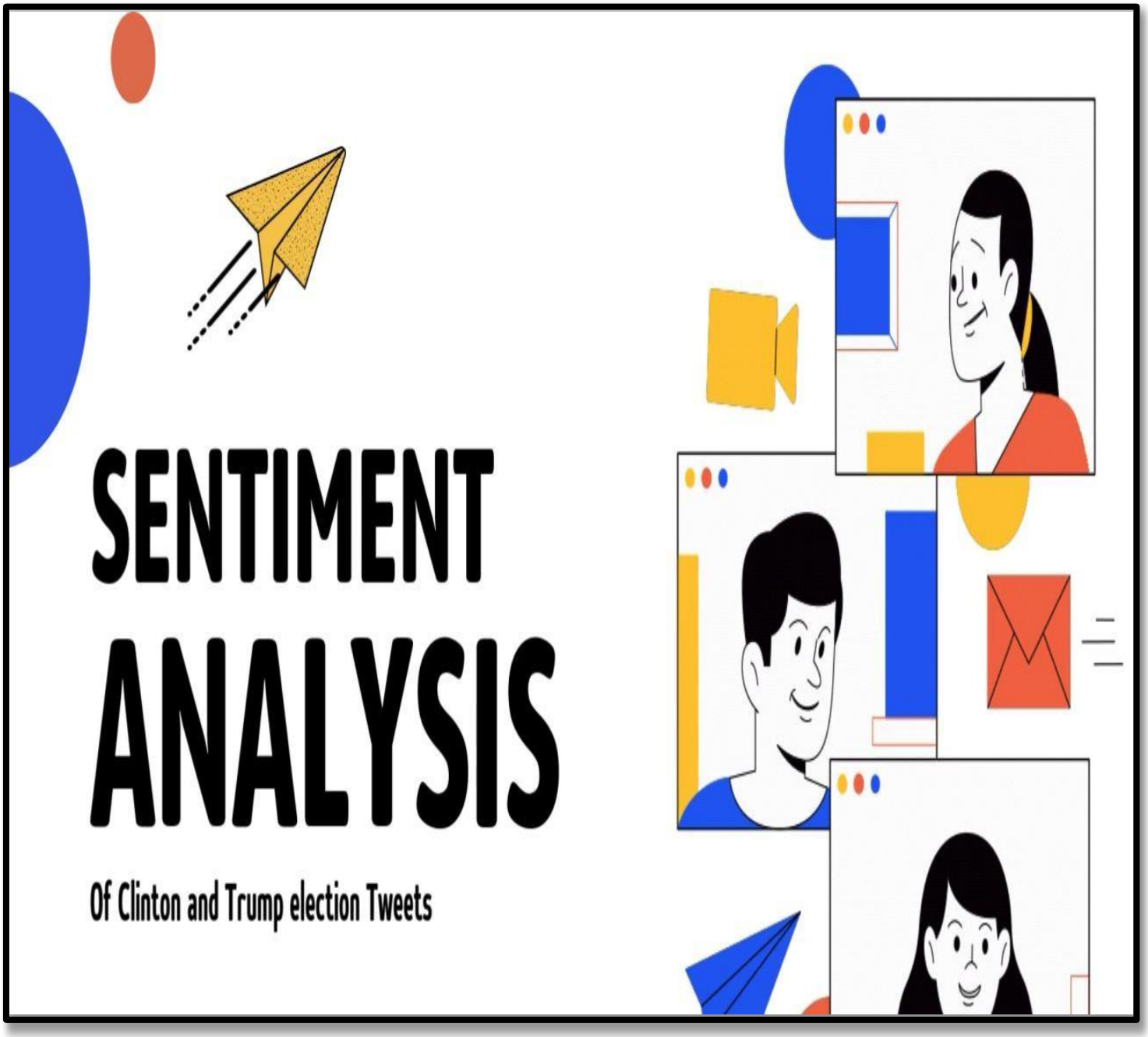

SENTIMENT
ANALYSIS
Of Clinton and Trump election Tweets

PROBLEM STATEMENT:

In today's digital era, social media platforms like Twitter play a significant role in shaping public opinion, disseminating information, and influencing behavior. Understanding the activity, engagement, and sentiment of tweets from key individuals or organizations can provide valuable insights into their communication strategies and audience engagement patterns.

This analysis focuses on a dataset of tweets from two prominent figures: Hillary Clinton and Donald Trump. The objectives of this analysis are as follows:

- Activity Analysis: Explore temporal patterns in tweeting behavior, such as monthly, daily, and hourly trends, to compare how frequently each individual uses Twitter.
- Engagement Metrics: Analyze retweet counts and favorites to understand audience interaction and content virality for each account.
- Textual Insights: Use text mining techniques to identify commonly used words, themes, and topics in tweets by Clinton and Trump.
- Sentiment Analysis: Evaluate the tone of tweets to classify them as positive, negative, or neutral and compare the overall sentiment distribution between the two accounts.
- Comparative Analysis: Highlight differences in behavior, language, and engagement between the two individuals.

The results of this analysis aim to provide actionable insights into the use of Twitter as a communication tool by public figures, emphasizing strategies for maximizing engagement and shaping public narratives.

DATASET:

The dataset, sourced from Kaggle, contains tweets from **Hillary Clinton** and **Donald Trump**, providing insights into their tweeting behavior and audience engagement. It consists of 6,444 rows and 28 columns with key attributes such as:

- **handle**: Twitter account (@HillaryClinton or @realDonaldTrump).
- **text**: Content of the tweet.
- **is_retweet**: Indicates if the tweet is a retweet.
- **time**: Timestamp of the tweet.
- **retweet_count** and favorite_count: Engagement metrics.
- **lang**: Language of the tweet.

This dataset allows for analyzing temporal patterns, engagement levels, and sentiment trends between the two accounts.

PROJECT STEPS:

#CODE AND OUTPUTS:

Preparation

```
# It is defined by the kaggle/python docker image: https://github.com/kaggle/docker-python
import os
import numpy as np # linear algebra
import pandas as pd # data processing, CSV file I/O (e.g. pd.read_csv)
import seaborn as sns
import matplotlib.pyplot as plt
```

```
[ ] # Input data file
    tweets = pd.read_csv('../content/tweets.csv')
```

```
#Separating out the time variable by Hour, Day, Month and Year
#for further analysis using datetime package
import datetime as dt
tweets['time'] = pd.to_datetime(tweets['time'])
tweets['hour'] = tweets['time'].apply(lambda x: x.hour)
tweets['month'] = tweets['time'].apply(lambda x: x.month)
tweets['day'] = tweets['time'].apply(lambda x: x.day)
tweets['year'] = tweets['time'].apply(lambda x: x.year)
```

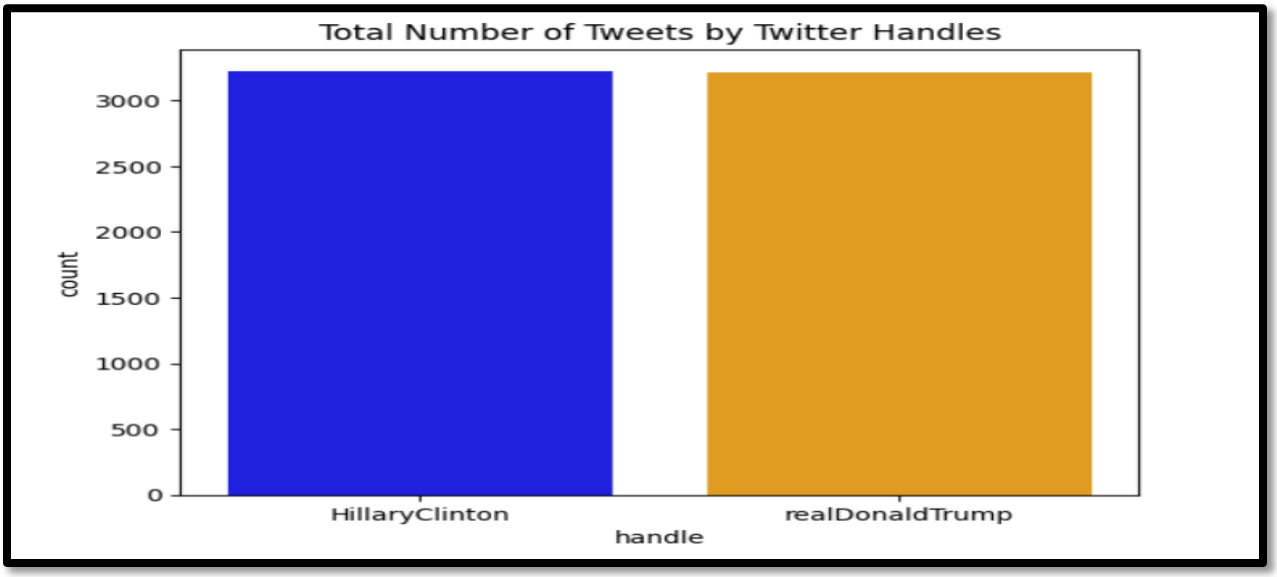
```
tweets.head(5)
```

	id	handle	text	is_retweet	original_author	time	in_reply_to_screen_name	in_reply_to_status_id	in_reply_to_user_id	is_quote_status	...	place_attri
0	780925634159796224	HillaryClinton	The question in this election: Who can put the...	False	NaN	2016-09-28 00:22:34	NaN	NaN	NaN	False	...	
1	780916180899037184	HillaryClinton	Last night, Donald Trump said not paying taxes...	True	timkaine	2016-09-27 23:45:00	NaN	NaN	NaN	False	...	
2	780911564857761793	HillaryClinton	Couldn't be more proud of @HillaryClinton. Her...	True	POTUS	2016-09-27 23:26:40	NaN	NaN	NaN	False	...	
3	780907038650068994	HillaryClinton	If we stand together, there's nothing we can't...	False	NaN	2016-09-27 23:08:41	NaN	NaN	NaN	False	...	
4	780897419462602752	HillaryClinton	Both candidates were asked about how they'd co...	False	NaN	2016-09-27 22:30:27	NaN	NaN	NaN	False	...	

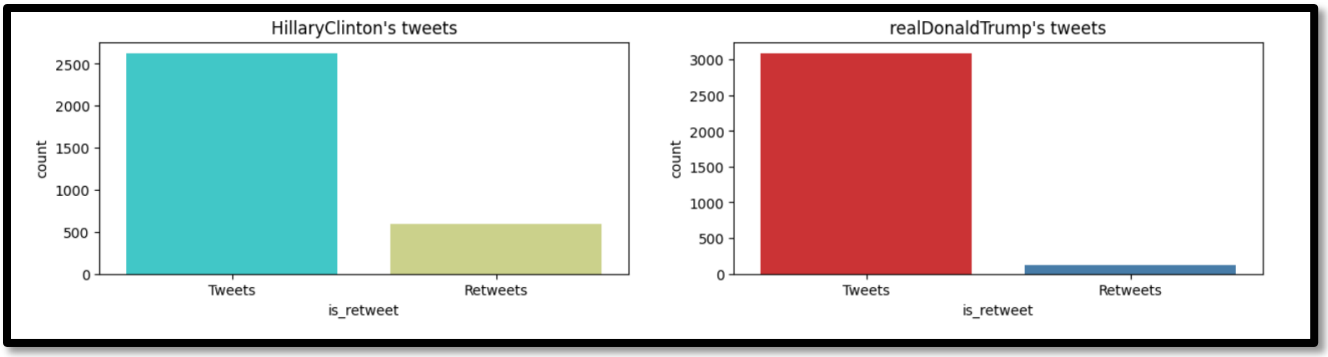
5 rows × 32 columns

Overall Tweets and Retweets visualization

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

# Plot with one bar in orange
sns.countplot(x='handle', data=tweets, palette=['blue', 'orange'])

# Show the plot
plt.title("Total Number of Tweets by Twitter Handles")
plt.show()
```



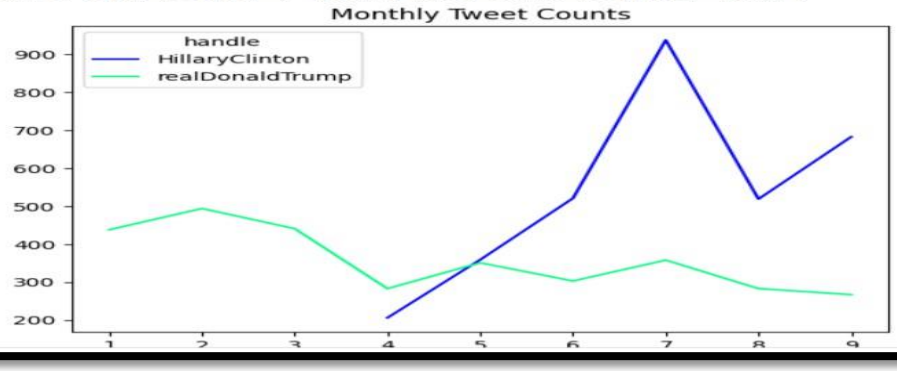

```
fig, (axis1, axis2) = plt.subplots(1, 2, figsize=(15, 3))

# Total number of original tweets and retweets for each contender
retweet_hc = tweets.loc[tweets['handle'] == 'HillaryClinton', ['is_retweet']]
retweet_dt = tweets.loc[tweets['handle'] == 'realDonaldTrump', ['is_retweet']]

# Updated countplot for Hillary Clinton
ax1 = sns.countplot(data=retweet_hc, x='is_retweet', palette='rainbow', ax=axis1)
ax1.set_title("HillaryClinton's tweets")
ax1.set_xticks([0, 1])  # Set fixed ticks
ax1.set_xticklabels(["Tweets", "Retweets"])

# Updated countplot for Donald Trump
ax2 = sns.countplot(data=retweet_dt, x='is_retweet', palette="Set1", ax=axis2)
ax2.set_title("realDonaldTrump's tweets")
ax2.set_xticks([0, 1])  # Set fixed ticks
ax2.set_xticklabels(["Tweets", "Retweets"])
```



Time Analysis of the number of tweets

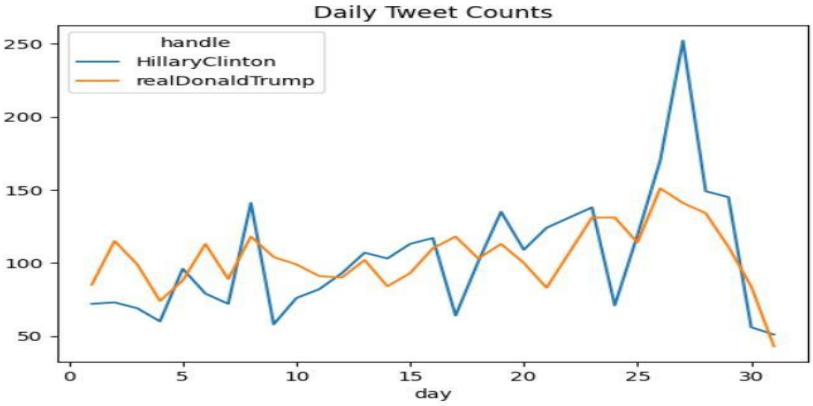
```
#Number of tweets by the months
monthly_tweets = tweets.groupby(['month', 'handle']).size().unstack()
monthly_tweets.plot(title='Monthly Tweet Counts', colormap='winter')
```

```
<Axes: title={'center': 'Monthly Tweet Counts'}, xlabel='month'>
```



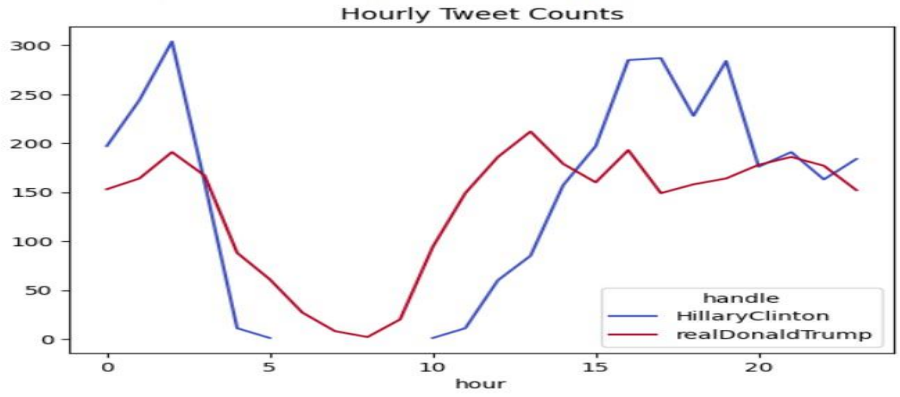

```
#Number of tweets daily
daily_tweets = tweets.groupby(['day', 'handle']).size().unstack()
daily_tweets.plot(title='Daily Tweet Counts')
```

```
<Axes: title={'center': 'Daily Tweet Counts'}, xlabel='day'>
```



```
#Number of tweets hourly
hourly_tweets = tweets.groupby(['hour', 'handle']).size().unstack()
hourly_tweets.plot(title='Hourly Tweet Counts', colormap='coolwarm')
```

```
<Axes: title={'center': 'Hourly Tweet Counts'}, xlabel='hour'>
```



TEXT MINING

```
from wordcloud import WordCloud
from wordcloud import STOPWORDS

tweets_hillary=tweets.loc[(tweets['handle']=='HillaryClinton'),['text']]
tweets_trump=tweets.loc[(tweets['handle']=='realDonaldTrump'),['text']]
```

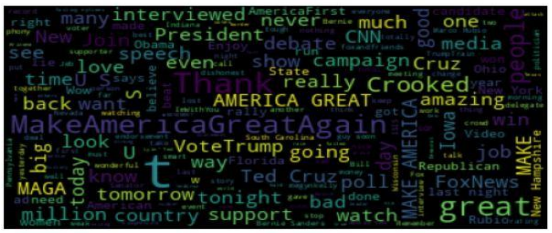
```
stopwords = set(STOPWORDS)
stopwords.add("http")
stopwords.add("https")
stopwords.add("amp")
stopwords.add("CO")
stopwords.add("Trump")
stopwords.add("Trump2016")
stopwords.add("Donald")
stopwords.add("Clinton")
stopwords.add("Hillary")
stopwords.add("realDonaldTrump")
stopwords.add("will")
stopwords.add("say")
stopwords.add("said")
stopwords.add("let")
stopwords.add("vote")
stopwords.add("now")
stopwords.add("go")
```

THE WORDS CLINTON USED THE MOST

```
wordcloud_hc = WordCloud(background_color='white',max_font_size=46, relative_scaling=0.5,stopwords=stopwords).generate(tweets_hillary['text'].str.cat())
plt.figure()
plt.imshow(wordcloud_hc)
plt.axis("off")
plt.show()
```

THE WORDS TRUMP USED THE MOST

```
[ ] wordcloud_dt = WordCloud(max_font_size=42, relative_scaling=.5,stopwords=stopwords).generate(tweets_trump['text'].str.cat())
    plt.figure()
    plt.imshow(wordcloud_dt)
    plt.axis("off")
    plt.show()
```

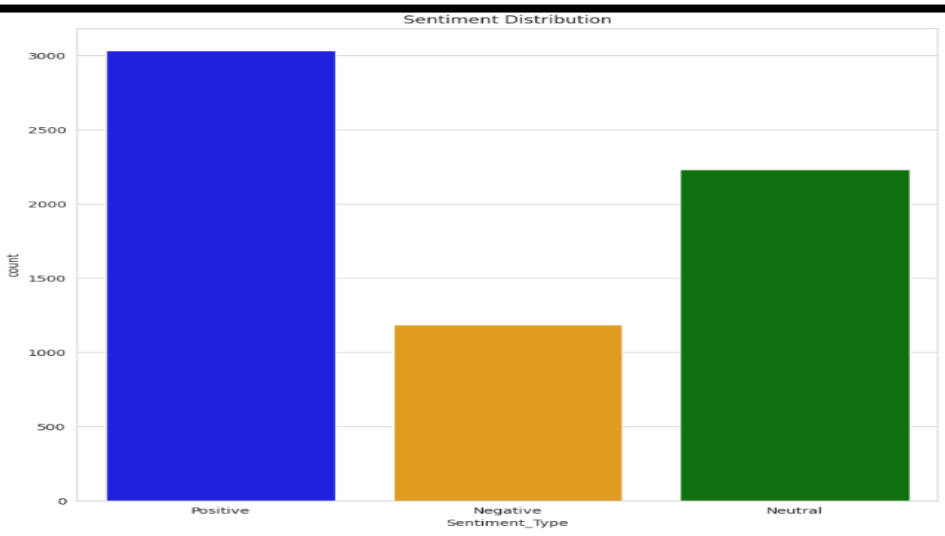


SENTIMENT ANALYSIS

Categorize the text column into Positive and Negative sentiments using TextBlob

ON THE ENTIRE DATASET

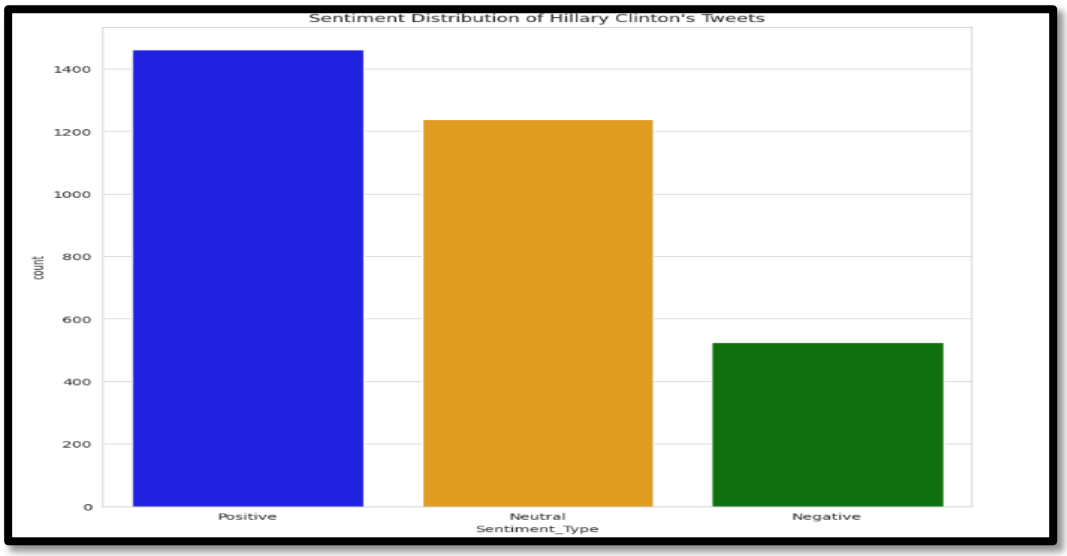
```
from textblob import TextBlob
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns

bloblist_desc = []

# Convert the 'text' column to string and process sentiment analysis
df_tweet_descr_str = tweets['text'].astype(str)
for row in df_tweet_descr_str:
    blob = TextBlob(row)
    bloblist_desc.append((row, blob.sentiment.polarity, blob.sentiment.subjectivity))

# Create a DataFrame with sentiment and polarity
df_tweet_polarity_desc = pd.DataFrame(bloblist_desc, columns=['sentence', 'sentiment', 'polarity'])

# Define sentiment type based on polarity
def f(row):
    if row['sentiment'] > 0:
        return "Positive"
    elif row['sentiment'] == 0:
```



CLINTON SENTIMENT ANALYSIS

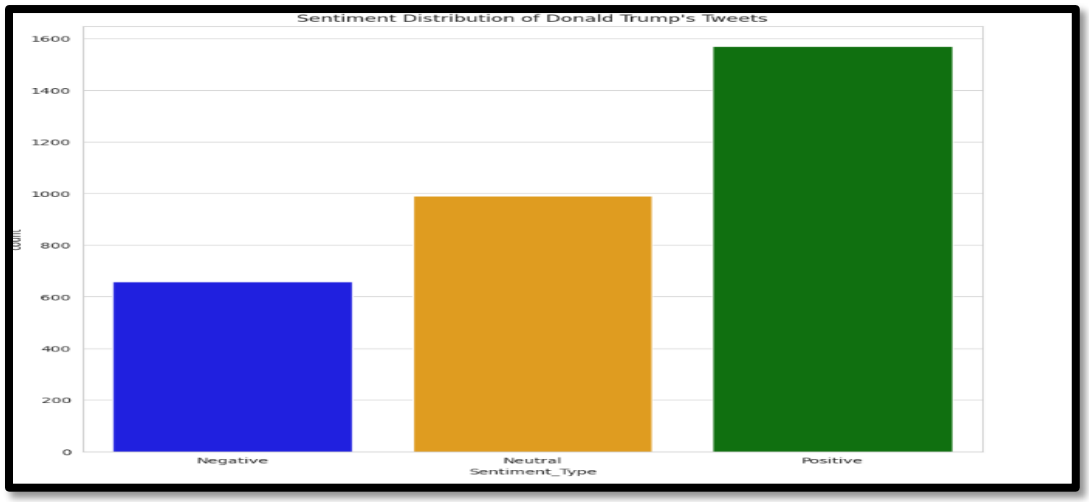
```
import matplotlib.pyplot as plt
import seaborn as sns
from textblob import TextBlob
import pandas as pd

# Filter Hillary Clinton's tweets
tweet_clinton = tweets.loc[tweets['handle'] == 'Hillaryclinton', ['text']]

# Analyze sentiment for Hillary Clinton's tweets
bloblist_desc = []
df_tweet_clinton_str = tweet_clinton['text'].astype(str)
for row in df_tweet_clinton_str:
    blob = TextBlob(row)
    bloblist_desc.append((row, blob.sentiment.polarity, blob.sentiment.subjectivity))

# Create a DataFrame for sentiment analysis results
df_tweet_clinton_polarity_desc = pd.DataFrame(bloblist_desc, columns=['sentence', 'sentiment', 'polarity'])

# Define sentiment type based on polarity
def f(row):
    if row['sentiment'] > 0:
        return "Positive"
    elif row['sentiment'] == 0:
        return "Neutral"
    else:
        return "Negative"

df_tweet_clinton_polarity_desc['Sentiment_Type'] = df_tweet_clinton_polarity_desc.apply(f, axis=1)

# Plot the countplot with specific colors
plt.figure(figsize=(10, 10))
sns.set_style("whitegrid")

# Define the custom palette
custom_palette = ['blue', 'orange', 'green']
ax = sns.countplot(x="Sentiment_Type", data=df_tweet_clinton_polarity_desc, palette=custom_palette)

# Add a title
plt.title("Sentiment Distribution of Hillary Clinton's Tweets")
plt.show()
```

```
<ipython-input-15-6ed4901e26c3>:36: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in v0.14.0. Assign the `x` variable to `hue` and set `legend=False` for the same effect.

  ax = sns.countplot(x="Sentiment_Type", data=df_tweet_clinton_polarity_desc, palette=custom_palette)
```

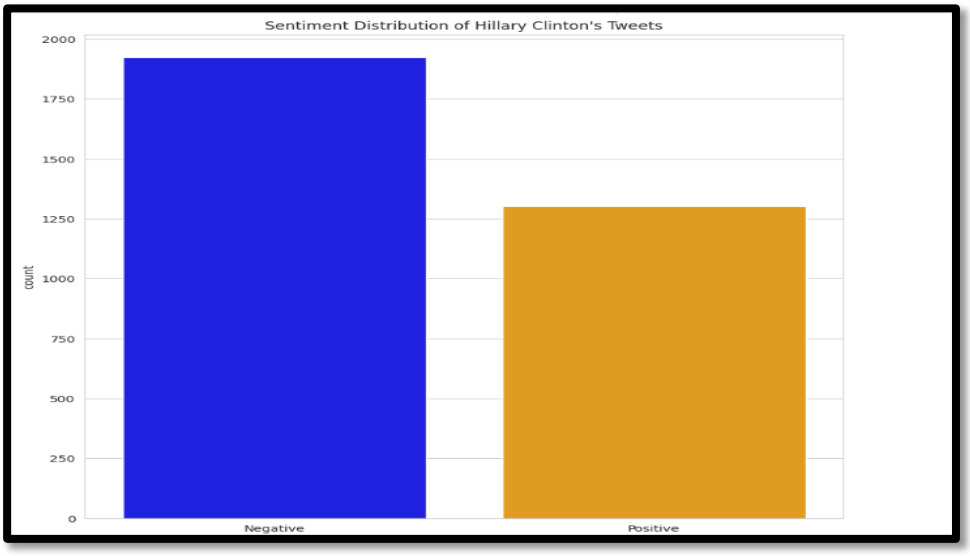


TRUMP SENTIMENT ANALYSIS

```
import matplotlib.pyplot as plt
import seaborn as sns
from textblob import TextBlob
import pandas as pd

# Filter Donald Trump's tweets
tweet_trump = tweets.loc[tweets['handle'] == 'realDonaldTrump', ['text']]

# Analyze sentiment for Trump's tweets
bloblist_desc = []
df_tweet_trump_str = tweet_trump['text'].astype(str)
for row in df_tweet_trump_str:
    blob = TextBlob(row)
    bloblist_desc.append((row, blob.sentiment.polarity, blob.sentiment.subjectivity))

# Create a DataFrame for sentiment analysis results
df_tweet_trump_polarity_desc = pd.DataFrame(bloblist_desc, columns=['sentence', 'sentiment', 'polarity'])

# Define sentiment type based on polarity
def f(row):
    if row['sentiment'] > 0:
        return "Positive"
    elif row['sentiment'] == 0:
        return "Neutral"
    else:
        return "Negative"

df_tweet_trump_polarity_desc['Sentiment_Type'] = df_tweet_trump_polarity_desc.apply(f, axis=1)

# Plot the countplot with specific colors
plt.figure(figsize=(10, 10))
sns.set_style("whitegrid")

# Define the custom palette
custom_palette = ['blue', 'orange', 'green']
ax = sns.countplot(x="Sentiment_Type", data=df_tweet_trump_polarity_desc, palette=custom_palette)

# Add a title
plt.title("Sentiment Distribution of Donald Trump's Tweets")
plt.show()
```




```python
import torch
from transformers import pipeline
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns

# Initialize the BERT sentiment-analysis pipeline
classifier = pipeline('sentiment-analysis')

# Load the CSV file into a DataFrame
tweets = pd.read_csv('/content/tweets.csv')

# Check the first few rows of the dataset to ensure it has the 'handle' and 'text' columns
print(tweets.head())

# Function to analyze sentiment using BERT model
def analyze_and_plot_sentiment_with_model(tweets, handle, title):
    # Filter tweets based on the handle (HillaryClinton or realDonaldTrump)
    tweet_data = tweets.loc[tweets['handle'] == handle, ['text']]

    # Analyze sentiment for the tweets using BERT
    sentiment_list = []
    df_tweet_str = tweet_data['text'].astype(str)
    for tweet in df_tweet_str:
        sentiment = classifier(tweet)
        sentiment_list.append((tweet, sentiment[0]['label'], sentiment[0]['score']))

    # Create a DataFrame for sentiment analysis results
    df_tweet_polarity_desc = pd.DataFrame(sentiment_list, columns=['sentence', 'sentiment', 'confidence'])

    # Map sentiment labels to more friendly terms (if necessary)
    sentiment_mapping = {'POSITIVE': 'Positive', 'NEGATIVE': 'Negative'}
    df_tweet_polarity_desc['Sentiment_Type'] = df_tweet_polarity_desc['sentiment'].map(sentiment_mapping)

    # Plot the sentiment distribution
    plt.figure(figsize=(10, 10))
    sns.set_style("whitegrid")

    # Define the custom color palette
    custom_palette = ['blue', 'orange']
    ax = sns.countplot(x="Sentiment_Type", data=df_tweet_polarity_desc, palette=custom_palette)

    # Add a title to the plot
    plt.title(title)
    plt.show()

# Analyze and plot sentiment for Hillary Clinton's tweets
analyze_and_plot_sentiment_with_model(tweets, 'HillaryClinton', "Sentiment Distribution of Hillary Clinton's Tweets")

# Analyze and plot sentiment for Donald Trump's tweets
analyze_and_plot_sentiment_with_model(tweets, 'realDonaldTrump', "Sentiment Distribution of Donald Trump's Tweets")
```



“PRESENTATION”