

MALIGNANT COMMENTS PROJECT

Submitted by:

JASMINE FRERNANDES

**ACKNOWLEDGMENT**

Some of the reference include:

YouTube

Lectures from data trained

**INTRODUCTION**

* Business Problem Framing

Social media has allowed people to express their opinions freely, however, it has also brough a lot of hate and malice, making online environments uninviting for users. Online hate, described as abusive language, aggression, cyberbullying, hatefulness and many others has been identified as a major threat on online social media platforms.

The objective is to predict and classify various statements into categories for further analysis, so that they can be removed online.

* Conceptual Background of the Domain Problem

The background for the problem originates from various online forums, where-in people participate actively and make comments. As the comments sometimes may be abusive, insulting or even hate-based, it becomes the responsibility of the hosting organizations to ensure that these conversations are not of negative type and an inviting zone for the users. Hence, the objective is to build a model which could make prediction to classify the comments into various categories.

* Motivation for the Problem Undertaken

Describe

Internet comments can be hateful and negative, and machine learning can be used to fight it. It can shatter the confidence of someone and lead to mental stress. The problem we aim to solve was the tagging and classifying Internet comments that are aggressive towards other users, so that it can be controlled and restricted from spreading hatred and cyberbullying.

.

**Analytical Problem Framing**

The negative comments and words, used by a user in an online platform, classify it to belong to one or more of the following categories — ***toxic, severe-toxic, obscene, threat, insult or identity-hate*** with values (0/1).

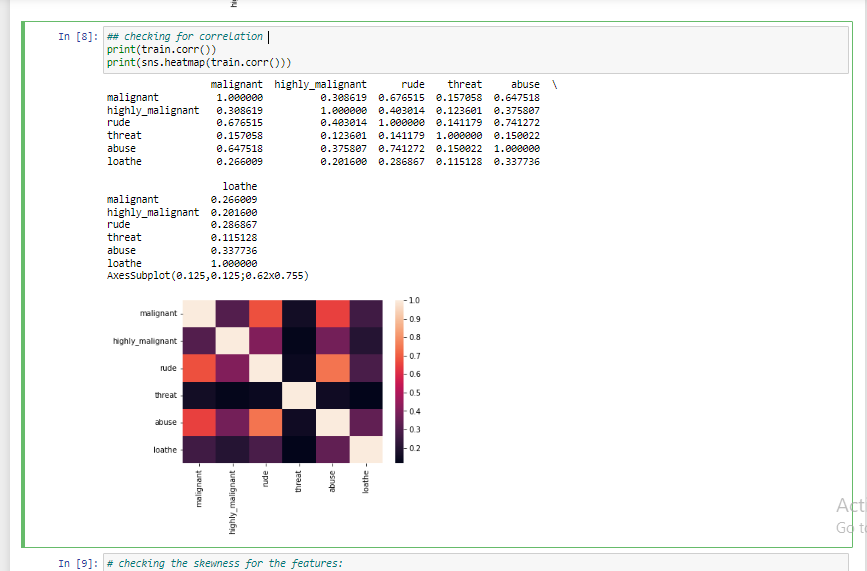
The task was to assign the correct value, 0, 1 or more than one categories of the six categories of comments listed above. One of the task is to differentiate between multi-label and multi-class classification.

The next step was to gain some useful insights from data which would aid further problem solving.

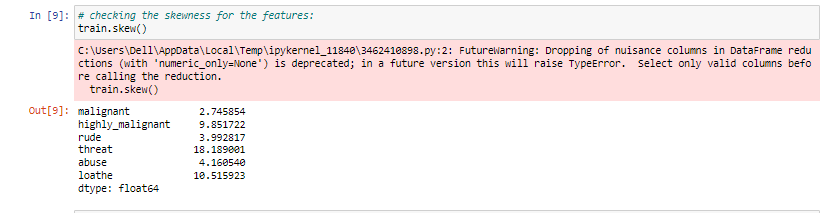
1. Checking for null values



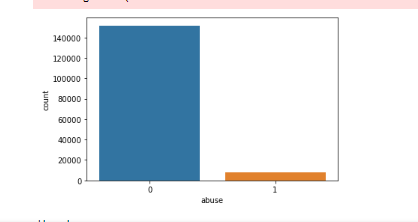
1. Checking for correlation

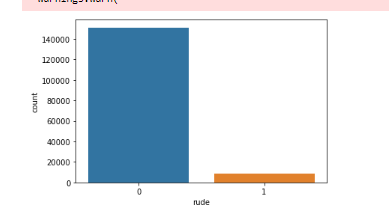


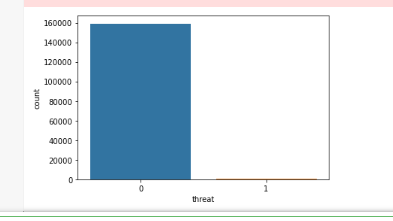
1. **Checking for skewness**

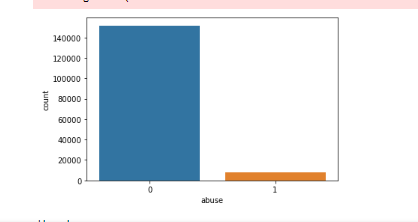


1. **Differentiating between malignant','highly\_malignant','loathe','rude','abuse','threat'**

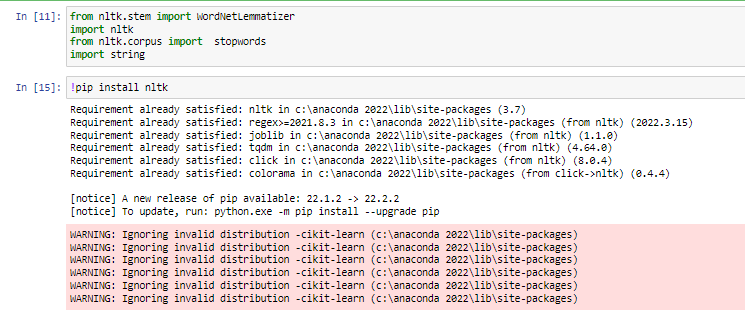


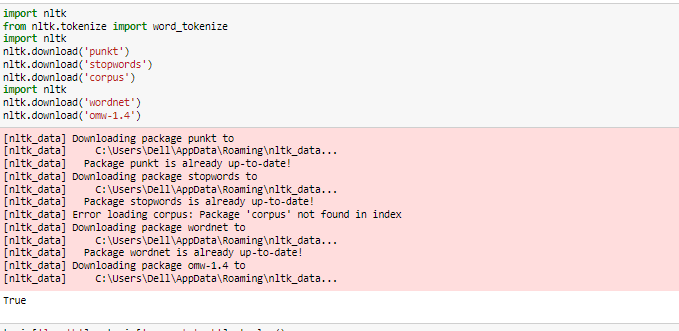




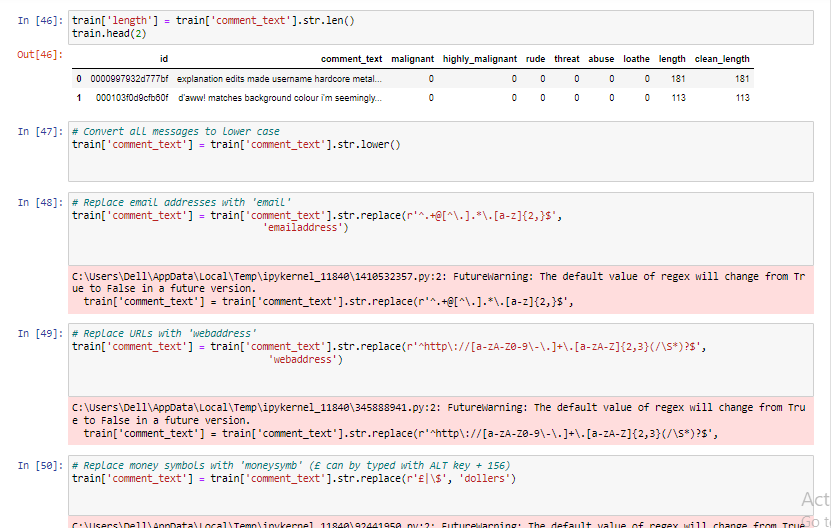


1. **Using NLTK**





**Removing any web addresses, email addresses and currency symbol**



**Replacing phone numbers**

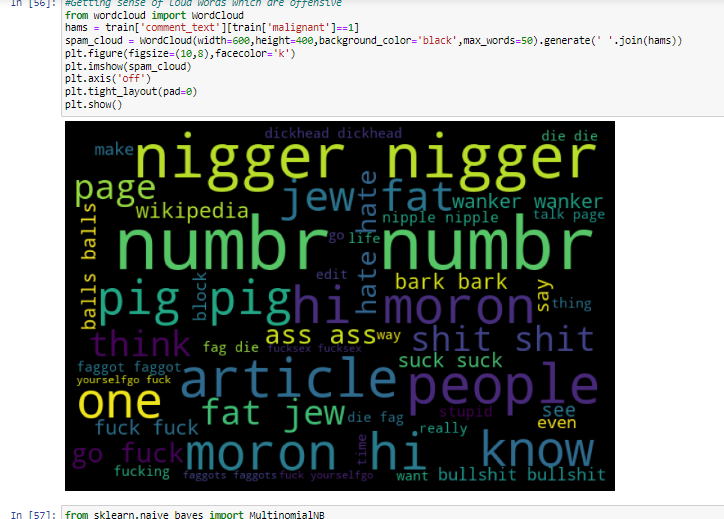
Removing phone numbers from the dataset





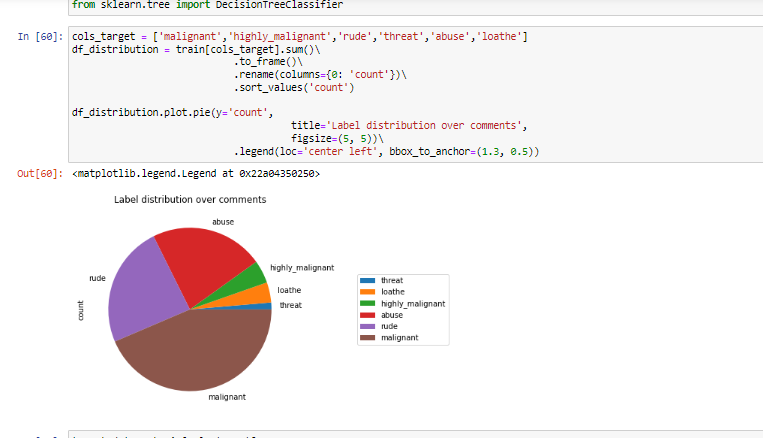
1. **Installed word cloud to highlight the top negative words on the platform**

**Some of them include numbr, moron, pig, nigger, moron**



**Categorizing malignment words as per the importance.**

**Malignant words seems to occupy most space in the data set**





**Model/s Development and Evaluation**

**Used the following methods:**

Logistic regression

Decision Tree

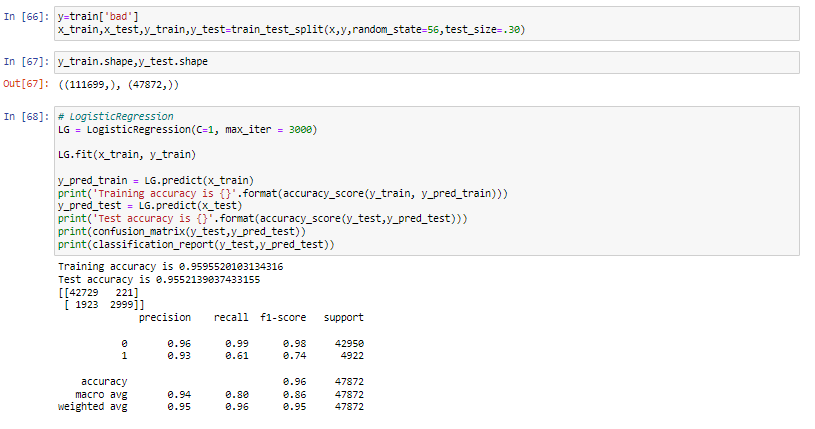
Random Forest

XGboost

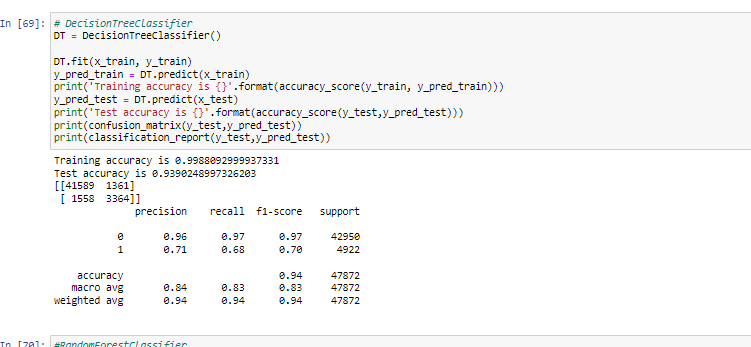
AdaBoostClassifier

KNeighborsClassifier

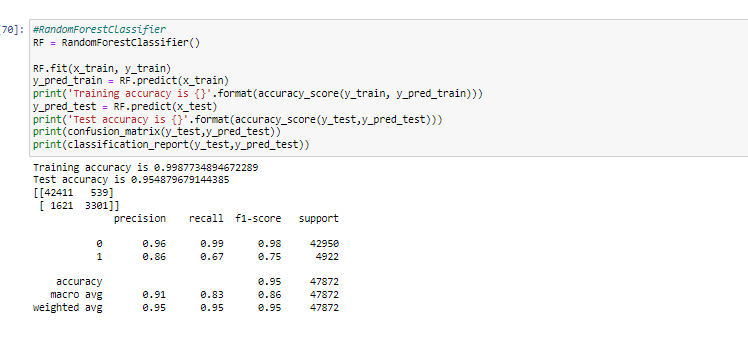
1. Training the data with Logistic regression



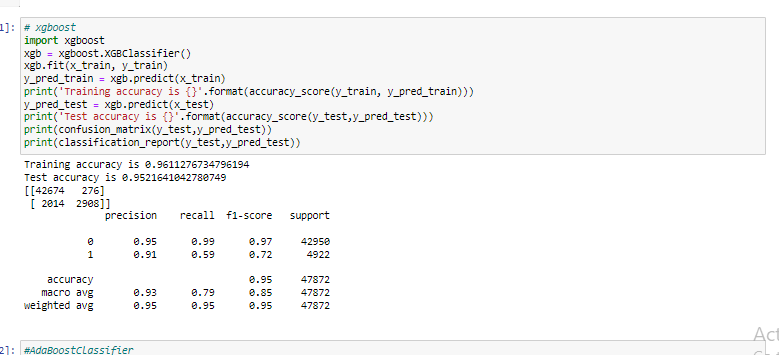
2.Training with decision Tree



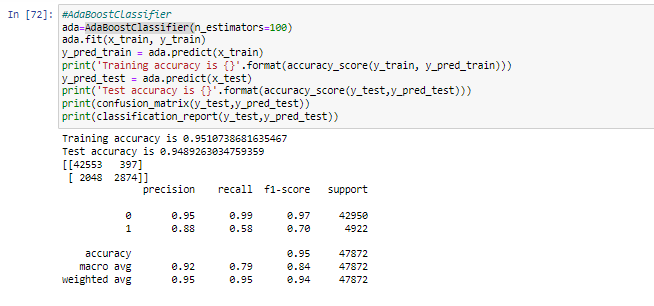
Training with Random Forest

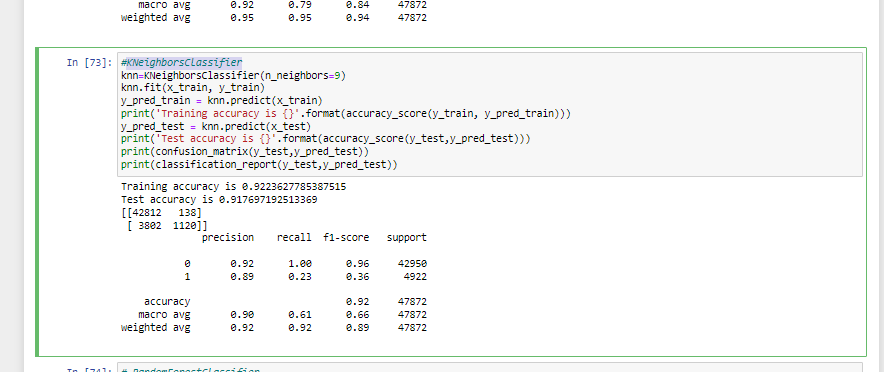


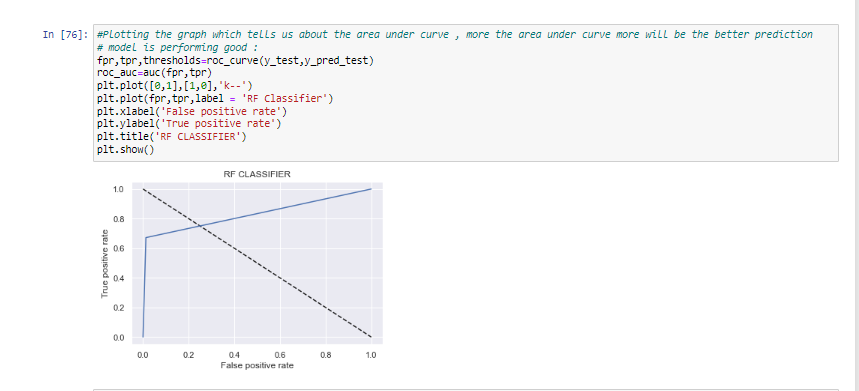
Training with XGBoost



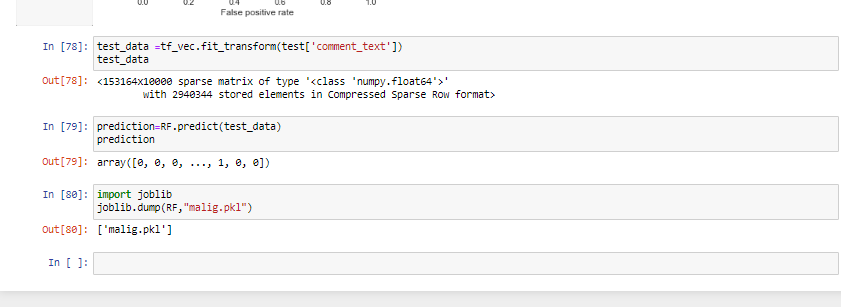
Training with Adaboost & KNeighborsClassifier







Saving the data



**CONCLUSION**

**1. Preparation for removal of punctuation marks:**I imported the string library comprising all punctuation characters and appended the numeric digits to it, as those were required to be removed too.

1. **Updating the list of stop words:**Python has a built-in dictionary of stop words. I used the same

**3. Stemming and Lemmatising:**use stemming as it helps in achieving the training process with a better accuracy.

4. This is quite similar to stemming in its working but not exactly same. I used the **word-net library** in nltk for this purpose. Stemmer and Lemmatizer were imported from nltk.

**5. Applying Count Vectorizer :**Count Vectorizer is used for converting a string of words into a matrix of words. Column headers have the words themselves and the cell values signify the frequency of occurrence of the word.