Government policies

and

public sentiments

2022S BIA 678-A

Big Data Technologies

Team 07 - A

Archana Kalburgi: 10469491

Table of contents

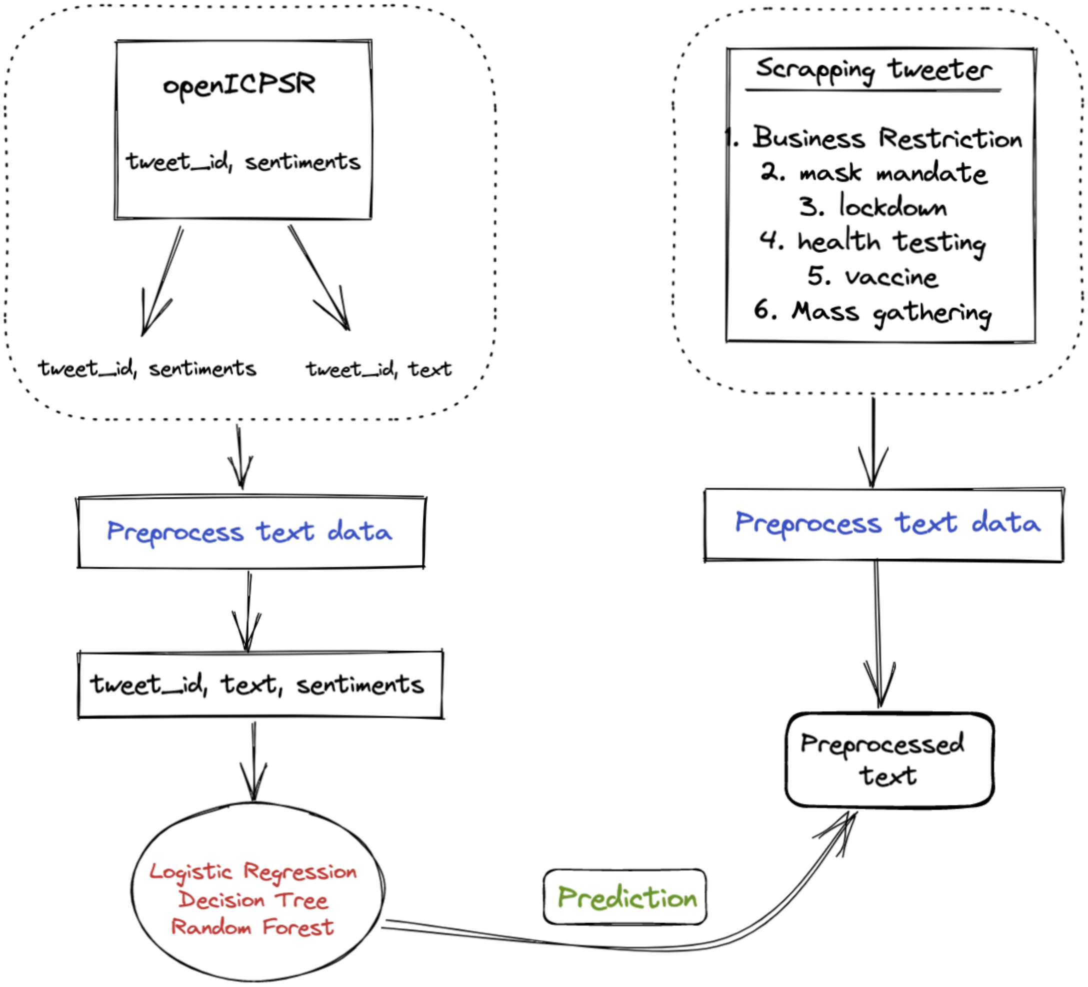
Introduction

Globally, governments have responded to the rapid growth of the Coronavirus pandemic by enacting different nationwide measures against it. In order to fight the pandemic situation, governments have framed new policies related to health testing, vaccines, mask mandates, mass gatherings and others. And public sentiment is an influential indicator of crisis response. With social media platforms, global populations have access to previously unmatched communication channels. We are using data from Twitter to gain an insight into the emotions of people over a particular category of policy.

The goal here is to predict how a certain category of a policy is going to be perceived on Twitter by social media users in the future. For which I have collected the text data (scraped from Twitter using SNScrape) to train machine learning models and record their accuracies.

In this project I have implemented logistic regression, Naive Bayes classifier and decision tree and analyzed the effect of size of data both on the time required to train the models and its effect on accuracy of the models. I have analyzed the effect of parallel computation done using Databricks. I have made use of Google Storage to store the data used to train the model.

Workflow


openICPSR
tweet_id, sentiments
tweet_id, sentiments
tweet_id, text
Scrapping tweeter
1. Business Restriction
2. mask mandate
3. lockdown
4. health testing
5. vaccine
6. Mass gathering
Preprocess text data
tweet_id, text, sentiments
Preprocess text data
Preprocessed text
Logistic Regression
Decision Tree
Random Forest
Prediction

Challenges

Due to non availability of data that could be used to train the models I have made use of two sources to collect data from.

1. First, from the Inter-university Consortium for Political and Social Research (ICPSR), openICPSR which is a self-publishing repository for social, behavioral, and health sciences research data.
2. Twitter social media platform

I have extracted the tweets (text data) using the tweet ID that we had collected from openICSPR. It was then necessary to merge this data with the data that contained the sentiment labels. By doing the above steps I was able to prepare the data that could be utilized to train the models.

Further I have scraped data based on the policies since the tweets in the openICSR dataset were generalized based on the keywords “Covid”, “Corona'' and “Wuhan”. This was achieved by incorporating the keywords based on the policies in the snscrape module.

Description of dataset and data statistics

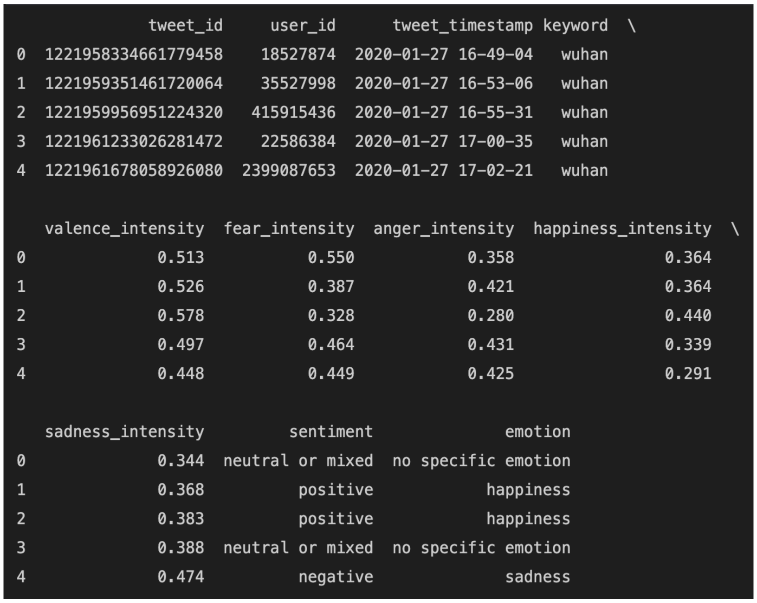
Dataset from openICPSR

The dataset I picked was a large dataset for researchers to discover public conversation on Twitter surrounding the COVID-19 pandemic.

It has over 198 million Twitter posts from more than 25 million unique users using four keywords: “corona”, “wuhan”, “nCov” and “covid”. Each tweet is being labeled with five quantitative emotion attributes indicating the degree of intensity of the valence or sentiment (from 0: very negative to 1: very positive), and the degree of intensity of fear, anger, happiness and sadness emotions (from 0: not at all to 1: extremely intense), and two qualitative attributes indicating the sentiment category (very negative, negative, neutral or mixed, positive, very positive) and the dominant emotion category (fear, anger, happiness, sadness, no specific emotion) the tweet is mainly expressing.

I have taken a subset of the data described above. In order to get the labeled sentiments, I have taken advantage of the data that is central to the United States which has 929337 data points. A snippet of the same can be found below:

A few rows of the merged dataset are as follows :



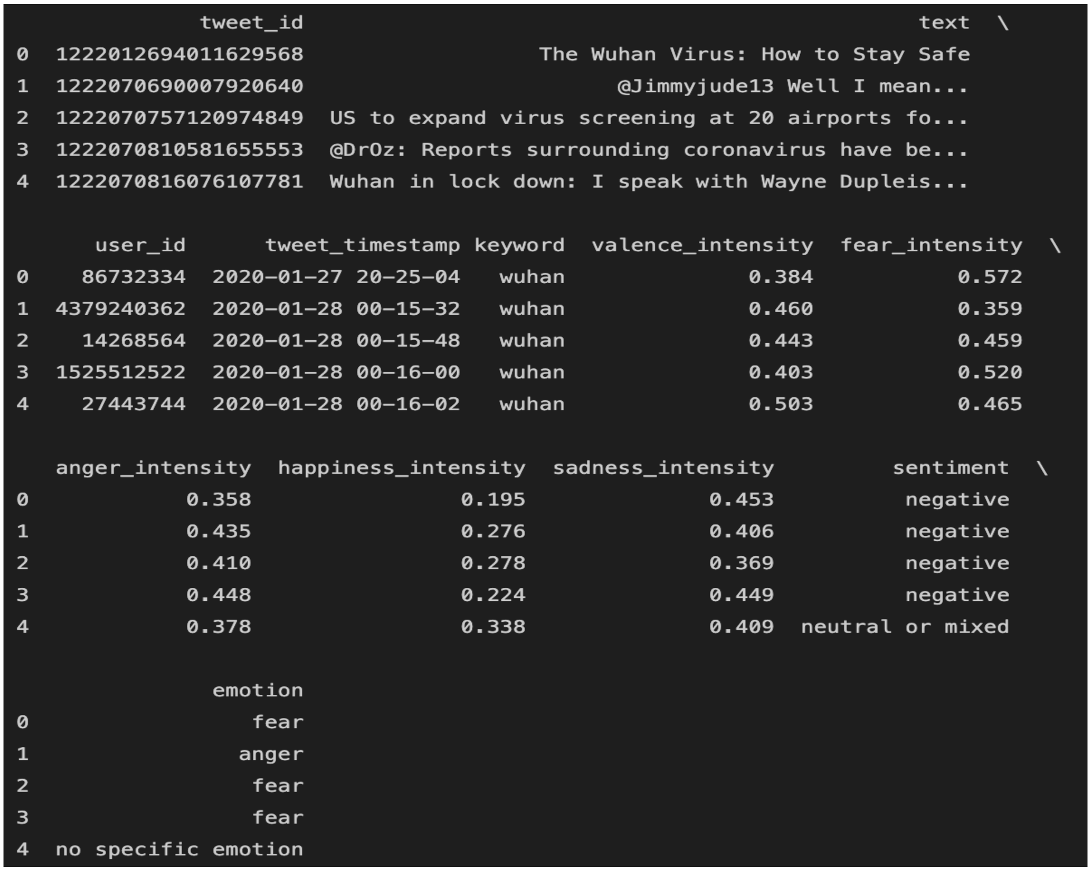
```
              tweet_id     user_id      tweet_timestamp keyword  \
0  1221958334661779458    18527874  2020-01-27 16-49-04   wuhan
1  1221959351461720064    35527998  2020-01-27 16-53-06   wuhan
2  1221959956951224320   415915436  2020-01-27 16-55-31   wuhan
3  1221961233026281472    22586384  2020-01-27 17-00-35   wuhan
4  1221961678058926080  2399087653  2020-01-27 17-02-21   wuhan

   valence_intensity  fear_intensity  anger_intensity  happiness_intensity  \
0              0.513           0.550            0.358                0.364
1              0.526           0.387            0.421                0.364
2              0.578           0.328            0.280                0.440
3              0.497           0.464            0.431                0.339
4              0.448           0.449            0.425                0.291

   sadness_intensity         sentiment              emotion
0              0.344  neutral or mixed  no specific emotion
1              0.368          positive            happiness
2              0.383          positive            happiness
3              0.388  neutral or mixed  no specific emotion
4              0.474          negative              sadness
```

fig1

As illustrated in fig1, only tweet_ids were present, which are unique identifiers assigned by Twitter to each tweet. Using this information, I was able to retrieve the tweet text from Twitter[4]. There were some rows that I had to drop since they could not be located because the user had deleted those tweets. The total number of rows was 909414 after we removed the dropped rows. Following the retrieval of the tweets, we had to merge the two datasets based on the column tweet_id.



```
             tweet_id                                              text  \
0  1222012694011629568                 The Wuhan Virus: How to Stay Safe   
1  1222070690007920640                       @Jimmyjude13 Well I mean...   
2  1222070757120974849  US to expand virus screening at 20 airports fo...   
3  1222070810581655553  @DrOz: Reports surrounding coronavirus have be...   
4  1222070816076107781  Wuhan in lock down: I speak with Wayne Dupleis...   

      user_id      tweet_timestamp keyword  valence_intensity  fear_intensity  \
0    86732334  2020-01-27 20-25-04   wuhan              0.384           0.572   
1  4379240362  2020-01-28 00-15-32   wuhan              0.460           0.359   
2    14268564  2020-01-28 00-15-48   wuhan              0.443           0.459   
3  1525512522  2020-01-28 00-16-00   wuhan              0.403           0.520   
4    27443744  2020-01-28 00-16-02   wuhan              0.503           0.465   

   anger_intensity  happiness_intensity  sadness_intensity         sentiment  \
0            0.358                0.195              0.453          negative   
1            0.435                0.276              0.406          negative   
2            0.410                0.278              0.369          negative   
3            0.448                0.224              0.449          negative   
4            0.378                0.338              0.409  neutral or mixed   

               emotion  
0                 fear  
1                anger  
2                 fear  
3                 fear  
4  no specific emotion  
```

fig2

Exploratory Data Analysis

Fig3 shows the Lengths of the tweets text over the train dataset and fig4 the distribution of sentiments in the train dataset

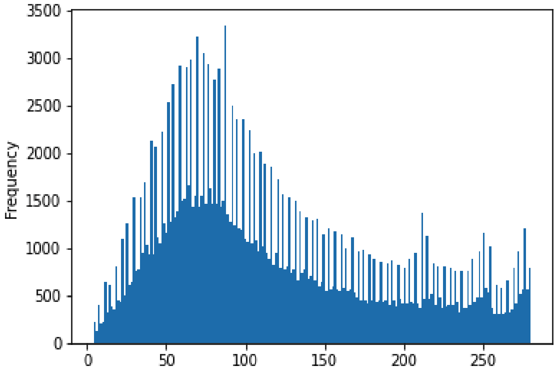


fig3

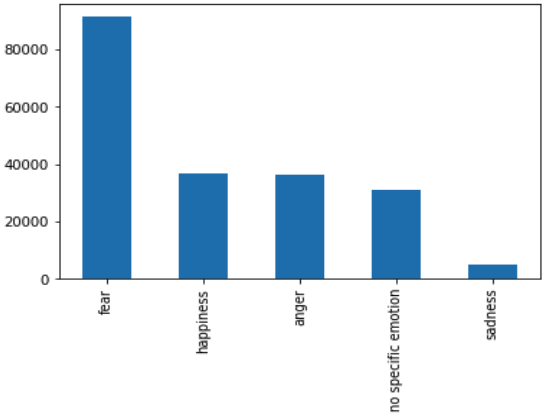


fig4

Distribution of the intensities of all the sentiments (fig5) and Distribution of the categories in the dataset (fig7)

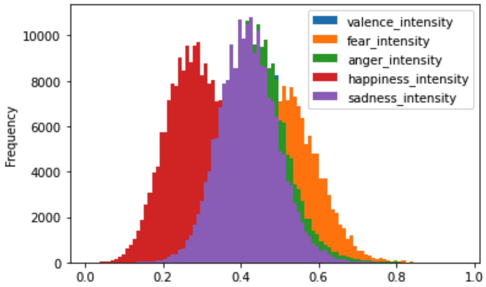


fig5

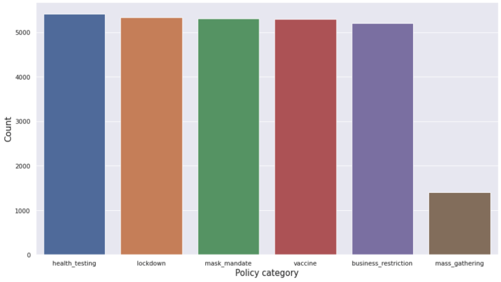


fig7

Dataset statistics:

openICPSR : 54,726,352

openicpsr-unique : 54,380,875

tweet_text fetched : 2,465,153

cleaned_tweet: : 1,001,286

clean_tweet_id : 929,336

unique_tweet: : 909,414

Merged Tweets and sentiments : 928,881

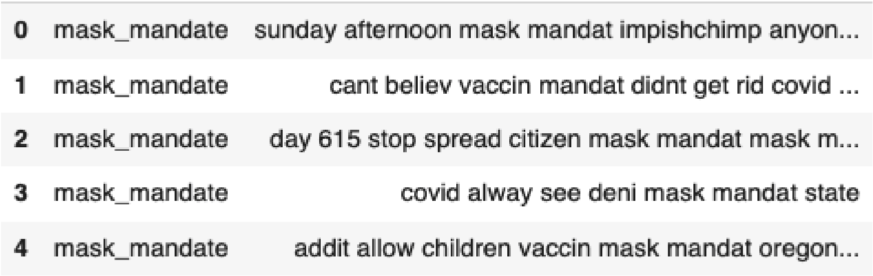
Dataset from Twitter

The number of tweets collected for each category after preprocessing is as indicated below

	Category	Tweets
1	Mass gathering	5306
2	Mask mandate	1409
3	Health testing	5339

4	Business restriction	5410
5	Lockdown	5291
6	Vaccine	5209

After combining the data from all the categories we ended up with 27964 rows of data on which we will be predicting the sentiments.



0	mask_mandate	sunday afternoon mask mandat impishchimp anyon...
1	mask_mandate	cant believ vaccin mandat didnt get rid covid ...
2	mask_mandate	day 615 stop spread citizen mask mandat mask m...
3	mask_mandate	covid alway see deni mask mandat state
4	mask_mandate	addit allow children vaccin mask mandat oregon...

Fig6

Models

Logistic regression

A supervised machine learning classification algorithm that is used to predict the probability of a categorical dependent variable which is used to predict a binary outcome based on a set of independent variables. In logistic regression, a binary outcome is where there are only two scenarios and every probability or possible outcome of the dependent variable can be converted into log odds by finding the odds ratio. The log odds algorithm or the logit function uses the formula ln(P/1-P) to make the conversion. It is used to calculate the probability of a binary event occurring and issues of classification. Logistic regression has proven to work well for cases where the dataset is linearly separable. It is more efficient to classify it into two separate classes.

Decision tree

In classification and regression, decision trees are used. Flowcharts visualize the decision-making process by outlining the various courses of action and their potential outcomes. When formed together, they typically consist of a root node, branches, and leaf nodes resembling a tree. A dataset is divided into smaller and smaller groups, attempting to make each as pure and homogeneous as possible. As soon as we have split the data, we use the final groups to make predictions based on the unseen data. For example, the Gini impurity can be used to evaluate splits in a decision tree.

Naive Bayes classifier

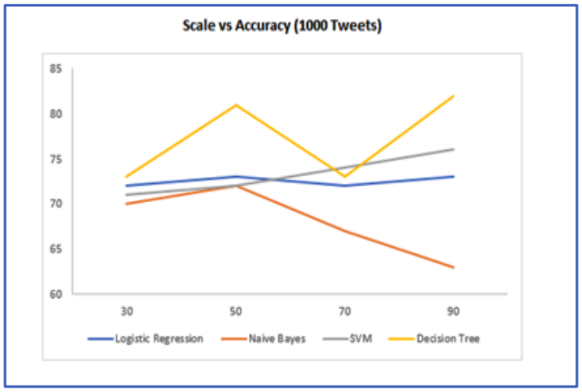
Naïve Bayes (NB) classifier is an algorithm for classifying data which assumes that each feature makes an independent and equal contribution to the target class. NB classifier assumes that each feature is independent and does not interact with each other, such that each feature independently and equally contributes to the probability of a sample to belong to a specific class. This classifier is simple to implement and computationally fast and performs well on large datasets having high dimensionality.

Support vector machine

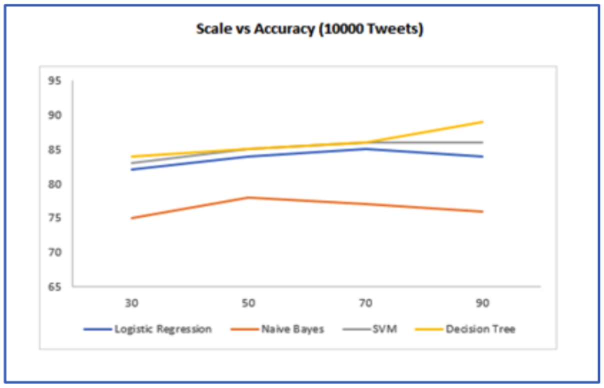
Support vector machines is a supervised machine learning model that uses classification algorithms for two-group classification problems. It can also be applied for both classifications and regression purposes. They use something called the hyperplane which is a line that linearly separates and classifies a set of data. Support vectors are data points nearest to the hyperplane so they are very important that when removed, would change, or alter the position of the dividing hyperplane. The distance between the hyperplane and the nearest data point from either set is also called the margin. To find the right hyperplane, you choose the one with the greatest possible margin between the hyperplane and any point within the training set, giving a greater chance of new data being classified correctly.

Performance evaluation

Scale vs Accuracy



X axis scale; y axis Accuracy %

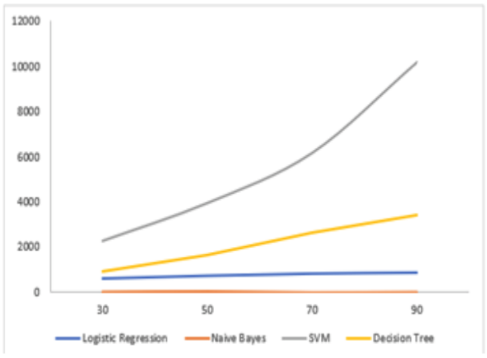


X axis scale; y axis Accuracy %

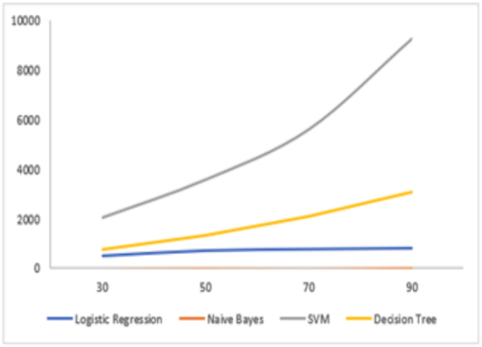
Here, I am evaluating scale vs. Accuracy. Above figure of scale vs accuracy using 1000 tweets, there is a fluctuation among few of the models. In decision Tree & Naïve Bayes for instance, we see evident fluctuations as the scale increases. Theoretically, a model is supposed to perform better when the training set increases, but we see some contradicting results in the graph. This is because, though the scale of the training set increases, 1000 tweets is not enough for a model to learn to its fullest and hence there is a lack of stability when it comes to returning the accuracy after predicting the test set. To resolve this issue and stabilize the accuracy of each model, the same model is trained with 10000 tweets as shown in the graph. As we can see, not only is there more stabilization between all four models, but we can also see the overall accuracy percentage shooting up significantly for every model.

Scale vs Time

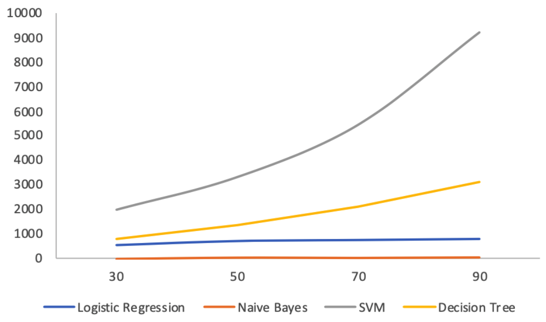
Scale vs server time(10000 Tweets - 3 nodes)

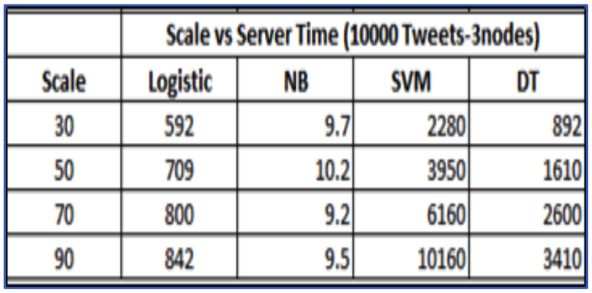


Scale vs server time(10000 Tweets - 5 nodes)

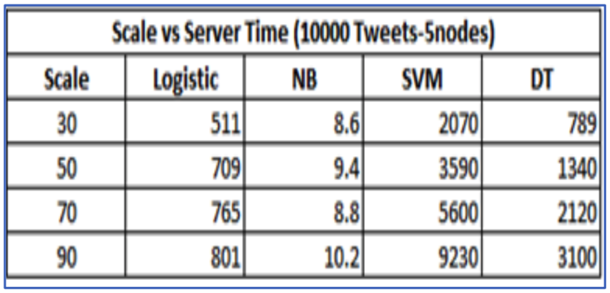


Scale vs server time(10000 Tweets - 7 nodes)

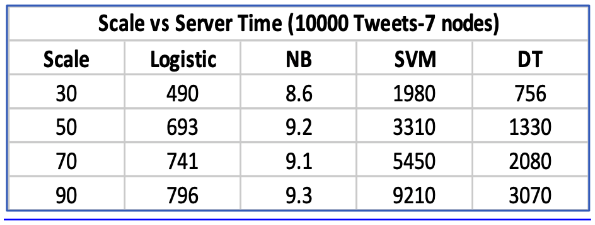




	Scale vs Server Time (10000 Tweets-3nodes)			
Scale	Logistic	NB	SVM	DT
30	592	9.7	2280	892
50	709	10.2	3950	1610
70	800	9.2	6160	2600
90	842	9.5	10160	3410



Scale vs Server Time (10000 Tweets-5nodes)				
Scale	Logistic	NB	SVM	DT
30	511	8.6	2070	789
50	709	9.4	3590	1340
70	765	8.8	5600	2120
90	801	10.2	9230	3100



Scale vs Server Time (10000 Tweets-7 nodes)				
Scale	Logistic	NB	SVM	DT
30	490	8.6	1980	756
50	693	9.2	3310	1330
70	741	9.1	5450	2080
90	796	9.3	9210	3070

I ran the program on 3, 5 and 7 nodes separately to compare the runtimes when being performed on multiple different nodes. The graph & table represents the running time taken to train each model respectively on 3 nodes. If you were to compare the tables of the time taken for each model we can see that parallel computing on 3 nodes performs a better job in terms of training a model in a shorter period.

The second graph & table represents the running time taken to train each model on 5 nodes. In this, if we were to compare the tables of the time taken for each model to get trained on 3 nodes vs 5 nodes, we see that parallel computing on 5 nodes performs a better job in terms of training a model in a shorter period of time compared to it being trained on 3 nodes.

Similarly, we see the same exact trend of the run times being faster in terms of training a model when it is run on 7 nodes. Though the speed jumps are not substantially great when increasing the number of nodes from 3 to 7, it still proves to us that the use of parallel computing when it comes to training a model achieves better results.

Conclusion and future work

Throughout this research project, I have analyzed the sentiments of tweets from all over the United States. A similar analysis can also be performed on tweets from other countries as well.

While I have assessed the effects of a particular category of policy, we can continue our analysis of the impact of a single policy rather than the consequences of a category of policies.

Complex deep learning models can be implemented to make more accurate predictions. Research can be conducted further, such as considering the use of the word2vec tool and multilayer convolutional neural networks

The analysis of a larger number of training data sets and other types of situation or status analysis can be done.

References

[1] Revealing Public Opinion Towards COVID-19 Vaccines With Twitter Data in the United States: Spatiotemporal Perspective

[2] Evolution of public sentiments during the COVID-19 pandemic: Case comparisons of India, Singapore, South Korea, the United Kingdom and the United States

[3] Öhman, A., Fear and anxiety. Handbook of Emotions, 709-729. (2008)

[4] Twitter preprocessor

[5] Emotion Prediction in Tweets with Bidirectional Long Short-Term Memory Neural Network