# Analysis of Covid-19 Sentiment

Yuming Chang*

CSE 8803 Urban Computing Course Project
*ychang394@gatech.edu

## Introduction

- It is important to analysis the sentiment of the tweet text to analysis the opinion on Covid-19 and to better provide the useful information and customized service
- There are lots of active users of tweet enable use to acquire the sufficient data
- The Natural Language Process , machine learning and deep learning machine enable us to do the text analysis and prediction

## Problem Definition

Using the Natural Language Process(NLP) method, sentiment analysis can automatically extract the information from the text and classify the sentiment from those text. In order to use the machine learning model to do the sentiment analysis, we need to convert the text into numeric values using Natural Language Process methods. So the problem in this project will be to convert the text information into a numeric value and train and test models using those data.

## Data

Data used in this project acquired from: https://github.com/rajeshmore1/Capstone-Project-2. The original dataset contain 41157 rows of tweet text data and five categories of sentiment. In order to get better results, I merge the positive and extremely positive to positive, negative and extremely negative to negative. And in this project, I am going to focus on the negative and positive tweet classification and prediction. So I also omit the neutral class. The I assigned the numeric value of negative and positive: Negative-0, Positive, Neutral-1, and in order to save the training time, I just choose 10000 records from each class, the number of processed dataset is 20000 and its distribution is shown in Fig 1

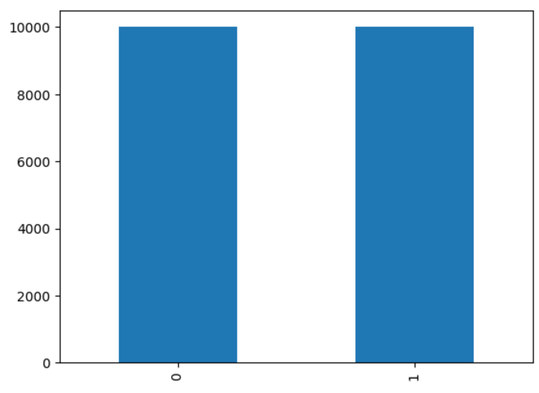


**Fig 1.** Distribution of different class after processing (0: negative, 1: positive, neutral)

## Method

- Clean the dataset: The tweet data contain special characters and links, we need to remove those before converting the text data into the numeric matrix, we also need to covert the Uppercase to Lowercase to exclude the influence of them
- Covert tweet text to numeric matrix using TF-IDF method: In order to using machine learning method to do the sentiment analysis, we need to convert the text to numeric value. And TF-IDF is widely used statistical method to vectorize text data. In this project, I use the max features=4000.
- Random Forest model: Random Forest combine multiple decision tree on sub datasets. In this project I used number of estimator as 100, criterion as Gini impurity. And random forest model will serve as baseline model in this project
- Naive Bayes: Based on the Bayes Rule, Naive Bayes model is one of statistical machine learning model to do the classification task. In this project, I am using the default parameters of the function of MultinomialNB from Skiti-learn and set the random state to 42.
- Support Vector Machine: SVM is one of the supervised learning methods to do classification. By mapping the data to high dimensional space,
the algorithm draws the hyperplane to separate different class data.
- XG Boost: XG Boost is an ensemble learning method that can handle large dataset and missing values and It is widely used in classification and regression. So in this project, the XG Boost classifier for boost library will be used.
- Convolutional Neural Network: CNN is the deep learning method model. Due to the convolutional layer of the CNN model, we can also use the CNN model to do text classification in the natural language process area. In this project, I used TensorFlow to develop the CNN model. I found the 3 convolutional layer have the best performance. I also set the optimization function as Adam. The learning rate is set to 0.0001 and other hyperparameters of the optimization function is default. And train the model using 10 epochs.
- Evaluation Method: Confusion Matrix, accuracy, precision, recall, F-1 score and training time.

## Results and Analysis

### Confusion Matrix

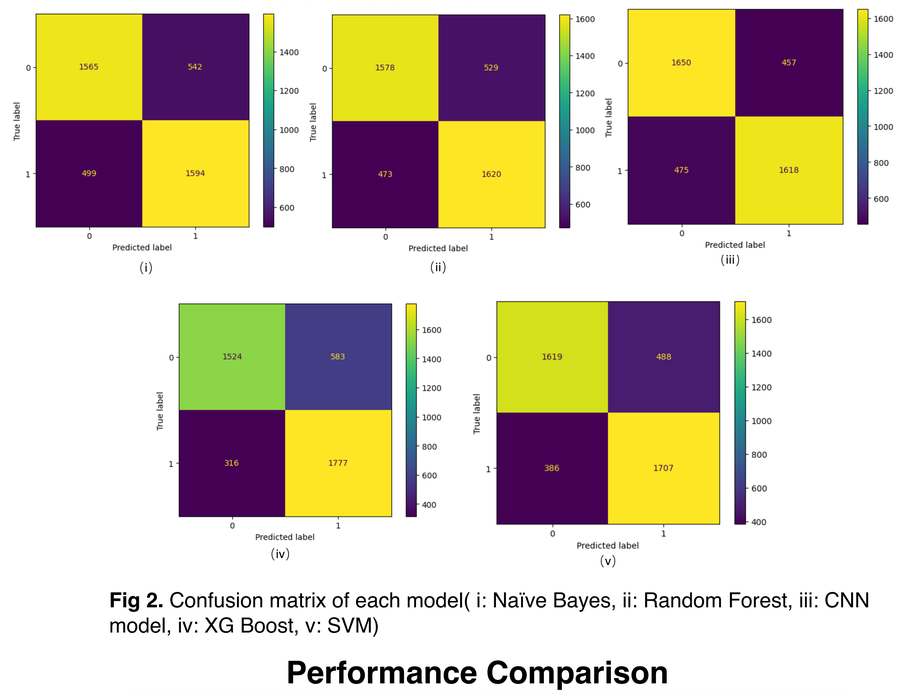


**Fig 2.** Confusion matrix of each model( i: Naïve Bayes, ii: Random Forest, iii: CNN model, iv: XG Boost, v: SVM)

### Performance Comparison

| Model | accuracy | precision | recall | F-1 score | Training Time (second) |
|---|---|---|---|---|---|
| Naïve Bayes | 0.75 | 0: 0.76<br>1: 0.75 | 0: 0.74<br>1: 0.76 | 0: 0.75<br>1: 0.75 | 0.34 |
| Random Forest | 0.76 | 0: 0.77<br>1: 0.75 | 0: 0.76<br>1:0.77 | 0: 0.76<br>1: 0.76 | 41.47 |
| XG Boost | 0.79 | 0: 0.83<br>1:0.75 | 0: 0.72<br>1:0.85 | 0:0.77<br>1: 0.80 | 118.38 |
| CNN | 0.78 | 0:0.78<br>1:0.78 | 0: 0.78<br>1: 0.77 | 0: 0.78<br>1: 0.78 | 543.86 |
| SVM | 0.79 | 0: 0.81<br>1: 0.78 | 0: 0.77<br>1:0.82 | 0: 0.79<br>1: 0.80 | 1267.70 |

## Conclusions

- From the performance comparison, we can see the in the project, the SVM model and XG Boost model have the best performance, which is around 0.79
- All other models have the accuracy above Baseline Model (Naïve Bayes: 0.75)
- SVM model has longest training time and Naïve Bayes Model has the fewest training time.
- We may need other parameters other than just convert the text data to vector to generate the better results.