FLIP ROBO

Malignant Comments Classifier Model

Submitted by:

RAHUL SINGH

ACKNOWLEDGMENT

Malignant Comments Classifier Model Project includes the details about the proliferation of social media which enables people to express their opinions widely online using there comments. The Social media platform like twitter, Instagram, Facebook ete. This Comment text are basically expression of people in from of words. In that some of comments are malignant , rude, threat, abuse, loathe.

Malignant Comments Classifier Model is basically a Natural language processing Model which is basically used to predict the type of comments written by user on the social media platform.

INTRODUCTION

- Business Problem Framing

  The Malignant Comments Classifier model is related to proliferation Social media .it is basically related to people to express their opinions widely online in there commets.This comments consist of Malignant comment, rude, threat, abuse etc.

- Conceptual Background of the Domain Problem

  The model is basically related to the Social media domain and consist of comments of people to express their opinions widely online.

  The model is basically a natural language processing model where model is build on based on understanding comments.

- Motivation for the Problem Undertaken

  The Malignant Comments Classifier Model is consist of only text comments which has been encoded in to binary format using multiple method like bag of words, TF/IDF, Stopwords in Natural language processing Model building. These methods used to filter the comments and then using steaming or lemmatizing to convert this words into binary format.

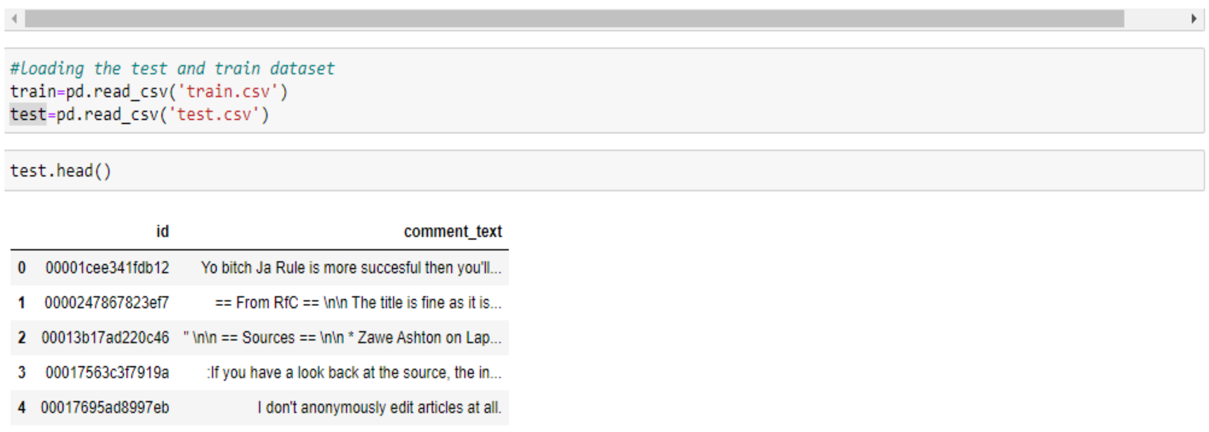
Analytical Problem Framing

- Mathematical/ Analytical Modeling of the Problem

  The Dataset basically train and test is in csv format which consist of all details i.e rows and columns. Analysing the data, there data type, which columns is impotent. Plotting the multiple plots like bar plot, WordCloud,heat map to analyse the data and which features are important. After that using multiple methods to filter the comments and then using lemmanizing to covert word into vectors.

- Data Sources and their formats

  Data(train and test) is in the CSV format.the data has been imported into the data frame using the python library.

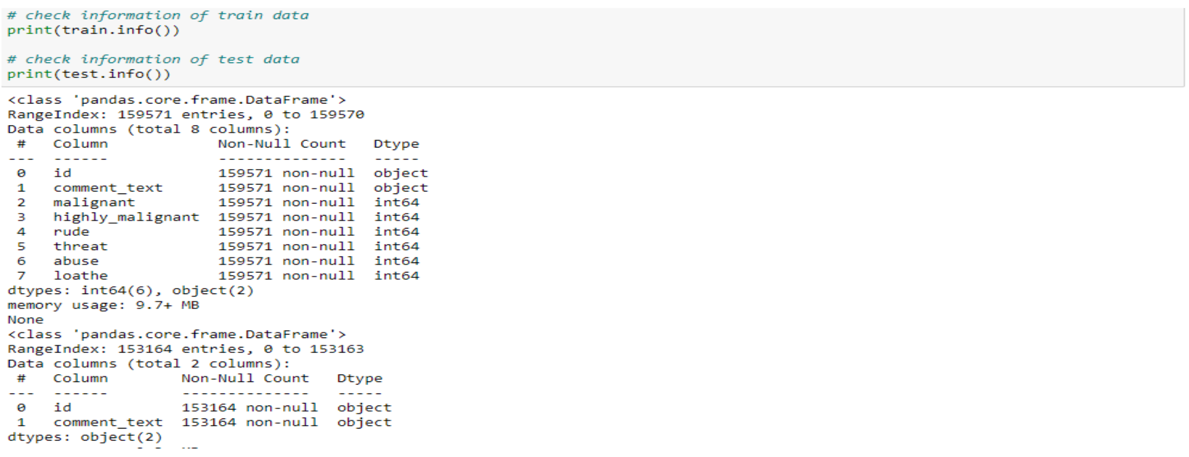


```
#loading the test and train dataset
train=pd.read_csv('train.csv')
test=pd.read_csv('test.csv')
```

```
test.head()
```

	id	comment_text
0	00001cee341fdb12	Yo bitch Ja Rule is more succesful then you'll...
1	0000247867823ef7	== From RfC == \n\n The title is fine as it is...
2	00013b17ad220c46	" \n\n == Sources == \n\n * Zawe Ashton on Lap...
3	00017563c3f7919a	:If you have a look back at the source, the in...
4	00017695ad8997eb	I don't anonymously edit articles at all.

The data having the different datatypes i.e. int, and object.



```
# check information of train data
print(train.info())

# check information of test data
print(test.info())
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 159571 entries, 0 to 159570
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)
memory usage: 9.7+ MB
None
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 153164 entries, 0 to 153163
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)
```

- Data Preprocessing Done

  The Data Preprocessing consist of below mentioned points:

  1. Checking the null values form the dataset, if null value is available them replace them with the suitable values.
  2. Describing the data.
  3. The train Data basically consist of only text_comments as feature which having only text data in form of people comments.
  4. Using multiple methods to filter the comments. Using stopwords, lower then lemmanizing to filter the words.
  5. After that using TF-IDF method to convert words into the vectors i.e basically in the binary format to build the model.
  6. This operation has been done on both train and test data separately.
  7. Plotting the multiple plots like bar plot,heat map, WordCloud to check the data and type of comments used for different words.
  8. There are multiple targets so combing all the target as one label and using that label to build the model.
  9. This is basically a NLP model so there is no need to check skewness and outliers.
  10. Separating the features and target to build the model.
  11. Checking the data imbalanced issue on the target variable using value counts.
  12. Here imbalanced issue is present and using upsampling SMOTE to solve the data imbalanced issue.

- Data Inputs- Logic- Output Relationships

  There are only one feature is available i.e Text comments which consist of comments in text format and multiple labels which has been converted to one target. So using the comment doing the prediction type on comments.

- Hardware and Software Requirements and Tools Used

The Tool used to build the model is anaconda jupyter.

Model/s Development and Evaluation

- Identification of possible problem-solving approaches (methods)

  Multiple problem faced during the model building. All are listed bellows.

  1. Checking for the unwanted columns analyse the relation with target and removal of that features.
  2. Checking the null values and replace it.
  3. Filtering the text using the stopwords and lemmanization.
  4. Using TP-ICP to convert cleaned word data into vector data.
  5. Performing both separately on the test and train data.
  6. Combining the multiple targets into single label.
  7. Checking of Data imbalanced issue on the target and then solve the data imbalnced suing upsampling SMOTE.

- Testing of Identified Approaches (Algorithms)

  This is a Natural language processing classification problem so multiple classification model has been used for the model building and basically training and testing the data on multiple models to select the best model.

  The multiple models are used as given below:

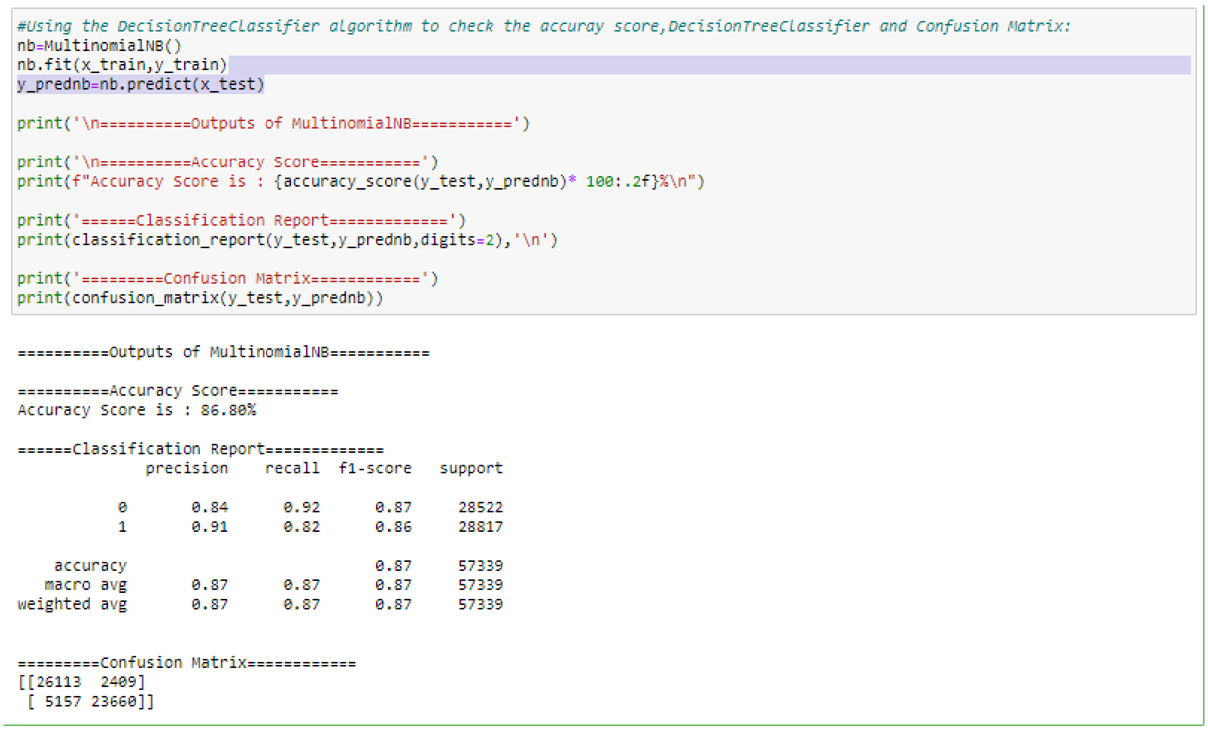
  1. MultinomialNB
  2. Random forest classifier

- ## Run and Evaluate selected models

  The model is a Natural language processing classification model so all the classification model building algorithm has been used to build the model and across all the model the best one has been selected as final model.

  The different models which has been used has been listed below:

  1. MultinomialNB Classifier: The multinomial Naive Bayes classifier is suitable for classification with discrete features (e.g., word counts for text classification). The multinomial distribution normally requires integer feature counts.



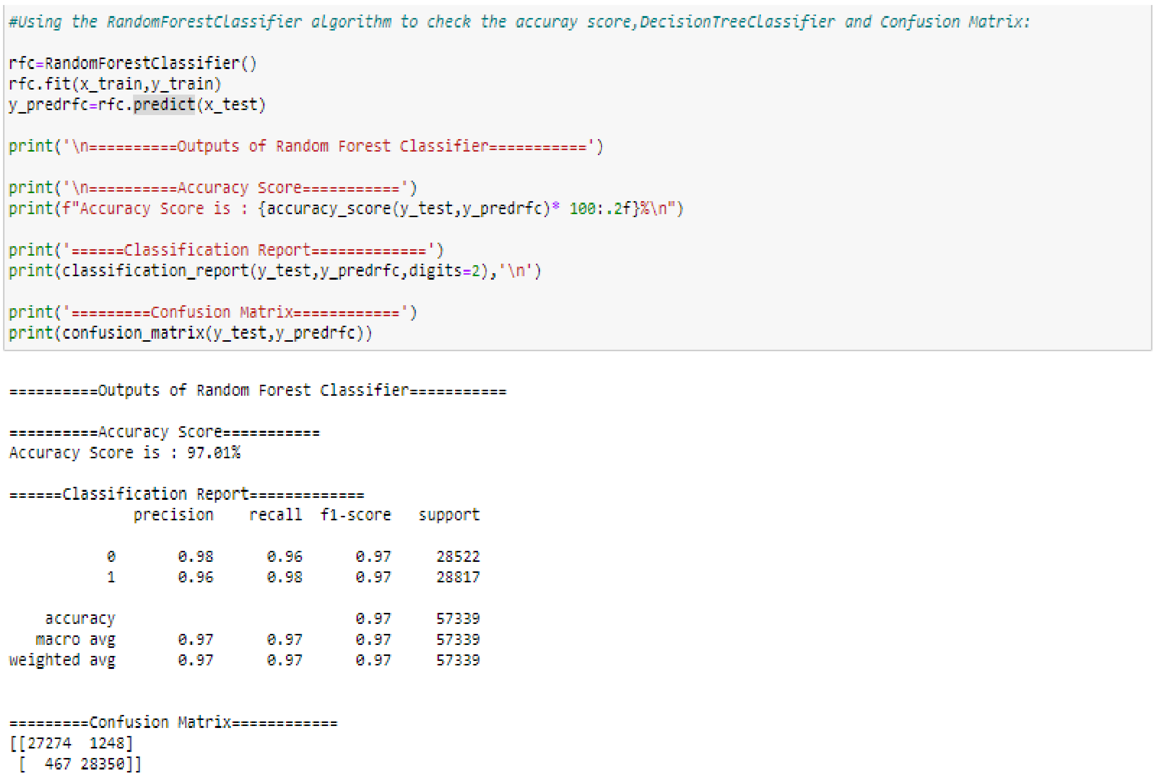
```
#Using the DecisionTreeClassifier algorithm to check the accuray score,DecisionTreeClassifier and Confusion Matrix:
nb=MultinomialNB()
nb.fit(x_train,y_train)
y_prednb=nb.predict(x_test)

print('\n==========Outputs of MultinomialNB===========')

print('\n==========Accuracy Score===========')
print(f"Accuracy Score is : {accuracy_score(y_test,y_prednb)* 100:.2f}%\n")

print('======Classification Report=============')
print(classification_report(y_test,y_prednb,digits=2),'\n')

print('=========Confusion Matrix============')
print(confusion_matrix(y_test,y_prednb))
```

```
==========Outputs of MultinomialNB===========

==========Accuracy Score===========
Accuracy Score is : 86.80%

======Classification Report=============
              precision    recall  f1-score   support

           0       0.84      0.92      0.87     28522
           1       0.91      0.82      0.86     28817

    accuracy                           0.87     57339
   macro avg       0.87      0.87      0.87     57339
weighted avg       0.87      0.87      0.87     57339


=========Confusion Matrix============
[[26113  2409]
 [ 5157 23660]]
```

After training the data in MultinomialNB the accuracy score of the model is 86.80%.
The Classification Report consist of all the details related to accuracy score, precision, recall and f1- score of MultinomialNB model.

It also consist of Confusion matrix values i.e True Positive, False Positive, True Negative and False Negative values MultinomialNB model.

2. Random forest Classifier: Random forests is a supervised learning algorithm. Random forests creates decision trees on randomly selected data samples, gets prediction from each tree and selects the best solution by means of voting.



```
#Using the RandomForestClassifier algorithm to check the accuray score,DecisionTreeClassifier and Confusion Matrix:

rfc=RandomForestClassifier()
rfc.fit(x_train,y_train)
y_predrfc=rfc.predict(x_test)

print('\n==========Outputs of Random Forest Classifier===========')

print('\n==========Accuracy Score===========')
print(f"Accuracy Score is : {accuracy_score(y_test,y_predrfc)* 100:.2f}%\n")

print('======Classification Report=============')
print(classification_report(y_test,y_predrfc,digits=2),'\n')

print('=========Confusion Matrix============')
print(confusion_matrix(y_test,y_predrfc))
```

```
==========Outputs of Random Forest Classifier===========

==========Accuracy Score===========
Accuracy Score is : 97.01%

======Classification Report=============
              precision    recall  f1-score   support

           0       0.98      0.96      0.97     28522
           1       0.96      0.98      0.97     28817

    accuracy                           0.97     57339
   macro avg       0.97      0.97      0.97     57339
weighted avg       0.97      0.97      0.97     57339


=========Confusion Matrix============
[[27274  1248]
 [  467 28350]]
```

After training, the data in Random forest Classifier the accuracy score of the model is 97.01%.
The Classification Report consist of all the details related to accuracy score, precision, recall and f1- score of Random forest Classifier model.
It also consist of Confusion matrix values i.e True Positive, False Positive, True Negative and False Negative values of Random forest Classifier model.

- ## Key Metrics for success in solving problem under consideration

  There are multiple points which impact the final outcome for the model.

  1. Converting the text comments into vectors using multiple methods.
  2. Combining multiple targets into single label and then build the NLP model.
  3. Performing the data pre-processing on the both train and test data separately.
  4. Checking the data imbalanced issue and solve it using upsampling SMOTE method.

- ## Visualizations

  Visualization is basically finding some outcomes after visualizing the data in form of some graph or some plots. Different visualizing methods has been use the to do the analysis on the data. In this model multiple plots has been used to do analysis on the provided data.

  1. Bar Plot – To check the relation between feature and target.
  2. Heat Map – To check the null count and multicolliniarity issue between the features.
  3. Word Cloud – To check the type of word used in particular comments.

- ## Interpretation of the Results

  From visualization, pre-processing and modelling the result should indicates label can be either 0 or 1, where 0 denotes a NO while 1 denotes a YES. There are various comments which have multiple labels and after pre-processing merged to one.

CONCLUSION

- Key Findings and Conclusions of the Study

  Main finding in the models are using multiple NLP methods to covert the word features to the vectors and using these vectors features building the NLP model.

- Learning Outcomes of the Study in respect of Data Science

  There are only one feature column which is consist of the text data.so the main learning is to analyse this data and cleaning the data using multiple method like stopwords, lowering the text, lemmanizing. Using TF-IDF method to convert cleaned data into vectors and using this vectors building the model. From the visualization the data imbalanced issue has been solved form the target columns using over sampling method called as SMOTE.