# Coronavirus Tweet Sentiment Analysis

**Vikramaditya Sah**
**Data science trainee,**
**AlmaBetter, Bangalore**

## Abstract:

Twitter is an American micro blogging and social networking service on which users post and interacts with messages known as "tweets".
We all know Covid-19 have greatly impacted our life in many ways. There have been good news and bad news all over the world and people have been responding to it on twitter. So my project is about classifying the tweets into positive, negative and neutral using different classification models. At the end we compare their performance and find which one is better.

***Keywords: machine learning, twitter, tweet, twitter, classification, supervised learning, NLP, sentiment analysis***

## 1.Problem Statement

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

The names and usernames have been given codes to avoid any privacy concerns.

You are given the following information:

Location

Tweet At

Original Tweet

Label.

Perform sentiment analysis using multiple classification models to predict the sentiment of the tweet and compare the evaluation metrics for all of them to find the best model.

## 2. Introduction

There have been a lot of incidents, both good and bad all around the world. People have written their views on those situations on many social media sites. One of them is twitter, and we have performed a sentiment analysis on thousands of tweets in our data set to come up with a classification model and segregate those tweets into – positive, negative and neutral.

Our data has the following features –

- UserName
- ScreenName,
- Location
- TweetAt
- OriginalTweet
- Sentiment.

Our goal here is to perform multiple techniques of classification Analysis to predict the sentiment and report our findings.

# 3. Steps involved:

- **Data overview**
  After we load our data, we simply take a look at it and perform sanity and null value checks. In our case, the only null values we found was in Location but since the column wasn't important for our project, we simply ignored them.

- **Exploratory Data Analysis**
  After overview, we performed EDA which consisted of seeing the distribution of the data, that is tweet count on the basis of sentiment and having a look at from which locations are we getting maximum tweets.

- **Text pre processing**
  Before we can fit any models we have to make certain changes to our tweets to ensure we get the best results.
  We will make comparisons at every step to get a better understanding of what's going on.

  **Orignal tweet** - As news of the region ' s first confirmed COVID-19 case came out of Sullivan County last week, people flocked to area stores to purchase cleaning supplies, hand sanitizer, food, toilet paper and other goods, @Tim_Dodson reports https://t.co/cfXch7a2lU

1. REMOVING LINKS/URLs

   As news of the region ' s first confirmed COVID-19 case came out of Sullivan County last week, people flocked to area stores to purchase cleaning supplies, hand sanitizer, food, toilet paper and other goods, reports

2. REMOVING USERNAMES/@USER

   As news of the regionÂ' s first confirmed COVID-19 case came out of Sullivan County last week, people flocked to area stores to purchase cleaning supplies, hand sanitizer, food, toilet paper and other goods, reports

3. REMOVING PUNCTUATIONS, NUMBERS AND SPECIAL CHARACTERS

   As news of the region s first confirmed COVID case came out of Sullivan County last week people flocked to area stores to purchase cleaning supplies hand sanitizer food toilet paper and other goods reports

4. REMOVING STOP WORDS, SHORT WORDS AND STEMMING
   **Stemming** means bring all the words in it's root form.

   news region first confirm covid case came sullivan counti last week peopl flock area store purchas clean suppli hand sanit food toilet paper good report

- **Fitting different models**
  For modeling we tried various regression algorithms like:

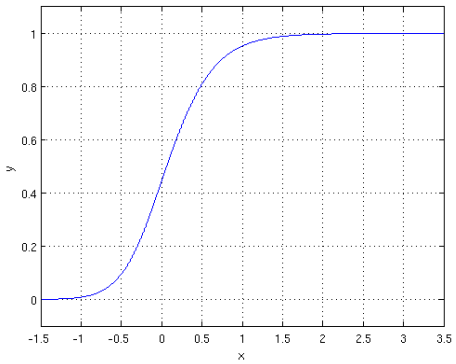
1. LOGISTIC REGRESSION
2. RANDOM FOREST CLASSIFIER
3. XGBoost CLASSIFIER
4. KNN CLASSIFIER
5. SVM CLASSIFIER

- **Comparing different models**
  After we were done with fitting all the models, we compared their metrics with each other to figure out the best regression technique for our dataset.

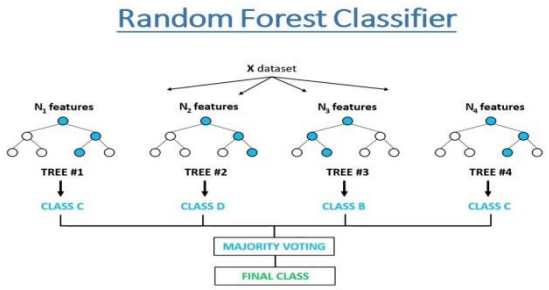
# 4. Algorithms:

- **Logistic regression**
  Logistic Regression is actually a classification algorithm that was given the name regression due to the fact that the mathematical formulation is very similar to linear regression.
  The function used in Logistic Regression is sigmoid function or the logistic function given by:

  $f(x)= 1/1+e ^(-x)$




- **Random forest classifier**
  Random Forest is a bagging type of Decision Tree Algorithm that creates a number of decision trees from a randomly selected subset of the training set, collects the labels from these subsets and then averages the final prediction depending on the most number of times a label has been predicted out of all.

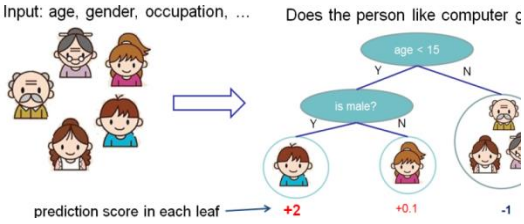


- **XGBoost CLASSIFIER**

  To understand XGBoost we have to know gradient boosting beforehand.

  - **Gradient Boosting-**

    Gradient boosted trees consider the special case where the simple model is a decision tree



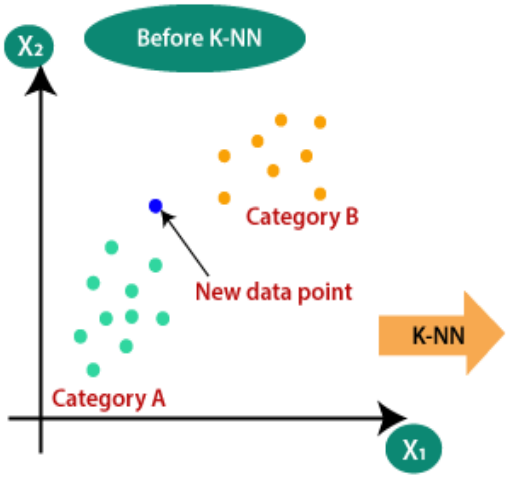

    In this case, there are going to be 2 kinds of parameters P: the weights at each leaf, w, and the number of leaves T in each tree (so that in the above example, T=3 and w=[2, 0.1, -1]).

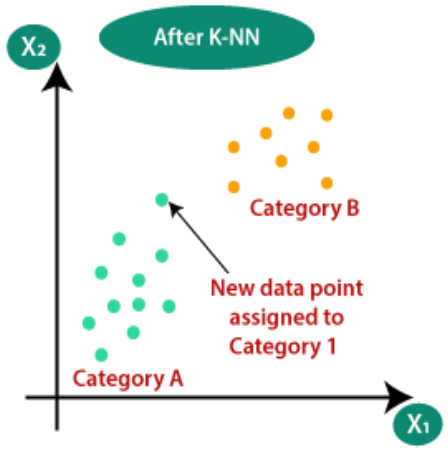
    When building a decision tree, a challenge is to decide how to split a current leaf. For instance, in the above image, how could I add another layer to the (age > 15) leaf? A 'greedy' way to do this is to consider every possible split on the remaining features (so, gender and occupation), and calculate the new loss for each split; you could then pick the tree which most reduces your loss.

**XGBoost** is one of the fastest implementations of gradient boosting. trees. It does this by tackling one of the major inefficiencies of gradient boosted trees: considering the potential loss for all possible splits to create a new branch (especially if you consider the case where there are thousands of features, and therefore thousands of possible splits). XGBoost tackles this inefficiency by looking at the distribution of features across all data points in a leaf and using this information to reduce the search space of possible feature splits.

- **KNN classifier**

  Suppose there are two categories, i.e., Category A and Category B, and we have a new data point x1, so this data point will lie in which of these categories. To solve this type of problem, we need a K-NN algorithm. With the help of K-NN, we can easily identify the category or class of a particular dataset. Consider the below diagram:

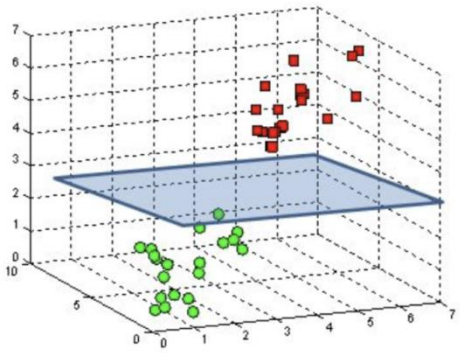





- **Support Vector Machine Classifier:**
  SVM is used mostly when the data cannot be linearly separated by logistic regression and the data has noise. This can be done by separating the data with a hyperplane at a higher order dimension.
  In SVM we use the optimization algorithm as:

$$\min_{\xi,w,b} \quad \frac{1}{2}||w||^2 + C\sum_{i=1}^{m}\xi_i$$
$$\text{s.t} \quad y^{(i)}\left(w^T x^{(i)} + b\right) \geq 1 -$$
$$\xi_i \geq 0;\ i = 1, \ldots, m.$$

where $C$ is a cost parameter and $\xi_i$'s are slack variables.



We use hinge loss to deal with the noise when the data isn't linearly separable.

Kernel functions can be used to map data to higher dimensions when there is inherent non linearity.

# 5. Evaluation metrics:

- **Precision/Recall-:**
  Precision is the ratio of correct positive predictions to the overall number of positive predictions : TP/TP+FP
  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

- **Root Mean Squared Error(RMSE)**
  Model F1 score represents the model score as a function of precision and recall score. F-score is a machine learning model performance metric that gives equal weight to both the Precision and Recall for measuring its performance in terms of accuracy, making it an alternative to Accuracy metrics.

  F1 Score = 2* Precision Score * Recall Score/ (Precision Score + Recall Score/)

# 6. Conclusion:

Starting with loading the data we did data overview and EDA. Then we performed text preprocessing.
After that we split and fitted many classification models on the data.
In the end we found out which is the most successful classifier for our dataset which was logistic regression.

**References-**


1. statisticshowto.com
2. Edx
3. Analytics Vidhya
4. Almabetter