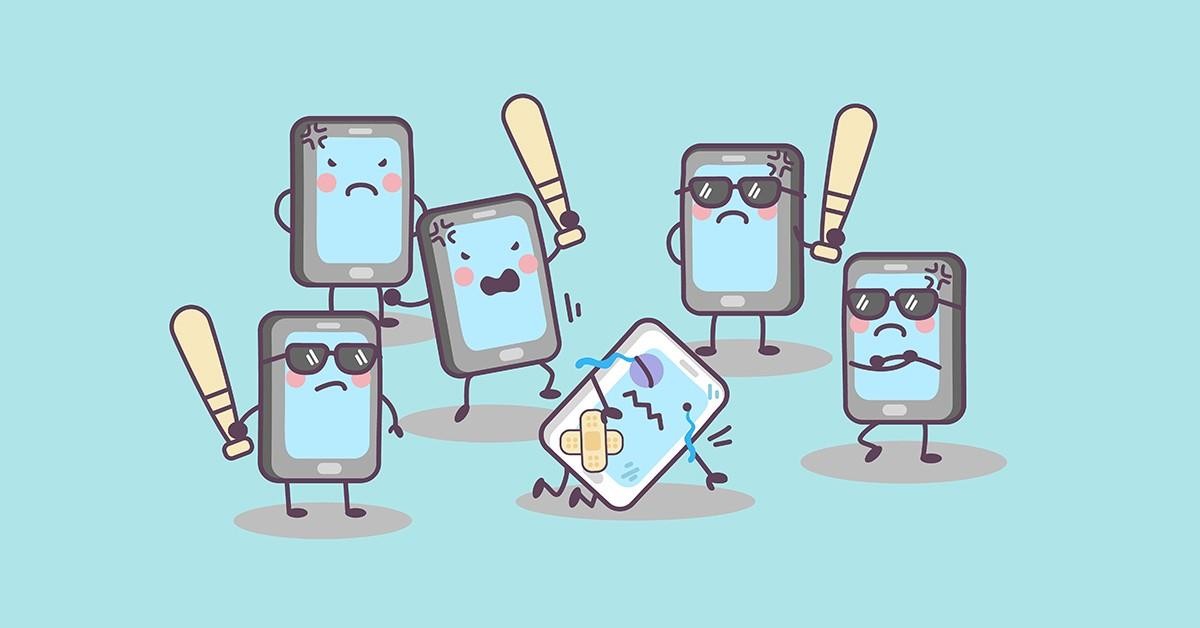


**MALIGNANT COMMENTS CLASSIFICATION**



**SME Name:**

Mohd. Kashif

**Prepared By:**

Abhishek Ranjan

### ACKNOWLEDGMENT

##### I would like to convey my heartfelt gratitude to Flip Robo Technologies for providing me with this wonderful opportunity to work on a Machine Learning project using NLP “***MALIGNANT COMMENTS CLASSIFICATION***” and also want to thanks my SME for providing the dataset and helping me to complete this project. This project would not have been accomplished without their help and insights.

##### I would also like to thank my academic “Data Trained Education” and their team who has helped me to learn Machine Learning and NLP.

* **INTRODUCTION**

1. **Business Problem Framing**

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 cyber bullying.

#### Conceptual Background of the Domain Problem

There has been a remarkable increase in the cases of cyber bullying 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.

#### Review of Literature

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, cyber bullying, 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 behavior.

#### Motivation for the Problem Undertaken

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.

* **Analytical Problem Framing**

#### Mathematical/ Analytical Modeling of the Problem

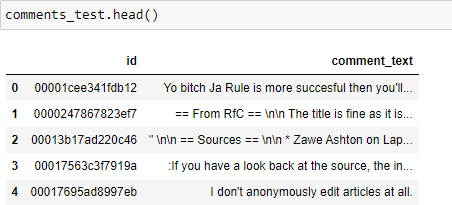
* 1. Used Panda’s Library to save data into csv file
  2. Cleaned Data by removing irrelevant features
  3. Descriptive Statistics
  4. Analyzed correlation
  5. Converted all messages to lower case
  6. Replaced email addresses with 'email'
  7. Replaced URLs with 'webaddress'
  8. Replaced money symbols with 'moneysymb' (£ can by typed with ALT key + 156)
  9. Replaced 10digit phone numbers (formats include parenthesis, spaces, no spaces, dashes) with 'phone number'
  10. Replace Numbers with 'number'
  11. Removed Punctuation
  12. Replaced extra space
  13. Replaced leading and trailing white space
  14. Removed \n
  15. Added and removed stop words
  16. Words of Sentence
  17. Calculated length of sentence
  18. Made one Target Column
  19. Removed Total length
  20. Checked the word which are offensive using WordCloud
  21. Checked the word which are not offensive using WordCloud
  22. Converted text into vectors using TF-IDF

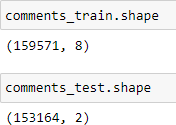
## Data Sources and their formats

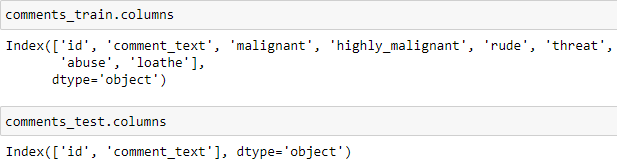
There are two data-set in csv format: **train and test dataset**. Features of this dataset are:

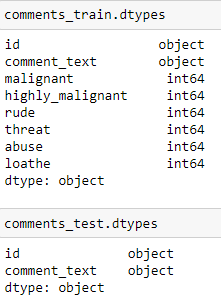
* Malignant: It is the Label column, which includes values 0 and 1, denoting if the comment is malignant or not.
* Highly Malignant: It denotes comments that are highly malignant and hurtful.
* Rude: It denotes comments that are very rude and offensive.
* Threat: It contains indication of the comments 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.

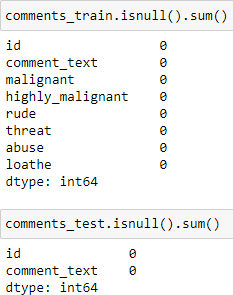
## Data Pre-processing:

1. Checked Top 5 Rows of both Dataset
2. Checked Total Numbers of Rows and Column

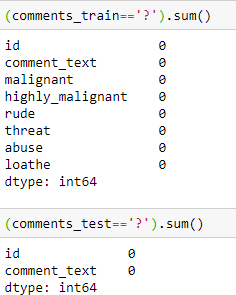
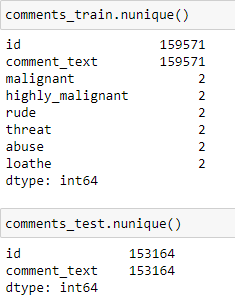


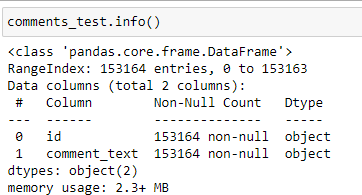
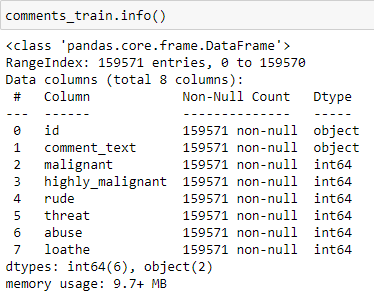
1. Checked All Column Name
2. Checked Data Type of All Data



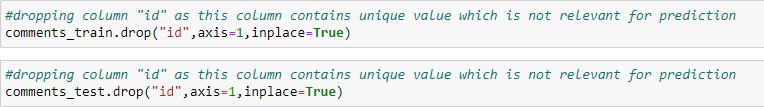
1. Checked for Null Values

There is no null value in the dataset.

1. Checking if "-" values present in dataset or not
2. Checked total number of unique values
3. Checking unique values present in the columns: ("malignant", "highly\_malignant", "rude", "threat", "abuse", "loathe")
4. Information about Data



1. Data cleaning
   * Dropped Column " id" as this column contains serial no.



1. Data Visualization
2. Uni-Variate Analysis
   * Used Countplot
3. Bivariate Analysis

(For comparison between each feature with target)

* + Used Bar plot

1. Multivariate Analysis

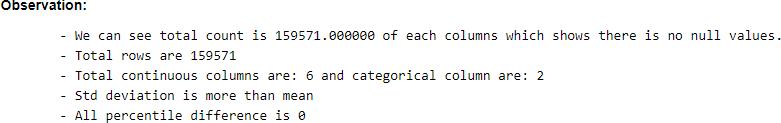
(For comparison between all features with target)

* + Used Pair plot

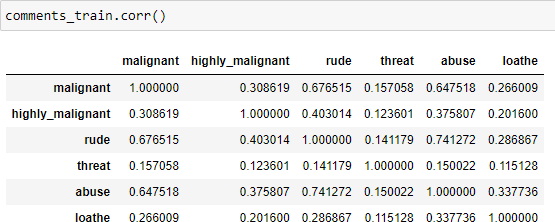
## Data Inputs- Logic- Output Relationships

### Descriptive Statistics

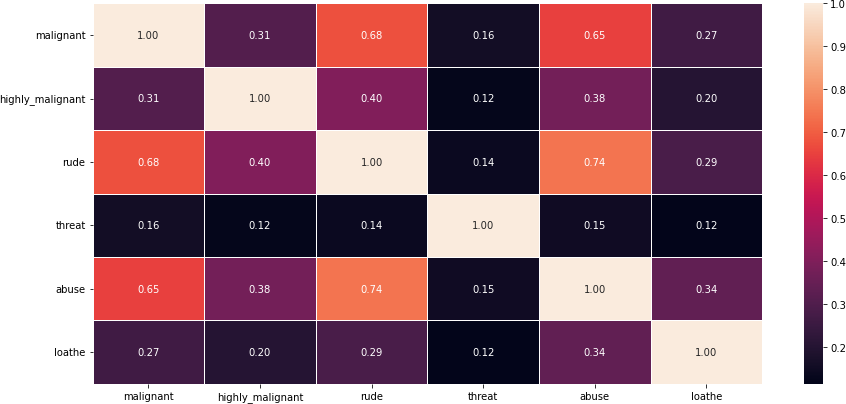


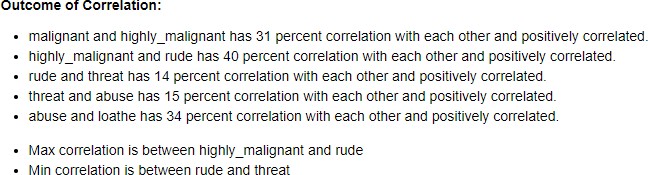


1. **Checking Correlation**

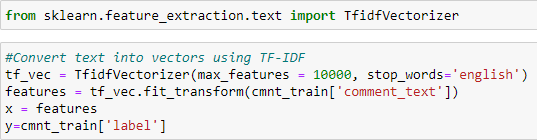
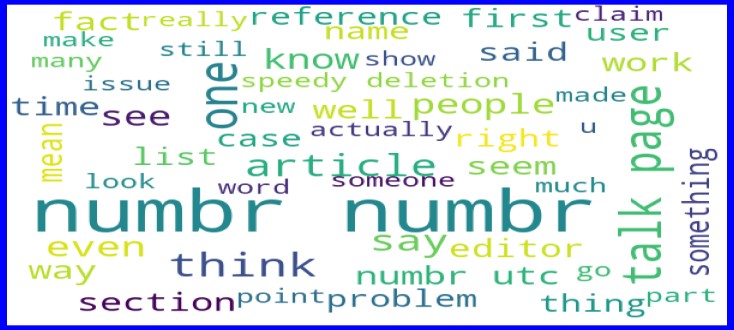
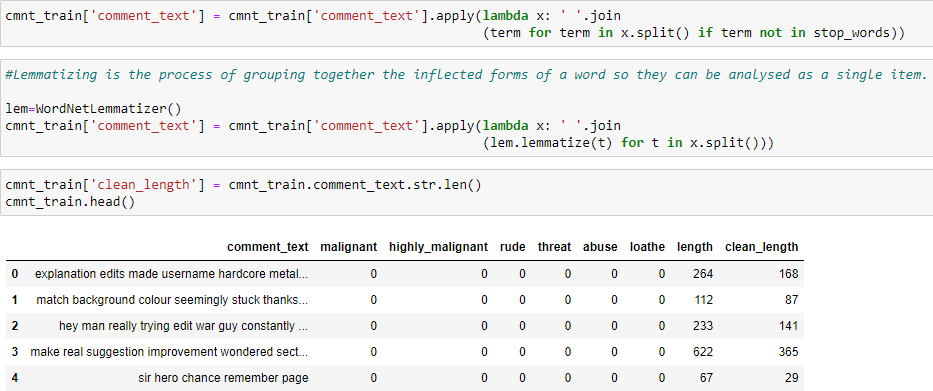


This gives the correlation between the dependent and independent variables.

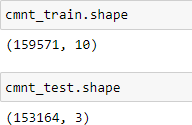
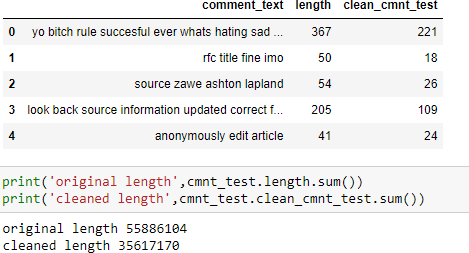




### Handling "cmnt\_train" Dataset



1. **Handling "cmnt\_test" Dataset**



#### State the set of assumptions (if any) related to the problem under consideration

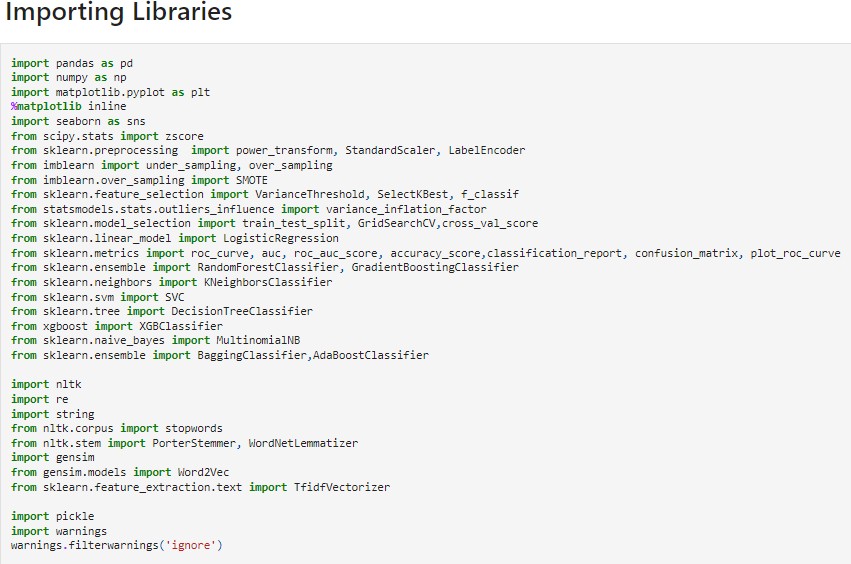
* It was observed that there is one column “id” which is irrelevant column as it contains serial no, so, have to drop this column.
* It was observed that in columns there are irrelevant values present in comment\_text. So, we need to drop, replace and remove those values.
* Also have to convert comment\_text into vectors using TF-IDF
* Have to create on Target column also.

#### Hardware and Software Requirements and Tools Used

###### Hardware used:

* + **Processor**: 11th Gen Intel(R) Core (TM) i3-1125G4 @ 2.00GHz 2.00 GHz
  + **System Type**: 64-bit OS

###### Software used:

* + **Anaconda** for 64-bit OS
  + **Jupyter** notebook
* **Tools, Libraries and Packages used:**



# * Model/s Development and Evaluation

1. Identification of possible problem-solving approaches (methods)

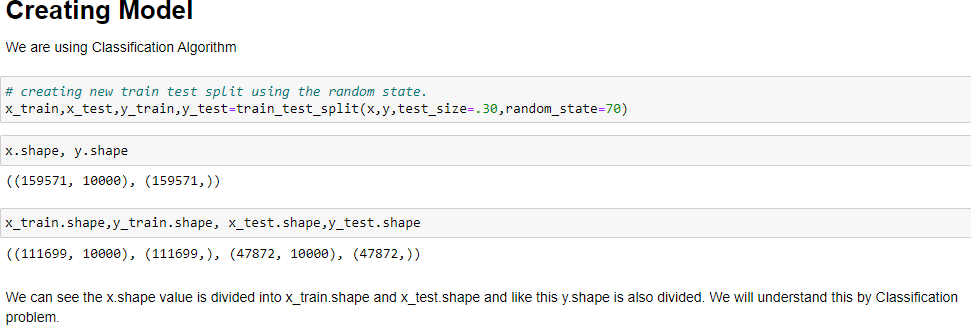
In this project, we want to differentiate between comments and its categories and for this we have used these approaches:

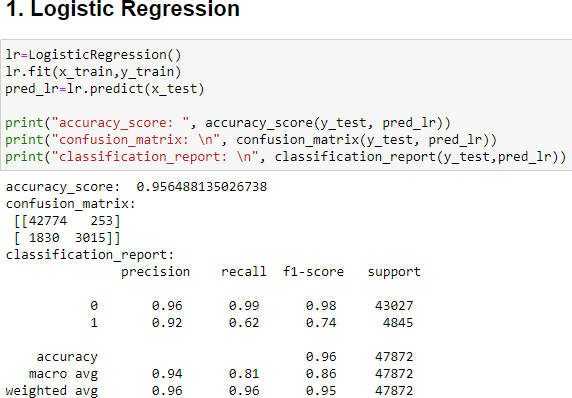
* Checked Total Numbers of Rows and Column
* Checked All Column Name
* Checked Data Type of All Data
* Checked for Null Values
* Checked for special character present in dataset or not
* Checked total number of unique values
* Information about Data
* Checked Description of Data and Dataset
* Dropped irrelevant Columns
* Replaced special characters and irrelevant data
* Checked all features through visualization.
* Checked correlation of features
* Converted all messages to lower case
* Replaced email addresses with 'email'
* Replaced URLs with 'webaddress'
* Replaced money symbols with 'moneysymb' (£ can by typed with ALT key + 156)
* Replaced 10digit phone numbers (formats include parenthesis, spaces, no spaces, dashes) with 'phone number'
* Replace Numbers with 'number'
* Removed Punctuation
* Replaced extra space
* Replaced leading and trailing white space
* Removed \n
* Added and removed stop words
* Words of Sentence
* Calculated length of sentence
* Made one Target Column
* Removed Total length
* Checked the word which are offensive using WordCloud
* Checked the word which are not offensive using WordCloud
* Converted text into vectors using TF-IDF

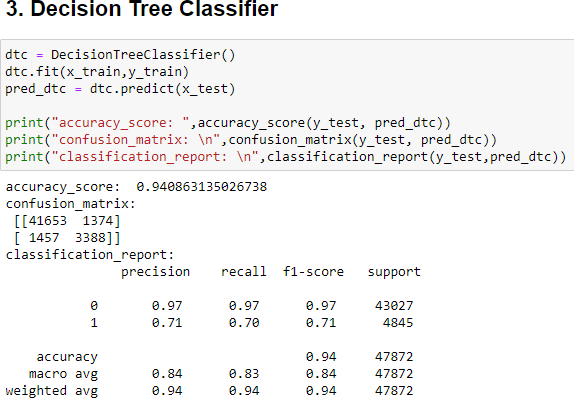
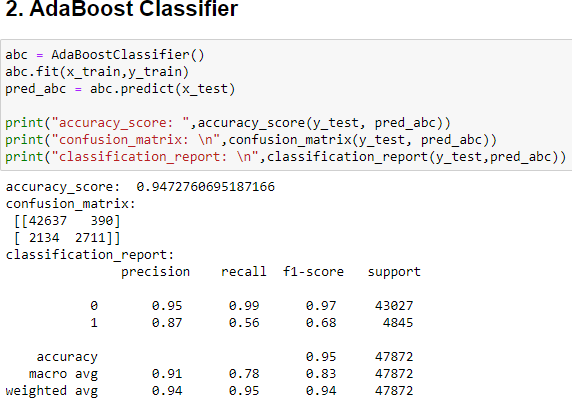
**Testing of Identified Approaches (Algorithms)**

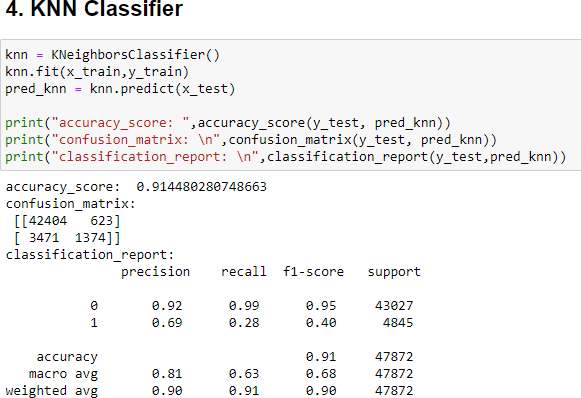
* 1. Logistic Regression
  2. AdaBoost Classifier
  3. Decision Tree Classifier
  4. KNN Classifier
  5. Gradient Boosting Classifier
  6. XGB Classifier
  7. MultinomialNB

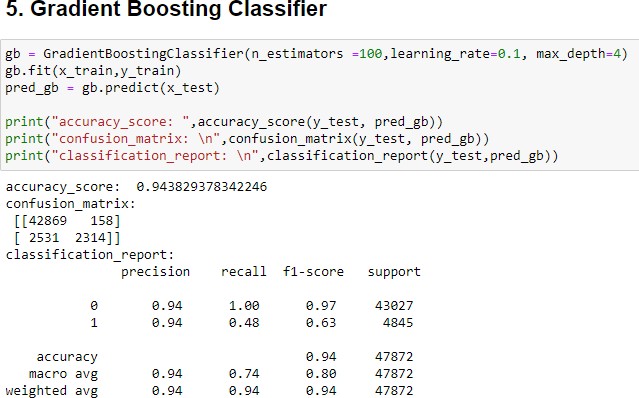
1. **Run and evaluate selected models**

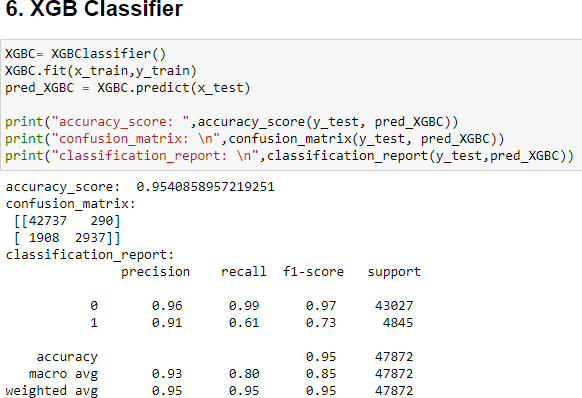


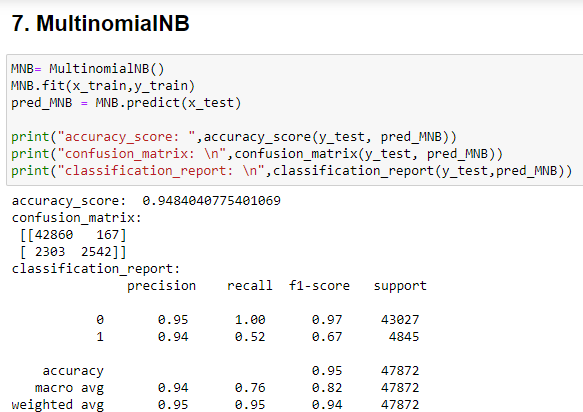


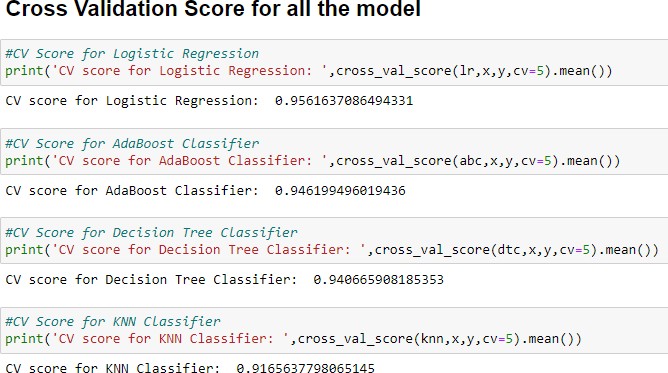


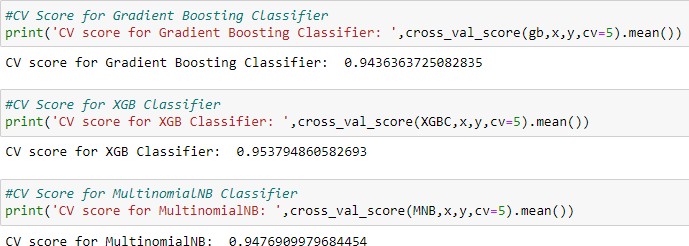


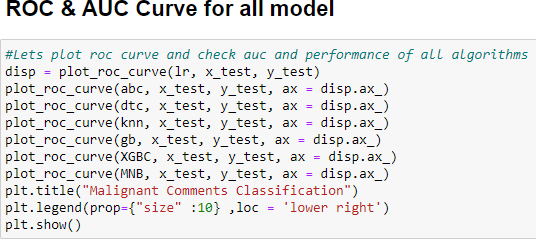


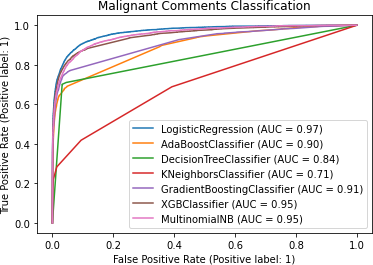




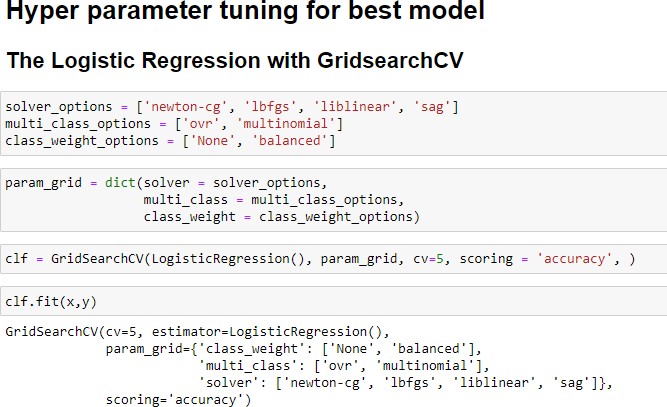


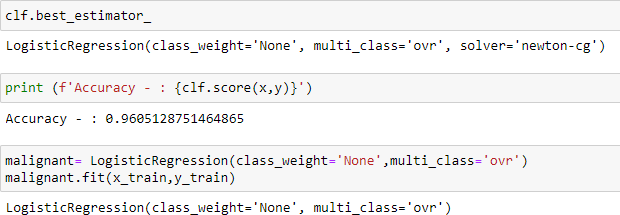


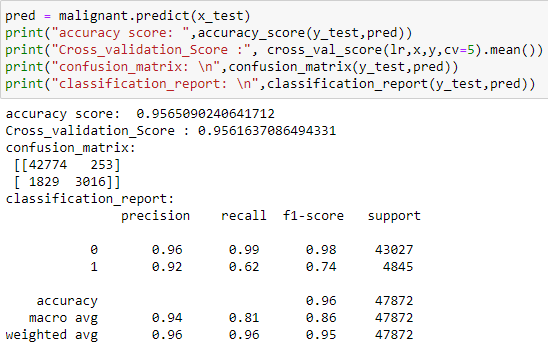


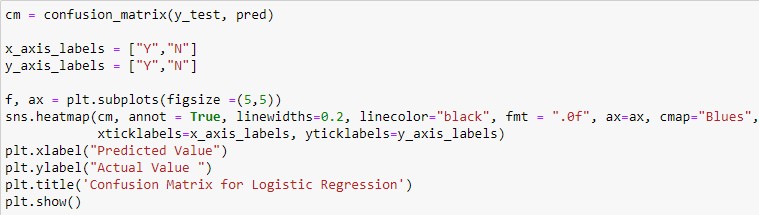


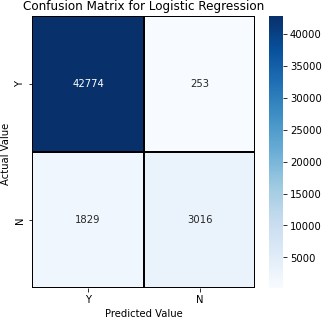




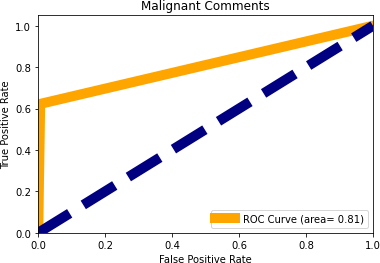
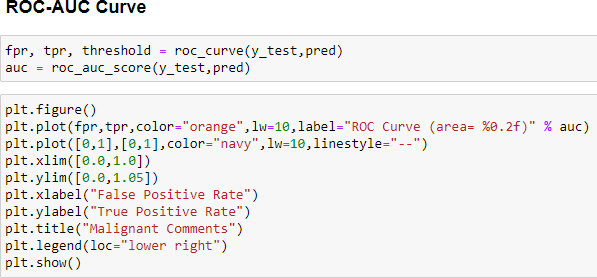


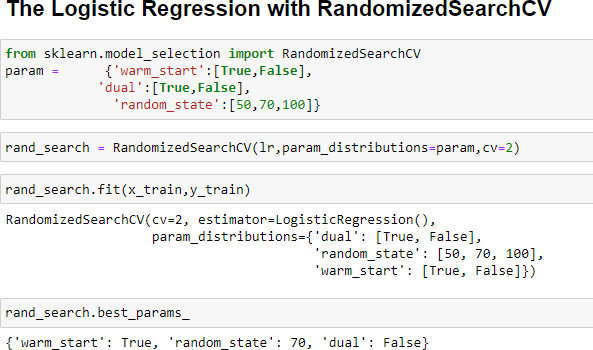


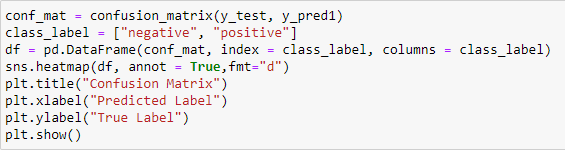
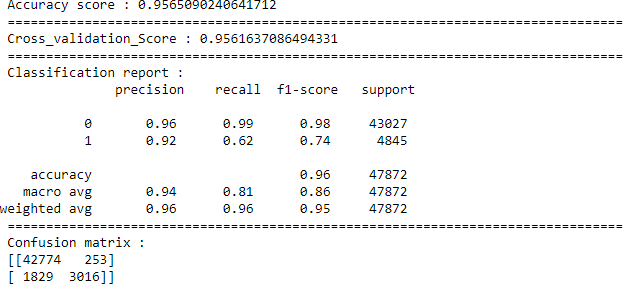
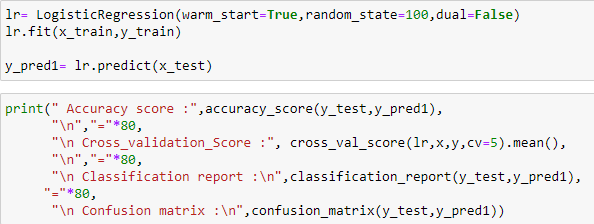


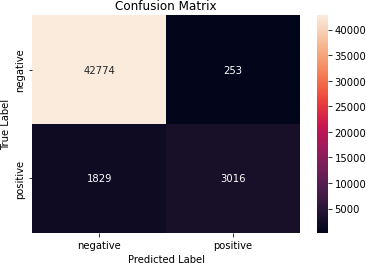




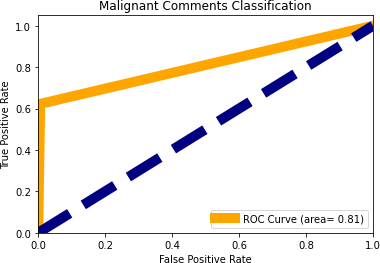
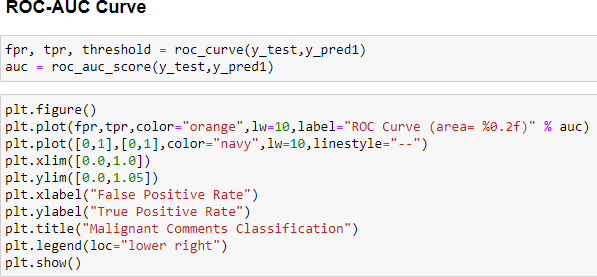










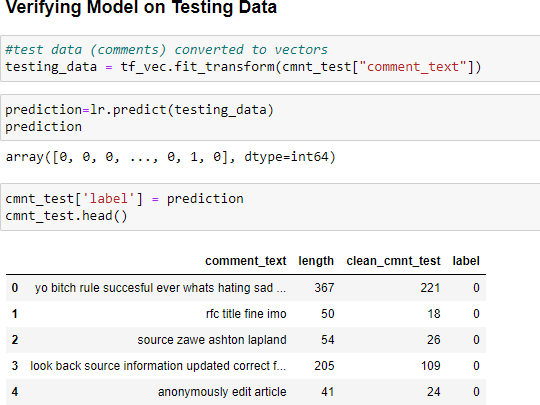


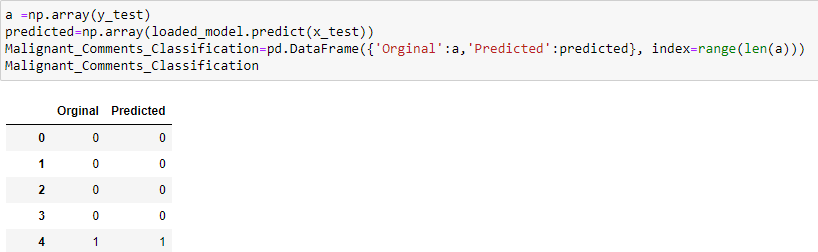


* **Saving The Predictive Model**



* **Comparing Actual and Predicted**





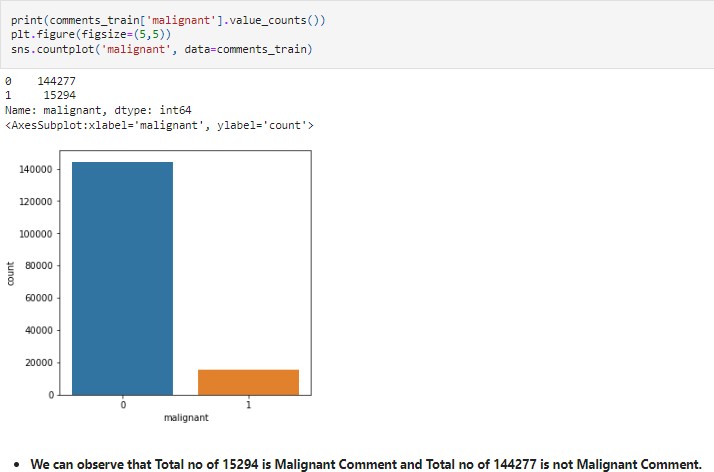
### Saving the model in CSV format



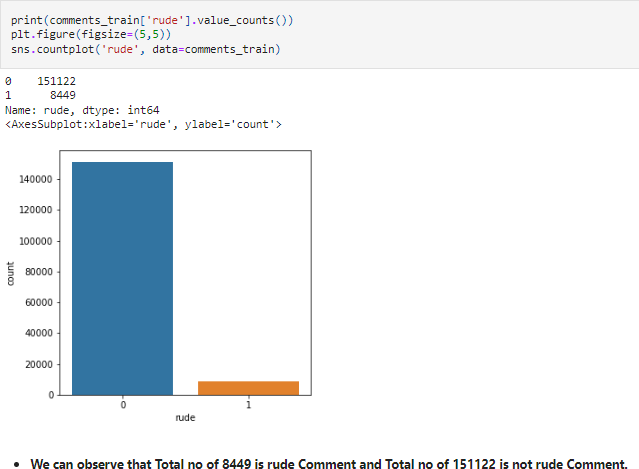
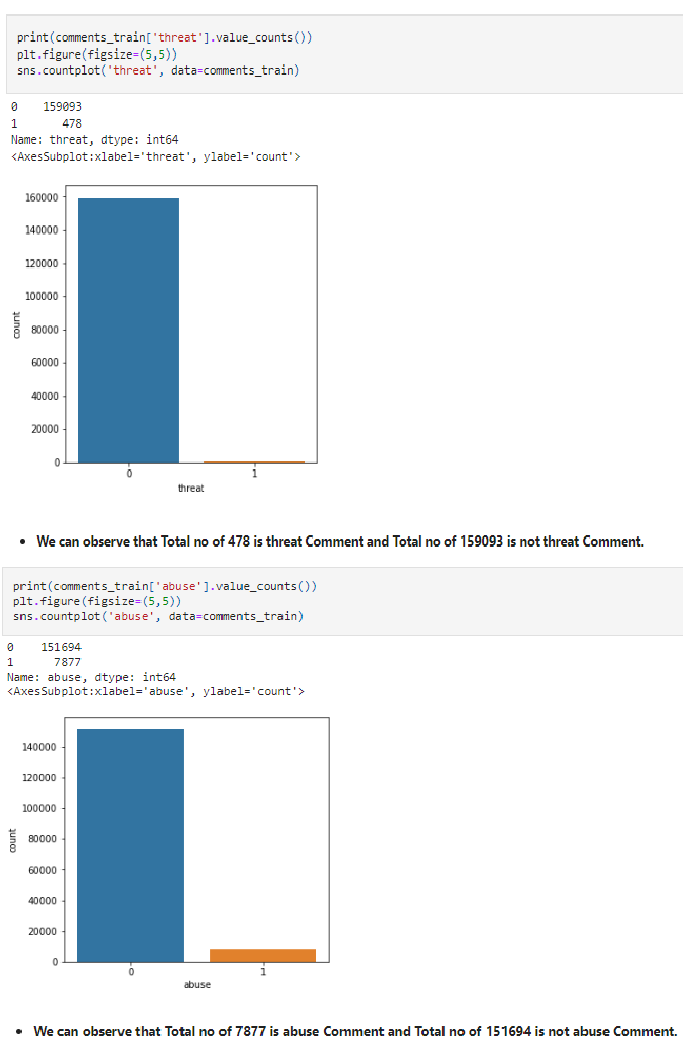
##### Key Metrics for success in solving problem under consideration

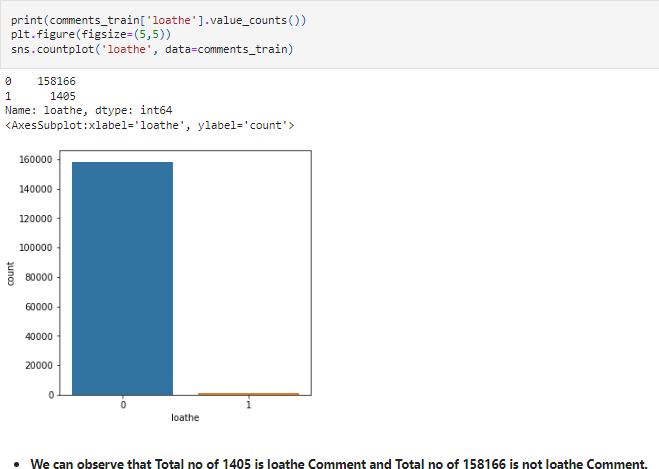
* Accuracy Score, Precision Score, Recall Score, F1-Score and CV score are used for success. Also, confusion matrix and AUC-ROC Curve is used for success.

##### Visualization

* Uni-Variate Analysis
  + Using Countplot



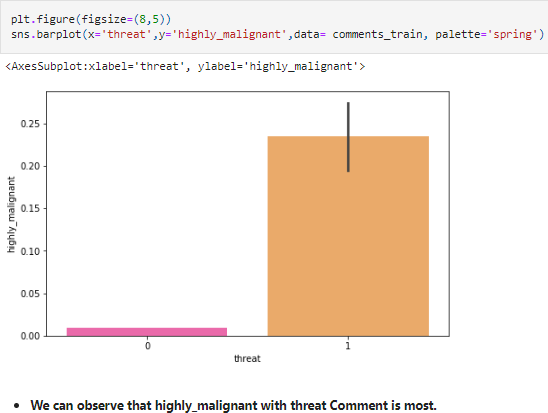
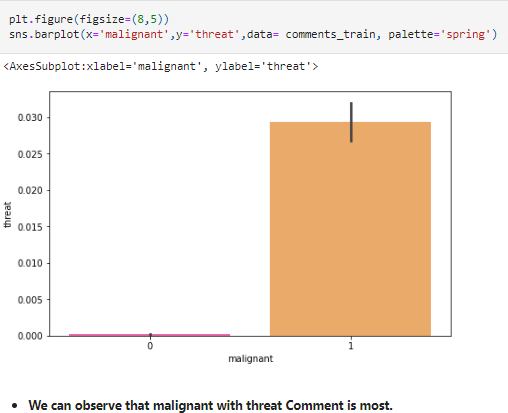


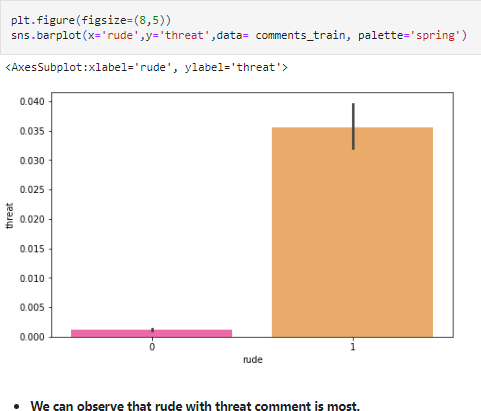


* Bivariate Analysis

(For comparison between each feature with target)

* + Using Bar plot

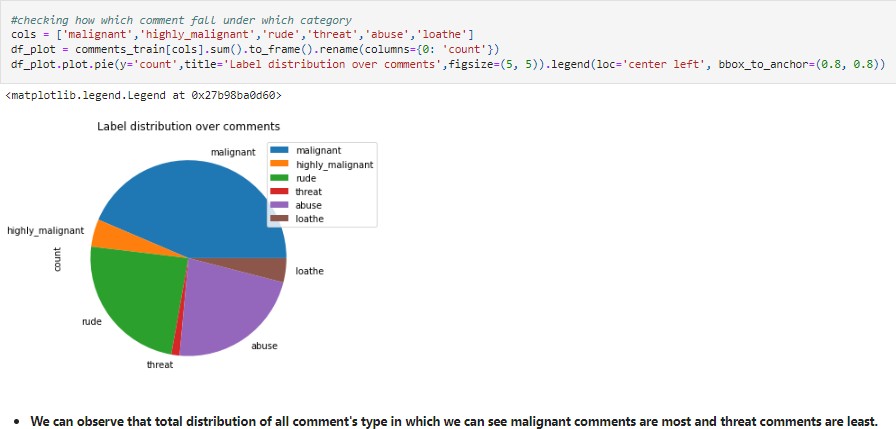




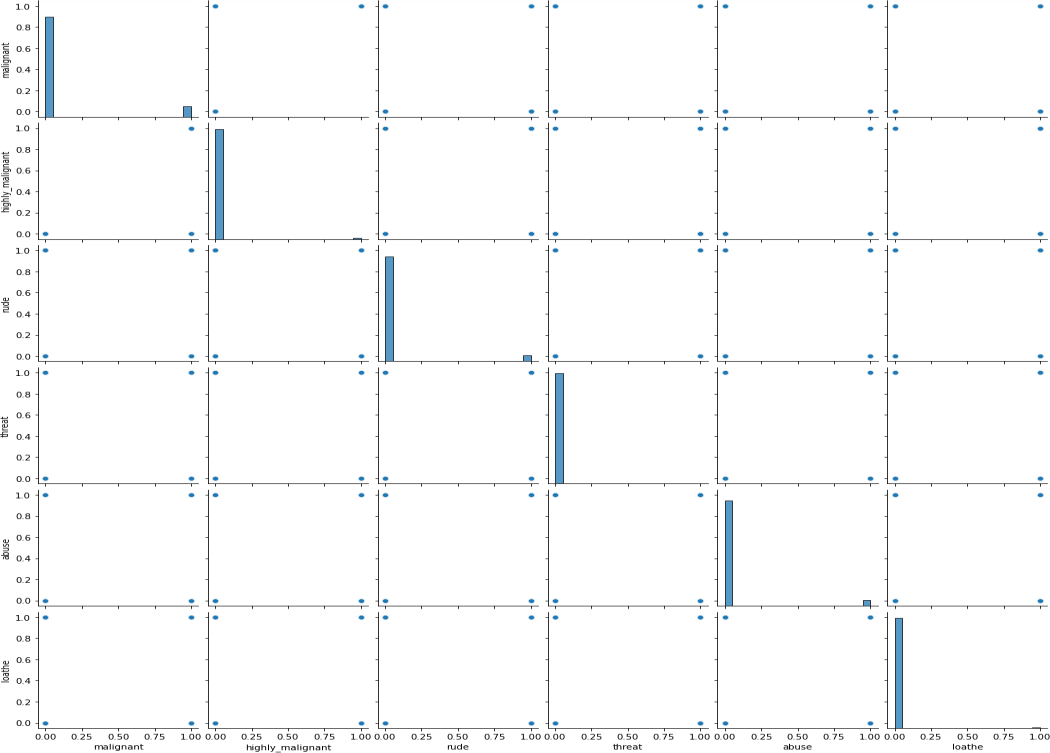


* Multivariate Analysis

(For comparison between all features with target)

* + Using Pie-Plot
  + Using Pair plot





##### Interpretation of the Results

* Through Pre-processing it is interpretive Converted all messages to lower case, Replaced email addresses with 'email', Replaced URLs with 'webaddress', Replaced money symbols with 'moneysymb', (£ can by typed with ALT key + 156), Replaced 10digit phone numbers (formats include parenthesis, spaces, no spaces, dashes) with 'phone number', Replace Numbers with 'number', Removed Punctuation, Replaced extra space, Replaced leading and trailing white space, Removed \n, Added and removed stop words, Calculated length of sentence, Made one Target Column, Removed Total length, Converted text into vectors using TF-IDF
* By creating/building model we get best model: Logistic Regression.

# * CONCLUSION

#### Key Findings and Conclusions of the Study

Here we have made a MALIGNANT COMMENTS CLASSIFICATION. We have done EDA, cleaned data and Visualized Data. While cleaning the data it is analyzed that:

* One column “id” is irrelevant so dropped this column.

After that we have done prediction on basis of Data using Data Pre- processing, Checked Correlation, removed email addresses, URLs, money symbols, 10digit phone numbers, Punctuation, extra space, leading and trailing white space, \n, stop words, converted text into vectors using TF-IDF and at last train our data by splitting our data through train-test split process.

Built our model using 7 models and finally selected best model which was giving best accuracy that is Logistic Regression. Then tuned our model through Hyper Tuning using GridSearchCV and RandomizedSearchCV, in which proceeded with RandomizedSearchCV. And at last compared our predicted and Actual test data. Thus, our project is completed.

#### Learning Outcomes of the Study in respect of Data Science

* This project has demonstrated the importance of NLP.
* Through different powerful tools of visualization, we were able to analyze and interpret the huge data and with the help of pie plot, count plot & word cloud, I am able to see the distribution of threat comments.
* Through data cleaning we were able to remove unnecessary columns, values, special characters, symbols, stop-words and punctuation from our dataset due to which our model would have suffered from over fitting or under fitting.

###### The few challenges while working on this project were: -

* To find punctuations & stop words, which took time to run using NLP.
* The data set is huge it took time to run some algorithms & to check the cross-validation score.

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

While we couldn’t reach out goal of 100% accuracy but created a system that made data get very close to that goal. This project allows multiple algorithms to be integrated together to combine modules and their results to increase the accuracy of the final result.

For better performance, we plan to judiciously design deep learning network structures, use adaptive learning rates and train on clusters of data rather than the whole dataset.