Prodigy InfoTech Internship

Task 4:

Analyse and visualize sentiment patterns in social media data to understand public opinion and attitudes towards specific topics or brands.

Sample Dataset: [Twitter Sentiment Analysis](#)

Analysation & Visualisation of sentiment patterns in **social media data**

Loading Libraries and Dataset:

```python
[122]  1 import pandas as pd
       2 import matplotlib.pyplot as plt
       3 import seaborn as sns
       4 import warnings
       5 warnings.filterwarnings('ignore')
       6 from textblob import TextBlob
```

```python
[123]  1 sns.set_theme(context='notebook', style='whitegrid', palette='muted')
```

```python
[124]  1 !pip install textblob
```
```
Requirement already satisfied: textblob in /usr/local/lib/python3.10/dist-packages (0.17.1)
Requirement already satisfied: nltk>=3.1 in /usr/local/lib/python3.10/dist-packages (from textblob) (3.8.1)
Requirement already satisfied: click in /usr/local/lib/python3.10/dist-packages (from nltk>=3.1->textblob) (8.1.7)
Requirement already satisfied: joblib in /usr/local/lib/python3.10/dist-packages (from nltk>=3.1->textblob) (1.4.2)
Requirement already satisfied: regex>=2021.8.3 in /usr/local/lib/python3.10/dist-packages (from nltk>=3.1->textblob) (2023.12.25)
Requirement already satisfied: tqdm in /usr/local/lib/python3.10/dist-packages (from nltk>=3.1->textblob) (4.66.4)
```

```python
[125]  1 col_names = ['Id', 'Entity', 'Sentiment', 'Tweet']
       2 valid_data = pd.read_csv("/content/drive/MyDrive/Project_Datasets/Sentiment_Analysis/twitter_validation.csv", names=col_names)
       3 train_data = pd.read_csv("/content/drive/MyDrive/Project_Datasets/Sentiment_Analysis/twitter_training.csv", names=col_names)
```

Understanding the shape of the data:

```python
[126]  1 train_data.head()
```

	Id	Entity	Sentiment	Tweet
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 ...

By: Aman Kumar Jha
[LinkedIn](#)

```
[127]   1 valid_data.head()
```

	Id	Entity	Sentiment	Tweet	
0	3364	Facebook	Irrelevant	I mentioned on Facebook that I was struggling ...	
1	352	Amazon	Neutral	BBC News - Amazon boss Jeff Bezos rejects clai...	
2	8312	Microsoft	Negative	@Microsoft Why do I pay for WORD when it funct...	
3	4371	CS-GO	Negative	CSGO matchmaking is so full of closet hacking,...	
4	4433	Google	Neutral	Now the President is slapping Americans in the...	

Data Cleaning:

```
[128]   1 # Check for missing values
        2 train_data.isnull().sum()
        3 valid_data.isnull().sum()
```

```
Id          0
Entity      0
Sentiment   0
Tweet       0
dtype: int64
```

```
[129]   1 # Checking the distribution of sentiment labels
        2 train_sentiment_distribution = train_data.iloc[:, 2].value_counts()
        3 valid_sentiment_distribution = valid_data.iloc[:, 2].value_counts()
        4 train_sentiment_distribution,valid_sentiment_distribution
```

```
(Sentiment
 Negative     22542
 Positive     20832
 Neutral      18318
 Irrelevant   12990
 Name: count, dtype: int64,
 Sentiment
 Neutral        285
 Positive       277
 Negative       266
 Irrelevant     172
 Name: count, dtype: int64)
```

```
   1 train_data.iloc[:, 1].nunique()
```

```
32
```

```
[131]   1 # Remove duplicate rows from the training set
        2 train_data_cleaned = train_data.drop_duplicates()
```

```
   1 # Drop rows with missing tweet/message values
   2 train_data_cleaned.dropna(subset=[train_data.columns[3]])
```

By: Aman Kumar Jha
LinkedIn

|	Id	Entity	Sentiment	Tweet
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 ...
...	...	...	...	...
74677	9200	Nvidia	Positive	Just realized that the Windows partition of my...
74678	9200	Nvidia	Positive	Just realized that my Mac window partition is ...
74679	9200	Nvidia	Positive	Just realized the windows partition of my Mac ...
74680	9200	Nvidia	Positive	Just realized between the windows partition of...
74681	9200	Nvidia	Positive	Just like the windows partition of my Mac is I...

71656 rows × 4 columns

```
[133]    1 # Verify the cleaning
         2 remaining_duplicates_train = train_data_cleaned.duplicated().sum()
         3 remaining_missing_train = train_data_cleaned.isnull().sum()
         4
         5 remaining_duplicates_train, remaining_missing_train
```

```
(0,
 Id            0
 Entity        0
 Sentiment     0
 Tweet       326
 dtype: int64)
```

Data Exploration:

```
 1 # Set up the plots
 2 fig, ax = plt.subplots(1, 2, figsize=(15, 5))
 3
 4 # Plot sentiment distribution for training data
 5 sns.countplot(data=train_data_cleaned, x=train_data_cleaned.columns[2], order=['Positive', 'Negative', 'Neutral', 'Irrelevant'], ax=ax[0])
 6 ax[0].set_title('Sentiment Distribution in Training Data')
 7 ax[0].set_ylabel('Count')
 8 ax[0].set_xlabel('Sentiment')
 9
10 # Plot sentiment distribution for validation data
11 sns.countplot(data=valid_data, x=valid_data.columns[2], order=['Positive', 'Negative', 'Neutral', 'Irrelevant'], ax=ax[1])
12 ax[1].set_title('Sentiment Distribution in Validation Data')
13 ax[1].set_ylabel('Count')
14 ax[1].set_xlabel('Sentiment')
15
16 plt.tight_layout()
17 plt.show()
```
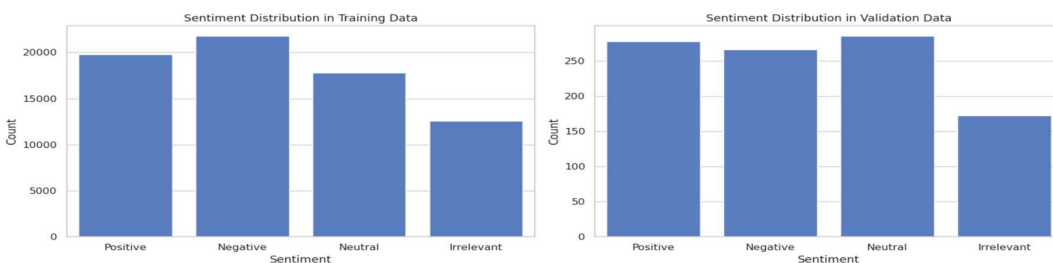
By: Aman Kumar Jha
LinkedIn

```python
1  # Bar Chart representation of top 10 most engaged entities on tweets
2  # Count the occurrences of each category
3  entity_counts = valid_data['Entity'].value_counts()
4
5  # Select the top 10 categories
6  top_10_entity = entity_counts.head(10)
7
8  # Define colors for the bars
9  colors = ['#1f77b4', '#ff7f0e', '#2ca02c', '#d62728', '#9467bd', '#8c564b', '#e377c2', '#7f7f7f', '#bcbd22', '#17becf']
10
11  # Plot a bar graph for the top 10 categories with colors
12  plt.figure(figsize=(10, 6))
13  top_10_entity.plot(kind='bar', color=colors)
14
15  # Remove the annotate part to not show values on top of each bar
16
17  plt.xlabel("Entities")
18  plt.ylabel('Tweet Count')
19  plt.title('Top 10 Entities with Highest Tweet Engagement')
20  plt.xticks(rotation=45, ha='right')  # Adjust rotation for better readability
21  plt.show()
22
23  #The highest tweet count is of RedDeadRedeemption entity which is 40, Lowest tweet count is of Fortnite.
```


Top 10 Entities with Highest Tweet Engagement
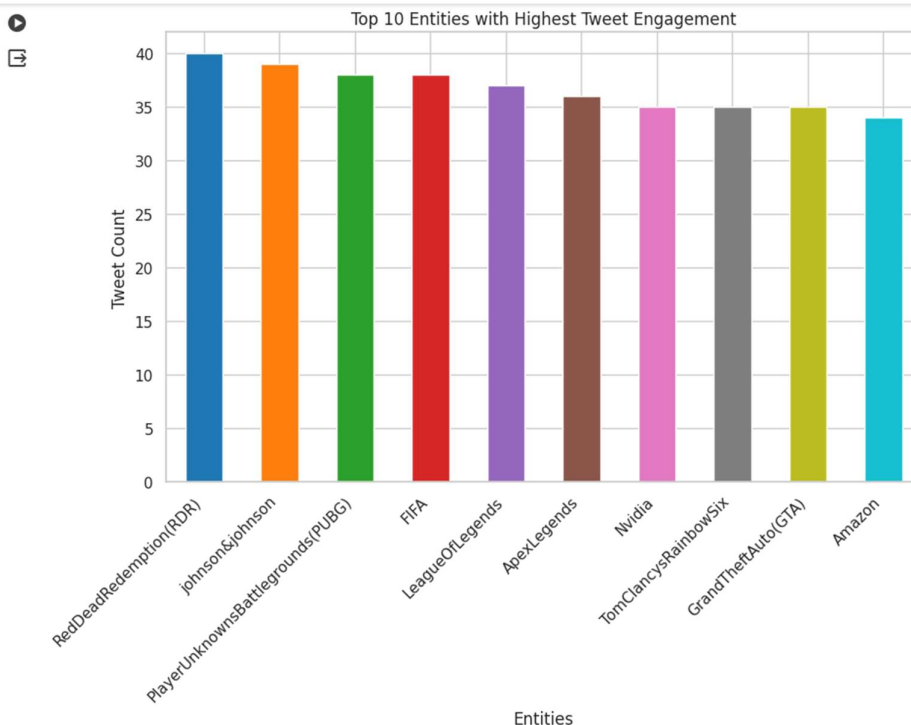
```python
1  #EXPLORATORY DATA ANALYSIS
2  #Pie Chart representation of sentiment distribution
3  # Count the occurrences of each category
4  category_counts =valid_data['Sentiment'].value_counts()
5
6  # Plot a pie chart
7  plt.figure(figsize=(6, 8))
8  plt.pie(category_counts, labels=category_counts.index, autopct='%1.1f%%', startangle=140)
9  plt.axis('equal')
10
11  plt.title('Distribution of Sentiments')
12  plt.show()
13
14  #There are 27.7% positive sentiment texts, 26.6% negative sentiment texts,
15  #17.1% Irrelevant sentiment texts and 28.5% neutral sentiment texts in the dataset
```

By: Aman Kumar Jha
LinkedIn
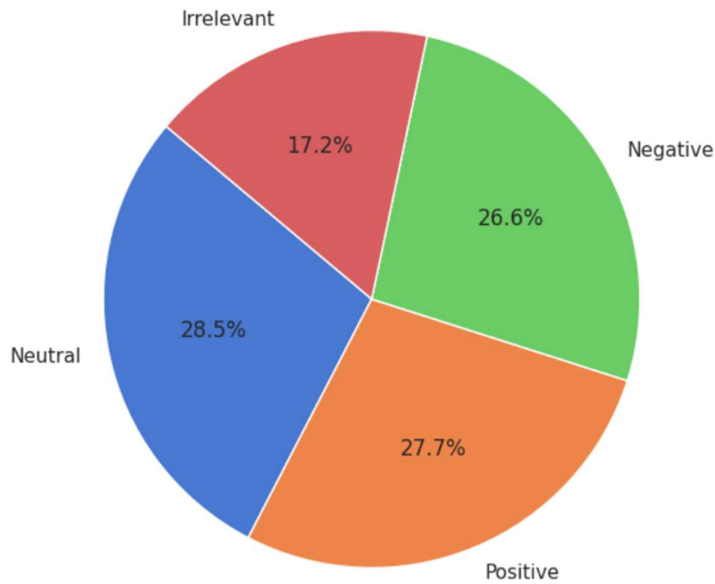
```
1 plt.figure(figsize=(15, 7))
2
3 # Plot entity distribution for training data
4 sns.countplot(data=train_data_cleaned, y=train_data_cleaned.columns[1], order=train_data_cleaned[train_data_cleaned.columns[1]].value_counts().index, palette='viridis')
5 plt.title('Entity Distribution in Training Data')
6 plt.xlabel('Count')
7 plt.ylabel('Entity')
8 plt.tight_layout()
9 plt.show()
```
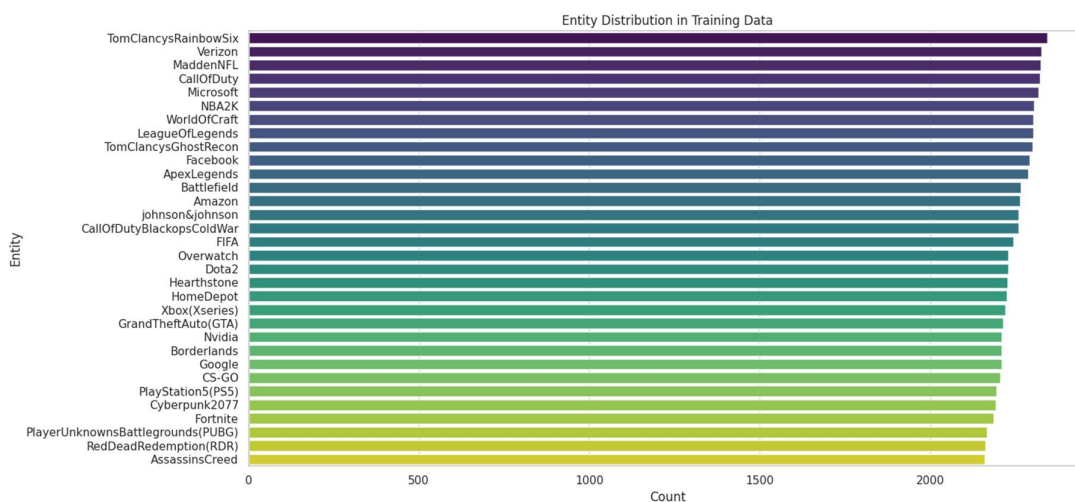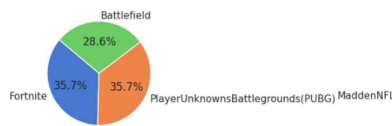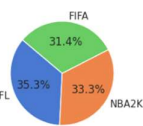
```
1 # Pie chart representation of the top three entities in each sentiment category
2 # Group the data by 'tweet_category' and 'entity' and count occurrences
3 category_entity_counts = valid_data.groupby(['Sentiment', 'Entity']).size().unstack(fill_value=0)
4
5 # Create a list to store the top 3 entities for each category
6 top_three_entities = []
7
8 # Iterate through each category
9 for category in category_entity_counts.index:
10     top_entities = category_entity_counts.loc[category].nlargest(3)
11     top_three_entities.append(top_entities)
12
13 # Determine the number of subplots based on the number of categories
14 num_categories = len(category_entity_counts.index)
15
16 # Create subplots for each category
17 plt.figure(figsize=(5 * num_categories, 5))
18 for i, (category, top_entities) in enumerate(zip(category_entity_counts.index, top_three_entities), start=1):
19     plt.subplot(1, num_categories, i)
20     plt.pie(top_entities, labels=top_entities.index, autopct='%1.1f%%', startangle=140)
21     plt.axis('equal')
22     plt.title(f'Top 3 Entities in {category}')
23
24 plt.tight_layout()
25 plt.show()
```
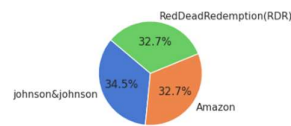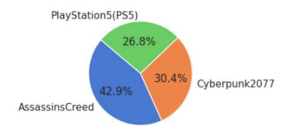

Top 3 Entities in Irrelevant | Top 3 Entities in Negative | Top 3 Entities in Neutral | Top 3 Entities in Positive

Building Basic Model & Testing:

```
1 train_data_cleaned['message_length'] = train_data_cleaned[train_data_cleaned.columns[3]].fillna('').apply(len)
2 valid_data['message_length'] = valid_data[valid_data.columns[3]].fillna('').apply(len)
```

```
[140]  1 fig, ax = plt.subplots(1, 2, figsize=(15, 5))
        2
        3 # Plot message length distribution for training data
        4 sns.histplot(train_data_cleaned['message_length'], bins=50, ax=ax[0], color='skyblue')
        5 ax[0].set_title('Message Length Distribution in Training Data')
        6 ax[0].set_ylabel('Frequency')
        7 ax[0].set_xlabel('Message Length')
        8
        9 # Plot message length distribution for validation data
       10 sns.histplot(valid_data['message_length'], bins=50, ax=ax[1], color='salmon')
       11 ax[1].set_title('Message Length Distribution in Validation Data')
       12 ax[1].set_ylabel('Frequency')
       13 ax[1].set_xlabel('Message Length')
       14
       15 plt.tight_layout()
       16 plt.show()
```

By: Aman Kumar Jha
LinkedIn
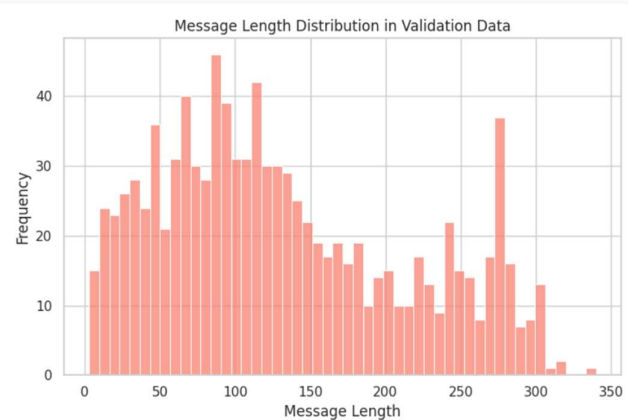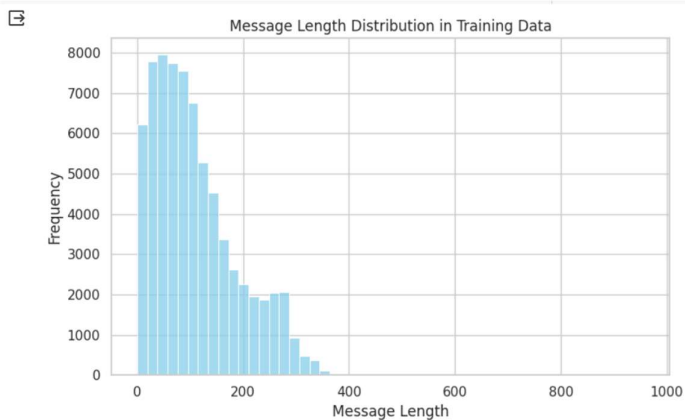
```
1 plt.figure(figsize=(18, 10))
2
3 # Plot sentiment distribution by entity
4 sns.countplot(data=train_data_cleaned, y=train_data_cleaned.columns[1], hue=train_data_cleaned.columns[2], order=train_data_cleaned[train_data_cleaned.columns[1]].value_counts().index, palette='viridis')
5 plt.title('Sentiment Distribution by Entity in Training Data')
6 plt.xlabel('Count')
7 plt.ylabel('Entity')
8 plt.legend(title='Sentiment')
9 plt.tight_layout()
10 plt.show()
```


Sentiment Distribution by Entity in Training Data
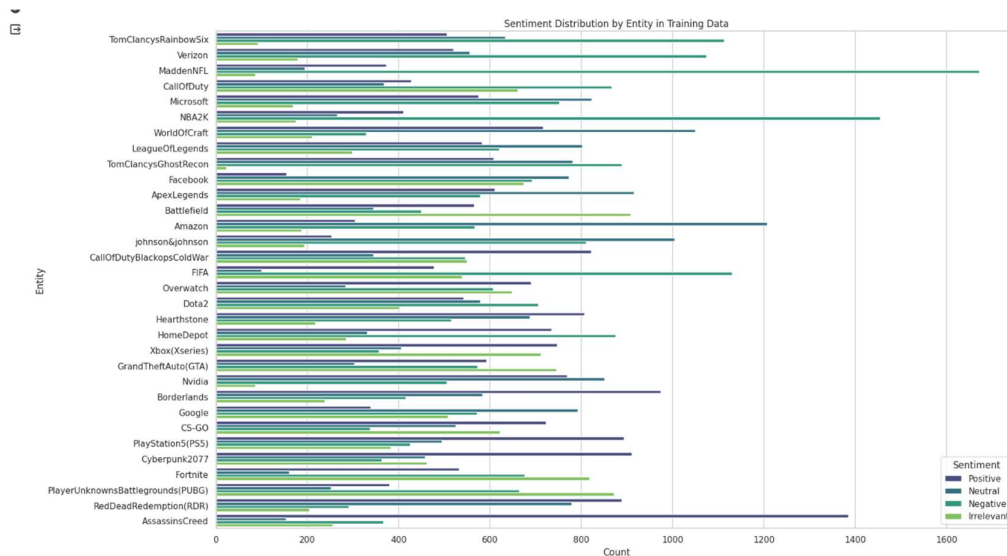
```
1 fig, ax = plt.subplots(1, 2, figsize=(15, 5))
2
3 # Plot message length distribution by sentiment for training data
4 sns.boxplot(data=train_data_cleaned, x=train_data_cleaned.columns[2], y='message_length', ax=ax[0], palette='viridis', order=['Positive', 'Negative', 'Neutral', 'Irrelevant'])
5 ax[0].set_title('Message Length Distribution by Sentiment in Training Data')
6 ax[0].set_ylabel('Message Length')
7 ax[0].set_xlabel('Sentiment')
8 ax[0].set_ylim(0, 300)
9
10 # Plot message length distribution by sentiment for validation data
11 sns.boxplot(data=valid_data, x=valid_data.columns[2], y='message_length', ax=ax[1], palette='viridis', order=['Positive', 'Negative', 'Neutral', 'Irrelevant'])
12 ax[1].set_title('Message Length Distribution by Sentiment in Validation Data')
13 ax[1].set_ylabel('Message Length')
14 ax[1].set_xlabel('Sentiment')
15 ax[1].set_ylim(0, 300)
16
17 plt.tight_layout()
18 plt.show()
```
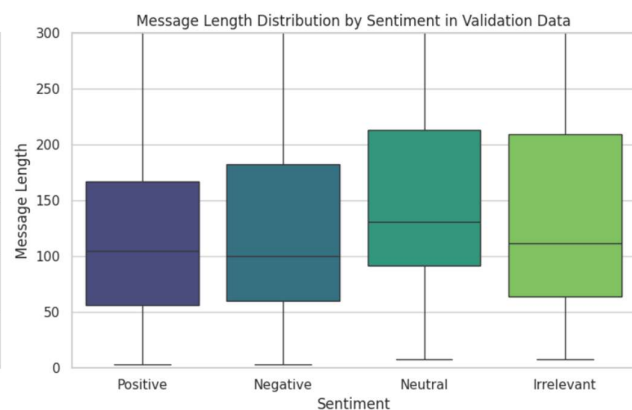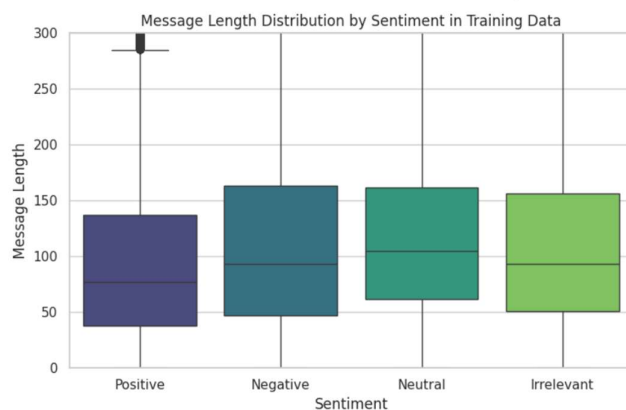


```
[143]  1 # Define a function to handle non-string values
       2 def analyze_sentiment(text):
       3     if isinstance(text, str):
       4         return TextBlob(text).sentiment.polarity
       5     else:
       6         return 0.0
```

By: Aman Kumar Jha

LinkedIn

```
[144]  1 # Perform sentiment analysis
       2 train_data['Polarity'] = train_data['Tweet'].apply(analyze_sentiment)
       3
       4 # Categorize sentiment
       5 train_data['Sentiment_Label'] = train_data['Polarity'].apply(lambda x: 'Positive' if x > 0 else 'Negative' if x < 0 else 'Neutral')
```

```
[145]  1 # Perform sentiment analysis
       2 valid_data['Polarity'] = valid_data['Tweet'].apply(analyze_sentiment)
       3
       4 # Categorize sentiment
       5 valid_data['Sentiment_Label'] = valid_data['Polarity'].apply(lambda x: 'Positive' if x > 0 else 'Negative' if x < 0 else 'Neutral')
```

```
1 # Analyze sentiment distribution
2 plt.figure(figsize=(10, 6))
3 sns.countplot(train_data['Sentiment_Label'])
4 plt.title('Sentiment Analysis')
5 plt.xlabel('Sentiment')
6 plt.ylabel('Count')
7 plt.show()
```
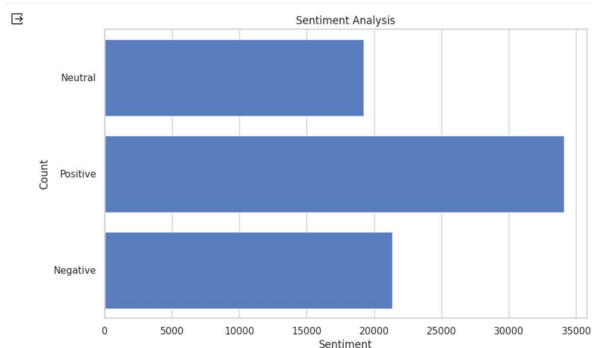


```
1 # Analyze sentiment by topic
2 plt.figure(figsize=(15,8))
3 sentiment_by_topic = train_data.groupby(['Entity', 'Sentiment_Label']).size().unstack(fill_value=0)
4 sentiment_by_topic.plot(kind='bar', stacked=True)
5 plt.title('Sentiment by Entity')
6 plt.xlabel('Entity')
7 plt.ylabel('Count')
8 plt.show()
```

<Figure size 1500x800 with 0 Axes>
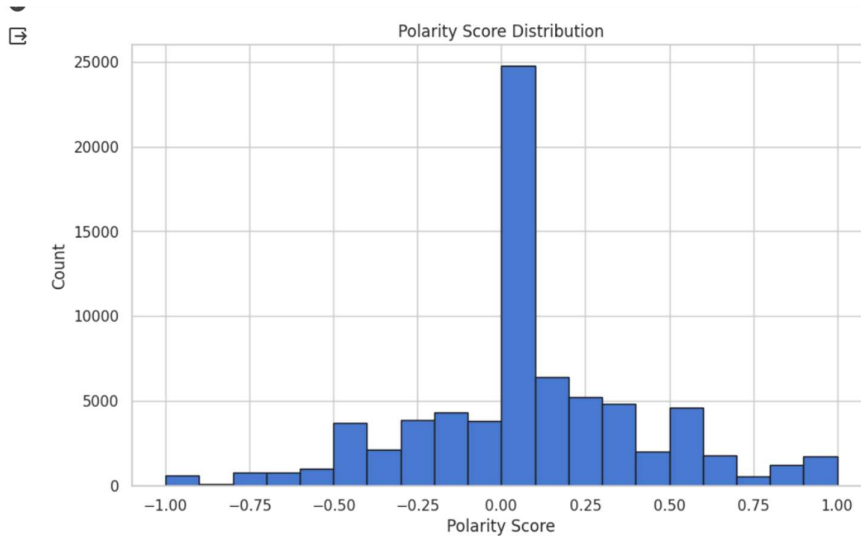

By: Aman Kumar Jha
LinkedIn
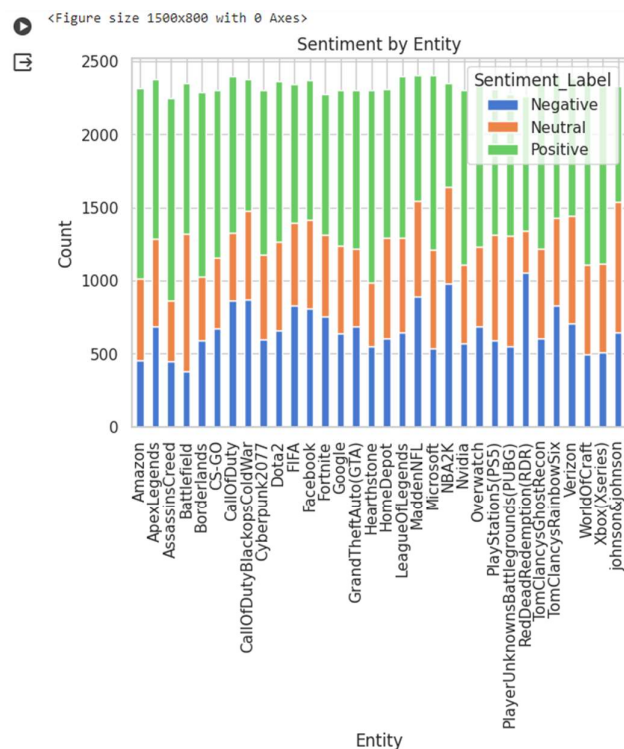
```
1 plt.figure(figsize=(10, 6))
2 plt.hist(train_data['Polarity'], bins=20, edgecolor='k')
3 plt.title('Polarity Score Distribution')
4 plt.xlabel('Polarity Score')
5 plt.ylabel('Count')
6 plt.show()
```
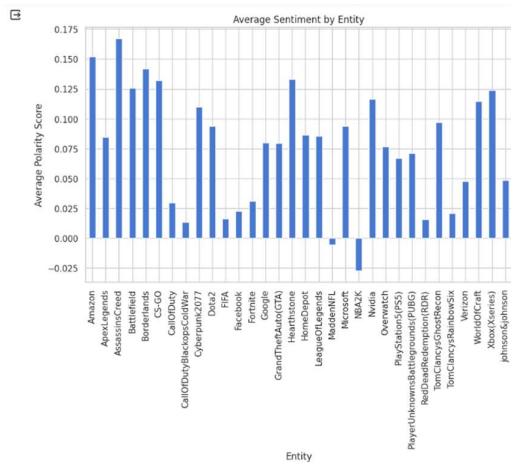

Polarity Score Distribution

```
1 plt.figure(figsize=(10, 6))
2 average_polarity_by_topic = train_data.groupby('Entity')['Polarity'].mean()
3 average_polarity_by_topic.plot(kind='bar')
4 plt.title('Average Sentiment by Entity')
5 plt.xlabel('Entity')
6 plt.ylabel('Average Polarity Score')
7 plt.show()
```


Average Sentiment by Entity

By: Aman Kumar Jha
LinkedIn

```
1 # Visualize the most positive and negative tweets
2 most_positive_tweet = train_data[train_data['Polarity'] == train_data['Polarity'].max()]['Tweet'].values[0]
3 most_negative_tweet = train_data[train_data['Polarity'] == train_data['Polarity'].min()]['Tweet'].values[0]
4
5 print('Most Positive Tweet:')
6 print(most_positive_tweet)
7
8 print('\nMost Negative Tweet:')
9 print(most_negative_tweet)
```

```
Most Positive Tweet:
Platinum is the best loot @Borderlands

Most Negative Tweet:
"What terrible bitch!"
```

```
1 # Visualize the most positive and negative tweets
2 most_positive_tweet = valid_data[valid_data['Polarity'] == valid_data['Polarity'].max()]['Tweet'].values[0]
3 most_negative_tweet = valid_data[valid_data['Polarity'] == valid_data['Polarity'].min()]['Tweet'].values[0]
4
5 print('Most Positive Tweet:')
6 print(most_positive_tweet)
7
8 print('\nMost Negative Tweet:')
9 print(most_negative_tweet)
```

```
Most Positive Tweet:
Best squad yet#pubg #pubgmobile #pubgkenya instagram.com/p/B-Obt_eAA4f/…

Most Negative Tweet:
@EAMaddenNFL franchise and face of the franchise are both terrible. @NFL #NFLdropEA #NFLDROPSEA
```

By: Aman Kumar Jha
LinkedIn