EXPERIMENT – 10

Objective: Develop social media text analytics model for improving existing products services by analysing customers reviews and comments.

To develop a text analytics model, begin by collecting social media comments from platforms like YouTube and Twitter. Clean and preprocess the text by removing unnecessary elements and normalizing it. Analyze the sentiment to categorize comments as positive, negative, or neutral. Extract keywords and topics to identify trends and themes. Finally, present insights through graphs and charts to enable better decision-making and trend analysis.

Step 1: Import & Setup

```python
import pandas as pd
import re
import nltk
import matplotlib.pyplot as plt
from nltk.sentiment.vader import SentimentIntensityAnalyzer
from wordcloud import WordCloud

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

[6] ✓ 19.2s Python

```
[nltk_data] Downloading package vader_lexicon to
[nltk_data]     C:\Users\jatin\AppData\Roaming\nltk_data...
[nltk_data] Downloading package stopwords to
[nltk_data]     C:\Users\jatin\AppData\Roaming\nltk_data...
[nltk_data]   Unzipping corpora\stopwords.zip.
```

```
True
```

Step 2: Load & Clean Data

```python
# Load social media comments
df = pd.read_csv("youtubereviews.csv")  # or any platform's comment data
df.dropna(subset=["results"], inplace=True)

# Clean text
def clean_text(text):
    text = re.sub(r"http\S+", "", text)  # Remove links
    text = re.sub(r"[^a-zA-Z\s]", "", text)  # Remove punctuation
    return text.lower().strip()

df["cleaned"] = df["results"].apply(clean_text)
```

[7] ✓ 0.1s Python

Step 3: Sentiment Analysis

```python
analyzer = SentimentIntensityAnalyzer()

def get_sentiment(text):
    score = analyzer.polarity_scores(text)['compound']
    if score >= 0.05:
        return "Positive"
    elif score <= -0.05:
        return "Negative"
    else:
        return "Neutral"

df["sentiment"] = df["cleaned"].apply(get_sentiment)
```

[8] ✓ 0.5s Python

Step 4: Keyword/Topic Extraction (Simple)

```python
from nltk.corpus import stopwords
from collections import Counter

stop_words = set(stopwords.words("english"))
all_words = ' '.join(df["cleaned"]).split()
filtered_words = [w for w in all_words if w not in stop_words and len(w) > 2]

word_freq = Counter(filtered_words).most_common(20)
print("Top Keywords:", word_freq)
```

[9] ✓ 0.0s Python

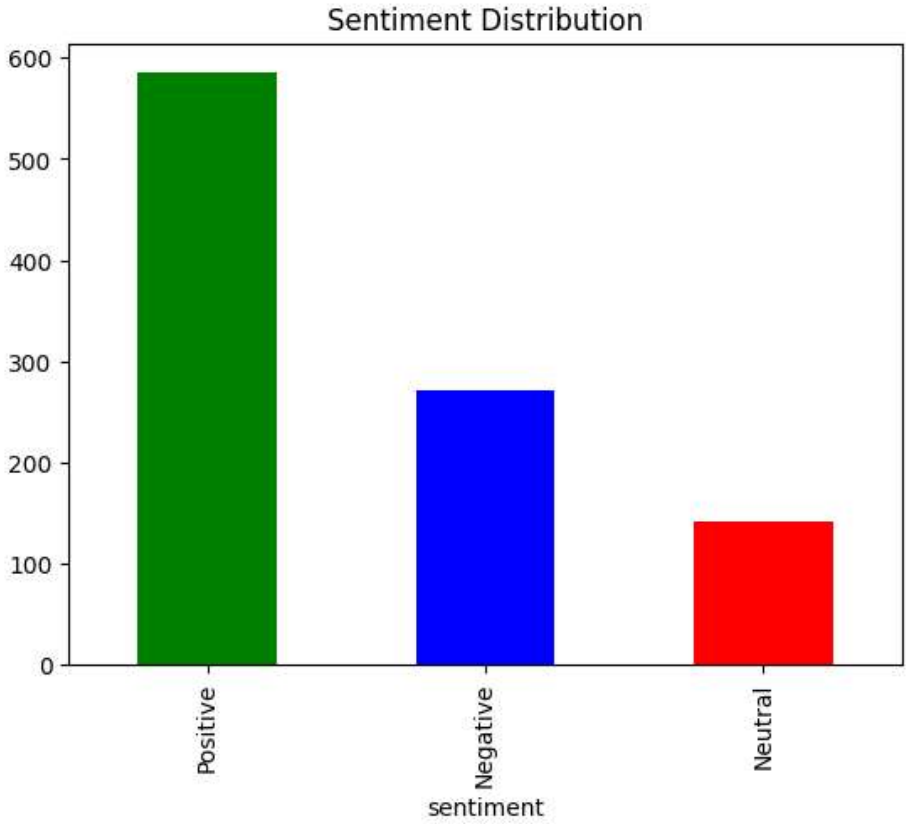
```
Top Keywords: [('video', 129), ('one', 118), ('like', 112), ('love', 85), ('life', 79), ('re
```

Step 5: Visualization

```python
# Sentiment distribution
df["sentiment"].value_counts().plot(kind="bar", title="Sentiment Distribution", color=
["green", "blue", "red"])
plt.show()

# Word Cloud
wc = WordCloud(width=800, height=400, background_color='white').generate(' '.join
(filtered_words))
plt.imshow(wc, interpolation='bilinear')
plt.axis('off')
plt.title("Word Cloud of Most Common Terms")
plt.show()
```

[13] ✓ 1.6s Python


Sentiment Distribution
600
500
400
300
200
100
0
Positive
Negative
Neutral
sentiment


Word Cloud of Most Common Terms
video
one
ant
time
life
thank
people
hope
love
make
still
really
always
much
year
tesla
many
song
guy
even
got
need
feel
never
see
good
thing
ive
man
made