```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
```

```
 import pandas as pd
 column_names = ['ID', 'entity', 'sentiment', 'comment']
df = pd.read_csv('/content/drive/MyDrive/twitter_training.csv', header=0, names=column_names)
```

```
df.head()
```

	ID	entity	sentiment	comment
0	2401	Borderlands	Positive	I am coming to the borders and I will kill you...
1	2401	Borderlands	Positive	im getting on borderlands and i will kill you ...
2	2401	Borderlands	Positive	im coming on borderlands and i will murder you...
3	2401	Borderlands	Positive	im getting on borderlands 2 and i will murder ...
4	2401	Borderlands	Positive	im getting into borderlands and i can murder y...

EDA

```
df.shape
```

```
(74681, 4)
```

```
# Count of unique entities
entity_count = df['entity'].value_counts()
print(entity_count)
```

```
entity
TomClancysRainbowSix                2400
MaddenNFL                           2400
Microsoft                           2400
LeagueOfLegends                     2394
CallOfDuty                          2394
Verizon                             2382
CallOfDutyBlackopsColdWar           2376
ApexLegends                         2376
Facebook                            2370
WorldOfCraft                        2364
Dota2                               2364
NBA2K                               2352
TomClancysGhostRecon                2346
Battlefield                         2346
FIFA                                2340
Xbox(Xseries)                       2334
Overwatch                           2334
johnson&johnson                     2328
Amazon                              2316
PlayStation5(PS5)                   2310
HomeDepot                           2310
Cyberpunk2077                       2304
CS-GO                               2304
GrandTheftAuto(GTA)                 2304
Hearthstone                         2298
Nvidia                              2298
Google                              2298
Borderlands                         2285
PlayerUnknownsBattlegrounds(PUBG)   2274
Fortnite                            2274
RedDeadRedemption(RDR)              2262
AssassinsCreed                      2244
Name: count, dtype: int64
```

```
df.info
```

```
pandas.core.frame.DataFrame.info
def info(verbose: bool | None=None, buf: WriteBuffer[str] | None=None, max_cols:
int | None=None, memory_usage: bool | str | None=None, show_counts: bool |
None=None) -> None
```

```
Print a concise summary of a DataFrame.

This method prints information about a DataFrame including
the index dtype and columns, non-null values and memory usage.

Parameters
```

Checking for duplicates

```
duplicates = df.duplicated()
duplicated_rows = df[duplicates]
duplicated_rows.count()
```

```
ID           2700
entity       2700
sentiment    2700
comment      2340
dtype: int64
```

checking for Missing values

```
df.isnull().sum()
```

```
ID             0
entity         0
sentiment      0
comment      686
dtype: int64
```

```
# Dropping missing value [ Using Dropna]
df = df.dropna()
```

```
df.isnull().sum()
```

```
ID           0
entity       0
sentiment    0
comment      0
dtype: int64
```

```
# Number of Unique Values
df.nunique()
```

```
ID           12447
entity          32
sentiment        4
comment      69490
dtype: int64
```

Displaying Sample

```
for i in range(5):
    print(f"{i+1}: {df['comment'][i]}   ->   {df['sentiment'][i]}")
```

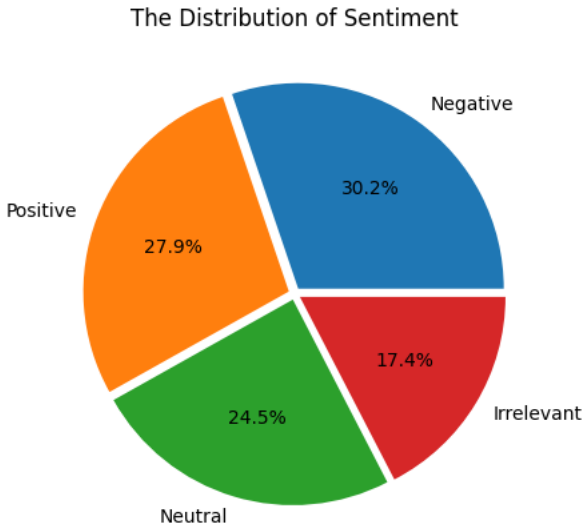
```
1: I am coming to the borders and I will kill you all,   ->   Positive
2: im getting on borderlands and i will kill you all,   ->   Positive
3: im coming on borderlands and i will murder you all,   ->   Positive
4: im getting on borderlands 2 and i will murder you me all,   ->   Positive
5: im getting into borderlands and i can murder you all,   ->   Positive
```

Sentiment Analysis

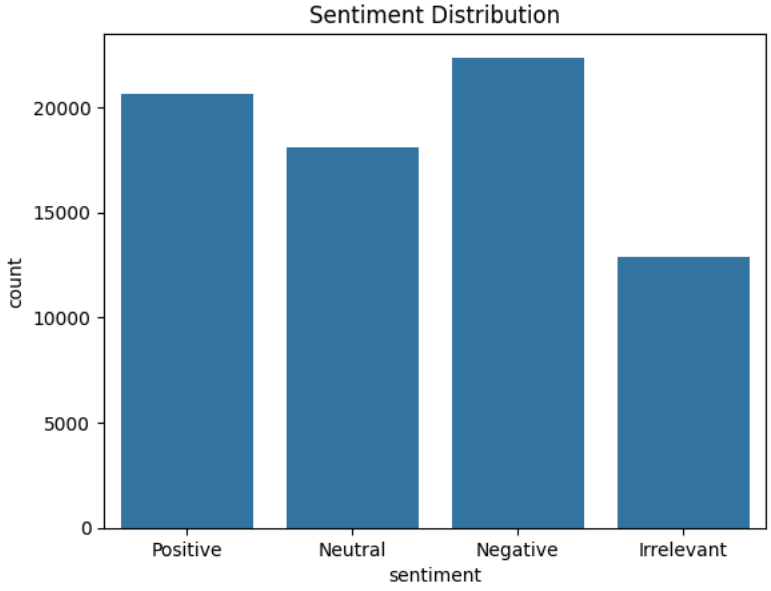
```
df['sentiment'].value_counts()
```

```
sentiment
Negative      22358
Positive      20654
Neutral       18108
Irrelevant    12875
Name: count, dtype: int64
```

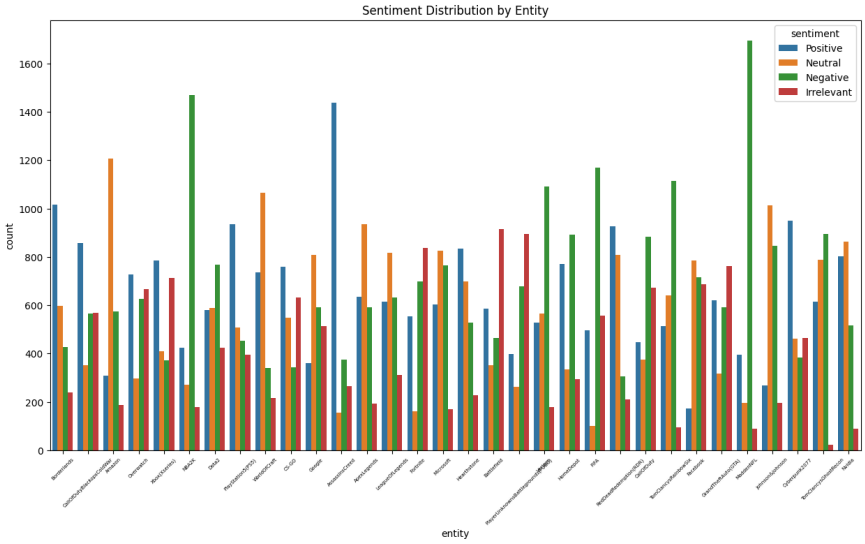
```
plt.figure(figsize=(10,5))
plt.pie(x=df['sentiment'].value_counts().values,
        labels=df['sentiment'].value_counts().index,
        autopct='%.1f%%', explode=[0.03, 0.03,0.03,0.03])
plt.title('The Distribution of Sentiment')
plt.show()
```



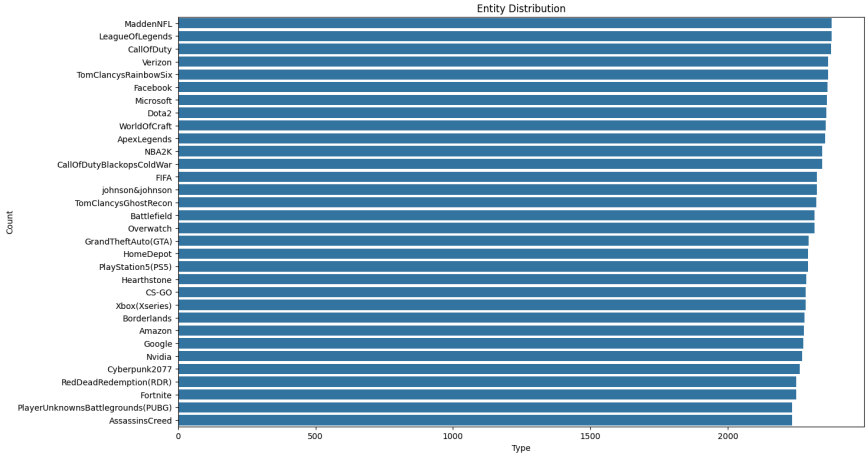

```
sns.countplot(x='sentiment', data=df)
plt.title('Sentiment Distribution')
plt.show()
```



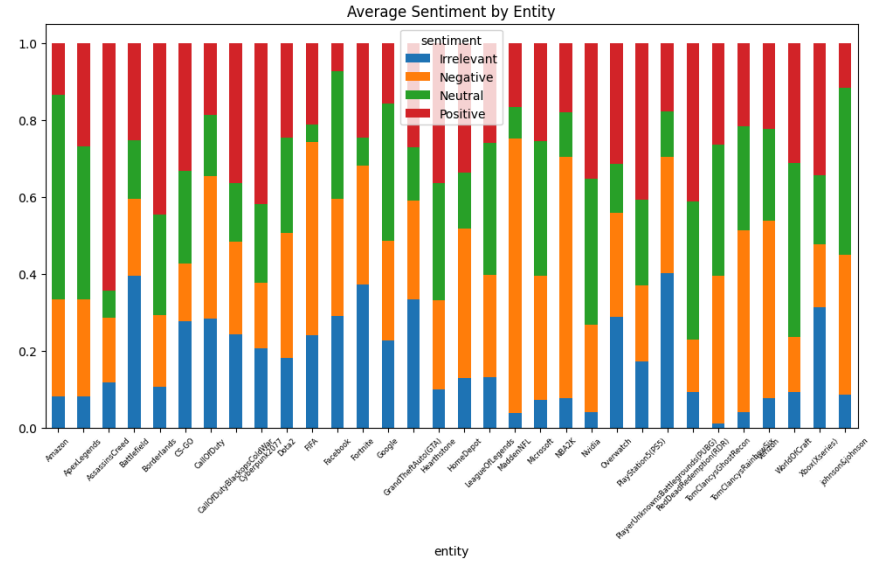

```
plt.figure(figsize=(15, 8))
sns.countplot(x='entity', hue='sentiment', data=df)
plt.title('Sentiment Distribution by Entity')
plt.xticks(rotation=45, fontsize=5)
plt.show()
```




```
plt.figure(figsize=(15,9))
sns.barplot(x=df['entity'].value_counts().values,y=df['entity'].value_counts().index)
plt.xlabel('Type')
plt.ylabel('Count')
plt.title('Entity Distribution')
plt.show()
```




```
average_sentiment_by_entity = df.groupby('entity')['sentiment'].value_counts(normalize=True).unstack()
average_sentiment_by_entity.plot(kind='bar', stacked=True, figsize=(12, 6))
plt.title('Average Sentiment by Entity')
plt.xticks(rotation=45, fontsize=6)
plt.show()
```



Word Cloud

```
import nltk
import re
stemmer = nltk.SnowballStemmer("english")
nltk.download('stopwords')
from nltk.corpus import stopwords
import string
stopword=set(stopwords.words('english'))
```

```
[nltk_data] Downloading package stopwords to /root/nltk_data...
[nltk_data]   Unzipping corpora/stopwords.zip.
```

```
def clean(text):
    text = str(text).lower()
    text = re.sub('\[.*?\]', '', text)
    text = re.sub('https?://\S+|www\.\S+', '', text)
    text = re.sub('<.*?>+', '', text)
    text = re.sub('[%s]' % re.escape(string.punctuation), '', text)
    text = re.sub('\n', '', text)
    text = re.sub('\w*\d\w*', '', text)
    text = [word for word in text.split(' ') if word not in stopword]
    text=" ".join(text)
    text = [stemmer.stem(word) for word in text.split(' ')]
    text=" ".join(text)
    return text
```

```
df["comment"] = df["comment"].apply(clean)
```

```
<ipython-input-31-213ab282b395>:1: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus
  df["comment"] = df["comment"].apply(clean)
```

```
df.head()
```

	ID	entity	sentiment	comment
0	2401	Borderlands	Positive	come border kill
1	2401	Borderlands	Positive	im get borderland kill
2	2401	Borderlands	Positive	im come borderland murder
3	2401	Borderlands	Positive	im get borderland murder
4	2401	Borderlands	Positive	im get borderland murder

!pip install wordcloud

```
from wordcloud import WordCloud,STOPWORDS
wc = WordCloud(width=800,height=500,min_font_size=10,background_color='white')
```

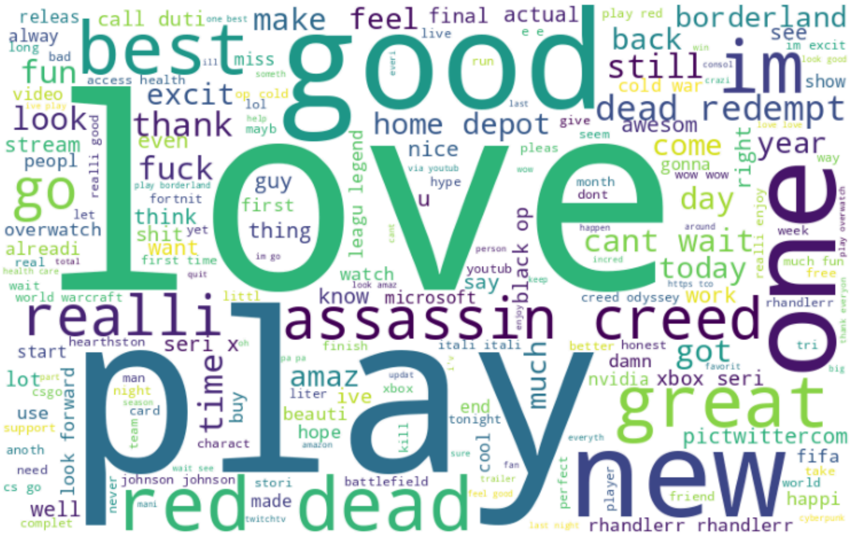
Positive Sentiment Word Cloud

```
from PIL import Image, ImageDraw, ImageFont
```

```
positive_data = df[df['sentiment'] == 'Positive']['comment'].str.cat(sep=" ")

# Exclude the word "game" from the text data
positive_data = positive_data.replace("game", "")
if positive_data:
    wc = WordCloud(width=800, height=500, background_color='white').generate(positive_data)
    plt.figure(figsize=(12, 6))
    plt.title('Positive Sentiment Word Cloud')
    plt.imshow(wc)
    plt.axis("off")
    plt.show()
else:
    print("No data available for positive sentiment.")
```

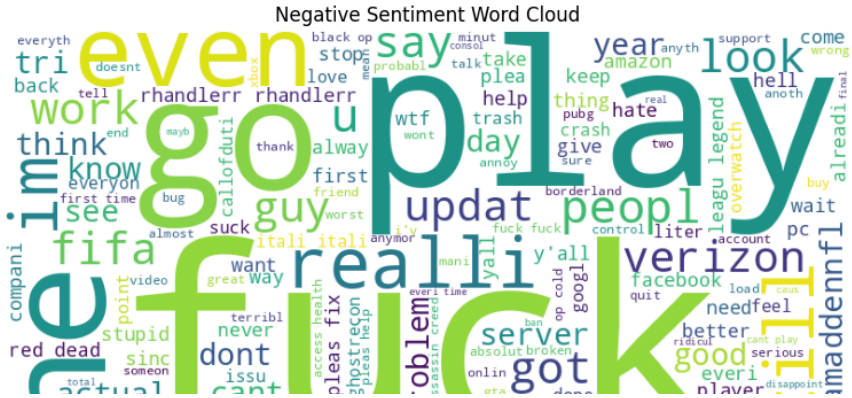
Positive Sentiment Word Cloud



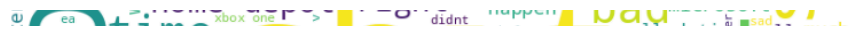
Negative Sentiment Word Cloud

```
negative_data = df[df['sentiment'] == 'Negative']['comment'].str.cat(sep=" ")

# Exclude the word "game" from the text data
negative_data = negative_data.replace("game", "")
if negative_data.strip():
    wc = WordCloud(width=800, height=500, background_color='white').generate(negative_data)
    plt.figure(figsize=(12, 6))
    plt.title('Negative Sentiment Word Cloud')
    plt.imshow(wc)
    plt.axis("off")
    plt.show()
else:
    print("No data available for negative sentiment.")
```



Neutral Sentiment Word Cloud



```
neutral_data = df[df['sentiment'] == 'Neutral']['comment'].str.cat(sep=" ")

# Exclude the word "game" from the text data
neutral_data = neutral_data.replace("game", "")
if neutral_data.strip():
    wc = WordCloud(width=800, height=500, background_color='white').generate(neutral_data)
    plt.figure(figsize=(12, 6))
    plt.title('Negative Sentiment Word Cloud')
```