

Malignant Comments Classifier

Submitted by:

HITESH KUMAR SINGH

**ACKNOWLEDGMENT**

1. Kevin Khieu and Neha Narwal: “Detecting and Classifying Toxic Comments” <https://web.stanford.edu/class/archive/cs/cs224n/cs224n.1184/reports/6837517.pdf>
2. ManavKohli, Emily Kuehler and John Palowitch: ”Paying attention to toxic comments online” <https://web.stanford.edu/class/archive/cs/cs224n/cs224n.1184/reports/6856482.pdf>
3. Sara Zaheri, Jeff Leath and David Stroud: “Toxic Comment Classification” <https://scholar.smu.edu/datasciencereview/vol3/iss1/13/>
4. <https://stackoverflow.com/>
5. <https://www.kaggle.com/>

**INTRODUCTION**

* Business Problem Framing

This study aims to analyse any piece of text and detecting different types of toxicity like malignant, highly malignant, rude, threat, abuse, loathe.

* Conceptual Background of the Domain Problem

Over a decade, social media have been growing, and people are able to express their opinions and also discuss among others via these platforms. These debates may arise due to differences in opinion and may often result in fights over the social media during which offensive language termed as malignant comments may be used from one side. This clearly pose the threat of abuse and harassment online. As such, some people stop giving their opinions or give up seeking different opinions which result in unhealthy and biased discussion. As a result, different platforms and communities find it very difficult to facilitate fair conversation and are often forced to either limit user comments or get dissolved by shutting down user comments completely.

* Review of Literature

Many machine learning approaches have been implemented for detecting types of malignant comments. We have tested the dataset on:

Multinomial Naive Bayes, Linear SVC, Logistic Regression, Decision Tree Classifier and KNeighbors Classifier.

* Motivation for the Problem Undertaken

The Conversation AI team, a research group founded by Jigsaw and Google have been working on providing an environment for healthy communication. They have also built publicly available models through the Perspective API on Comment Toxicity. But these models are sometimes prone to errors and does not provide the option to the users for choosing which type of toxicity, they are interested in finding. As such, a stable and improved intelligent system is required for Malignant Comment Prevention in social communication.

**Analytical Problem Framing**

* Mathematical/ Analytical Modeling of the Problem

**Df.describe():** To understand the mean, standard deviation, minimum, maximum, 25% and 75% data points.

Very less 1’s compared to 0’s; We can infer that majority of the dataset does not have what we want to classify. We can predict more of what we don’t want to predict, hence, giving better probability of distinguishing the comments.

**Seaborn.pairplot(df):** Display histograms of the complete data

**Non-unique:** Identify rows that has multiple 1’s in each rows {0: No multiple values, 1: Multiple values}

**Org\_length:** To find the number of words before pre-processing in each row.

**New\_length:** To find the number of words after pre-processing in each row.

* Data Sources and their formats



**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.

*There may be some comments which have multiple labels on them, i.e. some comments may be both malignant and loathe.*

**Train.csv**: the training set which contains comments with their binary labels.

**Test.csv**: The test set for which the predictions are to be done. It includes id and comments\_text.

* Data Preprocessing Done

1. Convert text to lower case and replaced email addresses with ‘email’
2. Replaced URLs with 'webaddress'
3. Replaced money symbols with 'dollars'
4. Replaced 10 digit phone numbers (formats include paranthesis, spaces, no spaces, dashes) with 'phonenumber'
5. Replaced numbers with 'numbr'
6. Removed punctuation
7. Replaced whitespace between terms with a single space
8. Removed leading and trailing whitespace
9. Removed stopwords with nltk stopwords corpus. Additional stopwords removed are 'u', 'ü', 'â', 'ur', '4', '2', 'im', 'dont', 'doin' and 'ure'

* Data Inputs- Logic- Output Relationships

1. comment\_text is string and has to be transformed into vector.
2. Id is a combination of alphanumeric values that is unique to each row.
3. All other columns are in binary {0: False, 1: True}

* State the set of assumptions (if any) related to the problem under consideration

1. Data might be imbalanced
2. Multi-target classification
3. Linear Kernel is the most appropriate SVM Kernel
4. Multinomial Naïve Bayes to be used

* Hardware and Software Requirements and Tools Used

Google Colab (Multiple sessions with at most session connected to 12 GB RAM and 100 GB Disk Space)

**Model/s Development and Evaluation**

* Identification of possible problem-solving approaches (methods)

**Df.describe():** To understand the mean, standard deviation, minimum, maximum, 25% and 75% data points.

Very less 1’s compared to 0’s; We can infer that majority of the dataset does not have what we want to classify. We can predict more of what we don’t want to predict, hence, giving better probability of distinguishing the comments.

**Seaborn.pairplot(df):** Display histograms of the complete data

**Non-unique:** Identify rows that has multiple 1’s in each rows {0: No multiple values, 1: Multiple values}

**Org\_length:** To find the number of words before pre-processing in each row.

**New\_length:** To find the number of words after pre-processing in each row.

* Testing of Identified Approaches (Algorithms)

MultinomialNB

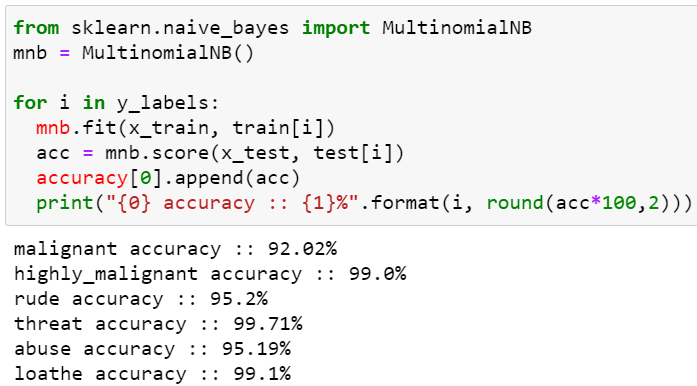
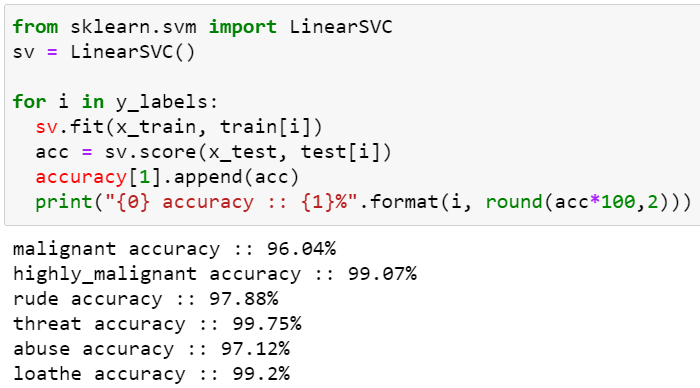
LinearSVC

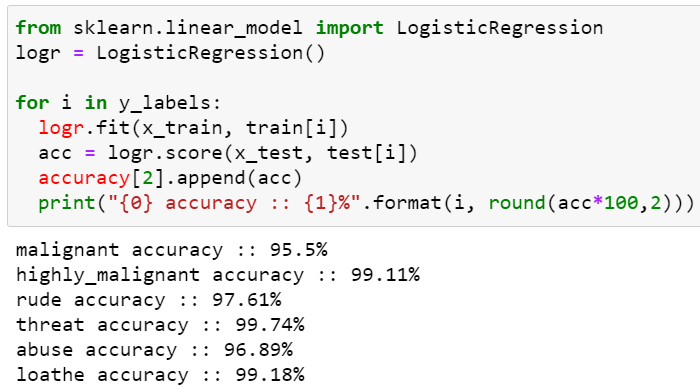
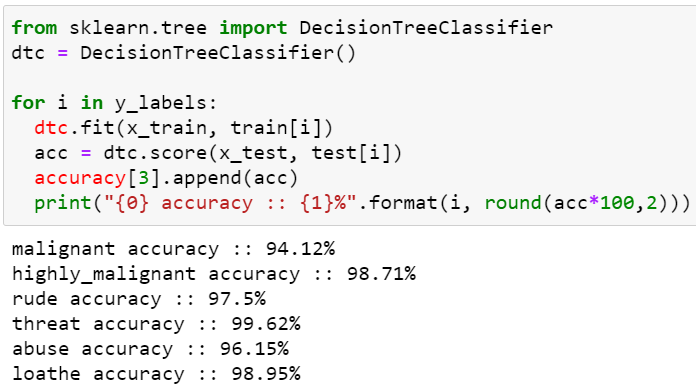
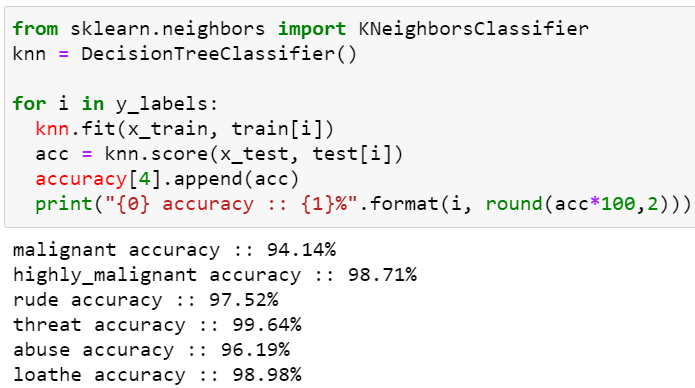
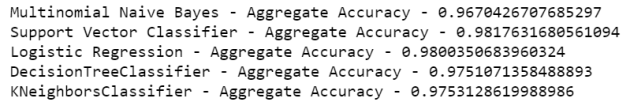
LogisticRegression

DecisionTreeClassifier

KNeighborsClassifier

* Run and Evaluate selected models

* Key Metrics for success in solving problem under consideration

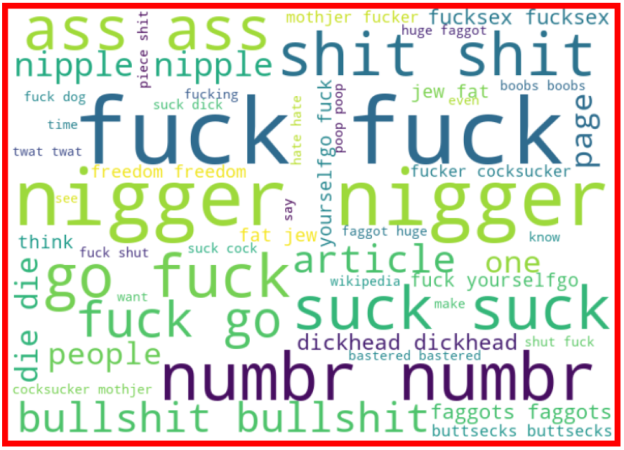
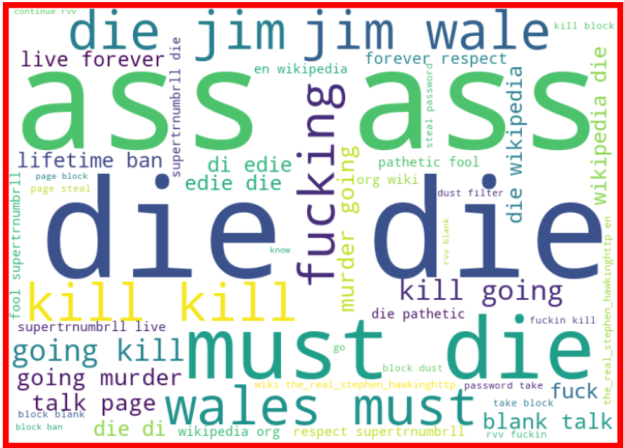
F1 Score, Accuracy Score, Confusion Matrix and ROC Curve

* Visualizations

Malignant Word Cloud Highly Malignant Word Cloud

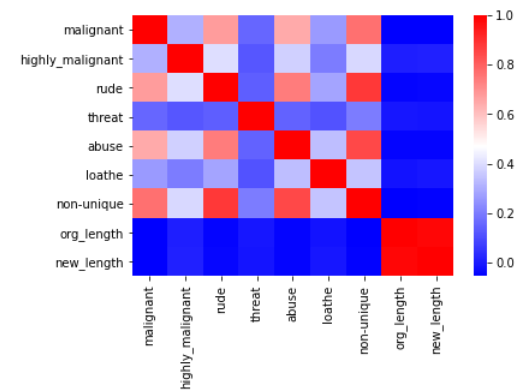
Rude Word Cloud Threat Word Cloud

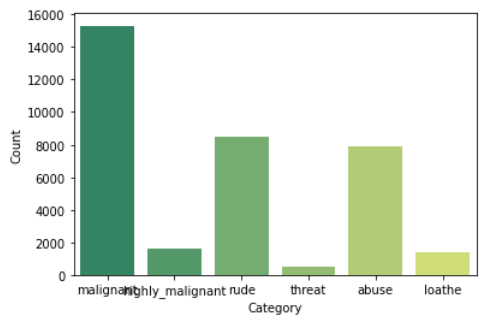
Abuse Word Cloud Loathe Word Cloud

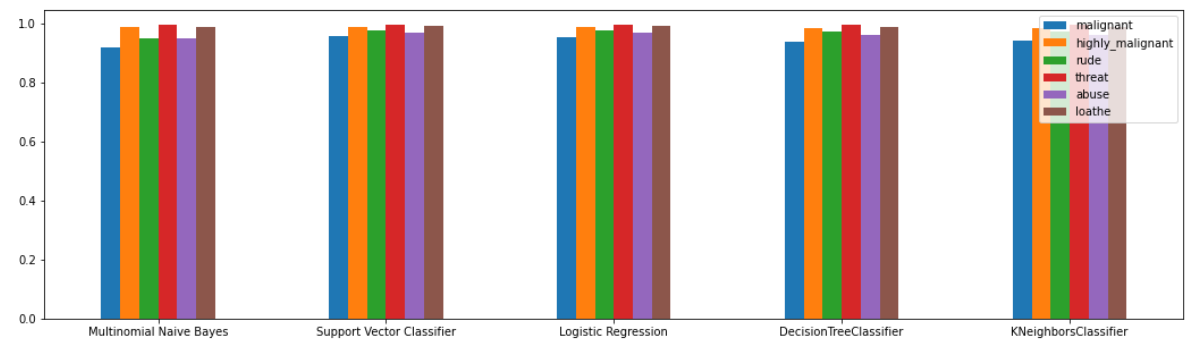
Non-unique Word Cloud Correlation Heatmap

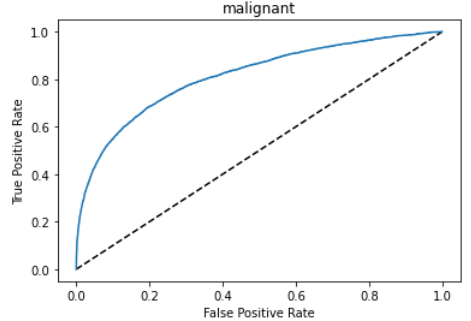
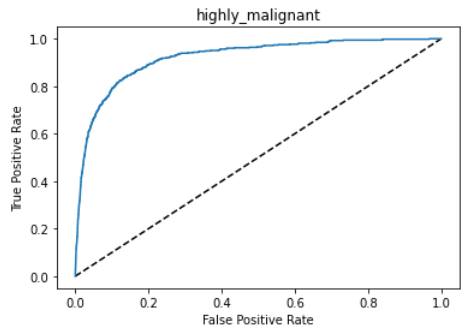
Category Vs Count Barplot



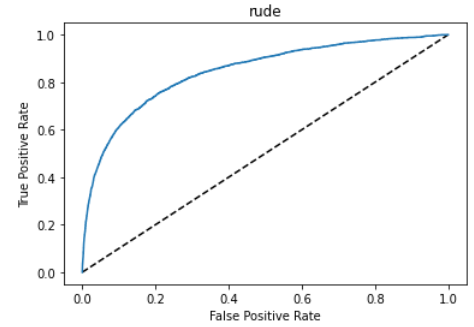
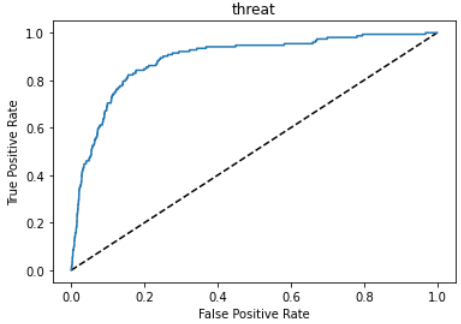
All Models Accuracy Comparison



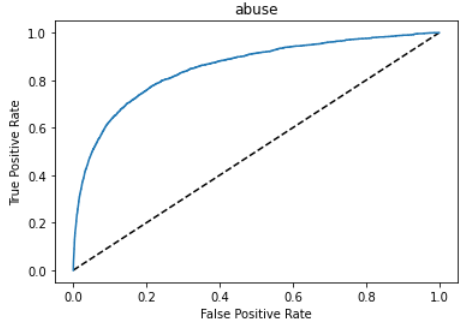
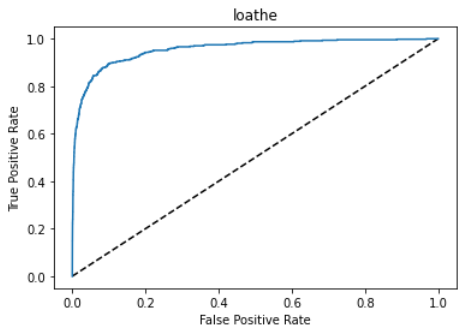
Malignant ROC Curve Highly Malignant ROC Curve

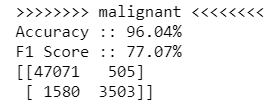
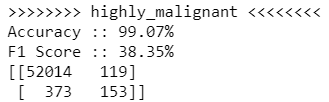
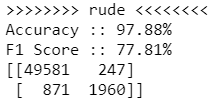
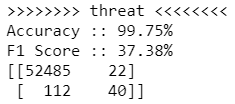
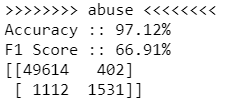
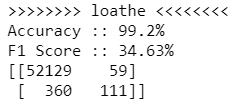
Rude ROC Curve Threat ROC Curve

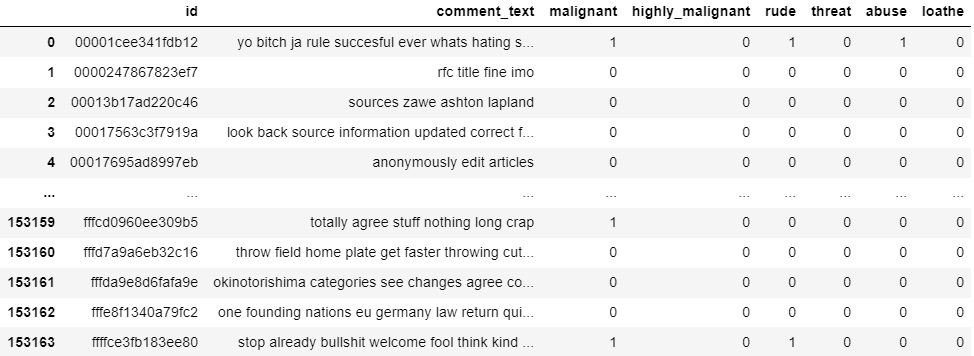
 

Abuse ROC Curve Loathe ROC Curve

* Interpretation of the Results



**CONCLUSION**

* Key Findings and Conclusions of the Study

Describe the key findings, inferences, observations from the whole problem.

* Learning Outcomes of the Study in respect of Data Science

List down your learnings obtained about the power of visualization, data cleaning and various algorithms used. You can describe which algorithm works best in which situation and what challenges you faced while working on this project and how did you overcome that.

* Limitations of this work and Scope for Future Work

What are the limitations of this solution provided, the future scope? What all steps/techniques can be followed to further extend this study and improve the results.