FLIP ROBO

Project Name

MALIGNANT COMMENTS CLASSIFIER PROJECT REPORT

Submitted by:

Priyanka Saikia

ACKNOWLEDGMENT

I would like to express my deep sense of gratitude to my SME (Subject Matter Expert) **Mr. Shubham Yadav** as well as **Flip Robo Technologies** who gave me the golden opportunity to do this data analysis project on **Malignant Comments Classifier**, which also helped me in doing lots of research and I came to know about so many new things.
I have put in my all efforts while doing this project.

Priyanka Saikia

INTRODUCTION

• Business Problem Framing:

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.

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 un-offensive, 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.

• Conceptual Background of the Domain Problem:

Online platforms and social media become the place where people share the thoughts freely without any partiality and overcoming all the race people share their thoughts and ideas among the crowd.

Social media is a computer-based technology that facilitates the sharing of ideas, thoughts, and information through the building of virtual networks and communities. By design, social media is Internet-based and gives users quick electronic communication of content. Content includes personal information, documents, videos, and photos. Users engage with social media via a computer, tablet, or smartphone via web-based software or applications.

While social media is ubiquitous in America and Europe, Asian countries like India lead the list of social media usage. More than 3.8 billion people use social media.

In this huge online platform or an online community there are some people or some motivated mob wilfully bully others to make them not to share their thought in rightful way. They bully others in a foul language which among the civilized society is seen as ignominy. And when innocent individuals are being bullied by these mob these individuals are going silent without speaking anything. So, ideally the motive of this disgraceful mob is achieved.

To solve this problem, we are now building a model that identifies all the foul language and foul words, using which the online platforms like social media principally stops these mob using the foul language in an online community or even block them or block them from using this foul language.

• Review of Literature

The purpose of the literature review is to:

1. Identify the foul words or foul statements that are being used.

2. Stop the people from using these foul languages in online public forum.

To solve this problem, we are now building a model using our machine language technique that identifies all the foul language and foul words, using which the online platforms like social media principally stops these mob using the foul language in an online community or even block them or block them from using this foul language.

I have used 9 different Classification algorithms and shortlisted the best on basis of the metrics of performance and I have chosen one algorithm and build a model in that algorithm.

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.

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.

• Motivation for the Problem Undertaken

Social media is reverting us back to those animalistic tantrums, schoolyard taunts and unfettered bullying that define youth, creating a dystopia.

With widespread usage of online social networks and its popularity, social networking platforms have given us incalculable opportunities than ever before, and its benefits are undeniable. Despite benefits, people may be humiliated, insulted, bullied, and harassed by anonymous users, strangers, or peers. In this study, we have proposed a cyberbullying detection framework to generate features from online content by leveraging a point-wise mutual information technique. Based on these features, we developed a supervised machine learning solution for cyberbullying detection and multi-class categorization of its severity. Results from experiments with our proposed framework in a multi-class setting are promising both with respect to classifier accuracy and f-measure metrics. These results indicate that our proposed framework provides a feasible solution to detect cyberbullying behaviour and its severity in online social networks.

Analytical Problem Framing

Mathematical/ Analytical Modeling of the Problem

In this project, we have been provided with two datasets namely train and test CSV files. We will build a machine learning model by using NLP using train dataset. And using this model we will make predictions for our test dataset.

We will need to build multiple classification machine learning models. Before model building we will need to perform all data pre-processing steps involving NLP. After trying different classification models with different hyper parameters then will select the best model out of it. We will need to follow the complete life cycle of data science that includes steps like -

1. Data Cleaning

2. Exploratory Data Analysis

3. Data Pre-processing

4. Model Building

5. Model Evaluation

6. Selecting the best model

Finally, we compared the results of proposed and baseline features with other machine learning algorithms. Findings of the comparison indicate the significance of the proposed features in cyberbullying detection.

Data Sources and their formats

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. There are various comments which have multiple labels. The first attribute is a unique ID associated with each comment.

The data set includes:

Malignant: It is the Label column, which includes values 0 and 1, denoting if the comment 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 to someone.

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

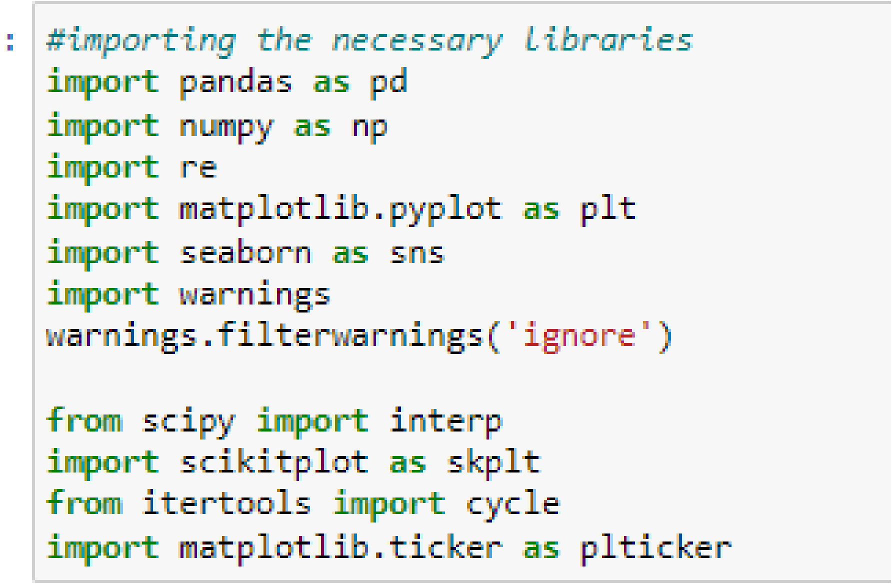
Loathe: It describes the comments which are hateful and loathing in nature.

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

Comment text: This column contains the comments extracted from various social media platforms.

This project is more about exploration, feature engineering and classification that can be done on this data. Since the data set is huge and includes many categories of comments, we can do good amount of data exploration and derive some interesting features using the comments text column available. We need to build a model that can differentiate between comments and its categories.

Importing Libraries

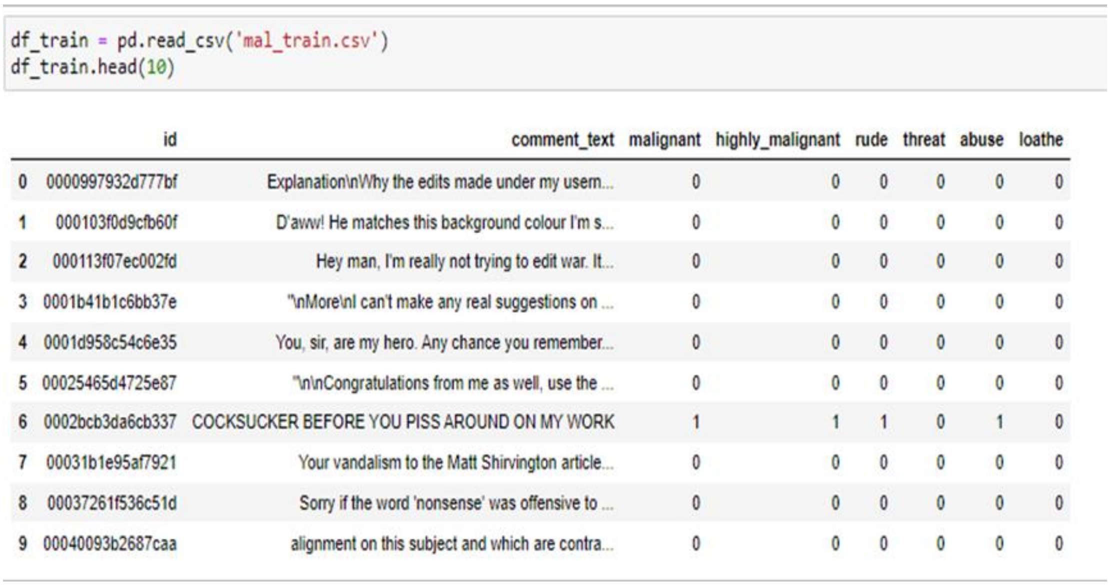


```
#importing the necessary libraries
import pandas as pd
import numpy as np
import re
import matplotlib.pyplot as plt
import seaborn as sns
import warnings
warnings.filterwarnings('ignore')

from scipy import interp
import scikitplot as skplt
from itertools import cycle
import matplotlib.ticker as plticker
```

- ## Data Pre-processing Done:

The following pre-processing pipeline is required to be performed before building the classification model prediction:

1. Load dataset:



```
df_train = pd.read_csv('mal_train.csv')
df_train.head(10)
```

	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
5	00025465d4725e87	"\n\nCongratulations from me as well, use the ...	0	0	0	0	0	0
6	0002bcb3da6cb337	COCKSUCKER BEFORE YOU PISS AROUND ON MY WORK	1	1	1	0	1	0
7	00031b1e95af7921	Your vandalism to the Matt Shirvington article...	0	0	0	0	0	0
8	00037261f536c51d	Sorry if the word 'nonsense' was offensive to ...	0	0	0	0	0	0
9	00040093b2687caa	alignment on this subject and which are contra...	0	0	0	0	0	0

2. Checking and removing null values.

3. Drop column ‘id’

4. Convert comment text to lower case and replace '\n' with single space.

5. Keep only text data ie. ‘a-z' and remove other data from comment text.

6. Remove stop words and punctuations

7. Apply Stemming using SnowballStemmer

8. Convert text to vectors using TfidfVectorizer

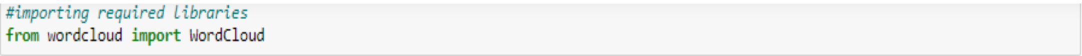
9. Load saved or serialized model

10. Predict values for multi class label.

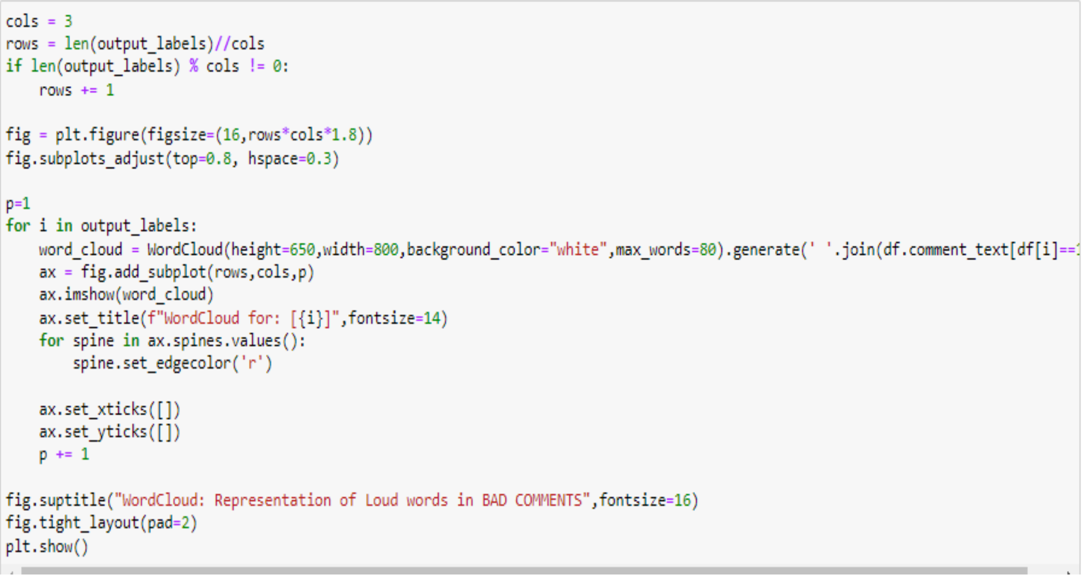
- ## Data Inputs- Logic- Output Relationships

We have analysed the input output logic with wordcloud and have word clouded the sentenced that are classified as foul language in every category. A tag/word cloud is a novelty visual representation of text data, typically used to depict keyword metadata on websites, or to visualize free form text. It’s an image composed of words used in a particular text or subject, in which the size of each word indicates its frequency or importance.

Code:

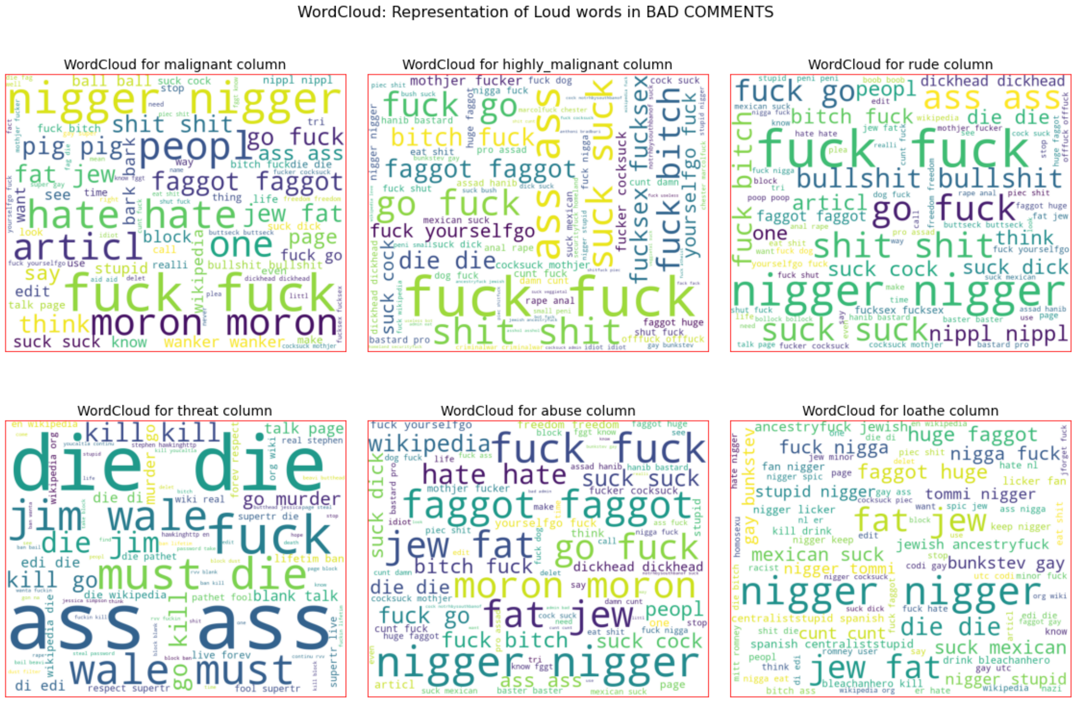


```
#importing required libraries
from wordcloud import WordCloud
```



```
cols = 3
rows = len(output_labels)//cols
if len(output_labels) % cols != 0:
    rows += 1

fig = plt.figure(figsize=(16,rows*cols*1.8))
fig.subplots_adjust(top=0.8, hspace=0.3)

p=1
for i in output_labels:
    word_cloud = WordCloud(height=650,width=800,background_color="white",max_words=80).generate(' '.join(df.comment_text[df[i]==1
    ax = fig.add_subplot(rows,cols,p)
    ax.imshow(word_cloud)
    ax.set_title(f"WordCloud for: [{i}]",fontsize=14)
    for spine in ax.spines.values():
        spine.set_edgecolor('r')

    ax.set_xticks([])
    ax.set_yticks([])
    p += 1

fig.suptitle("WordCloud: Representation of Loud words in BAD COMMENTS",fontsize=16)
fig.tight_layout(pad=2)
plt.show()
```

Output:



These are the comments that belongs to different type so which the help of word cloud we can see if there is abuse comment which type of words it contains and similar to other comments as well.

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

Cyberbullying has become a growing problem in countries around the world. Essentially, cyberbullying doesn't differ much from the type of bullying that many children have unfortunately grown accustomed to in school. The only difference is that it takes place online.

Cyberbullying is a very serious issue affecting not just the young victims, but also the victims' families, the bully, and those who witness instances of cyberbullying. However, the effect of cyberbullying can be most detrimental to the victim, of course, as they may experience a number of emotional issues that affect their social and academic performance as well as their overall mental health.

- ## Hardware and Software Requirements and Tools Used

➢ Hardware Used:
RAM: 8 GB
CPU: AMD A8 Quad Core 2.2 Ghz
GPU: AMD Redon R5 Graphics

➢ Software Tools used:
Programming language: Python 3.0
Distribution: Anaconda Navigator
Browser-based language shell: Jupyter Notebook

➢ Libraries/Packages Used:

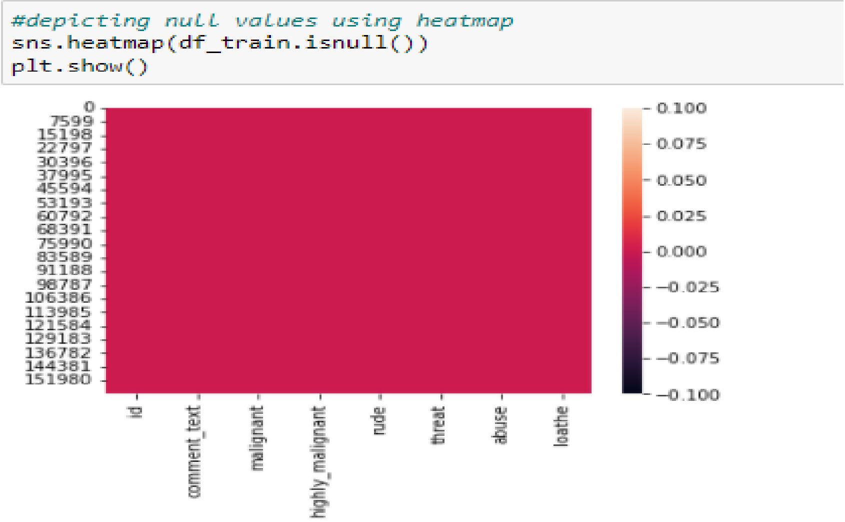
- Pandas
- Numpy
- Matplotlib
- Seaborn
- Scipy.stats
- Sklearn
- NLTK

Model/s Development and Evaluation

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

We checked through the entire training dataset for any kind of missing values information and all these pre-processing steps were repeated on the testing dataset as well.

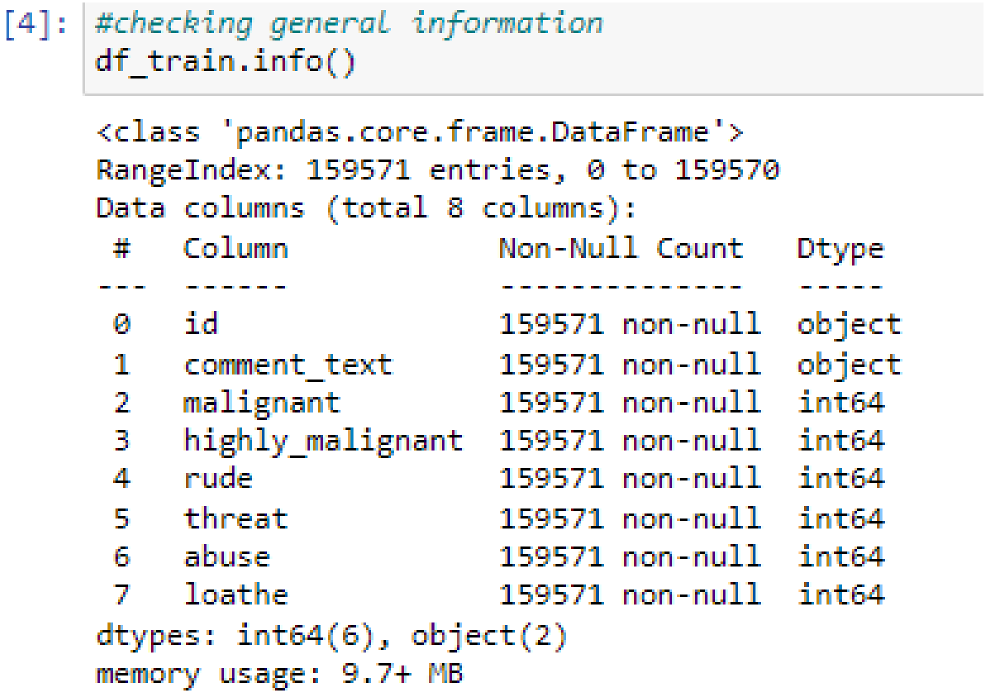
Code:



```
#depicting null values using heatmap
sns.heatmap(df_train.isnull())
plt.show()
```

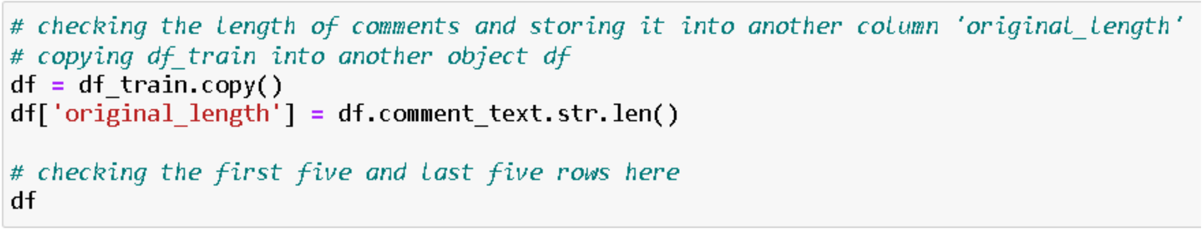
Then we went ahead and took a look at the dataset information. Using the info method, we are able to confirm the non-null count details as well as the datatype information. We have a total of 8 columns out of which 2 columns have object datatype while the remaining 6 columns are of integer datatype.

Code:



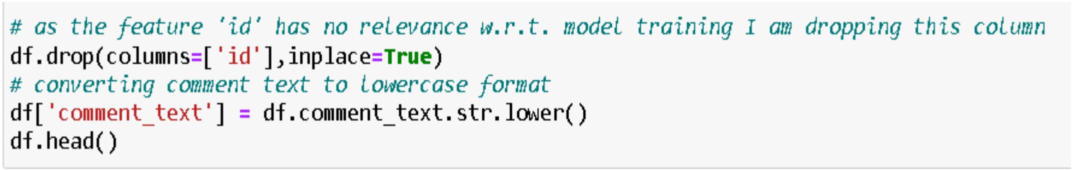
```
[4]: #checking general information
     df_train.info()

     <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
```

Then we went ahead and performed multiple data cleaning and data transformation steps. We have added an additional column to store the original length of our comment_text column.



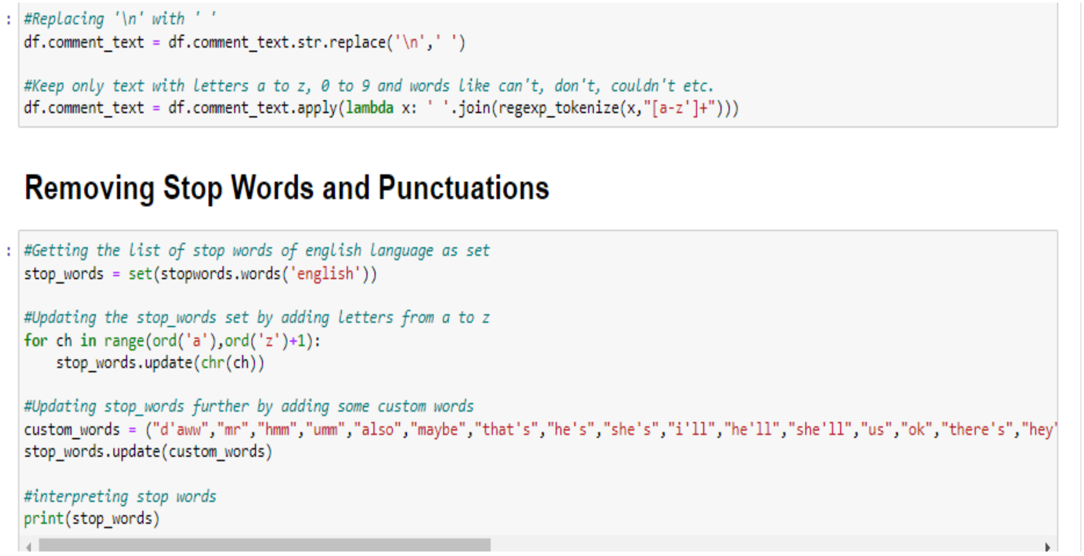
```
# checking the length of comments and storing it into another column 'original_length'
# copying df_train into another object df
df = df_train.copy()
df['original_length'] = df.comment_text.str.len()

# checking the first five and last five rows here
df
```

Since there was no use of the "id" column we have dropped it and converted all the text data in our comment text column into lowercase format for easier interpretation.



```
# as the feature 'id' has no relevance w.r.t. model training I am dropping this column
df.drop(columns=['id'],inplace=True)
# converting comment text to lowercase format
df['comment_text'] = df.comment_text.str.lower()
df.head()
```

Stemming is the process of reducing a word to its word stem that affixes to suffixes and prefixes or to the roots of words known as a lemma. Stemming is important in natural language understanding (NLU) and natural language processing (NLP).

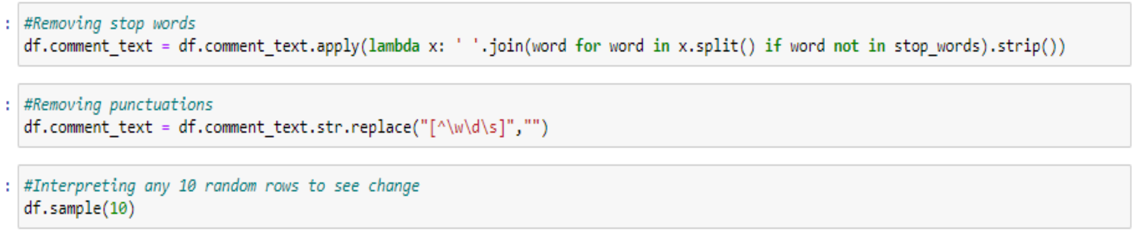


```
#Replacing '\n' with ' '
df.comment_text = df.comment_text.str.replace('\n',' ')

#Keep only text with letters a to z, 0 to 9 and words like can't, don't, couldn't etc.
df.comment_text = df.comment_text.apply(lambda x: ' '.join(regexp_tokenize(x,"[a-z']+")))
```

Removing Stop Words and Punctuations

```
#Getting the list of stop words of english language as set
stop_words = set(stopwords.words('english'))

#Updating the stop_words set by adding letters from a to z
for ch in range(ord('a'),ord('z')+1):
    stop_words.update(chr(ch))

#Updating stop_words further by adding some custom words
custom_words = ("d'aww","mr","hmm","umm","also","maybe","that's","he's","she's","i'll","he'll","she'll","us","ok","there's","hey'
stop_words.update(custom_words)

#interpreting stop words
print(stop_words)
```

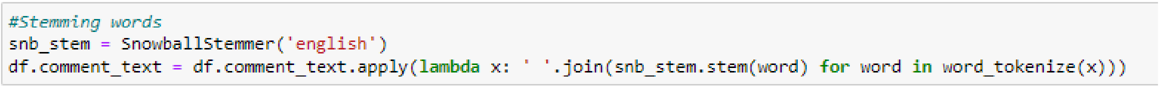
Here we have removed all the unwanted data from our comment column.



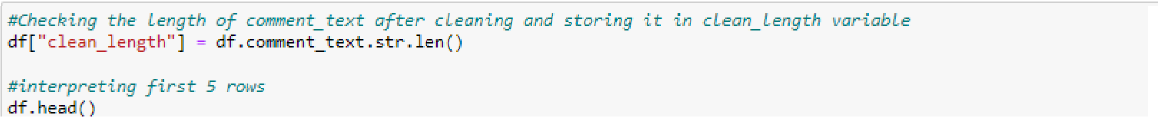
```
#Removing stop words
df.comment_text = df.comment_text.apply(lambda x: ' '.join(word for word in x.split() if word not in stop_words).strip())
```

```
#Removing punctuations
df.comment_text = df.comment_text.str.replace("[^\w\d\s]","")
```

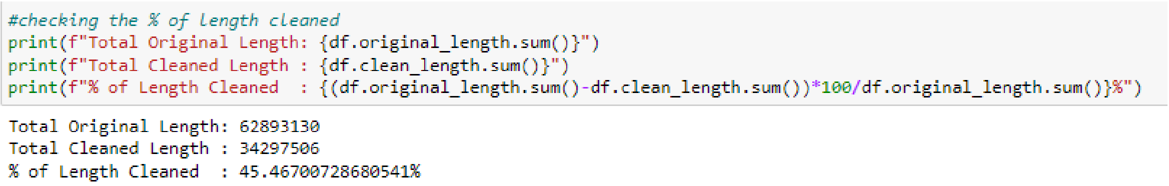
```
#Interpreting any 10 random rows to see change
df.sample(10)
```



```
#Stemming words
snb_stem = SnowballStemmer('english')
df.comment_text = df.comment_text.apply(lambda x: ' '.join(snb_stem.stem(word) for word in word_tokenize(x)))
```



```
#Checking the length of comment_text after cleaning and storing it in clean_length variable
df["clean_length"] = df.comment_text.str.len()

#interpreting first 5 rows
df.head()
```



```
#checking the % of length cleaned
print(f"Total Original Length: {df.original_length.sum()}")
print(f"Total Cleaned Length : {df.clean_length.sum()}")
print(f"% of Length Cleaned  : {(df.original_length.sum()-df.clean_length.sum())*100/df.original_length.sum()}%")
```

```
Total Original Length: 62893130
Total Cleaned Length : 34297506
% of Length Cleaned  : 45.46700728680541%
```

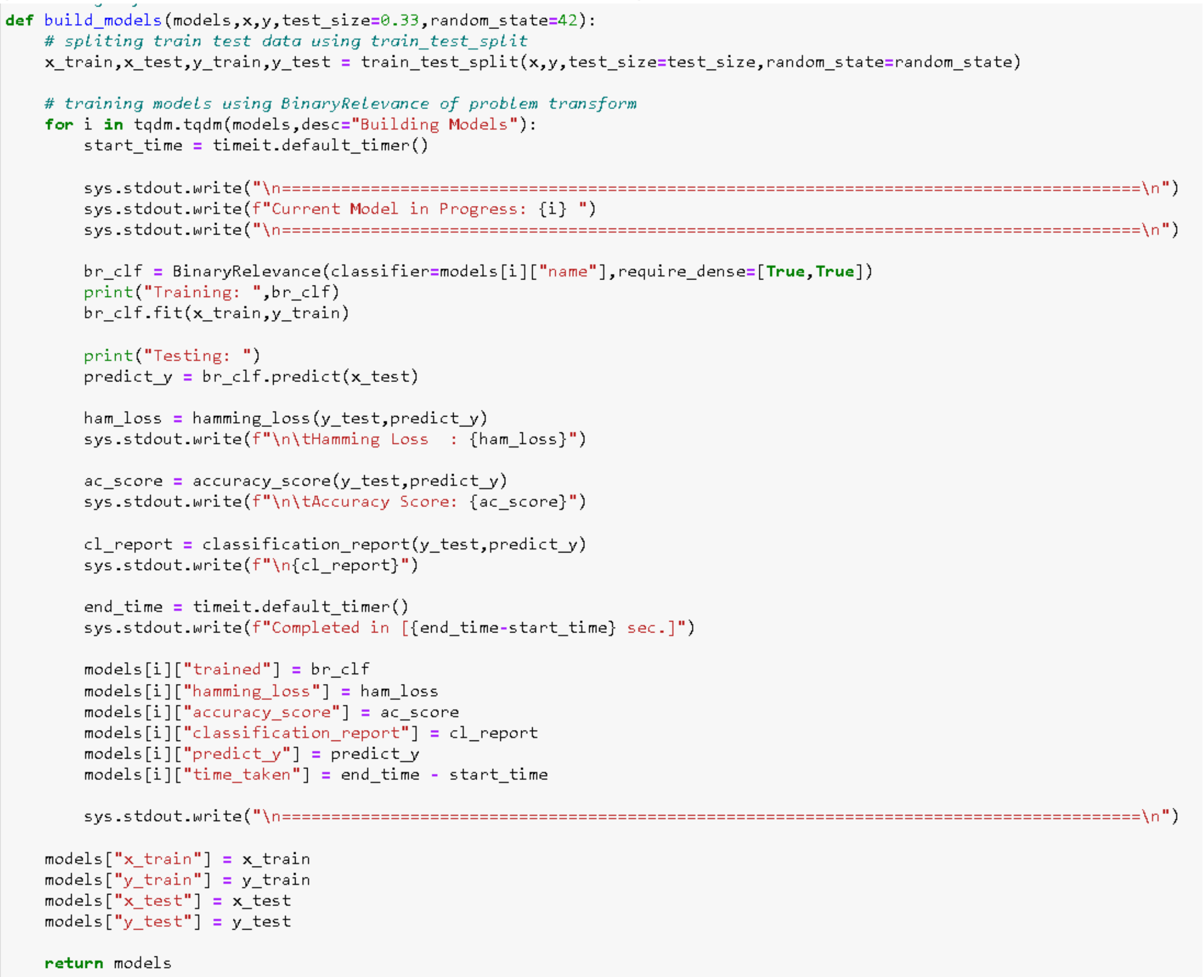
- ## Testing of Identified Approaches (Algorithms)

The list of all the algorithms used for the training and testing classification model are listed below:

1) Gaussian Naïve Bayes
2) Multinomial Naïve Bayes

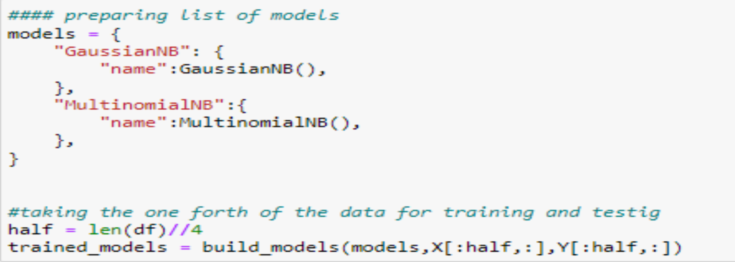
- ## Run and Evaluate selected models

We created a classification function that included the evaluation metrics details for the generation of our Classification Machine Learning models.



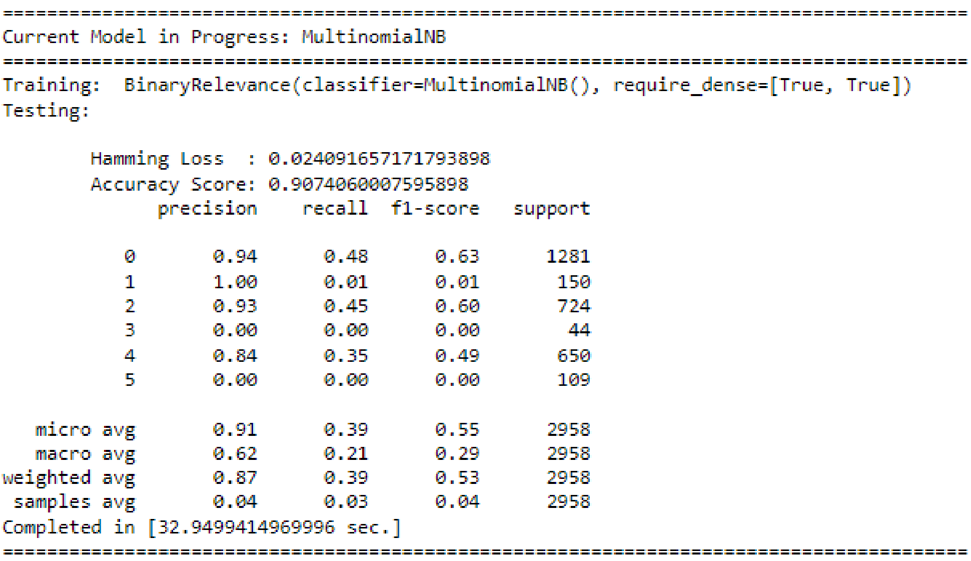
```
def build_models(models,x,y,test_size=0.33,random_state=42):
    # spliting train test data using train_test_split
    x_train,x_test,y_train,y_test = train_test_split(x,y,test_size=test_size,random_state=random_state)

    # training models using BinaryRelevance of problem transform
    for i in tqdm.tqdm(models,desc="Building Models"):
        start_time = timeit.default_timer()

        sys.stdout.write("\n======================================================================================\n")
        sys.stdout.write(f"Current Model in Progress: {i} ")
        sys.stdout.write("\n======================================================================================\n")

        br_clf = BinaryRelevance(classifier=models[i]["name"],require_dense=[True,True])
        print("Training: ",br_clf)
        br_clf.fit(x_train,y_train)

        print("Testing: ")
        predict_y = br_clf.predict(x_test)

        ham_loss = hamming_loss(y_test,predict_y)
        sys.stdout.write(f"\n\tHamming Loss  : {ham_loss}")

        ac_score = accuracy_score(y_test,predict_y)
        sys.stdout.write(f"\n\tAccuracy Score: {ac_score}")

        cl_report = classification_report(y_test,predict_y)
        sys.stdout.write(f"\n{cl_report}")

        end_time = timeit.default_timer()
        sys.stdout.write(f"Completed in [{end_time-start_time} sec.]")

        models[i]["trained"] = br_clf
        models[i]["hamming_loss"] = ham_loss
        models[i]["accuracy_score"] = ac_score
        models[i]["classification_report"] = cl_report
        models[i]["predict_y"] = predict_y
        models[i]["time_taken"] = end_time - start_time

        sys.stdout.write("\n======================================================================================\n")

    models["x_train"] = x_train
    models["y_train"] = y_train
    models["x_test"] = x_test
    models["y_test"] = y_test

    return models
```

Code:



```
#### preparing list of models
models = {
    "GaussianNB": {
        "name":GaussianNB(),
    },
    "MultinomialNB":{
        "name":MultinomialNB(),
    },
}


#taking the one forth of the data for training and testig
half = len(df)//4
trained_models = build_models(models,X[:half,:],Y[:half,:])
```

Output:

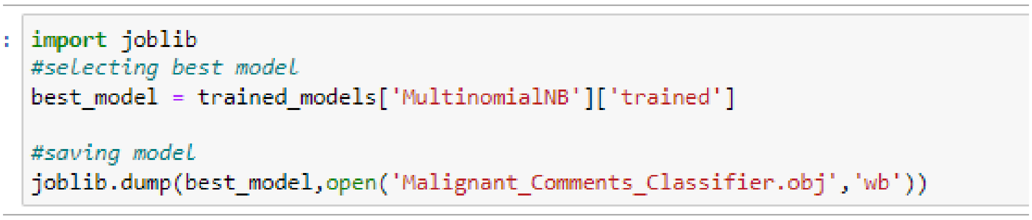


```
====================================================================================
Current Model in Progress: MultinomialNB
====================================================================================
Training:  BinaryRelevance(classifier=MultinomialNB(), require_dense=[True, True])
Testing:

        Hamming Loss  : 0.024091657171793898
        Accuracy Score: 0.9074060007595898
              precision    recall  f1-score   support

           0       0.94      0.48      0.63      1281
           1       1.00      0.01      0.01       150
           2       0.93      0.45      0.60       724
           3       0.00      0.00      0.00        44
           4       0.84      0.35      0.49       650
           5       0.00      0.00      0.00       109

   micro avg       0.91      0.39      0.55      2958
   macro avg       0.62      0.21      0.29      2958
weighted avg       0.87      0.39      0.53      2958
 samples avg       0.04      0.03      0.04      2958
Completed in [32.9499414969996 sec.]
====================================================================================
```

Observation:

From the above model comparision it is clear that MultinomialNB performs better with Accuracy Score: 90.74% and Hamming Loss: 2.4% than other models. Therefore, we will use MultinoimialNB.

Saving the best model:



```
import joblib
#selecting best model
best_model = trained_models['MultinomialNB']['trained']

#saving model
joblib.dump(best_model,open('Malignant_Comments_Classifier.obj','wb'))
```

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

To find out best performing model following metrices are used-

1. Accuracy Score: It is used to check the model performance score between 0.0 to 1.0.

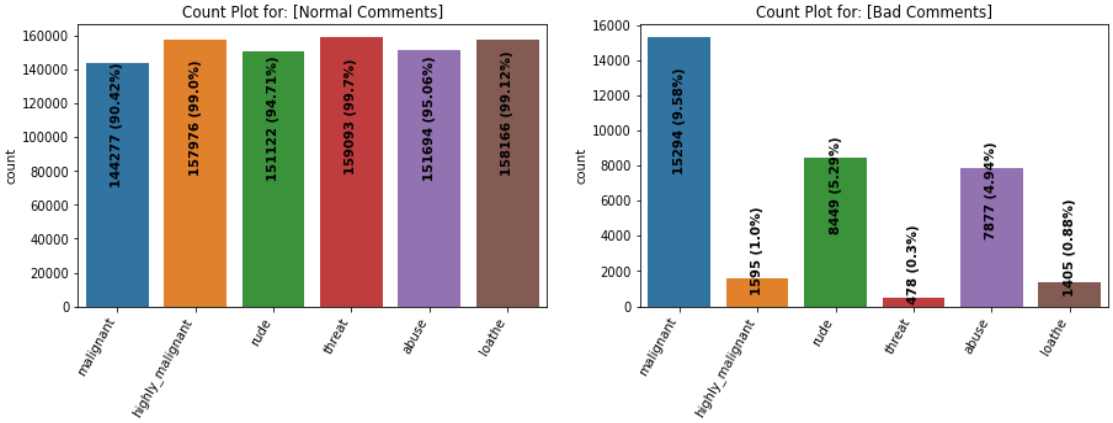
2. Hamming Loss: The Hamming Loss is the fraction of labels that are incorrectly predicted.
3. Classification report: A classification report is used to measure the quality of predictions from a classification algorithm. How many predictions are True and how many are False.

- ## Visualizations:

To better understand the data, the following types of visualizations have been used.

- Univariate analysis is the simplest form of data analysis where the data being analyzed contains only one variable. In this project, distribution plot, count plot, box plot and bar plot has been used.

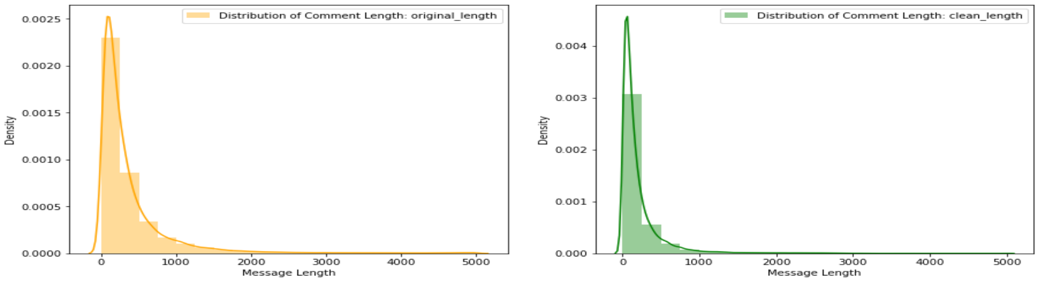
Count Plot:



Observations:

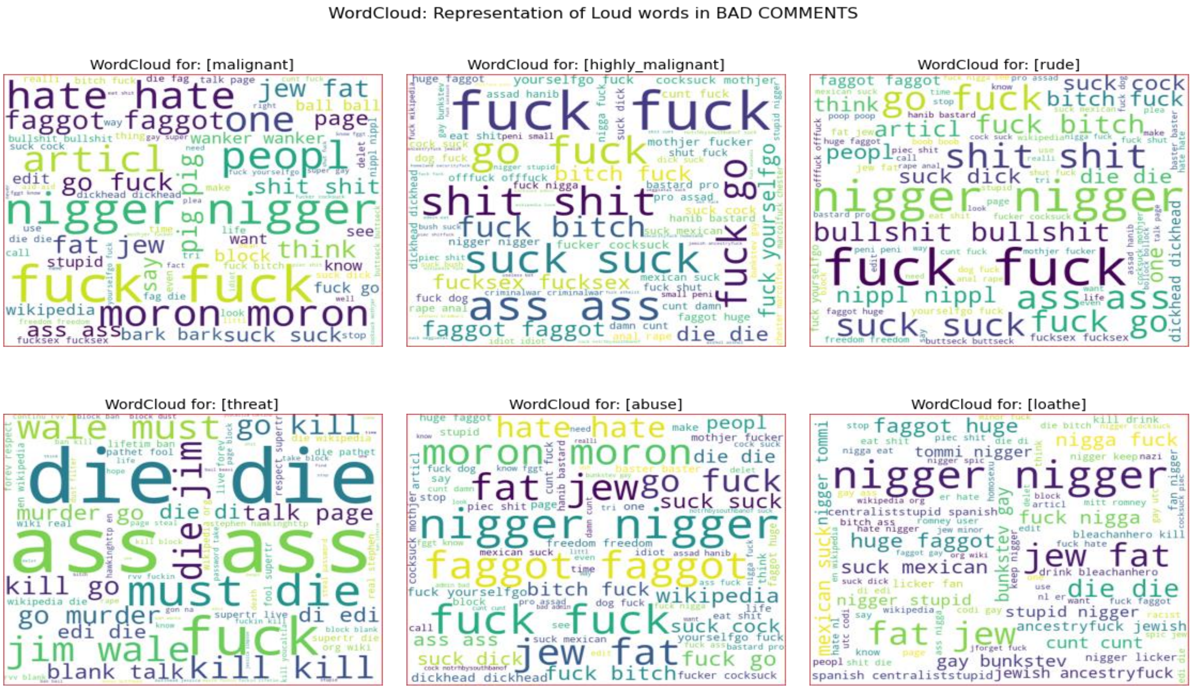
- Dataset consists of higher number of Normal Comments than Bad or Malignant Comments. Therefore, it is clear that dataset is imbalanced and needs to be handle accordingly.
- Most of the bad comments are of type malignant while least number of type threat is present in dataset.
- Majority of bad comments are of type malignant, rude and abuse.

Distribution Plot:



Observations: Before cleaning comment text, most of the comment's length lies between 0 to 1100 while after cleaning, it lies between 0 to 900.

Displaying with WordCloud:



Observations:

- From wordcloud of malignant comments, it is clear that it mostly consists of words like fuck, nigger, moron, hate, suck etc.
- From wordcloud of highly_malignant comments, it is clear that it mostly consists of words like ass, fuck, bitch, shit, die, suck, faggot etc.
- From wordcloud of rude comments, it is clear that it mostly consists of words like nigger, ass, fuck, suck, bullshit, bitch etc.
- From wordcloud of threat comments, it is clear that it mostly consists of words like die, must die, kill, murder etc.
- From wordcloud of abuse comments, it is clear that it mostly consists of words like moron, nigger, fat, jew, bitch etc.
- From wordcloud of loathe comments, it is clear that it mostly consists of words like nigga, stupid, nigger, die, gay cunt etc.

• Interpretation of the Results

Starting with univariate analysis, with the help of count plot it was found that dataset is imbalanced with having higher number of records for normal comments than bad comments (including malignant, highly malignant, rude, threat, abuse and loathe). Also, with the help of distribution plot for comments length it was found that after cleaning most of comments length decreases from range 0-1100 to 0-900. Moving further with word cloud it was found that malignant comments consists of words like fuck, nigger, moron, hate, suck etc. highly_malignant comments consists of words like ass, fuck, bitch, shit, die, suck, faggot etc. rude comments consists of words like nigger, ass, fuck, suck, bullshit, bitch etc. threat comments consists of words like die, must die, kill, murder etc. abuse comments consists of words like moron, nigger, fat, jew, bitch etc. and loathe comments consists of words like nigga, stupid, nigger, die, gay, cunt etc.

CONCLUSION

Key Findings and Conclusions of the Study

The finding of the study is that only few users over online use un-parliamentary language. And most of these sentences have more stop words and are being quite long. As discussed before few motivated disrespectful crowds use these foul languages in the online forum to bully the people around and to stop them from doing these things that they are not supposed to do. Our study helps the online forums and social media to induce a ban to profanity or usage of profanity over these forums.

Learning Outcomes of the Study in respect of Data Science

Through this project we were able to learn various Natural language processing techniques like lemmatization, stemming, removal of stopwords. We were also able to learn to convert strings into vectors through hash vectorizer. In this project we applied different evaluation metrics like log loss, hamming loss besides accuracy.

My point of view from the project is that we need to use proper words which are respectful and also avoid using abusive, vulgar and worst words in social media. It can cause many problems which could affect our lives. Try to be polite, calm and composed while handling stress and negativity and one of the best solutions is to avoid it and overcoming in a positive manner.

Limitations of this work and Scope for Future Work

Problems faced while working in this project:

- More computational power was required as it took more than 2 hours
- Imbalanced dataset and bad comment texts

Areas of improvement:

- Could be provided with a good dataset which does not take more time.
- Less time complexity
- Providing a proper balanced dataset with less errors.

References:

1) https://www.google.com/
2) https://www.youtube.com/
3) https://github.com/
4) https://www.kaggle.com/
5) https://towardsdatascience.com/
6) https://www.analyticsvidhya.com/