MALIGNANT COMMENT CLASSIFIER

Submitted by:

KISHAN BAROCHIYA

ACKNOWLEDGMENT

Here, I am acknowledging that presented all data and process are as per my best knowledge. I am also thankful to FlipRobo for providing me dataset for Study purpose. In this interesting project I take a help from google for best result, apart from it I thankful to my mentor Swati Mahaseth for guiding me in project with good clearance.

INTRODUCTION

- Business Problem Framing

We all are in the era of social media and spent a good part of time on social media. We noticed that in many social media platform like YouTube, LinkedIn, Facebook, Twitter etc have comment section where people can share review for that particular content. It can be helpful for content creator to improve best from it but sometimes we marked that many peoples use bad words, abusing or cyberbullying. Which leads to bad effect on peoples and it can generate big issues like comment on religion etc. We cannot remove option of comment section as it have more benefits than its misuse.

- Conceptual Background of the Domain Problem

Background of data is clear, as we all are aware from social media and comment section in it. Where we can share review. There are many objective to write malignant comments. For example, to share negative view of product or content, to make down impression of content owner, to get better rating or rating down of competitor.

- Review of Literature

First of all we understand from where we can get this type data, which type of comments are possible in dataset, how it is beneficial to content owner, what is the benefit if we can control malignant comment.

Data source: YouTube, Twitter, E-commerce platform, Facebook etc.

Types of comments: Positive comments, which are helpful to create good image of content, improvable, comment which can useful to content owner to improve content or service or product. Business comment which are not useful, like if someone put house selling content than share loan service offer in comment section, Malignant comment which is very bad for quality purpose because it leads to people on wrong way. Negative comment, someone who is not satisfied with product is

share their review but sometime in more aggression, they started abusing or bad words etc.

- Motivation for the Problem Undertaken

My object to control that malignant comments to improve better service where we can give original content and comments to customer, on basis of it they can take decision to buy or study or learning from it or not. Here our aim is not to remove negative comment from section but main aim is to remove abusing, bad words contain comments. If we can control than people will get quality and good filtered comment, which save their time. It is also useful for social world as we can control word or comments, which can create issues towards communities.

Analytical Problem Framing

- Mathematical/ Analytical Modeling of the Problem

Here, we have scraped data from social platform where we have 159571 comments data in training dataset, in this data we have analysis of the comment like is comments is malignant, rude, threat, abuse or not etc. In testing dataset we have 1.53 lakh comments, where we have only comments and we need to build model where we can understand and analyse that comment and revert is that comment is malignant or not?

Mathematical view of data:

- Out from 8 variable, 6 variable are in int form and 2 are in objective form.
- There is no any null value present in dataset.
- Data is looking skewed as its mean is far away from mid-point.
- For malignant comment, it have highly co-relation with rude and abuse comment.

- Data Sources and their formats

We are getting data in csv format, in which which have divided dataset in training and testing dataset. In training dataset, comments is analysed like is it abusing, rude or not.

In given data 6 variable are in int format which is in format of 0 or 1. Where 0 denotes that comments is rude or not and 1 denotes comment is rude. Same for other variable. Other 2 columns are in objective form (ID and comments). We know that id is not helpful for comment analysing.

- Data Pre-processing Done

There is no null values in data; we will pre-process data as below:

1) Drop ID column, as it is not helpful in model building.
2) Create new column of length, which contain length of comment.
3) Convert or clear comments as there are some emoji's are there, short form are there, mail id, contact no etc. are their which are not useful in model building so we will clear it.
4) Create column, which contain length of column after cleaning of comment.
5) Loc malignant comment and find out frequent words, which are used in that malignant comment and consider that words as malignant words.
6) Convert text into vectors for model building.

Now our dataset is good for further process

- Data Inputs- Logic- Output Relationships

It is very important to find out relation between each variable to get higher accuracy in model. Here we can see that Malignant column have higher relation with rude and abuse column, highly malignant column have relation with rude and abuse so we can mark that if rude or abuse words used in comment than it is higher chances that comment is malignant. Threat have lesser relation with malignant comment means if comments have threat words than we cannot say that it have higher chances that comments is malignant.

	malignant	highly_malignant	rude	threat	abuse	loathe
malignant	1.000000	0.308619	0.676515	0.157058	0.647518	0.266009
highly_malignant	0.308619	1.000000	0.403014	0.123601	0.375807	0.201600
rude	0.676515	0.403014	1.000000	0.141179	0.741272	0.286867
threat	0.157058	0.123601	0.141179	1.000000	0.150022	0.115128
abuse	0.647518	0.375807	0.741272	0.150022	1.000000	0.337736
loathe	0.266009	0.201600	0.286867	0.115128	0.337736	1.000000

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

1) Id column is not useful for model building, as we can understand that id is only for give unique id to that comment.
2) From given training set, we scraped bad words from malignant comments and consider them as malignant or bad words.

- Hardware and Software Requirements and Tools Used

First, we used Python software to run all programme. In python, we used many libraries as below.

Numpy: used for basic statistical review of data and for mathematical function.

Pandas: to load dataset and perform basic operations on dataset.

Seaborn and Matplotlib: Understand and visualisation of dataset.

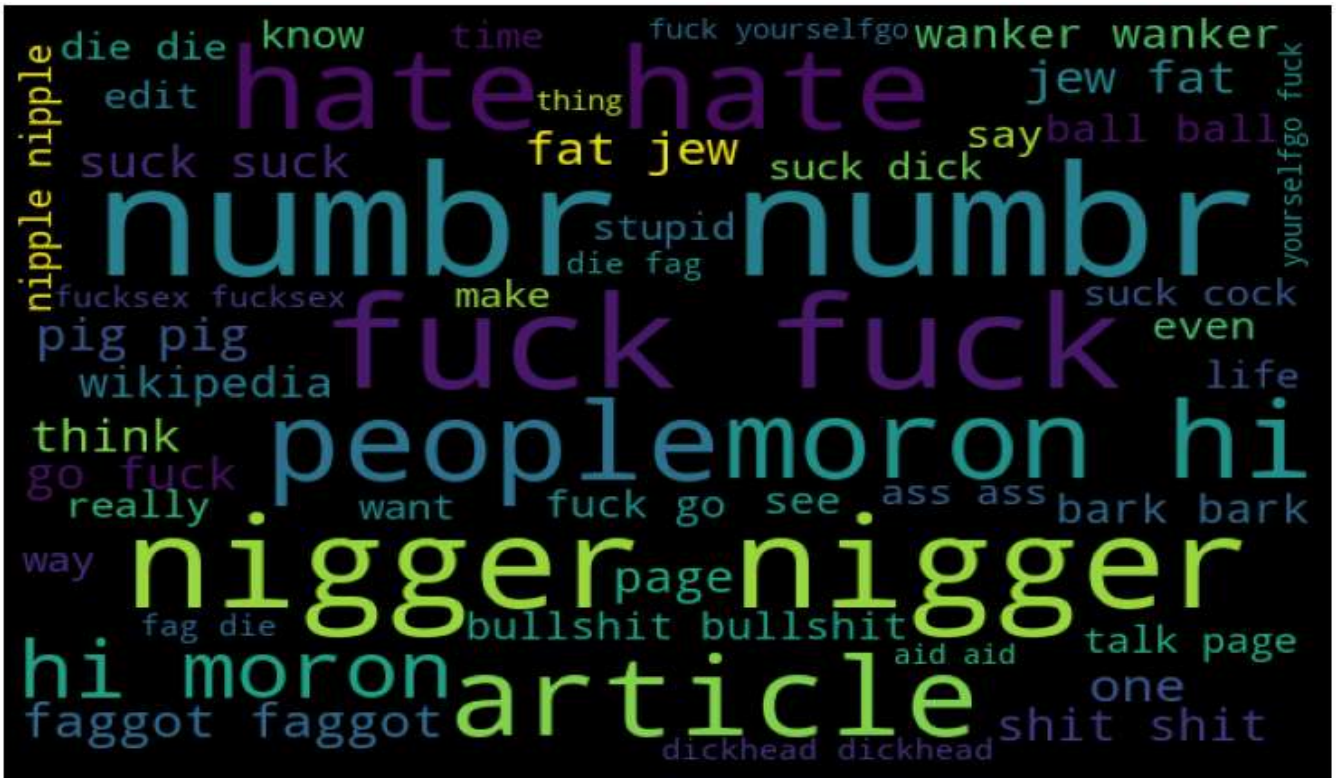
For Model building part:

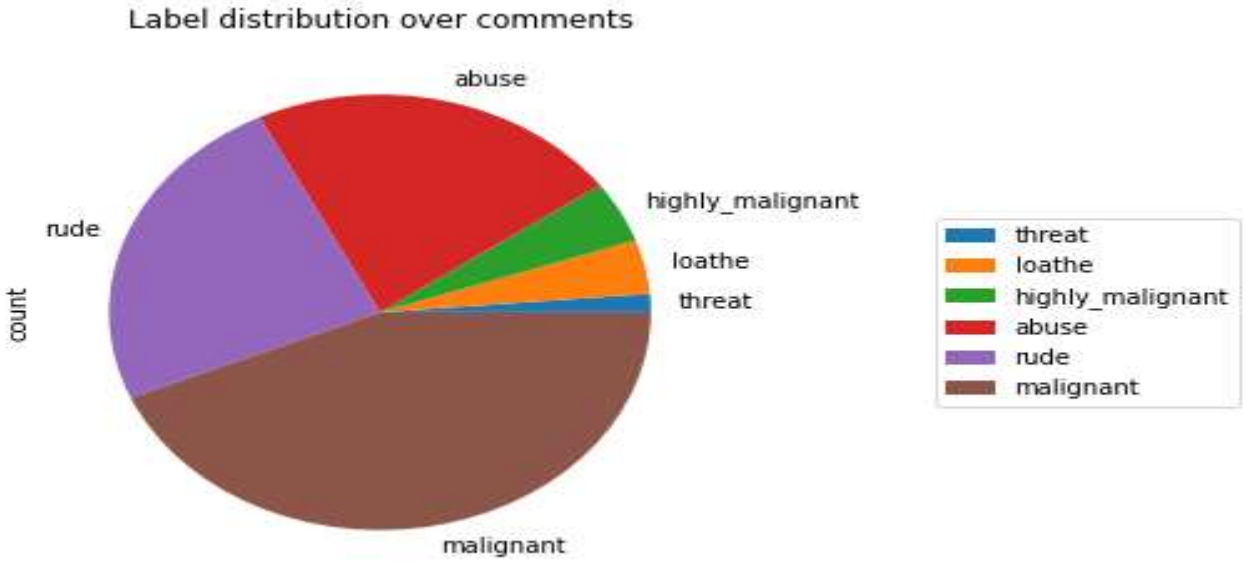
- train_test_split : divide dataset in training and testing.
- Models used for prediction:
  - LogisticRegression
  - DecisionTreeClassifier
  - KNeighborsClassifier
  - RandomForestClassifier
  - AdaBoostClassifier
  - GradientBoostingClassifier
  - GaussianNB
  - SVC
- accuracy_score,classification_report,confusion_matrix,f1_score
- accuracy_score,confusion_matrix,classification_report,roc_curve,roc_auc_score,auc
- import pickle : To dump, treated model for future use

Model/s Development and Evaluation

- Identification of possible problem-solving approaches (methods)

  First, I understand the data by statistical view as we show above then I analyse the relation between each variable by EDA then remove outliers and clean the comment where we have some short keywords, emoji, repeated words etc. then we loc malignant comment and scrape frequent words used in malignant comment. By using this words we build model where we can input comment and based on past data we can get that comment is malignant or not.





- Testing of Identified Approaches (Algorithms)

  Models used for prediction:

  - LogisticRegression
  - DecisionTreeClassifier
  - KNeighborsClassifier
  - RandomForestClassifier

- AdaBoostClassifier
- GradientBoostingClassifier
- GaussianNB
- SVC

- Run and Evaluate selected models

LOGISTIC REGRESSION:

```
fun(lg)
```

```
Score 0.9595341050501796
Accuracy Score 95.53183489304813
Confusion Matrix
 [[42729   221]
 [ 1918  3004]]
Classification Report
              precision    recall  f1-score   support

           0       0.96      0.99      0.98     42950
           1       0.93      0.61      0.74      4922

    accuracy                           0.96     47872
   macro avg       0.94      0.80      0.86     47872
weighted avg       0.95      0.96      0.95     47872

F1 score 73.7449367865472
```

- In Logistic regression, we are getting 95.5% accuracy in testing and training dataset. Precision and recall is also good while F1 score is 73.7%

RANDOMFOREST CLASSIFIER

```
fun(rndf)
pred=rndf.predict(x_test)
```

```
Score 0.9988540631518635
Accuracy Score 95.47125668449198
Confusion Matrix
 [[42393   557]
 [ 1611  3311]]
Classification Report
              precision    recall  f1-score   support

           0       0.96      0.99      0.98     42950
           1       0.86      0.67      0.75      4922

    accuracy                           0.95     47872
   macro avg       0.91      0.83      0.86     47872
weighted avg       0.95      0.95      0.95     47872

F1 score 75.33560864618887
```

- In random forest classifier, we are getting 95.47% accuracy in testing dataset and we have bit higher f1 score compare to logistic regression.

DECISIONTREE CLASSIFIER

```
fun(dtc)
```

```
Score 0.9988898736783678
Accuracy Score 93.97977941176471
Confusion Matrix
 [[41603  1347]
 [ 1535  3387]]
Classification Report
              precision    recall  f1-score   support

           0       0.96      0.97      0.97     42950
           1       0.72      0.69      0.70      4922

    accuracy                           0.94     47872
   macro avg       0.84      0.83      0.83     47872
weighted avg       0.94      0.94      0.94     47872

F1 score 70.1532725766363
```

- In DTC, we are getting lower accuracy compare to LG and RNDF.

KNN

```
fun(knn)
```

```
Score 0.9296681259456218
Accuracy Score 91.84909759358288
Confusion Matrix
 [[42615   335]
 [ 3567  1355]]
Classification Report
              precision    recall  f1-score   support

           0       0.92      0.99      0.96     42950
           1       0.80      0.28      0.41      4922

    accuracy                           0.92     47872
   macro avg       0.86      0.63      0.68     47872
weighted avg       0.91      0.92      0.90     47872

F1 score 40.986085904416214
```

- In KNN, we are get lower accuracy and F1 score is very low.

ADABOOST CLASSIFIER

```
fun(ad)
pred=ad.predict(x_test)
```

```
Score 0.9463737365598618
Accuracy Score 94.54169451871658
Confusion Matrix
 [[42587   363]
 [ 2250  2672]]
Classification Report
              precision    recall  f1-score   support

           0       0.95      0.99      0.97     42950
           1       0.88      0.54      0.67      4922

    accuracy                           0.95     47872
   macro avg       0.92      0.77      0.82     47872
weighted avg       0.94      0.95      0.94     47872

F1 score 67.16099032298605
```

- Accuracy is near 94% and F1 score is 67 which are lower than others.

GRADIENTBOOST CLASSIFIER

```
fun(gd)
pred=gd.predict(x_test)
```

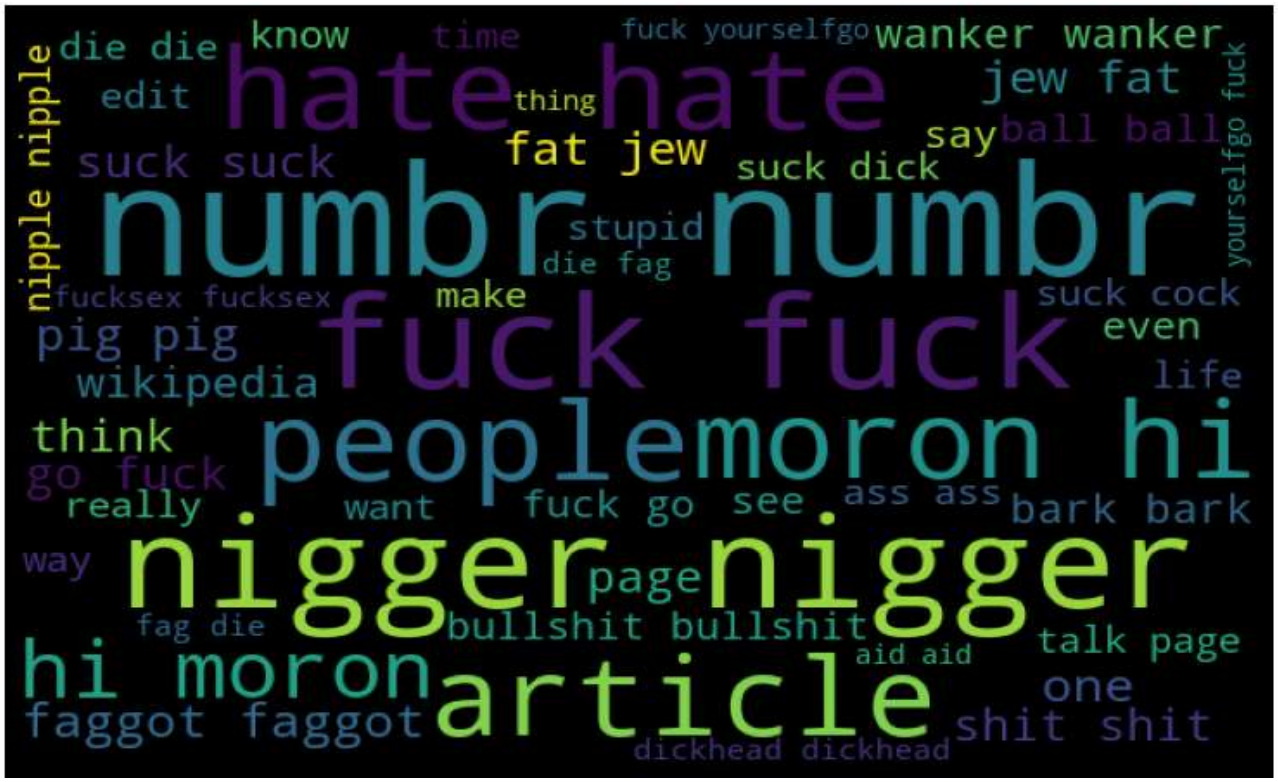
```
Score 0.9421391418007323
Accuracy Score 93.90040106951871
Confusion Matrix
 [[42812   138]
 [ 2782  2140]]
Classification Report
              precision    recall  f1-score   support

           0       0.94      1.00      0.97     42950
           1       0.94      0.43      0.59      4922

    accuracy                           0.94     47872
   macro avg       0.94      0.72      0.78     47872
weighted avg       0.94      0.94      0.93     47872

F1 score 59.44444444444444
```

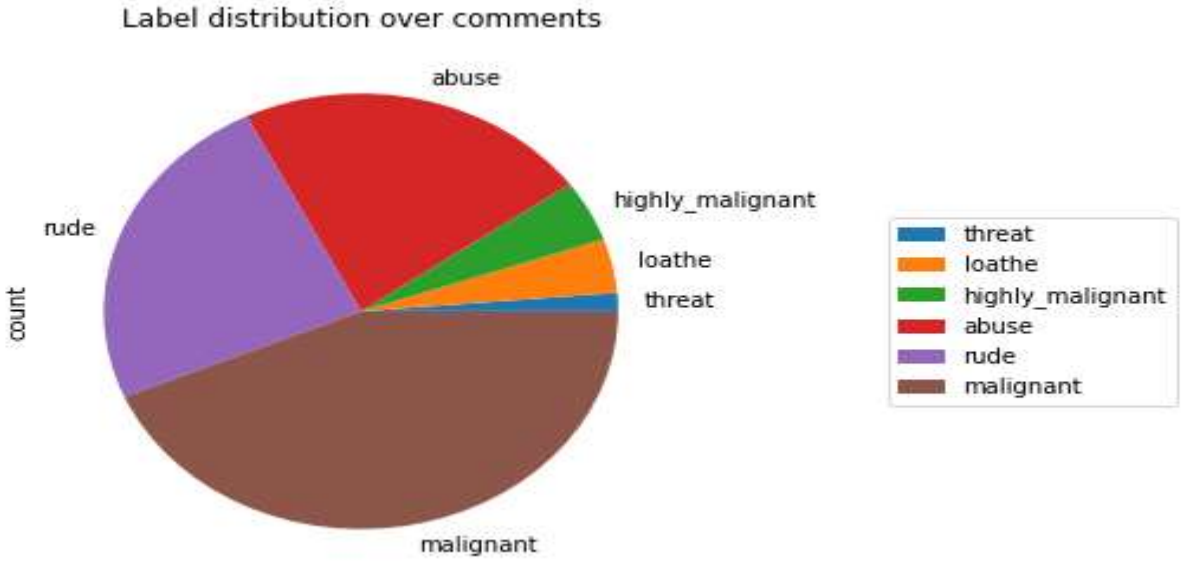
```
# RandomForestClassifier
RF = RandomForestClassifier()
RF.fit(x_train, y_train)
y_pred_train = RF.predict(x_train)
print('Training accuracy is {}'.format(accuracy_score(y_train, y_pred_train)))
y_pred_test = RF.predict(x_test)
print('Test accuracy is {}'.format(accuracy_score(y_test,y_pred_test)))
cvs=cross_val_score(RF, x, y, cv=10, scoring='accuracy').mean()
print('cross validation score :',cvs*100)
print(confusion_matrix(y_test,y_pred_test))
print(classification_report(y_test,y_pred_test))
```

```
Training accuracy is 0.9988540631518635
Test accuracy is 0.9548587901069518
cross validation score : 95.65835827444609
[[42396   554]
 [ 1607  3315]]
              precision    recall  f1-score   support

           0       0.96      0.99      0.98     42950
           1       0.86      0.67      0.75      4922

    accuracy                           0.95     47872
   macro avg       0.91      0.83      0.86     47872
weighted avg       0.95      0.95      0.95     47872
```

- Key Metrics for success in solving problem under consideration

  Here, we build a model, which work on given training dataset and according to it collect malignant words. Therefore, it might be possible that for other comments where extra new words use in malignant comments and accuracy may be decrease.

- Visualizations
- 

- 

- Interpretation of the Results

After working and testing dataset and different algorithm, we are on conclusion that Random forest classifier is the best-fit model for our dataset. In this algorithm, we are getting 95% accuracy which is very good accurate model.

CONCLUSION

Learning Outcomes of the Study in respect of Data Science

During this project, we can learn that how we can handle the malignant comment and it can be helpful and every social platform can be controllable over malignant comment. Data science is very useful in this case where we can build auto identifier malignant comment where if someone types malignant comment than automatically detect and remove it from section.

- Limitations of this work and Scope for Future Work

Accuracy based on given input data as we trained our model which that particular words which are used frequently in Malignant comments so it might be possible that accuracy will decrease.