MALIGNANT COMMENTS CLASSIFICATION

Submitted by:

Satarupa Goswami

ACKNOWLEDGMENT

The dataset was provided by FilpRobo technologies. The dataset mainly deals with the information ‘Malignant comments classification’ The project was done using python libraries using jupyter notebook. There libraries used for the processing of data and its analysis are numpy, pandas, sklearn . Along with the help from professionals at data trained for running various algorithms also from my SME at FlipRobo. Moreover, websites like https://towardsdatascience.com/ , https://scikit-learn.org/stable/ , were used to draw references and complete the project.

INTRODUCTION

- Business Problem Framing

  With the online world expanding exponentially many have the opportunity to express themselves and the words convey feelings which at times convey hate. These words which can be used to start online movements to start wars within countries. In fact, these hateful words are a serious problem that is worth looking into. If there were a way to control this spread of hatred then everyone might feel a little better.

- Conceptual Background of the Domain Problem

  Online hate, described as abusive language, aggression, cyberbullying, hatefulness and many others has been identified as a major threat on online social media platforms. With the increase of such instances, it is all the more necessary to make the online platform safer for it's users. So, the idea is to help separate the hateful words and filter them. It can be done by building a classification model. Which will help , control the spread of hate , cyberbullying online.

Analytical Problem Framing

- Mathematical/ Analytical Modelling of the Problem

  The dataset was divided into train and test datasets where in the train dataset set contains a total of 8 columns of which 6 columns had binary values.

  Train dataset = df_train

```
Data columns (total 8 columns):
 #   Column            Non-Null Count   Dtype
---  ------            --------------   -----
 0   id                159571 non-null  object
 1   comment_text      159571 non-null  object
 2   malignant         159571 non-null  int64
 3   highly_malignant  159571 non-null  int64
 4   rude              159571 non-null  int64
 5   threat            159571 non-null  int64
 6   abuse             159571 non-null  int64
 7   loathe            159571 non-null  int64
dtypes: int64(6), object(2)
```

  Test dataset = df_test

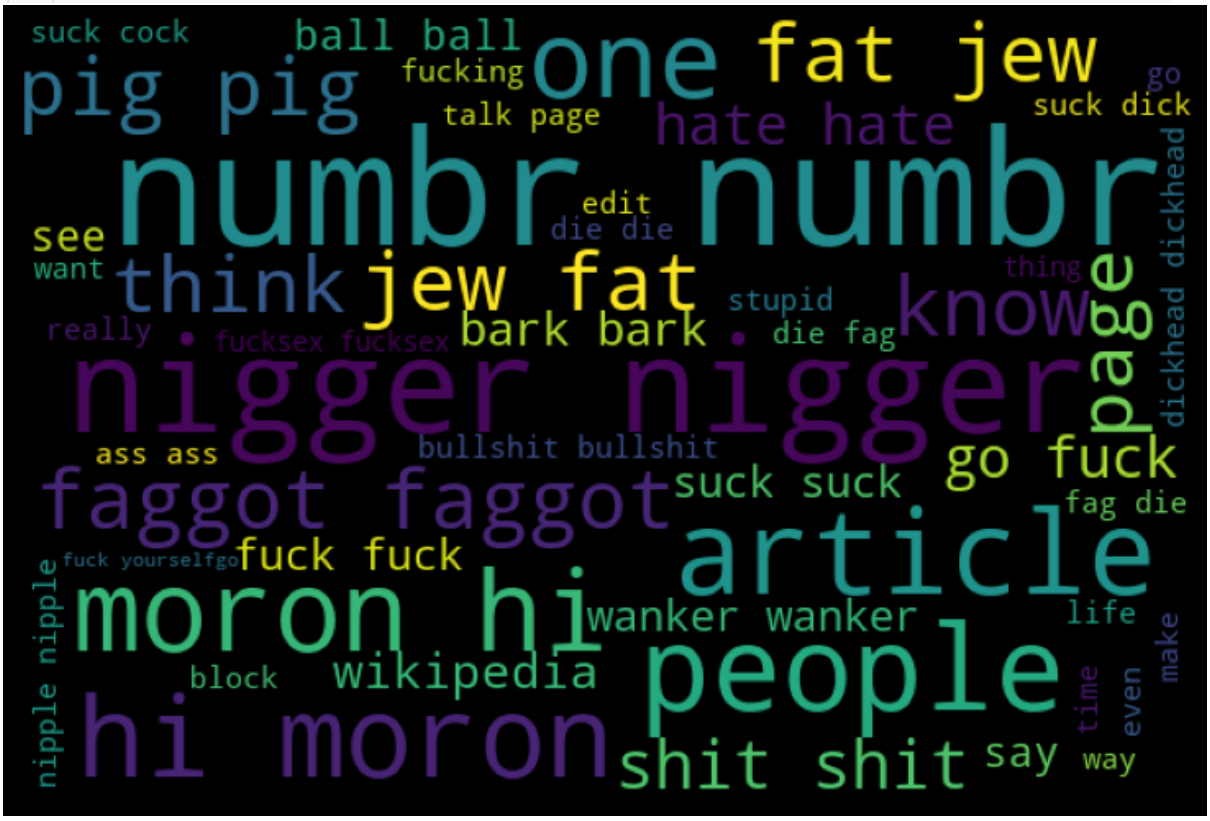
```
Data columns (total 2 columns):
 #   Column        Non-Null Count   Dtype
---  ------        --------------   -----
 0   id            153164 non-null  object
 1   comment_text  153164 non-null  object
dtypes: object(2)
```

  The issue at hand was to create a classification model that can qualify words as ‘malignant’ , ‘highly_ malignant’ , ‘rude’ , ‘threat’, ‘abuse’ , ‘loathe’ . These were classified as binary values. With a positive affirmation of negative word being 1 and the opposite being 0.

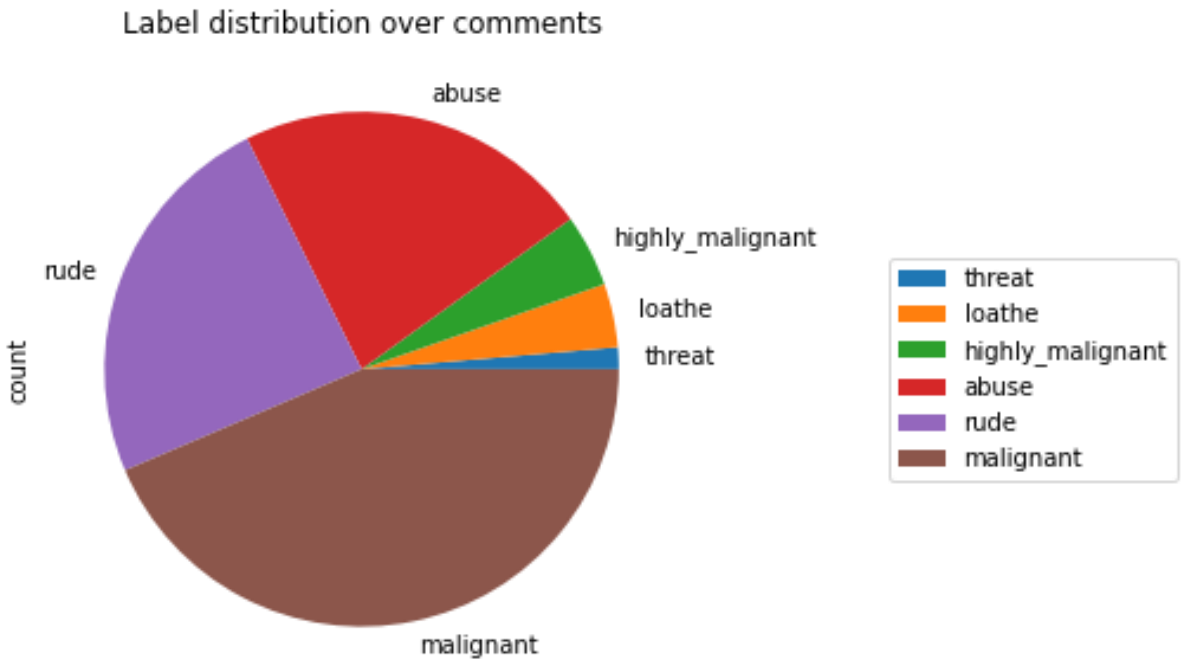
  As the comments contained a lot various other things that were not required and had to be removed from the comment itself. Like for instance the comments contained phone numbers, emails , numbers , symbols and stop words , which might be giving meaning to the comment as a whole but is not important for it’s classification . Since it had to be removed, another column which gave the total length to the comment string was created, and later another column which comprised of the cleaned column length was also created to understand how much of the total comments were removed.

The following figure was obtained that showed the most offensive words used in the comments.

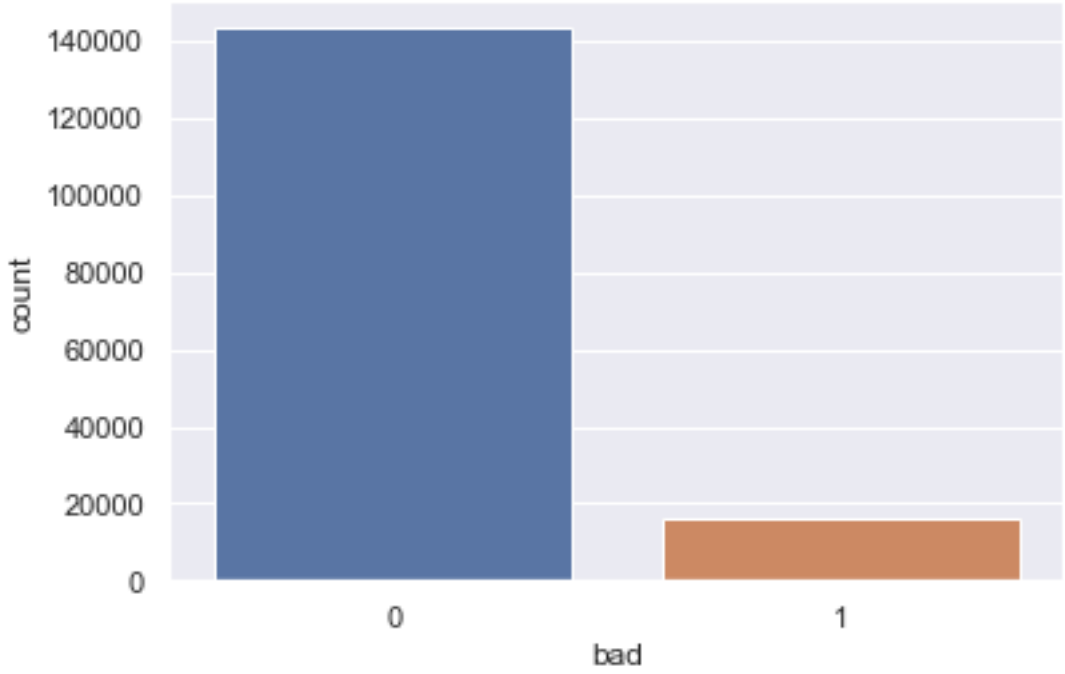
```
#Getting sense of loud words which are offensive
from wordcloud import WordCloud
hams = df_train['comment_text'][df_train['malignant']==1]
spam_cloud = WordCloud(width=600,height=400,background_color='black',max_words=50).generate(' '.join(hams))
plt.figure(figsize=(10,8),facecolor='k')
plt.imshow(spam_cloud)
plt.axis('off')
plt.tight_layout(pad=0)
plt.show()
```



This basically shows how the distribution is over all the different binary columns which classifies the words as malignant , highly_malignant , loathe, threat, rude, abuse.



Later these columns were merged into a single column as ‘bad’.
Where 0 is a bad word and 1 is considered to be not a bad word.



Model/s Development and Evaluation

- Testing of Identified Approaches (Algorithms)

The algorithms used were :

Logistic Regression

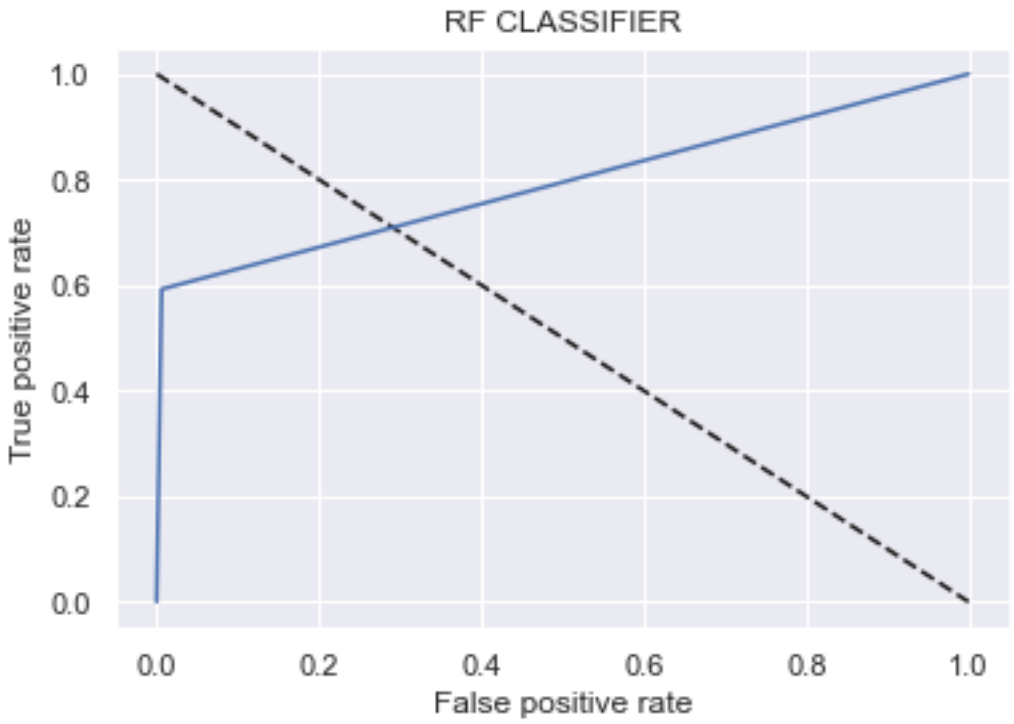
KNN Classifier

RF Classifier

XGBoost Classifier

The metric used to measure these classifiers is 'LOG-LOSS' , 'AUC_ROC', Using Precision , Recall , F1Score.

Of all the classifiers used RF Classifier had the best AUC_ROC score which enabled it to be the classifier used to obtain the final scoring and evaluation of the dataset.



Scoring Accuracy of the Algorithms Used:

Logistic Regression:

```
Training accuracy is 0.9595520103134316
Test accuracy is 0.9553392379679144
[[42729   221]
 [ 1917  3005]]
              precision    recall  f1-score   support

           0       0.96      0.99      0.98     42950
           1       0.93      0.61      0.74      4922

    accuracy                           0.96     47872
   macro avg       0.94      0.80      0.86     47872
weighted avg       0.95      0.96      0.95     47872
```

KNN Classifier :

```
Training accuracy is 0.9223985890652557
Test accuracy is 0.9176345254010695
[[42809   141]
 [ 3802  1120]]
              precision    recall  f1-score   support

           0       0.92      1.00      0.96     42950
           1       0.89      0.23      0.36      4922

    accuracy                           0.92     47872
   macro avg       0.90      0.61      0.66     47872
weighted avg       0.92      0.92      0.89     47872
```

RF Classifier :

```
Training accuracy is 0.9988630157834896
Test accuracy is 0.9552347927807486
[[42400   550]
 [ 1593  3329]]
              precision    recall  f1-score   support

           0       0.96      0.99      0.98     42950
           1       0.86      0.68      0.76      4922

    accuracy                           0.96     47872
   macro avg       0.91      0.83      0.87     47872
weighted avg       0.95      0.96      0.95     47872
```

XGBoost Classifier :

```
Training accuracy is 0.9614052050600274
Test accuracy is 0.9526236631016043
[[42689   261]
 [ 2007  2915]]
              precision    recall  f1-score   support

           0       0.96      0.99      0.97     42950
           1       0.92      0.59      0.72      4922

    accuracy                           0.95     47872
   macro avg       0.94      0.79      0.85     47872
weighted avg       0.95      0.95      0.95     47872
```

The final prediction.

```
1 prediction=RF.predict(test)
2 prediction
```

```
array([0, 0, 0, ..., 0, 0, 0])
```

Conclusion

As the internet finds new homes to expand to, along with it the opportunities it offers and also it's contretemps. Now, more than ever is the requirement for moderation. Moderation is something that doesn't sit well with the current world of decentralization, but some sort control needs to be implemented so as to protect the vast audience of the digital era. As, we are ushering in a new world of digital revolution, it is in our hands to create something that will make the world better for the coming generations.