NAME OF THE PROJECT

Malignant-Comments-Classifier

SUBMITTED BY:

AKANKSHA MISHRA

FLIPROBO SME:

MS. KHUSHBOO GaRG

ACKNOWLEDGMENT

I would like to express my special gratitude to "Flip Robo" team, who has given me this opportunity to deal with a beautiful dataset and it has helped me to improve my analyzation skills. And I want to express my huge gratitude to Ms. Khushboo Garg (SME Flip Robo), she is the person who has helped me to get out of all the difficulties I faced while doing the project.

A huge thanks to "Data trained" who are the reason behind my Internship at Fliprobo. Last but not least my parents who have been my backbone in every step of my life.

References use in this project:

1. SCIKIT Learn Library Documentation
2. Blogs from towardsdatascience, Analytics Vidya, Medium
3. Andrew Ng Notes on Machine Learning (GitHub)
4. Data Science Projects with Python Second Edition by Packt
5. Hands on Machine learning with scikit learn and tensor flow by Aurelien Geron
6. Stackoverflow.com to resolve some project related queries.
7. Predicting Credit Default among Micro Borrowers in Ghana Kwame Simpe Ofori, Eli Fianu
8. Predicting Microfinance Credit Default: A Study of Nsoatreman Rural Bank, Ghana Ernest Yeboah Boateng
9. A Machine Learning Approach for Micro-Credit Scoring Apostolos Ampountolas

And also thank you for many other persons who has helped me directly or indirectly to complete the project.

Chap 1. Introduction

Problem Statement:

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.

Data Set Description:

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.

Analytical Problem Framing

1. Mathematical / Analytical Modelling of the Problem

Whenever we employ any ML algorithm, statistical models or feature pre-processing in background lot of mathematical framework work. In this project we have done lot of data pre-processing & ML model building. In this section we dive into mathematical background of some of these algorithms.

1. Logistic Regression

The response variable, label, is a binary variable (whether the loan was repaid or not). Therefore, the logistic regression is a suitable technique to use because it is developed to predict a binary dependent variable as a function of the predictor variables. The logit, in this model, is the likelihood ratio that the dependent variable, non-defaulter, is one (1) as opposed to zero (0), defaulter. The probability, P, of credit default is given by;

$$\ln\left[\frac{P(Y)}{1-P(Y)}\right] = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \ldots + \beta_k X_k$$

Where;

$ln\left[\frac{P(Y)}{1-P(Y)}\right]$ is the log (odds) of credit default

Y is the dichotomous outcome which represents credit default (whether the loan was repaid or not) $X_1, X_2, \ldots, X_K$ are the predictor variables which are as educational level, number of dependents, type of loan, adequacy of the loan facility, duration for repayment of loan, number of years in business, cost of capital and period within the year the loan was advanced to the client $\beta_o, \beta_1, \beta_2, \ldots, \beta_k$ are the regression (model) coefficients

2. Data Sources and their formats

The data set comes from my internship company – Fliprobo technologies in excel format.

```
# Importing dataset CSV file using pandas
df= pd.read_csv('Data file.csv')
```

```
print('No. of Rows :',df.shape[0])
print('No. of Columns :',df.shape[1])
pd.set_option('display.max_columns',None) # # This will enable us to see truncated columns
df.head()
```

```
No. of Rows : 209593
No. of Columns : 37
```

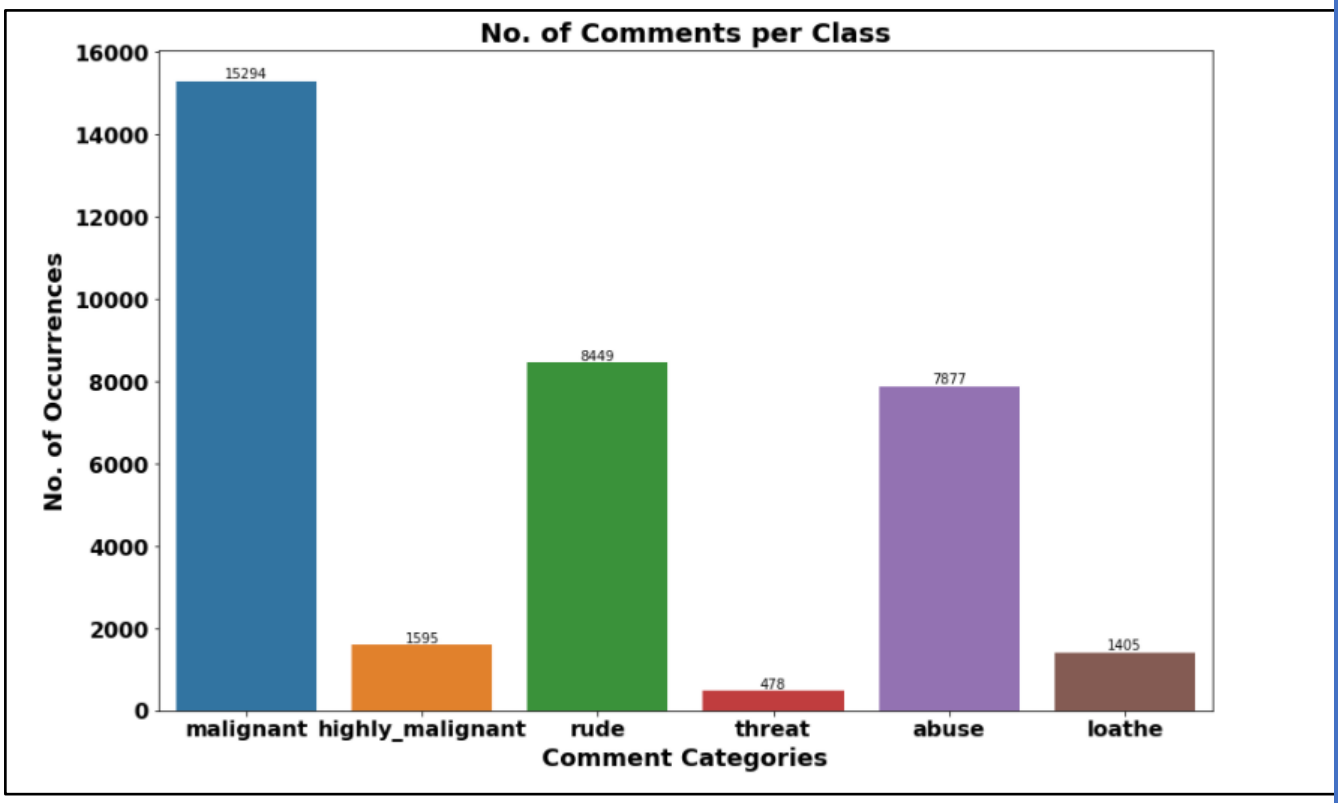
Malignant Commentes Classifier - Multi Label Classification

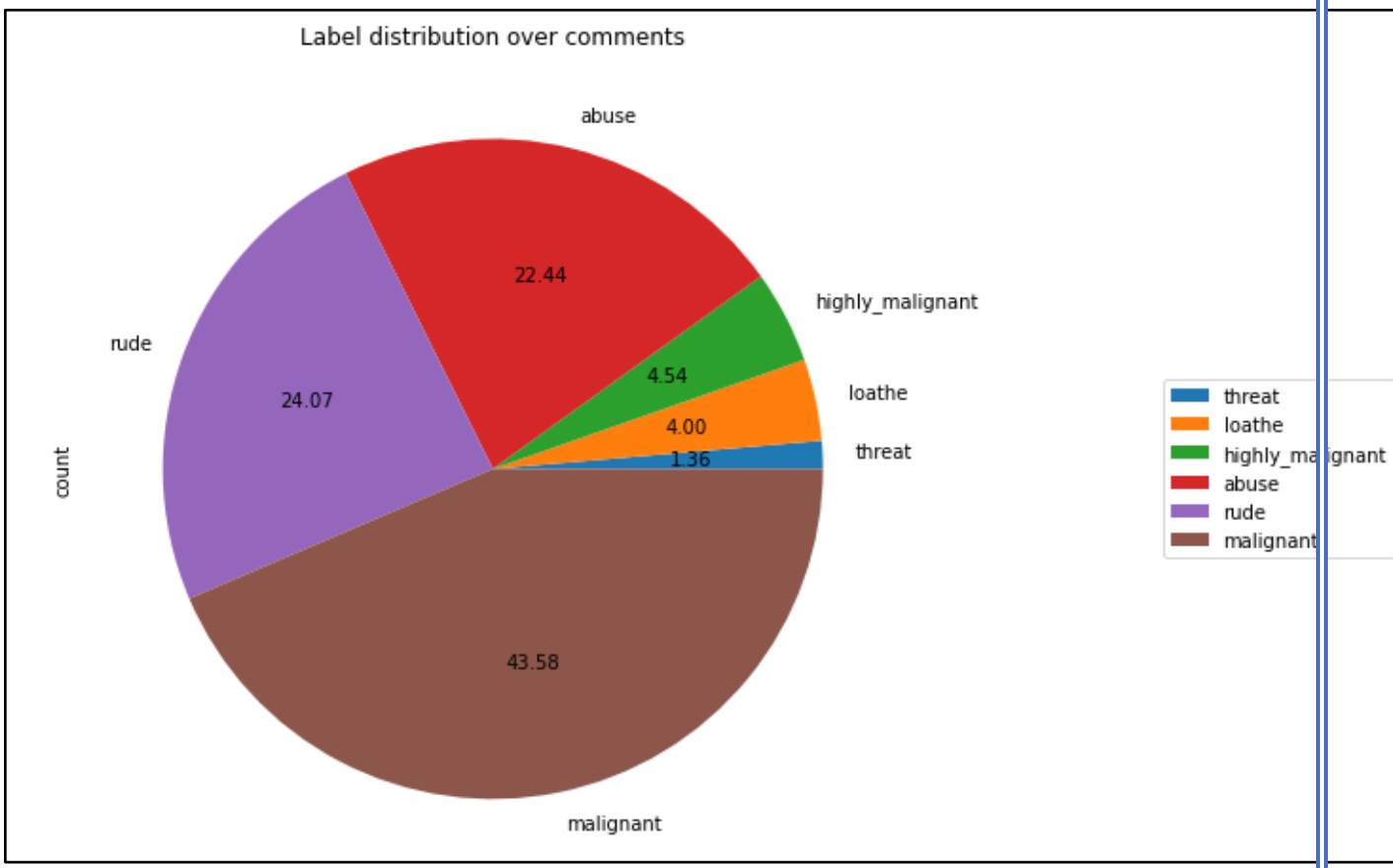
- There has been a remarkable increase in the cases of cyberbullying and trolls on various social media platforms. Many celebrities and influences are facing backlashes from people and have to come across hateful and offensive comments. This can take a toll on anyone and affect them mentally leading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

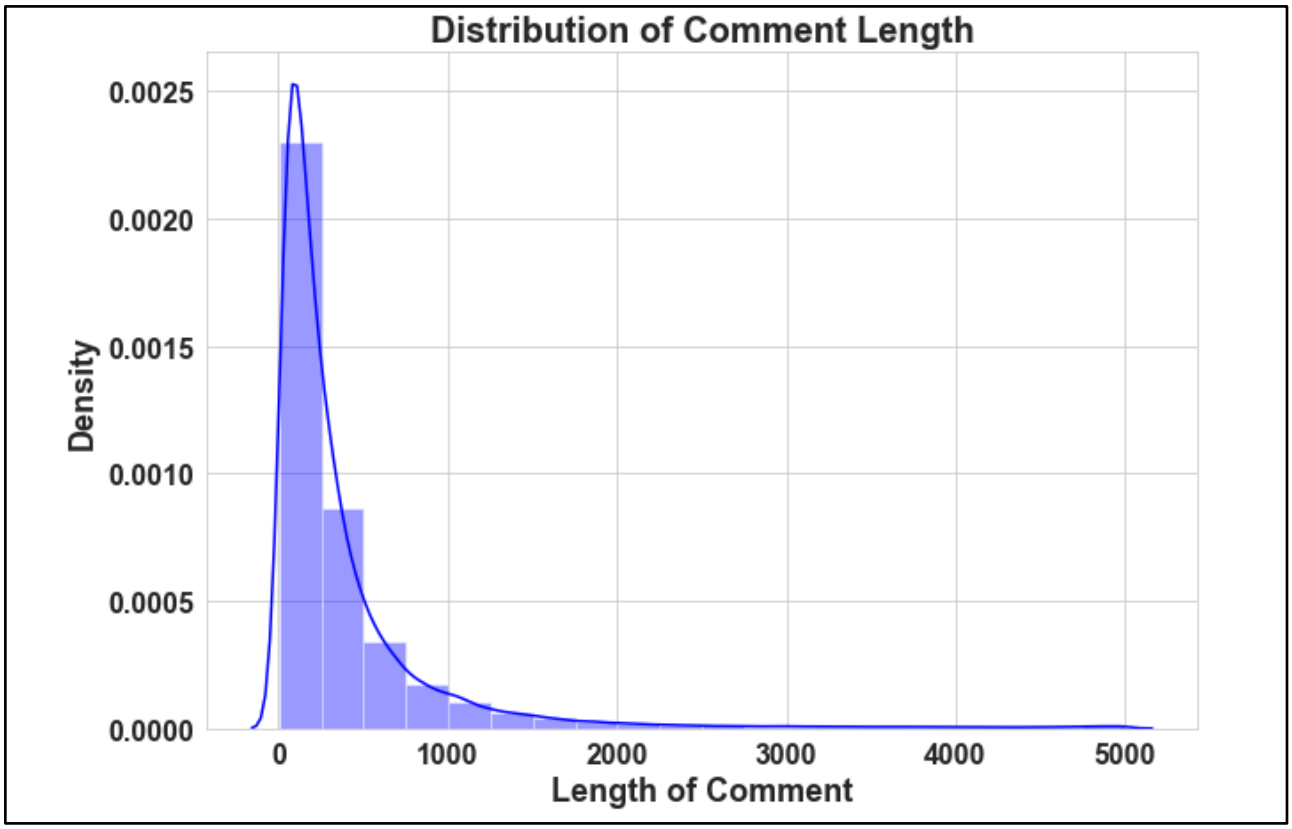
is clearly offensive.

- **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.**
-

Exploration of Target Variable Ratings :-


No. of Comments per Class
No. of Occurrences
Comment Categories
malignant
highly_malignant
rude
threat
abuse
loathe
15294
1595
8449
478
7877
1405
0
2000
4000
6000
8000
10000
12000
14000
16000


Label distribution over comments
abuse
highly_malignant
rude
loathe
threat
malignant
22.44
4.54
24.07
4.00
1.36
43.58
count
threat
loathe
highly_malignant
abuse
rude
malignant



Data Pre Processing

- Convert the text to lowercase
- Remove the punctuations, digits and special characters
- Tokenize the text, filter out the adjectives used in the review and create a new column in data frame
- Remove the stop words
- Stemming and Lemmatising

- Applying Text Vectorization to convert text into numeric

Multi-Label Classification Techniques

- **One Vs Rest**
- **Binary Relevance**
- **Classifier Chains**
- **Label Powerset**
- **Adapted Algorithm**

Word Cloud:-

- Word cloud is a visualization technique for text data wherein each word is picturized with its importance in the context or its frequency.

- THE MORE COMMONLY THE TERM APPEARS WITHIN THE TEXT BEING ANALYSED, THE LARGER THE WORD APPEARS IN THE IMAGE GENERATED.
- THE ENLARGED TEXTS ARE THE GREATEST NUMBER OF WORDS USED THERE AND SMALL TEXTS ARE THE SMALLER NUMBER OF WORDS USE.


WORDS TAGGED AS HIGHLY MALIGNANT
asshole
motherfucking
pathetic
stupid
dtype
u
shalom
blocking
hi
de
antonio
mercado
Length
kill
lol
co
know
walter
stuff
shit
around
stop
crow
cocksucker
work
object
semite
crap
moron
cunt
woman
peace
frankly
son
fuck
im
fuckhead
better
deleting
jimbo
feel
quite
wi
Name
good
stink
piss
back
fucker
cc
stinking
gay
comment_text
wale
fucking
dirty
bitch
rootmyass
never
mother
piece
filthy
leave


WORDS TAGGED AS RUDE
fucking
shit
piece
white
stuff
absurd
edits
gay
fucked
naked
fuckeeed
mother
great
libe
object
lolwat
got
conversation
stupid
fuck
dry
tiger
work
put
fat
meow
previous
eating
going
thought
shark
around
cant
posting
faggot
peace
obese
gpiss
deleting
keep
asshole
stop
sit
dtype
cocksucker
drink
archangel
filthy
deleted
comment_text
Length
Name
antisemitian

Visualization & Data Wrangling Library used

```
#Importing warning library to avoid any warnings
import pandas as pd # for data wrangling purpose
import numpy as np # Basic computation library
import seaborn as sns # For Visualization
import matplotlib.pyplot as plt # ploting package
%matplotlib inline
import warnings # Filtering warnings
warnings.filterwarnings('ignore')
```

Text Mining Library used

```
#Importing required libraries
import re
import string
import nltk
from nltk.corpus import stopwords
from nltk.tokenize import word_tokenize
from nltk.stem import SnowballStemmer, WordNetLemmatizer
from sklearn.feature_extraction.text import CountVectorizer, TfidfVectorizer
from wordcloud import WordCloud
```

Machine Learning Model Building Library used

```
#Importing Machine learning Model library
from sklearn.linear_model import LogisticRegression
from sklearn.naive_bayes import MultinomialNB
from sklearn.tree import DecisionTreeClassifier
from sklearn.neighbors import KNeighborsClassifier
from sklearn.ensemble import RandomForestClassifier
from sklearn.ensemble import AdaBoostClassifier
from sklearn.ensemble import GradientBoostingClassifier
from xgboost import XGBClassifier
from skmultilearn.problem_transform import BinaryRelevance
from sklearn.svm import SVC, LinearSVC
from sklearn.multiclass import OneVsRestClassifier
from sklearn.model_selection import train_test_split,cross_val_score
from sklearn.metrics import confusion_matrix,classification_report,accuracy_score
from sklearn.metrics import roc_auc_score, roc_curve, auc
from sklearn.metrics import hamming_loss, log_loss
```

The different classification algorithm used in this project to build ML model are as below:

- ❖ Random Forest classifier
- ❖ Support Vector Classifier
- ❖ Logistics Regression
- ❖ AdaBoost Classifier

Machine Learning Evaluation Matrix

- SUPPORT VECTOR CLASSIFIER GIVES MAXIMUM ACCURACY SCORE: 91.1508 % AND HAMMING LOSS: 2.0953% THAN THE OTHER CLASSIFICATION MODELS.
- HYPER PARAMETER TUNING IS PERFORM OVER THIS BEST MODEL USING BEST PARAM SHOWN BELOW :

```
Out[69]: {'estimator__loss': 'hinge',
          'estimator__multi_class': 'ovr',
          'estimator__penalty': 'l2',
          'estimator__random_state': 42}
```

Final ML Model

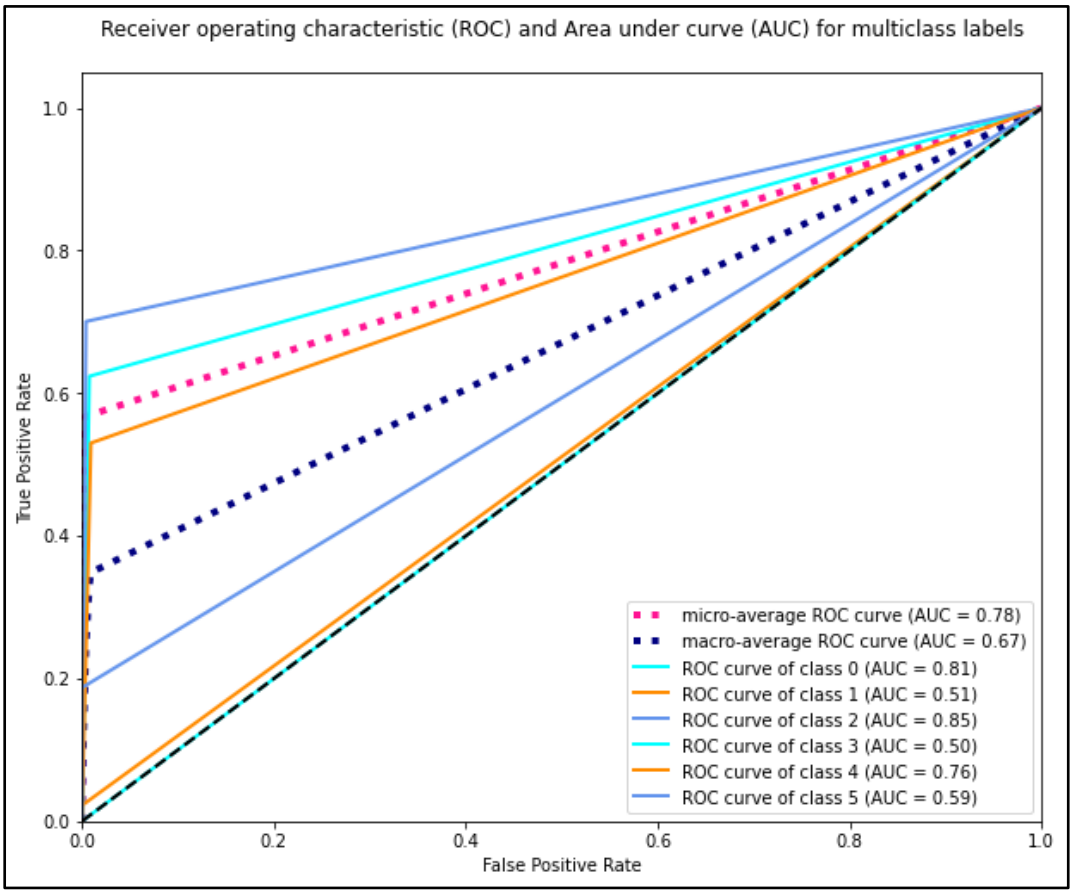
Final Model

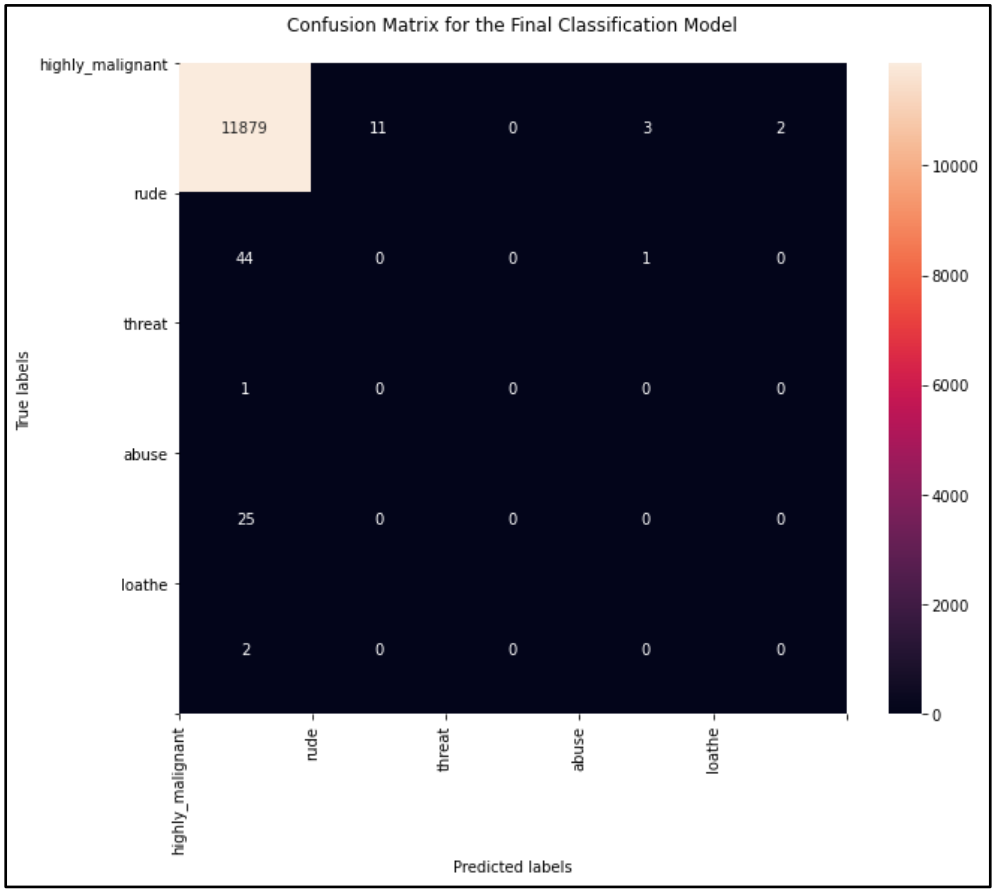
```
Final_Model = OneVsRestClassifier(LinearSVC(loss='hinge',
            multi_class='ovr', penalty='l2', random_state=42))

Classifier = Final_Model.fit(x_train, y_train)
fmod_pred = Final_Model.predict(x_test)
fmod_acc = (accuracy_score(y_test, fmod_pred))*100
print("Accuracy score for the Best Model is:", fmod_acc)
h_loss = hamming_loss(y_test,fmod_pred)*100
print("Hamming loss for the Best Model is:", h_loss)
```

```
Accuracy score for the Best Model is: 91.26002673796792
Hamming loss for the Best Model is: 2.0819407308377897
```

FINAL MODEL IS GIVING US ACCURACY SCORE OF 91.26% WHICH IS SLIGHTLY IMPROVED COMPARE TO EARLIER ACCURACY SCORE OF 91.15%.

AOC-ROC Curve & Confusion Matrix


Receiver operating characteristic (ROC) and Area under curve (AUC) for multiclass labels
True Positive Rate
False Positive Rate
0.0
0.2
0.4
0.6
0.8
1.0
micro-average ROC curve (AUC = 0.78)
macro-average ROC curve (AUC = 0.67)
ROC curve of class 0 (AUC = 0.81)
ROC curve of class 1 (AUC = 0.51)
ROC curve of class 2 (AUC = 0.85)
ROC curve of class 3 (AUC = 0.50)
ROC curve of class 4 (AUC = 0.76)
ROC curve of class 5 (AUC = 0.59)


Confusion Matrix for the Final Classification Model
True labels
Predicted labels
highly_malignant
rude
threat
abuse
loathe
11879
11
0
3
2
44
0
0
1
0
1
0
0
0
0
25
0
0
0
0
2
0
0
0
0
10000
8000
6000
4000
2000
0

Algorithm	Accuracy Score	Recall (Micro)
Logistics Regression	0.9123	0.89
Random Forest Classifier (RFC)	0.9074	0.56
Support Vector Classifier	0.9115	0.56
Ada Boost Classifier	0.9057	0.50

Machine Learning Evaluation Matrix

Learning Outcomes of the Study in respect of Data Science

1. FIRST TIME I HANDLE SUCH HUGE DATASET.
2. FIRST TIME ANY PROJECT I WORKED ON EVER NEED SUCH DATA CLEAN OPERATION. I PAID ATTENTION

REALISTIC & UNREALISTIC DATA, CONSIDERING IT CORRECTIVE MEASURE TAKEN AS PER NEED. THIS WAS BEYOND NORMAL MISSING VALUE IMPUTATION FOR ME.

3. AS DATA WAS HUGE REQUIRE HIGH COMPUTATIONAL CAPACITY, IT MADE ME SWITCH TO GOOGLE COLAB FOR RUNNING MODEL AND FOR HYPERPARAMETER TUNING. I HYPER TUNED FINAL MODEL WITH GOOGLE COLAB GPU.
4. I RUN HYPER PARAMETER TUNING 2-3 TIMES WITH SERVAL PARAMETER. IT WAS TAKING LOT OF TIMES SO AT END I REDUCE HYPERPARAMETER SEARCH PARAMETER AND STILL IT WAS TAKEN 6-7 HR FOR FINDING BEST PARAMETER.

Limitations of this work and Scope for Future Work

1. LIMITED COMPUTATIONAL

RESOURCES PUT LIMITATION ON OPTIMIZATION THROUGH HYPER PARAMETER TUNING. ACCURACY OF MODEL CAN INCREASE WITH HYPERPARAMETER TUNING WITH SEVERAL DIFFERENT PARAMETER. HERE WE USE ONLY TWO PARAMETERS FOR TUNING.