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"])

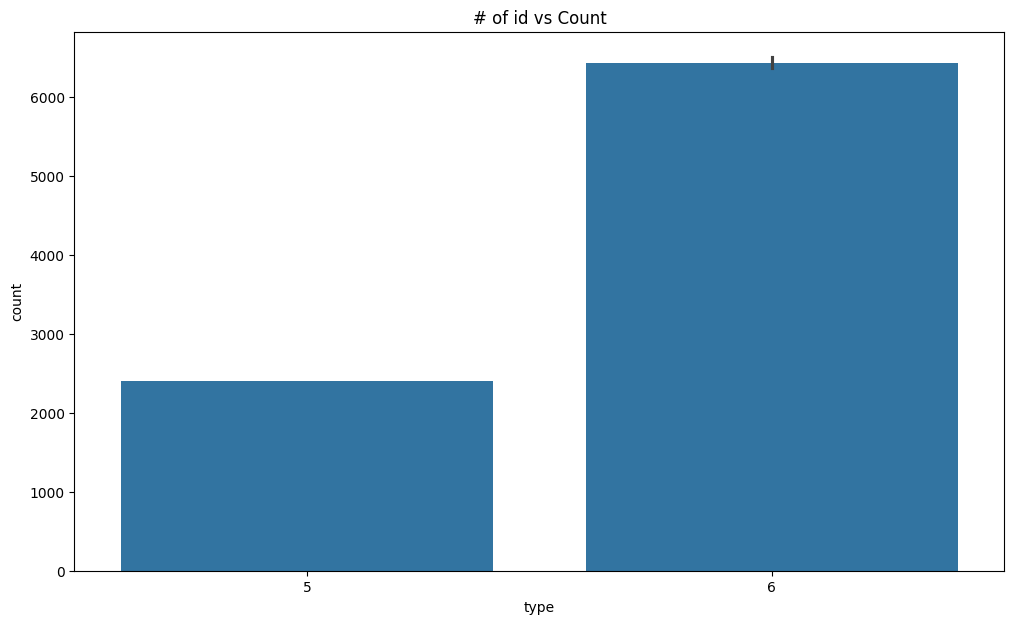
v\_data["game"]=label\_encoder.fit\_transform(v\_data["game"])

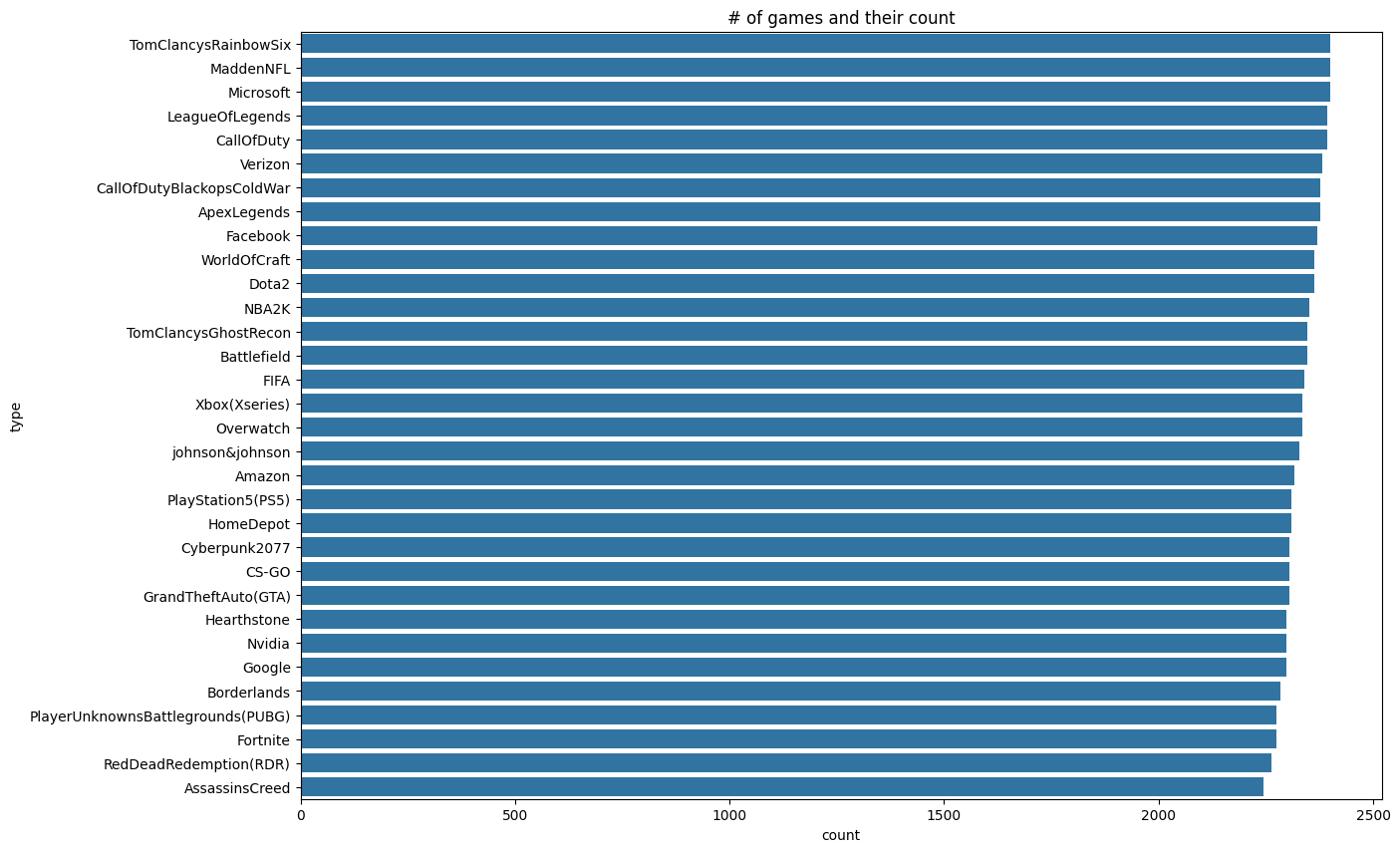
data=data.drop(['id'],axis=1)

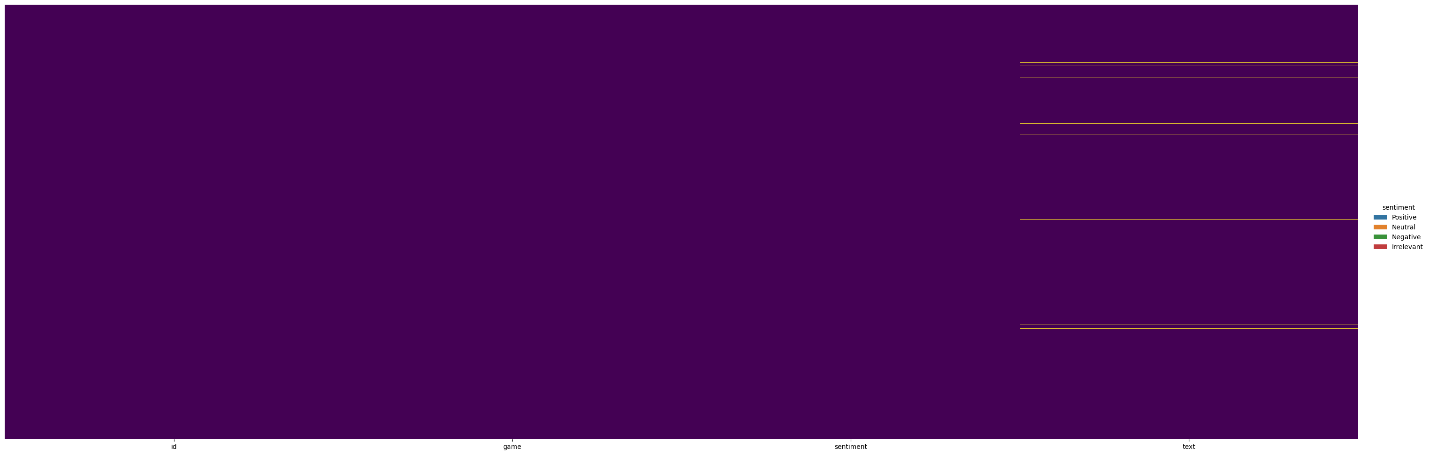
data

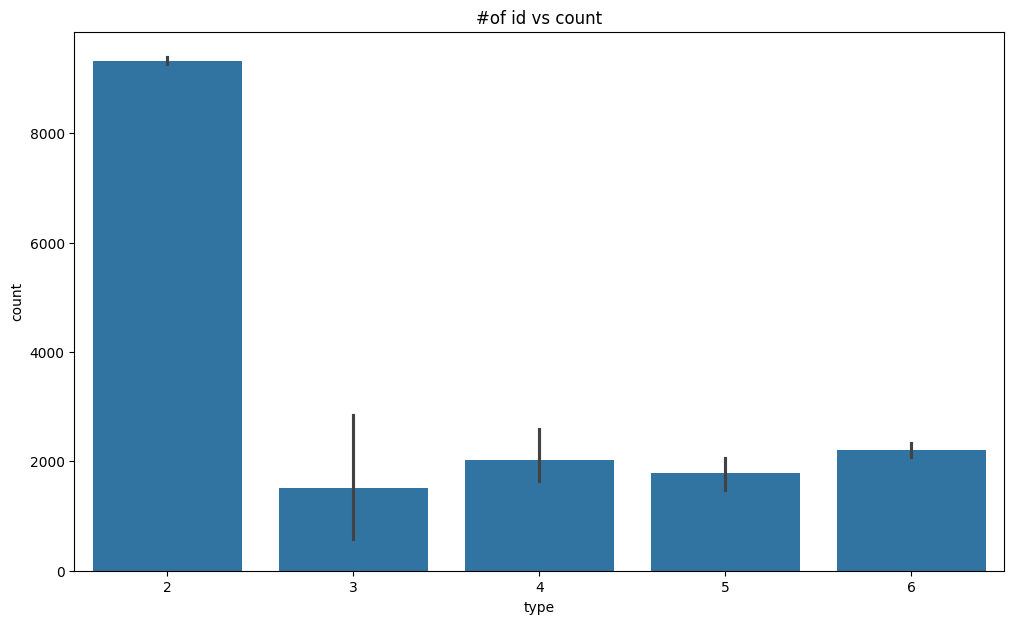
data.nunique()

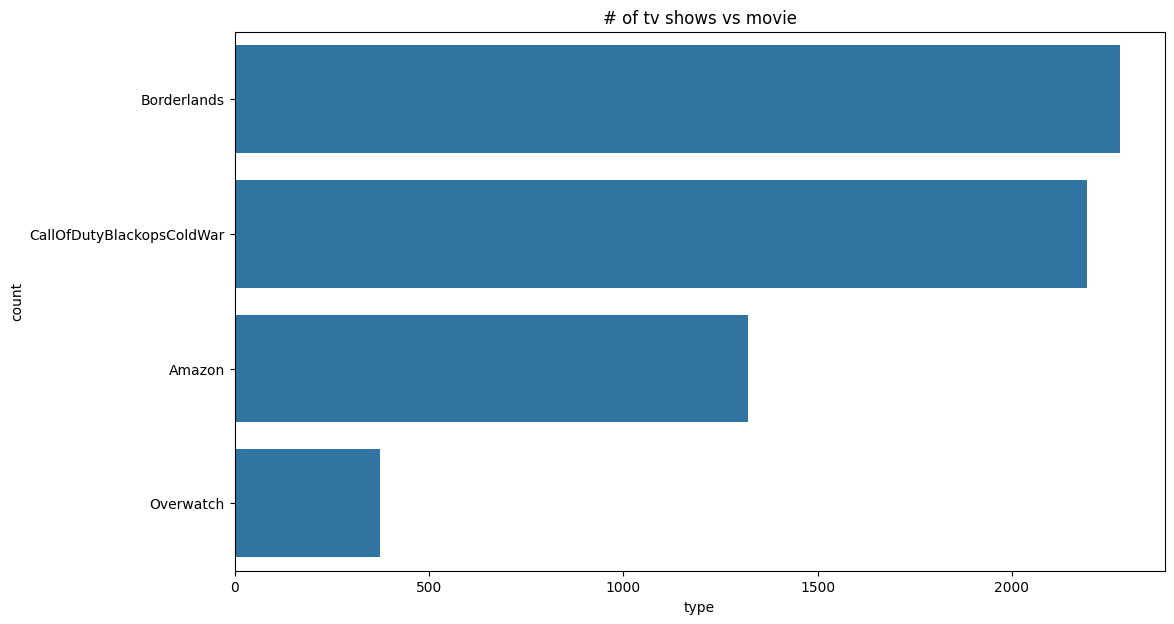
v\_data.nunique

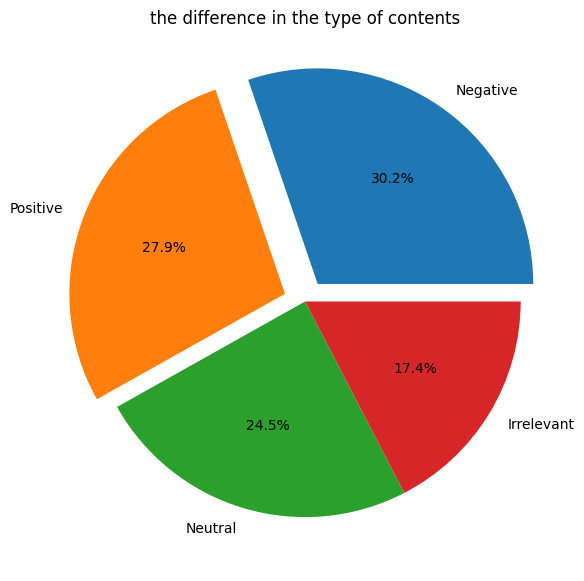










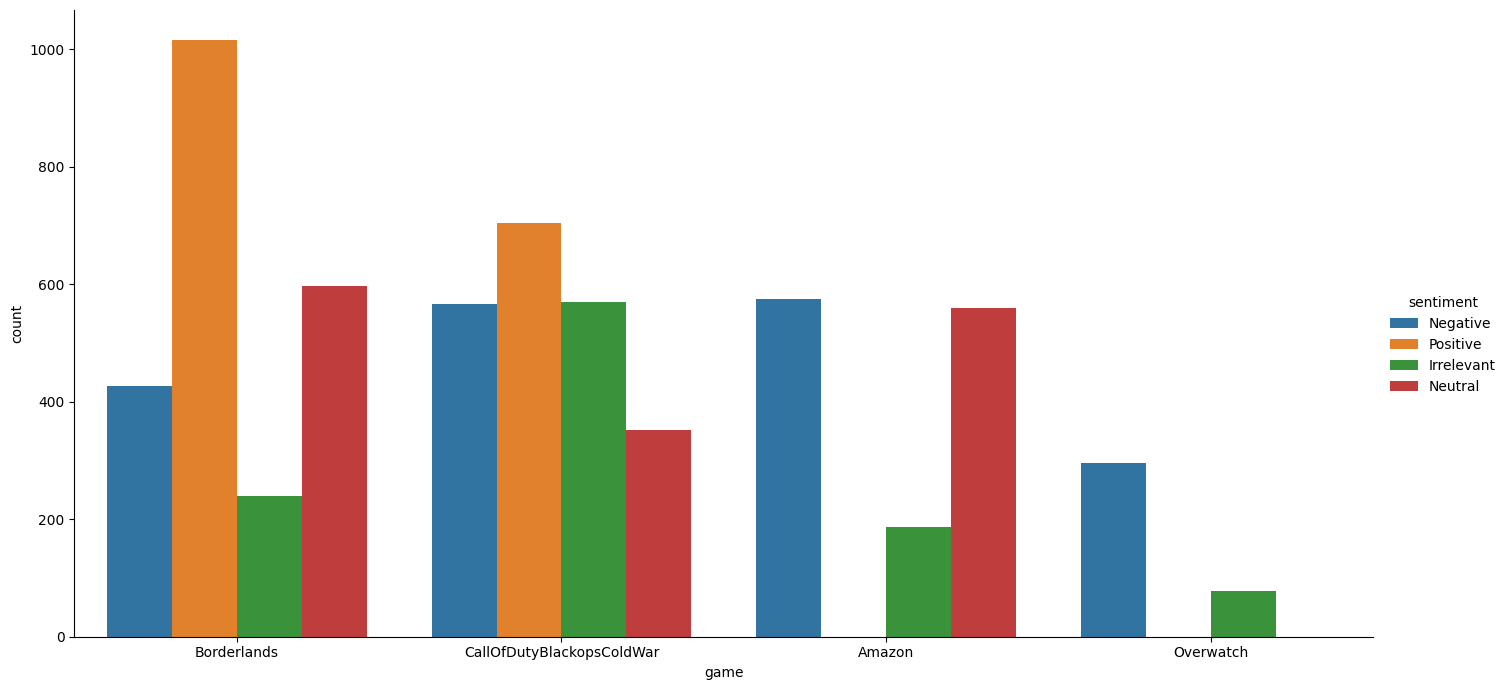


**0id 999**

**Game 32**

**Sentiment 4**

**Text 998**



0id999game32sentiment4text998