

Malignant

Comment

Classifier

**Submitted by:**

Ruchi Tiwari

**Contents**

[Acknowledgment 2](#_Toc65767591)

[Introduction 3](#_Toc65767592)

[Business Problem Framing 4](#_Toc65767593)

[Conceptual Background of the Domain Problem 4](#_Toc65767594)

[Motivation for the Problem Undertaken 4](#_Toc65767595)

[Analytical Problem Framing 4](#_Toc65767596)

[Mathematical/ Analytical Modeling of the Problem 5](#_Toc65767597)

[Data Sources and their formats 5](#_Toc65767598)

[Data Pre-processing Done 6](#_Toc65767599)

[Hardware and Software Requirements and Tools Used 7](#_Toc65767600)

[Model/s Development and Evaluation 8](#_Toc65767601)

[Testing of Identified Approaches (Algorithms) 8](#_Toc65767602)

[Run and Evaluate selected models 9](#_Toc65767603)

[Key Metrics for success in solving problem under consideration 12](#_Toc65767604)

[Visualizations 13](#_Toc65767605)

[Conclusion 20](#_Toc65767606)

[Key Findings and Conclusions of the Study 20](#_Toc65767607)

[Learning Outcomes of the Study in respect of Data Science 20](#_Toc65767608)

# Acknowledgment

Following are the external references which I used:

[www.w3school.com](http://www.w3school.com)

[www.stackoverflow.com](http://www.stackoverflow.com)

[www.google.com](http://www.google.com)

[www.geeksforgeeks.org](http://www.geeksforgeeks.org)

https://www.kaggle.com

# 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 inoffensive, but “u are an idiot” is clearly offensive.

## Conceptual Background of the Domain Problem

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

As this project is based on NLP and there are so many new libraries which are new for me so it is really motivates me to learn and understand the concept and do the coding on some new things which are totally new for me.

# Analytical Problem Framing

## Mathematical/ Analytical Modeling of the Problem

The statistical figure I get to know by the train.info, test.info, train. Describe (), test.describe(). The so many information the min max standard deviation the 25th percentile the 50th percentile the 75 percentile .Then by the help of correlation function I get to know the correlation of each columns with each other. From the heat map I can visualized to see them clearly that they are positive correlated or the negative correlated the dark side is show the negative correlation among each other the lighter side represent the positive correlation among the each other**. Check the skewness in the data.**

## 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.

## Data Pre-processing Done

There were no null value was present in the dataset. Here is a checklist to use the clean data Remove all irrelevant characters such as any non alphanumeric characters

1. Tokenize your text by separating it into individual words
2. Remove words that are not relevant,
3. Convert all characters to lowercase, in order to treat words such as “hello”, “Hello”, and “HELLO” the same
4. Consider combining misspelled or alternately spelled words to a single representation (e.g. “cool”/”kewl”/”cooool”)
5. Consider lemmatization (reduce words such as “am”, “are”, and “is” to a common form such as “be”)

Firstly I read the data and see the first five rows and the last five rows .Than in the dataset we don’t have null values for the preprocessing /cleaning the data I perform the stop words removal, Tokenization, Lemmatization .The Stop words are those common words that appear in a text many times and do not contribute to machine’s understanding of the text. We don’t want these words to appear in our data. So, we remove these words. Tokenization: Word tokenization is the process of splitting a large sample of text into words. Lemmatization: Lemmatization is the process of grouping together the different inflected forms of same root word so they can be analyzed as a single item. Then I convert all the text words in the lower case since python is a case sensitive language .Then I used regex for cleaning the sentence. Then I print the loud words of features news ,In order to use textual data for predictive modelling, the text must be parsed to remove certain words — this process is called tokenization. These words need to then be encoded as integers, or floating-point values, for use as inputs in machine learning algorithms. This process is called feature extraction (or vectorization).

Convert all messages to lower case. Replace email addresses with 'email'. Replace URLs with 'webaddress'. Replace money symbols with 'moneysymb' (£ can by typed with ALT key + 156). Replace 10 digit phone numbers (formats include paranthesis, spaces, no spaces, dashes) with 'phonenumber'. Replace numbers with 'numbr'. Total length removal. Original Length 62893130

Clean Length 43575187

TF-IDF Vectorizer TF-IDF stands for Term Frequency — Inverse Document Frequency. It is one of the most important techniques used for information retrieval to represent how important a specific word or phrase is to a given document. These are the things which I performed in the Preprocessing.

## Hardware and Software Requirements and Tools Used

* **Hardware** – Laptop
* **Software** - anaconda jupyter notebook
* **Libraries**

import pandas as pd

import numpy as np

import seaborn as sns

import matplotlib.pyplot as plt

#Pre-processing/ Cleaning the Data

#For preprocessing the data, we will need some libraries.

import nltk

import string

from nltk.stem import WordNetLemmatizer

from nltk.corpus import stopwords

from wordcloud import WordCloud

from sklearn.feature\_extraction.text import TfidfTransformer

#Importing Library to apply algorithms

from sklearn.naive\_bayes import MultinomialNB

from sklearn.model\_selection import train\_test\_split

from sklearn.metrics import accuracy\_score, confusion\_matrix, classification\_report,roc\_curve,roc\_auc\_score,auc

from sklearn.model\_selection import train\_test\_split

from sklearn.metrics import accuracy\_score,classification\_report,confusion\_matrix,f1\_score

from sklearn.linear\_model import LogisticRegression

from sklearn.model\_selection import cross\_val\_score,GridSearchCV

from sklearn.naive\_bayes import MultinomialNB

from sklearn.tree import DecisionTreeClassifier

from sklearn.neighbors import KNeighborsClassifier

from sklearn.ensemble import RandomForestClassifier,AdaBoostClassifier,GradientBoostingClassifier

from sklearn.naive\_bayes import GaussianNB

from sklearn.linear\_model import LogisticRegression

from sklearn.svm import SVC

from sklearn.tree import DecisionTreeClassifier

* **from sklearn.model\_selection import train\_test\_split,cross\_val\_score**

Train\_test\_split is a function in Sklearn model selection for splitting data arrays into two subsets: for training data and for testing data. With this function, you don't need to divide the dataset manually. By default, Sklearn train\_test\_split will make random partitions for the two subsets.

The algorithm is trained and tested K times, each time a new set is used as testing set while remaining sets are used for training. Finally, the result of the K-Fold Cross-Validation is the average of the results obtained on each set.

* **from sklearn.neighbors import KNeighborsClassifier**

K Nearest Neighbor(KNN) is a very simple, easy to understand, versatile and one of the topmost machine learning algorithms. KNN used in the variety of applications such as finance, healthcare, political science, handwriting detection, image recognition and video recognition

* **from sklearn.linear\_model import LogisticRegression**

The library sklearn can be used to perform logistic regression in a few lines as shown using the LogisticRegression class. It also supports multiple features. It requires the input values to be in a specific format hence they have been reshaped before training using the fit method.

* **from sklearn.tree import DecisionTreeClassifier**

Decision Tree is a white box type of ML algorithm. It shares internal decision-making logic, which is not available in the black box type of algorithms such as Neural Network. Its training time is faster compared to the neural network algorithm. The time complexity of decision trees is a function of the number of records and number of attributes in the given data. The decision tree is a distribution-free or non-parametric method, which does not depend upon probability distribution assumptions. Decision trees can handle high dimensional data with good accuracy

# Model/s Development and Evaluation

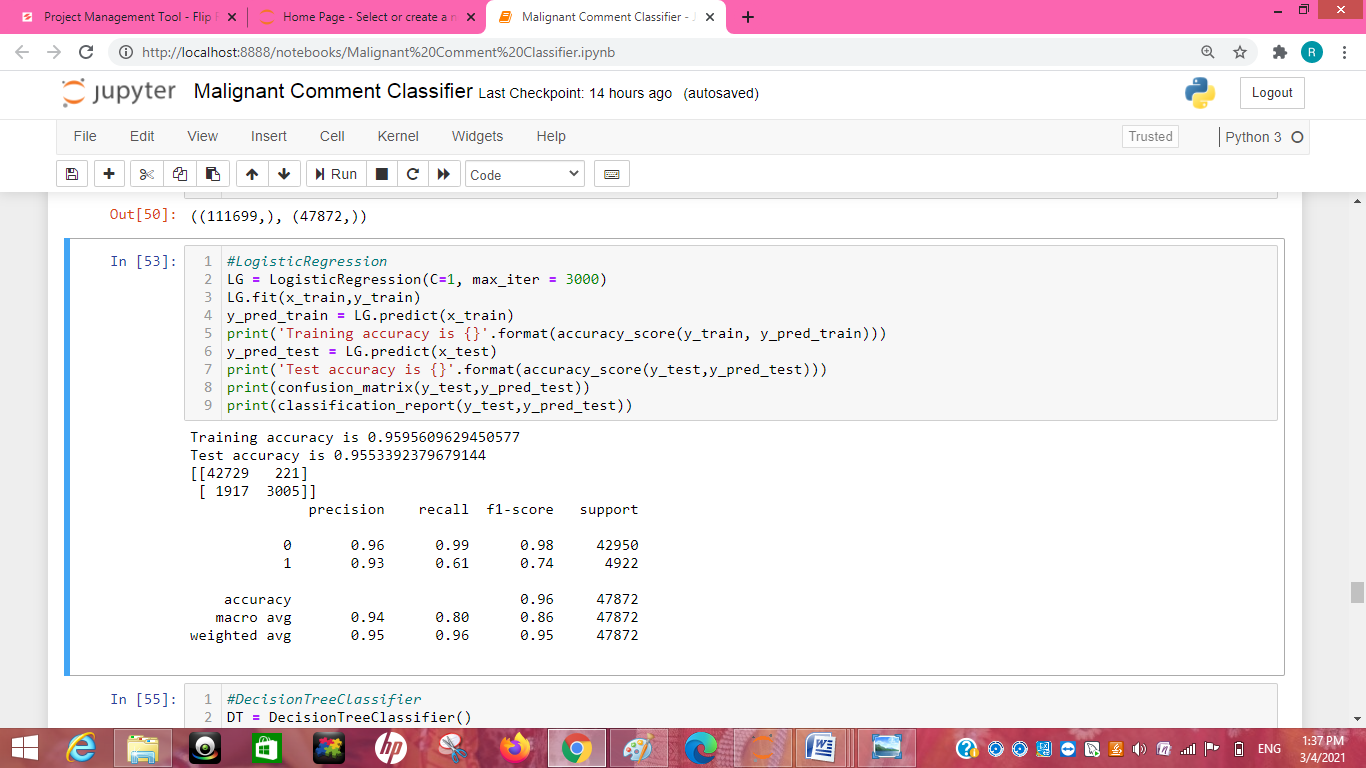
## Testing of Identified Approaches (Algorithms)

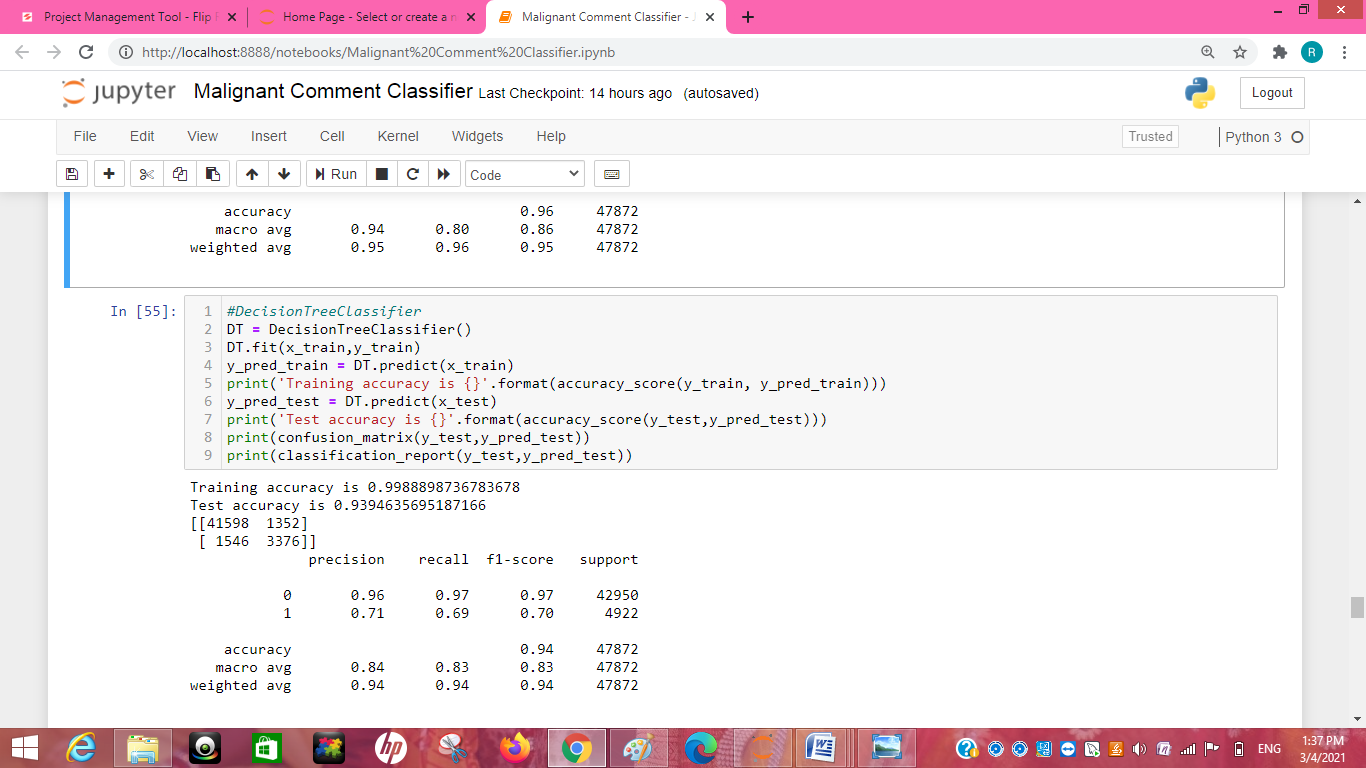
Listing down all the algorithms used for the training and testing.

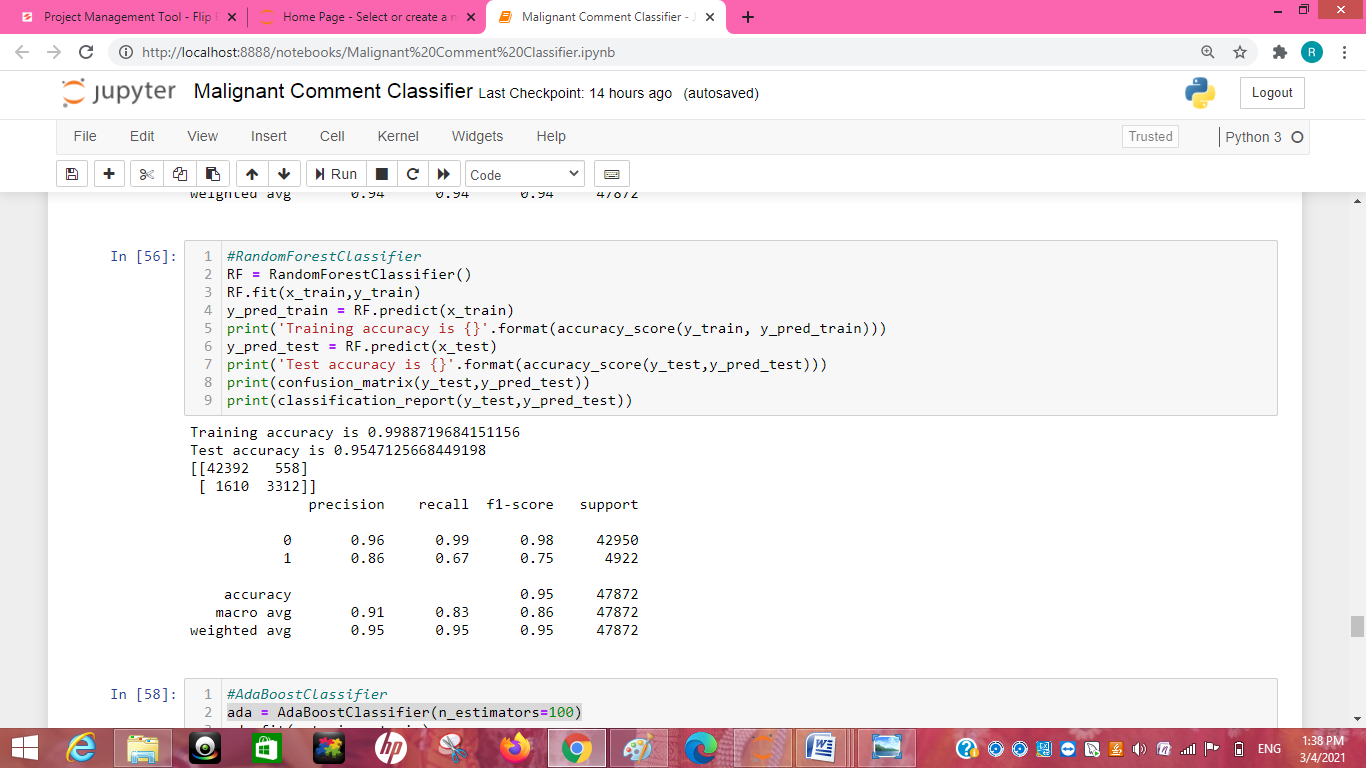
* KNN=KNeighborsClassifier
* LR=LogisticRegression()
* DT=DecisionTreeClassifier
* RF=RandomForestClassifier()
* ada = AdaBoostClassifier(n\_estimators=100)

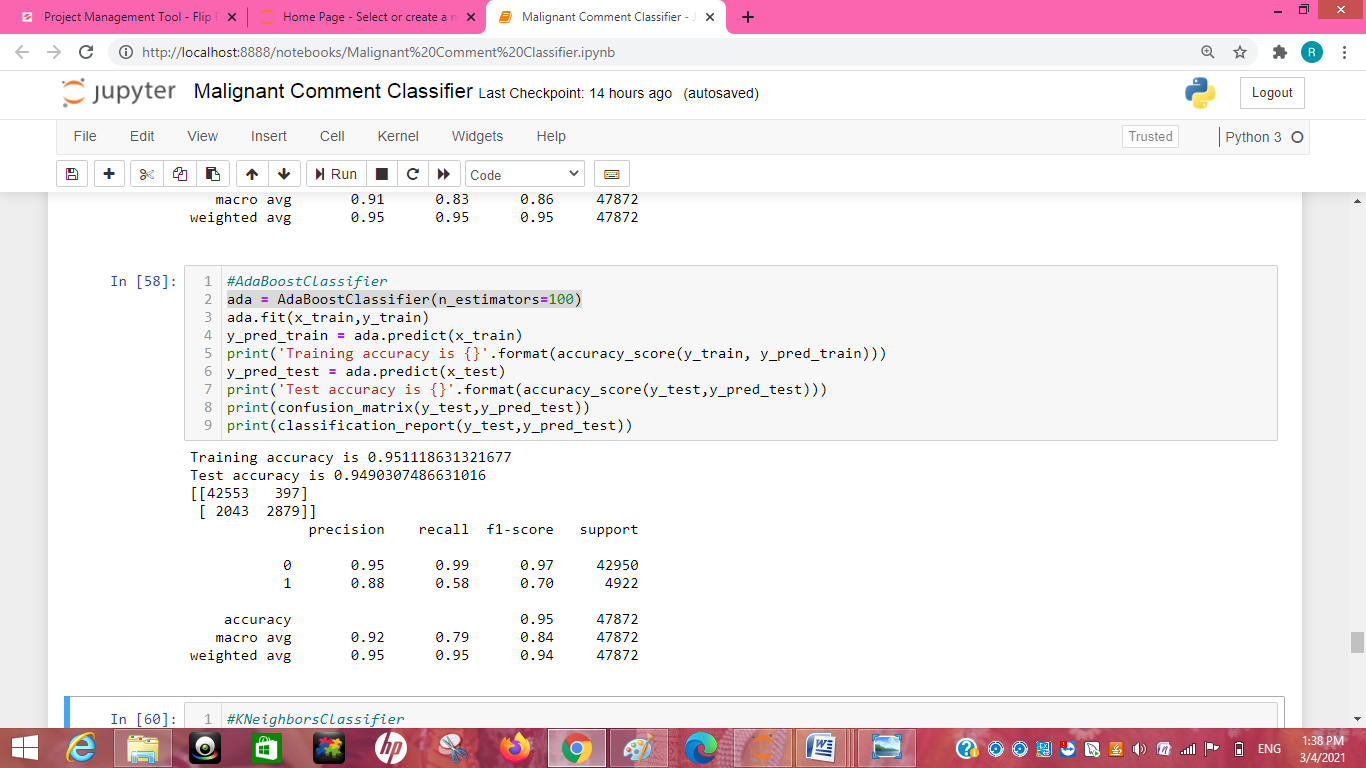
I applied all these algorithms in the dataset.

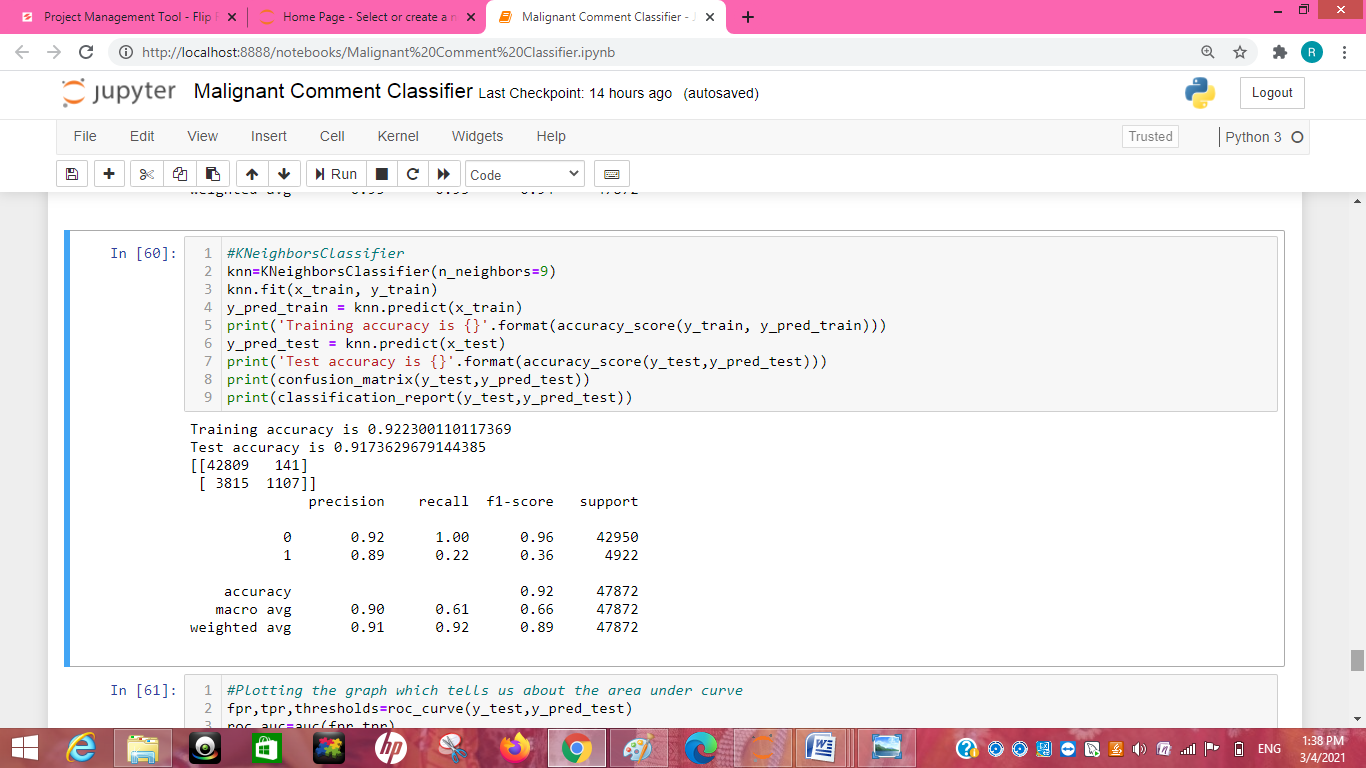
## Run and Evaluate selected models

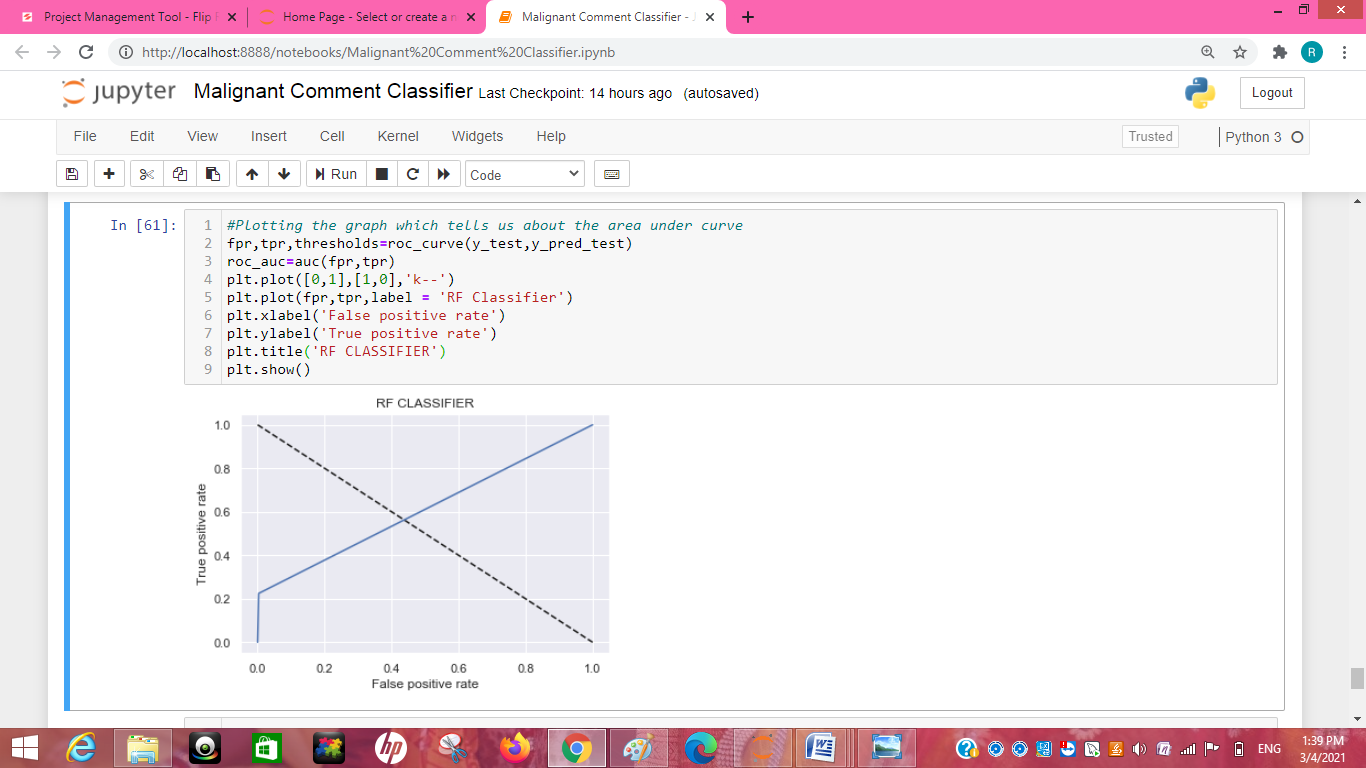








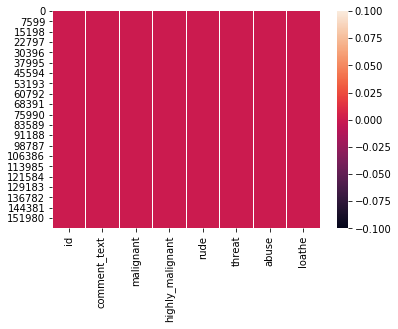




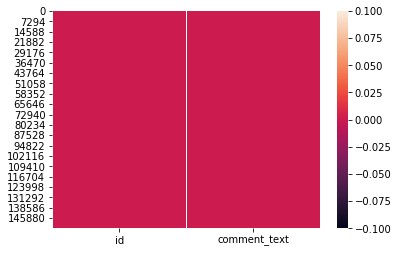
## Key Metrics for success in solving problem under consideration

* Precision: can be seen as a measure of quality, **higher** **precision** means that an algorithm returns more relevant results than irrelevant ones
* **Recall** is used as a measure of quantity and high recall means that an algorithm returns most of the relevant results.
* **Accuracy score** is used when the True Positives and True negatives are more important. **Accuracy** can be used when the class distribution is similar
* **F1**-**score** is used when the False Negatives and False Positives are crucial. While F1-score is a better metric when there are imbalanced classes.
* **Cross\_val\_score** :- To run **cross**-**validation** on multiple metrics and also to return train **scores**, fit times and **score** times. Get predictions from each split of **cross**-**validation** for diagnostic purposes. Make a scorer from a performance metric or loss function.
* roc \_auc \_score :-  **ROC curve**. It is a plot of the false positive rate (x-axis) versus the true positive rate (y-axis) for a number of different candidate threshold values between 0.0 and 1.0

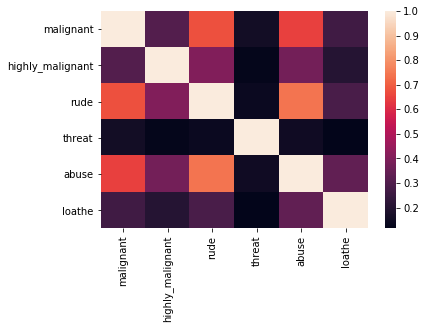
## Visualizations



sns.heatmap(train.isnull())



sns.heatmap(test.isnull())



sns.heatmap(train.corr())

#Visualizing the data with the help of for loop

col=['malignant', 'highly\_malignant', 'rude', 'threat','abuse', 'loathe']

for i in col:

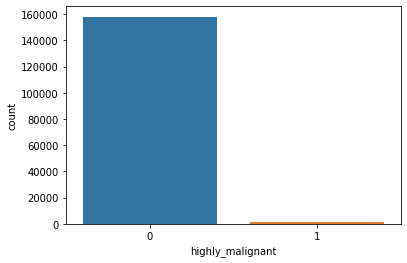
print(i)

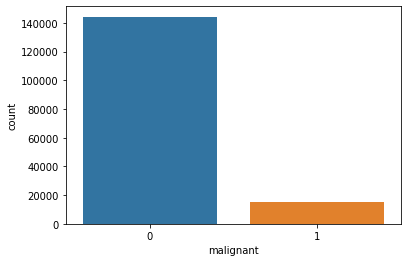
print('\n')

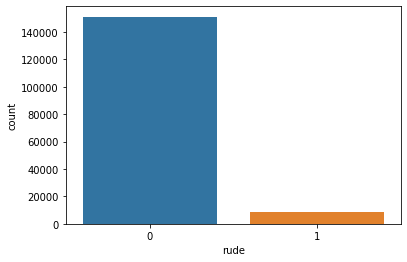
print(train[i].value\_counts())

sns.countplot(train[i])

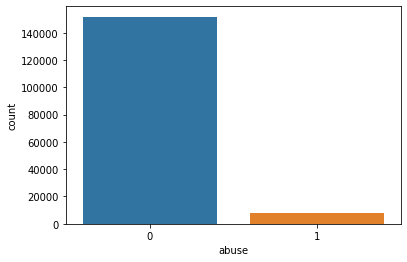
plt.show()

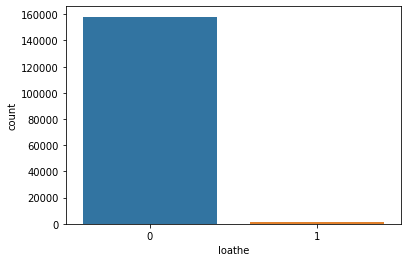












##Getting sense of loud words

from wordcloud import WordCloud

hams = train['comment\_text'][train['malignant']==1]

spam\_cloud = WordCloud(width=600,height=400,background\_color='black',max\_words=50).generate(' '.join(hams))

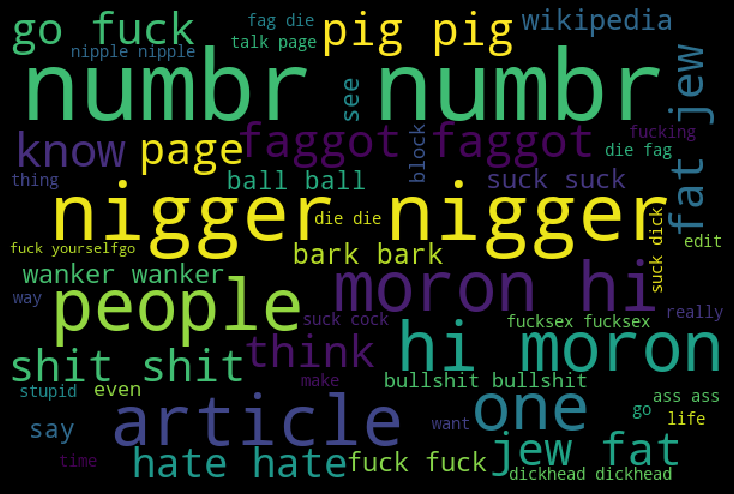
plt.figure(figsize=(10,8), facecolor = 'k')

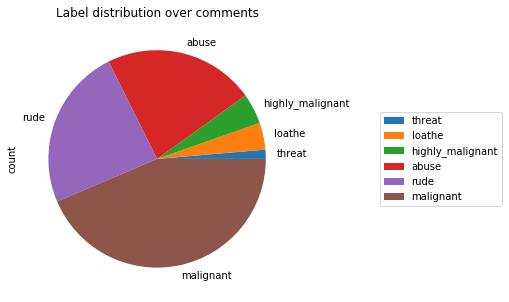
plt.imshow(spam\_cloud)

plt.axis('off')

plt.tight\_layout(pad=0)

plt.show()



****

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))

# Conclusion

## Key Findings and Conclusions of the Study

From this dataset I get to know that each feature play a very import role to understand the data. Data format plays a very important role in the visualization and Appling the models and algorithms. Pre-processing and data cleaning is the very import step than after only you will get the best accuracy score of the model

## Learning Outcomes of the Study in respect of Data Science

My learnings :-the power of visualization is helpful for the understanding of data into the graphical representation its help me to understand that what data is trying to say, Data cleaning is one of the most important step to remove missing value apply so many techniques like stop words removal, lemmatization check the loud words or null value fill it with appropriate method Various algorithms Lemmatization is the process of converting a word to its base form. Lemmatization considers the context and converts the word to its meaningful base form .Removing stop words. In this dataset I applied so many algorithms like Logistics Regression , DecisionTreeClassifie, RandomForestClassifier, AdaBoostClassifier, KNeighborsClassifier and I get to know that RandomForestClassifier perform the best result and save that model .