Coronavirus Tweet Sentiment Analysis

##### **Anjali Tidke, Shubham Dukare**

#### Data science trainee, AlmaBetter, Bangalore.

**Abstract:**

Coronavirus disease (COVID-19) is an infectious disease caused

by a newly discovered coronavirus. It was first recorded in Wuhan, Hubei Province of China in December 2019 (W.H.O., 2020).

It was spread to up to 213 countries in six continents (Worldometer, 2020).

Coronaviruses are a large family of zoonotic viruses that cause illnesses ranging from common cold to respiratory diseases (Lab-Manager, 2020).

As the Covid-19 outbreaks rapidly all over the world day by day and also affects the lives of millions, a number of countries declared complete lockdown to check its intensity. During this lockdown period, social media platforms have played an important role in spreading information about this pandemic across the world, as people used to express their feelings through the social networks.

During the lockdown period a lot of people have chosen the Twitter to share their expression about this disease so we have been inspired to measure the human sensations about this epidemic by analyzing this huge Twitter data.

In this experiment we will try and build a classification model which will classify the sentiments of the tweets.

***Keywords***: Covid-19, coronavirus, sentiment, pandemic, lockdown, virus, respiratory, diseases.

# Problem Statement

We have provided the data in the form of a csv file.

This challenge asks to build a classification model to predict the sentiment of COVID-19 tweets. These tweets have been pulled from Twitter and manual tagging has been done then.

#### Data Description

The dataset contains the information regarding the tweets related to covid-19.

The dataset consists of tweets made by users regarding Covid-19.

We have total 5 columns in the dataset as follows

1- Username

1. ScreenName
2. Location (Location of the tweet) 4- TweetAt (Date/Time of tweet)
3. OriginalTweet (The string of the tweet)
4. Sentiment (sentiment of the tweet)

As the first two columns username and screenname column contains integer data and it will be no use for performing the modeling and analysis. We need two columns, original tweet and sentiment to perform modeling and also need other features for EDA.

# Introduction

COVID-19 is not just an infectious disease which is transmitted through contact and by small droplets produced when people cough, sneeze or talk, it is now becoming a source of depression, stress and anxiety because of misleading information posted on social media. Mental health is directly affected because of the rapid spread of false information on social media. With the current situation of lockdown and social distancing, the prime dependency of individuals is on Internet and the highest activity has been reported on social media.

Social media has become a huge part of our life. It connects people to the outer world. Social media provides a way to showcase our lives, discreetly, conveniently and on our own terms. People rely more on the posts and tweets shared on social networking sites like Twitter®, Facebook®, and Instagram®. It is anticipated that social media should guide people in getting correct and authentic information on Corona case

Twitter is a big platform to post any status, the same we have provided with the posts data of twitter related to covid-19.

The data that we worked on here has been scrapped from twitter itself and manually labeled.

There are a total of 5 sentiments in the dataset as follows.

1. Extremely Positive
2. Positive
3. Neutral
4. Negative
5. Extremely Negative

We will be building a multiclass classification model here which can classify the sentiments of the tweets.

1. **Challenges Faced**

The provided dataset was fairly well built and not many challenges occurred in this experiment.

The basic challenge that we faced was with the ‘location’feature. The feature consisted of redundant data, but represented different locations for e.g. “London” and “London,

England”. The analysis of location vs no. of tweets was affected by this anomaly.

Second challenge was the number of classes. There were 5 different classes present in the sentiment i.e. target variable. This made the algorithms work harder and took too much computational time.

1. **Project Workflow**

The project workflow is as follows:

1. Import the required libraries
2. Load the dataset into dataframe
3. Data Inspection and Exploratory data analysis.
4. Feature Engineering.
5. Model Training.
6. Performance metrics and accuracy.
7. Conclusion.
8. **Steps involved:**

### Import the required libraries

Python is a primary language used by data scientists for data science projects, because of the presence of thousands of open-source libraries that can ease and perform a data scientist task.

Here, in this project, we added some of the required libraries. Some libraries added later according to the requirement as per the dataset and python code.

### Load the dataset

We loaded the dataset from google drive into a dataframe object.

The dataframe object will be used further to process the data according to the aim of the project.

.

* **Data Inspection & Exploratory Data Analysis**

After loading the dataset, Next process is to inspect the data.

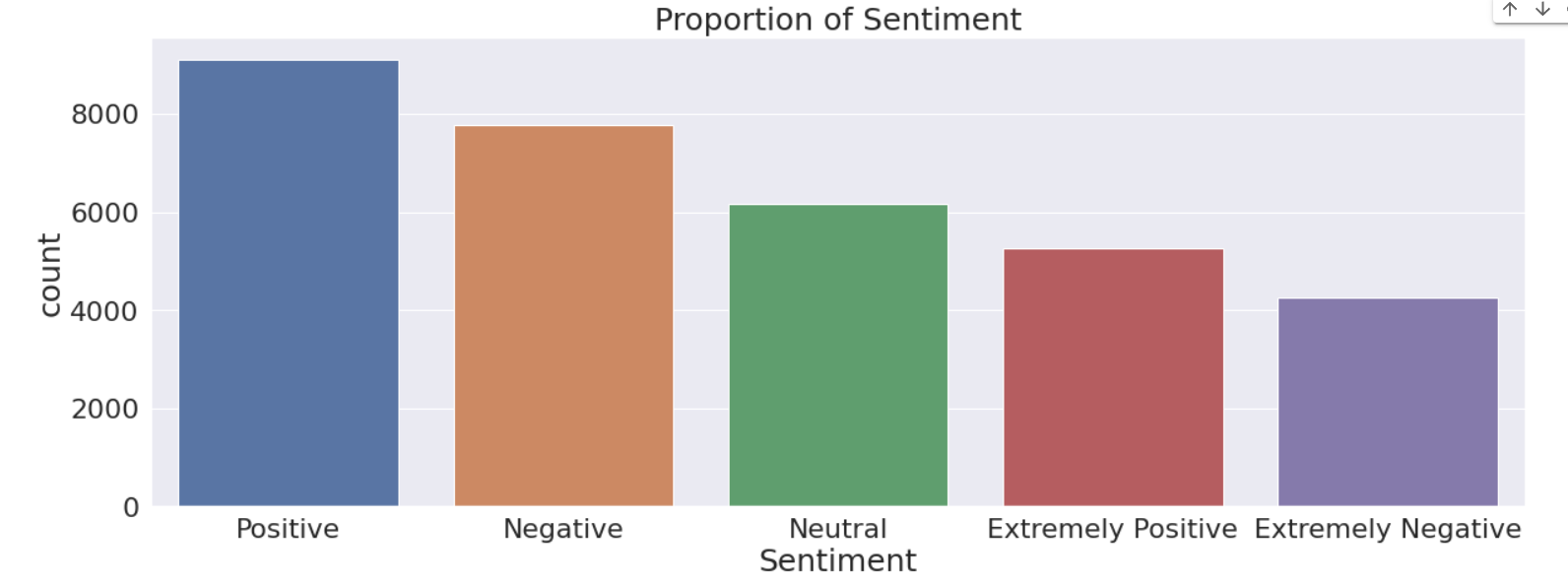
We checked the first few and last rows.

We have a total of 41157 rows and 6 columns.

Username and screenname columns have integer data type values while other columns don’t have specific data type.

Location column contains the 8590 null values.

We can see a total of 5 sentiments are in the dataset. Below image contains the proportion of sentiment.



We can see there are a maximum number of positive sentiments in the dataset followed by negative and neutral sentiments.

We also find out the original tweets according to the tweet at column. Most of the tweets are in between 17-03-2020 to 22-03-2020,

Also, by performing some EDA, Most of the tweets are from London followed by the United States.

### Feature Engineering

Here we wanted the data, especially the tweets to be properly processed before we proceeded to the modeling,

The processing of the data was done as follows-

We started by removing the web urls from the tweets. This was followed by removing the tagged usernames in the tweets because they would not be the contributors of the sentiment analysis.

Removal of punctuations and special characters was the next step,

followed by stop words removal and stemming.

### Model Training

We splitted the data into train data and test data using sklearn libraries.

We used 4 models in this project as follows

1. Logistic Regression
2. Decision Tree Classifier
3. Random Forest Classifier
4. Gradient Boosting Classifier

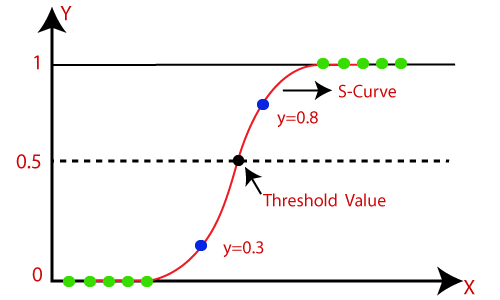
We will see each of them in details

# Algorithms:

1. **Logistic Regression:**

Logistic regression estimates the probability of an event occurring, such as voted or didn’t vote, based on a given dataset of independent variables. Since the outcome is a probability, the dependent variable is bounded between 0 and 1.

* Logistic Regression is much similar to Linear Regression except that how they are used. Linear Regression is used for solving Regression problems, whereas **Logistic regression is used for solving the classification problems**.
* In Logistic regression, instead of fitting a regression line, we fit an "S" shaped logistic function, which predicts two maximum values (0 or 1).
* The curve from the logistic function indicates the likelihood of something such as whether the cells are cancerous or not, a mouse is obese or not based on its weight, etc.
* Logistic Regression is a significant machine learning algorithm because it has the ability to provide probabilities and classify new data using continuous and discrete datasets.
* Logistic Regression can be used to classify the observations using different types of data and can easily determine the most effective variables used for the classification. The below image is showing the logistic function:

With logistic regression,

We are getting **accuracy of 77%,** precision 78%, recall 77%.

.

## Decision Tree Classifier:

* Decision Tree is a **Supervised learning technique**that can be used for both classification and Regression problems, but mostly it is preferred for solving Classification problems. It is a tree-structured classifier, where**internal nodes represent the features of a dataset, branches represent the decision rules** and **each leaf node represents the outcome.**
* In a Decision tree, there are two nodes, which are the **Decision Node** and**Leaf Node.** Decision nodes are used to make any decision and have multiple branches, whereas Leaf nodes are the output of those decisions and do not contain any further branches.
* The decisions or the test are performed on the basis of features of the given dataset.
* ***It is a graphical representation for getting all the possible solutions to a problem/decision based on given conditions.***
* It is called a decision tree because, similar to a tree, it starts with the root node, which expands on further branches and constructs a tree-like structure.
* We are getting **accuracy of 61%** with this algorithm.

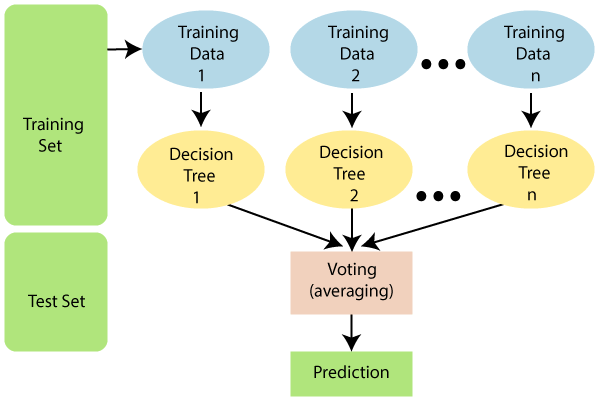
## 3. Random Forest Classifier:

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.*

As the name suggests, ***"***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***."*** Instead of relying on one decision tree, the random forest takes the prediction from each tree and based on the majority votes of predictions, and it predicts the final output.

**The greater number of trees in the forest leads to higher accuracy and prevents the problem of overfitting.**

The below diagram explains the working of the Random Forest algorithm:



Since the random forest combines multiple trees to predict the class of the dataset, it is possible that some decision trees may predict the correct output, while others may not. But together, all the trees predict the correct output. Therefore, below are two assumptions for a better Random forest classifier:

* There should be some actual values in the feature variable of the dataset so that the classifier can predict accurate results rather than a guessed result.
* The predictions from each tree must have very low correlations.

## Why use Random Forest?

Below are some points that explain why we should use the Random Forest algorithm:

* It takes less training time as compared to other algorithms.
* It predicts output with high accuracy, even for the large dataset it runs efficiently.
* It can also maintain accuracy when a large proportion of data is missing.

## How does the Random Forest algorithm work?

Random Forest works in two-phase first is to create the random forest by combining N decision tree, and second is to make predictions for each tree created in the first phase.

The Working process can be explained in the below steps and diagram:

**Step-1:** Select random K data points from the training set.

**Step-2:** Build the decision trees associated with the selected data points (Subsets).

**Step-3:** Choose the number N for decision trees that you want to build.

**Step-4:** Repeat Step 1 & 2.

**Step-5:** For new data points, find the predictions of each decision tree, and assign the new data points to the category that wins the majority votes.

We are getting an **accuracy of 71 %** with this model.

**4. Gradient Boosting Classifier**

In Gradient Boosting, each predictor tries to improve on its predecessor by reducing the errors. But the fascinating idea behind Gradient Boosting is that instead of fitting a predictor on the data at each iteration, it actually fits a new predictor to the residual errors made by the previous predictor.

We are getting an accuracy of 65% with this model.

# Model performance:

Just as we choose a certain learning model for a particular dataset, we need to choose the evaluator for type a dataset as well.

We choose the evaluator based on various particulars of the data such as balance, criticality of the results etc. Here, we focused on the balance in the dataset. The dataset had a fairly balanced distribution of classes; hence we went ahead with Accuracy as our main evaluation metric. We calculated precision and recall as well just to keep an eye on those scores as well.

* **Confusion Matrix**- Confusion Matrix is used to know the performance of a Machine learning classification. It is represented in a matrix form. Confusion Matrix has 4 terms to understand True Positive (TP), False Positive (FP), True Negative (TN) and False Negative (FN).
* **Precision and Recall**- ‘Precision’ is the ratio of correct positive predictions to the overall number of positive predictions: TP/TP+FP and ‘Recall’ is the ratio of correct positive predictions to the overall number of positive examples in the set: TP/FN+TP.

#### Accuracy-

Accuracy is given by the number of correctly classified examples divided by the total number of classified examples. In terms of the confusion matrix, it is given by: TP+TN/TP+TN+FP+FN

# Conclusion:

# 

* Dataset has a total of 41157 rows and 6 columns.
* We dropped the null values from the location column.
* "UserName" and "ScreenName" have only integer values. Which does not provide any meaningful information.
* For modeling, we required only two columns "OriginalTweet" and "Sentiment".
* We removed special characters from the original tweets column.
* After dropping and adding a new column, now we have 7 columns and 32567 rows.
* There are five types of sentiments - Extremely Positive, Positive, Extremely Negative, Negative and Neutral.
* We replaced Extremely positive to positive and Extremely negative to negative columns.
* The graphical representation of the top 10 locations shows us that most of the tweets came from London followed by the United States.
* Total we used four models here -

1. Logistic Regression Model

2. Decision Tree Classifier

3. Random Forest Classifier

4. Gradient Boosting Classifier.

We can see the better accuracy is with Logistic Regression that is 77%.

We can use the Logistic Regression Model.

#### References-

* 1. GeeksforGeeks
  2. Analytics Vidhya
  3. Almabetter