Task 04

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

Importing Data and Arranging them properly

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

```
data=pd.read_csv("/content/twitter_training.csv")
data.head()
```

	2401	Borderlands	Positive	im getting on borderlands and i will murder you all ,
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...

Next steps: Generate code with data View recommended plots New interactive sheet

```
cols=['ID', 'platform', 'Sentiment', 'review']
data=pd.read_csv("/content/twitter_training.csv",names=cols)
data.head()
```

	ID	platform	Sentiment	review
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 ...

Next steps: Generate code with data View recommended plots New interactive sheet

```
data.shape
```

```
(46295, 4)
```

```
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 46295 entries, 0 to 46294
Data columns (total 4 columns):
 #   Column     Non-Null Count  Dtype 
---  ------     --------------  ----- 
 0   ID         46295 non-null  int64 
 1   platform   46295 non-null  object
 2   Sentiment  46295 non-null  object
 3   review     45850 non-null  object
dtypes: int64(1), object(3)
memory usage: 1.4+ MB
```

```
data.describe()
```

	ID
count	46295.000000
mean	6093.146042
std	4090.867555
min	1.000000
25%	1986.500000
50%	5590.000000
75%	9558.000000
max	13200.000000

```
data.describe(include=object)
```

	platform	Sentiment	review
count	46295	46295	45850
unique	20	4	42998
top	Microsoft	Positive	It is not the first time that the EU Commissio...
freq	2400	13710	109

```
#missing data
missing_data = data.isna().sum()
pd.set_option('display.max_rows', None)
print(missing_data)
```

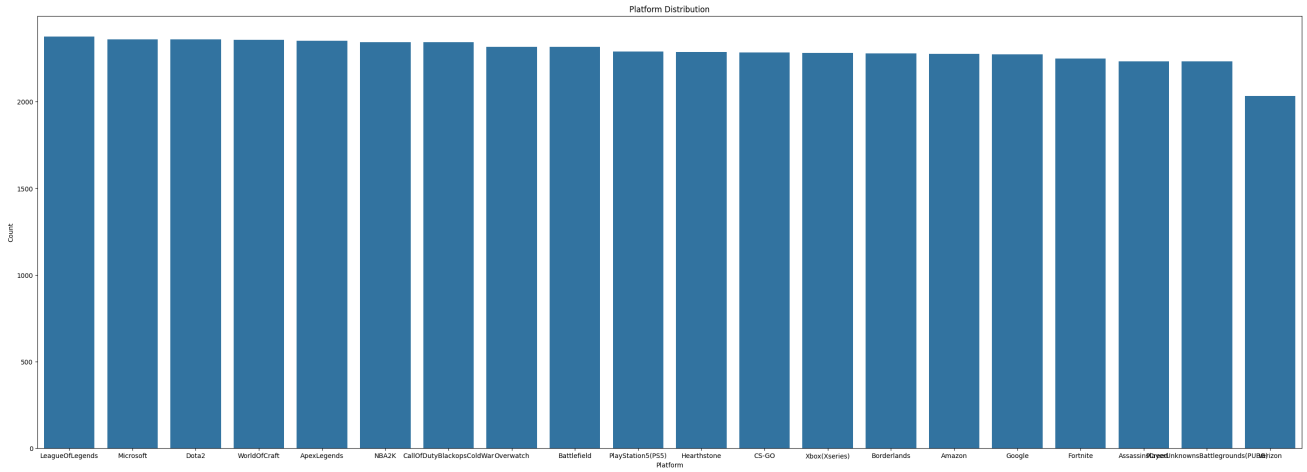
```
ID             0
platform       0
Sentiment      0
review       445
dtype: int64
```

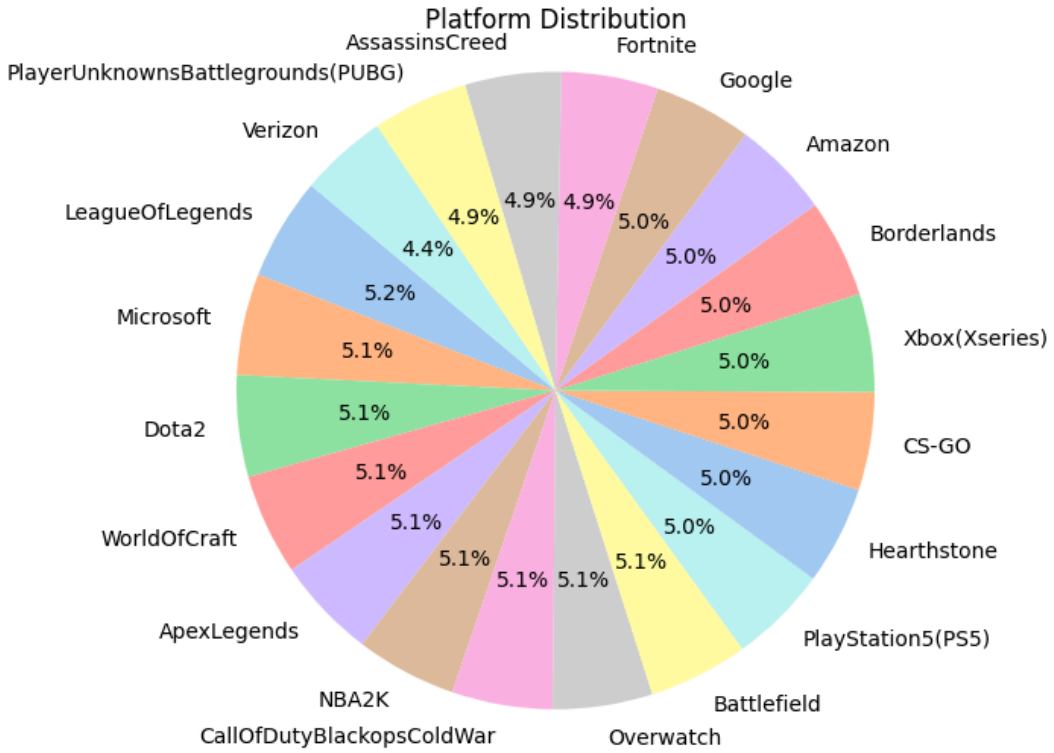
```
#removing null data
data=data.dropna()
#missing data
missing_data = data.isna().sum()
pd.set_option('display.max_rows', None)
print(missing_data)
```

```
ID           0
platform     0
Sentiment    0
review       0
dtype: int64
```

EDA

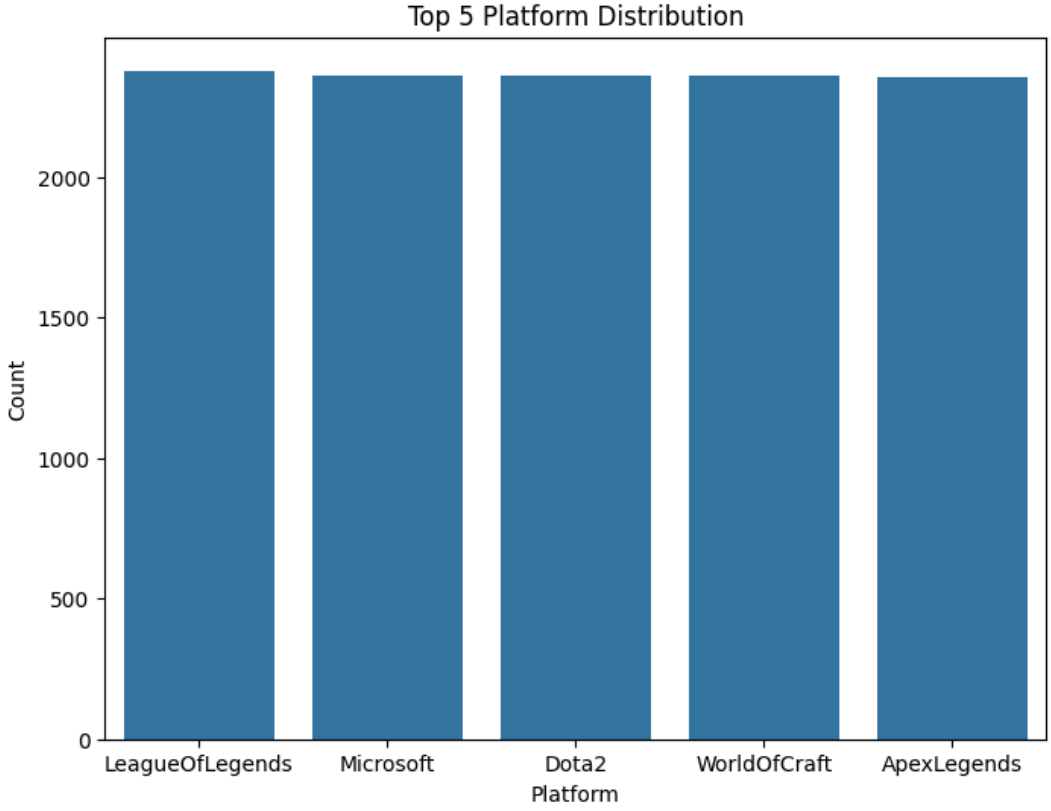
```
# Count of platform classes
platform_counts = data['platform'].value_counts()

# Bar Plot
plt.figure(figsize=(35, 12))
sns.barplot(x=platform_counts.index, y=platform_counts.values)
plt.title('Platform Distribution')
plt.xlabel('Platform')
plt.ylabel('Count')
plt.show()

# Pie Chart
plt.figure(figsize=(8, 6))
plt.pie(platform_counts, labels=platform_counts.index, autopct='%1.1f%%', startangle=140, colors=sns.color_palette("pa
plt.title('Platform Distribution')
plt.axis('equal')  # Equal aspect ratio ensures the pie is drawn as a circle.
plt.show()

print(platform_counts)
```





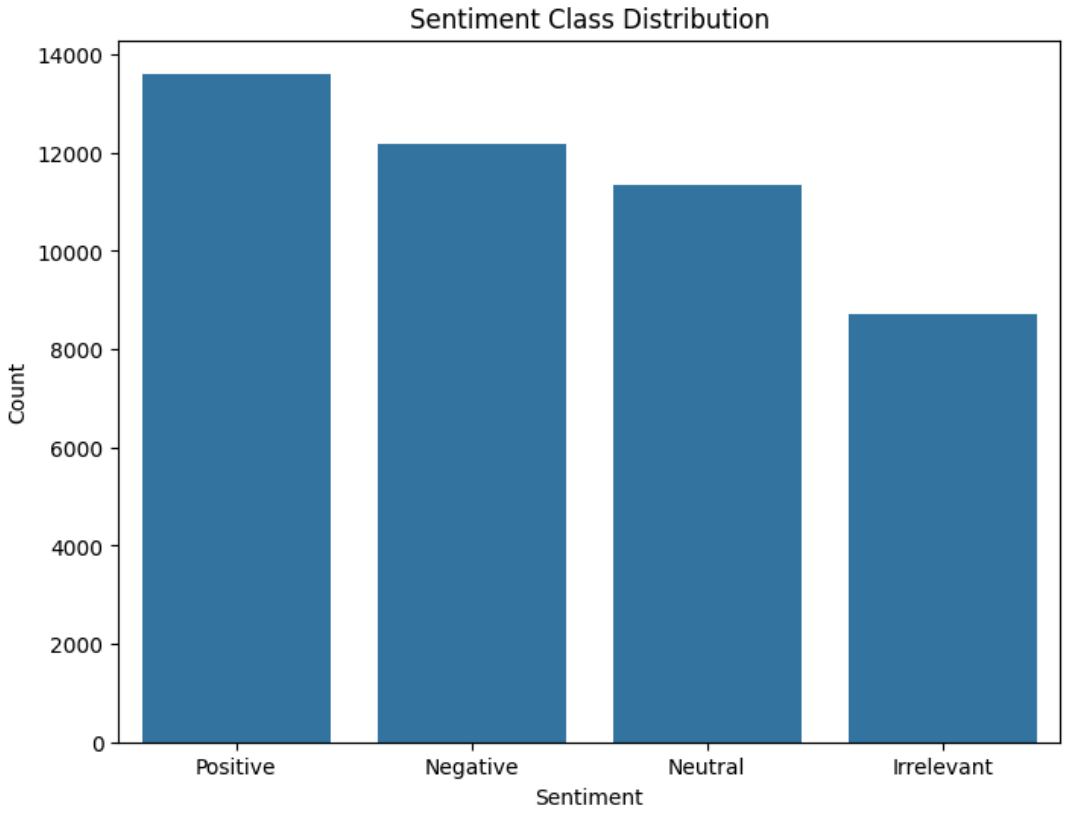

```
platform
LeagueOfLegends                  2377
Microsoft                        2361
Dota2                            2359
WorldOfCraft                     2357
ApexLegends                      2353
NBA2K                            2343
CallOfDutyBlackopsColdWar        2343
Overwatch                        2316
Battlefield                      2316
PlayStation5(PS5)                2291
Hearthstone                      2286
CS-GO                            2284
Xbox(Xseries)                    2283
Borderlands                      2280
Amazon                           2276
Google                           2274
Fortnite                         2249
AssassinsCreed                   2234
PlayerUnknownsBattlegrounds(PUBG)  2234
Verizon                          2034
Name: count, dtype: int64
```

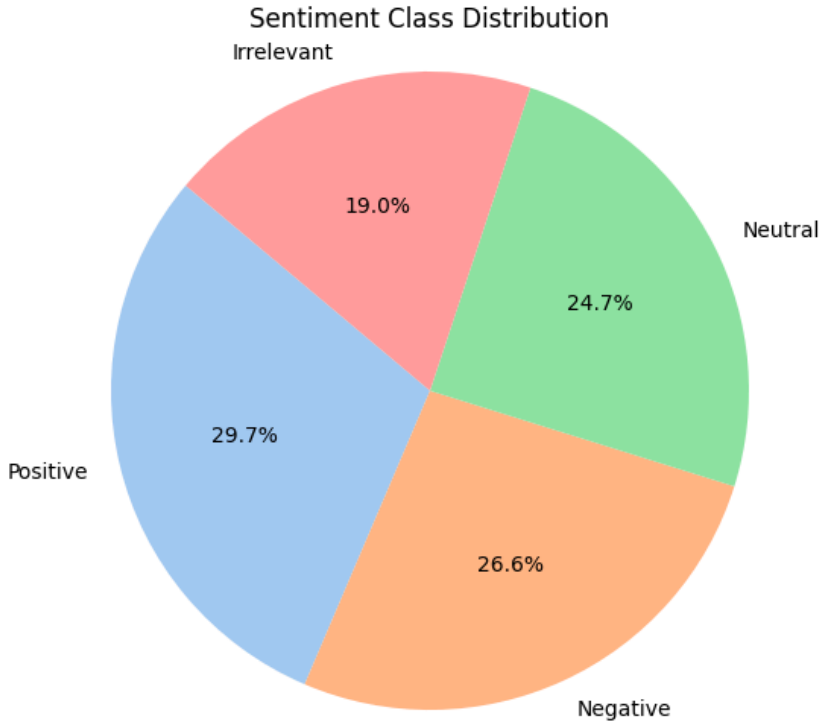
```
# Count of platform classes
platform_counts_top5 = data['platform'].value_counts().nlargest(5)

# Bar Plot
plt.figure(figsize=(8, 6))
sns.barplot(x=platform_counts_top5.index, y=platform_counts_top5.values)
plt.title('Top 5 Platform Distribution')
plt.xlabel('Platform')
plt.ylabel('Count')
plt.show()

print(platform_counts_top5)
```




```
platform
LeagueOfLegends    2377
Microsoft          2361
Dota2              2359
WorldOfCraft       2357
ApexLegends        2353
Name: count, dtype: int64
```

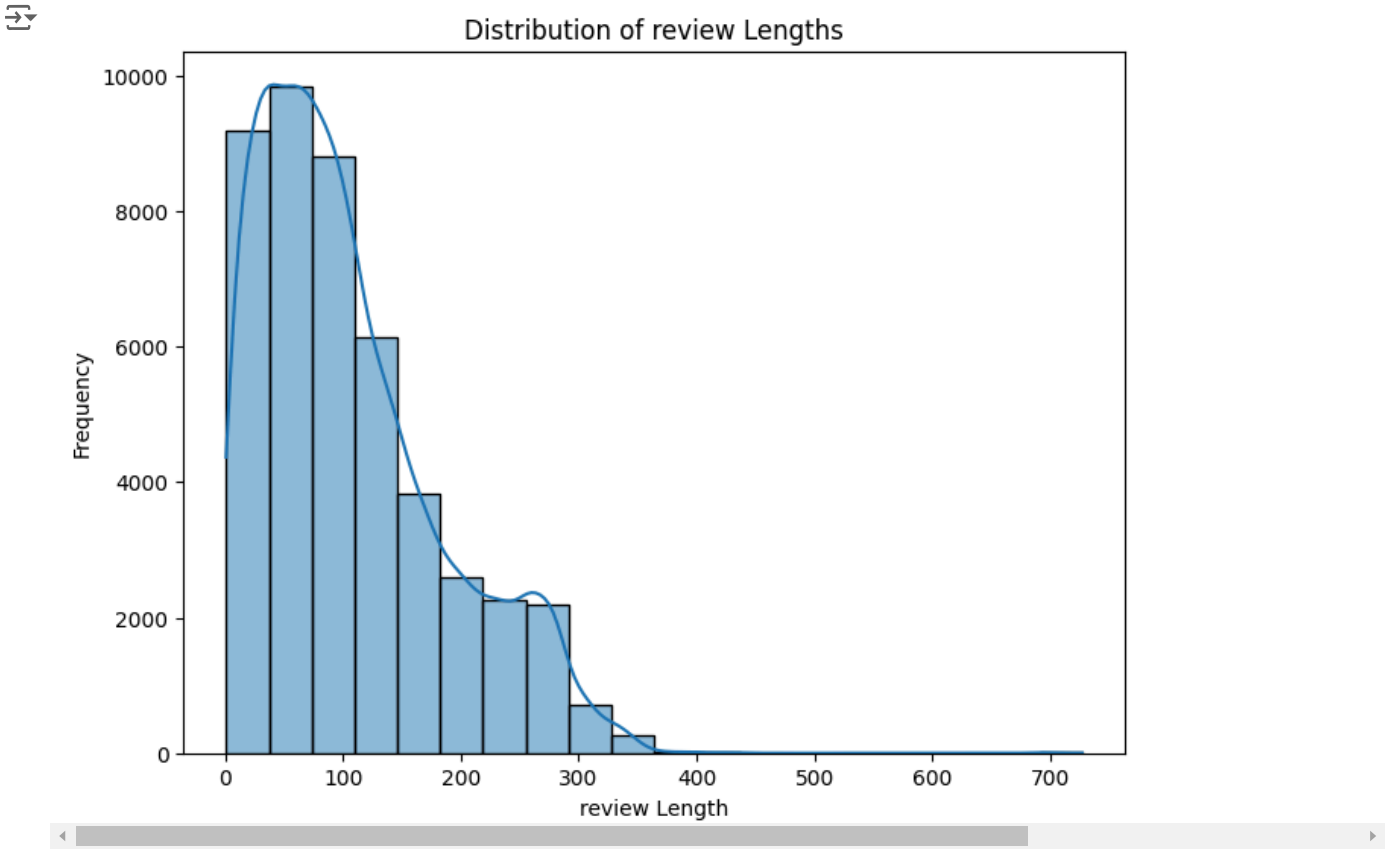
```
import matplotlib.pyplot as plt
import seaborn as sns

# Count of sentiment classes
sentiment_counts = data['Sentiment'].value_counts()

# Bar Plot
plt.figure(figsize=(8, 6))
sns.barplot(x=sentiment_counts.index, y=sentiment_counts.values)
plt.title('Sentiment Class Distribution')
plt.xlabel('Sentiment')
plt.ylabel('Count')
plt.show()

# Pie Chart
plt.figure(figsize=(8, 6))
plt.pie(sentiment_counts, labels=sentiment_counts.index, autopct='%1.1f%%', startangle=140, colors=sns.color_palette("
plt.title('Sentiment Class Distribution')
plt.axis('equal')  # Equal aspect ratio ensures the pie is drawn as a circle.
plt.show()

print(sentiment_counts)
```



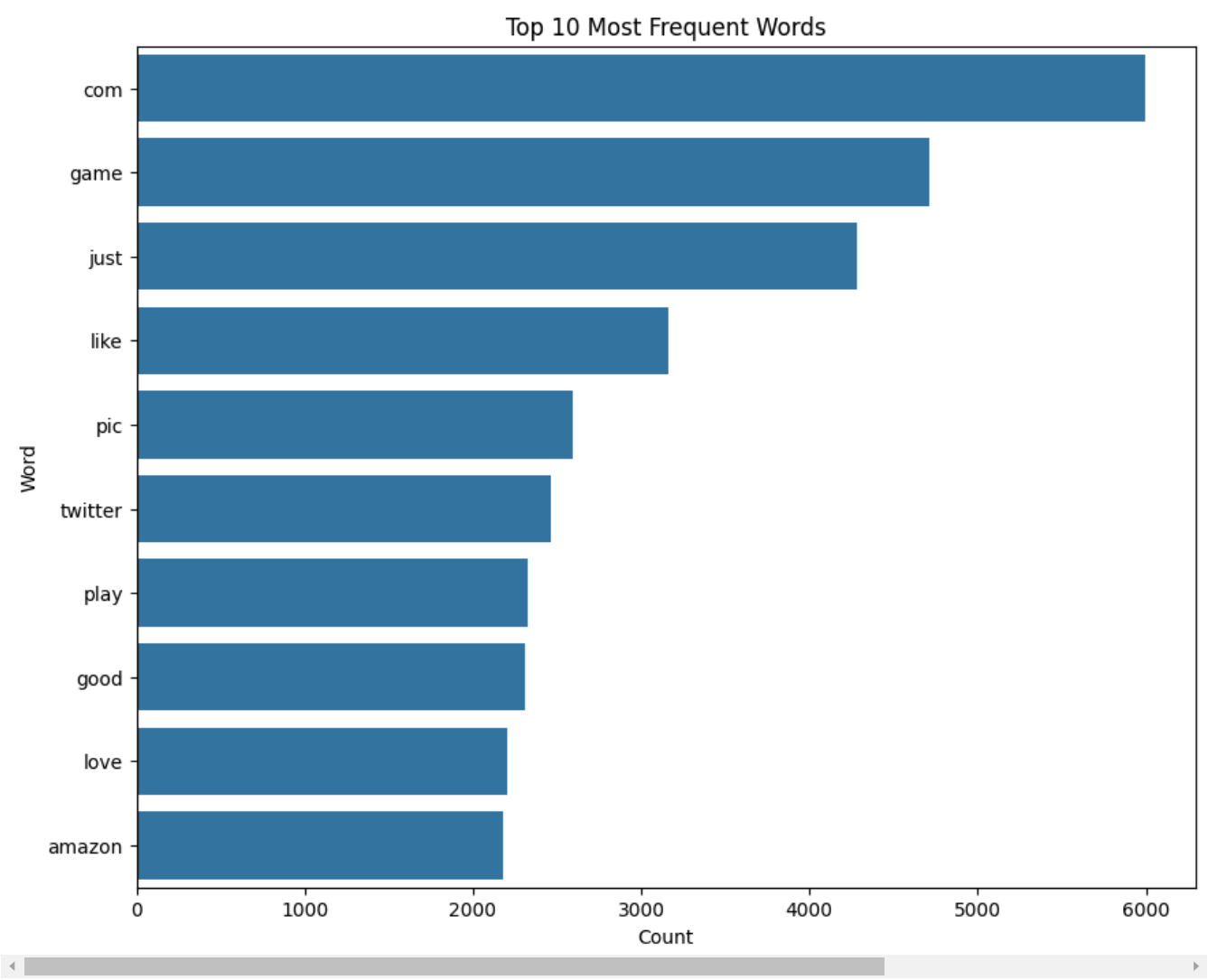



```
Sentiment
Positive      13610
Negative      12189
Neutral       11343
Irrelevant     8708
Name: count, dtype: int64
```

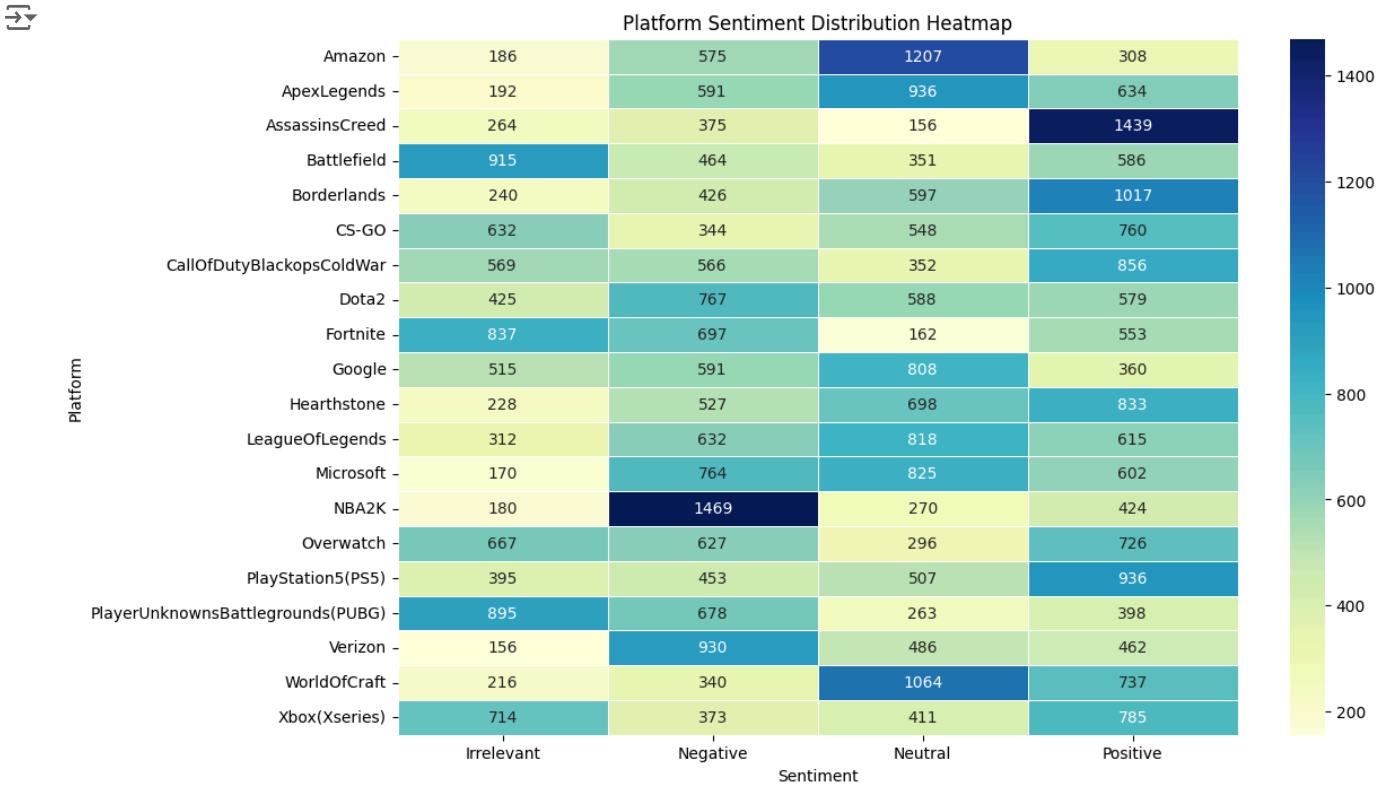
```
# Length of text
data['review_Length'] = data['review'].apply(len)

# Plot
plt.figure(figsize=(8, 6))
sns.histplot(data['review_Length'], bins=20, kde=True)
plt.title('Distribution of review Lengths')
plt.xlabel('review Length')
plt.ylabel('Frequency')
plt.show()
```



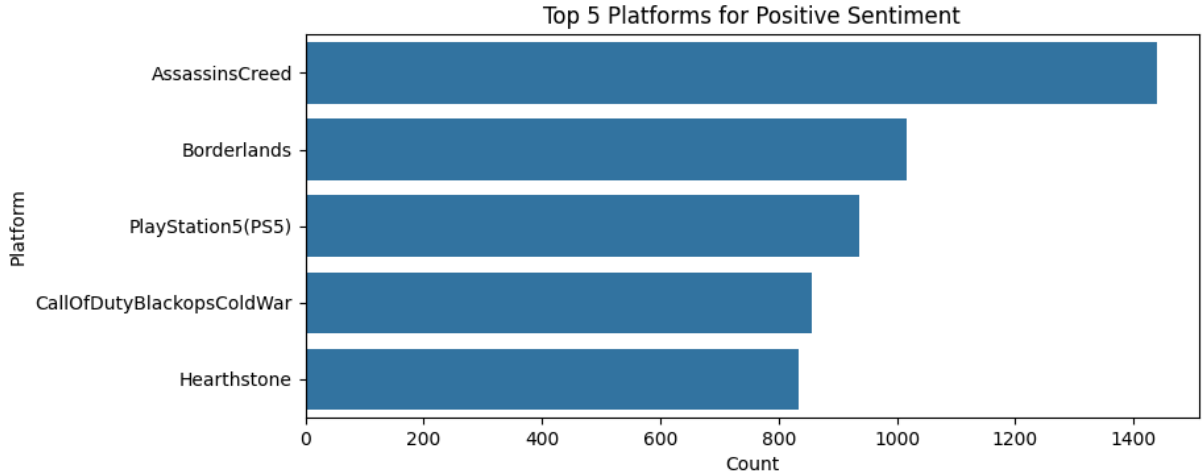

```
from sklearn.feature_extraction.text import CountVectorizer

# Vectorize text
vectorizer = CountVectorizer(stop_words='english')
X = vectorizer.fit_transform(data['review'])
word_counts = X.toarray().sum(axis=0)
words = vectorizer.get_feature_names_out()

# Create DataFrame for word counts
word_freq_df = pd.DataFrame({'word': words, 'count': word_counts})
word_freq_df = word_freq_df.sort_values(by='count', ascending=False)

# Plot
plt.figure(figsize=(10, 8))
sns.barplot(x=word_freq_df.head(10)['count'], y=word_freq_df.head(10)['word'])
plt.title('Top 10 Most Frequent Words')
plt.xlabel('Count')
plt.ylabel('Word')
plt.show()
```

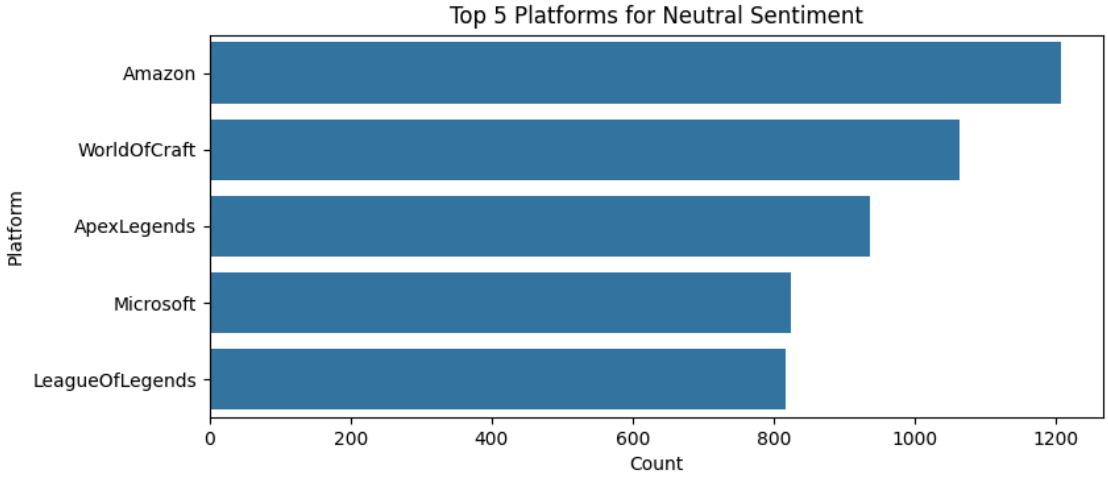


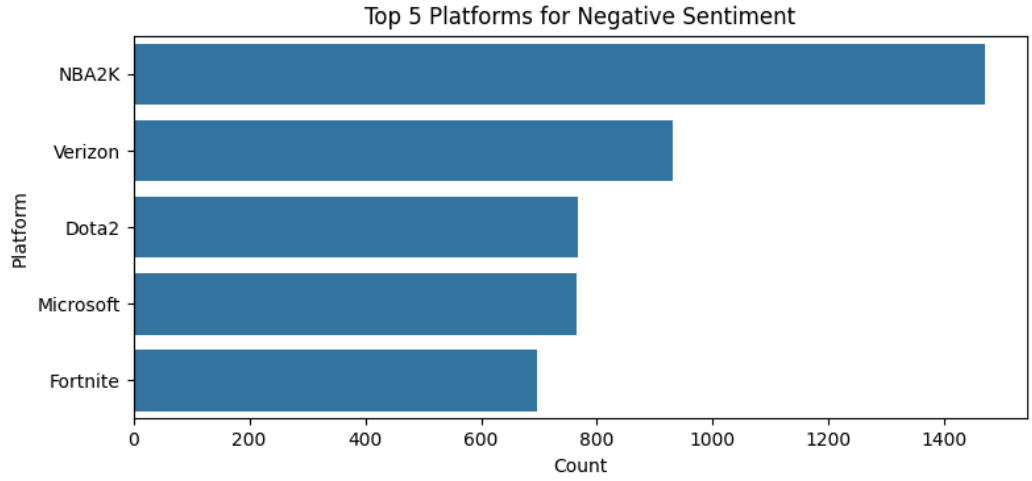

```
# Create the contingency table
platform_sentiment_table = pd.crosstab(data['platform'], data['Sentiment'])

# Plot the heatmap
plt.figure(figsize=(12, 8))
sns.heatmap(platform_sentiment_table, annot=True, fmt="d", cmap="YlGnBu", linewidths=.5)
plt.title('Platform Sentiment Distribution Heatmap')
plt.xlabel('Sentiment')
plt.ylabel('Platform')
plt.show()
```

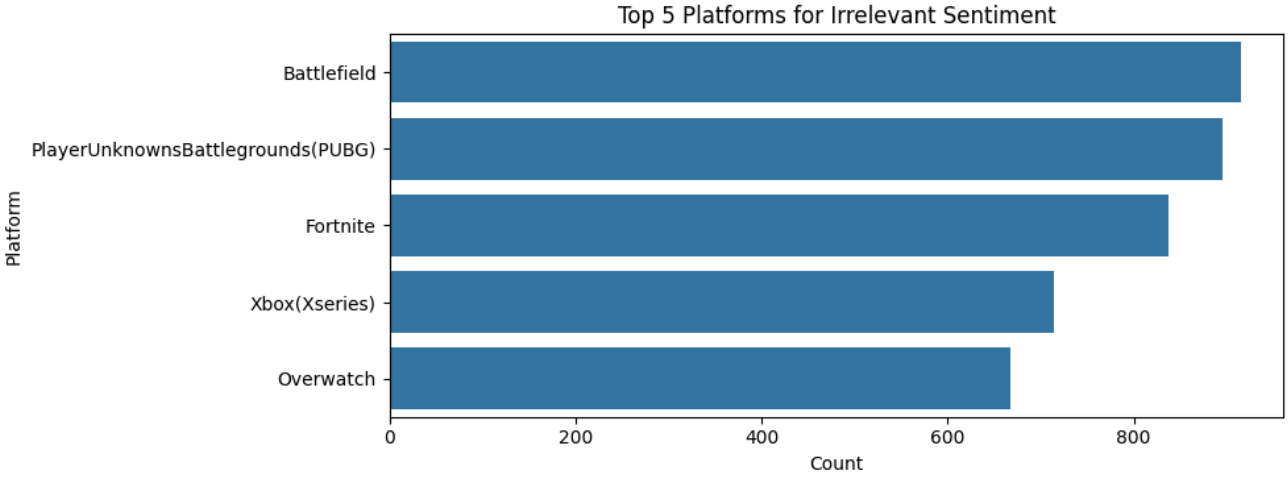



```
# Function to get top 5 platforms for each sentiment
def get_top_platforms(df, sentiment, top_n=5):
    filtered_df = df[df['Sentiment'] == sentiment]
    top_platforms = filtered_df['platform'].value_counts().nlargest(top_n)
    return top_platforms

# Sentiment categories
sentiments = data['Sentiment'].unique()

# Plotting
fig, axes = plt.subplots(nrows=len(sentiments), ncols=1, figsize=(10, 15))

for i, sentiment in enumerate(sentiments):
    top_platforms = get_top_platforms(data, sentiment)
    sns.barplot(x=top_platforms.values, y=top_platforms.index, ax=axes[i])
    axes[i].set_title(f'Top 5 Platforms for {sentiment} Sentiment')
    axes[i].set_xlabel('Count')
    axes[i].set_ylabel('Platform')

plt.tight_layout()
plt.show()
```


Top 5 Platforms for Positive Sentiment
AssassinsCreed
Borderlands
PlayStation5(PS5)
CallOfDutyBlackopsColdWar
Hearthstone
Platform
0
200
400
600
800
1000
1200
1400
Count


Top 5 Platforms for Neutral Sentiment
Amazon
WorldOfCraft
ApexLegends
Microsoft
LeagueOfLegends
Platform
0
200
400
600
800
1000
1200
Count


Top 5 Platforms for Negative Sentiment
NBA2K
Verizon
Dota2
Microsoft
Fortnite
Platform
0
200
400
600
800
1000
1200
1400
Count


Top 5 Platforms for Irrelevant Sentiment
Battlefield
PlayerUnknownsBattlegrounds(PUBG)
Fortnite
Xbox(Xseries)
Overwatch
Platform
0
200
400
600
800
Count

```
pip install wordcloud
```

```
Requirement already satisfied: wordcloud in /usr/local/lib/python3.10/dist-packages (1.9.3)
Requirement already satisfied: numpy>=1.6.1 in /usr/local/lib/python3.10/dist-packages (from wordcloud) (1.26.4)
Requirement already satisfied: pillow in /usr/local/lib/python3.10/dist-packages (from wordcloud) (9.4.0)
Requirement already satisfied: matplotlib in /usr/local/lib/python3.10/dist-packages (from wordcloud) (3.7.1)
Requirement already satisfied: contourpy>=1.0.1 in /usr/local/lib/python3.10/dist-packages (from matplotlib->wordc
Requirement already satisfied: cycler>=0.10 in /usr/local/lib/python3.10/dist-packages (from matplotlib->wordcloud
Requirement already satisfied: fonttools>=4.22.0 in /usr/local/lib/python3.10/dist-packages (from matplotlib->word
Requirement already satisfied: kiwisolver>=1.0.1 in /usr/local/lib/python3.10/dist-packages (from matplotlib->word
Requirement already satisfied: packaging>=20.0 in /usr/local/lib/python3.10/dist-packages (from matplotlib->wordcl
Requirement already satisfied: pyparsing>=2.3.1 in /usr/local/lib/python3.10/dist-packages (from matplotlib->wordc
Requirement already satisfied: python-dateutil>=2.7 in /usr/local/lib/python3.10/dist-packages (from matplotlib->w
Requirement already satisfied: six>=1.5 in /usr/local/lib/python3.10/dist-packages (from python-dateutil>=2.7->mat
```

```
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from wordcloud import WordCloud  # Importing WordCloud
import nltk
from nltk.corpus import stopwords
import string

# Ensure you have the stopwords downloaded
nltk.download('stopwords')
# Ensure you have the stopwords downloaded
nltk.download('stopwords')

data['review_length'] = data['review'].apply(lambda x: len(x.split()))

plt.figure(figsize=(12, 6))
sns.boxplot(x='Sentiment', y='review_length', data=data)
plt.title('Review Text Length Distribution by Sentiment')
plt.xlabel('Sentiment')
plt.ylabel('Review Length (Number of Words)')
plt.show()

# Function to create and display a word cloud for each sentiment
def plot_word_cloud(data, sentiment):
    reviews = data[data['Sentiment'] == sentiment]['review'].values
    text = ' '.join(reviews)
     # Remove stopwords and punctuation
    stop_words = set(stopwords.words('english'))
    text = ' '.join([word.lower() for word in text.split() if word.lower() not in stop_words and word not in string.pu

    wordcloud = WordCloud(width=800, height=400, background_color='white').generate(text)

    plt.figure(figsize=(10, 5))
    plt.imshow(wordcloud, interpolation='bilinear')
    plt.title(f'Word Cloud for {sentiment} Sentiment')
    plt.axis('off')
    plt.show()

# Plot word clouds for each sentiment
sentiments = data['Sentiment'].unique()
for sentiment in sentiments:
    plot_word_cloud(data, sentiment)
```