MALIGNANT COMMENT PREDICTION

Submitted by:

AVINASH PATEL

ACKNOWLEDGMENT

I would like to express my special thanks of gratitude to my teacher as well as my intern company FLIPROBO who gave me this golden opportunity to do this wonderful project on the topic MALIGNANT COMMENT PREDICTION, which also helped me in doing a lot of Research and I came to know about so many new things I am really thankful to them. Secondly I want to mention some sites which helped me when I got stuck somewhere while completing my projects.

Those sites are :

http://scikit-learn.org

https://www.w3schools.com

https://www.youtube.com

https://www.kaggle.com

INTRODUCTION

- ## Business Problem Framing

  Our client had provided us with data. So the data contains 159571 rows and 8 columns.

  So, in order to improve the prediction of a comment whether it is how much malignant, highly malignant, rude, threat, abuse, loathe comment it is the client wants some predictions that could help them in predicting which type of comment it is.

- ## Conceptual Background of the Domain Problem

  We need the data with different types of comments with there corresponding type so as to build a model on op of that data.

- ## Review of Literature

  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.

- Motivation for the Problem Undertaken

  As I am an intern I want to work on as many project as possible for me. So ,I am highly motivated to do this project and to learn new thing and also learn from my mistakes.

Analytical Problem Framing

- Mathematical/ Analytical Modelling of the Problem

  First I analysed the data for the basic stuff like number of rows columns, do some of the rows are null or not etc. After that I moved on to data cleaning by removing many junk data from comments column then I proceeded to model selection predicting the test dataset.

- Data Sources and their formats

  We have 159571 rows and 8 columns of data from which two of the columns , id and comment are character columns and the rest are integer columns.

- Data Pre-processing

  We only need to pre-process the comment column as the other columns have nothing to pre-process.

  For comment column: first I do the stemming for this columns to do so I used snowball stemmer because it is new and d other stemming quite well for huge data. Then I calculated the length of the comment for each row so that we could have a n idea that up to how much data we have cleaned so far. Then I moved on with the removal of punctuations , white spaces, special characters ,capital letters etc. After doing all these steps I calculated the length of comment for each row and find out that only 83% data is left , which means I have successfully cleaned 17% of data which is junk.

- Data Inputs- Logic- Output Relationships

  The Remaining Data after all the pre-processing acts as an input data for the Model(in our case LogisticRegression) with OneVsRestClassifier , the already trained model gives us an output of a percentage of how much each malignant or loath or other column does the comment belong to..

- Hardware and Software Requirements and Tools Used

  Ram:8GB
  ROM:200MB
  Processor : Ryzen 5 3600X(6 cores 12 threads)
  Tool : Jupiter Notebook
  Language Used: Python

Model/s Development and Evaluation

- Identification of possible problem-solving approaches (methods)

  First I imported all the basic libraries of python used in the problem solving. Then I read carefully the data and the description given to me. After that I imported the provided csv file into the Jupiter notebook. Now first I checked for any nan values in the dataset after that I checked how many number of rows and columns I am dealing with and I also done some more analysis of the raw data to get a general idea and the removed some rows as some of them are duplicate and some of them are null.

  After analysing the data I started cleaning the data as explained above.

  After cleaning I did some visualisation of the data to get some insights from our cleaned data.

  On the completion of all these steps I moved on to the model selection part. Here I tested many models and choose one of the model to proceed.

  After I selected one of the models and finally I saved my model.

- Testing of Identified Approaches (Algorithms)
  - LogisticRegression
  - OneVsRestClassifier

- Run and Evaluate selected models

  The AUC-ROC score for the for the train is 99% and for the test data is 97% so we selected this model.

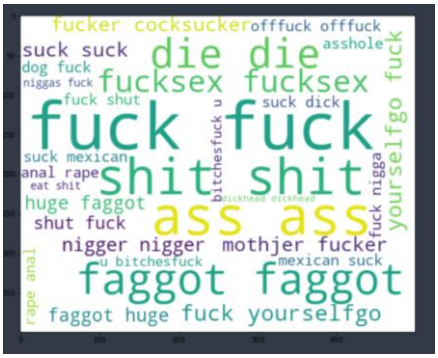
```
from sklearn.linear_model import LogisticRegression
from sklearn.multiclass import OneVsRestClassifier
from skmultilearn.problem_transform import BinaryRelevance
from sklearn.metrics import roc_auc_score , accuracy_score , confusion_matrix , f1_score

log_reg = LogisticRegression(C = 10, penalty='l2', solver = 'liblinear', random_state=7)

classifier = OneVsRestClassifier(log_reg)
classifier.fit(X_train_transformed, y_train)


y_train_pred_proba = classifier.predict_proba(X_train_transformed)
y_test_pred_proba = classifier.predict_proba(X_test_transformed)


roc_auc_score_train = roc_auc_score(y_train, y_train_pred_proba,average='weighted')
roc_auc_score_test = roc_auc_score(y_test, y_test_pred_proba,average='weighted')

print("ROC AUC Score Train:", roc_auc_score_train)
print("ROC AUC Score Test:", roc_auc_score_test)
```

```
ROC AUC Score Train: 0.9998364864278017
ROC AUC Score Test: 0.9781785111444917
```

That means SVM is our best model.

- Visualizations

  WordCloud:



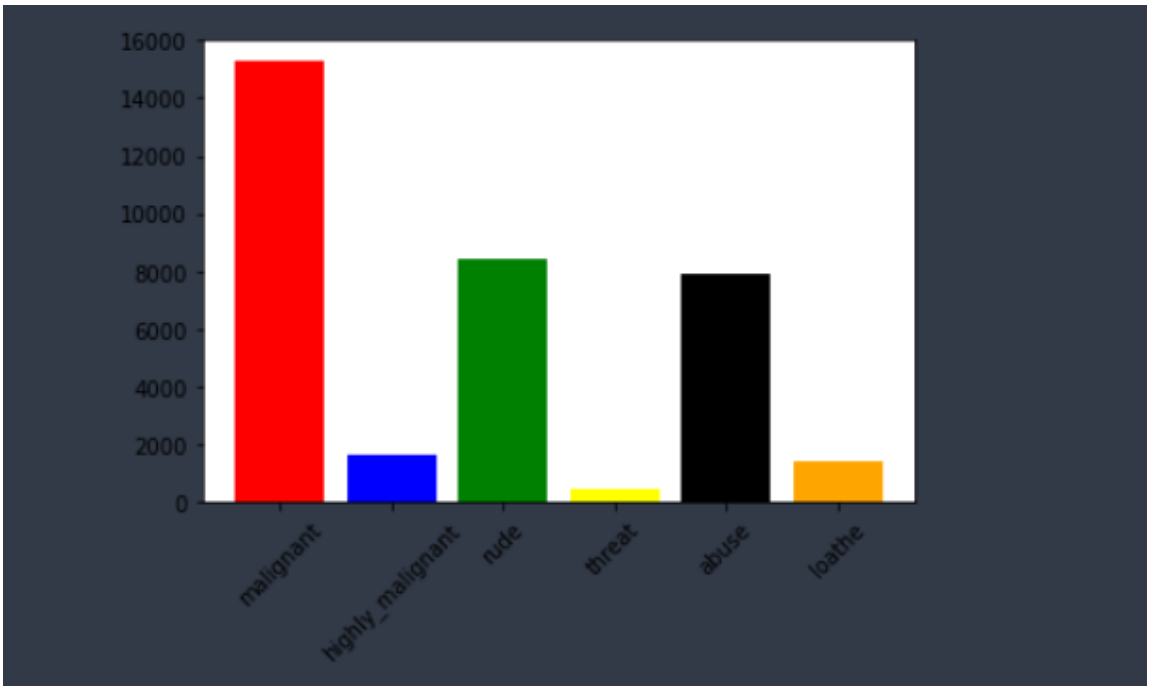

fucksex fucksex die die know
articl
stupid
asshole
fuck fuck
bullshit bullshit
shit shit
page
block
dickhead dickhead
buttseck buttseck
ass ass
suck dick
jew fat
think faggot faggot
fuck yourselfgo
mothjer fucker
nigger nigger
want
im
nippl nippl
fat jew
yourselfgo fuck
wikipedia


jim wale
talk page
forever dont
murder kill
die wikipedia
live forever
die die
die jim wale die
ass ass
dont respect
die di
edi die
fuck
di edi
kill murder
kill kill

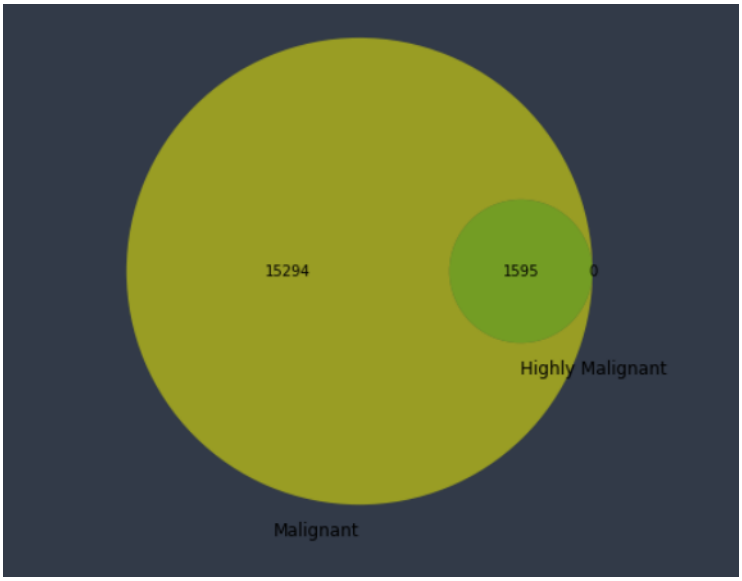

fuck yourselfgo
fucker cockvicker
fuck fuck
ass ass
die die suck dick
faggot huge
huge faggot
asshole
time
faggot faggot
idiot
block
.dickhead dickhead.
yourselfgo fuck
nigger nigger
fat jew
wikipedia
page
mothjer fucker
stupid
life
moron hi
know
jew fathi moron
peopl
hate hate
think


bleachanhero kill
licker fan
huge faggot
gay bunkstev
nigger tommy
niggas fuck
mexican suck
nigger licker
stupid nigger
fat jew
fan nigger
faggot huge
suck mexican
spanish centralistst
bunkstev gay
jew fat
tommy nigger
drink bleachanhero
die die
cunt cunt
fuck nigga
kill drink
nigger nigger
ancestryfuckoffjewish ancestryfuckoffjewish

Bargraph:


16000
14000
12000
10000
8000
6000
4000
2000
0
malignant
highly_malignant
rude
threat
abuse
loathe

Pie Chart:



- Interpretation of the Results

  After all the pre-processing and then visualising the data I find out that many words in comment column is useless and 17% of the data present in the comments column are useless.

CONCLUSION

- Key Findings and Conclusions of the Study

  After all the pre-processing and then visualising the data I find out that many words in comment column is useless and 17% of the data present in the comments column are useless and we need more data to accurately predict the type of comment column as we get an accuracy of 97% after model building which is quite good.

- Learning Outcomes of the Study in respect of Data Science

  As I have already told that this dataset has a lot of outliers and I have to look into each and every column to get some of the data cleaned. As per the visualisation part I plotted Word Cloud, pie chart, bar graph to see the Loud words etc in all the comment and other columns.

- Limitations of this work and Scope for Future Work

  I could get a better idea of how all the column are working and which outer columns needs to be in the dataset. As per the future scope, no there is not much room to grow as I could get the model efficiency to 97% it could be more accurate with more data and on further research.