Capstone Project -3

Coronavirus Tweet Sentiment Analysis

(individual)
Rajesh Kumar Patel

Introduction:

COVID-19 originally known as CoronaVirus Disease of 2019, has been declared as a pandemic by the World Health Organization (WHO) on 11th March 2020.

Sentiment Analysis is the process of computationally identifying and categorizing opinions expressed in a piece of text, especially in order to determine whether the writer's attitude towards a particular topic is positive, negative, or neutral.

Problem Statement:

The problem statement is to build a classification model to predict the sentiment of COVID-19 tweets.

About dataset:

This dataset has 41157 observations in it with 6 columns.

Sentiment column is target column in dataset.

In this dataset have 5 types of Sentiment:

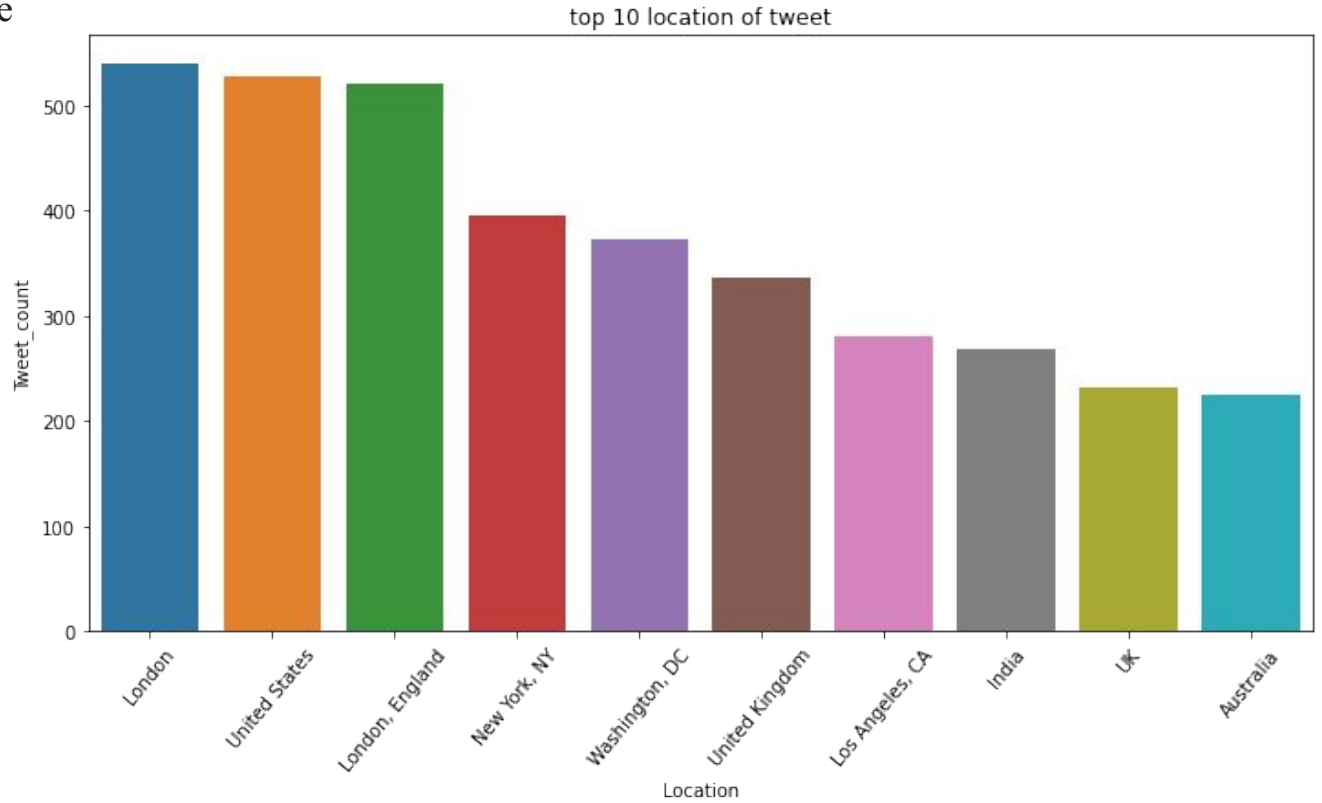
- Extremely Positive
- Positive
- Neutral
- Negative
- Extremely Negative

Feature information in details

- **UserName:** This is the username(encoded in number) unique for every data points
- **ScreenName:** This is the ScreenName(encoded in number) unique for every data points
- **Location:** places where from tweets are coming
- **TweetAt:** date of tweets data collected
- **OriginalTweet:** Original tweets are stored (This feature is more important in our analysis)
- **Sentiment:** types of sentiment (Positive, Negative, Neutral, Extremely Positive, Extremely Negative) it is a dependent variable.

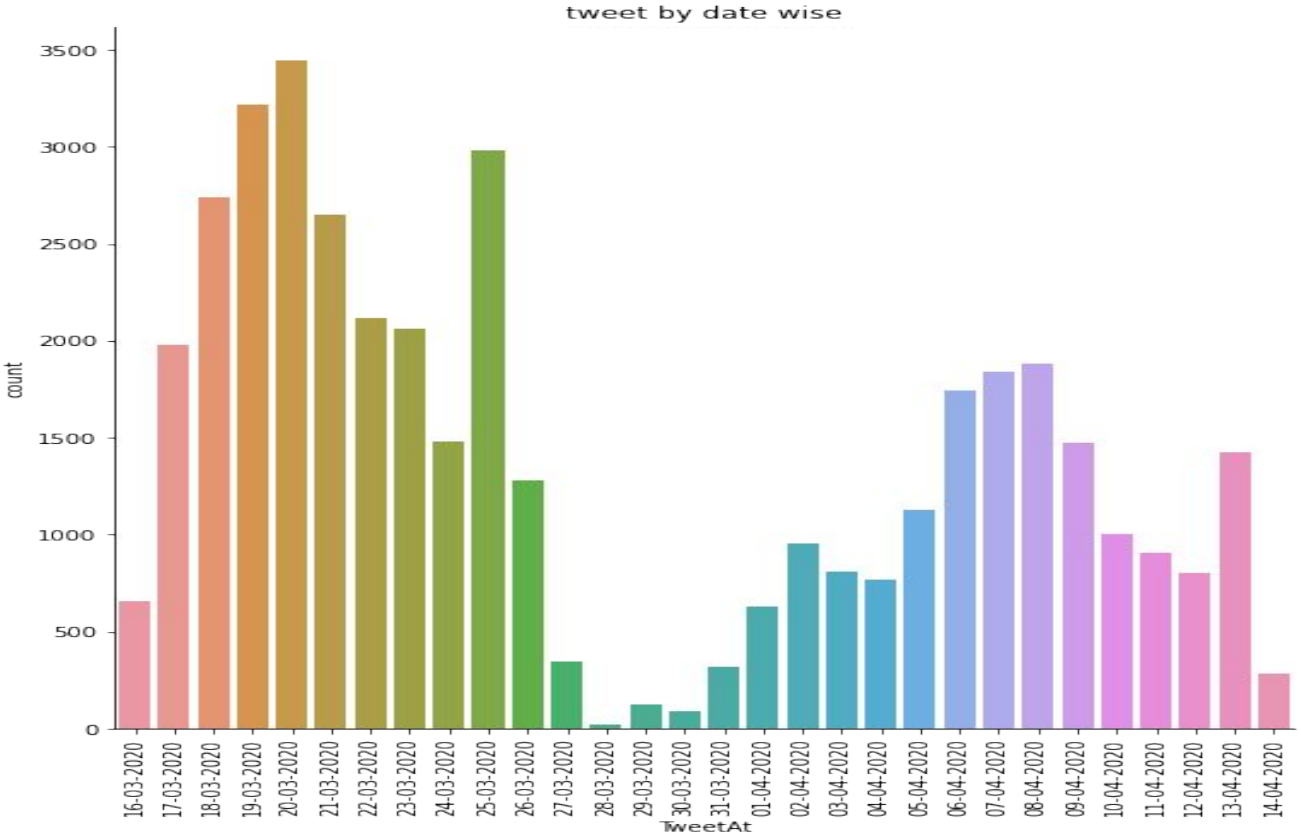
Exploratory Data Analysis

Location feature



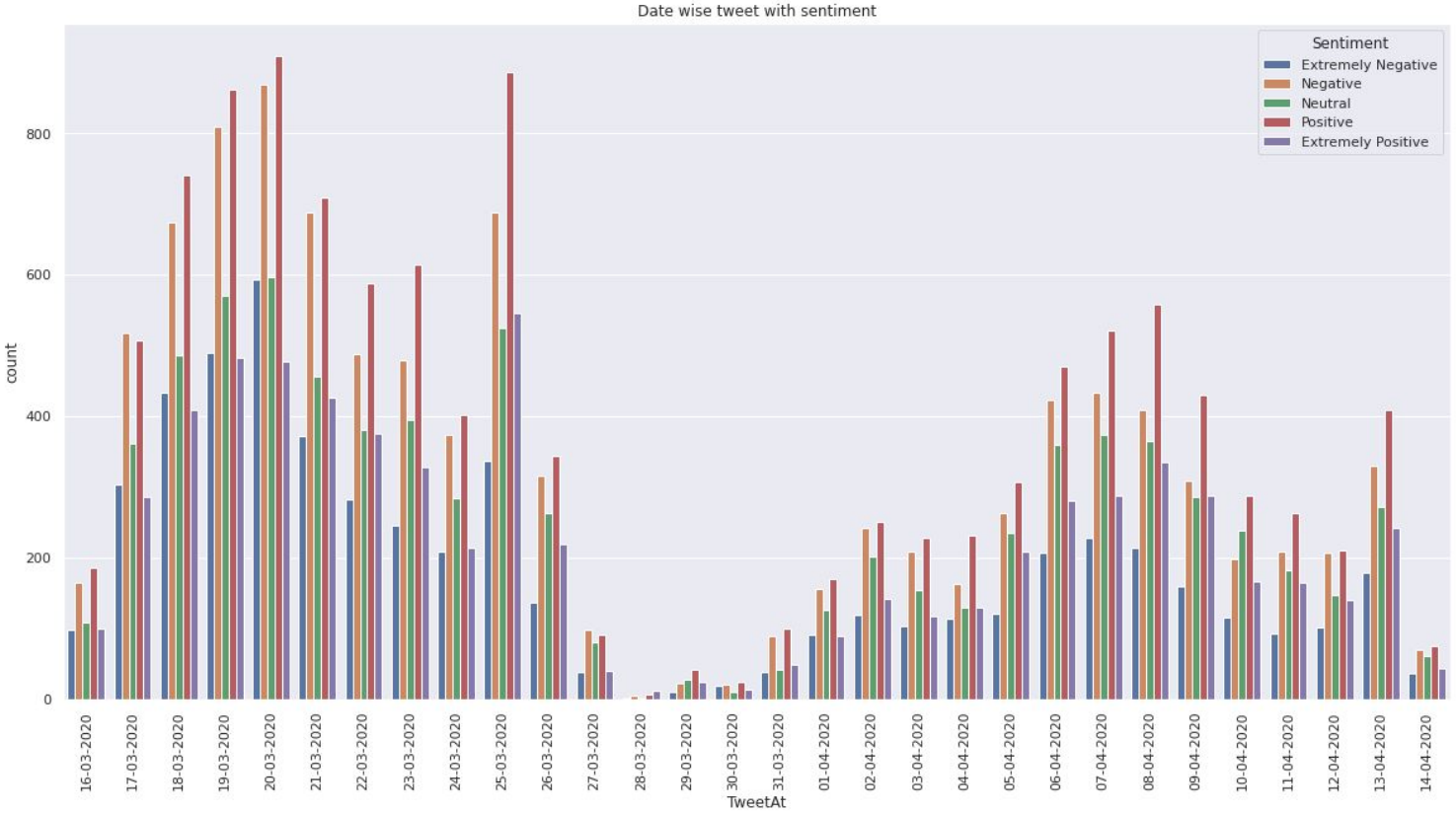
Most of the tweets comes from London, and US location.

TweetAt feature



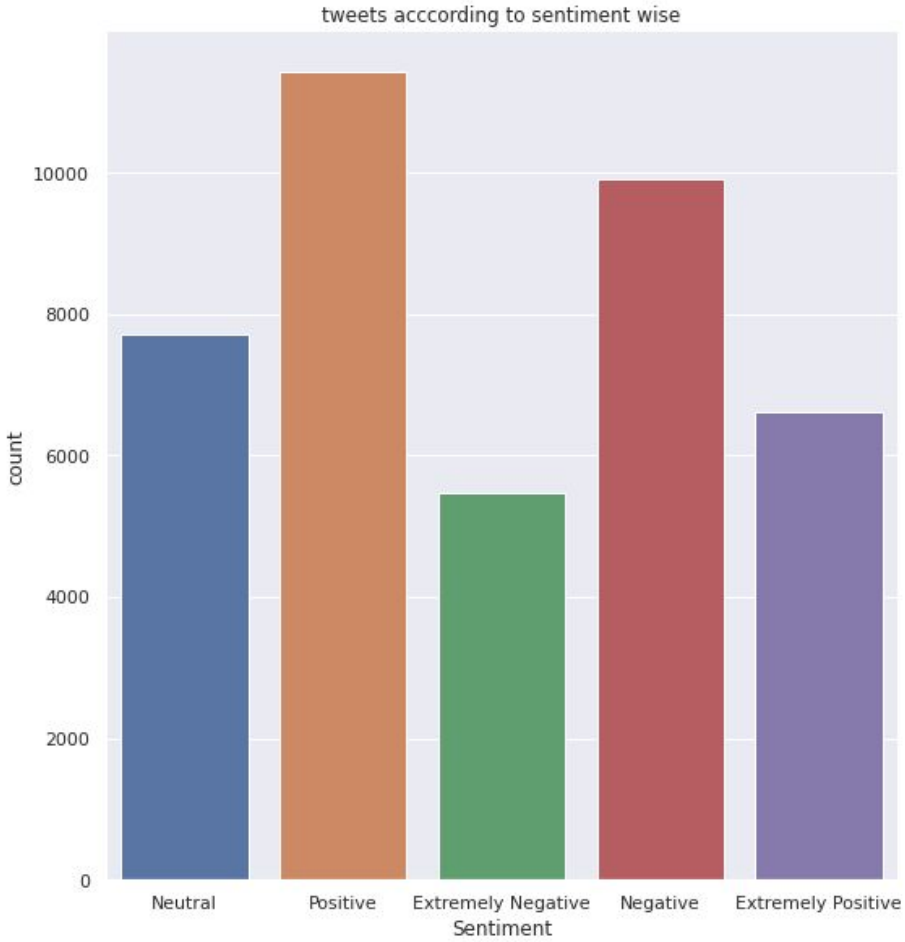
all tweets belongs to March and April months of 2020

Date-wise tweet with Sentiment



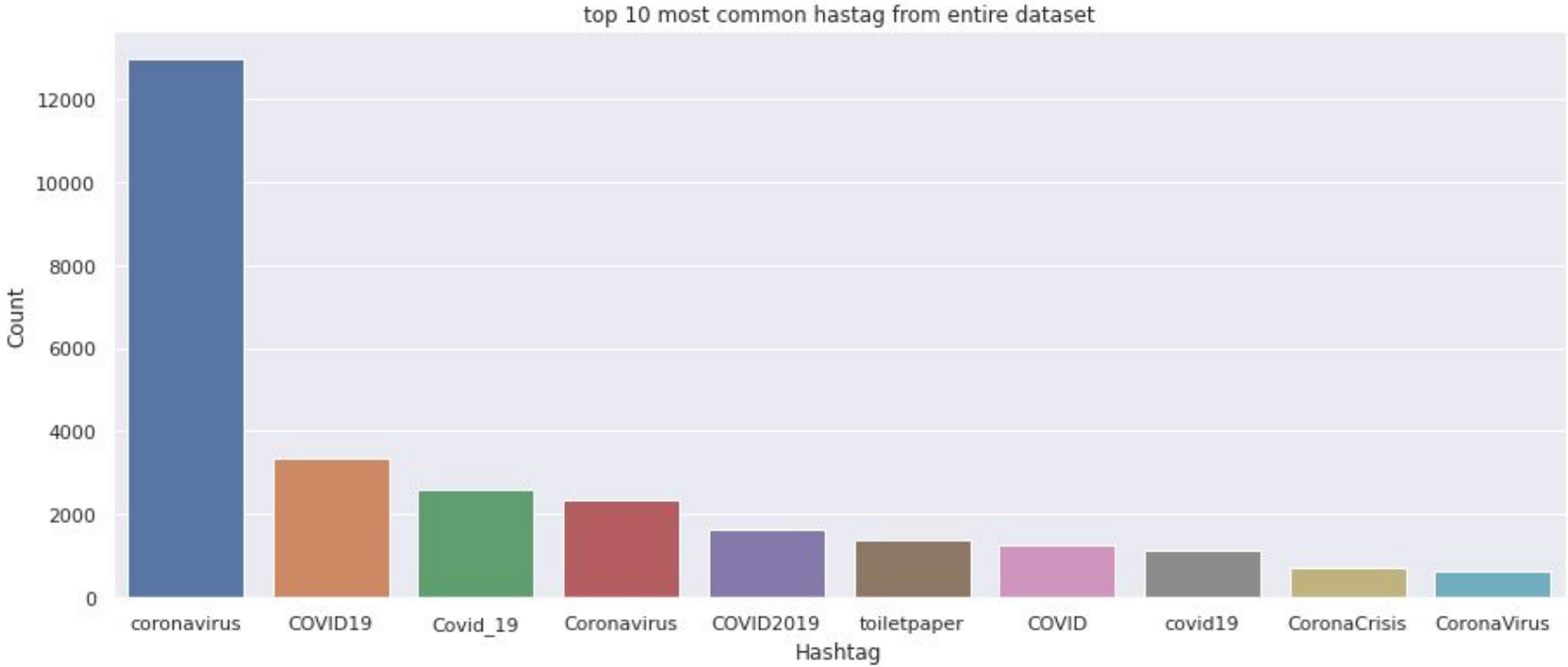
Every day tweets with Positive Sentiment is higher frequency

Sentiment feature



Positive Sentiment is most Frequent Compare than other Sentiment in Tweets

Covid, coronavirus, groceries store, help, hand sanitiser are the most common words in the entire dataset.



coronavirus and covid19 trends associated with Positive Sentiment in our dataset

Text Preprocessing

The preprocessing of the text data is an essential step as it makes the raw text ready for mining, i.e., it becomes easier to extract information from the text and apply machine learning algorithms to it.

The objective of this step is to clean noise that is less relevant to find the sentiment of tweets such as punctuation, special characters, numbers, and terms which don’t carry much weightage in context to the text.

Text Preprocessing

- removing @username
- removing special characters, numbers and punctuations (removed http:// also)
- removing short words(length equal to 2 alphabets)
- removing stop words(such as “the”, “a”, “an”, “in”)

Tokenization

Tokenization is essentially splitting a phrase, sentence, paragraph, or an entire text document into smaller units, such as individual words or terms. Each of these smaller units are called tokens.

Tokenization done using Python’s split() function.

Stemming

Stemming is the process of reducing a word to its word stem that affixes to suffixes and prefixes or to the roots of words known as a lemma.

Stemming is a rule-based process of stripping the suffixes (“ing”, “ly”, “es”, “s” etc) from a word.

Vectorization

To convert the text data into numerical data, we need some smart ways which are known as vectorization,

It is one of the simplest ways of doing text vectorization.

It creates a document term matrix, which is a set of dummy variables that indicates if a particular word appears in the document.

Countvectorizer will fit and learn the word vocabulary and try to create a document term matrix in which the individual cells denote the frequency of that word in a particular document, which is also known as term frequency, and the columns are dedicated to each word in the corpus.

Building Classification Model

we have used following model:

1. Naive Bayes Classifier
2. Stochastic Gradient Descent-SGD Classifier
3. Random Forest Classifier
4. Extreme Gradient Boosting
5. Support vector machine
6. Logistic Regression
7. Catboost

Naive Bayes Classifier

It is a classification technique based on Bayes' Theorem with an assumption of independence among predictors. In simple terms, a Naive Bayes classifier assumes that the presence of a particular feature in a class is unrelated to the presence of any other feature.

Stochastic Gradient Descent-SGD Classifier

It is a simple yet very efficient approach to fitting linear classifiers and regressors under convex loss functions such as (linear) Support Vector Machines and Logistic Regression. Even though SGD has been around in the machine learning community for a long time, it has received a considerable amount of attention just recently in the context of large-scale learning.

SGD has been successfully applied to large-scale and sparse machine learning problems often encountered in text classification and natural language processing.

Random Forest Classifier

Random Forest is a popular machine learning algorithm that belongs to the supervised learning technique. It can be used for both Classification and Regression problems in ML. It is based on the concept of ensemble learning, which is a process of combining multiple classifiers to solve a complex problem and to improve the performance of the model.

"Random Forest is a classifier that contains a number of decision trees on various subsets of the given dataset and takes the average to improve the predictive accuracy of that dataset."

Extreme Gradient Boosting

XGBoost is a decision-tree-based ensemble Machine Learning algorithm that uses a gradient boosting framework. Gradient Boosting: A special case of boosting where errors are minimized by gradient descent algorithm.

Support vector machine

It is used for Classification problems in Machine Learning.

The goal of the SVM algorithm is to create the best line or decision boundary that can segregate n-dimensional space into classes so that we can easily put the new data point in the correct category in the future. This best decision boundary is called a hyperplane.

Logistic Regression

Logistic regression is one of the most popular Machine Learning algorithms, which comes under the Supervised Learning technique. It is used for predicting the categorical dependent variable using a given set of independent variables.

Logistic regression is used for solving classification problems

Catboost

CatBoost is an open source, Gradient Boosted Decision Tree (GBDT) implementation for Supervised ML.

It provides a gradient boosting framework which attempts to solve for Categorical features using a permutation driven alternative compared to the classical algorithm.

CatBoost has gained popularity compared to other gradient boosting algorithms primarily due to the following features:

- Ordered Boosting to overcome overfitting
- Native handling for categorical features
- Using Oblivious Trees or Symmetric Trees for faster execution

Comparison of Model (Multiclass classification)

Model	Test accuracy	Train accuracy
CatBoost	0.617104	0.670251
Logistic Regression	0.616132	0.888018
Support Vector Machines	0.599004	0.898497
Stochastic Gradient Decent	0.571429	0.828155
Random Forest	0.550777	0.996203
XGBoost	0.487123	0.510554
Naive Bayes	0.477041	0.691329

The best model for this dataset would be Logistic Regression

Comparison of Model (Binary Class classification)

Model	Test accuracy	Train accuracy
Stochastic Gradient Decent	0.862609	0.934184
Logistic Regression	0.861030	0.938466
CatBoost	0.848639	0.884799
Support Vector Machines	0.836856	0.956568
Random Forest	0.823129	0.998481
Naive Bayes	0.786079	0.857099
XGBoost	0.746599	0.747213

The best model for this dataset would be Stochastic Gradient Descent

Conclusion

We have built models for multiclass classification and binary class classification and in binary class classification models perform very well compared to the multiclass classification model.

Out of Seven models, the best model for this dataset would be Logistic Regression, For multiclass classification with accuracy 88%(train data) and 61%(test data).

For binary classification, the best model for this dataset would be Stochastic Gradient Descent with accuracy 93%(train data) and 86%(test data).

Thank you!

Presented by: Rajesh Kumar Patel