

**Malignant Comment Classifier**



**Submitted by:**

**DEEPTHI MALLU**

**ACKNOWLEDGMENT**

I take this opportunity to acknowledge everyone who have helped me in every stage of this project.

Firstly, I am indebtedly grateful to my SME MR. Keshav Bansal sir, who helped me from beginning of my Project. Am also thankful to my Mentor Shankar Gowda Sir and my whole Data Trained team, where I have learnt Analysing the datasets and building the models using Machine learning and making the projects. Finally, am so thankful to my Flip Robo Technologies team, as they provided me the opportunity to work as intern in their company.

I feel pleasure, to make project report on “Malignant comment classifier”. It has been my privilege to have a team of project guide who have assisted me from the commencement of this project. The project is a result of my hard work, and determination put on by me with the help of Wikipedia, You Tube videos related to NLP Concepts, skikit-learn.org, Reffered some old projects on Google.

**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 an online hate and abuse comment classifier which can be 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:**

In the past few years its seen that the cases related to social media hatred have increased exponentially. The social media is turning into a dark venomous pit for people now a days. Online hate is the result of difference in opinion, race, religion, occupation, nationality etc.

In social media the people spreading or involved in such kind of activities uses filthy languages, aggression, images etc. to offend and gravely hurt the person on the other side. This is one of the major concerns now.

The result of such activities can be dangerous. It gives mental trauma to the victims making their lives miserable. People who are not well aware of mental health online hate or cyber bullying become life threatening for them. Such cases are also at rise. It is also taking its toll on religions. Each and every day we can see an incident of fighting between people of different communities or religions due to offensive social media posts.

Online hate, described as abusive language, aggression, cyberbullying, hatefulness, insults, personal attacks, provocation, racism, sexism, threats, or toxicity has been identified as a major threat on online social media platforms. These kinds of activities must be checked for a better future.

**Review of Literature:**

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.

It is also taking its toll on religions, each and every day we can see an incident of fighting between people of different communities or religions due to offensive social media posts.

**Motivation for the Problem Undertaken:**

The main aim of taking this project is, till now I have implemented only the machine learning algorithms where the features and labels having continuous data or categorical data, but I never worked on the data having long text which can be solved by using NLP concepts, so to learn different concepts in data science I choose this project.

The exposure to real world data and the opportunity to deploy my skillset in solving a real time problem has been the primary objective. However, the motivation for taking this project was that it is relatively a new field of research. Here we have many options but less concrete solutions.

The objective of this project 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. These comments which here are referred as malignant here represents the human behaviour as well as tendency of the individual in the society.

**Analytical Problem Framing**

**Mathematical/Analytical Modelling of the Problem:**

This is an NLP based Project and we deal with the textual data for understanding the data we used some methods like removing punctuations, numbers, stop words and using the lemmatization process convert the complex words into their simpler forms. These processes helped in cleaning the unwanted words form the comments and we were left with only those words which were going to help in our model building. After cleaning the data, we used TF-IDF Vectorizer technique to convert textual data into vector form. This technique works on the basis of the frequency of words present in the document. After training with train dataset, the same I have followed for the test dataset and converted all the text data and applied machine learning algorithms

**Data Sources and their formats:**

This data was provided to me by Flip Robo Technologies as a csv file format. This file contains training and testing dataset. On training dataset, we build a model and using this model we have to predict the outcomes from testing dataset. 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:

ID: It includes unique Ids associated with each comment text given.

Comment text: This column contains the comments extracted from various social media platforms.

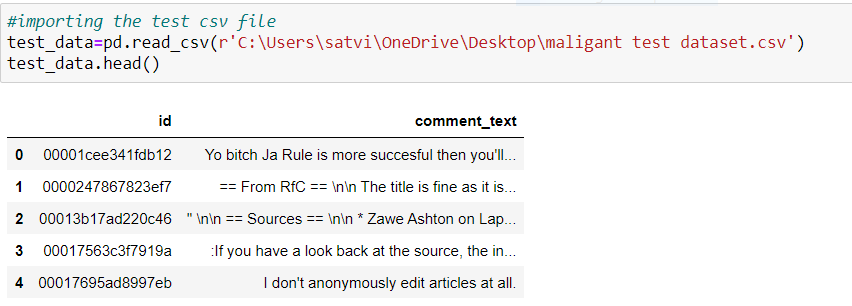
Malignant: It is the Label column, which includes values 0 and 1, denoting if the comment is malignant or not.

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.





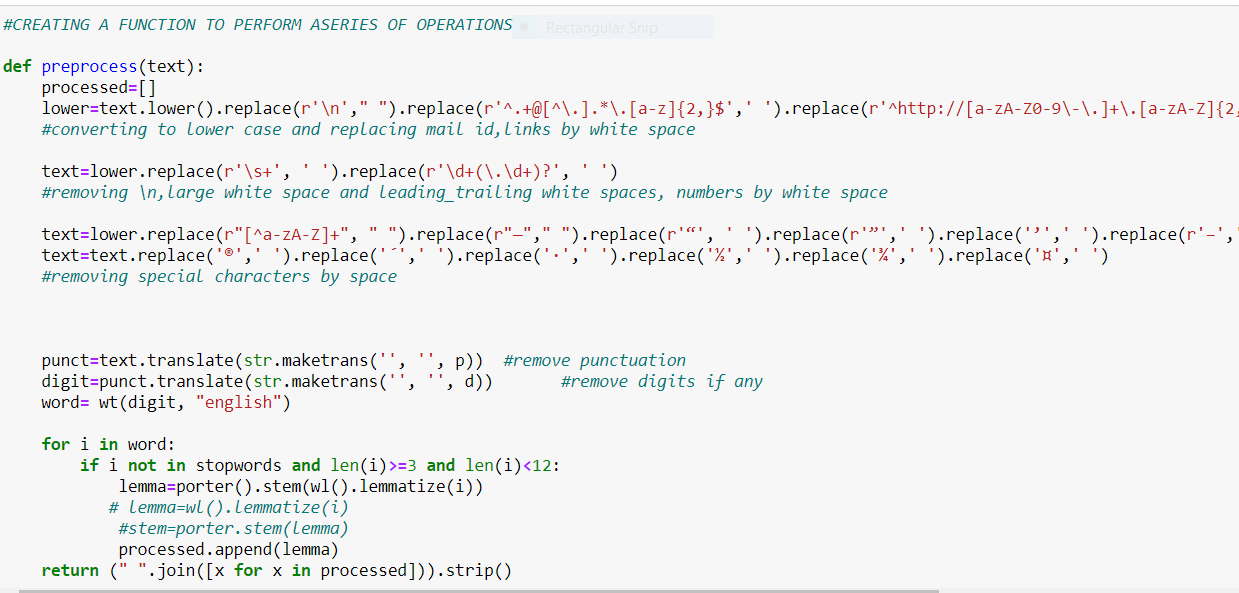
**Data Pre-processing:**

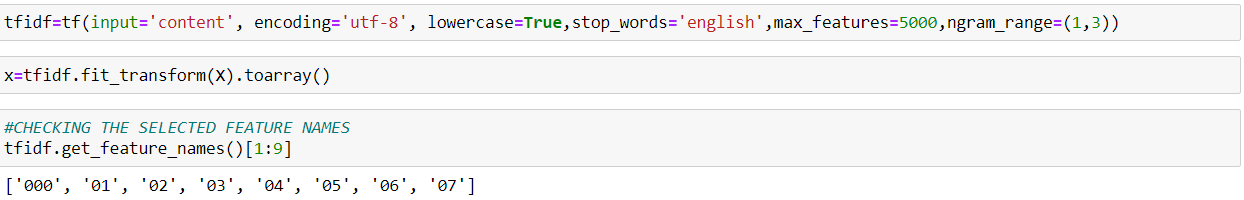
Data pre-processing is the data mining technique that involves transforming raw data into an understandable data format. So, in data pre-processing technique first step is import data and the libraries to be used in model building.

In the dataset there is a column of long text column, which cannot be understandable by the machine, so by using the Natural language processing techniques (NLP) we need to pre-process the data and apply various techniques to convert the text data and analyse the data. As a data scientist, the aim is to filter out these words from the social media from flashing of and blocking the user who does these types of acts on the regular basis a prototype of an online hate and abuse comment classifier which can be used to classify hate and offensive comments so that it can be controlled and restricted from spreading hatred and cyberbullying.

Since our feature column, has many stop-words, punctuations and special characters they are to be removed from the comments text because our machine cannot understand the characters and cannot be able to convert to the machine understandable language. So that need to be pre-processed, to be converted to lower cases and split the text.

Used TFIDF vectorizer to convert those text into vectors, and split the data and into test and train and trained various Machine learning algorithms.



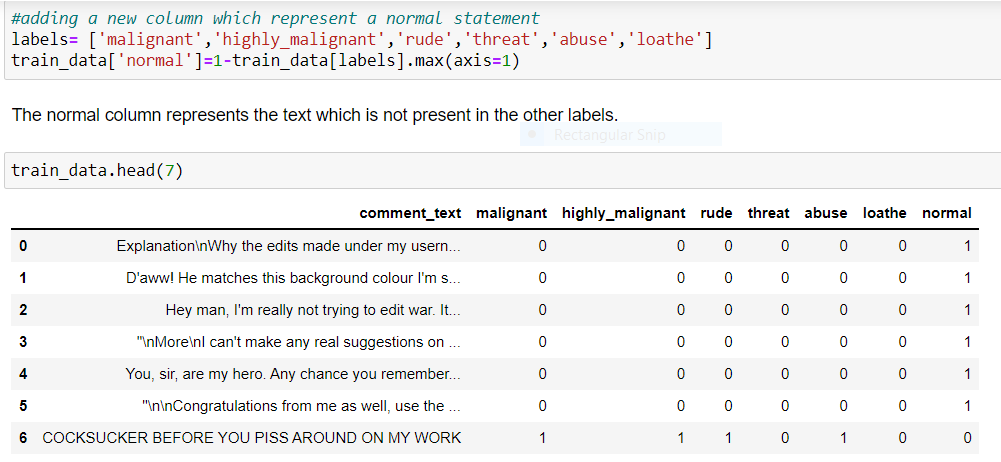


**Data Inputs- Logic- Output Relationships**

The input data for the processing and getting the output is converted in to the numerical forms or data words, in to the vector form and fed to the model one by one which analyses by the model providing the certain score through the medium of performance metrics.

So, there are no null or missing values in our dataset we move to next step of data cleaning. In data cleaning I have dropped the ID column as it gives no information. After dropping it we created another feature normal which shows the labelled data of positive and negative comments.

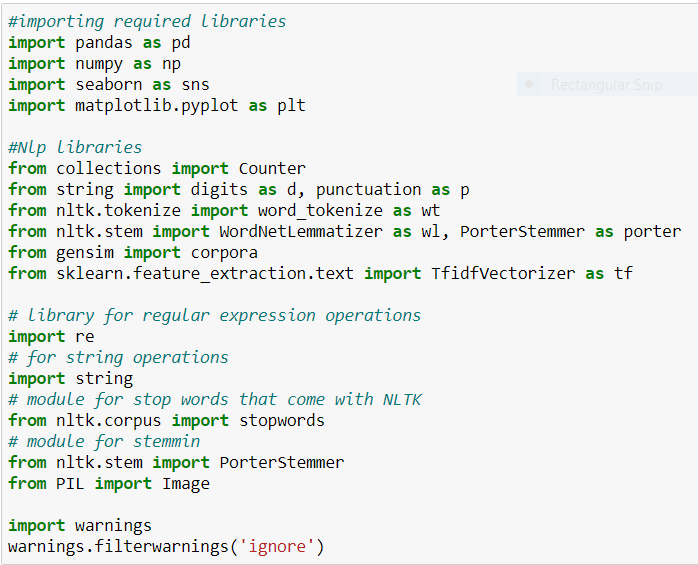
Now in cleaning the textual data we created a function to remove unwanted space, punctuation, numbers, emails, phone numbers etc and converted upper case letters into lower case and append the result into a new column. After the removal of unwanted notations, we moved to remove stop-words.





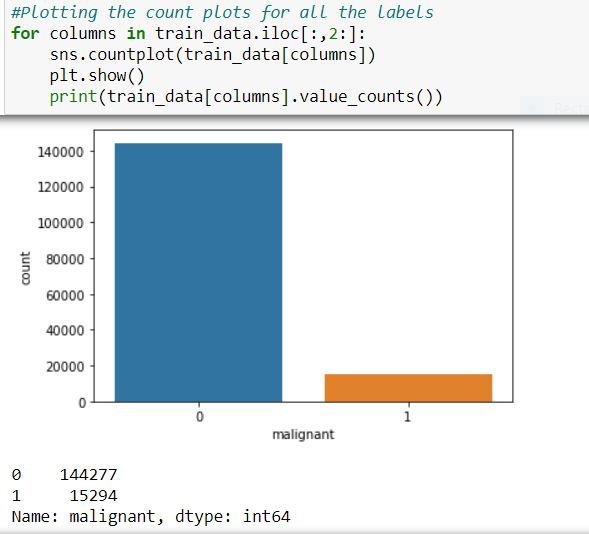
**Hardware and Software Requirements and Tools Used:**

I have used Laptop with i5 processor with 8 GB Ram, Jupyter Notebook, Python libraries for processing, Scikit learn library for using machine learning concepts and NLP concepts for pre-processing the data converting the text to the words using tokenization and then words to vectors using TDIF.

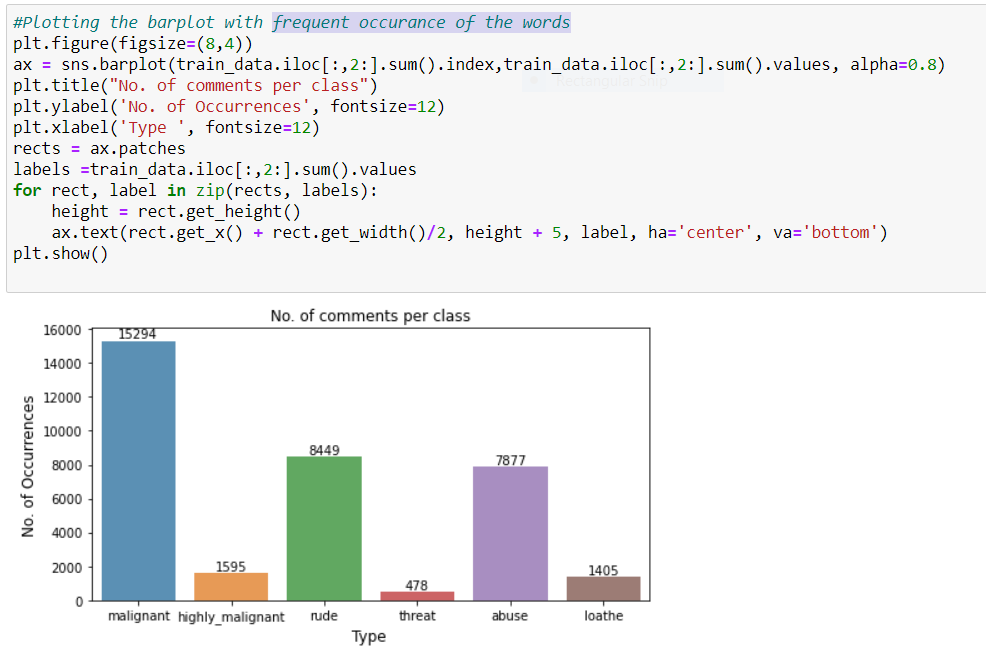


**Visualisation:**

**I have plotted count plots for all the labels which are having the classes as 0’s and 1’s and checked how the classes are distributed throughout the columns.**



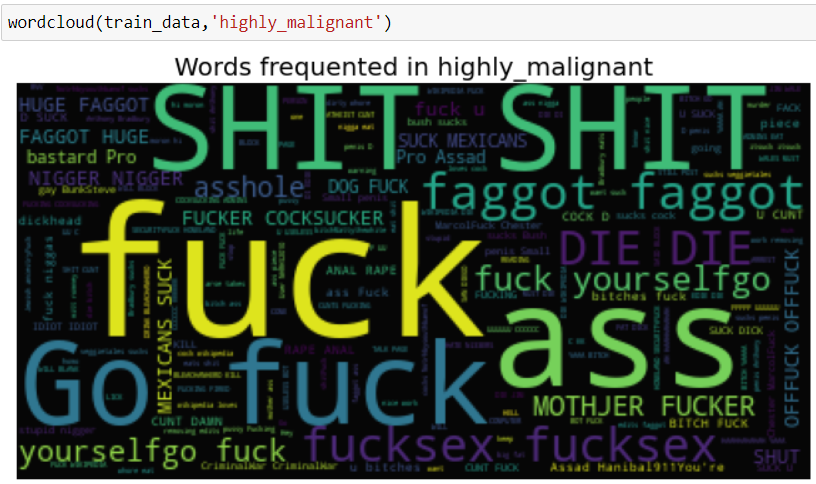
I have plotted bar graphs and checked the frequent occurances of the classes in the labels.

Percentages of the categories of the labels are taken and plotted the graph and checked how the comments are distributed in each label as shown in below screenshot.



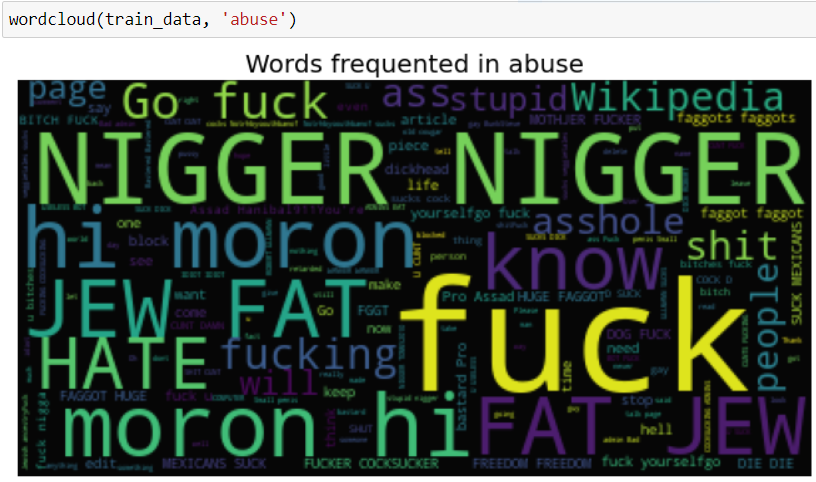
I have plotted and visualised the frequent used words in the comment texts as shown below using word cloud.











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

These are the following approaches I have used here:

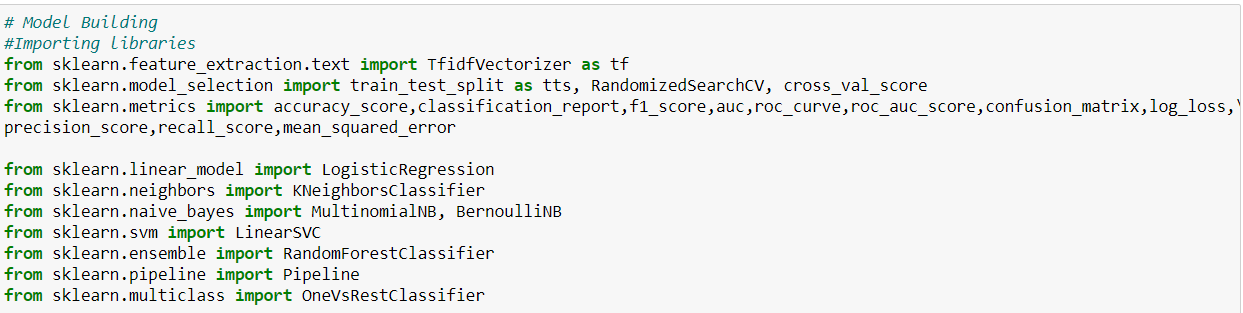
• Importing and drawing insights from the data using python libraries.

• Analysing the data and used proper pre-processing techniques to extract out important words from the comment text using NLP concepts.

• Applying data operations such as TDIF to convert text to vectorized form.

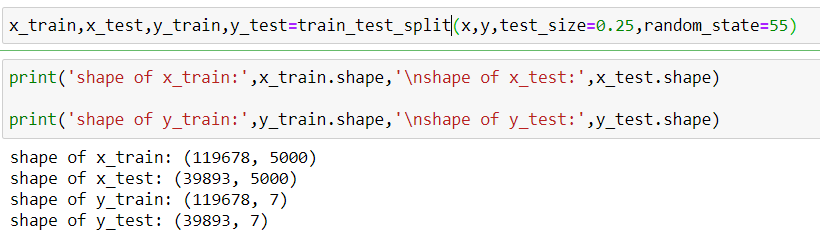
• Applied different Machine learning algorithms to check the performance using evaluation metrics.

In the given dataset there are 6 labels which defines the type of comments like malignant, highly malignant, hate, abuse, threat, loathe but we created another feature named as “normal” which is created based on the given labels. As the labels are in the form of 0’s and 1’s this type of problem comes under classification problem so I have used classification algorithms for the prediction. We cannot use regular algorithms for prediction because if we use regular SVC or ensemble techniques that may eat up all the memory and there is will be no output at last. So, I have used Linear SVC model rather than regular SVC and imported the algorithms and the methods required for evaluating the models and for splitting our dataset in to train and test data using train test split method.



**Testing of Identified Approaches (Algorithms)**

First, I have split the dataset into and x and y variables and then applied TDIF method on the words and converted the words into vectors and then split the x and y variables into train and test data. I have considered 75% train size and 25% test size and took the random state as 55. Then applied Machine learning algorithms on the train data and test data, Checked the performance, evaluated the model and finalised the model.



Following are the algorithms that applied on this dataset.

• LogisticRegression

• KNeighborsClassifier

• MultinomialNB, BernoulliNB

• LinearSVC

• RandomForestClassifier

I have used these models because when I tried applying different models for prediction but I have come across with memory related errors, so in order to avoid the errors I have used these models for prediction.

**Logistic Regression**













**Metrics for success in solving problem under consideration:**

For solving the problems and understanding the result of each algorithm, we have used different metrics like

a) Accuracy Score

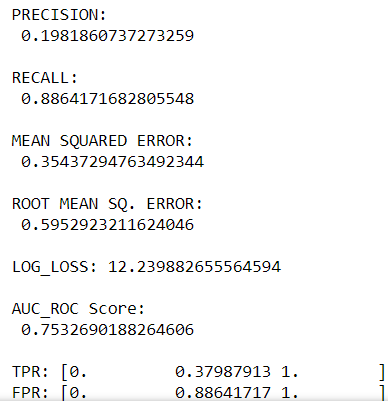
b) Classification Report

c) Confusion Matrix

d) Log Loss

e) Roc-Auc

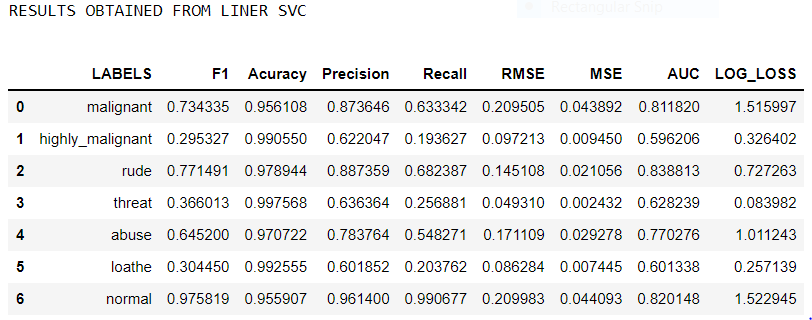
Key metrics are F1-score, Precision, Recall and Roc-Auc Score.

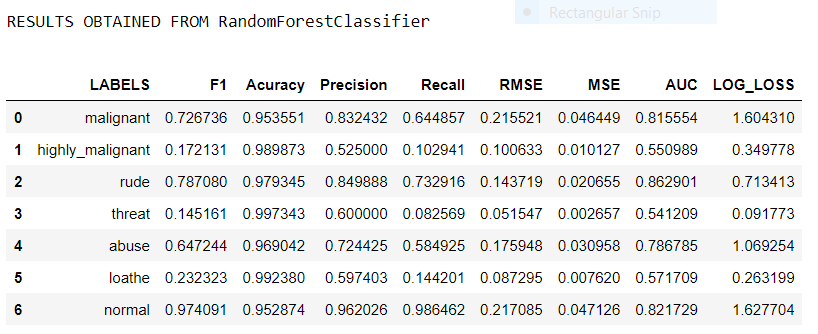
****

**Interpretation of the Results**

Almost all the algorithms which we applied are giving the good accuracy and the performance is also good. I have considered Linear-SVM and Random Forest classifier as my final models for building since it is giving best output results and evaluation metrics with minute differences.

As our target feature is imbalanced which means that alone accuracy score will not give best results but instead accuracy, we are using the classification report log loss score, Roc-AUC-score and confusion matrix to find the best algorithm. So, I have selected Linear SVM, Random Forest Classifier has 81% ROC\_AUC score and log loss of 1.51.





**CONCLUSION** **Key Findings and Conclusions of the Study**

Some of the findings from data explorations are:

• From displaying the data, it seems there are of special characters present in the data. So, it is better to proceed by filter it out.

• As the comments text data is in text format, so presence of special characters and stop-words will be there.

• After proper cleaning and processing, Linear SVM and Random Forest Classifier gives the highest accuracy and ROC AUC score.

From this project I have learnt, new techniques and ways to deal with uncleaned data. Find a solution to deal with multiple target variables. Tools used for visualizations, gives a better understanding of dataset. We have used different classification algorithms in which we have two labels. So, after applying many models, Linear SVC and Random-forest classifier gave better results compared to other models.

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

Some of the extension techniques that gives the best results are: multinomial Naive bayes algorithm and Random Forest Algorithm. After, analysing for each behaviour separately by creating models. • Support Vector Machine performs well on text data, but its hyperparameter tuning is very complex and takes much time for execution. So, I have not performed hyper parameter tuning on the Selected models. But, Hyper-parameter tuning would have resulted better accuracy.

Every effort has been put on it for perfection but nothing is perfect and this project is of no exception. There are certain areas which can be enhanced. Comment detection is an emerging research area with few public datasets. So, a lot of works need to be done on this field.

Working on this project created new ideas to think about but there were some limitations in this project like unbalanced dataset, multiple target features. To overcome these limitations, we have to use balanced dataset so that during the training of dataset our algorithm will not give biased result.