Project Report On

Malignant-Comments-Classifier

Submitted by

Aamina Ruvaida

ACKNOWLEDGMENT

I sincerely thank to “Flip Robo Technologies” who let me to work on this project. Also thanked to Data Trained team who gives the guidance and right direction in the completion of project. This project has helped me to understand a lot and learned various thing while completing the project.

References:

Creating Word Clouds in Python | Engineering Education (EngEd) Program | Section

Plot multiple plots in Matplotlib - GeeksforGeeks

Mental Health Effects of Reading Negative Comments Online (verywellmind.com)

TABLE OF CONTENTS:

1. Introduction

1.1 Business Problem Framing

1.2 Conceptual Background of the Domain Problem

1.3 Review of Literature

1.4 Motivation for the Problem Undertaken

2. Analytical Problem Framing

2.1 Mathematical/ Analytical Modelling of the Problem

2.2 Data Sources and their formats

2.3 Data Pre-processing Done

2.4 Data Inputs- Logic- Output Relationships

2.5 Hardware & Software Requirements & Tools Used

3. Model/s Development and Evaluation

3.1 Identification of possible Problem-solving approaches (Methods

3.2 Visualizations

3.3 Testing of Identified Approaches (Algorithms)

3.4 Run and Evaluate Selected Models

3.5 Key Metrics for success in solving problem under consideration

3.6 Interpretation of the Results

4. Conclusion

4.1 Key Findings and Conclusions of the Study

4.2 Learning Outcomes of the Study in respect of Data Science

4.3 Limitations of this work and Scope for Future Work

INTRODUCTION

• Business Problem Framing

❖ The proliferation of social media enables people to express their opinions widely online. However, at the same time, this has resulted in the emergence of conflict and hate, making online environments uninviting for users. Although researchers have found that hate is a problem across multiple platforms, there is a lack of models for online hate detection.

❖ Online hate, described as abusive language, aggression, cyberbullying, hatefulness and many others has been identified as a major threat on online social media platforms. Social media platforms are the most prominent grounds for such toxic behaviour.

❖ There has been a remarkable increase in the cases of cyberbullying and trolls on various social media platforms. Many celebrities and influences are facing backlashes from people and have to come across hateful and offensive comments. This can take a toll on anyone and affect them mentally leading to depression, mental illness, self-hatred and suicidal thoughts.

❖ Internet comments are bastions of hatred and vitriol. While online anonymity has provided a new outlet for aggression and hate speech, machine learning can be used to fight it. The problem we sought to solve was the tagging of internet comments that are aggressive towards other users. This means that insults to third parties such as celebrities will be tagged as unoffensive, but “u are an idiot” is clearly offensive.

❖ Our goal is to build a prototype of online hate and abuse comment classifier which can used to classify hate and offensive comments so that it can be controlled and restricted from spreading hatred and cyberbullying.

• Conceptual Background of the Domain Problem

❖ Internet comments are bastions of hatred and vitriol. While online anonymity has provided a new outlet for aggression and hate speech, machine learning can be used to fight it.

- The problem we sought to solve was the tagging of internet comments that are aggressive towards other users. This means that insults to third parties such as celebrities will be tagged as unoffensive, but “u are an idiot” is clearly offensive.

- Social media platforms are the most prominent grounds for such toxic behaviour. We should such negative comments that are spreading quickly in the social media platform.

- It will affect the people mental health and spread hatred, jealousy, revenge in the heart of people.

• Review of Literature

- The research starts with collecting the comments from different social media platforms in order to restrict and stop the increasing hatred through social media platform. There has been a remarkable increase in the cases of cyberbullying and trolls on various social media platforms.

- Many celebrities and influences are facing backlashes from people and have to come across hateful and offensive comments. This can take a toll on anyone and affect them mentally leading to depression, mental illness, self-hatred and suicidal thoughts. Internet comments are bastions of hatred and vitriol.

- While online anonymity has provided a new outlet for aggression and hate speech, machine learning can be used to fight it. The problem we sought to solve was the tagging of internet comments that are aggressive towards other users. This means that insults to third parties such as celebrities will be tagged as inoffensive, but “u are an idiot” is clearly offensive.

• Motivation for the Problem Undertaken

- This project helps me to understand the social media, how people express their feeling in the public platform. This Malignant Comments Classifier Project contains huge amount of data. Most important on this project is Data Exploration, Featuring Engineering and Visualization of data.

- By analysis the comments, and filter the frequently used comments that has the abusive behaviour. We can restrict those comments using the Machine Learning model. So, that it will not impact the society.

- It is highly possible that just one bad comment that leads to go against the people and create a flight against them. We see that in day-to-day life, with these abusive comments how it is affecting the people mental health. Some may even get depressed, even take extreme danger decision like suicide. We can prevent such cause in the future with our prediction.

Analytical Problem Framing

• Mathematical/ Analytical Modelling of the Problem

- Our goal is to build a prototype of online hate and abuse comment classifier which can used to classify hate and offensive comments so that it can be controlled and restricted from spreading hatred and cyberbullying.

- . I have used a Passive Aggressive Classifier model to classify the comments in terms malignant/highly malignant and also used cross validation to remove overfitting problem while predicted the correct outcome and validate the model.

- Data Analysis: After cleaning the data I have done some analysis on the data by using different types of visualizations.

- Model Building Phase: After collecting the data, I built a machine learning model. Before model building, have done all data pre-processing steps. The complete life cycle of data science that I have used in this project are as follows:

- · Data Cleaning
- · Exploratory Data Analysis
- · Data Pre-processing
- · Model Building
- · Model Evaluation
- · Selecting the best model

- **Data Sources and their formats**
  - ❖ · Data sources are provided internally by the enterprise.

    We have 2 separate data set one for training and another for testing.

    Uploaded the data in the jupyter notebook in the form of csv format.

  - ❖ The data set contains the training set, which has approximately 1,59,000 samples and the test set which contains nearly 1,53,000 samples. All the data samples contain 8 fields which includes ‘Id’, ‘Comments’, ‘Malignant’, ‘Highly malignant’, ‘Rude’, ‘Threat’, ‘Abuse’ and ‘Loathe’.

  - ❖ The label can be either 0 or 1, where 0 denotes a NO while 1 denotes a YES. There are various comments which have multiple labels. The first attribute is a unique ID associated with each comment.

  - ❖ The data set includes:
  - - Malignant: It is the Label column, which includes values 0 and 1, denoting if the comment is malignant or not.
  - - Highly Malignant: It denotes comments that are highly malignant and hurtful.
  - - Rude: It denotes comments that are very rude and offensive.
  - - Threat: It contains indication of the comments that are giving any threat to someone.
  - - Abuse: It is for comments that are abusive in nature.
  - - Loathe: It describes the comments which are hateful and loathing in nature.
  - - ID: It includes unique Ids associated with each comment text given.
  - - Comment text: This column contains the comments extracted from various social media platforms.

    Training dataset:

```
#Loading the train data set
df_train=pd.read_csv('train.csv')
df_train.head(8)
```

	id	comment_text	malignant	highly_malignant	rude	threat	abuse	loathe
0	0000997932d777bf	Explanation\nWhy the edits made under my usern...	0	0	0	0	0	0
1	000103f0d9cfb60f	D'aww! He matches this background colour I'm s...	0	0	0	0	0	0
2	000113f07ec002fd	Hey man, I'm really not trying to edit war. It...	0	0	0	0	0	0
3	0001b41b1c6bb37e	"\nMore\nI can't make any real suggestions on ...	0	0	0	0	0	0
4	0001d958c54c6e35	You, sir, are my hero. Any chance you remember...	0	0	0	0	0	0
5	00025465d4725e87	"\n\nCongratulations from me as well, use the ...	0	0	0	0	0	0
6	0002bcb3da6cb337	COCKSUCKER BEFORE YOU PISS AROUND ON MY WORK	1	1	1	0	1	0
7	00031b1e95af7921	Your vandalism to the Matt Shirvington article...	0	0	0	0	0	0

Test dataset:

```
#Loading the test data set
df_test=pd.read_csv('test.csv')
df_test.head(8)
```

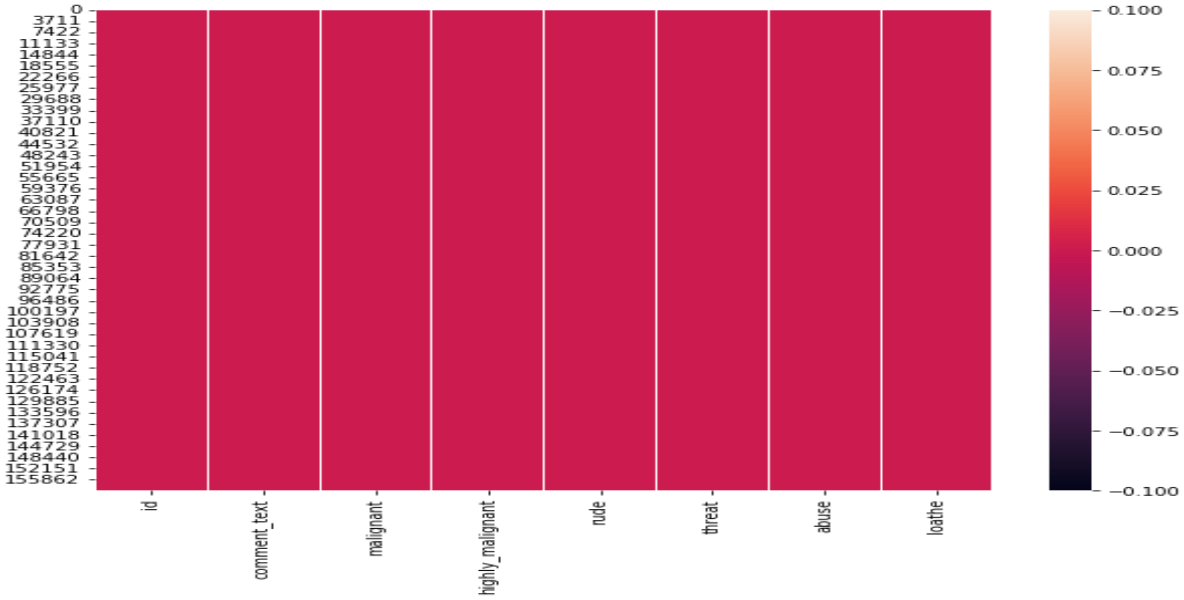
	id	comment_text
0	00001cee341fdb12	Yo bitch Ja Rule is more succesful then you'll...
1	0000247867823ef7	== From RfC == \n\n The title is fine as it is...
2	00013b17ad220c46	" \n\n == Sources == \n\n * Zawe Ashton on Lap...
3	00017563c3f7919a	:If you have a look back at the source, the in...
4	00017695ad8997eb	I don't anonymously edit articles at all.
5	0001ea8717f6de06	Thank you for understanding. I think very high...
6	00024115d4cbde0f	Please do not add nonsense to Wikipedia. Such ...
7	000247e83dcc1211	:Dear god this site is horrible.

- **Data Pre-processing Done**

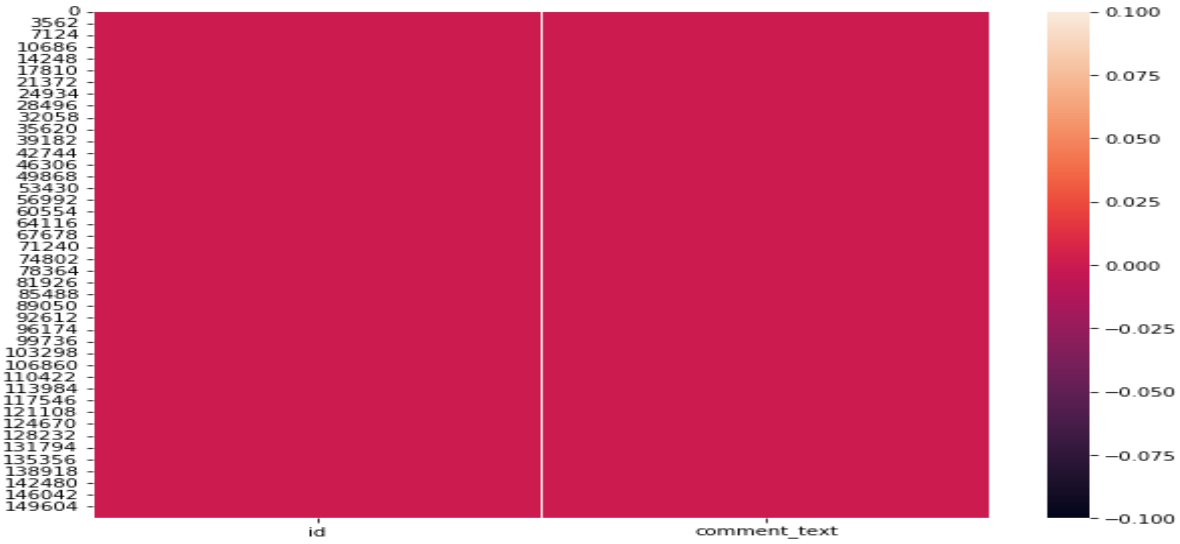
  In our dataset there are is no missing value. Checked the outliers, shape of data, data description, Correlation of data, data type. We have converted the object data type to int.

  Data pre-processing is most important step in any machine learning model.

Finding the Null value in train dataset:



Finding the null value in the test data:



In both train and test data there is no missing value present.

Removing the Stop words, Punctuation:

```
df_train['comment_text']=df_train['comment_text'].str.lower() #Making the comment text to lower case
```

```
#Replacing the '\n' with ''
df_train['comment_text']=df_train['comment_text'].str.replace('\n',' ')
```

```
#Keeping only the words which is starts with a-z, 0-9
from nltk.tokenize import regexp_tokenize
df_train['comment_text']=df_train['comment_text'].apply(lambda x: ' '.join(regexp_tokenize(x,"[a-z']+")))
```

```
#Getting the stop words in English

from nltk.corpus import stopwords
stopwords=set(stopwords.words('english'))
```

```
#Removing the punctuation
df_train['comment_text']=df_train['comment_text'].str.replace("[^\w\d\s]","")
```

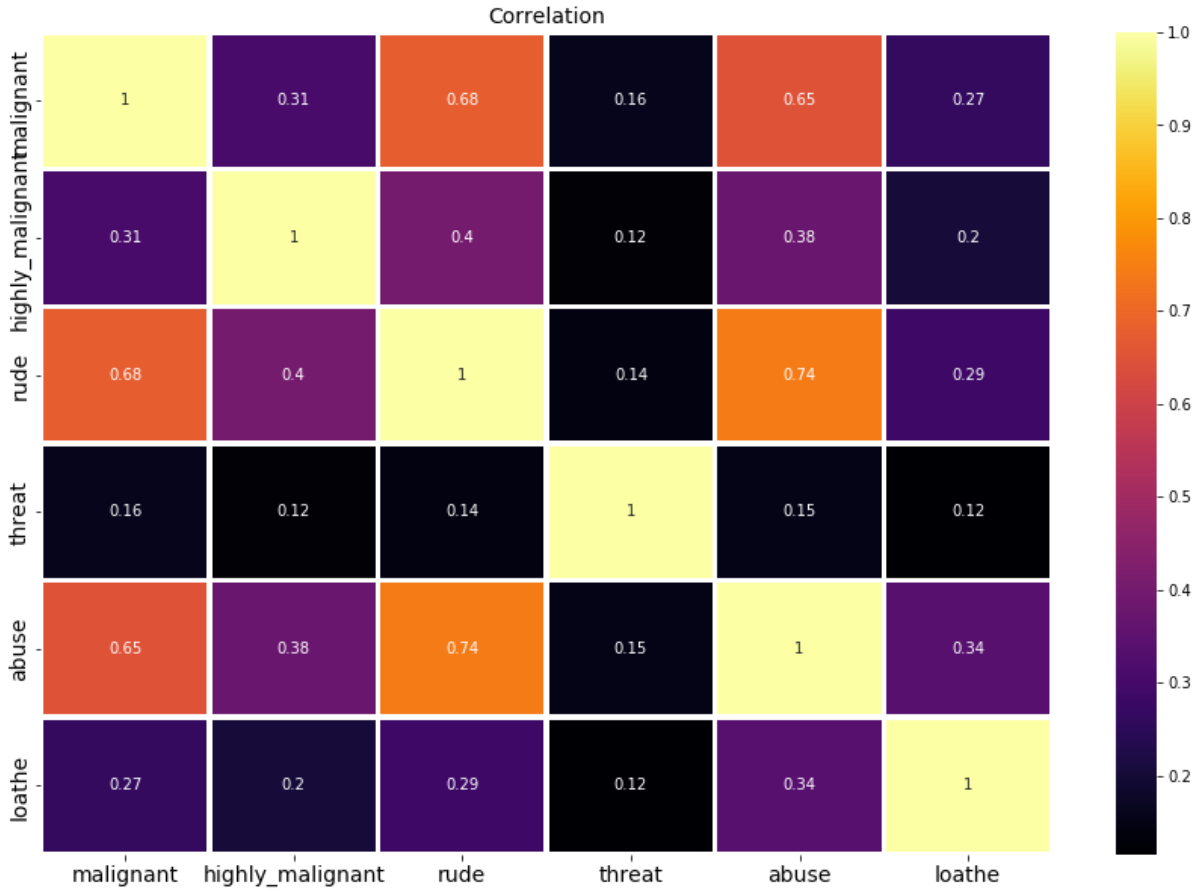
```
df_train.sample(10)
```

	comment_text	malignant	highly_malignant	rude	threat	abuse	loathe
99989	noticed think good start	0	0	0	0	0	0
8938	president rumors snipping confirmed wayne pres...	0	0	0	0	0	0
148146	meta language codes conflicts list conflicts l...	0	0	0	0	0	0
11356	would like suggest link gives detailed breakdo...	0	0	0	0	0	0
21539	nonsense thing formal warning anyways please b...	0	0	0	0	0	0
93401	stop lock hole	1	0	0	0	0	0
40250	yep ed kind enough semi talk hours theyll go p...	0	0	0	0	0	0
33006	retard shut fud good guy scientus user william...	1	0	0	0	1	0
46101	deeceevoice way thinking africoid something li...	0	0	0	0	0	0
78348	excellent observation frania wholeheartedly ag...	0	0	0	0	0	0

- **Data Inputs- Logic- Output Relationships**

➢ Our dataset contains both features and label. Malignant contains the higher count than others.

➢ Based on the comments the comments are classified though there are some common words like fuck , ass etc. which is most common in each kind of category but so far analysed based on the intensity of the words used it is classifying the comment



The above is the correlation of each class.

- **Hardware and Software Requirements and Tools Used**

Tool: Jupyter Notebook

• Web-based interactive computing notebook Environment. Software Requirement

The client environment used is Windows.

Hardware Requirement: • CPU: 2 x 64-bit, 2.8 GHz, 8.00 GT/s CPUs or better.

• Memory: RAM size of 12

Libraries: • Pandas: For reading CSV file, Converting dataset into a data frame, handling date datatype, and more. • Seaborn and matplotlib: For EDA and Visualization.

- import numpy as np: It is defined as a Python package used for performing the various numerical computations and processing of the multidimensional and single dimensional array elements. The calculations using Numpy arrays are faster than the normal Python array.

- import pandas as pd: Pandas is a Python library that is used for faster data analysis, data cleaning and data pre-processing. The data-frame term is coming from Pandas only.

- import matplotlib.pyplot as plt: Matplotlib and Seaborn acts as the backbone of data visualization through Python.

- Matplotlib: It is a Python library used for plotting graphs with the help of other libraries like Numpy and Pandas. It is a powerful tool for visualizing data in Python. It is used for creating statical interferences and plotting 2D graphs of arrays.

- import seaborn as sns: Seaborn is also a Python library used for plotting graphs with the help of Matplotlib considered as a superset of the Matplotlib library. It helps in visualizing univariate and bivariate data.

- With the above sufficient libraries, we can perform pre-processing, data cleaning and can build ML models.

```
#Importing Necessary Libraries

import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.model_selection import train_test_split
from sklearn.model_selection import cross_val_score

import string
import nltk
from nltk.corpus import stopwords
from nltk.stem import PorterStemmer
from nltk.stem import WordNetLemmatizer

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

Model/s Development and Evaluation

- **Identification of possible problem-solving approaches (methods)**

➢ I have checked the statistical description of data.

```
#Checking the description of train dataset
df_train.describe()
```

	malignant	highly_malignant	rude	threat	abuse	loathe
count	159571.000000	159571.000000	159571.000000	159571.000000	159571.000000	159571.000000
mean	0.095844	0.009996	0.052948	0.002996	0.049364	0.008805
std	0.294379	0.099477	0.223931	0.054650	0.216627	0.093420
min	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
25%	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
50%	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
75%	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
max	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000

We see the mean value of each class, their standard deviation, 25th Percentile, median, 75th percentile, minimum and maximum value. The minimum value is 0 and maximum value is 1. It is either 0 and 1.

```
#Checking the description of test dataset
df_test.describe()
```

	id	comment_text
count	153164	153164
unique	153164	153164
top	07262fce3be915b0	Both are about the same festival. Very interes...
freq	1	1

In the test data set only 2 columns are present. Id and comment_text

➢ For this particular project I have used different classification models to predict the outcome of this dataset. After the model implementation Passive Aggressive classifier method predicted the best outcome out of all the models in terms of accuracy score and also, I have used cross validation to flag the problem related overfitting or selection bias for the dataset and hence we can use this model for further evaluation.

- **Testing of Identified Approaches (Algorithms)**

```
from sklearn.linear_model import LogisticRegression, PassiveAggressiveClassifier
from sklearn.pipeline import Pipeline
from sklearn.multiclass import OneVsRestClassifier
from sklearn.metrics import classification_report, confusion_matrix, accuracy_score,auc,roc_curve,roc_auc_score,log_loss
from sklearn.ensemble import RandomForestClassifier
from sklearn.tree import DecisionTreeClassifier
from sklearn.svm import SVC
from sklearn.ensemble import AdaBoostClassifier
from sklearn.naive_bayes import MultinomialNB, ComplementNB
from sklearn.linear_model import SGDClassifier
```

In this project I have used below Machine Learning Algorithm to build the model.

- Logistic Regression
- Multinomial NB
- Complement NB
- Passive Aggressive Classifier
- SGD Classifier

- **Run and evaluate selected models**

Used various machine learning model, after analysis of model, we got the passive aggressive classifier model as the best model for this project. As it gives a good accuracy score.

Logistic Regression:

```
lg=LogisticRegression()
```

```
lg.fit(x_train,y_train)
```

```
LogisticRegression()
```

```
lg_pred=lg.predict(x_test)
```

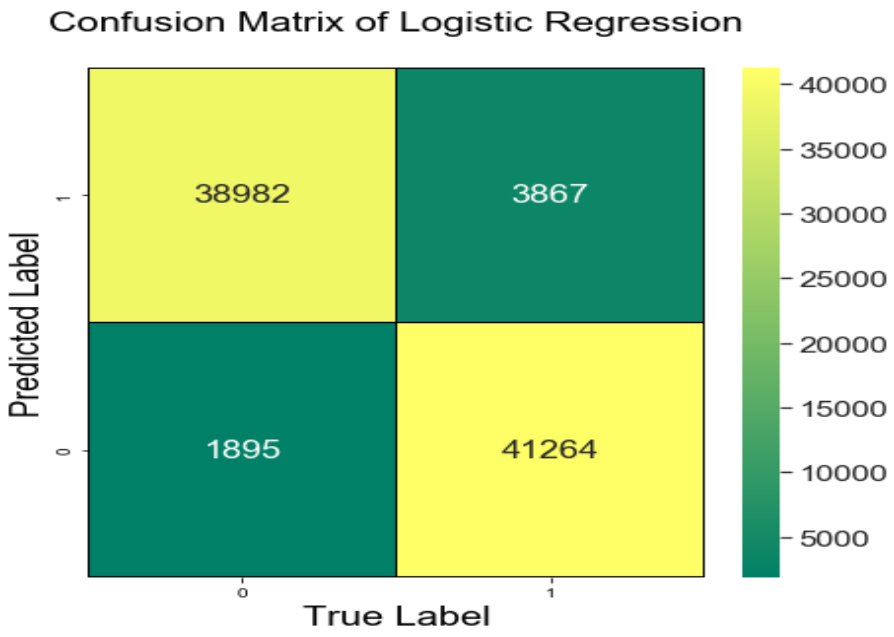
```
print("Accuracy Score is:",accuracy_score(lg_pred,y_test)*100)
print("Log Loss is:",log_loss(lg_pred,y_test))
print("Classification Report is:\n",classification_report(lg_pred,y_test))
print("Confusion Matrix is:\n",confusion_matrix(y_test,lg_pred))
```

```
Accuracy Score is: 93.30062319784206
Log Loss is: 2.3139003909474187
Classification Report is:
                precision    recall  f1-score   support

           0       0.91      0.95      0.93     40877
           1       0.96      0.91      0.93     45131

    accuracy                           0.93     86008
   macro avg       0.93      0.93      0.93     86008
weighted avg       0.93      0.93      0.93     86008

Confusion Matrix is:
 [[38982  3867]
 [ 1895 41264]]
```




```
tpr,fpr,threshold=roc_curve(y_test,lg_pred)
```

```
tpr
```

```
array([0.        , 0.09024715, 1.        ])
```

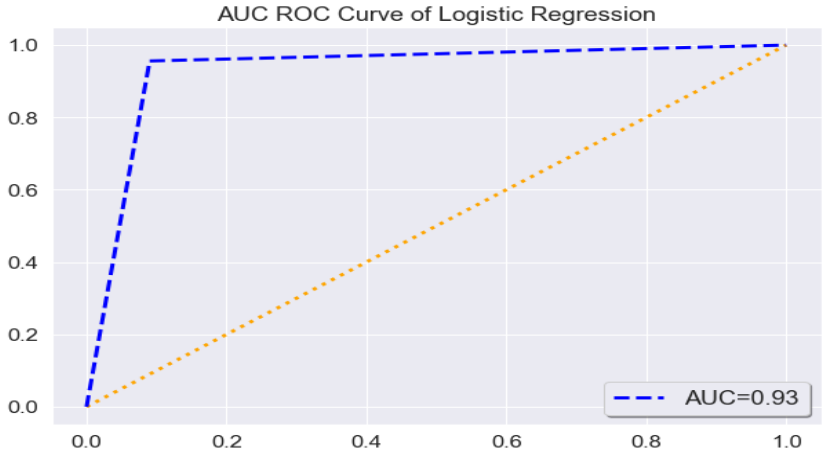
```
fpr
```

```
array([0.        , 0.95609259, 1.        ])
```

```
threshold
```

```
array([2, 1, 0])
```

```
plt.figure(figsize=(10,7))
plt.plot([0,1],[0,1],color='orange',linestyle=":",lw=3)
plt.plot(tpr,fpr,label="AUC=%0.2F"% auc_score,color='blue',linestyle="--",lw=3)
plt.legend(fancybox=True,shadow=True,fontsize='medium')
plt.title("AUC ROC Curve of Logistic Regression")
plt.show()
```



In Logistic Regression the accuracy score is 93%. AUC Score 93%. We see the Classification Report and Confusion Matrix chart in the above.

Passive Aggressive Classifier:

Passive Aggressive Classifier is an online learning algorithm where you train a system incrementally by feeding it instances sequentially, individually or in small groups called mini-batches.

```
pac=PassiveAggressiveClassifier()
```

```
pac.fit(x_train,y_train)
```

```
PassiveAggressiveClassifier()
```

```
pac_pred=pac.predict(x_test)
print(pac_pred)
```

```
[0 1 0 ... 0 1 0]
```

```
print("Accuracy score is:",accuracy_score(y_test,pac_pred)*100)
print("Log Loss is:",log_loss(y_test,pac_pred))
print("Classification report:\n",classification_report(y_test,pac_pred))
print("Confusion Matrix:\n",confusion_matrix(y_test,pac_pred))
```

```
Accuracy score is: 95.70737605804112
Log Loss is: 1.482652026019652
Classification report:
              precision    recall  f1-score   support

           0       0.99      0.92      0.96     42849
           1       0.93      0.99      0.96     43159

    accuracy                           0.96     86008
   macro avg       0.96      0.96      0.96     86008
weighted avg       0.96      0.96      0.96     86008

Confusion Matrix:
 [[39381  3468]
 [  224 42935]]
```

Passive Aggressive Classifier gives a good accuracy score 96%.

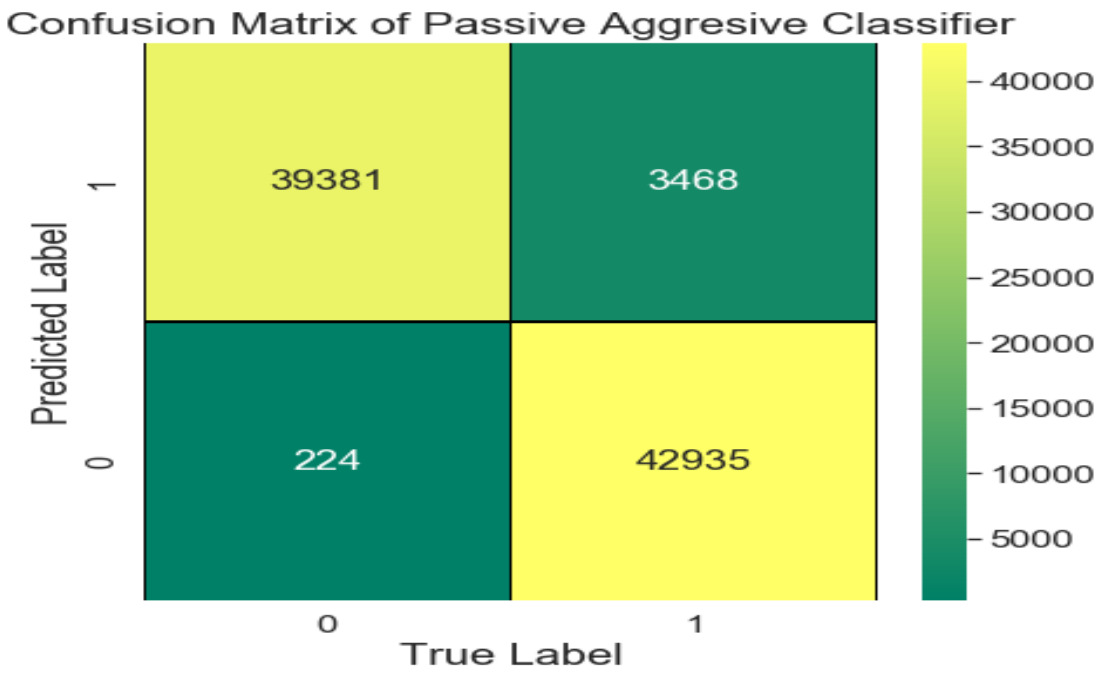
```
CM=confusion_matrix(y_test,pac_pred)

x_axis=['0','1']
y_axis=['1','0']

figure, axes=plt.subplots(figsize=(7,7))

sns.set(font_scale=1.5)

sns.heatmap(CM, axes=axes, annot=True, cmap='summer', xticklabels=x_axis,yticklabels=y_axis,fmt='0.0f',linecolor='black',linewidt
plt.xlabel("True Label",fontsize=20)
plt.ylabel("Predicted Label",fontsize=20)
plt.title("Confusion Matrix of Passive Aggresive Classifier",fontsize=20)
plt.show()
```



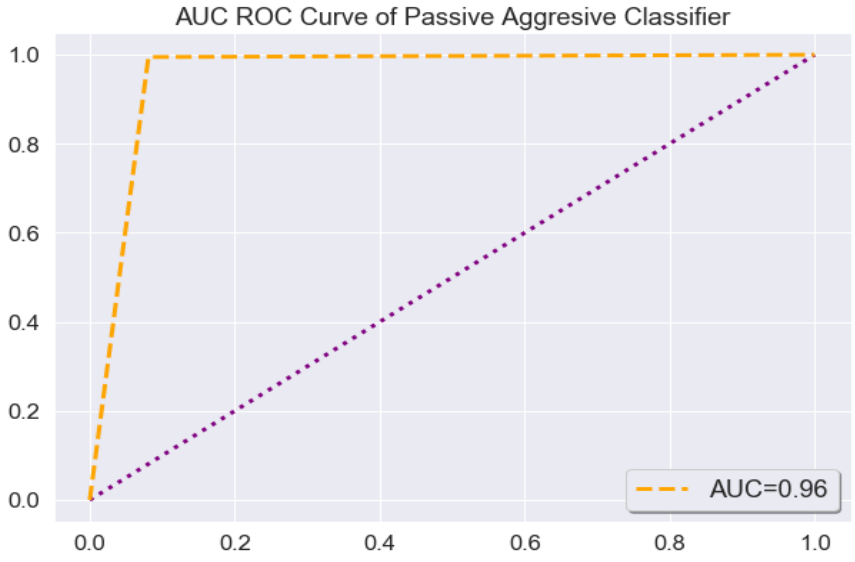

```
auc_score=roc_auc_score(y_test,pac_pred)
print("The value of AUC Score is:",auc_score)
```

```
The value of AUC Score is: 0.9569372556466351
```

```
tpr,fpr,threshold=roc_curve(y_test,pac_pred)
```

```
plt.figure(figsize=(10,7))
plt.plot([0,1],[0,1],color='purple',linestyle=":",lw=3)
plt.plot(tpr,fpr,label="AUC=%0.2F"% auc_score,color='orange',linestyle="--",lw=3)
plt.legend(fancybox=True,shadow=True,fontsize='medium')
plt.title("AUC ROC Curve of Passive Aggresive Classifier")
plt.show()
```

The AUC Score is 95.6%



Multinomial NB:

```
mnb=MultinomialNB()
```

```
mnb.fit(x_train,y_train)
```

```
MultinomialNB()
```

```
mnb_pred=mnb.predict(x_test)
mnb_pred
```

```
array([0, 1, 0, ..., 0, 1, 0])
```

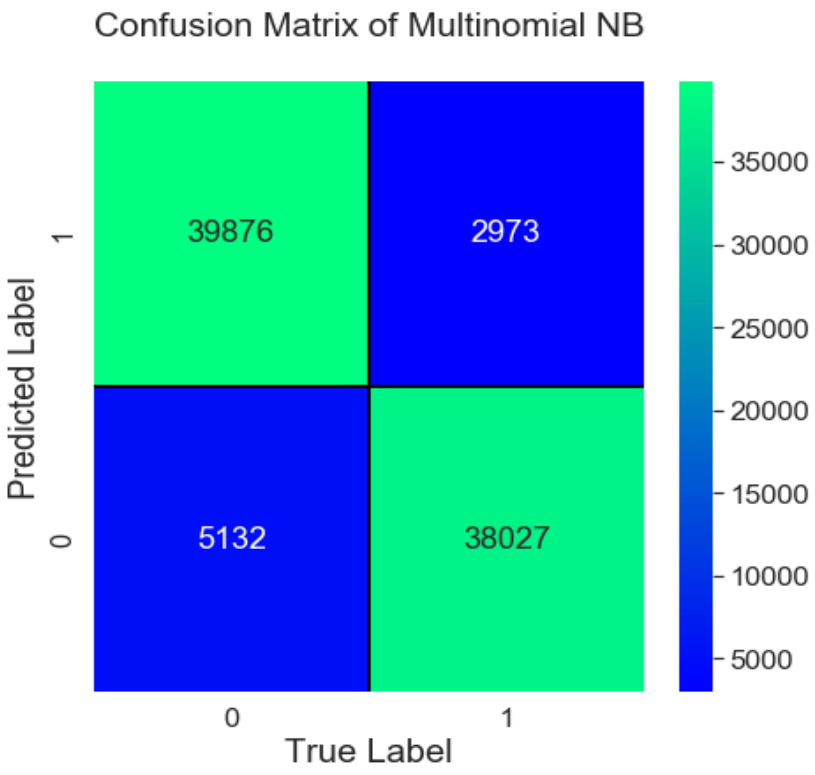
```
print("Accuracy Score is:",accuracy_score(y_test,mnb_pred)*100)
print("Log Loss is:",log_loss(y_test,mnb_pred))
print("Classification Report is:\n",classification_report(y_test,mnb_pred))
print("Confusion Matrix is:\n",confusion_matrix(y_test,mnb_pred))
```

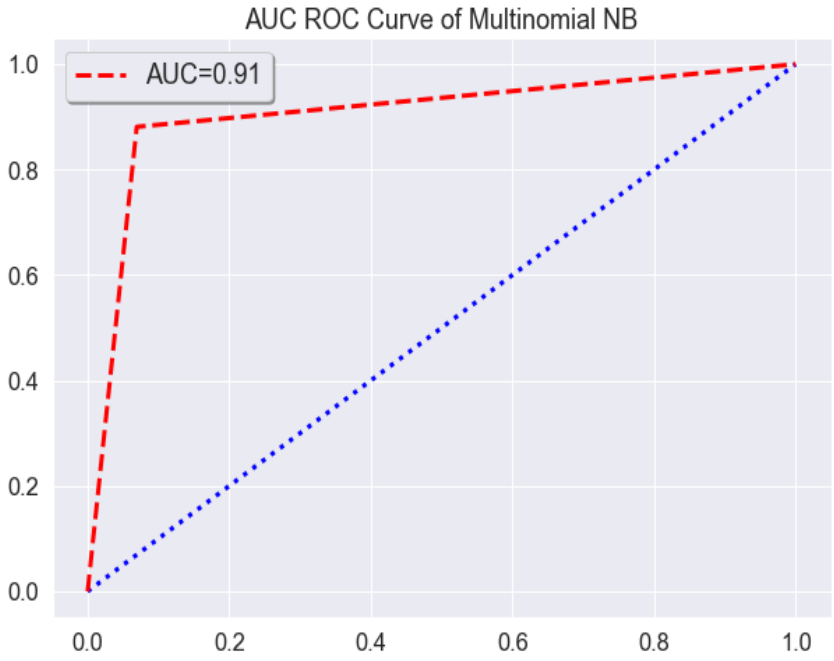
```
Accuracy Score is: 90.57645800390661
Log Loss is: 3.2548037378373067
Classification Report is:
               precision    recall  f1-score   support

           0       0.89      0.93      0.91     42849
           1       0.93      0.88      0.90     43159

    accuracy                           0.91     86008
   macro avg       0.91      0.91      0.91     86008
weighted avg       0.91      0.91      0.91     86008

Confusion Matrix is:
 [[39876  2973]
 [ 5132 38027]]
```





In Multinomial NB the accuracy score is 91%. AUC Score 91%. We see the Classification Report and Confusion Matrix chart in the above.

Complement NB:

In **complement Naive Bayes**, instead of calculating the probability of an item belonging to a certain class, we calculate the probability of the item belonging to all the classes. This is the literal meaning of the word, complement and hence is called Complement Naive Bayes.

```
cnb=ComplementNB()
```

```
cnb.fit(x_train,y_train)
```

```
ComplementNB()
```

```
cnb_pred=cnb.predict(x_test)
cnb_pred
```

```
array([0, 1, 0, ..., 0, 1, 0])
```

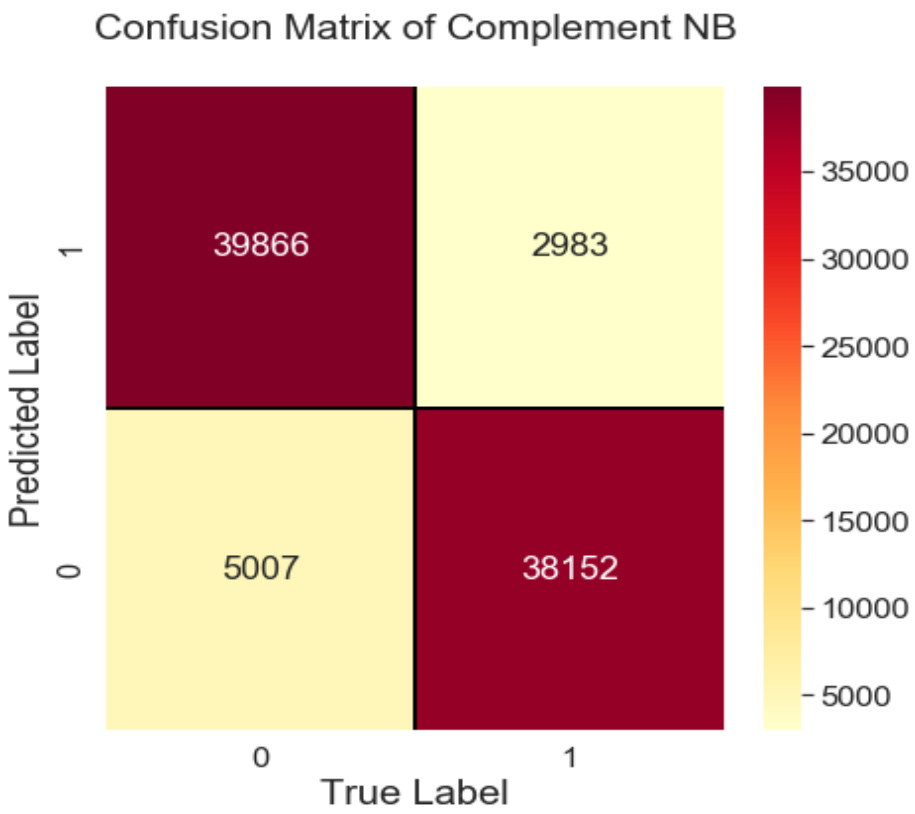
```
print("Accuracy Score is:",accuracy_score(y_test,cnb_pred)*100)
print("Log Loss is:",log_loss(y_test,cnb_pred))
print("Classification Report is:\n",classification_report(y_test,cnb_pred))
print("Confusion Matrix is:\n",confusion_matrix(y_test,cnb_pred))
```

```
Accuracy Score is: 90.7101664961399
Log Loss is: 3.208622553651645
Classification Report is:
               precision    recall  f1-score   support

           0       0.89      0.93      0.91     42849
           1       0.93      0.88      0.91     43159

    accuracy                           0.91     86008
   macro avg       0.91      0.91      0.91     86008
weighted avg       0.91      0.91      0.91     86008

Confusion Matrix is:
 [[39866  2983]
 [ 5007 38152]]
```

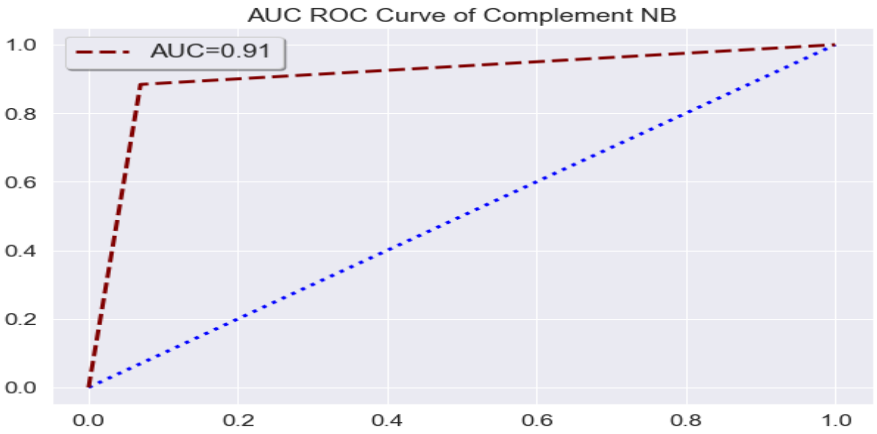



```
auc_score=roc_auc_score(y_test,cnb_pred)
print("The value of AUC Score:", auc_score)
```

```
The value of AUC Score: 0.9071852784616333
```

```
tpr,fpr,threshold=roc_curve(y_test,cnb_pred)
```

```
plt.figure(figsize=(10,7))
plt.plot([0,1],[0,1],color='blue',linestyle=":",lw=3)
plt.plot(tpr,fpr,label="AUC=%0.2F"% auc_score,color='maroon',linestyle="--",lw=3)
plt.legend(fancybox=True,shadow=True,fontsize='medium')
plt.title("AUC ROC Curve of Complement NB")
plt.show()
```



In Complement NB the accuracy score is 91%. AUC Score 91%. We see the Classification Report and Confusion Matrix chart in the above.

- **Key Metrics for success in solving problem under consideration**

In this project I have used Log Loss metrics, confusion matrix, classification report, accuracy score, roc curve, auc_roc score.

The key metrics that were mainly taken into consideration were the followings:

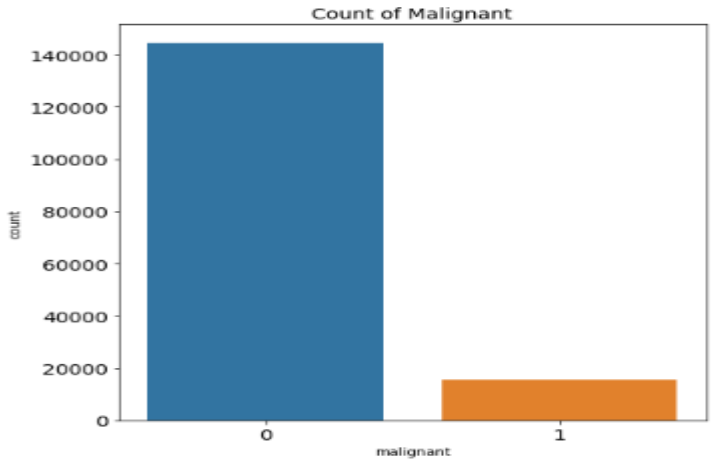
- Comments_text
- malignant
- highly_malignant
- rude
- threat
- abuse
- loathe

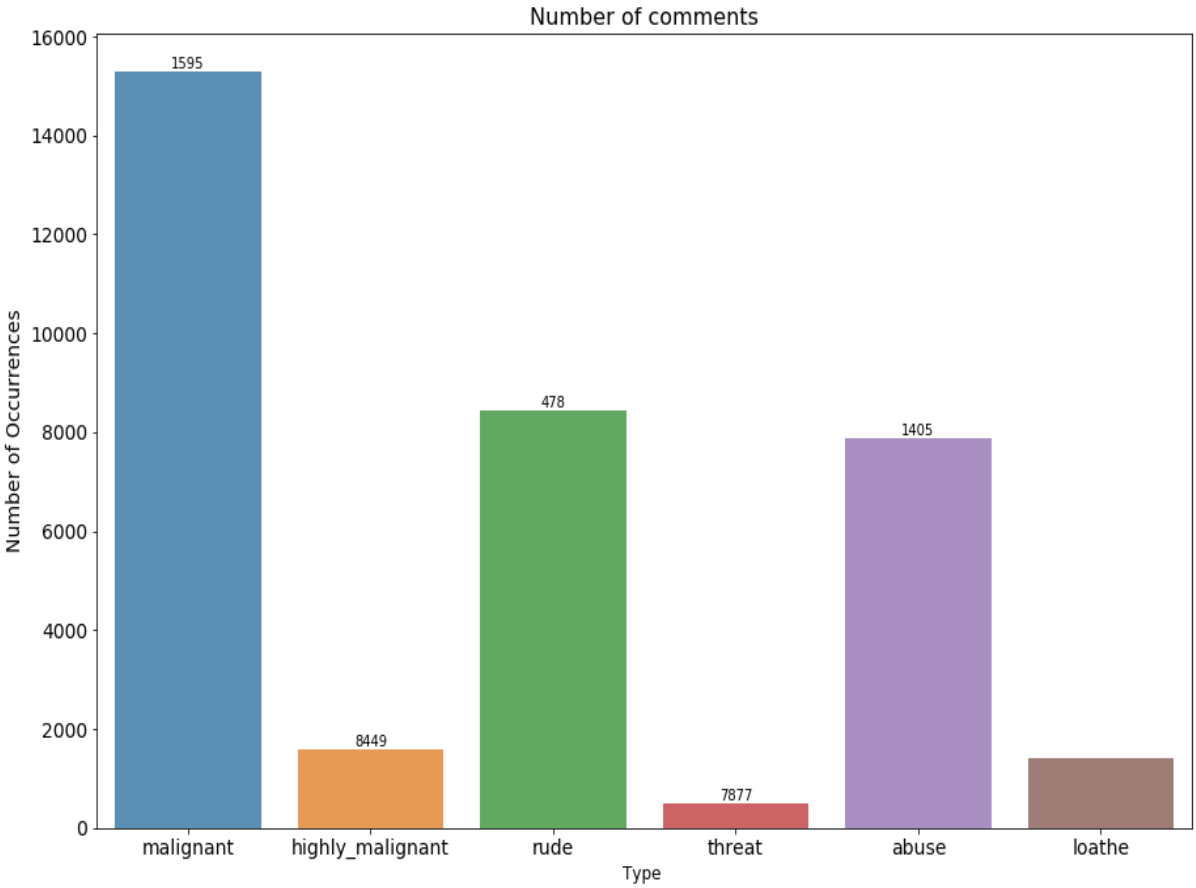
- **Visualizations**

  Using the countplot checked the count for Malignant how many bad comments and good comments are present.

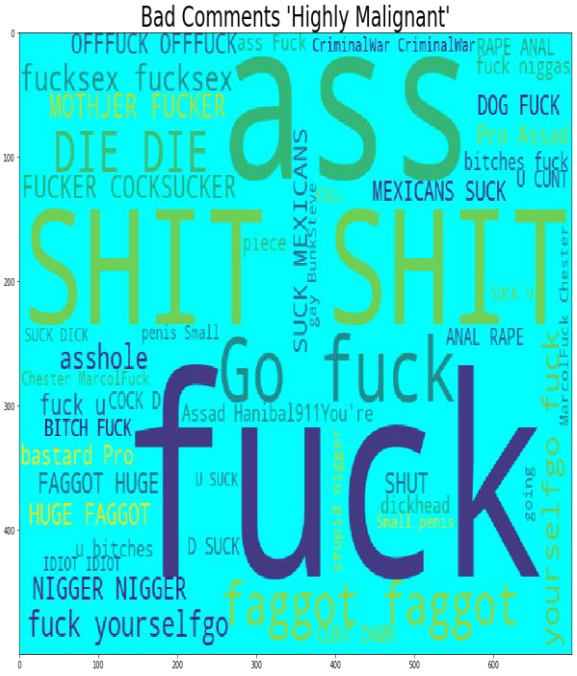
```
: #Checking the count of Malignant.
  plt.figure(figsize=(7,7))
  print(df_train['malignant'].value_counts())
  sns.countplot(df_train['malignant'])
  plt.xticks(fontsize=14)
  plt.yticks(fontsize=14)
  plt.title('Count of Malignant',fontsize=14)
  plt.show()
```

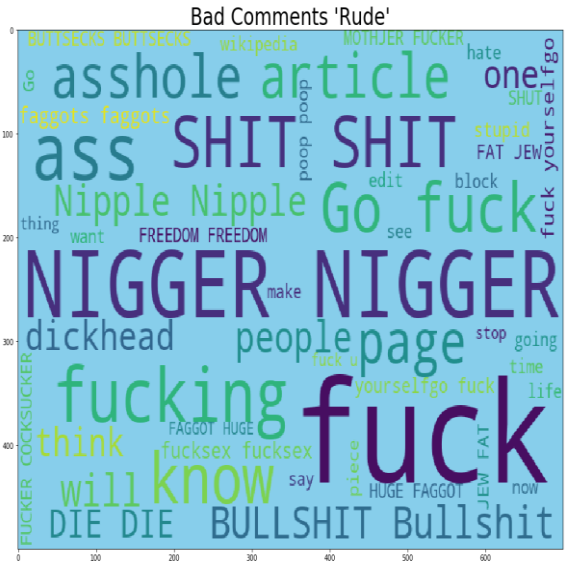
```
0    144277
1     15294
Name: malignant, dtype: int64
```



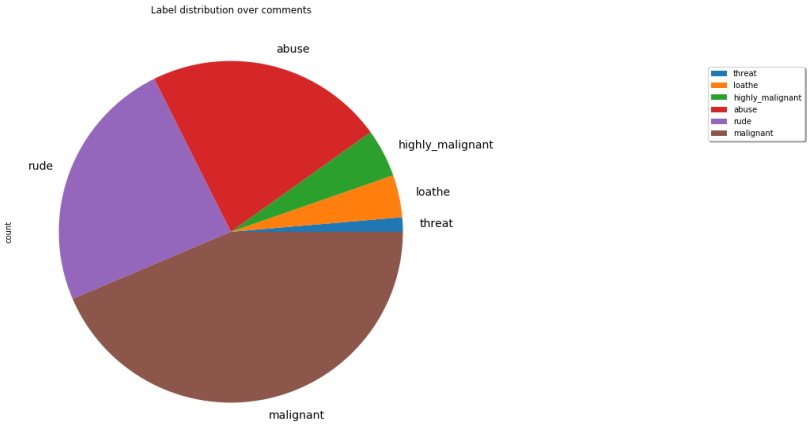


In the above barplot, we see the count of each type of comment that are bad. In Malignant the count is high as compare to other classes.

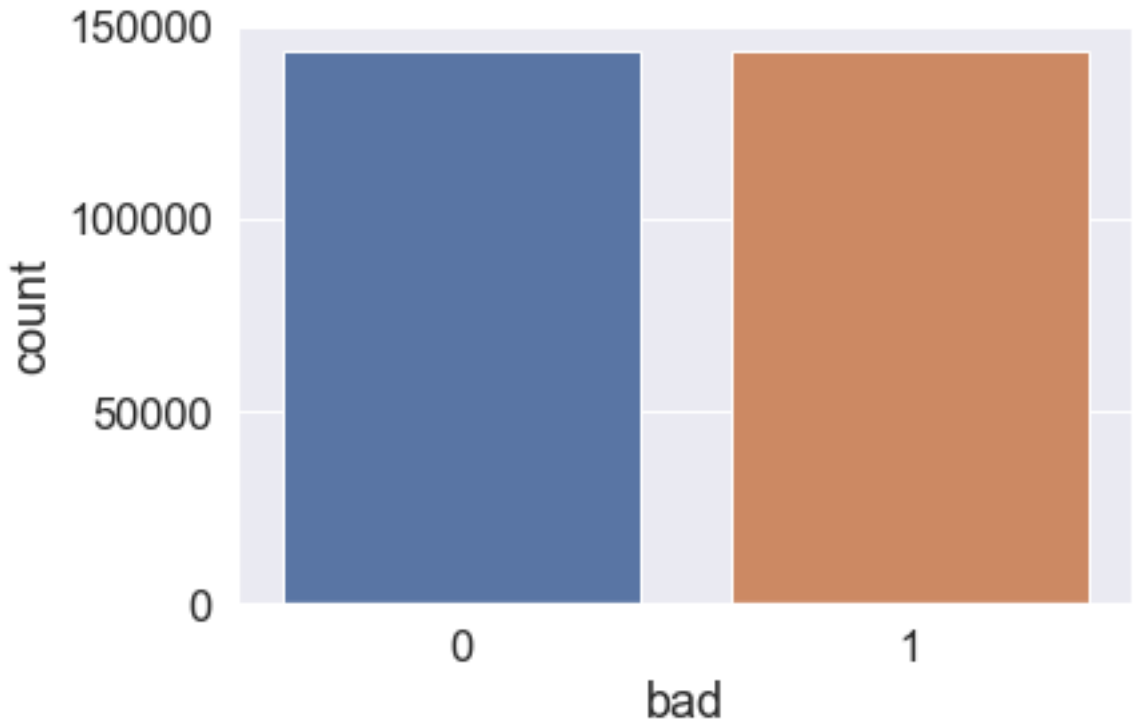




This are the frequently used comments in each class. Using the word cloud, we have visualized the frequently used comments.



In the pie chart we clearly see that Malignant count is high, as compare to other classes. Majority of person comments comes under the Malignant type category.



I have made all the good comments to 0 and all the bad comments to 1. Using the SMOTE Imbalance technique. I have balanced the target dataset.

- **Interpretation of the Results**

  After done the hyper parameter tuning Passive Aggressive Classifier gives a 96% score.

```
params={
        'n_features':[4,7,10],
       'random_state':[79,85,99]}
```

```
from sklearn.model_selection import GridSearchCV
```

```
model=PassiveAggressiveClassifier(fit_intercept=True, max_iter=500, tol=0.001, early_stopping=False, validation_fraction=0.1, n_
```

```
model.fit(x_train,y_train)
```

```
-- Epoch 1
Norm: 174.39, NNZs: 65388, Bias: 0.993626, T: 200506, Avg. loss: 0.209509
Total training time: 0.11 seconds.
-- Epoch 2
Norm: 222.26, NNZs: 67735, Bias: 1.006662, T: 401012, Avg. loss: 0.113731
Total training time: 0.24 seconds.
-- Epoch 3
Norm: 253.71, NNZs: 68595, Bias: 1.050360, T: 601518, Avg. loss: 0.086272
Total training time: 0.33 seconds.
-- Epoch 4
Norm: 277.91, NNZs: 69043, Bias: 1.093963, T: 802024, Avg. loss: 0.071542
Total training time: 0.43 seconds.
-- Epoch 5
Norm: 297.57, NNZs: 69373, Bias: 1.020929, T: 1002530, Avg. loss: 0.062509
Total training time: 0.51 seconds.
-- Epoch 6
Norm: 314.26, NNZs: 69555, Bias: 1.032661, T: 1203036, Avg. loss: 0.056223
Total training time: 0.62 seconds.
-- Epoch 7
```

```
model_pred=model.predict(x_test)
model_pred
```

```
array([0, 1, 0, ..., 0, 1, 0])
```

```
model_pred=model.predict(x_test)
model_pred
```

```
array([0, 1, 0, ..., 0, 1, 0])
```

```
print("Accuracy score is:",accuracy_score(y_test,model_pred)*100)
print("Log Loss is:",log_loss(y_test,model_pred))
print("Classification report:\n",classification_report(y_test,model_pred))
print("Confusion Matrix:\n",confusion_matrix(y_test,model_pred))
```

```
Accuracy score is: 95.75969677239327
Log Loss is: 1.4645803756293032
Classification report:
              precision    recall  f1-score   support

           0       0.99      0.92      0.96     42849
           1       0.93      0.99      0.96     43159

    accuracy                           0.96     86008
   macro avg       0.96      0.96      0.96     86008
weighted avg       0.96      0.96      0.96     86008

Confusion Matrix:
 [[39458  3391]
 [  256 42903]]
```

```
model.score(x_train,y_train)
```

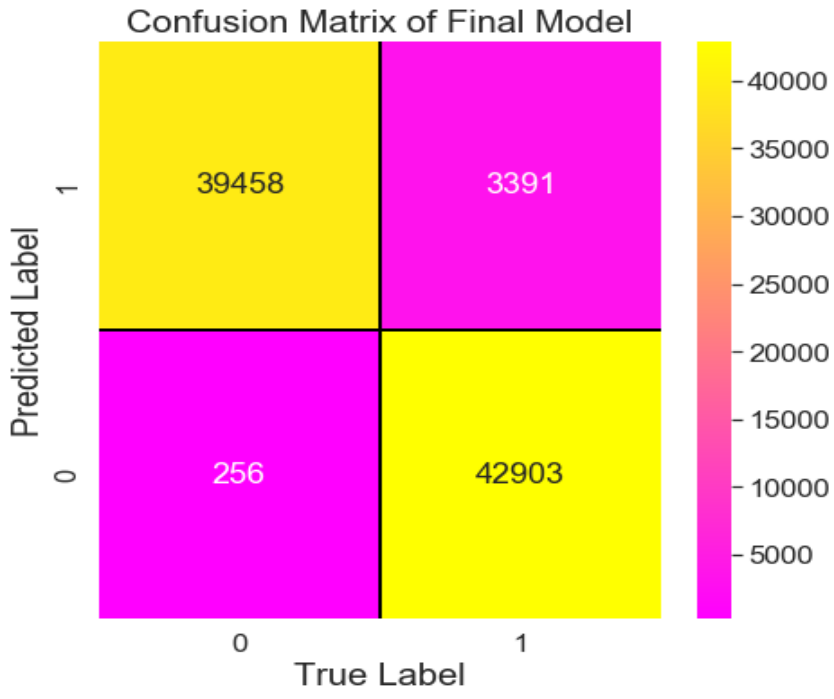
```
0.993686591855853
```

```
model.intercept_
```

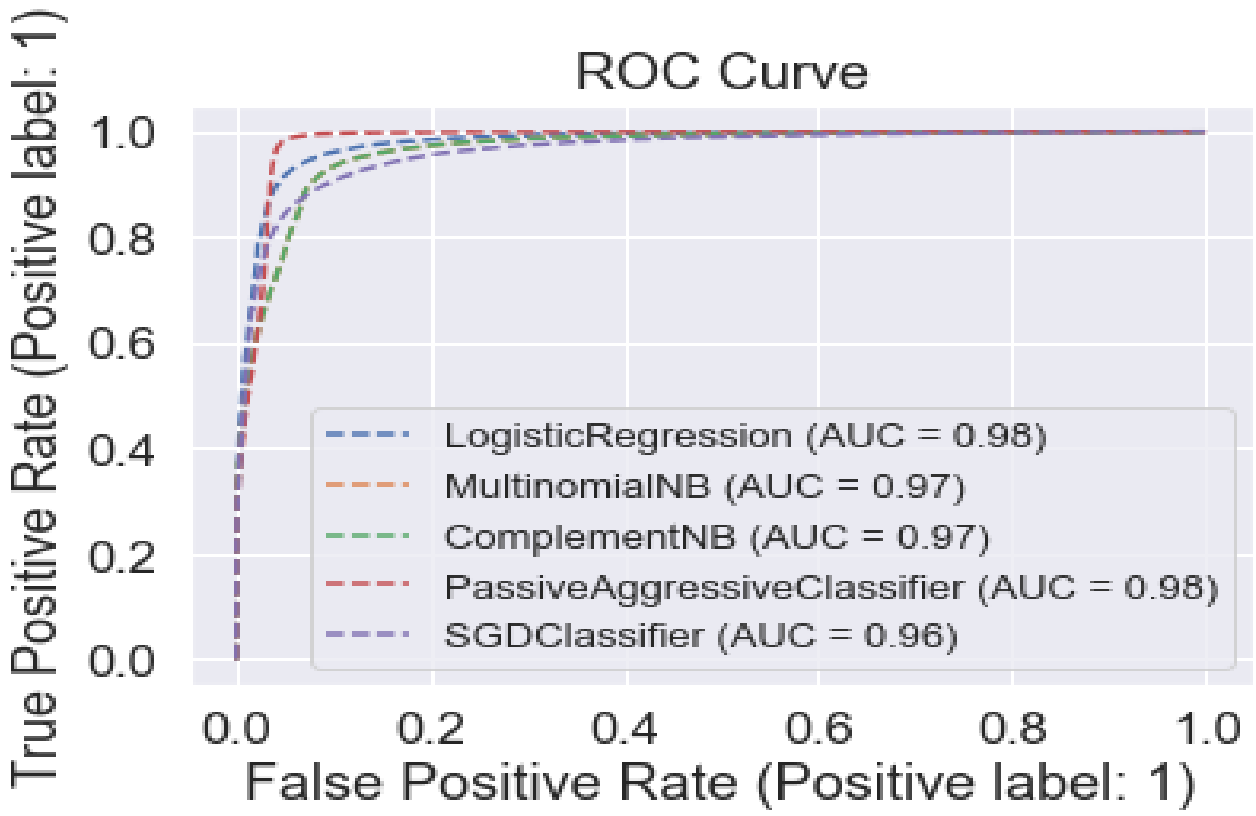
```
array([1.02013802])
```

```
model.coef_
```

```
array([[-2.69354523,  8.864979  , -0.12029241, ...,  1.01768785,
        -0.06495712, -2.07862774]])
```



This is the confusion matrix of our final model.



This the ROC Curve of various model used in this project. We see that Passive Aggressive Classifier gives 98% AUC Score.

Saving the Model:

```
import pickle
```

```
filename='Malignant-comments_classifier.pickle'
```

```
pickle.dump(model,open(filename,'wb'))
```

```
loaded_model=pickle.load(open(filename,'rb'))
```

```
loaded_model_pred=loaded_model.predict(x_test)
loaded_model_pred
```

```
array([0, 1, 0, ..., 0, 1, 0])
```

```
loaded_model.score(x_train,y_train)
```

```
0.993686591855853
```

```
df=pd.DataFrame([loaded_model.predict(x_test)[:],y_test[:]],index=['Predicted','Actual'])
df
```

	0	1	2	3	4	5	6	7	8	9	...	85998	85999	86000	86001	86002	86003	86004	86005	86006	86007
Predicted	0	1	0	1	0	1	1	1	1	0	...	0	0	0	1	0	0	1	0	1	0
Actual	0	1	0	1	0	1	1	1	1	0	...	0	0	0	1	0	0	1	0	1	0

2 rows × 86008 columns

We have saved our final model using the pickle. We have done the prediction from the final model and saved the predicted data in the csv format in the future use.

CONCLUSION

• Key Findings and Conclusions of the Study

- On visualization of countplot for Malignant, Highly Malignant, Abuse, Threat, Loathe most of them use the abuse language in the comments. We have visualized the bad comments in the word cloud. In the comments there are lots of stopwords, punctuation etc.

- I have cleaned the data and make the prediction. In the social media people shows the hatred in the comment. With this analysis and prediction, we came to conclude which comments is good and bad.

- So, that in future we can restrict those comments and spread peace, positive vibes in the social media platform.

- These malignant comments classification can be used by social media companies to filter and classify some keywords as highly malignant and set their own policy going forward for the customers and other shareholders.

• Learning Outcomes of the Study in respect of Data Science

- While working on this project, I have learned how to handle the multi class feature problem, how to cleaned the dataset, various visualization plot. Usage of nltk tool for Work Cloud.

- I have also used NLTK library to clean the text/comments and find out the actual length of the comments that can be used for further evaluation.

- Learned different machine learning algorithm and their usage. In respect to Data Science, it helps in all sector to analysis and visualized the data. Make a prediction that help in the future to increase their business many times more as that are now. I help

Businessman, Stakeholder, IT Sector any kind of business it is Data Science comes into play to help and increase the sales.

- **Limitations of this work and Scope for Future Work**

- The dataset is huge, it takes long time to visualized the data, run the model and do the hyper parameter tuning of final model. As it is multi class problem we can explore more.

- As this is the initial stage, we can explore data further. With our prediction we can filter those comments in the social media to stop influence the negative behaviour spread in the society.

- It is very exciting to work on this project, using various attractive visualization plot used. We can finally conclude that we should always encourage and give positive vibes in the social media instead of spreading the negative vibes.