“MACHINE LEARNING FOR E-COMMERCE PRODUCT EVALUATION,RANKING AND RECOMMENDATION WITH SENTIMENTAL ANALYSIS”

by

S. M. AKSHAYA 2021115012

S. SATHYA DHARINI 2021115097

A summer project report
submitted to the Faculty of

INFORMATION AND COMMUNICATION ENGINEERING

for the partial fulfillment of the
award of the degree of

Bachelor of Technology

in

Information Technology

DEPARTMENT OF INFORMATION SCIENCE AND TECHNOLOGY
COLLEGE OF ENGINEERING, ANNA UNIVERSITY
CHENNAI 600 025.

AUGUST 2023

CERTIFICATE

Certified that this report titled " **MACHINE LEARNING FOR E-COMMERCE PRODUCT EVALUATION ,RANKING AND RECOMMENDATION WITH SENTIMENTAL ANALYSIS** ", is a *bona fide* work of **Ms. AKSHAYA S M (2021115012), Ms. SATHYA DHARINI S (2021115097)** who carried out the work under my supervision, for the partial fulfillment of the requirements for the award of the degree of *Bachelor of Engineering* in *Information Technology*. Certified further that to the best of my knowledge and belief, the work reported herein does not form part of any other thesis or dissertation on the basis of which a degree or an award was conferred on an earlier occasion.

Place: Chennai.

Date:

Dr. K INDRA GANDHI
Associate Professor,
Dept. of Information Science and Techn.,
College of Engineering, Guindy
Anna University,
Chennai - 600 025.

COUNTERSIGNED

Dr. S. Sridhar
Professor and Head
Department of Information Science and Technology
College of Engineering, Anna University,
Chennai 600 025.

ACKNOWLEDGEMENT

I would like to express my sincere thanks and deep sense of gratitude to my guide *Dr. K. INDRA GANDHI*, Associate Professor, Department of Information Science and Technology, for her valuable guidance suggestions and constant encouragement paved way for the successful completion of the project work.

I also thank *Dr. S. Sridhar*, Professor & Head, Department of Information Science and Technology for his kind support and for providing necessary facilities to carry out the work.

I am bound to convey my sincere thanks to the project committee *Dr. P. Varalakhsmi*, Professor, *Dr. S. Bama*, Associate Professor, and *Dr. K. Arul Deepa*, Assistant Professor, Department of Information Science and Technology for their suggestions and motivations.

S. M. AKSHAYA

S. SATHYADHARINI

ABSTRACT

The "Machine Learning for E-commerce Product Evaluation ,Ranking and Recommendation with Sentimental Analysis" project aims to simplify the process of finding the perfect product on an ecommerce site for users by leveraging a combination of machine learning (ML) algorithms and web scraping techniques. By scraping data from Amazon, including product descriptions, URLs, review counts, overall ratings, and specific rating percentages, the system gathers the necessary information. This data is then processed and analyzed using Python, using clustering algorithms for optimal pricing and sentiment analysis performed using Hugging Face's pretrained NLP model.

The primary goal of this automation is to save user's time and provide accurate product recommendations based on ratings, reviews, and customer needs(specifications). Instead of analysing through thousands of products datas on a ecommerce site, users can rely on this system to rank products based on various filters, including data deduplication, data preprocessing, price optimization through clustering, review emotion classification using Hugging Face's pretrained ML model, and sentiment analysis using Vader scores. The final output includes Vader scores along with the top-ranked product's URL, ensuring a streamlined and efficient product selection process.

TABLE OF CONTENTS

CHAPTER 1

INTRODUCTION

1.1 OVERVIEW

Literature in the following areas has been analyzed:

a. WEB SCRAPPING

b. DATA PROCESSING USING PYTHON, CLASSIFICATION USING HUGGING PRE-TRAINED MODEL AND SENTIMENTAL ANALYSIS

WEB SCRAPING:-

Introduction

Web scraping is a technique used to extract data from websites for various purposes, including research, analysis, and business intelligence. This literature survey explores the historical context, legal and ethical aspects, techniques, applications, challenges, tools, and future trends in web scraping.

Historical Perspective

Web scraping has evolved alongside the internet. In the early days, it was often referred to as "screen scraping" and was used primarily for automating data entry tasks. Over time, web scraping has become more sophisticated and is now integral to data-driven decision-making processes.

Legal and Ethical Aspects

The legal landscape of web scraping is complex and varies by jurisdiction. Copyright, data protection, and terms of service agreements play a significant role. Ethical

considerations include issues of privacy, consent, and responsible data usage. Legal cases, such as HiQ v. LinkedIn, have shaped the legal understanding of web scraping.

Web Scraping Techniques

Web scraping techniques range from simple HTML parsing to more advanced methods like web automation and API integration. Tools such as BeautifulSoup, Scrapy, and Selenium are commonly used to facilitate scraping. The choice of technique depends on the specific requirements of the task.

Applications of Web Scraping

Web scraping finds applications across industries, including e-commerce, finance, healthcare, and academia. Examples include price monitoring in e-commerce, sentiment analysis in social media, and data collection for academic research.

References used:

https://www.selenium.dev/about/

https://pypi.org/project/beautifulsoup4/

https://chromedriver.chromium.org/downloads

DATA PROCESSING AND SENTIMENTAL ANALYSIS:-

Introduction

Pretrained models have revolutionized the field of Natural Language Processing (NLP) by providing powerful language representations that can be fine-tuned for various downstream tasks. This literature survey explores the historical context, architecture, training techniques, applications, challenges, and future trends in pretrained models for NLP.

Applications of Pretrained Models

Pretrained models have been applied to a wide range of NLP tasks, including sentiment analysis, named entity recognition, machine translation, question-answering, and text generation. They serve as strong feature extractors and can be fine-tuned with relatively small task-specific datasets.

Tools and Frameworks

Frameworks like Hugging Face Transformers have made it easier for researchers and developers to work with pretrained models. They offer prebuilt models and APIs for fine-tuning and inference, facilitating their adoption in various applications.

Future Trends

The future of pretrained models in NLP is promising. Ongoing research focuses on reducing model size and inference latency, improving interpretability, and addressing ethical concerns. Multilingual models, few-shot learning, and zero-shot learning are emerging trends.

References used:

https://huggingface.co/

https://www.kaggle.com/

1.2 LITERATURE SURVEY

1."EFFICIENT SCRAPING OF DATA FROM WEBSITES USING SELENIUM"

This paper concludes about the benefits of using selenium as web scrapper which reduces the manual work of extracting data from different websites and thus the automatic generation of data makes easier to process and analyse further.

Authors: [1]Shreya V. Dhoke student ,[2] Anupama D. Sakhare Asst Prof,[3] Satish J.Sharma Prof.

1,2,3 Dept of Electronics and Computer Science, Rakshtrasant Tukdoji Maharaj Nagpur University,India . 2022 jetir, volume 9, issue 6,issn:2349-5162

2."RoBERTa-GRU: A Hybrid Deep Learning Model for Enhanced Sentiment Analysis"

The results of the experiments carried out on this paper IMDb, Sentiment140, and Twitter US Airline datasets indicate that the RoBERTa-GRU model outperforms all other comparison methods, with accuracy scores of 94.63%, 89.59%, and 91.52%, respec-

tively. The combination of the RoBERTa and GRU results in a powerful and efficient model for sentiment analysis, making it a promising solution for a variety of NLP tasks.
Authors:Kian Long Tan ,Chin Poo Lee, Kian Ming Lim ,Muiltimedia University,Malaysia

3."The k-means Algorithm: A Comprehensive Survey and Performance Evaluation"

The paper made experimental analysis evaluated the performance of k means clustering algorithm,and proposing where it serves the best results i.e differing and being dependent on the nature dataset.
Authors: Mohiuddin Ahmed, Raihan Seraj, Syed Mohammed Shamsul Islam

4.*"Sentiment analysis of product reviews: A review "*

The paper made experimental analysis by doing sentimental analysis on various products reviews and proposes Support Vector Machine giving the high accuracy.
Author: T.K.Shivaprasad, Jyothi Shetty, NMAM Institute of Technology
Published in: 2017 International Conference on Inventive Communication and Computational Technologies (ICICCT)

5."A review on sentiment analysis and emotion detection from text"

This paper concludes that the lexicon based technique performs well in both sentiment and emotional analysis.
Authors: Nandwani, Rupali Verma

6."VADER: A Parsimonious Rule-based Model for Sentiment Analysis of Social Media Text"

This paper concludes how much is good to use the vader scores for sentimental analysis
Authors:C.J.Hutto, Eric Gilbert ,Georgia Institute Of Technology

7."An overview and comparison of free Python libraries for data mining and big data analysis"

This paper says about how much python is best language in the data -analysis and machine learning
Authors: Stacin, Jovic

1.3 ORGANIZATION OF THE REPORT

The rest of the report is organized as follows.

Chapter 2 describes the software requirements specifications.

Chapter 3 describes and shows the overall Architectural design of the system that has been developed.

Chapter 4 provides the results obtained by the system and the performance analysis of the system.

Chapter 5 deals with the concluding remarks about the project and the future areas of development in the project.

CHAPTER 2

SOFTWARE REQUIREMENTS SPECIFICATION

2.1 SOFTWARE SPECIFICATION

- **Python**: Code is written in python. Compatible version (3.x)
- **Bs4:** Beautiful Soup. Used for HTML parsing.
- **Selenium**: Used for browser automation.
- **Web Driver (Chrome)**: Used to automate browser automations for the Google Chrome web browser

- **LIBRARIES INCLUDED:**

1)csv

2)selenium

3)pandas

4)re

5)matplotlib.pyplot

6)sklearn.cluster

7)os

8)transformers

9)nltk

10)libraries

11)strings

CHAPTER 3

SYSTEM DESIGN

3.1 SYSTEM ARCHITECTURE:-

a. *Data Scraping:*

It includes a web scraper for Amazon product search results. It takes a user-entered product as input, uses BeautifulSoup and Selenium to scrape data from Amazon's search results pages, and then saves the extracted information, including product descriptions, prices, ratings, review counts, and URLs, into a CSV file named 'font67.csv'. The script iterates through multiple pages of search results and ensures that the extracted data is structured and saved for analysis or other purposes. It's important to use such web scraping scripts responsibly and in compliance with the website's terms of service and legal requirements.

b. *User specifications:*

It loads data from a CSV file named 'font67.csv' and filters out rows with missing values in specified columns like 'Description', 'Price', 'rating', 'reviewcount', and 'url'.It allows the user to input multiple keywords for product descriptions and filters the data based on these keywords.

c. *Optimum price range calculation:(K MEANS CLUSTERING)*

It calculates the distribution of review counts and divides the data into ranges. It performs clustering on the 'Price' column to identify clusters with the most data points. i.e., which are classified by k-means clustering algorithm considering products with high review counts and grouping them into ranges and considering the price ranges where most of the products lie in ,and filtering the datas in a separate csv file and proceeding with the next process for filtering out.

d. *Scraping rating's percentage and filtering:*

It uses Selenium to scrape additional data from Amazon product pages, including rating percentages for 5-star and 4-star ratings.It removes duplicate rows from the data based on the 'Description' column and saves the deduplicated data to 'filtered_data9_deduplicated.csv'.It combines the scraped data with the original data, including rating percentages and other information.It computes a combined score for each row based on selected columns like '5star', '4star', 'reviewcount', and 'o_rating'.It sorts the data based on the combined score in descending order.It saves the sorted data to a new CSV file named 'sorted_data1.csv'.

e. *Reviews Scraping:*

The script defines functions to scrape regular reviews and critical reviews from Amazon product pages.
It extracts review text by navigating to review pages, scraping the reviews, and handling pagination.The script loads product data from a CSV file ('sorted_data1.csv') containing product descriptions and URLs.
It creates a list ('products_data') to store product information.It iterates through the product data and for each product, collects its reviews (both regular and critical).
The reviews are written to separate CSV files ('merix1.csv' for regular reviews and 'scrapzz.csv' for critical reviews).

f. Sentimental Analysis:

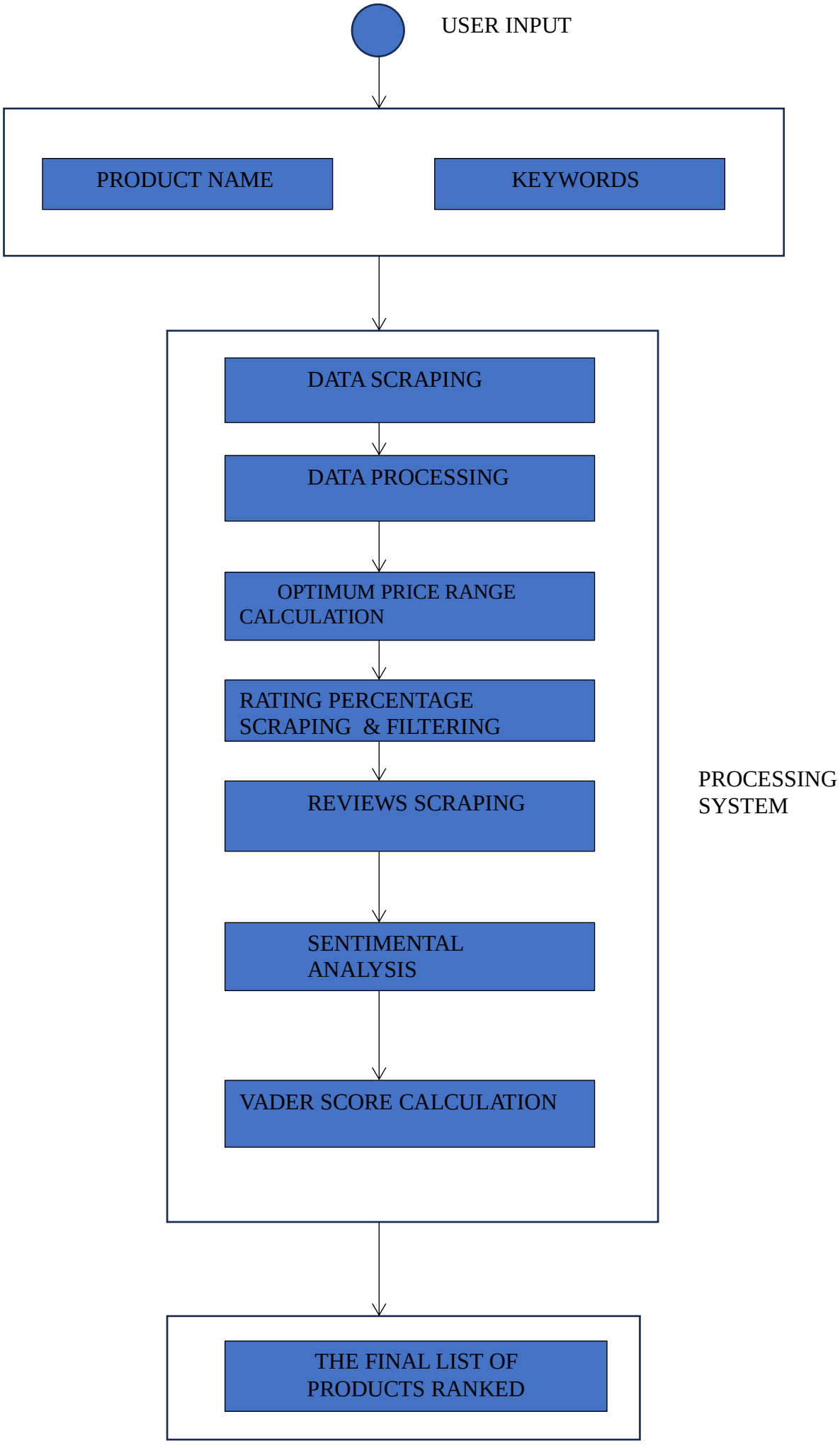
The Hugging Face Transformers library is used to perform sentiment analysis on the scraped reviews.
A pre-trained model ("SamLowe/roberta-base-go_emotions") is employed to classify the sentiments within the text.It analyses and counts the sentiment labels present in the text data from the CSV file, providing insights into the distribution of sentiments within the text.

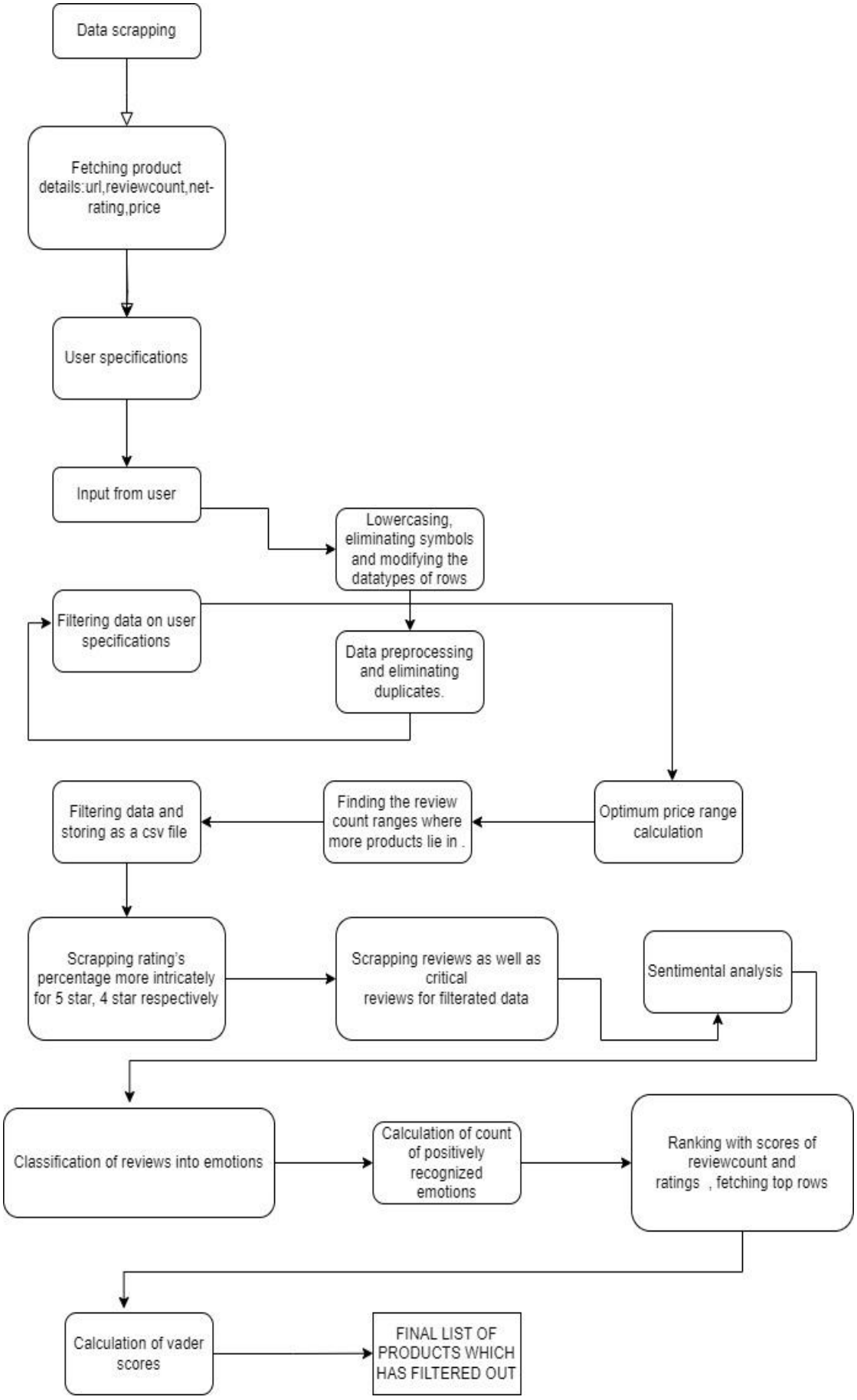
g. *Vader Score Calculation:*

NLTK resources such as a tokenizer, stopwords, a lemmatizer, and the VADER sentiment intensity analyzer are initialized.
The code reads a CSV file ('merix1.csv') and preprocesses the text in each cell.
Sentiment analysis using VADER is performed on the preprocessed words, and the compound scores are summed for each cell.
The summed sentiment scores are printed for each row.

h. Final Output:

Based on VADER score, final output is displayed. The product with higher VADER Score is the best recommended and considered by a lot of people as Amazon users.


USER INPUT
PRODUCT NAME
KEYWORDS
DATA SCRAPING
DATA PROCESSING
OPTIMUM PRICE RANGE CALCULATION
RATING PERCENTAGE SCRAPING & FILTERING
REVIEWS SCRAPING
SENTIMENTAL ANALYSIS
VADER SCORE CALCULATION
PROCESSING SYSTEM
THE FINAL LIST OF PRODUCTS RANKED

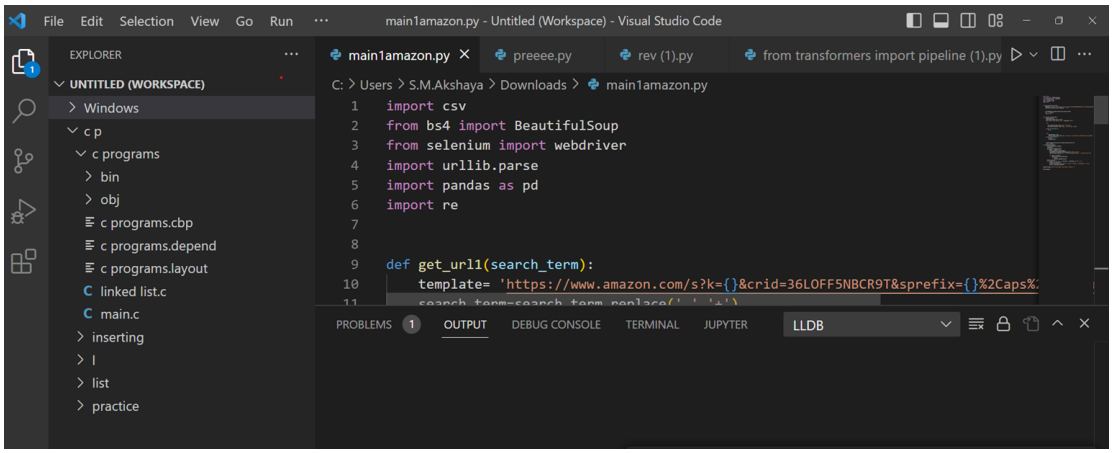

Data scrapping
Fetching product details:url,reviewcount,net-rating,price
User specifications
Input from user
Lowercasing, eliminating symbols and modifying the datatypes of rows
Filtering data on user specifications
Data preprocessing and eliminating duplicates.
Optimum price range calculation
Finding the review count ranges where more products lie in .
Filtering data and storing as a csv file
Scrapping rating's percentage more intricately for 5 star, 4 star respectively
Scrapping reviews as well as critical reviews for filterated data
Sentimental analysis
Classification of reviews into emotions
Calculation of count of positively recognized emotions
Ranking with scores of reviewcount and ratings , fetching top rows
Calculation of vader scores
FINAL LIST OF PRODUCTS WHICH HAS FILTERED OUT

IMPLEMENTATION AND RESULTS

4.1 IMPLEMENTATION

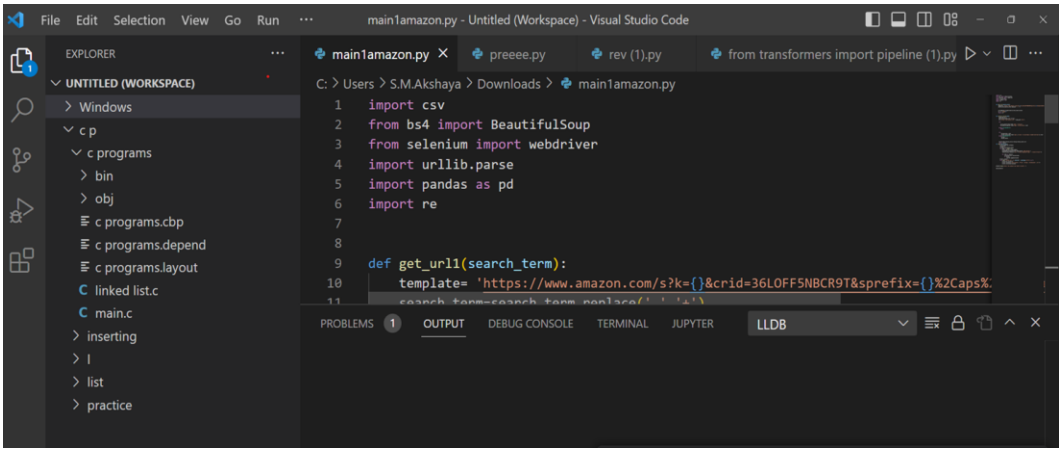
4.1.1 Web Scraping:

a.Importing Libraries:



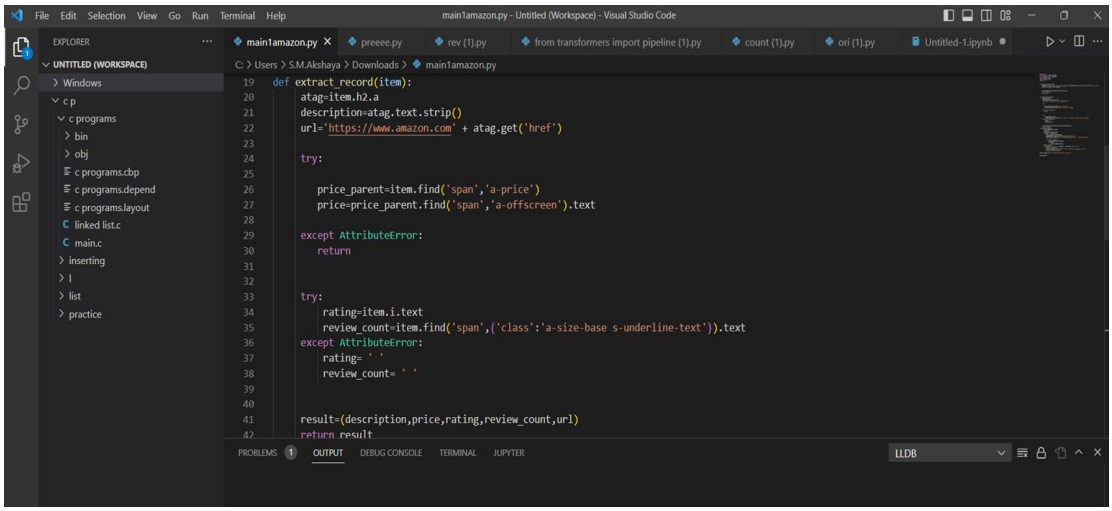
Importing all the necessary libraries including: csv for saving the data in excel, BeautifulSoup for parsing a web page and Selenium for web automation and a driver has been installed for the Google chrome browser and pandas for data manipulation and analysis and finally importing re ‘regular expressions’ modules.

b.Define Function **“get_url1(search_term)”**



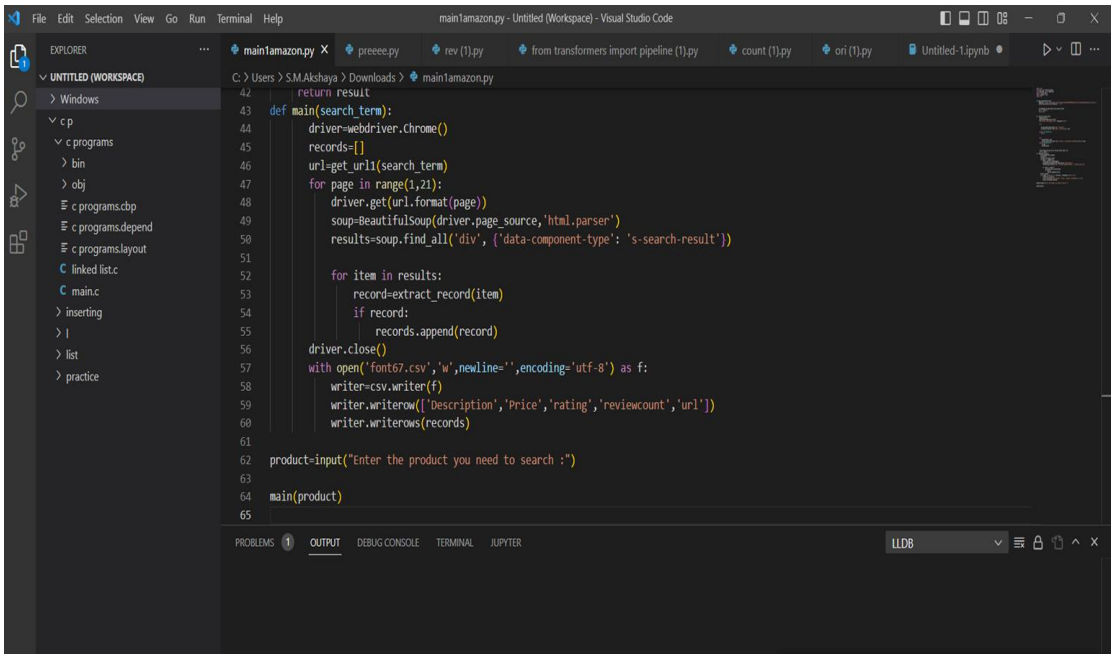
Defining a user-defined function named get_url1(search_term) so this helps to process the target url, Where the user specifies the product name, thus we add that as a substring to the common url and hence we get the target url which helps to direct the appropriate site containing the product details.

c. Define function **“extract_record(item)”**:



This function used to scrap the datas :”description”, “price”,”url”,“review count” and the overall “rating” of the product.

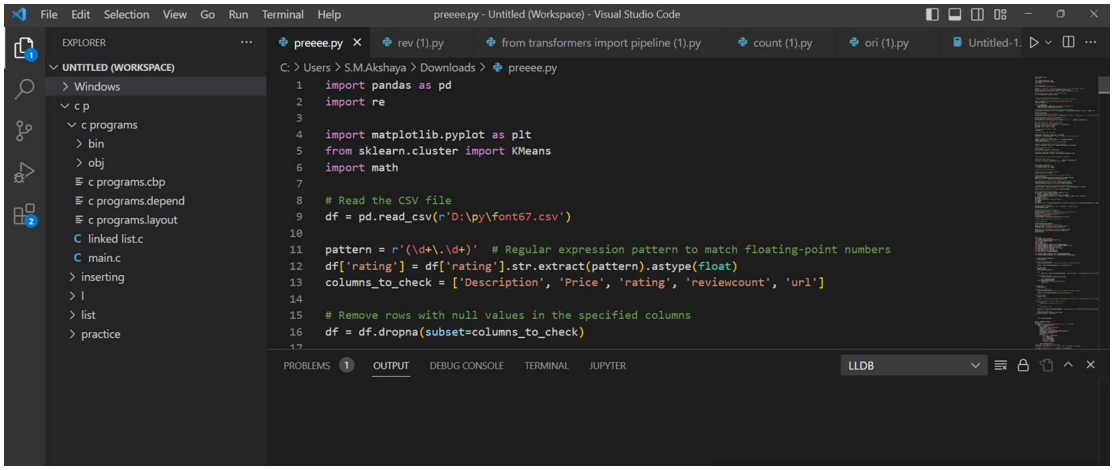
d.Define the main function **“main(search_item)”**:



This function main almost scraps a maximum of 21 pages of details containing product details
and storing all the datas in a excel sheet, this traverses to each and every page and scraps one by one.

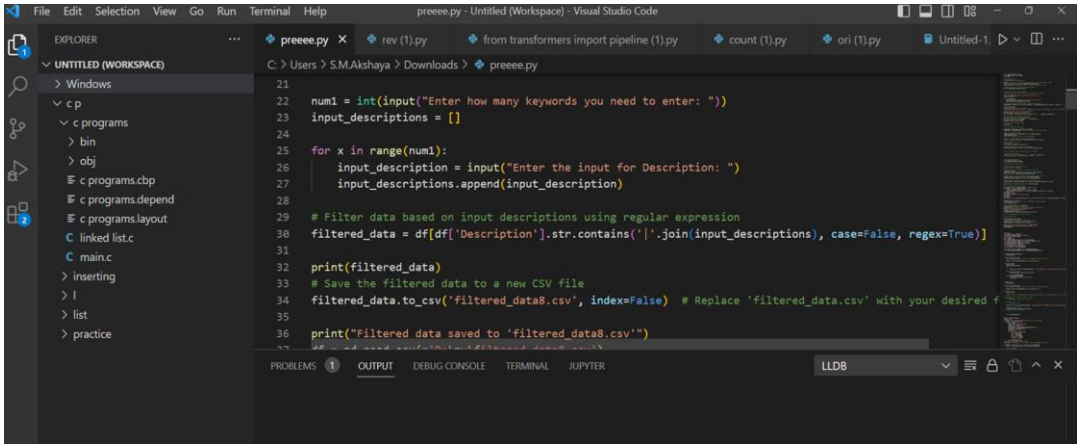
4.1.2 Data Filtering and Price clustering: -

a.Data Loading and Preprocessing:



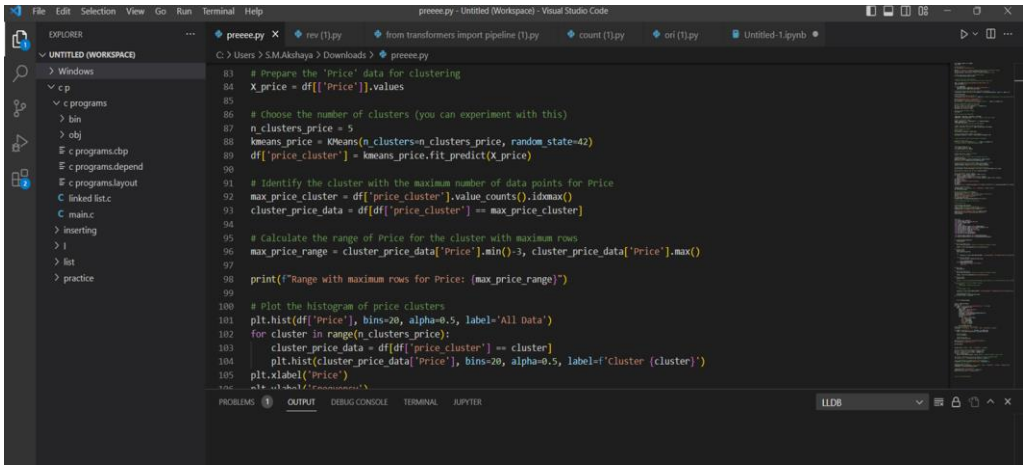
Preprocessing the datas that has been already stored to “font67.csv” and converting them to the appropriate data types that helps in the further processing.

b.User Input for **“FILTERING”:**



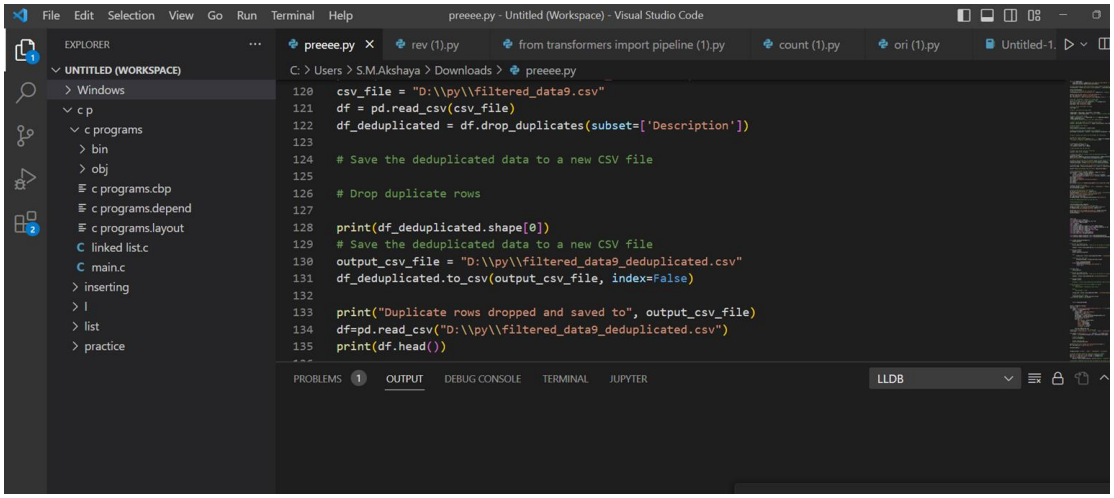
This part asks the user to enter the number of keywords and the keywords also thus using these keywords , we check up in all the product’s description which contain either of these keywords specified and filtering accordingly.

c.Price Clustering:



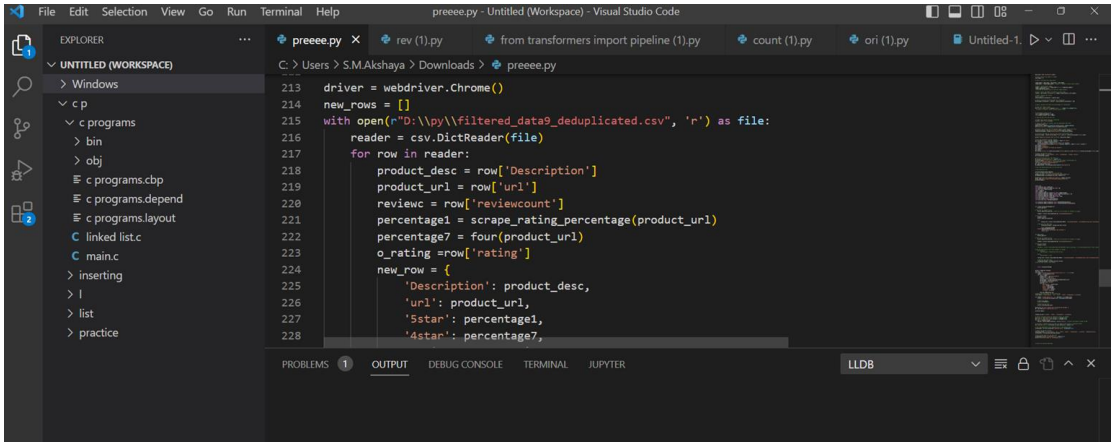
First grouping them based on the review count and creating 5 ranges of rc's and counting the number as well as percentages of products falling to each range.Using the Scikit-Learn library's KMeans clustering algorithm to cluster data based on the 'Price' column into a specified number of clusters (n_clusters_price).Identifying the cluster with the maximum number of data points and calculating the price range for the cluster with the maximum rows.

d.Data deduplication:



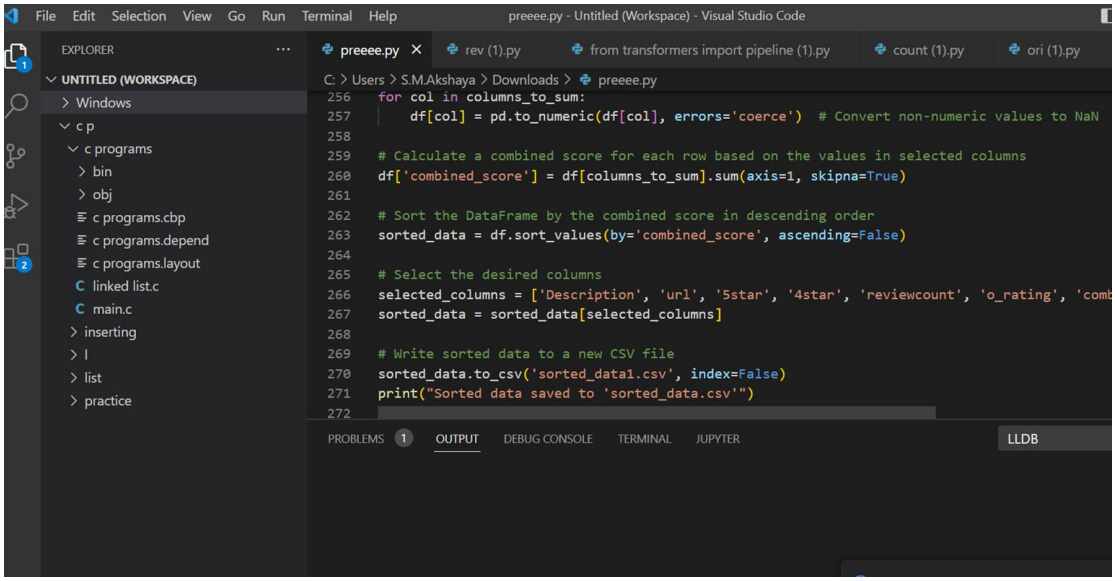
Checking if any duplicate data rows are present and eliminating the duplicates

e.Scrapping of rating percentages:



Now scrapping the individual rating percentages including 5 star and 4 star and thus this helps in filtering the more product intricately.

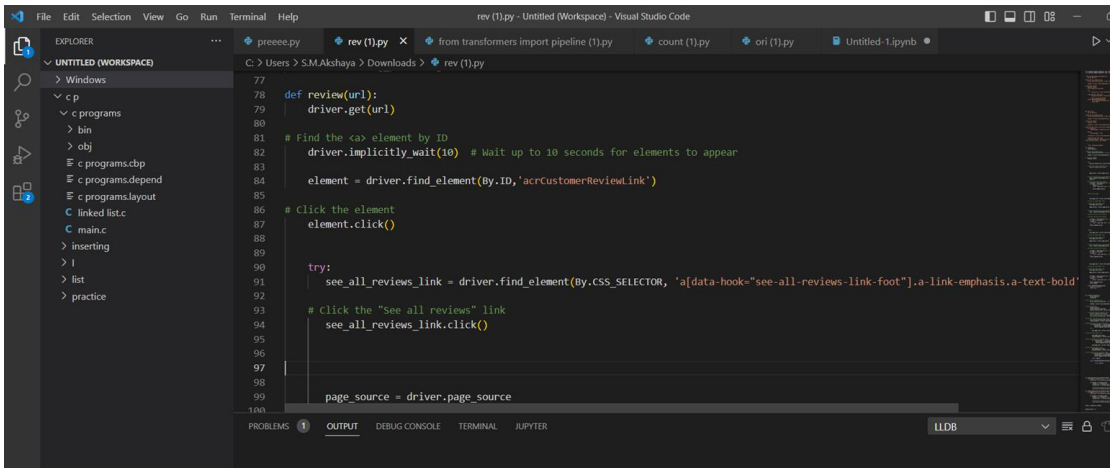
f.Combining and Sorting data:



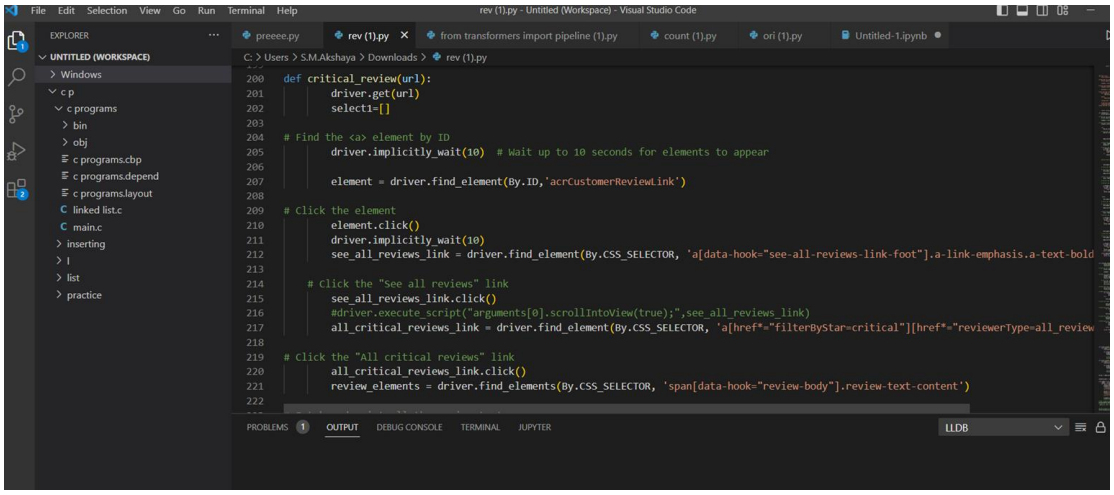
Now finding the combined score of reviewcount , 5 star , 4 star and calculating the overall score(summation of three values) which helps in the ranking of the products hence , this process helps to filter the product and saving into new csv file .

4.1.3 Scraping Reviews:

a.Web Scraping function:



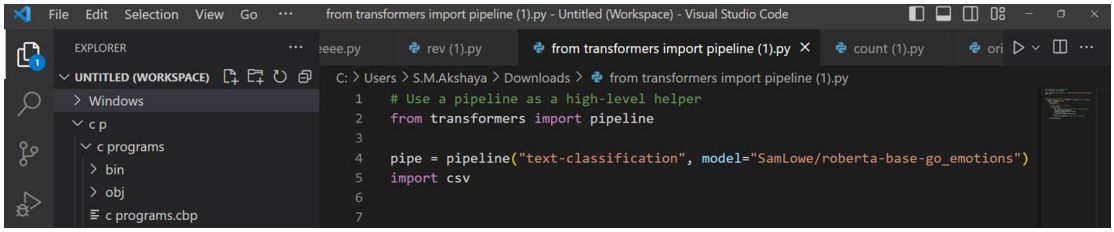
Now scrapping the reviews of the filtered data and this almost 50 -60 reviews of each product and get stored on a csv file.



This helps in the scrapping of reviews which are categorized under a tag named “critical reviews”

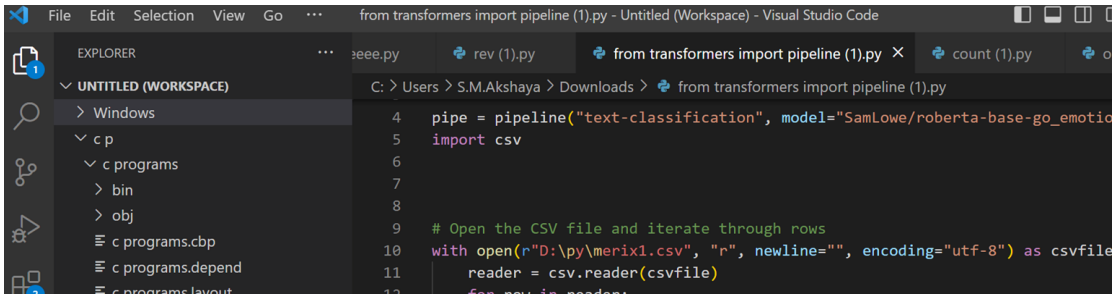
4.1.4 Sentimental Analysis:

a.Pipeline Creation:



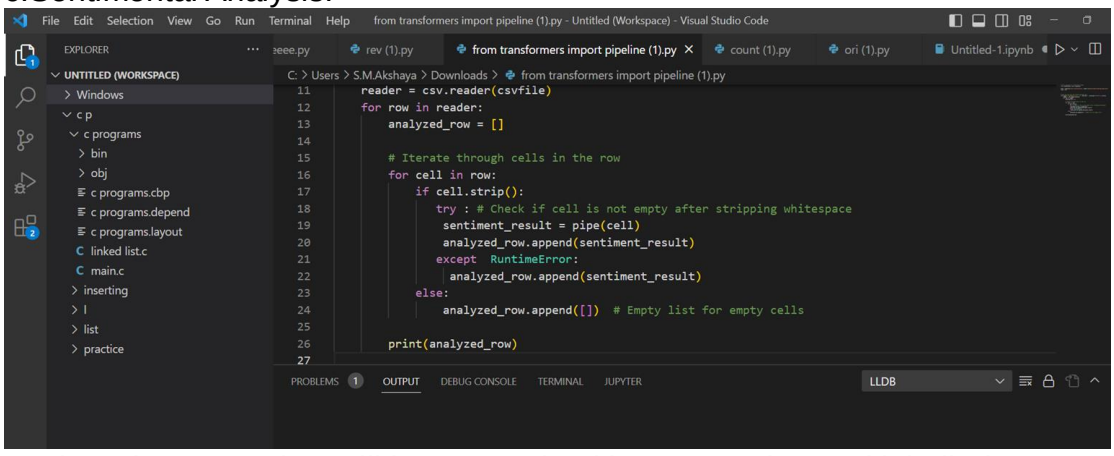
Pretrained Roberta model has been used which has linked using application interface using hugging face , thus which has a ability to categorize the reviews to several emotion tag namely joy, sad, disappointment and more on..,

b.Csv Processing:



Saving the reviews in a csv file .

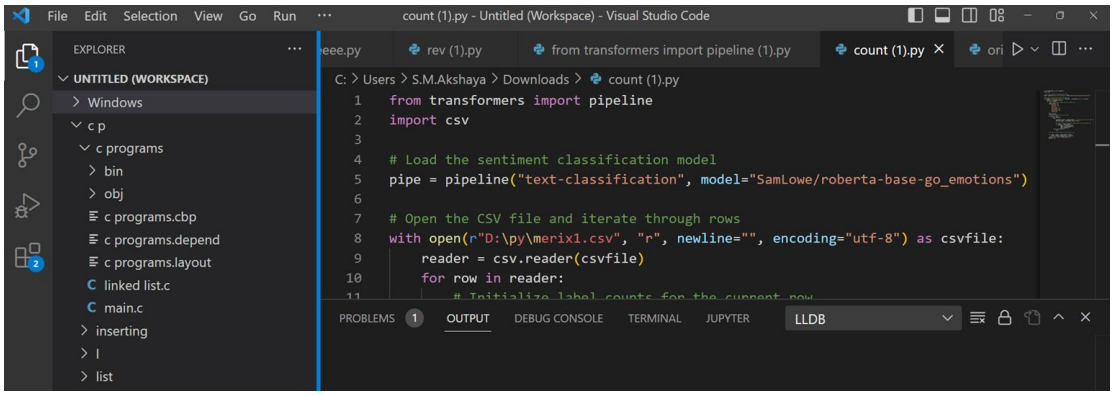
c.Sentimental Analysis:



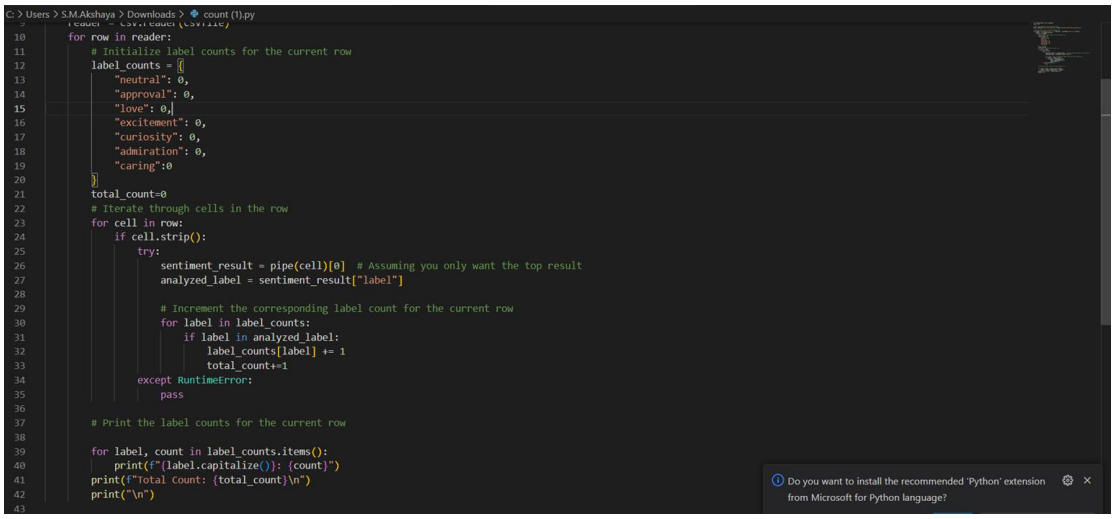
Each and every review will be categorized into a emotion and labelled

4.1.5 Label Counting of reviews using a pre-trained Model:

a.Sentimental Analysis Pipeline:



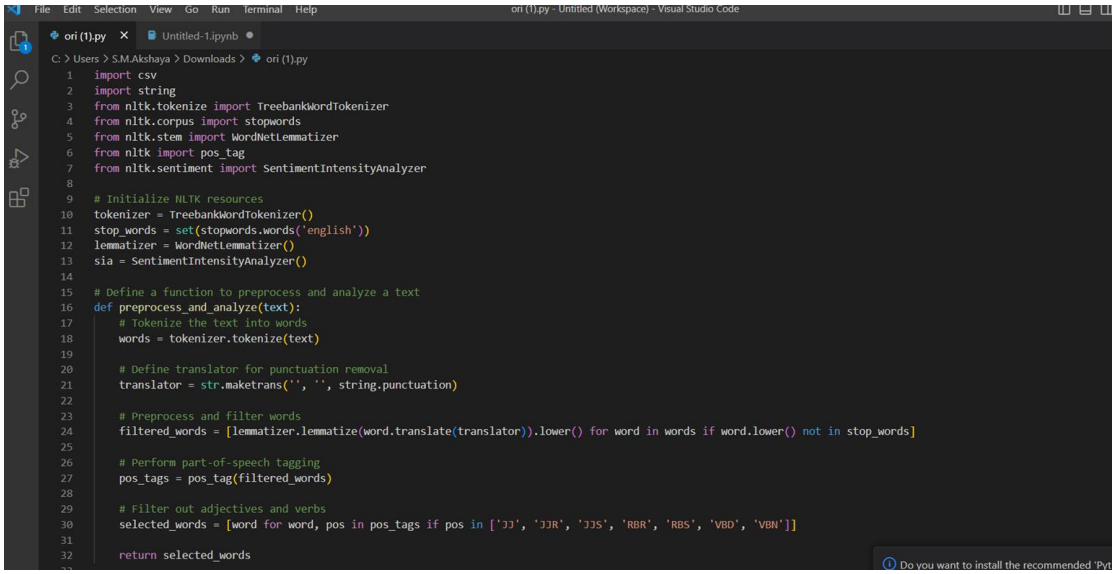
b.Label Counts and Analysis:



Counting the number of happy emotions that got labelled as a whole for every product.

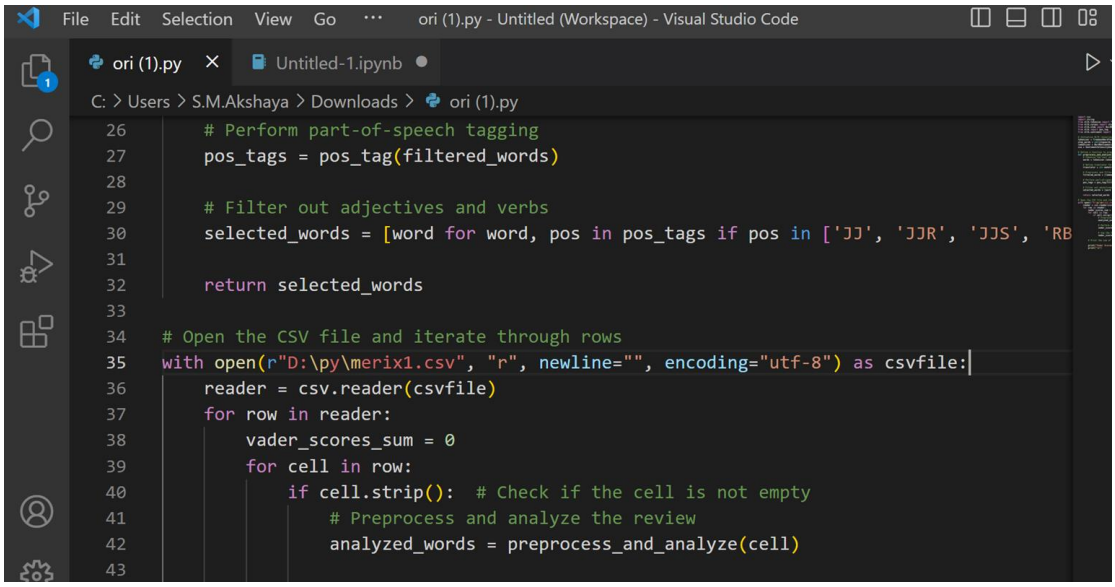
4.1.6 Summarized Sentiment Scoring of reviews

a.Data Reading and Preparation:



```python
import csv
import string
from nltk.tokenize import TreebankWordTokenizer
from nltk.corpus import stopwords
from nltk.stem import WordNetLemmatizer
from nltk import pos_tag
from nltk.sentiment import SentimentIntensityAnalyzer

# Initialize NLTK resources
tokenizer = TreebankWordTokenizer()
stop_words = set(stopwords.words('english'))
lemmatizer = WordNetLemmatizer()
sia = SentimentIntensityAnalyzer()

# Define a function to preprocess and analyze a text
def preprocess_and_analyze(text):
    # Tokenize the text into words
    words = tokenizer.tokenize(text)

    # Define translator for punctuation removal
    translator = str.maketrans('', '', string.punctuation)

    # Preprocess and filter words
    filtered_words = [lemmatizer.lemmatize(word.translate(translator)).lower() for word in words if word.lower() not in stop_words]

    # Perform part-of-speech tagging
    pos_tags = pos_tag(filtered_words)

    # Filter out adjectives and verbs
    selected_words = [word for word, pos in pos_tags if pos in ['JJ', 'JJR', 'JJS', 'RBR', 'RBS', 'VBD', 'VBN']]

    return selected_words
```

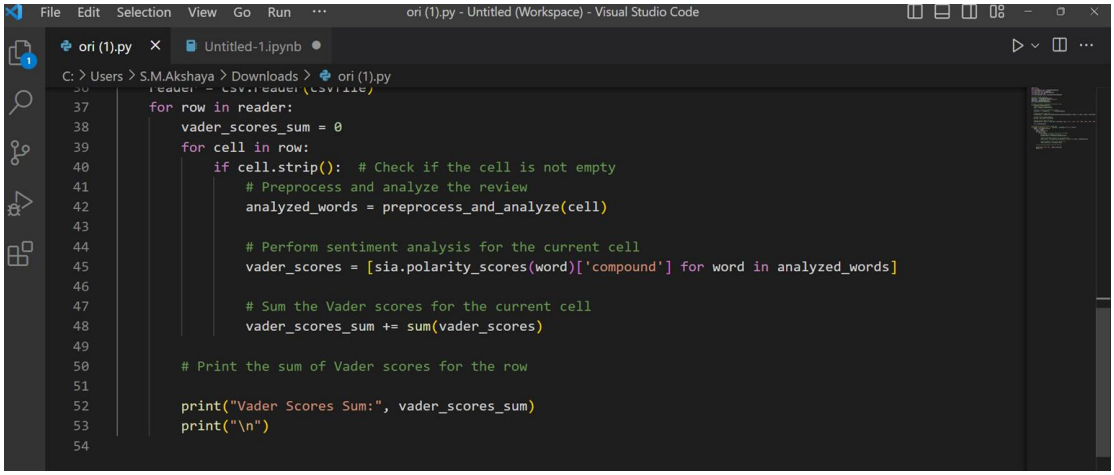
b.Text Preprocessing:



```python
    # Perform part-of-speech tagging
    pos_tags = pos_tag(filtered_words)

    # Filter out adjectives and verbs
    selected_words = [word for word, pos in pos_tags if pos in ['JJ', 'JJR', 'JJS', 'RB

    return selected_words

# Open the CSV file and iterate through rows
with open(r"D:\py\merix1.csv", "r", newline="", encoding="utf-8") as csvfile:
    reader = csv.reader(csvfile)
    for row in reader:
        vader_scores_sum = 0
        for cell in row:
            if cell.strip():  # Check if the cell is not empty
                # Preprocess and analyze the review
                analyzed_words = preprocess_and_analyze(cell)
```

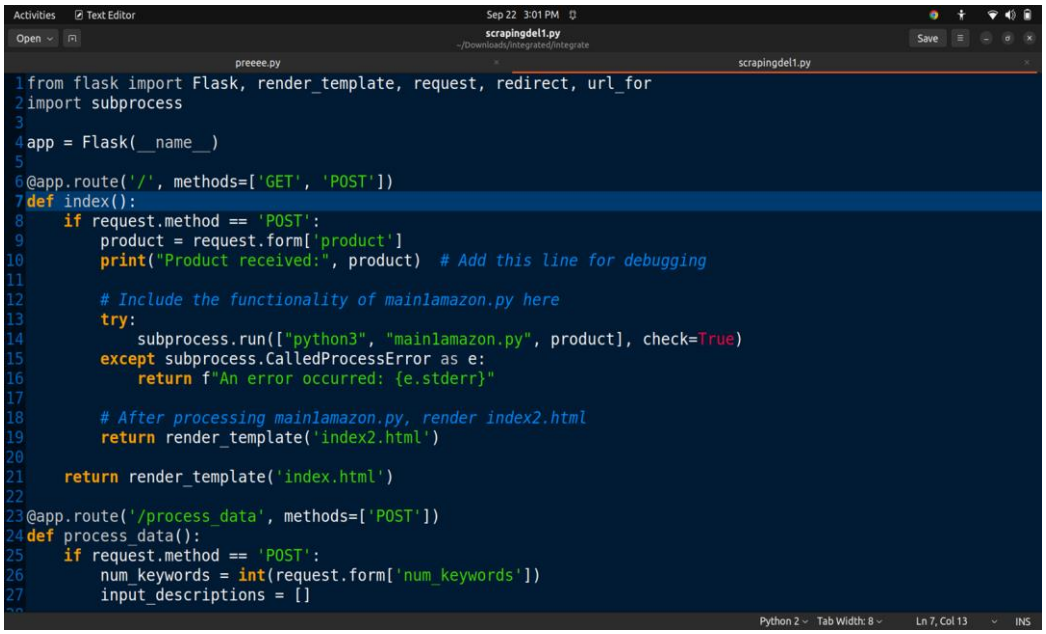
Using tokenizer and pos_tag which helps in filtering out the part of speech that are mentioned and we are eliminating the stopwords that are not necessary in the sentimental analysis.

c.Sentiment Analysis:



Finally to choose , we use the vader score which makes a score out of the sentimental analysis hence the scores are made to calculate and find the sum of vader scores of each row which contains all the reviews of a particular product.

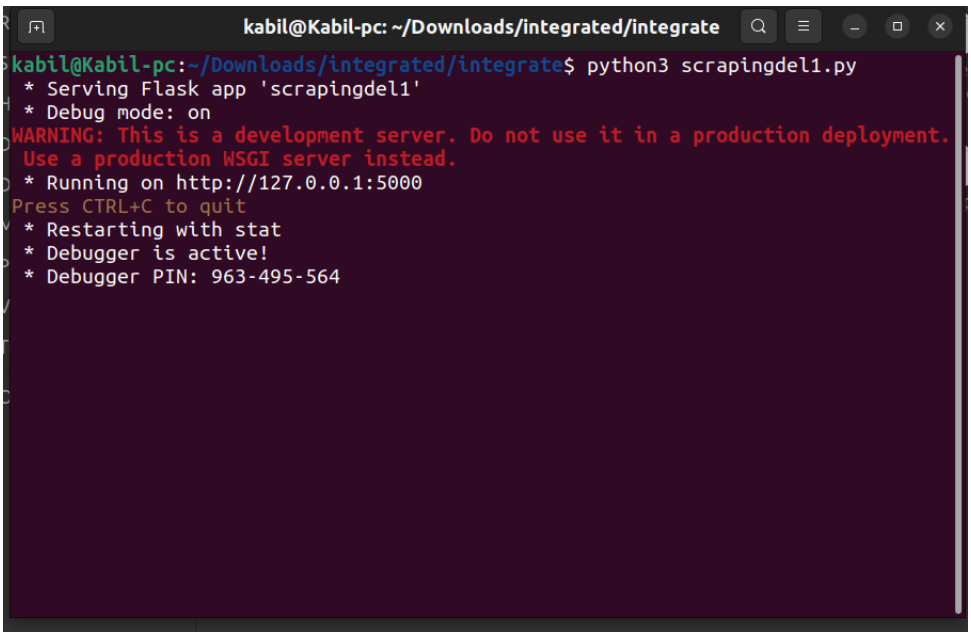
4.1.7 Frontend using Flask:



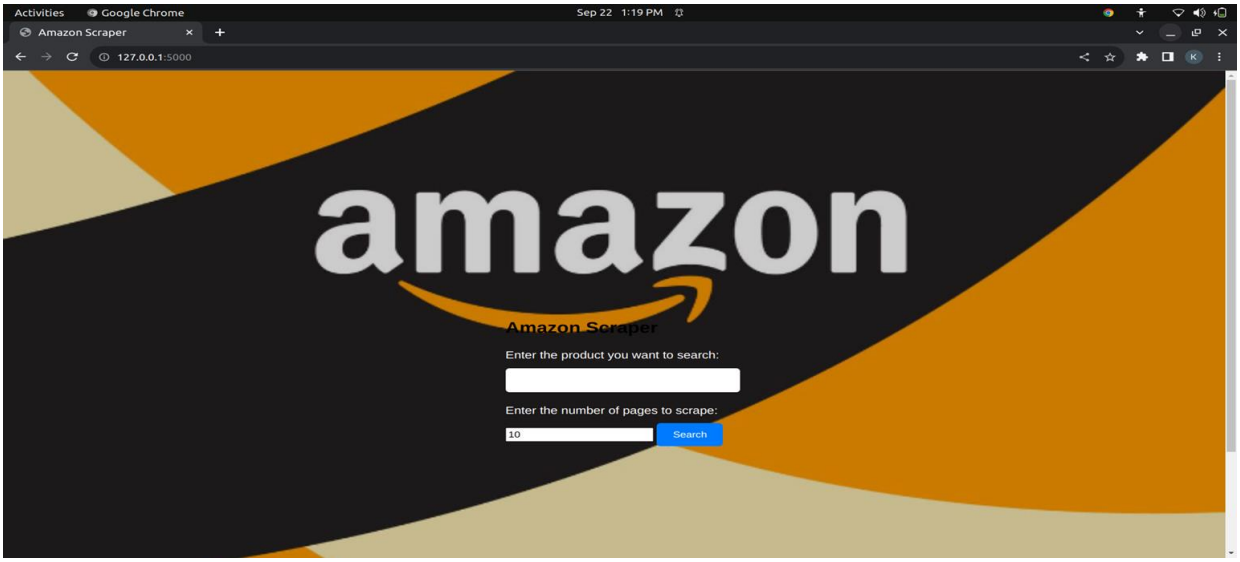
Finally to integrate all these backend with frontend website,Flask has been used.

4.2 RESULTS

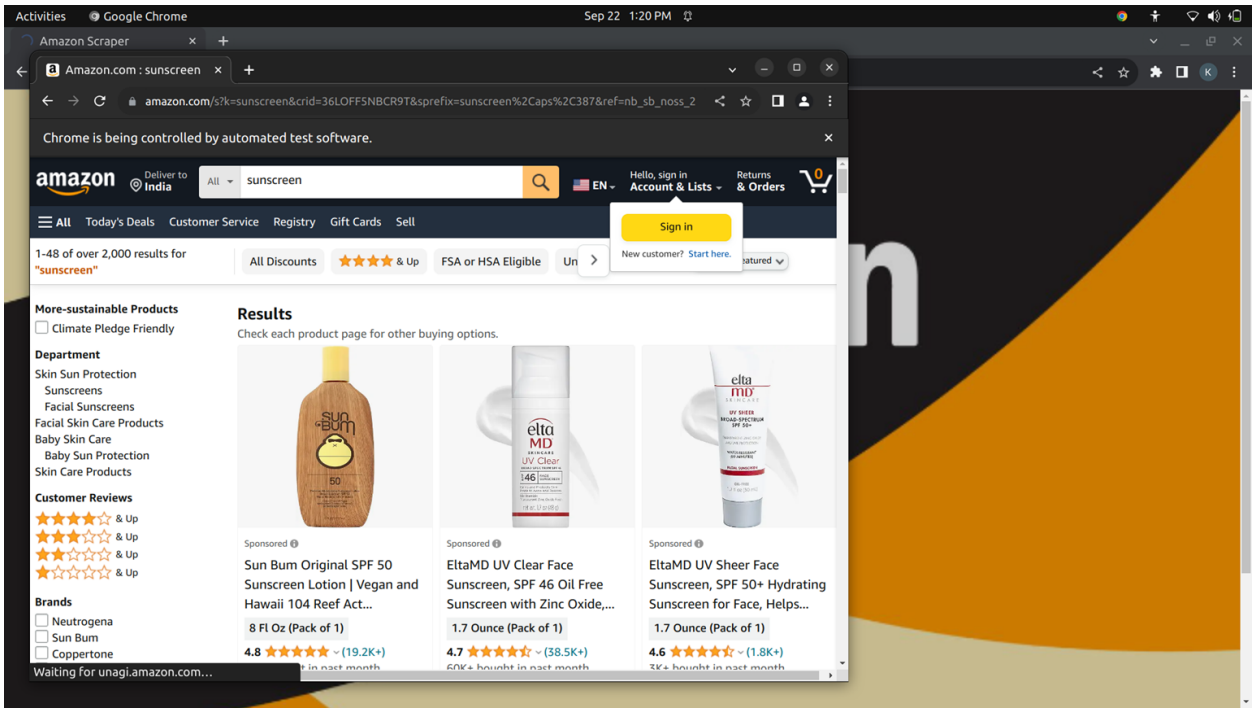
4.2.1 Web Scraping:-



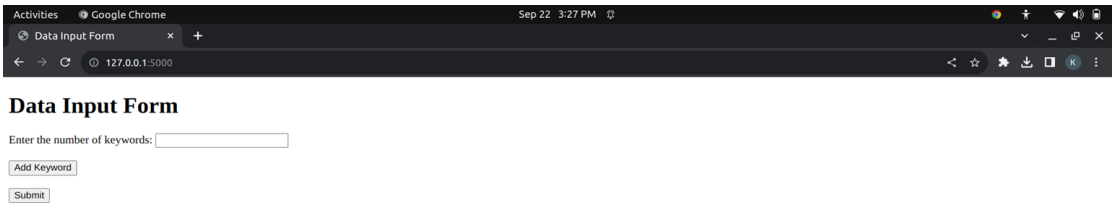
We have used the linux environment to the run the program, thus starting with the scrapping the data from the amazon site.



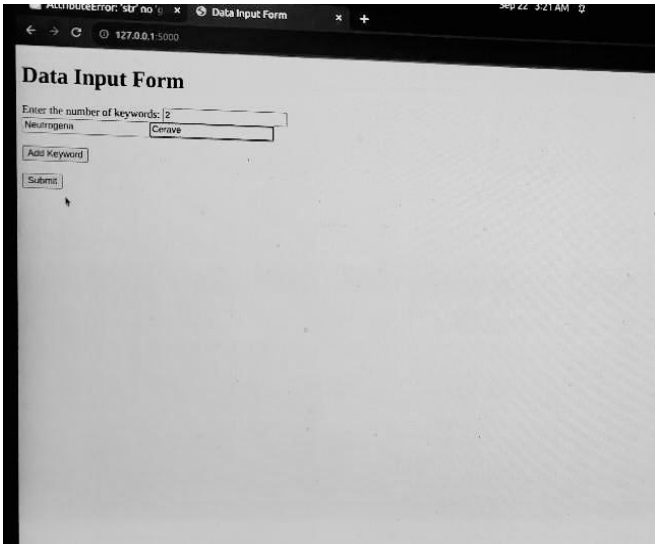
The front-end that we have incorporated with the flask which deploys where the user gives the input of the product name which is fed in input of amazon's search box.



This shows the scrapping of data which almost scraps nearly 1200 rows of data approximately, fetching the product's description, url, overall rating , reviewcount and the overall rating of the products , this almost scraps the datas of 21 pages from the Amazon's site.

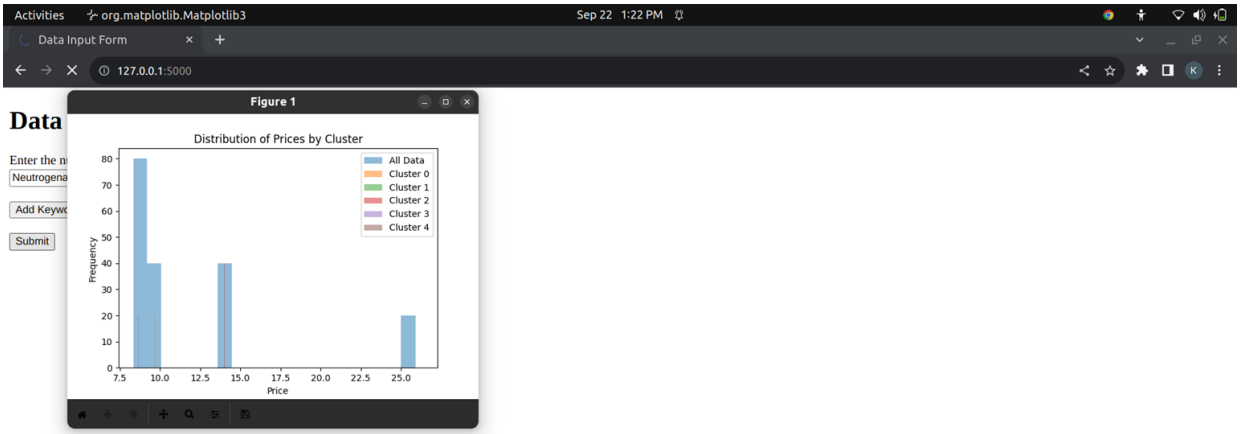


This page asks the user for the product's specifications as inputs by giving the number of inputs as a first step, where the code filters and saves the data rows accordingly to the given specifications.
And hence we use the saved present file for the upcoming processes.



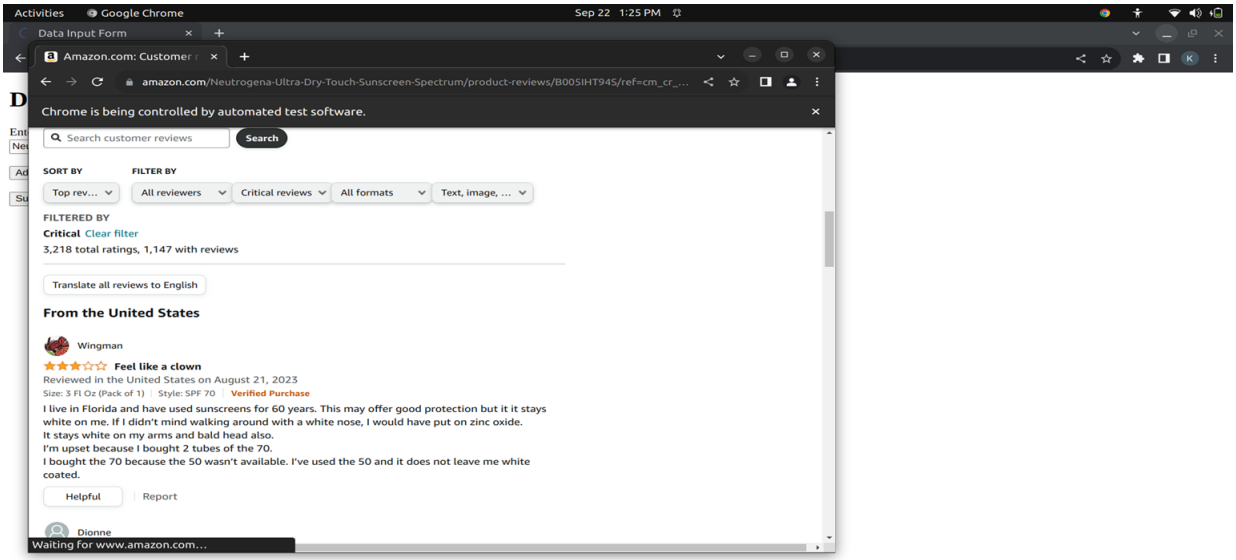
User specifying the inputs

4.2.3 Distribution of Prices:-



This picture shows the distributions of prices visually in graphs and which are classified by k-means clustering algorithm considering products with high review counts and grouping them into ranges and considering the price ranges where most of the products lie in ,and filtering the datas in a separate csv and proceeding with the next process for filtering out.

4.2.4 Review Scrapping:-

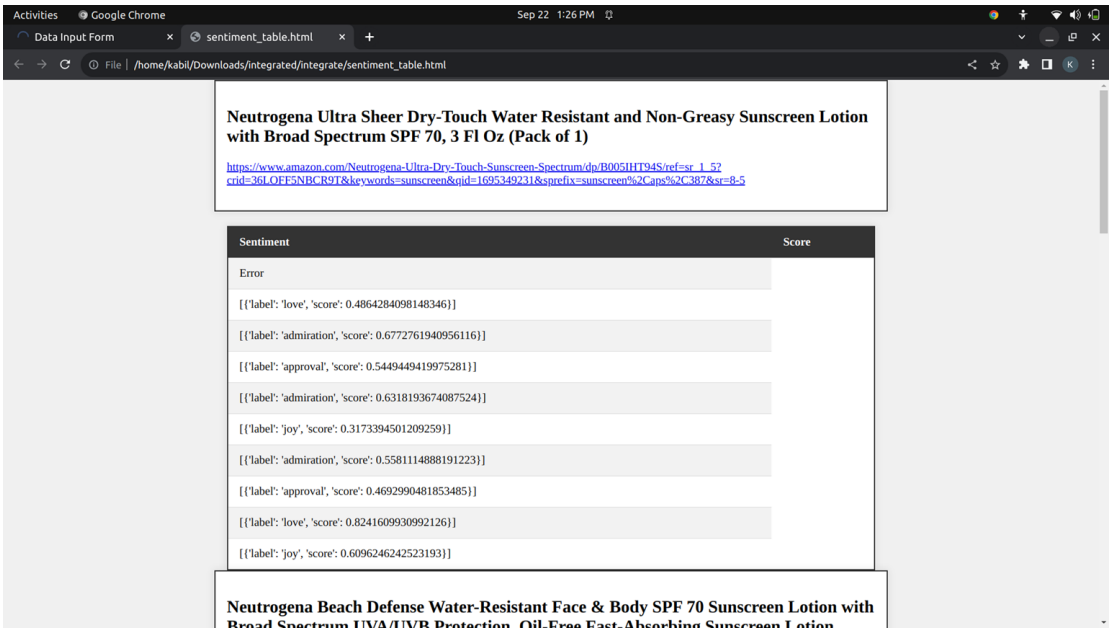


Scrapping all the reviews , even the critical reviews which are already categorized by Amazon and saving them in a csv , we scrap almost 40-50 reviews for each product and classifying them into emotion using the Roberta pretrained model which classifies them into emotions : admiration, amusement, anger,annoyance,approval, caring, desire, confusion, curiosity,disgust, excitement, disappointment,joy, love and more on.

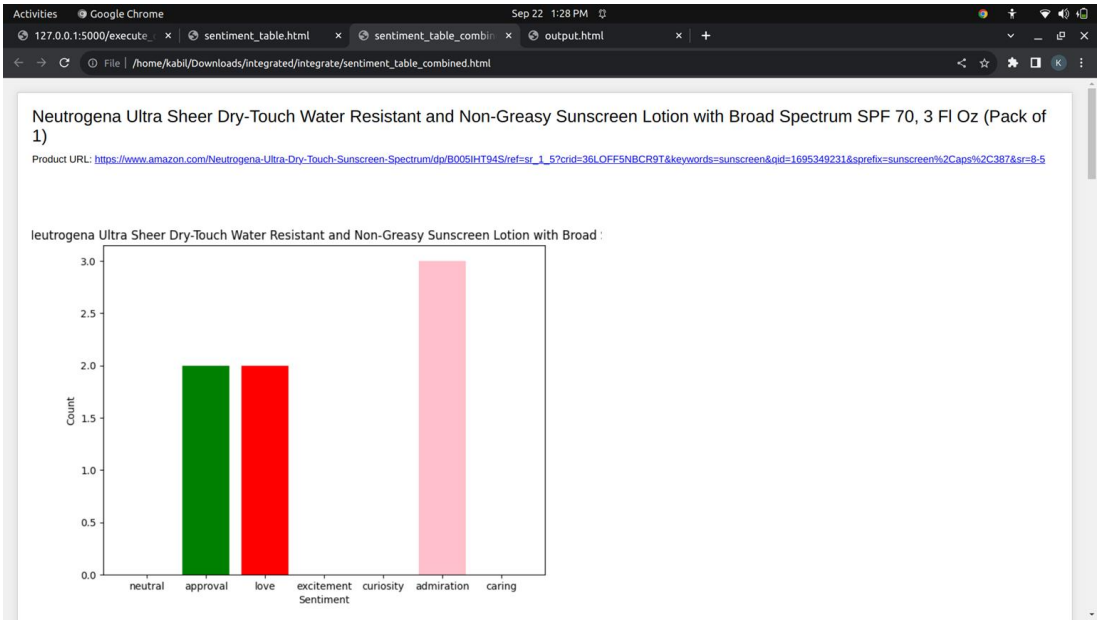
And counting the happy emotions alone i.e., neutral,approval,love,excitement,curiosity, admiration,caring for each product and classifying them which gets the higher score.

And finally taking the reviews of those filtered and removing all the stop words and using lemmatizers and tokenizers from the “Natural Language Tool kit” and even performing part of speech tagging with the reviews i.e,.taking /considering all the adjectives and verbs and calculating the vader sum and hence by comparing the resultant vader sum with all the products , we get the best recommended product that many people were happy about it and considering using it.

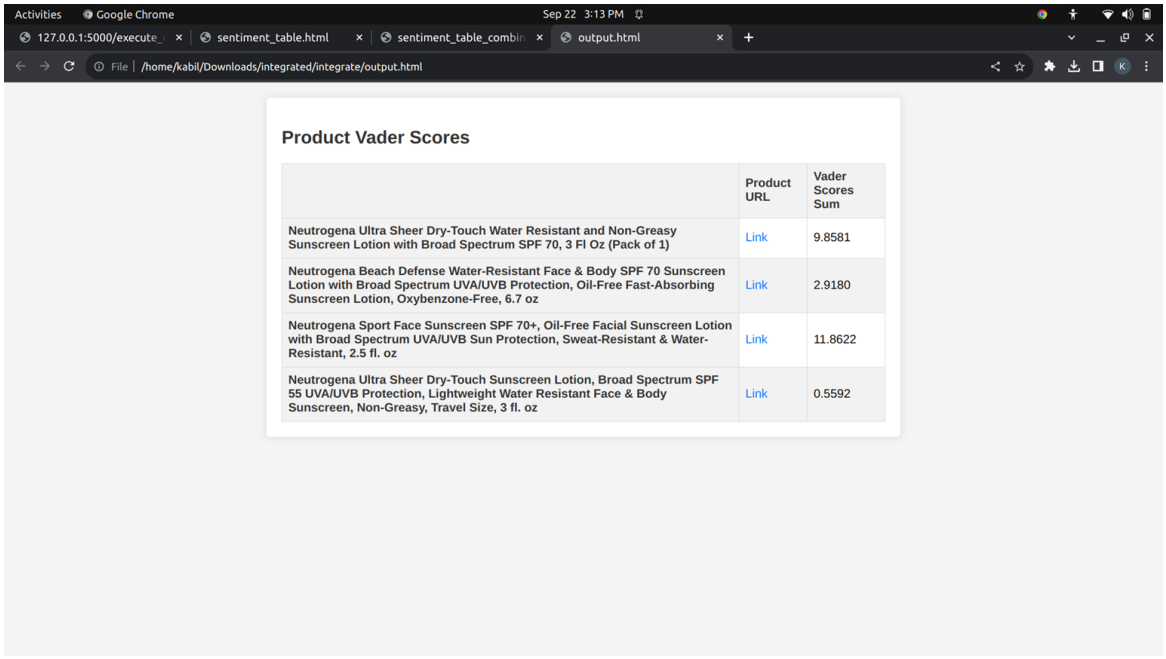
4.2.5 Final output:-



This shows the labelled classification of emotions using Roberta Model.



This shows that admiration got labelled on more number of reviews.



The shown vader scores are sum of the review's it is , by which we can conclude that the higher vader score id ,being great the product is with positive appreciation given by users.

5.1 CONCLUSION:

The usage of webscrapping by the automation of browser which in order collects the data at that instance available in the e-commerce site , and the data can be fetched and analysed at any time with the usage of webscrapper language comprising Selenium and Beautiful Soup with python.

After several filtrations of data containing the details of product and reviews , we are able to get the best products finalised list based on the ranking and vader scores .

Hence ,this resulting in an application which can be used to find the best product url list according to the specifications given as the input and produces results with vaderscores (higher the vader scores are implies that higher the good product it is with more good reviews).

This application saves the user a lot of time and one does not want spend hours on going through the list of product available on the e-commerce sites, searching for the desired they need with good reviews and meeting within their budget.

Here are some results given by our model:-

1)Sunscreen: 4.96 (Cerave)
2)Laptop: 4.24 (Lenovo)

5.2 FUTURE WORK:

Collecting product data from Amazon through web scraping and then determining the best product URL based on ratings and sentiment analysis of reviews has several potential applications and future prospects:

1)Competitor Analysis: You can track competitors' product ratings and reviews over time to gain insights into market trends and how they are perceived by customers.

2) Product Quality Assessment: Manufacturers and sellers can use this data to assess the quality of their products by analyzing customer feedback and making necessary improvements.

3)Customized Shopping Experiences: Leverage the data to offer personalized shopping experiences, suggesting products based on individual preferences and reviews they've given in the past.

4) Sentiment-Driven Marketing Campaigns: Develop marketing strategies based on sentiment analysis. For example, launch targeted advertising campaigns for products with overwhelmingly positive reviews and ratings.

5)Trend Forecasting: Analyze Amazon product reviews to identify emerging trends and consumer preferences. This information can be valuable for retailers and manufacturers planning their product lines.

6) Price Comparison Websites: You can use this data to build or enhance price comparison websites. Users often seek the best deals, and your analysis can help them find not only the cheapest products but also those with high ratings and positive sentiment in reviews.

7) Market Research: Analyzing sentiment from product reviews can provide insights into customer opinions and preferences. Companies can use this data to improve their products, marketing strategies, and customer service.

5.3References:

1."EFFICIENT SCRAPING OF DATA FROM WEBSITES USING SELENIUM"

Authors: [1]Shreya V. Dhoke student ,[2] Anupama D. Sakhare Asst Prof,[3] Satish J.Sharma Prof.

1,2,3 Dept of Electronics and Computer Science, Rakshtrasant Tukdoji Maharaj Nagpur University,India . 2022 jetir, volume 9, issue 6,issn:2349-5162

https://www.jetir.org/papers/JETIRFM06063.pdf

2."RoBERTa-GRU: A Hybrid Deep Learning Model for Enhanced Sentiment Analysis"

Authors:Kian Long Tan ,Chin Poo Lee, Kian Ming Lim ,Muiltimedia University,Malaysia

https://doi.org/10.3390/app13063915

3."The k-means Algorithm: A Comprehensive Survey and Performance Evaluation"

Authors: Mohiuddin Ahmed, Raihan Seraj, Syed Mohammed Shamsul Islam
https://www.mdpi.com/2079-9292/9/8/1295

4."Sentiment analysis of product reviews: A review"
Authors: T.K.Shivaprasad, Jyothi Shetty, NMAM Institute of Technology
https://ieeexplore.ieee.org/document/7975207

5."A review on sentiment analysis and emotion detection from text"
Authors: Nandwani, Rupali Verma
https://link.springer.com/article/10.1007/s13278-021-00776-6#article-info

6."VADER: A Parsimonious Rule-based Model for Sentiment Analysis of Social Media Text"
Authors:C.J.Hutto, Eric Gilbert ,Georgia Institute Of Technology
http://eegilbert.org/papers/icwsm14.vader.hutto.pdf

7."An overview and comparison of free Python libraries for data mining and big data analysis"
Authors: Stacin, Jovic
https://ieeexplore.ieee.org/document/8757088

8. *"An Automated Web Application Testing System"*
Authors:Tarek M Mahmoud,Moheb Girgis,Bahgat A.Abudullatif,Alaa Zaki
https://www.researchgate.net/publication/270569315_An_Automated_Web_Application_Testing_System