Step 1: scraping data from web

Please access the data_scraping.ipynb file and proceed by following the steps outlined below

1. **Importing Libraries:**

   - import the necessary libraries at the beginning of your script.
   - requests: Used to send HTTP requests to the Flipkart website.
   - BeautifulSoup: Used for web scraping to parse and extract data from the HTML content.
   - pandas: Used to organize and structure the data.
   - time: Used to introduce a delay between requests to avoid overloading the server.

2. **Initialization of Lists:**

   After that initialize six empty lists:

   - all_reviews
   - all_div_reviews
   - all_ratings.
   - Locations
   - Time
   - Cuctomers

   These lists will store the extracted data.

3. **Looping Through Pages**:

   We start a for loop that iterates from page 1 to page 200 (adjustable range).

   For each page, you construct the URL by appending the page_num variable to the base URL and Sending HTTP Request.

   We used the **requests.get()** method to send an HTTP GET request to the constructed URL.

   The response is stored in the variable r.

4. **Parsing HTML Content**:

   We used **BeautifulSoup** to parse the HTML content of the response using **'html.parser'**.

   The parsed HTML is stored in the variable soup.

5. **Extracting Data**:

   We used **soup.find_all()** to extract reviews, comments, and ratings based on their respective HTML classes. Reviews are stored in the reviews list. Comments are stored in the div_reviews list. Ratings are stored in the all_ratings list. Locations are stored in the Locations. Time are stored in the time. customers are stored in the customers.

6. **Extending Lists:**

We used **.extend()** to add the extracted text from reviews, comments, ratings, Locations, time and customer to their respective lists.

This is done for each page.

7. **Adding Delays:**

   We introduce a 3-second delay using **time.sleep(3)** to avoid sending too many requests too quickly and potentially getting blocked by the server.

8. **Organizing Data:**

   After scraping all the pages, we create a dictionary data that holds the collected **data with keys 'Review', 'Comments', 'Ratings' 'Locations','Cuctomers' and'Time'**

9. **Writing to CSV:**

   we specified a CSV filename, **'reviews_data.csv'**. I open this file in write mode and specify the encoding as 'utf-8'. we create a CSV writer and write the column headers based on the keys in the data dictionary.

Step 2: data cleaning and removing unnecessary data

Please access the data_cleaning.ipynb file and proceed by following the steps outlined below

1. **Importing Libraries:**

   We import the necessary libraries, including pandas, emoji, re, nltk, TextBlob, and SentimentIntensityAnalyzer from the NLTK library.

2. **Reading the CSV:**

   we used pd.read_csv() to read the CSV file named 'reviews_data.csv' into a DataFrame called df.

3. **Removing Emojis:**

   We define a function remove_emojis(text) that uses the emoji library to remove emojis from text.

4. **Downloading NLTK Resources:**

   We download required NLTK resources, including stopwords, the VADER lexicon, and the Punkt tokenizer.

5. **Initializing Variables:**

   We initialize variables, including stop_words (a set of English stopwords) and the VADER sentiment analyzer (vader).

6. **Creating Output DataFrame:**

   We create an empty DataFrame named output_df with columns for various sentiment-related metrics: Review, Comments, Ratings, Locations, Customers and Time Positive Score, Negative Score, Neutral Score, Sentiment, and Subjectivity Score.

7. **Analyzing Sentiment:**

   We define a function analyze_sentiment(text) to analyze the sentiment of a given text. Inside this function.

   We apply a regex pattern to remove unwanted characters and keep only alphabetic characters and spaces.

   We tokenize the cleaned text into words using NLTK's word_tokenize.

   We remove stopwords and convert words to lowercase.

   We calculate the VADER compound score for the cleaned text, as well as polarity and subjectivity scores using TextBlob.

   The function returns these scores.

8. **Iterating Over Rows:**

   we iterate over each row in the input DataFrame df. For each row, you extract the review text, comments text, and ratings. I call analyze_sentiment() on the combined text of the review and comments. Based on the compound score, you classify the sentiment as Positive, Neutral, or Negative .I calculate the Negative Score and Neutral Score accordingly.

9. **Building the Output DataFrame:**

   For each row, we create a dictionary (new_row) containing all the calculated values.

   We append this dictionary as a new row to the output_df DataFrame.

10. **Cleaning Emojis:**

    we apply the remove_emojis function to the 'Review' and 'Comments' columns of the output_df DataFrame to remove emojis from these columns.

11. **Saving the Results**:

    Finally, we saved the output_df DataFrame to a new CSV file named 'sentiment_analysis_results.csv' using to_csv() with index=False to exclude the index column.

Step 3: Performing EDA and Sentiment Analysis

Lastly, open the 'EDA and Final.ipynb' file and execute it

Introduction

This report presents the results of sentiment analysis conducted on a dataset of reviews. The analysis was performed using Python, with the NLTK library for sentiment scoring and Seaborn/Matplotlib for data visualization.

Data Preprocessing

The dataset was loaded from a CSV file, and any missing values were removed.

Duplicate rows, based on the columns 'Review', 'Comments', 'Ratings', 'Locations', and 'Time', were removed to ensure data integrity.

Sentiment Distribution

The sentiment distribution in the dataset was analyzed, and the following proportions were calculated:

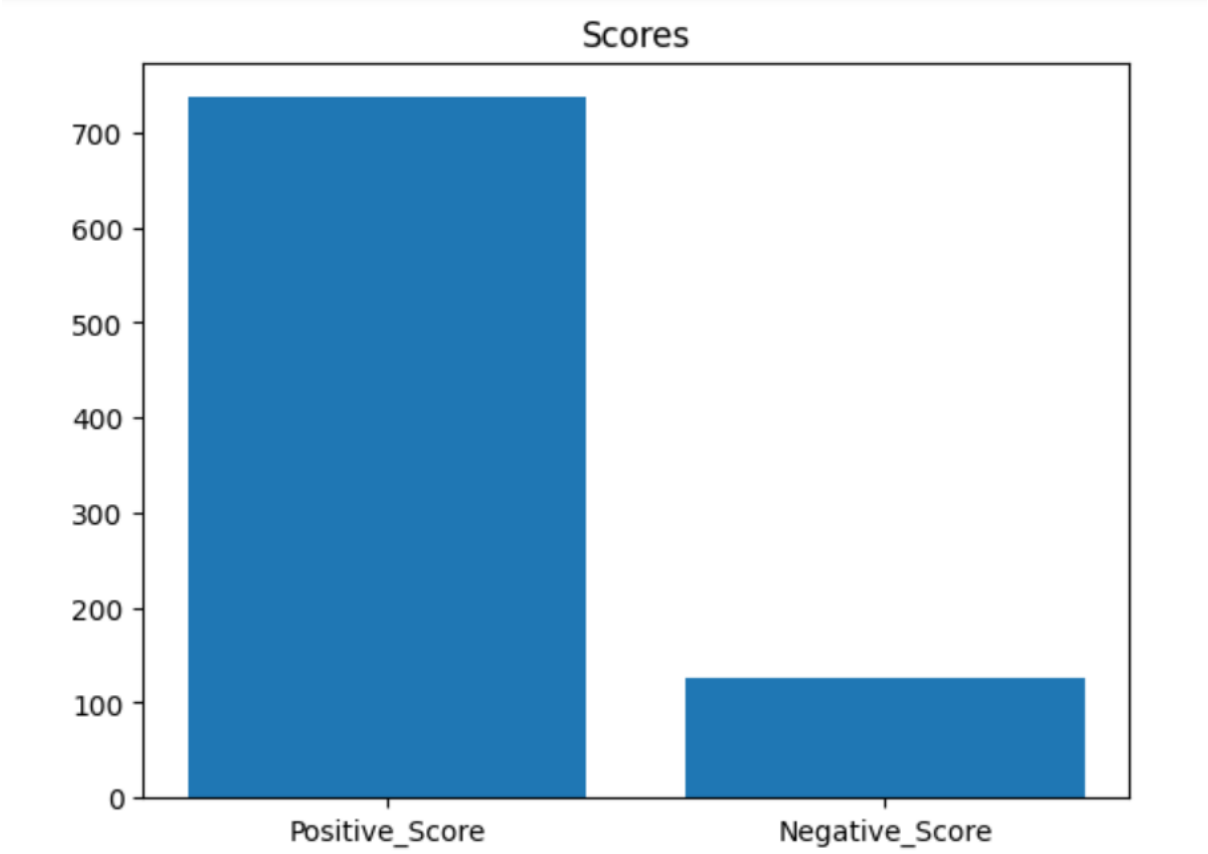
Proportion of Positive Sentiment: 0.8188105117565698
Proportion of Negative Sentiment: 0.16113416320885202
Proportion of Neutral Sentiment: 0.020055325034578148

This indicates that the majority of reviews in the dataset have a positive sentiment.

Sentiment Scores

The sum of positive sentiment scores is 737.2571.
The sum of negative sentiment scores is 126.9965.

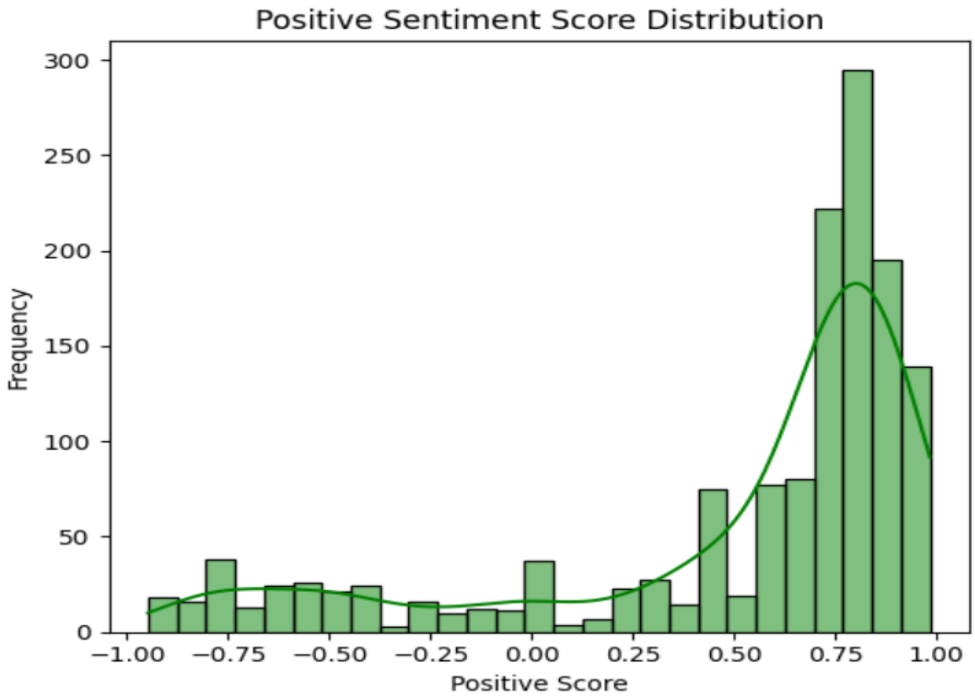
The bar chart below visually represents these scores



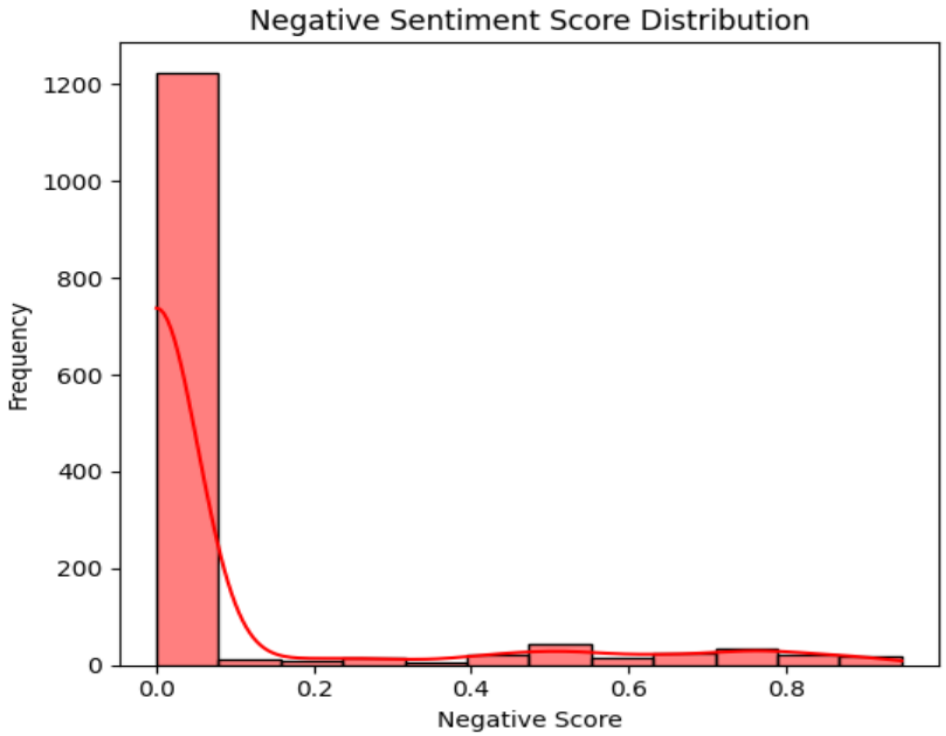
Sentiment Scores

Sentiment Score Distributions

Two histograms illustrate the distributions of positive and negative sentiment scores:



Positive Sentiment Score Distribution

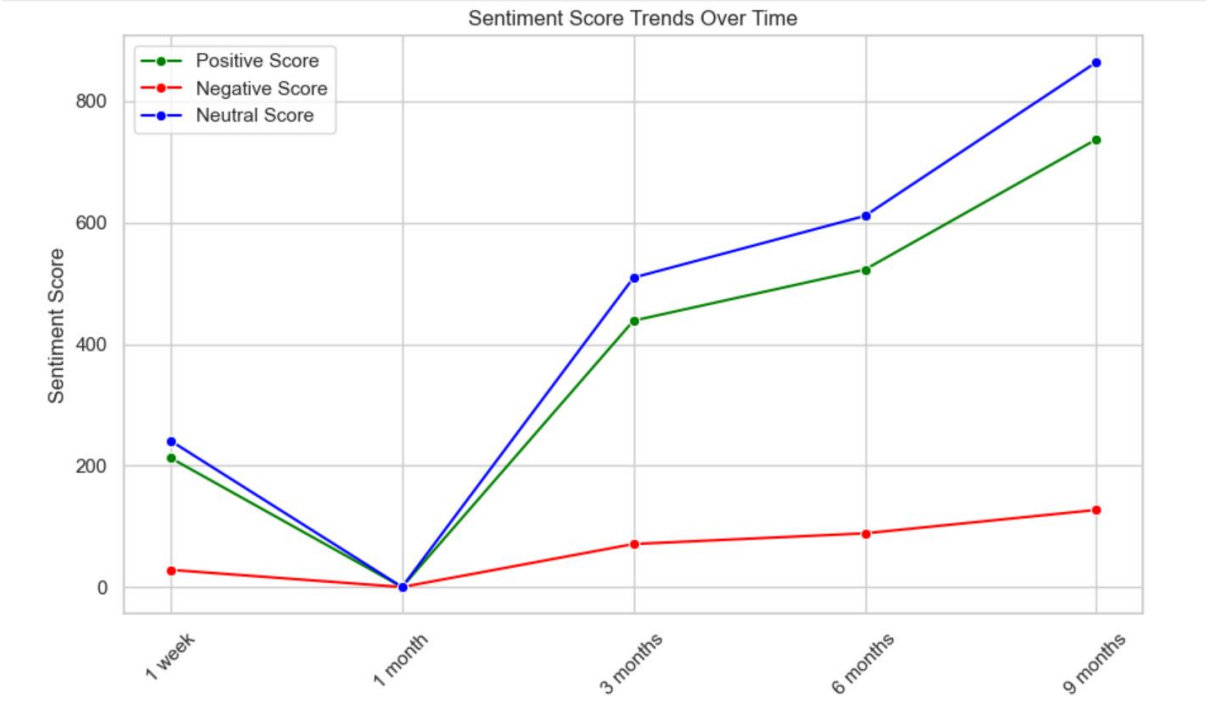


Negative Sentiment Score Distribution

From these distributions, it is evident that positive sentiment scores are skewed towards higher values, while negative sentiment scores tend to be lower.

Sentiment Trends Over Time

A line plot was created to visualize sentiment trends over time. The x-axis represents time intervals, and the y-axis represents sentiment scores. The plot includes positive, negative, and neutral sentiment scores.



Sentiment Trends Over Time

From the plot, we observe that sentiment scores fluctuate over time, with occasional spikes in both positive and negative sentiment. Neutral sentiment remains relatively stable.

Most Common Keywords

The most common keywords associated with positive and negative sentiment were identified.

Most Common Positive Keywords.

good: 1787
phone: 933
product: 844
nice: 738
camera: 637
best: 431
awesome: 408
battery: 326
mobile: 308
quality: 294

Most Common Negative Keywords.

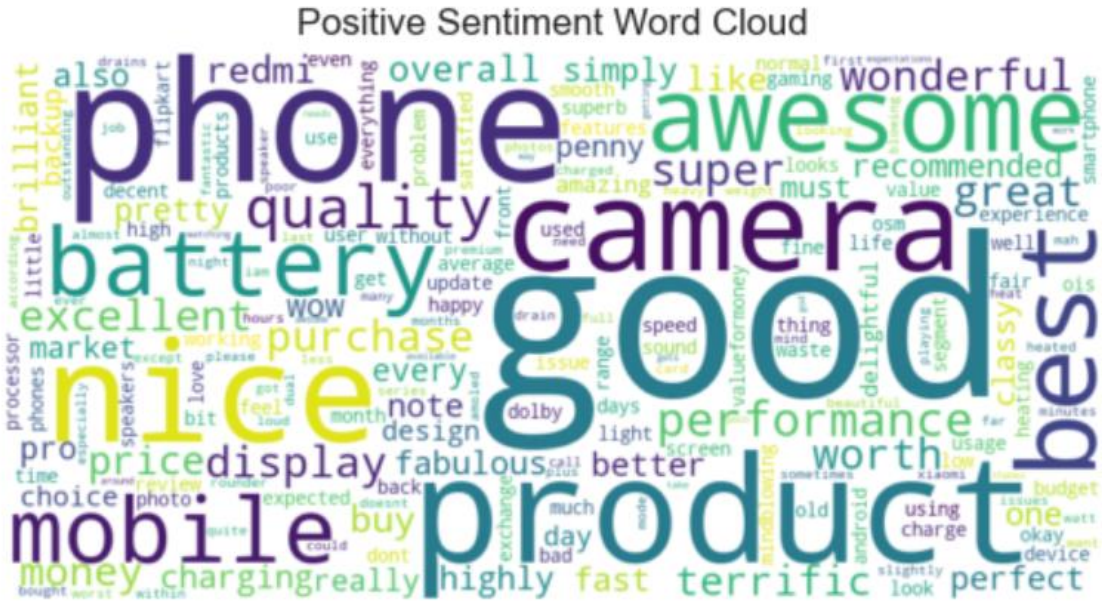
battery: 255

product: 236
camera: 234
good: 233
phone: 218
bad: 202
poor: 188
quality: 162
money: 123
performance: 122

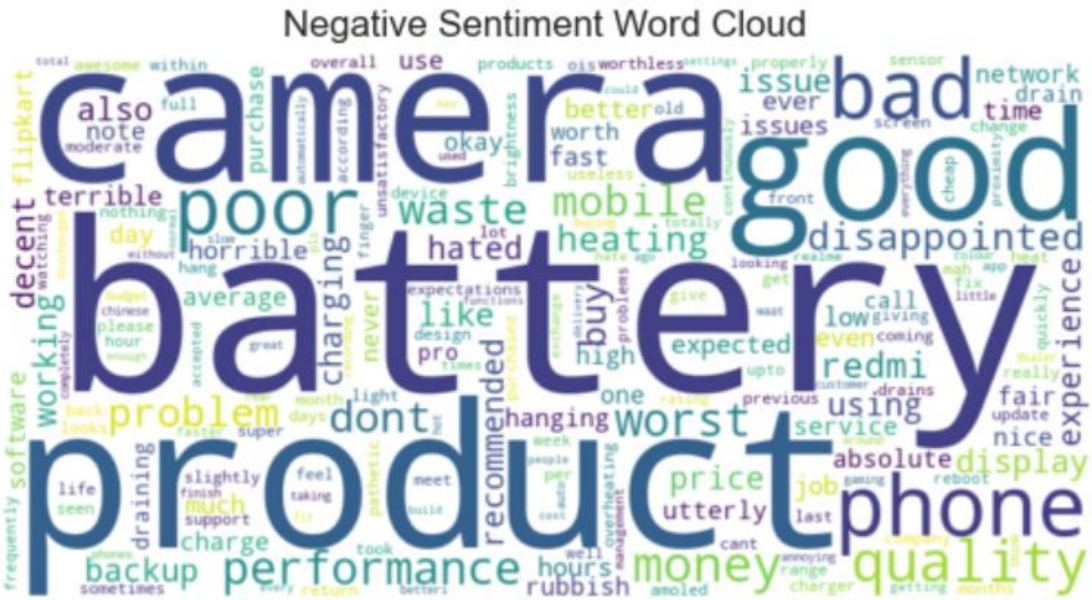
These keywords provide insights into the specific aspects of the reviews that contribute to their sentiment.

Word Clouds

Word clouds were generated to visually represent the most frequent words in positive and negative reviews. Larger words in the clouds indicate higher frequency.



Positive Sentiment Word Cloud



Negative Sentiment Word Cloud

The word clouds offer a quick glimpse into the most prominent words associated with each sentiment.

Conclusion

In conclusion, our sentiment analysis of customer reviews highlights a notably positive sentiment within the dataset. Approximately 81% of the reviews convey positive sentiments, indicating a strong overall satisfaction among customers. Negative sentiments are relatively infrequent, comprising only about 16% of the reviews, while neutral reviews make up the remaining 3%.

This overwhelmingly positive sentiment suggests that customers generally have a favorable opinion of the subject under review. However, it's important to pay attention to the smaller percentage of negative sentiments to identify areas that may require improvement or address specific concerns raised by some customers.