MALIGNANT COMMENTS CLASSIFICATION

Submitted By:

Ishmeet Kaur Sahota

ACKNOWLEDGMENT

I want to express my sincere thanks to Flip Robo Techniques and my SME Khusboo Garg who helped me in every possible way that she could and guide me through new things. without whom I won’t have been able to complete this project.

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 a problem across multiple platforms, there is a lack of models for online hate detection. This can take a toll on anyone and affect them mentally leading to depression, mental illness, self-hatred, and suicidal thoughts.

• Review of Literature:-

Our goal in this project is to build a prototype of online hate and abuse comment classifier which can be used to classify hate and offensive comments so that they can be controlled and restricted from spreading hatred and cyber bullying. These are some columns in our data 'Id', 'Comments', 'Malignant', 'Highly malignant', 'Rude', 'Threat', 'Abuse', and 'Loathe'.

• Motivation for the Problem Undertaken:-

There has been a remarkable increase in the cases of cyber bullying and trolls on various social media platforms. Many celebrities and influences are facing backlash 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.

Analytical Problem Framing

• Mathematical/ Analytical Modelling of the Problem :-

The Dataset contains 159571 rows and 8 columns. The columns contains both object data type and integer data type., i have seen the value of each columns by applying value_count method . Then I have used describe method . Is Null. Sum for checking the Nan values and getting their sums. Applied some visualization techniques for better understanding of the data. Then check the skewness of the data. I have replaced some numbers with numbers and data containing addresses, email, etc with some meaningful data by applying replace method .I have Stored all the targets in one single column. Convert text into vectors then split the data using train test split and used 4 different classification models.

• Data Sources and their formats :-

The Dataset contains 159571 rows and 8 columns respectively. containing all the necessary details.

These are some columns in our data 'Id', 'Comments', 'Malignant', 'Highly malignant', 'Rude', 'Threat', 'Abuse', and 'Loathe'.

• Data Preprocessing Done :-

The dataset contain some of object type data soI have replaced some numbers with numbers and data containing addresses, email, etc with some meaningful data by applying replace method. And Stored all the targets in one single column. Convert text into vectors and used all these methods for better model prediction.

• State the set of assumptions (if any) related to the problem under consideration: -

In The dataset we have taken assumption as : In every columns : 0 = NO , 1 = YES

• *Hardware and Software Requirements and Tools Used:-*

Libraries have I have used for data cleaning , Visualization and model building.

```
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
%matplotlib inline
import warnings
warnings.filterwarnings('ignore')
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score,classification_report,confusion_matrix,f1_score
from sklearn.metrics import roc_curve,roc_auc_score,auc
from sklearn.metrics import plot_roc_curve
from sklearn.linear_model import LogisticRegression
from sklearn.tree import DecisionTreeClassifier
from sklearn.metrics import roc_curve, roc_auc_score
from sklearn.metrics import plot_roc_curve
from sklearn.ensemble import RandomForestClassifier
from sklearn.neighbors import KNeighborsClassifier
```

Model/s Development and Evaluation

• *Identification of possible problem-solving approaches (methods):-*

I have used these approaches

pandas , numpy ,seaborn ,matplotlib.pyplot ,%matplotlib inline , import warnings , WordNetLemmatizer , stopwords , nltk , TfidfVectorizer.

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

from sklearn.feature_extraction.text import TfidfVectorizer

from sklearn.model_selection import train_test_split

from sklearn.metrics import accuracy_score,classification_report,confusion_matrix,f1_score

from sklearn.metrics import roc_curve,roc_auc_score,auc

from sklearn.metrics import plot_roc_curve

from sklearn.linear_model import LogisticRegression

from sklearn.tree import DecisionTreeClassifier

from sklearn.metrics import roc_curve, roc_auc_score

from sklearn.metrics import plot_roc_curve

from sklearn.ensemble import RandomForestClassifier

from sklearn.neighbors import KNeighborsClassifier

• Run and Evaluate selected models :-

 # KNeighbors Classifier

```
Training accuracy     : 0.9235053581500282
************************************************************************************************
Test accuracy         : 0.9181262729124237
************************************************************************************************
confusion matrix      : [[28491   119]
 [ 2494   811]]
************************************************************************************************
Classification Report :               precision    recall  f1-score   support

           0       0.92      1.00      0.96     28610
           1       0.87      0.25      0.38      3305

    accuracy                           0.92     31915
   macro avg       0.90      0.62      0.67     31915
weighted avg       0.91      0.92      0.90     31915
```

Logistic Regression

```
Training accuracy     : 0.9598138747884941
****************************************************************************************************
Test accuracy         : 0.9561961460128466
****************************************************************************************************
confusion matrix      : [[28458   152]
 [ 1246  2059]]
****************************************************************************************************
Classification Report :               precision    recall  f1-score   support

           0       0.96      0.99      0.98     28610
           1       0.93      0.62      0.75      3305

    accuracy                           0.96     31915
   macro avg       0.94      0.81      0.86     31915
weighted avg       0.96      0.96      0.95     31915
```

DecisionTreeClassifier

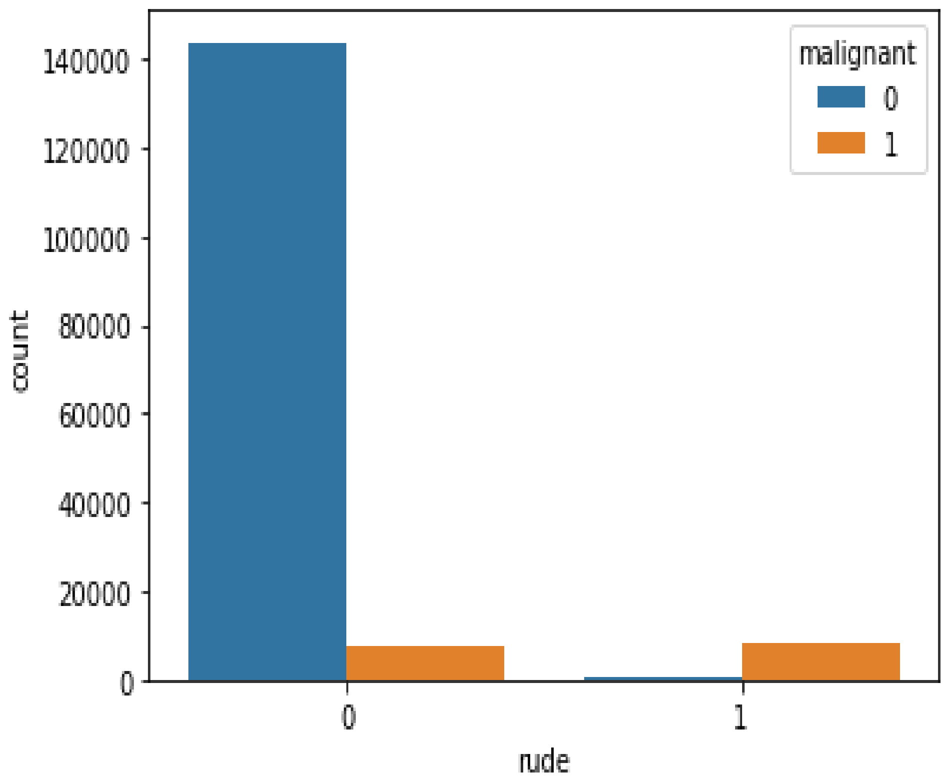
```
Training accuracy     : 0.9235053581500282
****************************************************************************************************
Test accuracy         : 0.9181262729124237
****************************************************************************************************
confusion matrix      : [[28491   119]
 [ 2494   811]]
****************************************************************************************************
Classification Report :               precision    recall  f1-score   support

           0       0.92      1.00      0.96     28610
           1       0.87      0.25      0.38      3305

    accuracy                           0.92     31915
   macro avg       0.90      0.62      0.67     31915
weighted avg       0.91      0.92      0.90     31915
```

RandomForestClassifier

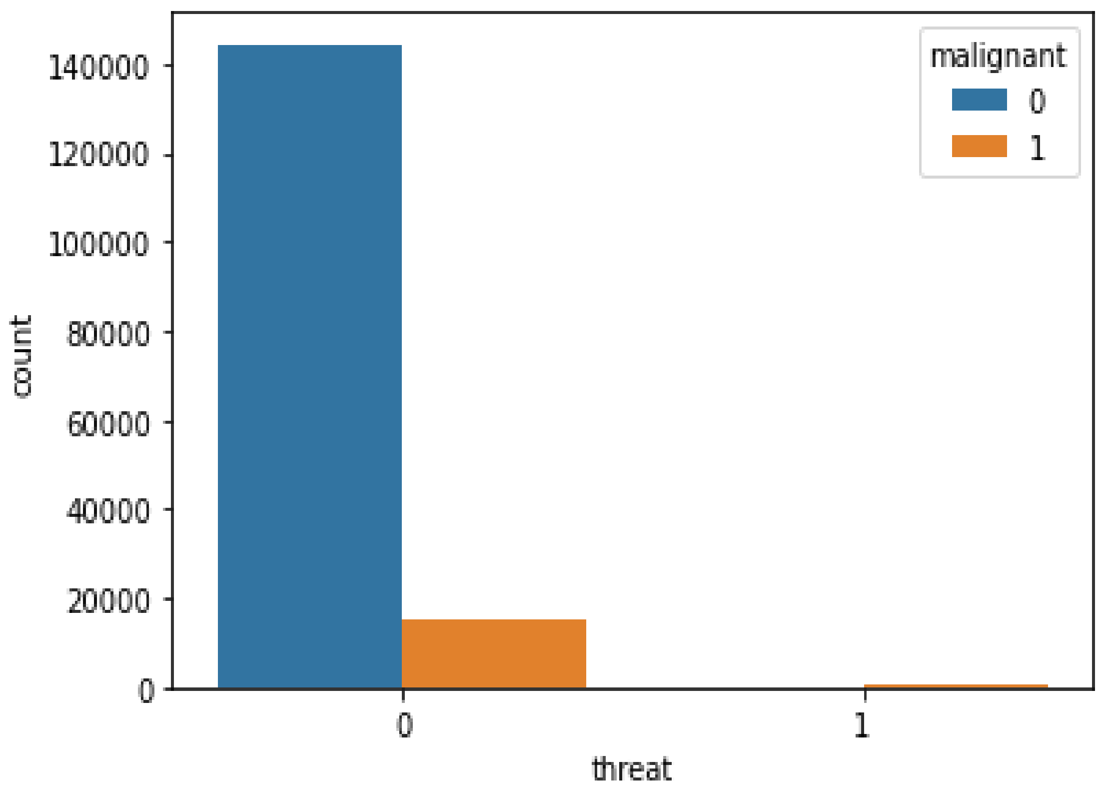
```
Training accuracy     : 0.9987074638089867
****************************************************************************************************
Test accuracy         : 0.9571361428795238
****************************************************************************************************
confusion matrix      : [[28270   340]
 [ 1028  2277]]
****************************************************************************************************
Classification Report :               precision    recall  f1-score   support

           0       0.96      0.99      0.98     28610
           1       0.87      0.69      0.77      3305

    accuracy                           0.96     31915
   macro avg       0.92      0.84      0.87     31915
weighted avg       0.96      0.96      0.95     31915
```

As we can see both Logistic regression and) RandomForestClassifier has the best accuracy score (0.96).

Visualization

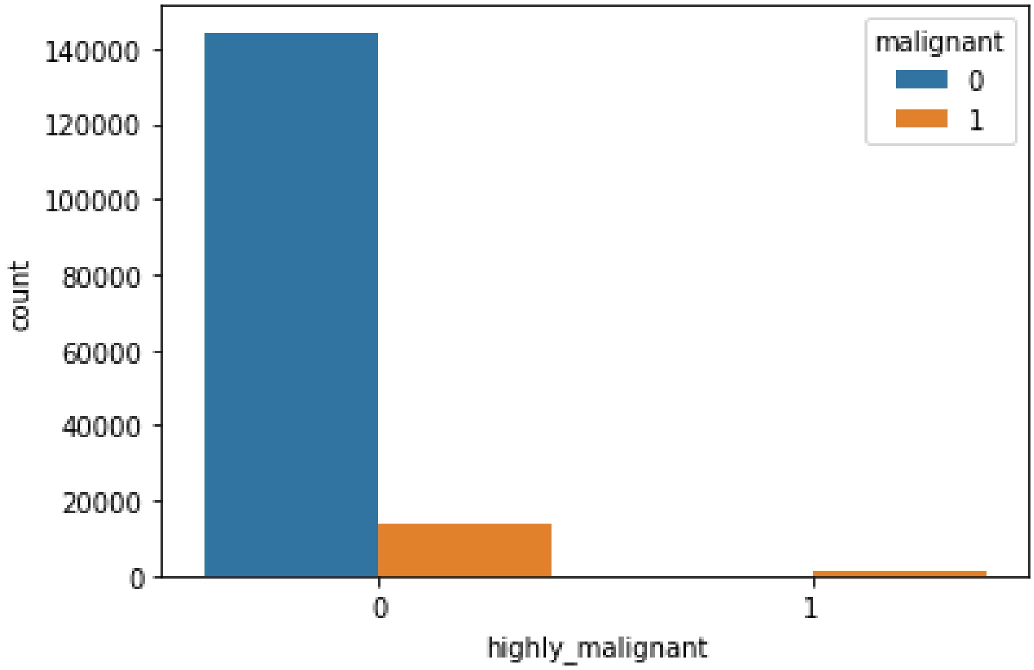


Used countplot for comparing 'rude' and 'malignant' columns.

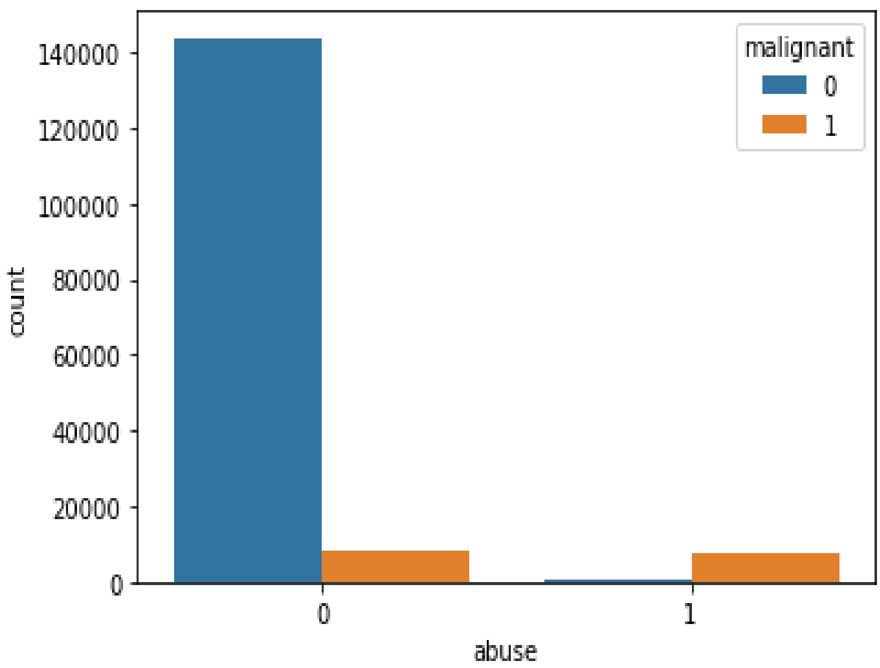


Compared ‘threat’ and ‘malignant’

Compared ‘Highly malignant’ and ‘malignant’

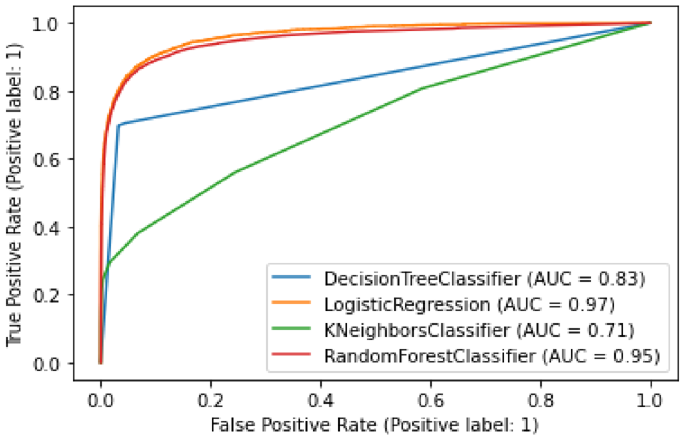


Compared ‘Abuse’ and ‘malignant’

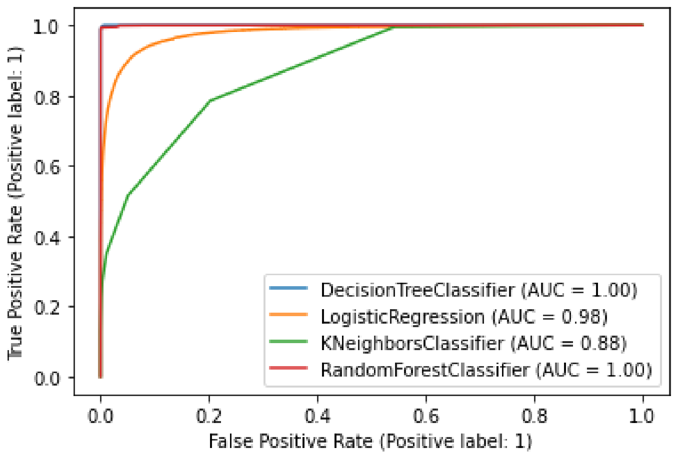


Comparing the score of different models on training data that which model fit best

- **Training data**



- **Testing data**



Checking the score of different models on testing data that which model fit best Logistic Regression fits best among all other models As Logistic Regression is given: 0.97 score while testing 0.98 score while training.

• Interpretation of the Results:-

After visualizing the data I have concluded that under all these columns : -

• 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.

As value of 0 as No are more than value of 1 as Yes in these columns. Majority of these texts 0 that is are Not malignant. Logistic Regression fits best among all other models As Logistic Regression is given: 0.97 score while testing 0.98 score while training.

CONCLUSION

• Key Findings and Conclusions of the Study :-

As value of 0 as No are more the n value of 1 as Yes in these columns. Majority of these texts 0 that is are Not malignant. Logistic Regress ion fits best among all other models As Logistic Regression is given: 0.97 score while testing 0.98 score while training.

• Learning Outcomes of the Study in respect of Data Science:-

As data contain some object type data with the help of replace method, I have replaced some data containing addresses, e mail, etc with meaningful data. And replaced numbers with numbers. Store all the targets in one single column. Convert text into vectors and used all these methods for better model prediction. As value of 0 as No are more than value of 1 as Yes in these columns. Majority of these texts 0 that is are Not malignant. Logistic Regression fits best among all other models As Logistic Regression is given: 0.97 score while testing 0.98 score while training.