Malignant Comment Prediction

Submitted by:

Gargi Saha Samanta

ACKNOWLEDGMENT

I would like to express my gratitude towards FlipRobo Technologies for their kind co-operation and encouragement which help me in completion of this project.

I would like to express my special gratitude and thanks to industry persons and my mentor(Ms. Swati Mahaseth) for giving me such attention and time as and whenever required.

INTRODUCTION

• Business Problem Statement

- 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. 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.

• 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.

• Analytical Problem Framing

Dataset Representation:

```
# Load the Train and Test data

df_train=pd.read_csv(r"C:\Users\saman\Downloads\Malignant-Comments-Classifier-Project--1---1-\Malignant Comments Classifier
df_test=pd.read_csv(r"C:\Users\saman\Downloads\Malignant-Comments-Classifier-Project--1---1-\Malignant Comments Classifier P
```

```
df_train.head(50)
```

	id	comment_text	malignant	highly_malignant	rude	threat	abuse	loathe
0	0000997932d777bf	Explanation\nWhy the edits made under my usern...	0	0	0	0	0	0
1	000103f0d9cfb60f	D'aww! He matches this background colour I'm s...	0	0	0	0	0	0
2	000113f07ec002fd	Hey man, I'm really not trying to edit war. It...	0	0	0	0	0	0
3	0001b41b1c6bb37e	"\nMore\nI can't make any real suggestions on ...	0	0	0	0	0	0
4	0001d958c54c6e35	You, sir, are my hero. Any chance you remember...	0	0	0	0	0	0

```
print(df_train.describe())
print(df_test.describe())
```

```
           malignant  highly_malignant           rude         threat  \
count  159571.000000     159571.000000  159571.000000  159571.000000
mean        0.095844          0.009996       0.052948       0.002996
std         0.294379          0.099477       0.223931       0.054650
min         0.000000          0.000000       0.000000       0.000000
25%         0.000000          0.000000       0.000000       0.000000
50%         0.000000          0.000000       0.000000       0.000000
75%         0.000000          0.000000       0.000000       0.000000
max         1.000000          1.000000       1.000000       1.000000

              abuse         loathe
count  159571.000000  159571.000000
mean        0.049364       0.008805
std         0.216627       0.093420
min         0.000000       0.000000
25%         0.000000       0.000000
50%         0.000000       0.000000
75%         0.000000       0.000000
max         1.000000       1.000000
                     id                                       comment_text
count            153164                                             153164
unique           153164                                             153164
top     00001cee341fdb12  Yo bitch Ja Rule is more succesful then you'll...
freq                  1                                                  1
```

So clearly it is a Multi Class classification problem.
In *multi-class classification*, we have one basic assumption that our data can belong to only one label out of all the labels we have.

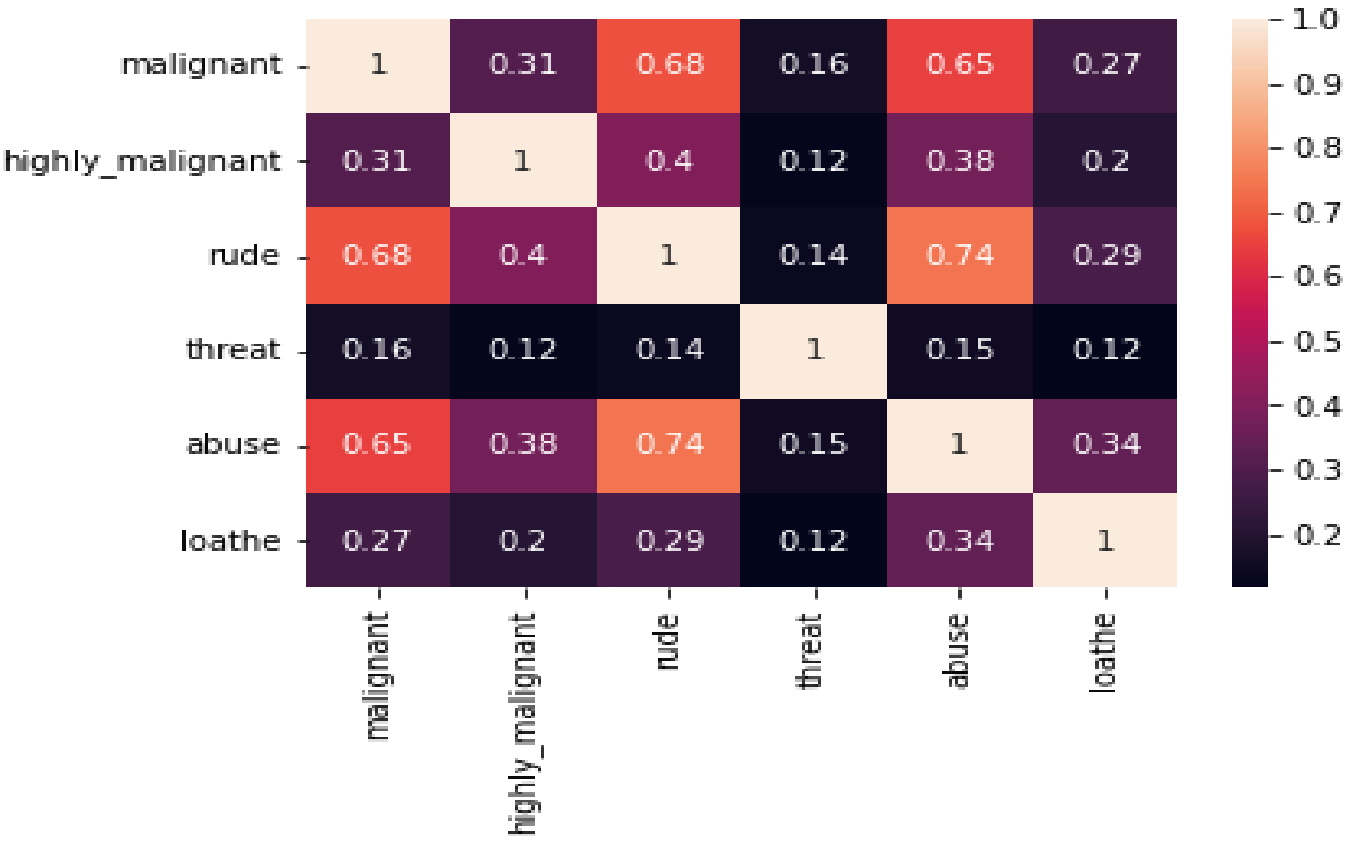
- # Data Sources and their formats & inferences

- Malignant: It is the Label column, which includes values 0 and 1, denoting if the comment is malignant or not.
- Highly Malignant: It denotes comments that are highly malignant and hurtful.
- Rude: It denotes comments that are very rude and offensive.
- Threat: It contains indication of the comments that are giving any threat to someone.
- Abuse: It is for comments that are abusive in nature.
- Loathe: It describes the comments which are hateful and loathing in nature.
- ID: It includes unique Ids associated with each comment text given.
- Comment text: This column contains the comments extracted from various social media platforms.

Observation:

- 2 Independent variables with 6 target variables.
- From the dataset we can infer that it is clearly classification problem.

- **Data Inputs- Logic- Output Relationships**
  - **Correlation** :



Observation:

Malignant, rude and abusive words are highly correlated words for my dataset.

- **Assumption for the problem:**

From the dataset we can infer that it is clearly a Classification problem.

- Hardware and Software Requirements and Tools Used

  Software Used:

  - Jupyter Notebook
  - Ms-Paint
  - MS-PowerPoint
  - MS-Word

  Hardware used:

  - Laptop
  - Good internet connectivity

Model/s Development and Evaluation

- **Testing of Identified Approaches (Algorithms)**

- LogisticRegression()
- DecisionTreeClassifier()
- KNeighborsClassifier()
- RandomForestClassifier()
- AdaBoostClassifier()
- XGBClassifier()

```
# Logistic Regression

LG = LogisticRegression()
#for trainoing data
LG.fit(x_train, y_train)
y_pred_train = LG.predict(x_train)
print('Training accuracy is {}'.format(accuracy_score(y_train, y_pred_train)))

# for testing data
y_pred_test = LG.predict(x_test)
print('Test accuracy is {}'.format(accuracy_score(y_test,y_pred_test)))
print(confusion_matrix(y_test,y_pred_test))
print(classification_report(y_test,y_pred_test))
```

```
# DecisionTree Regression

DTC = DecisionTreeClassifier()
#for trainoing data
DTC.fit(x_train, y_train)
y_pred_train = DTC.predict(x_train)
print('Training accuracy is {}'.format(accuracy_score(y_train, y_pred_train)))

# for testing data
y_pred_test = DTC.predict(x_test)
print('Test accuracy is {}'.format(accuracy_score(y_test,y_pred_test)))
print(confusion_matrix(y_test,y_pred_test))
print(classification_report(y_test,y_pred_test))
```

```
# KNeighborsClassifier

knn = KNeighborsClassifier()
#for trainoing data
knn.fit(x_train, y_train)
y_pred_train = knn.predict(x_train)
print('Training accuracy is {}'.format(accuracy_score(y_train, y_pred_train)))

# for testing data
y_pred_test = knn.predict(x_test)
print('Test accuracy is {}'.format(accuracy_score(y_test,y_pred_test)))
print(confusion_matrix(y_test,y_pred_test))
print(classification_report(y_test,y_pred_test))
```

```
# Random Forest Regression

RF = RandomForestClassifier()
#for trainoing data
RF.fit(x_train, y_train)
y_pred_train = RF.predict(x_train)
print('Training accuracy is {}'.format(accuracy_score(y_train, y_pred_train)))

# for testing data
y_pred_test = RF.predict(x_test)
print('Test accuracy is {}'.format(accuracy_score(y_test,y_pred_test)))
print(confusion_matrix(y_test,y_pred_test))
print(classification_report(y_test,y_pred_test))
```

```
# xgboost Regression

xgb = XGBClassifier()
#for trainoing data
xgb.fit(x_train, y_train)
y_pred_train = xgb.predict(x_train)
print('Training accuracy is {}'.format(accuracy_score(y_train, y_pred_train)))

# for testing data
y_pred_test = xgb.predict(x_test)
print('Test accuracy is {}'.format(accuracy_score(y_test,y_pred_test)))
print(confusion_matrix(y_test,y_pred_test))
print(classification_report(y_test,y_pred_test))
```

```
# AdaBoostClassifier Regression

ada = AdaBoostClassifier()
#for trainoing data
ada.fit(x_train, y_train)
y_pred_train = ada.predict(x_train)
print('Training accuracy is {}'.format(accuracy_score(y_train, y_pred_train)))

# for testing data
y_pred_test = ada.predict(x_test)
print('Test accuracy is {}'.format(accuracy_score(y_test,y_pred_test)))
print(confusion_matrix(y_test,y_pred_test))
print(classification_report(y_test,y_pred_test))
```

Accuracy of the Best Model (Random Forest Classifier)

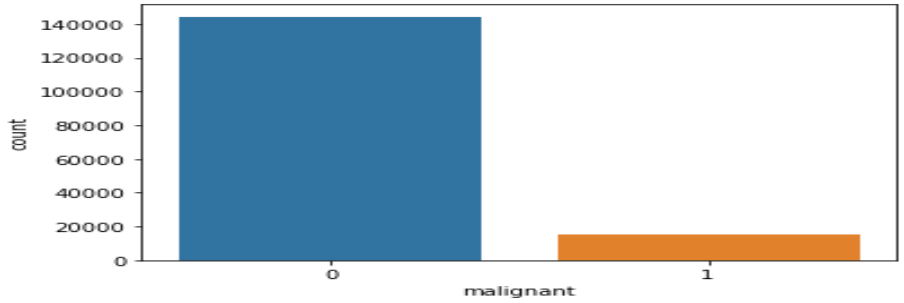
```
Training accuracy is 0.9988809210467416
Test accuracy is 0.9551512366310161
[[42399   551]
 [ 1596  3326]]
              precision    recall  f1-score   support

           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
```

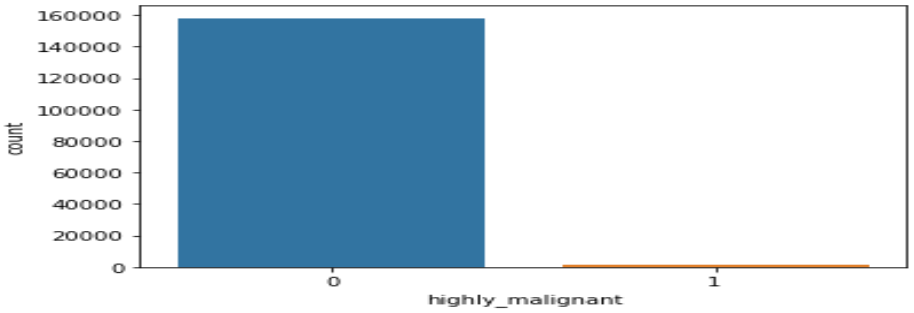
OBSERVATION:

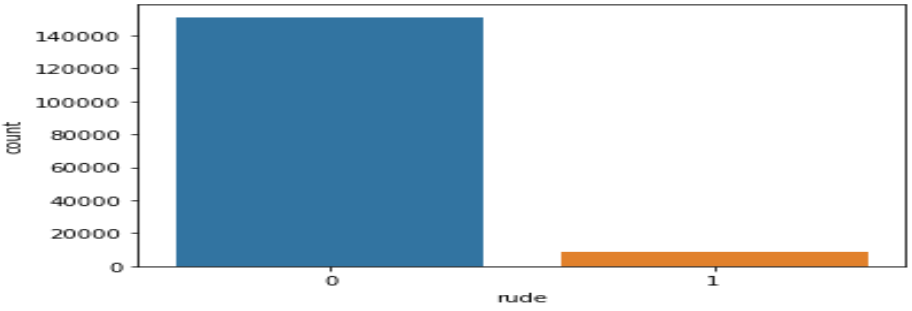
- Finally we have saved the Random Forest Classifier Model.

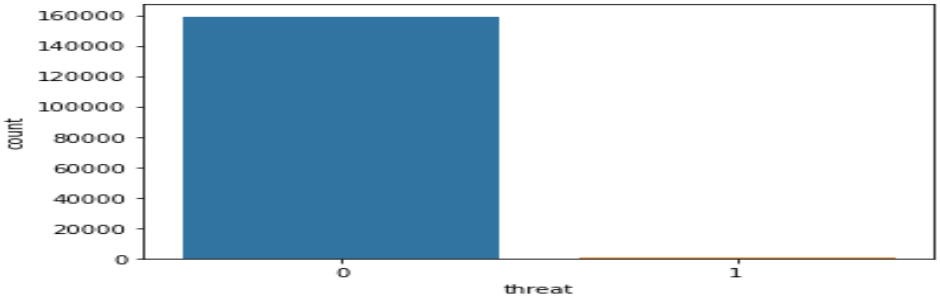
- **Visualizations**

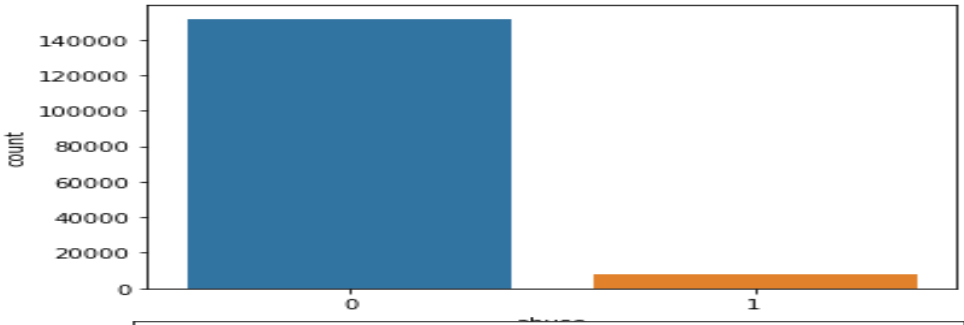
Univariate Analysis of categorical variables:

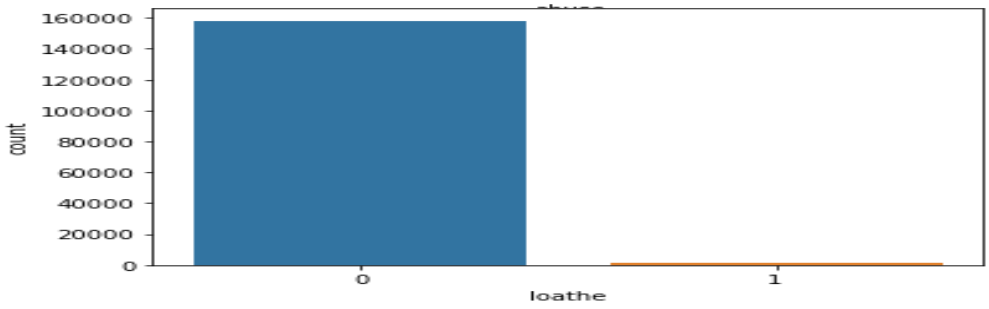












Observation :

- The malignant,highly malignant and rude data are highly imbalaced and seems that there are not much comments with loud data.
- Malignant data are most as compared to rude and highly malignant data.

- Similarly for threat,abuse and loathe also data are highly imbalaced and seems that there are not much comments with loud data.
- Abusive data are more than threat or loathe.

Interpretation of the Results

- Hence we can go with normal Random Forest Classifier model with is also giving good accuracy.
- This model may help to solve real life problems by identifying Harsh ,hurting and loud words and preventing it hurting to the readers.

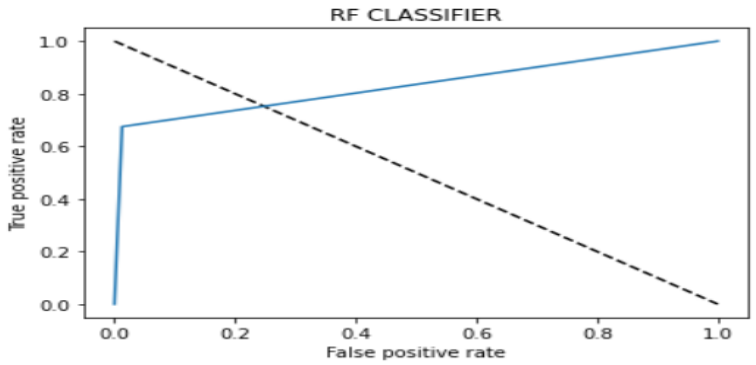
CONCLUSION

Finally we have saved the Random Forest Classifier Model.

Learning Outcomes of the Study in respect of Data Science

- Hence we can go with normal Random Forest Classifier model which is also giving good accuracy.
- Finally we have saved the Random Forest Regressor Model.

```
fpr,tpr,thresholds=roc_curve(y_test,y_pred_test)
roc_auc=auc(fpr,tpr)
plt.plot([0,1],[1,0],'k--')
plt.plot(fpr,tpr,label = 'RF Classifier')
plt.xlabel('False positive rate')
plt.ylabel('True positive rate')
plt.title('RF CLASSIFIER')
plt.show()
```



ROC_AUV CURVE

- **Limitations of this work and Scope for Future Work**

- In future this machine learning model using NLP may bind with various website which can provide real time data for malignant comment prediction.
- This project predicts the loud words and may predict it reaching to their user so that they are not hurt also 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.
- It may avoid the readers to spread negativity with their harsh words.

****