Malignant comments Prediction

Submitted By:

Aman Saxena

ACKNOWLEDGMENT

- https://scikit-learn.org/stable/ - For the libraries used in the project.
- Rest project is done by myself only.

INTRODUCTION

- Business Problem Framing

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

For more understanding we can simply correlate it with the project of Spam Mail detection.

- Motivation for the Problem Undertaken

In this project we have to build the model that will predict the Comments which are malignant and are abusive to the community.

Analytical Problem Framing

- Mathematical/ Analytical Modeling of the Problem

First of all we load the training and the test file into our jupyter notebook. Then we checked the shape of the dataset after that we described the dataset and collect some information. Then we checked for the null values in the training dataset and we found no null values in the datasets. Then the visualization of the dataset is done and we grab the information from that , like how may comments are malignant , abusive, loathe, rude, threat, etc.

After that data preprocessing is done in which we cleaned the comments in different ways i.e. convert all messages to lower case, Replace emailaddresses with 'emailaddress', Replace Urls with 'webaddress', Replace money symbols with 'moneysymbols', Replace 10 digit phone numbers, Replace numbers with 'number', Remove Punctuatuations, Remove White space between terms with single space, Remove leading and trailing white space, Remove Stopwords . After that Lematization is done . And then the new column is added ‘clean length’ after the punctuations and stopwords were removed.

After that we find that which words are the loud words or bad words means getting the sense of these words. After that we have done the sum of the dependent variables and created the final target column.

Then we converted the text into the vectors using TF-IDF. Then we divide the dataset into dependent and independent variables. After that dataset is trained with different models and prediction is made and found that the Logistic Regression is giving the best accuracy and made the ROC-AUC curve. Then the preprocessing of the test data is done and the prediction is made using our best model.

. Data Preprocessing Done

we cleaned the comments in different ways i.e. convert all messages to lower case, Replace emailaddresses with 'emailaddress', Replace Urls with 'webaddress', Replace money symbols with 'moneysymbols', Replace 10 digit phone numbers, Replace numbers with 'number', Remove Punctuatuations, Remove White space between terms with single space, Remove leading and trailing white space, Remove Stopwords . After that

Lematization is done . And then the new column is added ‘clean length’ after the punctuations and stopwords were removed.

After that we find that which words are the loud words or bad words means getting the sense of these words. After that we have done the sum of the dependent variables and created the final target column.

Then we converted the text into the vectors using TF-IDF. Then we divide the dataset into dependent and independent variables. After that dataset is trained with different models and prediction is made and found that the Logistic Regression is giving the best accuracy and made the ROC-AUC curve. Then the preprocessing of the test data is done and the prediction is made using our best model.

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

- import numpy as np
- import pandas as pd
- import seaborn as sns
- import matplotlib.pyplot as plt
- import warnings
- warnings.filterwarnings('ignore')
- from nltk.stem import WordNetLemmatizer
- import nltk
- from nltk.corpus import stopwords
- import string

Model/s Development and Evaluation

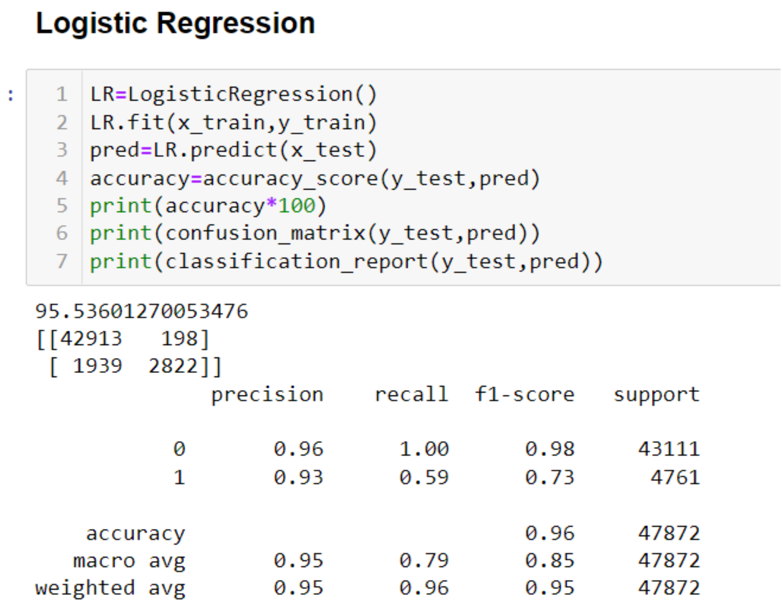
- Identification of possible problem-solving approaches (methods)

Since the problem is of NLP and we have to predict that wich comment is malignant or not.

Testing of Identified Approaches (Algorithms)

- Logistic Regression
- MultinomialNB
- Random forest Classifier
- Decision Tree Classifier

- Run and Evaluate selected models
  - Logistic Regression



Logistic Regression

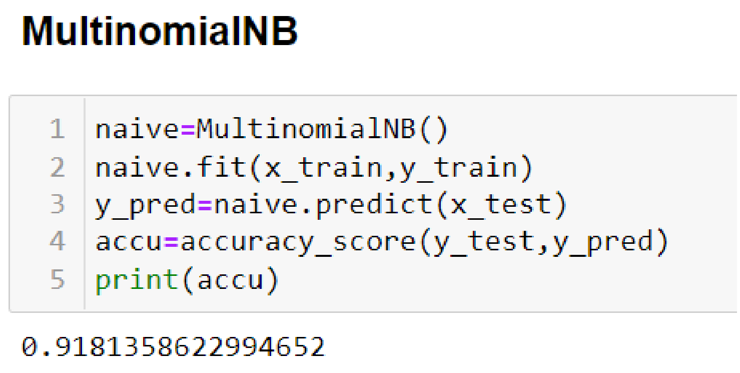
```
LR=LogisticRegression()
LR.fit(x_train,y_train)
pred=LR.predict(x_test)
accuracy=accuracy_score(y_test,pred)
print(accuracy*100)
print(confusion_matrix(y_test,pred))
print(classification_report(y_test,pred))
```

```
95.53601270053476
[[42913   198]
 [ 1939  2822]]
              precision    recall  f1-score   support

           0       0.96      1.00      0.98     43111
           1       0.93      0.59      0.73      4761

    accuracy                           0.96     47872
   macro avg       0.95      0.79      0.85     47872
weighted avg       0.95      0.96      0.95     47872
```

- ➢ MultinomialNB

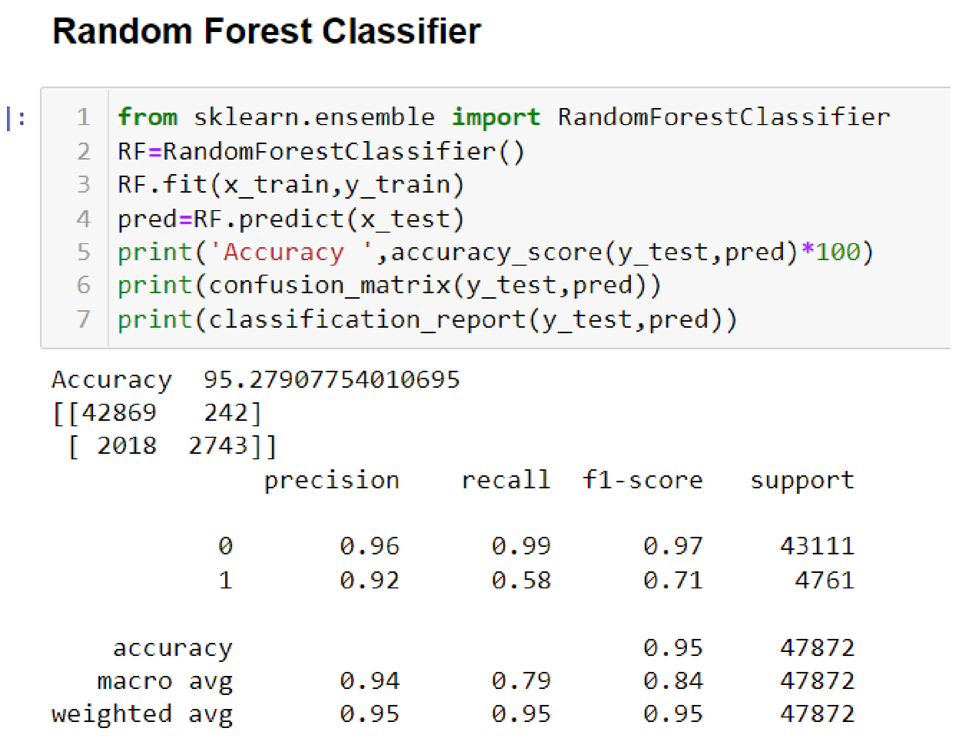


MultinomialNB

```
naive=MultinomialNB()
naive.fit(x_train,y_train)
y_pred=naive.predict(x_test)
accu=accuracy_score(y_test,y_pred)
print(accu)
```

```
0.9181358622994652
```

- ➢ Random Forest Classifier



Random Forest Classifier

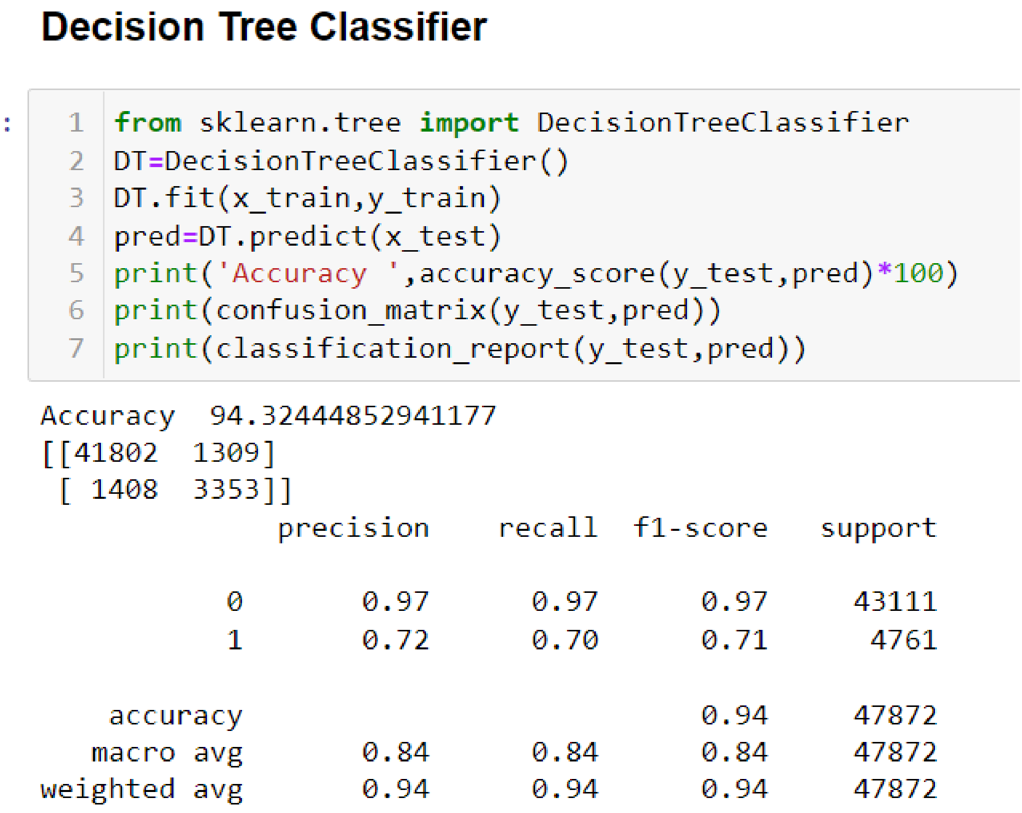
```
from sklearn.ensemble import RandomForestClassifier
RF=RandomForestClassifier()
RF.fit(x_train,y_train)
pred=RF.predict(x_test)
print('Accuracy ',accuracy_score(y_test,pred)*100)
print(confusion_matrix(y_test,pred))
print(classification_report(y_test,pred))
```

```
Accuracy  95.27907754010695
[[42869   242]
 [ 2018  2743]]
              precision    recall  f1-score   support

           0       0.96      0.99      0.97     43111
           1       0.92      0.58      0.71      4761

    accuracy                           0.95     47872
   macro avg       0.94      0.79      0.84     47872
weighted avg       0.95      0.95      0.95     47872
```

- Decision Tree Classifier



Decision Tree Classifier

```python
from sklearn.tree import DecisionTreeClassifier
DT=DecisionTreeClassifier()
DT.fit(x_train,y_train)
pred=DT.predict(x_test)
print('Accuracy ',accuracy_score(y_test,pred)*100)
print(confusion_matrix(y_test,pred))
print(classification_report(y_test,pred))
```

```
Accuracy  94.32444852941177
[[41802  1309]
 [ 1408  3353]]
              precision    recall  f1-score   support

           0       0.97      0.97      0.97     43111
           1       0.72      0.70      0.71      4761

    accuracy                           0.94     47872
   macro avg       0.84      0.84      0.84     47872
weighted avg       0.94      0.94      0.94     47872
```

- Key Metrics for success in solving problem under consideration

accuracy_score → for calculating the accuracy

CONCLUSION

First of all we load the training and the test file into our jupyter notebook. Then we checked the shape of the dataset after that we described the dataset and collect some information. Then we checked for the null values in the training dataset and we found no null values in the datasets. Then the visualization of the dataset is done and we grab the information from that , like how may comments are malignant , abusive, loathe, rude, threat, etc.

After that data preprocessing is done in which we cleaned the comments in different ways i.e. convert all messages to lower case, Replace emailaddresses with 'emailaddress', Replace Urls with 'webaddress', Replace money symbols with 'moneysymbols', Replace 10 digit phone numbers, Replace numbers with 'number', Remove Punctuatuations, Remove White space between terms with single space, Remove leading and trailing white space, Remove Stopwords . After that Lematization is done . And then the new column is added ‘clean length’ after the punctuations and stopwords were removed.

After that we find that which words are the loud words or bad words means getting the sense of these words. After that we have done the sum of the dependent variables and created the final target column.

Then we converted the text into the vectors using TF-IDF. Then we divide the dataset into dependent and independent variables. After that dataset is trained with different models and prediction is made and found that the Logistic Regression is giving the best accuracy and made the ROC-AUC curve. Then the preprocessing of the test data is done and the prediction is made using our best model.