Name: Ashish Gupta

Roll NO: 16

Class: D16AD

Sub: SMA

```
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
```

```
import csv

with open('tweeter_dataset.csv', 'r', encoding='latin-1') as file:
    reader = csv.reader(file)
    data = list(reader)

df = pd.DataFrame(data, columns=['target', 'ids', 'date', 'flag', 'user', 'text'])
```

```
df.head()
```

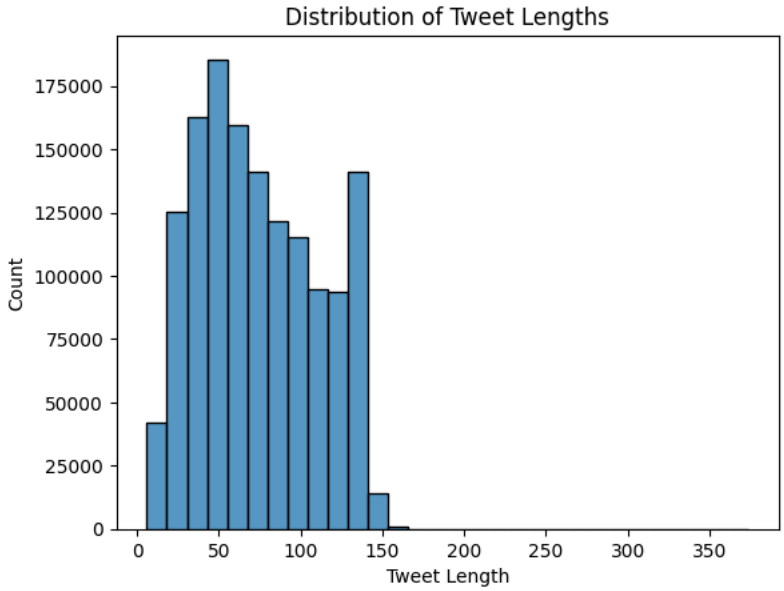
	target	ids	date	flag	user	text
0	0	1467810369	Mon Apr 06 22:19:45 PDT 2009	NO_QUERY	_TheSpecialOne_	@switchfoot http://twitpic.com/2y1zl - Awww, t...
1	0	1467810672	Mon Apr 06 22:19:49 PDT 2009	NO_QUERY	scotthamilton	is upset that he can't update his Facebook by ...
2	0	1467810917	Mon Apr 06 22:19:53 PDT 2009	NO_QUERY	mattycus	@Kenichan I dived many times for the ball. Man...
3	0	1467811184	Mon Apr 06 22:19:57 PDT 2009	NO_QUERY	ElleCTF	my whole body feels itchy and like its on fire
4	0	1467811193	Mon Apr 06 22:19:57 PDT 2009	NO_QUERY	Karoli	@nationwideclass no, it's not behaving at all....

```
#Check if dataset have missing values
df.isnull().sum()
```

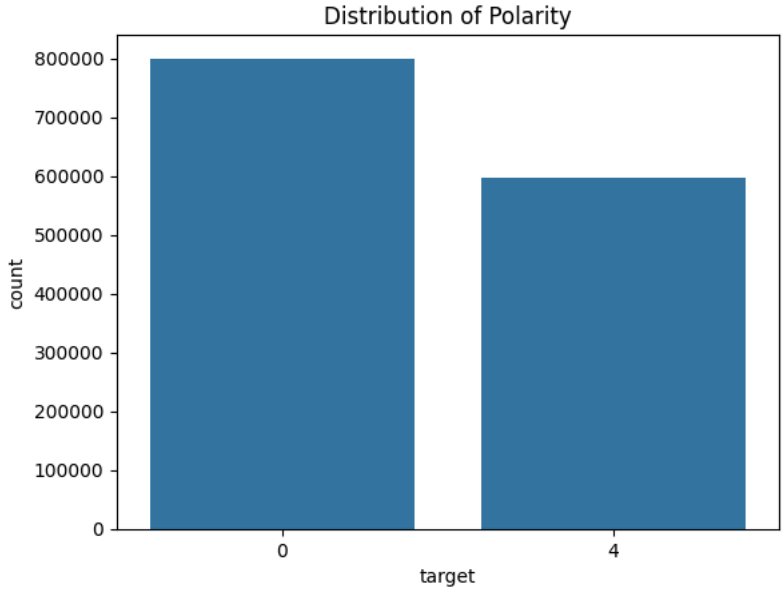
```
target    0
ids       0
date      0
flag      0
user      0
text      1
dtype: int64
```

```
df = df.dropna()
```

```
df['text_length'] = df['text'].apply(len)
sns.histplot(df['text_length'], bins=30)
plt.title('Distribution of Tweet Lengths')
plt.xlabel('Tweet Length')
plt.show()
```




```
sns.countplot(x='target', data=df)
plt.title('Distribution of Polarity')
plt.show()
```




```
print(df['target'].value_counts())
```

```
0    800000
4    597302
Name: target, dtype: int64
```

```
df = df.drop(["target", "ids", "date", "flag"], axis=1)
```

```
duplicate_count = df.duplicated(subset=['text']).sum()

print("Number of Duplicate Records:", duplicate_count)
print(df.shape)
```

```
Number of Duplicate Records: 16074
(1397302, 3)
```

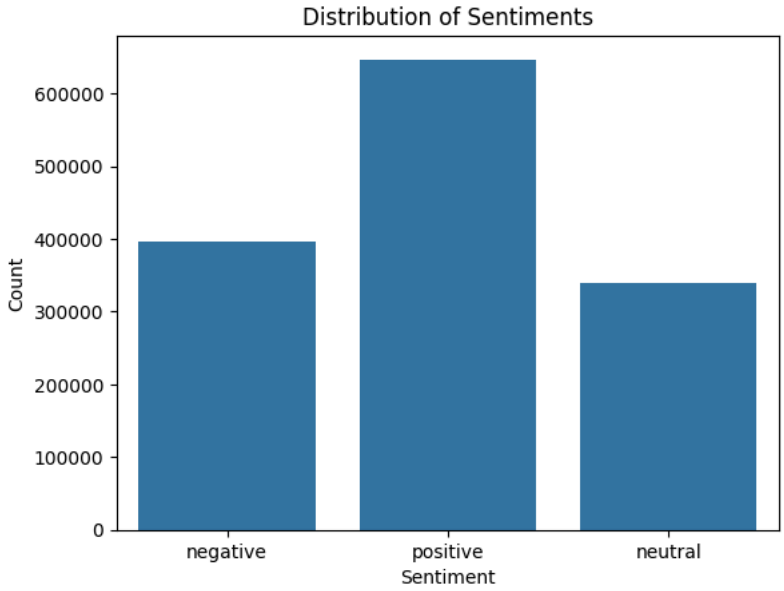
```
df = df.drop_duplicates(subset=['text'])

print(df.shape)
```

```
(1381228, 3)
```

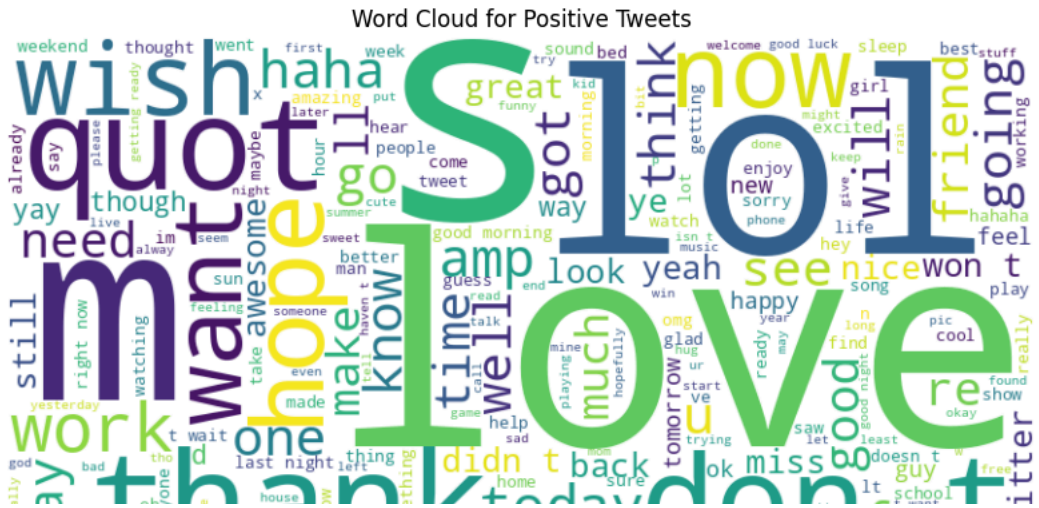
```
# Text cleaning
def clean_text(text):
    text = re.sub(r'@[A-Za-z0-9]+', '', text)  # Remove mentions
    text = re.sub('https?://[A-Za-z0-9./]+', '', text)  # Remove URLs
    text = re.sub("[^a-zA-Z]", " ", text)  # Remove special characters and numbers
    text = text.lower()  # Convert to lowercase
    return text

df['clean_text'] = df['text'].apply(clean_text)


df = df.drop('text',axis=1)


df = df.rename(columns={'clean_text': 'text'})


import nltk
from nltk.sentiment import SentimentIntensityAnalyzer


nltk.download('vader_lexicon')
```

```
[nltk_data] Downloading package vader_lexicon to /root/nltk_data...
[nltk_data]   Package vader_lexicon is already up-to-date!
True
```

```
sia = SentimentIntensityAnalyzer()

# Apply sentiment analysis to the clean_text column
df['sentiment_score'] = df['text'].apply(lambda x: sia.polarity_scores(x)['compound'])

# Convert the sentiment scores to categories (positive, negative, neutral)
df['sentiment'] = df['sentiment_score'].apply(lambda x: 'positive' if x > 0 else 'negative' if x < 0 else 'neutral')



 sns.countplot(x='sentiment', data=df)
plt.title('Distribution of Sentiments')
plt.xlabel('Sentiment')
plt.ylabel('Count')
plt.show()
```



```
from wordcloud import WordCloud

positive_tweets = df[df['sentiment'] == 'positive']['text']
negative_tweets = df[df['sentiment'] == 'negative']['text']

# Word cloud for positive tweets
wordcloud_positive = WordCloud(width=800, height=400, background_color='white').generate(' '.join(positive_tweets))
plt.figure(figsize=(10, 5))
plt.imshow(wordcloud_positive, interpolation='bilinear')
plt.title('Word Cloud for Positive Tweets')
plt.axis('off')
plt.show()
```




```
wordcloud_negative = WordCloud(width=800, height=400, background_color='white').generate(' '.join(negative_tweets))
plt.figure(figsize=(10, 5))
plt.imshow(wordcloud_negative, interpolation='bilinear')
plt.title('Word Cloud for Negative Tweets')
plt.axis('off')
plt.show()
```