MALIGNANT COMMENTS CLASSIFICATION

Submitted by:

MOHAMMED MINHAJ

ACKNOWLEDGMENT

Wikipedia, Youtube.com, google.com. The data were collected from the used car website cardheko.com. Referred in study materials of machine learning which I am studying.

INTRODUCTION

- ## Business Problem Framing

  Now people are using social media for daily use. Some people are addicted to that, they spent more times in it. People can express their opinion through comment for a subject or other purposes. Some of them were used to write the comments in 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. 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

  We can identify the comments using the ID is given. Through that we can categorize the comment whether is malignant or not and other categories.

- ## Review of Literature

  May be all the comments not to be malignant or other categories. We want to learn the machine which category of comments is that. For that purpose we want categorize the comments. We want to extract some common words to categorize the comment. Here in this project for analysing I used word cloud for easy understanding. So I can categorize the comments. The words used in comments can categorize the comment which is malignant or other categories.

- ## Motivation for the Problem Undertaken

  Social media have it’s own benefits. It can give us daily news and other up to date news or some tech news etcetera. But some people are using the social media platforms for bullying and giving the bad comments especially for women. They got bad comments and trolls for them. It affects badly for mental health of who suffer this type of comments or other activities. So in this project we were used machine learning for categorizing the comments. So it helps the person who are facing these problem for categorizing it. So these are main motivation to do this project.

Analytical Problem Framing

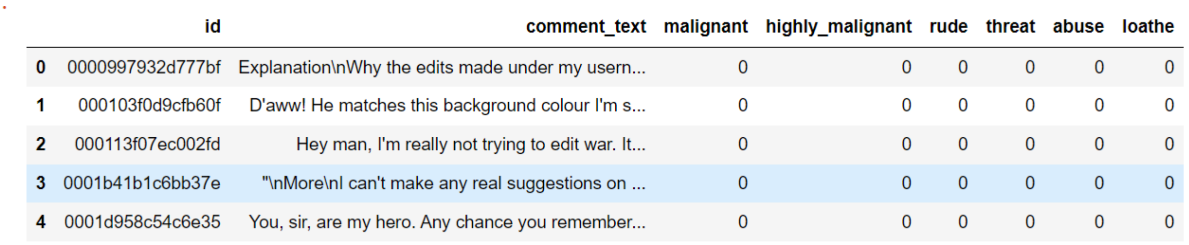
- Mathematical/ Analytical Modelling of the Problem

So as I mention our objective is to categorize the comments. Here by visualization we can understand the word that classifies the comments. That is, which word is used for each classification and machine can understand that which comment is it by analysing the words. So these are the major analytical step I done in this project.

- Data Sources and their formats

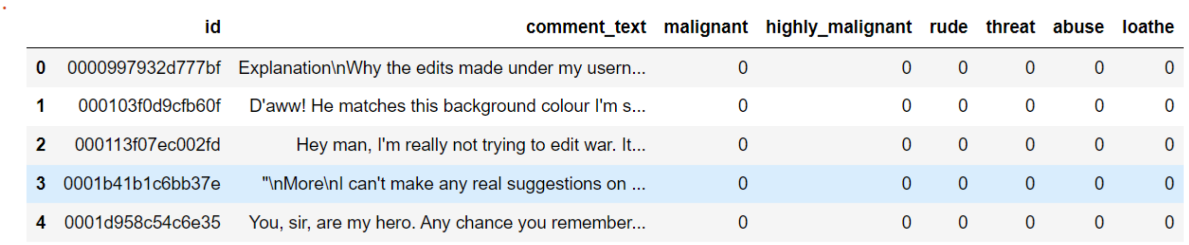
Data are given we need not to extract it from other sources. There are two separate datasets, one is for training and another for testing. In train dataset we have 159571 entries and 8 features. Whereas in test dataset we have 153164 and two features. Common features we have ID and Comments. In train dataset we have other features like categorizing the comments malignant, highly malignant, abuse, threat, loathe and rude. ID is used to find the comments that is which have unique ID for comments. Comments feature have the comment, which is our major input to find output. And other features have entries 0 and 1, where 0 is 'NO' and 1 represents 'YES'. So other features names itself is to denote whether the given comment categorize them with comment.

Train



	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

Test



	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

- Data Pre-processing Done

  As we have we have comment which is in text format so we want perform some NLP pre-processing in it such as remove punctuations, converting to lowercase etcetera. After removing all these I done lemmatization in this feature. After that I perform vectorization in this feature. Then we have ID which is unique for each comment here I used label encoding for encoding purpose. Then we have categorization of comments. For better understanding I extract the words from comments for categorizing the comments.

- Data Inputs- Logic- Output Relationships

  The major input for the classifying the comment is the feature comment. It classifies the comment whether it is malignant or not or other category or not. Machine can categorize it by the words. In pre-processing part we done the classification of category of comments by analysing the offensive words in each comment category. So the comment categorize the whether the comment is malignant or not and other category.

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

  The comment is categorized using the words used in it. That is if we have a comment we can categorize it using the words use in it whether the comment is malignant or not, highly malignant or not, loathe or not, abuse or not, rude or not, threat or not. Here threat may affects the life if that person face these comments.

- ## Hardware and Software Requirements and Tools Used

  As our dataset is very huge so we have to use high memory for it. Here for this project around 19 MB of memory is used. Here typical modules used for building the model. Pandas module for importing dataset and other analysing technique. And visualization modules for analysing the data visually the modules are seaborn and matplotlib. Warnings module used for ignoring the warning. NLTK(natural language tool kit) module is used for importing word net lemmatizer for performing lemmatization technique. For the visualization of the words I used word cloud module. For vectorization from sklearn module feature extraction is imported for tfidf vectorization. For encoding label encoder is imported from preprocessing module. And four models imported, from naïve bayes multinomial NB is imported. For importing decision tree, imported it from the module tree. For importing linear SVC, imported support vector machine. For importing random forest ensemble module is imported from sci-kit learn. For splitting train and test, for performing tuning and to find cross validation score I imported model selection module from sklearn. For perfroming the metrics like accuracy score, recall, precision I imported accuracy score, classification report from metrics module in sklearn. So these are the modules used for completing this project.

Model/s Development and Evaluation

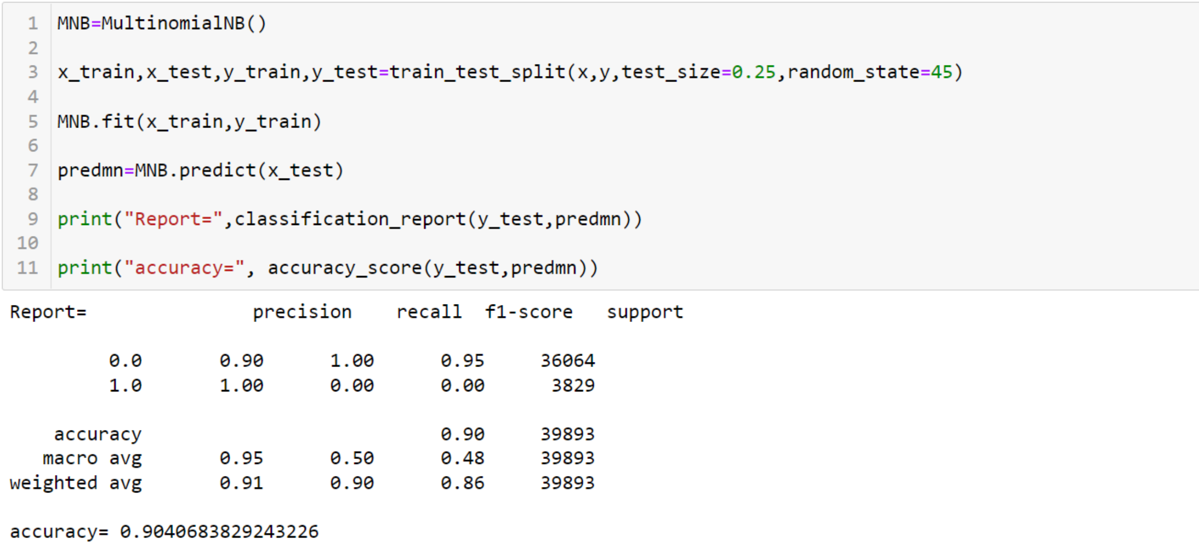
- Identification of possible problem-solving approaches (methods)

  As I mentioned we want to classify the comments using the words used in it. So we want to firstly analyse the words used in each category. There will be common words used to categorize the comments, but there will be some words which is unique in different category of comments. So we can easily classify it.

- Testing of Identified Approaches (Algorithms)
  - **Decision Tree**
  - **Random Forest**
  - **Multinomial NB**
  - **Linear SVC**

- Run and Evaluate selected models

<u>Multinomial NB</u>

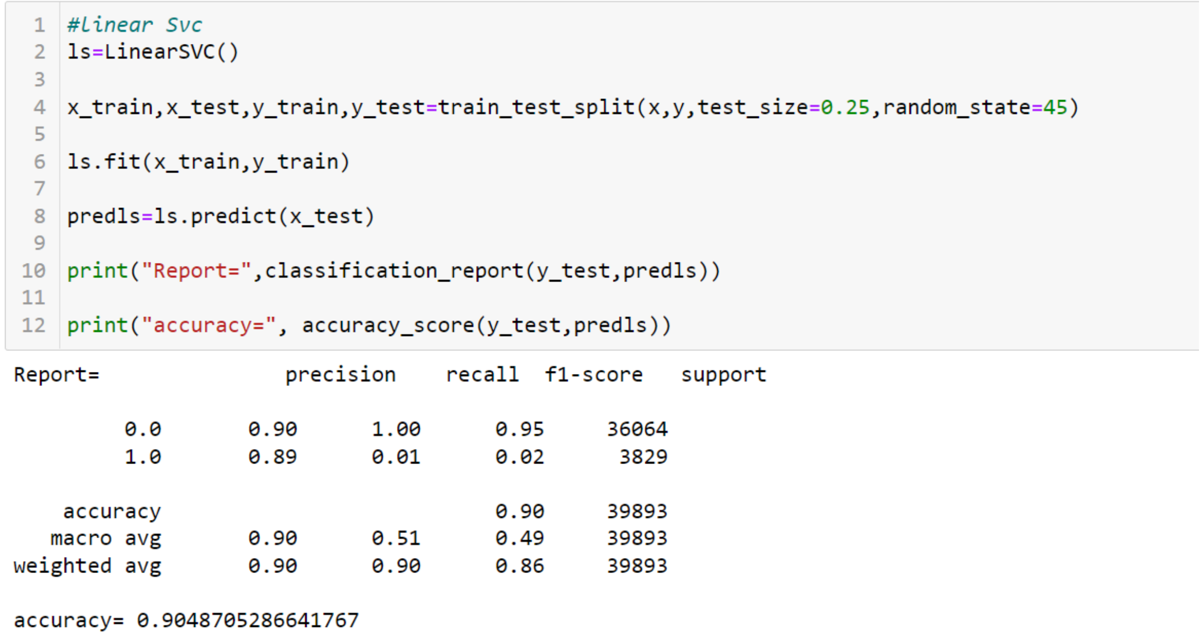


```
MNB=MultinomialNB()

x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.25,random_state=45)

MNB.fit(x_train,y_train)

predmn=MNB.predict(x_test)

print("Report=",classification_report(y_test,predmn))

print("accuracy=", accuracy_score(y_test,predmn))
```

```
Report=                precision    recall  f1-score   support

         0.0       0.90      1.00      0.95     36064
         1.0       1.00      0.00      0.00      3829

    accuracy                           0.90     39893
   macro avg       0.95      0.50      0.48     39893
weighted avg       0.91      0.90      0.86     39893

accuracy= 0.9040683829243226
```

Model split into train and test data, 25% of data were used for testing and 75% of the data were used for training. Accuracy score of Multinomial NB is 90% which is good. So we can say that multinomial NB perform good for this dataset. And we have precision 100% and recall is 95% for class 0. And for class 1 we have 100% of precision.

LinearSVC

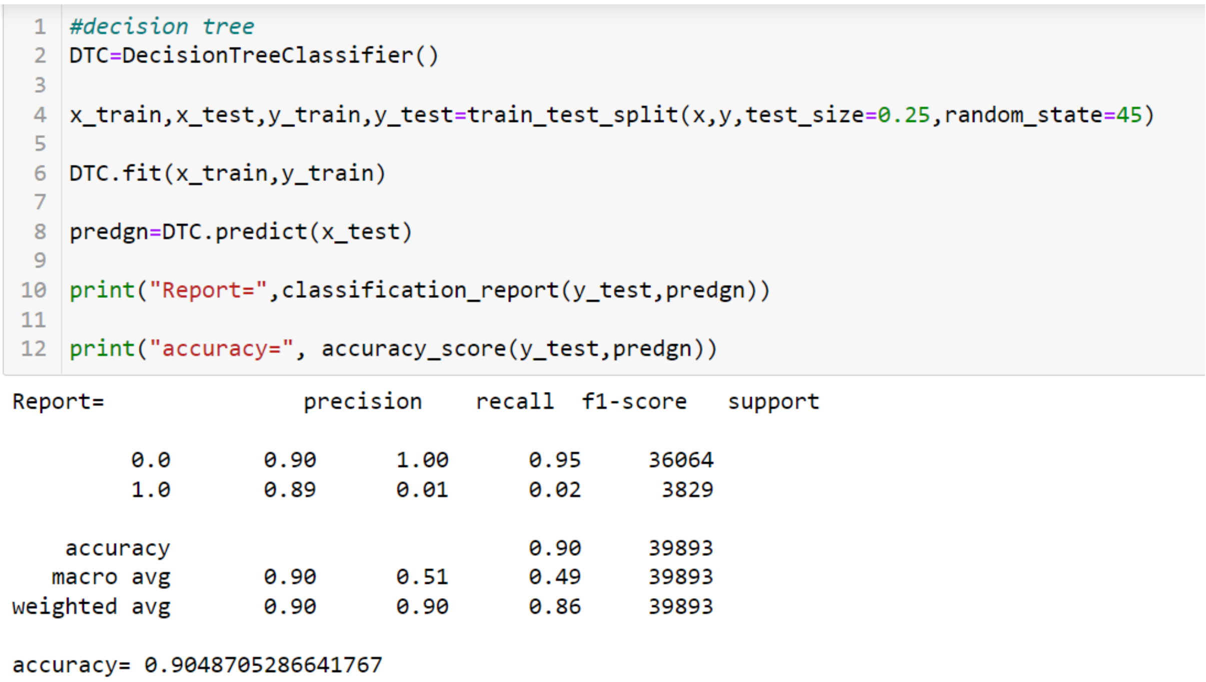


```
#Linear Svc
ls=LinearSVC()

x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.25,random_state=45)

ls.fit(x_train,y_train)

predls=ls.predict(x_test)

print("Report=",classification_report(y_test,predls))

print("accuracy=", accuracy_score(y_test,predls))
```

```
Report=               precision    recall  f1-score   support

         0.0       0.90      1.00      0.95     36064
         1.0       0.89      0.01      0.02      3829

    accuracy                           0.90     39893
   macro avg       0.90      0.51      0.49     39893
weighted avg       0.90      0.90      0.86     39893

accuracy= 0.9048705286641767
```

Model split into train and test data, 25% of data were used for testing and 75% of the data were used for training. Accuracy score of Linear SVC is 90% which is good. So we can say that Linear SVC perform good for this dataset. And we have precision 100% and recall is 95% for class 0. And for class 1 we have 89% of precision.

Decision Tree

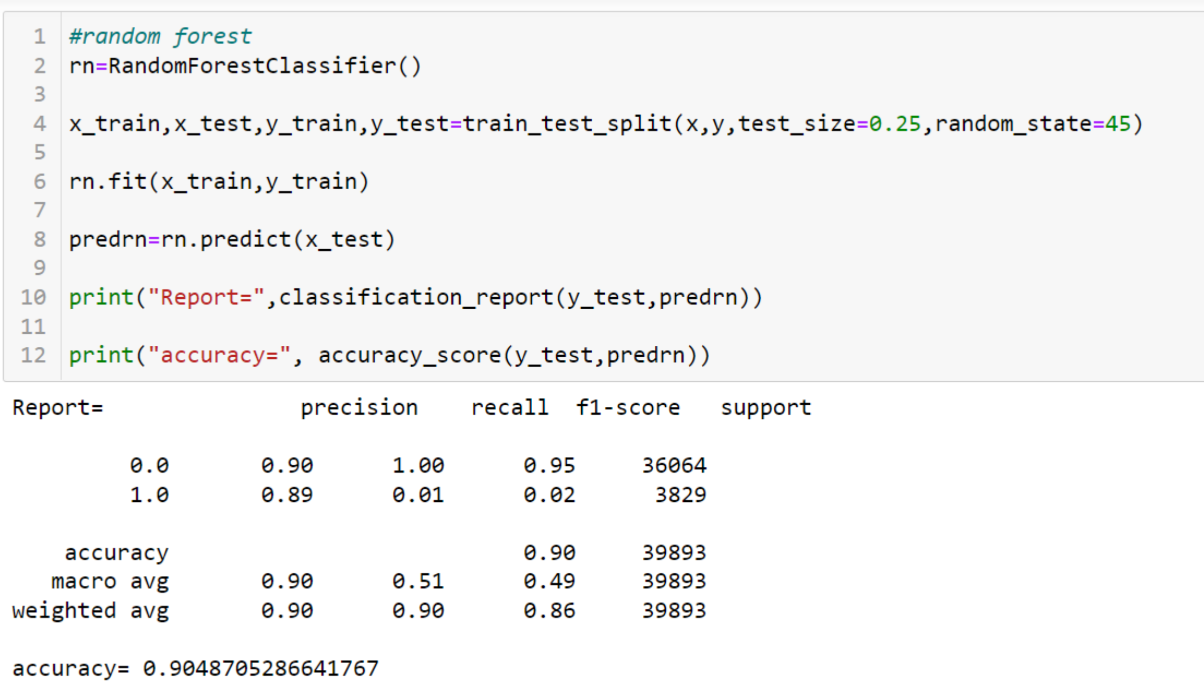


```python
#decision tree
DTC=DecisionTreeClassifier()

x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.25,random_state=45)

DTC.fit(x_train,y_train)

predgn=DTC.predict(x_test)

print("Report=",classification_report(y_test,predgn))

print("accuracy=", accuracy_score(y_test,predgn))
```

```
Report=               precision    recall  f1-score   support

         0.0       0.90      1.00      0.95     36064
         1.0       0.89      0.01      0.02      3829

    accuracy                           0.90     39893
   macro avg       0.90      0.51      0.49     39893
weighted avg       0.90      0.90      0.86     39893

accuracy= 0.9048705286641767
```

Model split into train and test data, 25% of data were used for testing and 75% of the data were used for training. Accuracy score of decision tree is 90% which is good. So we can say that decision tree perform good for this dataset. And we have precision 90% and recall is 100% for class 0. And for class 1 we have 89% of precision.

Random Forest



```
#random forest
rn=RandomForestClassifier()

x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.25,random_state=45)

rn.fit(x_train,y_train)

predrn=rn.predict(x_test)

print("Report=",classification_report(y_test,predrn))

print("accuracy=", accuracy_score(y_test,predrn))
```

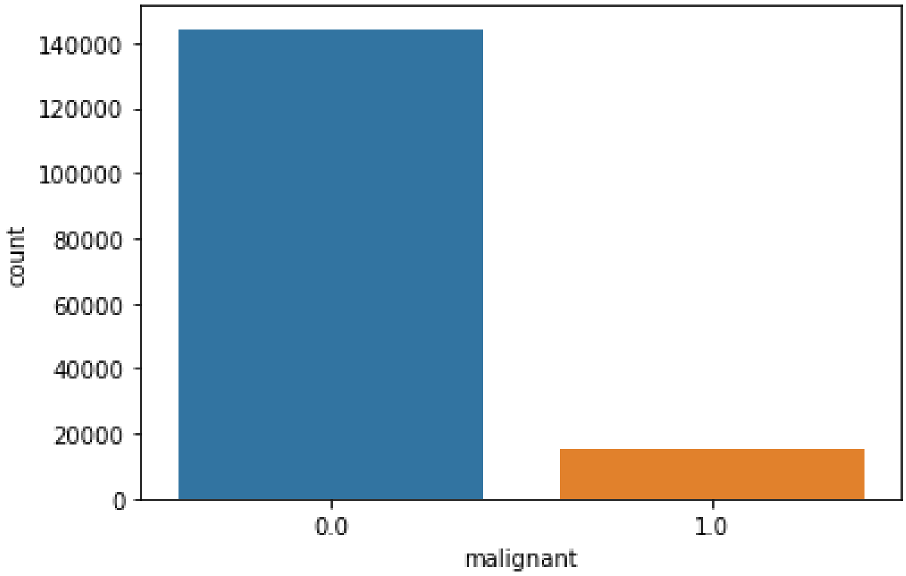
```
Report=               precision    recall  f1-score   support

         0.0       0.90      1.00      0.95     36064
         1.0       0.89      0.01      0.02      3829

    accuracy                           0.90     39893
   macro avg       0.90      0.51      0.49     39893
weighted avg       0.90      0.90      0.86     39893

accuracy= 0.9048705286641767
```

Model split into train and test data, 25% of data were used for testing and 75% of the data were used for training. Accuracy score of random forest is 90% which is good. So we can say that random forest also perform good for this dataset. And we have precision 90% and recall is 100% for class 0. And for class 1 we have 89% of precision.

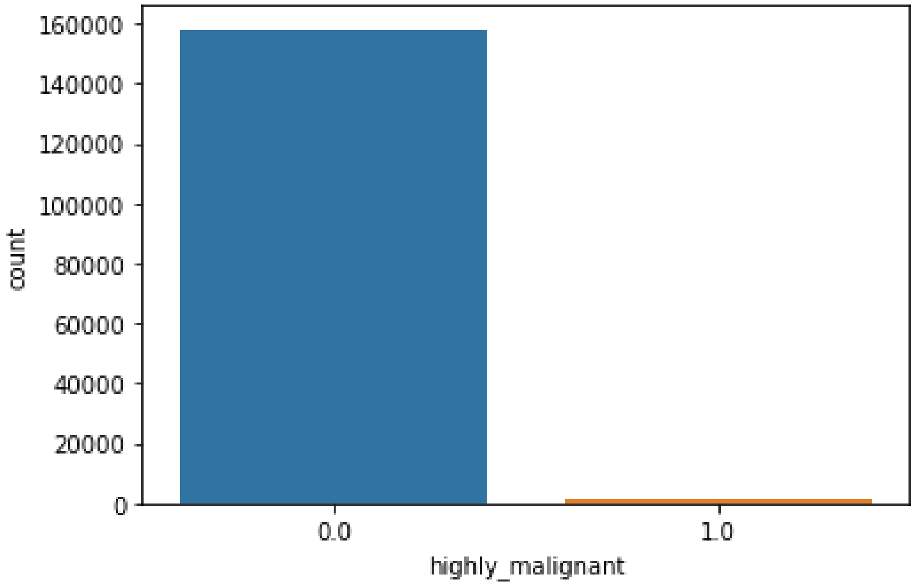
- Key Metrics for success in solving problem under consideration

The basic key metrics or commonly used key metrics for evaluating the classification model is accuracy score. A better accuracy score gives better performance of the model. Here in this project I used four different models. Which have 90% of accuracy. So in short I can say that four models have better performance. And other metrics I used to evaluate model is precision and recall to evaluate the performance of the model.

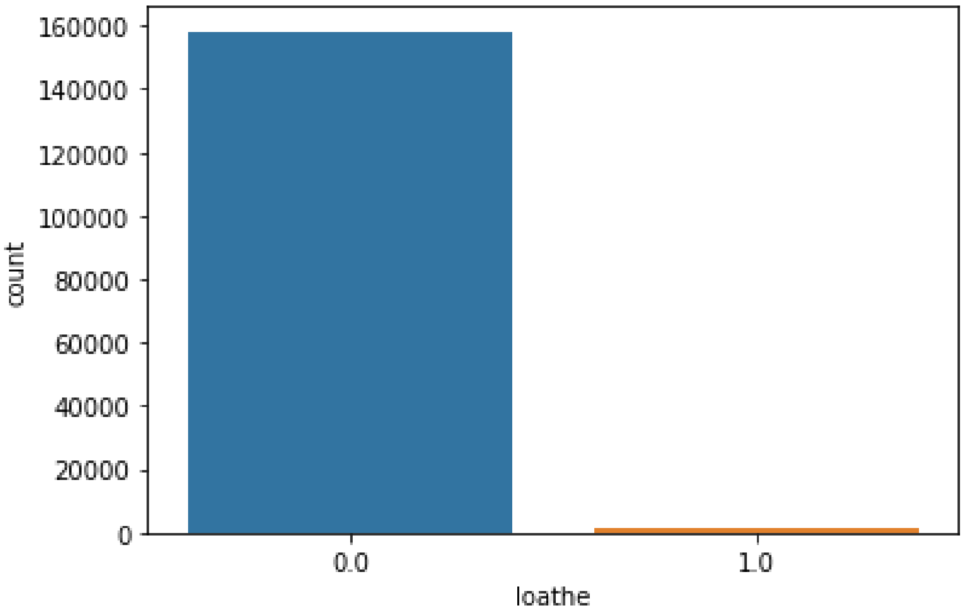
- Visualizations



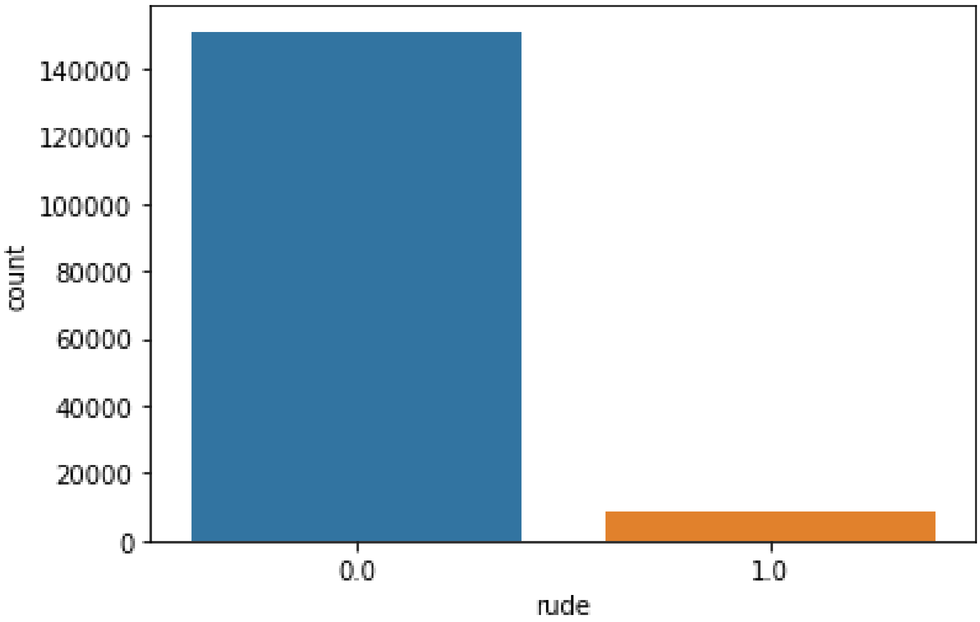
This shows the count of malignant comments, how many comments were malignant or not.



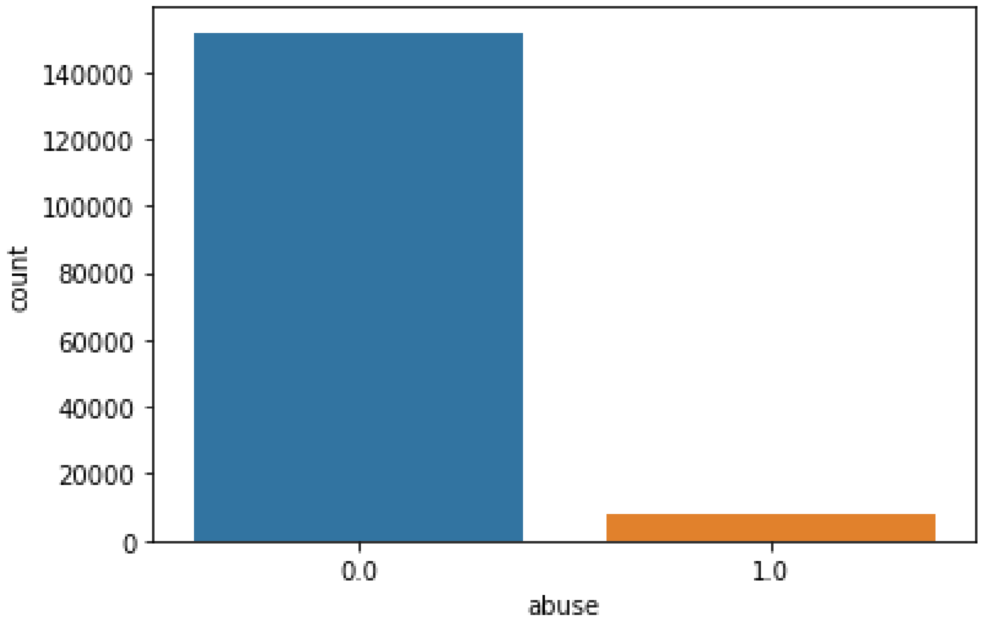
The count of comments which are highly malignant or not.



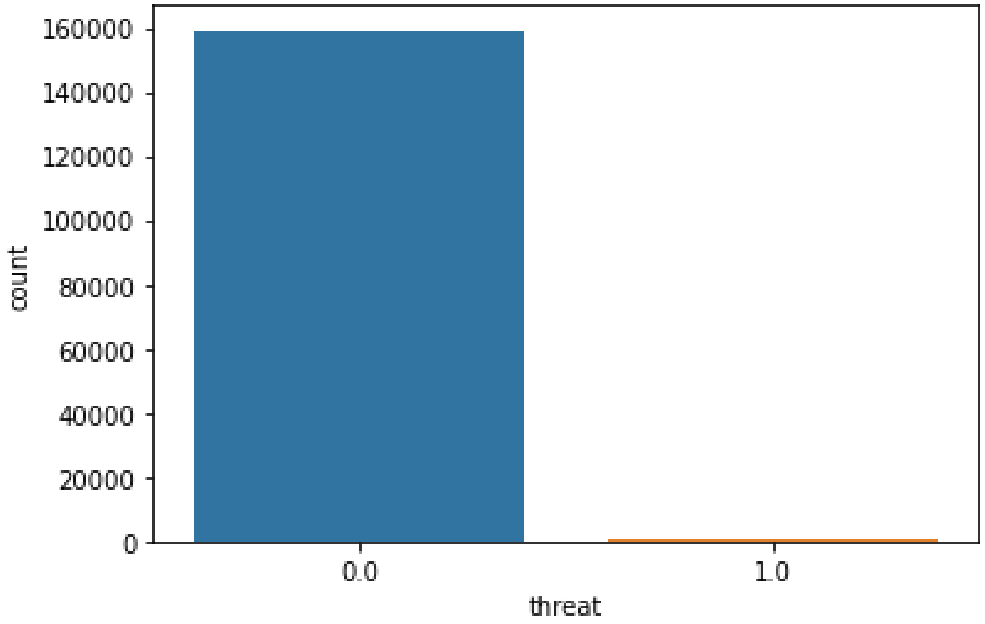
We can observe the count of comments which are loathe or not



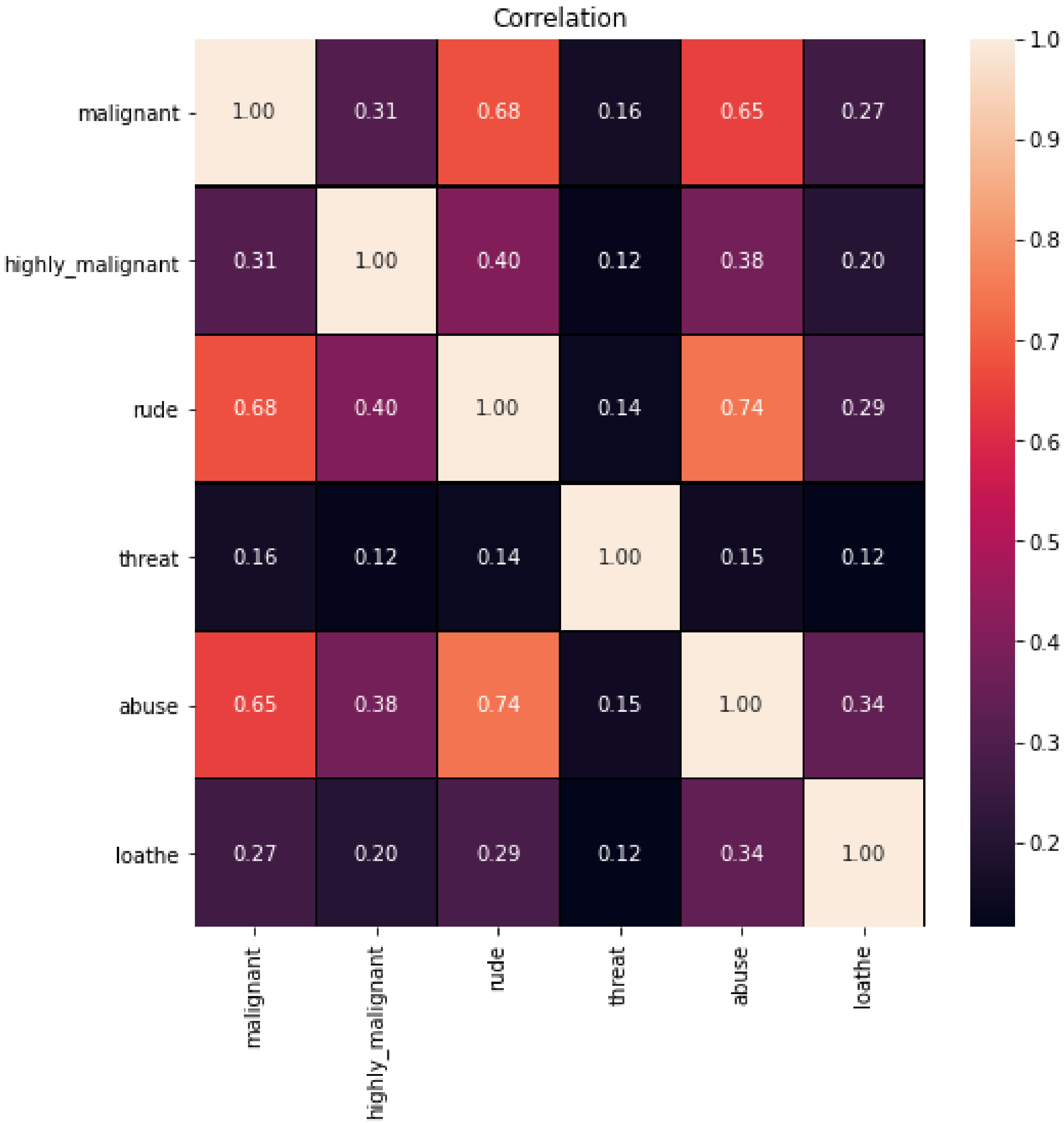
The count of comments which are rude or not.



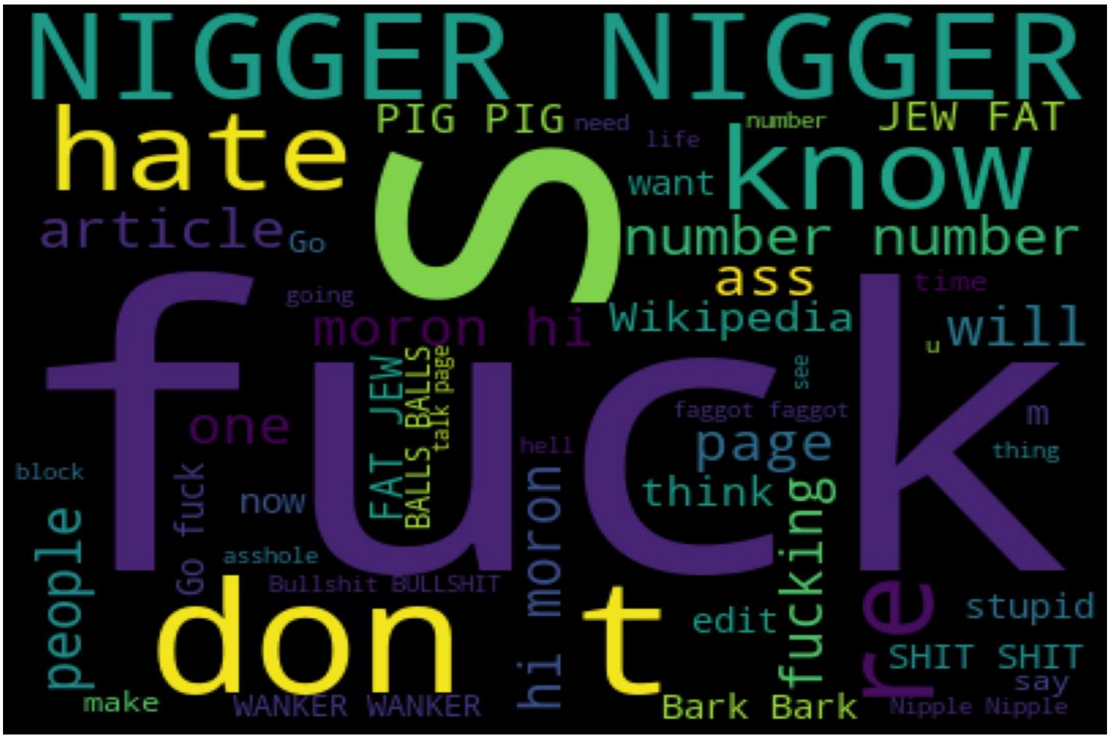
We can observe the comments which are abuse or not.



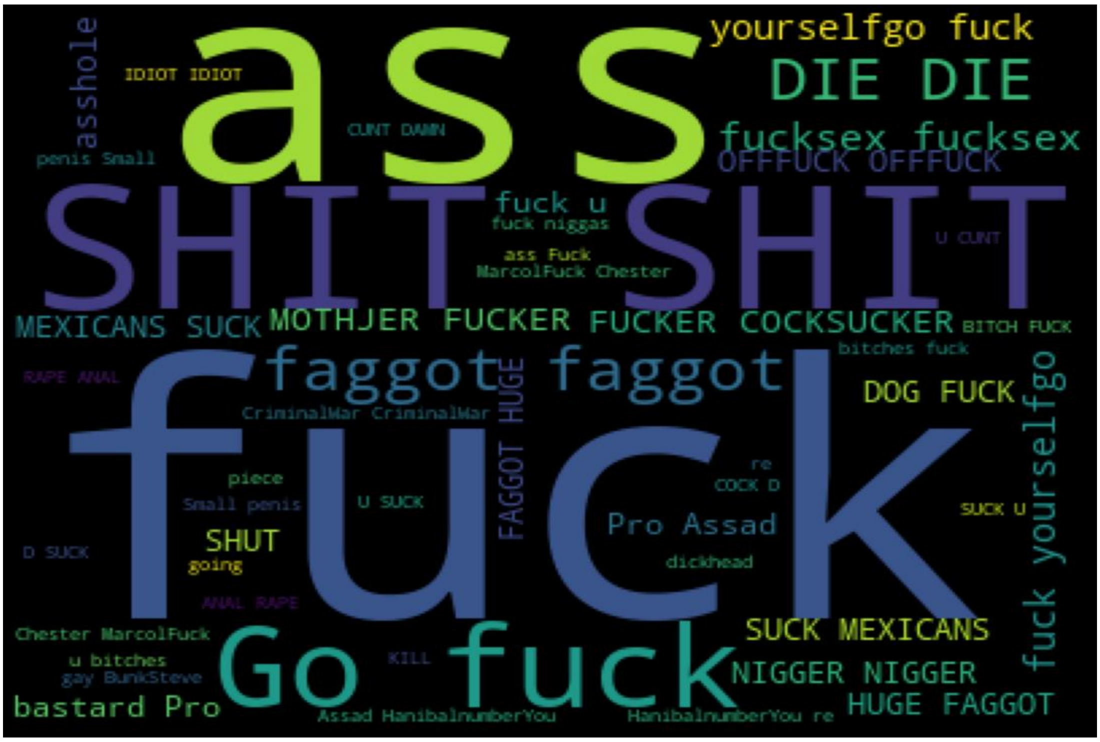
We can observe the comments which are threat or not.



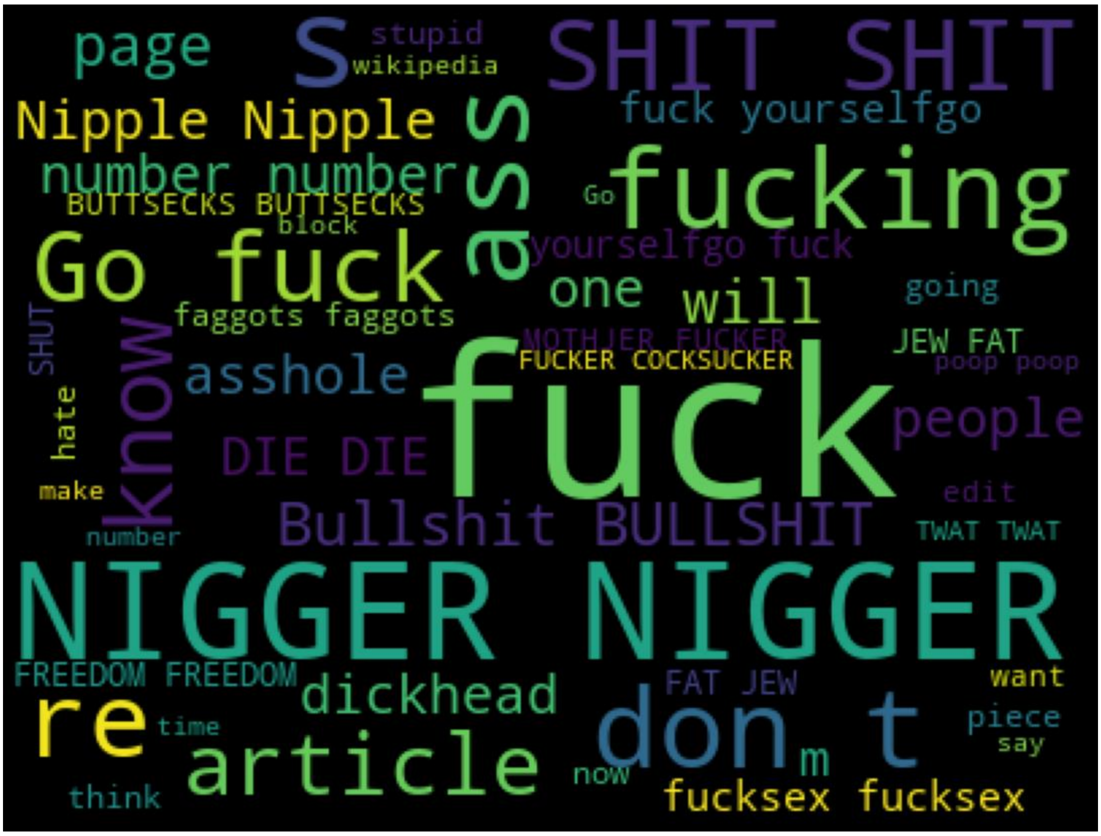
In this heat map we can observe the correlation between them. Rude and abuse were highly correlated, rude and malignant, abuse and malignant these all are highly correlated.



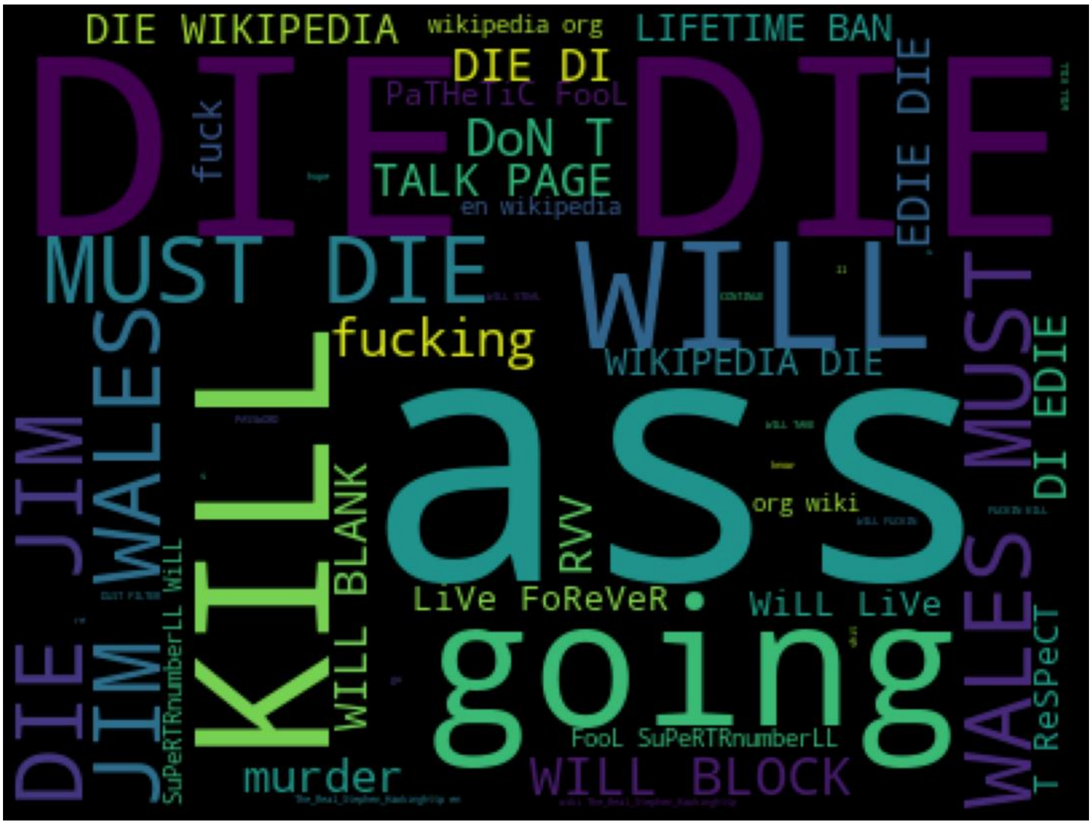
These are visualization of offensive words of malignant comments, these visualization is done using word cloud.



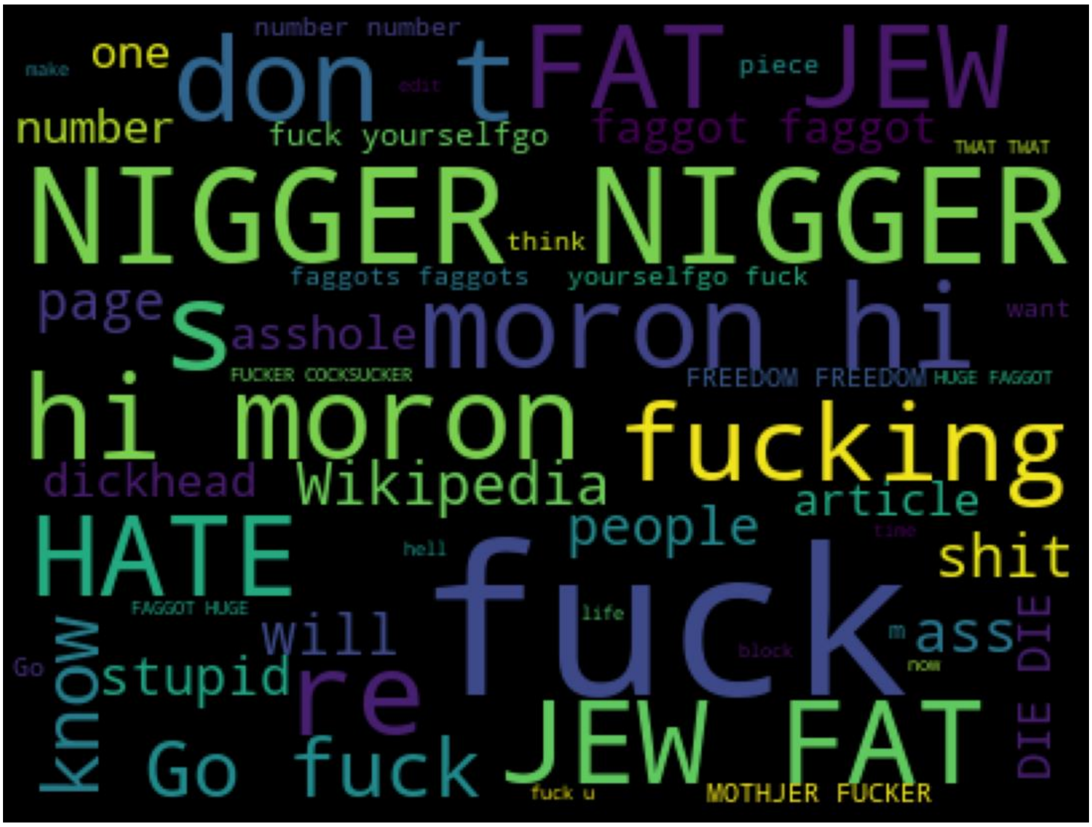
These are offensive words used in highly malignant.



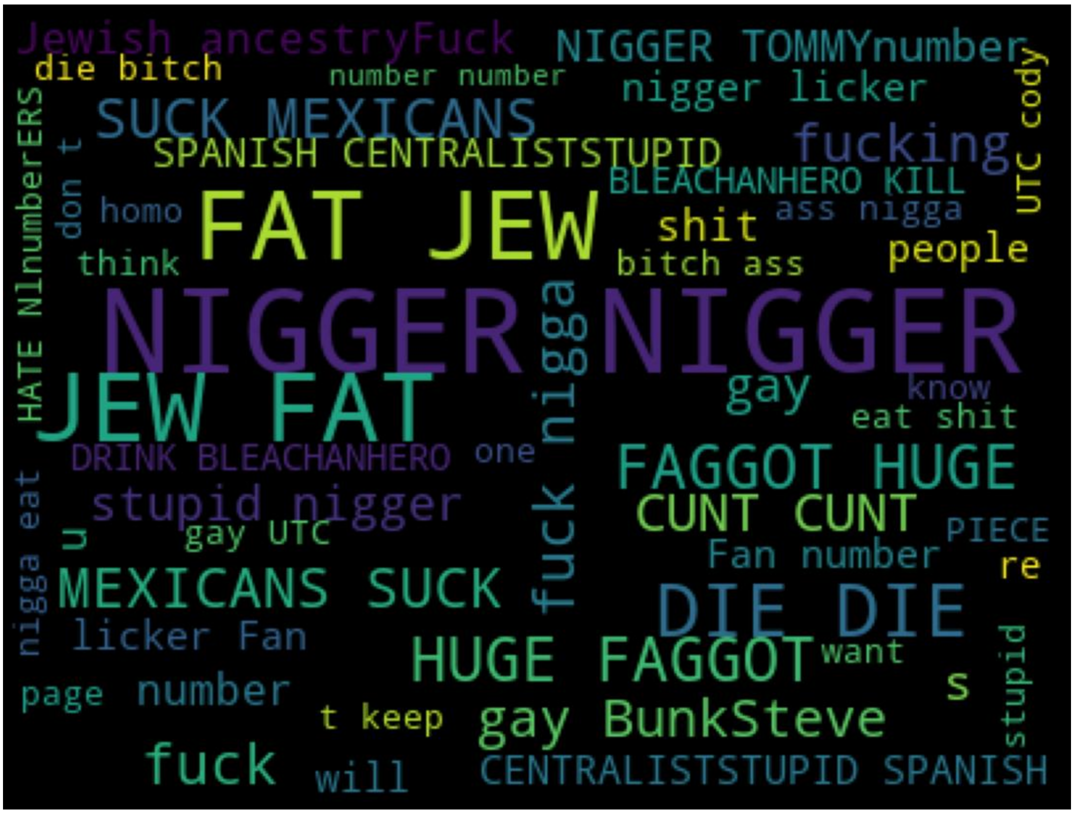
These are the offensive words in rude comments.



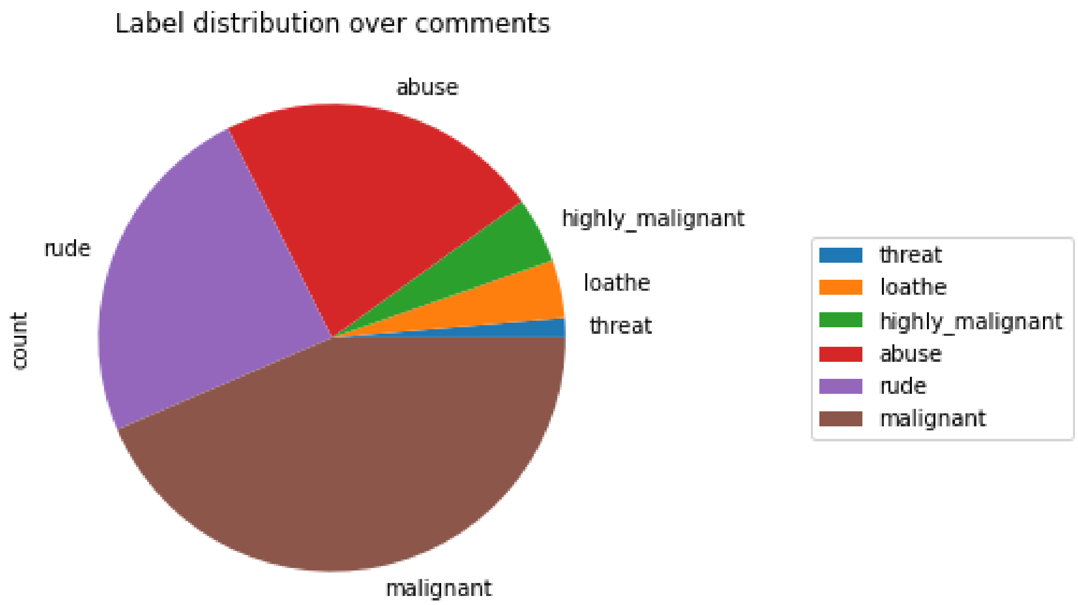
These are the offensive word used in threat type of comments.



These are the offensive words used in abuse comments.



These are offensive comments used in loathe comments.



This pie chart is used to analyse which type of comments are not offensive type. Here malignant type of comments are more number of comments which are not malignant.

• Interpretation of the Results

After a brief analysing of data, I have understand that the comments categorization is classified by the words. That is, the offensive words used in comments classification are common for every comments. In analysing the correlation I can understand that the comment which are rude, abuse, malignant are correlated. From that we can understand that these comments have some relation. May be the offensive words used in this type of comments are common in all of them.

CONCLUSION

- Key Findings and Conclusions of the Study

As I above mentioned the comments are classified among the words in the comment. That is how we categorize the comments. By analysing the most of the comments were rude, abuse and malignant. We have built machine learning to understand about the comments and categorize it.

- Learning Outcomes of the Study in respect of Data Science

Visualization one of the easy and better understanding of data. It gives the deep understanding of data. We have comment feature which is text data, I perform some NLP pre-processing in it that is, remove punctuations, convert to lowercase etcetera are performed in it. And we have ID which denote the uniqueness of comments, I perform the label encoding process in it. Then lemmatization done on comment, which is slow process it takes one day for the process lemmatization. And vectorization done on comment. For this project I used to perform four different models. Multinomial NB, linear SVC, Random forest, decision tree. Here Multinomial and linear svc gives the results fast. But decision tree and random forest gives the result slowly. As compared to random forest decision tree gives result fast. But we want to note that four algorithms gives better accuracy score and cross validation score. And I finalize the model by the fast performance in giving the result.

- Limitations of this work and Scope for Future Work

Every solution have a limitations. We want to recover that limitation by actual solution or make steps to get the actual solution. In this project we find the solution for understanding the comments categorization. But it may be not work for long time. So the time the solution did not work we want to find the better solution at that time.