

**MALIGNANT COMMENTS CLASSIFICATION**

**Submitted By:**

**Pratiksha Potghan**

**Internship 30**

**Aknowledgement**

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 “Malignant Comments Classification” and I also want to thank my SME “Mohd Kashif” for providing the dataset to complete this project.

This project would not have been accomplished without their help and insights.

Working on this project was a great experience.

**Introduction**

* In this project, we have been provided with two datasets namely train and test CSV files. We will build a machine learning model by using NLP using a training dataset. And using this model we will make predictions for our test dataset.
* We will need to build multiple classification machines and learning models. Before model building, we will need to perform all data pre-processing steps involving NLP. After trying different classification models with different hyperparameters than will select the best model out of them. We will need to follow the complete life cycle of data science that includes steps like –
* 1. Data Cleaning
* 2. Exploratory Data Analysis
* 3. Data Pre-processing
* 4. Model Building
* 5. Model Evaluation
* 6. Selecting the best model
* Finally, we compared the results of proposed and baseline features with other machine learning algorithms. The findings of the comparison indicate the significance of the proposed features in cyberbullying detection.

**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.

**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. Decisio](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)n 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.
* [**KNeighborsClassifier:**](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 KNeighborsClassifier 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.
* [**RandomForestClassifier:**](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 Randomforestclassifier 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.

**Conceptual Background of 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 un-offensive, but “u are an idiot” is clearly offensive.

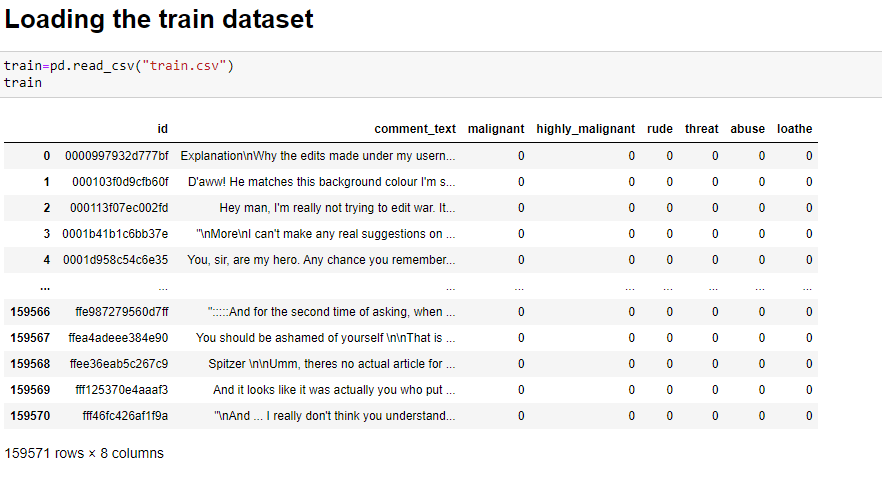
**Motivation for the Problem**

* The project was the first provided to me by FlipRobo as a part of the internship program. 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 main motivation is to build a prototype of online hate and abuse comment classifier which can be used to classify hate and offensive comments so that they can be controlled and restricted from spreading hatred and cyberbullying.

**Data Sources and their formats**

The dataset used for model making is provided by FlipRobo Technology, The data set has 159571 rows and 8 columns.

Dataset looks as below:



**About the dataset:**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  | | --- | --- | | ***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 Pre-processing Done**

For the purpose of the project the dataset has been pre-processed 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.

**Hardware and Software Requirements and tools used:**

Hardware : Laptop

Software : Anaconda, Jupyter Notebook

Libraries : pandas, numpy, matplot, seaborn, etc.

**Model/s Development and Evaluation**

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

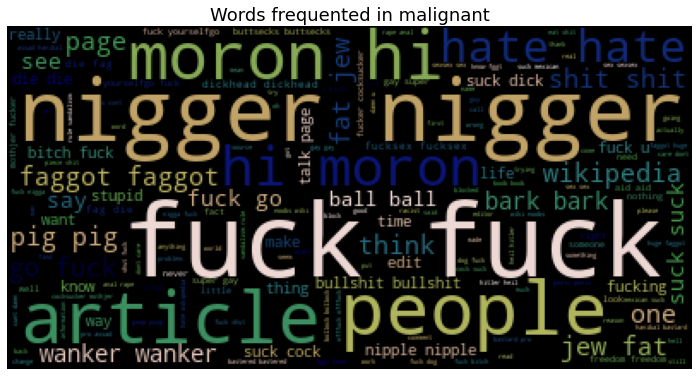
**There are 2 primary ways of handling missing values:**

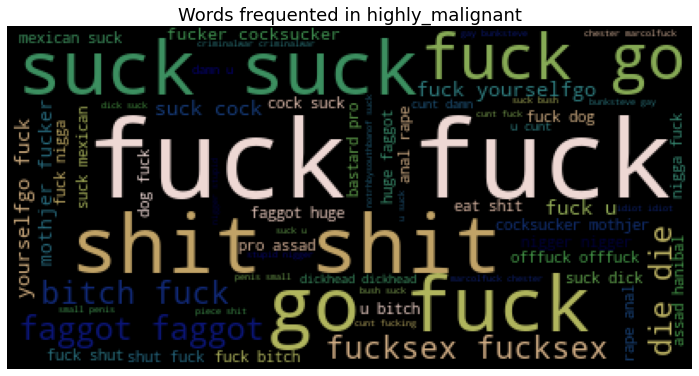
* Deleting the Missing values:-Generally, this approach is not recommended. It is one of the quick and dirty techniques one can use to deal with missing values.
* Imputing the Missing Values:- There are different ways of replacing the missing values

1. Replacing With Mean
2. Replacing With Mode ✓ Replacing With Median,etc.
3. We are free from missing value otherwise it is very important step for model building

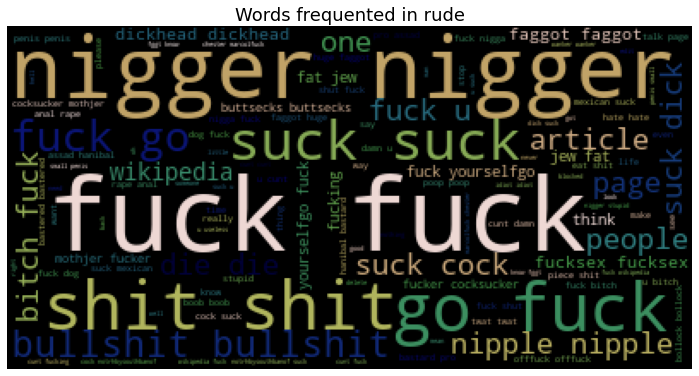
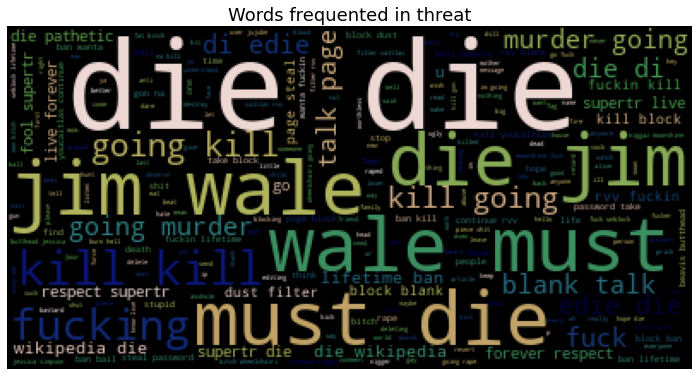
* **Testing of Identified Approaches (Algorithms)**

**Words Frequent in malignant and high malignant**

****

****

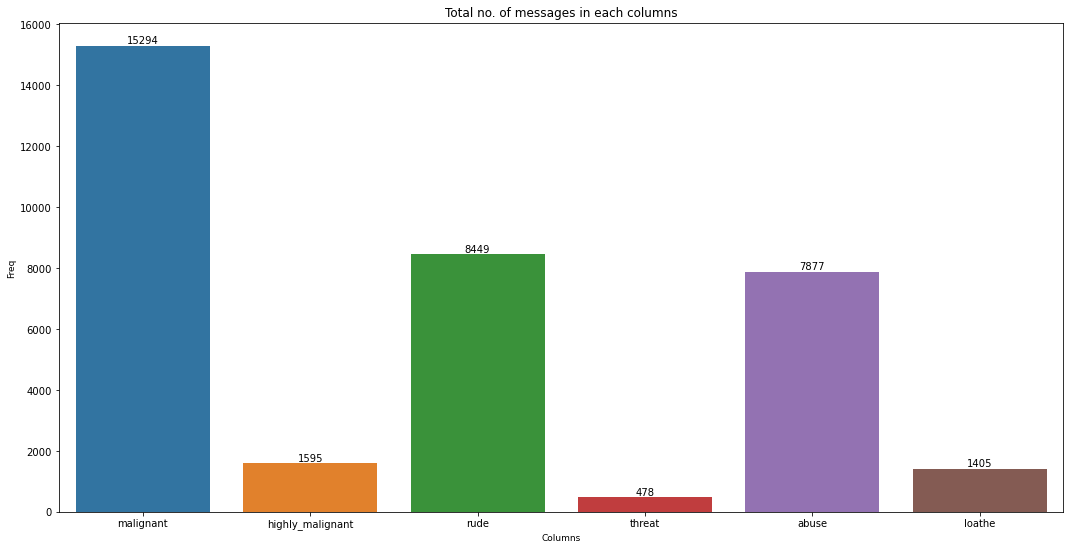
**Words Frequent in rude and high threat**

****

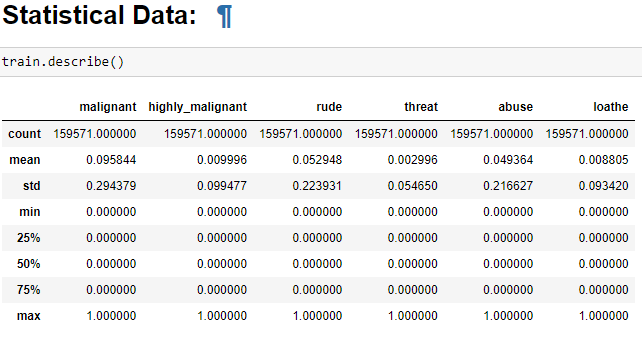
**Words Frequent in abuse and high loathe**

**** ****

**BarPlot of total no of messages in each column**

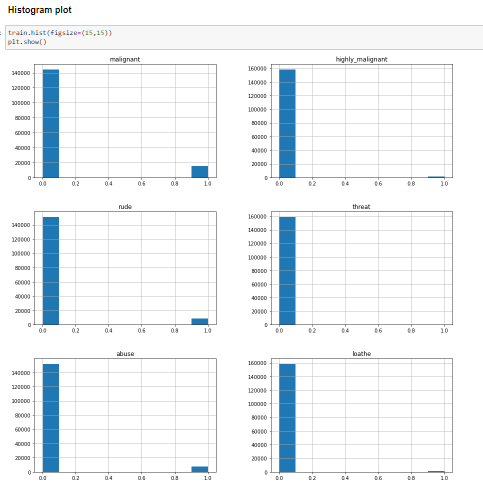
****

**Statistical Description**

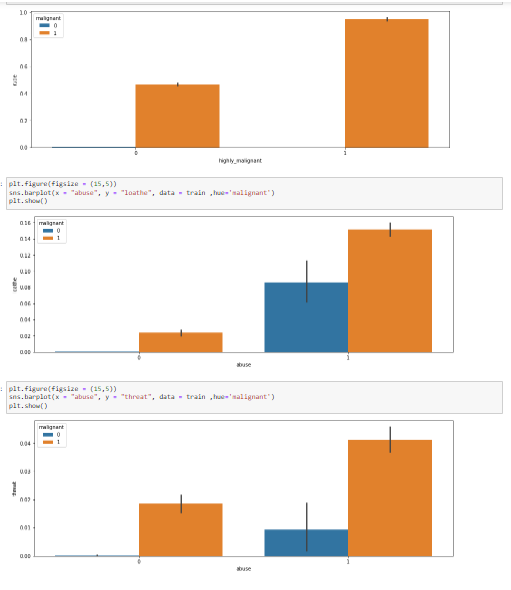
****

**Data Visualization**

* **Histogram**

****

* **Barplot**

****

* **Scatterplot**

****

**Correlation**

****

**Model Building**

* In this project there were 6 features which defines the type of comment like malignant, hate, abuse, threat, loathe but we created another feature named as “label” which is combined of all the above features and contains the labelled data into the format of 0 and 1 where 0 represent “NO” and 1 represents “Yes” . In this NLP based project we need to predict the multiple labels which are binary. I have converted text into feature vectors using TF-IDF vectorizer and separated our feature and labels. Also, before building the model, I made sure that the input data is cleaned and scaled before it was fed into the machine learning models.
* After the pre-processing and data cleaning I used remaining independent features for model building and prediction. The classification algorithms used on training the data are as follows.

1. Logistic Regression

2. MultinomialNB

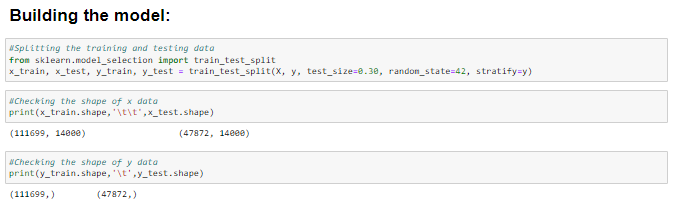
3. LineraSVC

4. Gradient Boosting Classifier

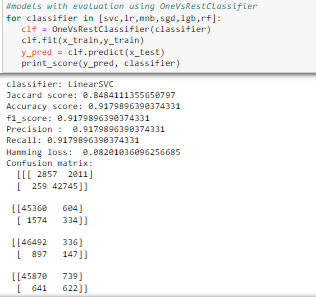
5. Decision Tree Classifier

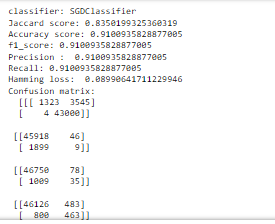
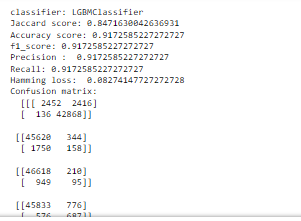
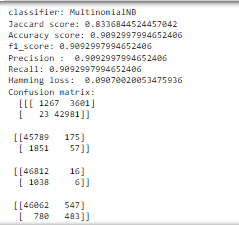
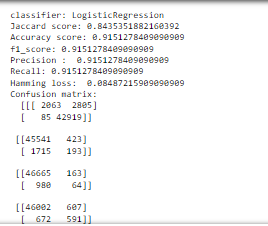
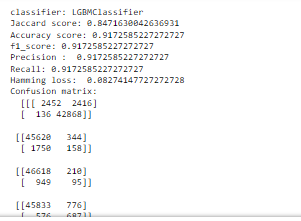
6. Ada Boost Classifier

****

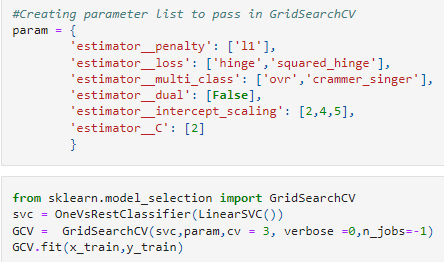
****

****

****

****

**Hyper parameter Tuning**

****

I have used 6 LinearSVC classifier parameters to be saved under the variable "parameters" that will be used in GridSearchCV for finding the best output. Assigned a variable to the GridSearchCV function after entering all the necessary inputs. And we used our training data set to make the GridSearchCV aware of all the hyper parameters that needs to be applied on our best model.

**Interpretation of the Results**

* LinearSVM and Random Forest models perform best in this project
* At first we get only 91.76% accuracy from LinearSVM But after parameter tuning we get 91.77% accuracy.There is no defference in accuracy score after parameter tuning.
* Using hyper parameter tunning we can improve our model accuracy, But here in this model the accuracy did not increased.
* It is always advised to all of us that atleast we need to use 5 Algorithm in order to figure out which one is performing best among them and we choose that one and we send that for hyper parameter tuning to know that best parameter.

**Future Work**

For future improvements, the following step we thought to take-

* + Replacing model with a latest/different model
  + Using other robust datasets
  + Predicting results on different attributes
  + It would seem that better performance might be achieved if multiple learners were combined.

**Conclusion**

* When I was working on this project, many complications were involved. There are many variables/attributes to consider in determining our target value, we need a lot of calculating power to get a near 100% accuracy result. And it is very difficult to accurately predict any comment good or bad.
* For any machine learning project my suggestion is first you have to understand the problem at 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 the case of machine learning but also in the life cycle would be futile. Maybe, my endeavor assist you whenever you will get stuck.