MALIGNANT COMMENTS CLASSIFIER PROJECT

Submitted by:

CHAMLIN NAJIR RAHMAN
INTERNSHIP 25
BATCH NUMBER 1838

ACKNOWLEDGMENT

I would like to thank FlipRobo Technologies for providing me this opportunity and guidance throughout the project and all the steps that are implemented.

I have primarily referred to various articles scattered across various websites for the purpose of getting an idea on project “Malignant Comments Classification” and for assisting me in conducting extensive research that allowed me to learn a lot of new things, particularly in the Natural Language Processing and Natural Language Toolkit sections.

I would like to thank my project SME Mr. Mohd Kashif for providing the flexibility in time and also for giving us guidance in creating the project.

I have referred to various articles in Towards Data Science and Kaggle

The following are all of the external resources that were utilised to create this project:

1) https://www.google.com/

2) https://www.youtube.com/

3) https://scikit-learn.org/stable/user_guide.html

4) https://towardsdatascience.com/

5) https://www.analyticsvidhya.com/

INTRODUCTION

• Business Problem Framing

People may now express themselves broadly online because to the advent of social media. However, this has led in the growth of violence and hatred, making online environments unappealing to users. Despite the fact that academics have discovered that hatred is an issue across numerous platforms, there are no models for detecting online hate.

Online hatred has been highlighted as a big problem on online social media platforms, and has been defined as abusive language, hostility, cyberbullying, hatefulness, and many other things. The most common venues for such toxic behaviour are social media platforms.

On numerous social media sites, there has been a significant increase in incidences of cyberbullying and trolls. Many celebrities and influencers face blowback from the public and are subjected to nasty and disrespectful remarks. This may have a negative impact on anyone, resulting in sadness, mental disease, self-hatred, and suicide thoughts.

Comments on the internet are hotbeds of hate and venom. Machine learning may be used to combat online anonymity, which has created a new venue for hostility and hate speech. The issue we were attempting to address was the labelling of internet remarks that were hostile to other users. This implies that insults directed towards third parties, such as celebrities, will be classified as non-offensive, whereas "u are an idiot" will be plainly offensive.

Our objective is to create a prototype of an online hate and abuse comment classifier that can be used to categorize and manage hate and offensive remarks in order to prevent the spread of hatred and cyber bullying.

• Conceptual Background of the Domain Problem

Online platforms and social media have evolved into places where individuals may freely discuss their beliefs without regard for race, and where people can share their thoughts and ideas with a large group of people.

Through the creation of virtual networks and communities, social media is a computer-based technology that allows the exchange of ideas, opinions, and information. Social media is Internet-based by design, allowing people to share material quickly via electronic means. Personal information, documents, movies, and images are all included in the content. Users interact with social media using web-based software or applications on a computer, tablet, or smartphone.

While social media is widely used in the United States and Europe, Asian nations such as India are at the top of the list. Social media is used by about 3.8 billion people.

Some people or a motivated mob on this massive internet platform or online community wilfully abuse others to prevent them from sharing their thoughts in a proper manner. They use filthy language to intimidate others, which is considered a form of ignominy in civilised society. When innocent people are intimidated by these mobs, they remain mute without saying anything. As a result, the disgusting mob's goal is excellently realised.

To address this issue, we are now developing a model that recognises all foul language and foul terms, with the goal of preventing these mobs from using foul language in online communities or perhaps blocking them from using foul language altogether.

- # 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 tested nine different classification algorithms and selected the best based on performance indicators. I then selected one approach and built a model in that algorithm.

Comments on the internet are hotbeds of hate and venom. Machine learning may be used to combat online anonymity, which has created a new venue for hostility and hate speech. The issue we were attempting to address was the labelling of internet remarks that were hostile to other users.

Our objective is to create a prototype of an online hate and abuse comment classifier that can be used to categorise and manage hate and offensive remarks in order to prevent the spread of hatred and cyberbullying.

- # Motivation for the Problem Undertaken

One of the first lessons we learn as children is that the louder you scream and the bigger of a tantrum you throw, you more you get your way. Part of growing up and maturing into an adult and functioning member of society is learning how to use language and reasoning skills to communicate our beliefs and respectfully disagree with others, using evidence and persuasiveness to try and bring them over to our way of thinking.

Social media is reverting us back to those animalistic tantrums, schoolyard taunts and unfettered bullying that define youth, creating a dystopia where even renowned academics and dispassionate journalists transform from Dr. Jekyll into raving Mr. Hydes, raising the critical question of whether social media should simply enact a blanket ban on profanity and name calling? Actually, ban should be implemented on these profanities and taking that as a motivation I have started this project to identify the malignant comments in social media or in online public forums.

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 pointwise mutual information technique. Based on these features, we

developed a supervised machine learning solution for cyber bullying 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 cyber bullying behavior and its severity in online social networks.

Analytical Problem Framing

• Mathematical/ Analytical Modeling of the Problem

The libraries/dependencies imported for this project are shown below:

```
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
```

```
# Importing all the required library

from collections import Counter
from string import digits as d, punctuation as p
from nltk.tokenize import word_tokenize as wt
from nltk.stem import WordNetLemmatizer as wl, PorterStemmer as porter
from gensim import corpora

from sklearn.feature_extraction.text import TfidfVectorizer as tf
from sklearn.model_selection import train_test_split as tts, RandomizedSearchCV as rsv, cross_val_score as cvs
from sklearn.metrics import accuracy_score,classification_report,f1_score,auc,roc_curve,roc_auc_score,confusion_matrix,log_loss,\
precision_score,recall_score,mean_squared_error

from sklearn.linear_model import LogisticRegression,PassiveAggressiveClassifier
from sklearn.naive_bayes import MultinomialNB, ComplementNB
from sklearn.svm import LinearSVC
from sklearn.pipeline import Pipeline
from sklearn.multiclass import OneVsRestClassifier
from sklearn.feature_extraction.text import TfidfVectorizer

from PIL import Image
import requests
from wordcloud import WordCloud
```

```
from sklearn import tree
```

```
#importing necessary libraries.
from sklearn.tree import DecisionTreeClassifier
from sklearn.ensemble import RandomForestClassifier
from sklearn.svm import SVC
from sklearn.naive_bayes import BernoulliNB
from sklearn.ensemble import ExtraTreesClassifier
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import classification_report
from sklearn.ensemble import GradientBoostingClassifier, AdaBoostClassifier, BaggingClassifier
from sklearn.neighbors import KNeighborsClassifier
from sklearn.naive_bayes import GaussianNB as NB
from sklearn.metrics import classification_report, confusion_matrix, roc_curve, roc_auc_score, accuracy_score
from sklearn.model_selection import cross_val_score
from sklearn.metrics import hamming_loss
```

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

I will need to build multiple classification machine learning models. Before model building 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. 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 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.

Variable	Definition
id	A unique id aligned with each comment text.
comment_text	It includes the comment text.
malignant	It is a column with binary values depicting which comments are malignant in nature.
highly_malignant	Binary column with labels for highly malignant text.
rude	Binary column with labels for comments that are rude in nature.
threat	Binary column with labels for threatening context in the comments.
abuse	Binary column with labels with abusive behaviour.
loathe	Label to comments that are full of loathe and hatred.

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. You need to build a model that can differentiate between comments and its categories.

• Data Preprocessing Done

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

1. Load dataset
2. Remove 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

I have analysed the input output logic with word cloud and I have word clouded the sentenced that as 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.

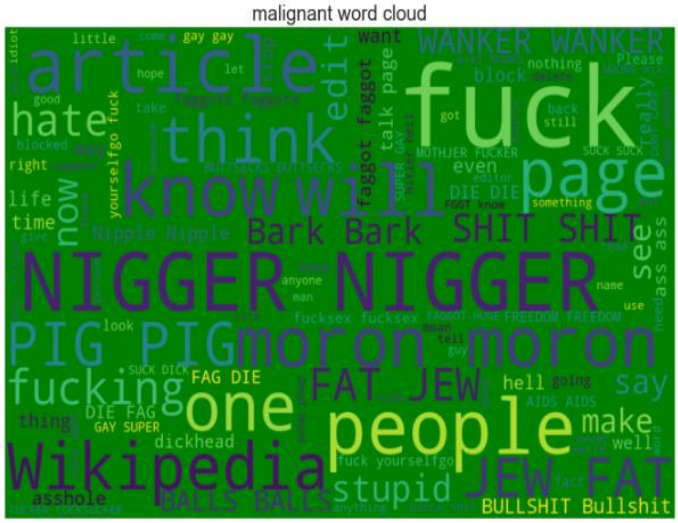
Code:

```
# Word cloud for the labels

for i in cat_features.columns:
    plt.figure(figsize=(12,8))
    label_words = " ".join(train[train[i] == 1]["comment_text"])

    word_cloud = WordCloud(width = 1200,
                           height = 800,
                           max_words = 400,
                           min_word_length = 3,
                           max_font_size = 180, min_font_size = 20,
                           background_color="green").generate(label_words)

    plt.title("{} word cloud".format(i), fontsize = 18)
    plt.imshow(word_cloud, interpolation="bilinear")
    plt.axis("off")
    plt.show()
```

Output:

highly_malignant word cloud

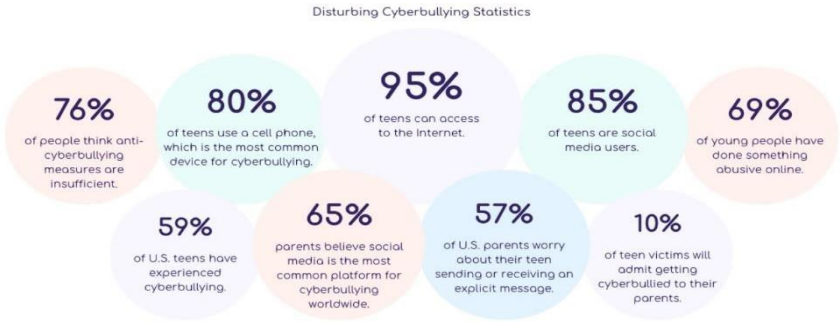
malignant word cloud



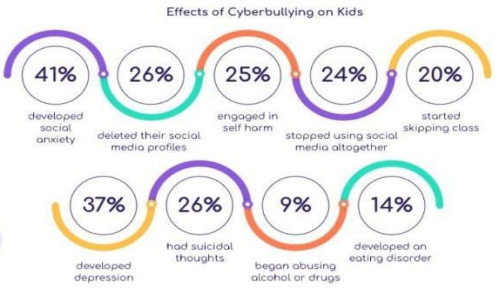
These are the comments that belongs to different type so which the help of word cloud we can see if there is abuse comment which type of words it contains and similar to other comments as well.

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

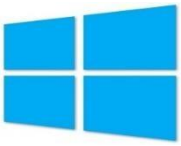
Cyberbullying has become a growing problem in countries around the world. Essentially, cyberbullying doesn't differ much from the type of bullying that many children have unfortunately grown accustomed to in school. Now-a-days the only difference is that it takes place online.



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 technology being used.

RAM : 8 GB

CPU : Intel(R) Core(TM) i3-7100U CPU @ 2.40GHz 2.40 GHz

Software technology being used.

Programming language : Python

Distribution : Anaconda Navigator

Browser based language shell : Jupyter Notebook/Googlecolab

Libraries/Packages specifically being used.

Pandas, NumPy, matplotlib, seaborn, scikit-learn, pandas-profiling, missingno, NLTK

- **Python**: Python is a general-purpose, and high-level programming language which is best known for its efficiency and powerful functions. Its ease to use, which makes it more accessible. Python provides data scientists with an extensive amount of tools and packages to build machine learning models. One of its special features is that we can build various machine learning with less-code.
- **Matplotlib** is a plotting library for the Python programming language and its numerical mathematics extension NumPy.
- **Seaborn** is a library for making statistical graphics in Python. It builds on top of matplotlib and integrates closely with pandas data structures. Seaborn helps you explore and understand
- **NumPy** is a general-purpose array-processing package. it provides a high-performance multidimensional array object and tools for working with these arrays. It is the fundamental package for scientific computing with Python.
- **Scikit-learn** provides a range of supervised and unsupervised learning algorithms via a consistent interface in Python. It is licensed under a permissive simplified BSD license and is distributed under many Linux distributions, encouraging academic and commercial use.
- **NLTK** The Natural Language Toolkit, or more commonly NLTK, is a suite of libraries and programs for symbolic and statistical natural language processing for English written in the Python programming language.
  NLTK is a standard python library that provides a set of diverse algorithms for NLP. It is one of the most used libraries for NLP and Computational Linguistics

Model/s Development and Evaluation

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

I 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:

```
# Checking null values for train dataset

train.isnull().sum()
```

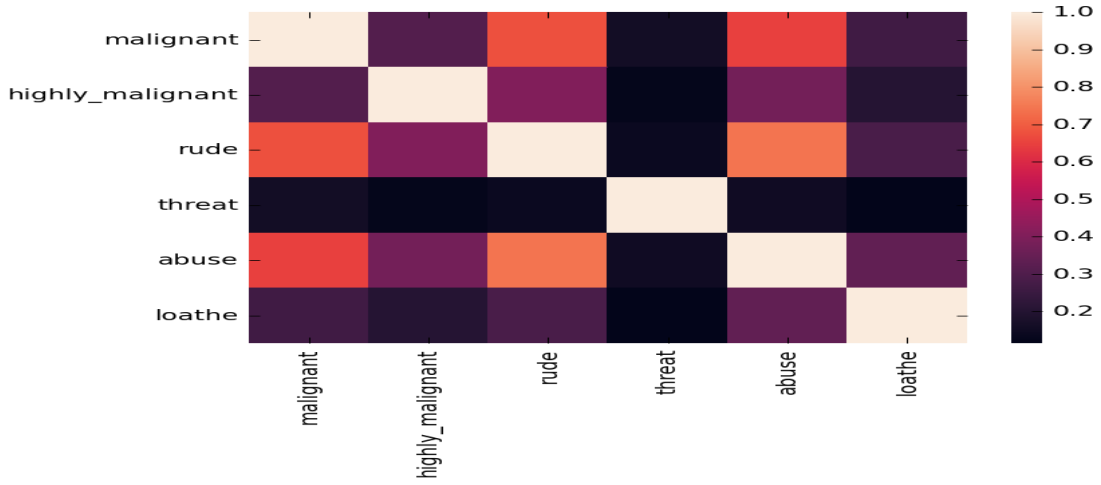
```
id                0
comment_text      0
malignant         0
highly_malignant  0
rude              0
threat            0
abuse             0
loathe            0
dtype: int64
```

Here we can see that there are no null values in train dataset.

```
# Checking the null values for test dataset

test.isnull().sum()
```

```
id              0
comment_text    0
dtype: int64
```

Visual Representation:



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:

```
# Checking the datatype of train dataset

train.dtypes
```

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

Then we went ahead and performed multiple data cleaning and data transformation steps. I have added an additional column to store the original length of our comment_text column.
Since there was no use of the "id" column I have dropped it and converted all the text data in our comment text column into lowercase format for easier interpretation.

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).

- # Testing of Identified Approaches (Algorithms)

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

1) Gaussian Naïve Bayes
2) Multinomial Naïve Bayes
3) Logistic Regression
4) Random Forest Classifier
5) Linear Support Vector Classifier
6) Ada Boost Classifier
7) K Nearest Neighbors Classifier
8) Decision Tree Classifier
9) Bagging Classifier

- # Run and Evaluate selected models

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

Train and test split:

```
from sklearn.model_selection import train_test_split
```

```
# Splitting the training and testing data

X_train, X_test, y_train, y_test = train_test_split(X_features, y, test_size=0.20, random_state=42)
```

```
conda install -c conda-forge scikit-learn
```

Models:

```
# Creating the instances for the algorithms.

lg = LogisticRegression()
mnb = MultinomialNB()
dt = DecisionTreeClassifier()
rf = RandomForestClassifier()
knn = KNeighborsClassifier()
svc = SVC()
ab = AdaBoostClassifier()
gb = GradientBoostingClassifier()
```

Observation:

Best Model

```
---------- RandomForestClassifier ----------

Training Accuracy : 0.9972112552484803

Test Accuracy : 0.9163089456368478

Hamming_loss : 1.9061047574285863

Log_loss : 1.5837183417939957

Classification Report :
```

	precision	recall	f1-score	support
0	0.87	0.67	0.76	3056
1	0.43	0.06	0.10	321
2	0.86	0.72	0.79	1715
3	0.40	0.05	0.10	74
4	0.76	0.57	0.65	1614
5	0.77	0.13	0.22	294
micro avg	0.83	0.60	0.70	7074
macro avg	0.68	0.37	0.43	7074
weighted avg	0.81	0.60	0.68	7074
samples avg	0.06	0.05	0.05	7074

Predictions :

Prediction values

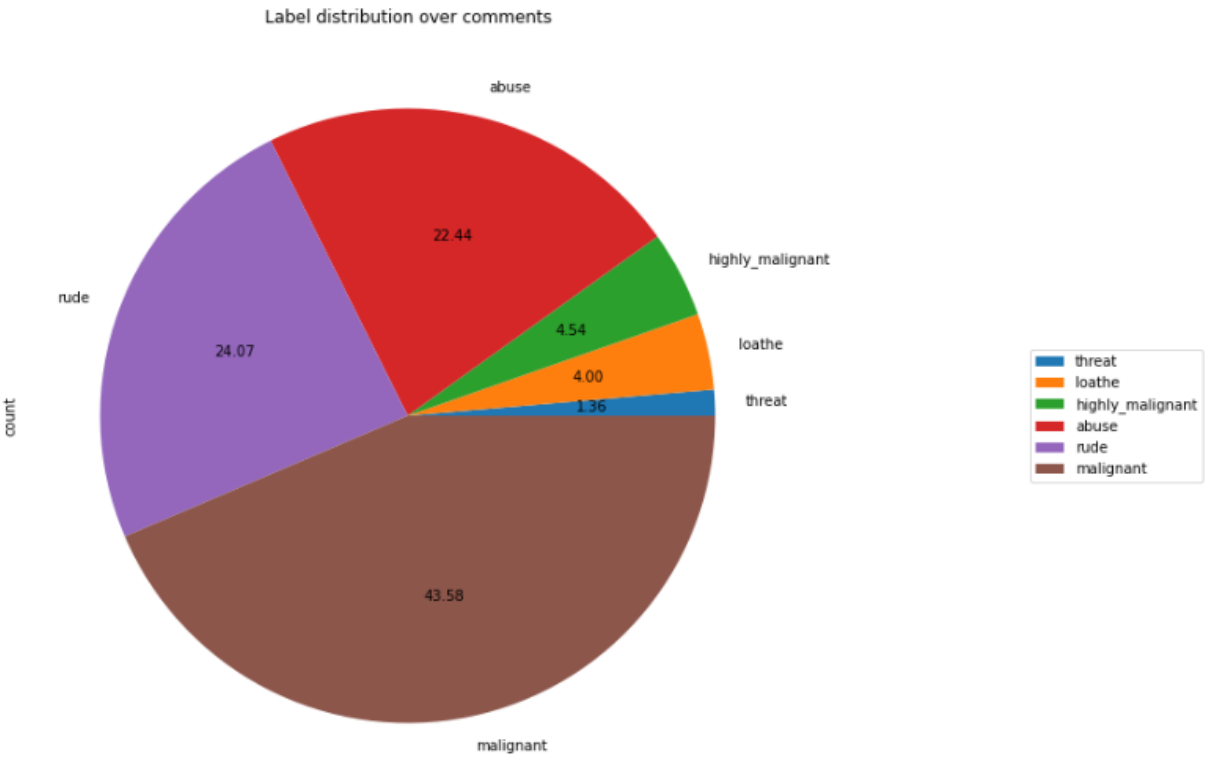
```
# Predicting the values

classifier = OneVsRestClassifier(svc)
classifier.fit(X_features, y)
predict = classifier.predict(test_X_features)
```

```
# Printing the predicted values.

predict
```

```
array([[0, 0, 0, 0, 0, 0],
       [0, 0, 0, 0, 0, 0],
       [0, 0, 0, 0, 0, 0],
       ...,
       [0, 0, 0, 0, 0, 0],
       [1, 0, 1, 0, 0, 0],
       [0, 0, 0, 0, 0, 0]])
```

- # Visualizations

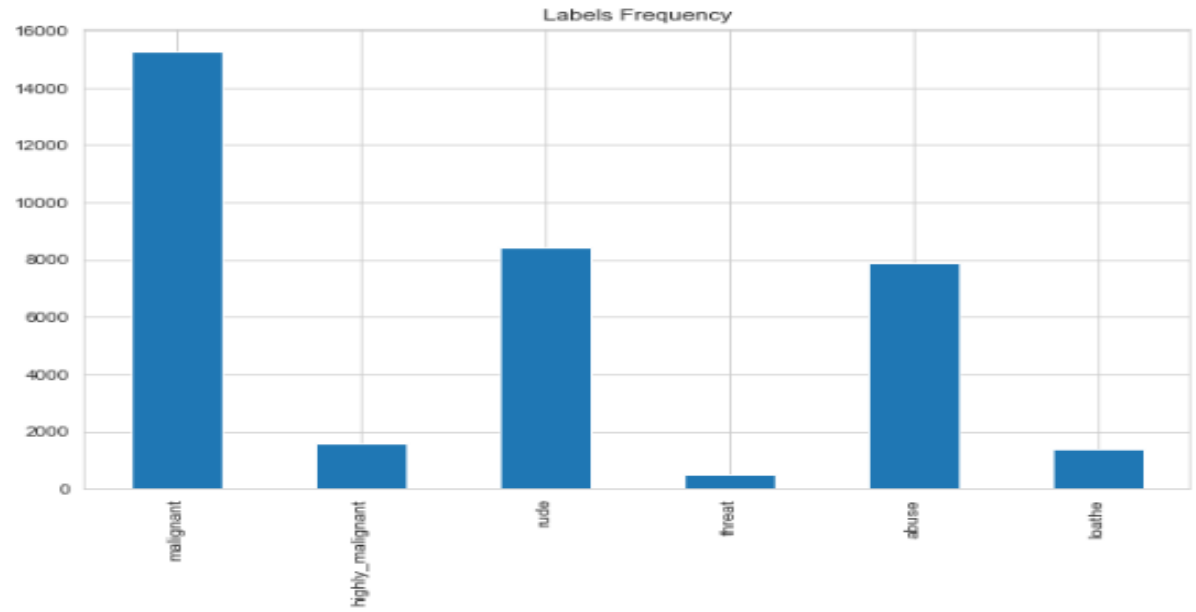
  I used the pandas profiling feature to generate an initial detailed report on my data-frame values. It gives us various information on the rendered dataset like the correlations, missing values, duplicate rows, variable types, memory size etc. This assists us in further detailed visualization separating each part one by one comparing and research for the impacts on the prediction of our target label from all the available feature columns.



Barplot :

```
# Plotting the bar plot that show the total comment counts for different labels.

sns.set_style('whitegrid')
plt.figure(figsize=(10,6))
train[comments_labels].sum(axis=0).plot.bar(title='Labels Frequency')
```

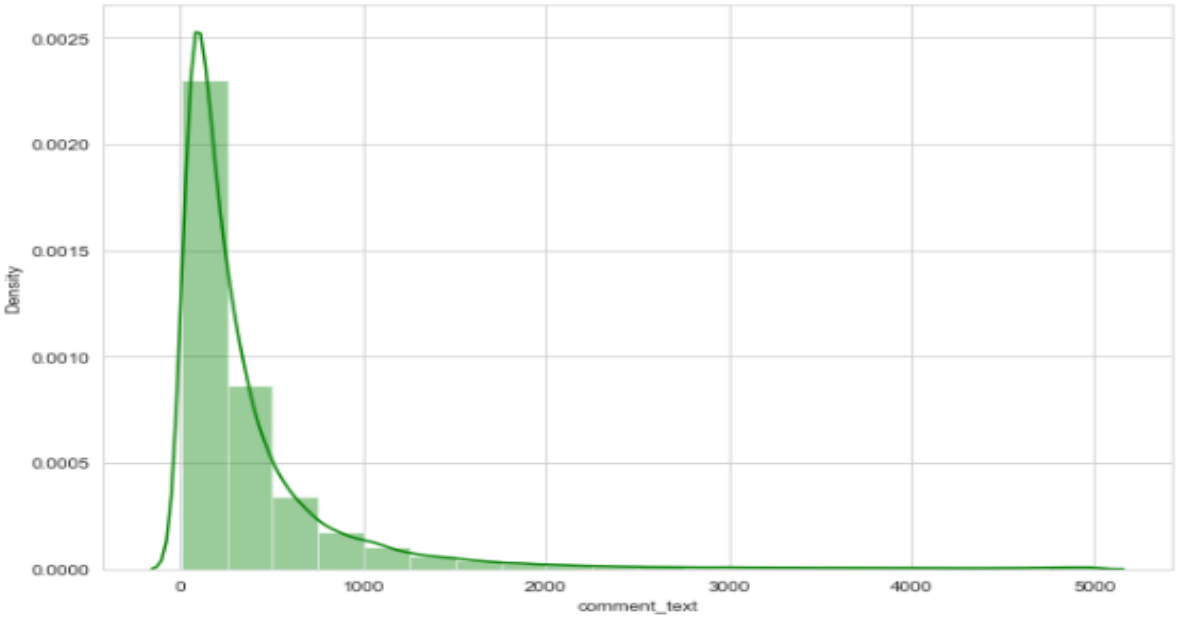
```
<AxesSubplot:title={'center':'Labels Frequency'}>
```



Distribution Plot :

```
# Distribution of comments length.

plt.figure(figsize=(10,7))
comment_len = train.comment_text.str.len()
sns.distplot(comment_len, bins=20, color = 'green')
```

```
<AxesSubplot:xlabel='comment_text', ylabel='Density'>
```



- # 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 unparliamentary 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 overthese 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 my 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
- Good parameters could not be obtained using hyper-parameter tuning

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 error