**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.

1. 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).

1. 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.

1. 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).

1. 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.

1. 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.

1. 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).

1. 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).

1. 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&marketplace=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)

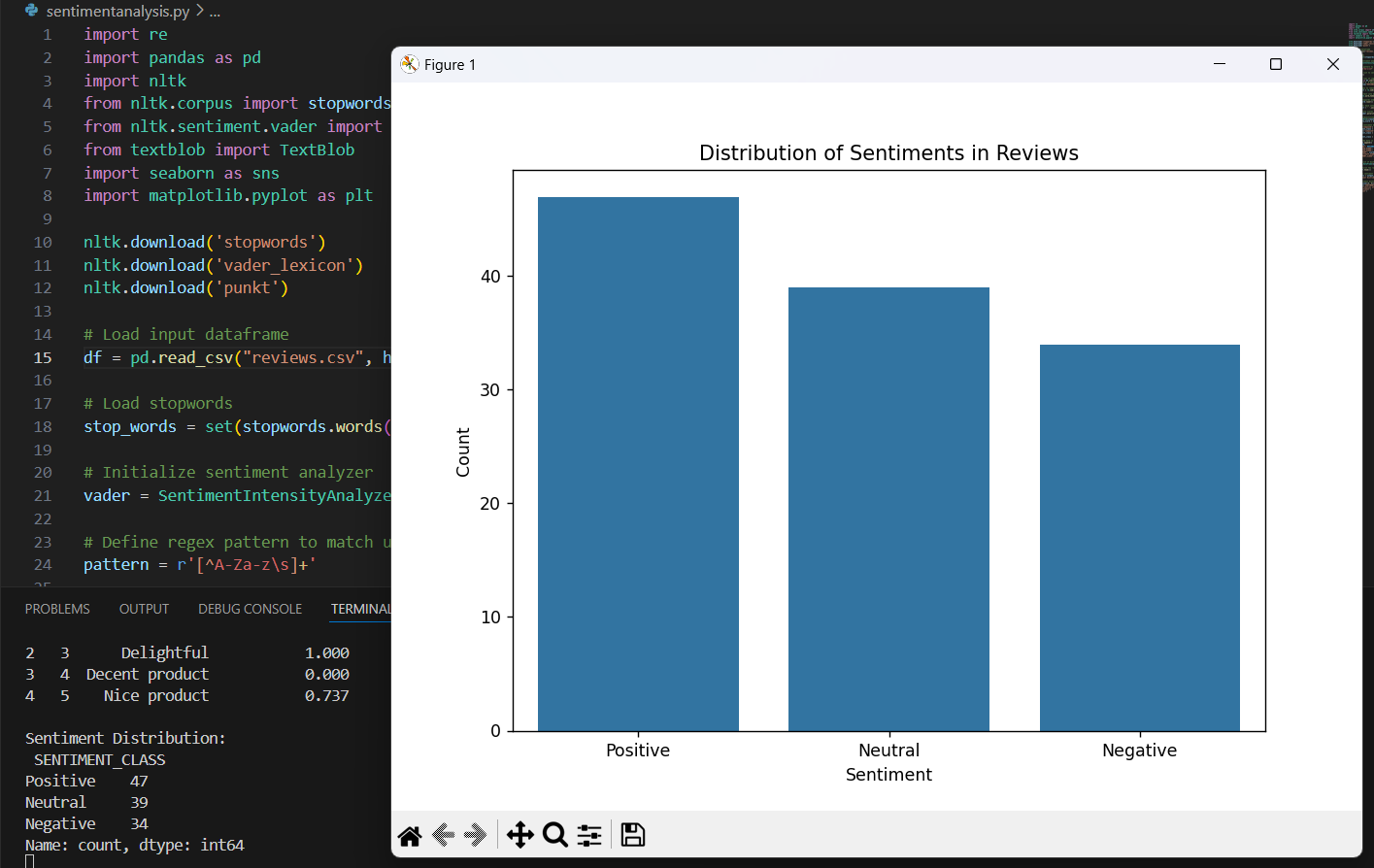
plt.title('Distribution of Sentiments in Reviews')

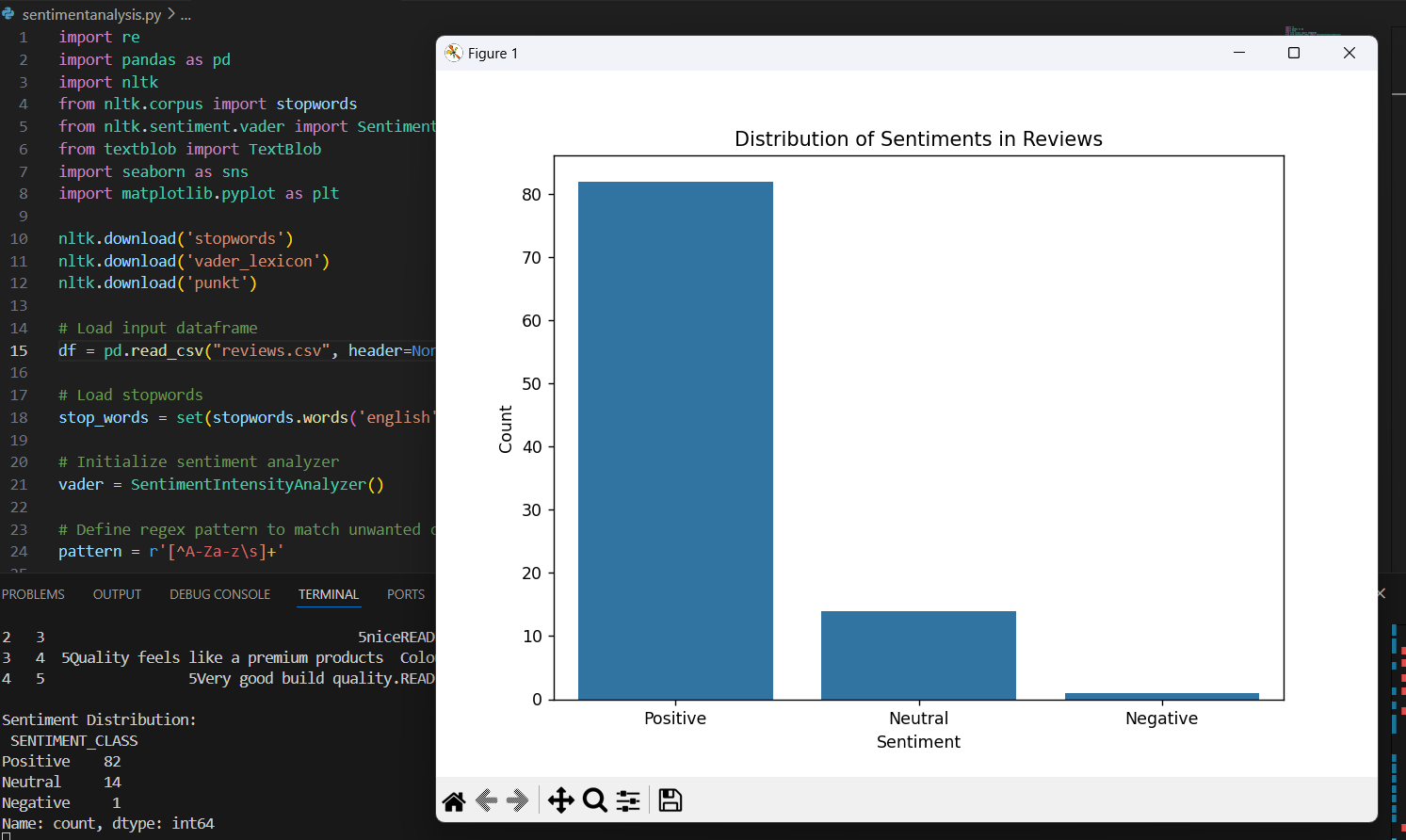
plt.xlabel('Sentiment')

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>

  
  
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>

*S Sharvesh Guru* | *CSBS* | *111722202043*