```
import pandas as pd
from google.colab import files
# Upload Dataset 1
uploaded_file_1 = files.upload()
```

Choose Files No file chosen Upload widget is only available when the cell has been executed in the current browser session. Please rerun this cell to enable.

Saving twitter_training.csv to twitter_training.csv

```
# Upload Dataset 2
uploaded_file_2 = files.upload()
```

Choose Files No file chosen Upload widget is only available when the cell has been executed in the current browser session. Please rerun this cell to enable.

Saving twitter_validation.csv to twitter_validation.csv

```
cols=['TweetID', 'Topic', 'Target', 'Text']
train=pd.read_csv('twitter_training.csv', names = cols)
valid=pd.read_csv('twitter_validation.csv', names = cols)
```

```
train
```

	TweetID	Topic	Target	Text
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 ...
...	...	...	...	...
74677	9200	Nvidia	Positive	Just realized that the Windows partition of my...
74678	9200	Nvidia	Positive	Just realized that my Mac window partition is ...
74679	9200	Nvidia	Positive	Just realized the windows partition of my Mac ...
74680	9200	Nvidia	Positive	Just realized between the windows partition of...
74681	9200	Nvidia	Positive	Just like the windows partition of my Mac is l...

74682 rows × 4 columns

```
valid
```

	TweetID	Topic	Target	Text
0	3364	Facebook	Irrelevant	I mentioned on Facebook that I was struggling ...
1	352	Amazon	Neutral	BBC News - Amazon boss Jeff Bezos rejects clai...
2	8312	Microsoft	Negative	@Microsoft Why do I pay for WORD when it funct...
3	4371	CS-GO	Negative	CSGO matchmaking is so full of closet hacking,...
4	4433	Google	Neutral	Now the President is slapping Americans in the...
...	...	...	...	...
995	4891	GrandTheftAuto(GTA)	Irrelevant	⭐ Toronto is the arts and culture capital of ...
996	4359	CS-GO	Irrelevant	tHIS IS ACTUALLY A GOOD MOVE TOT BRING MORE VI...
997	2652	Borderlands	Positive	Today sucked so it’s time to drink wine n play...
998	8069	Microsoft	Positive	Bought a fraction of Microsoft today. Small wins.
999	6960	johnson&johnson	Neutral	Johnson & Johnson to stop selling talc baby po...

```
df = pd.concat([train, valid], ignore_index = False)
df
```

	TweetID	Topic	Target	Text
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 ...
...	...	...	...	...
995	4891	GrandTheftAuto(GTA)	Irrelevant	⭐ Toronto is the arts and culture capital of ...
996	4359	CS-GO	Irrelevant	tHIS IS ACTUALLY A GOOD MOVE TOT BRING MORE VI...
997	2652	Borderlands	Positive	Today sucked so it’s time to drink wine n play...
998	8069	Microsoft	Positive	Bought a fraction of Microsoft today. Small wins.
999	6960	johnson&johnson	Neutral	Johnson & Johnson to stop selling talc baby po...

75682 rows × 4 columns

```
df.describe()
```

	TweetID
count	75682.000000
mean	6432.579583
std	3740.243463
min	1.000000
25%	3196.000000
50%	6423.000000
75%	9602.000000
max	13200.000000

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Index: 75682 entries, 0 to 999
Data columns (total 4 columns):
 #   Column   Non-Null Count  Dtype
---  ------   --------------  -----
 0   TweetID  75682 non-null  int64
 1   Topic    75682 non-null  object
 2   Target   75682 non-null  object
 3   Text     74996 non-null  object
dtypes: int64(1), object(3)
memory usage: 2.9+ MB
```

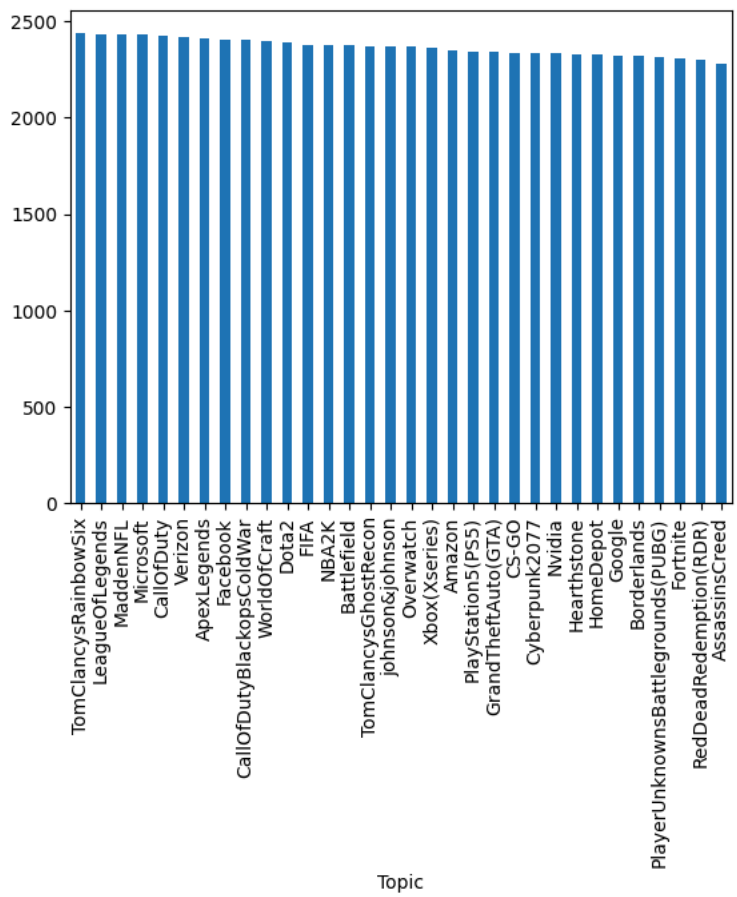
```
df['Topic'].unique()
df['Topic'].value_counts()
```

```
Topic
TomClancysRainbowSix                 2435
LeagueOfLegends                      2431
MaddenNFL                            2429
Microsoft                            2428
CallOfDuty                           2425
Verizon                              2414
ApexLegends                          2412
Facebook                             2403
CallOfDutyBlackopsColdWar            2403
WorldOfCraft                         2394
Dota2                                2391
FIFA                                 2378
NBA2K                                2373
Battlefield                          2372
TomClancysGhostRecon                 2368
johnson&johnson                      2367
Overwatch                            2366
Xbox(Xseries)                        2360
Amazon                               2350
```

```
PlayStation5(PS5)                    2343
GrandTheftAuto(GTA)                  2339
CS-GO                                2336
Cyberpunk2077                        2334
Nvidia                               2333
Hearthstone                          2330
HomeDepot                            2328
Google                               2322
Borderlands                          2319
PlayerUnknownsBattlegrounds(PUBG)    2312
Fortnite                             2308
RedDeadRedemption(RDR)               2302
AssassinsCreed                       2277
Name: count, dtype: int64
```

```
df['Topic'].value_counts().plot(kind = 'bar')
```

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




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

```
TweetID      0
Topic        0
Target       0
Text       686
dtype: int64
```

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

```
3217
```

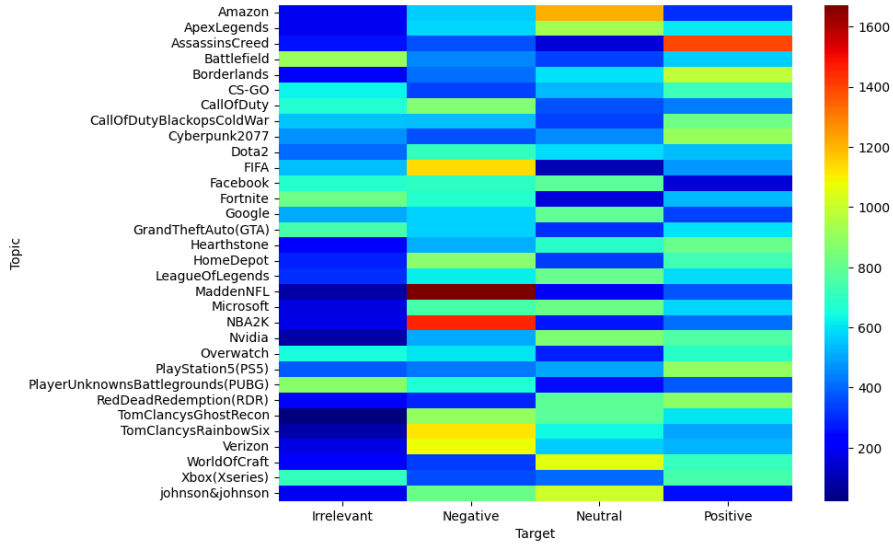
```
df.dropna(inplace=True)
df.drop_duplicates(inplace=True)
df.isnull().sum()
df.duplicated().sum()
```

```
0
```

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

```
plt.figure(figsize=(9, 7))
crosstab = pd.crosstab(index=df['Topic'], columns=df['Target'])
sns.heatmap(crosstab, cmap = 'jet')
```

```
<Axes: xlabel='Target', ylabel='Topic'>
```

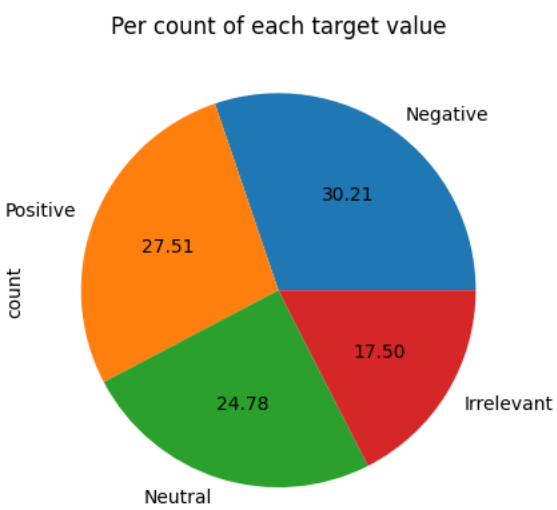



```
crosstab
```

Target Topic	Irrelevant	Negative	Neutral	Positive
Amazon	188	566	1210	306
ApexLegends	188	577	927	613
AssassinsCreed	257	366	155	1393
Battlefield	912	449	345	563
Borderlands	239	415	590	978
CS-GO	624	335	530	721
CallOfDuty	668	865	370	430
CallOfDutyBlackopsColdWar	548	542	343	823
Cyberpunk2077	462	361	458	908
Dota2	401	709	586	544
FIFA	544	1131	103	478
Facebook	676	693	777	154
Fortnite	823	680	160	528
Google	507	572	790	341
GrandTheftAuto(GTA)	749	574	303	592
Hearthstone	220	516	687	814
HomeDepot	284	873	332	732
LeagueOfLegends	300	617	811	586
MaddenNFL	87	1672	191	376
Microsoft	167	750	819	581
NBA2K	175	1454	265	410
Nvidia	86	509	857	762
Overwatch	650	608	283	690
PlayStation5(PS5)	382	423	496	897
PlayerUnknownsBattlegrounds(PUBG)	877	661	254	383
RedDeadRedemption(RDR)	205	290	786	890
TomClancysGhostRecon	23	893	781	608
TomClancysRainbowSix	92	1113	638	506
Verizon	177	1073	558	522
WorldOfCraft	213	328	1059	717

```
from wordcloud import WordCloud
import matplotlib.pyplot as plt
topic_list = ' '.join(crosstab.index)  # Assuming crosstab is your DataFrame or Series containing topics
wc = WordCloud(width=1000, height=500).generate(topic_list)

plt.figure(figsize=(10, 5))  # Adjusting figure size for better display
plt.imshow(wc, interpolation='bilinear')
plt.axis('off')  # Turning off axis labels
plt.show()
```

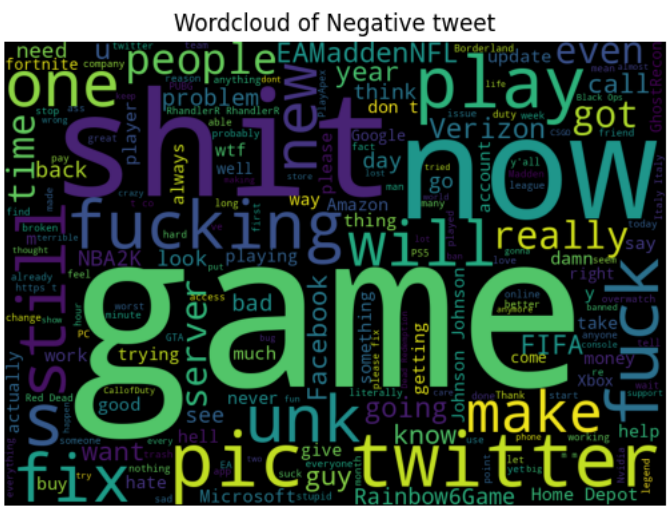

```
df['Target'].value_counts().plot(kind='pie',autopct='%.2f')
plt.title("Per count of each target value")
plt.show()
```




```
target_count = df['Target'].value_counts().reset_index()
target_count
plt.figure(figsize=(15,5))
ax = sns.barplot(data=target_count,x='Target',y='count',palette='cubehelix')
for bars in ax.containers:
    ax.bar_label(bars)

plt.title("Count of each target value")
plt.show()
```

```
<ipython-input-39-fb2c9f71b21f>:4: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in v0.14.0.

  ax = sns.barplot(data=target_count,x='Target',y='count',palette='cubehelix')
```


```
wc = WordCloud(width=1000,height=700,min_font_size=10,background_color='black')
negative = wc.generate(df[df['Target']=='Negative']['Text'].str.cat(sep=" "))
plt.title('Wordcloud of Negative tweet')
plt.axis('off')
plt.imshow(negative)
plt.show()
```




```
df['sentiment'] = df['Target'].replace({'Positive':1,'Negative':0,'Neutral':2,'Irrelevant':3})
```

```
df
```

	TweetID	Topic	Target	Text	sentiment
0	2401	Borderlands	Positive	im getting on borderlands and i will murder yo...	1
1	2401	Borderlands	Positive	I am coming to the borders and I will kill you...	1
2	2401	Borderlands	Positive	im getting on borderlands and i will kill you ...	1
[illegible]	[illegible]	[illegible]	[illegible]	Miracle World\...	[illegible]

```
!pip install nltk
from nltk.tokenize import word_tokenize
```

```
Requirement already satisfied: nltk in /usr/local/lib/python3.10/dist-packages (3.8.1)
Requirement already satisfied: click in /usr/local/lib/python3.10/dist-packages (from nltk) (8.1.7)
Requirement already satisfied: joblib in /usr/local/lib/python3.10/dist-packages (from nltk) (1.4.2)
Requirement already satisfied: regex>=2021.8.3 in /usr/local/lib/python3.10/dist-packages (from nltk) (2024.5.15)
Requirement already satisfied: tqdm in /usr/local/lib/python3.10/dist-packages (from nltk) (4.66.4)
```

```
import nltk
from nltk.corpus import stopwords

nltk.download('stopwords')
print(stopwords.words('english'))
```

```
['i', 'me', 'my', 'myself', 'we', 'our', 'ours', 'ourselves', 'you', "you're", "you've", "you'll", "you'd", 'your', 'yours', 'yourself',
[nltk_data] Downloading package stopwords to /root/nltk_data...
[nltk_data]   Unzipping corpora/stopwords.zip.
```

```
l = []
text = df['Text']

for t in text:
    if type(t) not in l:
        l.append(type(t))
print(l)
```

```
[<class 'str'>]
```

```
import nltk
from nltk.tokenize import word_tokenize
import re
nltk.download('punkt') # Download the 'punkt' resource for sentence tokenization

modified_text = []

rows = len(text)

for ithText in df['Text']:

    ithText = ithText.lower()  # Make text lowercase
    ithText = re.sub(r'[^\w\s]', '', ithText) # Remove punctuations and commas
    ithText = re.sub(r'\d+', '', ithText)


    tokens = word_tokenize(ithText)  # Extract tokens of each word
    words = set(stopwords.words('english'))
    doc = [word for word in tokens if word not in words]
    finalText = ' '.join(doc)
```