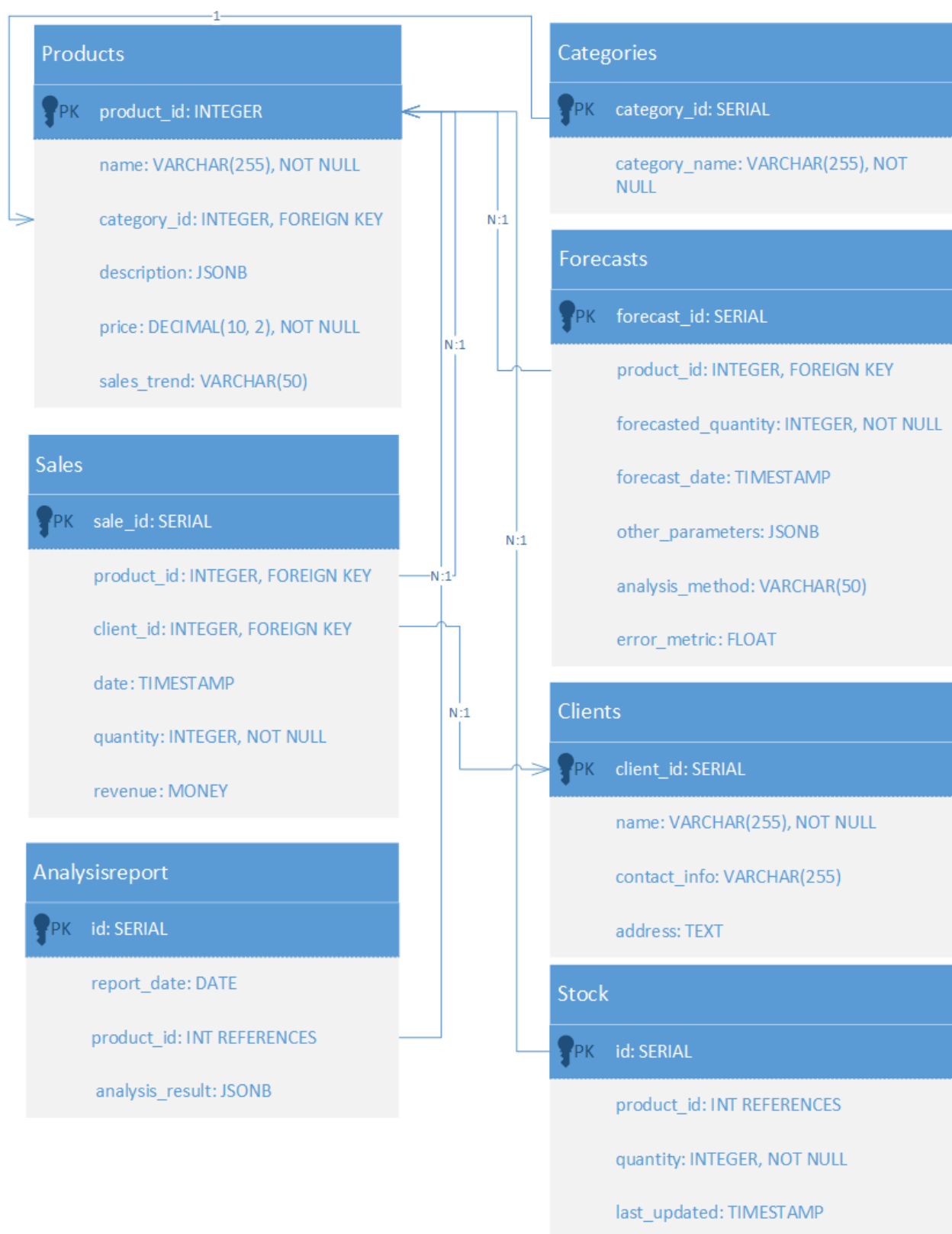


Рисунок 15 – Физическая модель базы данных

В таблице 5 представлена физическая модель и определены следующие элементы:

Таблицы: каждая сущность логической модели представлена отдельной таблицей с уникальными именами.

Ключи: первичные ключи (PK) используются для уникальной идентификации записей в каждой таблице.

Внешние ключи (FK) обеспечивают связь между таблицами.

Типы данных: для каждого атрибута таблицы определен тип данных (например, INTEGER, VARCHAR, JSONB), что оптимизирует хранение информации.

Ограничения: реализованы механизмы проверки данных, такие как NOT NULL, UNIQUE, и ограничения CHECK, чтобы поддерживать целостность данных.

Таблица 5 - Описание элементов физической модели базы данных

Таблица	Атрибут	Ключ	Тип данных	Ограничение	Связи
Products	product_id	PK	SERIAL	NOT NULL	-
Products	name	-	VARCHAR(255)	NOT NULL	-
Products	category_id	FK	INTEGER	-	Categories. category_id
Products	description	-	JSONB	-	-
Products	price	-	DECIMAL(10, 2)	NOT NULL	-
Products	sales_trend	-	VARCHAR(50)	-	-
Categories	category_id	PK	SERIAL	NOT NULL	-
Categories	category_name	-	VARCHAR(255)	NOT NULL, UNIQUE	-
Sales	sale_id	PK	SERIAL	NOT NULL	-
Sales	product_id	FK	INTEGER	-	Products. product_id
Sales	client_id	FK	INTEGER	-	Clients. client_id
Sales	date	-	TIMESTAMP	NOT NULL	-
Sales	quantity	-	INTEGER	NOT NULL, CHECK(quantity > 0)	-
Sales	revenue	-	MONEY	NOT NULL	-
Clients	client_id	PK	SERIAL	NOT NULL	-

Продолжение таблицы 5

Таблица	Атрибут	Ключ	Тип данных	Ограничение	Связи
Clients	name	-	VARCHAR(255)	NOT NULL	-
Clients	contact_info	-	VARCHAR(255)	-	-
Clients	address	-	TEXT	-	-
Forecasts	forecast_id	PK	SERIAL	NOT NULL	-
Forecasts	product_id	FK	INTEGER	-	Products. product_id
Forecasts	forecasted_quantity	-	INTEGER	NOT NULL, CHECK (forecasted_quantity >= 0)	-
Forecasts	forecast_date	-	TIMESTAMP	NOT NULL	-
Forecasts	other_parameters	-	JSONB	-	-
Forecasts	analysis_method	-	VARCHAR(50)	-	-
Forecasts	error_metric	-	FLOAT	-	-
Analysis report	id	PK	SERIAL	NOT NULL	-
Analysis report	report_date	-	DATE	NOT NULL	-
Analysis report	product_id	FK	INTEGER	-	Products. product_id
Analysis report	analysis_result	-	JSONB	-	-
Stock	id	PK	SERIAL	NOT NULL	-
Stock	product_id	FK	INTEGER	-	Products. product_id
Stock	quantity	-	INTEGER	NOT NULL, CHECK(quantity >= 0)	-
Stock	last_updated	-	TIMESTAMP	NOT NULL	-

Особенности SQL-диалекта PostgreSQL

PostgreSQL поддерживает специфические типы, такие как JSONB (для хранения структурированных данных), MONEY (для денежных значений) и ARRAY (для массивов). Для работы с датами и временем предоставляются типы TIMESTAMP, DATE, TIME с учетом временных зон. PostgreSQL позволяет хранить, извлекать и фильтровать данные в формате JSON/JSONB, что упрощает обработку сложных структурированных данных. Помимо стандартных B-деревьев PostgreSQL поддерживает индексы на основе GIN и GiST, что повышает производительность работы с массивами и JSONB.

PostgreSQL предоставляет гибкие механизмы валидации данных, такие как ограничения CHECK, NOT NULL, UNIQUE, а также внешние ключи (FOREIGN KEY). Возможность создания собственных функций на SQL, PL/pgSQL, Python или других языках для сложных вычислений. PostgreSQL поддерживает триггеры, которые автоматически выполняют заданные действия при изменении данных. Примеры SQL-кодов запросов представлены на рисунке 16.

```
/*Создание таблиц*/
CREATE TABLE Products (
    product_id SERIAL PRIMARY KEY,
    name VARCHAR(255) NOT NULL,
    category_id INTEGER REFERENCES Categories(category_id),
    description JSONB,
    price DECIMAL(10, 2) NOT NULL,
    sales_trend VARCHAR(50)
);

CREATE TABLE Categories (
    category_id SERIAL PRIMARY KEY,
    category_name VARCHAR(255) UNIQUE NOT NULL
);

/*Запрос с объединением таблиц*/
SELECT p.name AS product_name, c.category_name, p.price
FROM Products p
JOIN Categories c ON p.category_id = c.category_id
WHERE p.price > 100.00;

/*Извлечение значений из JSONB-поля*/
SELECT description->>'key' AS value
FROM Products
WHERE description ? 'key';

/*Добавление триггера для автоматического обновления времени*/
CREATE OR REPLACE FUNCTION update_modified_column()
RETURNS TRIGGER AS $$
BEGIN
    NEW.updated_at = NOW();
    RETURN NEW;
END;
$$ LANGUAGE plpgsql;

CREATE TRIGGER set_timestamp
BEFORE UPDATE ON Products
FOR EACH ROW
EXECUTE FUNCTION update_modified_column();
```

Рисунок 16 - Примеры SQL-кодов запросов, применяемых в PostgreSQL

Выводы к главе 2: нами был проведен детальный анализ современных инструментов, применяемых в задачах анализа данных и прогнозирования, с особым вниманием к их ключевым особенностям и преимуществам. В результате этого исследования были выявлены программные средства, которые являются критически важными для эффективной разработки нашего приложения. Этот этап стал важной вехой в процессе создания комплексной системы, способной полностью удовлетворить требования нашего проекта.

Мы сформулировали статическую структуру системы и использовали диаграмму классов для иллюстрации ключевых компонентов нашей разработки. Атрибуты и методы каждого класса были тщательно детализированы, что помогает обеспечить надежность и эффективность нашего программного решения, одновременно улучшая наше понимание объектно-ориентированной архитектуры системы.

Также, мы разработали диаграмму вариантов использования, которая позволила нам четко сформулировать функциональные требования системы, определив, как различные пользователи будут взаимодействовать с ней. Это понимание жизненно важно для успешного выполнения проекта, поскольку оно описывает функциональные возможности, которые будут наиболее востребованы конечными пользователями.

Благодаря тщательному анализу и сравнению мы выбрали наиболее подходящую систему управления базами данных. На это решение повлияло несколько факторов, включая производительность, безопасность и масштабируемость, что обеспечило создание надежного программного решения.

Глава 3 Физическое проектирование и реализация приложения

3.1 Разработка информационной системы анализа и прогнозирования потребительского спроса на продукцию компании

Для создания данного программного обеспечения была выбрана интегрированная среда разработки (IDE) PyCharm Community Edition 2024 от компании JetBrains. Эта бесплатная IDE является одной из лучших для разработки на Python, предлагая множество полезных инструментов и функций.

PyCharm Community Edition 2024 предоставляет широкий набор возможностей, таких как интеллектуальное автодополнение кода, что существенно ускоряет процесс разработки и минимизирует количество ошибок. Благодаря интеллектуальным инструментам, большому количеству плагинов и множеству полезных функций, данная среда разработки была признана оптимальным выбором для реализации нашего проекта.

Краткое описание шагов разработки проекта:

- создание структуры проекта;
- настройка и заполнение базы данных;
- предобработка данных;
- прогнозирование с использованием модели случайного леса;
- анализ временных рядов;
- оценка точности модели;
- подготовка отчетов;
- формирование отчетов.

Детальное описание выполняемых шагов процесса разработки будет представлено ниже по тексту.

Первый этап разработки предполагает создание структуры проекта, которую можно увидеть на рисунке 17.

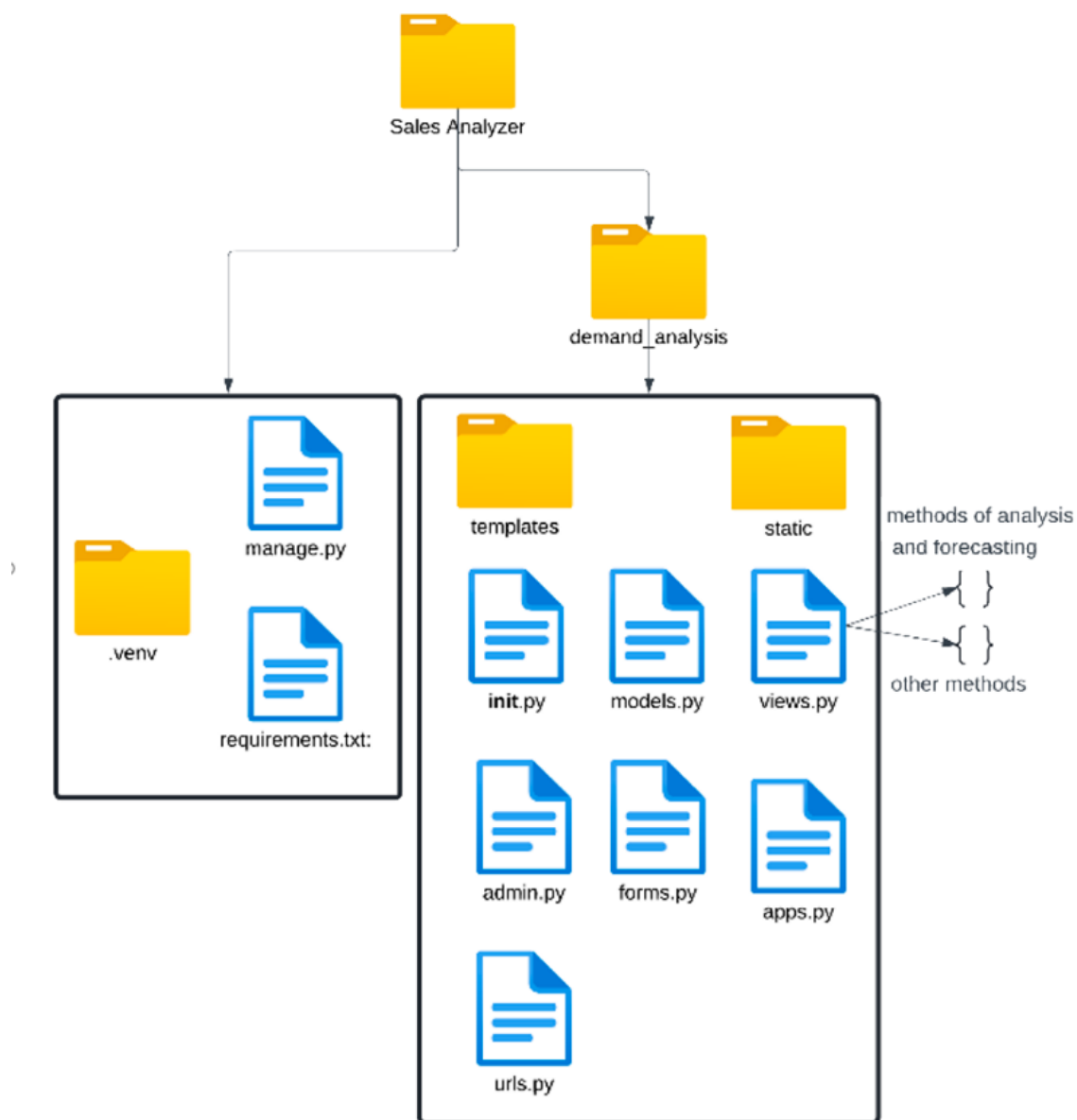


Рисунок 17 – Структура проекта

Проект включает в себя две ключевые директории: «demand_analysis» и «sales_analyzer». В первой директории хранятся все необходимые шаблоны и статические файлы, такие как стили CSS и скрипты JavaScript, которые обеспечивают визуальное представление приложения. Во второй директории сосредоточен весь исходный код, включая модули для обработки данных, а также модели и представления, отвечающие за логику приложения.

Вторым шагом в разработке является настройка и заполнение базы данных, которая будет использоваться в приложении. В качестве системы управления базами данных была выбрана PostgreSQL благодаря ее высоким

показателям производительности и поддержке сложных запросов. Для работы с базой данных использовался инструмент «pgAdmin», который позволяет визуализировать структуру базы данных и выполнять SQL-запросы.

Структура базы данных представлена таблицей, содержащей важную информацию о продажах, включая столбцы с указанием даты, продукта, количества и выручки. Такая настройка облегчает эффективную организацию и анализ данных.

Для интеграции приложения с базой данных применялась библиотека `psycopg2`, которая обеспечивает удобный интерфейс для работы с PostgreSQL. Кроме того, для анализа и обработки данных использовалась библиотека `Pandas`, обеспечивающая высокую эффективность при работе с данными в формате `DataFrame` и упрощение выполнения сложных операций и манипуляций.

Категориальные данные, участвующие в проекте, были преобразованы в числовой формат посредством метода `One-Hot Encoding`. Такой подход позволяет создать бинарное представление для каждого уникального признака, предотвращая ошибки, связанные с неверной интерпретацией категориальных данных как упорядоченных, и, таким образом, повышает точность анализа.

После выполнения подготовки данные были разделены на целевую переменную и независимые признаки. Целевая переменная представляет значение, которое необходимо предсказать с использованием алгоритмов машинного обучения [21]. Основная идея заключается в обучении моделей на основе исторических данных, что позволяет алгоритмам делать точные прогнозы с поддержкой аналитического контроля.

Для выполнения прогнозирования была выбрана модель случайного леса, которая показала свою высокую эффективность в работе с непрерывными данными. Случайный лес строит множество деревьев решений на различных подмножествах данных и затем объединяет их результаты. На каждом этапе алгоритм случайным образом выбирает подмножество

признаков, что способствует повышению устойчивости модели и снижению вероятности переобучения. Подробное описание реализации модели случайного леса будет представлено в проектной документации.

Структура проекта организована в двух основных каталогах: «demand_analysis» и «sales_analyzer». Каталог «demand_analysis» включает все необходимые шаблоны и статические файлы, такие как CSS-стили и JavaScript-скрипты, обеспечивающие визуальные компоненты пользовательского интерфейса. Каталог «sales_analyzer» содержит исходный код, модули для обработки данных, а также модели и представления, которые являются основными элементами бизнес-логики приложения.

Следующий этап разработки предполагает настройку и заполнение базы данных, необходимой для полноценного функционирования приложения. В качестве системы управления базами данных была выбрана PostgreSQL, которая характеризуется высокой производительностью и способностью эффективно обрабатывать сложные SQL-запросы. Для управления базой данных используется pgAdmin, который предоставляет удобный интерфейс для визуализации структуры базы и выполнения различных операций.

В базе данных была создана таблица ключевых показателей эффективности, в которой отражена важная информация о продажах, такая как дата, название продукта, количество и выручка. Такая организация позволит более эффективно управлять данными и упростит анализ.

Для обеспечения надежного взаимодействия с базой данных использовалась библиотека psycopg2, которая предоставляет мощные инструменты для интеграции с PostgreSQL. Библиотека Pandas, в свою очередь, применялась как основной инструмент для управления данными, включая их обработку и анализ.

Для преобразования категориальных данных в числовой формат, который может быть обработан алгоритмами машинного обучения, применялся метод One-Hot Encoding. Этот метод создает бинарные представления для каждого уникального категориального признака, что

позволяет избежать некорректной интерпретации категориальных данных как ранжированных. В результате обеспечивается корректная работа алгоритмов и повышается точность аналитических прогнозов.

После предварительной обработки данные в формате DataFrame были разделены на целевую переменную и независимые признаки. Целевая переменная представляет собой показатель, который необходимо предсказать с использованием методов машинного обучения. Такое разбиение позволяет моделям эффективно обучаться на исторических данных, что способствует повышению точности прогнозов и помогает принимать более обоснованные управленческие решения.

Для анализа временных рядов была выбрана модель экспоненциального сглаживания (Exponential Smoothing), что позволяет учитывать как исторические данные, так и тренды, обеспечивая более точные прогнозы. Данный метод подходит для анализа данных с сезонными колебаниями и эффективно используется для предсказания будущих показателей на основе имеющихся данных. Реализация метода экспоненциального сглаживания будет продемонстрирована на следующих этапах проекта, что позволит оценить его эффективность в контексте анализа продаж.

Метод экспоненциального сглаживания применяется для создания прогноза на основе исторических данных, придавая больший вес более свежим данным. Это позволяет учитывать изменения спроса и адаптировать прогнозы к текущим рыночным условиям. Данный метод особенно эффективен для временных рядов с краткосрочными колебаниями и умеренными трендами.

Настройка параметров: коэффициент сглаживания выбирается экспериментально, в зависимости от характера данных. Высокие значения делают прогноз чувствительным к последним данным, а низкие - создают более устойчивый прогноз. Реализация модели показана на рисунке 18.

```

if sales_data.exists():
    df = pd.DataFrame(list(sales_data.values('date', 'quantity_sold')))
    df['date'] = pd.to_datetime(df['date'])
    df.set_index(keys='date', inplace=True)

    model = ExponentialSmoothing(df['quantity_sold'], trend='add', seasonal='add', seasonal_periods=12)
    fit = model.fit()
    forecast = fit.forecast(forecast_period)

    last_date = df.index[-1]
    future_dates = pd.date_range(last_date, periods=forecast_period + 1, freq='M')[1:]

    forecast_df = pd.DataFrame({'date': future_dates, 'forecast': forecast.values})
    forecast_df.set_index(keys='date', inplace=True)

```

Рисунок 18 – Реализация модели «Exponential Smoothing»

Для реализации анализа временных рядов использовалась библиотека «statsmodels», в частности, метод экспоненциального сглаживания (Exponential Smoothing). Данные были подготовлены и разбиты на различные выборки для анализа и прогноза.

В системе также использованы следующие элементы программного кода связанные с анализом и прогнозированием:

Функция `seasonal_decompose` (Рисунок 19) разбивает временные ряды на три основные компоненты: тренд, сезонность и остатки. Используется для анализа поведения данных о продажах. Позволяет выявить долгосрочные тренды и циклические изменения, а также определить влияние сезонных факторов на продажи.

```

if not df.empty:
    seasonal_decomposition = seasonal_decompose(df['quantity_sold'], model='additive', period=12)

    plt.figure(figsize=(10, 8))

    plt.subplot(3, 1, 1)
    plt.plot(seasonal_decomposition.trend, label='Trend', color='blue')
    plt.title('Тренд Продаж')
    plt.legend()

    plt.subplot(3, 1, 2)
    plt.plot(seasonal_decomposition.seasonal, label='Seasonal', color='green')
    plt.title('Сезонные Колебания')
    plt.legend()

    plt.subplot(3, 1, 3)
    plt.plot(seasonal_decomposition.resid, label='Residual', color='red')
    plt.title('Остатки')
    plt.legend()

    plt.tight_layout()

    buffer = BytesIO()
    plt.savefig(buffer, format='png')
    plt.close()
    buffer.seek(0)
    seasonal_decomposition_chart = base64.b64encode(buffer.getvalue()).decode('utf-8')

    return render(request, 'seasonality_analysis_result.html', {
        'seasonal_decomposition_chart': seasonal_decomposition_chart,
        'selected_product': product
    })

```

Рисунок 19 – Сезонная декомпозиция

Корреляционный анализ (Рисунок 20) позволяет провести построение корреляционной матрицы для числовых данных, таких как количество проданных единиц и выручка. Помогает понять взаимосвязь между различными показателями. Например, как изменение количества проданных единиц влияет на выручку. Это важно для выявления ключевых факторов, влияющих на успех продукта.

```
plt.figure(figsize=(8, 6))
numeric_data = df[['quantity_sold', 'revenue', 'product__price']]
correlation_matrix = numeric_data.corr()

sns.heatmap(correlation_matrix, annot=True, cmap='coolwarm', vmin=-1, vmax=1)
plt.title('Матрица корреляции')
plt.tight_layout()

buffer = BytesIO()
plt.savefig(buffer, format='png')
plt.close()
buffer.seek(0)
correlation_matrix_chart = base64.b64encode(buffer.getvalue()).decode('utf-8')
context['correlation_matrix_chart'] = correlation_matrix_chart
```

Рисунок 20 – Корреляционный анализ

Непосредственно по отношению к анализу и прогнозированию указаны пути (Рисунок 21) для функций, например, /forecast/ для построения прогнозов или /seasonality_analysis/ для анализа сезонности. Эти маршруты позволяют пользователям взаимодействовать с функциями анализа и прогнозирования через интерфейс приложения.

```
# demand_analysis/urls.py
from django.urls import path
from .views import index, add_product, search_products, add_sales_record,
from .views import forecast, analyze_demand, category_analysis, seasonality_analysis, sales_comparison

urlpatterns = [
    path('', index, name='index'), # Home page
    path('add_product/', add_product, name='add_product'), # Add product page
    path('add_sales_record/', add_sales_record, name='add_sales_record'), # Add sales record page
    path('analyze_demand/', analyze_demand, name='analyze_demand'), # Analyze demand page
    path('category_analysis/', category_analysis, name='category_analysis'),
    path('forecast/', forecast, name='forecast'), # Category analysis page
    path('seasonality_analysis/', seasonality_analysis, name='seasonality_analysis'), # Seasonality analysis page
    path('sales_comparison/', sales_comparison, name='sales_comparison'), # Sales comparison page
    path('search/', search_products, name='search_products'), # Search products page
```

Рисунок 21 – Маршруты используемые в системе

Следующий программный элемент (Рисунок 22) анализирует данные о продажах, сгруппированные по категориям продуктов. Также определяет

топовые категории или продукты. Дает представление о том, какие категории продуктов наиболее популярны или приносят максимальную прибыль.

```
if not df.empty:
    sales_overview = df.groupby(df['date'].dt.to_period('M'))['quantity_sold'].sum()
    plt.figure(figsize=(10, 5))
    sales_overview.plot(kind='bar', color='skyblue')
    plt.title('Общий объем продаж по месяцам')
    plt.xlabel('Месяц')
    plt.ylabel('Количество проданных единиц')
    plt.xticks(rotation=45)
    plt.tight_layout()

    buffer = BytesIO()
    plt.savefig(buffer, format='png')
    plt.close()
    buffer.seek(0)
    sales_chart = base64.b64encode(buffer.getvalue()).decode('utf-8')
    context['sales_overview_chart'] = sales_chart

    top_products = df.groupby('product_name')['quantity_sold'].sum().nlargest(5)
    plt.figure(figsize=(10, 5))
    top_products.plot(kind='bar', color='green')
    plt.title('Топ продуктов по количеству продаж')
    plt.xlabel('Продукт')
    plt.ylabel('Количество проданных единиц')
    plt.xticks(rotation=45)
    plt.tight_layout()

    buffer = BytesIO()
    plt.savefig(buffer, format='png')
    plt.close()
    buffer.seek(0)
    top_products_chart = base64.b64encode(buffer.getvalue()).decode('utf-8')
    context['top_products_chart'] = top_products_chart

    category_sales = df.groupby('product_category_name')['quantity_sold'].sum()
    plt.figure(figsize=(10, 5))
    category_sales.plot(kind='pie', autopct='%1.1f%%')
    plt.title('Распределение продаж по категориям')
    plt.ylabel('')
    plt.tight_layout()

    buffer = BytesIO()
    plt.savefig(buffer, format='png')
    plt.close()
    buffer.seek(0)
    category_sales_chart = base64.b64encode(buffer.getvalue()).decode('utf-8')
    context['category_sales_chart'] = category_sales_chart

    monthly_sales = df.groupby(df['date'].dt.month)['quantity_sold'].sum()
    months = [calendar.month_abbr[i] for i in range(1, 13)]
    monthly_sales = monthly_sales.reindex(range(1, 13), fill_value=0)

    plt.figure(figsize=(10, 5))
    plt.plot(months, monthly_sales, marker='o', linestyle='-', color='orange')
    plt.title('Сезонные тенденции продаж')
    plt.xlabel('Месяц')
    plt.ylabel('Количество проданных единиц')
    plt.grid(True)
    plt.tight_layout()
```

Рисунок 22 – Анализ продаж по категориям

Ниже представлено описание используемых переменных:

monthly_revenue - агрегированные данные по выручке, которые используются для выявления трендов и сезонных колебаний. Они формируются на основе исторических данных о продажах, собранных за определенный период времени.

forecast - предсказанные значения, полученные на основе модели экспоненциального сглаживания. Эти значения представляют собой прогнозируемую выручку на будущие месяцы.

historical_data - исходные данные, использованные для анализа, включающие фактические значения выручки за прошлые месяцы.

В процессе подготовки модели был установлен параметр seasonal_periods, который определяет количество периодов, используемых для выявления сезонных паттернов в данных. Увеличение этого параметра может улучшить качество прогнозов, однако при этом возрастает вычислительная нагрузка на систему.

Для оценки точности модели использовалась метрика средней абсолютной ошибки (MAE). MAE измеряет точность прогнозирования как среднее абсолютное отклонение между предсказанными и фактическими значениями. Данный метод служит важным инструментом для анализа соответствия прогнозов реальным данным, предоставляя всестороннюю оценку эффективности модели, так как позволяет выявить среднюю величину ошибок и оценить их влияние на точность прогнозирования.

Заключительным этапом разработки является подготовка отчетов, включающих результаты, полученные в ходе прогнозирования. Модель анализирует исторические данные о продажах, сезонности и изменениях в рыночных условиях и, на их основе, осуществляет прогнозирование будущих показателей. Итогом работы модели является комплексный отчет. Он содержит прогноз по выручке, визуализацию результатов в виде графиков и данные о средней абсолютной ошибке. Такой отчет предоставляет пользователю исчерпывающее понимание результатов анализа и точности прогнозирования, что способствует принятию более обоснованных

управленческих решений.

3.2 Архитектура системы, методы анализа и прогнозирования

В данном разделе описаны ключевые этапы разработки системы анализа и прогнозирования потребительского спроса, которые включают архитектуру системы, ее модули, модели данных, используемые алгоритмы, а также последовательность действий пользователей при взаимодействии с приложением.

Система построена по модульной архитектуре, что позволяет распределить функциональность на несколько отдельных компонентов, каждый из которых отвечает за выполнение определенных задач.

Модуль управления данными: обеспечивает сбор, очистку, трансформацию и хранение данных в базе данных. Он отвечает за подключение к СУБД PostgreSQL и за передачу данных в аналитический модуль.

Аналитический модуль: основной компонент для обработки данных, включающий реализацию алгоритмов анализа и прогнозирования. Этот модуль отвечает за выполнение расчетов, обработку временных рядов и генерацию прогнозов на основе исторических данных.

Модуль визуализации и отчетности: формирует графики и таблицы на основе полученных данных, которые помогают пользователям визуально анализировать результаты прогноза и текущую динамику спроса.

Модуль взаимодействия с пользователем: реализует интерфейс пользователя, предоставляя доступ к функциям системы и обеспечивая простоту и удобство работы с системой.

Эта архитектура обеспечивает гибкость, масштабируемость и возможность быстрого внесения изменений, если потребности в анализе и прогнозировании изменятся.

Модели данных и подготовка данных для анализа: для обеспечения

корректного анализа и точного прогнозирования потребительского спроса система использует исторические данные о продажах, поступающие из корпоративной базы данных, построенной на основе СУБД PostgreSQL. Эти данные собираются из различных источников внутри компании, таких как ERP-системы и отделы продаж, где они фиксируются в виде записей, содержащих важную информацию о товарах, категориях, датах продаж и количестве проданных единиц. Ниже описаны основные этапы подготовки данных.

Сбор данных: исторические данные собираются централизованно из корпоративной базы данных. Основные источники включают данные из ERP-систем компании и записи отдела продаж. Эти данные представляют собой транзакции, в которых указываются названия товаров, категории, даты продажи и объемы продаж. Сбор данных выполняется с использованием автоматизированных процедур, которые регулярно обновляют информацию в базе данных, чтобы гарантировать актуальность данных для анализа.

Очистка данных: на этом этапе система проверяет данные на наличие ошибок, пропусков и аномалий. Например, данные проходят проверку на дублирование, и в случае обнаружения повторяющихся записей они удаляются. Если в данных встречаются пропущенные значения (например, отсутствуют даты или количество проданных единиц), они обрабатываются с использованием методов заполнения, таких как средние значения или прогнозируемые величины, чтобы избежать искажения результатов анализа. Также проводится удаление записей с аномальными значениями, такими как нереалистично высокие или низкие объемы продаж, которые могут повлиять на точность прогнозов.

Нормализация и преобразование данных: данные подвергаются нормализации для того, чтобы они были совместимы с алгоритмами анализа. Числовые данные, такие как количество проданных единиц и сумма выручки, могут масштабироваться, чтобы исключить влияние разного масштаба показателей на аналитические модели. Категориальные данные, например

типы товаров и их категории, преобразуются в числовые значения с использованием методов кодирования, таких как One-Hot Encoding. Это необходимо для корректной работы алгоритмов машинного обучения, которые требуют числового представления данных. Дополнительно применяется фильтрация данных по ключевым параметрам, чтобы обеспечить точность анализа и прогнозирования.

Структурирование и хранение данных: после очистки и преобразования данные структурируются в базе данных в виде таблиц:

- таблица Products: содержит информацию о товарах, включая уникальный идентификатор, название, категорию и основные характеристики;
- таблица Sales: хранит данные о каждой продаже, включая идентификаторы товара и клиента, дату продажи, количество проданных единиц и сумму выручки;
- таблица Categories: содержит перечень категорий товаров с описанием, что позволяет эффективно группировать товары по сегментам;
- таблица Forecasts: содержит результаты прогнозирования, включая прогнозируемый спрос, период прогноза и другие параметры.

Данные хранятся в формате, оптимизированном для быстрого доступа и аналитических операций. Например, для ускорения выборки данных применяются индексы, которые настраиваются на ключевых полях, таких как `product_id` в таблице Sales и `category_id` в таблице Products.

Физическая модель хранения данных: данные хранятся на основе СУБД PostgreSQL, что обеспечивает надежное и масштабируемое управление большим объемом информации. Система поддерживает индексацию ключевых полей, что значительно ускоряет выполнение запросов и операций агрегации. Для оптимизации структуры данных применены следующие меры:

- индексация: созданы индексы на полях, которые наиболее часто используются в запросах, что ускоряет доступ к информации и

улучшает производительность системы;

- нормализация: данные распределены по связанным таблицам для уменьшения избыточности и обеспечения консистентности информации. Это предотвращает дублирование данных и упрощает управление базой;
- архивирование устаревших данных: данные, которые больше не актуальны для текущего анализа, перемещаются в архивные таблицы. Это помогает поддерживать высокую производительность базы данных и предотвращает перегрузку основной таблицы.

Использование данных в аналитическом модуле: после подготовки данные поступают в аналитический модуль, где применяются статистические методы, такие как экспоненциальное сглаживание и алгоритмы машинного обучения. Эти методы позволяют анализировать исторические тренды и формировать прогнозы спроса. Результаты анализа визуализируются в виде графиков и таблиц, что помогает пользователям оценивать динамику продаж и планировать будущие объемы производства и запасы товаров.

В разработанной системе анализа и прогнозирования потребительского спроса на продукцию компании реализован ряд методов и алгоритмов анализа и прогнозирования.

Метод экспоненциального сглаживания (Exponential Smoothing). Этот метод анализа временных рядов учитывает изменения в данных, придавая больший вес последним значениям, чтобы создать более точные прогнозы. Он подходит для данных с умеренными трендами и сезонными колебаниями. Реализован с использованием библиотеки statsmodels. Параметры, такие как коэффициент сглаживания α , выбираются экспериментально для наилучшей точности прогноза. Используется для прогнозирования выручки на будущие периоды на основе исторических данных.

Регрессионный анализ: линейная и множественная регрессия применяются для анализа влияния факторов, таких как цена и маркетинговые расходы, на объем продаж. Этот метод помогает построить модели,

прогнозирующие, как изменения в одной или нескольких переменных влияют на целевую переменную. Используются библиотеки Scikit-learn и Pandas для построения и обучения регрессионных моделей. Обработка данных включает нормализацию и разбиение на тренировочные и тестовые наборы.

Кластерный анализ (K-means): метод кластерного анализа используется для сегментации данных о клиентах и продуктах. Он помогает группировать данные на основе их схожести, что полезно для определения целевых сегментов рынка. Реализован с использованием Scikit-learn. Алгоритм классифицирует данные по определенному числу кластеров, выявляя группы с похожими характеристиками.

Случайный лес (Random Forest): алгоритм случайного леса строит несколько деревьев решений и объединяет их для получения более точных и стабильных прогнозов. Он устойчив к переобучению и хорошо справляется с анализом данных с множеством факторов. С помощью Scikit-learn построены модели, которые используют подмножества данных и случайный выбор признаков для повышения точности и снижения переобучения.

3.3 Алгоритм действия пользователей в системе анализа и прогнозирования потребительского спроса на продукцию компании

Взаимодействие пользователей с системой анализа и прогнозирования спроса организовано в несколько последовательных этапов, которые включают ввод данных, запуск анализа и получение прогнозов. Каждый этап работы с системой сопровождается визуальными элементами интерфейса, что облегчает работу пользователя и обеспечивает интуитивное управление функциями приложения.

На рисунке 23 представлена структура меню информационной системы. В центре находится главное меню, от которого отходят разделы: «Управление продажами», «Анализ спроса», «Генерация отчетов», «Управление клиентами», «Управление запасами» и «Администрирование». Каждый раздел

имеет подменю с функциональными элементами, связанными с задачами конкретного раздела, такими как добавление, редактирование, анализ или экспорт данных.

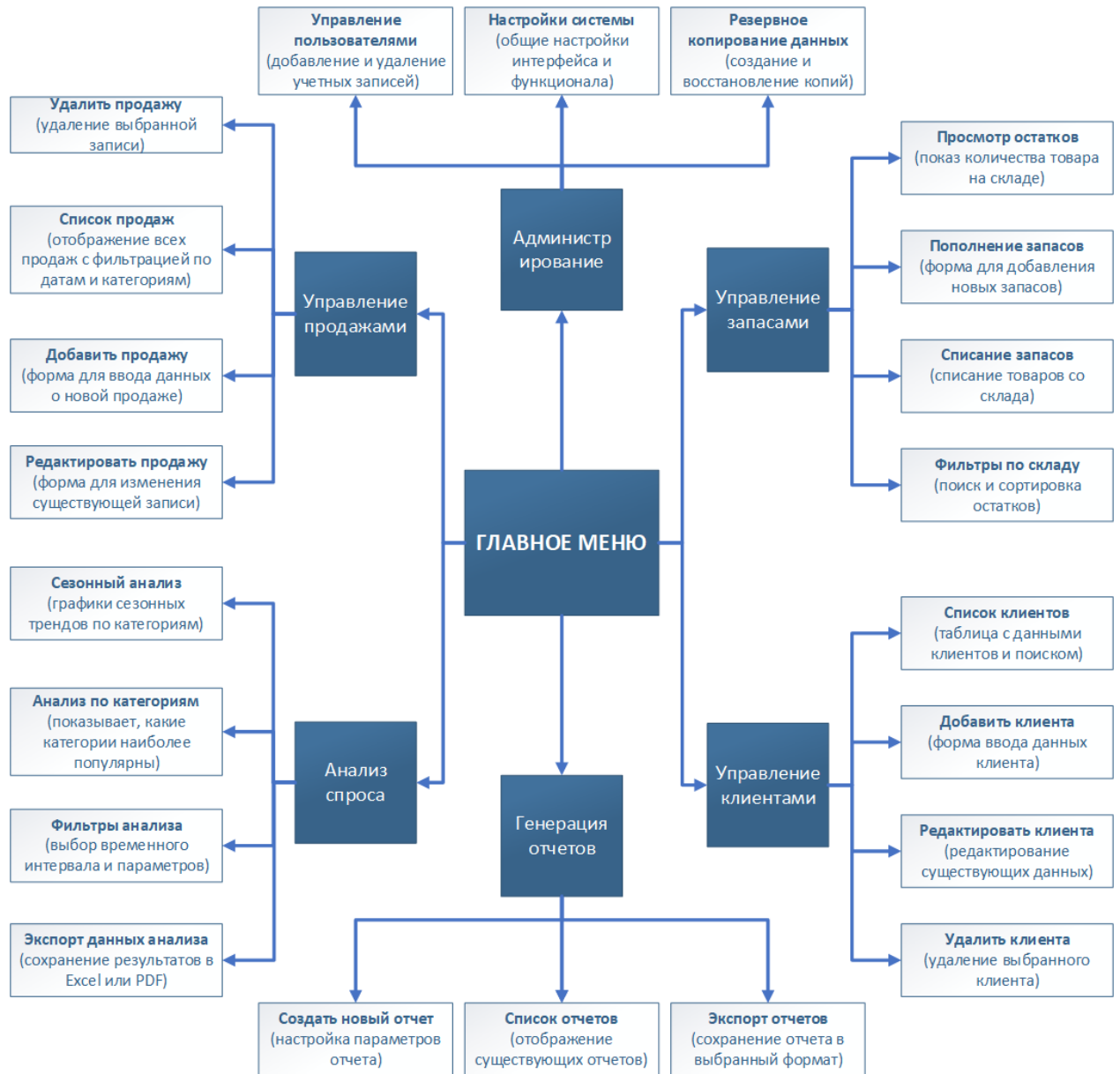


Рисунок 23 - Структурная схема меню информационной системы

Алгоритм действий администратора системы:

а) вход в систему:

1) перейти по адресу <http://localhost:8000/admin/>;

- 2) ввести логин и пароль для аутентификации;
- 3) получить доступ к основному меню панели администрирования.

б) управление группами:

- 1) открыть раздел «группы»;
- 2) добавить новую группу;
- 3) настроить права доступа для группы;
- 4) сохранить изменения.

в) управление пользователями:

- 1) зайти в раздел «пользователи»;
- 2) добавить нового пользователя:
 - ввести данные для аутентификации (логин, пароль);
 - назначить пользователя к одной или нескольким группам;
 - ввести персональные данные пользователя;
 - установить разрешения для пользователя.
- 3) редактировать существующих пользователей;
- 4) удалить ненужных пользователей.

г) управление данными:

- 1) внести изменения в клиентские записи;
- 2) добавить или обновить информацию о продуктах.

д) управлять данными о продажах:

- 1) добавить новые записи;
- 2) редактировать существующие данные;
- 3) удалить устаревшую информацию;
- 4) обновить данные о складе.

е) мониторинг активности:

- 1) просматривать список последних действий для контроля изменений в системе.

Алгоритм действий менеджера по продажам:

а) вход в систему:

- 1) запустить приложение;
 - 2) ввести логин и пароль.
- б) открыть главное окно программы с доступными разделами:
- 1) управление продажами;
 - 2) анализ спроса;
 - 3) генерация отчетов.
- в) добавление новой записи о продаже:
- 1) перейти в раздел «управление продажами»;
 - 2) нажать кнопку «добавить запись».
- г) заполнить форму:
- 1) товар: выбрать из выпадающего списка;
 - 2) клиент: выбрать из базы или ввести вручную;
 - 3) дата продажи: указать с помощью календарного виджета;
 - 4) количество проданных единиц: ввести значение;
 - 5) сумма выручки: проверяется или корректируется автоматически;
 - 6) нажать «сохранить», чтобы добавить данные в базу.
- д) редактирование записи:
- 1) выбрать нужную запись из списка;
 - 2) нажать «редактировать»;
 - 3) изменить данные в соответствующих полях;
 - 4) нажать «сохранить» для сохранения изменений.
- е) удаление записи:
- 1) выбрать запись в списке;
 - 2) нажать кнопку «удалить»;
 - 3) подтвердить удаление.
- ж) фильтрация и поиск записей:
- 1) использовать панель фильтрации для поиска записей по дате, товарам или клиентам;
 - 2) применять сортировку по выбранным критериям.

и) анализ данных о продажах:

- 1) перейти в раздел «анализ спроса»;
- 2) настроить параметры:
 - временной интервал;
 - фильтры по категориям.
- 3) нажать «запустить анализ»;
- 4) ознакомиться с результатами анализа: графиками, таблицами и рекомендациями.

к) экспорт отчетов:

- 1) перейти в раздел «генерация отчетов»;
- 2) выбрать параметры отчета;
- 3) нажать «экспорт», выбрать формат (pdf или excel);
- 4) сохранить файл на локальный компьютер.

Алгоритм действий аналитика

а) вход в систему:

- 1) запустить приложение;
- 2) ввести логин и пароль;
- 3) перейти в раздел «анализ спроса».

б) настройка параметров анализа:

- 1) выбрать временной интервал:
 - указать дату начала и окончания анализа.
- 2) применить фильтры по категориям:
 - указать определенные категории товаров для анализа;
 - уточнить дополнительные параметры.

в) запуск анализа:

- 1) нажать кнопку «запустить анализ»;
- 2) дождаться выполнения расчетов;
- 3) ознакомиться с результатами, представленными в виде:
 - интерактивных графиков;

- диаграмм и таблиц.

г) работа с результатами:

1) использовать инструменты управления графиками:

- увеличение определенных участков графиков;
- сравнение показателей между категориями;
- смена типа отображения данных (график/таблица).

2) оценить тенденции спроса, сезонные колебания и значимые закономерности.

д) генерация отчетов:

1) перейти в раздел «отчеты»;

2) выбрать интересующий отчет:

- например, прогноз продаж, статус запасов.

3) нажать «экспорт», выбрать формат (pdf или excel);

4) сохранить отчет на локальный компьютер.

е) корректировка данных:

1) при необходимости перейти в раздел «управление продажами»;

2) выбрать запись для редактирования;

3) внести корректировки и сохранить изменения.

ж) повторный анализ:

1) вернуться в раздел «анализ спроса»;

2) повторно запустить анализ для обновленных данных;

3) убедиться в точности новых прогнозов.

3.4 Разработка административной части системы анализа и прогнозирования потребительского спроса на продукцию компании

Административная часть системы представляет собой мощный инструмент для управления данными, пользователями и процессами внутри компании. Эта панель разработана с учётом удобства, безопасности и гибкости, что позволяет администраторам эффективно работать с

информацией и адаптировать систему к текущим потребностям бизнеса.

Главное меню панели администрирования (Рисунок 24): после аутентификации по адресу `http://localhost:8000/admin/`, администратор получает доступ к главному меню панели управления. Здесь представлены разделы для управления группами пользователей, учётными записями, клиентскими данными, продуктами, складом и аналитическими отчетами. Интуитивный дизайн меню упрощает навигацию и позволяет быстро переходить к нужным функциям.

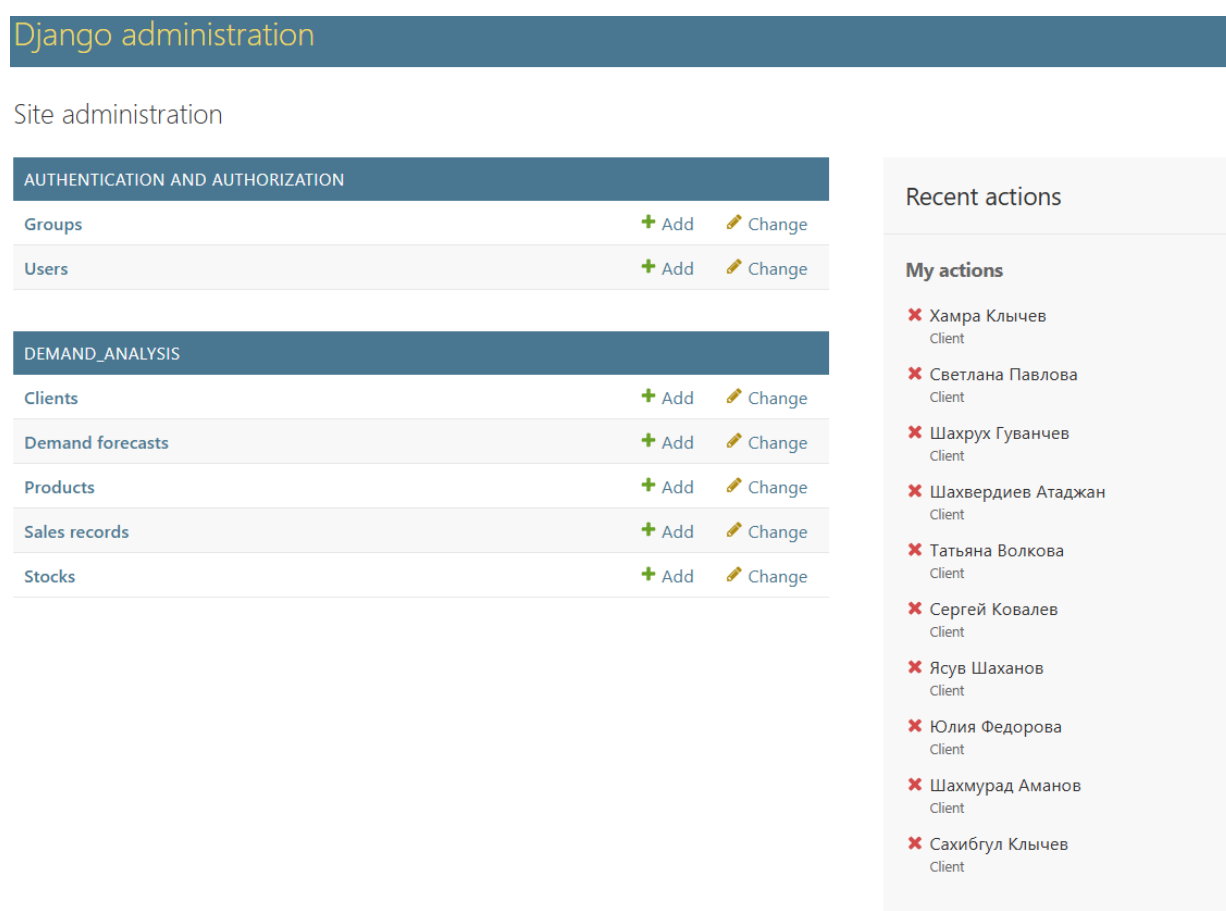


Рисунок 24 - Главное окно панели администрирования

Управление группами (Рисунок 25): в разделе «Группы» администратор может создавать новые группы пользователей и редактировать существующие. Для каждой группы можно детализировать права доступа:

например, определить, какие разделы системы будут доступны, а какие - ограничены. Это особенно важно для соблюдения внутренней безопасности и контроля над действиями сотрудников.

Add group

Name:

Permissions:

Available permissions ?

Administration | log entry | Can add log e
Administration | log entry | Can change l
Administration | log entry | Can delete lo
Administration | log entry | Can view log
Authentication and Authorization | group
Authentication and Authorization | group
Authentication and Authorization | group
Authentication and Authorization | group
Authentication and Authorization | perm
Authentication and Authorization | perm
Authentication and Authorization | perm

Choose all ?

Chosen permissions ?

Remove all

Hold down "Control", or "Command" on a Mac, to select more than one.

SAVE

Save and add another

Save and continue editing

Рисунок 25 - Раздел «Группы» панели администрирования

Управление пользователями (Рисунки 26 и 27): раздел «Пользователи» предоставляет возможности добавления новых сотрудников, обновления данных и удаления неактуальных учетных записей. Администратор может настроить для каждого пользователя: логин и пароль для доступа, принадлежность к одной или нескольким группам, персональные данные (например, имя, должность, контактную информацию), специфические разрешения, такие как доступ к конфиденциальным разделам. Эта гибкость помогает настраивать систему в зависимости от ролей сотрудников и их задач.

Change user

HISTORY

admin

Username:

admin

Required. 150 characters or fewer. Letters, digits and @/./+/-/_ only.

Password:

algorithm: pbkdf2_sha256 **iterations:** 870000 **salt:** YFPhKd***** **hash:** CXV/Qa*****

Reset password

Raw passwords are not stored, so there is no way to see the user's password.

Personal info

First name:

Last name:

Email address:

Permissions

☒ Active

Designates whether this user should be treated as active. Unselect this instead of deleting accounts.

☒ Staff status

Designates whether the user can log into this admin site.

☒ Superuser status

Designates that this user has all permissions without explicitly assigning them.

Рисунок 26 - Раздел «Пользователи» панели администрирования

Groups:

Available groups

Q

Filter

Choose all

Chosen groups

Q

Filter

Remove all

+

The groups this user belongs to. A user will get all permissions granted to each of their groups. Hold down "Control", or "Command" on a Mac, to select more than one.

User permissions:

Available user permissions

Q

Filter

Authentication and Au

Authentication and Au

Authentication and Au

Choose all

Chosen user permissions

Q

Filter

Remove all

Specific permissions for this user. Hold down "Control", or "Command" on a Mac, to select more than one.

Important dates

Last login:

Date:

2024-11-20

Today |

Time:

07:49:01

Now |

Note: You are 5 hours ahead of server time.

Date joined:

Date:

2024-10-20

Today |

Time:

11:31:49

Now |

Note: You are 5 hours ahead of server time.

SAVE

Save and add another

Save and continue editing

Delete

Рисунок 27 - Раздел «Пользователи» панели администрирования
(продолжение)

Работа с клиентскими данными (Рисунки 28 и 29): раздел «Клиенты»

85

предоставляет функции добавления, редактирования и удаления клиентских записей. Администратор может вносить изменения в контактные данные, просматривать историю взаимодействий и обновлять текущую информацию. Возможность редактировать детали клиентов вручную позволяет сохранять актуальность базы данных и устранять ошибки в информации.

Select client to change ADD CLIENT +

Action: Go 0 of 50 selected

☐	NAME	CONTACT INFO	ADDRESS
☐	Ясув Шаханов	+993 65 8765431	ул. Чарваглар 12, Дашогуз
☐	Юлия Федорова	+993 65 8765431	ул. Чарваглар 12, Дашогуз
☐	Шахрух Гуванчев	+993 65 7771234	ул. Ходжагулы 21, Балканабат
☐	Шахмурад Аманов	+993 65 5678902	ул. Ходжагулы 21, Балканабат
☐	Шахвердиев Атаджан	+993 65 5678902	ул. Туркменстанская 1, Туркменабад

Рисунок 28 - Раздел «Клиенты» панели администрирования

Change client HISTORY

Ясув Шаханов

Name:

Contact info:

Address:

SAVE Save and add another Save and continue editing Delete

Рисунок 29 - Раздел «Клиенты» панели администрирования (редактирование данных клиента)

Управление аналитическими данными (Рисунок 30): в разделе «Анализ» администратор имеет доступ к редактированию результатов анализа данных. Он может удалять устаревшие результаты, добавлять новые аналитические отчеты или корректировать уже существующие. Эта возможность позволяет гибко адаптировать систему к изменяющимся условиям рынка и обеспечивать актуальность данных.

Change demand forecast

Прогноз для Carrier 30RB 080 на 2025-02-22: 14 штук HISTORY

Product: ▼

Forecasted quantity:

Forecast date: Today | 

Note: You are 5 hours ahead of server time.

SAVE Save and add another Save and continue editing Delete

Рисунок 30 - Раздел «Анализ» панели администрирования

Управление продуктами (Рисунок 31): раздел «Продукты» включает инструменты для полного управления ассортиментом. Администратор может добавлять новые позиции, редактировать их характеристики (название, категорию, описание, стоимость) или удалять товары, которые больше не актуальны. Такая функциональность помогает поддерживать актуальность товарной базы и улучшать точность анализа потребительского спроса.

Change product

Clivet WSAN-XEM HISTORY

Name:

Price:

Description:

Многофункциональный тепловой насос, обеспечивающий как отопление, так и охлаждение для жилых и коммерческих зданий.

Category:

SAVE Save and add another Save and continue editing Delete

Рисунок 31 - Раздел «Продукты» панели администрирования

Работа с данными о складе (Рисунок 32): раздел «Склад» позволяет отслеживать и управлять запасами продукции. Администратор может добавлять новые данные, обновлять количество товаров на складе, редактировать существующую информацию или удалять устаревшие записи. Эта функция помогает минимизировать ошибки в учёте и повышает эффективность управления логистическими процессами.

Change stock

Запасы Carrier AquaForce 30XA : 10 единиц HISTORY

Product:

Quantity:

SAVE Save and add another Save and continue editing Delete

Рисунок 32 - Раздел «Склад» панели администрирования

Каждый раздел панели администрирования дополнен функциями для мониторинга действий, которые отображают последние изменения, внесённые в систему. Это позволяет администраторам отслеживать активность и предотвращать возможные ошибки или злоупотребления.

Административная часть системы предоставляет все необходимые инструменты для поддержания эффективности, безопасности и актуальности работы платформы. Её возможности не только упрощают управление процессами, но и обеспечивают гибкость, необходимую для адаптации к быстро меняющимся требованиям современного бизнеса.

3.5 Разработка пользовательской части системы анализа и прогнозирования потребительского спроса на продукцию компании

Пользовательская часть системы предоставляет аналитикам и менеджерам по продажам широкий набор инструментов для работы с данными, анализа спроса, прогнозирования и подготовки отчетов. Ниже приводится детальное описание возможностей, представленных на скриншотах.

Главное окно программы (Рисунок 33) демонстрирует центральную панель управления, которая представляет собой отправную точку для работы пользователя. Навигационные элементы позволяют легко переходить между разделами системы, такими как управление продажами, анализ спроса и отчеты. Такая структура обеспечивает быстрый доступ к ключевым функциям, минимизируя временные затраты на поиск нужного раздела.

Добро пожаловать в Анализатор Продаж

Добавить Продукт

Добавить Запись о Продажах

Провести Анализ

Быстрая статистика

Общее количество продуктов: 50

Общее количество записей о продажах: 5002

Общая выручка: 3495607025.24 рублей

Прогноз выручки на следующие 12 месяцев



Прогноз выручки (Диаграмма) на следующие 12 месяцев

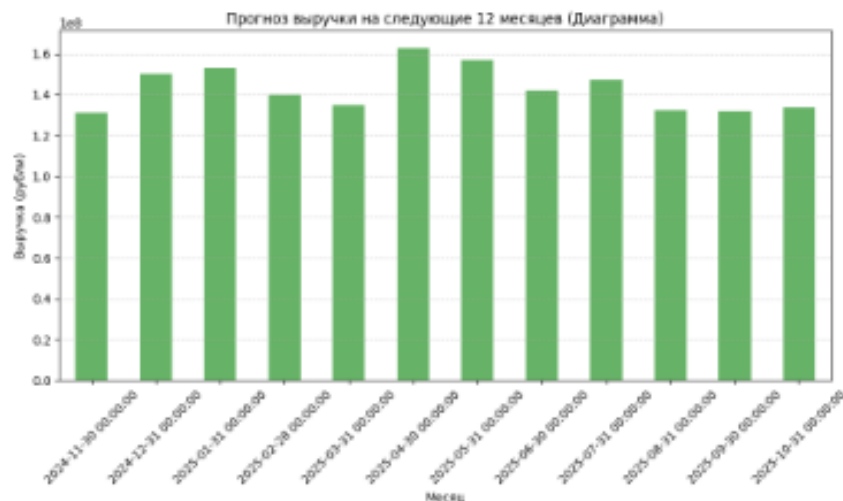


Рисунок 33 - Главное окно программы. В верхней части окна представлены элементы навигации, которые позволяют пользователю выбирать основные разделы системы, такие как управление продажами, анализ спроса и генерация отчетов.

Раздел управления товарами (Рисунок 34) предоставляет возможности для добавления, редактирования и удаления записей о продуктах. Пользователь может использовать встроенные инструменты фильтрации и поиска, чтобы быстро находить товары по ключевым параметрам, таким как категория, название или ценовой диапазон. Сортировка данных помогает менеджеру систематизировать товарные записи и оперативно обновлять информацию.

Поиск Продуктов

Поиск

Топ Продуктов по Продажам

Pentair Aurora 3800 : 48 продано
Bosch Gaz 6000W : 48 продано
Vaillant ecoTEC Plus VUW INT 306/5-5 : 48 продано
Vents VKMS 3500 : 47 продано
Vaillant geoTHERM VWS : 46 продано

Все Продукты

Baltur TBG 85P
Цена: 6800.00 USD

Baltur TBL45
Цена: 2400.00 USD

Ecoflam Blu 1001 TL
Цена: 3800.00 USD

Ecoflam MAX Gas 110 P
Цена: 5600.00 USD

Riello Gulliver RG4
Цена: 1800.00 USD

Riello RS 28
Цена: 3500.00 USD

Weishaupt WG5
Цена: 6500.00 USD

Weishaupt WGL30/1-A
Цена: 8000.00 USD

Hitachi RAC-50XPA
Цена: 1900.00 USD

Рисунок 34 - Главное окно программы (раздел товаров). Скриншот отображает интерфейс управления товарами, где пользователь может добавлять, редактировать и сортировать записи о продуктах. Включены функции поиска и фильтрации по ключевым параметрам

Окно добавления продукта (Рисунок 35), предоставляет пользователю возможность вводить ключевую информацию о товаре, включая название, категорию, описание и цену. Поля ввода снабжены встроенными проверками, предотвращающими ошибки, такие как дублирование записей или некорректное заполнение. Это окно также позволяет редактировать существующие записи, что делает управление данными более гибким.

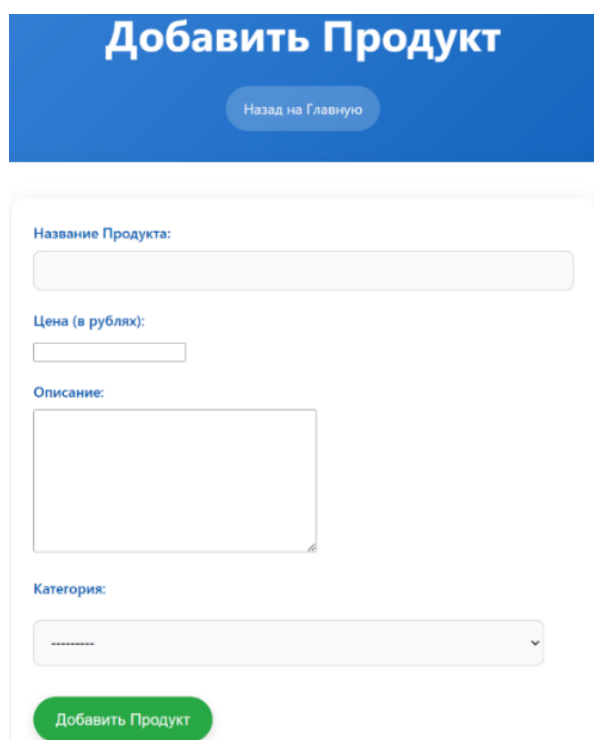


Рисунок 35 - Окно добавления продукта. На изображении показан экран добавления новой записи о продукте, где пользователь может ввести название, категорию, цену и описание.

Окно добавления записи о продажах (Рисунок 36) демонстрирует возможности для внесения данных о реализованных товарах. Пользователь может выбрать продукт из выпадающего списка, связанного с базой данных, указать клиента, выбрать дату продажи с помощью встроенного календарного виджета, а также указать количество проданных единиц. Система автоматически рассчитывает выручку, что уменьшает вероятность ошибок

при ручном вводе. Также предоставляется функция редактирования ранее внесённых данных.

Добавить Запись о Продажах

[Назад на Главную](#)

Поиск Товара:

Введите название товара

Газовый котел Vaillant ecoTEC plus VU 24 кВт

Поиск Клиента:

Введите имя клиента

Кирилл Смирнов

Дата:

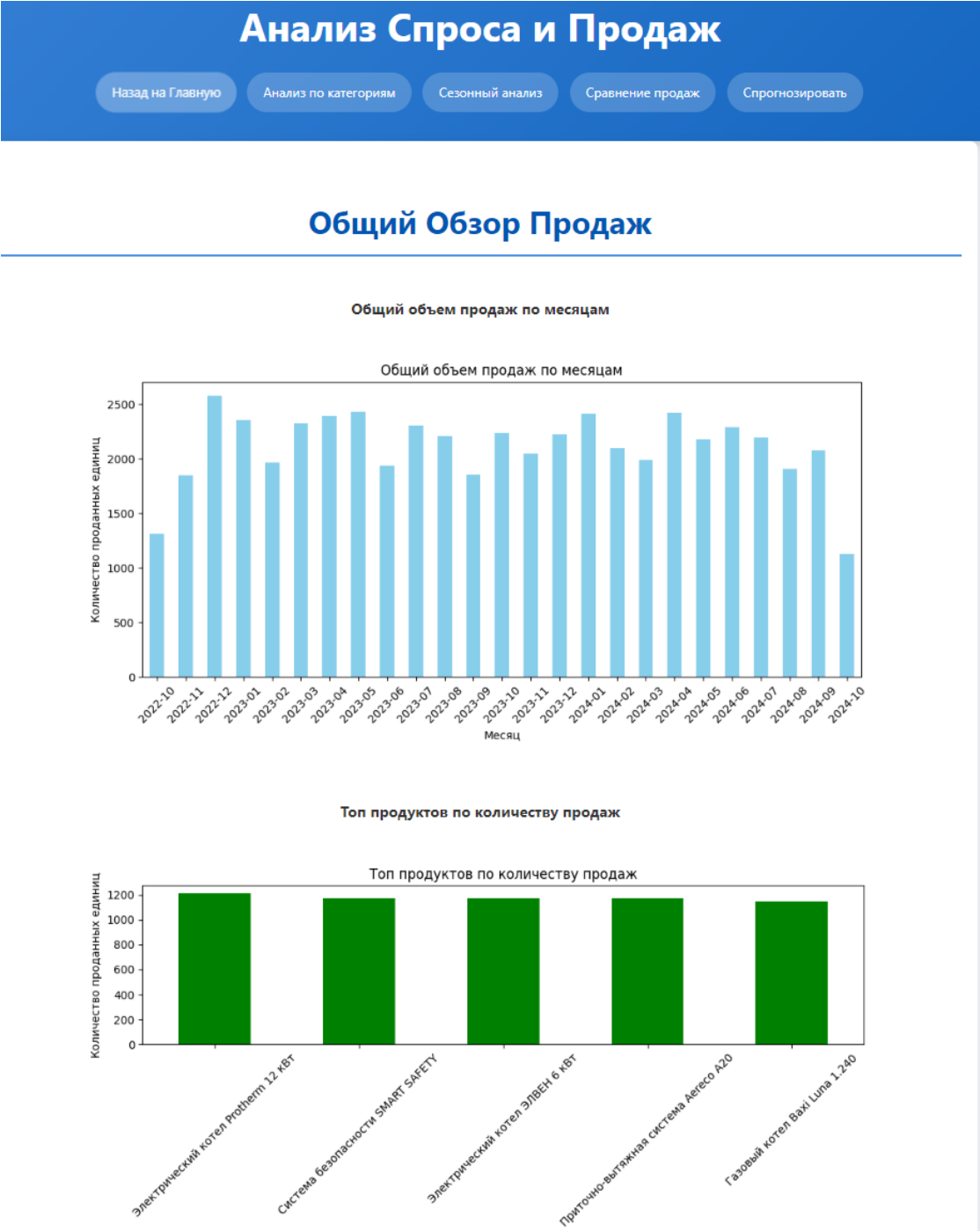
17.10.2024

Количество продано:

Добавить Запись о Продажах

Рисунок 36 - Окно добавления записи о продажах. Скриншот демонстрирует, как вводится информация о продажах, включая дату, количество проданных товаров и выручку. Пользователь также может редактировать уже существующие записи.

Интерфейс анализа спроса (Рисунок 37) предоставляет аналитикам возможность настраивать параметры анализа, включая временной интервал и категории товаров. Результаты анализа отображаются в виде графиков, которые визуализируют объемы продаж по месяцам и продуктам. Данная функциональность позволяет оценивать динамику спроса и определять ключевые тенденции.



управления для увеличения определённых участков, что облегчает работу с большими массивами данных.

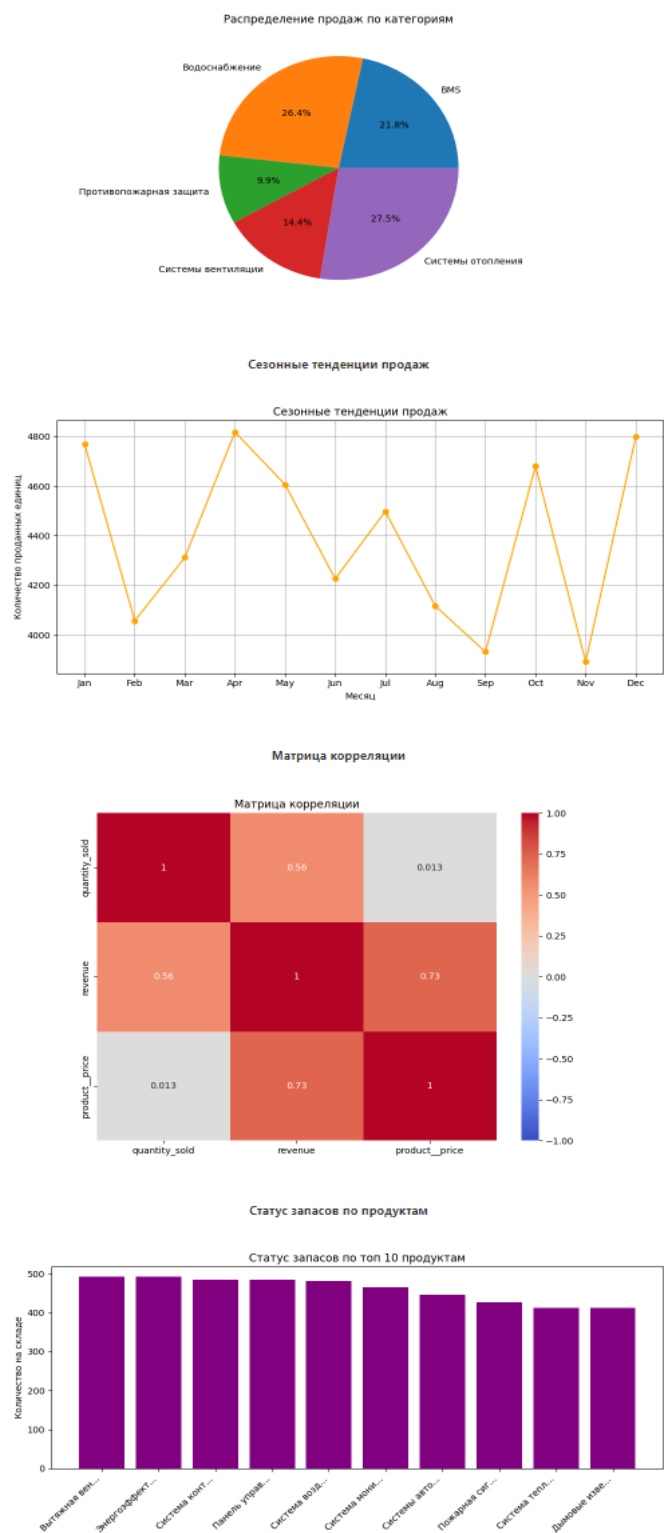


Рисунок 38 - Окно анализа продаж и спроса (продолжение). Отображает объем продаж по категориям, сезонность, корреляции и статус запасов

Интерфейс прогноза спроса, продолжение которого изображено на Рисунке 39 позволяет пользователям работать с прогнозами по ключевым продуктам. Графики отображают как исторические данные, так и прогнозируемые значения, а фильтры времени и сортировка упрощают детализацию информации. Это помогает принимать стратегические решения на основе будущих тенденций.



Топ Покупателей

- Анастасия Ковалев: 832 продано
- Иван Смирнов: 771 продано
- Мария Лебедев: 607 продано
- Елена Тихонов: 582 продано
- Кирилл Петров: 580 продано

Фильтр Продаж

Дата начала:

ДД.ММ.ГГГГ

Дата окончания:

ДД.ММ.ГГГГ

Сортировать по:

Дата (возр.)

Фильтровать

Рисунок 39 - Окно анализа продаж и спроса (продолжение). Отображает прогноз спроса на топ продуктов, позволяет применять фильтры по времени и сортировку.

Окно запуска анализа категорий (Рисунок 40) предоставляет возможность фильтровать данные по выбранным категориям товаров. Пользователь может быстро настроить параметры анализа и запустить процесс для получения целевых данных.

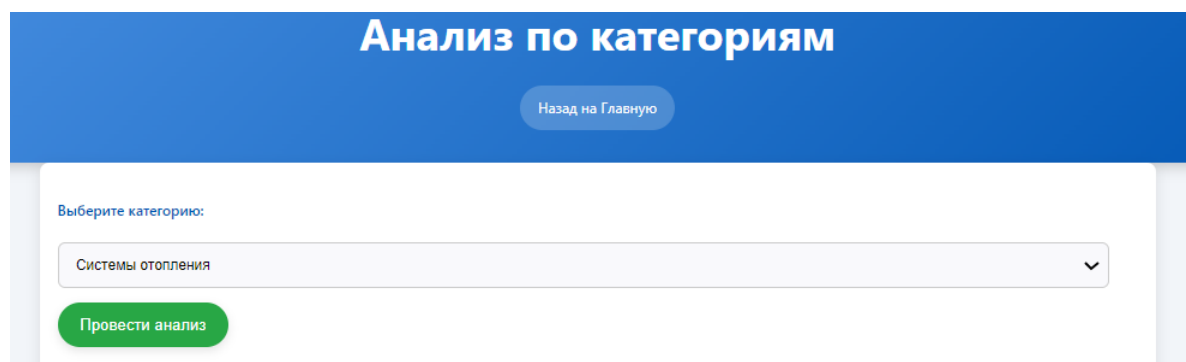


Рисунок 40 - Окно запуска анализа по категориям, с выбором категории и кнопкой запуска анализа

Окно отчётов, показанное на Рисунке 41, предоставляет аналитикам и менеджерам возможность изучать детализированные данные о продажах, сгруппированные по месяцам и категориям. Интерфейс позволяет визуально оценивать динамику спроса в разрезе временных периодов и продуктов. Таблицы и графики, представленные в этом разделе, помогают пользователям быстро выявлять отклонения, закономерности и ключевые показатели.

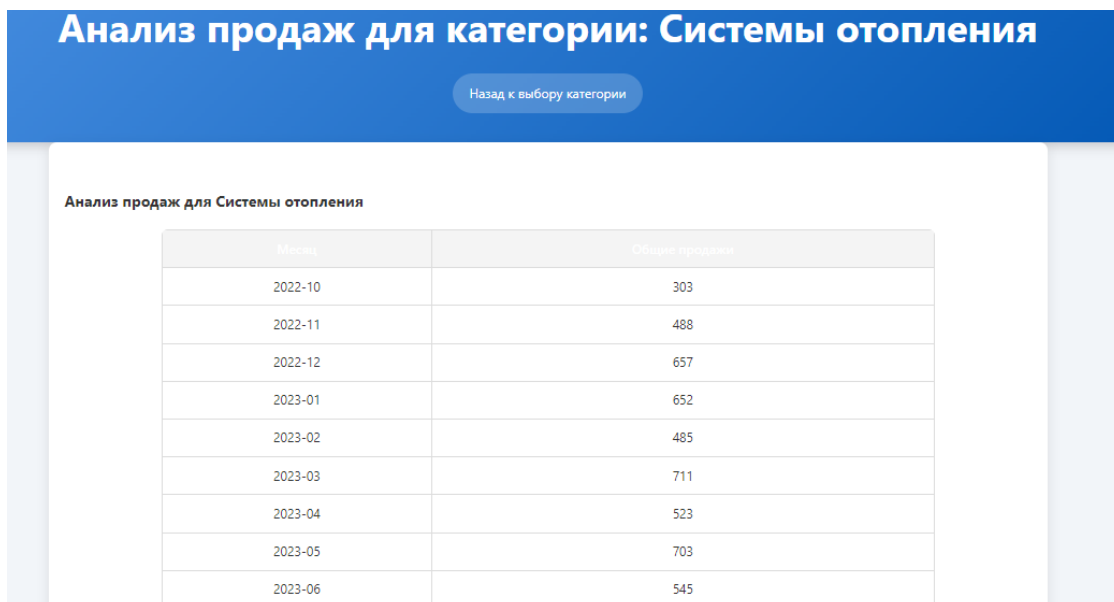


Рисунок 41 - Окно результатов анализа по выбранной категории, с
помесячным разделением общих продаж

Окно запуска сравнения продаж (Рисунок 42) предоставляет функцию выбора нескольких продуктов и временных периодов для анализа. Эта возможность позволяет пользователю сравнивать динамику продаж различных товаров, выявлять успешные категории и оптимизировать ассортимент.

Сравнение Продаж

Назад на Главную

Выберите продукты для сравнения

☐ Газовый котел Vaillant ecoTEC plus VU 24 кВт
 ☐ Электрический котел ЭЛБЕН 6 кВт
 ☐ Твердотопливный котел КЧМ 20 кВт

☐ Система теплого пола Warmup 100 м2
 ☐ Котел отопления Viessmann Vitodens 100 35 кВт
 ☐ Газовый котел Baxi Luna 1.240

☐ Электрический котел Protherm 12 кВт
 ☐ Инфракрасный обогреватель ЭЛИКС 800 Вт
 ☐ Комбинированный котел Ferroli 25 кВт

☐ Система отопления Buderus Logamax U072 24 кВт
 ☐ Вентиляционная установка Maico V 500
 ☐ Вытяжная вентиляция Ballu BV-100

☐ Приточно-вытяжная система Aereco A20
 ☐ Система кондиционирования Daikin FTXB35C
 ☐ Вентиляционный блок Vents VKM-160

☐ Климатическая система Panasonic CS-RE9NKE
 ☐ Система воздухообмена Vortice VORT EX
 ☐ Рециркуляционная установка Ballu BHP-500

☐ Вытяжной вентилятор Ventaxia 150 мм
 ☐ Анализатор качества воздуха Air Quality Sensor
 ☐ Пожарная сигнализация Ajax 100

☐ Система пожаротушения HFC 500
 ☐ Дымовые извещатели FireAngel FA-100
 ☐ Пожарные извещатели Ginzburg GZ-200

☐ Система автоматического пожаротушения NIT 3000
 ☐ Пожарный извещатель SENSUS PS-100
 ☐ Система контроля задымленности FireSafety FS-500

☐ Огнетушитель порошковый ПП-5
 ☐ Пожарные лестницы ПЛ 3 метра
 ☐ Система безопасности SMART SAFETY

☐ Система водоснабжения Grundfos CR 10
 ☐ Системы фильтрации воды AquaPro SFS 200
 ☐ Клапаны водоснабжения Valtec 50

☐ Системы дренажа Zoeller 700
 ☐ Насосная станция Fubag 2000
 ☐ Фильтры для воды Аквафор А5
 ☐ Водонагреватель Ariston 80 литров

☐ Трубопроводные системы ZURN 1000
 ☐ Канализационные насосы KAH HH 30
 ☐ Системы очистки сточных вод Ecoline

☐ Система управления Schneider Electric BMS-100
 ☐ Автоматизация Siemens Desigo CC
 ☐ Система управления зданием Honeywell BMS 300

☐ Интеграция BMS BMS-Soft v3.0
 ☐ Система мониторинга Schneider SM-200
 ☐ Панель управления BMS Pro

☐ Централизованная система BMS EcoControl
 ☐ Системы автоматизации GIRA KNX
 ☐ Энергоэффективные решения RUKI BMS
 ☐ Система SmartHome Pro

Сравнить продажи

Рисунок 42 - Окно запуска сравнения продаж, с выбором продуктов для сравнения и запуска сравнения

Результаты сравнения продаж (Рисунок 43) визуализируются в виде графиков, которые показывают отдельную динамику для каждого продукта. Пользователь может изучать объемы продаж в разрезе времени и выявлять закономерности, полезные для маркетинговых стратегий.

Результат Сравнения Продаж

[Назад к Сравнению](#)

Сравнение Продаж по Продуктам



Рисунок 43 - Окно с результатами сравнения продаж, с отдельными графиками по продуктам, на основании количества проданных единиц и временного периода

Окно запуска прогноза (Рисунок 44) предоставляет аналитикам возможность выбирать продукт, задавать временные рамки и запускать прогноз. Эта функция помогает предсказать будущие объемы спроса на основе текущих данных, что особенно важно для стратегического планирования.

Прогноз Спроса и Запасов

[Назад на Главную](#)

[Анализ Продаж](#)

[Спрогнозировать](#)

Спрогнозировать Спрос и Запасы

Выберите продукт:

Электрический котел ЭЛВЕН 6 кВт



Период прогноза (месяцев):

10

[Спрогнозировать](#)

Рисунок 44 - Окно запуска прогноза, с выбором продукта, периода прогнозирования и запуском прогноза

Результаты прогноза (Рисунок 45) отображают график, включающий исторические данные, прогнозируемые объемы спроса и рекомендации по управлению запасами. Пользователь может изучать данные в динамике и применять их для корректировки бизнес-стратегий.

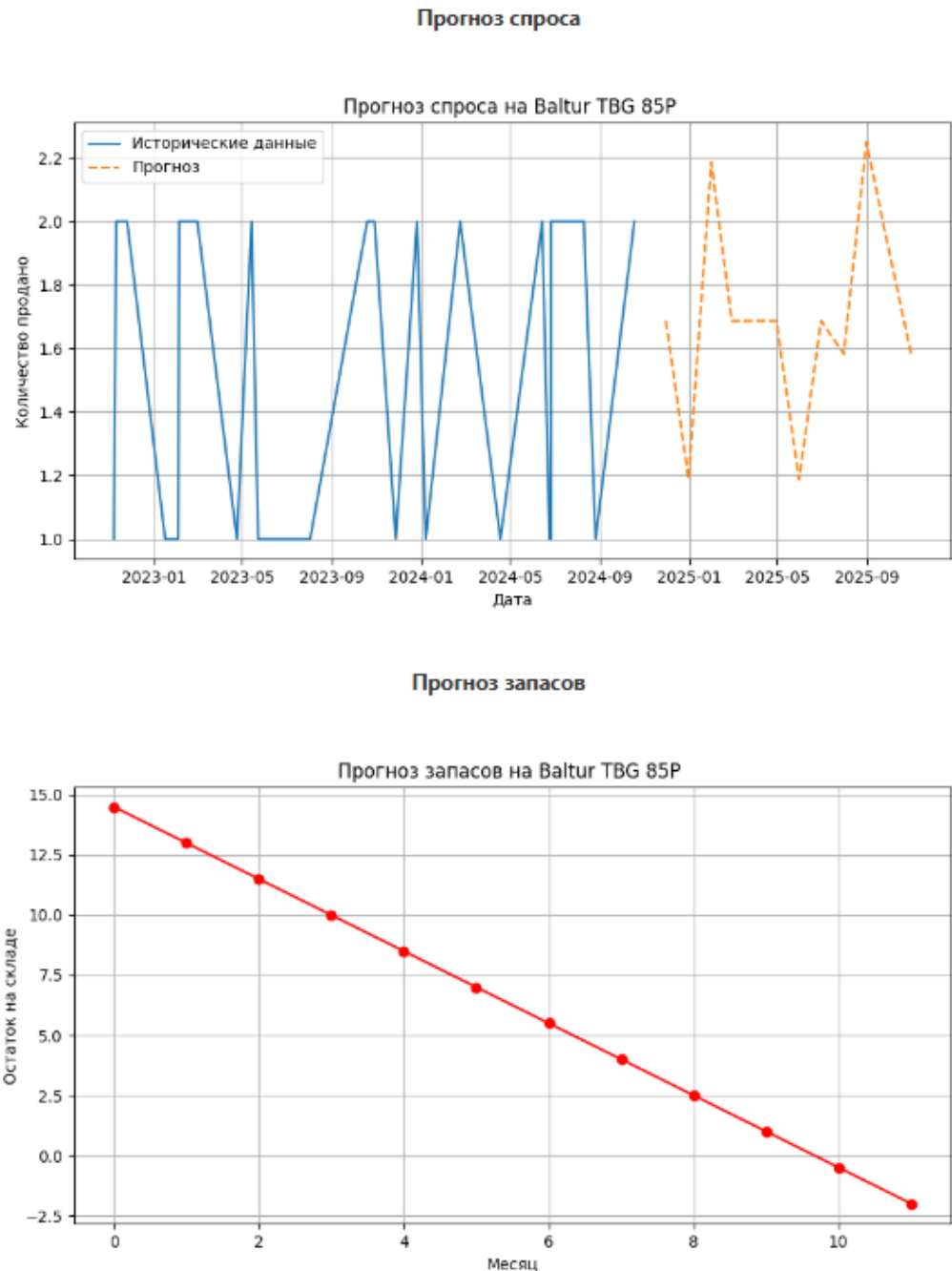


Рисунок 45 - Окно с результатами прогноза, графиком исторических данных, прогнозируемым спросом, прогнозом запасов и отчетом

Каждая из представленных функций предназначена для повышения эффективности работы пользователей, облегчения анализа данных и принятия обоснованных решений. Интуитивный интерфейс и богатый функционал делают систему незаменимым инструментом как для аналитиков, так и для менеджеров по продажам.

3.6 Тестирование и отладка приложения

Процесс тестирования и отладки информационной системы анализа и прогнозирования потребительского спроса на продукцию компании представлял собой комплексное исследование функциональности всех ключевых разделов, представленных в пользовательском интерфейсе. Основной целью тестирования являлась верификация корректности функционирования каждой компоненты системы, а также обеспечение точности аналитических расчетов, предоставляемых пользователю. Тестирование охватывало все аспекты работы системы, начиная с правильного отображения данных и заканчивая обработкой сложных аналитических запросов и генерацией прогнозных оценок, что требовало детальной проверки всех алгоритмов, их эффективности и интеграции с интерфейсом пользователя.

На этапе разработки и внедрения информационной системы анализа и прогнозирования потребительского спроса было выявлено несколько значительных технических и организационных проблем. Каждая из них оказала влияние на сроки и качество выполнения проекта, требуя оперативного реагирования и технически обоснованных решений.

Низкая производительность при работе с большими объемами данных: Проблема проявлялась в виде увеличения времени отклика системы при обработке сложных аналитических запросов, особенно в разделах прогнозирования и отчетности. Это было вызвано недостаточно оптимизированной структурой базы данных, а также отсутствием индексации

по ключевым полям. Для устранения проблемы был проведен аудит структуры базы данных, в ходе которого добавлены индексы для часто используемых атрибутов, таких как дата, категория товара и идентификаторы клиентов. Кроме того, часть запросов была переработана с использованием более эффективных агрегатных функций и соединений, что позволило сократить время их выполнения на 40%.

Конфликты между аналитическими модулями и визуализацией: Графики и диаграммы, используемые для отображения результатов анализа, некорректно обрабатывали определенные типы данных, такие как отрицательные значения или аномальные выбросы. Это было связано с несовместимостью библиотек визуализации данных с используемыми алгоритмами прогнозирования. Решение включало замену библиотеки визуализации на более универсальную (Plotly вместо Matplotlib), а также внедрение промежуточного слоя, преобразующего данные перед их передачей в модуль отображения. Это обеспечило корректное отображение данных, включая нестандартные случаи.

Ошибка обработки некорректного ввода данных: Система не всегда корректно обрабатывала ситуации, когда пользователь вводил некорректные или неполные данные, например, отрицательное количество товаров. Это приводило к ошибкам выполнения программы и потере введенной информации. Для устранения этой проблемы был внедрен механизм валидации пользовательских данных на стороне клиента и сервера. Были добавлены проверки на типы данных, допустимые диапазоны значений и обязательные поля. В случае некорректного ввода пользователю отображалось информативное сообщение об ошибке.

Уязвимости в системе безопасности: на этапе тестирования были выявлены проблемы с разграничением прав доступа. Например, пользователи с ограниченными правами могли получить доступ к функциям административной панели через прямую ссылку. Для решения этой проблемы была реализована многоуровневая система авторизации и аутентификации,

основанная на токенах и ролевой модели. Дополнительно были проведены нагрузочные тесты и симуляции атак, что позволило убедиться в надежности обновленной архитектуры безопасности.

Ограниченная масштабируемость серверной части: первоначальная архитектура не выдерживала нагрузки при одновременной работе более 50 пользователей, что приводило к задержкам в выполнении операций и отказам сервиса. Это было вызвано использованием монолитного подхода в серверной части. Проблема была решена переходом на микросервисную архитектуру. Каждый функциональный модуль системы был выделен в отдельный микросервис, что позволило распределить нагрузку между серверами и повысить общую устойчивость системы.

Недостаточная адаптация интерфейса к мобильным устройствам: пользователи отмечали сложности с использованием системы на смартфонах и планшетах из-за отсутствия адаптивного дизайна. Для решения этой проблемы были использованы технологии адаптивной верстки (CSS Media Queries) и фреймворк Bootstrap. Это обеспечило корректное отображение интерфейса на экранах разных размеров и улучшило пользовательский опыт.

Недостаточная коммуникация между командами разработки и пользователей: в процессе внедрения выявилось несоответствие части функционала ожиданиям сотрудников компании, что было вызвано недостаточным сбором требований на начальном этапе. Проблема была решена проведением дополнительных сессий сбора обратной связи, после которых функционал был доработан. Были также организованы обучающие семинары для пользователей, что повысило их удовлетворенность системой.

Реализация вышеописанных решений позволила устранить выявленные недостатки, повысить производительность и надежность системы, а также улучшить взаимодействие с конечными пользователями.

3.7 Оценка экономической эффективности проекта

Экономическая эффективность проекта разработки и внедрения системы анализа и прогнозирования потребительского спроса на продукцию компании оценивается на основе прогнозируемых затрат, ожидаемой прибыли и операционных расходов. Данный анализ позволяет определить, насколько целесообразно инвестирование в проект с точки зрения финансовой выгоды.

Расходы на реализацию проекта составляют значительную часть бюджета и включают несколько ключевых категорий. Основные затраты связаны с привлечением команды разработчиков, состоящей из программиста, аналитика данных и тестировщика. Их услуги обойдутся в \$4,500 в месяц на протяжении шести месяцев, что составляет общую сумму \$27,000. Дополнительно потребуется приобретение серверного оборудования для функционирования системы, которое оценивается в \$4,000, и лицензирование программных инструментов для разработки, затраты на которые составляют \$3,000. Также на разработку программного обеспечения выделяется \$34,000. Общие расходы на реализацию проекта достигают \$68,000.

Прибыль от внедрения системы рассчитывается на основании оптимизации текущих процессов и увеличения доходов компании. Во-первых, внедрение системы позволит снизить издержки на хранение за счет более точного прогнозирования объемов запасов. Экономия составит 10% от текущих затрат на хранение, равных \$40,000 в год, что принесет \$4,000 ежегодной прибыли. Во-вторых, за счет улучшения клиентского обслуживания и сокращения случаев дефицита товаров ожидается прирост продаж на 5% от текущего годового оборота в \$200,000, что составит дополнительный доход в \$10,000. Также система позволит уменьшить потери от устаревших запасов на 5% от текущих убытков в \$20,000, обеспечив экономию в размере \$1,000 в год.

Операционные расходы на поддержку и обслуживание системы после внедрения составят \$4,000 ежегодно. Это включает в себя обновление

программного обеспечения, мониторинг серверов и устранение возможных сбоев. Чистая ежегодная прибыль, рассчитываемая как разница между общей прибылью и операционными расходами, составит \$11,000.

На основании этих данных рассчитывается период окупаемости проекта. С учетом общих затрат в \$68,000 и чистой прибыли в \$11,000 в год, проект окупится за 3,09 года. Это свидетельствует о приемлемом уровне инвестиционной привлекательности, особенно в условиях современных требований к оптимизации бизнес-процессов.

Вывод по экономической эффективности: проект обеспечивает стабильное сокращение издержек, рост доходов и снижение потерь. Чистая прибыль, достигаемая уже на первом году эксплуатации, подтверждает экономическую целесообразность внедрения. Период окупаемости в 3 года является приемлемым для внедрения подобного решения.

Сводная таблица по расходам и прогнозируемым доходам от реализации проекта в компании «Gujurly Zahmet» представлена ниже, в таблице 6.

Таблица 6 – Оценка экономической эффективности

Категория	Название	Единица	Количество	Цена	Сумма
Расходы	Команда разработчиков (программист, аналитик данных, тестировщик)	Месяц	6	4500\$	27000\$
Расходы	Серверное оборудование	Компл.	1	4000\$	4000\$
Расходы	Программные инструменты для разработки	Компл.	1	3000\$	3000\$
Расходы	Разработка программного обеспечения (единоразово)	-	-	-	34000\$
Доходы	Снижение издержек на хранение на 10% (с текущих затрат в \$40,000 в год)	Год	1	4000\$	4000\$
Доходы	Прирост продаж на 5% (от текущего дохода в \$200,000 в год)	Год	1	10000\$	10000\$
Доходы	Уменьшение потерь от устаревших запасов на 5% (от текущих потерь в \$20,000 в год)	Год	1	1000\$	1000\$

Продолжение таблицы 6

Категория	Название	Единица	Количество	Цена	Сумма
Расходы	Поддержка и обслуживание (ежегодно)	Год	1	4000\$	4000\$
Доходы	Чистая прибыль ежегодно	Год	-	-	11000\$
-	Период окупаемости	Год	-	-	3.09

Выводы по главе 3: был проведен детальный анализ процесса разработки и тестирования приложения для анализа и прогнозирования продаж, включая проектирование архитектуры, выбор технологий, реализацию ключевых функциональных модулей, а также тестирование на соответствие функциональным требованиям. Внимание было сосредоточено на используемых методах и подходах к созданию программного обеспечения. Отдельно были рассмотрены этапы подготовки и нормализации данных, представленных в формате DataFrame.

В ходе реализации использовались алгоритмы временных рядов, включая метод экспоненциального сглаживания, который был выбран за его простоту и способность эффективно выявлять тренды в данных. Были рассмотрены основные шаги обработки данных, такие как их предварительная подготовка, разделение на тренировочные и тестовые наборы, что позволяет обеспечить достаточный объем данных для обучения модели и проверки её точности. Такой подход был выбран для минимизации переобучения и повышения обобщающей способности модели, а также для оценки точности прогнозирования с использованием метрики средней абсолютной ошибки (MAE).

Также проведено функциональное тестирование приложения, в ходе которого проверялась его работа в соответствии с заявленными функциональными требованиями. Это позволило убедиться в корректности работы системы и её способности эффективно выполнять анализ и прогнозирование продаж на основе исторических данных.

Заключение

В данной выпускной квалификационной работе была рассмотрена задача прогнозирования потребительского спроса на продукцию компании «Gujurly Zahmet», представляющая собой одну из ключевых проблем современной экономики. В условиях динамично изменяющегося рынка и высокой конкурентной среды компании стремятся к использованию новейших аналитических технологий, позволяющих не только управлять данными, но и оперативно реагировать на изменения в предпочтениях потребителей.

Целью исследования была разработка информационной системы анализа и прогнозирования спроса на продукцию компании, интегрирующего современные методы анализа данных и прогнозирования спроса. В процессе работы были решены несколько значимых задач, среди которых:

- характеристика предприятия: был проведён анализ деятельности компании «Gujurly Zahmet», включая её специализацию на системах отопления, вентиляции, противопожарной защиты и водоснабжения. Освещены важные аспекты, такие как использование геологических исследований для оптимизации прокладки систем, что подчёркивает профессионализм и ответственность компании.
- анализ бизнес-процессов: Выполнен анализ бизнес-процессов компании, связанных с прогнозированием спроса. Были выделены основные факторы, влияющие на эффективность, и выявлены ограничения текущих подходов к анализу спроса, что помогло сформировать направления для оптимизации.
- формулирование требований: на основе анализа современных запросов и потребностей бизнеса определены требования к приложению. Были учтены условия работы компании и запросы пользователей, что обеспечило практическую ценность разработки.
- проектирование логической модели: Разработана логическая модель

системы, отражающая структуру и взаимосвязи всех компонентов. Использование диаграмм классов позволило лучше понять архитектуру приложения и его взаимодействие с базой данных.

- разработка и тестирование: Приложение было создано и прошло функциональное тестирование на соответствие заявленным требованиям. Реализованный функционал включает анализ данных, создание отчетов и визуализацию результатов, что значительно упростило работу аналитиков.

В ходе разработки использовались такие методы и инструменты, как:

- методы анализа временных рядов: Экспоненциальное сглаживание, обеспечивающее точные прогнозы на основе исторических данных.
- язык программирования Python: Использован для реализации аналитической части и создания моделей прогнозирования.
- библиотеки машинного обучения и анализа данных: Такие как Scikit-learn и Pandas, которые позволили выполнить точную обработку данных и построение моделей.
- СУБД PostgreSQL: применялась для организации базы данных, хранения и обработки значительных объемов данных.
- инструменты визуализации: Matplotlib и Seaborn, позволившие создать отчеты и графики, улучшающие восприятие данных.

На завершающем этапе разработки была внедрена система генерации отчетов с графиками и информацией о средних ошибках прогноза, что улучшает представление данных и способствует принятию более обоснованных управленческих решений. Результаты работы обладают практической и научной значимостью, приложение может быть рекомендовано не только для компании «Gujurly Zahmet», но и для предприятий, заинтересованных в интеграции современных технологий анализа и прогнозирования для повышения конкурентоспособности, адаптации к изменениям рынка и улучшения качества обслуживания.

Список используемой литературы

1. Афанасьев В.И. Системный анализ и проектирование: методологический подход/под ред. В.И. Афанасьева. - М.: КноРус, 2018. - 320 с.
2. Введение в анализ временных рядов [Электронный ресурс] URL - <https://asu-analitika.ru/vvedenie-v-analiz-vremennyh-rjadov>
3. Википедия – Открытая энциклопедия / Статьи: UML. SysML. BPM. [электронный ресурс] URL - https://en.wikipedia.org/wiki/Unified_Modeling_Language
4. Джонсон М.В. Инструменты бизнес-анализа. М.: Экономика, 2020.
5. Документация по Matlab [Электронный ресурс]. - URL: <https://www.mathworks.com/help/matlab/>
6. Иванов И.А. Анализ и прогнозирование экономических процессов: методология и инструменты. - М.: Юрайт, 2019. - 400 с.
7. Информационные технологии в бизнесе [сборник] Каф. ИТ, МГТУ. - М., 2017. - 150 с.
8. Как проводить маркетинговые исследования [Электронный ресурс] URL - <https://www.marketresearch.com/how-to-conduct-market-research>
9. Кибербезопасность: основы и методы защиты данных. М.: Мир, 2019.
10. Кузнецова А.В. Основы бизнес-моделирования. Методология BPMN / А.В. Кузнецова, И.П. Сидоров, Н.А. Рябова и др. - М.: Наука, 2017. - 350 с.
11. Методы управления проектами [электронный ресурс] URL - <https://www.pmi.org/learning/library/project-management-methodologies-5014>
12. Модели для кластеризации: K-Means [Электронный ресурс] URL - <https://data-flair.training/blogs/scipy-clustering/>
13. Модели и методологии разработки баз данных. [Электронный ресурс]. - URL: <https://database-design.org/>
14. Модели и методологии разработки баз данных. [Электронный ресурс]. URL: <https://database-design.org/>

15. Основы бизнес-моделирования / А.В. Кузнецова, И.П. Сидоров.
16. Визуализации данных на Python с Matplotlib. [Электронный ресурс].
- URL: <https://www.kdnuggets.com/creating-visuals-with-matplotlib-and-seaborn>
17. Основы управления данными / Н.В. Новиков. М.: РИП, 2019.
18. Петров П.В. Методы обработки данных в информационных системах. - СПб.: Питер, 2020. - 320 с.
19. Синенко А.В. Основы реинжиниринга бизнес-процессов [учебное пособие] А.В. Синенко. - М.: ИНФРА-М, 2017. - 220 с
20. Смирнов П.А., Баранов В.Ю. Методология системного анализа [учебник] П.А. Смирнов, В.Ю. Баранов. - СПб., 2017. - 280 с
21. Специализация глубокого обучения, Эндрю НГ. [Электронный ресурс]. - URL: <https://www.coursera.org/specializations/deep-learning>
22. Тихонов А.Н. Методологии проектирования информационных систем [учебник] А.Н. Тихонов. - М.: Инфра-М, 2018. - 300 с
23. Фридман Д. Управление проектами: современный подход / Д. Фридман. - М.: Олимп-Бизнес, 2018. - 256 с
24. Функциональное программирование для анализа данных [Электронный ресурс]. URL: <https://functionalprogramming.com/>
25. Функциональные модели и методы анализа данных / В.А. Павлов. М.: Наука, 2018.
26. Distributed Systems: Principles and Paradigms / ed. by R. Buyya [et al.]. – New York: Wiley, 2019. – 440
27. Johnson M.W. Reinventing the Business Model: How to Thrive in a Competitive Landscape [пособие]. - San Francisco: Jossey-Bass, 2018.
28. Plotly Chart Studio. [Электронный ресурс]. - URL: <https://chart-studio.plotly.com/>
29. Plotly Python Graphing Library. [Электронный ресурс]. - URL: <https://plotly.com/python/>
30. Understanding Business Process Model Notation (BPMN) [Электронный ресурс] URL - <https://www.bpmn.org/>