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

Кафедра «Прикладная математика и информатика»
(наименование)

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

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

# ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА (БАКАЛАВРСКАЯ РАБОТА)

на тему «Анализ больших данных в социальных сетях для выявления тенденций и прогнозирования поведения пользователей»

Обучающийся В. Д. Положенцев _____
(Инициалы Фамилия) (личная подпись)

Руководитель канд.пед.наук, доцент, О.М. Гущина
(ученая степень (при наличии), ученое звание (при наличии), Инициалы Фамилия)

Консультант канд.пед.наук, доцент, Егорова А. В.
(ученая степень (при наличии), ученое звание (при наличии), Инициалы Фамилия)

Тольятти 2024

## Аннотация

Тема бакалаврской работы: «Анализ больших данных в социальных сетях для выявления тенденций и прогнозирования поведения пользователей».

Бакалаврская работа посвящена анализу больших данных в социальной сети ВКонтакте и на основе этого выявления тенденций и прогнозирования поведения пользователей.

В ходе выполнения исследований по бакалаврской работе был проведен сравнительный анализ различных методов сбора и анализа данных, описание математической модели метода k-ближайших соседей, функционально-ориентированное проектирование системы, а также разработка и тестирование ПО.

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

Во введении прописывается актуальность темы, написаны цель и задачи. В первом разделе рассматривается предметная область исследования и производится обзор социальных сетей для анализа, а также производится анализ характеристик пользователей в социальных сетях. Во втором разделе произведен обзор существующих метрик для оценки эффективности рекламной кампании, выбран метод k-ближайших соседей для поиска заинтересованной группы пользователей и описана его математическая модель, также мы описали и создали архитектуру системы и спроектировали систему. В третьем разделе было разработано ПО и протестировано методом функционального тестирования, а также произведен анализ полученных результатов. В заключении представлены результаты выполнения бакалаврской работы.

Бакалаврская работа состоит из 55 страниц, 22 рисунков, 4 таблиц, 31 источников и 2 листингов.

## Abstract

Topic of the bachelor's thesis: “Analysis of big data in social networks to identify trends and predict user behavior.”

The bachelor's thesis is devoted to the analysis of big data on the social network VKontakte and, based on this, identifying trends and predicting user behavior.

During the bachelor's thesis, a comparative analysis of various methods of data collection and analysis, a description of the mathematical model of the k-nearest neighbors method, function-oriented system design, as well as software development and testing were carried out.

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

The first section examines the research domain and provides an overview of social networks for analysis, as well as an analysis of the characteristics of users in social networks.

In the second section, we reviewed existing metrics for assessing the effectiveness of an advertising campaign, selected the k-nearest neighbors method to search for an interested group of users and described its mathematical model, we also described and created the system architecture and designed the system.

In the third section, the software was developed and tested using the functional testing method, and the results obtained were analyzed.

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

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

The bachelor's thesis consists of 55 pages, 22 figures, 4 tables, 31 sources and 2 listings.

## Содержание

# Введение

В условиях растущего объема данных, в социальных сетях, важной задачей становится анализ их содержимого с целью выявления актуальных тенденций и прогнозирования поведения пользователей. Для успешного решения этой задачи необходимо разработать инновационные методы и алгоритмы обработки больших данных, способные обеспечить высокую точность и надежность результатов анализа. При этом, актуальность и перспективность данной темы заключается в ее потенциале для создания эффективных технологий, позволяющих прогнозировать и предсказывать пользовательское поведение с высокой точностью. Такие технологии могут быть востребованы в различных областях, таких как маркетинг, социология, политический анализ и другие.

Современное состояние исследований в области данной темы активно развивается и получает все большую актуальность. Обзор литературы по данной проблеме показывает, что исследователи прикладывают значительные усилия для разработки новых методов и алгоритмов, а также для применения современных технологий, включая машинное обучение и искусственный интеллект.

Однако, несмотря на значительный прогресс в данной области, можно сделать вывод, что тема анализа больших данных в социальных сетях для выявления тенденций и прогнозирования поведения пользователей еще не полностью раскрыта. Существует ряд нерешенных проблем, связанных как с научной теорией, так и с практическими аспектами применения этих исследований.

Во-первых, с ростом объема данных, создаваемых в социальных сетях, становится сложнее обрабатывать и анализировать эти данные в реальном времени. Возникают проблемы с масштабируемостью и эффективностью алгоритмов обработки больших данных.

Во-вторых, существует необходимость в разработке новых моделей и методов, специфических для социальных сетей, учитывающих особенности взаимодействия пользователей, контекстуальную информацию и динамику поведения.

Кроме того, существует значительная потребность в разработке этических норм и правил для работы с данными пользователей в социальных сетях, так как важно обеспечить конфиденциальность и защиту личной информации.

Объектом исследования бакалаврской работы является разработка алгоритма и математической модели на основе анализа больших данных в социальных сетях для выявления поведения пользователей и тенденций.

Предмет исследования – базы данных социальных сетей, алгоритмы, программные решения, математическая модель, библиотеки, которые будут использованы для анализа больших данных.

Цель работы разработка алгоритма и математической модели на основе анализа больших данных в социальных сетях для развития деятельности кампаний в информационном пространстве, учитывая поведение пользователей и тенденций, оценка его эффективности.

Задачи выпускной квалификационной работы включают в себя:

- Изучение существующих методов анализа больших данных в социальных сетях и их применимости для выявления тенденций и прогнозирования поведения пользователей.
- Выбор социальной сети для анализа.
- Анализ характеристик поведения пользователей в социальных сетях для выявления основных характеристик для анализа больших данных.
- Проведение исследовательского анализа математических методов для поиска наиболее заинтересованной аудитории.
- Описание математической модели выбранного метода.

– Описание архитектуры и проектирование системы для разработки программного обеспечения.
– Описание реализации программного обеспечения, функциональное тестирование ПО и анализ полученных результатов.

Данная работа включает в себя введение, три главы, заключение и список литературы.

В первом разделе рассматривается постановка задачи на работу, а также описание методов анализа больших данных. Также производится ыыбор социальной сети и характеристик поведения пользователей для анализа.

Во втором разделе, описан метод к-ближайших соседей, сделана математическая модель, описана разработка программного модуля.

В третьем разделе, описан процесс реализации Программного обеспечения, и тестирования ПО.

# 1 Описание методов анализа больших данных и характеристик поведения пользователей в социальных сетях

## 1.1 Постановка задачи и описание существующих методов анализа больших данных

Задачей данной бакалаврской работы является выявление существующих на данный момент тенденций в социальных сетях и прогнозирование поведения пользователей в них. Выявление и прогнозирование будет осуществляться на основе разработанного алгоритма анализа больших данных и компьютерной модели данного алгоритма.

Целью работы является создание эффективного и оптимизированного алгоритма анализа больших данных в социальных сетях, который помогает владельцам среднего и малого бизнеса лучше продвигать свои товары и услуги в медиапространстве на основе компьютерной модели.

Анализ больших данных - это процесс извлечения ценной информации из больших объемов данных, которые обычно невозможно обработать с использованием традиционных методов анализа данных. Существует немало методов анализа больших данных, каждый из которых применяется для решения определённой задачи. Основные методы анализа больших данных включают в себя:

- Методы обработки данных.
- Методы исследовательского анализа данных (EDA).
- Методы машинного обучения.
- Методы глубокого обучения.
- Методы обработки потоков данных.

В своей работе я буду анализировать основные показатели рекламных кампаний в социальных сетях. Для этого я буду использовать методы обработки данных и методы исследовательского анализа данных. Вторая часть работы будет включать в себя выбор наиболее предпочтительной возрастной

группы, которая с большей вероятностью сделает предзаказ услуг на основе метода к-ближайших соседей. Для этого необходимо будет использовать методы машинного обучения. Рассмотрим каждый метод по отдельности (таблица 1).

Таблица 1 – Обзор методов обработки больших данных

| Название метода | Возможности метода | Преимущества метода | Недостатки метода |
|---|---|---|---|
| Методы обработки данных | Фильтрация данных;<br>Преобразование данных;<br>Объединение данных. | Позволяют обрабатывать большие объемы данных;<br>Улучшают качество данных;<br>Помогают подготовить данные для анализа. | Требуются специализированные навыки;<br>Могут потреблять много времени и ресурсов;<br>Риск потери информации. |
| Методы исследовательского анализа данных (EDA) | Статистический анализ;<br>Визуализация данных;<br>Кластерный анализ. | Позволяют исследовать данные без предварительных гипотез;<br>Могут выявить скрытые паттерны;<br>Помогают формулировать гипотезы. | Риск неправильных выводов из-за случайных факторов;<br>Требуются большие объемы данных;<br>Временные затраты. |
| Методы машинного обучения | Классификация;<br>Регрессионный анализ;<br>Кластеризация;<br>Ассоциативные правила. | Способность выявлять сложные взаимосвязи;<br>Скорость анализа;<br>Автоматизация. | Необходимость больших объемов данных;<br>Интерпретируемость. |

Все эти подходы будут эффективны для анализа больших данных в социальных сетях. Все вышеперечисленные недостатки не являются существенными в контексте моей работы. В своей работе я буду применять фильтрацию, визуализацию и объединение данных, а также классификацию и регрессионный анализ.

## 1.2 Выбор социальной сети для анализа больших данных

Для анализа больших данных важно выбрать подходящую платформу с целью анализа поведения пользователей и выявления трендов. Многие социальные сети на данным момент заблокированы в РФ, поэтому необходимо провести сравнительный анализ. Рассмотрим самые популярные социальные сети в России на данный момент (таблица 2).

Таблица 2 – Сравнительный анализ социальных сетей

| Социальная сеть | Преимущества соц. сети | Недостатки соц. сети |
|---|---|---|
| ВКонтакте (VK) | Большая база пользователей;<br>Много функций и соответственно мест для рекламы;<br>Много групп и сообществ по интересам. | Проблемы с конфиденциальностью данных пользователей;<br>Низкое качество контента;<br>Проблемы с цензурой. |
| Одноклассники | Фокус на аудитории старшего возраста;<br>Много функций для общения. | Низкая популярность среди молодежи;<br>Ограниченные возможности для рекламы. |
| Телеграм | Телеграм славится своей конфиденциальностью и шифрованием сообщений;<br>Богатый набор инструментов для разработки ботов и интеграцию с другими сервисами. | Для бизнеса может быть сложно монетизировать активность в Телеграме из-за ограниченных возможностей рекламы и монетизации контента. |
| ТикТок | ТикТок пользуется огромной популярностью среди молодежи;<br>Многие компании используют ТикТок для продвижения своих товаров и услуг, особенно если их аудитория молодежь. | Из-за высокой конкуренции в ТикТоке может быть сложно выделиться среди множества контента;<br>Ограниченные возможности для создания и распространения контента по сравнению с другими социальными сетями. |
| YouTube | Является крупнейшей видеоплатформой в мире, предоставляя бесконечные возможности для просмотра и создания контента;<br>Разнообразие контента. | Охват конкретной целевой аудитории может быть ограниченным из-за специфики платформы и алгоритмов рекомендаций;<br>Проблемы монетизации. |

Такие социальные сети как Instagram, Facebook и Twiter не учитывались

при сравнительном анализе, потому что они заблокированы на территории РФ и почти все никогда не были популярны в России. Рассмотрим сравнение среднесуточного охвата выбранных социальных сетей для выбора наиболее широкой и многочисленной аудитории для анализа больших данных и продвижения (рисунок 1).


Рисунок 1 – Охват самых популярных социальных сетей в России

В виду того, что некоторые социальные сети были заблокированы в РФ, стоит остановиться на доступных социальных сетях. В виду того, что средний возраст пользователей Одноклассников и TikTok не подходит для эффективного продвижения в социальных сетях продукции кампании, в которой я прохожу преддипломную практику, то они нам не подойдут. YouTube является видеохостингом, а телеграмм является мессенджером, что усложняет анализ их аудитории и не подходит в рамках поставленной задачи.

Учитывая всё вышесказанное, наиболее подходящим вариантом для анализа больших данных в социальных сетях в рамках поставленной задачи будет ВКонтатке.

## 1.3 Анализ характеристик, необходимых для прогнозирования поведения пользователей и выявления тенденций

В своей работе я буду производить анализ полученных статистических данных из кабинета ВКонтакте для того, чтобы спрогнозировать как пользователи отреагируют на то или иное объявление и какие пользователи наиболее предпочтительны для продвижения. В виду этого необходимо узнать какие основные характеристики поведения пользователей в социальных сетях существуют. Рассмотрим некоторые из них:

Пользователи социальных сетей активно общаются друг с другом, обмениваются мнениями, идеями и новостями [1]. Они создают и поддерживают социальные связи, общаются с друзьями, семьей и коллегами [2].

Пользователи регулярно обновляют свои профили, публикуют новости, фотографии и видеоролики. Также стоит отметить, что они просматривают и изучают контент, созданный другими пользователями, такие как статьи, видео, фотографии и т.д [3].

Пользователи используют социальные сети для поиска информации о товарах, услугах, событиях и на основе этого присоединяются к группам и сообществам, основанным на общих интересах и интересах [4].

Пользователи стремятся увеличить свою активность в социальных сетях, набирая новых друзей и подписчиков, получая больше лайков, комментариев и репостов [5].

Некоторые пользователи стремятся увеличить свою экспертизу в определенной области, публикуя контент, связанный с этой темой, а также стремятся увеличить свою нацеленность на определенную аудиторию, публикуя контент, ориентированный на эту аудиторию [6].

Однако каждый пользователь имеет свою уникальную активность в социальных сетях. Другими словами, не все выкладывают контент или просматривают новости, а например используют ВКонтакте только для

общения. Соответственно, необходимо выявить факторы, которые влияют на поведение пользователей в социальных сетях. К основным фактором относятся:

- Личностные качества.
- Социальные факторы.
- Цели пользователей.
- Особенности социальных сетей.
- Платформа, которую использует пользователь
- Правила и ограничения, установленные социальными сетями

Когда речь заходит об анализе больших данных все эти факторы точно отследить невозможно, в виду того, что ВКонтакте не предоставляет полной статистики взаимодействий пользователей. Однако, для выявления групп пользователей с примерно похожими характеристиками поведения в социальных сетях отлично подойдёт анализ по возрастным группам и полу.

Выводы по разделу 1

В первом разделе была поставлена задача на исследование, включающая в себя анализ больших данных в социальных сетях, на основе которого будет создано ПО. Оно поможет владельцам малого и среднего бизнеса отслеживать параметры продвижения в медиапространстве, а также поведение и реакции пользователей. Это необходимо для выбора наиболее подходящей группы пользователей для продвижения в информационном пространстве.

Также были произведен сравнительный анализ социальных сетей для анализа больших данных. В ходе анализа наиболее подходящей оказалась социальная сеть ВКонтакте.

В итоге были проанализированы характеристики поведения пользователей, которые можно выявить при анализе больших данных и было выявлено, что самыми важными являются пол и возраст пользователей.

# 2 Описание основных показателей рекламной кампании и метода к-ближайших соседей

## 2.1 Анализ основных показателей рекламной кампании

Организация, в которой я прохожу преддипломную практику, поставила передо мной задачу проанализировать эффективность внешних рекламных кампаний.

Эффективность рекламной кампании - это способность рекламного мероприятия достигать поставленных целей с минимальными затратами. Она измеряется в том, насколько успешно кампания привлекает внимание целевой аудитории, увеличивает узнаваемость бренда, стимулирует продажи или достигает других конкретных метрик, определяемых бизнес-целями [7].

Оценка эффективности рекламной кампании позволяет бизнесу понять, насколько успешно были использованы рекламные ресурсы, и проанализировать, какие аспекты кампании нужно улучшить [8]. Это также помогает оптимизировать бюджет на маркетинг и принимать обоснованные решения о том, какие стратегии и каналы коммуникации лучше всего работают для достижения поставленных целей [9].

Рекламные объявления были выложены в трёх группах ВКонтакте. Спустя пять дней внутренняя статистика в сообществах ВКонтакте предоставила статистику людей, которые перешли по ссылке с указанием их возрастной группы, пола, образования и города. Рекламодатели предоставили эту статистику в формате csv-файла (рисунок 2).

| | A | B | C | D | E | F | G |
|---|---|---|---|---|---|---|---|
| 1 | Возрастная группа | Город | Пол | Образование | Дата перехода | Семейное положение | Наличие детей |
| 2 | 42-48 | Челябинск | Ж | - | 12.02.2024 | холост | нет |
| 3 | 21-27 | Екатиринбург | Ж | Среднее профессиональное | 13.02.2024 | в активном поиске | нет |
| 4 | 42-48 | Челябинск | Ж | Среднее профессиональное | 14.02.2024 | женат | да |
| 5 | до 20 | Екатиринбург | М | Среднее профессиональное | 15.02.2024 | холост | нет |

Рисунок 2 – Пример статистического файла для анализа

На основе полученных данных необходимо применить методы фильтрации, объединения и визуализации данных для оценки эффективности рекламной кампании. Как мы уже выяснили, наиболее эффективным методом прогнозирования поведения пользователей и выявления тенденций является анализ по полам и возрастным группам. Соответственно, я буду использовать именно эти метрики для анализа. Остальные параметры не являются столь важными.

Для анализа эффективности рекламной кампании существует множество метрик, которые рассчитываются по формулам (формулы 1–6) [10]. Рассмотрим основные метрики в виде таблицы (таблица 3).

Таблица 3 – Обзор метрик анализа эффективности рекламной кампании

| Название метрики | Формула метрики | Определение метрики |
|---|---|---|
| ROI (Return on Investment). | $\frac{\text{Прибыль} - \text{Затраты}}{\text{Затраты}} * 100\%$ (1) | показывает, сколько денег компания заработала или потеряла относительно вложенных в рекламу средств [11]. |
| ROAS (Return on Advertising Spend). | $\frac{\text{Выручка от рекламы}}{\text{Затраты на рекламу}}$ (2) | измеряет, сколько денег компания заработала от рекламы [12]. |
| CTR (Click-Through Rate). | $\frac{\text{Количество кликов}}{\text{Количество показов}} * 100\%$ (3) | процент пользователей, которые кликнули на рекламу после ее просмотра [13]. |
| CPC (Cost Per Click). | $\frac{\text{Затраты на рекламу}}{\text{Количество кликов}}$ (4) | средняя стоимость одного клика на рекламу [14]. |
| CPA (Cost Per Acquisition). | $\frac{\text{Затраты на рекламу}}{\text{Количество действий}}$ (5) | средние затраты на привлечение одного нового клиента или выполнение желаемого действия [15]. |
| CPM (Cost Per Mille). | $\frac{\text{Затраты на рекламу}}{\text{Количество показов}} * 1000$ (6) | стоимость 1000 показов [16]. |

Самыми важными параметрами являются CTR, CPC и CPM. Это те параметры, по которым можно сказать, насколько рекламная кампания

показала свою эффективность [17].

Количество кликов будет определяться по количеству строк в csv файле со статистикой переходов, количество просмотров во ВКонтакте можно узнать в правом нижнем углу под каждой записью, а затраты на рекламную кампанию будут уточнены у руководителя практики.

Для визуализации этих параметров будет использоваться графический интерфейс библиотеки tkinter из набора стандартных библиотек Python.

Этот графический интерфейс был выбран мною ввиду его ряда преимуществ. Рассмотрим эти преимущества:

- Простота использования.
- Встроенная в Python.
- Переносимость.
- Обширная документация и сообщество.
- Мощный виджетный набор.
- Простая интеграция с другими библиотеками.

Для оценки эффективности рекламной кампании будут строиться два графика разных цветов для оценки женской и мужской аудитории с замерами CTR и CPC по возрастным группам. Также будут выведены CTR, CPC и CPM всей рекламной кампании. Данные для анализа будут извлекаться из csv файлов со статистикой переходов.

## 2.2 Выбор метода для поиска заинтересованной аудитории

Вторая часть моей бакалаврской работы будет заключаться в оценке аудитории, которая перешла по ссылке из рекламного объявления, с целью выявления наиболее заинтересованной группы пользователей, которые оформили предзаказ на услуги кампании.

Для этого мы будем использовать математический метод, который наиболее подходящий для поставленной задачи. Проведём сравнительный анализ нескольких математических методов, которые могут помочь

определить наиболее заинтересованную группы пользователей (таблица 4).

Таблица 4 – Сравнительный анализ математических методов для определения наиболее предпочтительной группы пользователей

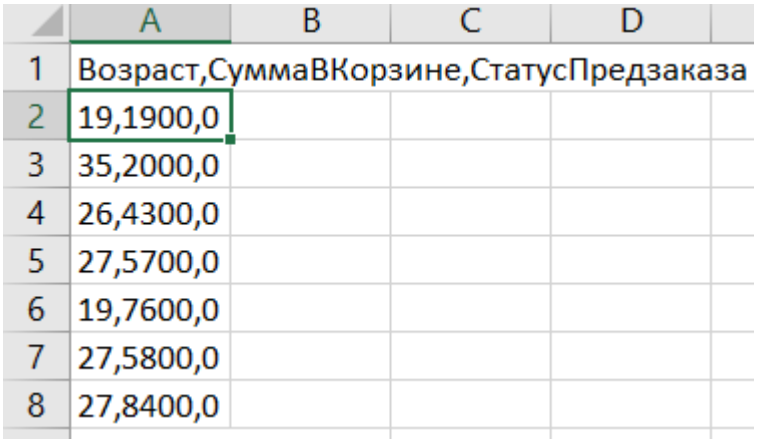
| Название метода | Принцип работы | Преимущества метода | Недостатки метода |
|---|---|---|---|
| Метод k-ближайших соседей. | Оценивает близость объекта к другим объектам в пространстве признаков и определяет принадлежность к определенному классу на основе классов его ближайших соседей [18]. | Простота реализации и понимания;<br>Хорошо работает, когда данные имеют простую структуру и нелинейные зависимости. | Требует настройки параметра k;<br>Вычислительно затратен при большом объеме данных. |
| Логистическая регрессия. | Модель, которая используется для оценки вероятности принадлежности объекта к определенному классу на основе линейной комбинации его признаков [19]. | Хорошо интерпретируема;<br>Эффективен при работе с линейно разделимыми данными. | Не способен улавливать нелинейные зависимости между признаками;<br>Требует предположения о линейности зависимости. |
| Деревья принятия решений. | Структура данных, которая разбивает набор данных на более мелкие группы на основе значений признаков [20]. | Могут обрабатывать как числовые, так и категориальные данные;<br>Не требуют масштабирования признаков. | Склонны к переобучению, особенно при большой глубине дерева;<br>Неустойчивы к изменениям в данных. |

На основе сравнительного анализа можно сделать вывод, что для поиска наиболее заинтересованной группы пользователей лучше всего подойдет метод k-ближайших соседей, потому что он проще в реализации, визуализации результатов и в понимании.

В своей работе я буду использовать метод k-ближайших соседей для анализа наиболее предпочтительной группы пользователей, которые перешли по ссылке из рекламных объявлений и оформили предзаказ. Выявление этой группы пользователей необходимо для наиболее эффективной настройки и

отладки внутренней таргетированной рекламы ВКонтакте с наименьшими затратами на продвижение.

На вход будут поступать значения из csv-файла, который был загружен из статистики на сайте кампании, в которой я прохожу преддипломную практику. В нём будет возраст пользователей, сумма предзаказа в корзине и статус оформления предзаказа, где 0 – оформили предзаказ, а 1 – не оформили. Пример данных из такого файла представлены на рисунке (рисунок 3).



| | A | B | C | D |
|---|---|---|---|---|
| 1 | Возраст,СуммаВКорзине,СтатусПредзаказа | | | |
| 2 | 19,1900,0 | | | |
| 3 | 35,2000,0 | | | |
| 4 | 26,4300,0 | | | |
| 5 | 27,5700,0 | | | |
| 6 | 19,7600,0 | | | |
| 7 | 27,5800,0 | | | |
| 8 | 27,8400,0 | | | |

Рисунок 3 – Данные из файла для построения графика метода k-ближайших соседей

На основе этих параметров будет строиться график метода k-ближайших соседей для определения наиболее предпочтительной группы пользователей для продвижения. Будет использоваться разделение всех пользователей на предпочтительную и непредпочтительную группу с небольшой погрешностью.

В дальнейшем полученную информацию можно будет использовать для таргетированной рекламы по наиболее активной и заинтересованной группе пользователей.

## 2.3 Математическая модель метода к-ближайших соседей

Метод k-ближайших соседей - это простой и интуитивно понятный алгоритм машинного обучения, который используется для классификации и регрессии. Его история уходит в прошлое, и он является одним из первых алгоритмов, который был применен в области паттерн-распознавания и классификации [22].

Идея метода k-ближайших соседей была предложена еще в 1950-х годах. Однако первое упоминание о нем в литературе относится к 1967 году, когда американские ученые Эдвард Харт и Фиона Роу применили этот метод для классификации различных растительных и геологических образцов [22]. Они использовали метод k-ближайших соседей для определения типа породы на основе ее химических свойств. Одна из основных причин использования этого метода - его простота, а также он не требует фазы обучения, что делает его привлекательным для простых задач классификации и регрессии [23].

Пусть у нас есть набор данных, состоящий из N объектов, каждый из которых описывается n признаками [24]. Обозначим этот набор данных D в виде общей формулы (формула 7).

$$\mathrm{D} = \{(x_1, y_1), (x_2, y_2), \dots, (x_N, y_N)\} \quad (7)$$

где:

$x_i$ - признаковое описание i-го объекта,

$y_i$ - его соответствующее значение целевой переменной.

В своей работе ВКР я буду использовать данные из отчёта, представленного на рисунке 3. В нем в качестве матрицы признаков используется возраст пользователей и сумма предзаказа, а в качестве вектора меток используется статус предзаказа. Соответственно, набор данных из формулы 7 будет представлен в виде матрицы X и вектора метода Y (формулы 8–9).

$$X = \begin{pmatrix} x_{11} & x_{12} \\ x_{21} & x_{22} \\ \cdots & \cdots \\ x_{i1} & x_{i2} \end{pmatrix} \quad (8)$$

$$Y = \begin{pmatrix} y_1 \\ y_2 \\ \cdots \\ y_i \end{pmatrix} \quad (9)$$

где:

$x_{ij}$ – значение j-ого признака (возраст и сумма предзаказа) для i-ого пользователя.

$y_i$ – метка класса (0 – не оформлен предзаказ, 1 – предзаказ оформлен) для i-ого пользователя.

Затем производится масштабирование признаков, которое представляет собой процесс преобразования значений признаков данных таким образом, чтобы они находились в определённом диапазоне. Это часто делается для улучшения работы алгоритмов машинного обучения [25].

В нашем случае для масштабирования признаков будет использоваться следующая формула (формула 10).

$$x'_{ij} = \frac{x_{ij} - \mu_j}{\sigma_j} \quad (10)$$

где:

$x_{ij}$ – набор данных в матрице X

$\mu_j$ – среднее значение признака j из матрицы X

$\sigma_j$ – стандартное отклонение признака j из матрицы X

Предполагается, что целевая переменная является дискретной (в задаче классификации) или непрерывной (в задаче регрессии) [26].

Параметр k представляет собой количество соседей, которые будут

использованы для классификации или регрессии нового объекта. Выбор оптимального значения k может влиять на качество модели. Параметр k обычно выбирается эмпирически или с помощью кросс-валидации. Оптимальное значение k может зависеть от конкретной задачи и набора данных [27].

После выбора параметра k необходимо определить метрику расстояния. Это функция, которая определяет расстояние между двумя точками в пространстве признаков. Наиболее распространенные метрики включают евклидово расстояние, манхэттенское расстояние и расстояние Минковского (формулы 11–13).

$$d(p,q) = \sqrt{\sum_{i=1}^{n}(q_i - p_i)^2} \quad (11)$$

$$d(p,q) = \sum_{i=1}^{n}|q_i - p_i| \quad (12)$$

$$d(p,q) = (\sum_{i=1}^{n}|q_i - p_i|^p)^{\frac{1}{p}} \quad (13)$$

где:

p и q - векторы признаков объектов,

n - количество признаков,

p - параметр, задающий степень.

В ходе решения поставленной задачи будет использовано Евклидово расстояние, которое с учетом полученных из отчёта данных будет выглядеть следующим образом (формула 14).

$$d(x_{new}, x_i) = \sqrt{(x'_{new1} - x'_{i1})^2 + (x'_{new2} - x'_{i2})^2} \quad (14)$$

где:

$x'_{new1}$ и $x'_{new2}$ – значения новой точки после масштабирования признаков.

$x'_{i1}$ и $x'_{i2}$ – значения из масштабированной обучающей выборки, которая будет использоваться в переменных с индексом train.

После определения метрики расстояния мы можем использовать задачу классификации или регрессии. В задаче классификации новый объект относится к классу, который наиболее часто встречается среди его k ближайших соседей. В задаче регрессии значение целевой переменной для нового объекта вычисляется как среднее (или взвешенное среднее) значение целевой переменной его k ближайших соседей [28].

Для классификации нового объекта в модели используется принцип голосования среди его k ближайших соседей. Например, в задаче бинарной классификации, классификатор может присваивать объекту класс, который является наиболее частым среди его соседей [29].

После нахождения k ближайших соседей объекта x, можно использовать их целевые значения для предсказания на основе общей формулы для предсказания значения целевой переменной для нового объекта x (формула 15).

$$\check{y}(x) = \frac{1}{k}\sum_{i=1}^{k} y_i \quad (15)$$

где:

$\check{y}(x)$- предсказанное значение для объекта x,

yi - значения целевой переменной для k ближайших соседей объекта x.

k – количество ближайших соседей.

Некоторые вариации этого метода позволяют взвешивать голоса соседей в зависимости от расстояния до нового объекта. Это означает, что ближайшие соседи, находящиеся ближе к новому объекту, могут иметь больший вес при классификации или регрессии. Веса соседей могут быть определены на основе расстояния до нового объекта. Чем ближе сосед к новому объекту, тем больше его вес. Формула взвешенного голосования может выглядеть следующим образом (формула 16) [30].

$$\check{y} = \frac{\sum_{i=1}^{k} w_i * y_i}{\sum_{i=1}^{k} w_i} \quad (16)$$

где:

$w_i$ – вес i-ого соседа,

$y_i$ - значение целевой переменной для i-го соседа.

В некоторых случаях можно использовать функцию ядра для взвешивания соседей, основанную не только на расстоянии, но и на других характеристиках объектов. Функция ядра может учитывать степень важности признаков и форму распределения данных. Функция ядра может использоваться для взвешивания соседей, учитывая не только расстояние, но и другие характеристики объектов. Одним из примеров функции ядра является Гауссовское ядро [31].

В своей работе я буду использовать в качестве матрицы признаков возраст пользователей и сумму предзаказа (формула 8), а в качестве вектора меток буду использовать статус предзаказа (формула 9). Для масштабирования признаков будет использоваться соответствующая формула (формула 10). Определение метрики расстояния будет производиться с помощью формулы Евклидового расстояния (формула 14) в виду того, что анализ будет производиться в рамках задачи классификации. Предсказания будут производиться на основе общей формулы для предсказания значения целевой переменной для нового объекта x (формула 15).

## 2.4 Описание компонентов системы и проектирование системы для разработки программного кода

Перед написанием программного кода необходимо подробно описать будущую систему, которая будет состоять из трёх компонентов: сбора, обработки и визуализации.

Компонент сбора данных будет использовать файлы со статистикой и информацией о пользователях, которая выгружена из сайта ВКонтакте в формате csv в первой части работы. Также пользователем будет вводиться информация о количестве просмотров и стоимости рекламной кампании. Во второй части работы будет использоваться csv файл, который был сформирован на сайте кампании. После сбора данных будет происходить фильтрация данных для использования только необходимых для анализа наборов данных.

Компонент обработки обрабатывает и анализирует полученные данные. Эта информация используется в методе k-ближайших соседей для формирования предпочтительных групп пользователей для дальнейшего анализа пользователем ПО. Также она будет использоваться для построения графиков CTR и CPC по половозрастным группам.

Компонент визуализации будет отображать графики CTR и CPC по половозрастным группам, а также CTR, CPC и CPM всей рекламной кампании. Помимо этого, компонент визуализации будет выводить график на основе метода k-ближайших соседей.

На основе описания компонентов системы мы можем спроектировать систему для дальнейшей разработки.

Программный код будет состоять из двух частей: первая часть будет анализировать эффективность рекламной кампании, а вторая часть будет определять наиболее заинтересованную группу пользователей на основе метода k-ближайших соседей.

Рассмотрим начало алгоритма для дальнейшей разработки первой части программного кода (рисунок 4).

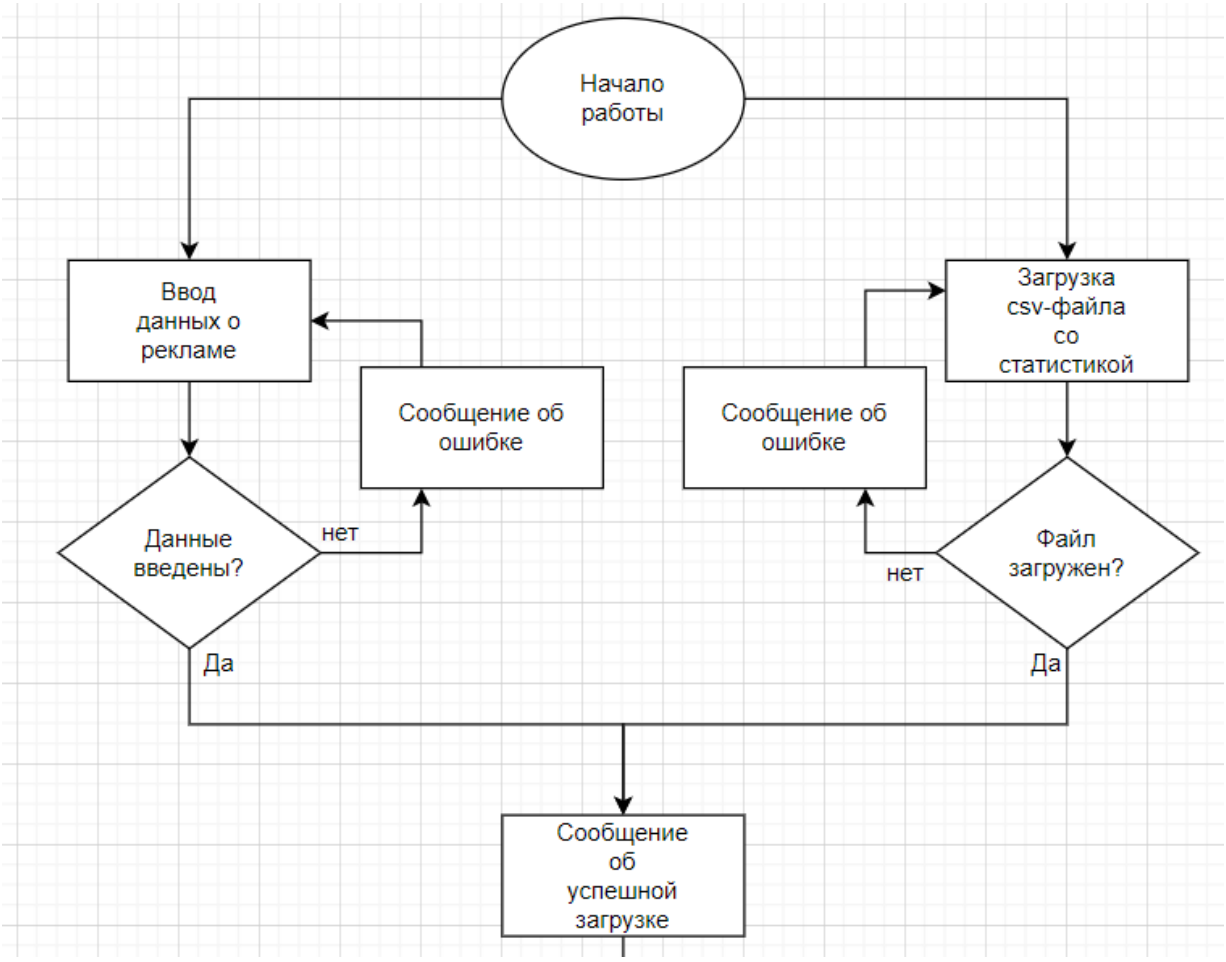



Рисунок 4 – Блок-схема, описывающая начало алгоритма для дальнейшей разработки первой части программного кода

В начале работы первой части программного кода пользователь будет вводить данные о рекламной кампании и загружать csv-файл со статистикой переходов с помощью проводника. Программа будет проверять правильность введенных данных и наличие csv-файла, а также выводить сообщение об успешной загрузке или ошибки в случае неудачи.

Рассмотрим завершение первой части программного кода (рисунок 5).

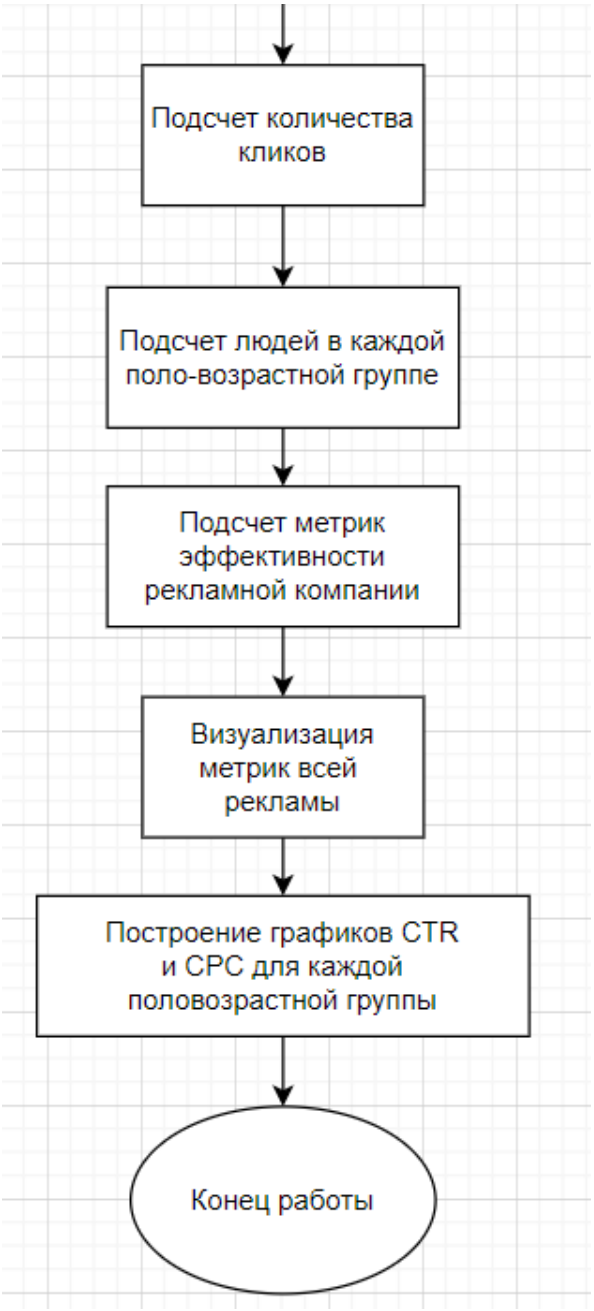



Рисунок 5 – Блок-схема, описывающая завершение алгоритма для дальнейшей разработки первой части программного кода

После сообщения об успешной загрузки ПО начинает свою работу и сперва подсчитывает количество кликов по количеству строк в csv-файле, а также подсчёт количества людей в каждой половозрастной группе. После чего происходит подсчёт CTR, CPC и CPM всей рекламной кампании для оценки

её эффективности в целом. Затем строятся графики CTR и CPC для каждой половозрастной группы по отдельности.

Во второй части программного кода будет применяться метод k-ближайших соседей для оценки наиболее заинтересованной группы пользователей. Разработаем алгоритм для дальнейшей разработки второй части кода (рисунок 6).

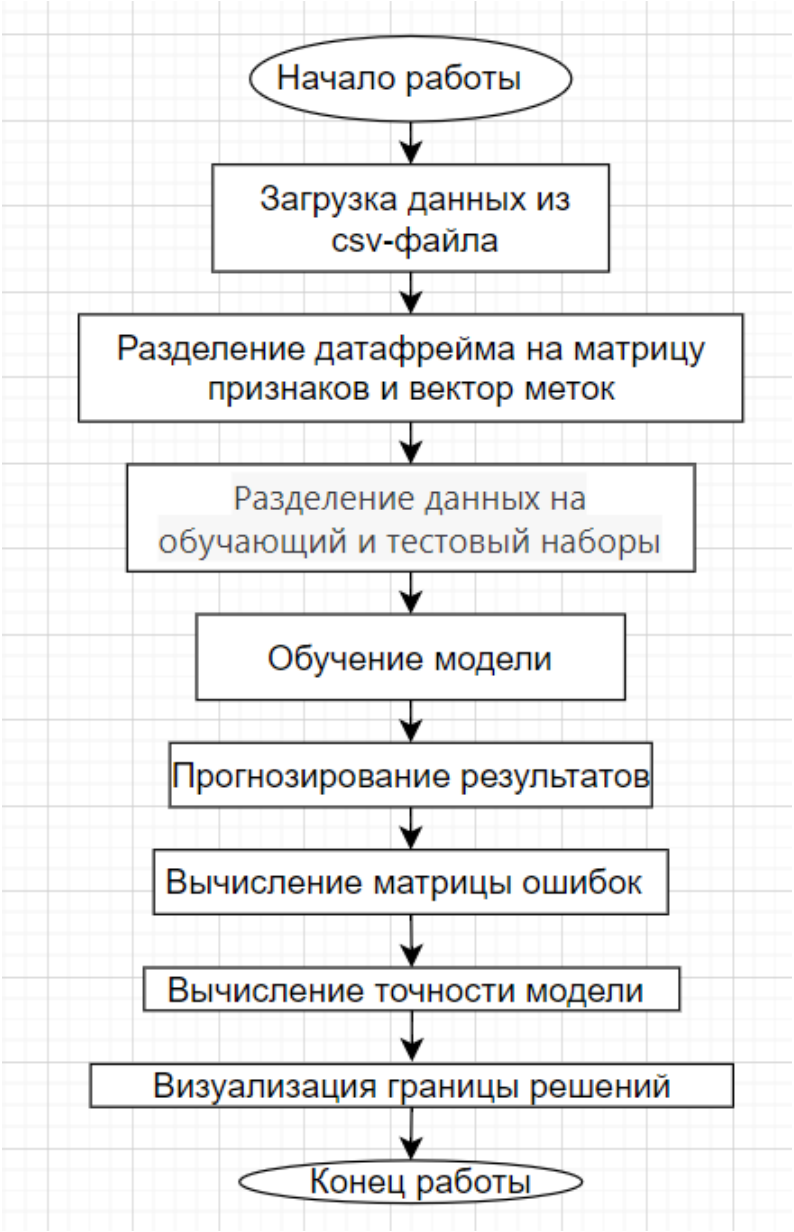



Рисунок 6 – Блок-схема, описывающая алгоритм для дальнейшей разработки второй части программного кода

Во второй части программного кода в первую очередь импортируются csv-файл со статистикой переходов в корзину. После этого датафрейм разделяется на матрицу признаков и вектор меток, а данные разделяются на тестовый и обучающий наборы. Затем модель обучается, и мы прогнозируем результаты с помощью функции predict. В конце вычисляется матрица ошибок и точность модели. После чего происходит визуализация результатов.

Выводы по разделу 2

Во втором разделе ВКР был произведен анализ основных показателей рекламной кампании для выявления самых важных метрик для оценки эффективности рекламной кампании. Также был подробно описан и проанализирован метод k-ближайших соседей, который будет оценивать наиболее предпочтительную группу пользователей.

Система была описана по компонентам сбора, анализа и визуализации данных, а также была спроектирована система для дальнейшей разработки программного обеспечения.

# 3 Реализация и тестирование программного обеспечения

## 3.1 Реализация программного обеспечения

Программный код будет написан на основе спроектированной системы для разработки на языке Python. В качестве среды разработки была выбрана программа PyCharm, которая предоставляет широкий функционал по работе с Python. Свою бакалаврскую работу я разделил на две подпрограммы. В первой происходит оценка эффективности рекламной кампании и вывод графиков CTR и CPC по половозрастным группам пользователей. Во второй программе используется метод k-ближайших соседей для определения наиболее предпочтительной группы пользователей. Обе программы были написаны на языке Python.

Начнём с первой работы в ней я использовал следующие библиотеки:

– Tkinter для создания графического пользовательского интерфейса.
– Pandas для обработки и анализа данных.
– matplotlib.pyplot для визуализации данных в виде графиков.
– seaborn для визуализации данных, основанная на matplotlib.

Объявление всех необходимых библиотек представлено ниже (рисунок 7).

```
1  import tkinter as tk
2  from tkinter import filedialog, messagebox
3  from tkinter import font
4  import pandas as pd
5  import matplotlib.pyplot as plt
6  from matplotlib.backends.backend_tkagg import FigureCanvasTkAgg
7  import seaborn as sns
```

Рисунок 7 – Объявление всех необходимых библиотек в первой подпрограмме

Для расчёта всех необходимых метрик я написал функцию calculate_metrics, которая принимает на вход путь к csv-файлу, количество просмотров и стоимость рекламы, введенные в консоль. Это основная функция программы и у неё широкий функционал. Рассмотрим его по отдельности.

Сначала производится чтение файла, в котором считается количество мужчин и женщин в каждой возрастной группе. Далее считается количество строк методом shape для определения количества кликов. После чего создаётся интерфейс для вывода параметров эффективности рекламной кампании. Результат представлен ниже (рисунок 8).

```
10  def calculate_metrics(filepath, views, CostAds):
11      df = pd.read_csv(filepath, encoding="UTF-8", sep=";")
12      male_20_count = df.loc[(df['Пол'] == 'М') & (df['Возрастная группа'] == 'до 20')].shape[0]
13      male_21_27_count = df.loc[(df['Пол'] == 'М') & (df['Возрастная группа'] == '21-27')].shape[0]
14      male_28_35_count = df.loc[(df['Пол'] == 'М') & (df['Возрастная группа'] == '28-35')].shape[0]
15      male_36_41_count = df.loc[(df['Пол'] == 'М') & (df['Возрастная группа'] == '36-41')].shape[0]
16      male_42_48_count = df.loc[(df['Пол'] == 'М') & (df['Возрастная группа'] == '42-48')].shape[0]
17      male_49_55_count = df.loc[(df['Пол'] == 'М') & (df['Возрастная группа'] == '49-55')].shape[0]
18      male_56_count = df.loc[(df['Пол'] == 'М') & (df['Возрастная группа'] == '56+')].shape[0]
19
20      female_20_count = df.loc[(df['Пол'] == 'Ж') & (df['Возрастная группа'] == 'до 20')].shape[0]
21      female_21_27_count = df.loc[(df['Пол'] == 'Ж') & (df['Возрастная группа'] == '21-27')].shape[0]
22      female_28_35_count = df.loc[(df['Пол'] == 'Ж') & (df['Возрастная группа'] == '28-35')].shape[0]
23      female_36_41_count = df.loc[(df['Пол'] == 'Ж') & (df['Возрастная группа'] == '36-41')].shape[0]
24      female_42_48_count = df.loc[(df['Пол'] == 'Ж') & (df['Возрастная группа'] == '42-48')].shape[0]
25      female_49_55_count = df.loc[(df['Пол'] == 'Ж') & (df['Возрастная группа'] == '49-55')].shape[0]
26      female_56_count = df.loc[(df['Пол'] == 'Ж') & (df['Возрастная группа'] == '56+')].shape[0]
27
28      clicks = df.shape[0]
29
30      # Создаем окно
31      root = tk.Tk()
```

Рисунок 8 – Объявление функции calculate_metrics

После этого прописывается верстка для окна интерфейса. Затем вводятся значения в консоль о количестве просмотров и стоимости анализируемой рекламной кампании (рисунок 9). Также считаются и выводятся все необходимые метрики для оценки эффективности рекламной кампании, которые описаны выше. Производится подсчёт CTR и CPC для

каждой половозрастной группы с проверкой на то, равно ли значению нулю для предотвращения возможных ошибок (рисунок 10).

```
67
68      print("Количество кликов:", clicks)
69      print("Общий CTR РК:", round((clicks / views * 100), 3))
70      print("Общий CPC РК:", round((CostAds / clicks), 3))
71      print("Общий CPM РК:", round((CostAds / views * 1000), 3))
72
73      age_groups = ['до 20', '21-27', '28-35', '36-41', '42-48', '49-55', '56+']
74
```

Рисунок 9 – Запись в консоль необходимых параметров

```
75      # Подсчет CTR для мужчин и женщин в каждой возрастной группе
76      male_ctr = [(male_20_count / views * 100), (male_21_27_count / views * 100),
77                  (male_28_35_count / views * 100), (male_36_41_count / views * 100),
78                  (male_42_48_count / views * 100), (male_49_55_count / views * 100),
79                  (male_56_count / views * 100)]
80      female_ctr = [(female_20_count / views * 100), (female_21_27_count / views * 100),
81                    (female_28_35_count / views * 100), (female_36_41_count / views * 100),
82                    (female_42_48_count / views * 100), (female_49_55_count / views * 100),
83                    (female_56_count / views * 100)]
84
85      # Подсчет CPC для мужчин и женщин в каждой возрастной группе
86      male_cpc = [(CostAds / male_20_count if male_20_count > 0 else 0),
87                  (CostAds / male_21_27_count if male_21_27_count > 0 else 0),
88                  (CostAds / male_28_35_count if male_28_35_count > 0 else 0),
89                  (CostAds / male_36_41_count if male_36_41_count > 0 else 0),
90                  (CostAds / male_42_48_count if male_42_48_count > 0 else 0),
91                  (CostAds / male_49_55_count if male_49_55_count > 0 else 0),
92                  (CostAds / male_56_count if male_56_count > 0 else 0)]
93
94      female_cpc = [(CostAds / female_20_count if female_20_count > 0 else 0),
95                    (CostAds / female_21_27_count if female_21_27_count > 0 else 0),
96                    (CostAds / female_28_35_count if female_28_35_count > 0 else 0),
97                    (CostAds / female_36_41_count if female_36_41_count > 0 else 0),
98                    (CostAds / female_42_48_count if female_42_48_count > 0 else 0),
99                    (CostAds / female_49_55_count if female_49_55_count > 0 else 0),
100                   (CostAds / female_56_count if female_56_count > 0 else 0)]
101
102     sns.set()
```

Рисунок 10 – Расчет CTR и CPC для каждой половозрастной группы

Затем происходит визуализация полученных значений в виде графиков CTR и CPC, а также они встраиваются внутрь интерфейса tkinter, чтобы не открывалось дополнительное окно. Результат представлен на рисунке ниже

(рисунок 11).

```
    # График для CTR
    axs[0].plot(age_groups, male_ctr, label='Мужчины', color='blue', marker='o', linestyle='-', linewidth=2)
    axs[0].plot(age_groups, female_ctr, label='Женщины', color='pink', marker='s', linestyle='--', linewidth=2)
    axs[0].set_xlabel('Возрастная группа', fontsize=14)
    axs[0].set_ylabel('CTR (%)', fontsize=14)
    axs[0].set_title('CTR по возрастным группам', fontsize=16)
    axs[0].grid(True)
    axs[0].legend(fontsize=12)
    axs[0].tick_params(axis='both', which='major', labelsize=12)

    # График для CPC
    axs[1].plot(age_groups, male_cpc, label='Мужчины', color='blue', marker='o', linestyle='-', linewidth=2)
    axs[1].plot(age_groups, female_cpc, label='Женщины', color='pink', marker='s', linestyle='--', linewidth=2)
    axs[1].set_xlabel('Возрастная группа', fontsize=14)
    axs[1].set_ylabel('CPC (руб.)', fontsize=14)
    axs[1].set_title('CPC по возрастным группам', fontsize=16)
    axs[1].grid(True)
    axs[1].legend(fontsize=12)
    axs[1].tick_params(axis='both', which='major', labelsize=12)

    # Отображение графиков
    plt.tight_layout()  # Улучшение размеров графиков для лучшего взаимодействия

    # Встраивание графиков в tkinter
    canvas = FigureCanvasTkAgg(fig, master=root)
    canvas.draw()
    canvas.get_tk_widget().pack(side=tk.TOP, fill=tk.BOTH, expand=True)
```

Рисунок 11 – Визуализация графиков

Также были написаны функции для открытия диалога выбора файла, для вывода ошибок в случае, если файл не был загружен и функция для отображения графиков и показателей после выбора файла, которая проверяет загрузку файла (рисунок 12).

После всего этого создаётся окно интерфейса и прописывается обращение ко всем функциям. В листинге 1 представлен код первой подпрограммы.

Во второй подпрограмме используется метод k-ближайших соседей и для этого сначала из CSV-файла считываются данные о показателях корзины. Данные разделяются на признаки (X) и целевую переменную (y). Признаки содержатся во всех столбцах, кроме последнего, а целевая переменная находится в последнем столбце.

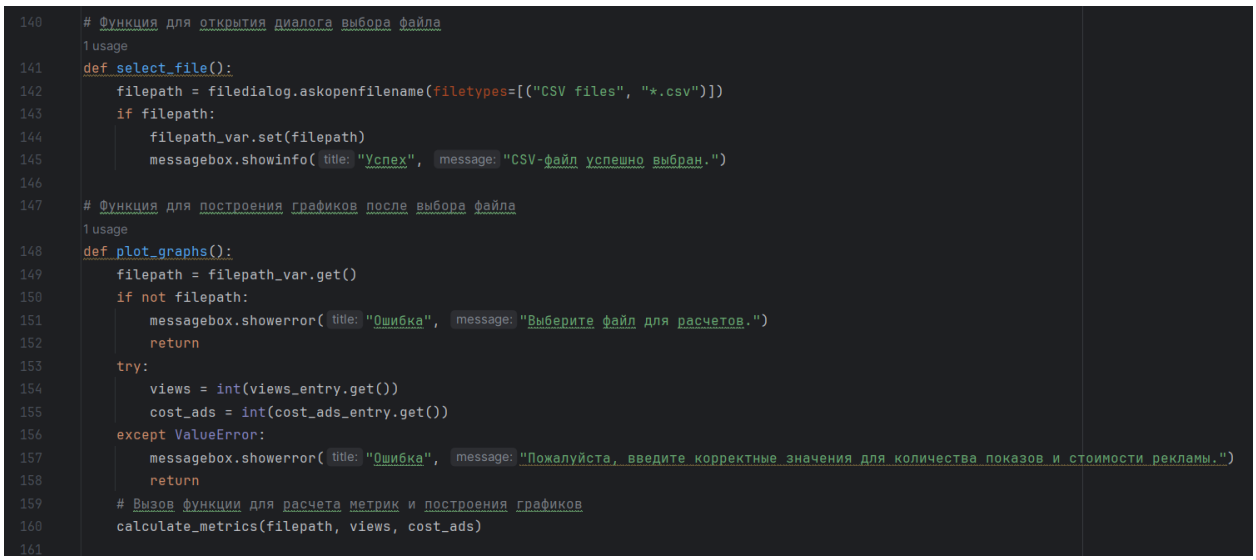



Рисунок 12 – Функции открытия диалога и проверки файла

Данные разделяются на обучающий и тестовый наборы с помощью функции train_test_split. Обучающий набор содержит 75% данных, а тестовый - 25%. Признаки масштабируются с помощью стандартизации средствами StandardScaler.

Модель классификации k-ближайших соседей инициализируется с параметрами, такими как количество соседей (5), метрика расстояния (евклидово расстояние) и степень для метрики Минковского (2). Затем модель обучается на обучающем наборе.

Модель применяется к тестовому набору для получения прогнозов. Вычисляется матрица ошибок (confusion_matrix) и точность (accuracy_score) для оценки качества модели. Ниже представлен реализация обучения тестовой выборки и предсказание значений (рисунок 13).

Результаты на тестовом наборе визуализируются на графике. Для этого используется двумерное представление данных, где каждая точка представляет собой экземпляр данных с координатами, соответствующими возрасту и сумме предзаказа. Регионы на графике закрашиваются цветом в зависимости от прогнозируемого класса, а также показываются точки данных для каждого класса.

```
# Splitting the dataset into the Training set and Test set
X_train, X_test, y_train, y_test = train_test_split( *arrays: X, y, test_size = 0.25, random_state = 0)

# Feature Scaling
sc = StandardScaler()
X_train = sc.fit_transform(X_train)
X_test = sc.transform(X_test)

# Training the K-NN model on the Training set
classifier = KNeighborsClassifier(n_neighbors = 5, metric = 'minkowski', p = 2)
classifier.fit(X_train, y_train)
# Predicting a new result
print(classifier.predict(sc.transform([[30,87000]])))

# Predicting the Test set results
y_pred = classifier.predict(X_test)
```

Рисунок 13 – Обучение тестовой выборки

Описание визуализации графиков представлено ниже (рисунок 14). В листинге 2 представлен код второй подпрограммы.

```
X_set, y_set = sc.inverse_transform(X_test), y_test
X1, X2 = np.meshgrid( *xi: np.arange(start = X_set[:, 0].min() - 10, stop = X_set[:, 0].max() + 10, step = 1),
np.arange(start = X_set[:, 1].min() - 1000, stop = X_set[:, 1].max() + 1000, step = 1))
plt.contourf( *args: X1, X2, classifier.predict(sc.transform(np.array([X1.ravel(), X2.ravel()]).T)).reshape(X1.shape),
alpha = 0.75, cmap = ListedColormap(('red', 'green')))
plt.xlim( *args: X1.min(), X1.max())
plt.ylim( *args: X2.min(), X2.max())
for i, j in enumerate(np.unique(y_set)):
  plt.scatter(X_set[y_set == j, 0], X_set[y_set == j, 1], c = ListedColormap(('blue', 'yellow'))(i), label = j)

plt.title('Метод k-ближайших соседей')
plt.xlabel('Возраст')
plt.ylabel('Сумма предзаказа')
plt.legend()
plt.show()
```

Рисунок 14 – Визуализация графиков второй подпрограммы

В ходе реализации программного кода были решены многие проблемы, например интеграция визуализации графиков внутри встроенного интерфейса tkinter. Программный код не выдает ошибок и стабильно запускается на любом компьютере за счёт низких требований графического интерфейса.

### 3.2 Тестирование программного обеспечения

Для тестирования ПО я выбрал метод функционального тестирования, который сосредотачивается на проверке функциональных требований системы. Он оценивает, насколько программа соответствует ожиданиям пользователя, выполняя проверку функций и возможностей, доступных в приложении. Цель тестирования - удостовериться, что программное обеспечение ведет себя так, как ожидается со стороны пользователей. Оно фокусируется на том, как приложение взаимодействует с пользователем и какие функции оно предоставляет. Основное внимание уделяется интерфейсу пользователя и функциональным возможностям.

Произведём тестирование первой подпрограммы. При запуске программы перед нами выводится интерфейс (рисунок 15), в котором мы должны ввести количество просмотров на рекламной записи и стоимость рекламной интеграции.

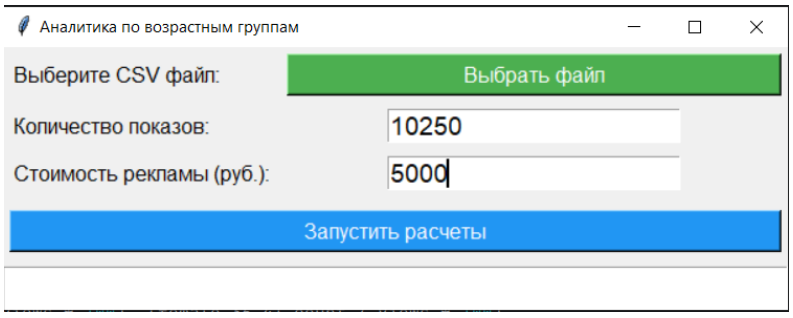



Рисунок 15 – Интерфейс программы

Если не ввести один из параметров, то программа выведет сообщение об ошибке (рисунок 16).

После чего нужно выбрать файл с помощью проводника, нажав на кнопку «Выбрать файл», и запустить расчёты, нажав на соответствующую кнопку.

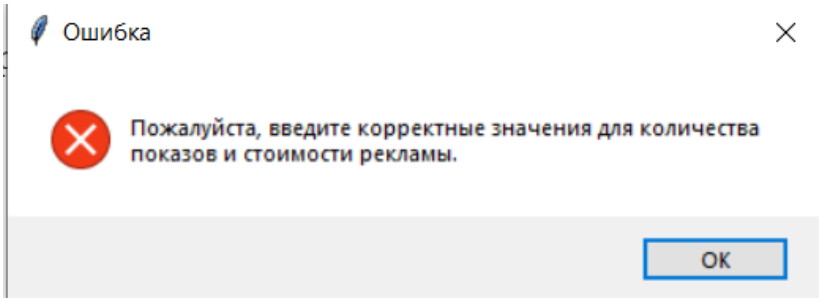



Рисунок 16 – Сообщение об ошибке, если мы не ввели параметры

Если мы не выбрали файл, то программа выведет сообщение об ошибки (рисунок 17). Когда мы выберем файл правильного формата программа выведет сообщение, что загрузка прошла успешно (рисунок 18).

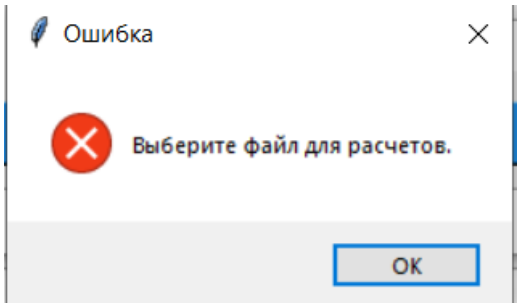



Рисунок 17 – Сообщение об ошибке в случае, если файл не загружен

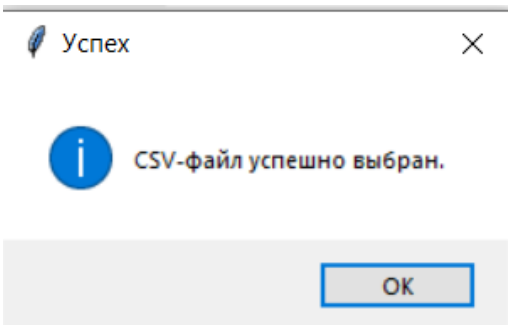



Рисунок 18 – Сообщение об успешной загрузки файла

В реализацию первой подпрограммы входит вывод информации об общей эффективности рекламной кампании, а также построение двух

половозрастных графиков на основе информации о пользователях для CTR и CPC. Рассмотрим и проанализируем реализацию программы для каждой рекламной интеграции (рисунки 19–21).

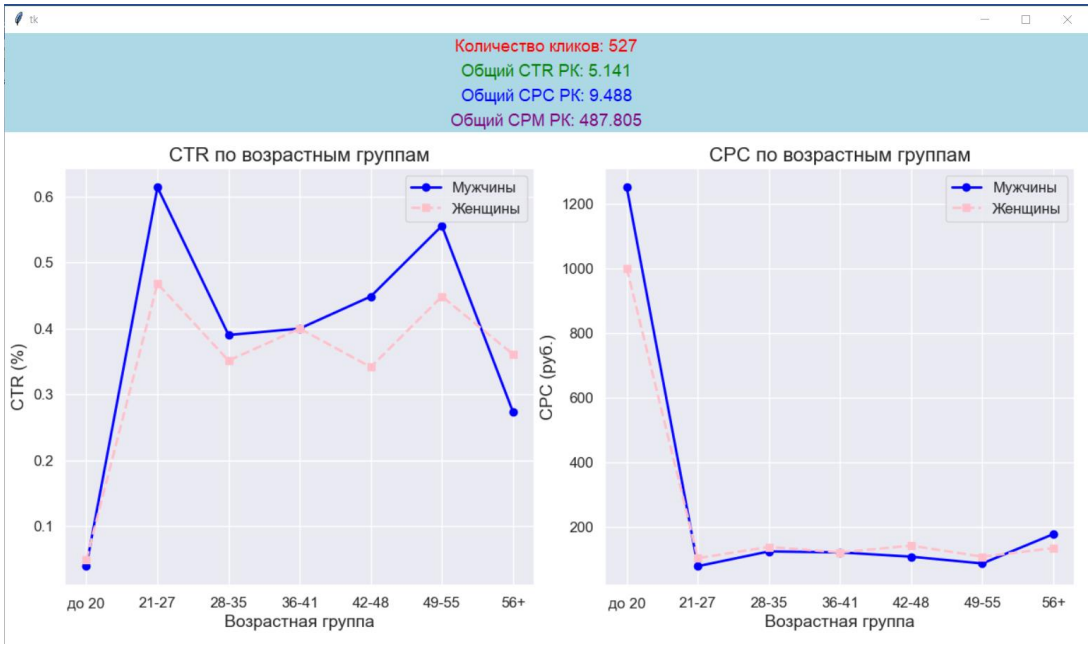



Рисунок 19 – Реализация программы для первой рекламной кампании

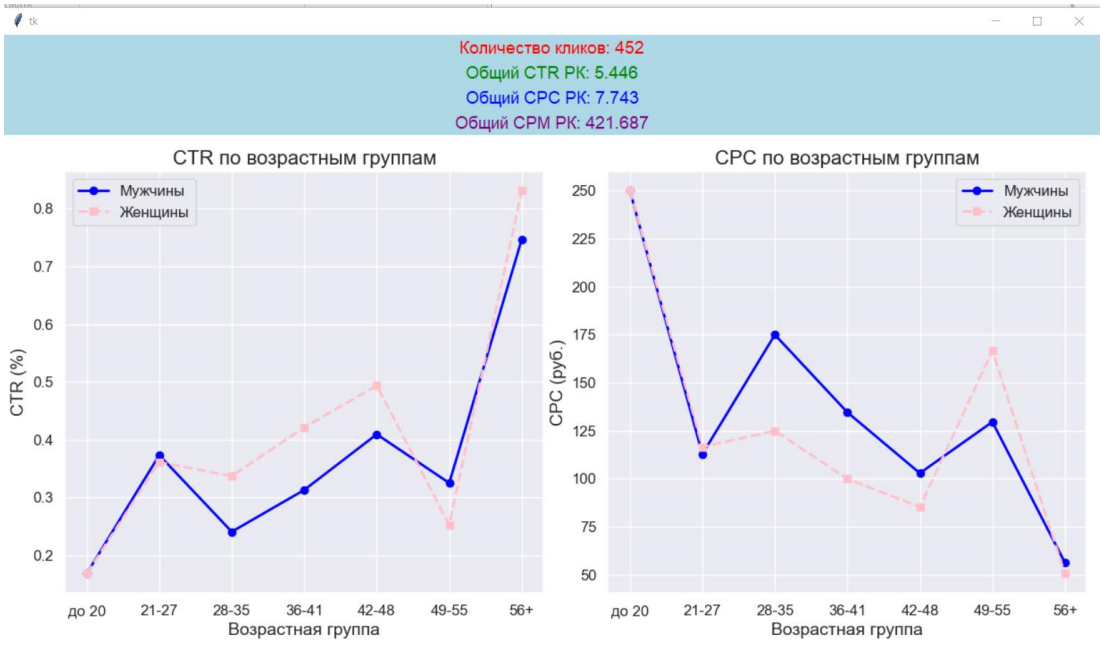



Рисунок 20 – Реализация программы для второй рекламной кампании


Рисунок 21 – Реализация программы для третьей рекламной кампании

На основе полученных результатов можно сделать вывод, что наиболее эффективные и рациональные, другими словами, имеющие наименьшую долю рекламных расходов, являются первая и вторая рекламные кампании. Однако, следует понимать, что в третьей рекламной кампании было больше просмотров, то есть охват аудитории, а иногда это бывает намного эффективнее, чем эффективная по CTR рекламная кампании.

Также можно отследить наиболее вовлеченную аудиторию по половозрастному признаку. По графикам видно, что в первой группе больше всего молодых людей от 21 до 27 лет, во второй группе наибольшее количество пользователей в возрасте от 49 до 56 и выше, а в третьей группе намного больше мужчин, чем женщин.

Тем не менее вовлеченность людей и большое количество кликов не означает, что эти же группы пользователей будут с наибольшей вероятностью покупать услуги и продукты кампании. Для отслеживания покупательской способности и спроса запустим вторую подпрограмму, которая использует для определения наиболее предпочтительной группы пользователей по

покупательской способности метод k-ближайших соседей (рисунок 22).

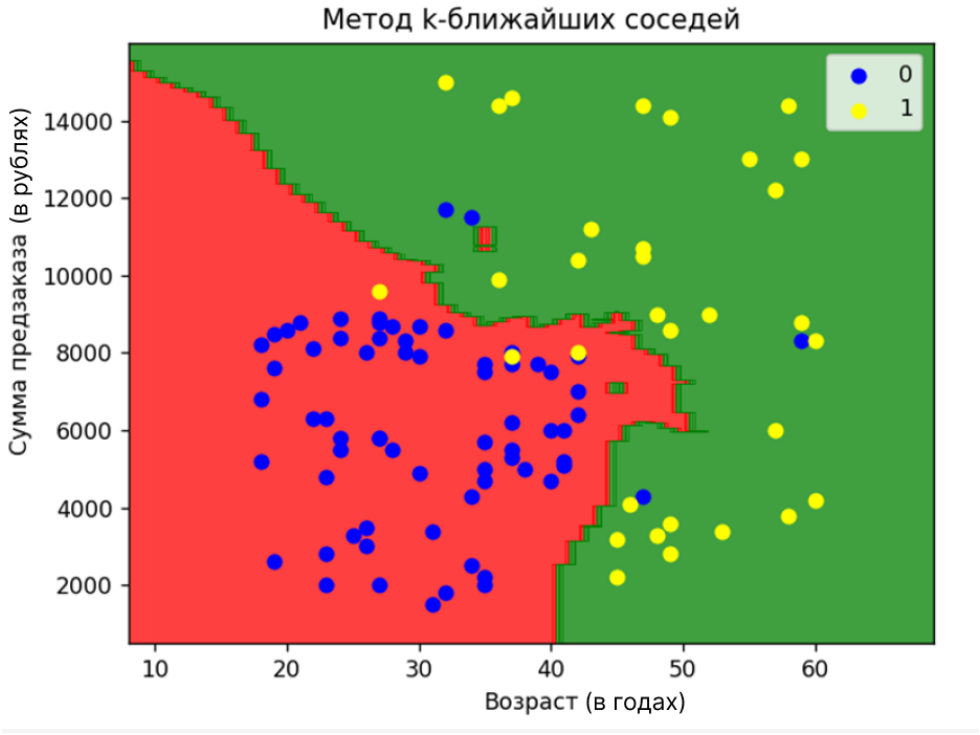



Рисунок 22 – Реализация второй подпрограммы

На рисунке 22 четко видно линию, которая отделяет наиболее предпочтительную группу пользователей от тех, кто положил в корзину, но не оформил предзаказ. На графике также видны аномалии, но можно сказать, что наиболее предпочтительная группа в возрасте от 30 до 60 лет.

Выводы по разделу 3:

В третьем разделе ВКР был описан принцип работы программного обеспечения и протестированы написанные программы. Также произведена проверка реакции системы на некорректный ввод данных со стороны пользователя и проведен анализ полученных результатов.

# Заключение

Бакалаврская работа посвящена анализу больших данных в социальных сетях для выявления тенденций и прогнозирования поведения пользователей.

В ходе выполнения ВКР были поставлены задачи на исследования.

Была поставлена задача на исследование, описаны методы анализа больших данных и характеристик поведения пользователей в социальных сетях для выявления ключевых факторов, которые влияют на анализ больших данных. Также была выбрана социальная сеть ВКонтаке, как основная социальная сеть в России на данный момент для анализа больших данных.

Далее были описаны основные показатели эффективности рекламной кампании и выявлены наиболее фундаментальные показатели. Затем была описана математическая модель метода k-ближайших соседей для определения наиболее предпочтительной группы пользователей.

Также программное обеспечение было описано по модулям, а также была спроектирована система для дальнейшей разработки программного обеспечения.

В заключительном разделе ВКР была представлена реализация программного обеспечения и проведено тестирование ПО, а также был произведён анализ полученных результатов. Тестирование производилось на трех разных рекламных записях с целью выявления наиболее эффективной для внешней рекламы.

Цель работы была выполнена: разработано программное обеспечение для выявления тенденций и прогнозирования поведения пользователей и пользовательского спроса на основе больших данных, полученных из социальной сети ВКонтакте.

Задачи, определённые для достижения цели работы, были выполнены в полном объёме.

## Список используемой литературы


1. Баженова Е.В. Анализ больших данных для выявления тенденций в социальных сетях / Е.В. Баженова. — М.: Издательство МГУ, 2018. — 256 с.
2. Баранов А.П. Прогнозирование поведения пользователей в социальных сетях / А.П. Баранов. — СПб.: Питер, 2019. — 320 с.
3. Беляев И.В. Методы анализа больших данных: теория и практика / И.В. Беляев. — Казань: Казанский федеральный университет, 2020. — 384 с.
4. Васильев С.Ю. Прогнозирование на основе анализа больших данных / С.Ю. Васильев. — М.: Альпина Паблишер, 2021. — 310 с.
5. Гаврилов Н.А. Методы машинного обучения для анализа больших данных / Н.А. Гаврилов. — Новосибирск: НГУ, 2019. — 270 с.
6. Громов В.И. Выявление тенденций в социальных сетях с помощью больших данных / В.И. Громов. — Екатеринбург: УрФУ, 2020. — 298 с.
7. Иванов А.В. Применение метода k-ближайших соседей в анализе данных / А.В. Иванов. — М.: Физматлит, 2018. — 225 с.
8. Капустин Д.С. Анализ данных в социальных сетях методом k-ближайших соседей / Д.С. Капустин. — Ростов-на-Дону: ЮФУ, 2020. — 290 с.
9. Киселёв П.Т. Метод k-ближайших соседей в задачах классификации / П.Т. Киселёв. — Пермь: ПГУ, 2019. — 245 с.
10. Ковалев М.В. Эффективность рекламных кампаний: методология и практика / М.В. Ковалев. — М.: Дашков и К°, 2021. — 315 с.
11. Кожевников Ю.В. Оценка эффективности рекламных кампаний с использованием больших данных / Ю.В. Кожевников. — Новосибирск: НГТУ, 2020. — 278 с.
12. Коротков В.А. Применение анализа больших данных для повышения эффективности рекламных кампаний / В.А. Коротков. — Екатеринбург: УрФУ, 2021. — 300 с.
13. Котов С.П. Машинное обучение в анализе больших данных / С.П.

Котов. — СПб.: СПбГУ, 2019. — 290 с.

14. Кузнецов Д.Н. Методы машинного обучения в анализе данных: k-ближайшие соседи и другие алгоритмы / Д.Н. Кузнецов. — М.: ИД Академия, 2018. — 250 с.

15. Лазарев И.В. Методы анализа больших данных и их применение в бизнесе / И.В. Лазарев. — СПб.: Питер, 2020. — 320 с.

16. Лебедев В.М. Анализ данных в социальных сетях: тенденции и прогнозы / В.М. Лебедев. — М.: Высшая школа экономики, 2021. — 340 с.

17. Михайлов А.Г. Эффективность рекламных кампаний в цифровых медиа / А.Г. Михайлов. — Казань: КФУ, 2019. — 275 с.

18. Новиков С.Н. Методы прогнозирования поведения пользователей / С.Н. Новиков. — Ростов-на-Дону: ЮФУ, 2020. — 260 с.

19. Петров А.И. Методы машинного обучения в маркетинговых исследованиях / А.И. Петров. — СПб.: СПбГУ, 2019. — 280 с.

20. Федоров Е.В. Анализ больших данных для прогнозирования поведения пользователей в социальных сетях / Е.В. Федоров. — М.: Инфра-М, 2021. — 310 с.

21. Akerkar, R. Big Data Analytics / R. Akerkar. — Oxford: Elsevier, 2013. — 580 p.

22. Bifet, A. Data Stream Mining: A Practical Approach / A. Bifet, G. Holmes, R. Kirkby, B. Pfahringer. — Cambridge: Cambridge University Press, 2011. — 251 p.

23. Breunig, M.M. LOF: Identifying Density-Based Local Outliers / M.M. Breunig, H.-P. Kriegel, R.T. Ng, J. Sander // ACM SIGMOD Record, 2000. — Vol. 29, No. 2. — P. 93-104.

24. Chandola, V. Anomaly Detection: A Survey / V. Chandola, A. Banerjee, V. Kumar // ACM Computing Surveys, 2009. — Vol. 41, No. 3. — Article 15.

25. Cristianini, N. An Introduction to Support Vector Machines and Other Kernel-based Learning Methods / N. Cristianini, J. Shawe-Taylor. — Cambridge: Cambridge University Press, 2000. — 189 p.

26. Friedman, J.H. Stochastic Gradient Boosting / J.H. Friedman // Computational Statistics & Data Analysis, 2002. — Vol. 38, No. 4. — P. 367-378.

27. Hastie, T. The Elements of Statistical Learning: Data Mining, Inference, and Prediction / T. Hastie, R. Tibshirani, J. Friedman. — New York: Springer, 2009. — 45 p.

28. Han, J. Data Mining: Concepts and Techniques / J. Han, M. Kamber, J. Pei. — Waltham: Morgan Kaufmann, 2011. — 74 p.

29. James, G. An Introduction to Statistical Learning with Applications in R / G. James, D. Witten, T. Hastie, R. Tibshirani. — New York: Springer, 2013. — 426 p.

30. Kotler, P. Marketing Management / P. Kotler, K.L. Keller. — London: Pearson Education, 2016. — 86 p.

31. Russel, S. Artificial Intelligence: A Modern Approach / S. Russel, P. Norvig. — Upper Saddle River: Prentice Hall, 2010. — 132 p.

Приложение А

**Листинг 1 (анализ эффективности рекламной кампании)**

Листинг 1 – Реализация первой программы по анализу эффективности рекламной кампании

```
import tkinter as tk
from tkinter import filedialog, messagebox
from tkinter import font
import pandas as pd
import matplotlib.pyplot as plt
from matplotlib.backends.backend_tkagg import FigureCanvasTkAgg
from matplotlib.figure import Figure
import seaborn as sns

# Функция для чтения CSV файла и выполнения расчетов
def calculate_metrics(filepath, views, CostAds):
 df = pd.read_csv(filepath, encoding="UTF-8", sep=";")
 male_20_count = df.loc[(df['Пол'] == 'M') & (df['Возрастная группа'] == 'до 20')].shape[0]
 male_21_27_count = df.loc[(df['Пол'] == 'M') & (df['Возрастная группа'] == '21-27')].shape[0]
 male_28_35_count = df.loc[(df['Пол'] == 'M') & (df['Возрастная группа'] == '28-35')].shape[0]
 male_36_41_count = df.loc[(df['Пол'] == 'M') & (df['Возрастная группа'] == '36-41')].shape[0]
 male_42_48_count = df.loc[(df['Пол'] == 'M') & (df['Возрастная группа'] == '42-48')].shape[0]
 male_49_55_count = df.loc[(df['Пол'] == 'M') & (df['Возрастная группа'] == '49-55')].shape[0]
```

```
male_56_count = df.loc[(df['Пол'] == 'М') & (df['Возрастная группа'] == '56+')].shape[0]

 female_20_count = df.loc[(df['Пол'] == 'Ж') & (df['Возрастная группа'] == 'до 20')].shape[0]
 female_21_27_count = df.loc[(df['Пол'] == 'Ж') & (df['Возрастная группа'] == '21-27')].shape[0]
 female_28_35_count = df.loc[(df['Пол'] == 'Ж') & (df['Возрастная группа'] == '28-35')].shape[0]
 female_36_41_count = df.loc[(df['Пол'] == 'Ж') & (df['Возрастная группа'] == '36-41')].shape[0]
 female_42_48_count = df.loc[(df['Пол'] == 'Ж') & (df['Возрастная группа'] == '42-48')].shape[0]
 female_49_55_count = df.loc[(df['Пол'] == 'Ж') & (df['Возрастная группа'] == '49-55')].shape[0]
 female_56_count = df.loc[(df['Пол'] == 'Ж') & (df['Возрастная группа'] == '56+')].shape[0]

 clicks = df.shape[0]

 # Создаем окно
 root = tk.Tk()

 # Изменяем цвет фона окна
 root.configure(bg='light blue')
```

```
# Изменяем размер окна
 root.geometry('400x300') # Ширина x Высота

 # Создаем текстовые метки для каждого параметра
 clicks_label = tk.Label(root, text="Количество кликов: " + str(clicks),
 font=('Arial', 14),
 bg='light blue')
 ctr_label = tk.Label(root, text="Общий CTR РК: " + str(round((clicks / views * 100), 3)),
 font=('Arial', 14),
 bg='light blue')
 cpc_label = tk.Label(root, text="Общий CPC РК: " + str(round((CostAds / clicks), 3)),
 font=('Arial', 14),
 bg='light blue')
 cpm_label = tk.Label(root, text="Общий CPM РК: " + str(round((CostAds / views * 1000), 3)),
 font=('Arial', 14),
 bg='light blue')

 # Размещаем метки в окне
 clicks_label.pack()
 ctr_label.pack()
 cpc_label.pack()
 cpm_label.pack()

 clicks_label.configure(fg='red')
 ctr_label.configure(fg='green')
```

```
cpc_label.configure(fg='blue')
cpm_label.configure(fg='purple')

# Выравниваем текст по центру и по горизонтали и вертикали
clicks_label.pack(fill='both', expand=True)
ctr_label.pack(fill='both', expand=True)
cpc_label.pack(fill='both', expand=True)
cpm_label.pack(fill='both', expand=True)

print("Количество кликов:", clicks)
print("Общий CTR РК:", round((clicks / views * 100), 3))
print("Общий CPC РК:", round((CostAds / clicks), 3))
print("Общий CPM РК:", round((CostAds / views * 1000), 3))

age_groups = ['до 20', '21-27', '28-35', '36-41', '42-48', '49-55', '56+']

# Подсчет CTR для мужчин и женщин в каждой возрастной группе
male_ctr = [(male_20_count / views * 100), (male_21_27_count / views * 100),
(male_28_35_count / views * 100), (male_36_41_count / views * 100),
(male_42_48_count / views * 100), (male_49_55_count / views * 100),
(male_56_count / views * 100)]
female_ctr = [(female_20_count / views * 100), (female_21_27_count / views *
100),
(female_28_35_count / views * 100), (female_36_41_count / views * 100),
(female_42_48_count / views * 100), (female_49_55_count / views * 100),
(female_56_count / views * 100)]
```

```
# Подсчет CPC для мужчин и женщин в каждой возрастной группе
male_cpc = [(CostAds / male_20_count if male_20_count > 0 else 0),
(CostAds / male_21_27_count if male_21_27_count > 0 else 0),
(CostAds / male_28_35_count if male_28_35_count > 0 else 0),
(CostAds / male_36_41_count if male_36_41_count > 0 else 0),
(CostAds / male_42_48_count if male_42_48_count > 0 else 0),
(CostAds / male_49_55_count if male_49_55_count > 0 else 0),
(CostAds / male_56_count if male_56_count > 0 else 0)]

female_cpc = [(CostAds / female_20_count if female_20_count > 0 else 0),
(CostAds / female_21_27_count if female_21_27_count > 0 else 0),
(CostAds / female_28_35_count if female_28_35_count > 0 else 0),
(CostAds / female_36_41_count if female_36_41_count > 0 else 0),
(CostAds / female_42_48_count if female_42_48_count > 0 else 0),
(CostAds / female_49_55_count if female_49_55_count > 0 else 0),
(CostAds / female_56_count if female_56_count > 0 else 0)]

sns.set()

# Построение графиков рядом друг с другом
fig, axs = plt.subplots(1, 2, figsize=(14, 6)) # 1 строка, 2 столбца

# График для CTR
axs[0].plot(age_groups, male_ctr, label='Мужчины', color='blue', marker='o',
linestyle='-', linewidth=2)
axs[0].plot(age_groups, female_ctr, label='Женщины', color='pink', marker='s',
linestyle='--', linewidth=2)
axs[0].set_xlabel('Возрастная группа', fontsize=14)
```

```
 axs[0].set_ylabel('CTR (%)', fontsize=14)
 axs[0].set_title('CTR по возрастным группам', fontsize=16)
 axs[0].grid(True)
 axs[0].legend(fontsize=12)
 axs[0].tick_params(axis='both', which='major', labelsize=12)

 # График для CPC
 axs[1].plot(age_groups, male_cpc, label='Мужчины', color='blue', marker='o', linestyle='-', linewidth=2)
 axs[1].plot(age_groups, female_cpc, label='Женщины', color='pink', marker='s', linestyle='--', linewidth=2)
 axs[1].set_xlabel('Возрастная группа', fontsize=14)
 axs[1].set_ylabel('CPC (руб.)', fontsize=14)
 axs[1].set_title('CPC по возрастным группам', fontsize=16)
 axs[1].grid(True)
 axs[1].legend(fontsize=12)
 axs[1].tick_params(axis='both', which='major', labelsize=12)

 # Отображение графиков
 plt.tight_layout() # Улучшение размеров графиков для лучшего взаимодействия

 # Встраивание графиков в tkinter
 canvas = FigureCanvasTkAgg(fig, master=root)
 canvas.draw()
 canvas.get_tk_widget().pack(side=tk.TOP, fill=tk.BOTH, expand=True)
```

```
 # Зафиксировать размер окна почти на весь экран
 screen_width = root.winfo_screenwidth()
 screen_height = root.winfo_screenheight()
 root.geometry(f"{int(0.8 * screen_width)}x{int(0.8 * screen_height)}")

# Функция для открытия диалога выбора файла
def select_file():
 filepath = filedialog.askopenfilename(filetypes=[("CSV files", "*.csv")])
 if filepath:
 filepath_var.set(filepath)
 messagebox.showinfo("Успех", "CSV-файл успешно выбран.")

# Функция для построения графиков после выбора файла
def plot_graphs():
 filepath = filepath_var.get()
 if not filepath:
 messagebox.showerror("Ошибка", "Выберите файл для расчетов.")
 return
 try:
 views = int(views_entry.get())
 cost_ads = int(cost_ads_entry.get())
 except ValueError:
 messagebox.showerror("Ошибка", "Пожалуйста, введите корректные значения
для количества показов и стоимости рекламы.")
 return
 # Вызов функции для расчета метрик и построения графиков
 calculate_metrics(filepath, views, cost_ads)
```

```
# Функция для отображения графиков и показателей
def display_results(metrics_data):
 # Очистка старого графика, если он есть
 for widget in graph_frame.winfo_children():
 widget.destroy()
 # Отображение показателей
 metrics_text.delete('1.0', tk.END)
 metrics_text.insert(tk.END, metrics_data)

# Создание главного окна
root = tk.Tk()
root.title("Аналитика по возрастным группам")

# Настройка стиля
root.configure(bg='#F0F0F0') # Светло-серый цвет фона

# Увеличение шрифта по умолчанию
default_font = font.Font(family="TkDefaultFont", size=12)

# Увеличение шрифта для полей ввода
entry_font = font.Font(family="TkDefaultFont", size=14)

# Переменная для хранения пути к файлу
filepath_var = tk.StringVar()

# Создание элементов интерфейса
tk.Label(root, text="Выберите CSV файл:", bg='#F0F0F0',
font=default_font).grid(row=0, column=0, sticky="w", padx=5, pady=5)
```

```
tk.Button(root, text="Выбрать файл", command=select_file, bg='#4CAF50', fg='white', font=default_font).grid(row=0, column=1, sticky="ew", padx=5, pady=5)

tk.Label(root, text="Количество показов:", bg='#F0F0F0', font=default_font).grid(row=1, column=0, sticky="w", padx=5, pady=5)
views_entry = tk.Entry(root, font=entry_font)
views_entry.grid(row=1, column=1, padx=5, pady=5)

tk.Label(root, text="Стоимость рекламы (руб.):", bg='#F0F0F0', font=default_font).grid(row=2, column=0, sticky="w", padx=5, pady=5)
cost_ads_entry = tk.Entry(root, font=entry_font)
cost_ads_entry.grid(row=2, column=1, padx=5, pady=5)

# Кнопка для запуска расчетов
tk.Button(root, text="Запустить расчеты", command=plot_graphs, bg='#2196F3', fg='white', font=default_font).grid(row=3, column=0, columnspan=2, sticky="ew", padx=5, pady=10)

# Настройка ширины столбцов
root.grid_columnconfigure(1, weight=1) # Столбец с полями ввода занимает всё доступное пространство

# Установка ширины окна
root.geometry('600x200') # Ширина x Высота
```

```
# Фрейм для отображения графиков
graph_frame = tk.Frame(root)
graph_frame.grid(row=4, column=0, columnspan=2, sticky="nsew")

# Поле для вывода показателей
metrics_text = tk.Text(root, height=10, width=50)
metrics_text.grid(row=5, column=0, columnspan=2, sticky="nsew")

# Запуск главного цикла обработки событий
root.mainloop()
```

Приложение Б

**Листинг 2 (поиск заинтересованной группы пользователей)**

Листинг 2 – Реализация второй программы по поиску наиболее заинтересованной группы пользователей

```
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd
import warnings
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
from sklearn.neighbors import KNeighborsClassifier
from matplotlib.colors import ListedColormap
from sklearn.metrics import confusion_matrix, accuracy_score

warnings.filterwarnings("ignore", category=UserWarning)

df = pd.read_csv(r"C:\VKR 1\Показатели корзины.csv", encoding="cp1251")
df.head()
df.describe()
df.info

X = df.iloc[:, :-1].values
y = df.iloc[:, -1].values

# Splitting the dataset into the Training set and Test set
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.25, random_state = 0)
```

```
# Feature Scaling
sc = StandardScaler()
X_train = sc.fit_transform(X_train)
X_test = sc.transform(X_test)

# Training the K-NN model on the Training set
classifier = KNeighborsClassifier(n_neighbors = 5, metric = 'minkowski', p = 2)
classifier.fit(X_train, y_train)
# Predicting a new result
print(classifier.predict(sc.transform([[30,87000]])))

# Predicting the Test set results
y_pred = classifier.predict(X_test)

cm = confusion_matrix(y_test, y_pred)
print(cm)
accuracy_score(y_test, y_pred)
# Visualising the Test set results
X_set, y_set = sc.inverse_transform(X_test), y_test
X1, X2 = np.meshgrid(np.arange(start = X_set[:, 0].min() - 10, stop = X_set[:, 0].max() + 10, step = 1),
np.arange(start = X_set[:, 1].min() - 1000, stop = X_set[:, 1].max() + 1000, step = 1))
plt.contourf(X1, X2, classifier.predict(sc.transform(np.array([X1.ravel(), X2.ravel()]).T)).reshape(X1.shape),
alpha = 0.75, cmap = ListedColormap(('red', 'green')))
plt.xlim(X1.min(), X1.max())
plt.ylim(X2.min(), X2.max())
```

```
for i, j in enumerate(np.unique(y_set)):
 plt.scatter(X_set[y_set == j, 0], X_set[y_set == j, 1], c = ListedColormap(('blue',
'yellow'))(i), label = j)

plt.title('Метод k-ближайших соседей')
plt.xlabel('Возраст')
plt.ylabel('Сумма предзаказа')
plt.legend()
plt.show()
```