```
from ast import increment_lineno
import pandas as pd
import numpy as np
from nltk.tokenize import sent_tokenize,word_tokenize
from sklearn.feature_extraction.text import CountVectorizer
from sklearn.model_selection import train_test_split
from sklearn.svm import SVC
from sklearn.datasets import fetch_20newsgroups
from nltk.corpus import stopwords
import string
from nltk import pos_tag
from nltk.stem import WordNetLemmatizer
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.naive_bayes import MultinomialNB
from sklearn.ensemble import RandomForestClassifier
from sklearn.svm import SVC
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn import preprocessing
import seaborn as sns
import matplotlib.pyplot as plt
%matplotlib inline
import nltk
nltk.download('stopwords')
data=pd.read_csv("/content/twitter_training.csv")
v_data=pd.read_csv("/content/twitter_validation.csv")
data




data.columns=['id','game','sentiment','text']
v_data.columns=['id','game','sentiment','text']
data


v_data


data.shape

data.columns

data.describe(include='all')
```

```
id_types=data['id'].value_counts()
id_types









plt.figure(figsize=(12,7))

sns.barplot(y=id_types.index,x=id_types.values)
plt.xlabel("type")
plt.ylabel("count")


plt.title('# of id vs Count')
plt.show()




game_types=data['game'].value_counts()
game_types





plt.figure(figsize=(14,10))
sns.barplot(x=game_types.values,y=game_types.index)
plt.title('# of games and their count')
plt.ylabel('type')
plt.xlabel('count')
plt.show()


sns.catplot(x='game',hue='sentiment',kind='count',height=10,aspect=3,data=
data)




sns.heatmap(data.isnull(),yticklabels=False,cbar=False,cmap="viridis")
```

```
total_null=data.isnull().sum().sort_values(ascending=False)
percent=((data.isnull().sum()/data.isnull().count())*100).sort_values(ascending=False)
print('total records=',data.shape[0])
missing_data=pd.concat([total_null,percent.round(2)],axis=1,keys=['total missing','In Percent'])
missing_data.head(10)
```

```
data.dropna(subset=['text'],inplace=True)
total_null=data.isnull().sum().sort_values(ascending=False)
percent=((data.isnull().sum()/data.isnull().count())*100).sort_values(ascending=False)
print('total records=',data.shape[0])
missing_data=pd.concat([total_null,percent.round(2)],axis=1,keys=['total missing','In Percent'])
missing_data.head(10)
```

```
train0=data[data['sentiment']=="Negative"]
train1=data[data['sentiment']=="Positive"]
train2=data[data['sentiment']=="Irrelevant"]
train3=data[data['sentiment']=="Neutral"]
```

```
train0.shape,train1.shape,train2.shape,train3.shape
```

```
train0=train0[:int(train0.shape[0]/12)]
train1=train1[:int(train1.shape[0]/12)]
train2=train2[:int(train2.shape[0]/12)]
train3=train3[:int(train3.shape[0]/12)]

train0.shape,train1.shape,train2.shape,train3.shape
```

```
data=pd.concat([train0,train1,train2,train3],axis=0)
data
```

```
id_types=data['id'].value_counts()
id_types
```

```
plt.figure(figsize=(12,7))
sns.barplot(x=id_types.values,y=id_types.index)
plt.xlabel("type")
plt.ylabel("count")
plt.title("#of id vs count")
plt.show()
```

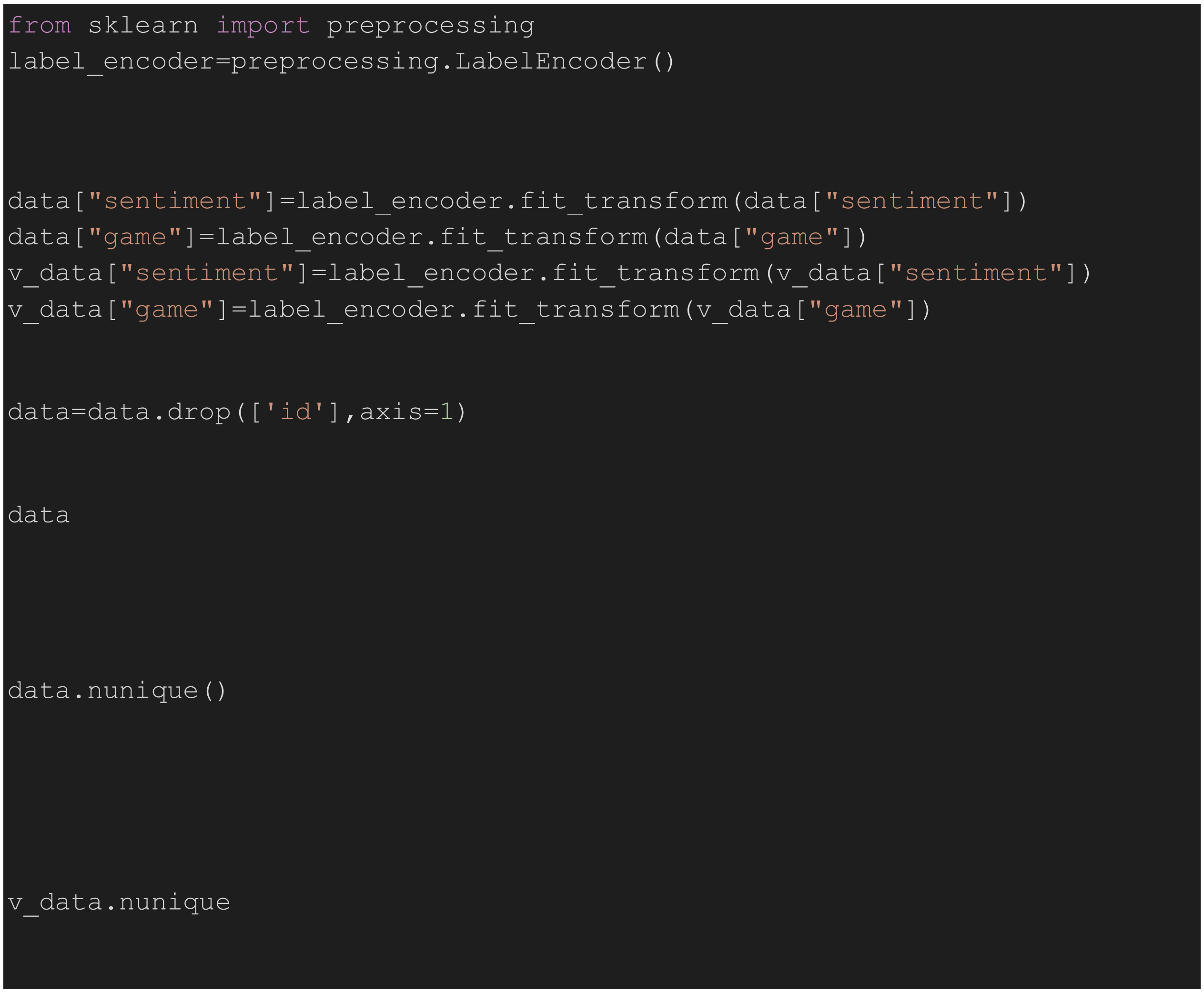
```
game_types=data['game'].value_counts()
game_types
```

```
plt.figure(figsize=(12,7))
sns.barplot(x=game_types.values,y=game_types.index)
plt.xlabel("type")
plt.ylabel("count")
plt.title('# of tv shows vs movie')
plt.show()
```

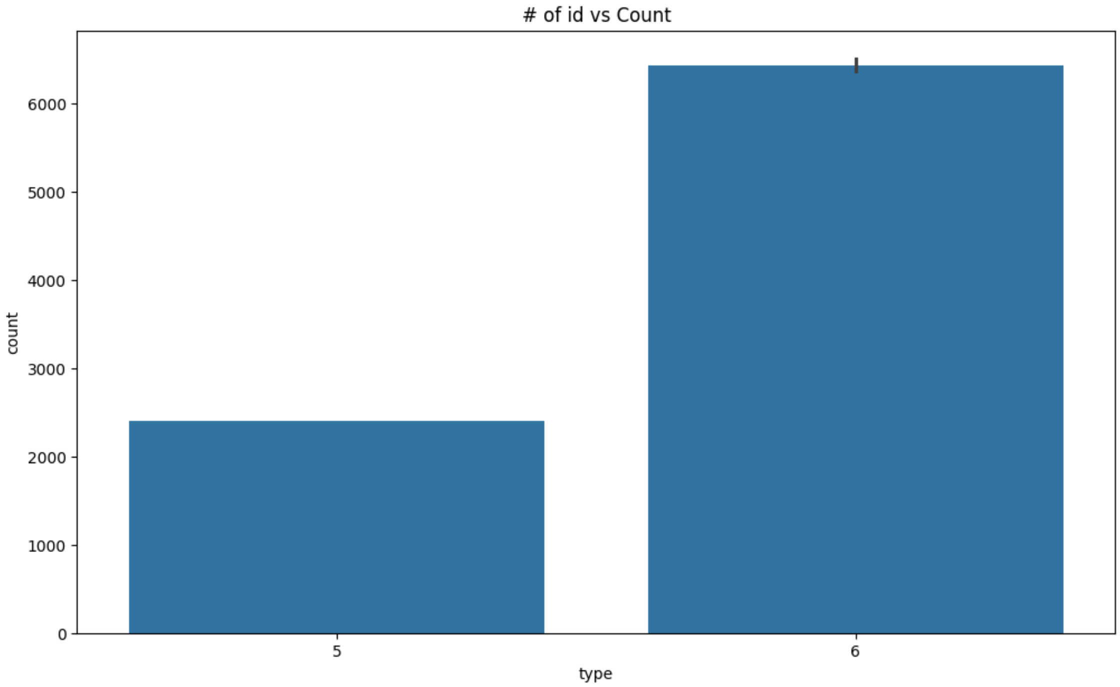
```
sentiment_types=data["sentiment"].value_counts()
sentiment_types
```

```
plt.figure(figsize=(12,7))
plt.pie(x=sentiment_types.values,labels=sentiment_types.index,autopct='%.1f%%',explode=[0.1,0.1,0,0])
plt.title("the difference in the type of contents")
plt.show()
```

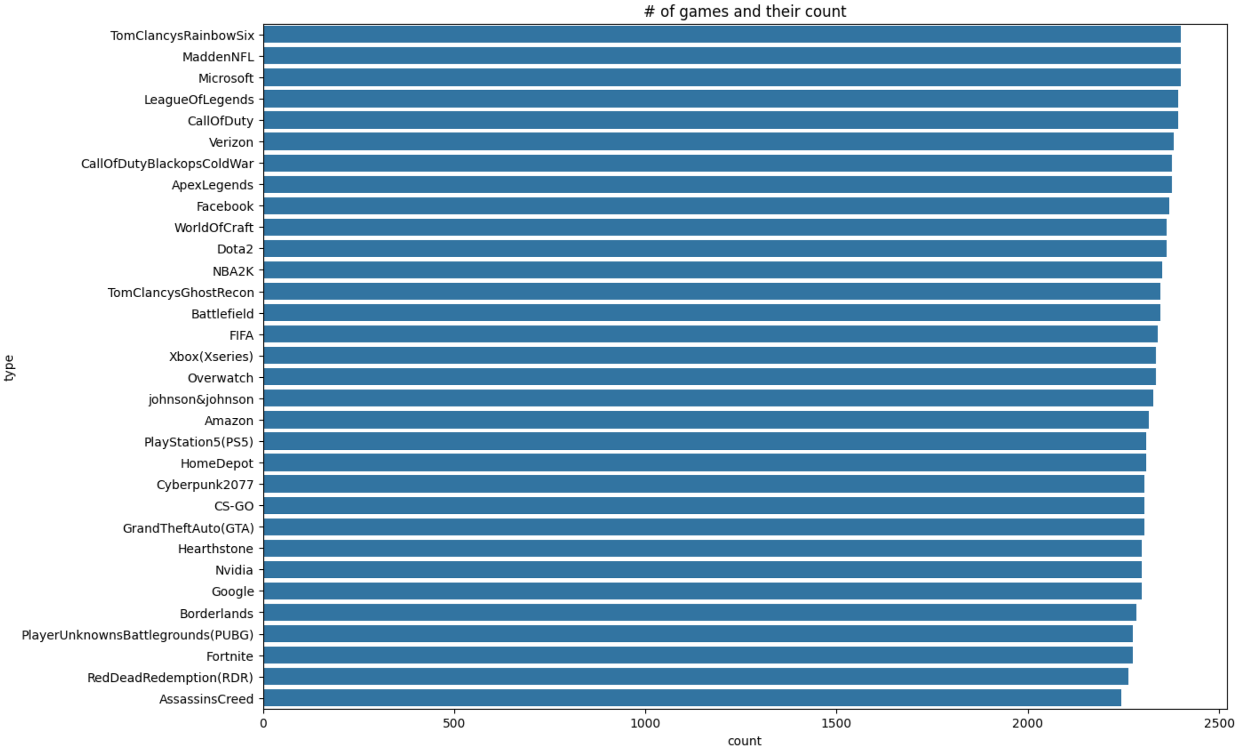
```
sns.catplot(x="game",hue="sentiment",kind="count",height=7,aspect=2,data=data)
```

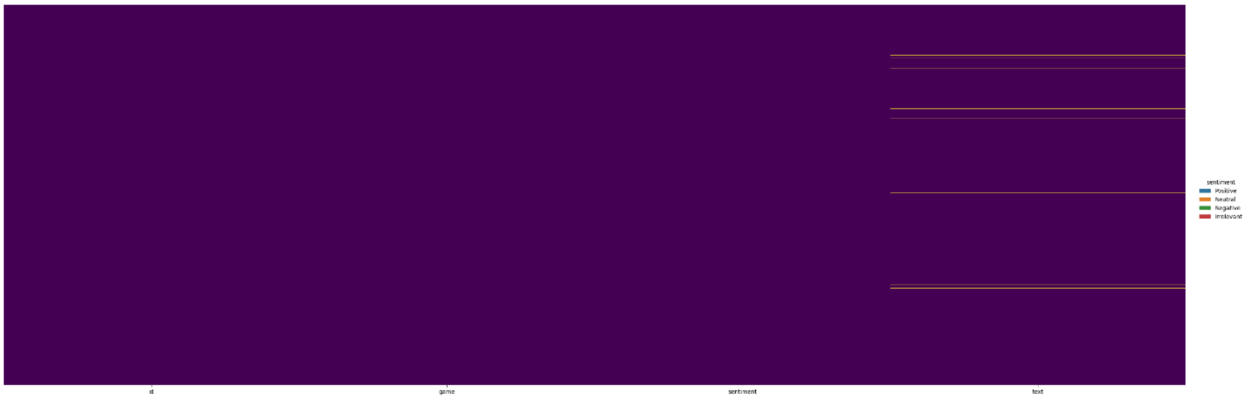


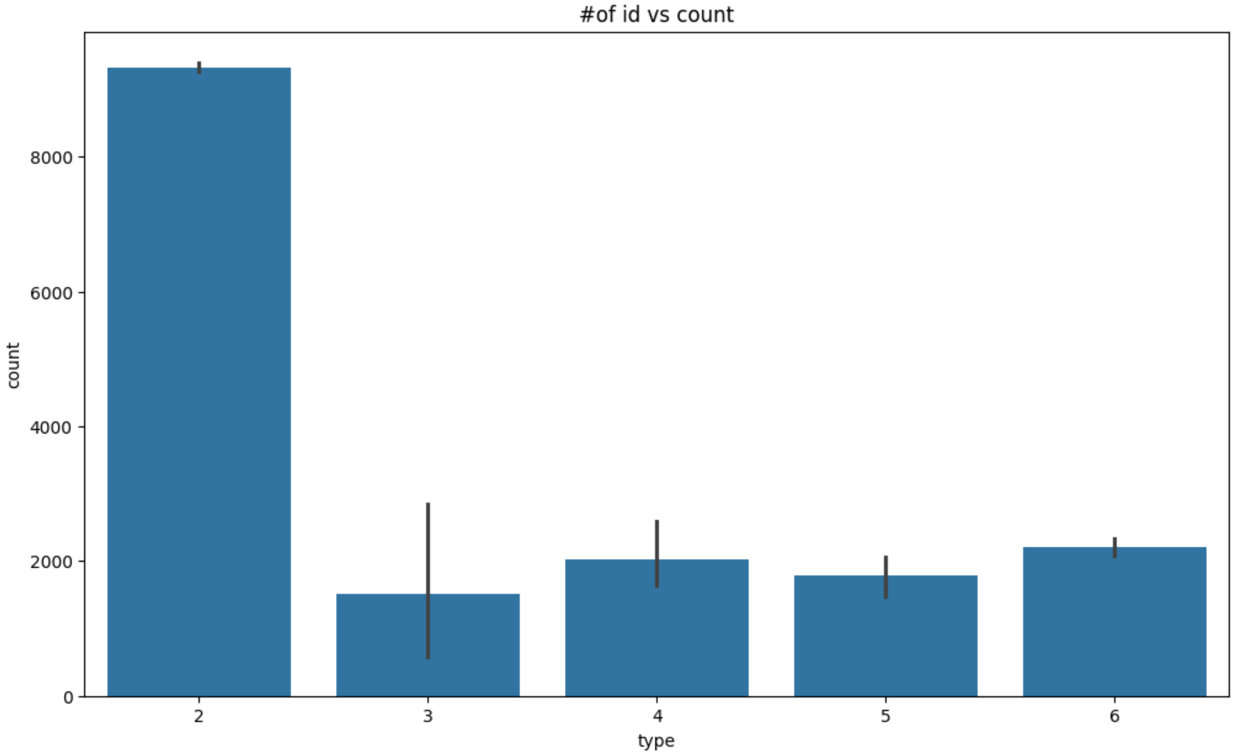
```
from sklearn import preprocessing
label_encoder=preprocessing.LabelEncoder()



data["sentiment"]=label_encoder.fit_transform(data["sentiment"])
data["game"]=label_encoder.fit_transform(data["game"])
v_data["sentiment"]=label_encoder.fit_transform(v_data["sentiment"])
v_data["game"]=label_encoder.fit_transform(v_data["game"])


data=data.drop(['id'],axis=1)


data




data.nunique()





v_data.nunique
```

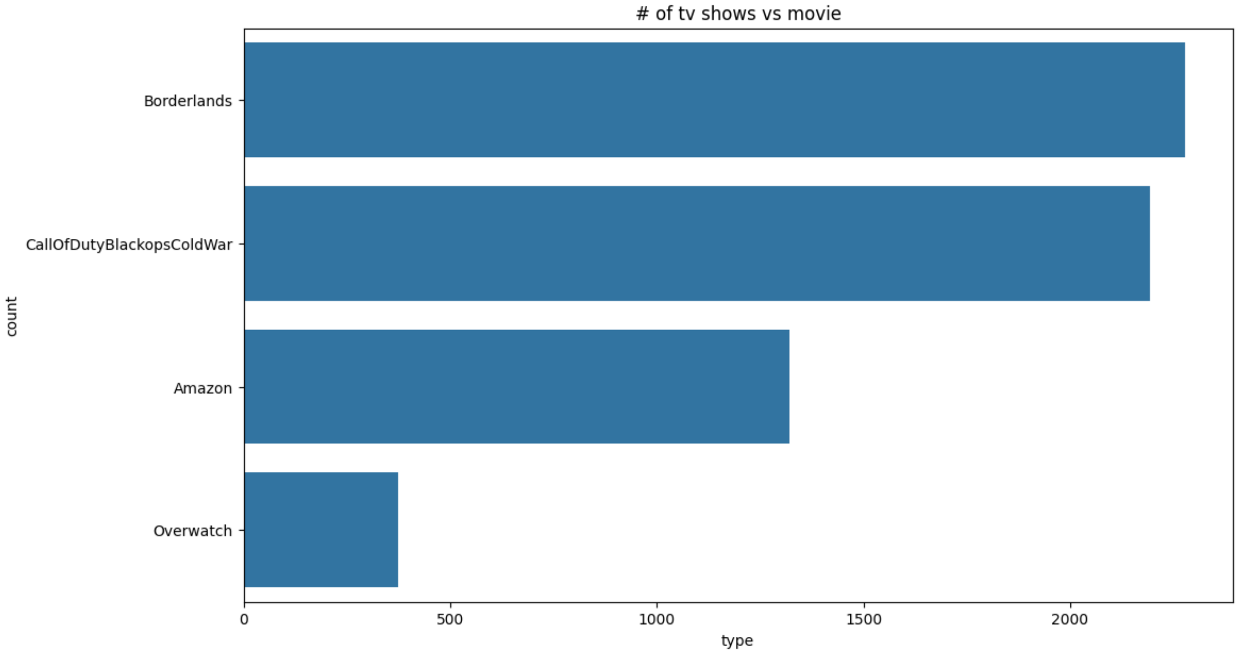


# of id vs Count
count
6000
5000
4000
3000
2000
1000
0
5
6
type

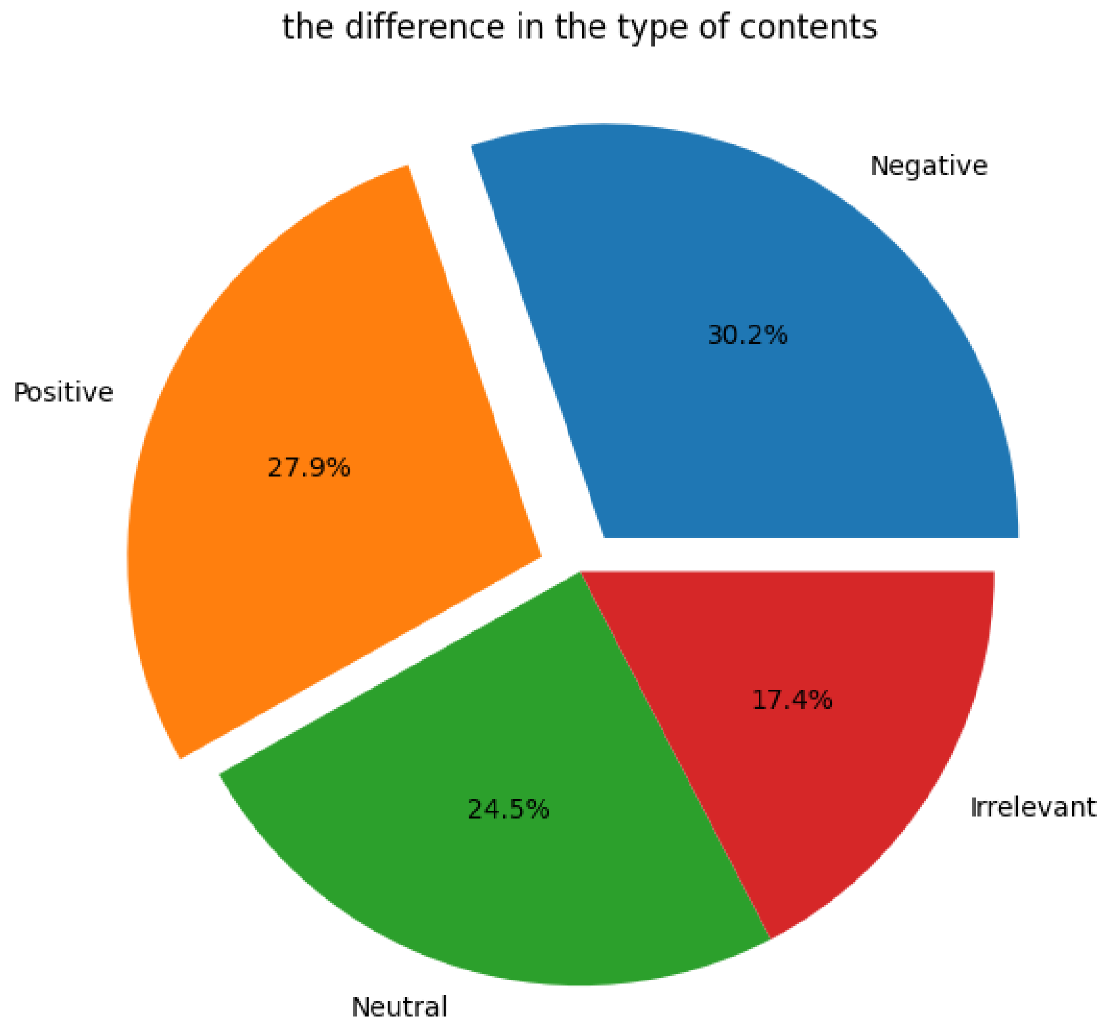

# of games and their count
TomClancysRainbowSix
MaddenNFL
Microsoft
LeagueOfLegends
CallOfDuty
Verizon
CallOfDutyBlackopsColdWar
ApexLegends
Facebook
WorldOfCraft
Dota2
NBA2K
TomClancysGhostRecon
Battlefield
FIFA
Xbox(Xseries)
Overwatch
johnson&johnson
Amazon
PlayStation5(PS5)
HomeDepot
Cyberpunk2077
CS-GO
GrandTheftAuto(GTA)
Hearthstone
Nvidia
Google
Borderlands
PlayerUnknownsBattlegrounds(PUBG)
Fortnite
RedDeadRedemption(RDR)
AssassinsCreed
type
0
500
1000
1500
2000
2500
count


id
game
sentiment
text
sentiment
Positive
Neutral
Negative
Irrelevant

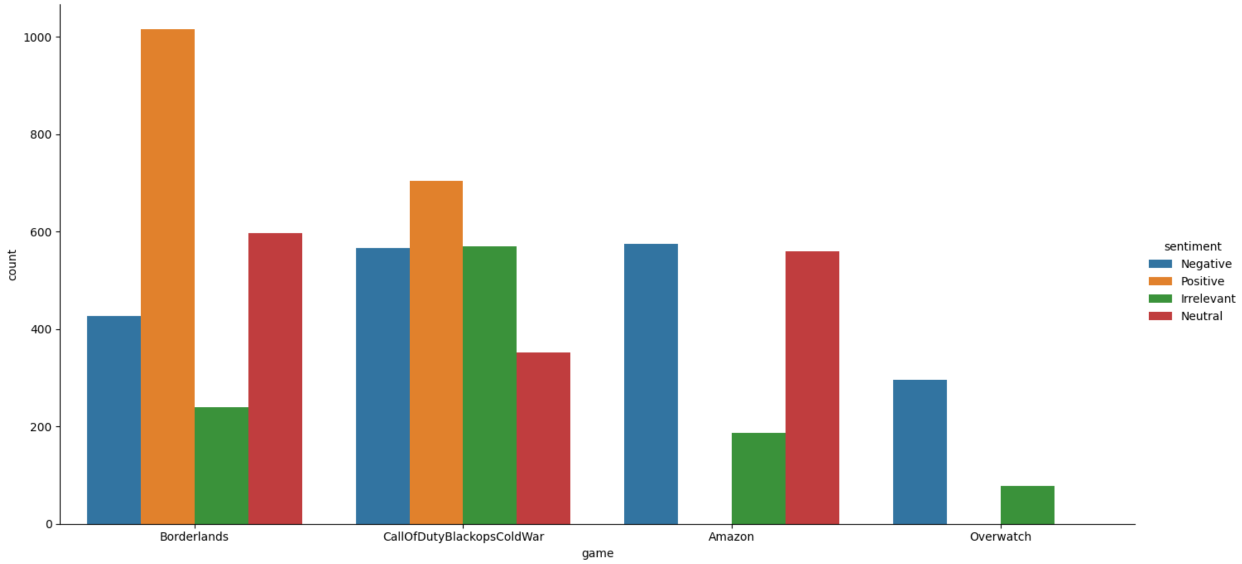

#of id vs count
count
8000
6000
4000
2000
0
2
3
4
5
6
type



# of tv shows vs movie
Borderlands
CallOfDutyBlackopsColdWar
Amazon
Overwatch
count
0
500
1000
1500
2000
type



0id	**999**
Game	**32**
Sentiment	**4**
Text	**998**



0id999game32sentiment4text998