МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное бюджетное образовательное учреждение высшего образования

«ПЕНЗЕНСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»

Факультет вычислительной техники

Кафедра «Вычислительная техника»

Направление подготовки: 09.04.01 «Информатика и вычислительная техника»

Магистерская программа: «Прикладной искусственный интеллект»

МАГИСТЕРСКАЯ ДИССЕРТАЦИЯ

на тему

ПРОГРАММА ДЛЯ ОБРАБОТКИ, ХРАНЕНИЯ И ВИЗУАЛИЗАЦИИ ИНФОРМАЦИИ

Студент _____ Горбунова Вероника Владимировна
(подпись, дата) (ФИО полностью)

Руководитель _____ Федюнин Р.Н.
(подпись, дата) 20.06.2024 (фамилия, инициалы)

Нормоконтролёр _____ Бычков А.С.
(подпись, дата) 20.06.2024 (фамилия, инициалы)

Рецензент нач. отдела _____
(должность, место работы) (подпись, дата)

(фамилия, инициалы)

Работа допущена к защите (протокол зас. кафедры от 27.05.2024 г. № 11)

Заведующий кафедрой _____ М.А. Митрохин
(подпись) (фамилия, инициалы)

Работа защищена с отметкой отлично (протокол заседания ГЭК от 21.06.2024 г. № 35)

Секретарь ГЭК _____ Е.И. Гурин
(подпись) (фамилия, инициалы)

Пенза 2024

Федеральное государственное бюджетное образовательное учреждение
высшего образования
«ПЕНЗЕНСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»

Факультет
вычислительной техники

Кафедра
«Вычислительная техника»

«Утверждаю»
Заведующий кафедрой
_____М.А. Митрохин
«__13__» _сентября___2022 г.

ЗАДАНИЕ
НА МАГИСТЕРСКУЮ ДИССЕРТАЦИЮ

1. Студент Горбунова Вероника Владимировна гр. 22ВВИм1 факультета вычислительной техники

2.Тема работы «Программа для обработки, хранения и визуализации информации»

Тема утверждена приказом ПГУ №765/о от "13" сентября 2022 г.

3. Руководитель магистерской диссертации к.т.н., доцент Федюнин Р.Н.

4. Задание на работу (назначение разработки, исходные данные и т.п.)

Разработать программу для анализа продаж товара на маркетплейсах. Приложение должно выполнять комплексный анализ данных, отображать полученные данные в виде таблиц и графиков, выдавать прогноз продаж на следующие 30 дней.

5. Перечень подлежащих разработке вопросов:

1. Теоретический анализ предметной области.
2. Обзор и сравнительный анализ необходимых средств и инструментов для разработки программы.
3. Обучение нейронной сети для прогноза продаж, тестирование. Разработка приложения для персонального компьютера.

6. Календарный график выполнения работы

№ п/п	Наименование этапов работы	Объем работы	Срок выполнения	Подпись руководителя
1.	Теоретический анализ предметной области	10%	21.09.2022 – 21.11.2022	
2.	Обзор и сравнительный анализ средств для разработки	10%	22.11.2022 – 22.01.2023	
3.	Разработка нейронной сети	40%	23.01.2023 – 23.10.2023	
4.	Разработка приложения	30%	24.10.2024 – 30.03.2024	
5.	Оформление выпускной работы	10%	31.03.2024 – 31.05.2024	

Дата выдачи задания "__13__" __сентября__ 2022 г.

Руководитель магистерской диссертации _____ Федюнин Р.Н.
(подпись, дата)

Задание к исполнению принял студент _____ Горбунова В.В.
(подпись, дата)

Работу к защите допустить
Декан факультета _____ Фионова Л.Р.
(подпись, дата)
19.06.2024

РЕФЕРАТ

Магистерская диссертация содержит 103 листа, 46 рисунков, 1 таблицу, 23 источника, 2 приложения

РАЗРАБОТКА ПРИЛОЖЕНИЯ, АНАЛИТИКА, ВИЗУАЛИЗАЦИЯ, ДАННЫЕ, ГРАФИКА, МАРКЕТПЛЕЙС, АНАЛИЗ, ОБРАБОТКА ИНФОРМАЦИИ, БАЗЫ ДАННЫХ, ПАРСИНГ ДАННЫХ С ВЕБ-САЙТОВ

Объектом исследования является извлечение данных и аналитика в области товарного бизнеса.

Предмет исследования – характеристики товара, размещенного на маркетплейсе «Wildberries».

Цель работы – повышение эффективности работы специалистов, ответственных за аналитику продаж товаров, поиск и анализ конкурентов на маркетплейсе «Wildberries». Проектирование и разработка приложения для извлечения данных с возможностью его дальнейшего включения в систему интеллектуального анализа данных. Данная цель обусловлена необходимостью иметь структурированные и подготовленные для анализа данные. Проблема сложности анализа информации человеком без подобных систем актуальна для продавцов маркетплейса «Wildberries».

В ходе работы были выработаны требования к разрабатываемому приложению, спроектирована архитектура и разработано приложение для торговой площадки «Wildberries», а также определено направление дальнейшей работы над проектом.

					ПГУ 1.090401.03.001 ПЗ			
Изм.	Лист	№ докум.	Подпись	Дата				
Разраб.		Горбунова В.В.		20.06.24	Приложение для обработки, хранения и визуализации информации	Лит.	Лист	Листов
Провер.		Федюнин Р.Н.		20.06.24				
Н. Контр		Бычков А.С.		20.06.24		Кафедра ВТ, гр. 22ВВИм1		
					Пояснительная записка			
Утверд.		Митрохин М.А.		20.06.24				

СОДЕРЖАНИЕ

ПЕРЕЧЕНЬ СОКРАЩЕНИЙ И ОБОЗНАЧЕНИЙ. 7
ВВЕДЕНИЕ. 8
1 Анализ предметной области. 13
1.1 Машинное обучение и нейронные сети. 13
1.2 Использование искусственного интеллекта в прогнозировании продаж в товарном бизнесе. 18
1.3 Анализ существующих аналогов разрабатываемого приложения. 20
1.3.1 Tableau. 21
1.3.2 Power BI. 22
1.3.3 SAP Analytics Cloud. 23
1.3.4 Google Data Studio. 25
1.3.5 Yandex Datalens. 26
1.3.6 Excel. 27
1.3.7 Выводы. 28
1.4 Анализ методов парсинга веб-сайтов. 29
1.5 Анализ функций активации. 34
1.5.1 Сигмоидная функция активации. 34
1.5.1 Функция активации ReLu. 36
1.5.2 Leaky ReLU. 38
1.6 Выводы. 39
2 Обоснование целесообразности использования заданных средств разработки. 40
2.1 Анализ ОС для функционирования приложения. 40
2.2 Анализ платформ для разработки приложения. 41
2.2.1 QT Creator. 43
2.2.2 Eclipse. 44
2.2.3 Microsoft Visual Studio. 45
2.2.4 IntelliJ IDEA. 46
2.2.5 NetBeans. 47

2.3 Выводы. 48
3 Техническое задание. 49
3.1 Назначение разработки. 49
3.2 Требования к программе. 49
3.3 Стадии и этапы разработки. 51
3.4 Выводы. 51
4 Описание программы. 52
4.1 Общие сведения. 52
4.2 Функциональное назначение. 52
4.3 Вызов и загрузка. 52
4.4 Входные данные. 52
4.5 Выходные данные. 52
4.6 Проектирование взаимодействия между клиентом и приложением 53
4.7 Разработка приложения. 53
4.8 Архитектура программы. 62
4.9 Структура хранения данных. 63
4.10 Описание пользовательской программы. 64
4.11 Описание тестовых данных. 64
4.12 Отладка и тестирование. 65
4.13 Руководство пользователя по использованию программы. 66
4.14 Выводы. 71
ЗАКЛЮЧЕНИЕ 72
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ. 73
ПРИЛОЖЕНИЕ А Листинг программы 76
ПРИЛОЖЕНИЕ В Презентация 97

ПЕРЕЧЕНЬ СОКРАЩЕНИЙ И ОБОЗНАЧЕНИЙ

ИИ – искусственный интеллект

ОС – операционная система

ПК – персональный компьютер

Яндекс Метрика – интернет-сервис компании «Яндекс», предназначенный для оценки посещаемости веб-сайтов и анализа поведения пользователей

HTML (HyperText Markup Language) – стандартизированный язык гипертекстовой разметки документов для просмотра веб-страниц в браузере

HTTP (HyperText Transfer Protocol) – протокол уровня приложений для распределённых, объединённых, гипермедийных информационных систем

SQL (Structured Query Language) – декларативный язык программирования, применяемый для создания, модификации и управления данными в реляционной базе данных, управляемой соответствующей системой управления базами данных

URL (Uniform Resource Locator) – адрес ресурса в сети Интернет

ВВЕДЕНИЕ

Визуализация и структурирование данных играют критическую роль в обработке информации. Представление данных в графическом виде значительно облегчает их восприятие, помогая выявить важные закономерности и тенденции.

Работа с графикой требует глубокого понимания математики и геометрии, что выделяет эту область от разработки обычных прикладных программ. Библиотеки и компоненты, специализированные на визуализации данных, становятся популярными из-за их способности упростить процесс создания сложных графиков.

Помимо упорядочивания данных, визуализация также помогает в обнаружении пропущенных значений и выявлении скрытых паттернов в данных. Этот подход становится важным средством для анализа информации и принятия обоснованных решений [1].

Таким образом, анализ данных через визуализацию и работу с графикой играет ключевую роль как в научной сфере, так и в бизнесе, способствуя более эффективному и качественному использованию информации для прогнозирования и принятия стратегических решений.

Сейчас на рынке есть множество программных продуктов, предназначенных для обработки данных, а также для хранения и визуализации информации [2].

Рассмотрим стандартные возможности упомянутых выше программ:

- наличие непосредственной связи с базами данных или возможности выполнения разовых процедур импорта данных из формата хранения популярных баз данных и табличных редакторов,
- наличие возможности редактирования и/или просмотра в табличном режиме данных, поступающих из подключенной базы данных, введенных в ручном режиме или импортированных извне,

– возможность выбора типа диаграммы или графика, используемого для отображения данных, настройки цветов линий и плоскостей на графике, ориентации пределов измерений шкал, подписей и т. д.,

– возможность манипуляции шкалами: нормирование отображаемых величин, установление масштаба отображения, изменение закона распределения делений на шкалах,

– возможность вывода на печать и экспорта в другие программы и сохранения представления в иных форматах.

Такие возможности характерны для различных программ, например, табличные редакторы(Microsoft Excel, Lotus Notes), среды математического моделирования(MathLab, MathCAD) и многие другие инструменты для обработки и анализа данных.

Разработчики программ часто прибегают к сохранению результатов обработки в форматах, совместимых с распространенными программами для просмотра данных. Это делается для того, чтобы пользователь мог удобно просматривать и использовать данные с помощью встроенных средств отображения популярных программных продуктов, что эффективно помогает решить проблему отображения данных [3].

Применение средств визуального моделирования и проектирования на начальной стадии формализации упрощает понимание проекта в целом, так как оно:

– стимулирует мыслительные усилия эксперта за счет придания абстрактному процессу наглядной формы,

– способствует формированию и развитию системного мышления,

– способствует проявлению скрытых логических конфликтов, а также логической неполноты модели,

– позволяет создать основу для развития модели, дальнейшей формализации отношений, перехода от качественных показателей к количественным.

Множество компаний анализируют данные для увеличения эффективности работы.

Анализ данных - это ключевой процесс изучения информации и выявления закономерностей. Он занимает центральное место в области анализа данных и включает в себя два основных аспекта: описательный анализ (анализ данных и описание того, что происходит) и прогностический анализ (предсказание того, что произойдет в будущем, на основе текущих и прошлых действий) [4].

Описательный анализ является наиболее общим подходом к анализу данных и пытается ответить на вопрос «Что произошло в прошлом?». Для проведения описательного анализа используются исторические данные, которые могут охватывать как недавние временные интервалы, так и более длительные периоды.

Здесь применяются различные методы описательной статистики, такие как агрегации (например, сумма, количество, минимум и максимум), меры центральной тенденции (среднее, медиана, мода), и меры разброса значений (например, размах, стандартное отклонение).

Прогностический анализ, напротив, пытается ответить на вопрос «Что может произойти в будущем?». Для составления прогнозов используются исторические данные.

В прогностическом анализе используются такие инструменты, как отслеживание трендов, линии регрессии, тестирование гипотез и прогнозирование. Прогнозы строятся на основе статистических моделей, однако всегда остаются события, которые невозможно предсказать, как, например, пандемия COVID-19. Поэтому, с увеличением времени от текущего момента, точность прогнозов снижается.

Методы сбора данных - это техники и процедуры, используемые для сбора информации в исследовательских целях. Они могут варьироваться от простых опросов до сложных экспериментов и могут применять как количественный, так и качественный подходы.

Некоторые распространенные методы сбора данных включают опросы, интервью, наблюдения, фокус-группы, эксперименты и анализ вторичных данных [5].

Один из методов сбора данных - парсинг веб-страниц, который предполагает использование автоматических ботов или скриптов для извлечения данных с веб-сайтов [6].

Искусственный интеллект (ИИ) — это способность компьютера имитировать человеческое мышление. Для его обучения необходимо использовать большие объемы данных. Электронные устройства (компьютеры, телефоны) могут проанализировать изображения, распознавать речь, имитировать общение человека, а так же делать прогнозы на основе исходных данных с помощью искусственного интеллекта.

При анализе данных, для внесения изменений необходим большой человеческий ресурс. ИИ помогает упросить этот процесс, так как может самостоятельно вносить изменения, в особенности, если реализовано машинное обучение.

Машинное обучение — это метод ИИ, использующий математические алгоритмы для создания прогнозных моделей. Алгоритм нужен для анализа полей данных и «обучения» на основе закономерностей, которые получилось обнаружить в данных, для создания моделей. Далее модели можно применять для принятия решений, а так же для создания прогнозов на основе новых данных.

Для обучения модель необходимо протестировать на подготовленных данных, далее по заданным критериям, соответствующим определенному бизнес-сценарию, оценивается результат и, при необходимости, вносятся изменения. Модель машинного обучения можно улучшить путем повторного обучения.

Алгоритм — это последовательность вычислений и правил, используемых для решения проблемы или анализа набора данных. Он подобен

блок-схеме с пошаговыми инструкциями для задаваемых вопросов, но написанными на математическом языке или языке программирования.

С помощью алгоритма можно описать к какой категории относится животное, например, домашнее или дикое, ведущее ночной или дневной образ жизни. Можно составить более сложный алгоритм, который сможет анализировать речь, обрабатывать текст, а так же переводить его на другие языки.

Алгоритмы способны классифицировать информацию на основе заданных критериев, самостоятельно формировать эти критерии, исследовать закономерности и тенденции, а также генерировать отчеты на основе своих выводов быстрее, чем человек.

Исходя из бизнес-целей, искусственный интеллект способен оперативно выявлять закономерности в данных и извлекать ценную информацию из них. Эта информация может быть использована для разработки маркетинговых стратегий, оценки эффективности рекламных кампаний, оптимизации издержек и других задач [7].

1 Анализ предметной области

Для эффективной торговли необходим конкурентный анализ. Конкурентный анализ — это процесс изучения конкурентов, выводы после которого помогут принять стратегические решения. Анализ помогает найти ключевые показатели конкурентов и сравнить их со своими.

Это может быть анализ торгового предложения, ценообразования, ассортимента, отношения к клиентам, прибыли, оборотов, программы лояльности, каналов сбыта, рекламы. В частности, контроль продаж конкурентов для детской игрушки на маркептлейсе «Wildberries».

Входными данными для данной программы является ссылка на товар на маркетплейсе «Wildberries».

Выходными данными являются текущие цена, рейтинг, процент выкупа, данные по продажам за последний месяц (дата, цена, количество продаж, сумма продаж).

Основные задачи, которые необходимо решить за счёт данного приложения:

- получение данных по продажам товара,
- визуализация данных в виде графиков и таблиц,
- получение прогноза продаж на следующие 30 дней.

Таким образом, разработка программы для обработки, хранения и визуализации информации актуальна и востребована.

1.1 Машинное обучение и нейронные сети

Единого и общепринятого определение термина «машинное обучение» на данный момент нет, однако в общем виде можно зафиксировать следующую трактовку. Машинное обучение (Machine Learning) – множество математических, статистических и вычислительных методов для разработки алгоритмов, способных решить задачу не прямым способом, а на основе поиска закономерностей в разнообразных входных данных [8].

Это означает, что в процессе работы алгоритма, реализованного на принципах машинного обучения, задача решается не по четкой формуле, а по

установленным зависимостям итоговых результатов от конкретного набора признаков и их значений во входных данных. Можно сказать, что машинное обучение реализует подход Case Based Reasoning – метод решения проблем рассуждением по аналогии, путем предположения на основе подобных случаев (прецедентов).

Среди множества методов для реализации машинного обучения можно выделить 3 категории:

1) обучение с учителем (Supervised learning). В ходе данной модели обучения машинная система исследует примеры категории «стимул-реакция». Между входами (стимулами) и выходами (реакциями) существует зависимость, неизвестная системе, и изначально доступно только конечное множество «прецедентов» – пар «стимул-реакция», которая называется обучающей выборкой. Задача машины при таком подходе заключается в построении отношений между стимулами и реакцией, то есть в построении алгоритма, который для произвольного набора входных значений может выдать достаточно точный ответ. Для решения задач прогнозирования, классификации, ранжирования и регрессии обычно применяются методы данной категории,

2) обучение без учителя (Unsupervised learning). Данная модель предполагает отсутствие вмешательства со стороны эксперта. Таким образом предполагается, что машинная система будет спонтанно обучаться выполнять поставленную задачу на основе описания множества объектов. При данном подходе к обучению от системы требуется обнаружить внутренние взаимосвязи и закономерности, которые существуют между объектами. С помощью методов этой категории решаются задачи кластеризации, поиска правил ассоциации, заполнения пропущенных значений, сокращения размерности и визуализации данных,

3) обучение с подкреплением (Reinforcement learning). В данной категории предполагается, что машинная система, называемая агентом, обучается, взаимодействуя с некоторой средой. Откликом среды на принятые

агентом решения являются сигналы подкрепления, которые воздействуют на агента по изначально определённым правилам, меняя его решение до тех пор, пока реакция среды не станет оптимальной. Данная система взаимодействия агента со средой может рассматриваться как частный случай обучения с учителем, где в качестве учителя выступает не определённый конечный набор примеров «стимул-реакция», а окружающая агента среда. Однако надо учитывать, что правила подкрепления могут базироваться на неявных реакциях среды, что уже приближает данный подход к обучению без учителя. Именно из-за этой особенности отношений агента и среды данные методы машинного обучения принято выделять в отдельную самостоятельную категорию.

После рассмотрения сущности и методов машинного обучения следует перейти к изучению общих характеристик искусственных нейронных сетей и выявлению связей сетей и машинного обучения [9].

Основная идея искусственных нейронных сетей состоит в имитации работы естественных нейронных сетей. Таким образом, суть искусственных нейронных сетей можно выделить, рассмотрев приближённую работу человеческого мозга, который является естественной нейронной сетью.

Работа естественной сети состоит из взаимодействия огромного количества биологических нейронов (рисунок 1) – клеток, основные составляющие которой: ядро, отростки, тела и компоненты для связи с другими нейронами. Через связь нейронов друг с другом передаются электрохимические импульсы, приводящие всю сеть в состояние спокойствия или возбуждения.

По упрощенной теории естественной нейросети проведём изучение структуры биологических нейронов и принципах их работы. Нейрон состоит из ядра и тела клетки. У тела нейрона есть множество ответвлений, называемых дендритами. Среди них выделяют особо длинные, которые называются аксонами. Аксоны используются для установления связей между

нейронами через специальные точки контакта, называемые синапсами, через которые проходит импульс.

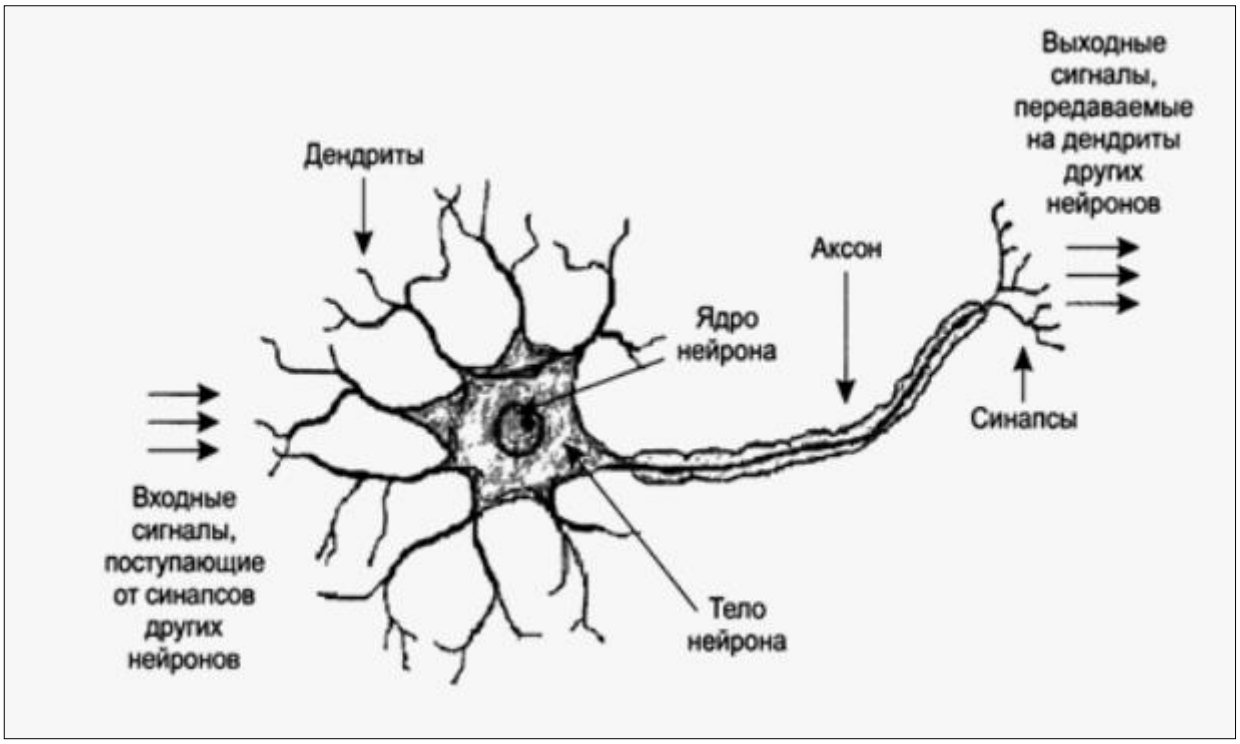


Рисунок 1 – Обобщённая структура биологического нейрона

На дендрит приходит сигнал, который по аксону передается другому нейрону. В точке синаптической передачи импульса, его частота и амплитуда может изменяться. Это общая схема работы человеческого мозга.

В 1957 году психологом Корнеллского университета Франком Розенблаттом была продемонстрирована первая обучаемая искусственная нейронная сеть, которая называлась «Перцептрон», и базировалась на принципах работы естественных нейронных сетей, описанных выше. Дендрит выступает роли информационного входа, аксон – выхода, тело нейрона послужило прототипом сумматора, а кодом, состоящим из 0 и 1 происходит регулирование частоты и импульса сигнала перед подачей на сумматор в разделе «Веса». Это основные составляющие искусственного нейрона (рисунок 2).

Учитывая всё вышенаписанное, можно заключить, что ИНС – математическая модель, а также её программное или аппаратное воплощение.

Она построена по принципу организации и функционирования биологических нейронных сетей, то есть сетей нервных клеток живого организма. ИНС состоит из комплекса взаимосвязанных простых процессоров, имитирующих нейроны. Такие процессоры обычно довольно просты (особенно в сравнении с процессорами, используемыми в персональных компьютерах). Каждый процессор подобной сети имеет дело только с сигналами, которые он периодически получает, и сигналами, которые он периодически посылает другим процессорам. И, тем не менее, будучи соединёнными в достаточно большую сеть с управляемым взаимодействием, такие, по отдельности простые, процессоры вместе способны выполнять довольно сложные задачи.

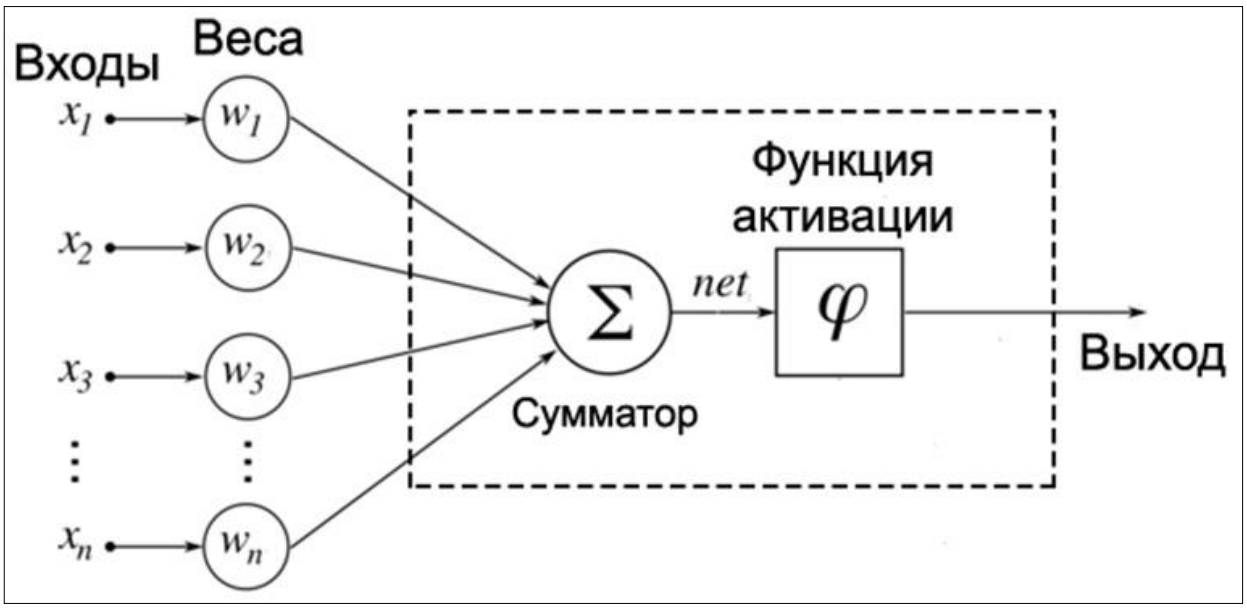


Рисунок 2 – Структура искусственного нейрона

Однако, для успешной работы нейронной сети её необходимо обучить, т.е. правильно настроить веса и смещения внутри сети. В этом процессе как раз и применяются методы машинного обучения практически всех категорий, что обеспечивает глубокую связь нейронных сетей и машинного обучения.

Таким образом, для задачи создания экспертной системы, которая требует решения в рамках магистерской ВКР, можно использовать искусственную нейросеть, обученную одним из методов машинного обучения.

1.2 Использование искусственного интеллекта в прогнозировании продаж в товарном бизнесе

Продажи — важный аспект для развития товарного бизнеса. Каждое принятое решение в организации, зависит от дохода от продаж, получаемого за определенный период времени. Прием сотрудников, маркетинг, постановка целей и так далее, зависят от получаемой прибыли.

Поэтому важно иметь оценку того, какой доход сможет приносить бизнес через месяц или год на основе текущей ситуации. Этот процесс называется прогнозированием продаж [10].

Данные, необходимые для прогнозирования будущих доходов от продаж и принятия важных решений для бизнеса:

- отраслевые тенденции,
- трубопровод продаж,
- отчеты по продажам,
- исторические данные покупателей,
- опросы клиентов и так далее.

Точные прогнозы позволят принимать обоснованные решения о последующих действиях, логически распределять ресурсы и удовлетворять потребности клиентов.

Технология искусственного интеллекта будет полезна компаниям различных отраслей, включая продажи. Отделы продаж могут стимулировать рост доходов, повысить производительность и получить конкурентное преимущество, используя искусственный интеллект.

Возможность точно предсказывать и прогнозировать продажи является одним из основных преимуществ искусственного интеллекта. Модели прогнозирования продаж на основе искусственного интеллекта используют информацию о клиентах и исторические данные для предсказания будущих тенденций продаж.

Ритейл – это механизм продажи услуг и товаров, сеть из покупателей и продавцов. Продавцы закупают товары оптом и продают по определённой

цене в различных розничных сетях и на торговых площадках. Покупатели берут представленный товар исходя из своих предпочтений.

Перед компанией всегда будет стоять вопрос прогнозирования: «Сколько каждого из товаров нужно закупить, чтобы создать ему оптимальные условия для хранения и при этом реализовать его до окончания срока годности?». В настоящее время недостаточно просто привозить продукцию в торговые точки и выставлять её на прилавки. Отчёты по продажам за месяц были хорошим инструментом для анализа, но при этом требовали большого количества времени и человеческих ресурсов [11].

Товарный бизнес может получать значительную экономию, с помощью решений по точному прогнозированию спроса с использованием искусственного интеллекта: не закупая продукты, не пользующиеся спросом в определенное время года или в отдельных регионах, минимизируя издержки на их хранение. Это является главным основанием востребованности таких решений у компаний.

Технологии, автоматизирующие процессы, востребованы уже давно. Использование искусственного интеллекта будет эффективно везде, где есть большое количество данных и необходимо выявлять корреляции между ними, а так же разрабатывать модели для прогноза продаж.

В сегменте онлайн-торговли имеются готовые оцифрованные данные, готовые к обработке. В офлайн-торговле данные сложнее получать, по этой причине этот сегмент отстает по применению искусственного интеллекта. Высокая стоимость разработки мешала их массовому внедрению, так как оплату искусственного интеллекта не каждая торговая сеть могла себе позволить, следовательно, такие проекты в товарный бизнес приходили медленно.

Избежать перегрузки складов, повысить точность планирования запасов и снизить возможные товарные и денежные потери позволит разработка автоматизированной системы прогнозирования спроса на товары.

После внедрения искусственного интеллекта, с одной стороны вырастет эффективность логистикой закупок, управления складами и так далее, с другой – снизится нагрузка на фонд заработной платы. Внедрение искусственного интеллекта позволит автоматизировать по возможности все процессы – от закупки и до продажи товара. Прогнозирование закупок – это только небольшая часть процессов, которые может выполнять искусственный интеллект. Максимальная автоматизация торговли позволит повысить качество обслуживания и лучше понимать потребителей.

Искусственный интеллект рассчитает более эффективное расположение товаров в магазине с учетом сезонного и дневного спросов, составит прогноз нагрузки на магазин, повысит точность рекомендательной системы, для конкретного клиента сформирует целевые предложения. Искусственный интеллект может значительно повысить эффективность работы всех сегментов ритейла [12].

Применение решений с искусственным интеллектом улучшает качество аналитики и очень востребовано. Однако, при их внедрении, нужно понимать, что модель обучается на исторических данных и не способна работать в ситуации, когда покупательская способность внезапно меняется, например, под влиянием скачков валюты или пандемии коронавируса.

1.3 Анализ существующих аналогов разрабатываемого приложения

Во многих сферах жизни приходится обрабатывать и анализировать информацию. В основном ее собирают в табличном виде — это удобно для хранения, но не для восприятия. Для упрощения процесса анализа данных активно применяются средства визуализации, инструменты для создания диаграмм и графиков. Они способствуют быстрому восприятию и анализу информации, позволяют сравнивать выполнение задач за разные периоды, видеть динамику изменения показателей и существенно сокращать время принятия решений.

Рассмотрим сервисы визуального представления данных и их особенности [13].

Платформы могут работать в облачных пространствах, на серверах компании, а так же в виде программы, установленной на компьютер. Они могут работать в открытом доступе, либо только на платной основе. Также можно разделить эти инструменты по уровню компетенций, необходимых специалистам: одни работают по принципу no-code, а другие требуют знания языков программирования.

1.3.1 Tableau

Популярная BI-платформа, обеспечивает полный цикл работы с данными, от процесса ETL до визуализации информации. Позволяет легко подключаться к различным источникам данных. Для построения моделей необходимы знания SQL, после загрузки данных удобно создавать разнообразные визуальные элементы, включая графики, диаграммы, карты и другие сложные элементы.

Серсис позволяет:

- собирать и объединять данные из разных источников,
- обрабатывать и трансформировать данные,
- строить графики,
- делать прогнозы,
- делать интерактивные дашборды и презентации,
- работать с проектами на локальном компьютере или в облаке.

Для пользователей с опытом работы в области аналитики она представляет собой мощный инструмент для работы с данными и принятия обоснованных решений. При этом, незнание русского языка в интерфейсе может стать небольшим препятствием, однако она легко устраняется опытом и знаниями специалистов в данной области.

Работа программы проиллюстрирована на рисунке 3.

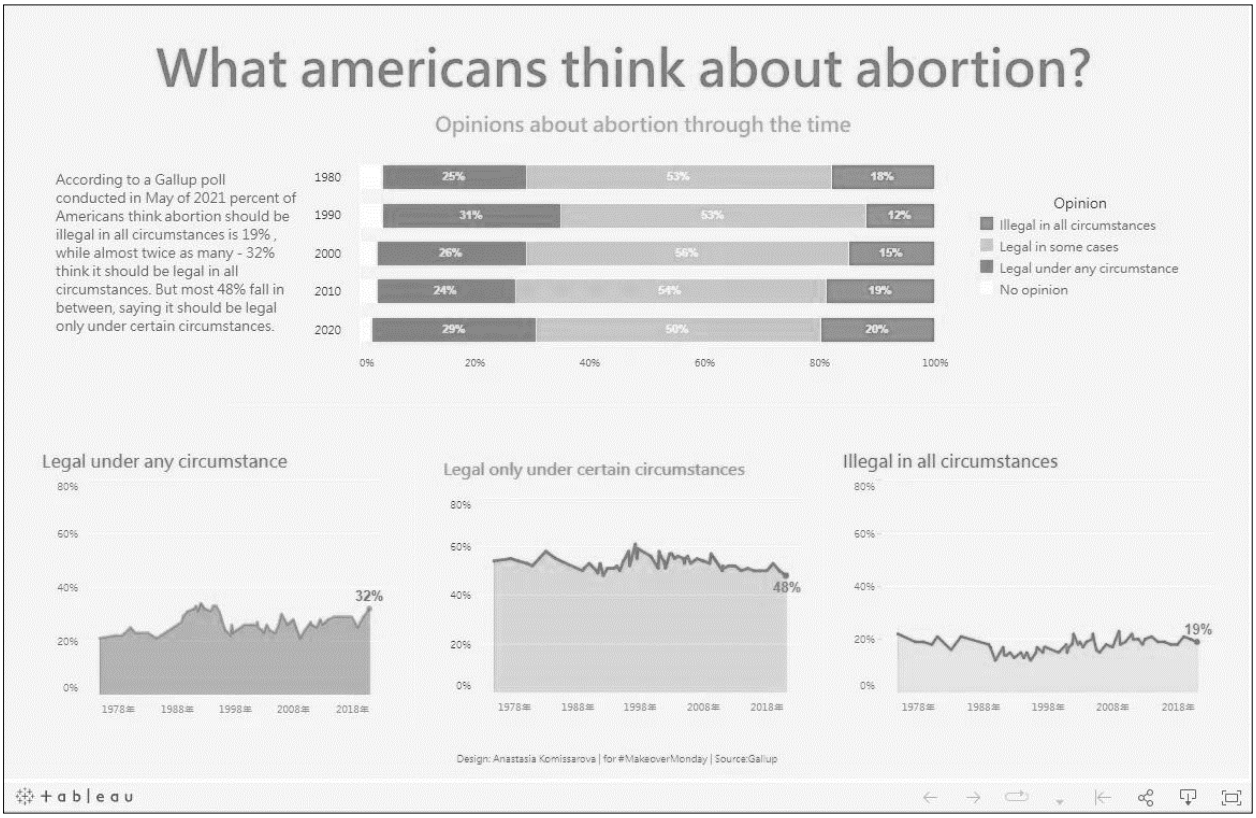


Рисунок 3 — Дашборд Tableau

1.3.2 Power BI

Power BI является одним из самых популярных инструментов. Он может работать как веб-версия, десктопное приложение, а так же мобильное приложение.

Платформа позволяет подключаться к разным источникам, визуально отображать информацию из баз данных, облачных хранилищ Google Docs и Яндекс-Диск, API, Excel и .csv-файлов. С помощью надстройки Power Query можно очистить исходные сведения, не меняя источник. Дашборды формируются на основе полученных данных, объединенных в модель. Готовые отчеты можно сохранить в виде файла, а так же опубликовать в сети интернет.

Возможности сервиса:

– сбор воедино данных по разным источникам, не связанным между собой,

– удобная визуализация полученной информации в графическом виде,

– приведение разрозненных сведений к единому стандарту, который позволяет обрабатывать информацию с самых «глубин» и выявлять причинно-следственные связи происходящих процессов,

– генерация собственных формул, показателей KPI и метрик для контроля состояния и оптимизации бизнеса,

– возможность представить любые отчеты через интернет с помощью онлайн-сервиса Power Bi Service или приложения Power Bi Mobile,

– возможность предоставления доступа индивидуально для каждого сотрудника,

– настройка оповещений для конкретных сотрудников,

– автоматическая актуализация данных.

Работа программы проиллюстрирована на рисунке 4.

Рисунок 4 — Дашборд Power BI

1.3.3 SAP Analytics Cloud

SAP Analytics Cloud представляет собой инновационный облачный инструмент, объединяющий функции планирования, аналитики и прогнозирования. Хорошо вписывается в инфраструктуру крупного бизнеса.

Пользовательский интерфейс SAP Analytics Cloud обеспечивает интуитивно понятные инструменты визуализации и взаимодействия с данными, что позволяет руководителям и аналитикам принимать обоснованные и быстрые решения [14].

Отличительная черта SAP Analytics Cloud — возможность использования искусственного интеллекта в прогнозной аналитике при обработке информации на доступном для пользователей уровне, без необходимости прописывать алгоритмы машинного обучения.

Возможности сервиса:

– глубокое и простое исследование данных всей организации с помощью интуитивно понятных инструментов,

– совместное планирование, обеспечивающее возможность оптимизировать основное финансовое и оперативное планирование,

– использование искусственного интеллекта в прогнозной аналитике упрощает доступ к важной информации.

Работа программы проиллюстрирована на рисунке 5.

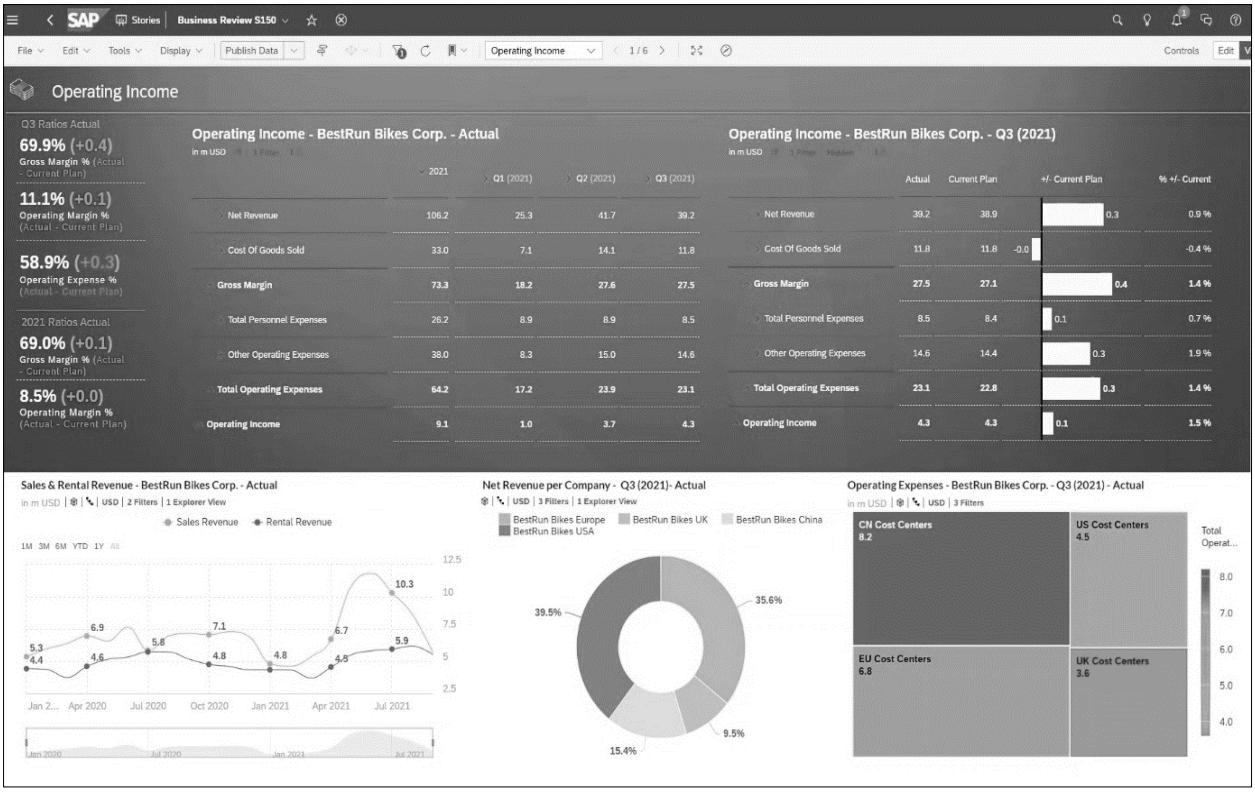


Рисунок 5 — Дашборд SAP Analytics Cloud

1.3.4 Google Data Studio

Google Data Studio (GDS) представляет собой бесплатный сервис, который обеспечивает возможность создания интерактивных отчетов и дашбордов с использованием разнообразных графиков, диаграмм и таблиц.

GDS позволяет легко объединять информацию из Google Analytics, CRM, баз данных, социальных сетей, Google Таблиц и других источников, представляя ее в виде интерактивных панелей. Отчеты автоматически обновляются благодаря интеграции.

Инструмент множество шаблонов для создания визуальных элементов: линейные, гистограммы, круговые диаграммы, географические карты и так далее. При помощи срезов и фильтров придается интерактивность.

Возможности сервиса:

- сбор данных из разных источников,
- глубокий анализ данных,
- создание отчётов для отдельных метрик или показателей,
- использование шаблона отчёта или создание собственного,
- контроль настройки отчётов,
- динамическое управление фильтрами и сегментами,
- создание нестандартных показателей на основе формулы,
- работа с несколькими аккаунтами,
- выбор представления данных,
- мониторинг показателей,
- можно открыть онлайн-доступ к файлам,
- работает в браузере на всех устройствах,
- можно настроить отправку отчетов на почту,
- доступна история изменений,
- доступен бесплатно.

Работа программы проиллюстрирована на рисунке 6.

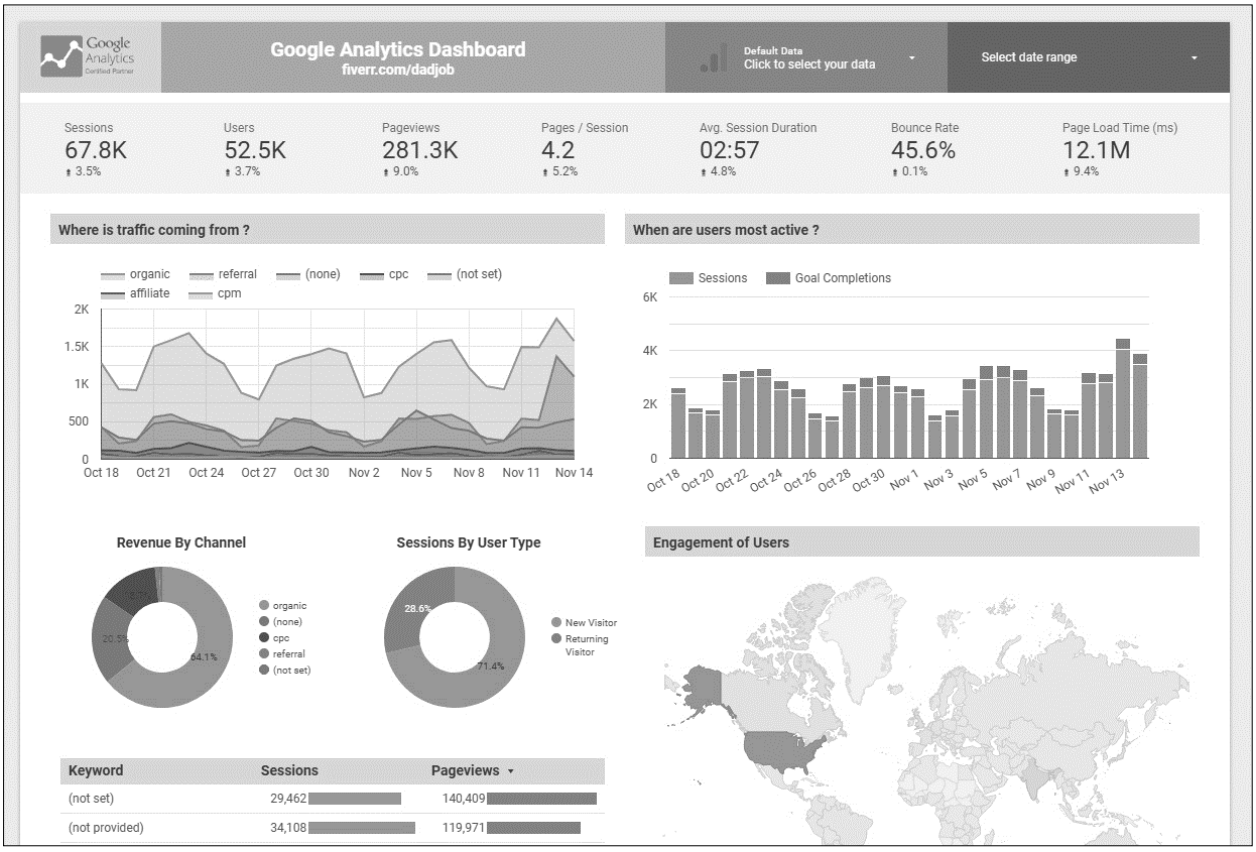


Рисунок 6 — Дашборд Google Data Studio

1.3.5 Yandex Datalens

Это платформа, созданная российскими разработчиками, может составить конкуренцию зарубежным аналогам.

Она представляет собой облачный сервис для подключения к источникам и подготовки интерактивных отчетов. Для ее использования не требуется технических навыков. Инструмент имеет множество шаблонов для создания стандартных диаграмм. Готовые дашборды легко скачивать и публиковать на различных ресурсах. Платформа предоставляет возможность совместной работы.

Возможности сервиса:

– широкий выбор инструментов для создания интерактивных дашбордов и отчетов: графиков, диаграмм и таблиц,

– возможность создания настраиваемых фильтров для быстрого поиска нужной информации,

– интеграция с различными источниками данных, такими как «Яндекс.Метрика», ClickHouse, PostgreSQL, MySQL, Microsoft SQL Server, «Google Таблицы» «Яндекс.Карт» и так далее,

– работа с данными в реальном времени.

Работа программы проиллюстрирована на рисунке 7.

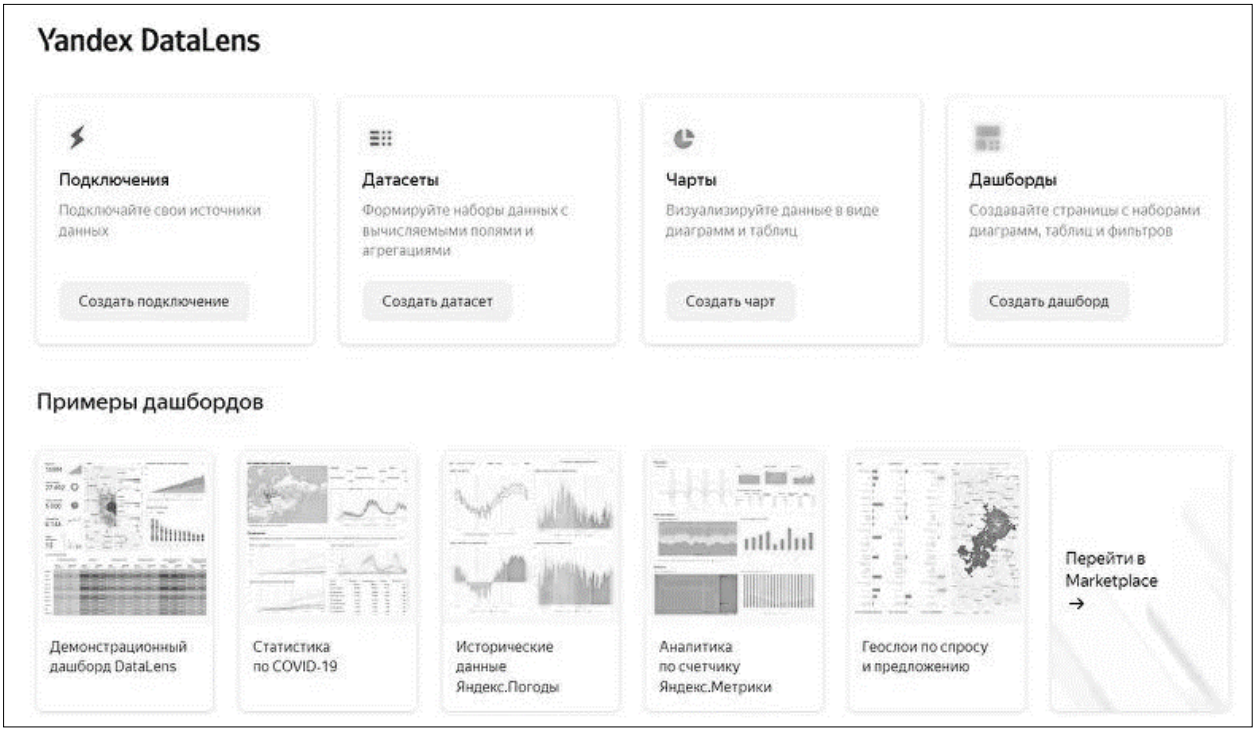


Рисунок 7 — Дашборд Yandex Datalens

1.3.6 Excel

Excel является одним из самых популярных офисных инструментов. Программа имеет очень широкие возможности, однако проигрывает облачным в удобстве работы. Она позволяет делать как статичные визуальные элементы, так и полноценные интерактивные панели мониторинга.

Excel предоставляет возможность собирать и обрабатывать данные, вследствие этого инструмент часто используется в аналитике данных [15].

Возможности сервиса:

– создание и редактирование электронных таблиц: пользователи могут создавать новые электронные таблицы, вводить и редактировать данные в ячейках, форматировать их, добавлять и удалять столбцы и строки,

– работа с формулами и функциями: Excel предлагает большой набор формул и функций для выполнения различных математических операций, статистических расчетов, обработки текста,

– графики и диаграммы: пользователи могут визуализировать данные с помощью диаграмм и графиков, которые позволяют наглядно представить информацию и облегчить ее анализ,

– объединение данных: с помощью Excel можно связывать данные из разных источников, например, из других файлов Excel или баз данных. Это помогает глубже анализировать и обрабатывать информацию,

– защита данных: можно защитить отдельные листы или рабочие книги от несанкционированного доступа, изменения или удаления.

Работа программы проиллюстрирована на рисунке 8.

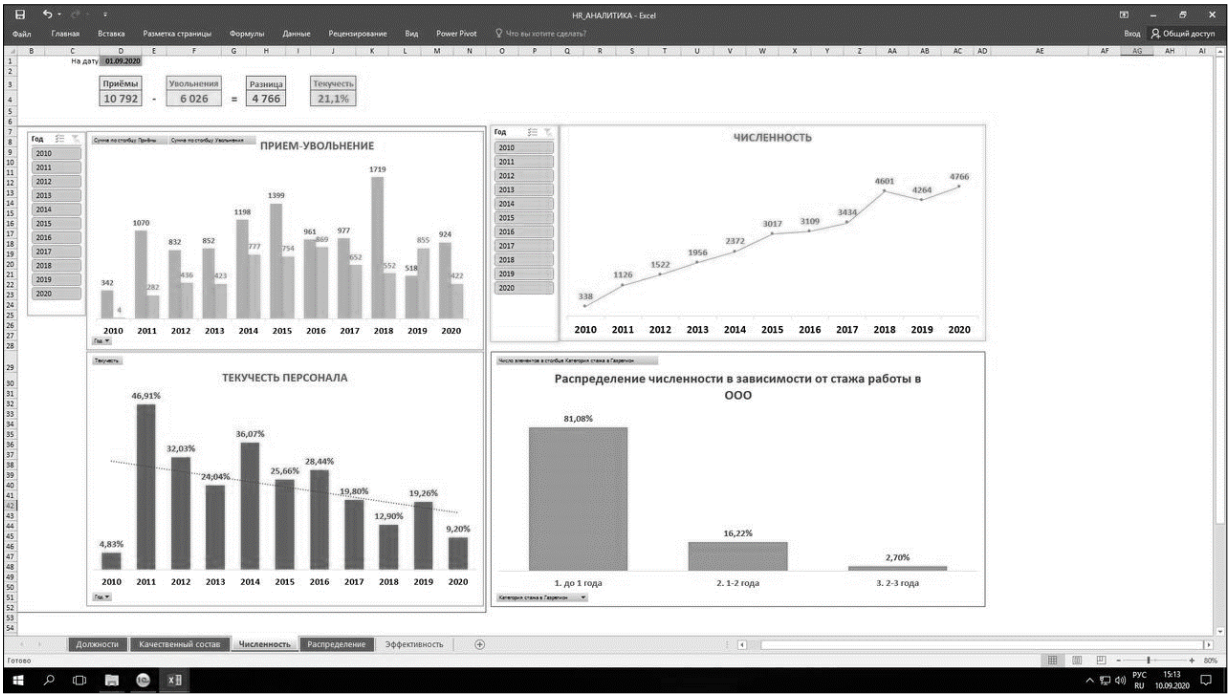


Рисунок 8 — Дашборд Excel

1.3.7 Выводы

Для эффективной подачи информации при помощи графиков и таблиц, необходимо обратить внимание на несколько важных аспектов: подбор правильных типов графиков и диаграмм, а также создание интуитивно понятного интерфейса играют ключевую роль в улучшении восприятия информации.

Поэтому, при разработке приложения, важно уделить внимание не только качеству визуализации данных, но и удобству интерфейса для пользователей, чтобы обеспечить оптимальный опыт работы и понимание сути данных без необходимости глубокого вникания в цифры.

В разрабатываемом приложении информация будет предоставляться в виде графиков и таблиц.

1.4 Анализ методов парсинга веб-сайтов

У современного человека часто возникает потребность в доступе к большому объему информации из интернет-ресурсов. Иногда, для поиска нужных данных требуется большое количество времени. Анализировать обширные объемы информации нелегко, и всегда существует риск упустить важные детали [16].

Парсинг веб-сайтов дает возможность:

– автоматизировать процессы сбора данных в необходимом масштабе,

– получить доступ к источникам данных, которые могут повысить эффективность бизнеса,

– принимать решения, опираясь на данные.

Эти факторы объясняют увеличивающийся интерес к парсингу веб-сайтов, что отражается в популярности запросов в Google Trends (рисунок 9).

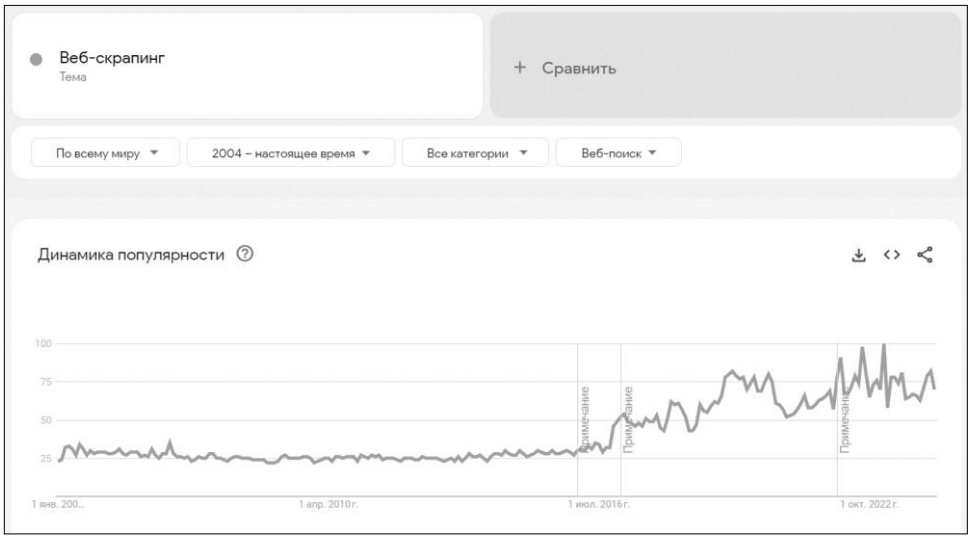


Рисунок 9 – Количество запросов по фразе «Веб-скрапинг» с 2004 г. по настоящее время

В данном проекте есть необходимость использовать автоматизированный поиск информации, который можно осуществить через API или с помощью парсинга веб-сайтов. Использование API возможно только тогда, когда веб-ресурс его предоставляет. Далее рассмотрим процесс парсинга веб-ресурсов.

Парсинг — это синтаксический анализ веб-документа и извлечение необходимой информации. Обычно таким образом происходит преобразование неструктурированных данных в Интернете в структурированные данные: базы данных или электронные таблицы. Под парсером понимается скрипт, используемый для автоматического сбора информации со страниц сайта и предоставления ее пользователю в структурированном виде.

Парсер позволяет выделять и импортировать повторяющиеся данные автоматически, что значительно экономит время и предотвращает возникающие при копировании информации вручную ошибки.

Парсинг сайта включает в себя доступ к Интернету напрямую через HTTP или через веб-браузер [17].

Работа парсера происходит следующим образом:

– идентификация целевых URL-адресов,

– если веб-сайт, сканируемый для сбора данных, использует инструменты противодействия парсингу, то парсеру, возможно, придется выбрать подходящий прокси-сервер, чтобы получить новый IP-адрес, через который парсер будет отправлять свой запрос,

– отправка запросов на эти URL-адреса для получения HTML-кода,

– использование указателей для обнаружения местонахождения данных в HTML-коде,

– аналитический разбор строки данных, которая содержит нужную информацию,

– преобразование собранных данных в желаемый формат,

– передача собранных данных в выбранное хранилище данных.

Рассмотрим сферы применения парсинга веб-сайтов:

– аналитика данных и наука о данных (data science). Сбор «тренировочных» данных для машинного обучения. Наполнение корпоративных баз данных,

– маркетинг и продажи. Сравнение цен. Извлечение описания товаров. Отслеживание в поисковых системах конкурентов и текущего состояния компаний в рамках усилий по поисковой оптимизации (SEO). Лидогенерация. Тестирование веб-сайтов. Отслеживание потребительских настроений,

– PR. Агрегация новостей о компании,

– торговля. Сбор финансовых данных,

– стратегическое планирование. Исследование рынка.

Инструменты для парсинга с веб-сайтов:

– фреймворки с открытым исходным кодом делают парсинг веб-сайтов дешевле и проще для личного использования. Наиболее часто используются следующие инструменты: Scrapy, Selenium, BeautifulSoup и Puppeteer. Пользователи могут собирать информацию, используя библиотеки, чтобы автоматизировать процесс;

– проприетарные решения. Они делятся на 2 типа:

1) решения, для взаимодействия с которыми нужно писать программный код. Поставщики таких решений предоставляют ключ API для доступа к данным;

2) решения без кода, для использования которых необязательно обладать навыками программирования и которые предоставляют инструменты, делающие сканирование веб-сайтов более доступным;

– облачные решения. Работа таких парсеров происходит в так называемом «облаке».

Рассмотрим следующие методы извлечения данных: регулярные выражения, модули DOM и XPath Simple, библиотеки QueryList и HTML DOM, Curl.

Регулярные выражения являются инструментом для извлечения информации с помощью шаблонов [18]. Переход на новую торговую площадку будет сопровождаться созданием большого количества новых сложных регулярных выражений.

Поэтому, чем больше торговых площадок бы охватывал модуль, тем большее количество регулярных выражений пришлось бы хранить. А так же, при больших объемах данных, работа регулярных выражений занимает большое количество времени.

Согласно DOM-модели, документ является иерархией («деревом»). Каждый HTML-тег представляет собой узел дерева с типом «элемент». Вложенные в него теги становятся дочерними узлами. Для представления текста создаются узлы с типом «текст» [19]. Пример такой модели приведен на рисунке 10.

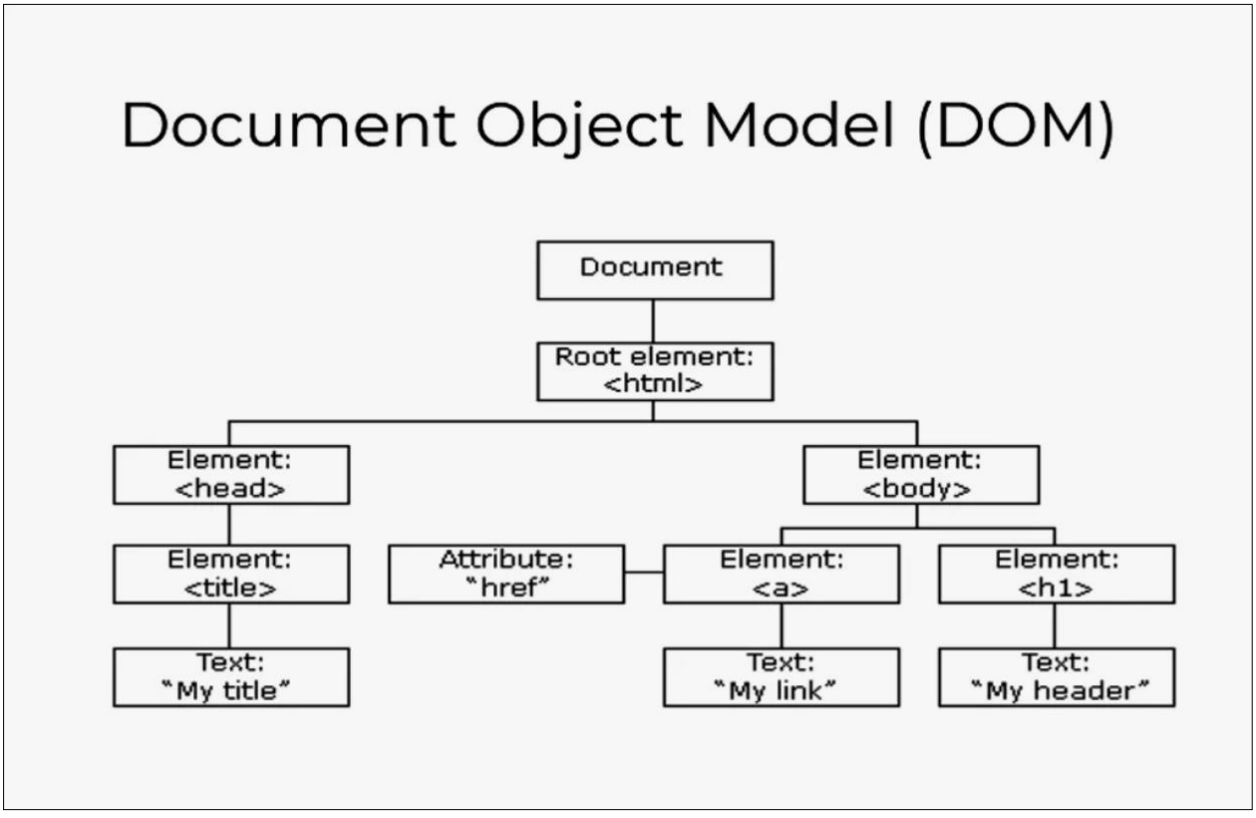


Рисунок 10 — Модель DOM

С помощью данного метода данные можно извлечь по идентификатору, имени, а так же другим атрибутам элемента дерева или с помощью уникального пути, спускаясь вниз по дереву. При изменении структуры веб-

ресурса путь к элементу может получиться очень сложным и потребовать изменений.

XPath представляет собой язык запросов к элементам XML- или XHTML-документа [20]. Для получения необходимых данных, требуется создать запрос, описывающий эти данные. Пример модели приведен на рисунке 11.

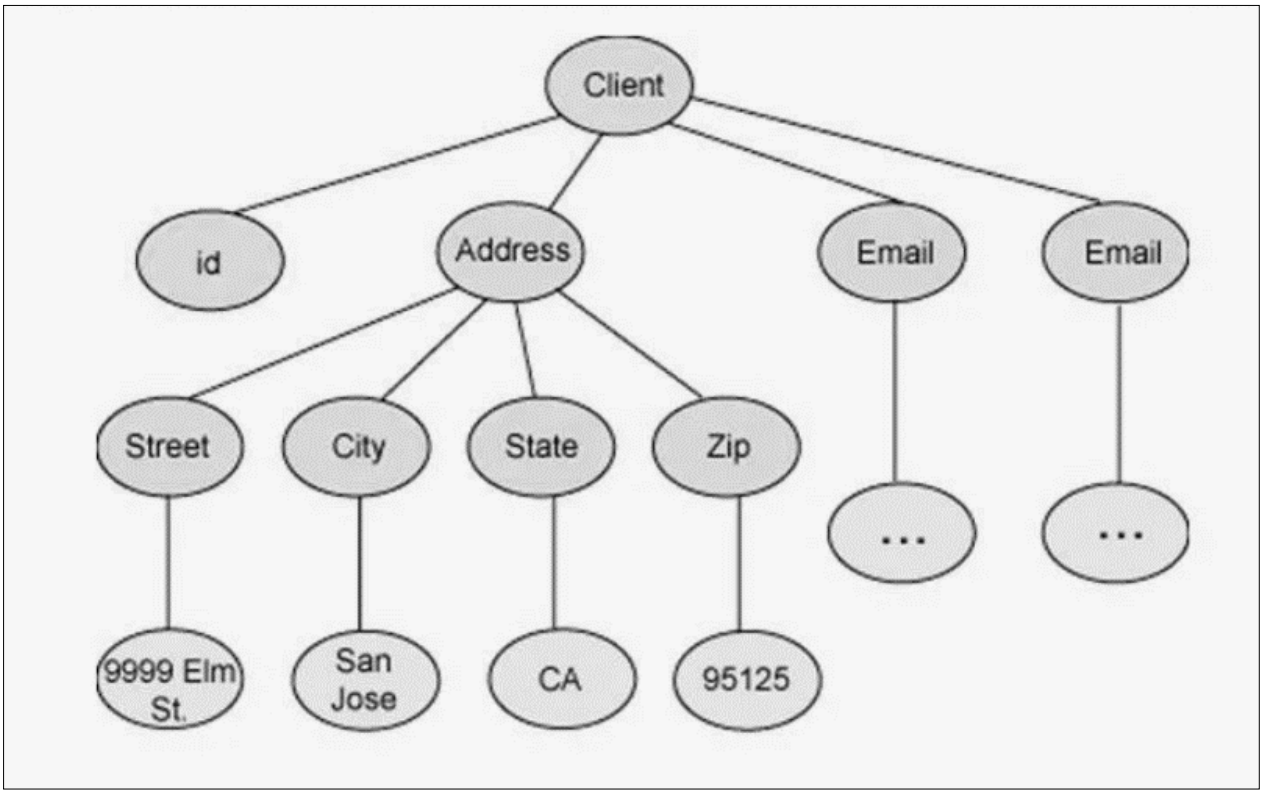


Рисунок 11 — Модель XPath

У библиотек QueryList и Simple HTML DOM не имеют минусов, указанных ранее. Они имеют удобные для поиска элементов HTML-кода возможности, например, обращение к HTML-элементам с помощью CSS селекторов. Это значительно упрощает парсинг новых торговых площадок. Производительность работы библиотеки QueryList выше, чем у Simple HTML DOM [21].

Проект Curl выделяется возможностью получения и отправки файлов на различные источники: http, https, ftp сервера и другие. Его основными преимуществами являются возможность установки логина и пароля для доступа к серверу, обработка редиректов и таймаутов, а также хорошая

документация. Кроме того, проект кроссплатформенный, с открытым исходным кодом.

Проект Curl доступен в виде консольной программы curl.exe, а также в виде библиотек libcurl-x64.dll и libcurl-x64.lib. Это позволяет интегрировать Curl непосредственно в свою программу без запуска дополнительного процесса, что дает возможность работать с буфером в памяти. Несмотря на наличие исходных кодов, создание необходимых библиотек из них может потребовать определенных усилий.

Таким образом, для использования в работе выбрана библиотека Curl.

1.5 Анализ функций активации

В машинном обучении и нейронных сетях слои слои активации играют ключевую роль в обработке данных. Они определяют, как нейрон должен реагировать на сумму входных сигналов. Такие функции могут быть простыми, например, линейные, или более сложными, как сигмоидная или ReLU. Основная задача функции активации заключается в добавлении нелинейности в нейронную сеть, что позволяет модели эффективнее обучаться и успешно решать более сложные задачи.

Рассмотрим несколько вариантов функций.

1.5.1 Сигмоидная функция активации

Сигмоидная функция активации, часто обозначаемая как σ(x), представляет собой нелинейную функцию, которая принимает любое вещественное число и преобразует его в диапазоне от 0 до 1. Математически она выражается формулой:

$$\sigma(x) = \frac{1}{1 + e^{-x}}$$

Где e – основание натурального логарифма.

Графически сигмоидная функция активации выглядит как S-образная кривая, которая монотонно возрастает и имеет асимптоты на 0 и 1. В

частности, если x > 0, то f(x) > 0.5, а если x < 0, то f(x) < 0.5. Значение 0.5 достигается при x = 0. График фцункции представлен на рисунке 12.

Сигмоидная функция активации используется для преобразования выходного значения нейрона в вероятность, т.е. вероятность того, что входное значение относится к классу 1, если мы работаем с задачей бинарной классификации. Если значение сигмоидной функции близко к 1, то вероятность того, что входное значение относится к классу 1, высока. Если значение близко к 0, то вероятность того, что входное значение относится к классу 1, низкая.

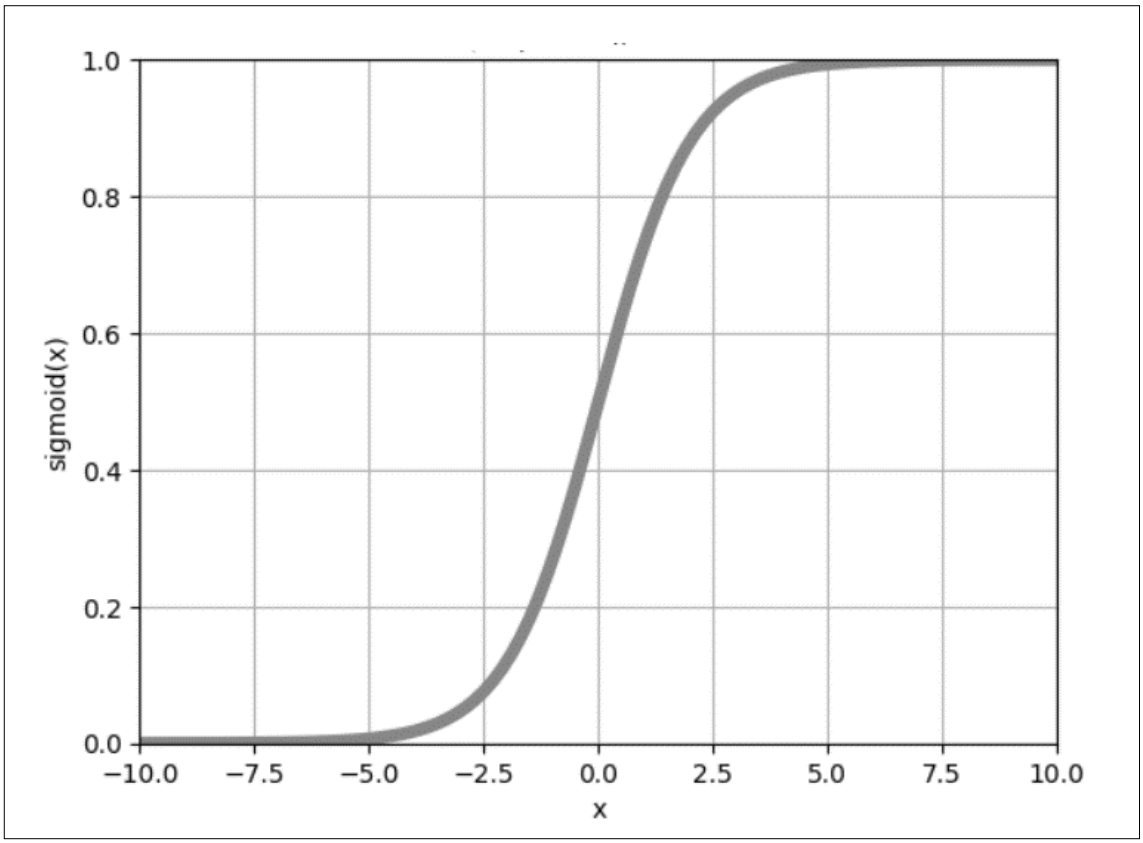


Рисунок 12 — График сигмоидной функции

Преимущества:

– гладкая градация: сигмоид обеспечивает плавный переход выходных значений, что полезно для предсказания вероятностей,

– дифференцируемость: эта функция дифференцируема в любой точке, что позволяет использовать её в методах градиентного спуска при обучении нейронных сетей,

– выходные значения между 0 и 1: это делает её удобной для задач, где требуется вероятностный вывод, как в бинарной классификации.

Недостатки:

– исчезающий градиент: в областях, где |x| очень велик, производная функции становится очень мала, что приводит к исчезающему градиенту и замедляет обучение,

– нецентрированный вывод: выходы сигмоиды не центрированы вокруг нуля, что может привести к смещению весов в нейронной сети,

– вычислительная затратность: из-за экспоненциальной операции в формуле сигмоидная функция требует больше вычислительных ресурсов.

1.5.1 Функция активации ReLu

ReLU (Rectified Linear Unit) - это нелинейная функция активации, которая широко используется в глубоком обучении. Она преобразует входное значение в значение от 0 до положительной бесконечности. Если входное значение меньше или равно нулю, то ReLU выдает ноль, в противном случае - входное значение.

Математически ReLU выражается следующим образом:

$$\mathrm{ReLU}(x) = \max(0, x),$$

где max - функция, возвращающая максимальное значение из двух.

Графически ReLU выглядит как линейная функция с нулевым отсечением на оси абсцисс в точке 0 (рисунок 13). Это значит, что функция имеет постоянный наклон во всех точках, кроме точки 0, где происходит отсечение.

Преимущества:

– устранение проблемы исчезающего градиента: в отличие от сигмоида и Tanh, градиент ReLU не сходится к нулю при больших

положительных значениях, что помогает ускорить обучение глубоких нейронных сетей,

– вычислительная эффективность: ReLU требует меньше вычислительных ресурсов, так как она включает в себя простые операции сравнения и присвоения, в отличие от экспоненциальных вычислений в сигмоиде и Tanh,

– способствует разреженности активаций: в ReLU, все отрицательные входы обнуляются, что приводит к разреженности активаций в нейронной сети. Это может улучшить эффективность и уменьшить переобучение,

– хорошие практические результаты: во многих практических приложениях, особенно в глубоких нейронных сетях, ReLU показала отличные результаты, опережая другие функции активации.

У ReLU есть и недостаток: проблема "мертвых нейронов", в таком случае нейроны, которые начинают выдавать отрицательные значения, перестают активироваться и вносить вклад в обучение сети.

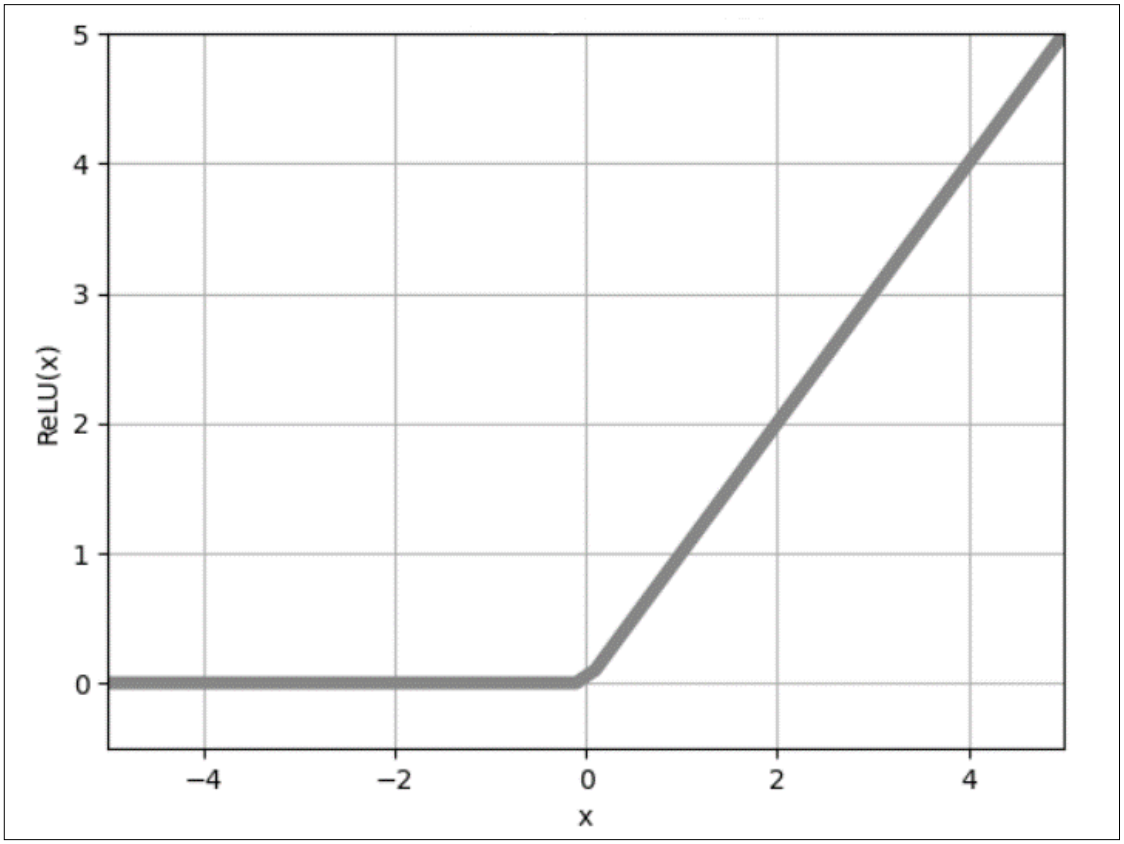


Рисунок 13 — График функции ReLU

1.5.2 Leaky ReLU

Leaky ReLU (Rectified Linear Unit) - это функция активации, которая используется в нейронных сетях для введения нелинейности в выходные данные каждого нейрона (рисунок 14).

Для решения проблемы "мертвых нейронов" был разработан Leaky ReLU. Он возвращает само значение при положительном входном значении, а при отрицательных значениях возвращает линейную функцию от входа, умноженную на небольшой коэффициент, называемый отрицательным уклоном (leak). Таким образом, у нейрона всегда есть возможность вносить вклад в выходную функцию, даже если входные данные отрицательны.

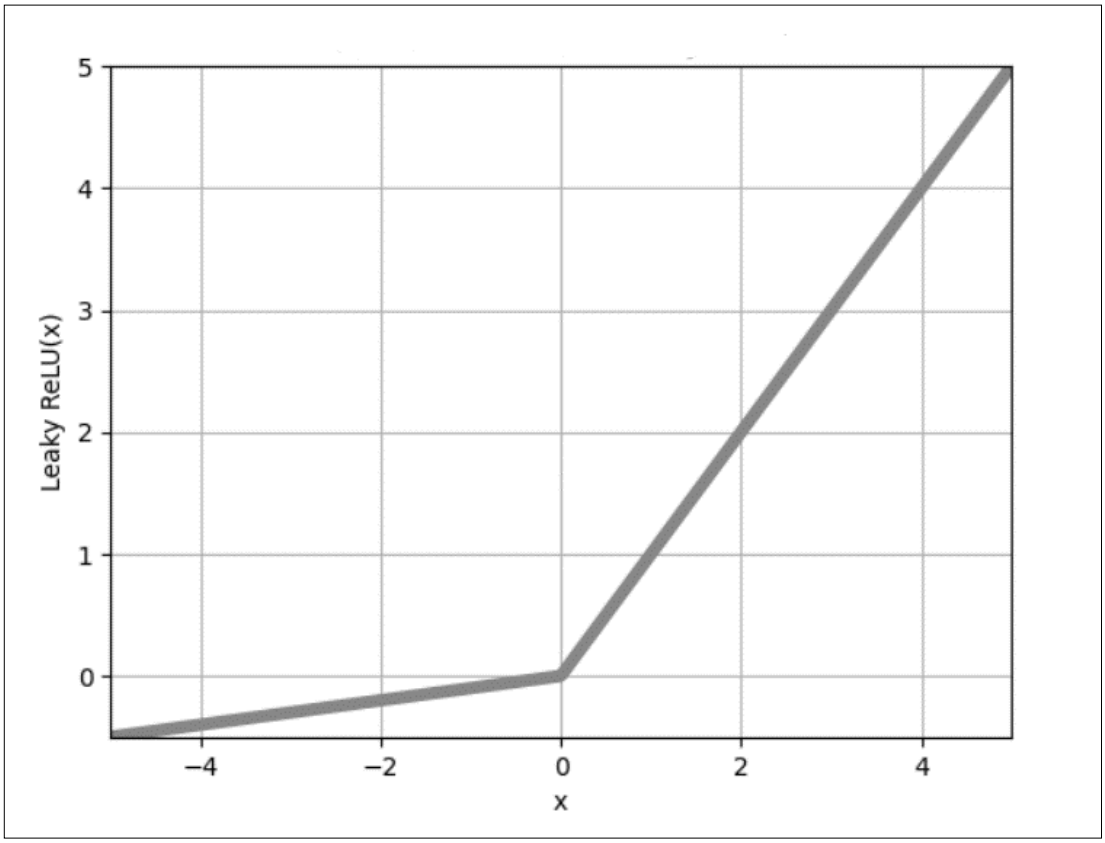


Рисунок 14 — График функции Leaky ReLU

Формула для Leaky ReLU выглядит следующим образом:

$$\text{Leaky ReLU}(x) = \max(a * x, x),$$

где a (alpha) - отрицательный уклон, который является маленьким положительным числом, например, 0,01.

Преимуществом Leaky ReLU является устойчивость к "умиранию" нейронов и лучшая сходимость в процессе обучения, что приводит к более быстрому и точному обучению нейронных сетей.

1.6 Выводы

Итак, в рамках этого раздела были достигнуты следующие результаты: произведен анализ предметной области, в рамках которой планируется вести разработку приложения, было рассмотрено машинное обучение и нейронные сети, проанализированы аналоги разрабатываемого приложения, проанализированы методы парсинга веб сайтов, рассмотрены функции активации нейронов.

2 Обоснование целесообразности использования заданных средств разработки

2.1 Анализ ОС для функционирования приложения

Для функционирования программы для обработки, хранения и визуализации информации необходимо выбрать операционную систему (ОС).

В соответствии с ГОСТ 15971-90, ОС – это совокупность системных программ, предназначенная для обеспечения определенного уровня эффективности системы обработки информации за счет автоматизированного управления ее работой и предоставляемого пользователю определенного набора услуг [22].

При выборе ОС необходимо обратить внимание на такие особенности, как совместимость с различными ПО, надежность, простота установки и обслуживание и стоимость приобретения.

Одними из распространенных ОС на сегодняшний день являются Windows, Mac OS и Linux.

Windows является самой распространенной ОС. Она установлена на большую часть устройств, будь то планшеты, ноутбуки или компьютеры. Windowsиспользуется как дома, так и на предприятии. Основными достоинствами ОС Windows является отличная совместимость с различными программными средствами и широкая распространенность.

Однако с распространением мобильной ОС Android, Linux стала теснить Windows, т. к. стала основой для Android. Linux принадлежит к семейству открытых и свободных систем семейства Unix. Основное достоинство данной ОС – это возможность доработки недочетов и неточностей благодаря открытому исходному коду.

Mac OS (Macintosh Operating System) – семейство ОС производства корпорации Apple. Эта система разработана для линейки персональных компьютеров (ПК) Macintosh. Одной из последних версий данной ОС является

Mac OS X, которая в недавнем времени была переименована в macOS. Преимуществами MacOS являются стабильность и производительность.

По данным Яндекс.Метрики 92,58% пользователей ПК используют ОС Windows. Соотношение ОС у пользователей ПК представлено на рисунке 15.

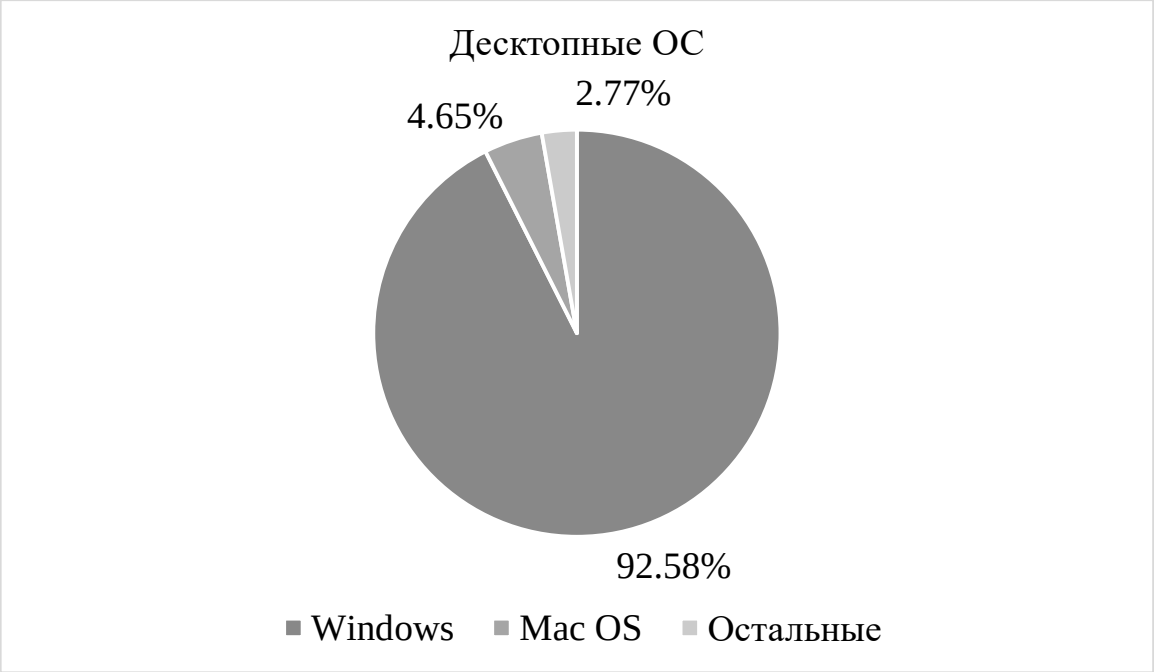


Рисунок 15 — Соотношение десктопных ОС

Целесообразность использования данной ОС подтверждается следующим рядом достоинств:

– возможность установки на многие компьютеры,

– высокий уровень надежности,

– возможность резервного копирования важных файлов, что обеспечивает сохранность данных,

– возможность одновременной работы нескольких приложений, не теряя при этом производительности,

– совместимость с наиболее распространенными программами,

– и многие другие.

Таким образом, для работы программы для сбора, анализа, хранения и визуализации данных была выбрана ОС Windows.

2.2 Анализ платформ для разработки приложения

Для разработки программного обеспечения используются различные инструменты, например, компоновщики, ассемблеры, компиляторы, а также отладчики, приложения для анализа производительности, дизайнеры

графических интерфейсов. Правильно подобранные инструменты значительно повышают производительность и результативность проекта.

Платформы для разработки приложений предоставляют разработчикам единую среду, в которой есть все инструменты, необходимые для создания программных приложений. Некоторые платформы обрабатывают множество различных типов приложений, в то время как другие ориентированы на конкретный вариант использования.

Компании, нуждающиеся в программном приложении, предоставляют своим разработчикам подобные платформы для написания, управления и развертывания приложений. Обычно этот процесс происходит полностью в отделе разработки или IT, но может происходить в рамках небольшой компании или на индивидуальной основе.

Платформы для разработки приложений обладают характеристиками, аналогичными интегрированным средам разработки (IDE), но они, как правило, более надежны в предоставлении разработчикам полного набора инструментов разработки.

Чтобы претендовать на включение в категорию платформ для разработки приложений, продукт должен:

- предоставлять возможность разработки программного обеспечения с помощью текстового редактора или графического пользовательского интерфейса (GUI),
- предоставлять интерфейс прикладного программирования (API) платформы и позволять выполнять компиляцию, отладку, контроль версий, предложения кода для конкретной платформы или развертывание кода,
- обладать интерфейсными инструментами разработки для создания пользовательского интерфейса / UX, внутренними инструментами разработки для служб передачи данных,
- предоставлять набор разработчика программного обеспечения с дополнительными инструментами для таких действий, как совместная работа, отладка или анализ кода.

Проведем анализ существующих платформ для разработки программного обеспечения, а так же рассмотрим их преимущества и недостатки.

2.2.1 QT Creator

Qt Creator – это кроссплатформенная интегрированная среда разработки C ++, JavaScript и QML, входящая в состав SDK для Qt Framework [23]. Внешний вид представлен на рисунке 16.

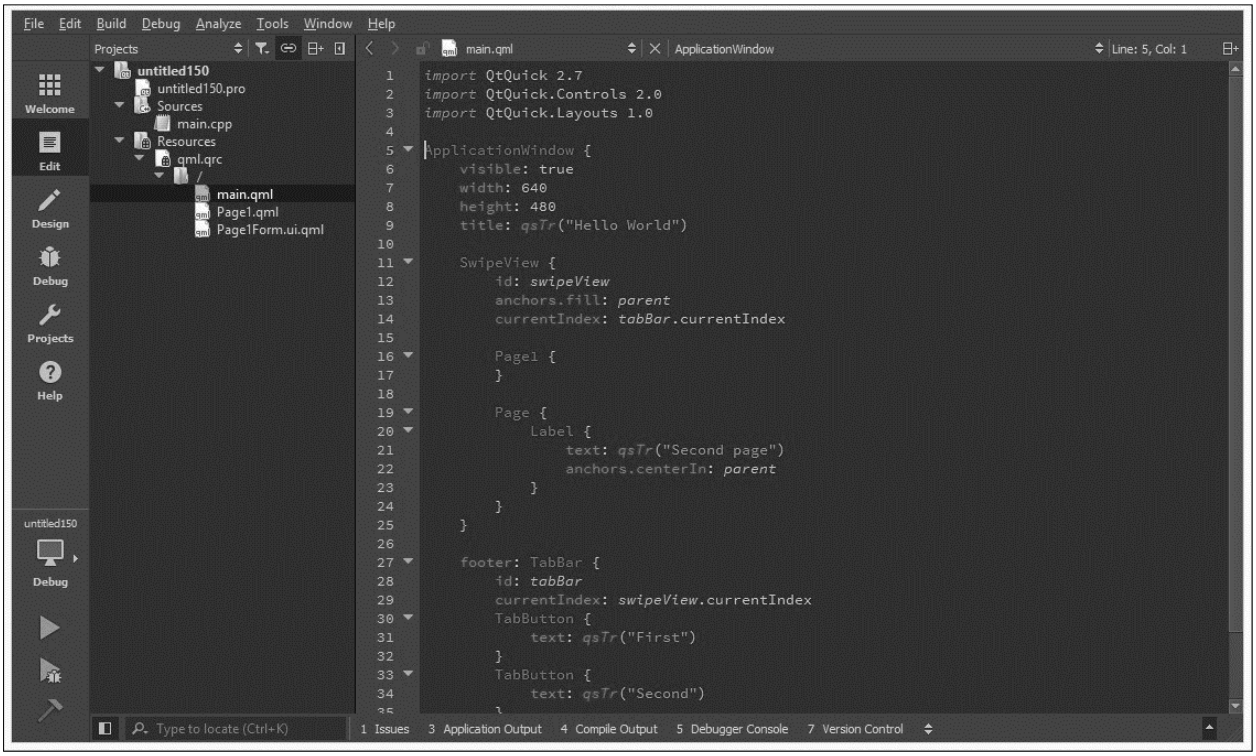


Рисунок 16 — Среда программирования Qt Creator

Он включает в себя визуальный отладчик, встроенный графический интерфейс и конструктор форм. Среди функций редактора - подсветка синтаксиса и автозаполнение.

Qt Creator использует компилятор C++ из коллекции компиляторов GNU для Linux и FreeBSD. Под Windows может работать с MinGW или MSVC с установкой по умолчанию, а также поддерживает отладчик консоли Microsoft при компиляции из исходного кода. Также есть поддержка Clang.

Qt Creator включает в себя менеджер проектов, который может использовать различные форматы проектов, такие как .pro, CMake, Autotools и

другие. Файл проекта содержит информацию о включенных файлах, пользовательских этапах сборки и настройках для запуска приложений.

Кроме того, Qt Creator включает редактор кода и интегрирует Qt Designer для проектирования и создания графических пользовательских интерфейсов (GUI) из виджетов Qt.

2.2.2 Eclipse

Eclipse представляет собой расширяемую платформу разработки со средами исполнения и прикладными платформами для создания, развертывания и управления программным обеспечением на всех этапах его жизненного цикла.

Внешний вид представлен на рисунке 17.

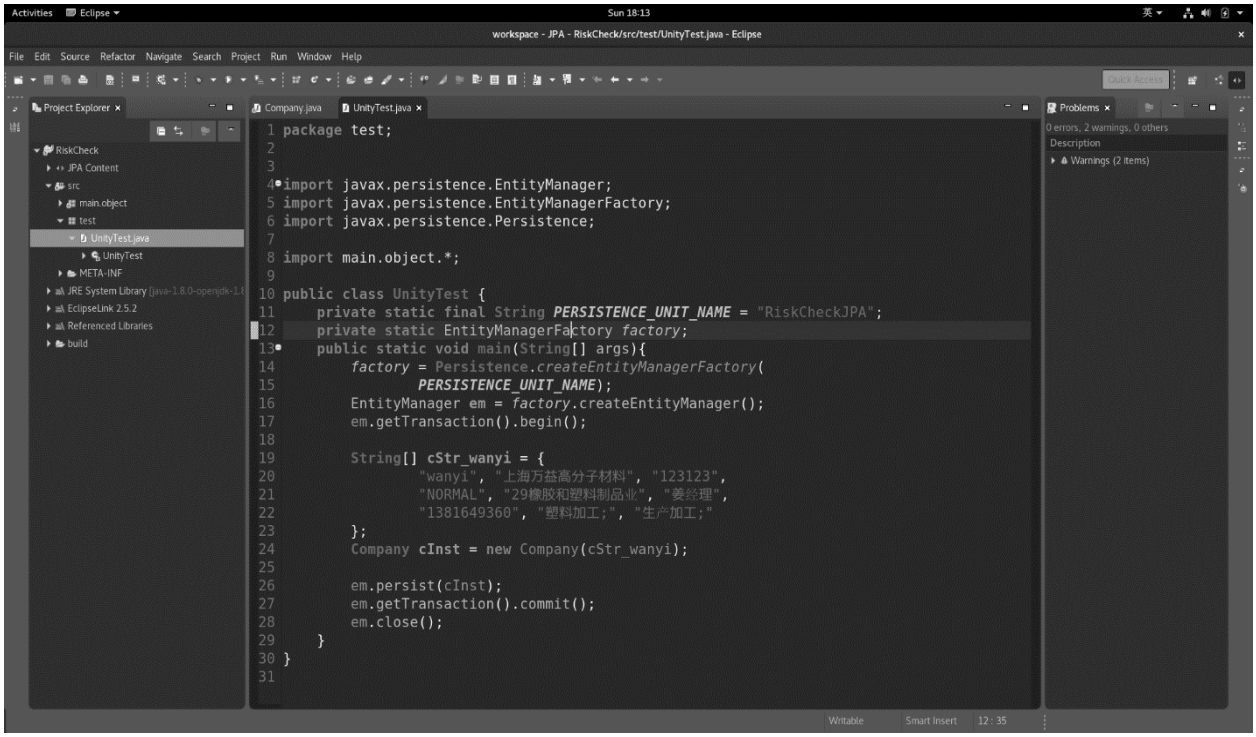


Рисунок 17 — Среда программирования Eclipse

Eclipse тесно связан с экосистемой Java, часто сопровождая новые версии Java обновлениями своей платформы. Переход к регулярным квартальным обновлениям от ежегодных выпусков отражает стремление к более частым итерациям разработки. Модульность является одной из уникальных особенностей Eclipse. Он представляет собой практически полностью модульную систему, состоящую из различных плагинов.

2.2.3 Microsoft Visual Studio

Microsoft Visual Studio - это интегрированная среда разработки (IDE). Она предназначена для создания компьютерных программ для операционной системы Microsoft Windows, а также веб-сайтов, веб-приложений и веб-сервисов. Visual Studio использует платформы разработки программного обеспечения от Microsoft, включая Windows API, Windows Forms, Windows Presentation Foundation, Windows Store и Microsoft Silverlight. Среда поддерживает как нативный код, так и управляемый. Внешний вид представлен на рисунке 18.

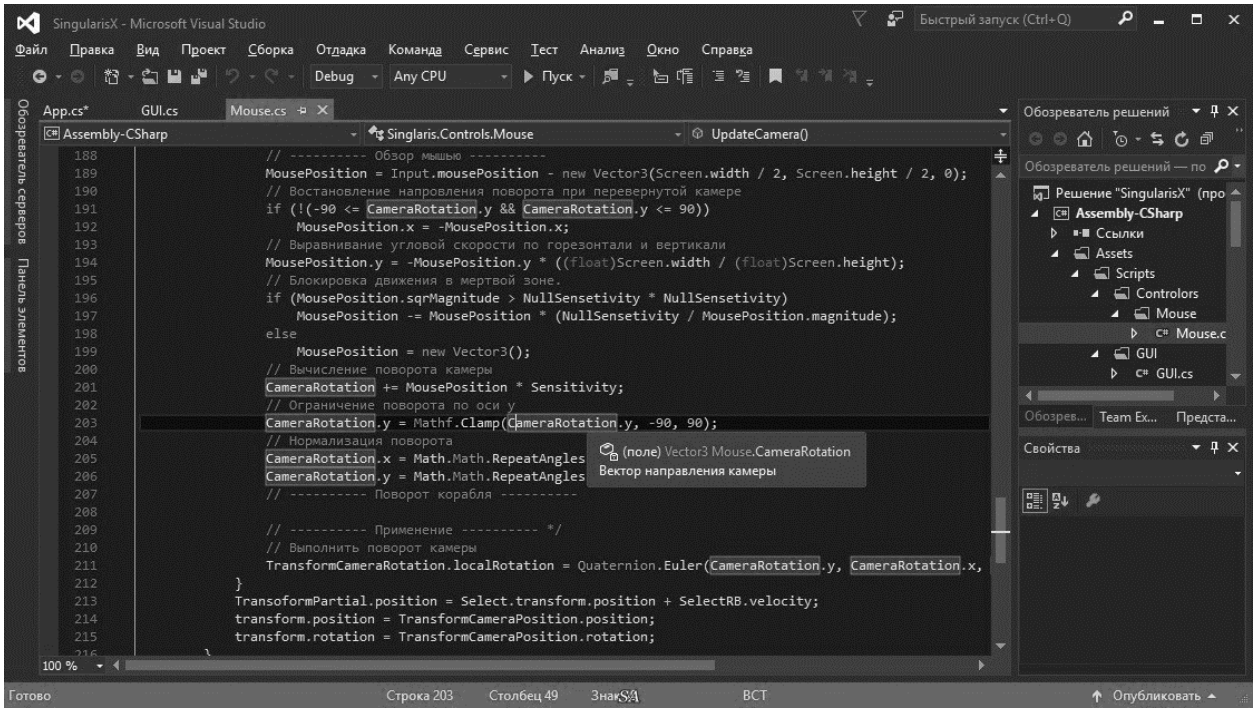


Рисунок 18 — Среда программирования Microsoft Visual Studio

Visual Studio включает в себя редактор кода, поддерживающий IntelliSense, а также рефакторинг кода. Интегрированный отладчик работает как отладчик уровня источника и отладчик уровня машины. Другие встроенные инструменты включают конструктор форм для создания приложений с графическим интерфейсом, веб-дизайнер, дизайнер классов и конструктор схем базы данных.

Visual Studio поддерживает разные языки программирования и позволяет редактору кода и отладчику поддерживать практически любой язык программирования при условии, что существует языковая служба.

Встроенные языки включают C, C ++ и C ++ / CLI (через Visual C ++), VB.NET (через Visual Basic .NET), C # (через Visual C #) и F # (начиная с Visual Studio 2010). Поддержка других языков, таких как M, Python и Ruby и др., доступна через языковые службы, устанавливаемые отдельно. Он также поддерживает XML / XSLT, HTML / XHTML, JavaScript и CSS.

2.2.4 IntelliJ IDEA

IntelliJ IDEA - это бесплатная коммерческая среда Java IDE, разработанная компанией JetBrains. Его дизайн ориентирован на производительность программиста. Она ориентирована на повышение производительности программиста, предлагая ряд функций, направленных на ускорение разработки, в то время как IntelliJ IDEA выполняет рутинные задачи кодирования. Внешний вид представлен на рисунке 19.

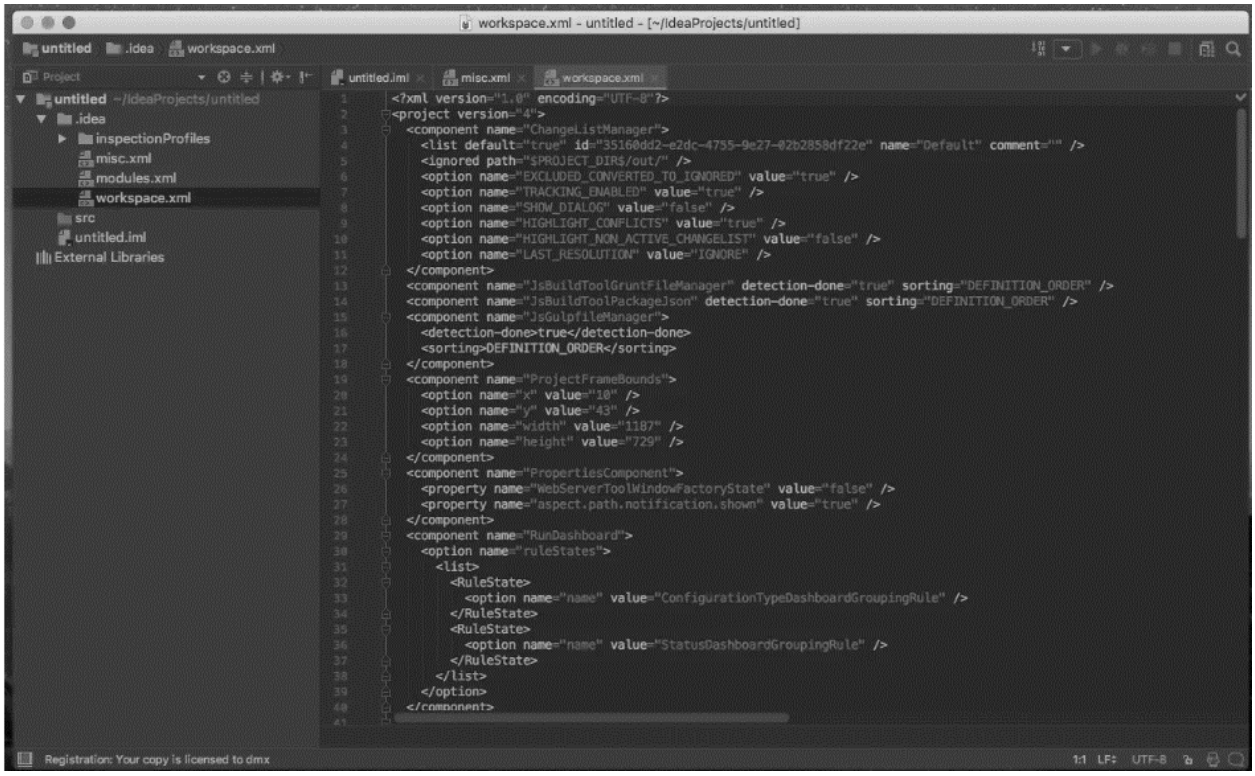


Рисунок 19 — Среда программирования IntelliJ IDEA

IntelliJ IDEA доступен платной (Ultimate) и бесплатной (Community) версиях.

IDEA поддерживает различные языки программирования с расширенными функциями проверки кода и рефакторинга, такие как Groovy, Javascript, Actionscript, Ruby / JRuby, PHP и другие.

Версия Community Edition является открытым исходным кодом, но пропускает большинство функций для серверной и веб-разработки.

2.2.5 NetBeans

NetBeans IDE - это бесплатная интегрированная среда разработки с открытым исходным кодом.Она обеспечивает все необходимые инструменты для создания профессиональных приложений: от настольных и корпоративных до веб-приложений и мобильных приложений на различных языках, таких как Java, C/C++, а также динамические языки программирования, включая PHP, JavaScript, Groovy и Ruby.

Среда IDE NetBeans проста в установке и использовании прямо из коробки и работает на многих платформах, включая Windows, Linux, Mac OS X и Solaris.

В разделе загрузки удобно представлены предварительно упакованные пакеты/установщики для каждого поддерживаемого языка программирования, содержащие только необходимые плагины.

Внешний вид представлен на рисунке 20.

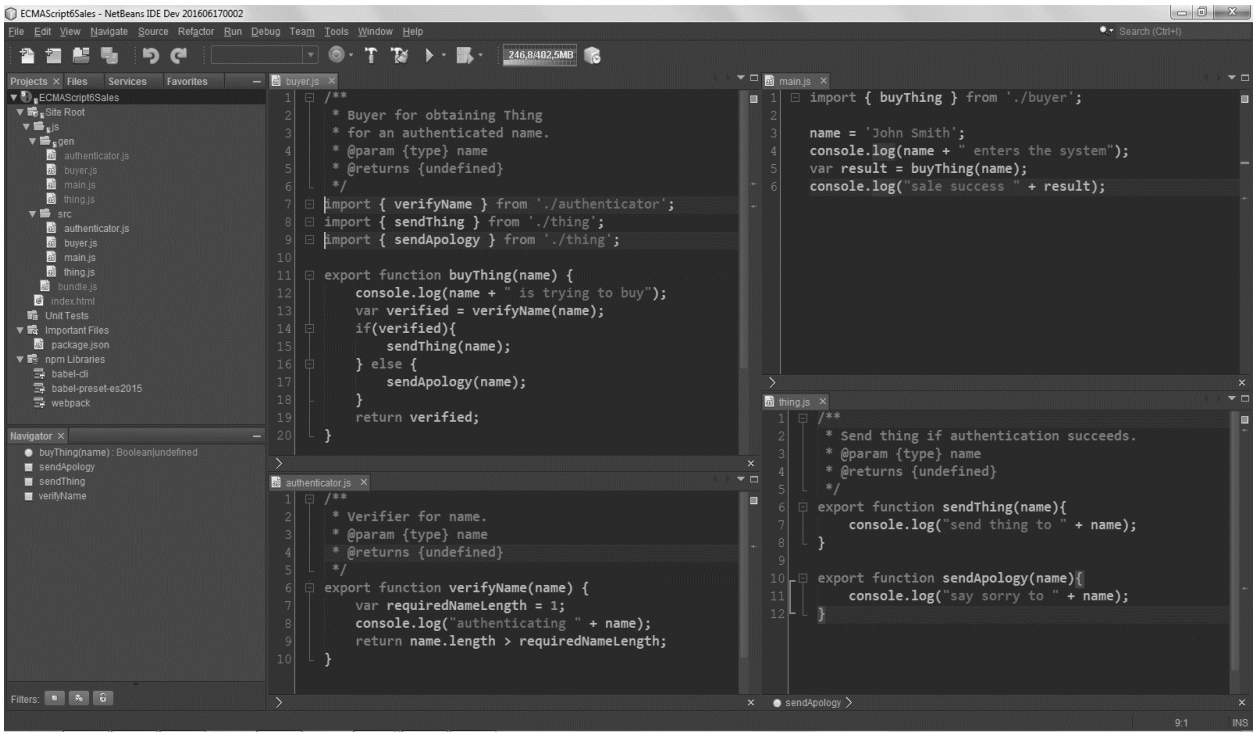


Рисунок 20 — Среда программирования NetBeans

В качестве среды разработки выбрана Qt Creator.

2.3 Выводы

Итак, в рамках этого раздела были достигнуты следующие результаты: произведён анализ ОС для функционирования приложения, а так же анализ платформ для разработки приложения.

3 Техническое задание

3.1 Назначение разработки

Программа должна быть предназначена для обработки, хранения и визуализации данных для товара на маркетплейсе «Wildberries».

3.2 Требования к программе

Целью дипломного проекта является проектирование и разработка приложения, главная задача которого – предоставлять данные в различном виде, а так же делать прогнозирование.

Проект нужно реализовать в виде десктопного приложения. Для этого необходимо реализовать проектирование и разработку пользовательского интерфейса, разработать структуру хранения данных в файлах. Данные необходимо получать из открытых источников с помощью LibCurl.

Разрабатываемое программное обеспечение должно представлять собой приложение, которое собирает данные с сайта маркетплейса «Wildberries» по заданным характеристикам, производит обработку и вывод данных в виде таблиц и графиков.

Программное обеспечение должно эксплуатироваться на персональном компьютере.

Программа должна обеспечивать возможность выполнения перечисленных ниже функций:

- сбор данных о товаре с сайта маркетплейса «Wildberries» (с помощью технологии парсинга данных):
  1) цена товара;
  2) рейтинг товара;
  3) артикул товара;
  4) процент выкупов;
- отображение данных по цене, количеству продаж, сумме продаж в виде таблиц и графиков;
- прогноз количества продаж на следующие 30 дней.

Главным источником информации о характеристиках товара является официальный сайт маркетплейса «Wildberries». Для автоматизации сбора информации будет написана программа для парсинга данных. Парсинг – это технология получения данных путем извлечения их со страниц сайтов.

Для реализации данной технологии будет использован язык программирования C++ и сопутствующие ему библиотеки.

После получения данных, происходит их консолидация.

Выходные данные приложения будут поступать в .csv файл. Они будут представлять из себя массивы данных с численной нформацией о характеристиках товара. При необходимости данные могут быть представлены в виде обобщенных таблиц и графиков.

При повторном запросе, данные о товаре будут обновлены. Данные распределятся по заранее прописанным параметрам.

В открытых источниках сети Интернет можно найти данные, необходимые для подробного анализа характеристик товаров. Официальный сайт «Wildberries» в открытом доступе предоставляет информацию о каждом товаре. Данные представлены в типизированном виде, что сильно облегчает работу с ними (рисунки 21, 22).

Рисунок 21 – Представление характеристик товара на официальном сайте Wildberries

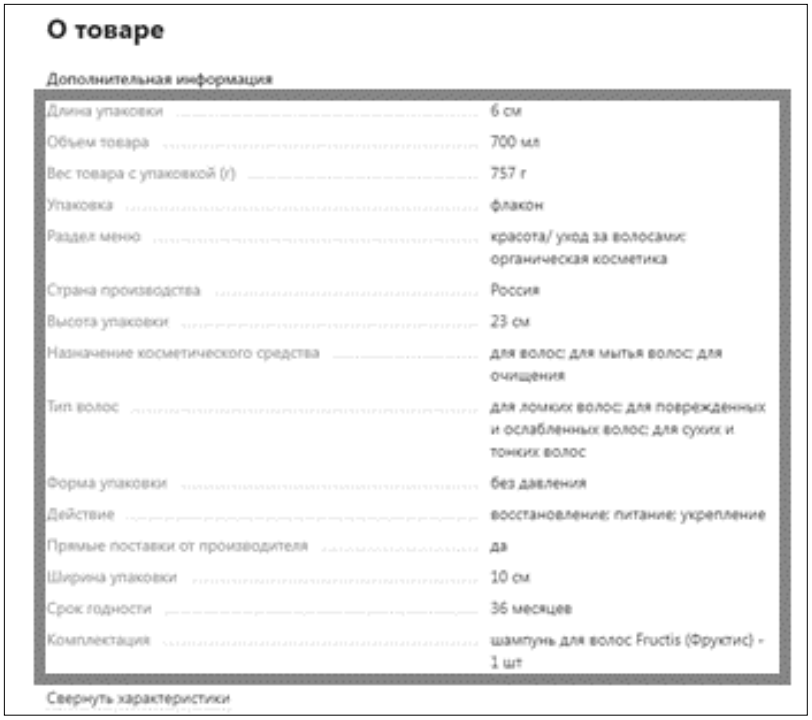


Рисунок 22 – Представление характеристик товара на официальном сайте Wildberries

3.3 Стадии и этапы разработки

Стадии и этапы разработки приложения представлены в таблице 3.1.

Таблица 3.1 – Стадии и этапы разработки приложения

Стадии разработки	Этапы разработки
Техническое задание	Обоснование необходимости разработки программы
	Научно-исследовательские работы
	Разработка и утверждение технического задания
Рабочий проект	Разработка программы
	Испытания программы
Внедрение проекта	Подготовка и передача программы

3.4 Выводы

В рамках этого раздела определены требования к программе, а так же стадии и этапы разработки.

4 Описание программы

4.1 Общие сведения

Приложение для обработки, хранения и визуализации информации написано на языке C++ в среде разработки QT Creator.

Разработанная программа имеет название «Аналитика товаров на маркетплейсе Wildberries» и располагается в файле Analytics_WB.exe. Листинг программы приведен в приложении А.

4.2 Функциональное назначение

Функциональное назначение приложения для сбора, хранения и визуализации информации следующее:

- сбор данных о товаре с сайта маркетплейса «Wildberries» (с помощью технологии парсинга данных),
- комплектация собранной информации,
- отображение данных в виде таблиц и диаграмм,
- получение прогноза продаж на следующие 30 дней.

4.3 Вызов и загрузка

Для корректной работы приложения для обработки, хранения и визуализации информации персональный компьютер должен быть продключен к сети Интернет.

Запуск программы осуществляется запуском исполняемого файла Analytics_WB.exe.

4.4 Входные данные

В качестве входных данных для приложения для обработки, хранения и визуализации информации используется ссылка на товар на маркетплейсе «Wildberries».

4.5 Выходные данные

Выходными данными для приложения для обработки, хранения и визуализации информации должны быть:

- цена товара,
- рейтинг товара,

- артикул товара,
- процент выкупа товара,
- количество продаж за последний месяц,
- сумма продаж за последний месяц,
- прогноз продаж на следующие 30 дней.

4.6 Проектирование взаимодействия между клиентом и приложением

Для более полного понимания взаимодействия пользователя с приложением необходимо создать UML-диаграмму последовательностей. UML-диаграмма последовательностей, представленная на рисунке 23, будет показывать, как пользователь взаимодействует с приложением. Такая диаграмма помогает наглядно отобразить динамику работы всей системы и взаимодействие её основных объектов.

Программа представлена в виде приложения, которое запускается на персональном компьютере с помощью ярлыка на рабочем столе или через менб «Пуск». Она позволяет пользователям выполнять различные действия над данными. Для получения информации необходимо вставить ссылку в требуемое окно. Далее на экране отобразятся данные по количеству продаж, цене и сумме продаж за последние 30 дней в виде таблиц и графиков. Так же имеется возможность изучить прогноз количества продаж на следующие 30 дней на основе текущих показателей. Пользователь имеет возможность сохранить полученные результаты в файл, а так же загрузить их позже для повторного анализа.

4.7 Разработка приложения

Для начала разработки необходимо установить среду программирования QT Creator. Для установки можно использовать разные способы: использование одного из менеджеров пакетов, установка вручную, установка с помощью инсталлятора.

В данном случае установка будет производиться с поиощью офлайн инсталлятора. Его можно скачать с официального сайта QT. Для установки

потребуется запустить установочный пакет (рисунок 24), создать или войти в учетную запись (рисунок 25), выборать каталог установки (рисунок 26), а так же необходимые компоненты (рисунок 27). На этом настройка завершена и можно начинать работу с проектом.

Рисунок 23 – UML-диаграмма последовательности

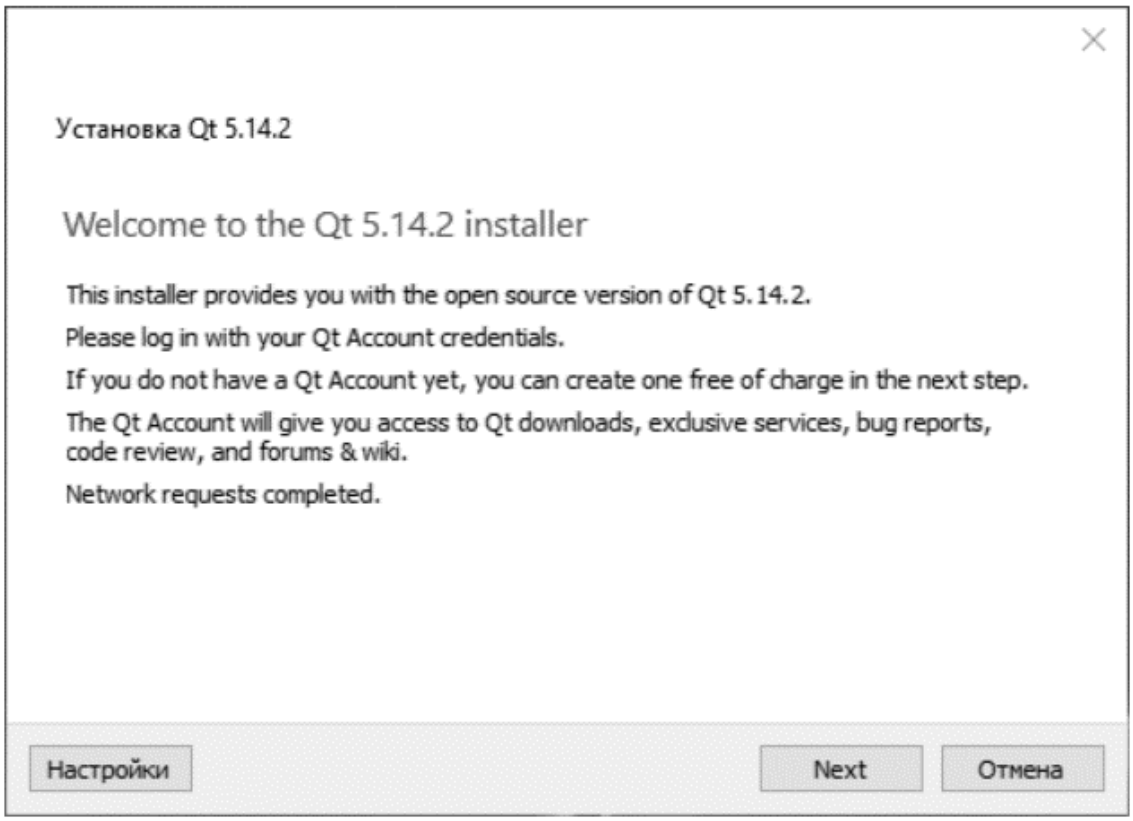


Рисунок 24 – Запуск установочного пакета

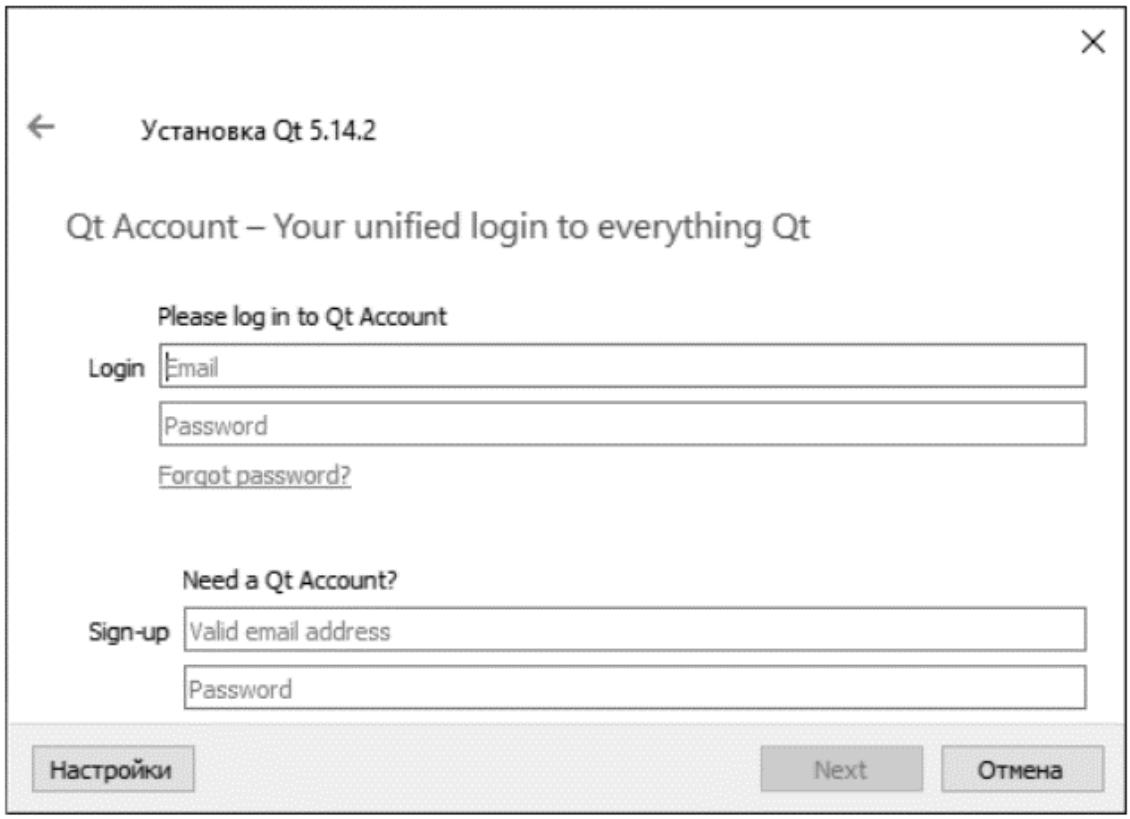


Рисунок 25 – Авторизация

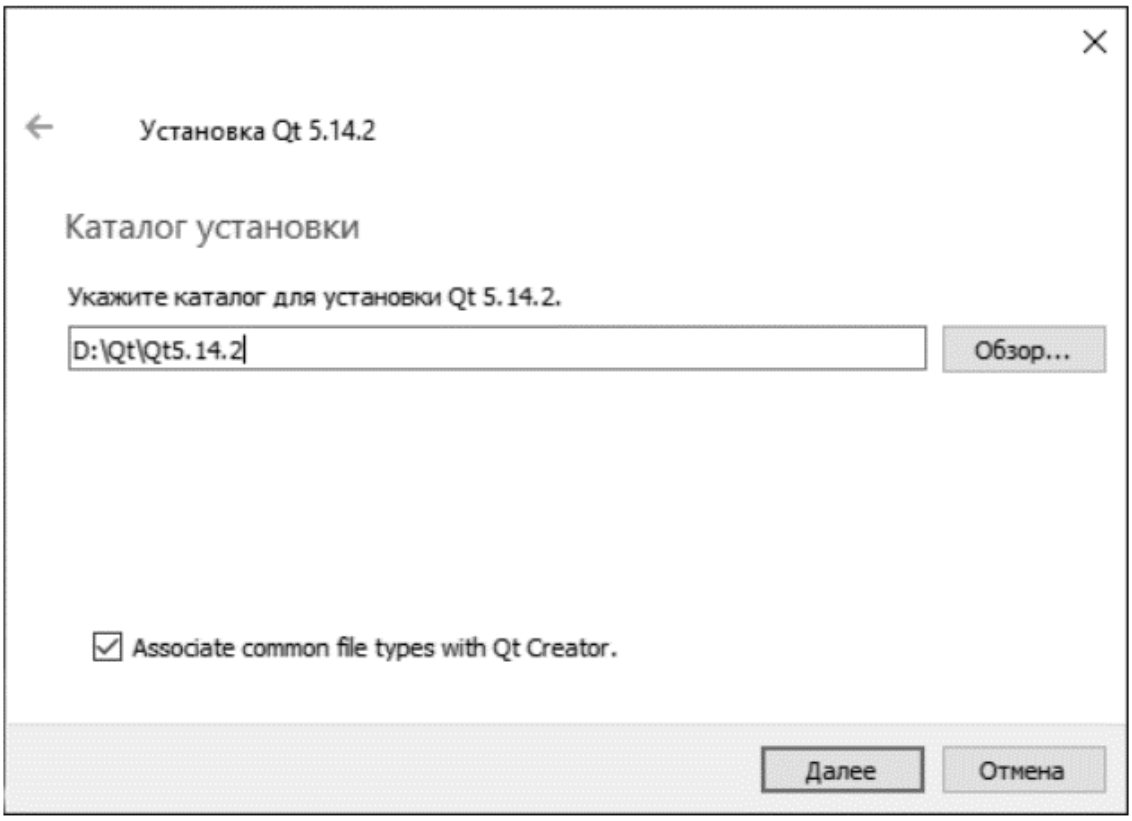


Рисунок 26 – Каталог для установки

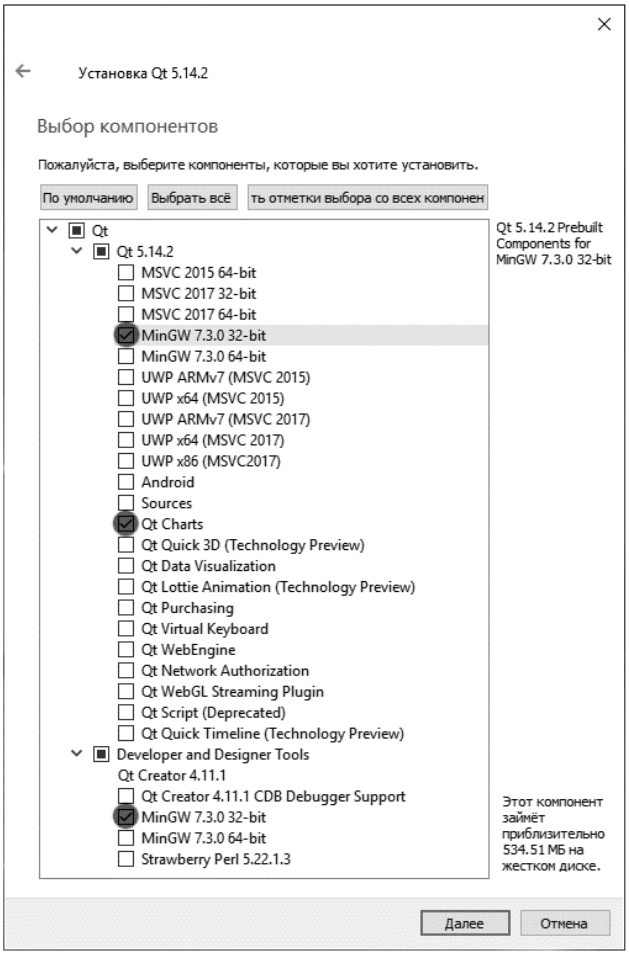


Рисунок 27 – Выбор компонентов

Создаем проект «Приложение QT Widgets» (рисунок 28), заносим на форму необходимые элементы (рисунок 29), такие как кнопки, элементы для ввода и вывода информации: окно для ввода текста, надписи, набор виджетов с вкладками, таблицы, виджеты для отрисовки графиков.

Рисунок 28 – Создание проекта

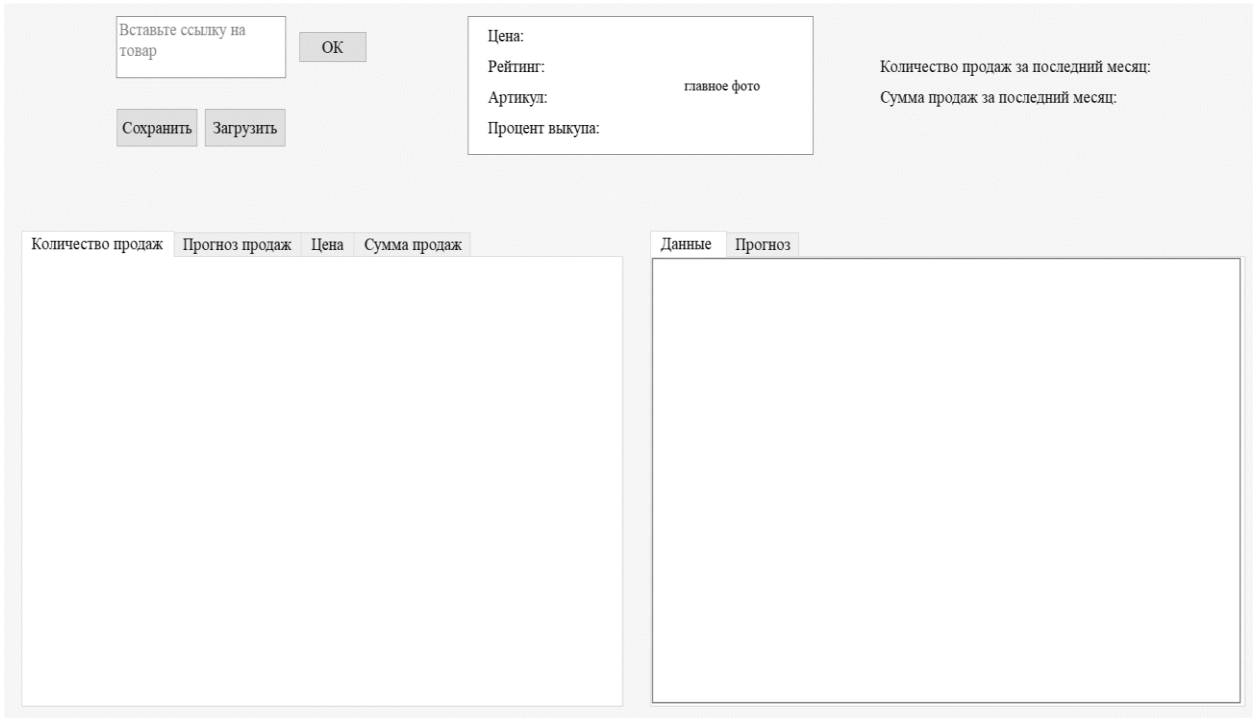


Рисунок 29 – Создание пользовательского интерфейса

Далее создаем слот для кнопки «ОК», который запускает процесс получения и обработки данных. Для этого необходимо получить содержание окна для ввода ссылки на товар и передать информацию в блок, который реализует загрузку html-страницы.

```
std::string url = article.toLocal8Bit().constData();
        std::string filename = "info.html";

        if (downloader.download(url, filename)) {
            std::cout << "File downloaded successfully" <<
std::endl;
        } else {
            std::cerr << "Failed to download file" <<
std::endl;
        }
```

Далее необходимо выполнить парсинг страницы в .csv файл, таким образом получив следующие данные:

- дата (последние 30 дней),
- изменения цены за последние 30 дней,
- количество продаж за последние 30 дней,
- сумма продаж за последние 30 дней,
- общее количество продаж за последние 30 дней,
- общая сумма продаж за последние 30 дней,
- текущая цена товара,
- текущий рейтинг товара,
- артикул товара,
- текущий процент выкупа,
- главное фото.

```
ifstream inputFile(input_path);
 ofstream outputFile(output_csv);
 string line, buf1;
 // Извлекаем данные начиная с chartchart
 while(getline(inputFile,buf1))
```

```
        line += buf1;
    line.erase(0, line.find("chartchart"));

    // Получаем диапазоны данных
    vector<string> lines;
    unsigned long left = 0, right = 0;
    for (int i = 0; i < 5; ++i) {
        left = line.find('[', right);
        if(i==1)
            left = line.find('[', left+1);
        right = line.find(']', left);
        lines.push_back(line.substr(left, right - left +
1));
    }
    // Убираем одинарные кавычки из первой строки
    std::string buf;
    buf.resize(lines[0].size());
    auto it = std::copy_if(lines[0].begin(), lines[0].end(),
buf.begin(), [](char x) { return x != '\''; });
    buf.resize(std::distance(buf.begin(), it));
    lines[0] = move(buf);

    // Парсим строки
    vector<string> data = parse(lines[0]);
    vector<string> sales = parse(lines[1]);
    vector<string> price = parse(lines[3]);
    vector<string> revenue = parse(lines[4]);

    // Записываем данные в CSV
    outputFile << "Date,Price,Sales,Revenue\n";
    for (size_t i = 0; i < data.size(); ++i) {
        outputFile << data[i] << ',' << price[i] << ',' <<
sales[i] << ','  << revenue[i] << '\n';
    }

    inputFile.close();
```

```
    outputFile.close();
```

После завершения записи в файл переходим к прогнозированию количества продаж на следующие 30 дней. Для этого берем данные за последние 30 дней и обучаем на них нейросеть.

Нейросеть обучается с помощью модели «с учителем», для этого инициализируем веса случайными значениями для четырех полносвязных слоев. В качестве функции активации используется Leaky ReLU, так как она добавляет нелинейность, что позволяет модели учить сложные зависимости. Обучение модели происходит в цикле на 10 000 эпох с приминением градиентного спуска.

```
vector<float> Model::leaky_relu(const vector<float>& x,
float alpha) {
    vector<float> result(x.size());
    for (size_t i = 0; i < x.size(); ++i) {
        result[i] = x[i] > 0 ? x[i] : alpha * x[i];
    }
    return result;
}
```

Результат работы заносится в .csv файл.

Далее необходимо представить полученные данные в удобном пользователю виде. Для этого выведем в виде графика зависимости цены, количества продаж и суммы продаж товара от даты, а так же прогнозируемую зависимость количества продаж от даты.

```
    QCustomPlot *wGraphic_price;
    wGraphic_price = new QCustomPlot();
    QGridLayout *layout = new QGridLayout(ui->tab_price);
    layout->addWidget(wGraphic_price,1,0,1,1);

    QVector<double> col_Price =
extractColumn("base_look.csv", 1);

    // Добавляем график на полотно
```

```
    wGraphic_price->addGraph(wGraphic_price->xAxis,
wGraphic_price->yAxis);
    // Устанавливаем координаты точек графика
    wGraphic_price->graph(0)->setData(x1,col_Price);

    // Подписываем оси координат
    wGraphic_price->xAxis->setLabel("Дни");
    wGraphic_price->yAxis->setLabel("Изменения цены");

    // Устанавливаем максимальные и минимальные значения
координат
    wGraphic_price->xAxis->setRange(0,30);
    wGraphic_price->yAxis-
>setRange(*std::min_element(col_Price.begin(),col_Price.end(
))-
10,*std::max_element(col_Price.begin(),col_Price.end())+10);

    // Отрисовываем содержимое полотна
    wGraphic_price->replot();
```

Далее выведем данные за последние 30 дней в виде обобщенной таблицы, а так же таблицу прогноза количества продаж на следующие 30 дней.

```
// Создаём модель данных для отображения таблицы из CSV
файла
    csvModel_isx = new QStandardItemModel(this);
    csvModel_isx->setColumnCount(4);
    csvModel_isx->setHorizontalHeaderLabels(QStringList() <<
"Дата" << "Цена" << "Количество продаж" << "Сумма продаж");
    ui->tableView_vir_table->horizontalHeader()-
>setSectionResizeMode(QHeaderView::Stretch);
    ui->tableView_vir_table->setModel(csvModel_isx);
// Устанавливаем модель в таблицу

    QFile file("base_look.csv");
    if ( !file.open(QFile::ReadOnly | QFile::Text) ) {
```

```
            qDebug() << "File not exists";
        } else {
            // Создаём поток для извлечения данных из файла
            QTextStream in(&file);
            // Считываем данные до конца файла
            while (!in.atEnd())
            {
                // ... построчно
                QString line = in.readLine();
                // Добавляем в модель по строке с элементами
                QList<QStandardItem *> standardItemsList;
                // учитываем, что строка разделяется точкой с
запятой на колонки
                for (QString item : line.split(";")) {
                    standardItemsList.append(new
QStandardItem(item));
                }
                csvModel_isx->insertRow(csvModel_isx-
>rowCount(), standardItemsList);
            }
            file.close();
        }
```

Так же добавим пользователю возможность сохранять полученные данные в файл, а так же загрузить и изучить их повторно, при необходимости.

4.8 Архитектура программы

Архитектура программы состоит из хранилища данных, модулей передачи данных, формирования прогноза, заполнения таблиц и построения графиков. Их взаимодействие показано на рисунке 30.

Рассмотрим основные функции программы:

– on_pushButton_tovar_ok_clicked: обрабатывает ссылку на товар и запускает процесс парсинга данных и вывода результатов,

– downloader: получение страницы HTML,

– parser2: парсинг страницы и запись результата в файл,

– predict: прогнозирование продаж на основе исходных данных,
– csv_isx: запись исходных данных в таблицу,
– csv_progn: запись прогноза продаж в таблицу,
– graph: построение графика.

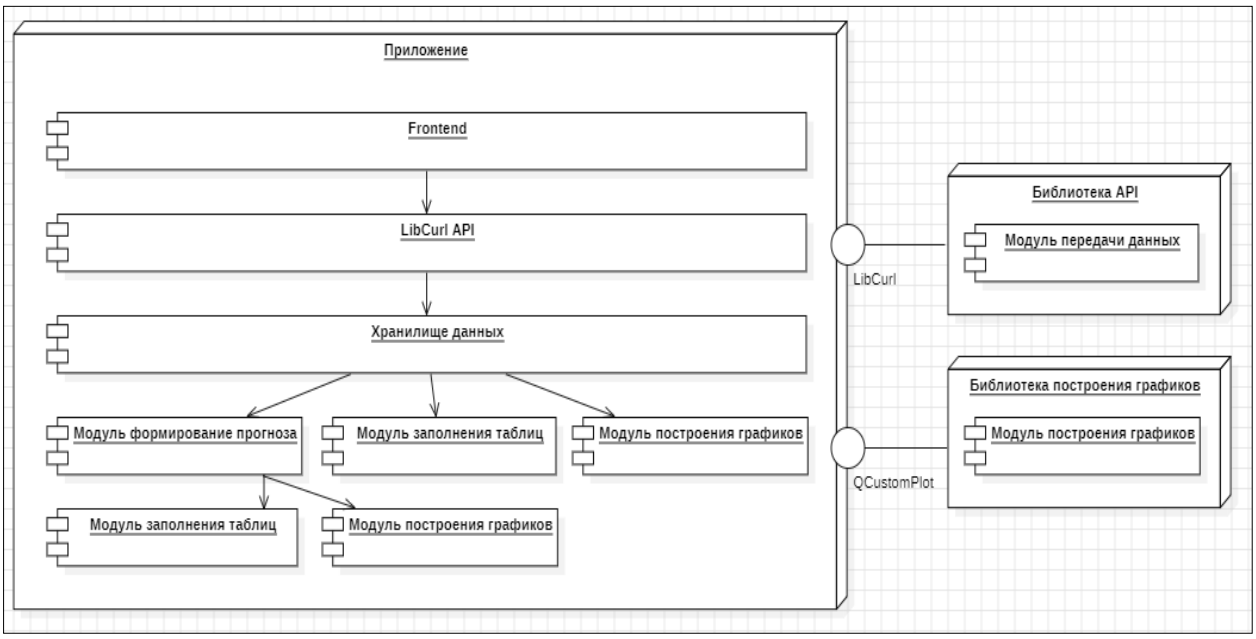


Рисунок 30 – Архитектура программы

4.9 Структура хранения данных

После парсинга исходной страницы данные по продажам товара заносятся в .csv файл, имеющий следующую структуру (рисунок 31).

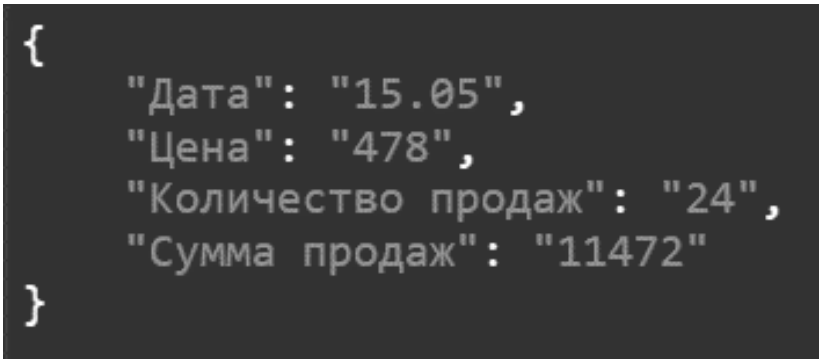


Рисунок 31 – Структура .csv файла с данными по продажам

Файл, хранящий прогноз продаж на следующие 30 дней имеет следующую структуру, представленную на рисунке 32. Пример файла рассмотрен на рисунке 33.

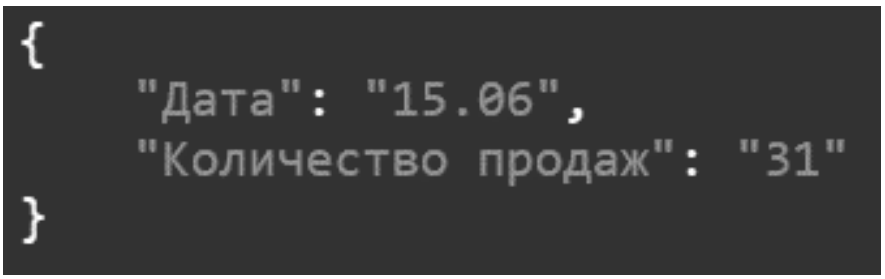


Рисунок 32 – Структура .csv файла с прогнозом продаж

4.10 Описание пользовательской программы

Для запуска приложения необходимо открыть его с помощью ярлыка на рабочем столе, либо через меню «Пуск». После открытия перед пользователем стоит задача вставить ссылку на нужный товар в соответстсвующее окно и нажать кнопку «ОК». Приложение выведет результатвыполнения программы на экран.

4.11 Описание тестовых данных

Для тестирования работоспособности был выбран товар «Генератор мыльных пузырей». Данные были заполнены в формате .csv файла. Пример файла представлен на рисунке 24.

Далее данные по продажам используются для составления обучающей выборки по прогнозированию продаж.

16.май	299	22	6578
17.май	299	20	5980
18.май	299	15	4485
19.май	299	11	3289
20.май	298	26	7748
21.май	298	15	4470
22.май	298	12	3576
23.май	298	19	5662
24.май	298	18	5364
25.май	289	10	2890
26.май	298	15	4470
27.май	298	5	1490
28.май	298	11	3278

Рисунок 33 – Пример .csv файла

Для обучения выбран метод с учителем. Обучение с учителем - это метод машинного обучения, при котором алгоритм обучается на основе

размеченных данных. Это означает, что алгоритму предоставляется набор данных, в котором для каждого примера имеется "правильный ответ" или метка. Алгоритм анализирует эти данные и стремится научиться предсказывать метки для новых, ранее не виденных примеров.

Модель состоит из 4-х слоев, обучение происходит в цикле на 10 000 эпох с применением схемы обновления весов методом градиентного спуска. Для активации используется функция Leaky ReLU.

4.12 Отладка и тестирование

Отладка в ходе разработки проводилась с использованием встроенных в QT Creator средств: точка останова, пошаговое выполнение кода программы, анализ содержимого глобальных и локальных переменных, результаты работы некоторых функций, трассировка, вывод сообщений в консоль. Тестирование проводилось в рабочем порядке в ходе всего цикла разработки. Все выявленные ошибки были исправлены.

В качестве примера хода тестирования и отладки на рисунке 34 представлен процесс просмотра парсинга данных о товаре и записи в переменные.

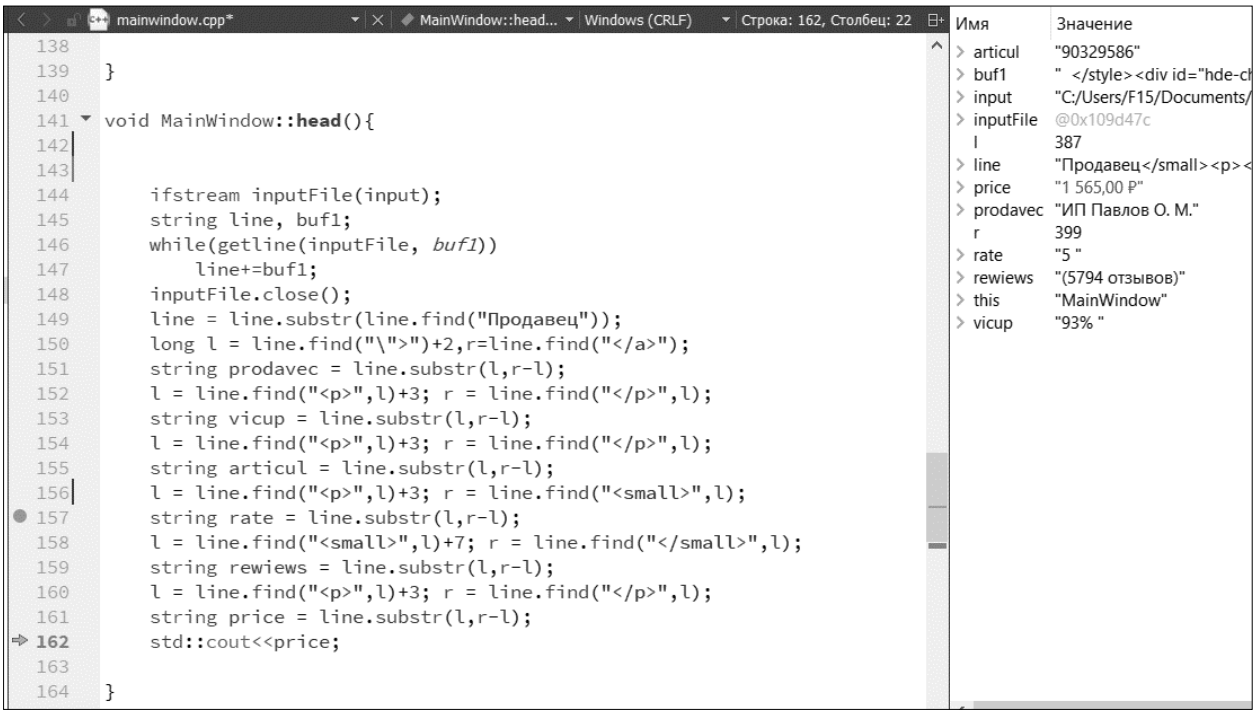


Рисунок 34 – Представление характеристик товара на официальном сайте Wildberries

В рамках работ, описанных в этом разделе, каждый компонент итоговой программы проверен на работоспособность в ходе итераций тестирования, которые выполнялись на протяжении всего цикла разработки каждого из компонентов программы.

4.13 Руководство пользователя по использованию программы

После установки и запуска пользователю отображается окно, представленное на рисунке 35. В этом окне необходимо ввести ссылку на товар и нажать кнопку «ОК» (рисунок 36).

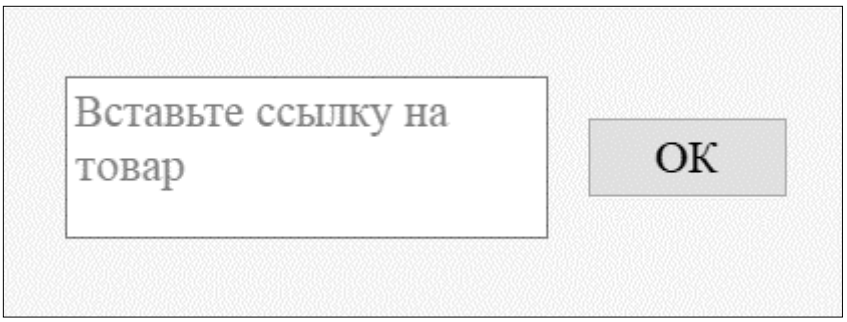


Рисунок 35 – Стартовое окно приложения

Рисунок 36 – Ввод ссылки

Далее пользователю отображается основное окно программы, представленное на рисунке 37. В верхней средней части окна (рисунок 38) можно найти основную информацию по товару.

Рисунок 37 – Основная информация по товару

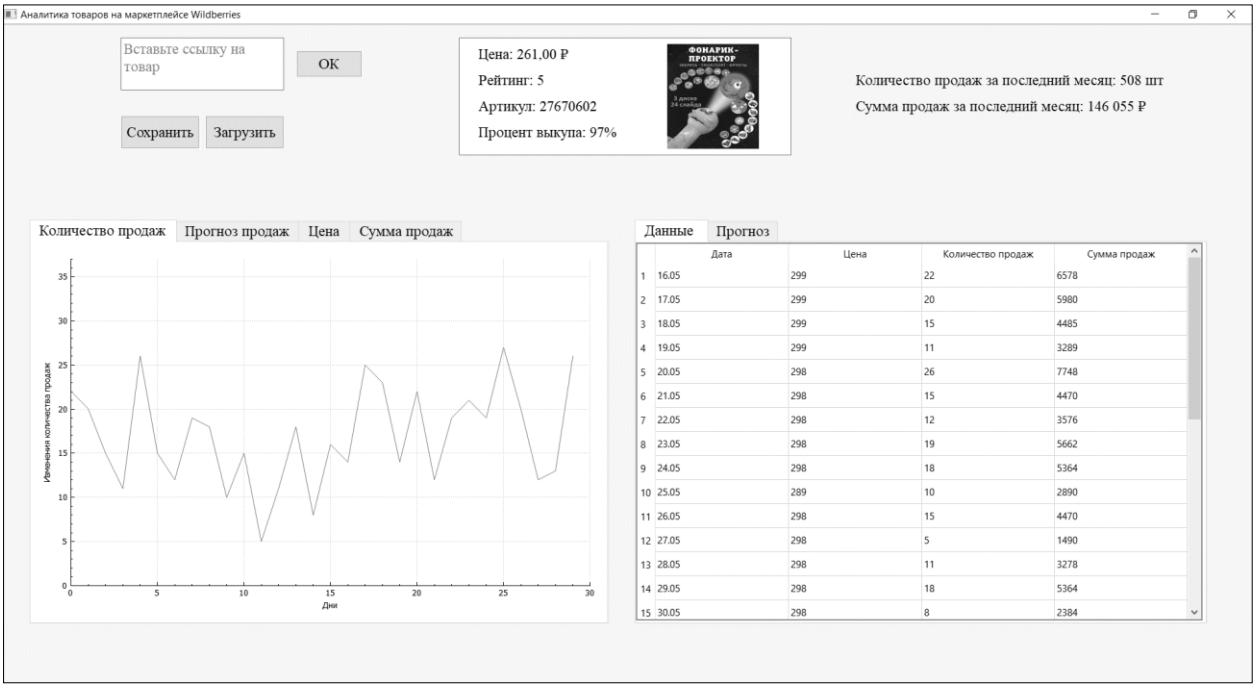


Рисунок 38 – Основное окно приложения

В верхней правой части окна (рисунок 39) можно найти количество продаж и сумму продаж за последний месяц.

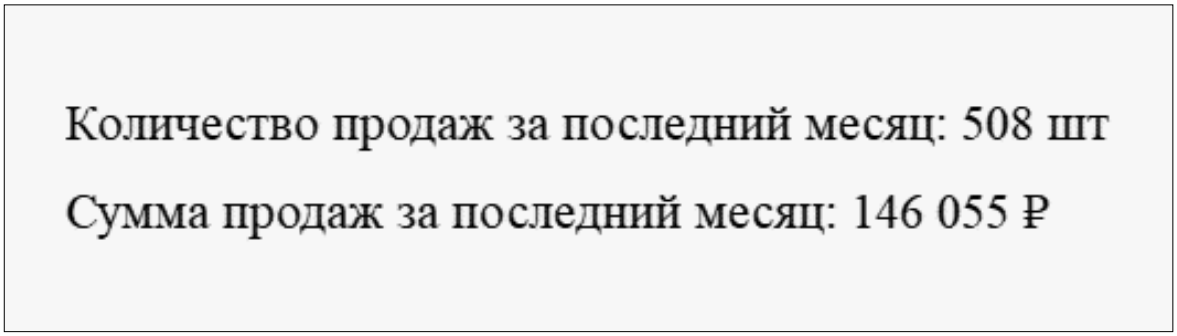


Рисунок 39 – Количество продаж и сумма продаж за последний месяц

В нижней левой части можно ознакомиться с изменением цены, количества продаж, и суммой продаж в течение последнего месяца, а так же прогнозом количества продаж на следующие 30 дней. Графики изменений представлены на рисунках 40, 41, 42, 43.

В нижней правой части окна в виде таблицы представлены данные по цене, количеству продаж и сумме продаж за последний месяц, а так же проноз количества продаж на следующие 30 дней. Таблицы показаны на рисунках 44, 45.

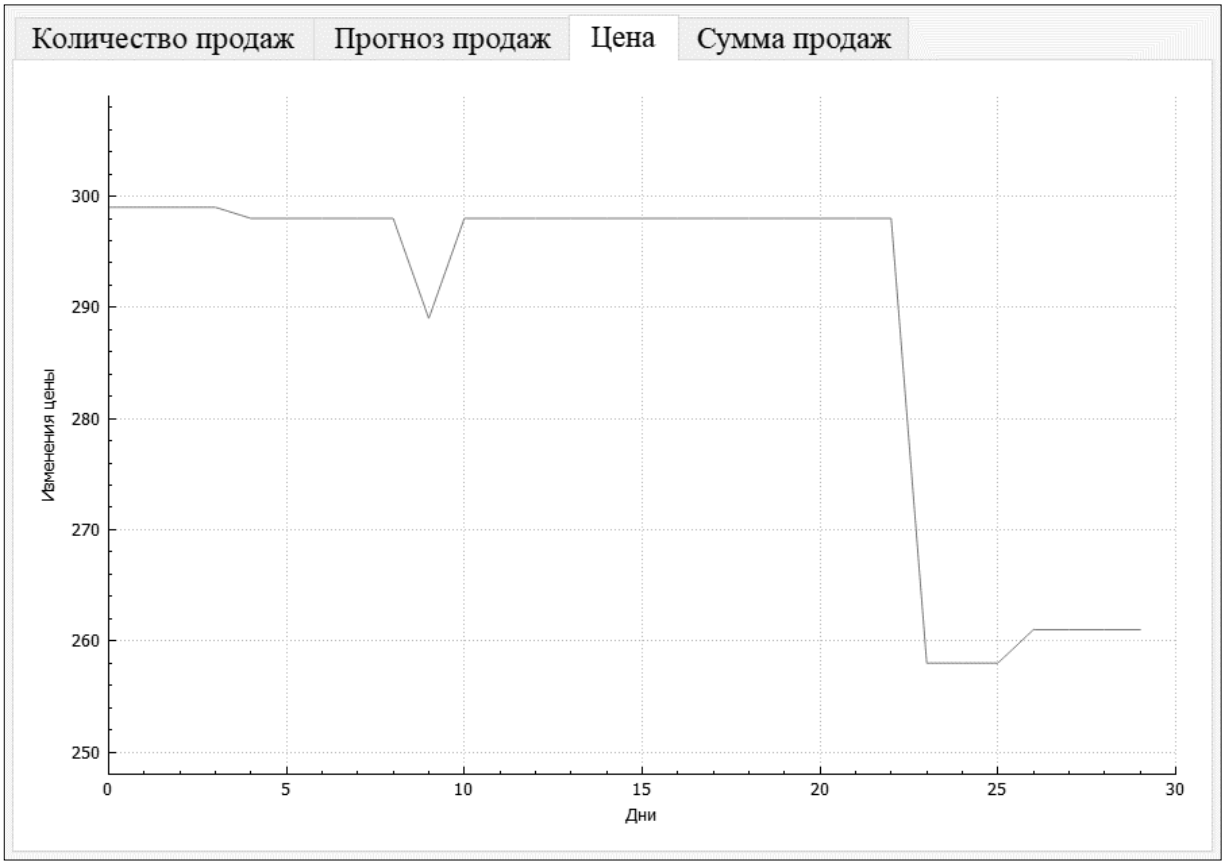


Рисунок 40 – График отношения даты к цене за последний месяц

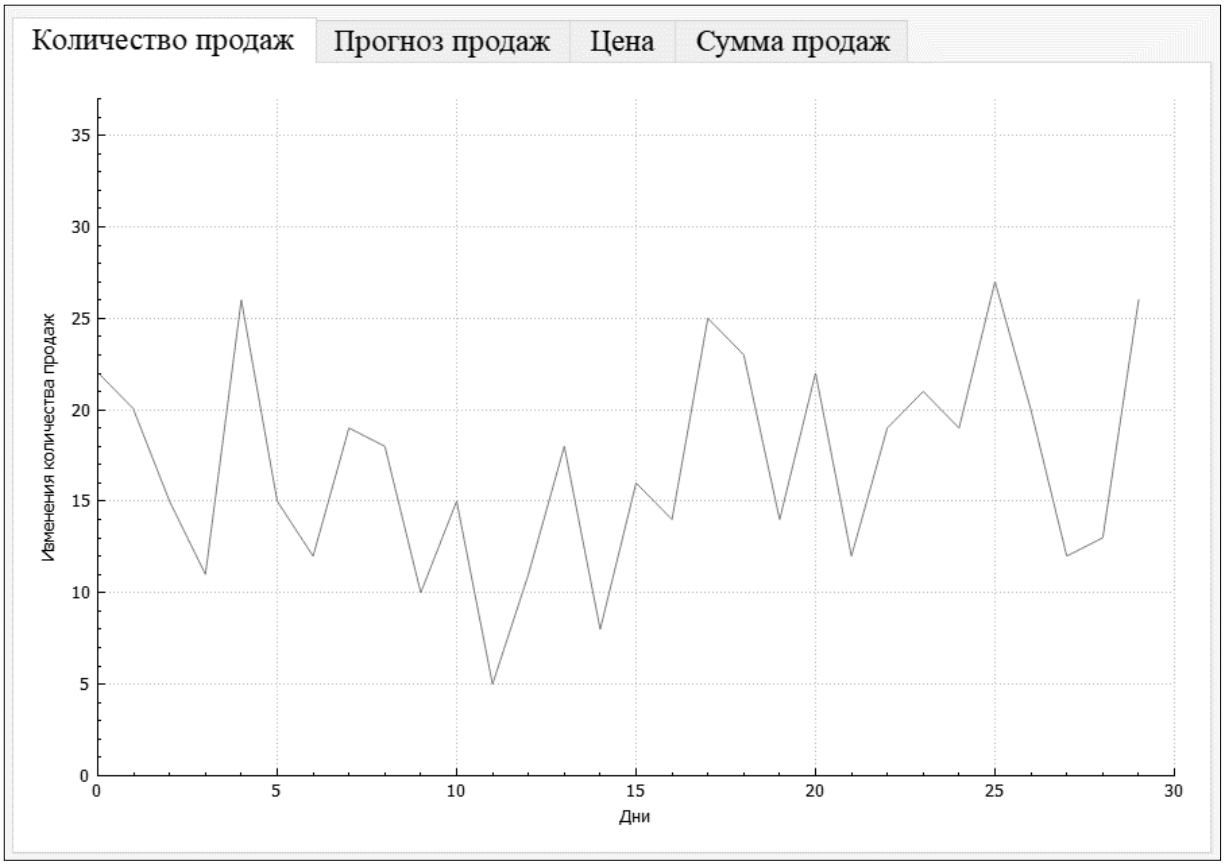


Рисунок 41 – График отношения даты к количеству продаж
за последний месяц

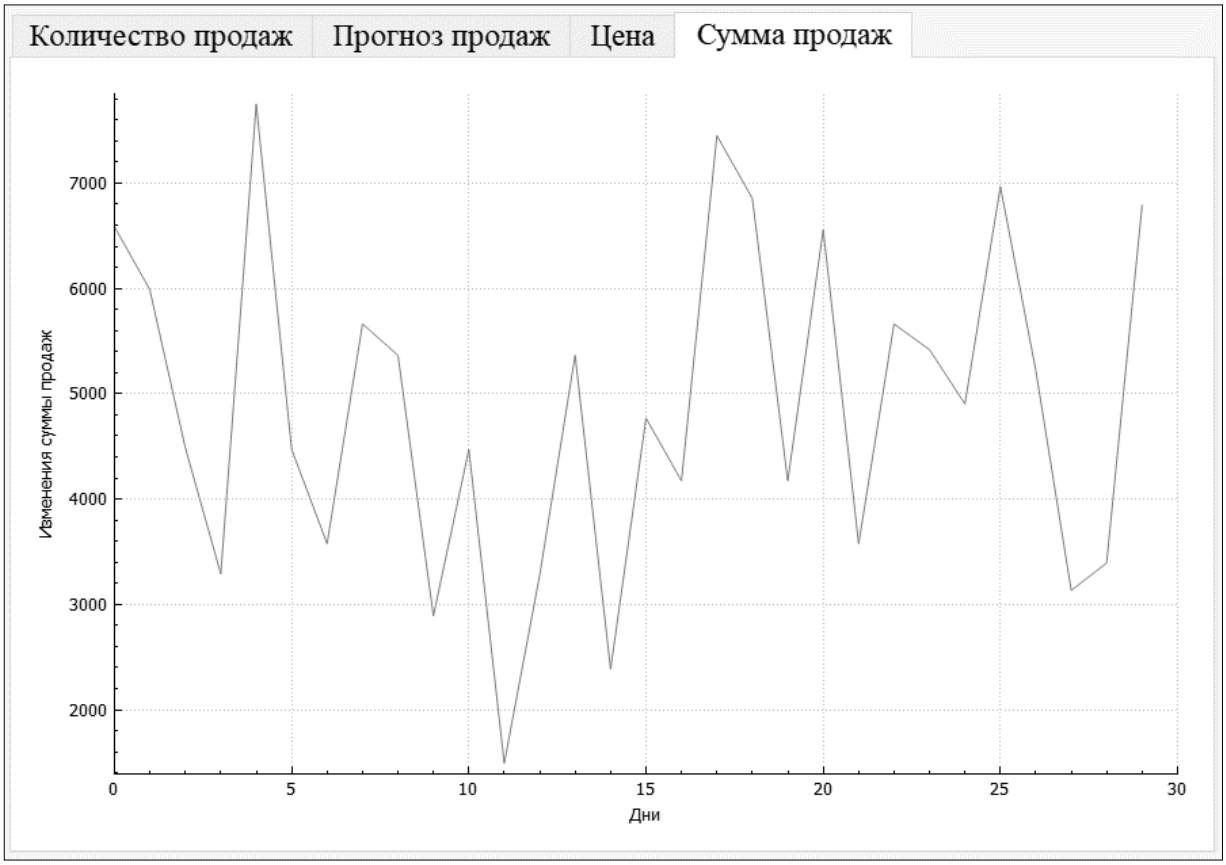


Рисунок 42 – График отношения даты к сумме продаж за последний месяц

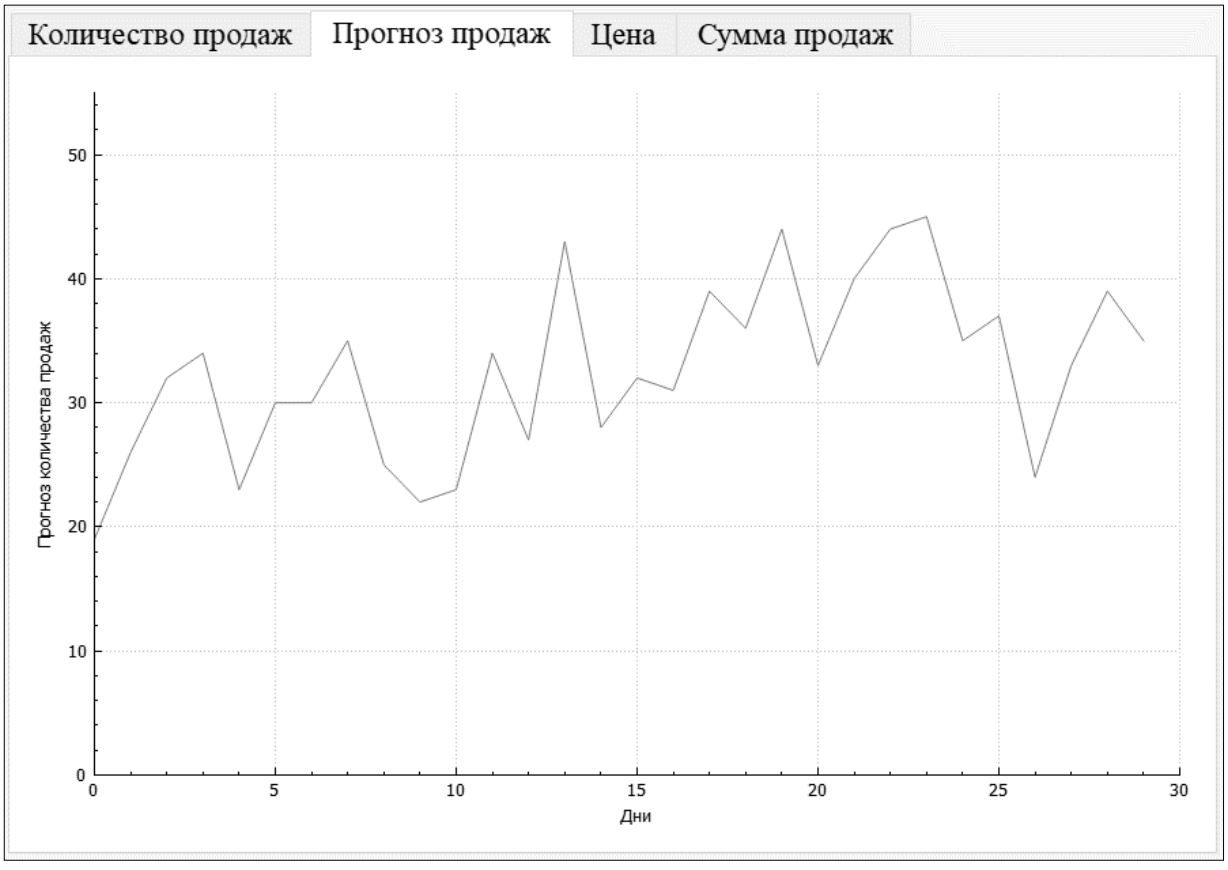


Рисунок 43 – График отношения даты к количеству продаж
на следующие 30 дней

Данные | Прогноз

	Дата	Цена	Количество продаж	Сумма продаж
1	16.05	299	22	6578
2	17.05	299	20	5980
3	18.05	299	15	4485
4	19.05	299	11	3289
5	20.05	298	26	7748
6	21.05	298	15	4470
7	22.05	298	12	3576
8	23.05	298	19	5662
9	24.05	298	18	5364
10	25.05	289	10	2890
11	26.05	298	15	4470
12	27.05	298	5	1490
13	28.05	298	11	3278
14	29.05	298	18	5364
15	30.05	298	8	2384

Рисунок 44 – Таблица данных за предыдущий месяц

Данные | Прогноз

	Дата	Количество продаж
1	15.06	19
2	16.06	26
3	17.06	32
4	18.06	34
5	19.06	23
6	20.06	30
7	21.06	30
8	22.06	35
9	23.06	25
10	24.06	22
11	25.06	23
12	26.06	34
13	27.06	27
14	28.06	43
15	29.06	28

Рисунок 45 – Таблица данных на следующие 30 дней

С помощью кнопок «Сохранить» и «Загрузить» (рисунок 46) пользователь имеет возможность сохранить полученный данные в файл, а так же использовать их в дальнейшей работе.

Рисунок 46 – Кнопки «Сохранить» и «Загрузить»

4.14 Выводы

В этом разделе было приведено описание итогового приложения, описаны тестовые данные и методы, с помощью которых проводилось итоговое функциональное тестирование разработанного приложения, а также составлено руководство пользователя, где подробно с иллюстрациями был описан алгоритм взаимодействия пользователя с приложением.

ЗАКЛЮЧЕНИЕ

В рамках данной выпускной квалификационной работы была разработана программа для обработки, хранения и визуализации информации для использования продавцами на маркетплейсе «Wildberries». В качестве пользовательского интерфейса было реализовано приложение для ПК на ОС «Windows». В основу механизма прогноза количества продаж были положены искусственные нейронные сети, которые обучены одним из методов машинного обучения с учителем. Таким образом, можно заключить, что разработанный проект полностью соответствует поставленным целям.

Данным приложением, благодаря простому пользовательскому интерфейсу, может пользоваться любой уверенный пользователь ПК.

Дальнейшее развитие разработанной системы может включать в себя разработку мобильного приложения, добавление новых критериев для пронозирования.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Маккэндлесс Д. Инфографика. Самые интересные данные в графическом представлении. – М.: Манн, Иванов и Фербер, 2013. – 264 с.
2. StudFiles: сайт. – URL: https://studfile.net/preview/5411061/page:63/ (дата обращения: 21.10.2022). – Средства структурирования и визуализации данных. Электронные помощники аналитика.
3. Алексей Колоколов. Институт бизнес аналитики: сайт. – URL: https://alexkolokolov.com/ru/blog/instrumenty-vizualizacii (дата обращения: 17.11.2022). – Инструменты визуализации: сервисы, программы и средства обработки данных.
4. Нейт Сильвер. Сигнал и Шум. Почему одни прогнозы сбываются, а другие — нет // Азбука-Аттикус, КоЛибри, 2015
5. Агапов, А.В. Обработка и обеспечение безопасности электронных данных: Учебное пособие / А.В. Агапов, Т.В. Алексеева и др. - М.: МФПУ Синергия, 2012. - 592 с.
6. Захаренко, С. П. Парсер. Получение информации с сайтов / С. П. Захаренко, К. А. Гарамов. — Текст : непосредственный // Юный ученый. — 2023. — № 3 (66). — С. 97-100. — URL: https://moluch.ru/young/archive/66/3490/.
7. Используйте искусственный интеллект для повышения точности прогнозов продаж // Блог Втигер URL: https://www.vtiger.com/ru/blog/leverage-ai-to-boost-sales-forecast-accuracy/ (дата обращения 12.05.2023).
8. Хенрик Б., Джозеф Р., Марк Ф. Машинное обучение. – Издательский дом «Питер», 2017. – 336 с.
9. Академия искусственного интеллекта [Электронный ресурс]. – URL: https://ai-academy.ru/ (дата обращения: 03.09.2023).
10. Артемьев Е.А., Мокшин В.В. Прогнозирование продаж в ритейле на основе методов машинного обучения // Сборник научных статей. – Казань: Казанский национальный исследовательский технический университет, 2020. – С. 1-3.

11. Сара Бослаф. Статистика для всех. – М.: ДМК Пресс, 2017 – 586.

12. О внедрении программ по искусственному интеллекту [Электронный ресурс]. – URL: https://goo.su/zyUUMdF (дата обращения: 03.11.2022).

13. Лучшие платформы для разработки приложений в 2023 году: сайт. – URL: https:// g2.com/categories/application-development-platforms (дата обращения: 19.11.2022). – Лучшие платформы для разработки приложений.

14. Желязны Д. Говори на языке диаграмм: Пособие по визуальным коммуникациям для руководителей / Пер. с англ. - М.: Институт комплексных стратегических исследований, 2004 - 220 с.

15. Винстон Уэйн, Бизнес-моделирование и анализ данных. Решение актуальных задач с помощью Microsoft Excel. 6-е издание. — СПб.: Питер, 2021. — 944 с.: ил. — (Серия «IT для бизнеса»).

16. Парсинг веб-сайтов / [Электронный ресурс] // VC.ru : [сайт]. — URL: https://vc.ru/marketing/239915-rukovodstvo-po-parsingu-veb-saytov-v-2021-godu.

17. Хуан Д. Алгоритмы извлечения информации из текстов, парсинг веб-страниц // Актуальные исследования. 2022. №30 (109). С. 21-24. URL: https://apni.ru/article/4419-algoritmi-izvlecheniya-informatsii-iz-tekstov.

18. Фридл Дж. Регулярные выражения, 3-е издание. – Пер. с англ. – СПб.: Символ Плюс, 2008. – 608 с.

19. Дерево DOM : сайт. — URL: https://maxello.gitbooks.io/js-note/content/15_dom_2/derevo_dom.html (дата обращения: 16.02.2023).

20. XPath — примеры запросов в html для парсинга сайта : сайт. – URL: https://habr.com/ru/articles/753332/ (дата обращения: 03.03.2023)

21. Официальный сайт «QueryList». [Электронный ресурс]. URL: https://querylist.cc/ – Свободный доступ. (дата посещения 21.03.2023).

22. Батаев А.В. Операционные системы и среды / Батаев А.В., Налютин Н.Ю., Синицын С.В. – М.: Академия, 2015. – 272 с.

23. Qt Creator : сайт. – URL: https://progsoft.net/ru/software/qt-creator (дата обращения: 11.12.2022). – Qt Creator альтернативы и похожие программы.ё

ПРИЛОЖЕНИЕ А

(обязательное)

Программа для обработки, хранения и визуализации информации

Листинг программы

Файл «mainwindow.cpp»

```
#include "mainwindow.h"
#include "ui_mainwindow.h"

MainWindow::MainWindow(QWidget *parent)
    : QMainWindow(parent)
    , ui(new Ui::MainWindow)
{
    ui->setupUi(this);
    setlocale(LC_ALL, "");


    QWidget::showMaximized();

    ui->tableView_vir_table = new QTableView;
    ui->scrollArea_3->setWidget(ui->tableView_vir_table);

    ui->tableView_progn_table = new QTableView;
    ui->scrollArea_4->setWidget(ui->tableView_progn_table);


}

MainWindow::~MainWindow()
{
    delete ui;
}


void MainWindow::on_pushButton_tovar_ok_clicked()
{
    article = ui->textBrowser_tovar->toPlainText();
    ui->textBrowser_tovar->clear();

        HtmlDownloader downloader;

        std::string url = article.toLocal8Bit().constData();
        std::string filename = "info.html";

        if (downloader.download(url, filename)) {
            std::cout << "File downloaded successfully" <<
std::endl;
        } else {
            std::cerr << "Failed to download file" << std::endl;
        }

    parser2 base("info.html", "base.csv");
    neuro model("base.csv");
    for(auto i: model.predict(30))
        cout<<i.first<<", "<<i.second<<endl;
    model.create_base();
```

```
    //заголовки
    head();
    //таблицы
    csv_isx();
    csv_progn();
    //построить график
    graph();

}

//заполнение таблиц исходными данными
void MainWindow::csv_isx(){

    // Создаём модель данных для отображения таблицы из CSV
файла
    csvModel_isx = new QStandardItemModel(this);
    csvModel_isx->setColumnCount(4);
    csvModel_isx->setHorizontalHeaderLabels(QStringList() <<
"Дата" << "Цена" << "Количество продаж" << "Сумма продаж");
    ui->tableView_vir_table->horizontalHeader()-
>setSectionResizeMode(QHeaderView::Stretch);
    ui->tableView_vir_table->setModel(csvModel_isx); //
Устанавливаем модель в таблицу

    QFile file("base_look.csv");
    if ( !file.open(QFile::ReadOnly | QFile::Text) ) {
        qDebug() << "File not exists";
    } else {
        // Создаём поток для извлечения данных из файла
        QTextStream in(&file);
        // Считываем данные до конца файла
        while (!in.atEnd())
        {
            // ... построчно
            QString line = in.readLine();
            // Добавляем в модель по строке с элементами
            QList<QStandardItem *> standardItemsList;
            // учитываем, что строка разделяется точкой с
запятой на колонки
            for (QString item : line.split(";")) {
                standardItemsList.append(new
QStandardItem(item));
            }
            csvModel_isx->insertRow(csvModel_isx->rowCount(),
standardItemsList);
        }
        file.close();
    }

}

//заполнение таблицы прогноз
```

```
void MainWindow::csv_progn(){

    // Создаём модель данных для отображения таблицы из CSV
файла
    csvModel_progn = new QStandardItemModel(this);
    csvModel_progn->setColumnCount(2);
    csvModel_progn->setHorizontalHeaderLabels(QStringList() <<
"Дата" << "Количество продаж" );
    ui->tableView_progn_table->horizontalHeader()-
>setSectionResizeMode(QHeaderView::Stretch);
    ui->tableView_progn_table->setModel(csvModel_progn); //
Устанавливаем модель в таблицу

    QFile file("predict.csv");
    if ( !file.open(QFile::ReadOnly | QFile::Text) ) {
        qDebug() << "File not exists";
    } else {
        // Создаём поток для извлечения данных из файла
        QTextStream in(&file);
        // Считываем данные до конца файла
        while (!in.atEnd())
        {
            // ... построчно
            QString line = in.readLine();
            // Добавляем в модель по строке с элементами
            QList<QStandardItem *> standardItemsList;
            // учитываем, что строка разделяется точкой с
запятой на колонки
            for (QString item : line.split(";")) {
                standardItemsList.append(new
QStandardItem(item));
            }
            csvModel_progn->insertRow(csvModel_progn-
>rowCount(), standardItemsList);
        }
        file.close();
    }


}

//заполнение данными заголовков
void MainWindow::head(){

    using namespace std;
    string input = "info.html";
    ifstream inputFile(input);
    string line, buf1;
    while(getline(inputFile, buf1))
        line+=buf1;
    inputFile.close();
    line = line.substr(line.find("Продавец"));
    long l = line.find("\">")+2,r=line.find("</a>");
```

```
    string prodavec = line.substr(l,r-l);
    l = line.find("<p>",l)+3; r = line.find("</p>",l);
    string vicup = line.substr(l,r-l);
    ui->label_har_percent->setText("Процент выкупа: " +
QString::fromStdString(vicup));
    l = line.find("<p>",l)+3; r = line.find("</p>",l);
    string articul = line.substr(l,r-l);
    ui->label_har_article->setText("Артикул: " +
QString::fromStdString(articul));
    l = line.find("<p>",l)+3; r = line.find("<small>",l);
    string rate = line.substr(l,r-l);
    ui->label_har_reiting->setText("Рейтинг: " +
QString::fromStdString(rate));
    l = line.find("<small>",l)+7; r = line.find("</small>",l);
    string rewiews = line.substr(l,r-l);
    l = line.find("<p>",l)+3; r = line.find("</p>",l);
    string price = line.substr(l,r-l);
    ui->label_har_price->setText("Цена: " +
QString::fromStdString(price));


    QPixmap pixmap("1.jpg");
    ui->label_har_photo->setPixmap(pixmap);
    ui->label_har_photo->setMask(pixmap.mask());
    ui->label_har_photo->setScaledContents( true );
    ui->label_har_photo->setSizePolicy( QSizePolicy::Ignored,
QSizePolicy::Ignored );


    string input2 = "info.html";
    ifstream inputFile2(input2);
    string line2, buf2;
    while(getline(inputFile2, buf2))
        line2+=buf2;
    inputFile2.close();
    line2.erase(0,line2.find("Выручка")+20);
    int rev1 = line2.find("Выручка");
    string revenue1 = line2.substr(rev1+15,line2.find("и",rev1)-
rev1-15);
    ui->label_sum_prod->setText("Сумма продаж за последний
месяц: " + QString::fromStdString(revenue1));
    int sol1 = line2.find("продано");
    string sold1 = line2.substr(sol1+15,line2.find("шт.",sol1)-
sol1-15);
    sold1+="шт";
    ui->label_kol_prod->setText("Количество продаж за последний
месяц: " + QString::fromStdString(sold1));

}
//запись данных в вектор
QVector<double> extractColumn(const string& filename, int
columnNumber) {
    ifstream file(filename);
```

```
    QVector<double> columnData;

    if (!file.is_open()) {
        cerr << "Could not open the file " << filename << " for
reading." << endl;
        return columnData;
    }

    string line;
    while (getline(file, line)) {
        istringstream iss(line);
        string token;
        int currentColumn = 0;

        while (getline(iss, token, ';')) {
            if (currentColumn == columnNumber) {
                try {
                    columnData.append(stod(token));
                } catch (const invalid_argument& e) {
                    cerr << "Invalid data: " << token << endl;
                } catch (const out_of_range& e) {
                    cerr << "Data out of range: " << token <<
endl;
                }
                break;
            }
            currentColumn++;
        }
    }

    file.close();
    return columnData;
}

//построить график
void MainWindow::graph(){

    QVector<double> x1(30);

        x1[0] = 0;
        x1[1] = 1;
        x1[2] = 2;
        x1[3] = 3;
        x1[4] = 4;
        x1[5] = 5;
        x1[6] = 6;
        x1[7] = 7;
        x1[8] = 8;
        x1[9] = 9;
        x1[10] = 10;
        x1[11] = 11;
        x1[12] = 12;
        x1[13] = 13;
```

```
        x1[14] = 14;
        x1[15] = 15;
        x1[16] = 16;
        x1[17] = 17;
        x1[18] = 18;
        x1[19] = 19;
        x1[20] = 20;
        x1[21] = 21;
        x1[22] = 22;
        x1[23] = 23;
        x1[24] = 24;
        x1[25] = 25;
        x1[26] = 26;
        x1[27] = 27;
        x1[28] = 28;
        x1[29] = 29;


//---график для цены
    QCustomPlot *wGraphic_price;
    wGraphic_price = new QCustomPlot();
    QGridLayout *layout = new QGridLayout(ui->tab_price);
    layout->addWidget(wGraphic_price,1,0,1,1);

    QVector<double> col_Price = extractColumn("base_look.csv",
1);

    // Добавляем график на полотно
    wGraphic_price->addGraph(wGraphic_price->xAxis,
wGraphic_price->yAxis);
    wGraphic_price->graph(0)->setData(x1,col_Price);     //
Устанавливаем координаты точек графика

    // Подписываем оси координат
    wGraphic_price->xAxis->setLabel("Дни");
    wGraphic_price->yAxis->setLabel("Изменения цены");

    // Устанавливаем максимальные и минимальные значения
координат
    wGraphic_price->xAxis->setRange(0,30);
    wGraphic_price->yAxis-
>setRange(*std::min_element(col_Price.begin(),col_Price.end())-
10,*std::max_element(col_Price.begin(),col_Price.end())+10);

    // Отрисовываем содержимое полотна
    wGraphic_price->replot();


//--график для суммы продаж
    QCustomPlot *wGraphic_sum;
    wGraphic_sum = new QCustomPlot();
    QGridLayout *layout_sum = new QGridLayout(ui-
>tab_summ_prod);
```

```
    layout_sum->addWidget(wGraphic_sum,1,0,1,1);

    QVector<double> col_Sum = extractColumn("base_look.csv", 3);

    // Добавляем график на полотно
    wGraphic_sum->addGraph(wGraphic_sum->xAxis, wGraphic_sum-
>yAxis);
    wGraphic_sum->graph(0)->setData(x1,col_Sum);     //
Устанавливаем координаты точек графика

    // Подписываем оси координат
    wGraphic_sum->xAxis->setLabel("Дни");
    wGraphic_sum->yAxis->setLabel("Изменения суммы продаж");

    // Устанавливаем максимальные и минимальные значения
координат
    wGraphic_sum->xAxis->setRange(0,30);
    wGraphic_sum->yAxis-
>setRange(*std::min_element(col_Sum.begin(),col_Sum.end())-
100,*std::max_element(col_Sum.begin(),col_Sum.end())+100);

    // Отрисовываем содержимое полотна
    wGraphic_sum->replot();


//-----график для количества продаж
    QCustomPlot *wGraphic_kol;
    wGraphic_kol = new QCustomPlot();
    QGridLayout *layout_kol = new QGridLayout(ui->tab_kol_prod);
    layout_kol->addWidget(wGraphic_kol,1,0,1,1);

    QVector<double> col_Kol = extractColumn("base_look.csv", 2);

    // Добавляем график на полотно
    wGraphic_kol->addGraph(wGraphic_kol->xAxis, wGraphic_kol-
>yAxis);
    wGraphic_kol->graph(0)->setData(x1,col_Kol);     //
Устанавливаем координаты точек графика


    // Подписываем оси координат
    wGraphic_kol->xAxis->setLabel("Дни");
    wGraphic_kol->yAxis->setLabel("Изменения количества
продаж");

    // Устанавливаем максимальные и минимальные значения
координат
    wGraphic_kol->xAxis->setRange(0,30);
    wGraphic_kol->yAxis-
>setRange(0,*std::max_element(col_Kol.begin(),col_Kol.end())+10)
;

    // Отрисовываем содержимое полотна
```

```
    wGraphic_kol->replot();

//----график для прогноза продаж
    QCustomPlot *wGraphic_progn;
    wGraphic_progn = new QCustomPlot();
    QGridLayout *layout_progn = new QGridLayout(ui-
>tab_progn_prod);
    layout_progn->addWidget(wGraphic_progn,1,0,1,1);

    QVector<double> col_Progn = extractColumn("predict.csv", 1);

    // Добавляем график на полотно
    wGraphic_progn->addGraph(wGraphic_progn->xAxis,
wGraphic_progn->yAxis);
    wGraphic_progn->graph(0)->setData(x1,col_Progn);     //
Устанавливаем координаты точек графика

    // Подписываем оси координат
    wGraphic_progn->xAxis->setLabel("Дни");
    wGraphic_progn->yAxis->setLabel("Прогноз количества
продаж");

    // Устанавливаем максимальные и минимальные значения
координат
    wGraphic_progn->xAxis->setRange(0,30);
    wGraphic_progn->yAxis-
>setRange(0,*std::max_element(col_Progn.begin(),col_Progn.end())
+10);

    // Отрисовываем содержимое полотна
    wGraphic_progn->replot();

}
```

Файл «mainwindow.h»

```
#ifndef MAINWINDOW_H
#define MAINWINDOW_H

#include <QMainWindow>
#include <QTableWidget>
#include <QTabWidget>
#include <QScrollArea>
#include <QDebug>
#include <QtCore>
#include <qcustomplot.h>
#include <neuro.h>
#include <parser2.h>
#include <download.h>
#include <QLabel>
#include <QDate>
#include <QVector>
#include <QStandardItemModel>
```

```
QT_BEGIN_NAMESPACE
namespace Ui { class MainWindow; }
QT_END_NAMESPACE

class MainWindow : public QMainWindow
{
    Q_OBJECT

public:
    MainWindow(QWidget *parent = nullptr);
    ~MainWindow();
    QString article;
    QString article_mayak;

    void graph();
    QVector< QVector<double> > vec_sale;
    QVector <double> vec_s;

    QStandardItemModel *csvModel_isx;
    void csv_isx();
    QStandardItemModel *csvModel_progn;
    void csv_progn();

    void head();//заполнение шапки


private slots:
    void on_pushButton_tovar_ok_clicked();

private:
    Ui::MainWindow *ui;
};
#endif // MAINWINDOW_H
```

Файл «download.cpp»

```
#include <download.h>

bool download(const std::string& url, const std::string&
filename) {
    if (!curl) {
        return false;
    }

    // Открываем файл для записи
    FILE *file = fopen(filename.c_str(), "wb");
    if (!file) {
        std::cerr << "Failed to open file for writing" <<
std::endl;
        return false;
```

```
    }

    // Устанавливаем URL для скачивания
    curl_easy_setopt(curl, CURLOPT_URL, url.c_str());

    // Устанавливаем функцию обратного вызова для записи данных
в файл
    curl_easy_setopt(curl, CURLOPT_WRITEFUNCTION, write_data);

    // Устанавливаем файл для записи данных
    curl_easy_setopt(curl, CURLOPT_WRITEDATA, file);

    // Выполняем запрос
    CURLcode res = curl_easy_perform(curl);
    if (res != CURLE_OK) {
        std::cerr << "Failed to download file: " <<
curl_easy_strerror(res) << std::endl;
        fclose(file);
        return false;
    }

    // Закрываем файл
    fclose(file);

    return true;
}
```

Файл «download.h»

```
#ifndef DOWNLOAD_H
#define DOWNLOAD_H
#include <iostream>
#include <fstream>
#include <curl/curl.h>

class HtmlDownloader {
public:
    HtmlDownloader() {
        curl_global_init(CURL_GLOBAL_ALL);
        curl = curl_easy_init();
        if (!curl) {
            std::cerr << "Failed to initialize libcurl" <<
std::endl;
        }
    }

    ~HtmlDownloader() {
        if (curl) {
            curl_easy_cleanup(curl);
        }
        curl_global_cleanup();
    }
```

```
private:
    CURL *curl;

    // Колбэк-функция для записи данных в файл
    static size_t write_data(void *ptr, size_t size, size_t
nmemb, FILE *stream) {
        size_t written = fwrite(ptr, size, nmemb, stream);
        return written;
    }
};

#endif // DOWNLOAD_H
```

Файл «parser2.cpp»

```
#include "parser2.h"

vector<string> parser2::parse(const string& str) {
    setlocale(LC_ALL, "");
    string buf = "";
    vector<string> res;
    for (int i = 0; i < str.size(); i++) {
        if (str[i] == '[')
            continue;
        if (str[i] == ']') {
            res.push_back(buf);
            break;
        }
        if (str[i] == ',') {
            res.push_back(buf);
            buf = "";
        } else {
            buf += str[i];
        }
    }
    return res;
}
parser2::parser2(const string& input_path, const string&
output_csv) {
    setlocale(LC_ALL, "");
    ifstream inputFile(input_path);
    ofstream outputFile(output_csv);
    string line, buf1;
    // Извлекаем данные начиная с chartchart
    while(getline(inputFile,buf1))
        line += buf1;
    line.erase(0, line.find("chartchart"));

    // Получаем диапазоны данных
    vector<string> lines;
    unsigned long left = 0, right = 0;
```

```
    for (int i = 0; i < 5; ++i) {
        left = line.find('[', right);
        if(i==1)
            left = line.find('[', left+1);
        right = line.find(']', left);
        lines.push_back(line.substr(left, right - left + 1));
    }
    // Убираем одинарные кавычки из первой строки
    std::string buf;
    buf.resize(lines[0].size());
    auto it = std::copy_if(lines[0].begin(), lines[0].end(),
buf.begin(), [](char x) { return x != '\''; });
    buf.resize(std::distance(buf.begin(), it));
    lines[0] = move(buf);

    // Парсим строки

    vector<string> data = parse(lines[0]);
    vector<string> sales = parse(lines[1]);
    vector<string> price = parse(lines[3]);
    vector<string> revenue = parse(lines[4]);

    // Записываем данные в CSV
    outputFile << "Date,Price,Sales,Revenue\n";
    for (size_t i = 0; i < data.size(); ++i) {
        outputFile << data[i] << ',' << price[i] << ',' <<
sales[i] << ','  << revenue[i] << '\n';
    }

    inputFile.close();
    outputFile.close();

    cout << "Data has been written to parser.csv" << endl;
}
```

Файл «parser2.h»

```
#ifndef PARSER2_H
#define PARSER2_H

#include <iostream>
#include <fstream>
#include <string>
#include <vector>
#include <algorithm>
#include <map>
#include <QtCore>

using namespace std;

class parser2 {
    vector<string> parse(const string& str);
public:
```

```
    parser2(const string& input_path = "info.html", const
string& output_path="parser.csv");
};



#endif //PARSER2_H
```

Файл «neuro.cpp»

```
#include "neuro.h"

using namespace std;

std::string dayOfYearToDate(int dayOfYear) {
    // Количество дней в каждом месяце для невисокосного года
    int daysInMonth[] = {31, 28, 31, 30, 31, 30, 31, 31, 30, 31,
30, 31};

    int month = 0;
    // Найти месяц и день
    while (dayOfYear > daysInMonth[month]) {
        dayOfYear -= daysInMonth[month];
        month++;
    }

    // month + 1, потому что индексация месяцев начинается с 0
    std::ostringstream oss;
    oss << std::setw(2) << std::setfill('0') << dayOfYear << "."
        << std::setw(2) << std::setfill('0') << (month + 1);

    return oss.str();
}
void writeSetToCSV(const std::set<std::pair<std::string,
int>,StringComparator>& data, const std::string& filename) {
    std::ofstream file(filename);

    if (!file.is_open()) {
        std::cerr << "Could not open the file " << filename << "
for writing." << std::endl;
        return;
    }

    // Записываем заголовок CSV
    //file << "Date; Prediction\n";

    // Записываем данные
    for (const auto& entry : data) {
        file << entry.first << "; " << entry.second << "\n";
    }

    file.close();
}
```

```
// Конструктор модели
Model::Model() {
    // Инициализация весов случайными значениями и смещений
нулями
    srand(static_cast<unsigned int>(time(nullptr)));

    fc1_weights = vector<vector<float>>(1, vector<float>(32));
    for (auto &weights : fc1_weights) {
        for (auto &weight : weights) {
            weight = static_cast<float>(rand()) / RAND_MAX *
0.01f; // Уменьшаем дисперсию инициализации
        }
    }

    fc2_weights = vector<vector<float>>(32, vector<float>(64));
    for (auto &weights : fc2_weights) {
        for (auto &weight : weights) {
            weight = static_cast<float>(rand()) / RAND_MAX *
0.01f;
        }
    }

    fc3_weights = vector<vector<float>>(64, vector<float>(32));
    for (auto &weights : fc3_weights) {
        for (auto &weight : weights) {
            weight = static_cast<float>(rand()) / RAND_MAX *
0.01f;
        }
    }

    fc4_weights = vector<vector<float>>(32, vector<float>(1));
    for (auto &weights : fc4_weights) {
        for (auto &weight : weights) {
            weight = static_cast<float>(rand()) / RAND_MAX *
0.01f;
        }
    }

    fc1_biases = vector<float>(32, 0.0f);
    fc2_biases = vector<float>(64, 0.0f);
    fc3_biases = vector<float>(32, 0.0f);
    fc4_biases = vector<float>(1, 0.0f);
}

vector<float> Model::leaky_relu(const vector<float>& x, float
alpha) {
    vector<float> result(x.size());
    for (size_t i = 0; i < x.size(); ++i) {
        result[i] = x[i] > 0 ? x[i] : alpha * x[i];
    }
    return result;
}
```

```
vector<float> Model::linear(const vector<float>& x, const
vector<vector<float>>& weights, const vector<float>& biases) {
    vector<float> result(biases.size());
    for (size_t i = 0; i < biases.size(); ++i) {
        result[i] = biases[i];
        for (size_t j = 0; j < x.size(); ++j) {
            result[i] += x[j] * weights[j][i];
        }
    }
    return result;
}

// Прямой проход
vector<float> Model::forward(const vector<float>& x) {
    vector<float> out = leaky_relu(linear(x, fc1_weights,
fc1_biases));
    out = leaky_relu(linear(out, fc2_weights, fc2_biases));
    out = leaky_relu(linear(out, fc3_weights, fc3_biases));
    out = linear(out, fc4_weights, fc4_biases);
    return out;
}

neuro::neuro(const string& path) : path(path) {
    setlocale(LC_ALL, "");
    // Загрузка данных из CSV
    vector<pair<float, float>> data;
    ifstream file(path);
    if (file.is_open()) {
        string line;
        getline(file, line); // Пропускаем заголовок
        while (getline(file, line)) {
            istringstream iss(line);
            size_t buf = line.find(',');
            string date = line.substr(0, buf);
            buf = line.find(',',buf+1);
            string sales = line.substr(buf + 1, line.find(',',
buf) - buf - 1);
            cout<<sales;
            float sal = stof(sales);
            float time = c2c(date);
            data.push_back({time, sal});
            cout << data.back().first << ", " <<
data.back().second << endl;
        }
        file.close();
        last_day = int(data.back().first);
    } else {
        cerr << "Unable to open file" << endl;
    }

    // Подготовка данных для обучения
    vector<float> inputs(data.size());
    vector<float> targets(data.size());
```

```
    for (size_t i = 0; i < data.size(); ++i) {
        inputs[i] = data[i].first;
        targets[i] = data[i].second;
    }

    // Нормализация данных
    float inputs_sum = accumulate(inputs.begin(), inputs.end(),
0.0f);
    float targets_sum = accumulate(targets.begin(),
targets.end(), 0.0f);
    inputs_mean = inputs_sum / inputs.size();
    targets_mean = targets_sum / targets.size();

    float inputs_sq_sum = 0.0f;
    float targets_sq_sum = 0.0f;
    for (size_t i = 0; i < inputs.size(); ++i) {
        inputs_sq_sum += pow(inputs[i] - inputs_mean, 2);
        targets_sq_sum += pow(targets[i] - targets_mean, 2);
    }
    inputs_std = sqrt(inputs_sq_sum / inputs.size());
    targets_std = sqrt(targets_sq_sum / targets.size());

    for (auto& input : inputs) {
        input = (input - inputs_mean) / inputs_std;
    }
    for (auto& target : targets) {
        target = (target - targets_mean) / targets_std;
    }

    // Обучение модели
    const size_t num_epochs = 10000;
    float learning_rate = 0.001f;  // Уменьшаем начальную
скорость обучения
    for (size_t epoch = 0; epoch < num_epochs; ++epoch) {
        vector<float> predictions(data.size());
        for (size_t i = 0; i < data.size(); ++i) {
            predictions[i] = model.forward({inputs[i]})[0];
        }

        // Вычисление функции потерь
        float loss_value = 0.0f;
        for (size_t i = 0; i < targets.size(); ++i) {
            loss_value += pow(predictions[i] - targets[i], 2);
        }
        loss_value /= targets.size();

        // Проверка на NaN
        if (isnan(loss_value)) {
            cerr << "Loss is NaN at epoch " << epoch << endl;
            break;
        }

        // Обратный проход и оптимизация (простейшая реализация)
```

```
        for (size_t i = 0; i < inputs.size(); ++i) {
            float error = predictions[i] - targets[i];
            for (auto& weight : model.fc4_weights) {
                for (auto& w : weight) {
                    w -= learning_rate * error * inputs[i];
                }
            }
            for (auto& bias : model.fc4_biases) {
                bias -= learning_rate * error;
            }
        }

        // Увеличиваем скорость обучения на каждой эпохе
        learning_rate *= 1.00001;

        // Вывод прогресса
        if ((epoch + 1) % 500 == 0) {
            cout << "Epoch [" << (epoch + 1) << "/" <<
num_epochs << "], Loss: " << loss_value << endl;
        }
    }
}

float neuro::predict(const string& current) {
    float cur = c2c(current);
    // Нормализация текущего значения
    float current_input = (cur - inputs_mean) / inputs_std;
    // Прогнозирование
    float prediction = model.forward({current_input})[0];
    // Денормализация результата
    return prediction * targets_std + targets_mean;
}

float neuro::c2c(const string& cur) {
    setlocale(LC_ALL, "");
    const int days_in_month[13] = {0, 31, 28, 31, 30, 31, 30,
31, 31, 30, 31, 30, 31};
    static map<string, int> month_map = {
        {"янв.", 1}, {"фев.", 2}, {"мар.", 3}, {"апр.", 4},
        {"мая", 5}, {"июня", 6}, {"июля", 7}, {"авг.", 8},
        {"сен.", 9}, {"окт.", 10}, {"ноя.", 11}, {"дек.", 12}
    };

    // Разбиваем строку на число и месяц
    int day;
    string month_str;
    istringstream ss(cur);
    ss >> day >> month_str;

    // Получаем порядковый номер месяца
    int month = month_map[month_str];

    // Вычисляем номер дня с начала года
```

```
    int days_since_start_of_year = 0;
    for (int m = 1; m < month; ++m) {
        // Суммируем количество дней в каждом месяце до
указанного месяца
        days_since_start_of_year += days_in_month[m];
    }
    days_since_start_of_year += day;

    // Возвращаем номер дня с начала года
    return static_cast<float>(days_since_start_of_year);
}
std::string neuro::day_to_date(float day_of_year) {
    setlocale(LC_ALL, "");
    // Мапа для соответствия порядкового номера месяца его
названию
    std::map<int, std::string> month_map = {
        {1, "янв."}, {2, "фев."}, {3, "мар."}, {4, "апр."},
        {5, "мая"}, {6, "июня"}, {7, "июля"}, {8, "авг."},
        {9, "сен."}, {10, "окт."}, {11, "ноя."}, {12, "дек."}
    };

    // Количество дней в каждом месяце
    const int days_in_month[13] = {0, 31, 28, 31, 30, 31, 30,
31, 31, 30, 31, 30, 31};

    int day = static_cast<int>(day_of_year);
    int month = 1;

    // Определяем месяц и день в месяце
    while (day > days_in_month[month]) {
        day -= days_in_month[month];
        month++;
    }

    return std::to_string(day) + " " + month_map[month];
}

set<pair<string, int>,StringComparator> neuro::predict(int n) {
    setlocale(LC_ALL, "");
    set<pair<string, int>,StringComparator> ret;
    for(int i=last_day+1;i<last_day+n+2;i++) {

ret.insert({dayOfYearToDate(i),round(predict(day_to_date(i))+int
((targets_std)+targets_mean))});
    }
    writeSetToCSV(ret, "predict.csv");
    return ret;
}
void neuro::create_base(){
    ifstream in("base.csv");
    ofstream of("base_look.csv");
    string line,buf;
    vector<string> base;
```

```
    bool f = true;
    while(getline(in,line)){
        if(f) {
            f=false;
            continue;
        }
       buf=line.substr(0,line.find(','));
       buf = dayOfYearToDate(int(c2c(buf)));
       line = line.substr(line.find(','),line.size()-1);
       line = buf+line;
       replace(line.begin(), line.end(), ',', ';');
       of<<line<<endl;
    }
    in.close();
    of.close();
}
```

Файл «neuro.h»

```
#ifndef NEURO_H
#define NEURO_H

#include <iostream>
#include <fstream>
#include <sstream>
#include <vector>
#include <cmath>
#include <chrono>
#include <algorithm>
#include <numeric>
#include <map>
#include <set>
#include <QtCore>
#include <sstream>
#include <iomanip>


// Определение модели нейронной сети

class Model {
public:
    Model();
    std::vector<float> forward(const std::vector<float>& x);
    std::vector<float> leaky_relu(const std::vector<float>& x,
float alpha = 0.01f);
    std::vector<float> linear(const std::vector<float>& x, const
std::vector<std::vector<float>>& weights, const
std::vector<float>& biases);

    std::vector<std::vector<float>> fc1_weights, fc2_weights,
fc3_weights, fc4_weights;
    std::vector<float> fc1_biases, fc2_biases, fc3_biases,
fc4_biases;
```

```
};
struct StringComparator {

    bool operator() (const std::pair<std::string, int>& lhs,
const std::pair<std::string, int>& rhs) const {
        using namespace std;
        string day1 = lhs.first.substr(0,2),month1 =
lhs.first.substr(3,4);
        string day2 = rhs.first.substr(0,2),month2 =
rhs.first.substr(3,4);
        if(month1==month2)
            return day1<day2;
        return month1<month2;

    }
};

class neuro {
public:
    neuro(const std::string& path);
    float predict(const std::string& current);
    float c2c(const std::string& cur);
    std::string day_to_date(float day_of_year);
    std::set<std::pair<std::string, int>,StringComparator>
predict(int n);
    void create_base();
private:
    int last_day;
    std::string path;
    Model model;
    float inputs_mean, inputs_std, targets_mean, targets_std;
};

#endif // NEURO_H
```

ПРИЛОЖЕНИЕ В

(обязательное)

Программа для обработки, хранения и визуализации информации

Презентация

Программа для обработки, хранения и визуализации информации

Автор работы: студент группы 22ВВИм1
Горбунова Вероника Владимировна
Научный руководитель: к.т.н., доцент
Федюнин Роман Николаевич

Пенза, 2024

Цель работы

Целью дипломного проекта является проектирование и разработка приложения, главная задача которого – предоставлять данные и делать прогнозирование.

Разрабатываемое программное обеспечение должно представлять собой приложение, которое собирает данные с сайта маркетплейса «Wildberries» по заданным характеристикам, производит обработку и вывод данных в виде таблиц и графиков.

Задачи приложения

Программа должна обеспечивать возможность выполнения перечисленных ниже функций:

– сбор данных о товаре с сайта маркетплейса «Wildberries» (с помощью технологии парсинга данных):

1) цена товара;
2) рейтинг товара;
3) артикул товара;
4) процент выкупов;

– отображение данных по цене, количеству продаж, сумме продаж в виде таблиц и графиков;

– прогноз количества продаж на следующие 30 дней.

2

Актуальность

Для эффективной торговли необходим конкурентный анализ. Конкурентный анализ — это процесс изучения конкурентов, выводы после которого помогут принять стратегические решения. Анализ помогает найти ключевые показатели конкурентов и сравнить их со своими.

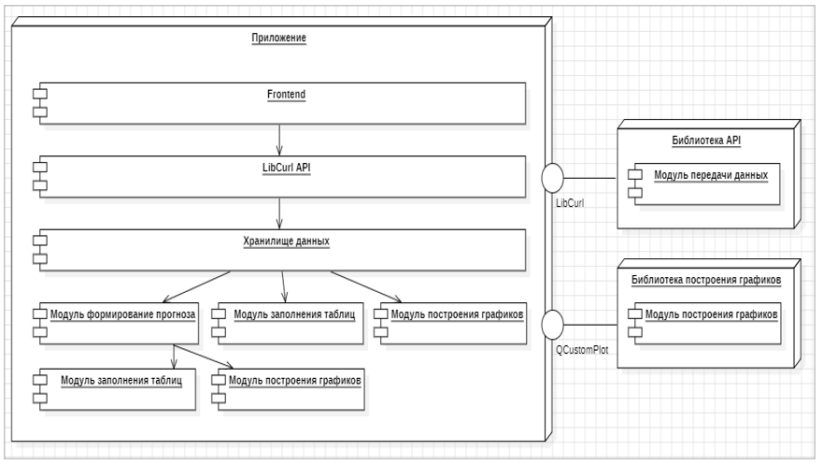
Данный проект позволит проводить анализ продаж автоматизированно, что значительно снизит временные затраты человека на обработку данных.

3

Аналоги

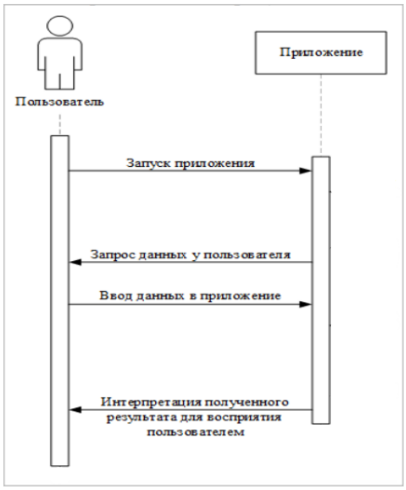
4

Архитектура проекта



5

Диаграмма последовательности



6

Структура хранения данных

```
{
    "Дата": "15.05",
    "Цена": "478",
    "Количество продаж": "24",
    "Сумма продаж": "11472"
}
```

```
{
    "Дата": "15.06",
    "Количество продаж": "31"
}
```

7

Пример .csv файла

16.май	299	22	6578
17.май	299	20	5980
18.май	299	15	4485
19.май	299	11	3289
20.май	298	26	7748
21.май	298	15	4470
22.май	298	12	3576
23.май	298	19	5662
24.май	298	18	5364
25.май	289	10	2890
26.май	298	15	4470
27.май	298	5	1490
28.май	298	11	3278

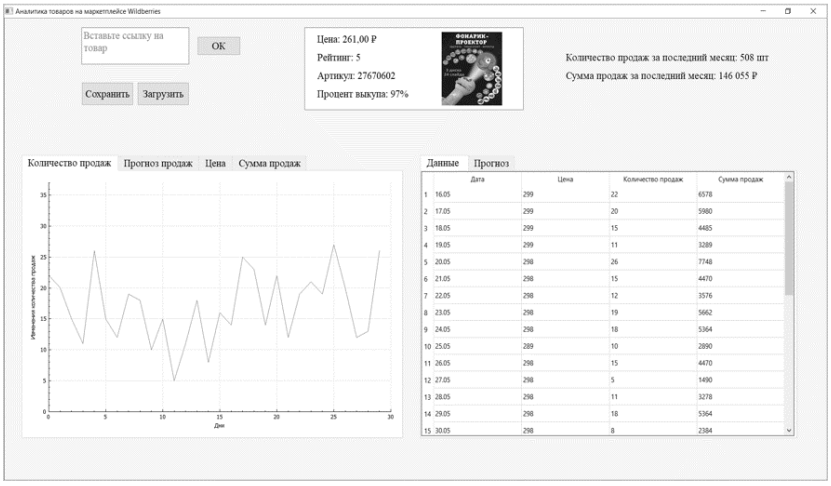
8

Тестирование

В рамках работ, описанных в этом разделе, каждый компонент итоговой программы проверен на работоспособность в ходе итераций тестирования, которые выполнялись на протяжении всего цикла разработки каждого из компонентов программы.

9

Результат работы приложения



10

Спасибо за внимание!

11