22CB903

MINI-PROJECT 1

OBJECTIVE:

Develop a model to classify customer reviews as positive, negative or neutral by choosing any two products.

ALGORITHM:

Extracting Reviews:

1. Initialize:
    - Start by importing the required libraries like requests and BeautifulSoup.
    - Initialize an empty list all_reviews to store extracted reviews.
2. Scrape Reviews:
    - Loop through the desired number of pages (e.g., first 10 pages) using a for loop.
    - For each page, generate the URL to fetch reviews using string formatting.
    - Send a GET request to fetch the page content.
    - Use BeautifulSoup to parse the page's HTML content.
    - Extract reviews by searching for specific HTML tags and classes (<p> and <div> with respective classes).
3. Combine and Clean Reviews:
    - Loop through the extracted reviews and clean the text by replacing commas and stripping unwanted characters.
    - Append the cleaned reviews to the all_reviews list.
4. Save Reviews:
    - Open a CSV file (e.g., reviews.csv) in append mode and write all the cleaned reviews to it.

Sentiment Analysis

1. Initialize:
    - Import necessary libraries such as re, pandas, nltk, textblob, and visualization libraries like seaborn and matplotlib.
    - Download NLTK resources (stopwords, vader_lexicon, and punkt).
2. Load and Clean Data:
    - Load the CSV file containing reviews into a DataFrame (df).
    - Define a regex pattern to remove unwanted characters.
    - Initialize the VADER sentiment analyzer and load the stopwords.
3. Process Reviews:
    - Loop through each review in the DataFrame.
    - Clean the review text by removing unwanted characters and stopwords.
    - Tokenize and rejoin the cleaned text.
4. Analyze Sentiment:
    - Use VADER to calculate sentiment polarity scores (positive, negative, neutral, compound).
    - Use TextBlob to calculate the subjectivity score.
    - Classify the sentiment based on the compound score (positive, negative, neutral).
5. Store Results:
    - Create a new list of dictionaries to store results, including ID, review text, sentiment scores, subjectivity score, and sentiment classification.
    - Convert this list into a DataFrame (output_df).
6. Visualize Results:
    - Generate a count plot of sentiment distribution using seaborn and matplotlib.
    - Display the first few rows of the output DataFrame and summarize the sentiment distribution.

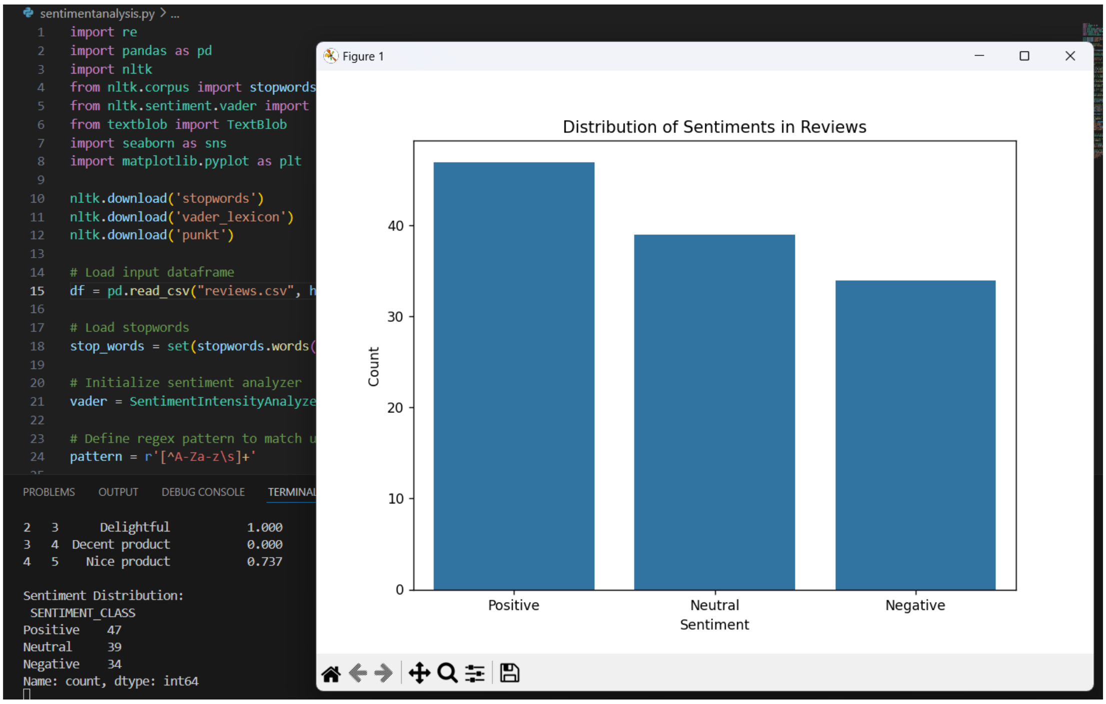
CODE:

```
import requests

from bs4 import BeautifulSoup

# Initialize an empty list to store all reviews

all_reviews = []

# Loop through the first 10 pages of reviews

for page_num in range(1, 10):

    # URL to get the reviews for the specified page

    url_source = 'https://www.flipkart.com/jio-b1-keypad-phone-upi-locked-
blue/product-
reviews/itm21336a1a4737b?pid=MOBGVJFHMG8PFNBK&lid=LSTMOBGVJFHMG8PFNBKXZ3SFN&m
arketplace=FLIPKART&page={page_num}'

    url = url_source.format(page_num=page_num)

    r = requests.get(url)

    # Extract data using BeautifulSoup

    soup = BeautifulSoup(r.content, 'lxml')

    # Extract reviews using the appropriate HTML tags and classes

    reviews = soup.find_all('p', {'class': "z9E0IG"})

    div_reviews = soup.find_all('div', {'class': "ZmyHeo"})

    # Combine reviews from <p> and <div> tags

    review2 = ''

    for review in reviews:

        review2 += review.text.strip().replace(',', ' ') + '\n'  # Replace
commas with spaces

    for div_review in div_reviews:

        review2 += div_review.text.strip().replace(',', ' ') + '\n'  #
Replace commas with spaces

    # Add the combined reviews to the list

    all_reviews.append(review2)

# Write all reviews to a CSV file

for review in all_reviews:

    with open('reviews.csv', 'a', encoding='utf-8') as f:

        f.write(review + '\n')


import re
import pandas as pd

import nltk

from nltk.corpus import stopwords

from nltk.sentiment.vader import SentimentIntensityAnalyzer

from textblob import TextBlob

import seaborn as sns

import matplotlib.pyplot as plt

nltk.download('stopwords')

nltk.download('vader_lexicon')

nltk.download('punkt')

# Load input dataframe

df = pd.read_csv("reviews.csv", header=None, names=["REVIEWS"])

# Load stopwords

stop_words = set(stopwords.words('english'))

# Initialize sentiment analyzer

vader = SentimentIntensityAnalyzer()

# Define regex pattern to match unwanted characters

pattern = r'[^A-Za-z\s]+'

# Create empty list to collect data for output dataframe

output_data = []

# Loop through rows of input dataframe

for index, row in df.iterrows():

    ID = index + 1  # Generate an ID based on the index

    REVIEWS = row["REVIEWS"]

    # Apply regex to remove unwanted characters

    cleaned_text = re.sub(pattern, ' ', REVIEWS)

    # Tokenize text into words

    words = nltk.word_tokenize(cleaned_text)

   # Remove stopwords and lowercase

    words = [word.lower() for word in words if word.lower() not in
stop_words]

    # Join words back into cleaned text

    cleaned_text = ' '.join(words)

    # Get polarity scores for cleaned text
    scores = vader.polarity_scores(cleaned_text)
    # Get the subjectivity score
    blob = TextBlob(cleaned_text)
    subjectivity_score = blob.sentiment.subjectivity
    # Classify sentiment based on compound score
    if scores['compound'] > 0.05:
        sentiment_class = 'Positive'
    elif scores['compound'] < -0.05:
        sentiment_class = 'Negative'
    else:
        sentiment_class = 'Neutral'
    # Collect the data in a dictionary
    output_data.append({
        "ID": ID,
        "REVIEWS": REVIEWS,
        "POSITIVE_SCORE": scores["pos"],
        "NEGATIVE_SCORE": scores["neg"],
        "SENTIMENT": scores["compound"],
        "SUBJECTIVITY_SCORE": subjectivity_score,
        "SENTIMENT_CLASS": sentiment_class
    })
# Convert the list of dictionaries into a DataFrame
output_df = pd.DataFrame(output_data)
# Summarize the sentiment distribution
sentiment_counts = output_df['SENTIMENT_CLASS'].value_counts()
print("\nSentiment Distribution:\n", sentiment_counts)
# Plot the sentiment distribution
plt.figure(figsize=(8, 6))
sns.countplot(x='SENTIMENT_CLASS', data=output_df)
plt.title('Distribution of Sentiments in Reviews')
plt.xlabel('Sentiment')
plt.ylabel('Count')
plt.show()
```

OUTPUT:

Product: Jio B1 Keypad Phone (UPI) Locked Blue

https://www.flipkart.com/jio-b1-keypad-phone-upi-locked-blue/product-reviews/itm21336a1a4737b?pid=MOBGVJFHMG8PFNBK&lid=LSTMOBGVJFHMG8PFNBKXZ3SFN&marketplace=FLIPKART

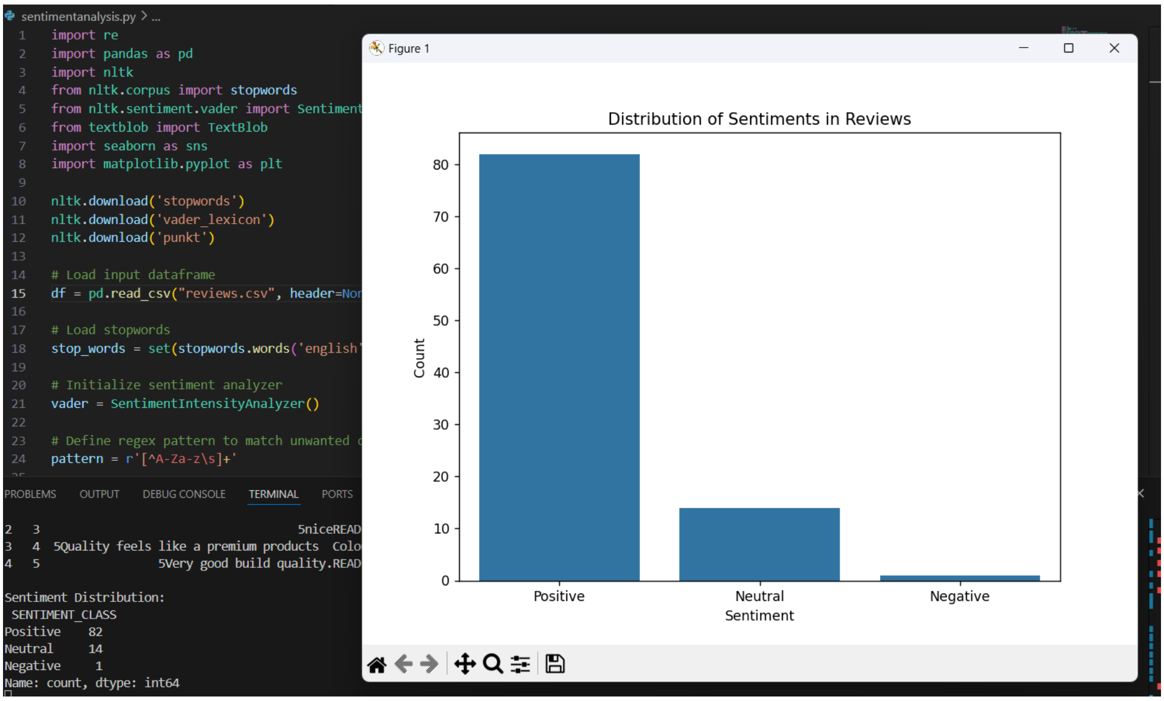


```python
import re
import pandas as pd
import nltk
from nltk.corpus import stopwords
from nltk.sentiment.vader import
from textblob import TextBlob
import seaborn as sns
import matplotlib.pyplot as plt

nltk.download('stopwords')
nltk.download('vader_lexicon')
nltk.download('punkt')

# Load input dataframe
df = pd.read_csv("reviews.csv", h

# Load stopwords
stop_words = set(stopwords.words(

# Initialize sentiment analyzer
vader = SentimentIntensityAnalyze

# Define regex pattern to match u
pattern = r'[^A-Za-z\s]+'
```

```
2   3      Delightful           1.000
3   4  Decent product           0.000
4   5    Nice product           0.737

Sentiment Distribution:
 SENTIMENT_CLASS
Positive    47
Neutral     39
Negative    34
Name: count, dtype: int64
```

Product: ARISTOCRAT Airstop 53 - Hardbody Trolley Bag Cabin Suitcase 4 Wheels - 21 Inch

https://www.flipkart.com/aristocrat-airstop-53-hardbody-trolley-bag-cabin-suitcase-4-wheels-21-inch/product-reviews/itm5a5d80f729676?pid=STCGGDWPMHQRVHZX&lid=LSTSTCGGDWPMHQRVHZXDWCVWS&marketplace=FLIPKART&page=1



```python
import re
import pandas as pd
import nltk
from nltk.corpus import stopwords
from nltk.sentiment.vader import Sentiment
from textblob import TextBlob
import seaborn as sns
import matplotlib.pyplot as plt

nltk.download('stopwords')
nltk.download('vader_lexicon')
nltk.download('punkt')

# Load input dataframe
df = pd.read_csv("reviews.csv", header=Nor

# Load stopwords
stop_words = set(stopwords.words('english'

# Initialize sentiment analyzer
vader = SentimentIntensityAnalyzer()

# Define regex pattern to match unwanted c
pattern = r'[^A-Za-z\s]+'
```

```
2  3                                  5niceREAD
3  4  5Quality feels like a premium products  Colo
4  5                   5Very good build quality.READ

Sentiment Distribution:
 SENTIMENT_CLASS
Positive    82
Neutral     14
Negative     1
Name: count, dtype: int64
```