

**Malignant Comment Classifier**

Submitted by:

Sahib Hussain

**ACKNOWLEDGMENT**

This project would not have been possible without the support of many people. Many thanks to my SME Keshav Bansal who gave suggestion how to proceed in building this model and helped make some sense of the confusion. Also thanks to my Internship Company FlipRobo Technologies Pvt. Ltd for giving me this opportunity to work on such a project which can help the industry to grow and achieve success.

**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.

* Conceptual Background of the Domain Problem

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.

* Review of Literature

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.

* Motivation for the Problem Undertaken

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

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’.

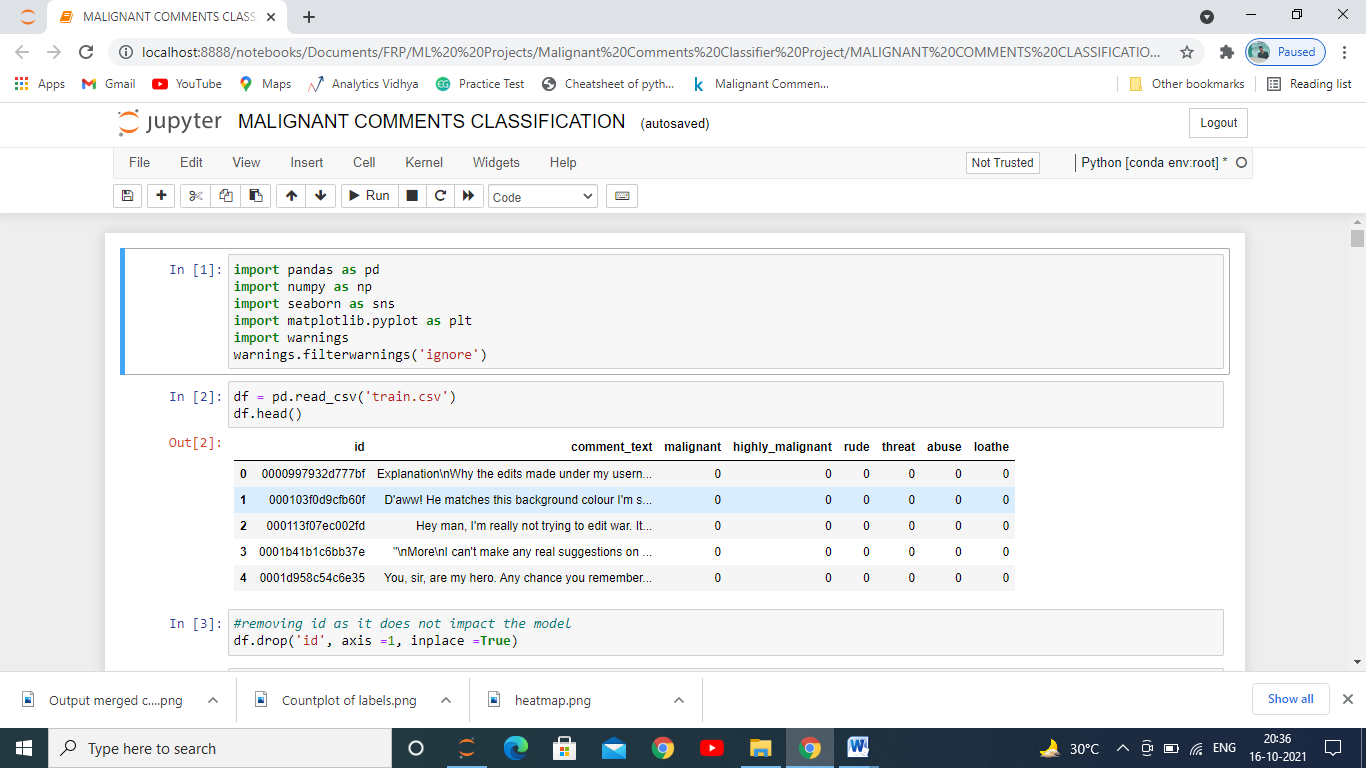
By using shape we saw the Number of rows and columns in the dataset.

By using heatmap we found the collinearity among various features which we will see later in this documentation.

To make this model we used Naïve Bayes Classifier, Passive Aggressive Classifier, Logistic Regression, Decision Tree Classifier, Random Forest Classifier and XGB Classifier algorithm and used the best one out of these by evaluating the necessary metrics, we predicted the test set.

* Data Sources and their formats

The sample data is provided to us from our client database.



Above figure shows us that there are 8 columns in it.

It has one independent Variable only which is our comment\_text and rest of the columns are dependent.

* Data Preprocessing Done

The clean the data we had to delete id as it would had matter in getting the predictions.

A common label was created by taking out the values of all the dependepent variables for building the model and predictions.

Then each of the comments where tokenized into words for better model building.

After tokenization, Lemmatization was performed on these words for each and every comment. It is the process of grouping together the different inflected forms of a word so that they can be analysed as a single item.

Also stop words technique was used to clear out the words which does not add any meaning to a sentence for building this model. They can safely be ignored without sacrificing the meaning of the sentence. For example, the words like the, he, have etc.

On these sentence we then performed **re** known as regular expression. It specifies a set of strings that matches it; the functions in this module let you check if a particular string matches a given regular expression (or if a given regular expression matches a particular string, which comes down to the same thing).

Finally at the end, we used **TfidVectorizer** for every comments which transforms text to feature vectors that can be used as input to estimator.

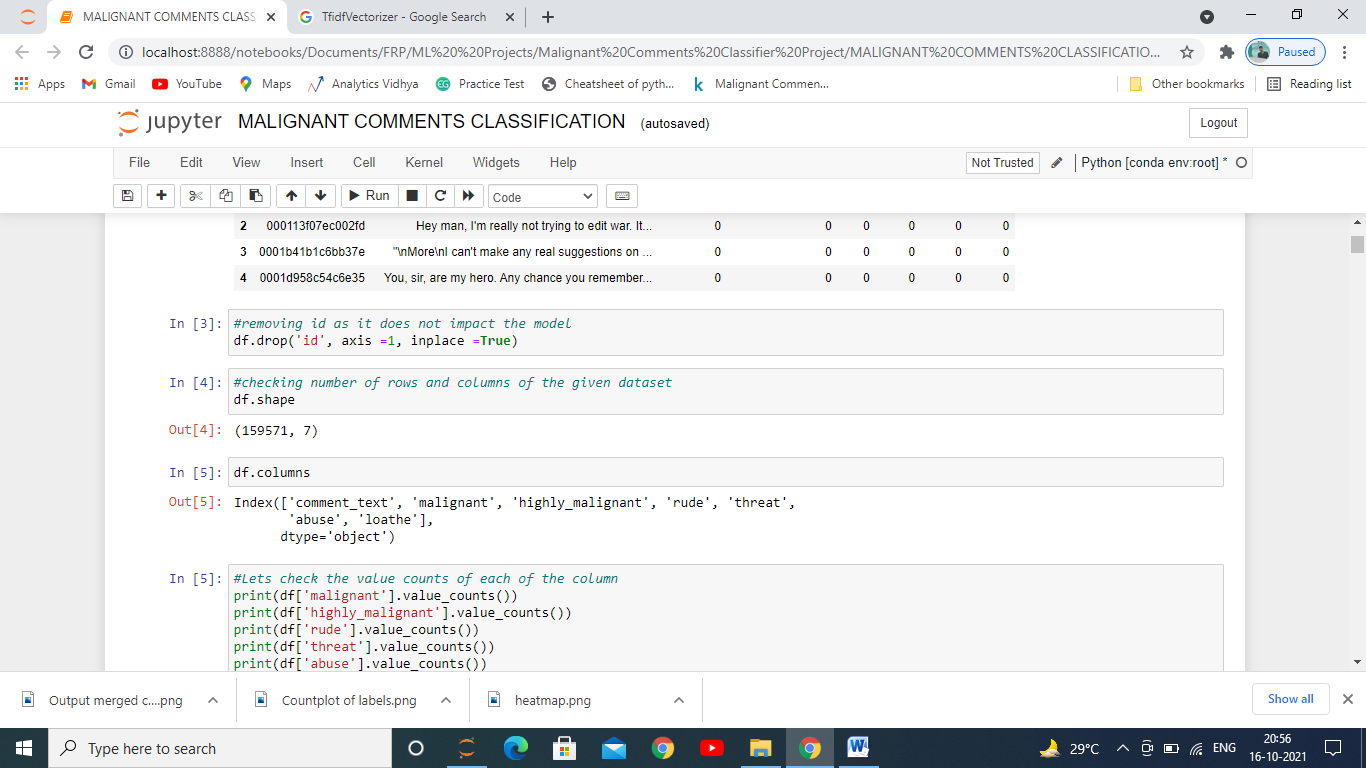
* Hardware and Software Requirements and Tools Used

**Software used** : In this we used Jupyter Notebook from the Anaconda to build the model, Miscrosoft powerpoint to make the powerpoint presentation and Miscrosoft word for documentation.

**Libraries used**: Pandas to read the dataset, re, stopwords, NLTK, word tokenizer, lemmatizer, tfidvectorizer, Matplotlib and Seaborn to analyse the data and clean the data and Scikit-Learn to build our models.

**Model/s Development and Evaluation**

* Identification of possible problem-solving approaches (methods)

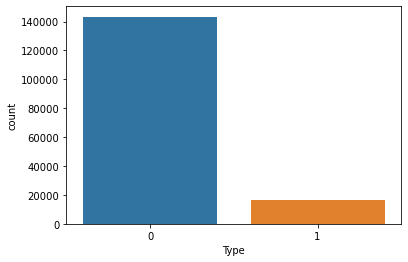


As we can see that the dataset was huge there were 159571 rows and 7 columns which was after removing unnecessary features. 6 of the 7 columns were dependent one. We clubbed the dependent ones into a single output which had 0 for toxic and 1 for non-toxic comments.

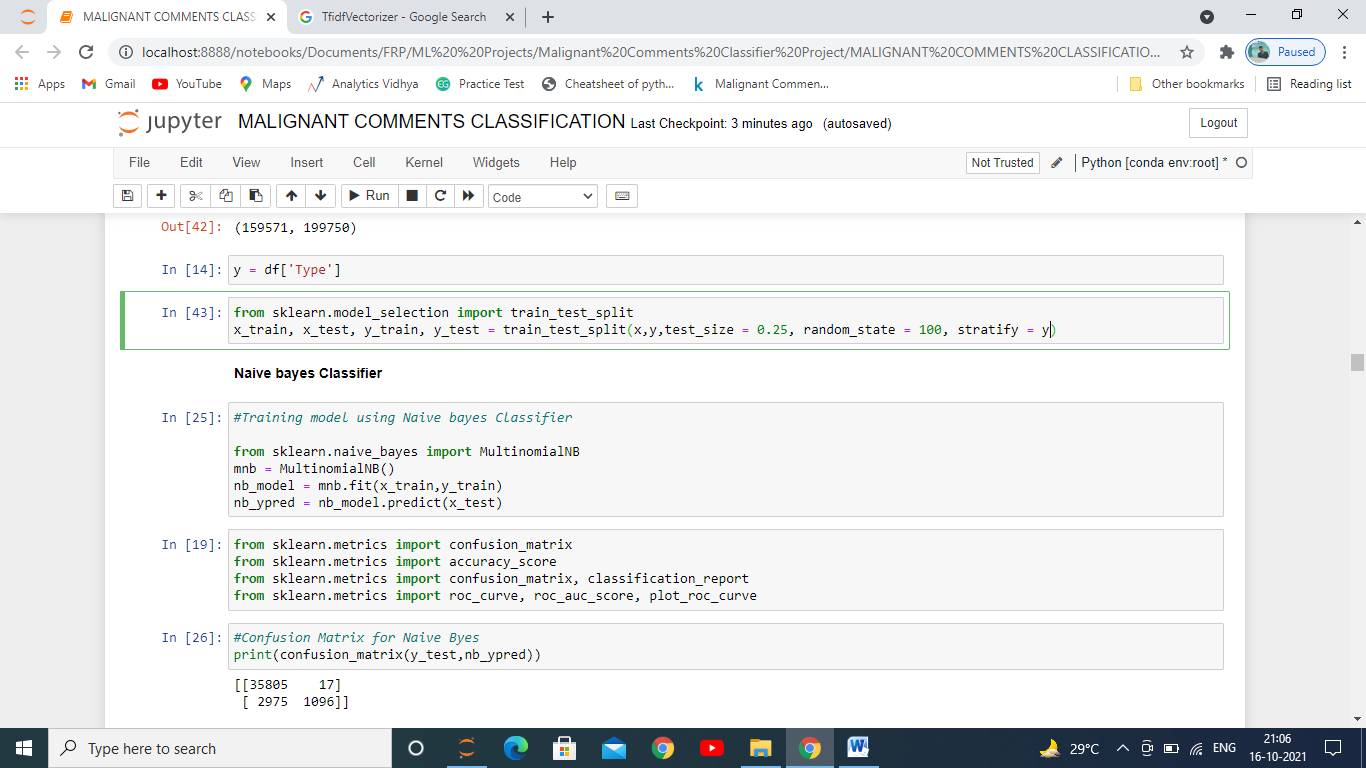
The comments were than pre-processed and clean to build a better model using the steps mentioned earlier.

We plotted heatmap to see if there is collinearity among the original dependent variables.

Also we saw that there was lot of imbalance in the label (Dependent Variable) so we had to overcome it again because model in that case would have predicted mostly the highest one, it was done by using Stratify method used as a parameter in train-test split. We can see below that mostly the label had one if it hadn’t been balanced the model would have given only that prediction everytime.

****

* Testing of Identified Approaches (Algorithms)

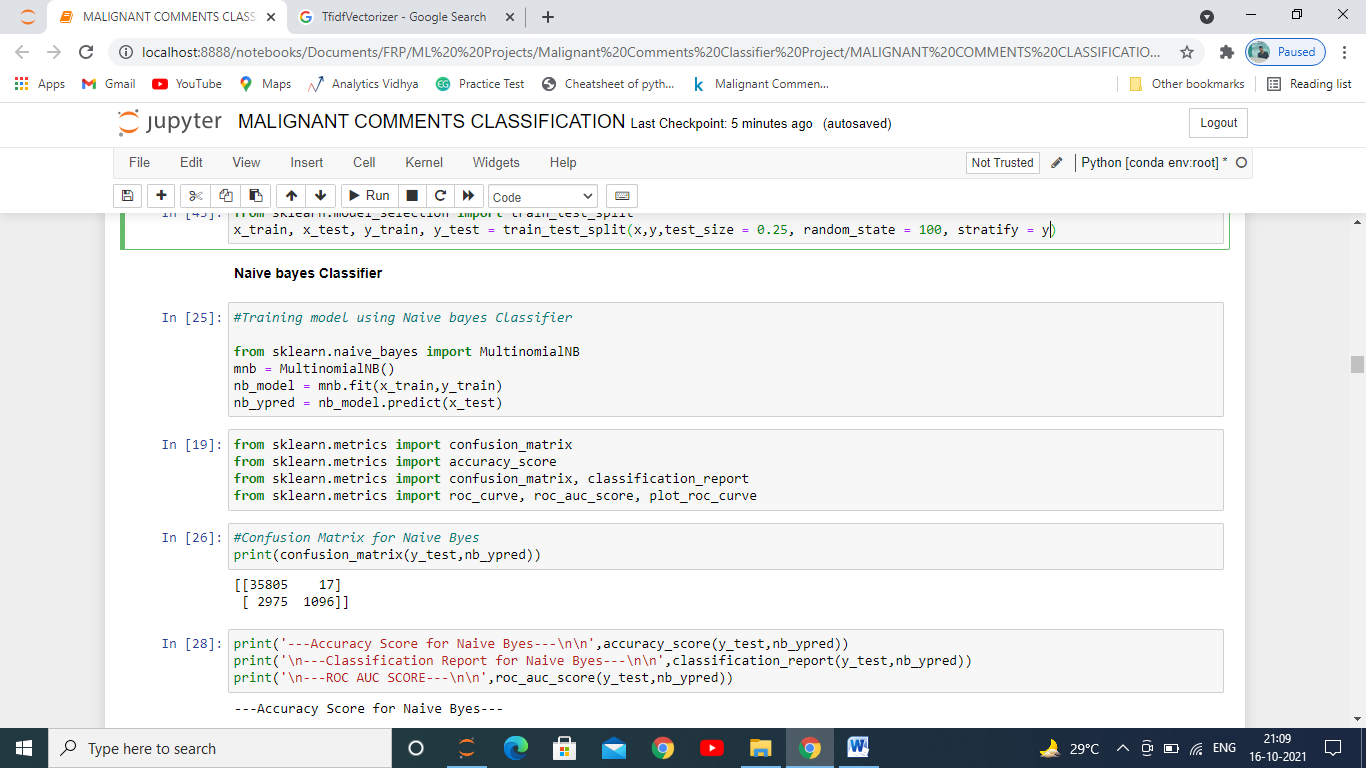


We first imported the train test split as you can see in the above image.

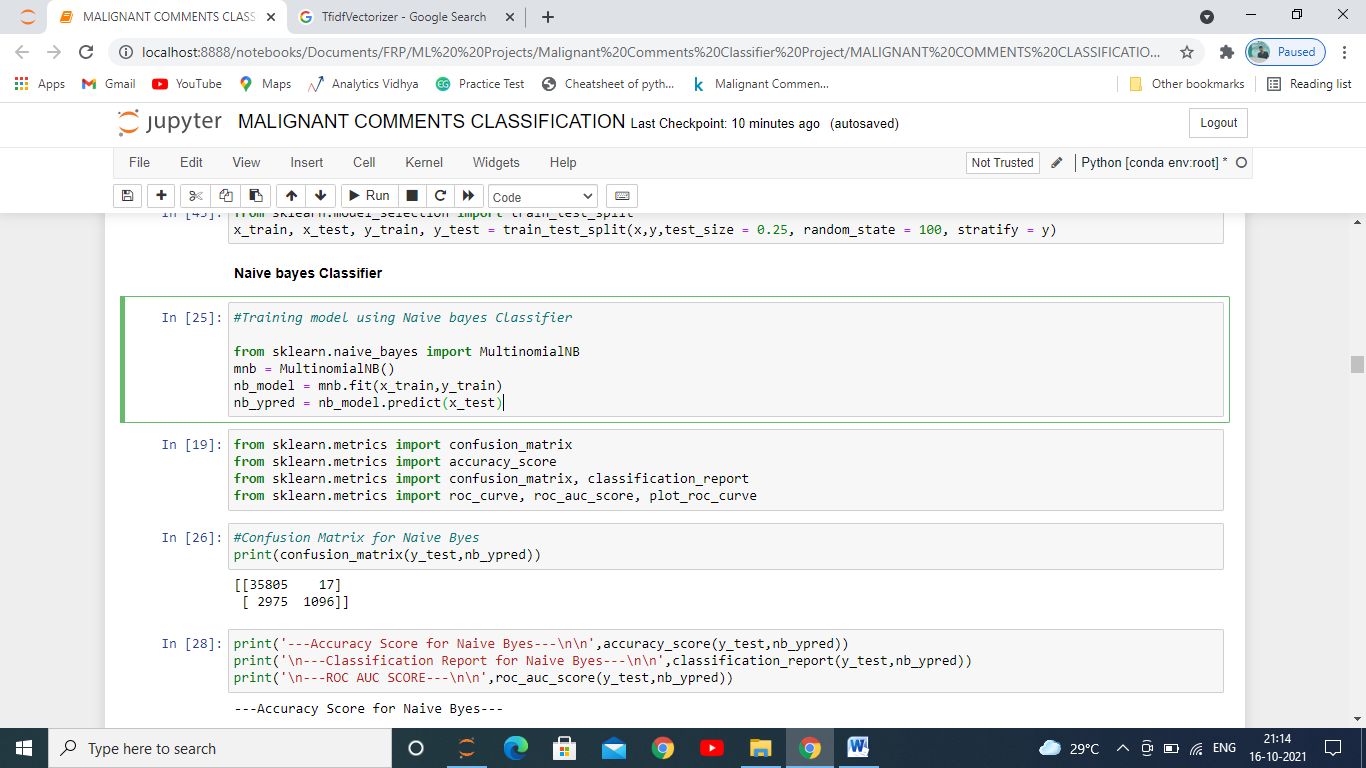
The whole dataset is broken down into 4 parts each for train and test, 2 parts for traning the model and 2 parts for testing, in it we kept 75% of the data for training and remaining 25% for testing the data. We used stratify because there was imbalance in the label. It will take equal proportion of each output to train and test the model.

* Run and Evaluate selected models

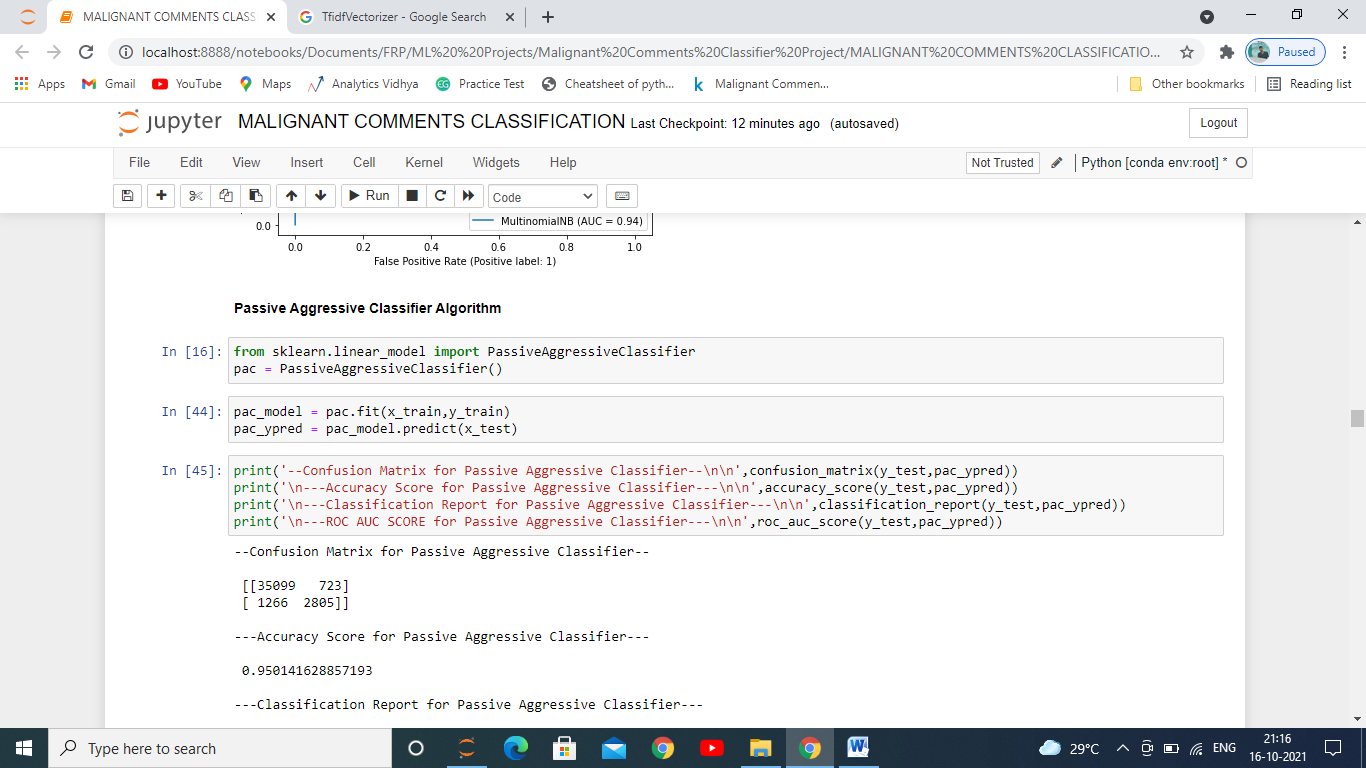
**Evaluation Matrics:**



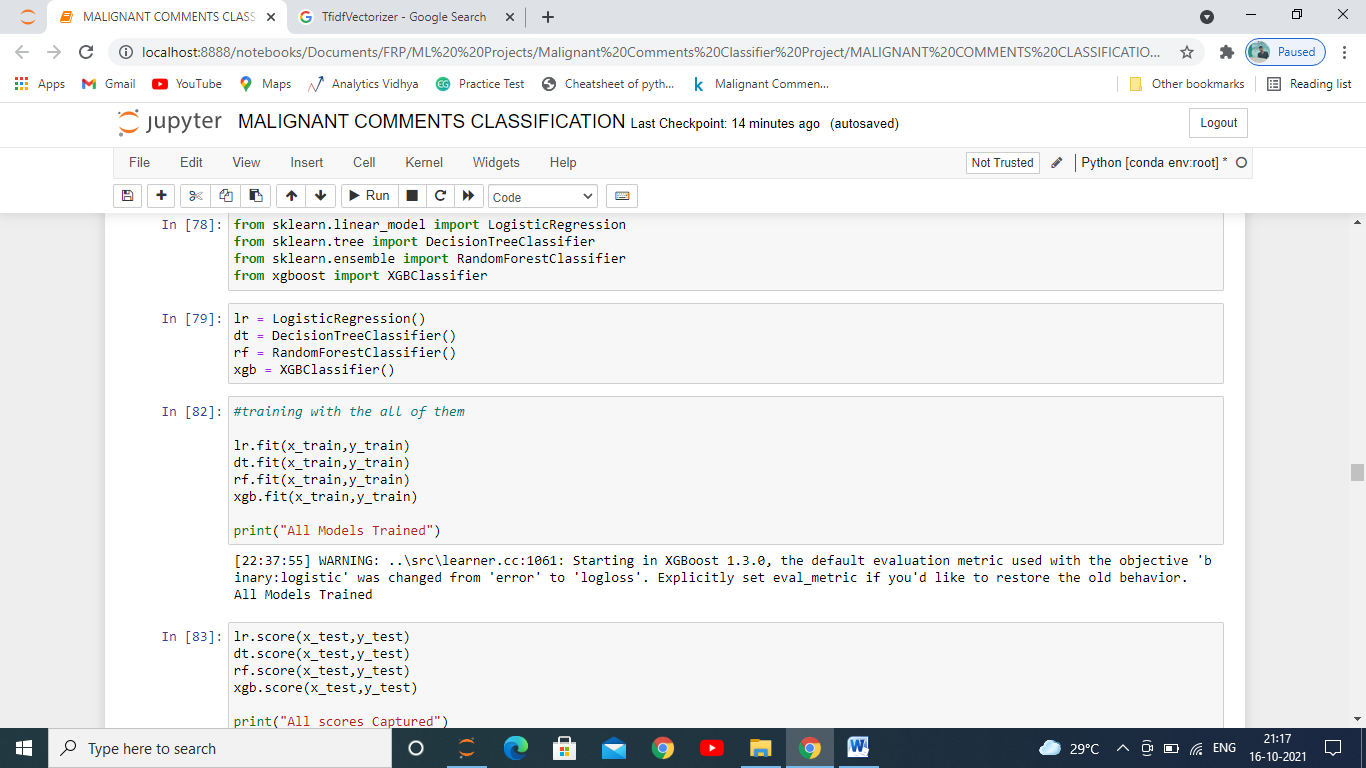
**Naïve Bayes Classifier:**



**Passive Aggressive Classifier:**



**Other Classification Models:**

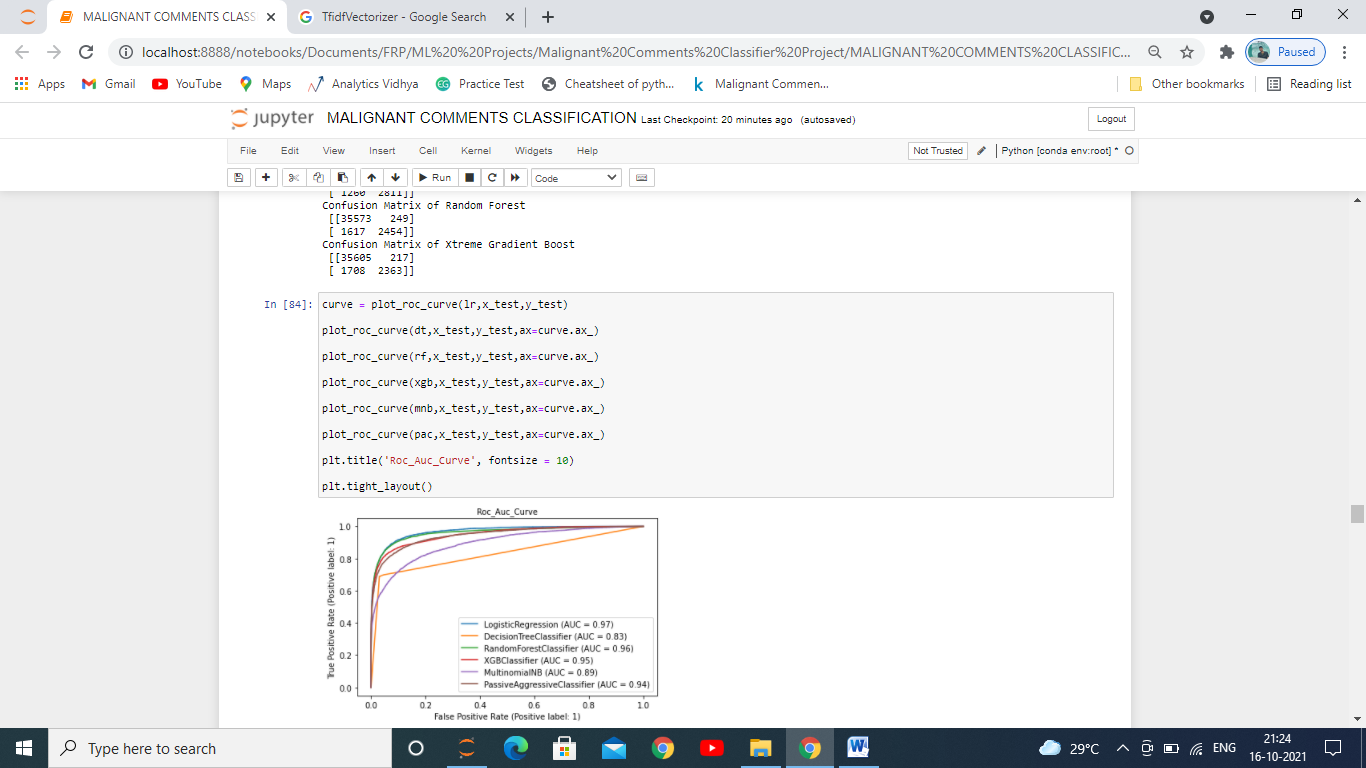


First we imported various evaluation metrics used for evaluating a classification model since the label had to predict to whether the comment given by a user is toxic or not, then imported various models we are used for building classification model as you can see we imported 6 models: Naïve Bayes Classifier, Passive Aggressive Classifier, Logistic Regression, Decision Tree Classifier, Random-Forest Classifier, Xtreme Gradient Boost Classifer then we stored them in variables for each one of them which we are going to use further in building the model.We then trained it with x\_train and y\_train which we got from the dataset after splitting it in train and test set. This train set composes of 75% of the data which is going to be used to study the data by the model to predict the test set.

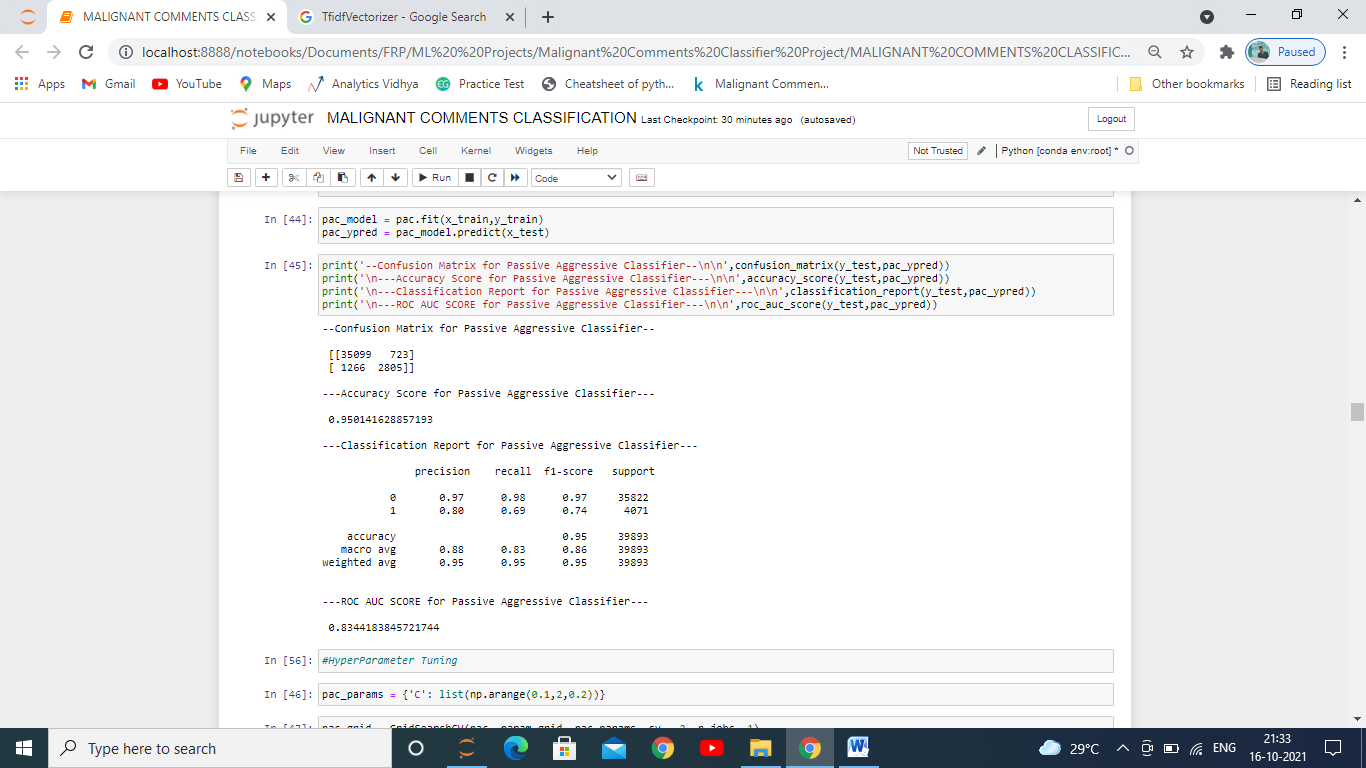
Then we captured the scores of each with our test set divided in 2: x\_test, y\_test.

After evaluating we saw that Passive Aggressive Classifier and Logistic regression were giving us the highest accuracy however the type 1 and type 2 error which we got with the help of confusion matrix was lower in case of Passive Aggressive Classifier.

* Key Metrics for success in solving problem under consideration



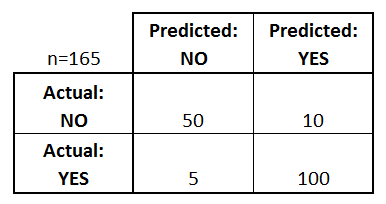
The above image is showing us the area covered by different models and the highest area covered was by Logistic Regression as we can see in the above picture.



In the above image we can see that the accuracy score given by the Passive Aggressive Classifier is 5% and has the roc\_auc\_score of 83.44%. Accuracy is one metric for evaluating classification models. Informally, **accuracy** is the fraction of predictions our model got right. Formally, accuracy has the following definition:

Accuracy=Number of correct predictions/ Total number of predictions

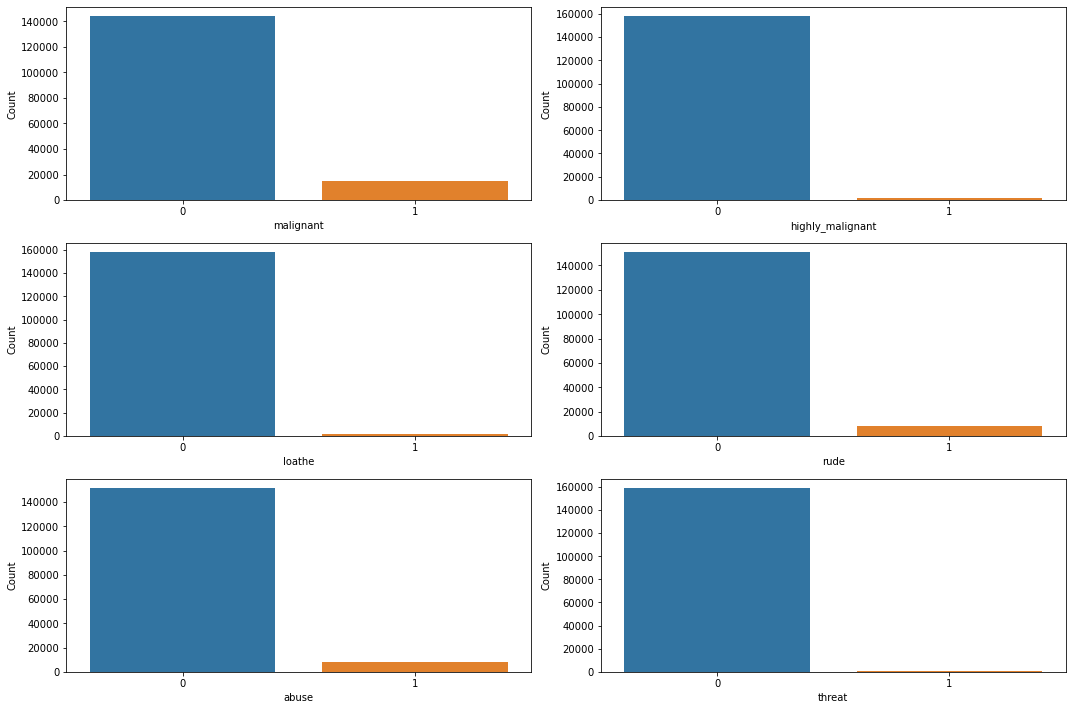
We also evaluated it with A confusion matrix is a table that is often used to **describe the performance of a classification model** on a set of test data for which the true values are known as shown in the below figure.



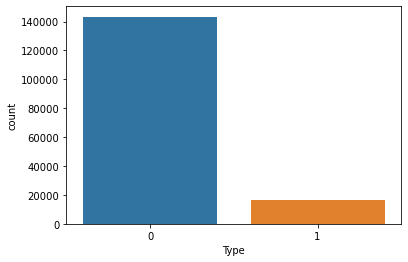
We saw that the model with Passive Aggressive classifier is giving us AUC-ROC score of 83.44%. AUC - ROC is a performance measurement for the classification problems at various threshold settings. ROC is a probability and AUC represents the degree or measure of separability. It tells how much the model is capable of distinguishing between classes. Higher the AUC, the better the model is at predicting 0 classes as 0 and 1 classes as 1.

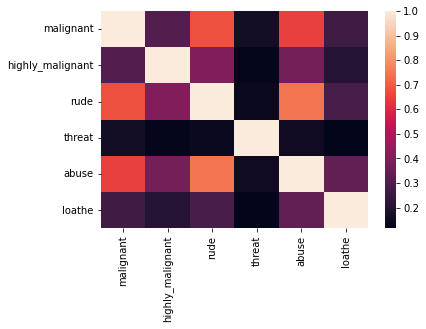
We also printed the Confusion metric for evaluating the model.   
A classification report is **a performance evaluation metric in machine learning**. It is used to show the precision, recall, F1 Score, and support of your trained classification model.

* Visualizations

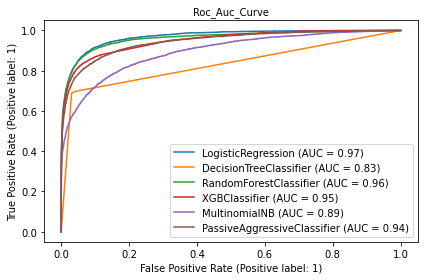


Above is the count plot of each of the output which we combined into one dependent variable.





It is a heatmap which shows us the colinearity amoung all the dependent variables , how much they are co-related near to 1 has high colinearity among them.



It showed us which of the model covered most of the area(AUC- Area Under Curve).

* Interpretation of the Results

The dataset is huge and it was regarding the comments given by users on social media platform.

After building the we found that Passive Aggressive classifier gave us the best result taking into consideration all the metrics scores and parameters, this model had covered most of the Area.

**CONCLUSION**

* Key Findings and Conclusions of the Study

This model helped us to analyse the comments type and which type of comments fall under which category.

* Learning Outcomes of the Study in respect of Data Science

While doing this project we observed that data visualization plays a key role in analysis the data and further processing it to build a model which has very less error and more accuracy. Analysis helps us to find out which are the feature mostly affecting the label and has more weightage and on this basis the data needs to be cleaned so that the data which we get to build our model gives us the best results. Shape is used to find the number of rows and columns, info for getting what are the data types of features and whether there are any null values in it or not, if there are this needs to be either filled up or removed from the dataset so that it does not build an erroneous model for us, describe has also been used to check the quartiles and mean, standard deviation, min value and maximum value of each of the features.