

Twitter Sentiment Analysis

Step 1:Importing the libraries

```
In [7]: import numpy as np
import pandas as pd
import re
import nltk
from nltk.corpus import stopwords
from nltk.stem import WordNetLemmatizer
from wordcloud import WordCloud
import matplotlib.pyplot as plt
from plotly.subplots import make_subplots
import plotly.express as px
import plotly.graph_objects as go
import plotly.figure_factory as ff
from sklearn.pipeline import Pipeline
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import accuracy_score
from sklearn.model_selection import GridSearchCV

In [8]: col_names = ['ID', 'Entity', 'Sentiment', 'Content']
tr_df = pd.read_csv("C:\\Users\\Ankit\\Desktop\\Data Science\\Twitter Sentiment Analysis\\Dataset\\twitter_training.csv", names=col_names)
te_df = pd.read_csv("C:\\Users\\Ankit\\Desktop\\Data Science\\Twitter Sentiment Analysis\\Dataset\\twitter_validation.csv", names=col_names)

In [9]: tr_df.head()
```

	ID	Entity	Sentiment	Content
0	2401	Borderlands	Positive	im getting on borderlands and i will murder yo...
1	2401	Borderlands	Positive	I am coming to the borders and I will kill you...
2	2401	Borderlands	Positive	im getting on borderlands and i will kill you ...
3	2401	Borderlands	Positive	im coming on borderlands and i will murder you...
4	2401	Borderlands	Positive	im getting on borderlands 2 and i will murder ...

```
In [10]: tr_df.isnull().sum()

Out[10]: ID          0
Entity      0
Sentiment   0
Content    686
dtype: int64

In [11]: tr_df.dropna(subset=['Content'], inplace=True)

In [13]: tr_df['Sentiment'] = tr_df['Sentiment'].replace('Irrelevant', 'Neutral')
te_df['Sentiment'] = te_df['Sentiment'].replace('Irrelevant', 'Neutral')
```

2.Exploratory Data Analysis

```
In [14]: import plotly.express as px

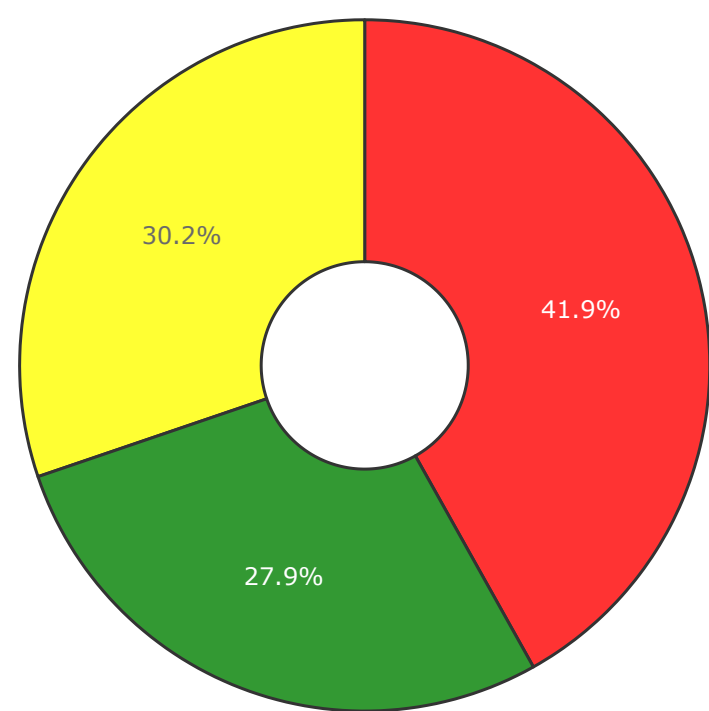
sentiment_counts = tr_df['Sentiment'].value_counts()

fig = px.pie(sentiment_counts,
             values=sentiment_counts.values,
             names=sentiment_counts.index,
             title='Sentiment Distribution',
             hole=0.3,
             color_discrete_sequence=['red', 'yellow', 'green'])

fig.update_traces(marker_line_color='black',
                  marker_line_width=1.5,
                  opacity=0.8)

fig.show()
```

Sentiment Distribution



Distribution of Entity

```
In [17]: import plotly.express as px

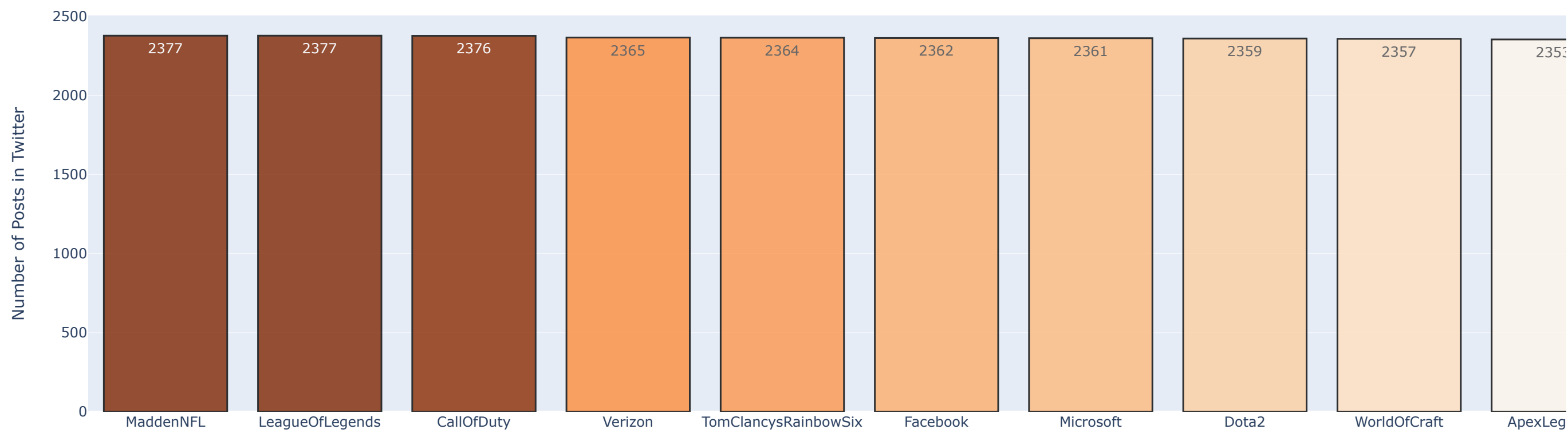
top10_entity_counts = tr_df['Entity'].value_counts().head(10)

fig = px.bar(top10_entity_counts,
             x=top10_entity_counts.index,
             y=top10_entity_counts.values,
             text=top10_entity_counts.values,
             color=top10_entity_counts.values,
             color_continuous_scale='oranges',
             labels={'x': 'Entity', 'y': 'Number of Posts in Twitter'},
             title='Top 10 Twitter Entity Distribution')
```

```
fig.update_traces(marker_line_color='black',
                  marker_line_width=1.5,
                  opacity=0.8)

fig.show()
```

Top 10 Twitter Entity Distribution



```
In [29]: import plotly.graph_objects as go

top3_entity_df = tr_df['Entity'].value_counts().head(3).index.tolist()
sentiment_labels = ['Negative', 'Neutral', 'Positive']
sentiment_colors = ['red', 'grey', 'green']

fig = go.Figure(
    data=[
        go.Pie(
            labels=sentiment_counts.index,
            values=sentiment_counts.values,
            textinfo='percent+value+label',
            marker=dict(colors=sentiment_colors, line=dict(color='black', width=1.5)),
            textposition='auto',
        )
    ],
    layout=go.Layout(
        title_text='Sentiment Distribution by Top 3 Entities',
        grid={'rows': 1, 'columns': 3},
        showlegend=False,
    ),
)

fig.show()
```

Sentiment Distribution by Top 3 Entities

