PROBLEM DEFINITION:

1.PROJECT OVERVIEW:

Online platform when used by normal people feel that they can express freely and without any reluctance.If they come across any type of malignant or toxic comment which can also be thread or insult or any type of harrassment which makes them uncomfortable ,so they differ to use the social media.Thus,it became essential for an organisatio or community to keep a watch on such comment and thus take respective action for it, such as blocking or reporting the same to prevent such mishap in the future.

This is the huge concern as there are 7.7 billion people in the world and out of which 3.5 billion people use some kind of social media platform. Thus this problem can be eliminated through the use of Natural Language Processing. In this we try to understand the intention of the speaker by building the model that's capable of detecting different types of toxicity like thread, insult, abuse and hate. Moreover it is crucial to handle such type of nuisance, to make more user-friendly experience, only after which people will enjoy discussion in online platform.

2.PROBLEM STATEMENT:

Given a number of comments, sentences or paragraph as a comment by the user, our task is to identify whether the comment is a malignant or not. Thus this problem comes under the category of multi-label-classification.

In multi-class-classification ,each instance is classified into one or mor classes, whereas in multi-label classification, multiple labels (such as malignant, highly malignant, toxic, threat, hate and the loathe) are to be predicted for same instance.

Adapted algorithm such as Random Forest classifier, Linear SVC, One VS Classifier and the Multi-Output classifier.

3.EVALUATION METRICS:

We have used accuracy score and the hamming loss. these are calculated each of the example and then averaged across the test set.

Accuarcy is defined as the proportion of correctly predicted label to the number of labels of each instance.

Hamming loss is defined as the symmetric difference between predicted and the true labels, divided by the total number of the labels.

2. ANALYSIS

A) Data Extraction:

One of the most- time consuming task in Data science is the collection of the data and the labelling of the data. What we noticed was that if we gave the track parameter(" ") that is the space or (\n) which is present in every comment which means many of the comment are vague or non-toxic.

B) Data Exploration.

The data consist of train and test data.

In train data we have following columns

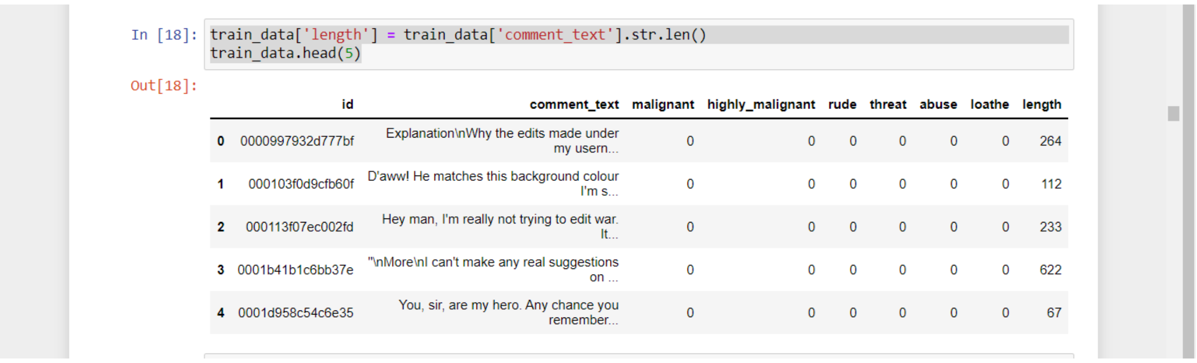
- **ID**
- **Comment_text: A multi-line text field which contain the unfiltered comment.**
- **Malignant: binary variable which contain 0 and 1(0 for no and 1 for yes)**
- **Highly_malignant: binary variable which contain 0/1**

- **Rude: binary variable which contain 0/1**
- **Threat: binary variable which contain 0/1**
- **Abuse: binary variable which contain 0/1**
- **Loathe: binary variable which contain 0/1**

Out of these field comment_text will be preprocessed and fitted into different classifier to predict whether it belongs to one or more label/output variable(malignant, highly malignant, rude, threat, abuse, loathe)

We have total 159571 rows × 8 columns that has been loaded from train.csv file.

C) Exploratory Visualization:

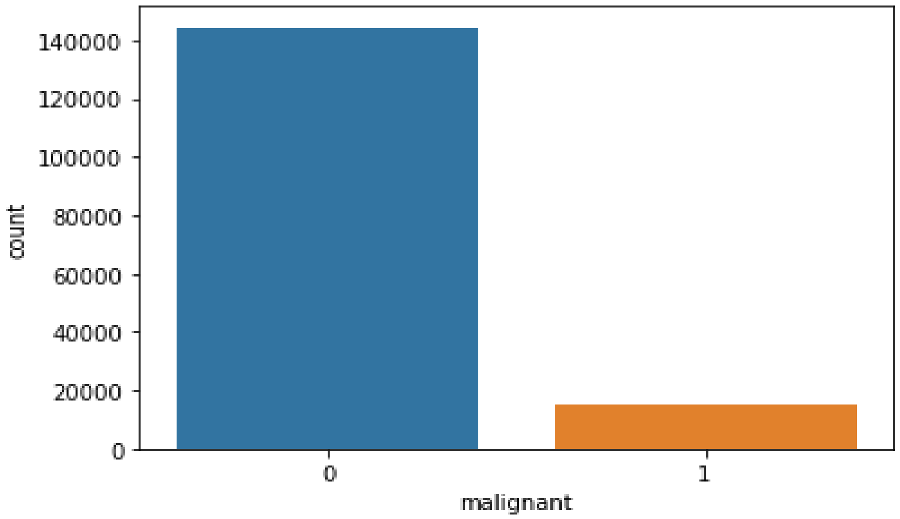


```
In [18]: train_data['length'] = train_data['comment_text'].str.len()
         train_data.head(5)
```

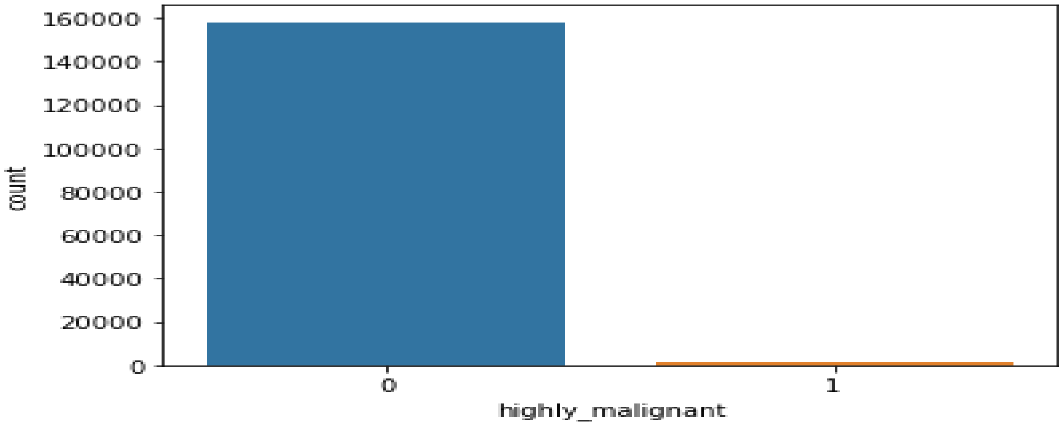
Out[18]:

	id	comment_text	malignant	highly_malignant	rude	threat	abuse	loathe	length
0	0000997932d777bf	Explanation\nWhy the edits made under my usern...	0	0	0	0	0	0	264
1	000103f0d9cfb60f	D'aww! He matches this background colour I'm s...	0	0	0	0	0	0	112
2	000113f07ec002fd	Hey man, I'm really not trying to edit war. It...	0	0	0	0	0	0	233
3	0001b41b1c6bb37e	"\nMore\nI can't make any real suggestions on ...	0	0	0	0	0	0	622
4	0001d958c54c6e35	You, sir, are my hero. Any chance you remember...	0	0	0	0	0	0	67

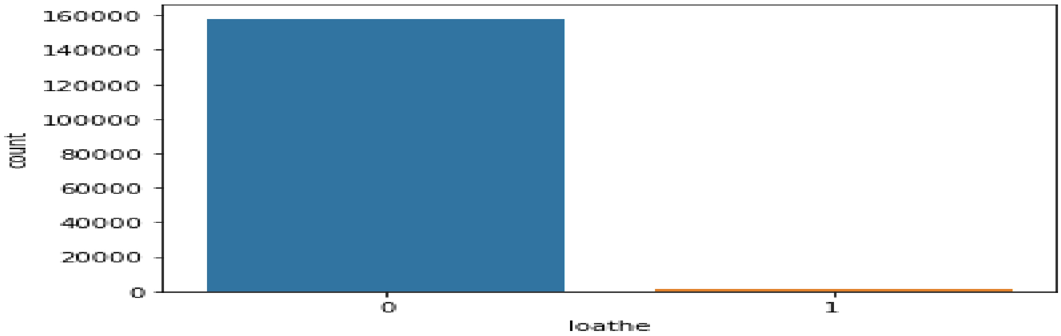
In the above figure we can see the length of the comment.



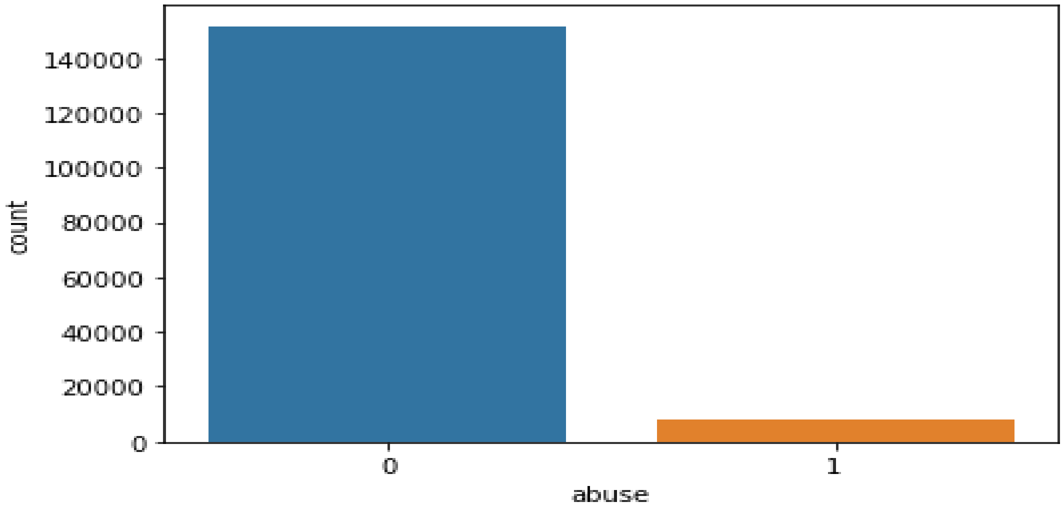
The malignant contains 0 as 144277 and 1 as the 15294.



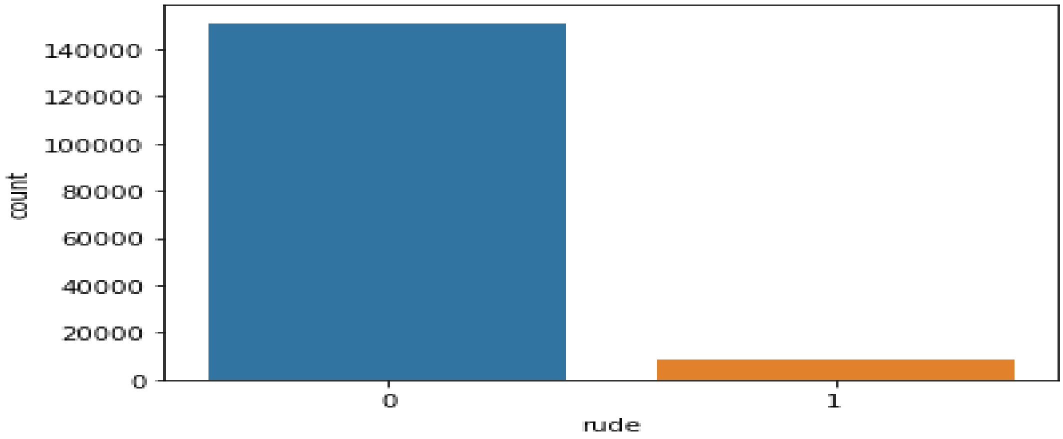
The highly malignant contain 0 for 157976 and 1 contain as 1595.



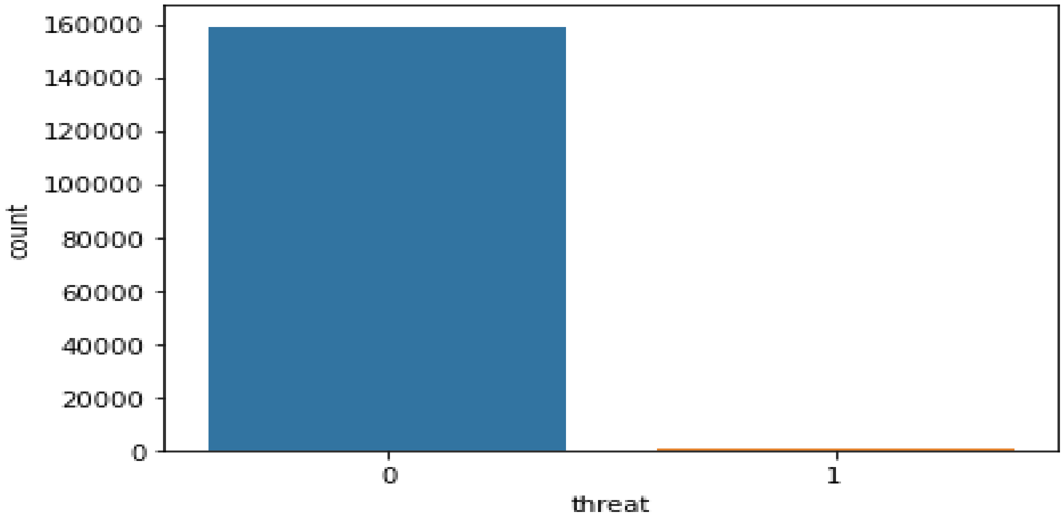
The loathe contain 0 label as the 158166 and 1 label as the 1405



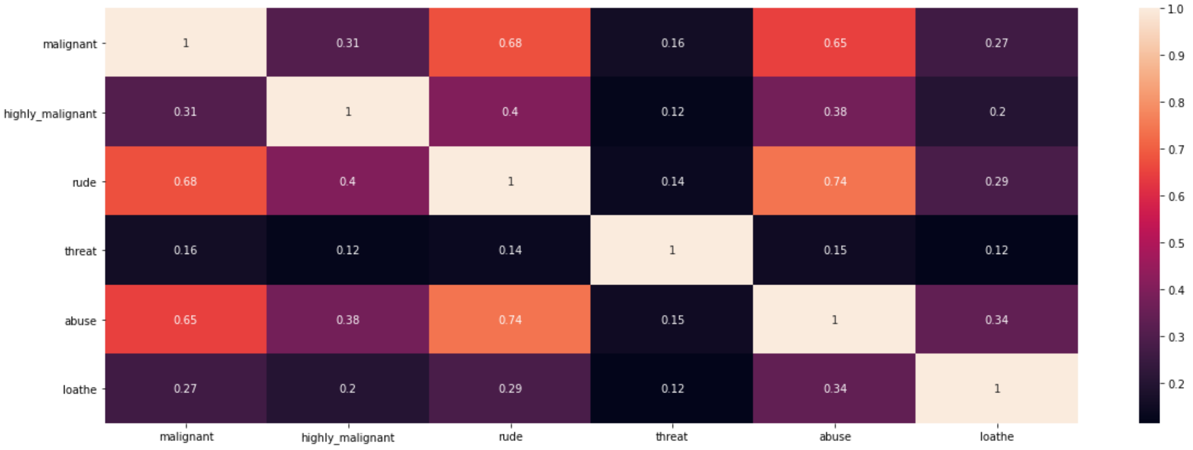
The abuse column contain 0 label as the 151694 and the 1 label contain 7877.



The rude column contains 0 label as the 151122 and 1 label as the 8449.



The threat contain 0 label as the 159093 and the 1 label as the 478.



In the above correlation figure we find that the highest correlation with each other is with the column malignant, rude and the abuse.

3.Methodology

Data Preprocessing:

- **Stop-Words:**

stopwords are those words that are frequently used in both written and verbal communication and thereby do not have any negative or positive impact on our statement such as “is,this,us,etc”. single letter word if excited or created due to any preprocessing step do not convey any meaningful statement, so they can be removed directly.

- **Stemming and Lemmatizing:**

  The process of converting the inflected/derived from the word stem or the root form is known as stemming. for example words like “stems”, “stemming”, “stemmer”, “stemmed” all are based on stem. Lemmatizing is the process of grouping together the inflected form of words so that they can be analysed as a single item. This is quite similar in stemming in its working but it differs since it depend upon correctly identifying the intended part of speech and meaning of the word in a sentence, as well as with the larger context surrounding that sentence, such as neighboring sentence or in the entire document.

  The “wordnet” library in the nltk will be used for these purpose. Stemmer and the lemmatizing are also imported from nltk library.

- **Applying count vectorizer:**

  To convert the string of words to a matrix of words with column header represented by words and their value signifying the frequency of occurrence of the word count vectorizer is used.

  Stopwords are accepted and converted to lowercase, and regular expression as its parameter. Here we will be supplying our custom list of stopwords created earlier and using lowercase option. Regular expression will have its default value.

- **Spitting dataset into training and testing:**

  We have assigned ¾ value as the train data and ¼ as the test data.

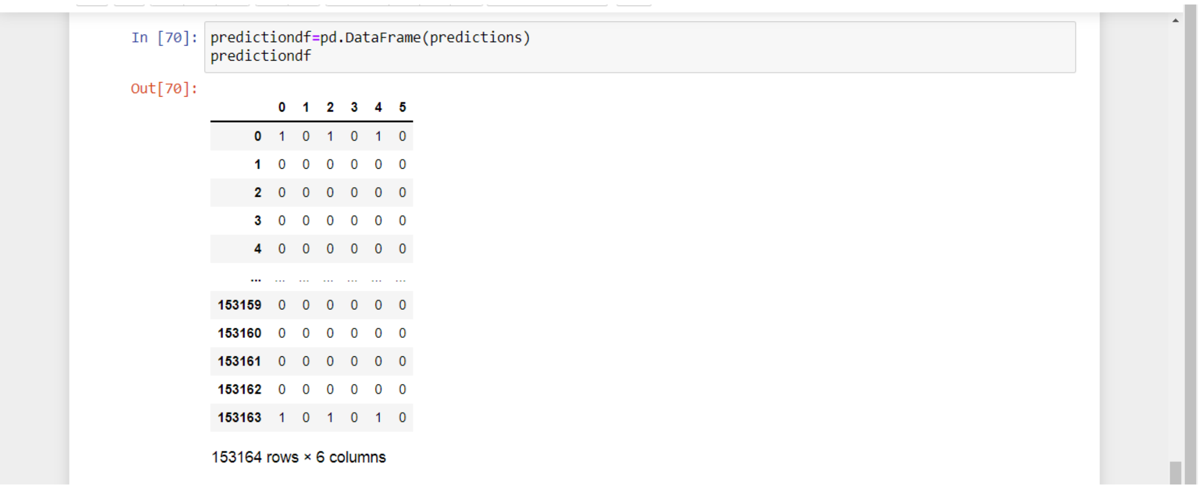
- **Implementation:**

  We will be using accuracy score and the hamming loss for predicting the train and test dataset.

  We will be using Random Forest classifier, Linear SVC, One Vs Rest classifier.

  We will take the test data and then we will clean the comment text and the predict which comment comes

under which category such as (Malignant, highly malignant, rude, threat, loathe and the abuse.)



```
In [70]: predictiondf=pd.DataFrame(predictions)
         predictiondf
```

```
Out[70]:
                0  1  2  3  4  5
             0  1  0  1  0  1  0
             1  0  0  0  0  0  0
             2  0  0  0  0  0  0
             3  0  0  0  0  0  0
             4  0  0  0  0  0  0
           ... .. .. .. .. .. ..
        153159  0  0  0  0  0  0
        153160  0  0  0  0  0  0
        153161  0  0  0  0  0  0
        153162  0  0  0  0  0  0
        153163  1  0  1  0  1  0

       153164 rows × 6 columns
```

The above figure shows the prediction of the test data which predicted the columns such as malignant, highly malignant, rude, threat, abuse, loathe.