Sentiment classification of tweets regarding coronavirus

Name:	Sharanyak Podder
Roll No.:	19287
Institute Name:	IISER Bhopal
Program:	DSE
Problem Release date:	August 15, 2022
Date of Submission:	September 29, 2022

1 Introduction

Sentiment classification of Tweets is the task of computationally classifying tweets depending on their arrangements of words and contents. It is also a task of significantly categorizing the opinion expressed in the text, i.e., Tweet, to determine the writer's attitude to the particular topics. COVID-19 is the Corona Virus Disease of 2019, declared a pandemic by the World Health Organization (WHO) in March 2020. Facebook, blogs, Instagram and Twitter and other social media platforms have become places where people post their opinions on specific topics. The objective is to classify the tweets into the following classes: *extremely positive*, *positive*, *neutral*, *negative* and *extremely negative*. The training data set[1] has six columns: *UserName*, *ScreenName*, *Location*, *TweetAt*, *OriginalTweet*, and *Sentiment*. Also, this training data set1 has the range index of 39664 entries, 0 to 39663. The data set contains a total number of 1311293 words. After the manual correction in the training data set, there are no null values in *UserName*, *ScreenName*, *TweetAt*, *OriginalTweet*, and *Sentiment*. The *Location* has the most number of null values and that is 8306, and also this column has 20.94 percent null values. The training data set[1] has the Tweets from 4th of January, 2020 to 4th of December, 2020. The most of the Tweets were done in the month of March.1

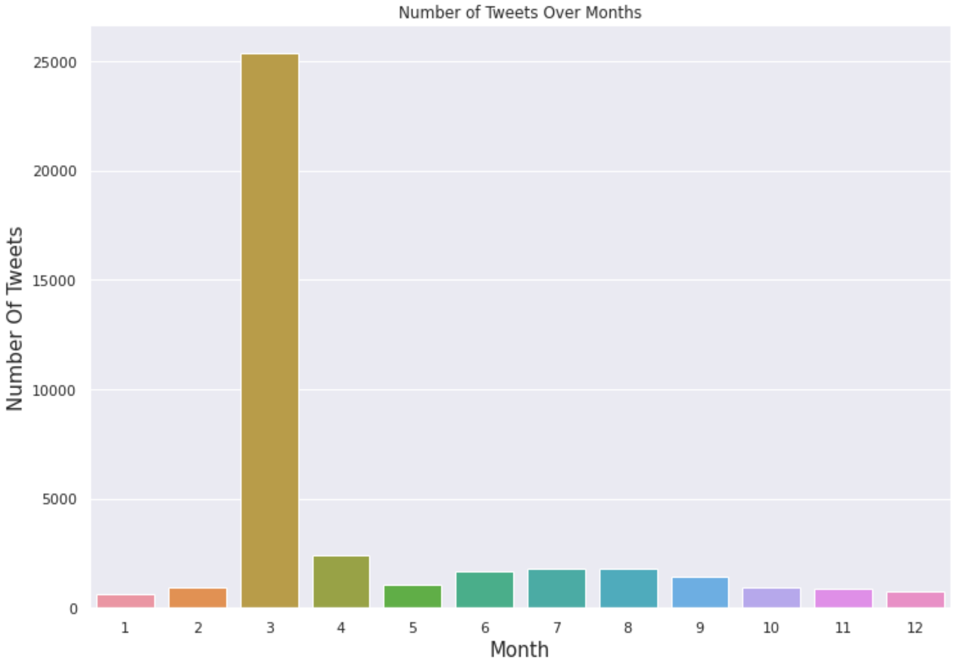
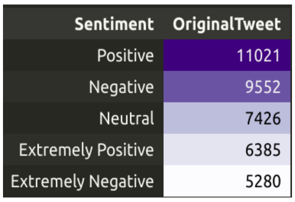


Figure 1: The Number of Tweets In Specific Months

[1] The provided training data set has anomalies in the row index 6728, 6729, 9776, 9777, 12295, 12296, 34160, 34161. And, the test data set has anomalies in the row index 819, and 820. These were corrected manually, we urge to use the training data set given in the folder of Anusandhan Server, else you can get the access of it from the links of Manually Corrected Data Sets.

The number of each classified sentiments can be observed from this figure2.



Sentiment	OriginalTweet
Positive	11021
Negative	9552
Neutral	7426
Extremely Positive	6385
Extremely Negative	5280

Figure 2: The Number of Tweets on Different Sentiment Class

It can easily be observed that the number of positive sentimental tweets is the highest and the number of extremely negative sentimental tweets is the lowest. In the figure below3, we observe the number of unique values per column present in the training data set.[1]

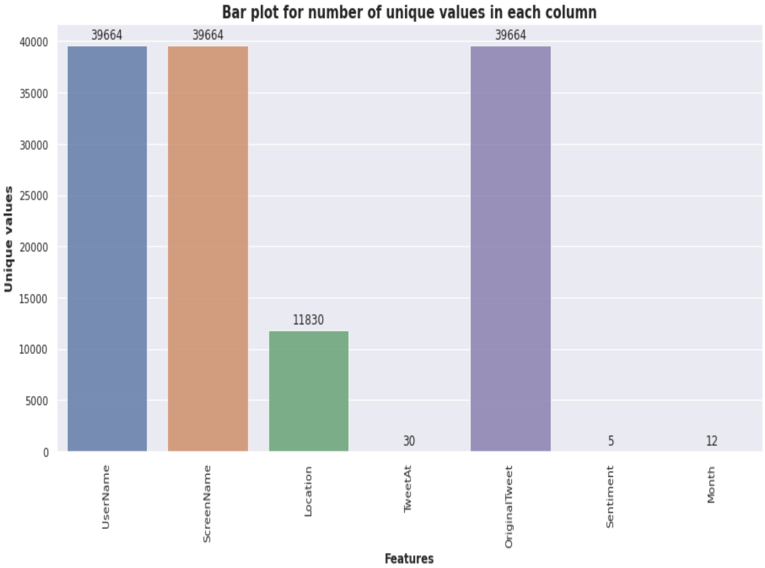


Figure 3: The Number of Unique Values per Column

2 Methods

In this project we have used the following machine learning techniques: *Logistic Regression*, *CatBoost*, *Support Vector Machines*, *XGBoost*, *Random Forest*, *Naive Bayes*, *Stochastic Gradient Decent*. We split the training data set 1 into two portions, 80% for the training and 20& for the testing. We used stratification, which means that the **train test split** method returns training and test subsets that have the same proportions of class labels as the input data set. We used the CountVectorizer from **sklearn** to prepare the **tf-idf** model. The training Accuracy, Testing Accuracy, Precision, Recall, and f1-score for the various models are listed in the table.3

We have five class of **Sentiments** and have extracted the Hashtags from the Original tweet and created a separated column. We have separated five list of Hashtags for the given five classes in the labeled data.

3 Experimental Analysis

For this project, we have experimented with various machine learning classifiers and fed them data by **Tokenization**, and **Lemmatization**. While using **tf-idf**, the table3 shows the comparison of the ML models on the processed data. Above all the models, that are used for this project Logistic Regression and CatBoost has the highest validation accuracy.

	Train Accuracy	Validation Accuracy	precision	recall	f1-score
Naive Bayes	0.75	0.46	0.43	0.53	0.45
SGD	0.89	0.56	0.59	0.57	0.57
Random Forest	0.99	0.55	0.52	0.60	0.53
SVC	0.91	0.58	0.57	0.63	0.58
Logistic Regression	0.95	0.60	0.61	0.62	0.61
XGBosst	0.68	0.56	0.56	0.59	0.56
CatBosst	0.66	0.60	0.60	0.63	0.60

The figure4 illustrates the result obtained using different ML classifiers.

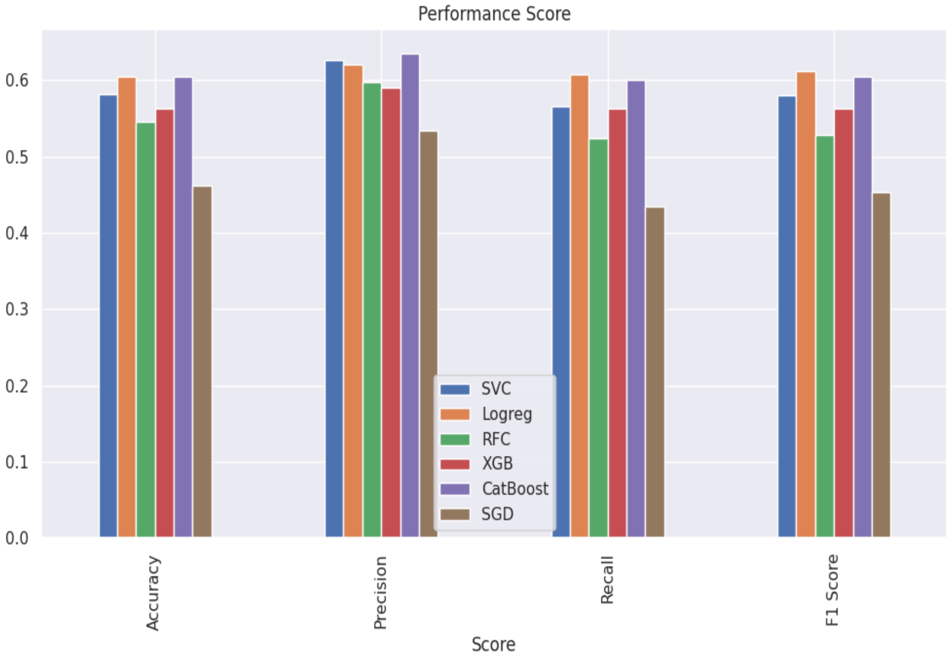


Figure 4: Comparison between the ML Models

Lemmatization, unlike-stemming, reduces the inflected words properly ensuring that the root word belongs to the language. Lemmatization considers the context and convert the words to its meaningful base form. This is why we got better results with *Lemmatization.*

4 Discussions

We have use Grid Search CV Hyper-parameter Tuning for this multi class classification. In the table below4, the performance of **SVC**, **SGD**, **Random Forest**, **Logistic Regression** are shown.

	Grid Search CV Accuracy
SGD	0.57
Random Forest	0.38
SVC	0.30
Logistic Regression	0.55

5 Contribution

I, Sharanyak Podder, have done the pre-processing of the data, extracting salient features of the data and have mentioned it in the introduction, applied variuos classical ML modles with the tokenization, lemmatization, and pos tagging and also applied Gridsearch(with K = 10 folds) with hyperparameter tuning.