**HATE SPEECH DETECTION**

PREPROCESSING OF TEXT DATA INCLUDES:

1. Replacing URLS, usernames
2. Using Emoji Python library to detect emojis
3. Replacing Hashtags , punctuations
4. Removing stop words
5. Word Lemmatization

Vectorization is performed on the clean data using:

1. Word Embeddings
2. TF-IDF Word Vectorizer
3. Count Vectorizer - Bag of Words model

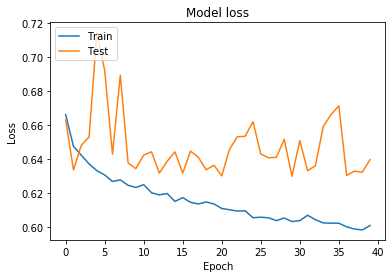
Multiple Classifiers were used in each form of vectorization:

As it is a binary classification, I used classifiers such as MLP, Logistic Regression, SVM with a gaussian kernel and Random Forest Classifier.

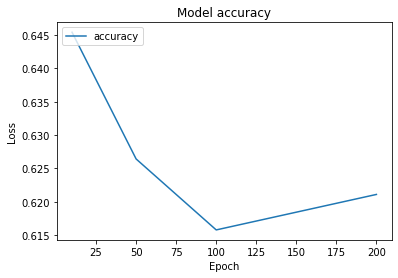
GridSearchCV was used for hyperparameter tuning that gave the best parameters as output that produced the most accurate results.

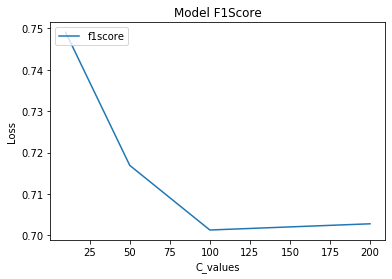
**1.WORD EMBEDDINGS:**

Using MLP:

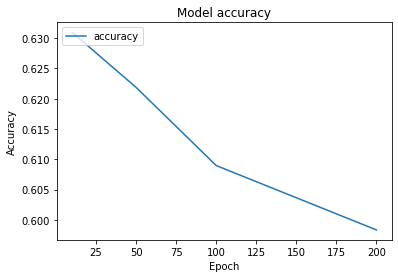


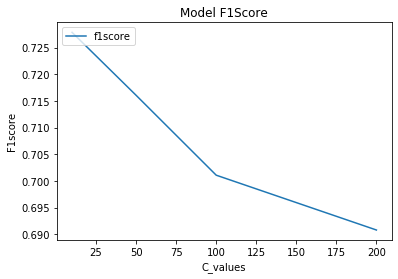
USING SVM: Different values of C and kernel = ‘rbf’





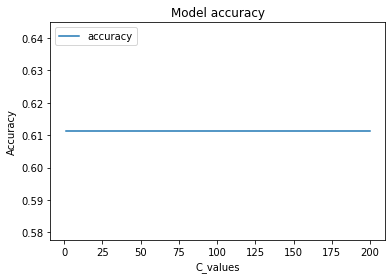
Using Logistic Regression:





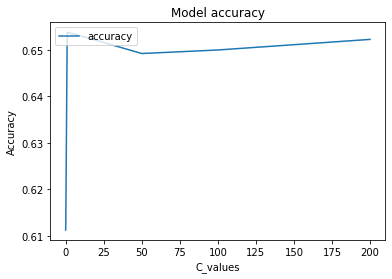
**2. TF - IDF**

Using SVM:

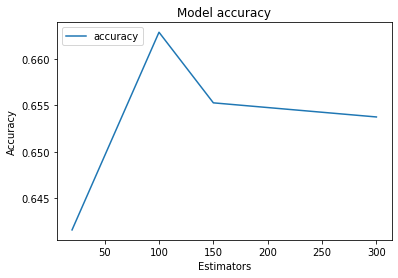


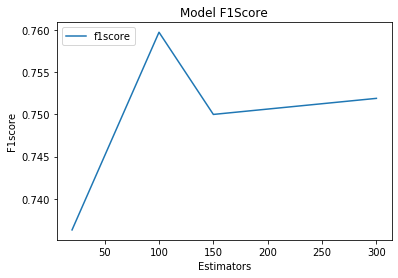


Using Logistic Regression :



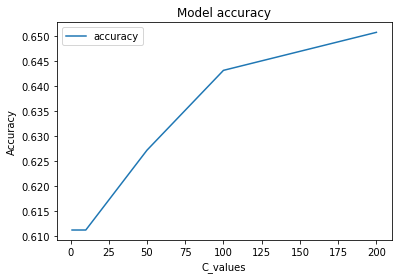
Using Random Forest Classifier:

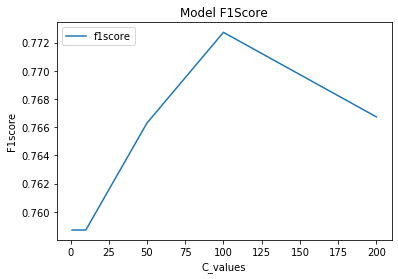




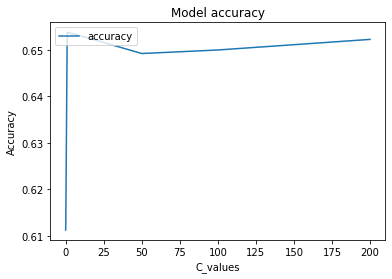
**3. Using Count Vectorizer:**

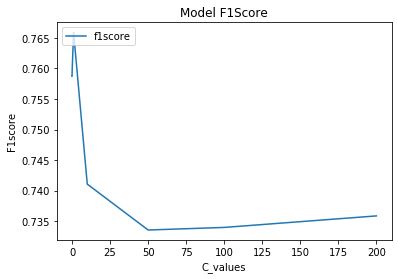
Using SVM:



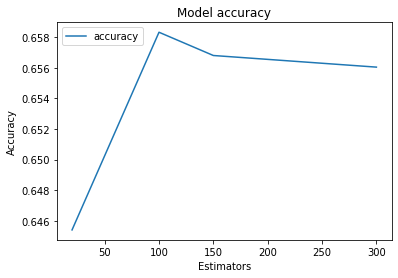


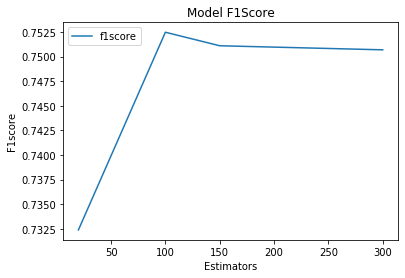
Using Logistic Regression:





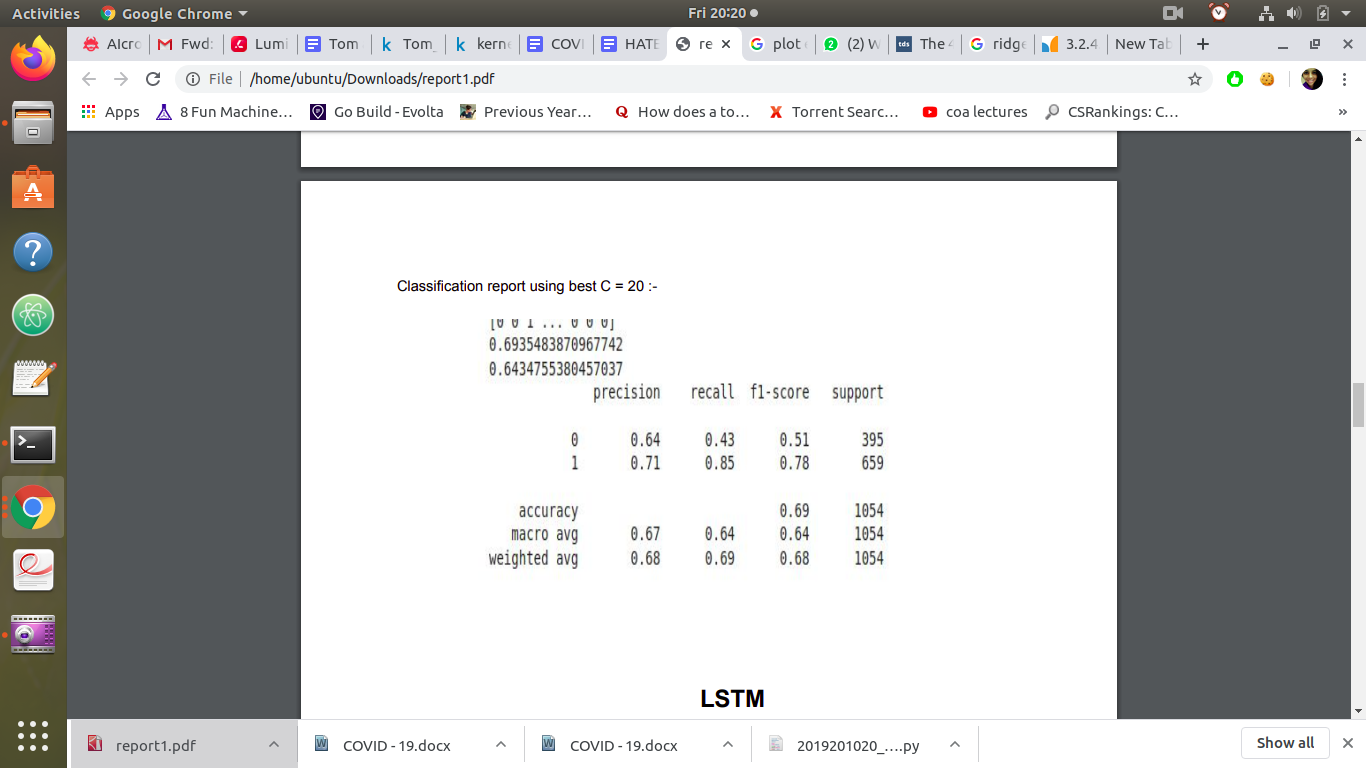
Using Random Forest:





**Observation**:

As we can see from the above graphs that data vectorized using TF-IDF Vectorizer and SVM classifier outperforms all the other classifiers and vectorizers and hence selected it as the classifier for the final prediction.



The above confusion matrix shows how well the classifier performed.

**How to run the program**:

The paths are specified in the program itself. Those are absolute paths. So if one needs to run the program, he/she needs to change the paths in the program manually

And then run the program as a normal python file.