Malignant Comment Classification

Submitted by:

Amol Wakchaure

Intern Data Science

Acknowledgment

I would like to express my special thanks of Flip Robo Pvt.Ltd as well as Data Trained Education who gave me the golden opportunity to do this wonderful project on the topic Malignant which also helped me in doing a lot of Research and i came to know about so many new things I am really thankful to them. Secondly i would also like to thanks below mentioned source was used by me for doing this project.

https://www.analyticsvidhya.com/blog/2017/01/ultimate-guide-to-understand-implement-natural-language-processing-codes-in-python/

INTRODUCTION

Business Problem

The proliferation of social media enables people to express their opinions widely online. However, at the same time, this has resulted in the emergence of conflict and hate, making online environments uninviting for users. Although researchers have found that hate is a problem across multiple platforms, there is a lack of models for online hate detection.

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.

.

Conceptual Background of the Domain Problem

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

Analytical Problem Framing

Mathematical/ Analytical Modeling of the Problem

This was a NLP Project and in this project we deals withthe textual data and for understanding the data we used some methods like removing punctuations ,numbers, stopwords and using the lemmatization process convert the complex words into their simpler forms. These processes helped in cleaning the unwanted words form the comments and we were left with only those words which were going to help in our model building. After cleaning the data we used TF-IDF Vectorizer technique to convert textual data into vector form. This technique works on the basis of the frequency of words present in the document. After training with train dataset we use this technique into test dataset.

Data Sources

This data was provided to me by FlipRobo Technologies into a csv file format. This file contains training and testing dataset. On training dataset we build a model and using this model we have to predict theoutcomes from testing dataset.

The data set contains the training set, which has approximately 1,59,000 samples and the test set which contains nearly 1,53,000 samples. All the data samples contain 8 fields which includes 'Id', 'Comments', 'Malignant', 'Highly malignant','Rude', 'Threat', 'Abuse' and 'Loathe'.

The label can be either 0 or 1, where 0 denotes a NO while 1 denotes a YES. Thereare various comments which have multiple labels. The first attribute is a unique IDassociated with each comment.

The data set includes:

-**Malignant**: It is the Label column, which includes values 0 and 1, denoting if thecomment is malignant or not.

-**Highly** Malignant: It denotes comments that are highly malignant and hurtful.

-**Rude**: It denotes comments that are very rude and offensive.

-**Threat**: It contains indication of the comments that are giving any threat tosomeone.

-**Abuse**: It is for comments that are abusive in nature.

-**Loathe**: It describes the comments which are hateful and loathing in nature.

-ID: It includes unique Ids associated with each comment text given.

-Comment text: This column contains the comments extracted from various socialmedia platforms.

First of all we upload the training dataset into train dataframe and check the data types present in it.

print('train shape is ',train.shape)

print('test shape is ',test.shape)

print('test info',test.info)

print('train info',train.info)

```
train shape is  (159571, 8)
test shape is  (153164, 2)
test info <bound method DataFrame.info of                       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.
...                  ...                                                ...
153159  fffcd0960ee309b5  . \n i totally agree, this stuff is nothing bu...
153160  fffd7a9a6eb32c16  == Throw from out field to home plate. == \n\n...
153161  fffda9e8d6fafa9e  " \n\n == Okinotorishima categories == \n\n I ...
153162  fffe8f1340a79fc2  " \n\n == ""One of the founding nations of the...
153163  ffffce3fb183ee80  " \n :::Stop already. Your bullshit is not wel...

[153164 rows x 2 columns]>
train info <bound method DataFrame.info of                       id
comment_text  \
0       0000997932d777bf  Explanation\nWhy the edits made under my usern...
1       000103f0d9cfb60f  D'aww! He matches this background colour I'm s...
2       000113f07ec002fd  Hey man, I'm really not trying to edit war. It...
3       0001b41b1c6bb37e  "\nMore\nI can't make any real suggestions on ...
4       0001d958c54c6e35  You, sir, are my hero. Any chance you remember...
...                  ...                                                ...
159566  ffe987279560d7ff  ":::::And for the second time of asking, when ...
```

So the data present in training data is both object and integer in nature. 2 columnsare object dtypes and 6 columns are integer in dtypes.

Data Preprocessing Done

Data preprocessing is the data mining techniquethat involves transforming raw data into an understandable data format. So in data preprocessing technique first step is import data and the libraries to be used in model building.

In [1]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import re


from nltk import pos_tag
from nltk.stem import WordNetLemmatizer
from gensim.parsing.preprocessing import STOPWORDS
from nltk.corpus import wordnet
import warnings
warnings.filterwarnings('ignore')

# Importing libraries for classification
from sklearn.model_selection import train_test_split, cross_val_score

from sklearn.linear_model import LogisticRegression
from sklearn.tree import DecisionTreeClassifier
from sklearn.svm import SVC
from sklearn.neighbors import KNeighborsClassifier
from sklearn.ensemble import RandomForestClassifier, AdaBoostClassifier, GradientBoostingClassifier
```

Now after loading the dataset we check the missing values

```
id                0
comment_text      0
malignant         0
highly_malignant  0
rude              0
threat            0
abuse             0
loathe            0
    dtype: int64
```

So there were no null or missing values in our dataset we move to next step of data cleaning –

In data cleaning we dop the ID column as it gives no information. After droppingit we created another feature "negative_cmnts" which shows the labelled data ofpositive and negative comments.Now in cleaning the textual data we created a function to remove unwanted space, punctuation , numbers, emails , phone numbers etc and converted upper case letters into lower case and append the result into a new column.

```
# Convert all messages to lower case
train['comment_text'] = train['comment_text'].str.lower()

# Replace email addresses with 'email'
train['comment_text'] = train['comment_text'].str.replace(r'^.+@[^\.].*\.[a-z]{2,}$',
                    'emailaddress')

# Replace URLs with 'webaddress'
train['comment_text'] = train['comment_text'].str.replace(r'^http\://[a-zA-Z0-9\-\.]+\.[a-zA-Z]{2,3}(/\S*)?$',
                    'webaddress')

# Replace money symbols with 'moneysymb' (£ can by typed with ALT key + 156)
train['comment_text'] = train['comment_text'].str.replace(r'£|\$', 'dollers')

# Replace 10 digit phone numbers (formats include paranthesis, spaces, no spaces, dashes) with 'phonenumber'
train['comment_text'] = train['comment_text'].str.replace(r'^\(?[\d]{3}\)?[\s-]?[\d]{3}[\s-]?[\d]{4}$',
                    'phonenumber')


# Replace numbers with 'num'
train['comment_text'] = train['comment_text'].str.replace(r'\d+(\.\d+)?', 'num')


train['comment_text'] = train['comment_text'].apply(lambda x: ' '.join(
   term for term in x.split() if term not in string.punctuation))

stop_words = set(stopwords.words('english') + ['u', 'ü', 'ur', '4', '2', 'im', 'dont', 'doin', 'ure'])
train['comment_text'] = train['comment_text'].apply(lambda x: ' '.join(
   term for term in x.split() if term not in stop_words))

lem=WordNetLemmatizer()
train['comment_text'] = train['comment_text'].apply(lambda x: ' '.join(
 lem.lemmatize(t)fortinx.split(
```

After the removal of unwanted notations we moved to remove stopwords fromour dataset. For removing them we created another function named stopwords and append the text into another new column.

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

After removing all the unnecessary words or numbers we converted the wordsinto their simpler form using the Lemmatization process**.** In this process we defined two function,first one will tag the words into proper format and the other function will convert them into simpler form using the tagged alphabet.

```
cols_target = ['malignant','highly_malignant','rude','threat','abuse','loathe']
df_distribution = train[cols_target].sum()\
.to_frame()\
.rename(columns={0: 'count'})\
.sort_values('count')

df_distribution.plot.pie(y='count',
title='Label distribution over comments',
figsize=(5, 5))\
.legend(loc='center left', bbox_to_anchor=(1.3, 0.5))
```

Now we took a random sample from our dataset and compared the text of beforeand after the treatment.

Encoding the categorical data (Feature Extraction and scaling)

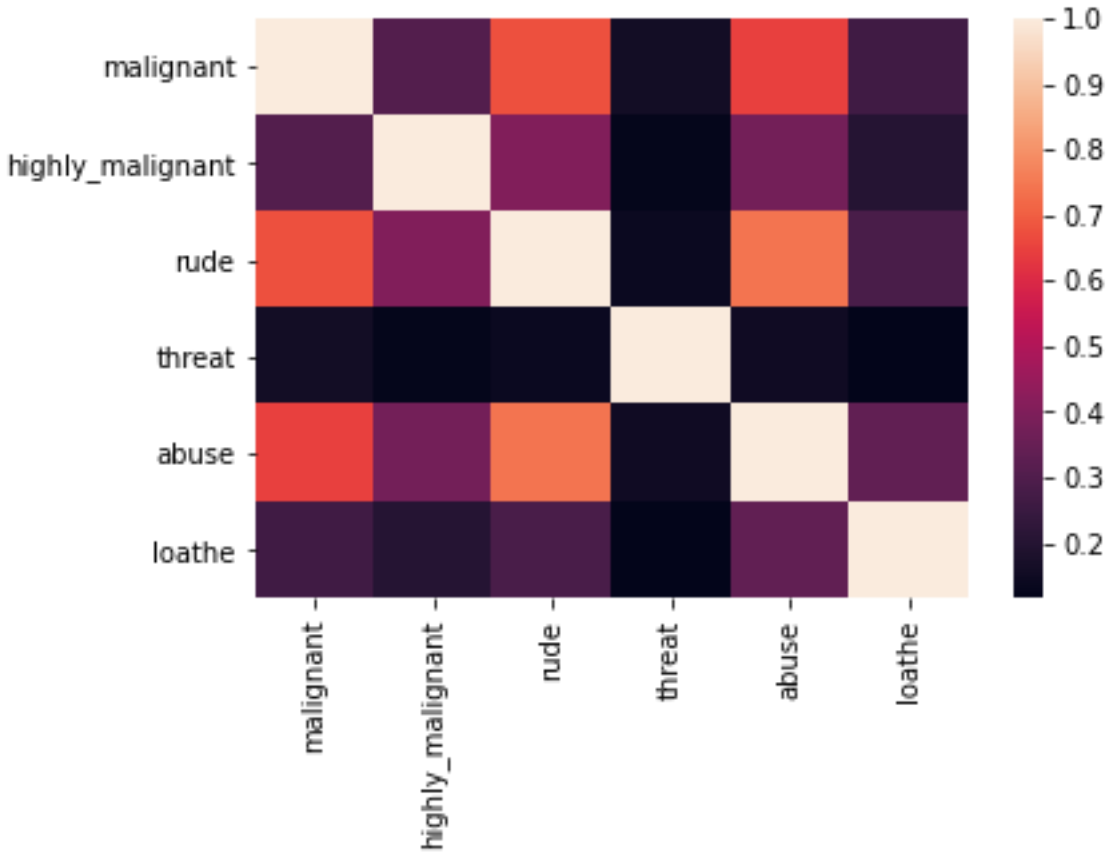
As of now we had cleaned the data , now another major step towards building a model is to convert the textual data into numericalform because our algorithms understand only the numerical data. So for converting into numerical or vector form we used Tf-Idf Vectorizer technique which converted the textual data into the vectors using the terms frequency method.

Splitting the dataset

The last step for data-preprocessing is splitting the dataset into training and testing dataset Using the train test split model selectionmethod we converted the dataset into training and testing.

```
y=train['bad']
x_train,x_test,y_train,y_test=train_test_split(x,y,random_state=56,test_size=.30)
```

Multivariate Analysis



All the harmful comments present in dataset are having positive correlation witheach others. Rude and abuse are highly correlated.

Hardware and Software Requirements and Tools Used

There is no such requirement for hardware ,but I have used intel i5 8th generation processor with 4 GB Ram.

Software: Jupyter Notebook (Anaconda 3)Language

Python 3.9

Libraries used in project:

Pandas ,Numpy ,Matplotlib, Seaborn ,Sklearn ,NLTK

```
from nltk.stem import WordNetLemmatizer
import nltk
from nltk.corpus import  stopwords
import string

import numpy as np
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
import warnings
warnings.filterwarnings('ignore')
```

Model/s Development and Evaluation

Identification of possible problem-solving approaches

In this project there were 5-6 features which defines the type of comment like malignant , hate , abuse, threat, loathe but we created another feature named as “negative_cmnts” which is combined of all the above features and contains the labelled data into the format of 0 and 1 where 0 represents “NO” and 1 represents “Yes”.

As we have labelled data into our target feature which is the case of classification method.

from sklearn.linear_model import LogisticRegression

from sklearn.tree import DecisionTreeClassifier

from sklearn.neighbors import KNeighborsClassifier

from sklearn.ensemble import RandomForestClassifier,AdaBoostClassifier,GradientBoostingClassifier

from sklearn.svm import SVC

Based on the classification approach we are going touse following approaches :

I. Logistic Regression
II. Decision Tree Classifier
III. RandomForest Classifier
IV. AdaBoost Classifier
V. KNN Classifier

In NLP we use Naïve Bayes algorithms mostly but due to our systems we facedmemory error to deal with them**.**

Performance of models

I. Logistic Regresison -

```
Training accuracy is 0.9595699155766838
Test accuracy is 0.9552974598930482
[[42729   221]
 [ 1919  3003]]
              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
```

II. Decision Tree Classifier –

```
Training accuracy is 0.9988898736783678
Test accuracy is 0.9388786764705882
[[41604  1346]
 [ 1580  3342]]
              precision    recall  f1-score   support

           0       0.96      0.97      0.97     42950
           1       0.71      0.68      0.70      4922

    accuracy                           0.94     47872
   macro avg       0.84      0.82      0.83     47872
weighted avg       0.94      0.94      0.94     47872
```

III. Random Forest Classifier –

```
Training accuracy is 0.9988719684151156
Test accuracy is 0.9550467914438503
[[42388   562]
 [ 1590  3332]]
              precision    recall  f1-score   support

           0       0.96      0.99      0.98     42950
           1       0.86      0.68      0.76      4922

    accuracy                           0.96     47872
   macro avg       0.91      0.83      0.87     47872
weighted avg       0.95      0.96      0.95     47872
```

IV. AdaBoost Classifier –

```
Training accuracy is 0.951118631321677
Test accuracy is 0.9490516377005348
[[42554   396]
 [ 2043  2879]]
              precision    recall  f1-score   support

           0       0.95      0.99      0.97     42950
           1       0.88      0.58      0.70      4922

    accuracy                           0.95     47872
   macro avg       0.92      0.79      0.84     47872
weighted avg       0.95      0.95      0.94     47872
```

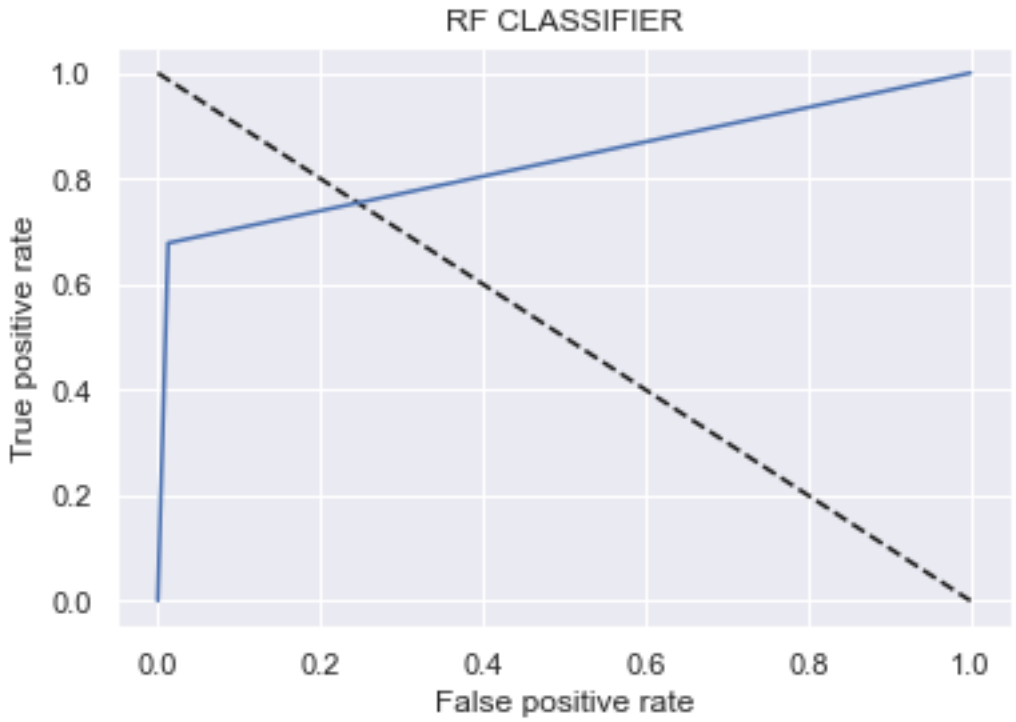
V. KNN Classifier –

```
Training accuracy is 0.9221837259062301
Test accuracy is 0.91717496657754
     [[42803   147]
     [ 3818  1104]]
              precision    recall  f1-score   support

           0       0.92      1.00      0.96     42950
           1       0.88      0.22      0.36      4922

    accuracy                           0.92     47872
   macro avg       0.90      0.61      0.66     47872
weighted avg       0.91      0.92      0.89     47872
```

Key Metrics for success in solving problem under consideration

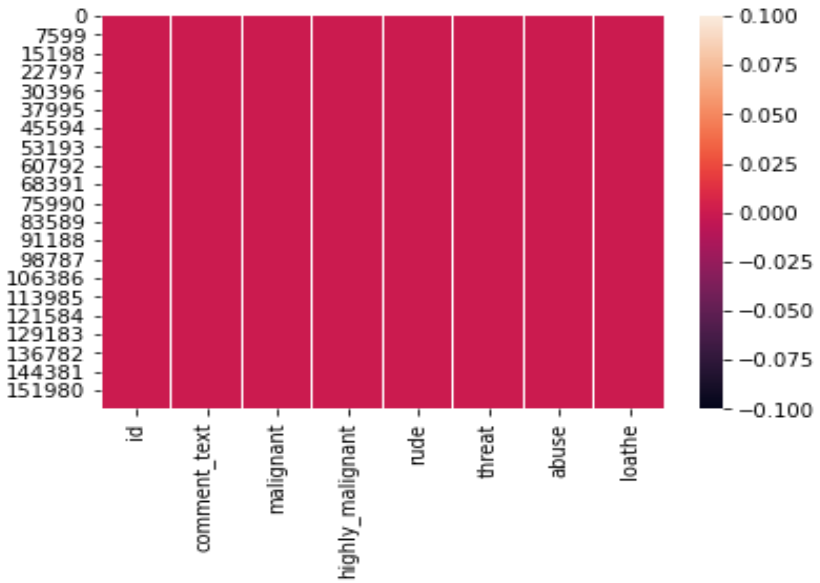
For solving the problems and understanding the resultof each algorithms we have used a lot of metrics :

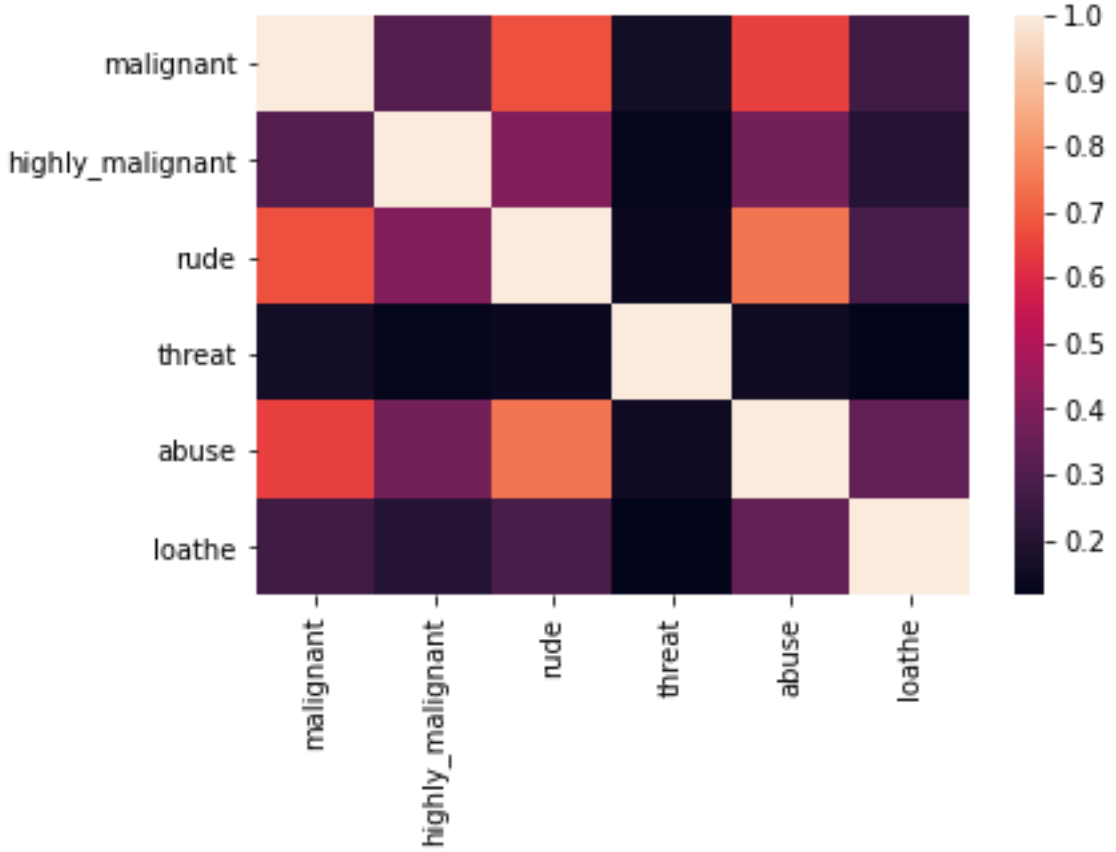
a) Accuracy Score
b) Classification Report
c) Confusion Matrix
d) Log Loss
e) Roc-Auc

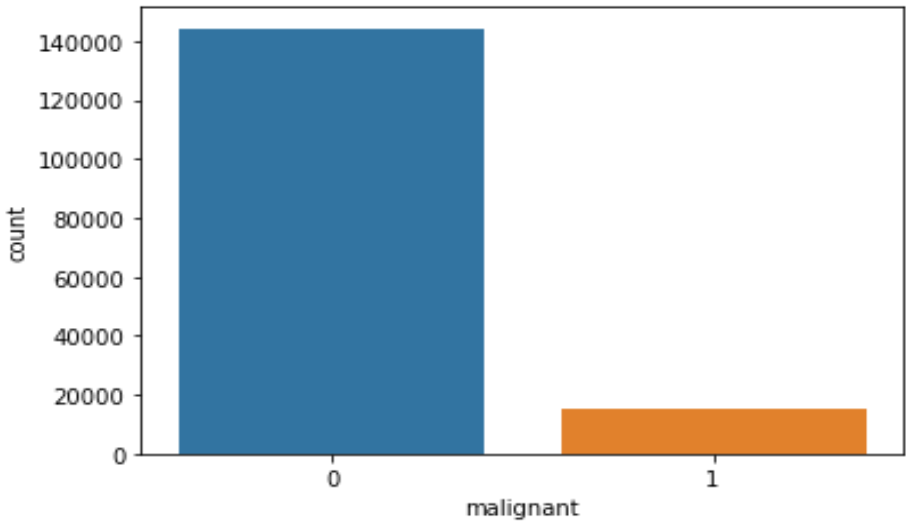


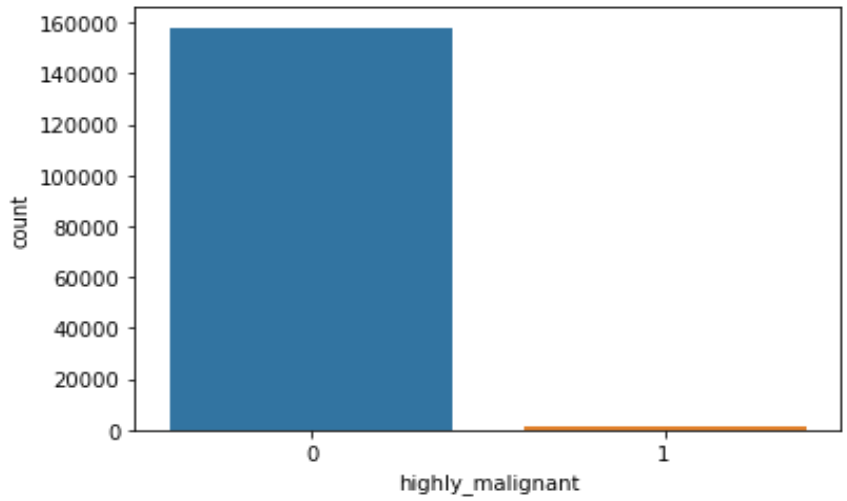
Visualizations

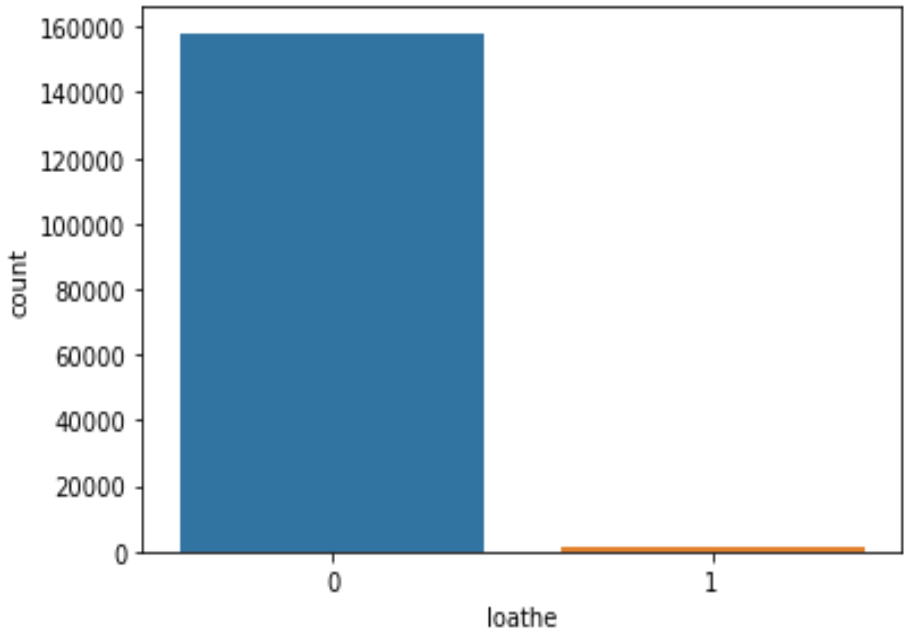
For the Visualization we have used Matplotlib and Seaborn library to plot the numerical data into graphs –

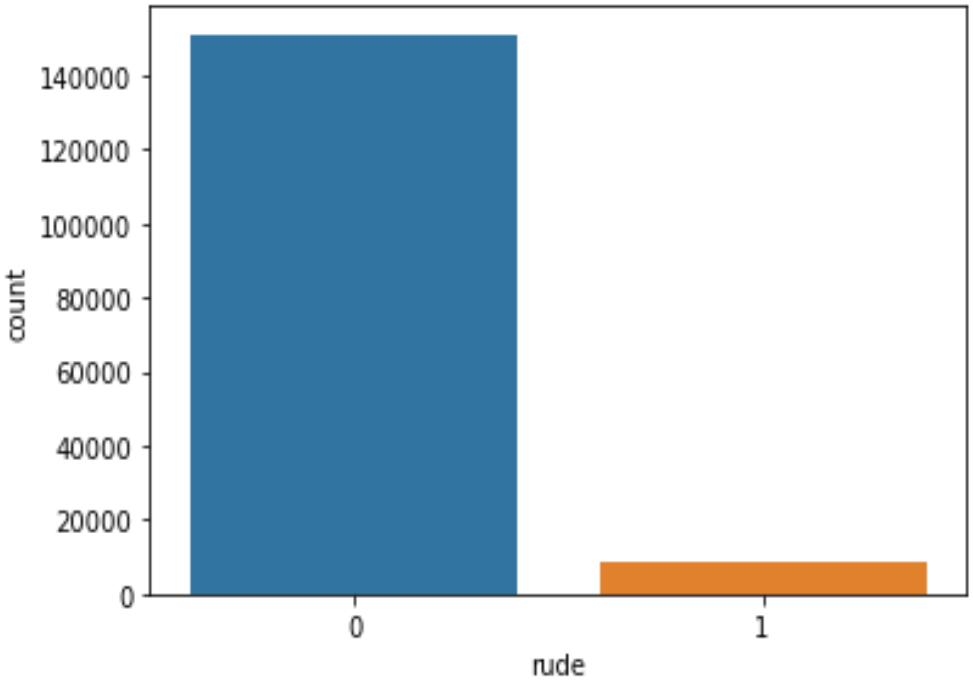


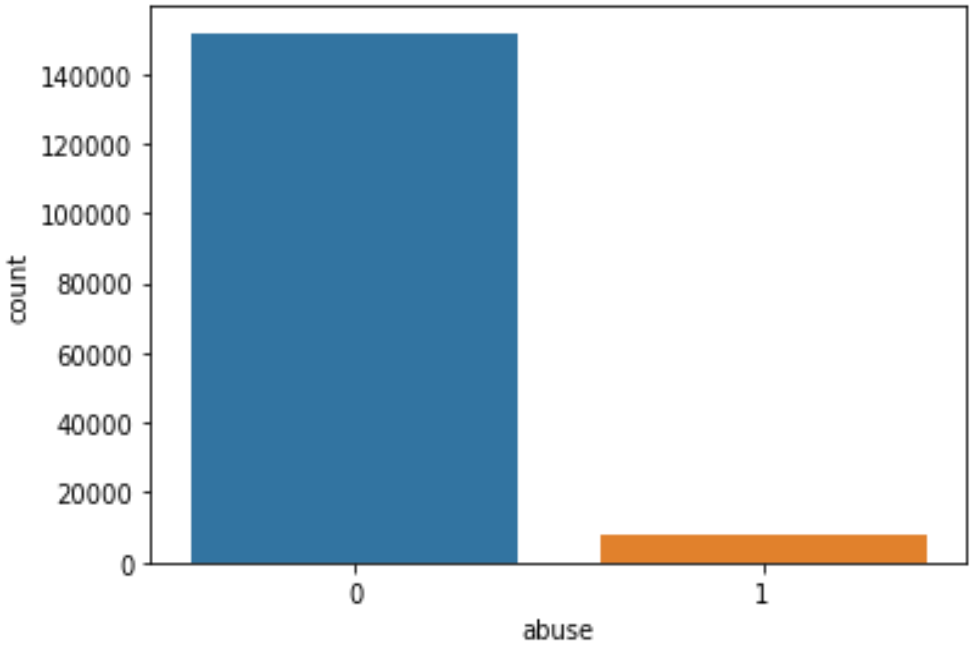

malignant
highly_malignant
rude
threat
abuse
loathe
1.0
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2

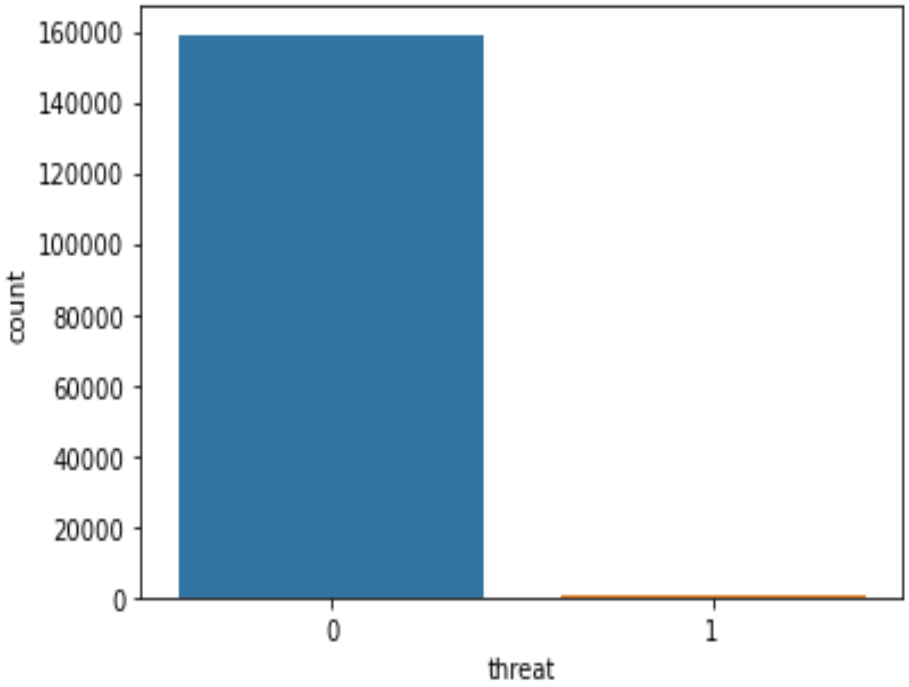

140000
120000
100000
80000
60000
40000
20000
0
count
0
1
malignant

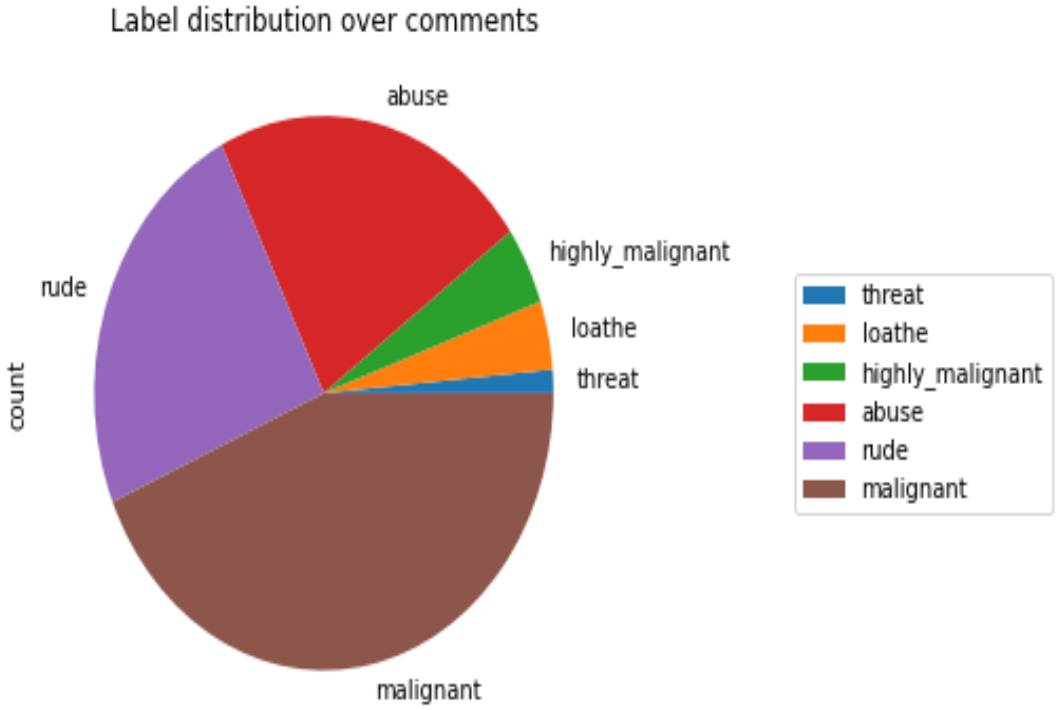

160000
140000
120000
100000
80000
60000
40000
20000
0
count
0
1
highly_malignant

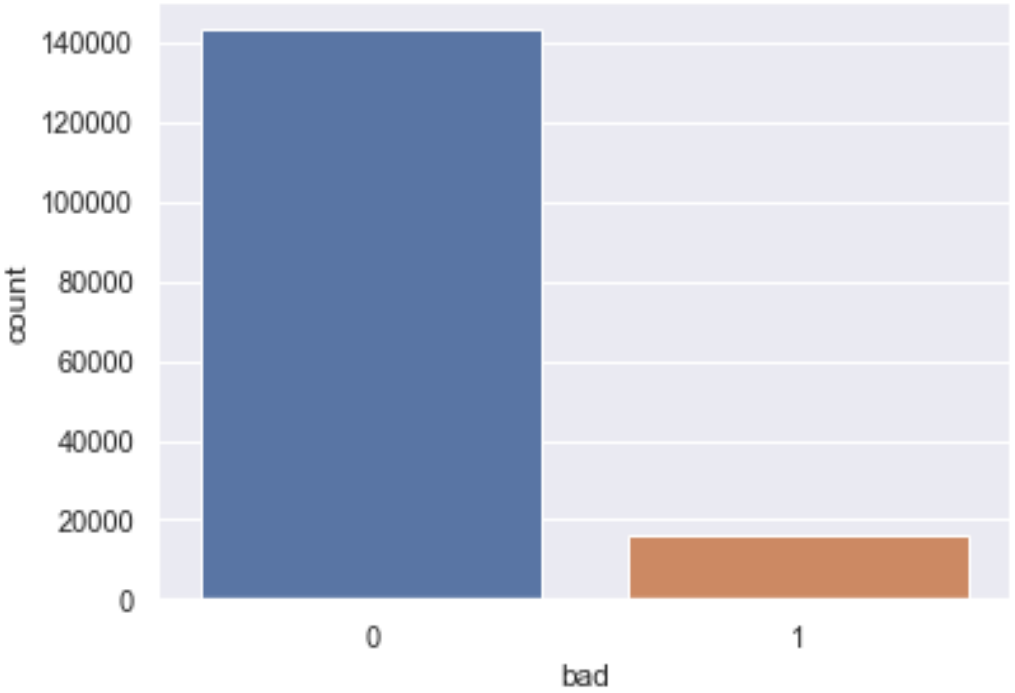

160000
140000
120000
100000
80000
60000
40000
20000
0
count
0
1
loathe

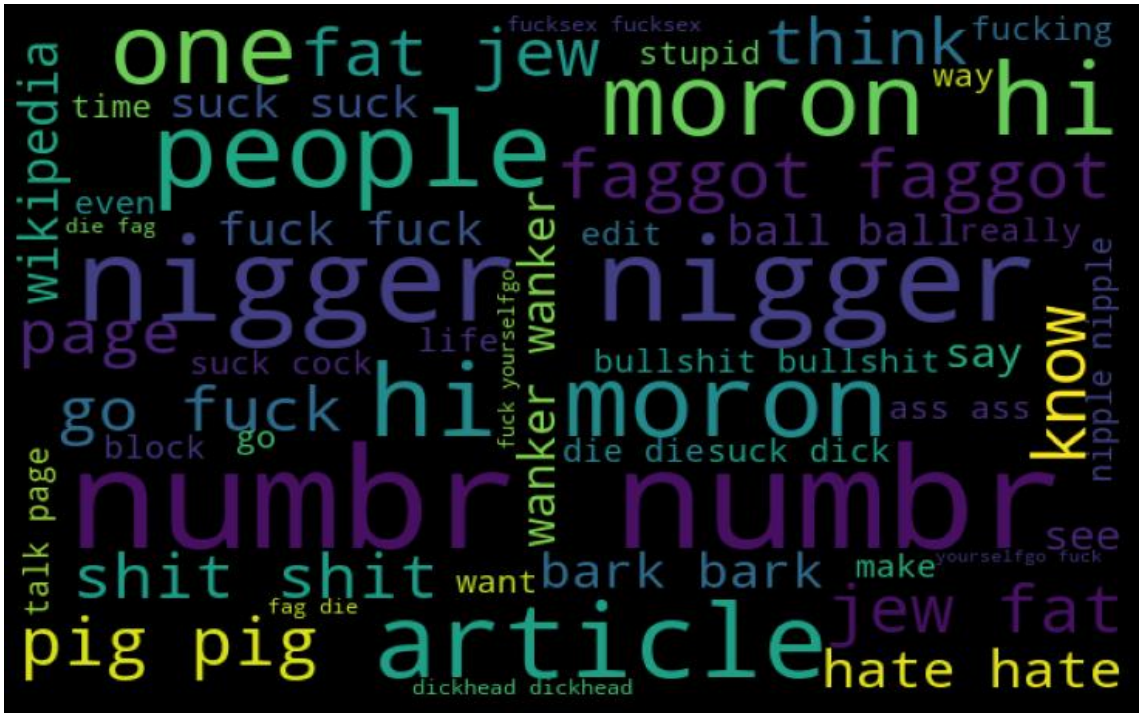

140000
120000
100000
80000
60000
40000
20000
0
count
0
1
rude


140000
120000
100000
80000
60000
40000
20000
0
count
0
1
abuse


160000
140000
120000
100000
80000
60000
40000
20000
0
count
0
1
threat


Label distribution over comments
abuse
highly_malignant
rude
loathe
threat
count
malignant
threat
loathe
highly_malignant
abuse
rude
malignant


140000
120000
100000
80000
60000
40000
20000
0
count
0
1
bad


fucksex fucksex
one
fat jew
stupid
think
fucking
wikipedia
time
suck suck
way
moron
hi
people
faggot faggot
even
die fag
fuck fuck
edit
ball ball
really
nigger
nigger
page
life
fuck yourselfgo
wanker
wanker
nipple
nipple
suck cock
bullshit bullshit
say
know
go fuck
hi
moron
ass ass
block
go
die diesuck dick
numbr
numbr
talk page
see
shit shit
want
bark bark
make
yourselfgo fuck
fag die
jew fat
pig pig
article
hate hate
dickhead dickhead

Interpretation of the Results

As our target feature is imbalanced which means thatalone accuracy score will not give best results but instead accuracy we are using theclassification report log loss score , Roc-Auc score and confusion matrix to find the best algorthim which is above all.

So we selected Logistic Regression and RandomForest Classifier based on above condition and use Randomized SearchCV for hyper tunning the parameters of these 2 selected algorithms .

After hyper tunning the parameters we selected the Logistic Regression algorithm as it give upto 85% score of Roc-Auc and losgg loss of 1.38 far better than RandomForest Classifier whose log loss was above 2.0 and again the classification report also tells that it is better than Random forest classifier

Conclusion

The conclusion for our study :-

a) In training dataset we have only 10% of data which is spreading hate onsocial media.
b) In this 10% data most of the comments are malignant, rude or abuse.
c) After using the wordcloud we find that there are so many abusive words present in the negative comments. While in positive comments there is nouse of such comments.
d) Some of the comments are very long while some are very short.

.

Learning Outcomes of the Study in respect of Data Science

From this project we learned a lot. Gains new techniques and ways to deal with uncleaned data. Find a solution to deal with multiple target features. Tools used for visualizations gives a better understanding of dataset. We have used a lot of algorithms and find that in the classification problem where we have only two labels , Logistic Regression gives better resultscompared to others.

But due to our system we could not use algorithms which gives much better results in NLP project like GaussinaNB, MultinomailNM. We also used googlecolab and some pipelines techniques but none of them worked here and also it was too muchtime consuming.

Limitations of this work and Scope for Future Work

This project was amazing to work on , it creates new ideas to think about but there were some limitations in this project like unbalanced dataset, multiple target features. To overcome these limitations we have to use balanced datset so that during the training of dataset our algorithm will not givebiased result.