

**MALIGNANT COMMENTS CLASSIFICATION PROJECT**

**Submitted by:**

Akash Kanjwani

**ACKNOWLEDGMENT**

During the process of completing this project, I have referred following materials for which I owe them great gratitude.

1. For theoretical knowledge https://towardsdatascience.com/

2. Data trained video tutorials.

3. Scikit-learn <https://scikit-learn.org/stable/>

4. Machine Learning for Dummies by John Mueller and Luca Massaron - Easy to understand for a beginner book.

5. Geeksforgeeks. <https://www.geeksforgeeks.org/>

Besides that all the observation, creations of the models and graphs done by self help.

**Problem Statement**

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 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.

**Data Set Description**

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.

**Analytical Problem Framing**

Mathematical/ Analytical Modelling of the Problem

1) The size of table is 159571 × 8 i.e. no. of rows are 159571 and no. of columns are 8.

2) Out of 8 columns 6 columns are numeric type and 2 columns are object type.

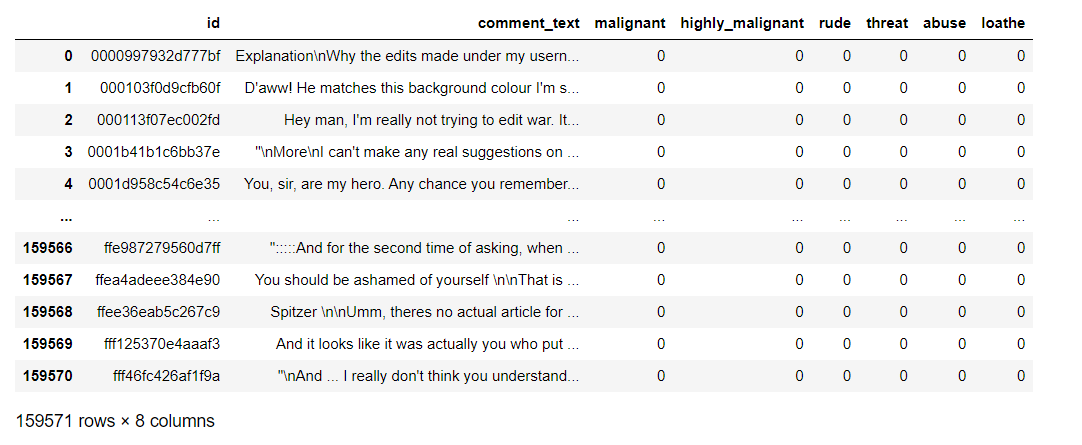
3) Null values are not present in the data set as we can see in this seaborn heatmap, so there is no need to adopt imputation technique.

4) In case of object data type, we will apply the NLP technique to convert the values in the numeric format.

Because this project is based on Natural language processing that is why we will have to adopt NLP technique such as Word Net Lemmatizer, Stop words, Vectorization etc.

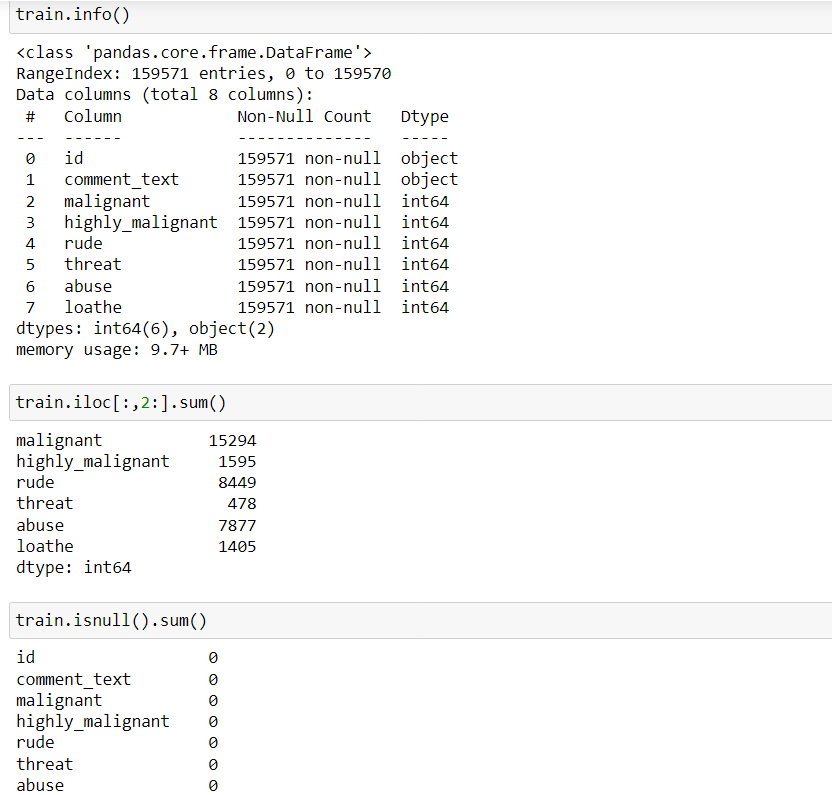
Data Sources and their formats

Data has been provided by the Flip Robo technology.

**Train Data -**

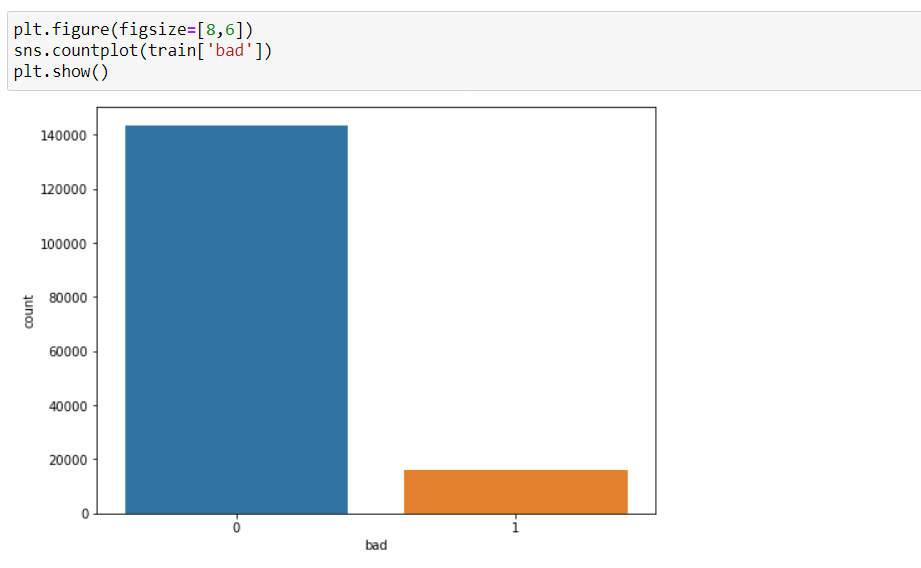
**Test Data –**

****



**Data Pre-processing Done**

****Because the project is based on NLP that is why we have adopted some NLP technique like Vectorization, Lemmatization, stop words.

****

**Hardware and Software Requirements and Tools Used**

**Anaconda Navigator**

**Jupyter Notebook**

**Language-Python**

**Many lib.-------**

import pandas as pd

import matplotlib.pyplot as plt

import seaborn as sns

import numpy as np

import warnings

warnings.filterwarnings('ignore')

import sklearn

from sklearn.linear\_model import Logistic Regression

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

from sklearn.tree import DecisionTreeClassifier

from sklearn.ensemble import RandomForestClassifier,

AdaBoostClassifier,

GradientBoostingClassifier

import xgboost as xg

from sklearn.metrics

import mean\_squared\_error, mean\_absolute\_error, r2\_score

**Pandas**- For making data frame

**Matplotlib and seaborn-** For data visualization

**Numpy-** For numerical python

**From metrice** – Classification Report , Confusion metrix , Accuracy score -For checking the model accuracy.

**Ensamble-** For boosting and bagging

**Cross\_Val\_Score**- For cross validation

**Algorithms**

• Logistic Regression

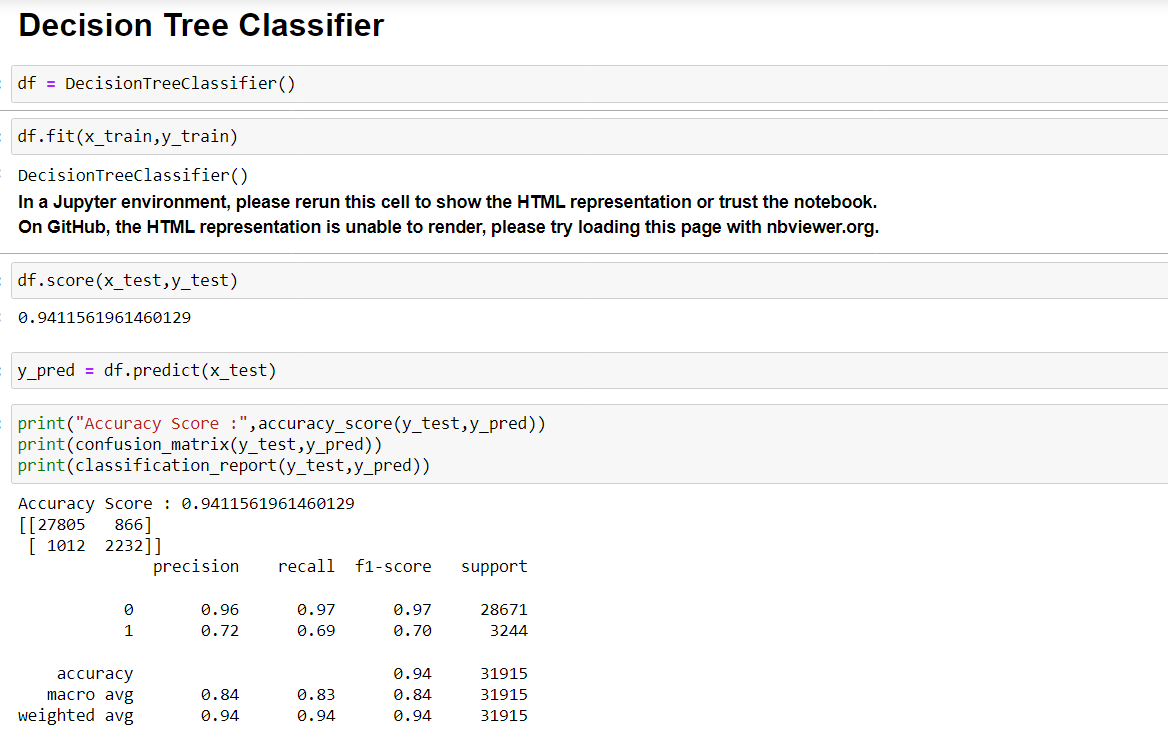
• Decision Tree Classifier

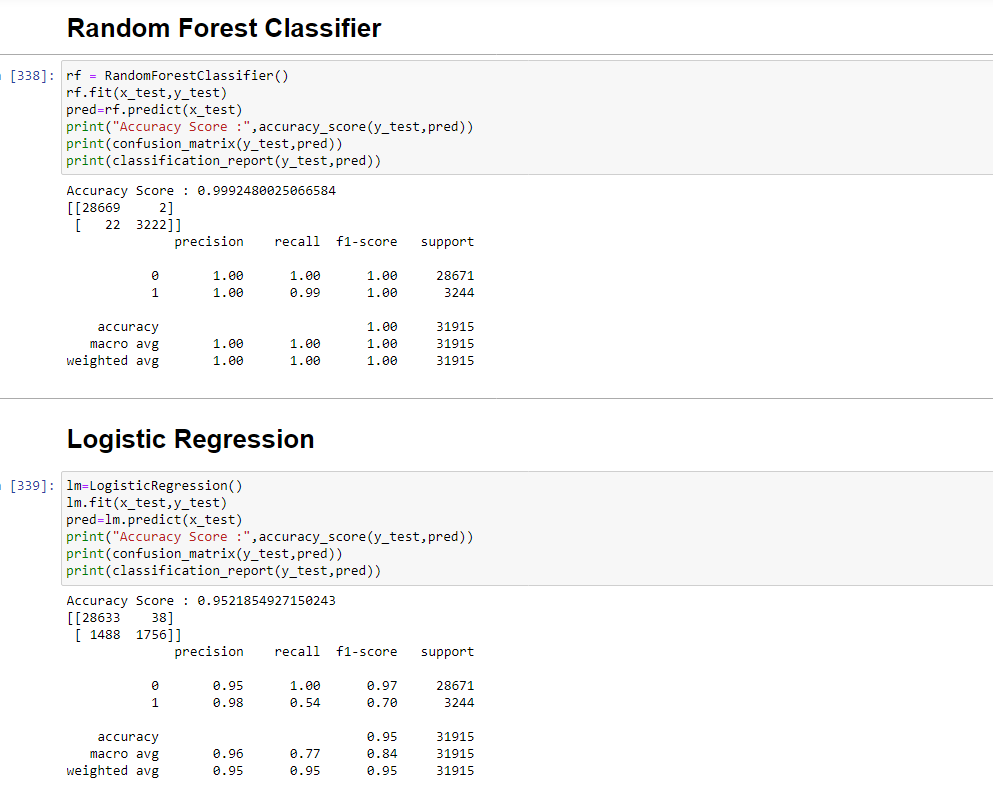
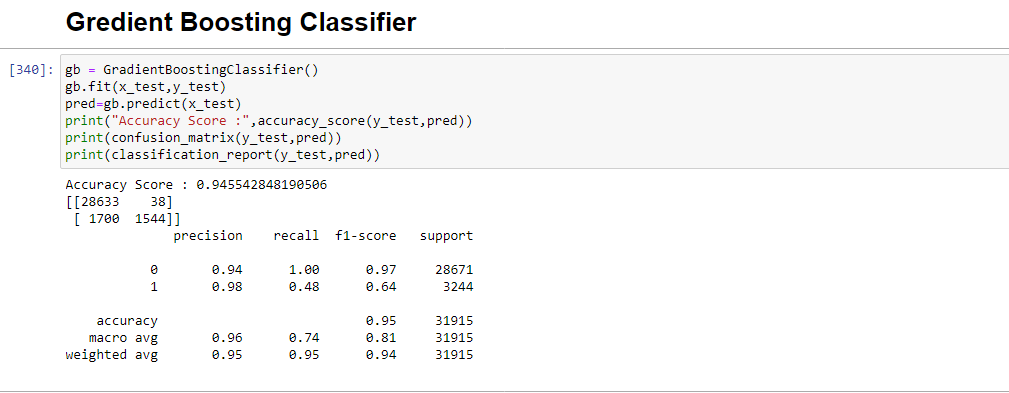
**For Bagging and boosting:**

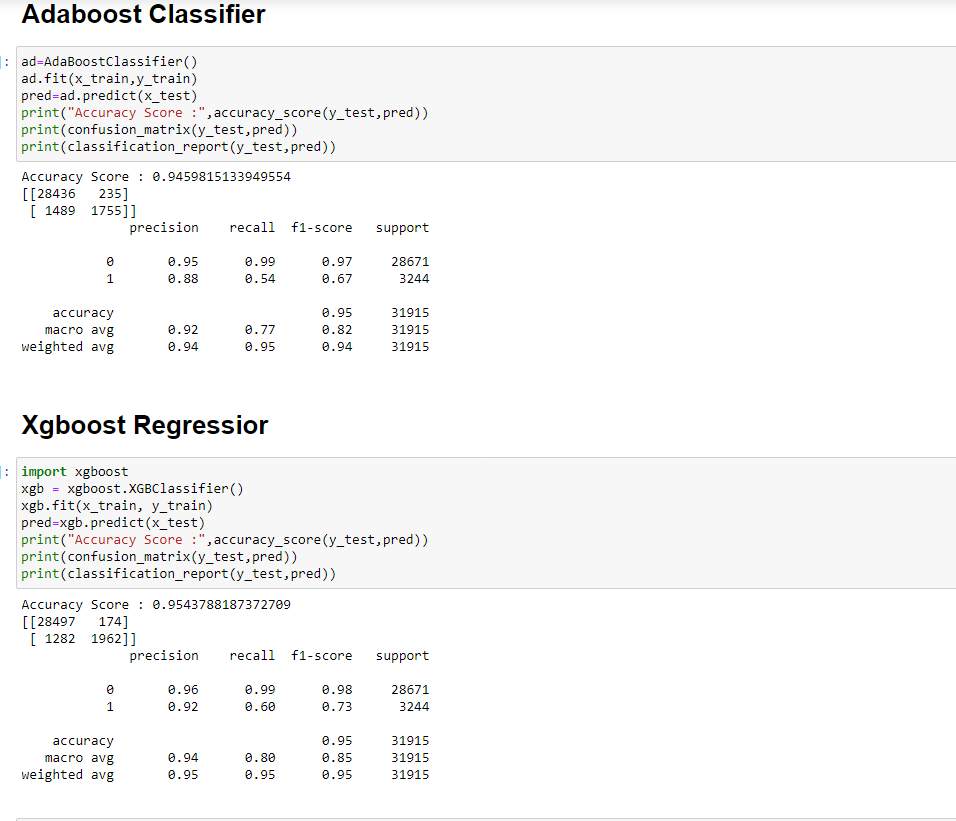
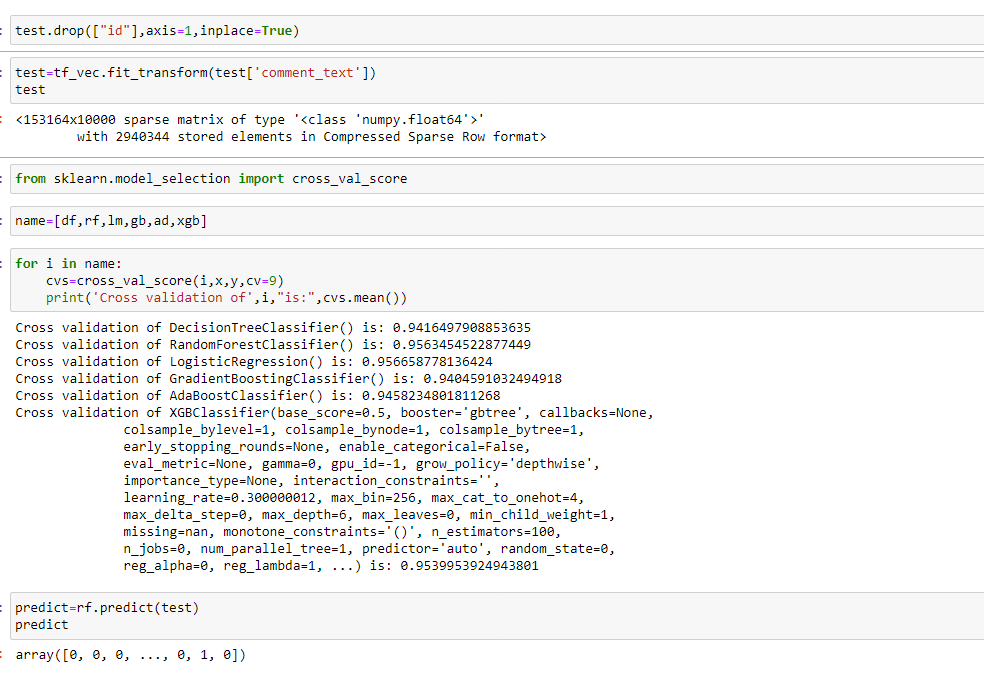
• Random Forest Classifier

• Gradient Bossting Classifier

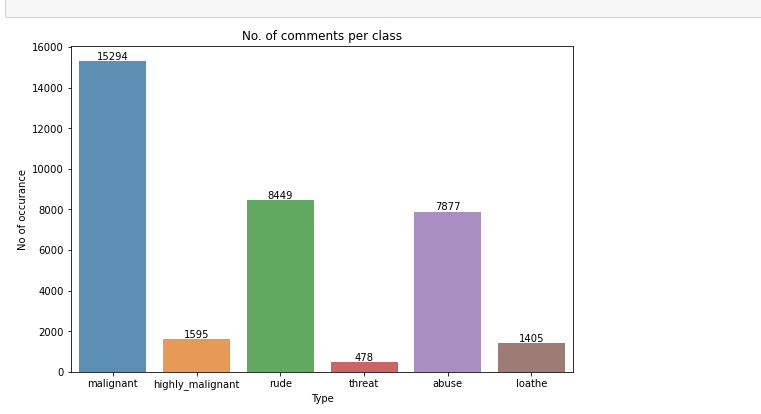
• AdaBoost Classifier

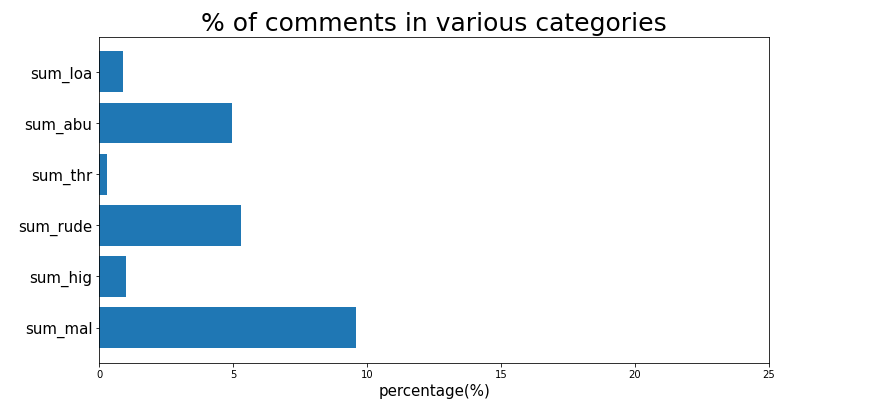
• XgBoost Classifier

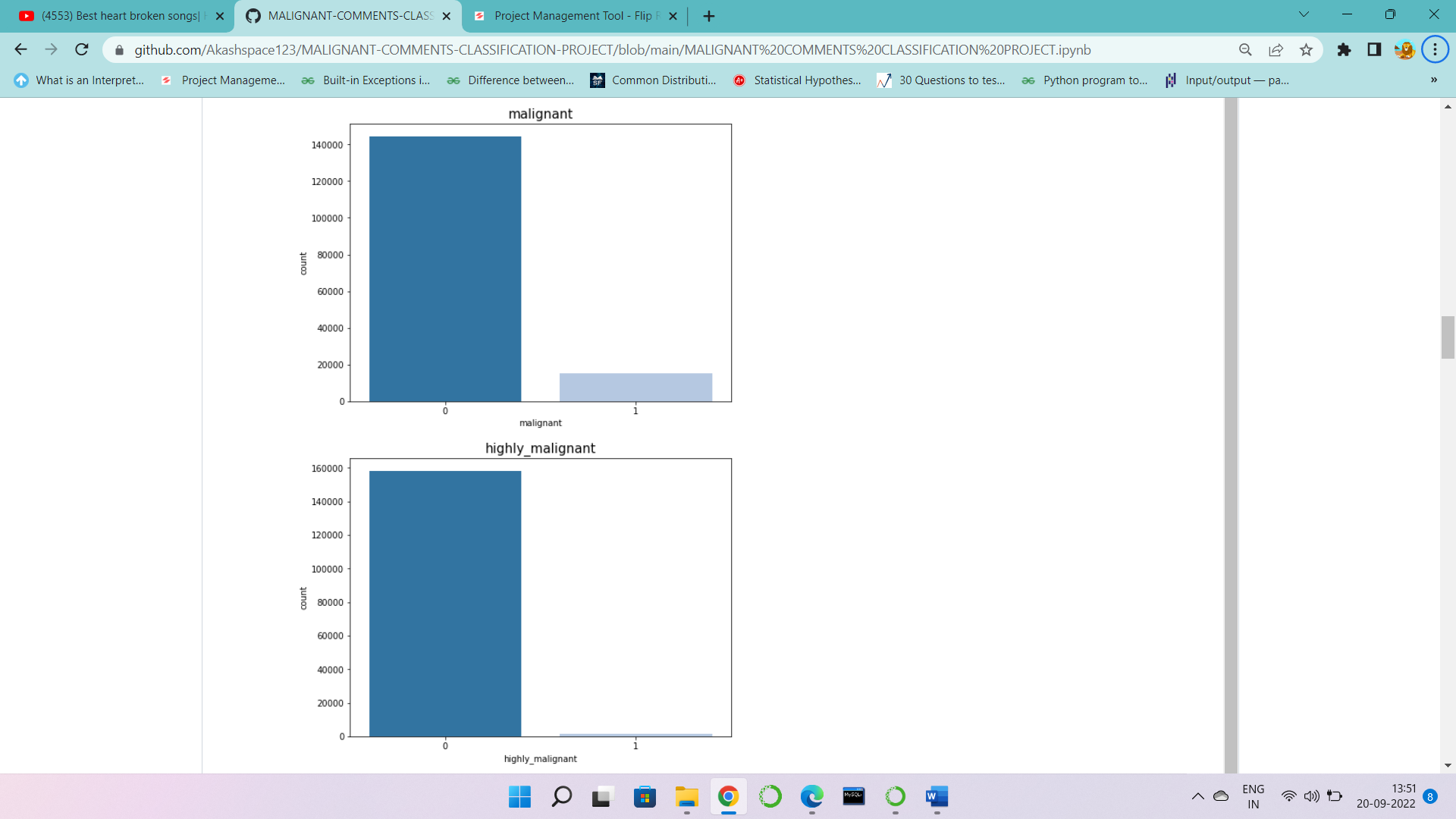
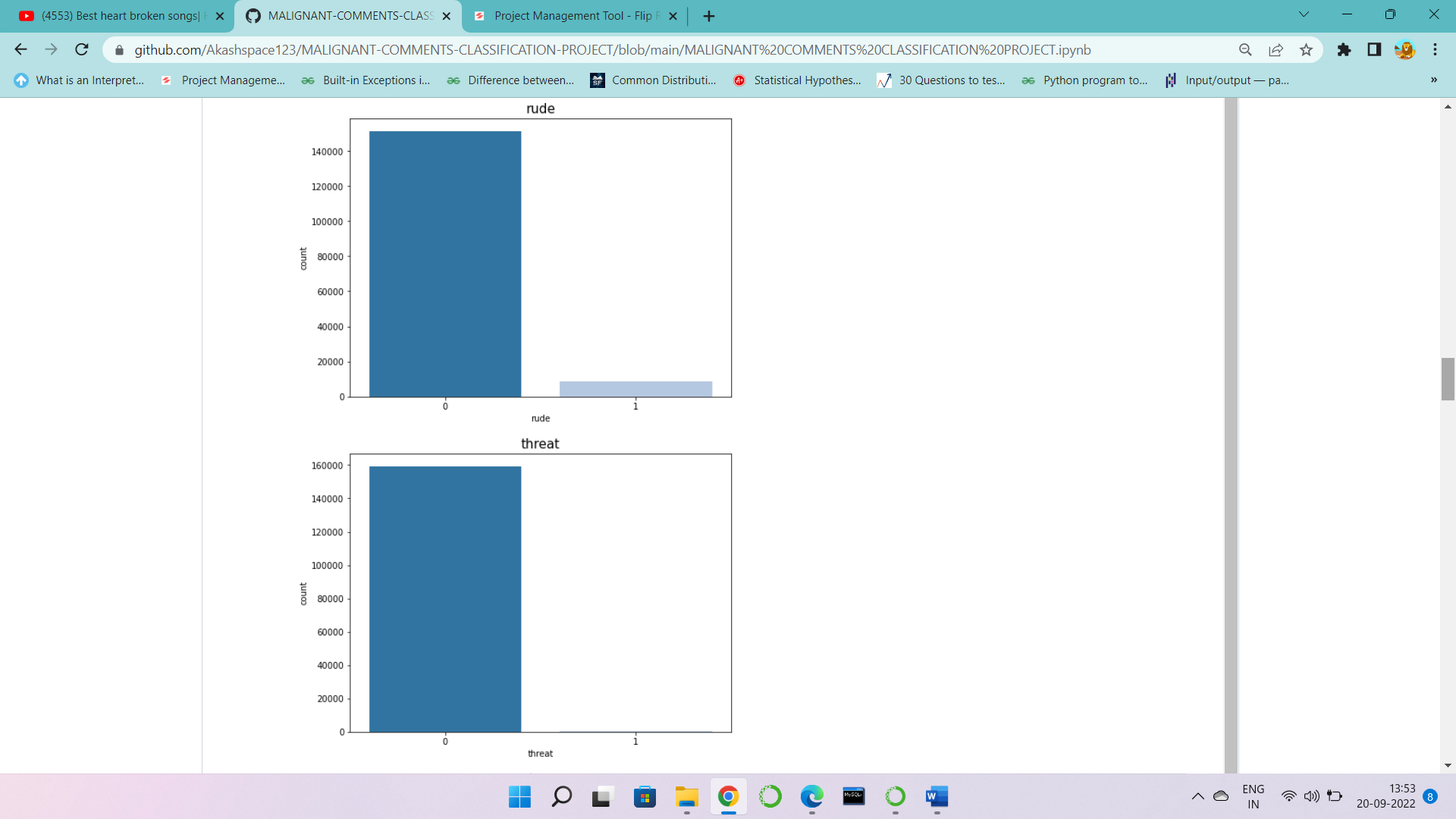
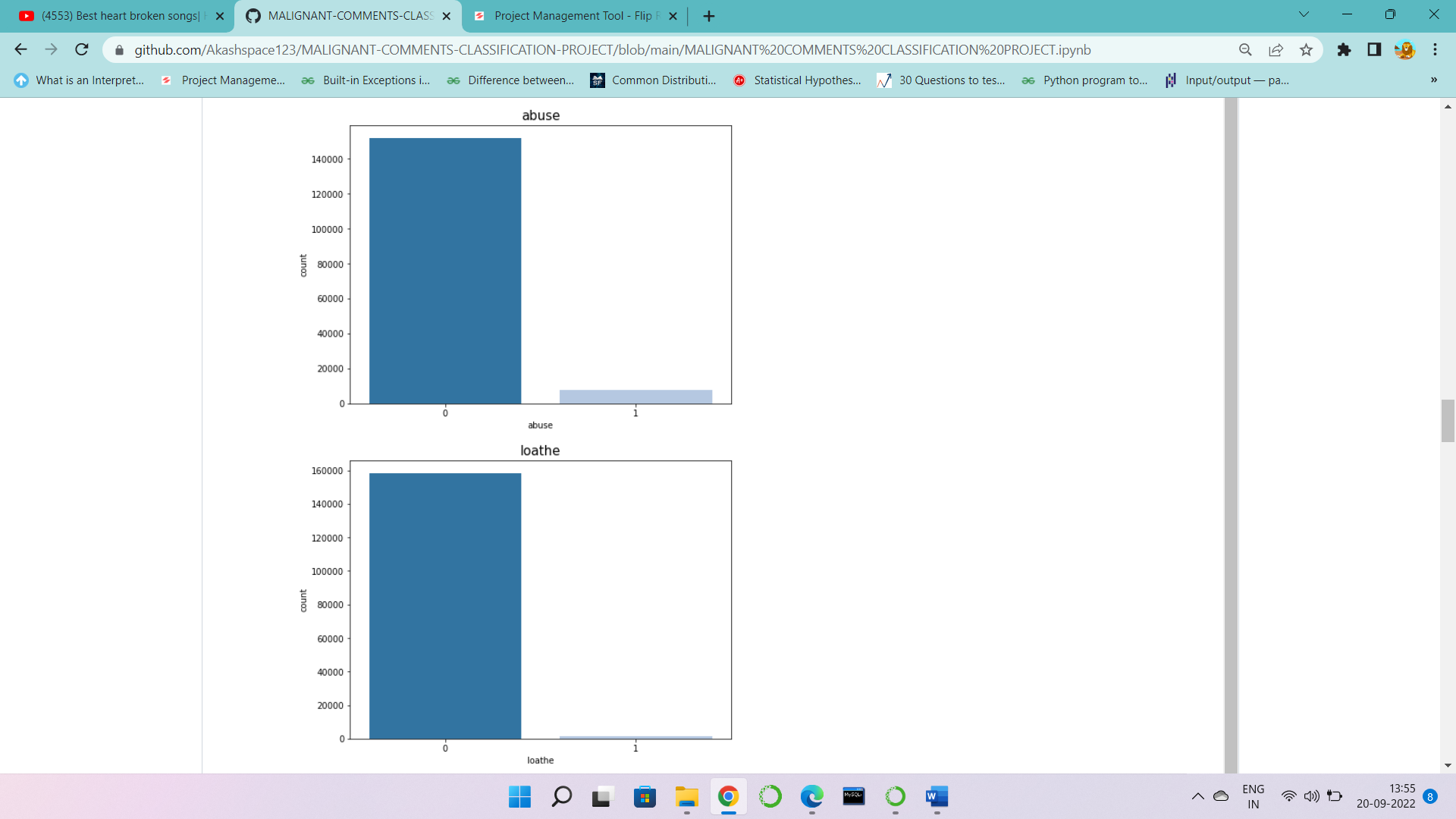




**Visualization**

****

****



**Finally we have selected the Random forest classifier model because it is giving the highest accuracy as compare to other models.**

**Accuracy score- 99.92 %**

**Cross validation score – 95.63 %**