

*MALIGNANT COMMENT CLASSIFIER*

Submitted by:

VIVEK KUMAR SINGH

**ACKNOWLEDGMENT**

The completion of this thesis study would not have been possible without the support and extensive knowledge of several website.

I cannot begin without expressing my thanks to my SME from *Flip Robo Technologies* , ***MR.SHWETAK MISHRA*** for the valuable experience and insight in conducting and writing scientific reports. Thank you for always answering all questions and giving me helpful practical suggestions.

Finally, I would like to express my sincere gratitude to my family for their constant encouragement and support throughout my time on this thesis.

**INTRODUCTION**

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.

**ANALYTICAL PROBLEM FRAMING**

* *Mathematical/ Analytical Modeling of the Problem*
* In this project, we will develop and evaluate the performance and predictability of trained and tested models based on comments which is provide by flip robo technology. Once we get a good fit, we will apply on our test data.
* In here we will use various classification algorithm to predict our target. Let's have an overview of the algorithms we will use for our predictions. To read more about these algorithms , just click on the algorithms name.

[Logistic Regression](https://www.google.com/search?q=linear+regression&rlz=1C1CHBF_enIN997IN998&oq=&aqs=chrome.1.69i59i450l8.734952339j1j15&sourceid=chrome&ie=UTF-8):- Logistic regression analysis is valuable for predicting the likelihood of an event. It helps determine the probabilities between any two classes. In a nutshell, by looking at historical data, logistic regression can predict whether: An email is a spam

[Decision Tree Classifier](https://www.google.com/search?q=about+DecisionTreeRegressor&rlz=1C1CHBF_enIN997IN998&ei=7kG5YoWNM6fA3LUPqcGy8AQ&ved=0ahUKEwiFvLry9Mz4AhUnILcAHamgDE4Q4dUDCA4&uact=5&oq=about+DecisionTreeRegressor&gs_lcp=Cgdnd3Mtd2l6EAM6BAgAEA1KBAhBGABKBAhGGABQAFjfEWDdFWgAcAF4AIABqQKIAZYLkgEDMi02mAEAoAEBwAEB&sclient=gws-wiz):- Decision trees help you to evaluate your options. Decision Trees are excellent tools for helping you to choose between several courses of action. They provide a highly effective structure within which you can lay out options and investigate the possible outcomes of choosing those options.

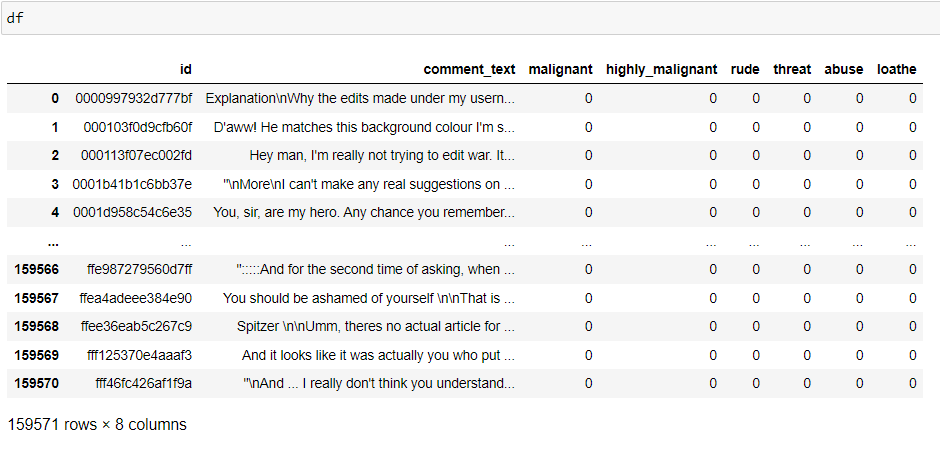
[SVR](https://www.google.com/search?q=about+SVR&rlz=1C1CHBF_enIN997IN998&oq=about+SVR&aqs=chrome..69i57j0i10i22i30l6j0i390l3.4767j1j15&sourceid=chrome&ie=UTF-8):- The basic idea behind SVR is to find the best fit line. In SVR, the best fit line is the hyperplane that has the maximum number of points. Unlike other Regression models that try to minimize the error between the real and predicted value, the SVR tries to fit the best line within a threshold value.

[K Neighbours Classifier](https://www.google.com/search?q=about+KNeighborsRegressor&rlz=1C1CHBF_enIN997IN998&oq=about+KNeighborsRegressor&aqs=chrome..69i57j33i160.3952j1j15&sourceid=chrome&ie=UTF-8):- By default, the K Neighbors Classifier looks for the 5 nearest neighbors. We must explicitly tell the classifier to use Euclidean distance for determining the proximity between neighboring points. Using our newly trained model, we predict whether a tumor is benign or not given its mean compactness and area..

[Random Forest Classifier](https://www.google.com/search?q=about+RandomForestRegressor&rlz=1C1CHBF_enIN997IN998&ei=n0a5Yq5xxJWx4w_O07lA&ved=0ahUKEwjuvN-u-cz4AhXESmwGHc5pDggQ4dUDCA4&uact=5&oq=about+RandomForestRegressor&gs_lcp=Cgdnd3Mtd2l6EAMyBwghEAoQoAEyBwghEAoQoAE6BwgAEEcQsAM6CggAEOQCELADGAE6BQgAEIAEOggIABCxAxCDAToICAAQgAQQsQM6CwgAEIAEELEDEIMBOgUIABCRAjoLCC4QgAQQsQMQgwFKBAhBGABKBAhGGAFQywhYjiBgrzJoAXABeAKAAbQDiAH5EJIBCTAuMi4yLjAuM5gBAKABAaABAsgBDcABAdoBBggBEAEYCQ&sclient=gws-wiz):- What is Random forest classifier in Python?A random forest classifier. A random forest is a meta estimator that fits a number of decision tree classifiers on various sub-samples of the dataset and uses averaging to improve the predictive accuracy and control over-fitting.

* *Data Sources and their formats*

The dataset which I use for model making is provide by Flip Robo Technology, The data set has 159571 rows and 8 columns



***About The Data Set:-***

|  |  |
| --- | --- |
| Variable | Definition |
| id | A unique id aligned with each comment text. |
| Comment\_text | It includes the comment text. |
| malignant | It is a column with binary values depicting which comments are malignant in nature. |
| Highly\_malignant | Binary column with labels for highly malignant text. |
| rude | Binary column with labels for comments that are rude in nature. |
| threat | Binary column with labels for threatening context in the comments. |
| abuse | Binary column with labels with abusive behaviour. |
| loathe | Label to comments that are full of loathe and hatred. |

* *Data Preprocessing Done*

For the purpose of the project the dataset has been preprocessed as follows:

* Checking shape of the dataframe
* Checking Missing Value
* Checking which type of data stored in each columns
* Text processing
* Plot Word cloud
* Visualization
* Describing the dataset
* Checking correlation and using heatmap for better understanding

We’ll now open a python 3 Jupyter Notebook and execute the following code snippet to load the dataset and remove the non-essential features. Receiving a success message if the actions were correctly performed.

* *Hardware and Software Requirements and Tools Used*

*Hardware:- Laptop*

*Software:- Anaconda*

*Libraries:-* *Numpy, Pandas, Matplot, Seaborn, nltk, etc*

**Model/s Development and Evaluation**

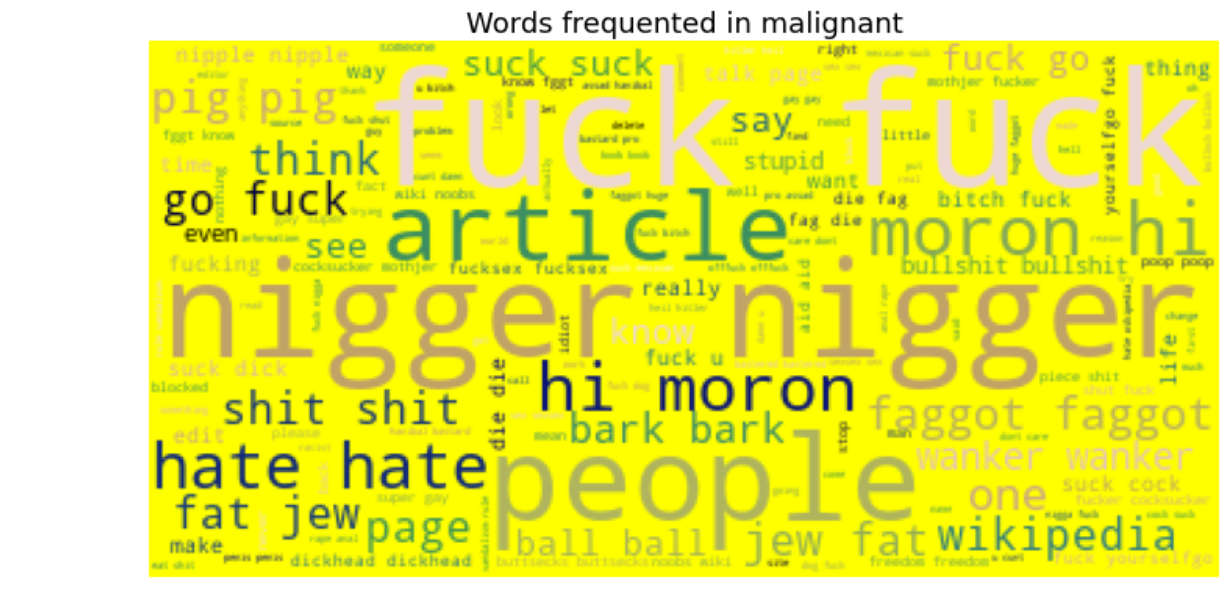
* *Identification of possible problem-solving approaches (methods)*
* *Missing Value Handling:*

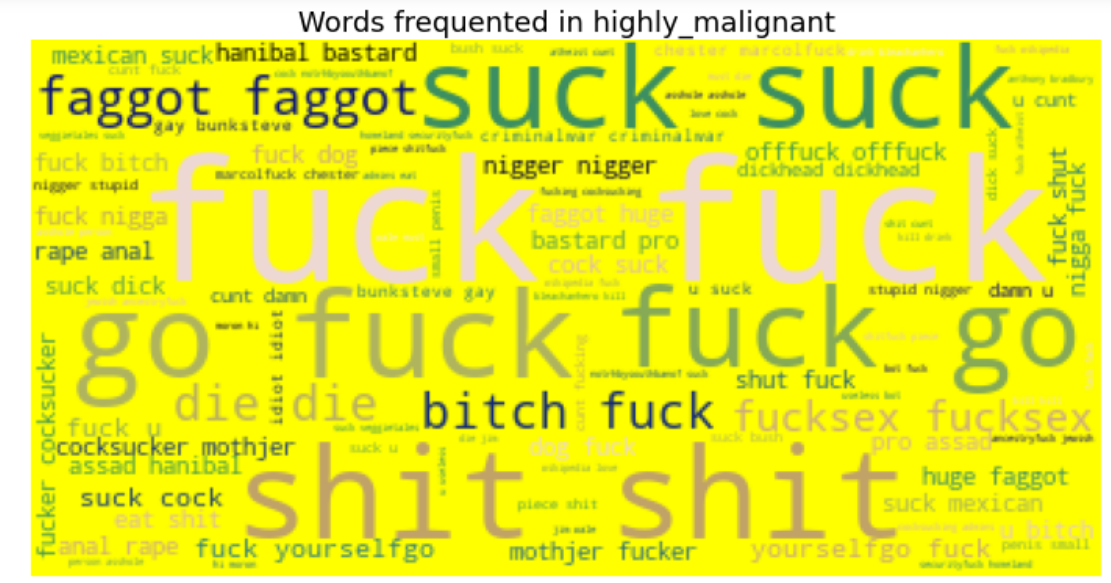
**There are 2 primary ways of handling missing values:**

* Deleting the Missing values:-Generally, this approach is not recommended. It is one of the quick and dirty techniques one can use to deal with missing values.

* Imputing the Missing Values:- There are different ways of replacing the missing values
* Replacing With Mean
* Replacing With Mode
* Replacing With Median, etc.
* We are free from missing value otherwise it is very important step for model building
* *Testing of Identified Approaches (Algorithms)*

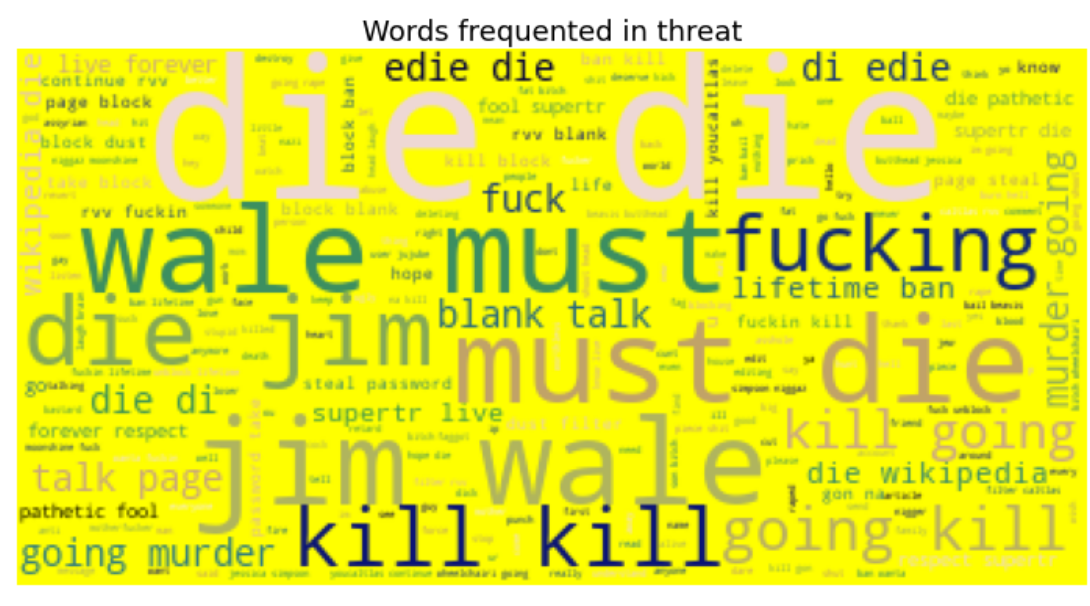
I have analysed the input output logic with word cloud and I have word clouded the sentenced that as classified as foul language in every category. A tag/word cloud is a novelty visual representation of text data, typically used to depict keyword metadata on websites, or to visualize free from text. It’s an image composed of words used in a particular text or subject, in which the size of each word indicates its frequency or importance.

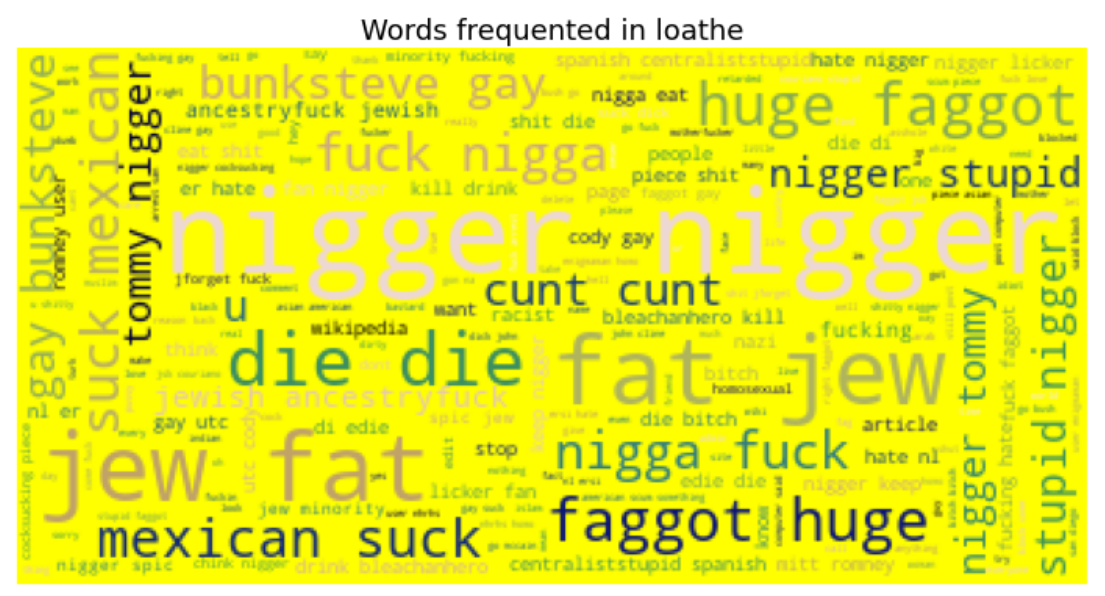




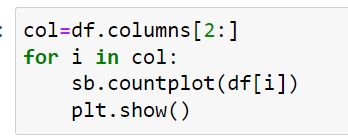


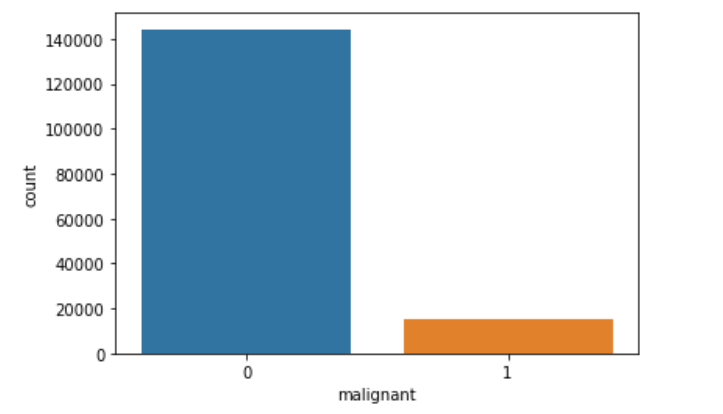


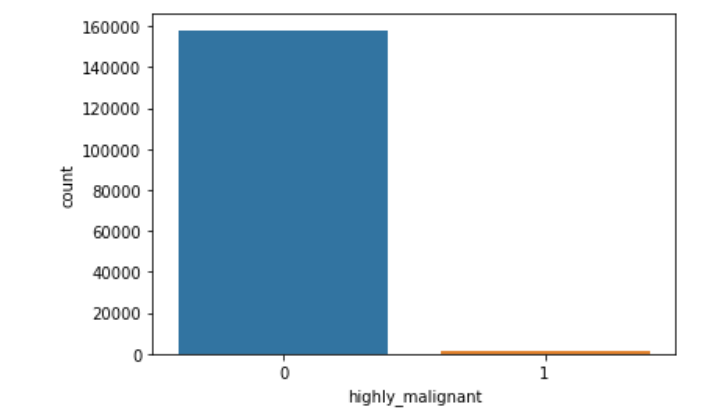


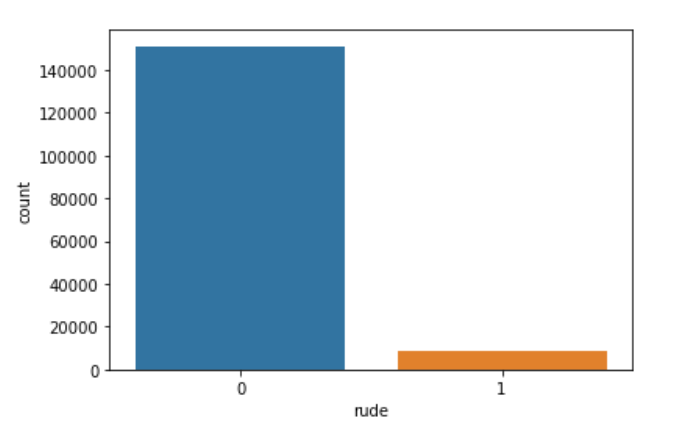


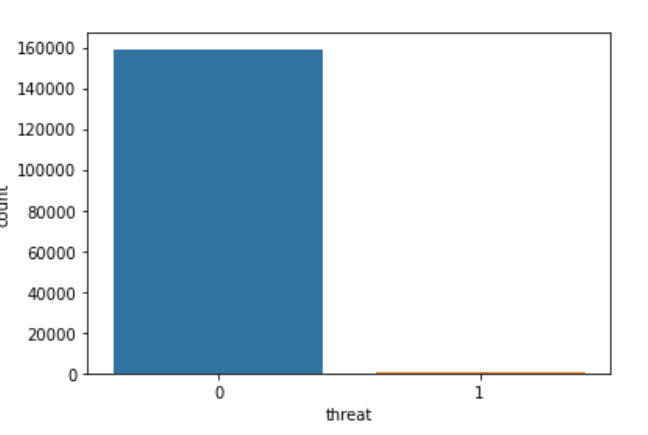
***Plotting count plot: -***

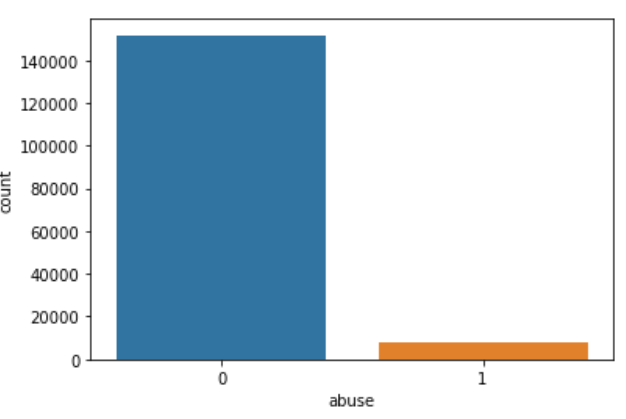


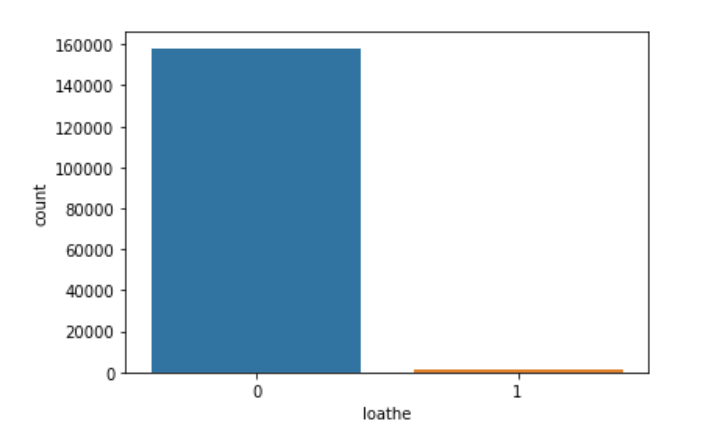








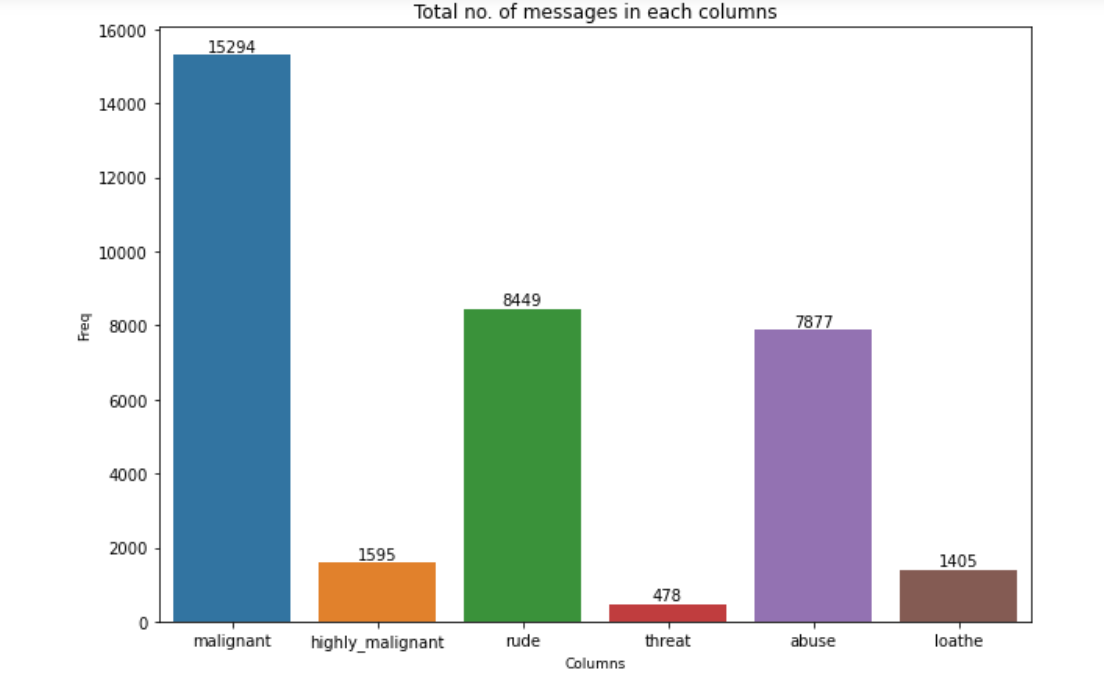




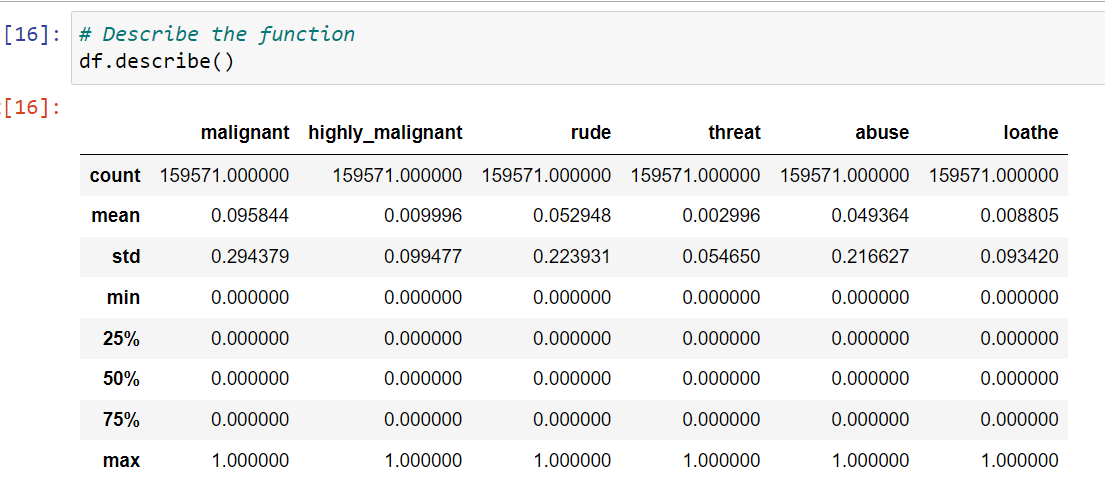
***Observation:-***

* Here in the first graph of malignant we can clearly observe that most of the messages are not malignant.
* In the second image we can clearly observe that there are very less highly malignant messages.
* Same in third picture there are few rude comments in the dataset.
* In 4th we can clearly see that there are very few cases/almost negligible of threat comments
* In 5th image we can clearly see that there are some messages with abusive language.
* While in the sixth image we can clearly see that there are very few cases of loathe messages.

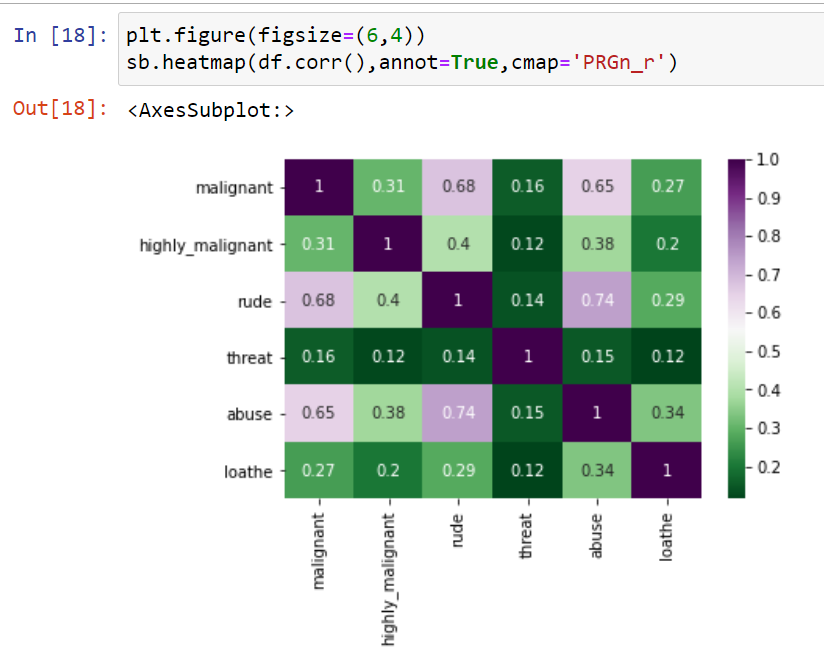
***Total number of messages in each column: -***

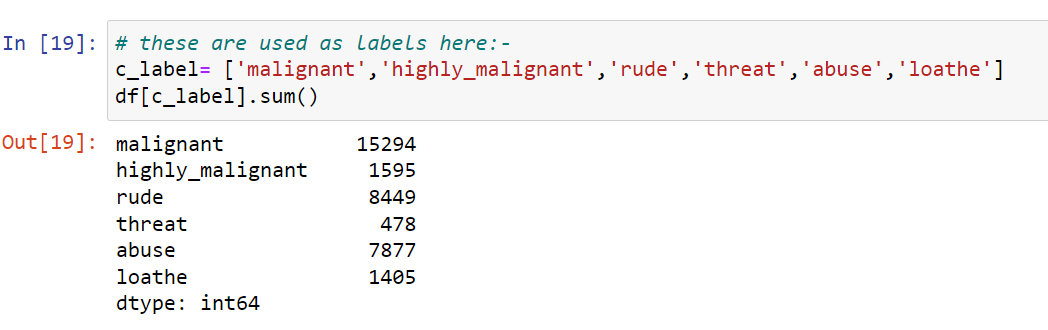


***Describe the dataset: -***

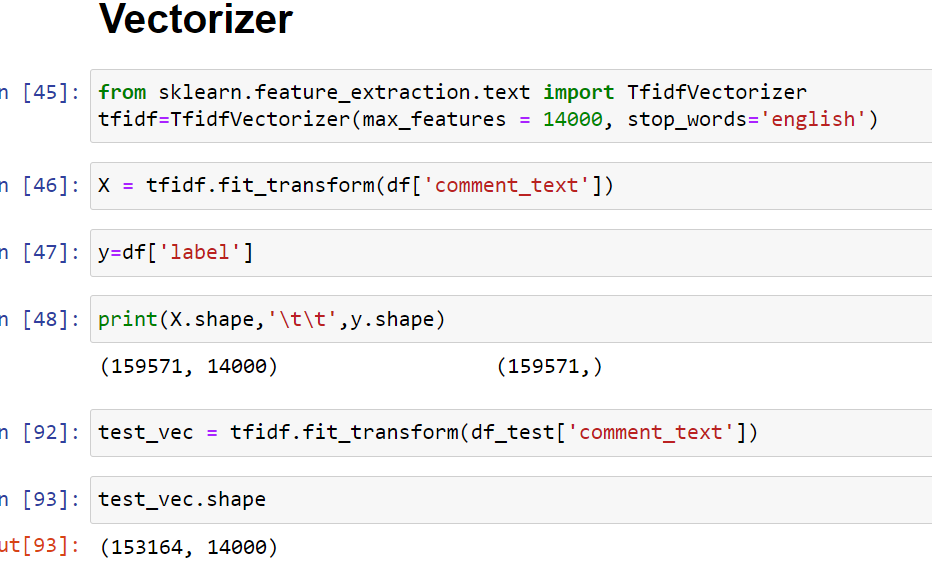


***Heatmap of the Dataset:-***

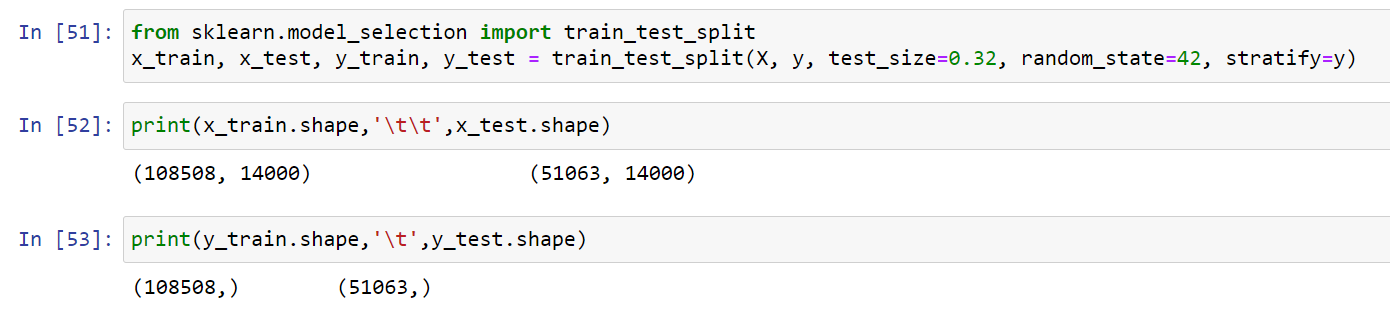




***Preprocessing:-***



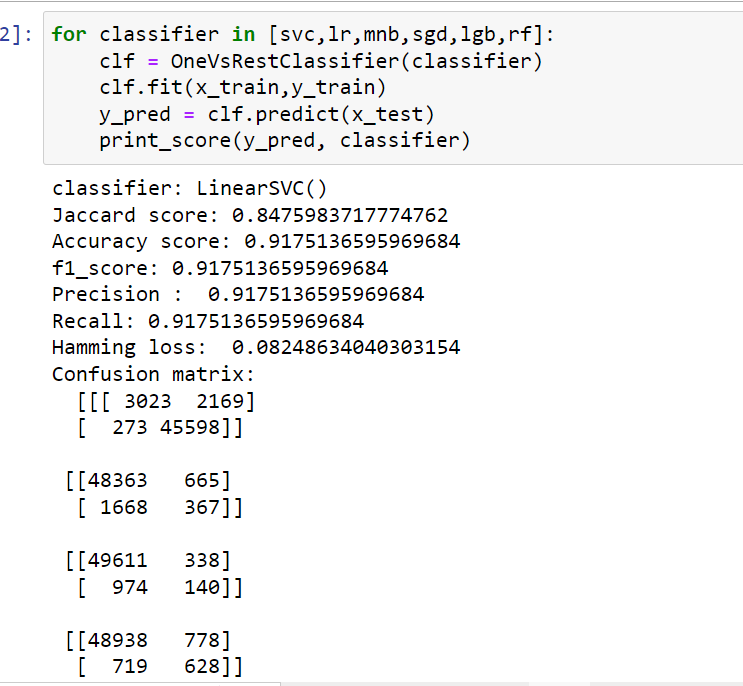
***Train test spilt: -***

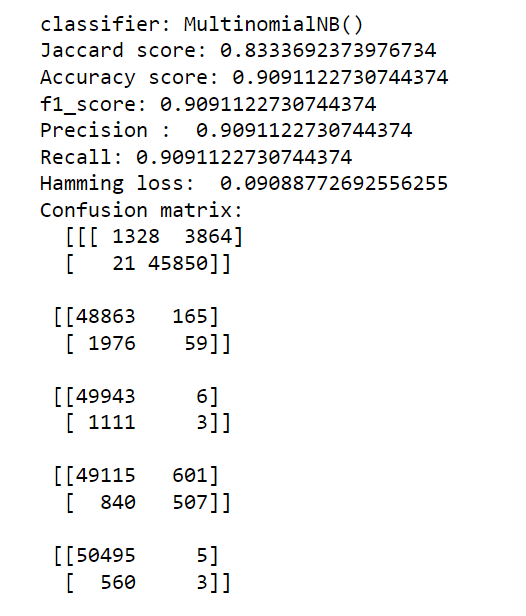


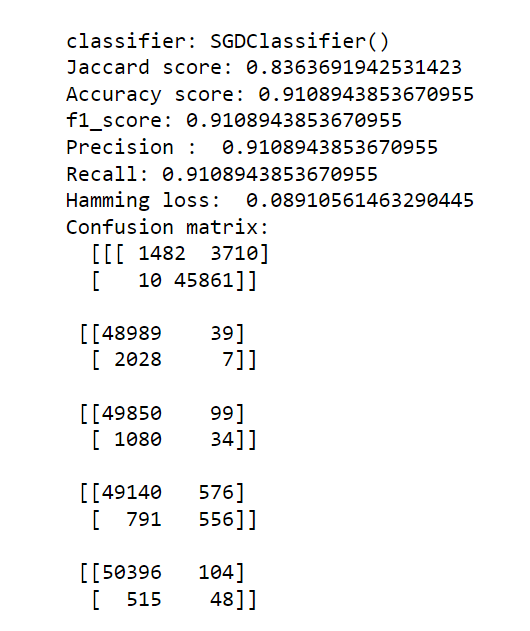
***Model selection: -***

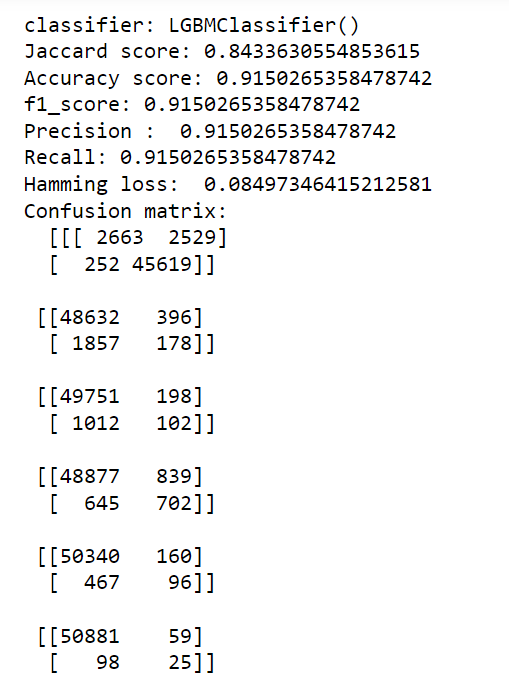


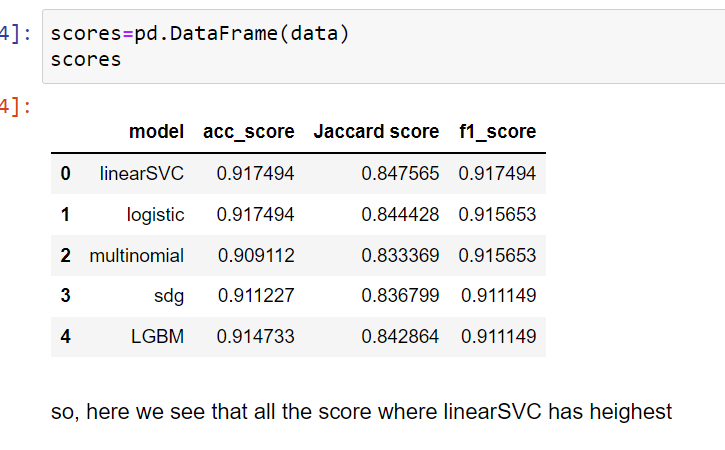
Output:-



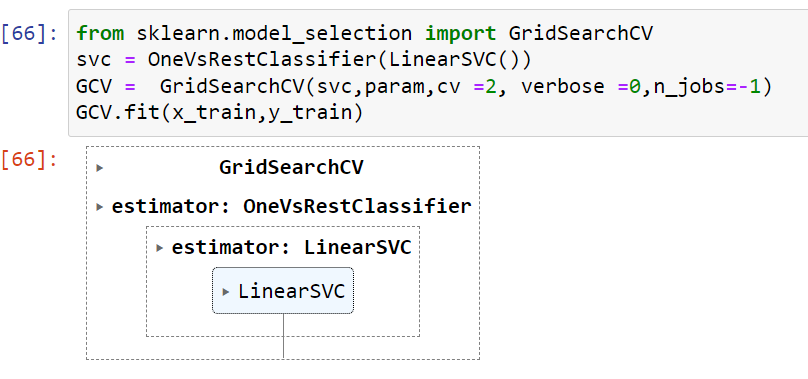




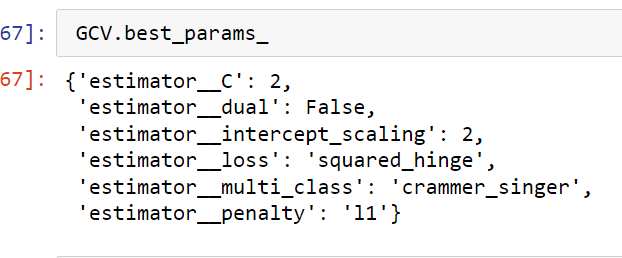




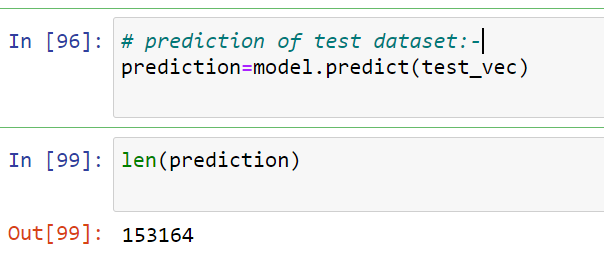
***Hyperparameter tuning:-***



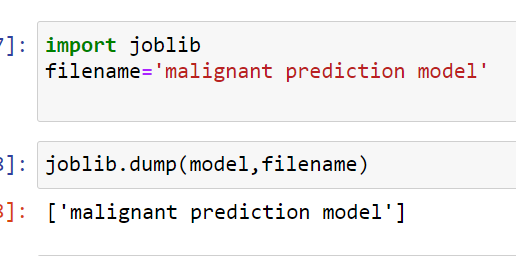
***Best parameter:-***



***Predating test dataset:-***



***Saving the model: -***



**CONCLUSION**

For any of machine learning project my suggestion is first you have to understand the problem on ground level. if you don’t allow yourself to work with diligence. if you don’ t work harder anything that you are doing or will do, not only in case of machine learning but also in life cycle would be futile. Maybe, my endeavour assists you whenever you will get stuck

* For future improvements, following step we thought to took-

* + Replacing model with a latest/different model

* + Using other robust datasets

* + More focus on NLP properties

* It would seem that better performance might be achieved if multiple learners were combined.