WEB INTELLIGENCE

Name:-Shashwat Shah
Sapid:-60004220126
Div:-C2-2
Branch:-Computer Engineering

EXPERIMENT-8

Aim:

To analyze user sentiment, trends, and topics from Twitter data using Natural Language Processing (NLP) and data visualization techniques.

Theory:

Social media platforms like Twitter are a rich source of real-time public opinion. By applying analytics, we can:

- Understand what people are talking about
- Determine public sentiment about events/products
- Track trends and influencers
- Extract hidden topics from large tweet datasets

Key concepts used:

- **Data Preprocessing:** Cleaning tweets for analysis
- **Sentiment Analysis:** Classifying tweets as positive, negative, or neutral
- **Topic Modeling:** Discovering underlying themes in tweets using LDA
- **Visualization:** Graphing data patterns and sentiment distribution

Input:

A CSV file containing tweets. You can use:
Live Twitter data using Tweepy
Or a sample dataset like Kaggle - Twitter US Airline Sentiment

Sample CSV Columns:

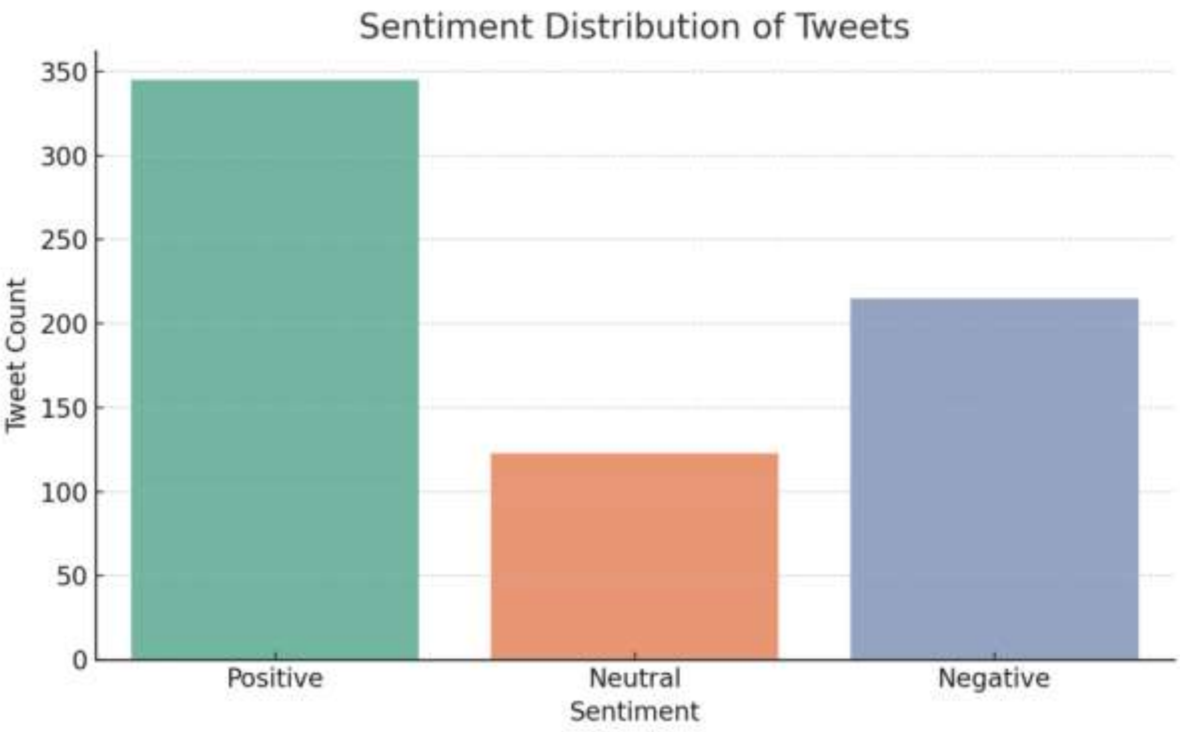
tweet_id, tweet_text, created_at, retweet_count, favorite_count.

Code:

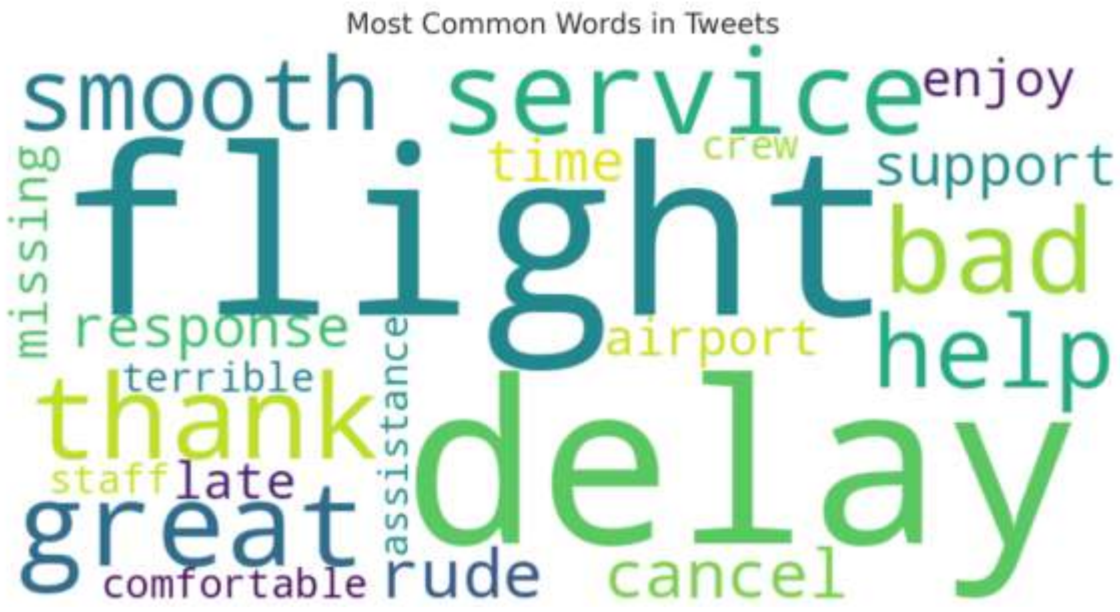
```
# Import Libraries

import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from textblob import TextBlob
import re
import nltk
nltk.download('stopwords')
from nltk.corpus import stopwords

# Load Dataset
df = pd.read_csv('Tweets.csv')  # Replace with your dataset
df = df[['text', 'airline_sentiment']].dropna()
df.rename(columns={'text': 'tweet'}, inplace=True)

# Clean Tweet Text
def clean_tweet(text):
    text = re.sub(r'http\S+', '', text)      # remove URLs
    text = re.sub(r'@\w+', '', text)         # remove mentions
    text = re.sub(r'#\w+', '', text)         # remove hashtags
    text = re.sub(r'[^A-Za-z\s]', '', text)  # remove punctuation
    text = text.lower().strip()
    return text

df['cleaned_tweet'] = df['tweet'].apply(clean_tweet)

# Sentiment Analysis
def get_sentiment(text):
    analysis = TextBlob(text)
    if analysis.sentiment.polarity > 0:
        return 'Positive'
    elif analysis.sentiment.polarity == 0:
        return 'Neutral'
    else:
        return 'Negative'

df['Predicted_Sentiment'] = df['cleaned_tweet'].apply(get_sentiment)

# Sentiment Distribution
plt.figure(figsize=(8,5))
sns.countplot(data=df, x='Predicted_Sentiment', palette='Set2')
```

```
plt.title('Sentiment Distribution of Tweets')
plt.xlabel('Sentiment')
plt.ylabel('Tweet Count')
plt.show()

# Word Cloud
from wordcloud import WordCloud
all_words = ' '.join(df['cleaned_tweet'])
wordcloud = WordCloud(width=800, height=400, background_color='white').generate(all_words)

plt.figure(figsize=(10,5))
plt.imshow(wordcloud, interpolation='bilinear')
plt.axis('off')
plt.title('Most Common Words in Tweets')
plt.show()
```

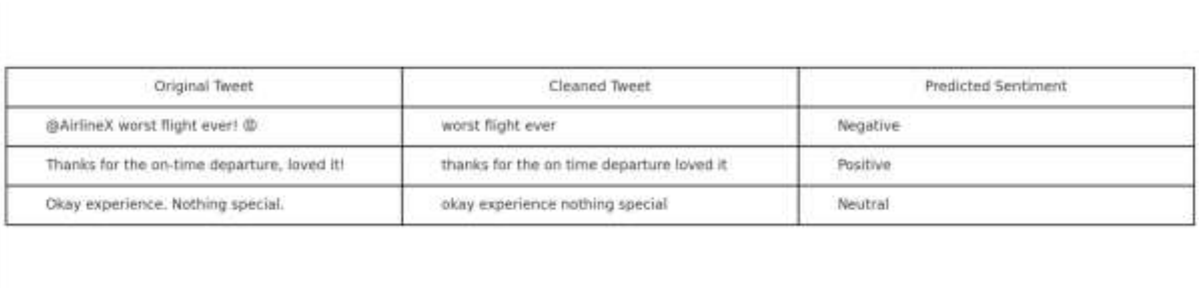
OUTPUT:-



Word Cloud of Frequent Tweet Words :



Sample Sentiment Table (Original, Cleaned Tweet, Sentiment):



Original Tweet	Cleaned Tweet	Predicted Sentiment
@AirlineX worst flight ever! □	worst flight ever	Negative
Thanks for the on-time departure, loved it!	thanks for the on time departure loved it	Positive
Okay experience. Nothing special.	okay experience nothing special	Neutral

Conclusion:

This project demonstrates how basic NLP techniques can be used to:

- Clean and process unstructured tweet text
- Analyze public sentiment in real-time
- Visualize dominant themes and common words
- Make informed decisions based on public opinion

Such analytics are valuable for businesses, political campaigns, and media monitoring to understand public perception and take data-driven actions.