Sentiment Analysis: Predicting sentiment of Covid-19 Tweets

By Harshavardhan Shete

Twitter is a microblogging site in which users can post updates (tweets) to friends (followers). It has become an immense dataset of *sentiments.* The goal of sentiment analysis is to determine opinions, emotions, and attitudes presented in the source material. In tweet sentiment analysis, opinions in messages can be typically categorized as positive, negative, and neutral. Now-a-days the popularity of the Twitter platform has increased tremendously and because huge userbase and worldwide spread, it has become the opinion platform of the entire world. Thus, to understand people's perception about some topic or issues, sentiment analysis of the tweets related to that is a useful tool. It is heavily used during elections or to understand the perception of people newly launched product/ service by any company or to understand the perception of people about an IPO or shares of a particular company etc.

Problem Statement:

The task is to build a classification model to predict the Covid-19 related tweets. The tweets have been pulled from Twitter and manual tagging has been done. The dataset provided has the following columns:

1. Location: The location at which the tweet was made
2. TweetAt: The date on which the tweet was made
3. OriginalTweet: This is the actual text of the tweet
4. Sentiment: This is the sentiment of the tweet, which is manually tagged

Extremely Positive, Positive, Extremely Negative, Negative, and Neutral are the tags provided in the dataset, thus the problem is of multi-class classification.

Solutioning

The solutioning of the problem involved the following steps:

1. Exploratory Data Analysis
2. Data Pre-processing
3. Model training and performance comparison

1. **Exploratory Data Analysis**
   Exploratory Data Analysis (EDA) as the name suggests, is used to analyze and investigate datasets and summarize their main characteristics, often employing data visualization methods. It helps determine how best to manipulate data sources to get the answers you need, making it easier for data scientists to discover patterns, spot anomalies, test a hypothesis, or check assumptions. It also helps to understand the relationship between the variables (if any) and it will be useful for feature engineering. It helps to understand data well before making any assumptions, to identify obvious errors, as well as better understand patterns within data, detect outliers, anomalous events, find interesting relations among the variables.

Exploration and visualizations performed with Covid-19 Tweet dataset:

a. Sentiment-wise tweet count
b. Date-wise tweet count
c. Top 15 locations with the highest number of tweets
d. Handling null values:
   Only the
   'Location' column has 20.9%`null values and all other columns have 0 null values. As the ' Location' column will not be used for analysis, need not worry about it.
e. Checking for duplicate records:
   There are no duplicate records

2. Data Pre-processing

Data pre-processing is the process of transforming the raw data into a usable and or understandable format. Major tasks in data pre-processing are data cleaning, data integration, data reduction, and data transformation.

Steps involved in data pre-processing (Source: medium.com)

It is important to note that not all the time you will need to perform all the steps, it depends upon where and how do you want to use the data.

Steps were taken to pre-process the data for tweet sentiment analysis:

a. Removing Twitter handles/ user names
b. Removing URL links
c. Removing # symbols and retaining the tags
d. Removing Punctuations and stop words
e. Removing short words
f. Tokenization and stemming

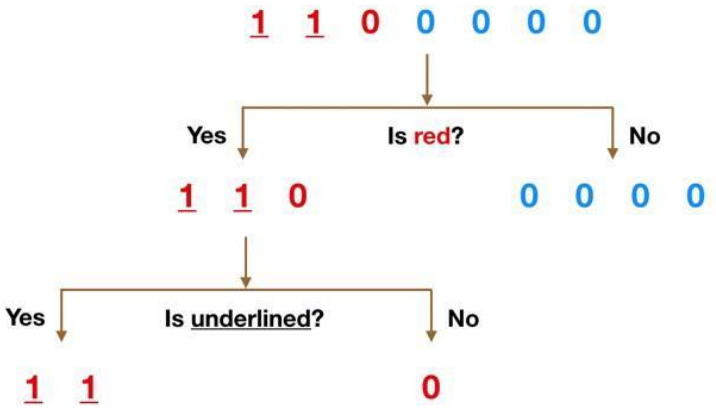
Tokenization is the process of breaking down the given text into the smallest unit in a sentence called token and Stemming is the process of finding the root of words.

3. Model training and performance metrics

Various multiclass classification algorithms were tried with given data and they were compared using certain performance parameters to choose the best one. The algorithms tried are- Decision tree classifier, Support vector machine classifier, K-nearest neighbor classifier, Naïve-bayes classifier, Random forest classifier, XGBoost classifier, Stochastic gradient descent classifier, and the performance parameters used were accuracy, precision, and recall.

a. Decision tree classifier

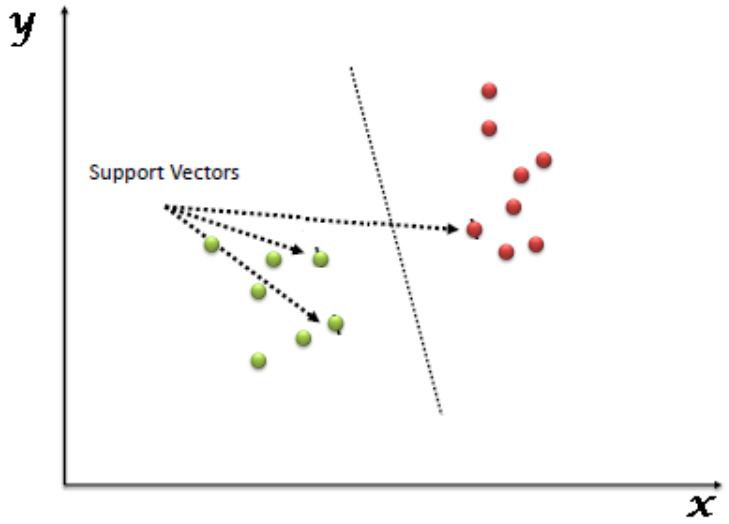
Decision tree classifier is a systematic approach for multiclass classification. It poses a set of questions to the dataset (related to its attributes/features). The decision tree classification algorithm can be visualized on a binary tree. On the root and each of the internal nodes, a question is posed and the data on that node is further split into separate records that have different characteristics. The leaves of the tree refer to the classes in which the dataset is split.



An image depicting functioning of Decision tree classifier.

b. Support vector machine classifier

"Support Vector Machine" (SVM) is a supervised machine learning algorithm that can be used for both classification or regression challenges. However, it is mostly used in classification problems. In the SVM algorithm, we plot each data item as a point in n-dimensional space (where n is a number of features you have) with the value of each feature being the value of a particular coordinate. Then, we perform classification by finding the hyper-plane that differentiates the two classes very well (look at the below snapshot).



Support vectors belonging to two different classes.

c. K-nearest neighbor classifier

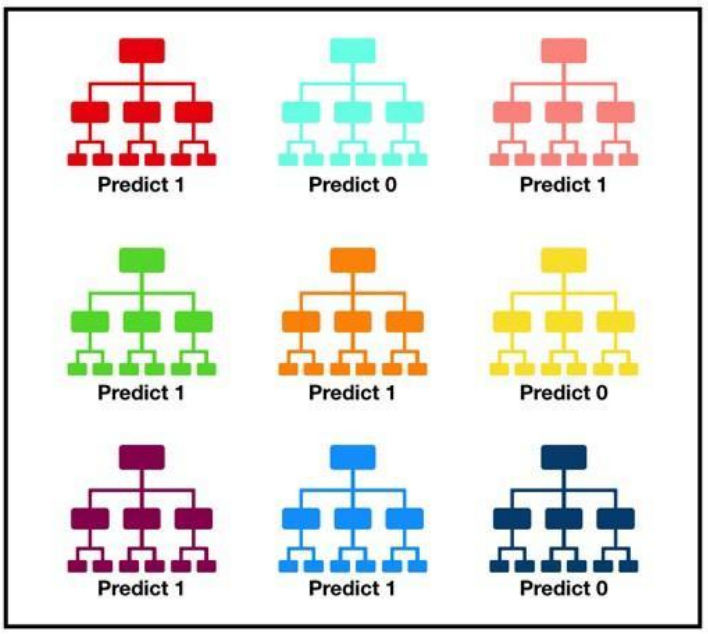
KNN or k-nearest neighbors is the simplest classification algorithm. This classification algorithm does not depend on the structure of the data. Whenever a new example is encountered, its k nearest neighbors from the training data are examined. Distance between two examples can be the Euclidean distance between their feature vectors. The majority class among the k nearest neighbors is taken to be the class for the encountered example.

d. Naïve-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. The Naive Bayes model is easy to build and particularly useful for very large data sets. Along with simplicity, Naive Bayes is known to outperform even highly sophisticated classification methods.

e. Random forest classifier

Random forest, like its name implies, consists of a large number of individual decision trees that operate as an ensemble. Each individual tree in the random forest spits out a class prediction and the class with the most votes becomes our model's prediction. The fundamental concept behind random forest is a simple but powerful one — the wisdom of crowds.



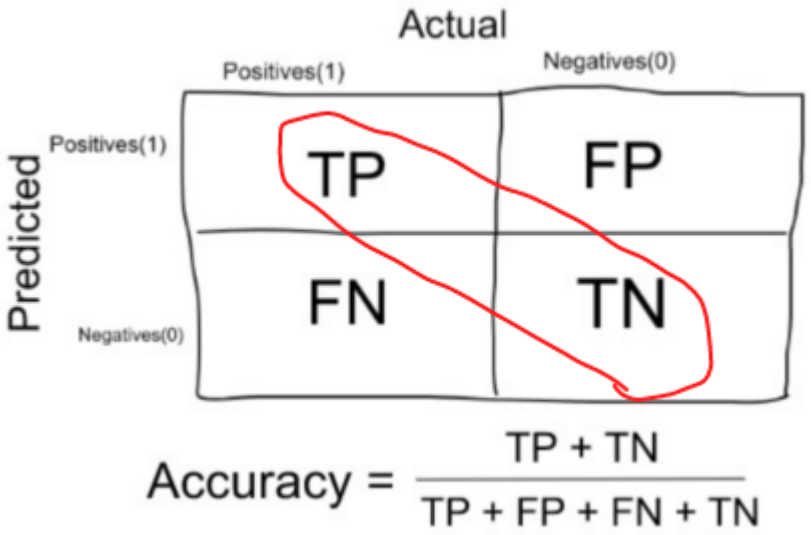
Visualization of Random forest model making a prediction.

f. XGBoost classifier
XGBoost is short for "eXtreme Gradient Boosting." The "eXtreme" refers to speed enhancements such as parallel computing and cache awareness that makes XGBoost approximately 10 times faster than traditional Gradient Boosting. In addition, XGBoost includes a unique split-finding algorithm to optimize trees, along with built-in regularization that reduces overfitting. Generally speaking, XGBoost is a faster, more accurate version of Gradient Boosting.

g. Stochastic gradient descent classifier
Stochastic Gradient Descent (SGD) is a simple yet efficient optimization algorithm used to find the values of parameters/coefficients of functions that minimize a cost function. Stochastic Gradient Descent (SGD) classifier basically implements a plain SGD learning routine supporting various loss functions and penalties for classification. Advantages of SDG are efficiency and ease of implementation (lots of opportunities for code tuning) and its disadvantages are that it requires a number of hyperparameters (such as regularization parameter and the number of iterations) and SDG are sensitive to feature scaling.

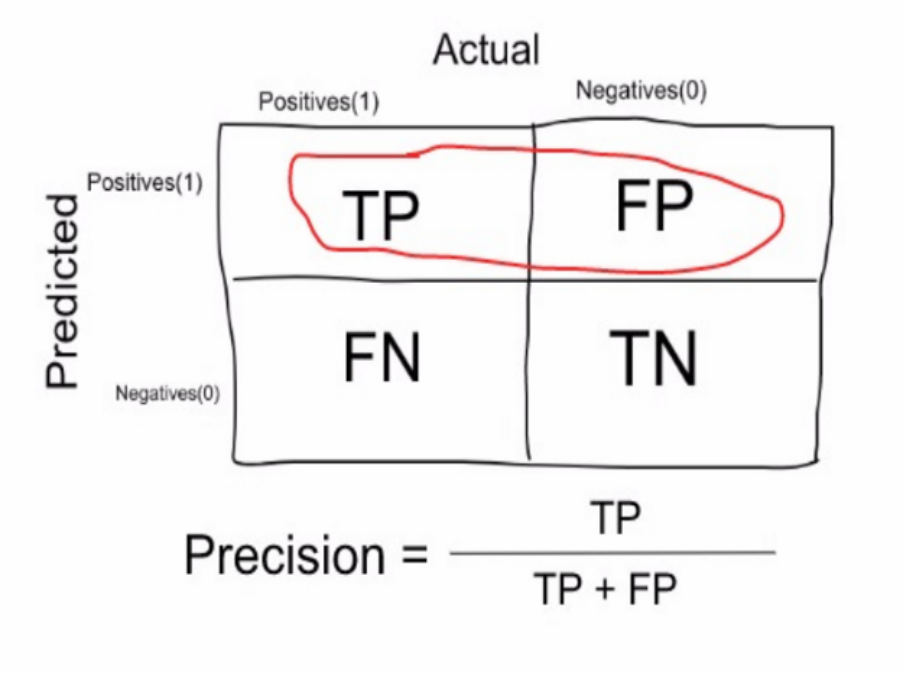
Performance Metrics:

a. Accuracy:
Accuracy in classification problems is the number of correct predictions made by the model over all kinds of predictions made. Accuracy is a good measure when the target variable classes in the data are nearly balanced. Accuracy should never be used as several measures when the target variable classes in the data are a majority of one class.



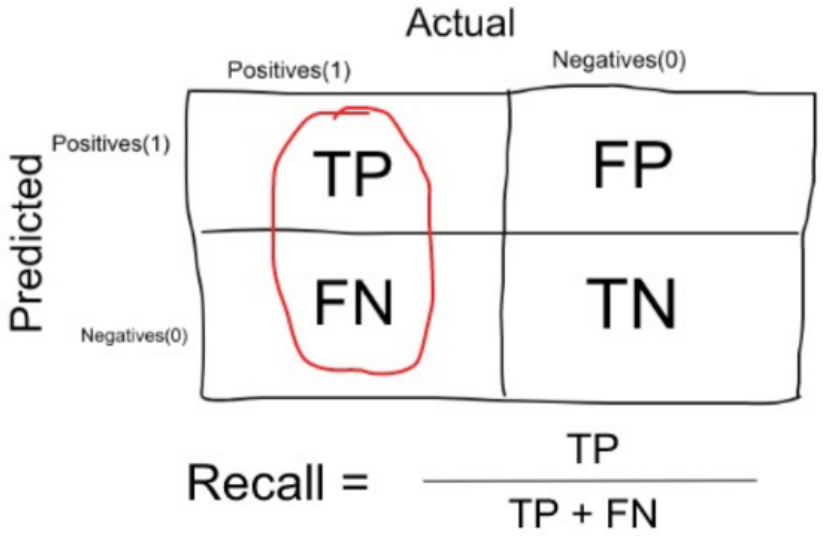
Accuracy calculation

b. Precision:
It is the number of correct positive results divided by the number of positive results predicted by the classifier.



Precision Calculation

c. Recall or Sensitivity:
It is the number of correct positive results divided by the number of *all* relevant samples (all samples that should have been identified as positive).



Recall calculation

Final verdict:

After comparing the results of various classifiers for the tweet sentiment analysis, it is concluded that Support Vector Machine Classifier has performed best amongst all the models tried.

Sources

- Towardsdatascience.com
- Ibm.com
- Analyticsvidhya.com
- Geeksforgeeks.org