

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)

Разработка социальных и экономических информационных систем

(направленность (профиль)/специализация)

**ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА
(БАКАЛАВРСКАЯ РАБОТА)**

на тему «Разработка приложения для анализа и прогнозирования продаж организации»

Обучающийся

Д.В. Логинов

(И.О. Фамилия)

(личная подпись)

Руководитель

канд. эк. наук, доцент, Раченко Т.А.

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

Консультант

Усатова И.Ю.

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

Аннотация

Тема бакалаврской работы – «Разработка приложения для анализа и прогнозирования продаж организации».

В условиях современного рынка, успешность бизнеса во многом определяется способностью к анализу и прогнозированию продаж. Это требует оперативного доступа к данным и возможности их анализа за кратчайший промежуток времени.

Объектом исследования бакалаврской работы является разработка и интеграция современных инструментов для анализа данных в информационную систему организации.

Предметом исследования бакалаврской работы является проект по реализации и интеграции инструментов для анализа данных в IT-инфраструктуру организации.

Цель бакалаврской работы – проектирование и разработка приложения, использующего современные инструменты для анализа и прогнозирования продаж организации.

Актуальность работы подчеркивается необходимостью повышения конкурентоспособности организации за счёт более точного и оперативного анализа и прогнозирования продаж, что является важным аспектом в условиях быстро меняющегося рынка.

Практическая значимость бакалаврской работы заключается в разработке готового IT-решения, включающего в себя современные инструменты для анализа и прогнозирования данных, с последующей возможностью интеграции в IT - инфраструктуру организации.

Данная работа состоит из введения, трёх глав, заключения и списка литературы.

Работа включает в себя 54 страницы текста, 26 рисунков, 3 таблицы и 25 источников.

Abstract

The title of the graduation work is “The Development of an Application for Analysis and Forecasting of Organization Sales”.

The senior paper consists of an introduction, three parts, a conclusion and list of references including foreign sources.

The key issue of the thesis depends on the ability to analyze and forecast sales. This requires prompt access to data and the ability to analyze it in the shortest possible time.

The aim of the work is the project to implement and integrate data analysis tools into the organization’s IT infrastructure.

The object of the work is development and integration of modern tools for data analysis into the organization’s information system.

The purpose of the work is to design and development of an application that uses modern tools for analyzing and forecasting sales of an organization.

The relevance of the work is emphasized by the need to increase the competitiveness of the organization through more accurate and timely analysis and sales forecasting, which is an important aspect in a rapidly changing market.

In conclusion we’d like to stress this work is relevant in development of a ready-made IT solution, including modern tools for data analysis and forecasting, with the subsequent possibility of integration into the organization’s IT infrastructure.

Оглавление

| | |
|--|----|
| Введение | 5 |
| Глава 1 Функциональное моделирование предметной области | 7 |
| 1.1 Характеристика ООО «СМЦ «Металл-Маркет» | 7 |
| 1.2 Выбор технологии концептуального моделирования | 9 |
| 1.4 Формирование требований | 16 |
| Глава 2 Логическое проектирование приложения | 20 |
| 2.1 Описание инструментов для анализа и прогнозирования данных, использующих машинное обучение..... | 20 |
| 2.2 Проектирование статической структуры приложения | 23 |
| 2.3 Выбор СУБД | 35 |
| Глава 3 Физическое проектирование приложения | 40 |
| 3.1 Разработка приложения..... | 40 |
| 3.1 Тестирование приложения | 43 |
| Список используемой литературы | 52 |

Введение

На сегодняшний день технологии машинного обучения в сфере обработки данных набирают огромный темп развития. Различные предприятия, в план которых входит анализ и прогнозирование продаж все больше используют современные инструменты, позволяющие в кратчайшие сроки получить точный отчёт в результате имеющейся информации.

Практика подтверждает, что эффективное управление данными не только ускоряет процесс принятия решений, но и обеспечивает значительное конкурентное преимущество на рынке. Предприятиям, стремящимся оставаться в конкурентной среде необходимо прибегать к технологиям машинного обучения. Интеграция таких систем открывает новые горизонты анализа больших данных и управления продажами в реальном времени.

Методы машинного обучения представляют собой множество полезных алгоритмов, способных исполнять объёмные и трудоёмкие задачи, опираясь лишь на шаблоны и выводы, а не на явные инструкции.

Проектирование ИТ-решения, обеспечивающего интеграцию технологий машинного обучения в информационную систему организации, представляет актуальный и научно-практический интерес.

Объектом исследования бакалаврской является разработка и интеграция современных инструментов для анализа данных в информационную систему организации.

Предметом исследования бакалаврской работы является проект по реализации и интеграции инструментов для анализа данных в ИТ-инфраструктуру организации.

Целью бакалаврской работы является проектирование и разработка приложения, использующего современные инструменты для анализа и прогнозирования продаж организации.

Актуальность работы подчеркивается необходимостью повышения конкурентоспособности организации за счёт более точного и оперативного

анализа и прогнозирования продаж, что является важным аспектом в условиях быстро меняющегося рынка.

Для достижения данной цели необходимо реализовать следующие задачи:

1. характеристика ООО «СМЦ «Металл-Маркет»;
2. исследование бизнес-процессов анализа и прогнозирования продаж организации ООО «СМЦ «Металл-Маркет»;
3. сформулировать требования к приложению и выбрать средства разработки;
4. спроектировать логическую модель приложения;
5. разработать и провести тестирование приложения;
6. произвести оценку эффективности приложения для анализа и прогнозирования продаж организации.

Во время исследования будут использованы методы моделирования, а также инструменты машинного обучения для работы с большим объемом данных.

Бакалаврская работа будет ориентирована на разработку приложения, направленного на анализ и прогнозирование продаж организации.

Данная работа состоит из введения, трёх глав, заключения и списка использованной литературы.

В первой главе будет отражен анализ бизнес-процессов организации, создание модели программного обеспечения и формулировку требований к нему.

Во второй главе будет отражен логический этап и описание стадий проектирования приложения, а также выбор и описание программных средств для разработки.

В третьей главе будет отражена реализация и тестирование приложения, согласно требованиям, сформулированным в первой главе.

В заключении подводятся результаты проведенной выпускной квалификационной работы.

Глава 1 Функциональное моделирование предметной области

1.1 Характеристика ООО «СМЦ «Металл-Маркет»

ООО «СМЦ «Металл-Маркет» занимается оптовой торговлей металлами в первичных формах. Компания предлагает широкий ассортимент металлопродукции, а также услуги по металлообработке. Являясь лидером на рынке, бизнес нуждается во внедрении современных инструментов для анализа и прогнозирования данных. ИТ-решением, которое решает эту проблему может являться приложение, использующее методы машинного обучения, обеспечивающее повышение эффективности, производительности и адаптируемости. Использование подобного функционала будет играть решающую роль в решении вопросов, связанных с трансформацией бизнеса, обеспечивая фундаментальные изменения в бизнес-операциях для лучшего реагирования на рыночные изменения, что приводит к созданию единой информационной среды и оптимизированной поддержки сквозных корпоративных и организационных процессов. [9]

Услугами ООО «СМЦ «Металл-Маркет» пользуется широкая часть предприятий, использующих металлопродукцию в своих производствах. В связи с высоким интересом к продукции компании, каждый день происходит сотни сделок, из-за чего предприятие имеет большое количество отделов для работы с различными задачами. Организационная структура предприятия показана на рисунке 1.

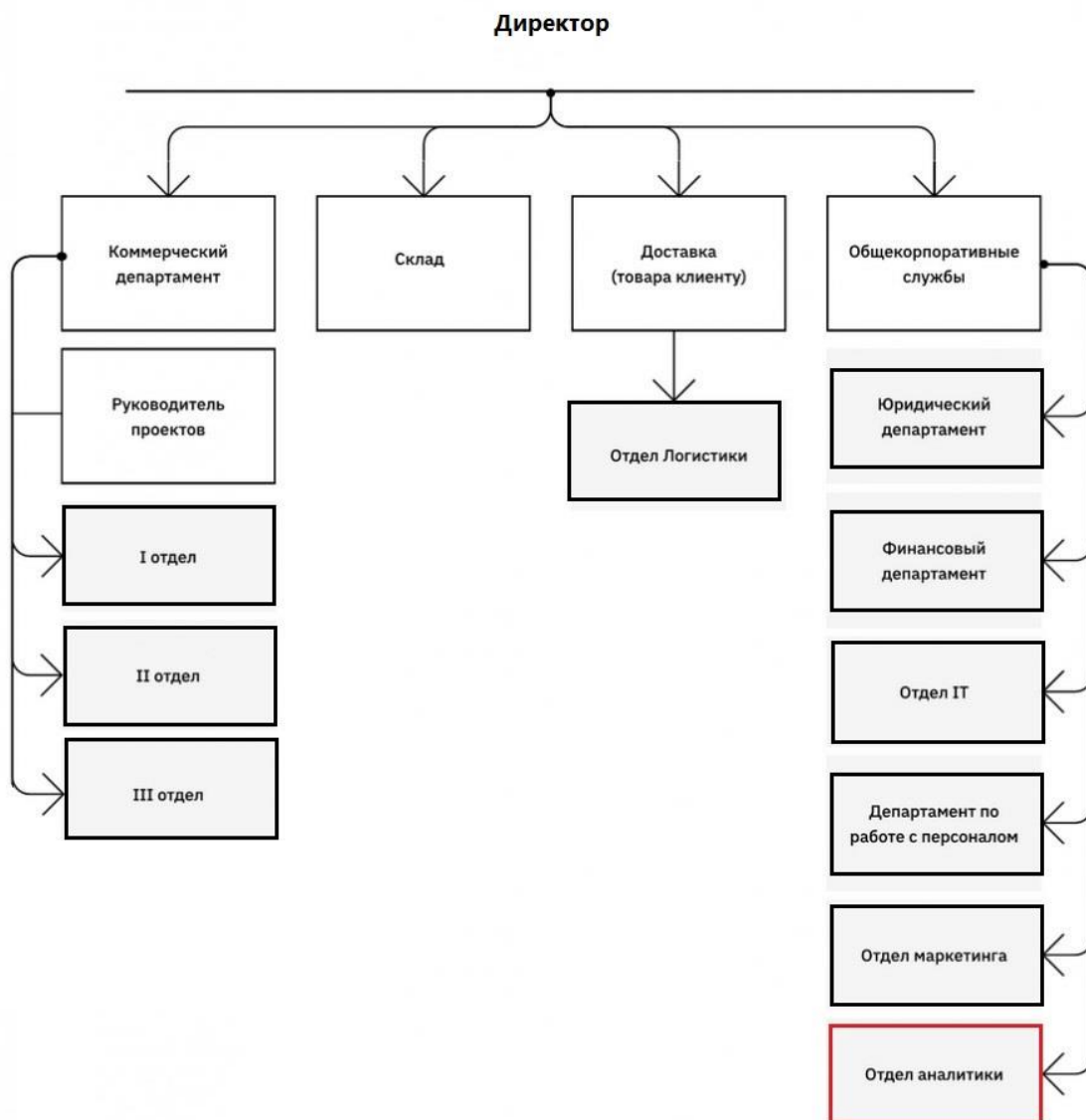


Рисунок 1 – Организационная структура ООО «СМЦ «Металл-Маркет»

Рассмотрим отдел аналитики более подробно. Он играет важную роль в организации, так как отвечает за сбор, обработку и анализ данных, что помогает руководству принимать обоснованные решения. Аналитики извлекают полезные знания из данных, находят взаимосвязи, визуализируют информацию и предоставляют рекомендации, которые способствуют улучшению продуктов или процессов.

Задачи отдела аналитики включают разработку стратегий и рекомендаций, поддержку принятия решений на основе фактов и аналитических выводов. Также оценивать финансовое состояние компании, определять ликвидность и рентабельность, делать прогнозы будущей продуктивности и помогать компании оставаться на шаг впереди конкурентов. [8]

К аналитике данных может быть привлечено большое количество специалистов, таких как: руководители, технические специалисты, маркетологи и социологи. Однако самыми важными кадрами являются аналитики. В зависимости от поставленной задачи, аналитики также делятся на несколько типов: бизнес-аналитик, системный аналитик, аналитик данных.

1.2 Выбор технологии концептуального моделирования

В данном разделе будут описаны основные современные методологии и технологии концептуального моделирования.

Методология объектно-ориентированного анализа и проектирования, основанные на стандартах языка UML.

Основной данной методологии является объектная модель. Большинство современных методов методологии построены на применении языка UML. UML – это унифицированный язык моделирования. Он является языком для представления, определения, документирования и проектирования систем. [12]

Создавать модель UML можно в программах, таких как: Draw.io, Gliffy, Visual Paradigm и др. В моем случае будет использоваться бесплатное программное обеспечение Creatly.

Объектно-ориентированный анализ и проектирование подразумевает использование диаграмм классов, компонентов и размещений, позволяющие моделировать архитектуру, иерархию и структуру системы.[22] Также

используются диаграммы поведения, которые позволяют описать поведение классов в системе. [10]

Методология IDEF0 представляется процесс взаимодействия как нечто, что получает на вход информацию для обработки, этой информацией может быть, к примеру, звонок клиента. А на выходе получается итог обработки данной информации, к примеру, покупка товара. Данная методология также включает в себя стратегии и процедуры необходимые для выполнения (управляющие), а также описывает необходимые для проведения работы ресурсы (механизмы).

Создавать модель IDEF0 можно программами, такими как: ERWIN, BPWIN, Visio и др. В моем случае будет использоваться бесплатное программное обеспечение Ramus.

Модель IDEF0 можно декомпозировать, разбив на дочерние диаграммы, которые более подробно описывают процессы, происходящие во время работы.

Назначение модели IDEF0 заключается в предоставлении инструмента для понимания, анализа и принятия решений относительно реконструкции, замены или проектирования новой системы. Она описывает систему как взаимосвязанное совокупное целое, состоящее из разнообразных элементов, таких как люди, информация, программное обеспечение, оборудование, продукция, сырье или энергия. Модель детально описывает функции системы, методы управления ею, воздействие на сущности, которые она обрабатывает, используемые ресурсы для выполнения задач и результаты ее функционирования.

Основные элементы диаграммы IDEF0:

– Блоки функций (Function Boxes) – представляют функции системы и отображают, что система делает. Они обозначают конкретные действия или процессы, выполняемые в рамках системы;

- Связи (Arrows) – отображают потоки информации или материальных объектов между блоками функций. Они описывают потоки и передачу данных между различными функциями системы;
- Управляющие блоки (Control Nodes) – помечают особые условия, ограничения или решения, влияющие на выполнение функций или потоки данных в системе;
- Текстовые аннотации (тильды) – дополнительные пояснения, комментарии или описания, которые могут быть добавлены к диаграмме для уточнения деталей и описания конкретных элементов. [19]

Блоки в IDEF0 представляют основные функции моделируемого объекта и могут быть разбиты на более детальные составные части через процесс декомпозиции. Этот процесс декомпозиции продолжается до достижения необходимого уровня детализации для достижения целей конкретного проекта. Диаграмма верхнего уровня обеспечивает общее или абстрактное описание объекта моделирования. За ней следует серия дочерних диаграмм, которые предоставляют более подробное представление об объекте. [11]

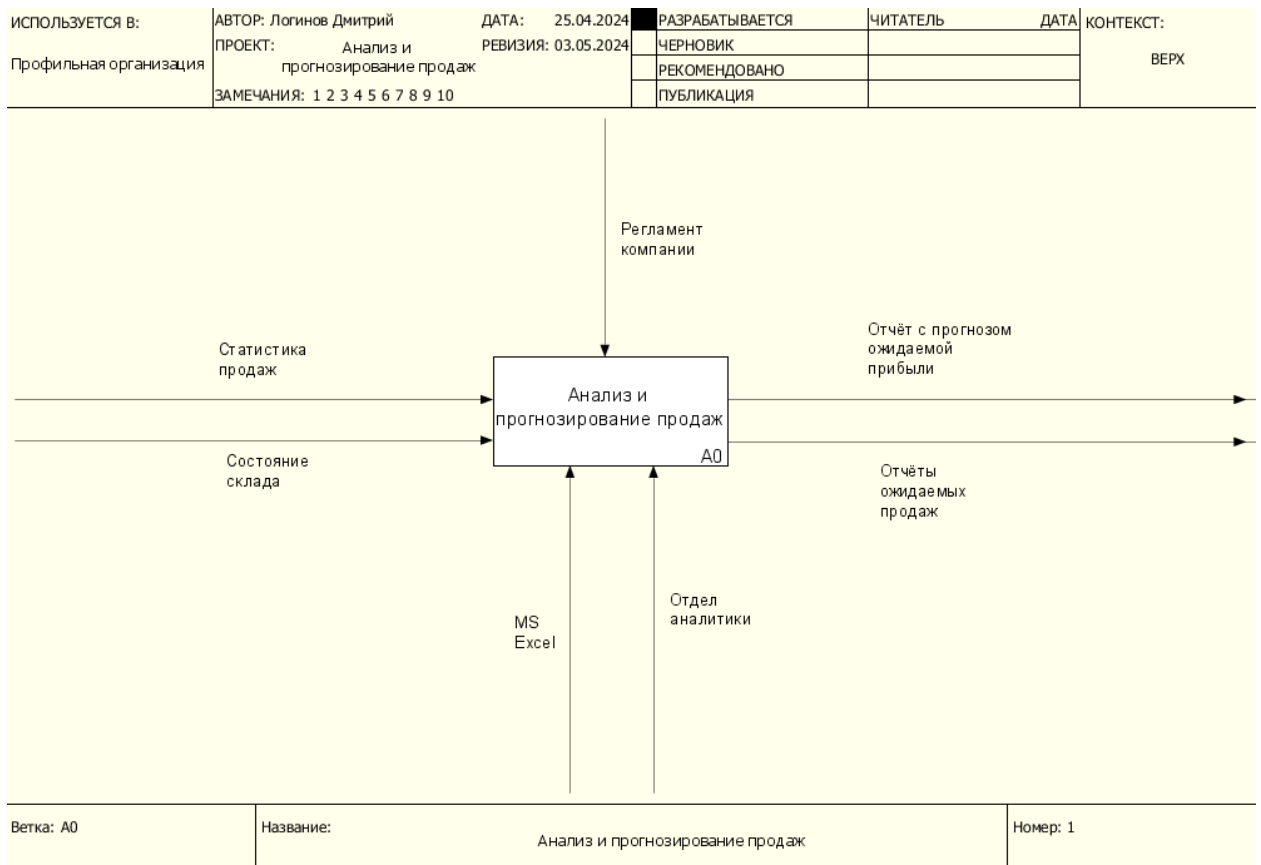


Рисунок 2 – Контекстная диаграмма бизнес-процесса «Анализ и прогнозирование продаж» AS-IS

После того, как описали контекстную диаграмму, необходимо разбить ее на подсистемы и описывать уже их. Данный процесс называется декомпозицией и представлен на рисунке 3.

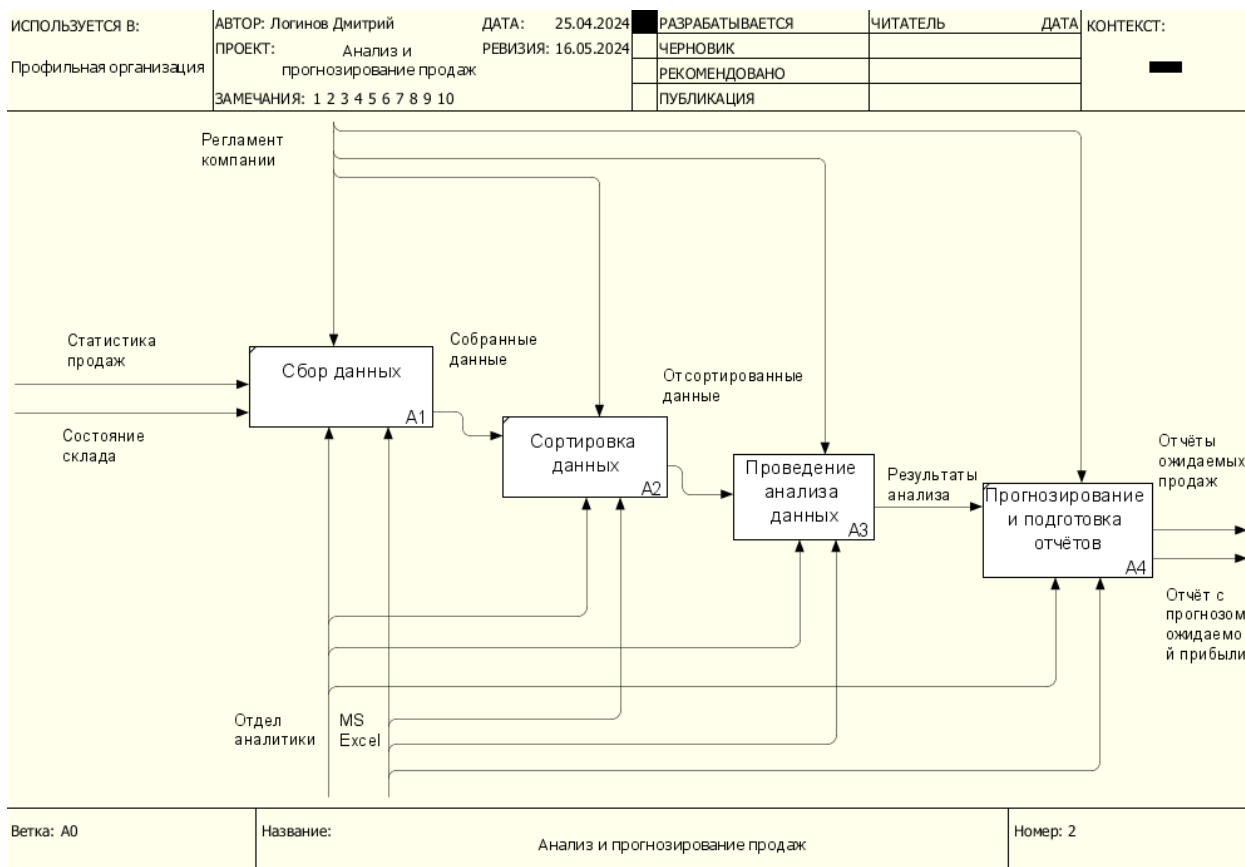


Рисунок 3 – Детализация контекстной диаграммы «Анализ и прогнозирование продаж»

Деятельность аналитического отдела профильной организации представлена в четырех этапах:

1. «Сбор данных» – процесс сбора всей необходимой для анализа информации, включающая историю продаж организации, состояние склада и ситуацию на рынке.
2. «Сортировка» – процесс ручной обработки данных и проверки их на точность. Этот этап повышает целостность и качество данных. На выходе получают чистые данные, подготовленные к анализу.
3. «Аналитика» – процесс математической обработки чистых данных. Является важнейшим этапом всего процесса, декомпозиция которого представлена на рисунке 4.

4. «Прогнозирования и подготовка отчётов» – процесс, включающий в себя прогнозирование и визуализацию результатов анализа в графическом и письменном виде.

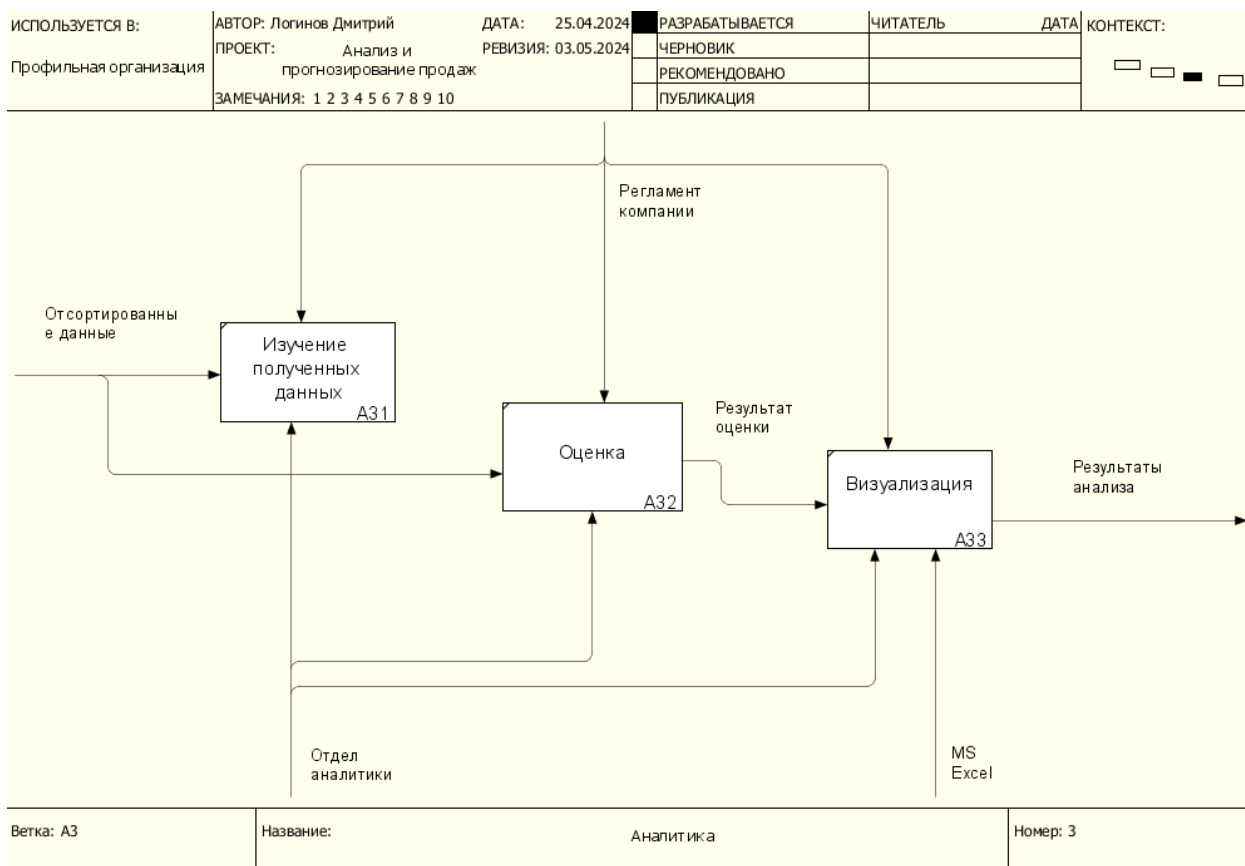


Рисунок 4 – Детализация контекстной диаграммы «Аналитика»

Процесс анализа данных разделен еще на 3 этапа:

- «Изучение полученных данных» – процесс изучения и преобразования имеющихся данных в цифровые значения, пригодные для математического анализа.
- «Оценка» – процесс применения алгоритмов ручного математического анализа данных для возможности прогнозирования данных.
- «Визуализация» – процесс прогнозирования предстоящих события на основании полученных данных.

Благодаря построению модели в предыдущем этапе можно провести сравнительный анализ. Мое предложение заключается в интеграции современных инструментов анализа данных в бизнес-процесс организации. Таким образом получится не только значительно уменьшить нагрузку на сотрудников предприятия, но и увеличить скорость обработки большого объема данных. Для расчёта эффективности будем использовать формулу интенсивности труда, далее ИТ.
$$ИТ = \frac{\text{Человеко-часы}}{\text{Объем работы}}. \quad [14]$$
 Оценка эффективности предложенного метода представлена в таблице 1.

Таблица 1 – Сравнительный анализ моделей процесса аналитики и прогнозирования данных организации

| Модель | Количество человеко-часов, необходимое для выполнения задачи | Объем работы | Коэффициент интенсивности труда |
|---|--|--------------|---------------------------------|
| Модель, использующая традиционные методы анализа данных | 35 | 100 | 0,35 |
| Модель, использующая современные методы анализа данных | 10 | 100 | 0,10 |

В результате сравнения можно заметить, что предложенная модель с использованием современных методов анализа данных более чем в три раза эффективней текущей. В связи с этим добавляем новую подсистему в нашу организацию – «Блок анализа данных с использованием машинного обучения», представленную на рисунке 5.

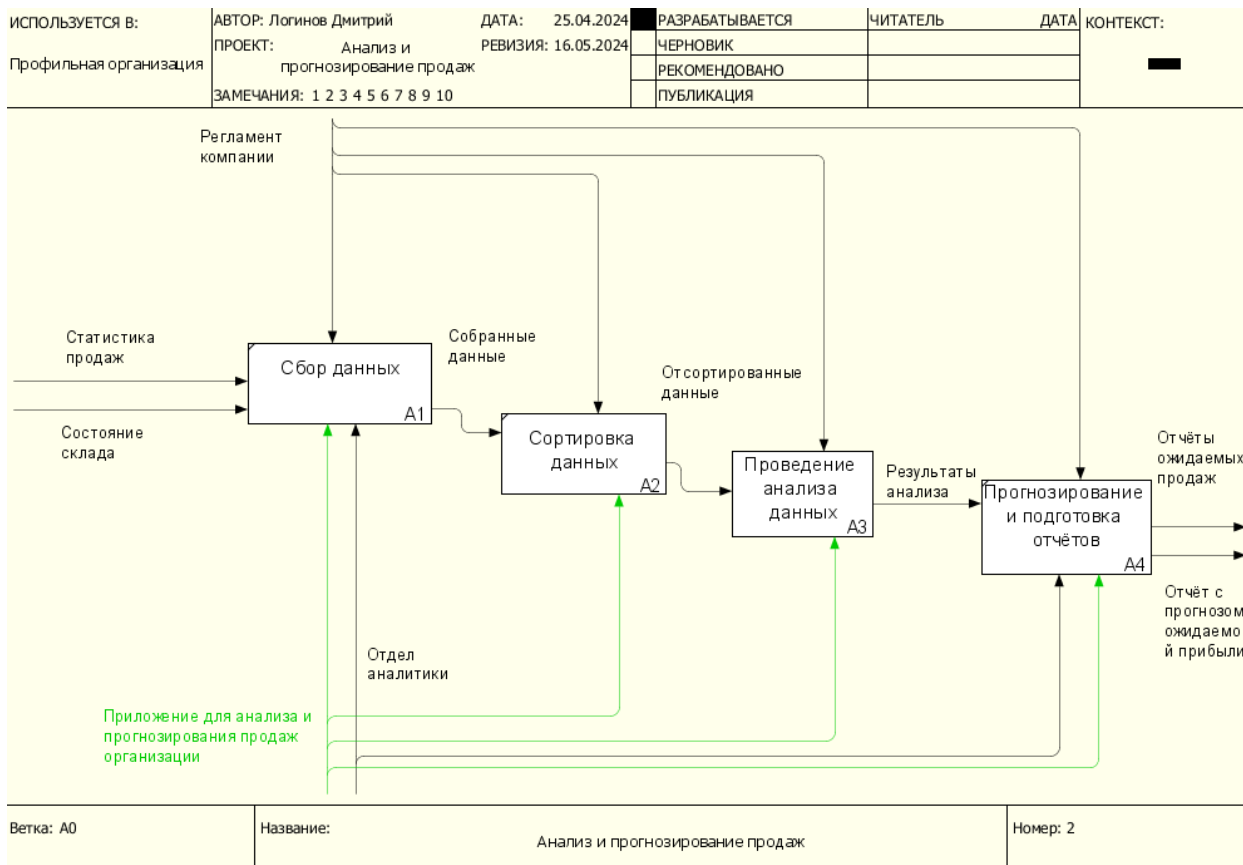


Рисунок 5 – Контекстная диаграмма бизнес-процесса «Анализ и прогнозирование продаж» ТО-ВЕ

На диаграмме видно, что приложение для анализа и прогнозирования продаж сильно оптимизирует бизнес-процесс. Благодаря такому механизму организация может уменьшить количество сотрудников для исполнения поставленной задачи, а также оптимизировать затрачиваемое время и точность результатов. [13]

1.4 Формирование требований

Программное обеспечение для анализа и прогнозирования продаж организации должно удовлетворять требования потребителя. В данном случае должен быть составлен грамотный и точный алгоритм, задачей

которого является обрабатывать большие объемы данных за короткий промежуток времени. [15]

Алгоритм разработки показан на рисунке 6.

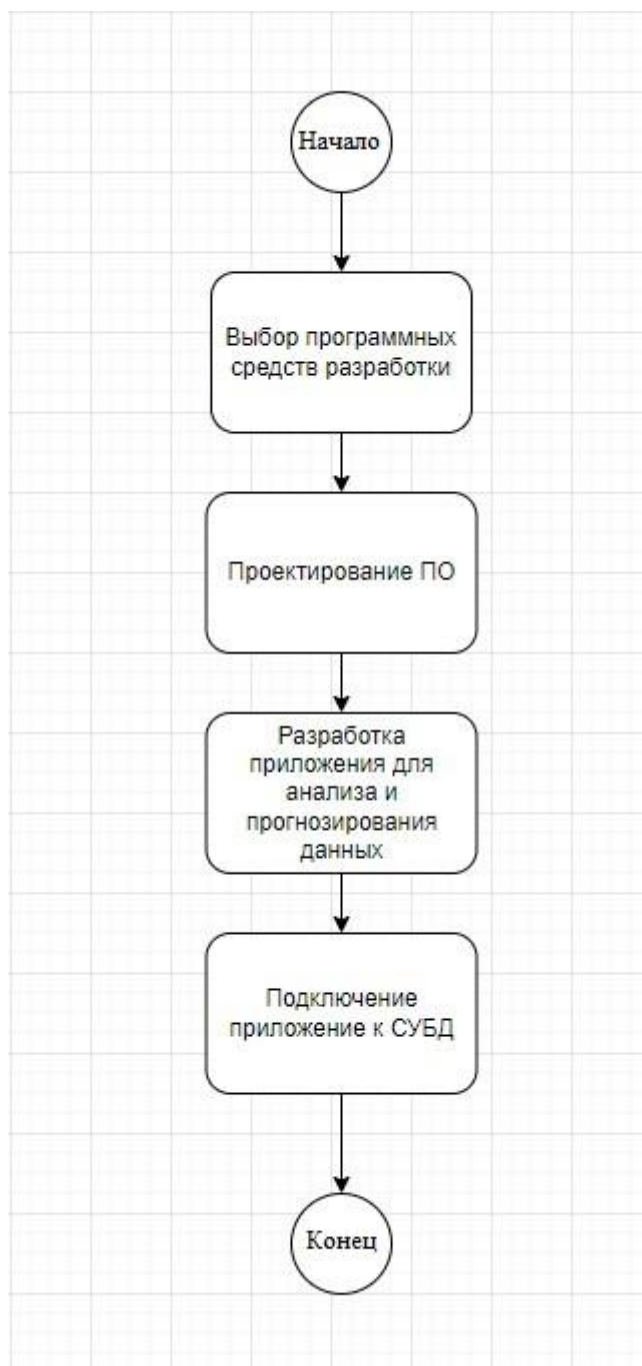


Рисунок 6 – Алгоритм разработки

Для этого необходимо выбрать наиболее подходящие средства разработки программного продукта. Важнейшей же частью является процесс анализа данных. Система должна автоматически принимать внешние данные с возможностью их нормализации. Во время анализа система должна учитывать все необходимые меры безопасности. Для этого необходимо создать конфигурацию, в которой будет использоваться последняя версия выбранного языка программирования, а также только проверенные и безопасные пакеты. Система также должна быть защищена от инъекций баз данных, поэтому необходимо передавать только параметризованные запросы или ORM. Программное обеспечение должно быть устойчивым к различным видам атак и легко восстанавливаемым. Система должна эффективно использовать выделенные ей ресурсы. В случае если физическая часть системы выйдет из строя, то нужна возможность быстрого переноса программного обеспечения на новую рабочую систему. [25]

На основании подготовленной информации составим список требований к ПО. [20]

Функциональные требования:

- Требование 1: Система должна собирать данные в блок анализа;
- Требование 2: Система должна классифицировать, кодировать и нормализовать данные для последующего анализа.
- Требование 3: Система должна анализировать данные и прогнозировать их для выявления ключевых трендов и тенденций;
- Требование 4: Система должна генерировать отчеты, содержащие: прогноз о выручке, прогноз о полном количестве продаж, прогноз о продажах по категориям.

Нефункциональные требования:

- Требование 1: Система должна быть доступной для пользователей с различными уровнями технических навыков;

- Требование 2: Система должна быть безопасной;
- Требование 3: Система должна быть масштабируемой, чтобы поддерживать потенциальное увеличение количества обрабатываемых данных;
- Требование 4: Система должна быть отказоустойчивой, чтобы обеспечить непрерывную работу в случае сбоев;
- Требование 5: Система должна быть совместимой с различными операционными системами и браузерами.

Выводы к главе 1

Первая глава была посвящена определению предметной области, определению модели бизнес-процесса, а также формулировке требований к программному обеспечению.

Использование моделей «AS-IS» и «TO-DO», а также проведение сравнительного анализа привело нас к выводу, что для улучшения эффективности бизнес-процесса «Анализ и прогнозирование продаж» в профильной организации необходимо использовать современный алгоритм, содержащий инструменты для анализа большого объема данных, включающие технологии машинного обучения.

По результатам анализа были составлены требования для программного обеспечения, основанные на современных запросах от организаций для использования системы анализа и прогнозирования продаж.

Полученный набор требований будет служить основой для реализации программного решения, целью которого является разработка системы анализа и прогнозирования продаж для профильной организации.

Глава 2 Логическое проектирование приложения

2.1 Описание инструментов для анализа и прогнозирования данных, использующих машинное обучение

Анализ и понимание исторических данных о продажах являются ключевыми шагами для определения будущего спроса. Используя библиотеки Python, такие как NumPy, Pandas и Scikit-learn, аналитики могут проводить детальный анализ данных и строить модели прогнозирования спроса.[3] Например, применение методов временных рядов, таких как Prophet или ARIMA, позволяет предсказать будущие тренды и сезонные колебания спроса. Кроме того, методы машинного обучения, включая регрессию и нейронные сети, могут быть использованы для более точного прогнозирования спроса на основе различных факторов, таких как маркетинговые активности или экономические показатели. [1]

Оптимизация ассортимента продукции включает в себя выбор наиболее подходящего набора товаров для предложения потребителям. Задача оптимизации может быть сформулирована как задача линейного программирования или смешанного целочисленного программирования. Для решения таких задач можно использовать библиотеки, включая PuLP или Gurobi. Эти инструменты позволяют определить оптимальный набор товаров, учитывая ограничения, такие как бюджет, доступность или предпочтения клиентов, и максимизировать ожидаемую прибыль или другие ключевые показатели.

Для успешного планирования ассортимента необходимо учитывать актуальные данные о наличии товаров, продажах и других факторах, влияющих на планирование.

Интеграция программного обеспечения с системами управления складом и продажами играет важную роль в этом процессе. Используя API и стандартные протоколы связи, такие как REST API, данные могут быть

синхронизированы между различными системами, обеспечивая достоверность и актуальность информации, а также автоматизируя процессы планирования и управления запасами. [2]

Визуализация данных является мощным средством для понимания результатов анализа и планирования ассортимента. [6] Библиотеки для визуализации данных, такие как Matplotlib, Seaborn или Plotly, позволяют создавать графики, диаграммы и дашборды, которые помогают визуализировать результаты прогнозирования спроса, оптимизации ассортимента и других аналитических выводов.[4] Кроме того, инструменты для создания отчетов, такие как Jupyter Notebook или Power BI, обеспечивают возможность создания интерактивных отчетов и дашбордов, которые можно использовать для мониторинга и принятия решений на основе данных о планировании ассортимента. [5]

Для обработки больших объемов данных, масштабирования и обеспечения высокой доступности программного обеспечения для планирования ассортимента, рекомендуется использовать облачные сервисы, такие как Amazon Web Services (AWS) или Microsoft Azure. Эти сервисы предоставляют гибкую и масштабируемую инфраструктуру, которая позволяет запускать и масштабировать вычислительные ресурсы, хранить и обрабатывать большие объемы данных, а также обеспечивать высокую доступность и надежность системы планирования ассортимента.

Стоит выделить определение анализа данных и прогнозирования спроса – это процессы, основанные на обработке и анализе исторических данных, с целью выявления закономерностей и трендов, а также предсказания будущего спроса. Применение методов аналитики и машинного обучения позволяет автоматизировать и улучшить этот процесс, обеспечивая более точные и надежные прогнозы.

При этом, с целью проведения анализа данных необходимо выявить методы для проведения анализа данных.

Анализ данных включает в себя различные методы и техники, направленные на извлечение значимых информации из исторических данных о продажах и других факторах, влияющих на спрос. Примеры таких методов включают:

- Исследование временных рядов: Методы анализа временных рядов, такие как экспоненциальное сглаживание, авторегрессионные модели (ARIMA) и сезонные модели, позволяют выявлять сезонные колебания, тренды и цикличность в данных, что важно для прогнозирования будущего спроса.
- Регрессионный анализ: Регрессионные модели позволяют установить связь между зависимыми и независимыми переменными и использовать их для прогнозирования спроса на основе различных факторов, таких как цены, маркетинговые активности и экономические показатели.
- Кластерный анализ: Кластерный анализ позволяет группировать схожие товары или клиентов на основе их характеристик и поведения, что помогает более точно понять структуру спроса и разработать более целевые стратегии маркетинга.
- Применение аналитики и машинного обучения в планировании ассортимента продукции открывает новые возможности для оптимизации бизнес-процессов, повышения эффективности и увеличения прибыли. Библиотеки, инструменты и интеграции предоставляют средства для анализа данных, прогнозирования спроса, оптимизации ассортимента и визуализации результатов.

Таким образом, при использовании классификационных методов можно выявить влияние различных факторов на определенные категории. Регрессионные методы могут использоваться для предсказания числовых значений. [7]

С помощью машинного обучения можно создать предсказательные модели для прогнозирования будущих тенденций и результатов.

В совокупности эти инструменты обеспечат нас всеми необходимыми средствами для тщательного анализа данных, их визуализации и применения методов машинного обучения для получения ценной информации.

2.2 Проектирование статической структуры приложения

Диаграмма классов (class diagram) – это диаграмма, предназначенная для представления модели статической структуры программной системы в терминологии классов объектно-ориентированного программирования.

Диаграмма классов применяется для отображения статической архитектуры системы и основана на классах, характерных для объектно-ориентированного программирования. Эта диаграмма демонстрирует связи между разными сущностями в областях, включая объекты и их группировки. Также выявляет структуру этих сущностей, включая атрибуты, поведение и разнообразные связи между ними, такие как иерархия и интерфейсы. Важно отметить, что классовая диаграмма не отражает динамические процессы системы и отличается от диаграммы последовательности отсутствием элементов временной зависимости, таких как начальные и конечные точки, разветвления или параллельные действия. Вместо этого, она сфокусирована на классах и их взаимодействиях. [18]

В объектно-ориентированных информационных системах классы являются фундаментальными блоками. Они определяют набор объектов, объединённых общими свойствами, поведением, взаимоотношениями и предназначением. Классовые диаграммы неотъемлемо входят в процесс проектирования, способствуя разбору и пониманию основных элементов и их связей в рамках проекта. Они служат для создания терминологии, отражающей важнейшие концепции и отношения в системе. Классы могут отражать как абстрактные идеи, так и конкретные программные или технические детали, лежащие в основе разработки. [22]

Рассмотрим основные элементы диаграммы классов.

Класс – это описание набора объектов с одинаковыми атрибутами.

Атрибут (свойство) – это именованное свойство класса, описывающее диапазон значений, которые может принимать экземпляр атрибута.

Операция (метод) – это реализация метода класса. Класс может иметь любое число операций либо не иметь ни одной.

Объект – это элемент, который может представлять собой как физический предмет в реальности, так и теоретическую сущность. Он обладает ясно выраженными контурами в рамках системы, имеет свою ценность, состояние, индивидуальность и специфическое поведение.

Состояние объекта – определенное положение, в котором находится объект. С течением времени его положение может меняться, что обусловлено изменениями в характеристиках, известных как атрибуты.

Поведение объекта – это метод, с помощью которого объект отвечает на внешние воздействия и устанавливает набор операций, которые он способен осуществлять самостоятельно.

Индивидуальность объекта – это особенность, которая позволяет каждому элементу в системе быть уникальным, даже если их характеристики схожи. Благодаря этому качеству, каждый элемент обладает своей уникальностью и может быть точно распознан в пределах системы.

Стереотип – это процесс, который облегчает группировку классов в различные группы и формирование новых видов классов. Данный процесс дает возможность назначать специфические свойства и функции классам, что упрощает их характеристику и управление ими.

Управляющий класс – это тип класса, задача которого — управлять взаимодействиями между другими классами внутри системы. Он регулирует порядок происходящих событий и управляет действиями, связанными с различными сценариями, гарантируя эффективную работу системы.

Класс в графическом представлении представляется в виде прямоугольника, разделенного на три горизонтальных блока. Первый блок

содержит имя класса, второй - атрибуты или свойства класса, а третий - операции или методы класса.

Имя каждого класса должно быть уникальным и может содержать буквы, цифры и знаки препинания, за исключением двоеточия и точки. Это текстовая строка, которая может быть записана на нескольких строках.

Кроме описания свойств созданных классов, для формирования информационной системы необходимо установить между ними логические связи. [17]

Далее рассмотрим основные виды связей, используемых в нашей диаграмме.

Ассоциация (association) — произвольное отношение или взаимосвязь между классами.

Агрегация (aggregation) — уникальный тип связи, который представляет собой структурное соотношение между целым и его составными элементами. Такая связь обозначается как именованное соединение между вместилищем и его содержимым.

Композиция (composition) — строгий вариант агрегации, также известный как агрегация по значению.

Зависимость (dependency) — особый тип связи, который демонстрирует необходимость одного или множества элементов в других элементах для их определения или создания.

Такая зависимость дает возможность отражать взаимосвязи между классами, интерфейсами, пакетами и прочими компонентами модели. Она подразумевает, что функционирование или создание одного элемента модели обусловлено наличием другого элемента.

Обобщение — это процесс, который обеспечивает возможность отражения иерархии и отношений наследования, при котором производный элемент (наследник) формируется на базе определений базового элемента (предка). Наследник наследует структуру и поведение предка, что дает возможность строить классовые иерархии и детализировать их атрибуты и

методы. Этот процесс позволяет разрабатывать универсальные классы для использования в разнообразных сегментах системы, а также воплощать принципы наследования и инкапсуляции.

Реализация — это тип взаимодействия между классами, при котором один класс (изготовитель) устанавливает набор правил, следование которым является обязательным для другого класса (потребителя). Этот вид связи устанавливается между интерфейсами и классами, осуществляющими эти интерфейсы. Она обеспечивает возможность установления связей между абстрактными классами и их конкретными реализациями, а также между интерфейсами и классами, реализующими их. В UML реализация часто применяется для описания отношений между абстракциями и их реальными реализациями, а также для разграничения функций и обязанностей среди классов. [16]

Диаграмма классов для системы анализа и прогнозирования продаж организации представлена на рисунке 7.

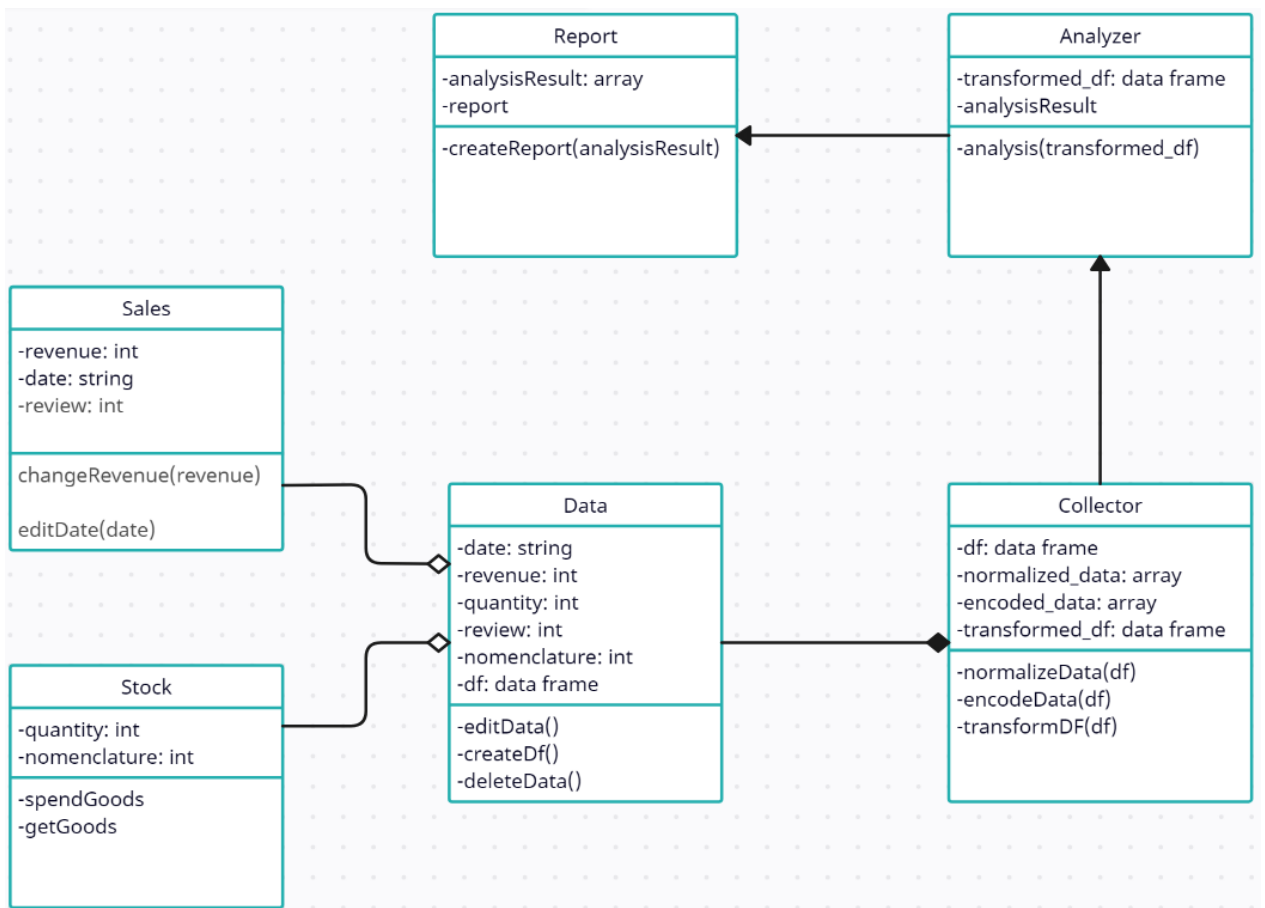


Рисунок 7 – Диаграмма классов

Таблица 2 – Перевод терминов диаграммы

| Термин на диаграмме | Перевод |
|---------------------|------------|
| Report | Отчёт |
| Analyzer | Аналитик |
| Collector | Сборщик |
| Data | Данные |
| Sales | Продажи |
| Stock | Склад |
| Revenue | Выручка |
| Date | Дата |
| Review | Отзыв |
| Quantity | Количество |

Продолжение таблицы 2

| Термин на диаграмме | Перевод |
|---------------------|------------------------|
| Nomenclature | Номенклатура |
| Data Frame | Дата фрейм |
| Normalized Data | Нормализованные данные |
| Encoded Data | Закодированные данные |
| Analysis Result | Результат анализа |
| Report | Отчёт |
| Integer | Целочисленный |
| String | Строка |
| Array | Массив |

Система анализа данных состоит из большого количества классов. На примере диаграммы, мы можем рассмотреть каждый процесс более подробно.

На рисунке 8 представлен класс «Sales» (продажи). Он имеет 3 атрибута: «price», «date» и «review», а также 2 метода: `changeRevenue(dynamic)` и `editDate(date)`.

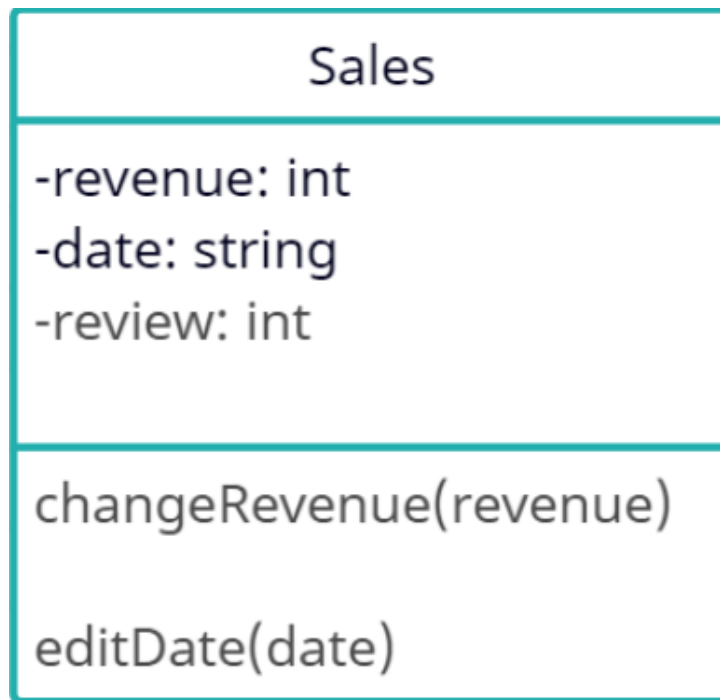


Рисунок 8 – Класс «Sales»

Описание атрибутов класса «Sales»:

- атрибут «price» представляет из себя данные о денежном эквиваленте проданных товаров и имеет целочисленный тип данных;
- атрибут «date» несет в себе данные о дате предыдущих продаж и имеет тип данных строка;
- атрибут «review» несет в себе данные об отзывах пользователей и имеет целочисленный тип данных.

Описание методов класса «Sales»:

- метод «changeRevenue» выполняет функцию отслеживания изменения динамики продаж и использует атрибут «Revenue» в качестве аргумента.
- метод «editData» выполняет функцию изменения данных датах продажи и использует атрибут «Date» в качестве аргумента.

Класс «Sales» имеет связь с классом «Data» в виде агрегации.

Теперь рассмотрим класс «Stock» (Склад), представленный на рисунке 9. Он имеет 2 атрибута: «nomenclature» и «quantity», а также 2 метода: spendGoods(quantity) и getGoods(quantity).

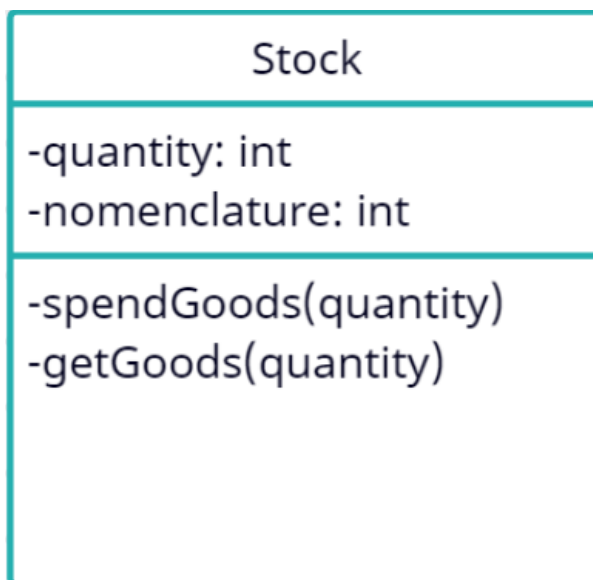


Рисунок 9 – Класс «Stock»

Описание атрибутов класса «Stock»:

- атрибут «nomenclature» представляет из себя данные о перечне товаров, услуг или работ, используемых в организации, имеет тип строковый тип данных;
- атрибут «quantity» несет в себе данные о количестве и состоянии товара на складе, имеет целочисленный тип данных.

Описание методов класса «Stock»:

- метод «spendGoods» выполняет функцию отслеживания убывания товара со склада и использует атрибут «quantity» в качестве атрибута;
- метод «getGoods» выполняет функцию отслеживания прибытия товаров на склад и использует атрибут «quantity» в качестве аргумента.

– Класс «Stock» имеет связь с классом «Data» в виде агрегации.

На рисунке 10 представлен класс «Data» (данные). Он является основной базой данных во всей системе и имеет уникальный атрибут «df». Этот класс самостоятельный, но имеет связи с классами «Stock» и «Sales» в виде агрегации, а также имеет зависимую связь с классом «Collector» в виде композиции.

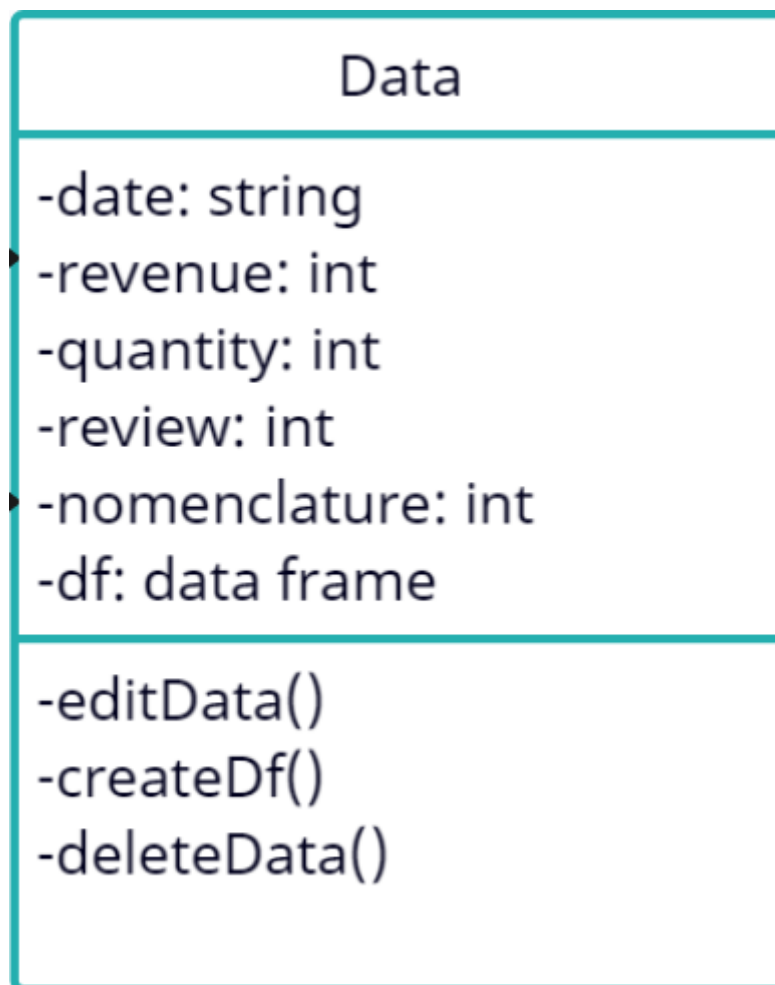


Рисунок 10 – Класс «Data»

Атрибуты «revenue», «date», «review», «quantity» и «nomenclature» передаются из классов «Stock» и «Sales».

Описание атрибутов класса «Data»:

– атрибут «df» является созданным из всех собранных данных дата фреймом;

Описание методов класса «Data»:

- метод «editData» выполняет функцию возможности редактирования данных, содержащихся в базе данных;
- метод «createDf» создает дата фрейм из всех собранных данных;
- метод «deleteData» позволяет удалять любые данные из системы.

Рассмотрим класс «Collector» (сборщик), представленный на рисунке 11. Этот класс является зависимым от «Data» и имеет тип связи в виде композиции.

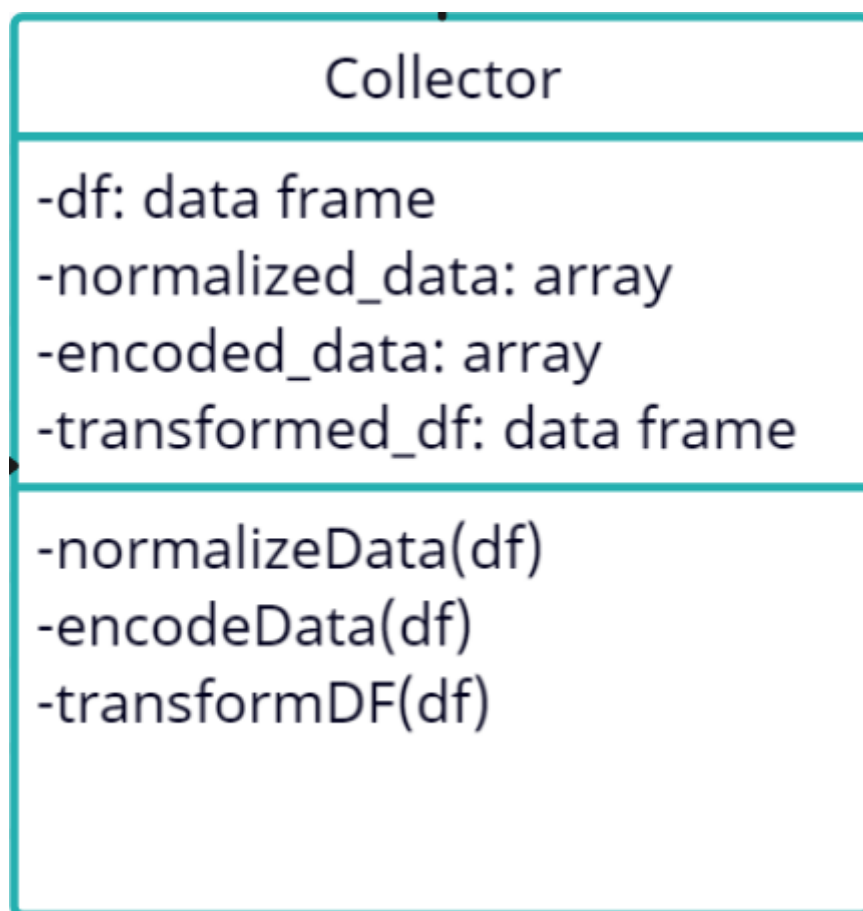


Рисунок 11 – Класс «Collector»

Класс «Collector» выполняет важную задачу в виде кодирования и нормализации данных, получаемых из базы данных.

Описание атрибута класса «Collector»:

- атрибут «normalized_data» является результатом работы метода «normalizeData» и содержит в себе очищенные и нормализованные данные для последующего анализа. Представляет из себя массив;
- атрибут «encoded_data» содержит в себе закодированные данные и является результатом работы метода «encodeData». Представляет из себя массив массив;
- атрибут «transformed_df» является результатом работы метода «transformDf» и содержит подготовленные для анализа данные. Представляет из себя дата фрейм.

Описание методов класса «Collector»

- метод «normalizeData» выполняет нормализацию данных, а также делит их на цель и атрибуты для анализа, использует атрибут «df» в качестве аргумента;
- метод «encodeData» выполняет кодирование данных, непригодных для анализа и использует атрибут «df» в качестве аргумента;
- метод «transformDF» преобразовывает закодированные и нормализованные данные в дата фрейм, готовый к анализу.

Класс «Analyzer» (анализатор) выполняет лишь одну функцию — анализ данных. Он является наследником класса «Collector». Класс «Analyzer» представлен на рисунке 12.

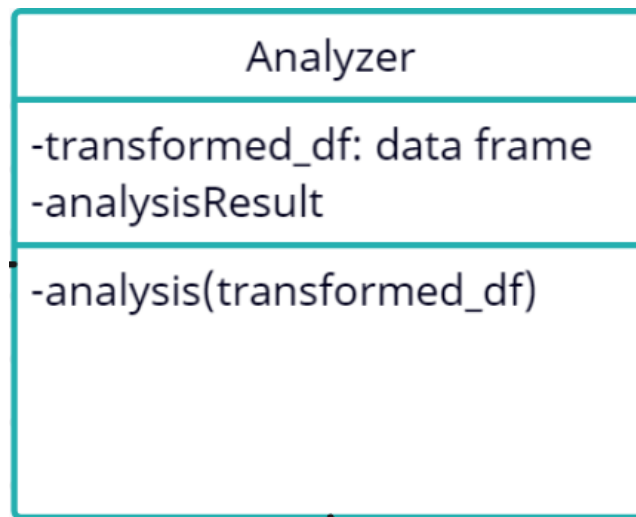


Рисунок 12 – Класс «Analyzer»

Описание атрибута класса «Analyzer»:

- атрибут «analysisResult» является результатом работы метода «analysis» и содержит в себе результат анализа предоставленных данных.

Описание метода класса «Analyzer»:

- метод «analysis» выполняет анализ данных и использует атрибут «transformed_df» в качестве аргумента.

Класс «Report» (отчёт) подводит итоги анализа и составляет отчёт с прогнозами продаж организации. Является наследником класса «Analyzer». Класс «Report» представлен на рисунке 13.

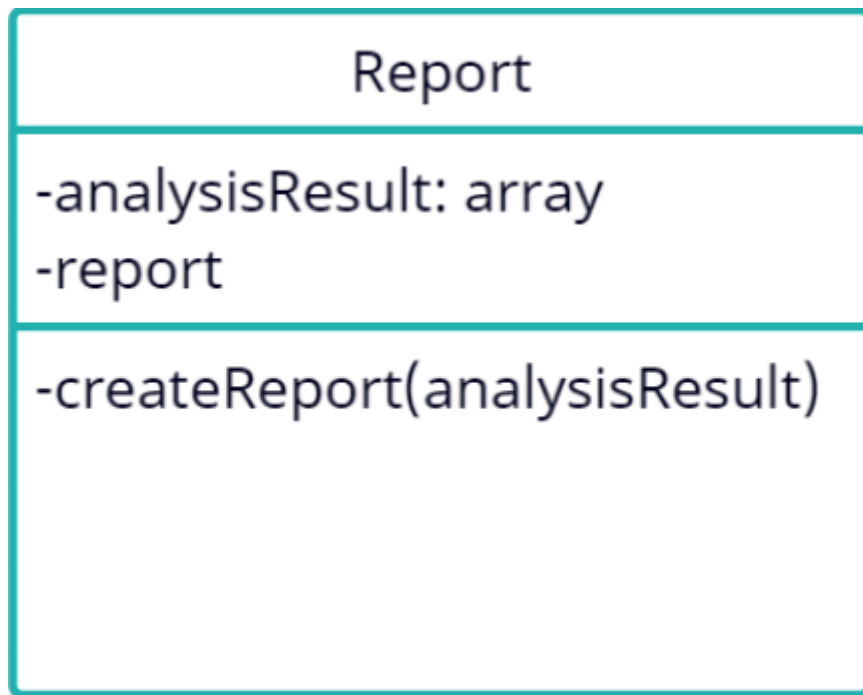


Рисунок 13 – Класс «Report»

Описание атрибута класса «Report»:

- атрибут «report» является финальным продуктом всей систем. Он содержит отчёт с прогнозом продаж организации.

Описание метода класса «Report»:

- метод «createReport» составляет отчёт с прогнозом продаж организации по результатам данных с анализа. Использует атрибут «analysisResult» в качестве аргумента.

2.3 Выбор СУБД

Система управления базами данных, далее СУБД, является важнейшей частью любой информационной системы организации. СУБД представляет собой комплекс программно-языков средств, предназначенных для эффективного создания, управления и манипулирования данными в базах

данных. Также обеспечивает структурированное хранение и защиту данных, вместе с этим предоставляя пользователю инструменты для их обработки.

Выбор подходящей СУБД имеет ключевое значение в разработке ПО, ведь от него зависят производительность и удобство работы с данными.

Существует множество различных СУБД, каждая из которых предназначена для определенных задач и сценариев использования. Например, реляционные СУБД, основанные на модели реляционных таблиц, в которых данные хранятся в виде строк или столбцов, или noSQL-СУБД, предназначенных работы с неструктурированными данными, такими как JSON и XML документы. Существуют также In-Memory СУБД, которые хранят данные в оперативной памяти, что обеспечивает высокую скорость доступа. [21]

В нашем случае остановимся на реляционных СУБД. Благодаря гибкости, безопасности, а также возможности нормализации данных, они отлично подойдут к структурированной системе. Рассмотрим три примера реляционных СУБД: Oracle, MySQL и PostgreSQL. Для выбора необходимой для нас, воспользуемся диаграммой вариантов использования.

Диаграмма вариантов использования — это инструмент в языке моделирования UML, который помогает описать функциональные требования к системе путем визуализации всех потенциальных взаимодействий с системой. Она показывает, как пользователи, далее актёры, будут взаимодействовать с системой в различных ситуациях.

Вот основные элементы диаграммы вариантов использования:

- Варианты использования — представляют функциональные возможности системы, а также то, как пользователь будет с ней взаимодействовать;
- Актёры — пользователи и другие системы, которые взаимодействуют с вариантами использования;
- Связи — отношения между актёрами и вариантами использования.

Диаграмма вариантов использования представлена на рисунке 14.

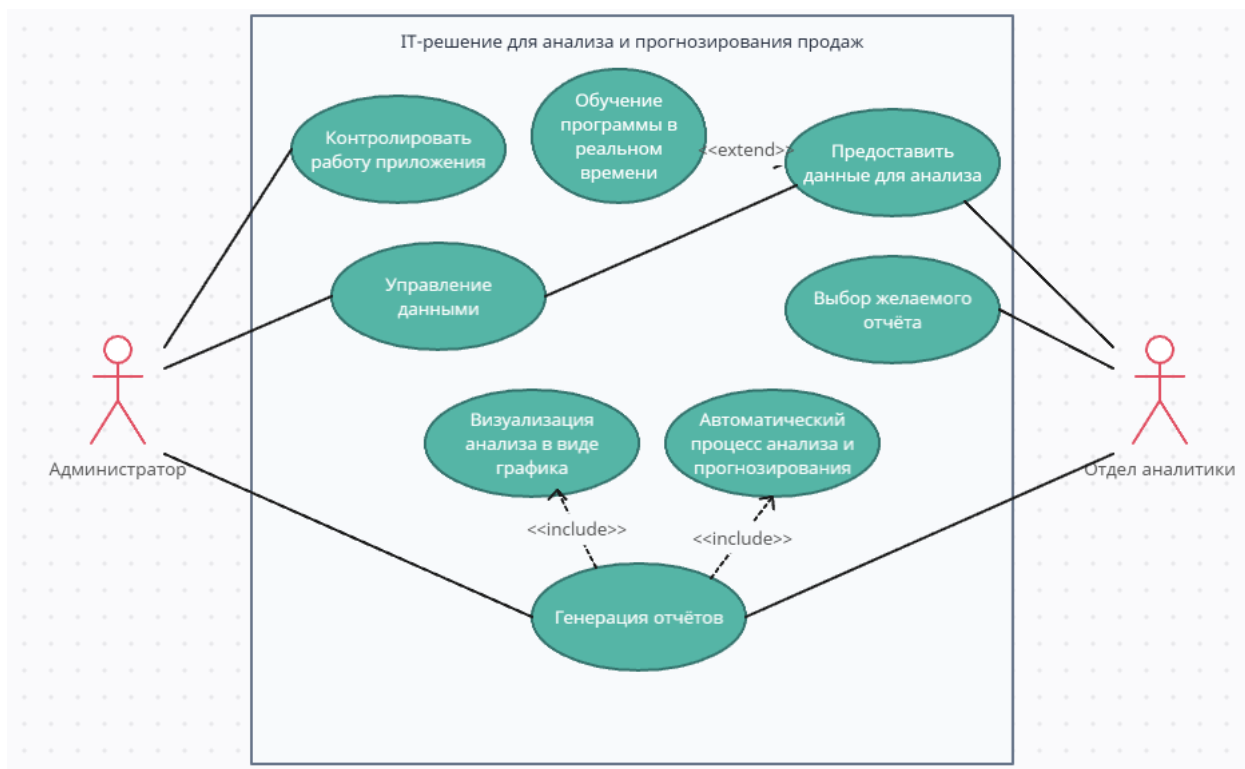


Рисунок 14 – Диаграмма вариантов использования приложения

На диаграмме можно увидеть сильную перегрузку СУБД, что говорит о том, что этот фрагмент должен быть особенно эффективным. СУБД должна обрабатывать запросы клиента в кратчайшие сроки.

В связи с этим можно выделить основные критерии для выбора базы данных: гибкость, эффективность, возможность нормализации, простота использования, совместимость, поддержка стандартов, безопасность, структуризация, OpenSource и масштабируемость.

Структуризация таблицы подразумевает собой используемую таблицей структуру для хранения данных.

Для финального выбора СУБД составим таблицу, в которой сравним все предложенные системы управления базами данных.

Таблица 3 - Сравнение СУБД

| Требование | MySQL | Oracle | PostgreSQL |
|--------------------------|-------|--------|------------|
| Гибкость | + | + | + |
| Эффективность | + | + | - |
| Возможность нормализации | + | + | + |
| Простота использования | + | + | - |
| Совместимость | + | - | - |
| Поддержка стандартов | + | - | - |
| Безопасность | + | + | + |
| Структуризация | + | - | + |
| OpenSource | + | - | + |
| Масштабируемость | + | - | + |
| Итого | 10 | 5 | 6 |

По результатам анализа видно, что MySQL имеет ряд преимуществ над остальными реляционными СУБД.

MySQL представляет собой открытую реляционную базу данных, созданную для управления информацией с высокой эффективностью. Эта система позволяет организовать, контролировать и работать с данными, используя универсальный язык запросов SQL. Она совместима с различными операционными системами, в том числе с Windows.

Выводы к главе 2

Во второй главе были описаны современные инструменты анализа и прогнозирования данных, а также тонкости их использования. Были определены программные средства для разработки приложения.

Была спроектирована статическая структура системы. Благодаря диаграмме классов, визуализировали все особенности разрабатываемой системы, а также описали каждый класс, его атрибуты и методы. Это поможет нам создать более надежную и эффективную систему и лучше разобраться в её объектно-ориентированной структуре.

Помимо этого мы построили диаграмму вариантов использования, благодаря которой были описаны функциональные требования к системе. Далее, с помощью сравнения была выбрана СУБД.

Выбранные инструменты программной разработки будут необходимы для реализации программного обеспечения.

Глава 3 Физическое проектирование приложения

3.1 Разработка приложения

Для разработки программного обеспечения будет использоваться бесплатная среда разработки от JetBrains.

«PyCharm 2021 Community Edition» является одной из мощнейших IDE на рынке, созданных специально для разработки на языке Python. В его функции входит автоматическое заполнение кода, что позволит экономить время, а также снизить количество ошибок. Благодаря своим интеллектуальным функциям, обширной экосистеме плагинов и мощным инструментам разработки PyCharm является лучшим выбором для разработки приложения.

Первым этапом разработки является создание структуры проекта, визуализация которой представлена на рисунке 15.

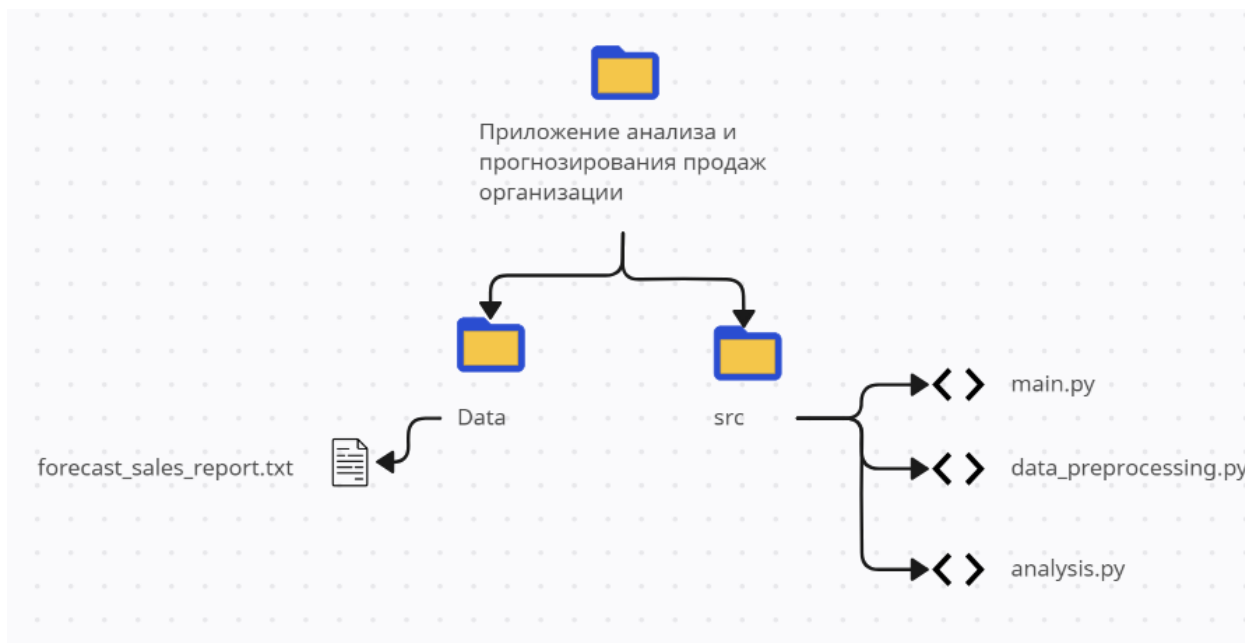


Рисунок 15 – Структура проекта

Проект содержит в себе две директории: «Data» и «src». В первой содержится вся генерируемая приложением информация в виде отчётов и графиков. В «src» содержится весь код программы.

Следующим этапом является создание и заполнение базы данных. Для реализации этой части использовалось бесплатное программное обеспечение «MySQL Workbench 8.0». Далее была создана и заполнена таблица, включающая в себя 10 столбцов с информацией о продажах организации.

Теперь необходимо установить соединение приложения с базой данных. Для реализации этого была использована библиотека «MySQL Connector», а затем «Pandas» для создания и обработки дата фрейма.

Для кодирования информации был применен «One-Hot Encoder». Принцип его работы заключается в преобразовании категориальных данных в числовой формат, который может быть использован для машинного обучения. Само преобразование достигается путем создания бинарных векторов, где каждый уникальный категориальный признак представлен отдельным столбцом.

Затем дата фрейм был разбит на две части: цель и атрибуты. Цель — это переменная, которую необходимо спрогнозировать методами машинного обучения, в основе которого лежит идея, что машина может учиться из данных, находить закономерности и принимать решения с минимальным вмешательством человека.

В машинном обучении существует множество различных алгоритмов. В приложении была выбрана модель обучения «случайного леса». Это алгоритм, который использует ансамбль деревьев решений для прогнозирования непрерывных значений. Каждое дерево в случайном лесу обучается на разных выборках данных. Кроме того, при построении каждого узла дерева алгоритм выбирает случайное подмножество признаков, что делает модель более устойчивой к переобучению. [23] Реализация модели случайного леса показана на рисунке 16.

```

from sklearn.metrics import mean_absolute_error
from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestRegressor

def analysis(attributes, target, df):
    # Разделение данных на обучающий и тестовый наборы
    attributes_train, attributes_test, target_train, target_test = train_test_split(attributes, target,
                                                                                      test_size=0.2,
                                                                                      random_state=42)

    # Инициализация и обучение модели случайного леса
    model = RandomForestRegressor(n_estimators=100, random_state=42)
    model.fit(attributes_train, target_train)

    # Предсказание значений целевой переменной
    target_prediction = model.predict(df)
    mae = model.predict(attributes)
    print('Коэффициент средней абсолютной ошибки MAE: ', mean_absolute_error(target, mae))
    return target_prediction

```

Рисунок 16 – Реализация модели «случайного дерева»

Для реализации «случайного леса» использовалась библиотека «scikit-learn». Данные были разбиты на «attributes_train», «attributes_test», «target_test» и «target_train».

Описание переменных:

- «attributes_train» — это часть данных, используемая для обучения модели. Она содержит переменные, которые применяются для изучения и выявления закономерностей;
- «attributes_test» представляет из себя данные, которые используются для тестирования модели после обучения;
- «target_train» является ключевыми зависимыми переменными и целевыми значениями для предсказания значений;
- target_test — это данные, используемые для оценки точности модели.
- Далее происходит настройка модели. Параметр «n_estimators» определяет количество деревьев решений в ансамбле случайного леса. Увеличение количества значений может улучшить качество модели, однако негативно скажется на времени обработки

информации. Параметр «random_state» используется для инициализации генератора случайных чисел, определяющий случайность выбора данных для обучения деревьев, а также признаков при разделении узлов.

Для проверки модели был использован метод средней абсолютной ошибки, далее MAE. MAE представляет из себя метрику для измерения точности в задачах регрессии. Он вычисляется как среднее абсолютное значение разностей между предсказанными и истинными значениями. [24]

Заключительным этапом разработки является подготовка отчётов. Модель получает на вход данные для анализа, в результате которых предсказывает значения. На выходе генерирует отчёт с прогнозом, график и информацию о средней абсолютной ошибке.

3.1 Тестирование приложения

Используем метод функционального тестирования чтобы проверить соответствие приложения к заявленным требованиям и спецификациям.

Необходимо проверить, выполняет ли приложение все свои функции правильно.

Первым требованием к приложению был сбор данных в блок анализа. Данные из СУБД, представленные на рисунке 17. Передаются в дата фрейм, обработав который вся необходимая информация передается в блок анализа.

| Date | Revenue | Nomenclature | Review | Pipes | Constructions | Hardware | Stainless | Grid | Quantity |
|-----------|----------|--------------|--------|-------|---------------|----------|-----------|------|----------|
| January | 79134.00 | 35 | 4.2 | 71 | 51 | 79 | 66 | 104 | 371 |
| February | 76331.00 | 39 | 6.1 | 100 | 107 | 43 | 48 | 116 | 414 |
| March | 55856.00 | 31 | 6.4 | 79 | 80 | 103 | 34 | 43 | 339 |
| April | 96578.00 | 32 | 5.4 | 80 | 38 | 98 | 85 | 100 | 401 |
| May | 96894.00 | 34 | 5.5 | 30 | 55 | 64 | 40 | 65 | 254 |
| June | 79244.00 | 37 | 4.7 | 50 | 103 | 69 | 77 | 65 | 364 |
| July | 81935.00 | 38 | 6.4 | 68 | 115 | 81 | 64 | 70 | 398 |
| August | 86956.00 | 39 | 7.6 | 37 | 89 | 86 | 42 | 31 | 285 |
| September | 85137.00 | 36 | 4.7 | 85 | 99 | 92 | 97 | 31 | 404 |
| October | 61674.00 | 30 | 8.2 | 54 | 51 | 76 | 41 | 62 | 284 |
| November | 61383.00 | 35 | 5.3 | 62 | 44 | 56 | 37 | 94 | 293 |
| December | 60052.00 | 31 | 4.2 | 100 | 104 | 68 | 60 | 98 | 430 |
| January | 56532.00 | 33 | 6.3 | 106 | 40 | 40 | 106 | 33 | 325 |
| February | 51656.00 | 38 | 9.9 | 75 | 77 | 79 | 48 | 44 | 323 |
| March | 55318.00 | 38 | 5.0 | 45 | 58 | 42 | 73 | 37 | 255 |
| April | 63140.00 | 38 | 8.0 | 92 | 68 | 64 | 41 | 37 | 302 |
| May | 75732.00 | 34 | 7.9 | 111 | 85 | 105 | 119 | 63 | 483 |
| June | 60546.00 | 35 | 4.6 | 108 | 36 | 109 | 67 | 95 | 415 |
| July | 59432.00 | 32 | 9.3 | 76 | 117 | 61 | 89 | 58 | 401 |
| August | 88416.00 | 37 | 7.9 | 108 | 67 | 107 | 34 | 71 | 387 |
| September | 57974.00 | 32 | 7.6 | 64 | 36 | 58 | 31 | 52 | 241 |

Рисунок 17 – Представление данных в СУБД

Второе требование заключалось в классификации, кодировании и нормализации данных. Это требование было выполнено в процессе обработки данных. Пример нормализованного дата сета представлен на рисунке 18.

| | 0 | 1 | 2 | 3 | 4 | 5 | ... | 15 | 16 | 17 | 18 | 19 | 20 |
|-----|-----|-----|-----|-----|-----|-----|-----|-------|-------|-------|-------|-------|-------|
| 0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | ... | 71.0 | 51.0 | 79.0 | 66.0 | 104.0 | 371.0 |
| 1 | 0.0 | 0.0 | 0.0 | 1.0 | 0.0 | 0.0 | ... | 100.0 | 107.0 | 43.0 | 48.0 | 116.0 | 414.0 |
| 2 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | ... | 79.0 | 80.0 | 103.0 | 34.0 | 43.0 | 339.0 |
| 3 | 1.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | ... | 80.0 | 38.0 | 98.0 | 85.0 | 100.0 | 401.0 |
| 4 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | ... | 30.0 | 55.0 | 64.0 | 40.0 | 65.0 | 254.0 |
| ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... |
| 163 | 0.0 | 1.0 | 0.0 | 0.0 | 0.0 | 0.0 | ... | 99.0 | 63.0 | 31.0 | 76.0 | 77.0 | 346.0 |
| 164 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | ... | 51.0 | 117.0 | 46.0 | 93.0 | 40.0 | 347.0 |
| 165 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | ... | 39.0 | 51.0 | 99.0 | 91.0 | 109.0 | 389.0 |
| 166 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | ... | 92.0 | 117.0 | 79.0 | 118.0 | 100.0 | 506.0 |
| 167 | 0.0 | 0.0 | 1.0 | 0.0 | 0.0 | 0.0 | ... | 46.0 | 103.0 | 95.0 | 54.0 | 78.0 | 376.0 |

Рисунок 18 – Нормализованный дата сет

Третье требование запрашивает от приложения анализ и прогноз данных. Анализ данных происходит благодаря методу случайного дерева, подробно описанным в пункте 3.1. Результатом анализа является массив с прогнозами, представленный на рисунке 19.

```
[80543.32 71362.74 70525.18 83172.16 69118.77 70597.49 79501.64 73536.9  
81174.43 73320.29 66578.48 77637.05]
```

Рисунок 19 – Массив с прогнозами

Четвертое требование заключается в необходимости генерации приложением отчётов о выручке и продажах. Это и является результатом работы приложения, представленные на рисунках 20, 21, 22, 23.

```
Отчет с прогнозом выручки за следующие 12 месяцев:  
Month Revenue  
Jan 80543.32  
Feb 71362.74  
Mar 70525.18  
Apr 83172.16  
May 69118.77  
Jun 70597.49  
Jul 79501.64  
Aug 73536.90  
Sep 81174.43  
Oct 73320.29  
Nov 66578.48  
Dec 77637.05
```

Рисунок 20 – Отчёт о прогнозируемой выручке

Отчет с прогнозом продаж на следующие 12 месяцев:

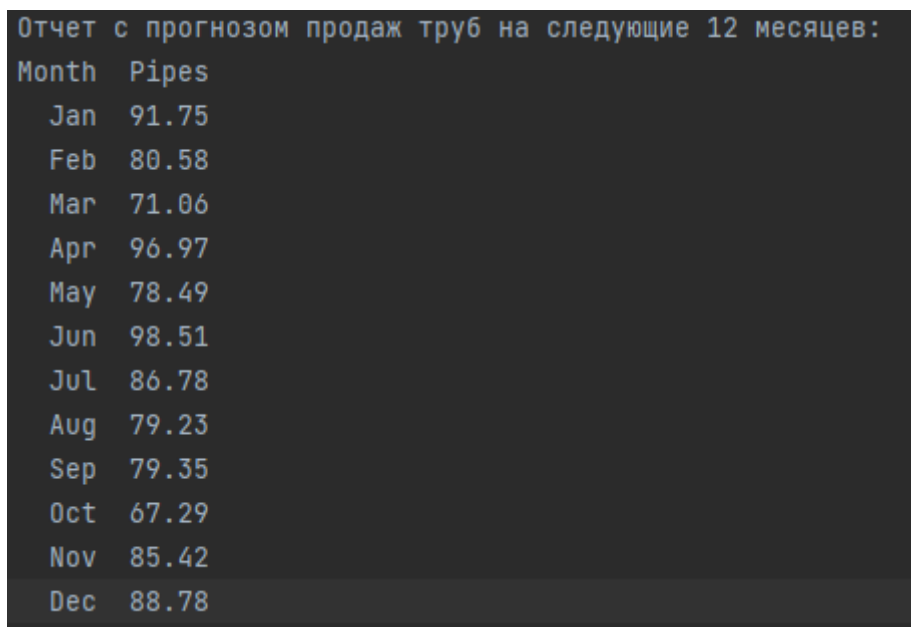
| Month | Quantity |
|-------|----------|
| Jan | 432.83 |
| Feb | 396.29 |
| Mar | 339.96 |
| Apr | 377.51 |
| May | 345.39 |
| Jun | 395.04 |
| Jul | 386.74 |
| Aug | 353.64 |
| Sep | 414.38 |
| Oct | 345.44 |
| Nov | 408.95 |
| Dec | 395.47 |

Рисунок 21 – Отчёт о прогнозируемых продажах



Рисунок 22 – График прогнозируемых продаж

Помимо представленных вариантов, имеется возможность генерации отчёта с прогнозом продаж конкретных категорий. Например, отчёт с прогнозом продаж труб представлен на рисунке 23.



```
Отчет с прогнозом продаж труб на следующие 12 месяцев:  
Month Pipes  
Jan 91.75  
Feb 80.58  
Mar 71.06  
Apr 96.97  
May 78.49  
Jun 98.51  
Jul 86.78  
Aug 79.23  
Sep 79.35  
Oct 67.29  
Nov 85.42  
Dec 88.78
```

Рисунок 23 – Отчёт о прогнозируемых продажах труб

Рассмотрим интерфейс приложения. Первым делом выводится сообщение о состоянии подключения к базе данных. Далее пользователю предлагается выбрать тип отчёта и временные рамки для прогноза, рисунок 24.

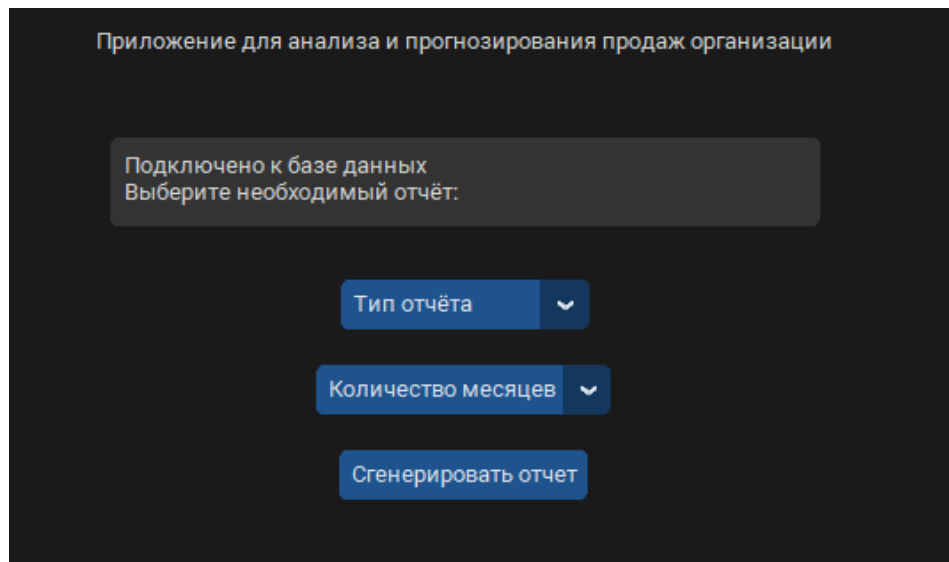


Рисунок 25 – Интерфейс приложения

После нажатия на кнопку «Сгенерировать отчёт», программа составляет и отображает выбранный отчёт и график, рисунок 26.

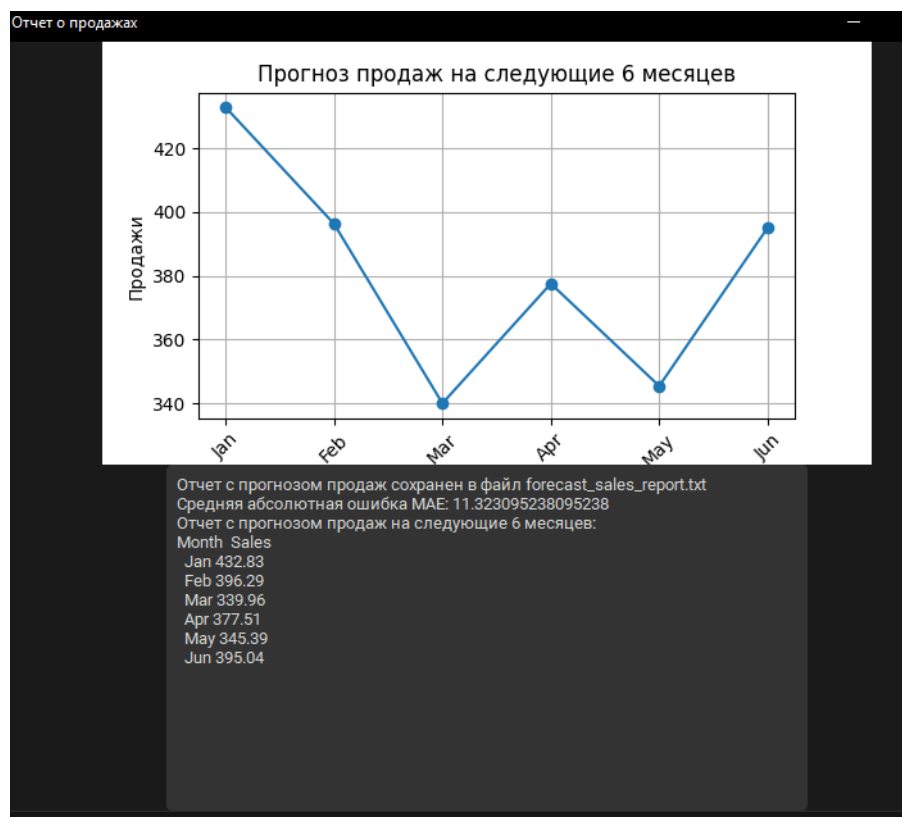


Рисунок 26 – Отчёт внутри приложения

Таким образом приложение проходит функциональное тестирование путем соответствия всем функциональным требованиям и готова к использованию организацией ООО «СМЦ «Металл Маркет».

Вывод по главе 3

Третья глава посвящена разработке и тестированию приложения по анализу и прогнозированию продаж. Были приведены методы разработки программного обеспечения, а также описаны процессы кодирования и нормализации дата фрейма.

Реализован и описан алгоритм машинного обучения для анализа и прогнозирования продаж организации.

Проведено функциональное тестирование приложения путем проверки его на соответствие заявленным функциональным требованиям.

Заключение

В данной выпускной квалификационной работе исследуется актуальная проблема, связанная с анализом и прогнозированием продаж ООО «СМЦ «Металл Маркет». Инструменты машинного обучения предоставляют аналитикам возможность более эффективного и точного прогнозирования данных.

Цель данной работы заключается в проектировании и разработке приложения, использующего современные инструменты для анализа и прогнозирования продаж организации. В процессе работы были решены следующие задачи:

- была дана характеристика ООО «СМЦ «Металл-Маркет»;
- было проведено исследование бизнес-процессов анализа и прогнозирования продаж организации ООО «СМЦ «Металл-Маркет»;
- сформулированы требования к приложению и выбрать средства разработки;
- спроектирована логическая модель приложения;
- разработано и проведено тестирование приложения.

Использованы модели «AS-IS» и «TO-BE», а также проведен сравнительный анализ, после чего был сделан вывод, что для улучшения эффективности бизнес-процесса «Анализ и прогнозирование продаж» в профильной организации необходимо использовать современный алгоритм, содержащий инструменты для анализа большого объема данных, включающие технологии машинного обучения.

По результатам анализа были составлены требования для программного обеспечения, основанные на современных запросах от организаций для использования системы анализа и прогнозирования продаж.

Была спроектирована статическая структура системы. Благодаря диаграмме классов, визуализировали все особенности разрабатываемой

системы, а также описали каждый класс, его атрибуты и методы. Это помогло создать более надежную и эффективную систему, а также разобраться в её объектно-ориентированной структуре.

Были приведены методы разработки программного обеспечения, а также описаны процессы кодирования и нормализации дата фрейма.

Реализован и описан алгоритм машинного обучения для анализа и прогнозирования продаж организации.

Проведено тестирование приложения, а также проверка его на соответствие заявленным функциональным требованиям.

В результате разработки данного IT-решения были реализованы функции, позволяющие аналитикам более эффективно строить прогнозы продаж организации.

Результаты данной бакалаврской работы представляют научно-практический интерес и могут быть рекомендованы бизнес-аналитикам и разработчикам приложений, занимающимся проблемами анализа и прогнозирования продаж организации.

Список используемой литературы

1. Основные модели и методологии разработки программного обеспечения. [Электронный ресурс]. - URL: <https://baikov.dev/software-development-methodologies/>
2. Основы анализа данных на Python с использованием Pandas и Sklearn. [Электронный ресурс]. - URL: <https://habr.com/ru/articles/111674/>
3. Pandas Documentaion. [Электронный ресурс]. - URL: <https://pandas.pydata.org/docs/>
4. Matplotlib 3.8.2 Documentation. [Электронный ресурс]. - URL: <https://matplotlib.org/stable/index.html>
5. Python API reference for plotly. [Электронный ресурс]. - URL: <https://plotly.com/python-api-reference/>
6. Seaborn Library [Электронный ресурс]. - URL: <https://seaborn.pydata.org/>
7. Scikit-learn. Machine learning in Python. [Электронный ресурс]. - URL: <https://scikit-learn.org/stable/index.html>
8. Виды организационной структуры предприятия [электронный ресурс] URL - <https://blog.iteam.ru/vidy-organizatsionnoj-struktury-predpriyatiya/>
9. Организационная структура предприятия [Электронный ресурс] URL - <https://lektsii.org/12-6075.html>
10. Актуальность использования информационных систем управления для современного предприятия [Электронный ресурс] URL - <https://cyberleninka.ru/article/n/aktualnost-ispolzovaniya-informatsionnyh-sistem-upravleniya-dlya-sovremennogo-predpriyatiya>
11. Wikipedia – Свободная библиотека / Статьи: IDEF0. EPC. BPMN. [электронный ресурс] URL - <https://ru.wikipedia.org/wiki/IDEF0>
12. What is Unified Modeling Language (UML) [Электронный

ресурс] URL - <https://www.visual-paradigm.com/guide/uml-unified-modeling-language/what-is-uml/>

13. Кутелев П.В. Технология реинжиниринга бизнеса [учебное пособие] П.В. Кутелев, И.В. Мишурова. — М.: MapT; Ростов-на-Дону: MapT, 2003. — 176 с.

14. Инжиниринг информационных и деловых процессов [сборник] Моск. гос. ун-та экономики, статистики и информатики. — М., 1997. — 138 с.

15. Davenport T.H. Business Innovation, Reengineering Work through Information Technology [пособие] T.H. Davenport. — Boston: Harvard Business School Press, 1993.

16. Вендров А.М. CASE-технологии. Современные методы и средства проектирования информационных систем [учебник] А.М. Вендров. — М: Финансы и статистика, 1998. — 176 с.

17. Марка Д.А., МакГоуэн К. Методология структурного анализа и проектирования [учебник] Д.А. Марка, К. МакГоуэн. — М., 1993. — 240 с.

18. ARENA Users Guide. — Sewickley: System Modeling Co., 1996.

19. Каменнова М.С. Моделирование бизнеса. Методология ARIS / М.С. Каменнова, А.И. Громов, М.М. Ферапонтов и др. — М.: Весть-МетаТехнология, 2001. — 333 с.

20. Саати Т. Аналитическое планирование. Организация систем: пер. с англ. / Т. Саати, К. Кернс. — М.: Радио и связь, 1991. — 224 с.

21. Models and Analysis in Distributed Systems / ed. by S. Haddad [et al.]. — London [et al.] : ISTE [et al.], 2011. — XV, 355, [1] p.

22. Самуйлов, С. В. Объектно-ориентированное моделирование на основе UML : учеб. пособие [Электронный ресурс] URL-www.iprbookshop.ru/47277.html

23. Модели для классификации: Случайный лес (Random Forest) [Электронный ресурс] URL - <https://python-school.ru/blog/osnovy-ml/randomforestclassifier/>

24. Регрессионный анализ [Электронный ресурс] URL - http://www.machinelearning.ru/wiki/index.php?title=Регрессионный_анализ

25. Как провести анализ продаж [Электронный ресурс] URL - <https://developers.sber.ru/help/business-development/how-to-analyze-sales>