

task-4

April 30, 2025

```
[7]: import pandas as pd
import zipfile
import os

# Path to your zip file
zip_path = r"C:\pypractice\twitter_training.csv.zip"

# Extract the zip file in the same directory
extract_path = r"C:\pypractice"
with zipfile.ZipFile(zip_path, 'r') as zip_ref:
    zip_ref.extractall(extract_path)

# Now read the extracted CSV
csv_path = os.path.join(extract_path, "twitter_training.csv")
df = pd.read_csv(csv_path)

# Display the first few rows
df.columns = ['ID', 'Topic', 'Sentiment', 'Message']

# Preview the cleaned data
print(df.head())
print(df.columns)
```

```
      ID      Topic Sentiment \
0  2401  Borderlands  Positive
1  2401  Borderlands  Positive
2  2401  Borderlands  Positive
3  2401  Borderlands  Positive
4  2401  Borderlands  Positive

      Message
0  I am coming to the borders and I will kill you...
1  im getting on borderlands and i will kill you ...
2  im coming on borderlands and i will murder you...
3  im getting on borderlands 2 and i will murder ...
4  im getting into borderlands and i can murder y...
Index(['ID', 'Topic', 'Sentiment', 'Message'], dtype='object')
```

```

[10]: import pandas as pd

# Load the data

df['Message'] = df['Message'].astype(str).str.lower()

from textblob import TextBlob

# Define a function to get sentiment polarity
def get_sentiment_polarity(text):
    return TextBlob(text).sentiment.polarity

# Apply sentiment scoring
df['Polarity'] = df['Message'].apply(get_sentiment_polarity)

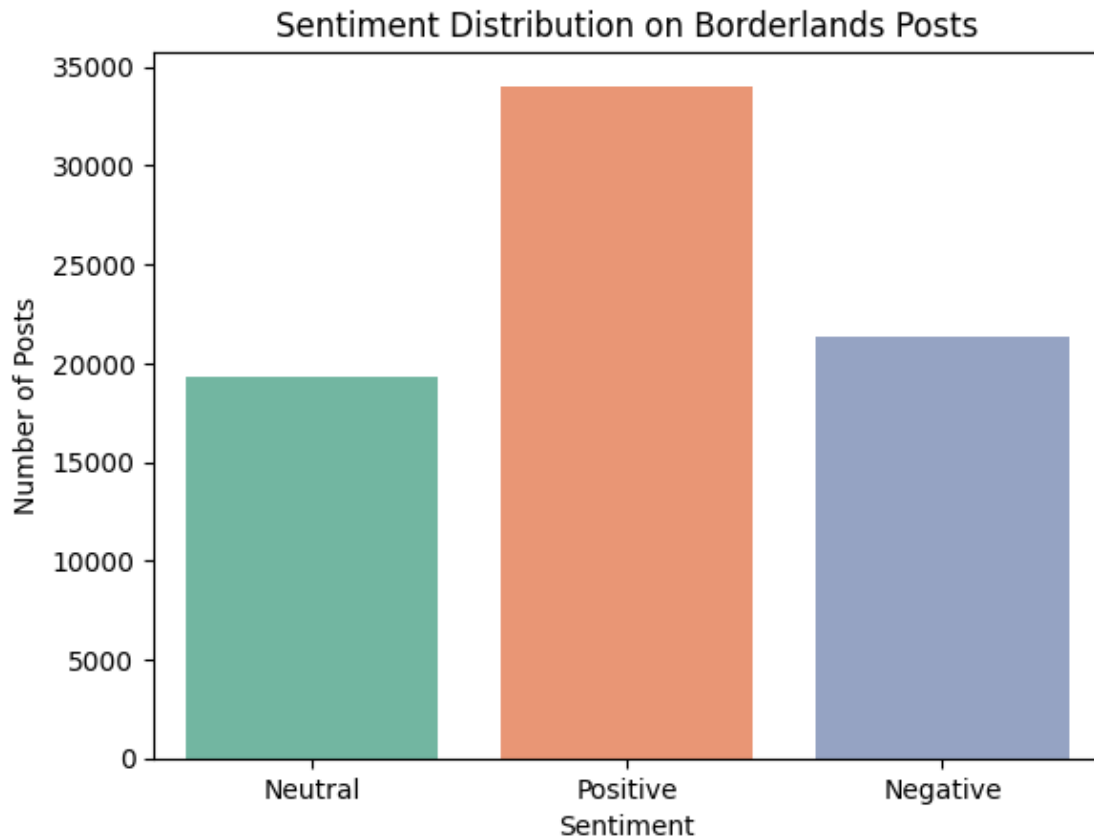
# Classify sentiment
df['Sentiment_Label'] = df['Polarity'].apply(
    lambda x: 'Positive' if x > 0 else ('Negative' if x < 0 else 'Neutral')
)

import seaborn as sns
import matplotlib.pyplot as plt

# Plot with hue explicitly set to the same as x
sns.countplot(x='Sentiment_Label', hue='Sentiment_Label', data=df,
    palette='Set2', legend=False)

plt.title('Sentiment Distribution on Borderlands Posts')
plt.xlabel('Sentiment')
plt.ylabel('Number of Posts')
plt.show()

```



```
[12]: from wordcloud import WordCloud

# Combine all messages into one string
all_text = ' '.join(df['Message'])

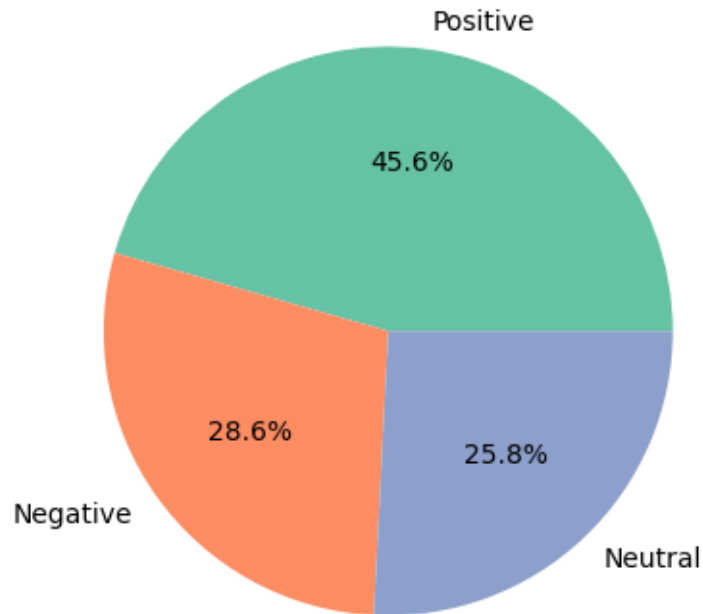
# Generate word cloud
wordcloud = WordCloud(width=800, height=400, background_color='white').
    generate(all_text)

# Plot
plt.figure(figsize=(10, 5))
plt.imshow(wordcloud, interpolation='bilinear')
plt.axis('off')
plt.title('Common Words in Borderlands Posts')
plt.show()
```

[illegible]

```
#pie chart of sentiment distribution
df['Sentiment_Label'].value_counts().plot.pie(
    autopct='%1.1f%%',
    colors=sns.color_palette('Set2'),
    title='Sentiment Share of Borderlands Posts',
    ylabel=''
)
plt.show()
```

Sentiment Share of Borderlands Posts



```
[14]: # Common aggressive terms
aggressive_words = ['kill', 'murder', 'death', 'blood', 'die']
for word in aggressive_words:
    df[word] = df['Message'].str.contains(word).astype(int)

# Count how many posts contain each aggressive word
aggression_df = df[aggressive_words].sum().sort_values(ascending=False)
print(aggression_df)
```

```
kill      1379
die       952
death     308
blood     260
murder     99
dtype: int64
```

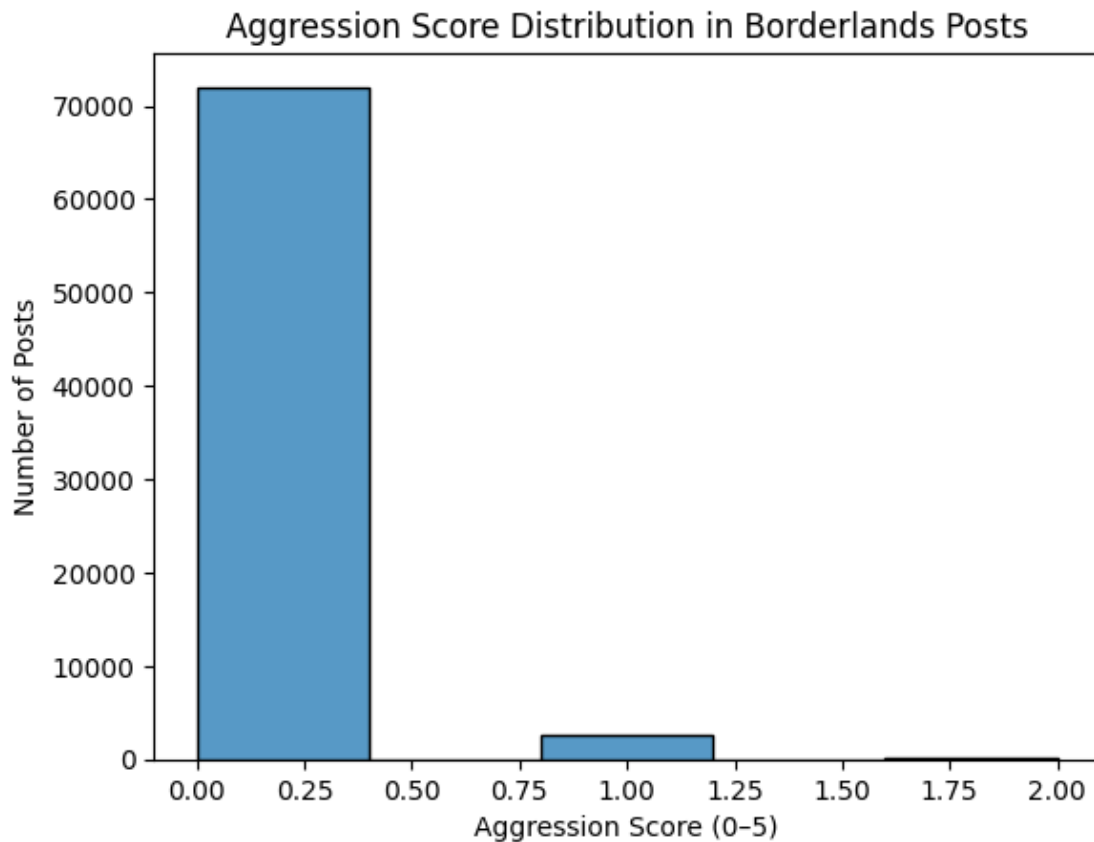
```
[17]: #Aggression Index (Custom Feature)
aggressive_words = ['murder', 'kill', 'die', 'blood', 'violence']
df['Aggression_Score'] = df['Message'].apply(lambda x: sum(word in x for word_
    ↪in aggressive_words))

# Histogram of aggression scores
```

```

sns.histplot(df['Aggression_Score'], bins=5, kde=False)
plt.title('Aggression Score Distribution in Borderlands Posts')
plt.xlabel('Aggression Score (0-5)')
plt.ylabel('Number of Posts')
plt.show()

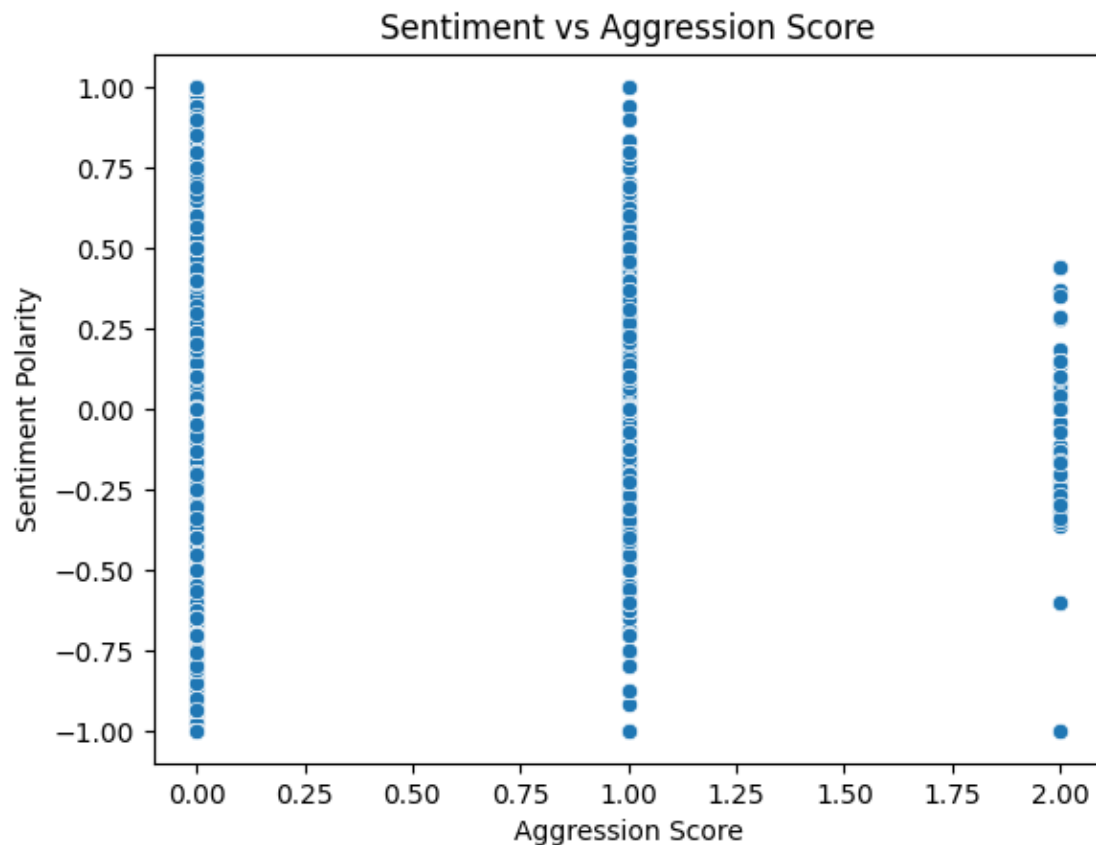
```



```

[18]: #Correlation between Sentiment and Aggression score
# Scatterplot: sentiment polarity vs aggression score
sns.scatterplot(x='Aggression_Score', y='Polarity', data=df)
plt.title('Sentiment vs Aggression Score')
plt.xlabel('Aggression Score')
plt.ylabel('Sentiment Polarity')
plt.show()

```



```
[ ]: #Wordcloud by sentiment Label (Visualizing Tone difference)

from wordcloud import WordCloud

# Positive wordcloud
pos_text = ' '.join(df[df['Sentiment_Label'] == 'Positive']['Message'])
WordCloud(width=600, height=300, background_color='white').generate(pos_text).
    ↪to_image().show()
```

```
[ ]:
```