СОДЕРЖАНИЕ:

[ВВЕДЕНИЕ 2](#_Toc197081418)

[1.ТЕОРЕТИЧЕСКАЯ ЧАСТЬ. 3](#_Toc197081419)

[1.1 Терминология по проекту и глоссарий. 3](#_Toc197081420)

[1.2. РАСПРЕДЕЛЕНИЕ РОЛЕЙ И РАБОТЫ. 8](#_Toc197081421)

[1.3. СТЕК ТЕХНОЛОГИЙ. 10](#_Toc197081422)

2.ПРАКТИЧЕСКАЯ ЧАСТЬ…………………………………………………... 11

2.1. Архитектура ПО…………………………………………………………….11

2.2. Разработка проекта по ролям………………………………………………13

2.3. Контроль выполнения плана……………………………………………….18

ЗАКЛЮЧЕНИЕ………………………………………………………………….19

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ……………………………21

# ВВЕДЕНИЕ

В наше время современный рынок электронной коммерции характеризуется высокой конкуренцией, где ключевым фактором успеха становится удовлетворенность клиентов. Одним из основных источников обратной связи являются отзывы покупателей, анализ которых позволяет компаниям улучшать продукты, корректировать маркетинговые стратегии и повышать уровень сервиса.

Одним из самых популярных маркетплейсов является Wildberries который является одним из самых крупнейших маркетплейсов в России, и в некоторых странах СНГ, на котором ежедневно публикуются десятки тысяч отзывов. Ручной анализ такого объема данных будет очень трудно, поэтому программа веб парсинга сайтов является самым удобным способом для анализа большого объема отзывов.

Цель работы: Разработка веб-парсинга сайта Wildberries для сбора отзывов с использованием семантического анализа текста.

Задачами проекта является:

1. Разработка программы веб парсинг сайта wildbberies на языке программирования Python.

2. Разработка и работа с моделькой (сентимент анализ).

3. Разработка функции графиков после анализа.

4. Разработка интерфейса.

Предметы исследования: Процесс сбора и анализа отзывов покупателей на платформе Wildberries с использованием методов веб-парсинга и семантического анализа текста.

Объект исследования: Разработка веб-парсинга сайта Wildberries для сбора отзывов с использованием семантического анализа текста.

После того как рассмотрели введению перейдем к теоретической части и потом приступим к самой разработки программы.

# 1.ТЕОРЕТИЧЕСКАЯ ЧАСТЬ.

# 1.1 Терминология по проекту и глоссарий.

Веб парсинг – автоматизированное получение информации с веб-сайтов с помощью программ, которые называются парсерами. Парсеры собирают и систематизируют определенный контент с веб-страниц, документов, API или даже исходного кода, например, с помощью веб-парсинга можно собрать отзывы. Парсинг имеет широкий круг применения можно использовать для анализа различных типов контента с веб-сайтов, такие как сведения о конкурентах, отзывы клиентов, новостные статьи, контент в социальных сетях, аудитории групп, частоту поисковых запросов, популярных заголовки и связанные ключевые слова.

Некоторые задачи, которые можно решить с помощью парсинга:

1. Сбор цен на товары и услуги. Парсинг сайтов помогает быстро собрать информацию о ценах на продукты конкурентов.
2. Сбор поисковых фраз. Парсер может сохранять фразы из поисковых подсказок и блока «Люди ищут» внизу.
3. Сбор целевой аудитории в соцсетях. С помощью парсинга собирают людей по нужным критериям — например, тех, кто проявляет активность в сообществах конкурентов.
4. Поиск битых ссылок. Парсер может быстро обойти весь сайт и сохранить в документе все битые ссылки — и страницы, на которых они расположены.
5. Сбор идей для контента. Парсер может собрать в одном месте посты или статьи на выбранную тему.
6. Поиск отзывов. Программы каждый день будут собирать отзывы, появившиеся за последние сутки, а менеджеры быстро отреагируют на них.
7. Парсинг может использоваться в разных сферах, например в финансах, аудите ссылок, автоматизации бизнес-процессов, создании контента, HR, образовании и недвижимости.

Важно учитывать, что парсинг имеет этические и юридические аспекты. Некоторые сайты запрещают парсинг в своих правилах использования. Кроме того, данные, извлечённые с сайта, могут быть защищены авторскими правами.

И теперь чтобы сортировать отзывы на отрицательные и положительные, разберем что такое сентимент анализ или же анализ тональности текста.

Анализ тональности — это многоэтапный процесс, определяющий эмоциональную тональность текста, помогая бизнесу и организациям понять мнения клиентов, рыночные тренды и общественные настроения. Процесс начинается со сбора данных, где информация поступает из различных источников, таких как социальные сети, отзывы, новостные статьи и опросы. Эти источники предоставляют богатый массив данных, отражающий мнения и чувства людей.

Далее следует обработка текста, включающая очистку собранных данных. На этом этапе удаляются нерелевантные элементы, такие как URL, специальные символы и стоп-слова, которые не имеют значения для анализа. Затем данные разбиваются на более мелкие единицы, такие как слова или фразы (токенизация), чтобы упростить анализ. Иногда добавляются дополнительные этапы, такие как стемминг (приведение слов к корню) и лемматизация (приведение слов к их базовой форме) для повышения точности.

После этого происходит определение тональности. На этом этапе продвинутые алгоритмы или модели машинного обучения анализируют обработанный текст, чтобы классифицировать тональность. Оно может быть положительным, отрицательным или нейтральным в зависимости от контекста. Анализ тональности может быть детализированным, выявляя такие эмоции, как радость, гнев, грусть или удивление, а также интенсивность этих чувств.

Наконец, результаты анализа представляются в понятной форме, например, в виде отчетов, интерактивных панелей или визуальных графиков, позволяя пользователям легко интерпретировать данные и принимать меры. Эти данные помогают организациям принимать решения на основе аналитики, отслеживать репутацию бренда или анализировать обратную связь для улучшения продуктов и услуг.

Основная цель анализа тональности — автоматизация процесса понимания и классификации мнений, эмоций или отношения, выраженных в текстовых данных.

Некоторые задачи анализа тональности:

1. анализ отзывов о товарах и услугах;
2. определение языка вражды;
3. оценка общественного мнения;
4. анализ обратной связи от клиентов;
5. отслеживание постов в социальных сетях.

Процесс анализа тональности включает несколько этапов:

1. Сбор данных. Информация поступает из различных источников, таких как социальные сети, отзывы, новостные статьи и опросы.
2. Обработка текста. Включает очистку собранных данных, удаление нерелевантных элементов и разбиение текста на более мелкие единицы.
3. Определение тональности. На этом этапе продвинутые алгоритмы или модели машинного обучения анализируют обработанный текст, чтобы классифицировать тональность.
4. Представление результатов. Анализ представляет данные в понятной форме, например, в виде отчётов, интерактивных панелей или визуальных графиков.

После выполнения сентимент анализа часто возникает необходимость визуализировать результаты для облегчения понимания и интерпретации.

Для этого мы разработаем функции графиков для дальнейшей работы, и чтобы разработать функции графиков будем использовать библиотеки для python такие как: Matplotlib и Seaborn.

Matplotlib — базовая библиотека построения графиков на Python. Она предоставляет комплексные функции построения графиков и параметры настройки.

Некоторые возможности Matplotlib:

* создание базовых графиков, таких как линейные графики, точечные диаграммы, столбчатые диаграммы, гистограммы и многое другое;
* настройка внешнего вида графиков, настройка меток, заголовков, легенд, цветов и стилей.

Seaborn — библиотека визуализации статистических данных, построенная поверх Matplotlib. Она предлагает высокоуровневый интерфейс и привлекательные стили по умолчанию.

Некоторые возможности Seaborn:

* создание статистических графиков, таких как прямоугольные графики, графики скрипок, парные графики и тепловые карты.
* использование встроенных тем и цветовых палитр для улучшения эстетики сюжета.

Преимущества Matplotlib: гибкость и контроль над каждым аспектом графика, поддержка анимаций и интерактивных графиков, широкий спектр типов графиков. Преимущества Seaborn: простота использования и высокоуровневые интерфейсы, красивые и стильные графики по умолчанию, отличная интеграция с pandas DataFrame.

Обе библиотеки интегрируются с другими инструментами анализа данных в Python. Matplotlib легко работает с числовыми массивами NumPy, что делает его незаменимым для научных вычислений. Seaborn, в свою очередь, отлично понимает структуры данных pandas (DataFrame), позволяя создавать визуализации буквально одной строкой кода.

И наконец, разберем что такое интерфейс.

Интерфейс – это набор функций, с помощью которых можно взаимодействовать с данными. Он содержит возвращаемое значение, имена функций и их сигнатуры — количество и типы входящих параметров.

Интерфейс помогает определить общее поведение группы классов, возможно, не связанных между собой.

Для разработки интерфейса в Python используются специальные библиотеки и фреймворки. Одна из таких библиотек — Tkinter. Она входит в стандартный пакет Python и позволяет создавать приложения для Windows, mac OS и Linux.

Процесс разработки интерфейса с помощью Tkinter включает следующие шаги:

– Создание главного окна. Это окно будет служить основой интерфейса и контейнером для всех других элементов.

– Добавление виджетов. Используются базовые элементы, такие как кнопки, текстовые поля, метки и таблицы.

– Настройка взаимодействий. Для обработки пользовательских действий добавляются функции-обработчики событий.

– Сохранение состояния. Для сложных приложений важно обеспечить сохранение введённых данных или текущего состояния интерфейса, например, через локальные файлы или базы данных.

– Запуск цикла событий. Все приложения с графическим интерфейсом на базе Tkinter используют основной цикл для обработки событий в реальном времени. Это необходимо для корректной работы всех элементов.

# 1.2. РАСПРЕДЕЛЕНИЕ РОЛЕЙ И РАБОТЫ.

Мы наконец то разобрали теоретическую часть о веб парсинге, сентимент анализе, функции графиков, и о интерфейсе. Далее перейдем к командной работе.

Команда состоит из пяти человек, таких как Мазуров Андрей КИСП-9-22(2), Романов Дмитрий КИСП-23(2), Антонов Викториан КИСП-9-22(1), Егоров Артем КИСП-9-22(1), Борисов Айтал КИСП-9-22(1).

И теперь можем распределить роли и работы для нашего проекта на тему веб – парсинг сайта wildberries с использованием сентимент анализа текста.

Роли и работы:

1. Романов Дмитрий и Егоров Артем будут отвечать за веб-парсинг сайта wildberries.

План по выполнению работы по части Веб – парсинга:

* Скачать библиотеку Selenium.
* Написать функцию для обозревания web view сайта Wildberries
* Написать функцию для поиска отзывов конкретного продукта
* С помощью исходного html кода страницы, найти отзывы по типу «<div class = “feedback\_info” </div>»
* Выгрузить данные по отзывам
* Перевести в формат csv

1. Антонов Викториан после того как будет готов веб парсинг будет заниматься анализом тональности текста.

План по выполнению работы по части анализа тональности:

* Изучение анализа тональности.
* Написать функцию для обработки ответа csv файла.
* Написать функцию для сентимент анализа по отзывам в csv файле.
* Распределить количество позитивных, негативных,

нейтральных отзывов по возрастанию

* Сохранить проанализированный csv файл

1. Мазуров Андрей будет заниматься разработкой функции графиков после анализа тональности текста.

– Разработать диаграмму для распределения позитивных и негативных отзывов:

Подзадачи:

– Определить тип диаграммы (например, круговая и тд)

– Реализовать функцию для построения диаграммы с использованием библиотеки Matplotlib.

­– Разработать гистограммы для распределения оценок:

Подзадачи:

– Определить диапазоны оценок (например, от 1 до 5).

– Реализовать функцию для построения гистограммы с использованием библиотеки Matplotlib.

1. Борисов Айтал будет заниматься интерфейсом нашей программы после всех этих действий.

* Разработать консольный интерфейс для вывода результатов в консоль:

Подзадачи:

– Определить структуру вывода данных (например, форматирование текста, разделение на секции).

– Реализовать функции для отображения результатов анализа тональности (позитивные, негативные, нейтральные отзывы).

– Добавить возможность взаимодействия с пользователем (например, выбор опций для отображения различных графиков).

# 1.3. СТЕК ТЕХНОЛОГИЙ.

Для реализации проекта по веб-парсингу сайта Wildberries с использованием сентимент анализа текста и визуализации результатов, мы будем использовать следующий стек технологий:

1. Язык программирования Python: Основной язык для разработки, который поддерживает все необходимые библиотеки для парсинга, анализа и визуализации данных.

2. Библиотеку для веб – парсинга будем использовать Selenium – Библиотека для автоматизации браузеров, которая позволяет взаимодействовать с веб-страницами и извлекать данные.

3. Для анализа тональности будем использовать NLTK (Natural Language Toolkit) – Библиотека для обработки естественного языка, которая может быть использована для анализа тональности текста.

4. Библиотека для построения графиков будем использовать Matplotlib – Основная библиотека для построения графиков и визуализации данных.

5. Для работы с данными подойдет библиотека Pandas – Библиотека для работы с данными в формате таблиц (DataFrame), которая упрощает обработку и анализ данных.

6. Интерфейс пользователя сделаем с помощью Tkinter – Стандартная библиотека Python для создания графических интерфейсов пользователя (GUI).

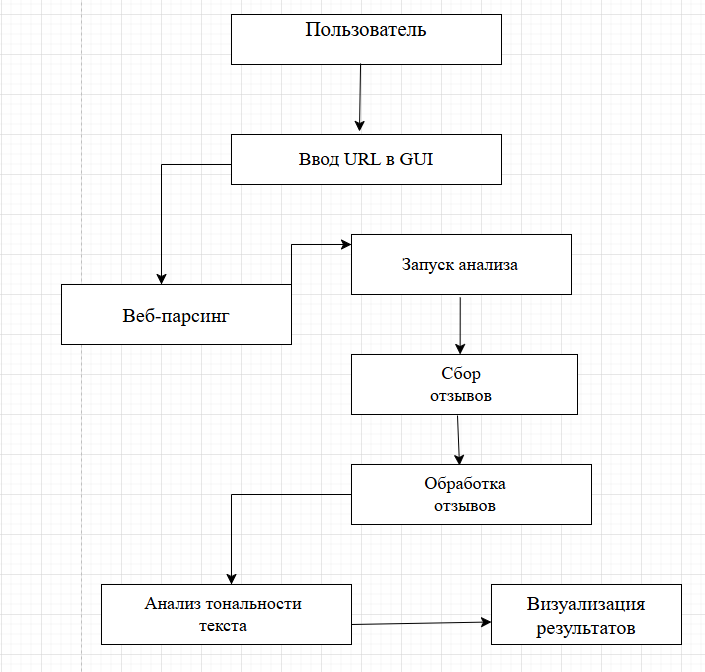
7. Среда разработки будет VS Code.

8. Вся программа будет хранится в GitHub.

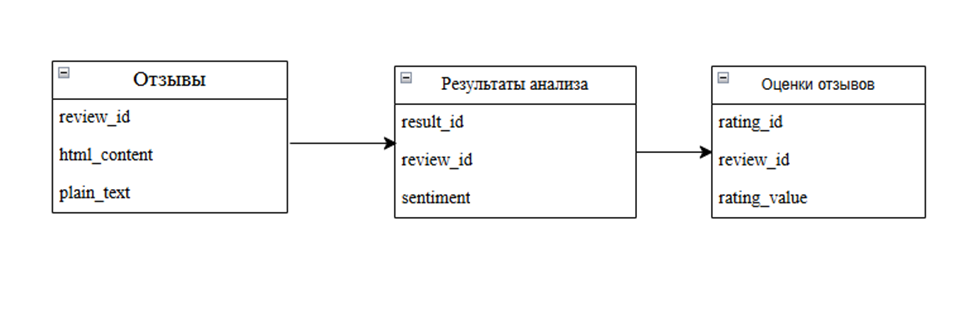
2. ПРАКТИЧЕСКАЯ ЧАСТЬ.

2.1. Архитектура ПО.

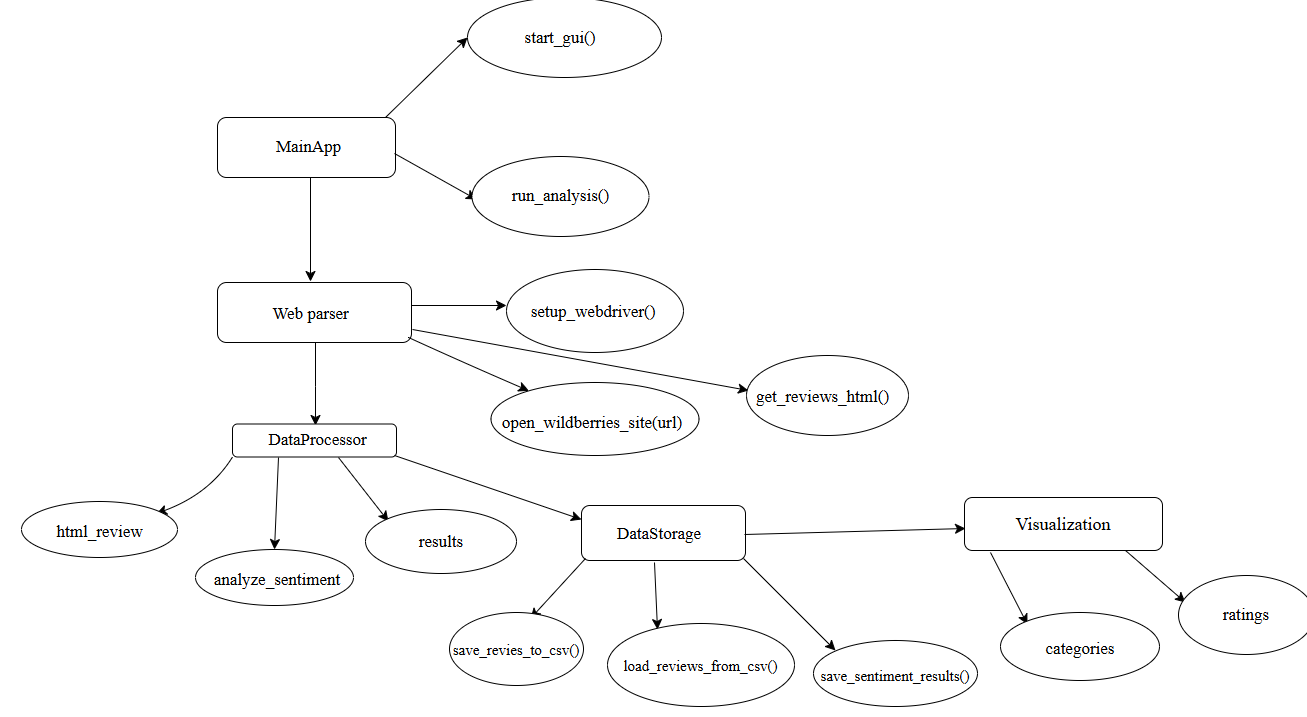
1) Диаграмма USE CASE.



2) ERD – диаграмма.



3) UML – диаграмма.



2.2. Разработка проекта по ролям.

Общая структура проекта:

Проект разбит на 4 части, каждый из которых выполняет свою задачу.

1. Веб парсинг выполнили Романов Дмитрий и Егоров Артем:

Для веб-парсинга будем использовать библиотеку Selenium чтобы извлечь отзывы с маркетплейса.

Функции для веб-парсинга:

1) Setup\_webdriver(): настраивает драйвер Chrome для запуска в фоновом режиме (см. Рис 1):

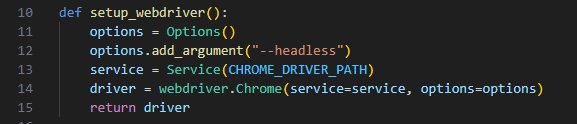


Рисунок 1 – Setup\_webdriver().

2) Open\_wildberries\_site(driver, url): открывает сайт Wildberries по указанному URL (см. Рис 2):

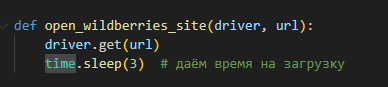


Рисунок 2 – Open\_wildberries\_site(driver, url).

3) Get\_reviews\_html(driver): извлекает HTML-код отзывов из элементов с классом feedback\_info (см. Рис 3):

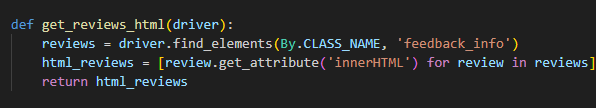


Рисунок 3 – Get\_reviews\_html.

4) Save\_reviews\_to\_csv(reviews, filename='reviews.csv'): сохраняет полученные отзывы в CSV файл (см. Рис 4):

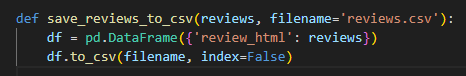


Рисунок 4 – Save\_reviews\_to\_csv.

2. Анализ тональности текста выполнил Антонов Викториан.

После сбора отзывов, с использованием библиотеки NLTK, отзывы будем анализировать на положительные, нейтральные и отрицательные.

Функции для анализа тональности текста:

1) load\_reviews\_from\_csv(filename='reviews.csv'): загружает отзывы из CSV файла (см. Рис 5):

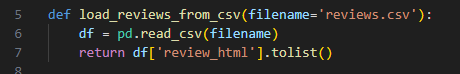


Рисунок 5 – Загрузка отзывов из CSV.

2) Preprocess\_review(html\_review): предварительная обработка отзывов для получения текста без HTML-разметки (см. Рис 6):

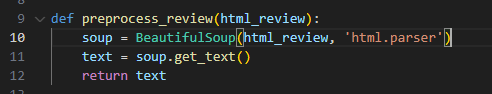


Рисунок 6 – Предварительная обработка.

3) Analyze\_sentiment(texts): анализирует тональность каждого отзыва и классифицирует как позитивный, негативный или нейтральный (см. Рис 7):

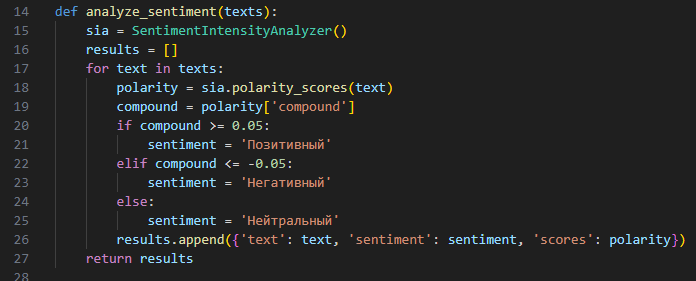


Рисунок 7 – Анализ тональности.

4) Save\_sentiment\_results(results, filename='reviews\_sentiment.csv'): сохраняет результаты анализа в новый CSV файл (см. Рис 8):

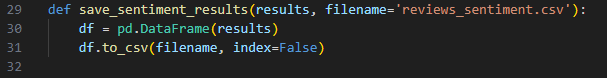


Рисунок 8 – Сохранение результатов.

5) Categorize\_reviews(results): подсчитывает количество отзывов в каждой категории (см. Рис 9):

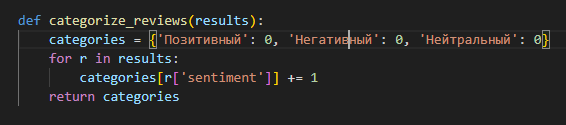


Рисунок 9 – Подсчет отзывов.

3. Функция графиков после анализа тональности выполнил Мазуров Андрей.

После того как создали функцию анализа тональности текста, необходимо сделать визуализацию, а точнее создаnm диаграмму для распределения тональности отзывов с использованием библиотеки Matplotlib.

Функции для визуализации:

1) Plot\_sentiment\_distribution(categories): создает круговую диаграмму для распределения тональности отзывов (см. Рис 10):

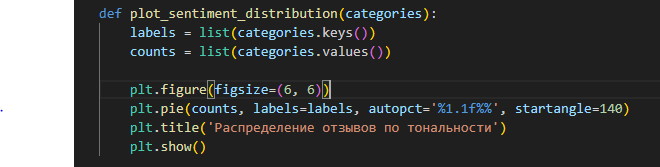


Рисунок 10 – Распределение отзывов.

2) Plot\_rating\_histogram(ratings): строит гистограмму для распределениеоценок, если они доступны (см. Рис 11) :

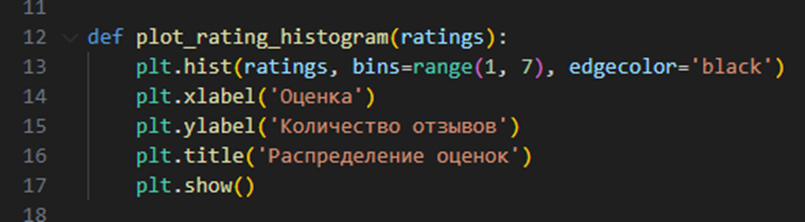


Рисунок 11 – Гистограмма.

4. Интерфейс разработал Борисов Айтал.

У нас будет консольный интерфейс для запуска всех процессов.

Функции создания интерфейса:

Start\_gui(): Создает окно с полем для ввода URL и кнопкой для запуска анализа (см. Рис 12):

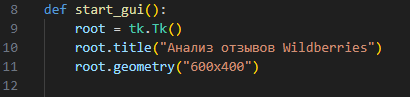


Рисунок 12 – Поле ввода.

Run\_analysis(): последовательное выполнение всех этапов: веб-парсинг, анализ тональности, визуализация результатов (см. Рис 13):



Рисунок 13 – Последовательность.

2.3. Контроль выполнения плана.

Выполненные задачи по разработке:

1. Веб – парсинг сайта Wildberries:

– Настройка Selenium WebDriver (setup\_webdriver).

– Открытие сайта по URL (open\_wildberries\_site).

– Получение отзывов с помощью поиска элементов с классом 'feedback\_info' (get\_reviews\_html).

– Сохранение отзывов в CSV (save\_reviews\_to\_csv).

2. Анализ тональности:

– Загрузка отзывов из CSV (load\_reviews\_from\_csv).

– Обработка HTML отзывов в текст (preprocess\_review с BeautifulSoup).

– Анализ сентимента с помощью SentimentIntensityAnalyzer из NLTK (analyze\_sentiment).

– Сохранение результатов анализа в CSV (save\_sentiment\_results).

– Категоризация отзывов по типам (позитивный, негативный, нейтральный) (categorize\_reviews).

3. Визуализация:

– Построение круговой диаграммы распределения отзывов по тональности (plot\_sentiment\_distribution).

4. Интерфейс:

– GUI на Tkinter, позволяющее ввести URL и запустить процесс анализа (start\_gui).

Использованные технологии:

– Язык программирования: Python 3.9+

– Библиотеки Python: selenium, pandas, beautifulsoup4, nltk, matplotlib tkinter

– Система контроля версий: GitHub

– Среда разработки: Visual Studio Code.

– Операционная система: Windows 10 Pro.

ЗАКЛЮЧЕНИЕ

В ходе выполнения проекта по разработке системы веб-парсинга сайта Wildberries с использованием методов семантического анализа текста были достигнуты поставленные цели и выполнены все запланированные задачи. В рамках проекта была создана автоматизированная программа, которая позволяет эффективно собирать отзывы покупателей с платформы Wildberries, сохранять их в удобном формате CSV и проводить их аналитическую обработку.

Использование библиотеки Selenium обеспечило надежное взаимодействие с динамическим содержимым сайта, позволяя извлекать отзывы по заданным критериям. Обработка HTML-кода отзывов с помощью BeautifulSoup позволила выделить чистый текст для дальнейшего анализа. Для определения эмоциональной окраски отзывов и классификации их по тональности применена модель SentimentIntensityAnalyzer из библиотеки NLTK, что обеспечило автоматическую сегментацию мнений пользователей на позитивные, негативные и нейтральные.

Для визуализации результатов анализа были реализованы функции построения круговых диаграмм и гистограмм с использованием библиотек Matplotlib и Seaborn, что позволило наглядно представить распределение отзывов по категориям и оценкам. Интерфейс программы, разработанный на базе Tkinter, обеспечивает удобное взаимодействие пользователя с системой, позволяя запускать процесс сбора и анализа данных, а также просматривать полученные графики и отчеты.

Используемый стек технологий, включающий Python, Selenium, NLTK, Pandas, Matplotlib, Seaborn и Tkinter, обеспечивает широкие возможности для автоматизации, анализа и визуализации данных, а также легкую интеграцию различных компонентов системы.

В результате работы созданная система способствует автоматизации сбора отзывов, их анализа и интерпретации, что значительно облегчает мониторинг репутации продукта и помогает принимать обоснованные управленческие решения. В дальнейшем проект может быть дополнен расширенными модулями анализа, интеграцией с базами данных и возможностью автоматического формирования отчетов в реальном времени.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОНИКОВ.