Assignment 4

1. Name : Vaibhav Bichave
2. class : BE 10 (IT)
3. Roll No.: 43209

Problem Statement :

Text classification for Sentimental analysis using KNN.

In [1]:
```
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
import warnings
warnings.filterwarnings
%matplotlib inline
import re
import string
from wordcloud import WordCloud
from nltk.tokenize import sent_tokenize, word_tokenize
from nltk.corpus import stopwords
from nltk.stem import WordNetLemmatizer, PorterStemmer
from nltk import pos_tag, ne_chunk
from nltk.chunk import tree2conlltags

import seaborn as sns
import matplotlib.pyplot as plt
from collections import Counter

import nltk
nltk.download('stopwords')
nltk.download('punkt')
nltk.download('wordnet')
nltk.download('averaged_perceptron_tagger')
nltk.download('maxent_ne_chunker')
nltk.download('words')
nltk.download('omw-1.4')

import warnings
warnings.filterwarnings("ignore")
```

```
[nltk_data] Downloading package stopwords to
[nltk_data]     C:\Users\Admin\AppData\Roaming\nltk_data...
[nltk_data]   Package stopwords is already up-to-date!
[nltk_data] Downloading package punkt to
[nltk_data]     C:\Users\Admin\AppData\Roaming\nltk_data...
[nltk_data]   Package punkt is already up-to-date!
[nltk_data] Downloading package wordnet to
[nltk_data]     C:\Users\Admin\AppData\Roaming\nltk_data...
[nltk_data]   Package wordnet is already up-to-date!
[nltk_data] Downloading package averaged_perceptron_tagger to
[nltk_data]     C:\Users\Admin\AppData\Roaming\nltk_data...
[nltk_data]   Package averaged_perceptron_tagger is already up-to-
[nltk_data]       date!
[nltk_data] Downloading package maxent_ne_chunker to
[nltk_data]     C:\Users\Admin\AppData\Roaming\nltk_data...
[nltk_data]   Package maxent_ne_chunker is already up-to-date!
[nltk_data] Downloading package words to
[nltk_data]     C:\Users\Admin\AppData\Roaming\nltk_data...
[nltk_data]   Package words is already up-to-date!
[nltk_data] Downloading package omw-1.4 to
[nltk_data]     C:\Users\Admin\AppData\Roaming\nltk_data...
[nltk_data]   Package omw-1.4 is already up-to-date!
```

In [2]:
```
columns = ["Id","Entity","Target","Text"]
data = pd.read_csv("twitter_training.csv", names=columns,header=None)
```

In [3]:
```
data.head()
```

Out[3]:

	Id	Entity	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 ...

In [4]:
```
df = data[["Text","Target"]]
```

In [5]:
```
df.head()
```

Out[5]:

	Text	Target
0	im getting on borderlands and i will murder yo...	Positive
1	I am coming to the borders and I will kill you...	Positive
2	im getting on borderlands and i will kill you ...	Positive
3	im coming on borderlands and i will murder you...	Positive
4	im getting on borderlands 2 and i will murder ...	Positive

In [6]:
```
df.shape
```

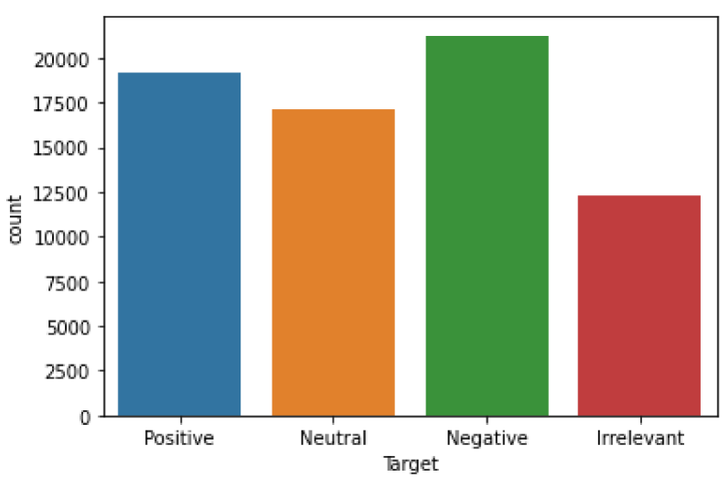
Out[6]:
```
(74682, 2)
```

In [7]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 74682 entries, 0 to 74681
Data columns (total 2 columns):
 #   Column  Non-Null Count  Dtype 
---  ------  --------------  ----- 
 0   Text    73996 non-null  object
 1   Target  74682 non-null  object
dtypes: object(2)
memory usage: 1.1+ MB
```

In [8]:
```
df= df.drop_duplicates()
```

In [9]:
```
sns.countplot(x="Target",data=df);
```



In [10]:
```
sentiment = []

for i in df["Target"]:
    if i == "Positive":
        sentiment.append(1)
    elif (i == "Irrelevant") or (i == "Neutral"):
        sentiment.append(0)
    else:
        sentiment.append(-1)
df["Sentiment"] = sentiment
```

In [11]:
```
df.head()
```

Out[11]:

	Text	Target	Sentiment
0	im getting on borderlands and i will murder yo...	Positive	1
1	I am coming to the borders and I will kill you...	Positive	1
2	im getting on borderlands and i will kill you ...	Positive	1
3	im coming on borderlands and i will murder you...	Positive	1
4	im getting on borderlands 2 and i will murder ...	Positive	1

In [12]:
```
stop_words = set(stopwords.words("english"))
```

Text Cleaner

In [13]:
```
df["Text"] = df["Text"].str.replace("\d","")
```

In [14]:
```
def cleaner(data):
    # Tokens
    tokens = word_tokenize(str(data).replace("'", "").lower())

    # Remove Puncs
    without_punc = [w for w in tokens if w.isalpha()]

    # Stopwords
    without_sw = [t for t in without_punc if t not in stop_words]

    # Lemmatize
    text_len = [WordNetLemmatizer().lemmatize(t) for t in without_sw]
    # Stem
    text_cleaned = [PorterStemmer().stem(w) for w in text_len]

    return " ".join(text_cleaned)
```

In [15]:
```
df["Text"] = df["Text"].apply(cleaner)
df["Text"].head()
```

Out[15]:
```
0     im get borderland murder
1              come border kill
2       im get borderland kill
3    im come borderland murder
4     im get borderland murder
Name: Text, dtype: object
```

In [16]:
```
df["Text"]=df["Text"].str.replace("im","")
df["Text"].head()
```

Out[16]:
```
0      get borderland murder
1           come border kill
2        get borderland kill
3     come borderland murder
4      get borderland murder
Name: Text, dtype: object
```

Rare Words

In [17]:
```
rare_words = pd.Series(" ".join(df["Text"]).split()).value_counts()
rare_words
```

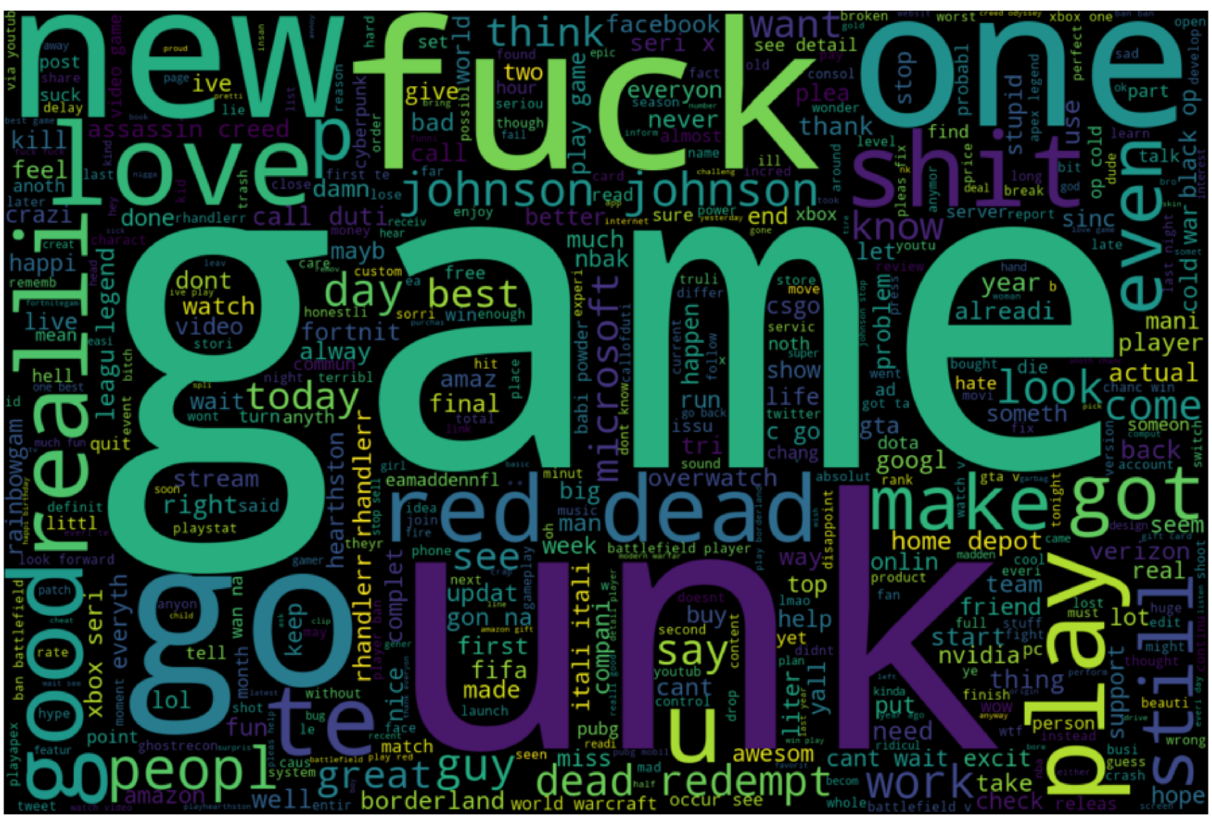
Out[17]:
```
game              10787
play               6822
get                5567
like               5153
go                 4216
                   ...
spokesperson          1
tgo                   1
thatwhat              1
hentaithick           1
adh                   1
Length: 22234, dtype: int64
```

In [18]:
```
rare_words = rare_words[rare_words <= 2]
```

In [19]:
```
df["Text"] = df["Text"].apply(lambda x: " ".join([i for i in x.split() if i n
```

Word Cloud

In [20]:
```
plt.figure(figsize=(16,12))
wordcloud = WordCloud(background_color="black",max_words=500, width=1500, hei
plt.imshow(wordcloud, interpolation='bilinear')
plt.axis("off")
plt.show()
```

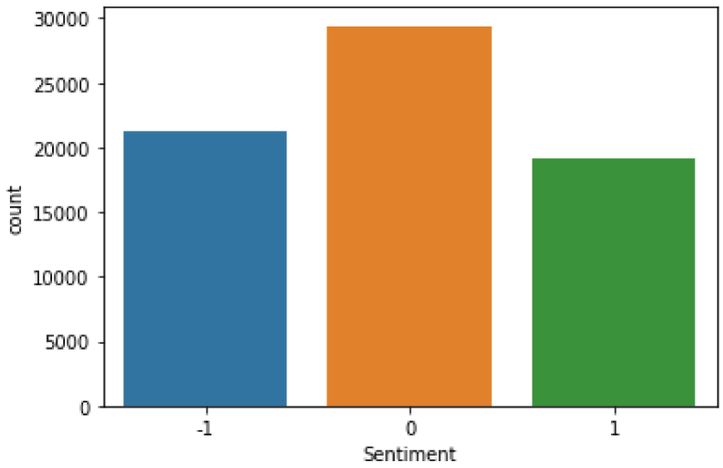


Train test split

In [21]:
```
from sklearn.model_selection import train_test_split
from sklearn.metrics import confusion_matrix,classification_report,accuracy_s
from sklearn.naive_bayes import MultinomialNB, BernoulliNB
from sklearn.ensemble import RandomForestClassifier
```

In [22]:
```
X = df["Text"]
y = df["Sentiment"]
```

In [23]:
```
sns.countplot(y,data=df);
```



In [24]:
```
X_train,X_test,y_train,y_test = train_test_split(X,y, test_size = 0.30,random
```

Count Vectorizer

In [25]:
```
from sklearn.feature_extraction.text import CountVectorizer
```

In [26]:
```
vt = CountVectorizer(analyzer="word")
X_train_count = vt.fit_transform(X_train)
X_test_count = vt.transform(X_test)
```

In [27]:
```
print(X_train_count.toarray())
X_train_count
```

```
[[0 0 0 ... 0 0 0]
 [0 0 0 ... 0 0 0]
 [0 0 0 ... 0 0 0]
 ...
 [0 0 0 ... 0 0 0]
 [0 0 0 ... 0 0 0]
 [0 0 0 ... 0 0 0]]
```

Out[27]:
```
<48841x14108 sparse matrix of type '<class 'numpy.int64'>'
        with 480354 stored elements in Compressed Sparse Row format>
```

KNN

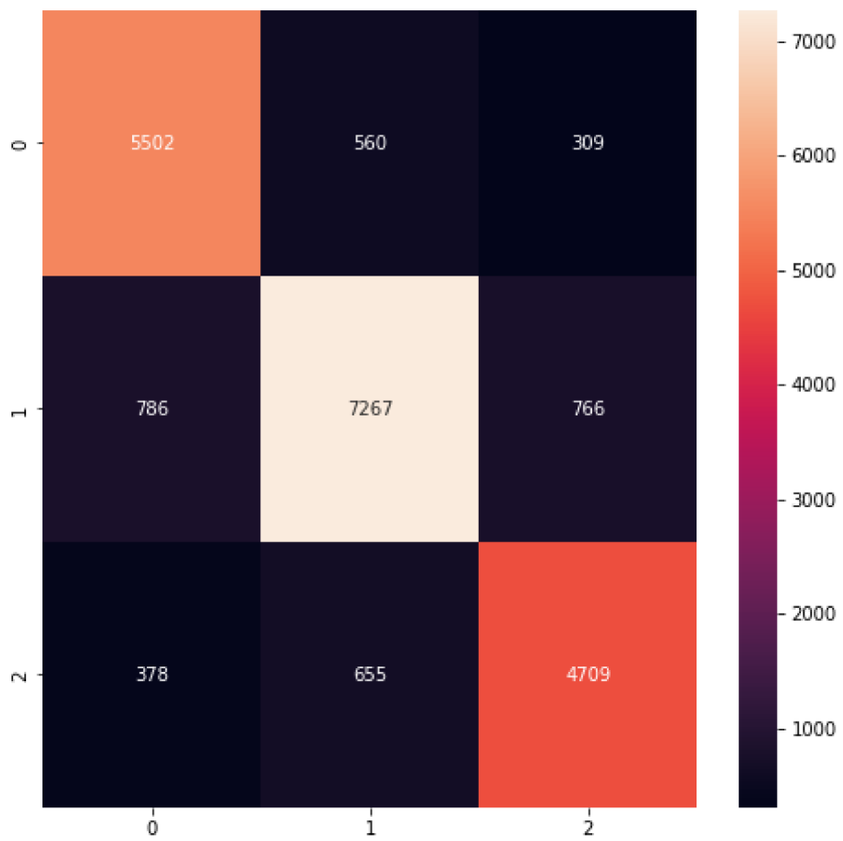
In [28]:
```
from sklearn.neighbors import KNeighborsClassifier
knn = KNeighborsClassifier()
knn_model = knn.fit(X_train_count,y_train)
```

In [29]:
```
knn_pred = knn_model.predict(X_test_count)
knn_train_pred = knn_model.predict(X_train_count)
```

In [30]:
```
print("X Test")
print(classification_report(y_test,knn_pred))
print("X Train")
print(classification_report(y_train,knn_train_pred))

plt.figure(figsize=(8,8))
sns.heatmap(confusion_matrix(y_test,knn_pred),annot = True,fmt = "d")
```

```
X Test
              precision    recall  f1-score   support

          -1       0.83      0.86      0.84      6371
           0       0.86      0.82      0.84      8819
           1       0.81      0.82      0.82      5742

    accuracy                           0.83     20932
   macro avg       0.83      0.84      0.83     20932
weighted avg       0.84      0.83      0.83     20932

X Train
              precision    recall  f1-score   support

          -1       0.92      0.94      0.92     14867
           0       0.93      0.92      0.92     20577
           1       0.92      0.91      0.91     13397

    accuracy                           0.92     48841
   macro avg       0.92      0.92      0.92     48841
weighted avg       0.92      0.92      0.92     48841
```

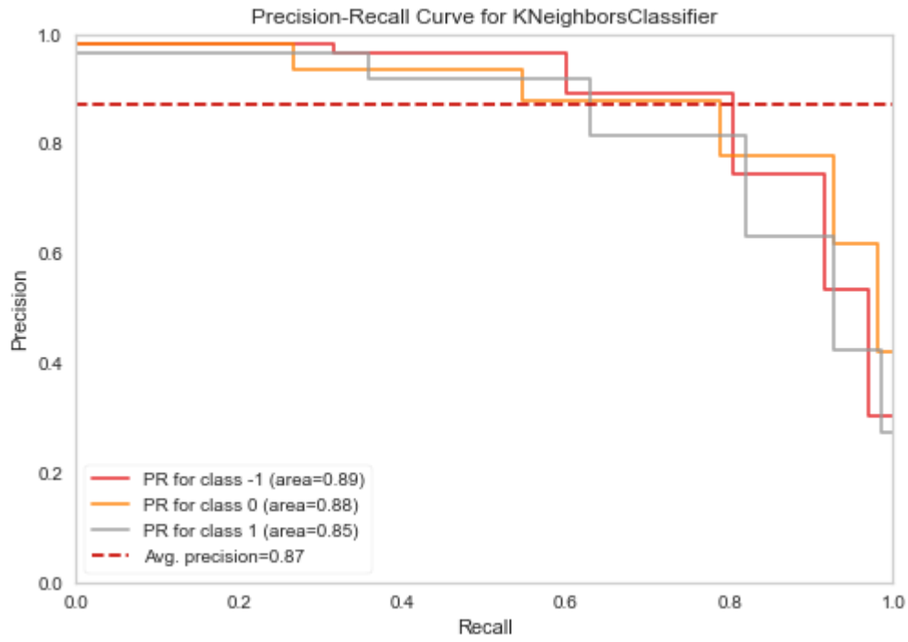
Out[30]: <AxesSubplot:>



In [32]:
```
pip install yellowbrick
```

```
Collecting yellowbrickNote: you may need to restart the kernel to use update
d packages.
  Downloading yellowbrick-1.5-py3-none-any.whl (282 kB)
     ------------------------------------ 282.6/282.6 kB 268.6 kB/s eta 0:0
0:00
Requirement already satisfied: scikit-learn>=1.0.0 in c:\users\admin\appdata
\local\programs\python\python38\lib\site-packages (from yellowbrick) (1.0.1)
Requirement already satisfied: cycler>=0.10.0 in c:\users\admin\appdata\loca
l\programs\python\python38\lib\site-packages (from yellowbrick) (0.11.0)
Requirement already satisfied: numpy>=1.16.0 in c:\users\admin\appdata\local
\programs\python\python38\lib\site-packages (from yellowbrick) (1.21.4)
Requirement already satisfied: matplotlib!=3.0.0,>=2.0.2 in c:\users\admin\a
ppdata\local\programs\python\python38\lib\site-packages (from yellowbrick)
(3.5.0)
Requirement already satisfied: scipy>=1.0.0 in c:\users\admin\appdata\local
\programs\python\python38\lib\site-packages (from yellowbrick) (1.7.2)
Requirement already satisfied: pillow>=6.2.0 in c:\users\admin\appdata\local
\programs\python\python38\lib\site-packages (from matplotlib!=3.0.0,>=2.0.2-
>yellowbrick) (8.4.0)
Requirement already satisfied: fonttools>=4.22.0 in c:\users\admin\appdata\l
ocal\programs\python\python38\lib\site-packages (from matplotlib!=3.0.0,>=2.
0.2->yellowbrick) (4.28.2)
Requirement already satisfied: kiwisolver>=1.0.1 in c:\users\admin\appdata\l
ocal\programs\python\python38\lib\site-packages (from matplotlib!=3.0.0,>=2.
0.2->yellowbrick) (1.3.2)
Requirement already satisfied: setuptools-scm>=4 in c:\users\admin\appdata\l
ocal\programs\python\python38\lib\site-packages (from matplotlib!=3.0.0,>=2.
0.2->yellowbrick) (6.3.2)
Requirement already satisfied: pyparsing>=2.2.1 in c:\users\admin\appdata\lo
cal\programs\python\python38\lib\site-packages (from matplotlib!=3.0.0,>=2.
0.2->yellowbrick) (3.0.6)
Requirement already satisfied: python-dateutil>=2.7 in c:\users\admin\appdat
a\local\programs\python\python38\lib\site-packages (from matplotlib!=3.0.0,>
=2.0.2->yellowbrick) (2.8.2)
Requirement already satisfied: packaging>=20.0 in c:\users\admin\appdata\loc
al\programs\python\python38\lib\site-packages (from matplotlib!=3.0.0,>=2.0.
2->yellowbrick) (21.3)
Requirement already satisfied: threadpoolctl>=2.0.0 in c:\users\admin\appdat
a\local\programs\python\python38\lib\site-packages (from scikit-learn>=1.0.0
->yellowbrick) (3.0.0)
Requirement already satisfied: joblib>=0.11 in c:\users\admin\appdata\local
\programs\python\python38\lib\site-packages (from scikit-learn>=1.0.0->yello
wbrick) (1.1.0)
Requirement already satisfied: six>=1.5 in c:\users\admin\appdata\local\prog
rams\python\python38\lib\site-packages (from python-dateutil>=2.7->matplotli
b!=3.0.0,>=2.0.2->yellowbrick) (1.16.0)
Requirement already satisfied: setuptools in c:\users\admin\appdata\local\pr
ograms\python\python38\lib\site-packages (from setuptools-scm>=4->matplotli
b!=3.0.0,>=2.0.2->yellowbrick) (49.2.1)
Requirement already satisfied: tomli>=1.0.0 in c:\users\admin\appdata\local
\programs\python\python38\lib\site-packages (from setuptools-scm>=4->matplot
lib!=3.0.0,>=2.0.2->yellowbrick) (1.2.2)
Installing collected packages: yellowbrick
Successfully installed yellowbrick-1.5
```

In [33]:
```
from yellowbrick.classifier import PrecisionRecallCurve
viz = PrecisionRecallCurve(KNeighborsClassifier(),
                           classes=knn_model.classes_,
                           per_class=True,
                           cmap="Set1")
viz.fit(X_train_count,y_train)
viz.score(X_test_count, y_test)
viz.show();
```



In [38]:
```
accuracy = accuracy_score(y_test,knn_pred)
print("KNN accuracy score :",accuracy)
```

```
KNN accuracy score : 0.8349894897764188
```