

**Malignant Comments Classification**



Submitted by:

**Pradeep Kr. Singh**

**ACKNOWLEDGMENT**

In preparation of this Project, I had to take the help and guidance of some NLP articles and Kaggle advises by the experts posted there, who deserve my deepest gratitude. As the completion of this Project gave me much pleasure, I would like to show my gratitude towards Kris Naik tutorial channel on YouTube, sessions from Shankar sir articles on towardsdatascience.com and medium.com for giving me a good guidelines for this Project throughout start to end using some sample data. I would also like to expand my gratitude to all those who have directly and indirectly who helped me to use wide thinking to explore here in dataset.

In addition, a thank you to my mentor Astha Mishra, who introduced me to the Methodology of work, and who support always wherever I stuck ever. I also thank FlipRobo Technologies for providing such an opportunity to work on various types of projects which gradually improve my vision to apply for the ease to solve the complexity in the data.

**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.

* Conceptual Background of the 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 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.

* Review of Literature

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.

* Motivation for the Problem Undertaken

These comments which here are referred as malignant here represents the human behaviour as well as tendency of the individual in the society. As a data scientist, the aim is to filter out these words from the social media from flashing of and blocking the user who does these types of act on the regular basis.

**Analytical Problem Framing**

* Mathematical/ Analytical Modeling of the Problem

Here we are dealing with one main text columns which held some importance of the data and others shows the multiple types of behaviour inferred from the text. I prefer to select on focus more on the words which has great value of importance in the context. TF-IDF is the NLP terms I am going to apply on text columns. This converts the important words a proper vectors with some weights.

* Data Sources and their formats

I received the dataset for this project through my mentor on this project and is in the .csv file. Some of the info other than malignant features are seems unnecessary with respect to the problem. So I drop it before any processing. Here is the snap of the data.



* Data Pre-processing Done

Removing the stop-words, punctuations and special characters from the text from text feature is done here.

* Data Inputs- Logic- Output Relationships

The input data for the processing and getting the output is converted in to the numerical forms or it data words, in to the vector form. It feeded to the model in the form of series (one by one) which analyses by the model providing the certain score through the medium of performance metrics.

* Hardware and Software Requirements and Tools Used

Hardware And Software required for this project:

* Laptop with I5 processor and 4 GB Ram
* Anaconda (Jupyter Notebook) or google colab
* Python Pandas for processing
* Scikit learn library for model deployment

**Model/s Development and Evaluation**

Identification of possible problem-solving approaches (methods)

These are the following approaches I have used here:

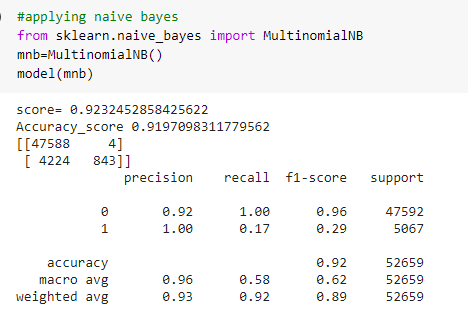
* Importing and displaying insights of the data using pandas Dataframe.
* Analysing the data and use proper pre-processing techniques to extract out important words.
* Applying data operations such as Tf-IDF to convert text to vectorized form
* Applying algorithms to check the performance using evaluation metrics.

Testing of Identified Approaches (Algorithms)

* Naïve Bayes-Multinomial
* Decision Tree Classifier
* Random Forest Classifier

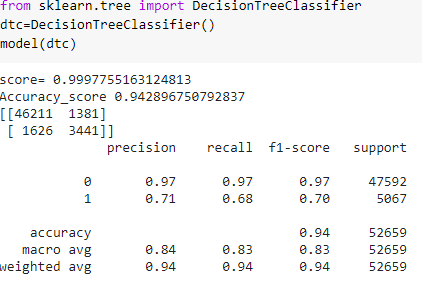
Run and Evaluate selected models

* **Multinomial naive Bayes** : The multinomial Naive Bayes classifier is suitable for classification with discrete features (e.g., word counts for text classification). The multinomial distribution normally requires integer feature counts.



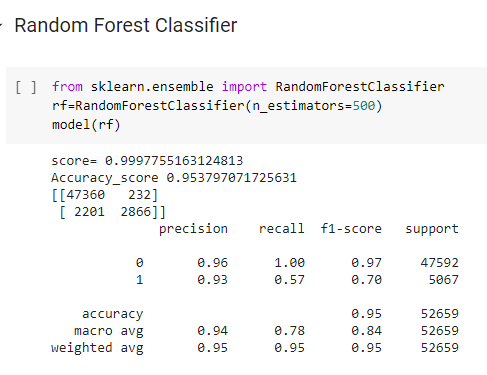
**Decision Tree**

Decision Tree is one of the earlier classification algorithm for text and data mining .Decision tree classifiers (DTC’s) are used successfully in many diverse areas of classification. The structure of this technique includes a hierarchical decomposition of the data space (only train dataset). The main idea is creating trees based on the attributes of the data points.



**Random Forest Classifier**

Random forests or random decision forests technique is an ensemble learning method for text classification. This method implies that the decision taken is depend on trees in parallel.



Key Metrics for success in solving problem under consideration

The performance metrics here mainly considered for the problem solving are:

* Accuracy score
* Confusion matrix
* Roc Curve & Auc-RoC score.

Interpretation of the Results

Decision Tree Classifier performs well from the applied two algorithms along with performance metrics as well i.e.Accuracy,F1 score, ROC-AUC score

**CONCLUSION**

* Key Findings and Conclusions of the Study

Some of the findings from data explorations are as follows:

* From displaying the data, it seems there is lot of special characters present in the data. So, it is better to proceed by filter it out.
* As the above data is in text, so presence of special characters and stopwords is always there.
* The data difference ratio in target label is around 9:10.
* After proper cleaning and processing, decision tree classifier gives the highest accuracy as well as roc score.
* Limitations of this work and Scope for Future Work

Some of the extention techniques that give the best results after optimization are:

* Multinomial naïve bayes and Random Forest using hyperparameter tuning.
* After analysing for each behaviour separately by creating models.
* Support Vector Machine also performs well in text data but its hyperparameter tuning is very complex and takes much more time.