SENTIMENT ANALYSIS

mood

IMPORTING LIBRARIES

```
#importing libraries and reading the files
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
```

READING CSV FILES

```
a=pd.read_csv(r'C:\Users\Windows\Downloads\twitter_training.csv')
b=pd.read_csv(r'C:\Users\Windows\Downloads\twitter_validation.csv')
```

Setting column names

```
column_name=['tweetID','entity','sentiment','tweet_content']
a.columns=column_name
b.columns=column_name
df=pd.concat([a,b],ignore_index=False)
df.head()
```

```
   tweetID       entity sentiment  \
0     2401  Borderlands  Positive
1     2401  Borderlands  Positive
2     2401  Borderlands  Positive
3     2401  Borderlands  Positive
4     2401  Borderlands  Positive

                                       tweet_content
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...
```

QUICK VIEW OF DATA

```
#list columns in our dataframe
df.columns.tolist()
```

```
['tweetID', 'entity', 'sentiment', 'tweet_content']
```

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Index: 75680 entries, 0 to 998
Data columns (total 4 columns):
 #   Column         Non-Null Count  Dtype 
---  ------         --------------  ----- 
 0   tweetID        75680 non-null  int64 
 1   entity         75680 non-null  object
 2   sentiment      75680 non-null  object
 3   tweet_content  74994 non-null  object
dtypes: int64(1), object(3)
memory usage: 2.9+ MB
```

EDA

Looking for null Values

```
df.isnull().sum().sort_values(ascending = False)
```

```
tweet_content    686
tweetID            0
entity             0
sentiment          0
dtype: int64
```

Checking for dupilcates

```
df.duplicated().sum()
```

```
np.int64(3216)
```

Dropping null values and duplicated values

```
df.dropna(inplace = True)
df.drop_duplicates(inplace = True)
print("NULL VAUES:", "\n", df.isnull().sum())
print("Duplicate Values:", df.duplicated().sum())
```

```
NULL VAUES: 
 tweetID          0
entity           0
sentiment        0
tweet_content    0
dtype: int64
Duplicate Values: 0
```

Dropping Unneeded Columns

```
df.drop(columns=['tweetID','tweet_content'],inplace=True)
df.head()
```

```
        entity sentiment
0  Borderlands  Positive
1  Borderlands  Positive
2  Borderlands  Positive
3  Borderlands  Positive
4  Borderlands  Positive
```

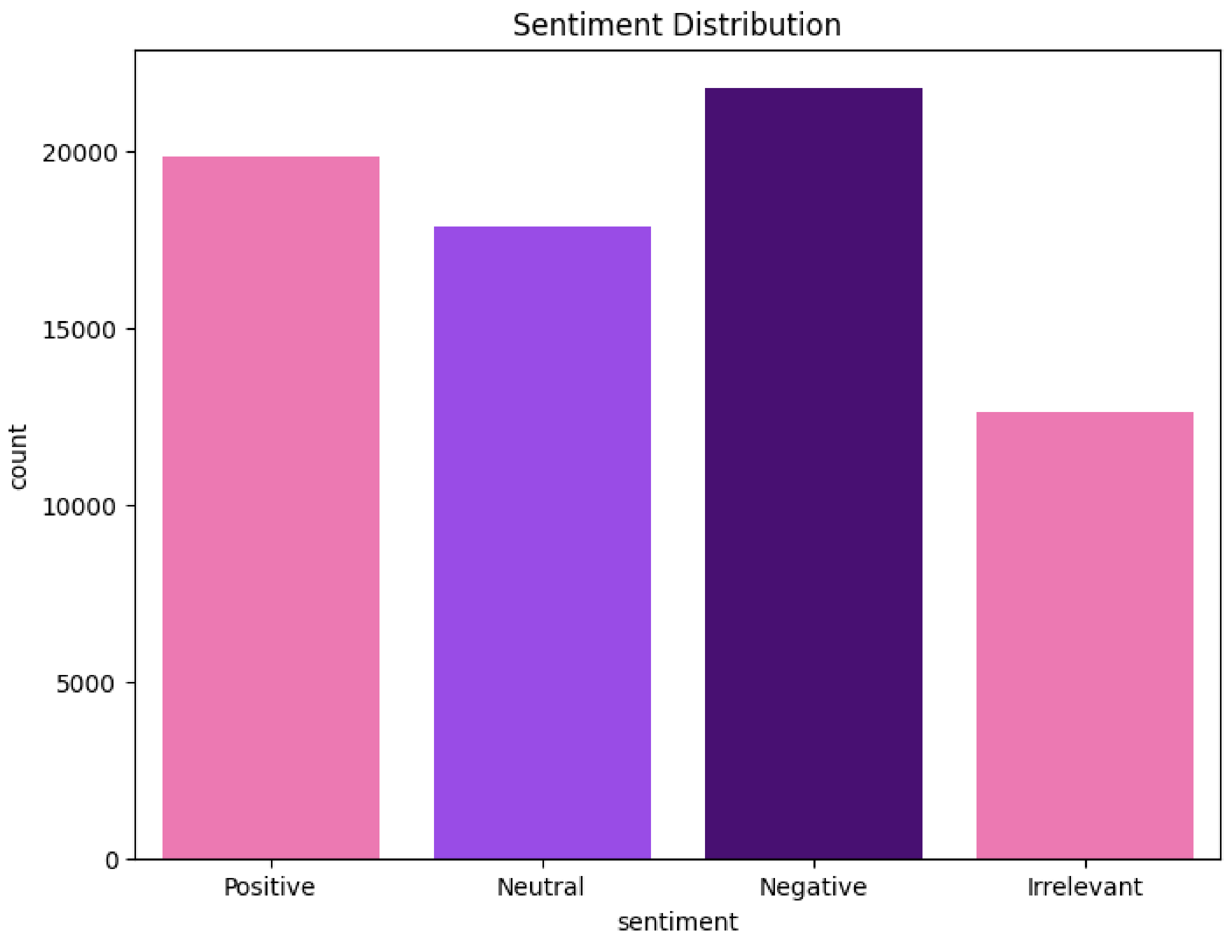
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Index: 72138 entries, 0 to 995
Data columns (total 2 columns):
 #   Column     Non-Null Count  Dtype 
---  ------     --------------  ----- 
 0   entity     72138 non-null  object
 1   sentiment  72138 non-null  object
dtypes: object(2)
memory usage: 1.7+ MB
```

VISUALISATION

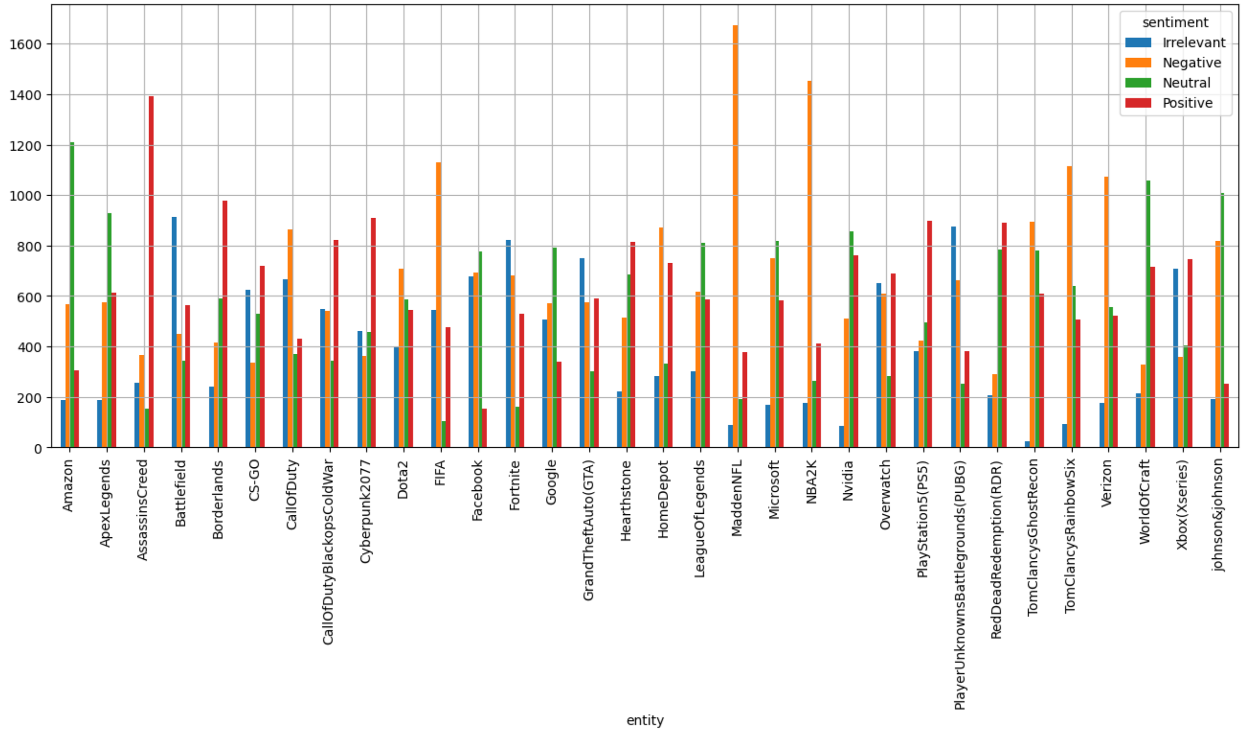
```
custom_palette = sns.color_palette(["#FF66B2", "#9933FF", "#4B0082"])

plt.figure(figsize=(8,6))
sns.countplot(x='sentiment', data=df, palette=custom_palette)
plt.title('Sentiment Distribution')
plt.show()
```

```
C:\Users\Windows\AppData\Local\Temp\ipykernel_11908\2557904149.py:4: 
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.

  sns.countplot(x='sentiment', data=df, palette=custom_palette)
C:\Users\Windows\AppData\Local\Temp\ipykernel_11908\2557904149.py:4: 
UserWarning: 
The palette list has fewer values (3) than needed (4) and will cycle, 
which may produce an uninterpretable plot.
  sns.countplot(x='sentiment', data=df, palette=custom_palette)
```

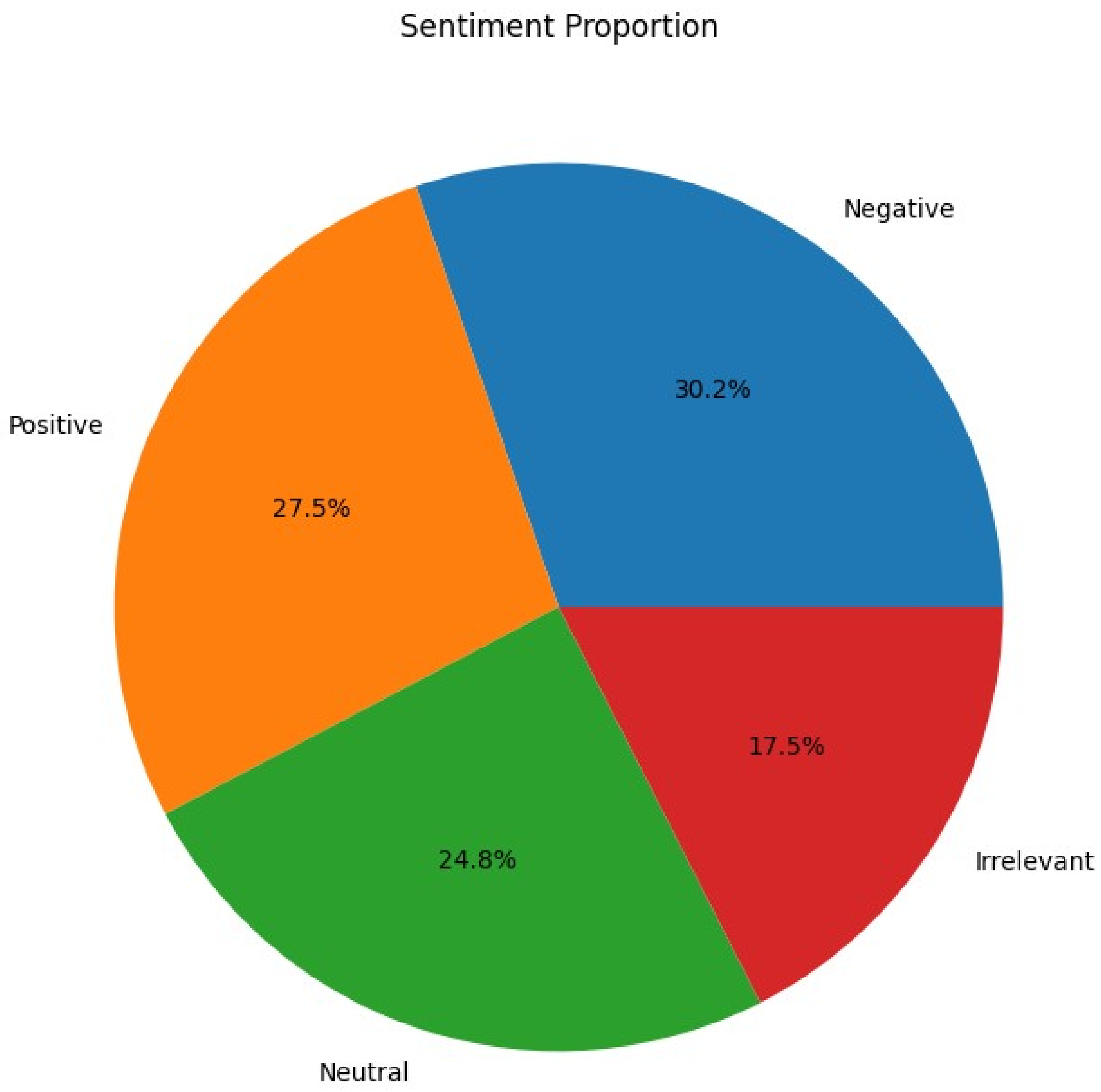


```
reactions_entities = pd.crosstab(df['entity'],df['sentiment'])
reactions_entities.plot(kind='bar', figsize=(16, 6),grid=True)
```

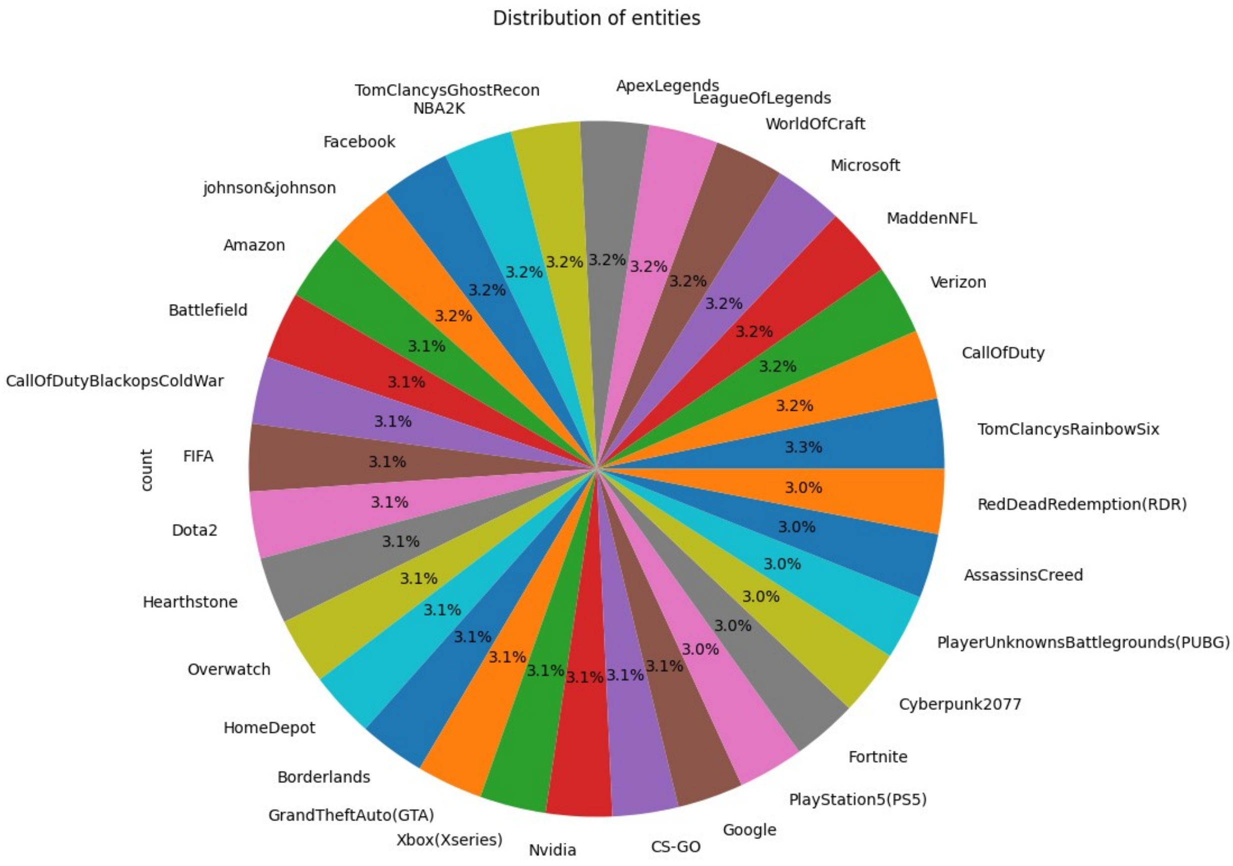
```
<Axes: xlabel='entity'>
```



```
df['sentiment'].value_counts().plot.pie(autopct='%1.1f%%',
figsize=(8,8), title='Sentiment Proportion')
plt.ylabel('')
plt.show()
```

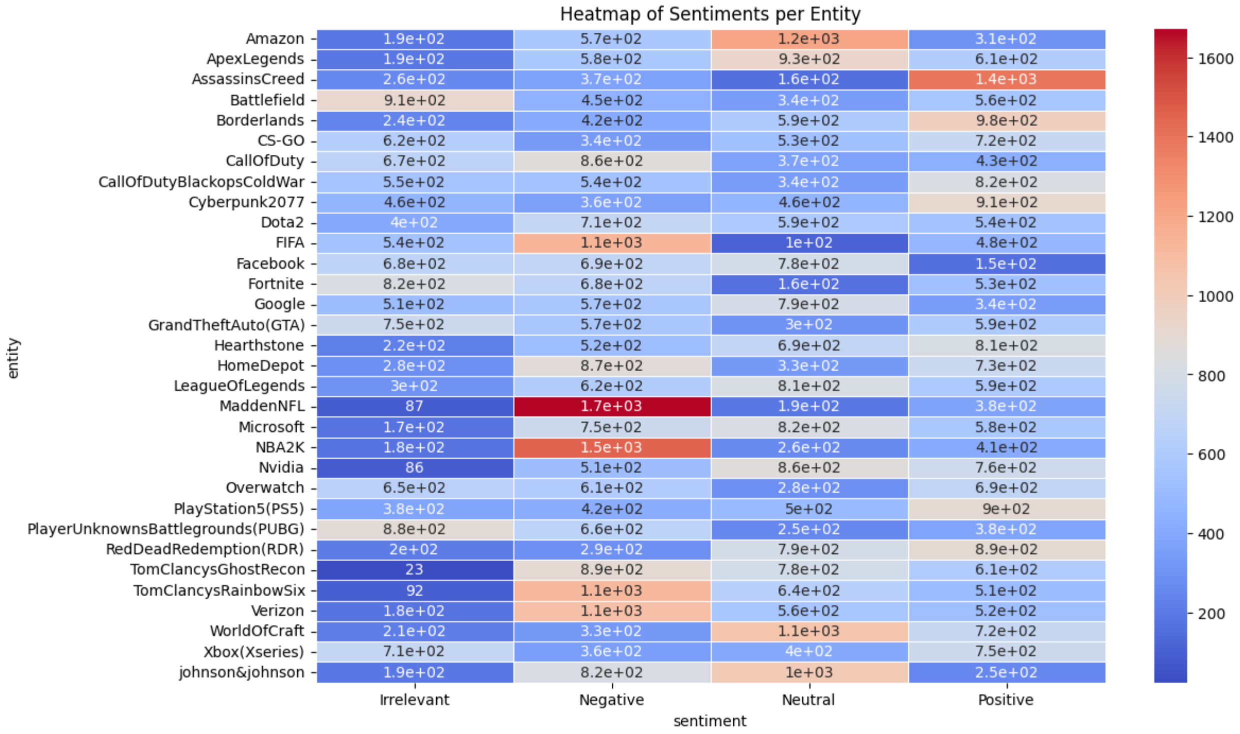


```
entity_content=df['entity'].value_counts()
entity_content.plot(kind='pie', autopct='%1.1f%%', figsize=(10, 12))
plt.title('Distribution of entities')

plt.show()
```



From the above figure the highest negative reactions in the MaddenNFL entity The highest irrelevant reactions in the Battlefield entity The highest neutral reactions in the Amazon entity The highest positive reactions in the AssassinsCreed entity

```
pivot_table = df.pivot_table(index='entity', columns='sentiment',
aggfunc='size', fill_value=0)
plt.figure(figsize=(12,8))
sns.heatmap(pivot_table, annot=True, cmap='coolwarm', linewidths=0.5)
plt.title('Heatmap of Sentiments per Entity')
plt.show()
```



The overall sentiment analysis reveals a dominant trend where either positive or negative opinions heavily influence the public's perception. If positive, it suggests satisfaction and approval, while a high negative sentiment indicates dissatisfaction or criticism towards the entities. Neutral sentiments reflect a lack of strong emotional engagement. These patterns help gauge general audience behavior and attitudes.OVET

THANK YOU !!!!