Project Report on
Malignant Comment Classification

BY:

MAYANK SHUKLA

Background & Introduction

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:

-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.

This project is more about exploration, feature engineering and classification that can be done on this data. Since the data set is huge and includes many categories of comments, we can do good amount of data exploration and derive some interesting features using the comments text column available.

You need to build a model that can differentiate between comments and its categories.

Refer to the data set file provided along with this.

Business Problem

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.

Approach to Data Cleaning

* First we check the information of the given dataset because it tells that how many rows and columns are present in our dataset and data type of the columns whether they are object, integer or float.

- Drop duplicates rows if present in dataset.+Then we check for the null values present in our dataset.
- If null values are present then fill it via mean, median or mode. Or also you can remove that rows but kindly check it properly.
- After that we check the summary statistics of our dataset. This part tells about the statistics of our dataset i.e. mean, median, max value ,min values and also it tell whether outliers are present in our dataset or not.

Approach to Data Cleaning (Cntd...)

*We also check the correlation of our dataset to check the correlation of the columns with each other. If columns are highly correlated with each other let's say 90% or above then remove those columns to avoid multi coli-nearity problem.

* We extract data from date column and make new columns like day, month and year to see the outcomes with our target column that is label.

* We delete the pcircle column because it has only one unique value that tells that collected data is only for one circle.

* We cannot remove outliers because more than 20% of our data are removed.

Visualization.

* We plot correlation matrix via heatmap to see the correlation of the columns with other columns.

* We also visualize the correlation of columns with target column via bar graph to see which column is highly correlated with target column.

* We see the number of defaulter and non defaulter customers with the help of count plot.

* We plot histogram to displays the shape and spread of continuous sample data.

* We also see the customers labels i.e defaluter/Non-defaulter according to date and month with count plot.

* We also see the distribution of the data with the help of distribution plot whether it is left skewed or right skewed.

Modelling part

*We know that this is classification problem so we use accuracy score, classification report and confusion matrix as our evaluation matrix. We also see the AUC score and also plot the AUC_ROC curve for our final model.

* As we know this dataset is imbalance so we don't too much focus on accuracy score . We see the precision and recall value along with f1_score.

* First we see the result without doing any sampling technique and for that I Logistic Regression Model

* And then we use KNeighborsRegressor where the input consists of the k closest training.

* At the third time we use DecisionTreeRegressor model, which is a model of decisions and all of their possible results, including outcomes, input costs, and utility.

* Then we use RandomForestRegressor Model, which performing both regression and classification tasks with the use of multiple decision trees and a technique called Bootstrap and Aggregation, commonly known as bagging.

* Finally we use AdaBoostRegressor and ExtremeGradientBoostingRegressor model, which is used to generate an ensemble model by combining the weak learners or weak predictive models.

Conclusion

So here **'RandomForestRegressor'** is the best model out of all model tested above which gives score, Test accuracy is 0.955610795454545454, which is the greatest.