FLIP ROBO

MALIGNANT COMMENT CLASSIFIER PROJECT

Submitted by:

ANITA THAPA

ACKNOWLEDGMENT

I would like to express my special thanks of gratitude to my SME Swati Ma’am for giving me an opportunity to work on the Malignant Comment Classifier Project. During my research on this project internet sites like towardsdatascience, medium, Machine Learning Mastery, Analytics Vidya, American Statistical Association, scikit learn and Stack Overflow helped me a lot in the project.

INTRODUCTION

- ## Business Problem Framing

  Social media has given a lot of things to people which beyond imagination. In this era of technology, it has become the hub of information. The numbers of contents on social media are vast and rich and everything has found a place on social media that may be anything. It has given wings to its users to fly high and express their feelings. It has become a boon for the mankind but we all know that if there is good there must be some bad. Likewise, social media has also got the dark side.

  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

  In the past few years its seen that the cases related to social media hatred have increased exponentially. The social media is turning into a dark venomous pit for people now a days. Online hate is the result of difference in opinion, race, religion, occupation, nationality etc. The result of such activities can be dangerous. It gives mental trauma to the victims making their lives miserable. People who are not well aware of mental health online hate or cyber bullying become life threatening for them. These kinds of activities must be checked for a better future.

- ## Review of Literature

  The dataset contains 1 feature column ‘comment_text’ and 6 target columns which are ‘malignant’, ’highly_malignant’, ’rude’, ’loathe’, ’threat’ and ‘abuse’.

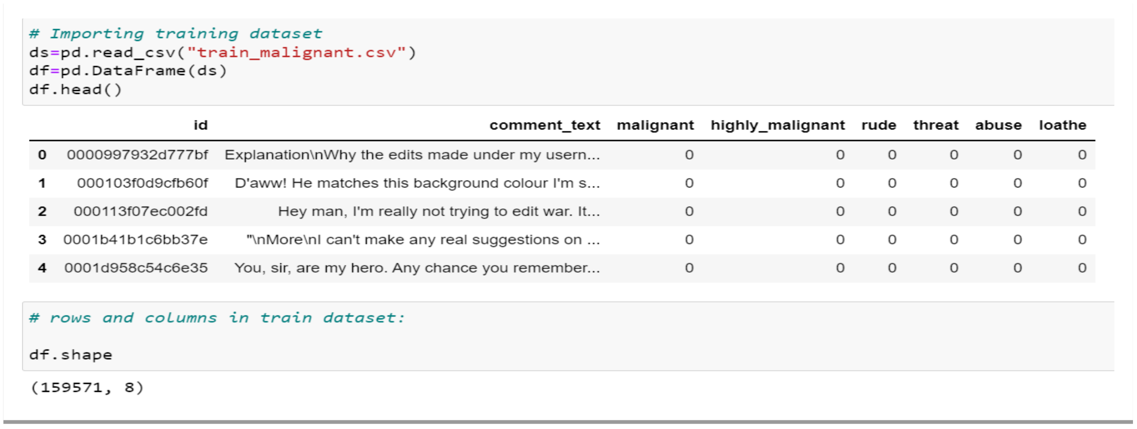
- Motivation for the Problem Undertaken

  The main motivation was to classify the news in order to bring awareness and reduce unwanted chaos and make a good model which will help us to know such kind of miscreants.

Analytical Problem Framing

- ## Mathematical/ Analytical Modeling of the Problem

The data was present in excel format so by using pandas the excel file was read. Then the dataset details were analysed using. shape () to find the number of rows and columns, using dtypes () we found the dataset datatype like object and numeric datatype. Dataset has 159571 Rows and 8 Columns; 6 columns are numeric and 1 column is object type and the Target columns are ‘malignant’, ’highly_malignant’, ’rude’, ’loathe’, ’threat’ and ‘abuse’. Dataset is a Classification model.



```
# Importing training dataset
ds=pd.read_csv("train_malignant.csv")
df=pd.DataFrame(ds)
df.head()
```

	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

```
# rows and columns in train dataset:

df.shape
```

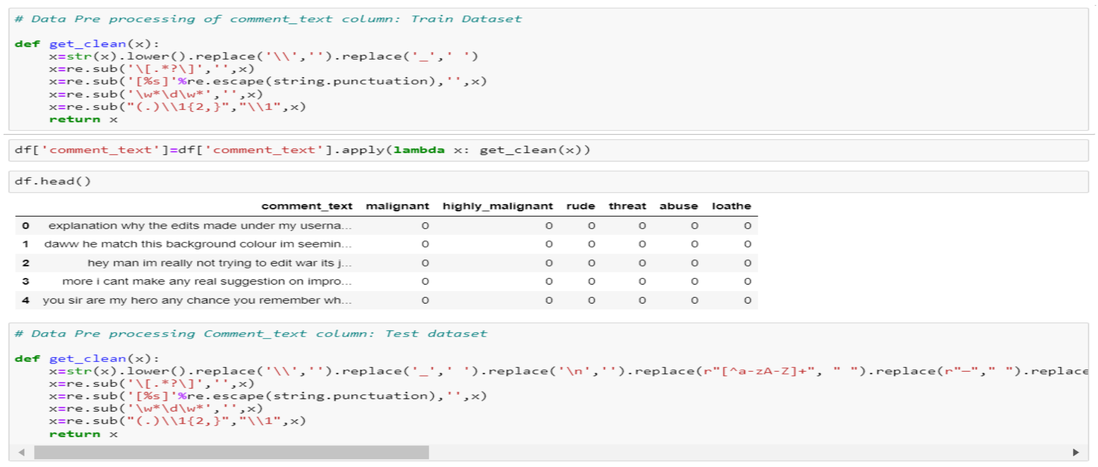
```
(159571, 8)
```

- ## Data Sources and their formats

Data was provided for project. Then data was imported in python using pandas. As per the information each column has count of 159571 rows and 8 columns.

- ## Data Preprocessing Done

ID columns dropped both in train and test dataset. Then the data in comment_text column had punctuation, uppercase and lowercase, emoji and numeric terms which were corrected using Natural Language Processing.



```
# Data Pre processing of comment_text column: Train Dataset

def get_clean(x):
    x=str(x).lower().replace('\\','').replace('_',' ')
    x=re.sub('\[.*?\]','',x)
    x=re.sub('[%s]'%re.escape(string.punctuation),'',x)
    x=re.sub('\w*\d\w*','',x)
    x=re.sub("(.)\\1{2,}","\\1",x)
    return x
```

```
df['comment_text']=df['comment_text'].apply(lambda x: get_clean(x))
```

```
df.head()
```

	comment_text	malignant	highly_malignant	rude	threat	abuse	loathe
0	explanation why the edits made under my userna...	0	0	0	0	0	0
1	daww he match this background colour im seemin...	0	0	0	0	0	0
2	hey man im really not trying to edit war its j...	0	0	0	0	0	0
3	more i cant make any real suggestion on impro...	0	0	0	0	0	0
4	you sir are my hero any chance you remember wh...	0	0	0	0	0	0

```
# Data Pre processing Comment_text column: Test dataset

def get_clean(x):
    x=str(x).lower().replace('\\','').replace('_',' ').replace('\n','').replace(r"[^a-zA-Z]+", " ").replace(r"—"," ").replace
    x=re.sub('\[.*?\]','',x)
    x=re.sub('[%s]'%re.escape(string.punctuation),'',x)
    x=re.sub('\w*\d\w*','',x)
    x=re.sub("(.)\\1{2,}","\\1",x)
    return x
```

- ## Data Inputs- Logic- Output Relationships

Input is the comment_text Column and output are ‘malignant’, ’highly_malignant’, ’rude’, ’loathe’, ’threat’ and ‘abuse’.The value of target columns is changed to 0 and 1 value using Tfidf where 0 means not a malignant comment and 1 represents malignant comment..

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

Here no such assumptions are considered.

- ## Hardware and Software Requirements and Tools Used

Hardware used:

1. LAPTOP: Lenovo yoga
2. OS: WIN 11
3. PROCESSOR: AMDA Radeon
4. RAM: 16GB

Libraries used were:

1. Numpy : It is a Numerical Python library used for numerical computation like arrays in Python.

2. Panda: It was used for reading and converting excel/csv file into dataframe.

3. matplotlib: this library was used for plotting the graph in the EDA.

4. Seaborn: This library is also used for visualization as it has helped to plot different types of plots like countplot in the notebook.

5. nltk, re: Used for Natural Language Processing of review column.

6. sklearn.preprocessing and power transform: It was used to remove the skewness from the dataset.

7. Tfidf Vectorizer : It was used for Review column.

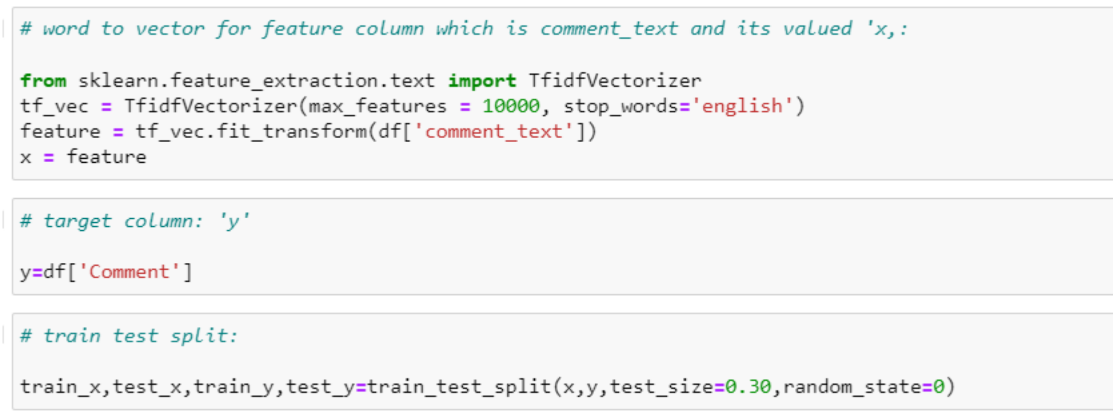
8. sklearn: This library was used to import train_test_split, metrics like accuracy score, classification report and also importing Algorithms likes Linear classification, Decision Tree classification, SV classification, Random Forest classification and K-Neighbors classification.

9. pickle: This library was used to save the final model.

Model/s Development and Evaluation

- ## Identification of possible problem-solving approaches (methods)

  The Tfidf Vectorizer is used in Model building. As the model is Classification so Logistic Regressor library along with accuracy, cross validation score, confusion matrix was imported. Then 4 Algorithms were used which were Random Forest Classification, XG-Boost Classifier, Ada-Boost Classifier and K-Neighbors Classification. Then after selecting best model, then the final model was deployed and the accuracy of model was 95% as it predicted values approximately equal to actual values.



```
# word to vector for feature column which is comment_text and its valued 'x,:

from sklearn.feature_extraction.text import TfidfVectorizer
tf_vec = TfidfVectorizer(max_features = 10000, stop_words='english')
feature = tf_vec.fit_transform(df['comment_text'])
x = feature
```

```
# target column: 'y'

y=df['Comment']
```

```
# train test split:

train_x,test_x,train_y,test_y=train_test_split(x,y,test_size=0.30,random_state=0)
```

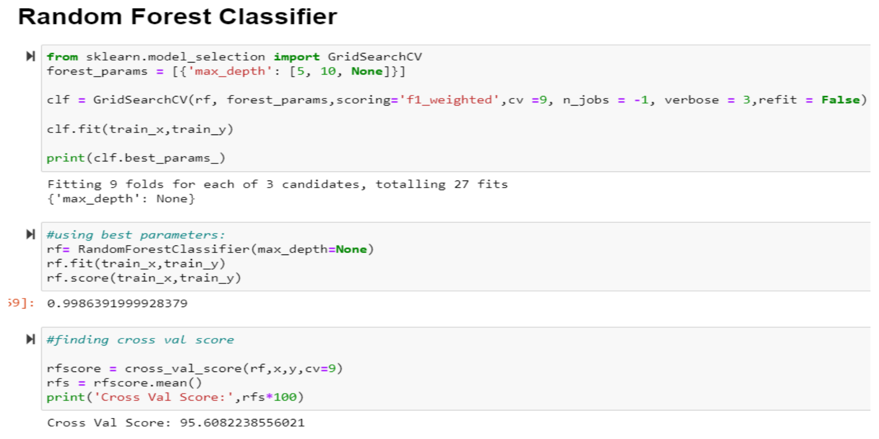
- ## Testing of Identified Approaches (Algorithms)

  Algorithms used for the training and testing are:

1. K-Neighbors Classifier: Accuracy and Cross validation score are 92.2% and 95.3%.
2. ADA Boost Classifier: Accuracy and Cross validation score are 92.3% and 92.2%.
3. Random Forest Classifier: Accuracy and Cross validation score are 95.6% and 95.5%.
4. XG-Boost Classifier: Accuracy and Cross validation score are 96.1% and 95.3%.

- ## Run and Evaluate selected models

Random Forest Classifier Algorithm is chosen the best model:



Random Forest Classifier

```
from sklearn.model_selection import GridSearchCV
forest_params = [{'max_depth': [5, 10, None]}]

clf = GridSearchCV(rf, forest_params,scoring='f1_weighted',cv =9, n_jobs = -1, verbose = 3,refit = False)

clf.fit(train_x,train_y)

print(clf.best_params_)
```

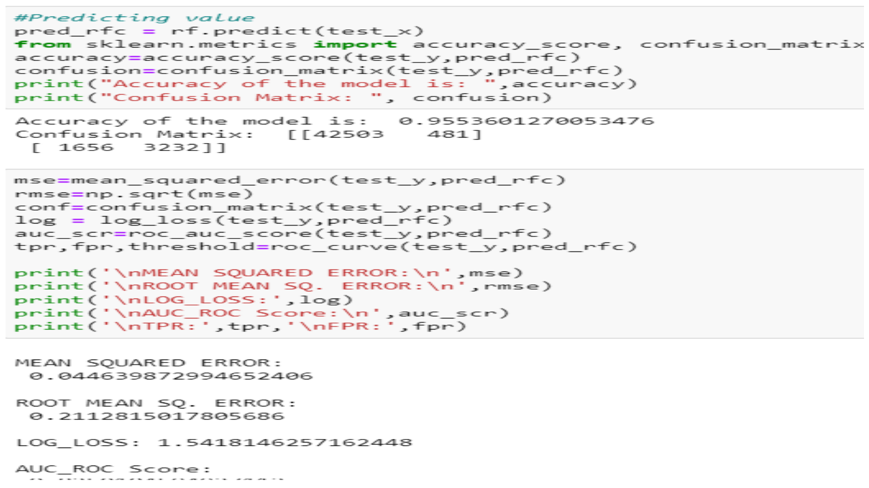
```
Fitting 9 folds for each of 3 candidates, totalling 27 fits
{'max_depth': None}
```

```
#using best parameters:
rf= RandomForestClassifier(max_depth=None)
rf.fit(train_x,train_y)
rf.score(train_x,train_y)
```

```
;9]: 0.9986391999928379
```

```
#finding cross val score

rfscore = cross_val_score(rf,x,y,cv=9)
rfs = rfscore.mean()
print('Cross Val Score:',rfs*100)
```

```
Cross Val Score: 95.6082238556021
```



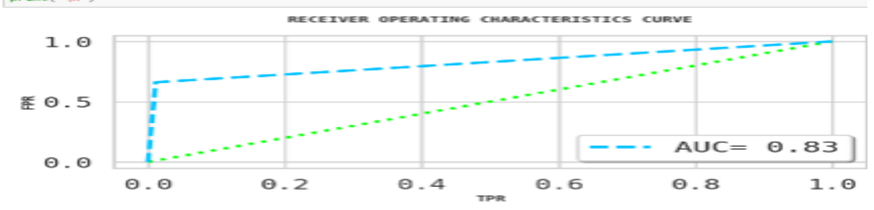
```
#Predicting value
pred_rfc = rf.predict(test_x)
from sklearn.metrics import accuracy_score, confusion_matrix
accuracy=accuracy_score(test_y,pred_rfc)
confusion=confusion_matrix(test_y,pred_rfc)
print("Accuracy of the model is: ",accuracy)
print("Confusion Matrix: ", confusion)
```

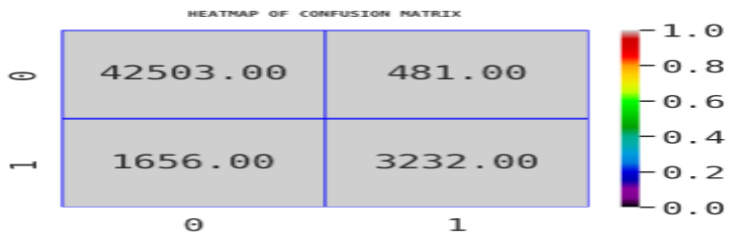
```
Accuracy of the model is:  0.9553601270053476
Confusion Matrix:   [[42503    481]
 [ 1656   3232]]
```

```
mse=mean_squared_error(test_y,pred_rfc)
rmse=np.sqrt(mse)
conf=confusion_matrix(test_y,pred_rfc)
log = log_loss(test_y,pred_rfc)
auc_scr=roc_auc_score(test_y,pred_rfc)
tpr,fpr,threshold=roc_curve(test_y,pred_rfc)

print('\nMEAN SQUARED ERROR:\n',mse)
print('\nROOT MEAN SQ. ERROR:\n',rmse)
print('\nLOG_LOSS:',log)
print('\nAUC_ROC Score:\n',auc_scr)
print('\nTPR:',tpr,'\nFPR:',fpr)
```

```
MEAN SQUARED ERROR:
 0.044639872994652406

ROOT MEAN SQ. ERROR:
 0.2112815017805686

LOG_LOSS: 1.5418146257162448

AUC_ROC Score:
```

```
print('\n')
```



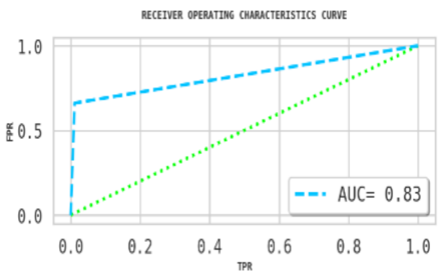


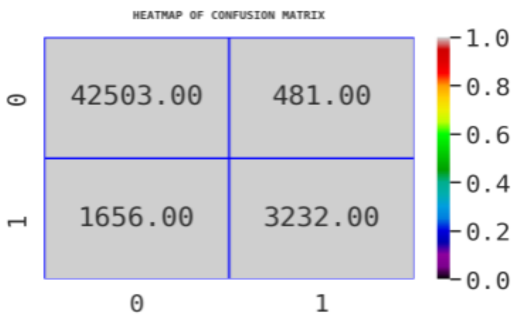
- Key Metrics for success in solving problem under consideration

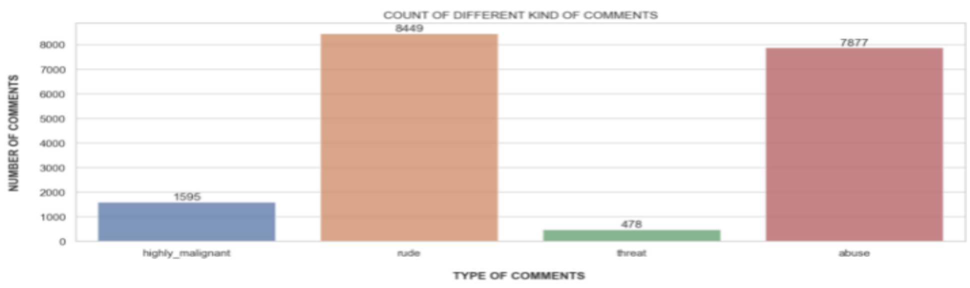
  Validation score, Accuracy, confusion matrix and the Classification report helped in building project.

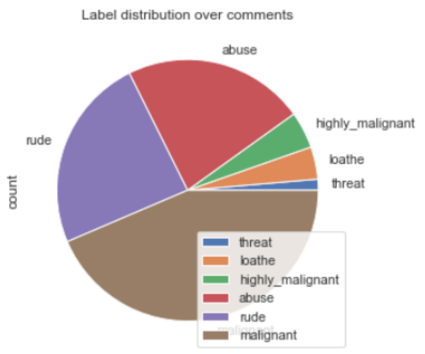
- Visualizations

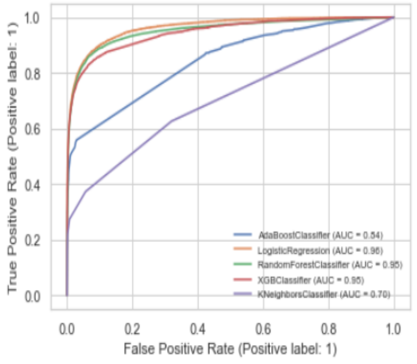
  Used only one plot which is catplot, distribution plot, roc curve, heatmap.











- Interpretation of the Results

  From the visualizations, pre-processing and modelling of the data it was interpreted that comment_text help in predicting malignant comment by the use of Natural Language Processing. As per the model the data can be predicted with 95% accuracy.

CONCLUSION

- Key Findings and Conclusions of the Study

  From the above analysis the below mentioned results were achieved which depicts the chances and conditions of a comment being a hateful comment or a normal comment. With the increasing popularity of social media, more and more people consume feeds from social media and due differences they spread hate comments to instead of love and harmony. It has strong negative impacts on individual users and broader society.

- Learning Outcomes of the Study in respect of Data Science

  It is possible to classify the comments content into the required categories of authentic and However, using this kind of project an awareness can be created to know what is fake and authentic.

- Limitations of this work and Scope for Future Work

  Every effort has been put on it for perfection but nothing is perfect and this project is of no exception. There are certain areas which can be enhanced. Comment detection is an emerging research area with few public datasets. So, a lot of works need to be done on this field.