NAME OF THE PROJECT

MALLIGNANT COMMENT CLASSIFIER USING NLP

Submitted by:

RAVINDER SINGH

FLIPROBO SME: MS. KHUSHBOO GARG

ACKNOWLEDGMENT

I would like to express my special gratitude to “Flip Robo” team, who has given me this opportunity to deal with a beautiful dataset and it has helped me to improve my analyzation skills. And I want to express my huge gratitude to Ms. Khushboo Garg (SME Flip Robo), she is the person who has helped me to get out of all the difficulties I faced while doing the project.
A huge thanks to “Data trained” who are the reason behind my Internship at Fliprobo. Last but not least my parents who have been my backbone in every step of my life.
References use in this project:

- SCIKIT Learn Library Documentation
- Blogs from towardsdatascience, Analytics Vidya, Medium
- Andrew Ng Notes on Machine Learning (GitHub)
- Data Science Projects with Python Second Edition by Packt
- Hands on Machine learning with scikit learn and tensor flow by Aurelien Geron
- Lackermair, G., Kailer, D. & Kanmaz, K. (2013). Importance of online product reviews from a consumer’s perspective. Horizon Research Publishing, 1-5. doi: 10.13189/aeb.2013.010101
- Baccianella, S., Esuli, A. & Sebastiani, F. (2009). Multi-facet rating of product reviews. Proceedings of the 31st European Conference on Information Retrieval (ECIR), 461- 472
- Project BY : https://github.com/NikhilGohil/Social-Media-and-Data- MiningProject/

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 aproblem 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.

Conceptual Background of the Domain Problem

A person who is not happy with the product of a company can respond in several ways ,person can write comment in English and their native Language . also it depends on the vocabulary of the person that what words he chooses to express their feelings.It will be considered malignant

Review of Literature

According to the Lackermair, Kailer and Kanmaz (2013), product reviews and ratings represent an important source of information for consumers and are helpful tools in order to support their buying decisions [6]. They also found out that consumers are willing to compare both positive and negative reviews when searching for a specific product. The authors argue that customers need compact and concise information about the products. Therefore, consumers first need to pre-select the potential products matching their requirements. With this aim in mind, consumers use the star ratings as an indicator for selecting products. Later, when a limited number of potentials products have been chosen, reading the associated text review will reveal more details about the products and therefore help consumers making a final decision.

It becomes daunting and time-consuming to compare different products in order to eventually make a choice between them. Therefore, models able to predict the user rating from the text review are critically important (Baccianella, Esuli & Sebastiani, 2009) [7].
Pang, Lee and Vaithyanathan (2002) [9] approach this predictive task as an opinion mining problem enabling to automatically distinguish between positive and negative reviews. In order to determine the reviews polarity, the authors use text classification techniques by training and testing binary classifiers on movie reviews containing 36.6% of negative reviews and 63.4% of positive reviews. On the top of that, they also try to identify appropriate features to enhance the performance of the classifiers.
Dave, Lawrence, and Pennock (2003) [10] also deal with the issue of class imbalance with a majority of positive reviews and show similar results. SVM outperforms Naïve Bayes with an accuracy greater than 85% and the implementation of part-of-speech as well as stemming is also ineffective. However, this work demonstrates that bigrams turn out to be more successful at capturing context than unigrams in the specific situation of their datasets, despite earlier research having produced better results with unigrams.
to capture the weights of such characteristics by minimising the mean square error.

Motivation for the Problem Undertaken

My first project from Flip Robo Technologies under the internship programme was the project. The main drivers behind this were the chance to apply my skill set to a real-world problem and the exposure to data from the actual world.
The project has implementation in the various domain like analysis of twitter comment and youtube comment which can help a company to get aware of the threat, hate they will be facing against their product.

Analytical Problem Framing

Mathematical / Analytical Modelling of the Problem

The first step involved collecting data and deciding what part of it is suitable for training : This step was extremely crucial since including only very small length comments would give poor results if the length was increased whereas including very long length comments would increase the number of words drastically, hence increasing the training time exponentially and causing system (jupyter kernel) to go out of memory and die eventually. The second major step was performing cleaning of data including punctuation removal, stop word removal, stemming and lemmatizing: This step was also crucial since the occurrence of similar origin words but having different spellings will intend to give similar classification, but computer cannot recognize this on its own. Hence, this step helped to a large extent in both removing and modifying existing words.

Data Sources and their formats

Data source was provided but was not read by the Jupyter usual command so in order to read we have to change our Engine to Python which rather slow than the usual Engine

```
# Importing dataset excel file using pandas.
train_data=pd.read_csv('train.csv',engine='python') # Engine was set to python
```

```
print('No. of Rows :',train_data.shape[0])
print('No. of Columns :',train_data.shape[1])
pd.set_option('display.max_columns',None) # This will enable us to see truncated columns
train_data.head()
```

```
No. of Rows : 159571
No. of Columns : 8
```

This is multi-classification problem only Mallignant comments are targeted here to find whether the made comment is malignant or not.

```
Value Counts of malignant
0    144277
1     15294
Name: malignant, dtype: int64
==================================================================================================
Value Counts of highly_malignant
0    157976
1      1595
Name: highly_malignant, dtype: int64
==================================================================================================
Value Counts of rude
0    151122
1      8449
Name: rude, dtype: int64
==================================================================================================
Value Counts of threat
0    159093
1       478
Name: threat, dtype: int64
==================================================================================================
Value Counts of abuse
0    151694
1      7877
Name: abuse, dtype: int64
==================================================================================================
Value Counts of loathe
0    158166
1      1405
Name: loathe, dtype: int64
==================================================================================================
```

Data Pre-processing

The dataset is large and it may contain some data error. In order to reach clean, error free data some data cleaning & data pre-processing performed data.

Missing Value Imputation:

Missing value were not there

Data is pre-processed using the following techniques:

Convert the text to lowercase
Remove the punctuations, digits and special characters
Tokenize the text, filter out the adjectives used in the review and create a new column in data frame
Remove the stop words
Stemming and Lemmatising
Applying Text Vectorization to convert text into numeric

Data Inputs- Logic- Output Relationships

The dataset consists of 6 features with a label. The comments are independent and Nature of comment is dependent as our label varies the values (text) of our independent variable's changes. Using word cloud, we can see most occurring word for different categories.

Hardware & Software Requirements with Tool Used

Hardware Used -
Processor — Intel i3 processor with 2.4GHZ
RAM — 4 GB
GPU — 2GB AMD Radeon Graphics card Software utilised -
Anaconda – Jupyter Notebook
Selenium – Web scraping
Google Colab – for Hyper parameter tuning
Libraries Used – General library for data wrangling & visualsation

Models Development & Evaluation

Identification Of Possible Problem-Solving Approaches (Methods)

The first step involved collecting data and deciding what part of it is suitable for training : This step was extremely crucial since including only very small length comments would give poor results if the length was increased whereas including very long length comments would increase the number of words drastically, hence increasing the training time exponentially and causing system (jupyter kernel) to go out of memory and die eventually. The second major step was performing cleaning of data including punctuation removal, stop word removal, stemming and lemmatizing: This step was also crucial since the occurrence of similar origin words but having different spellings will intend to give similar classification, but computer cannot recognize this on its own. Hence, this step helped to a large extent in both removing and modifying existing words.0

Testing of Identified Approaches (Algorithms)

The different classification algorithm used in this project to build ML model are as below:
Logistic Regression
Support vector Machine
Multinomial Naïve Bayes
Decision Tress classifier

Key Metrics for Success in Solving Problem Under Consideration

Label bases metrics include one-error, average precision, etc. These can be calculated for each labels, and then can be averaged for all without taking into account any relation between the labels if exists.
Average Precision (AP): Average precision is a measure that combines recall and precision for ranked retrieval results. For one information need, he average precision is the mean of the precision scores after each relevant document is retrieved, where, and are the precision and recall at the threshold.
Accuracy is defined as the proportion of correctly predicted labels to the total no. of labels

for each instance.
Hamming-loss is defined as the symmetric difference between predicted and true labels, divided by the total no. of labels.

Run And Evaluate Selected Models

Logistic Regression

```
Building Models:   0%|          | 0/3 [00:00<?, ?it/s]
*******
Current Model in Progress: Logistic Regression
*******
Training:  BinaryRelevance(classifier=LogisticRegression(), require_dense=[True, True])
Testing:

        Hamming Loss  : 0.021765467686009107
        Accuracy Score: 0.9118937210176714
              precision    recall  f1-score   support

           0       0.93      0.53      0.67       790
           1       0.60      0.16      0.25        93
           2       0.94      0.58      0.72       456
           3       0.40      0.08      0.13        26
           4       0.81      0.48      0.60       404
           5       0.77      0.16      0.26        64

   micro avg       0.89      0.49      0.63      1833
   macro avg       0.74      0.33      0.44      1833
weighted avg       0.88      0.49      0.62      1833
 samples avg       0.05      0.04      0.04      1833
Completed in [26.014041900000166 sec.]
**********
```

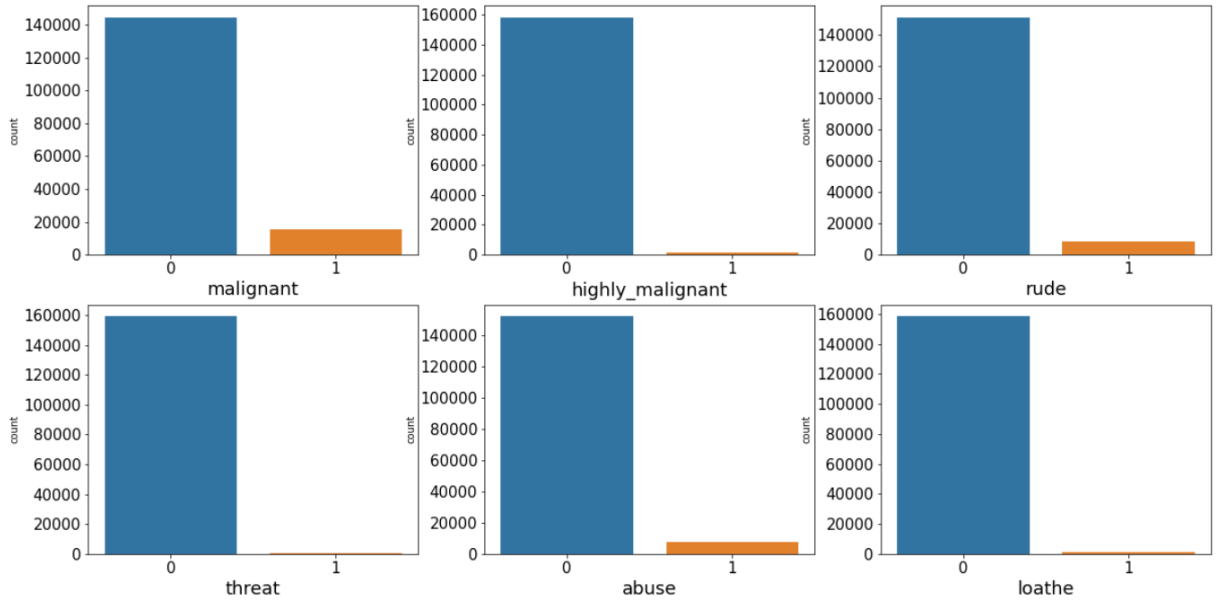
Support Vector Machine

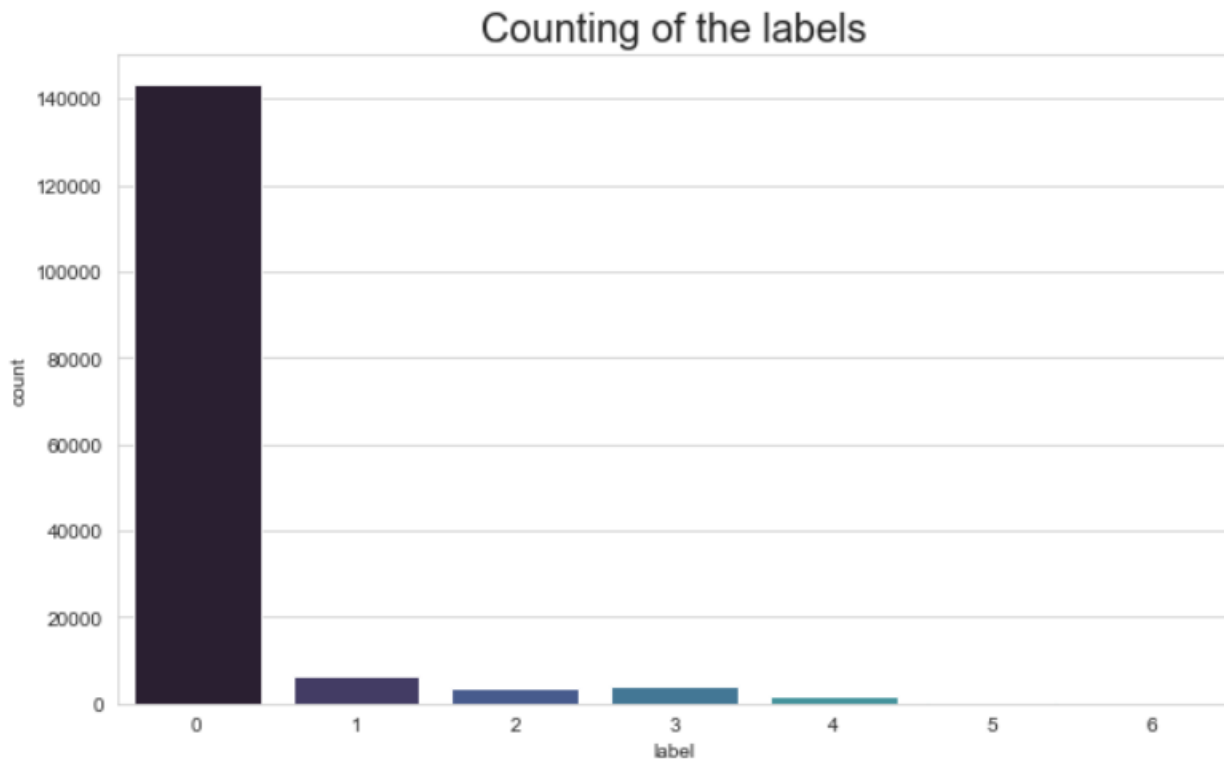
```
*******
Current Model in Progress: Support Vector Classifier
*******
Training:  BinaryRelevance(classifier=LinearSVC(max_iter=3000), require_dense=[True, True])
Testing:

        Hamming Loss  : 0.020449513305760957
        Accuracy Score: 0.9128963529264319
              precision    recall  f1-score   support

           0       0.87      0.62      0.72       790
           1       0.59      0.18      0.28        93
           2       0.91      0.66      0.76       456
           3       0.42      0.19      0.26        26
           4       0.76      0.56      0.65       404
           5       0.66      0.30      0.41        64

   micro avg       0.84      0.58      0.68      1833
   macro avg       0.70      0.42      0.51      1833
weighted avg       0.83      0.58      0.68      1833
 samples avg       0.06      0.05      0.05      1833
Completed in [8.209033300000101 sec.]
**********
```

Multinomial Naïve Bayes

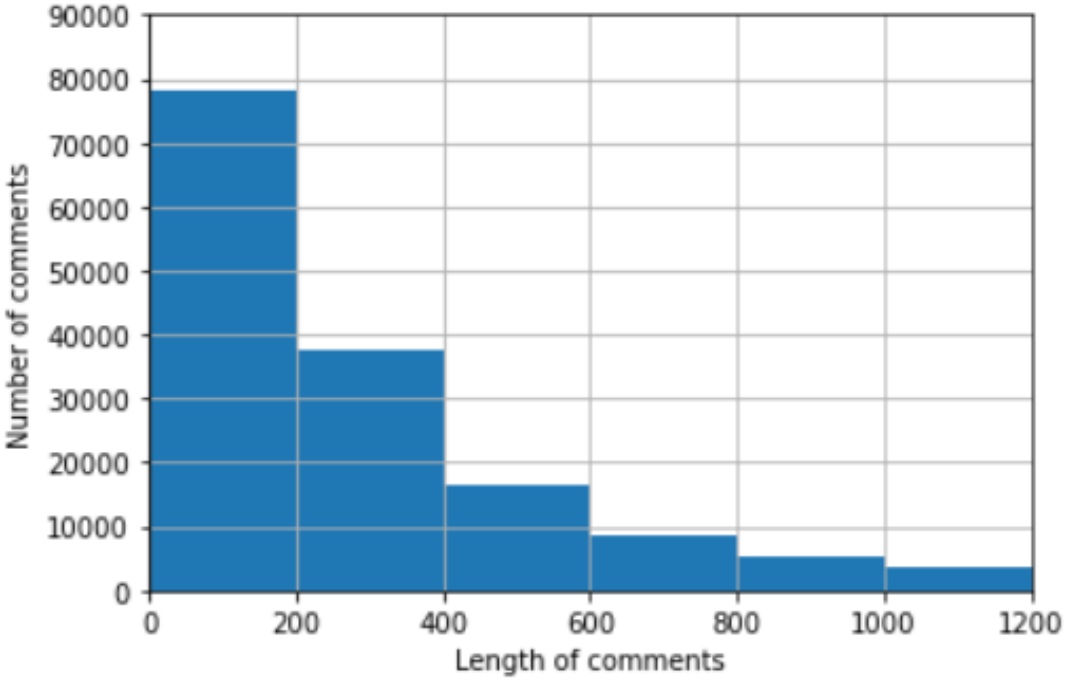
```
*******
Current Model in Progress: Multinomial Naive Bayes
*******
Training:  BinaryRelevance(classifier=MultinomialNB(), require_dense=[True, True])
Testing:

        Hamming Loss  : 0.022433888958516106
        Accuracy Score: 0.9103897731545306
              precision    recall  f1-score   support

           0       0.95      0.50      0.66       790
           1       0.62      0.09      0.15        93
           2       0.96      0.54      0.69       456
           3       0.00      0.00      0.00        26
           4       0.84      0.44      0.58       404
           5       1.00      0.05      0.09        64

   micro avg       0.92      0.45      0.61      1833
   macro avg       0.73      0.27      0.36      1833
weighted avg       0.90      0.45      0.59      1833
 samples avg       0.05      0.04      0.04      1833
Completed in [5.720615299999963 sec.]
**********
```

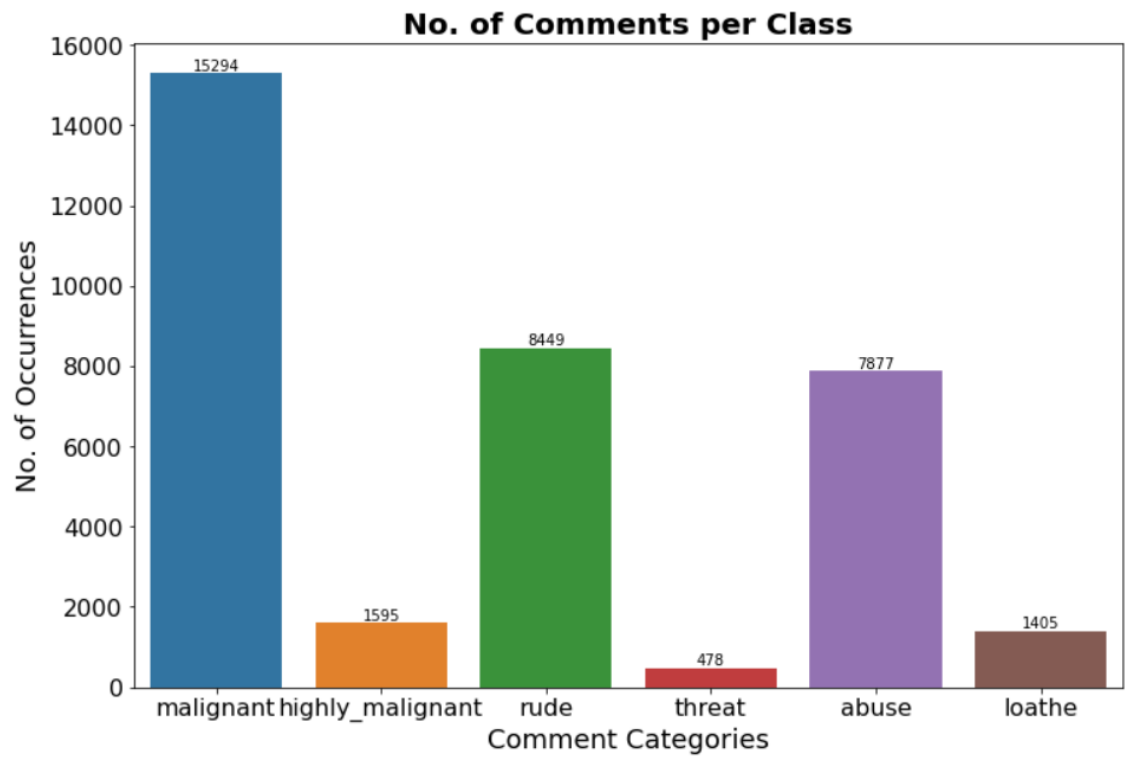
Visualizations






```
average length of comment: 394.139
```





Comment:

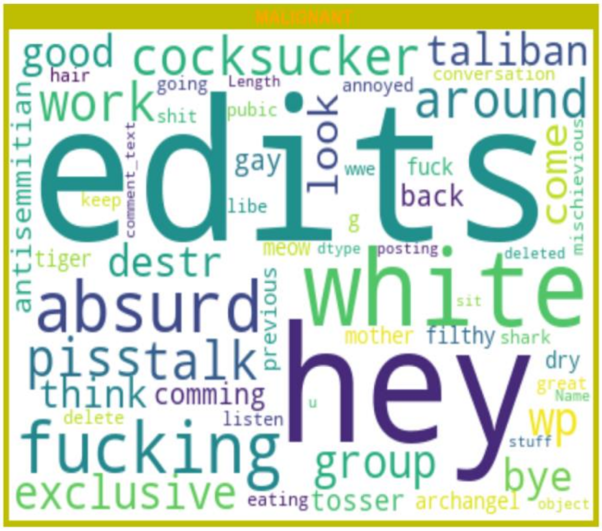
Out of all negative comments, malicious remarks make up the majority, followed by impolite categories.There aren't many remarks that are threatening

Although malignant remarks make up the majority of negative comments, many comments are also abusive and unpleasant, while threat comments make up the least amount of negative comments.

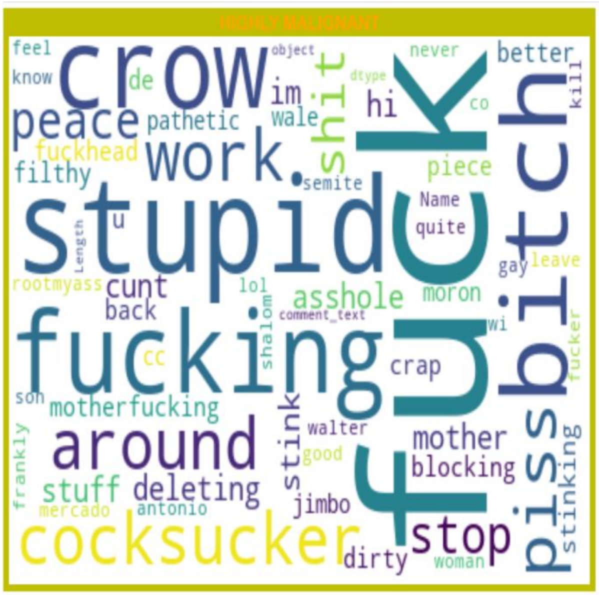
Word Cloud:

Word Cloud is a visualization technique for text data wherein each word is picturized with its importance in the context or its frequency.

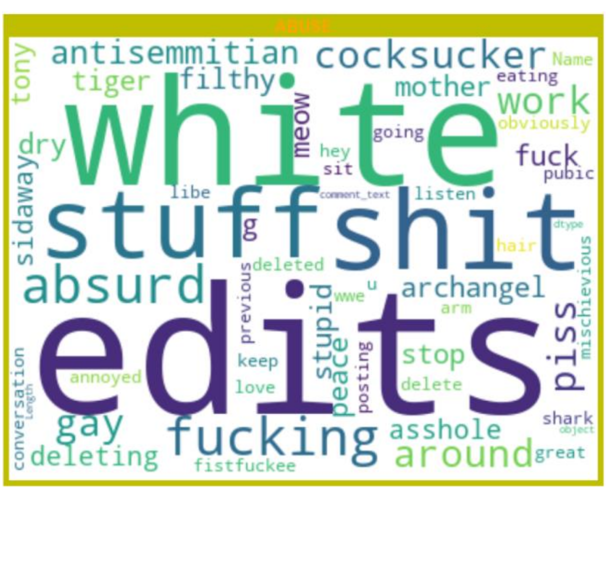
The more commonly the term appears within the text being analysed, the larger the word appears in the image generated.The enlarged texts are the greatest number of words used here and small texts are the smaller number of words use


MALIGNANT
edits
hey
white
cocksucker
taliban
around
good
work
absurd
pisstalk
fucking
exclusive
group
think
destr
look
come
antisemmitian
comming
tosser
archangel
bye
wp


RUDE
shit
fucking
white
piecestuff
absurd
edits
cocksucker
piss
gay
fucked
obese
cant
put
filthy
stop
archangel
asshole
antisemmitian
around
work
fuckeeed
drink
naked
lolwat
conversation


HIGHLY MALIGNANT
fuck
stupid
fucking
bitch
piss
crow
peace
work
around
cocksucker
stop
shit
asshole
cunt
mother
blocking
deleting
stuff
motherfucking
pathetic
fuckhead
stink
stinking


THREAT
die
fuck
bitch
suck
stupid
hey
back
stop
hell
last
edits
hi
oife
fagget
hate
warning
undoing
think
burn
puppet
sorry
sodomize
notified
dumb
account

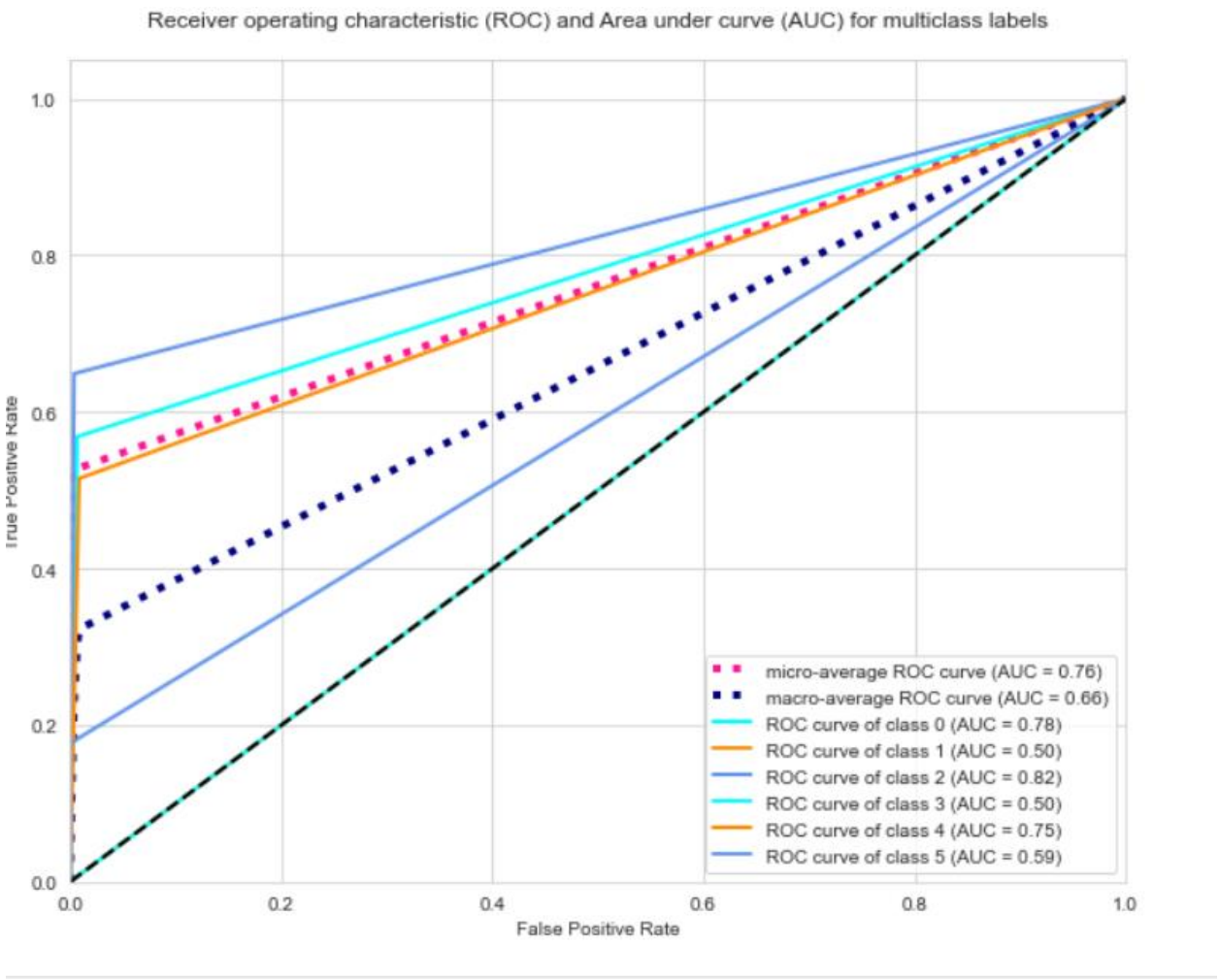

ABUSE
white
edits
stuffshit
absurd
fucking
antisemmitian
cocksucker
around
archangel
piss
work
tiger
filthy
mother
gay
deleting
asshole
stop
fuck


LOATHE
fuck
gay
u
thinkkill
jew
white
weiner
meow
antisemmitian
archangel
tiger
fagget
hating
nazi
pair
nigger
schmuck
sorry
hate
racist

Conclusion

Key Findings and Conclusion of the Study

Algorithm	Accuracy Score	Recall	Precision	F1 Score	Humming Loss
Logistic Regression	0.9118	0.49	0.89	0.63	0.0217
Support Vector Machine	0.9128	0.58	0.84	0.68	0.0204
Multinomial Naïve Bayes	0.9103	0.45	0.92	0.61	0.0224



- Linear Support Vector Classifier performs better with Accuracy Score: 91.15077857956704 % and Hamming Loss: 2.0952019242942144 % than the other classification models.

- Final Model (Hyperparameter Tuning) is giving us Accuracy score of 91.26% which is slightly improved compare to earlier Accuracy score of 91.15%.

SVM classifier is fastest algorithm compare to others

Learning Outcomes of Data Science

In this project we got to know that kernel of jupyter has limitation and in order to takle that we have to shorten our data by One- Fourth

In this project we were able to learn various Natural language processing techniques like lemmatization, stemming, removal of Stop words.

This project has demonstrated the importance of sampling effectively, modelling and predicting data.

Limitations of this work and Scope for the Future

- The Maximum feature used while vectorization is 2000. Employing more feature in vectorization lead to more accurate model which I not able to employed due computational resources.
- Data is imbalanced in nature but due to computational limitation we have not employed balancing techniques here.
- Deep learning CNN, ANN can be employed to create more accurate model.

.