COVID-19 Vaccination Tweets Sentiment Analysis

Marissa Long
mlong25@wisc.edu

Hailey Park
hpark353@wisc.edu

Yating Tian
ytian83@wisc.edu

Abstract

COVID-19 vaccines are endorsed by the CDC as a safe and effective way to prevent the spread of the virus. However, the public had mixed opinions regarding COVID-19 vaccines. In this study, we apply natural language processing and sentiment analysis to develop machine learning models to further study the public sentiment toward COVID-19 vaccines. We use Twitter data due to its diverse users and its nature of real-time data. We train, validate, and test several machine learning models, namely Random Forest, Naive Bayesian, Support Vector Machine, and Logistic Regression, to predict the sentiment given a text data. This project will help public health policymakers understand the public's attitude toward COVID-19 vaccines and better design campaigns that can encourage more people to get vaccinated. As a result, we were able to achieve the highest test accuracy of 97.324 % on Support Vector Machine after applying Grid Search.

1. Introduction

COVID-19 is a contagious respiratory disease that has had a tremendous effect across the globe since December 2019. The pandemic has caused a devastating loss of life as well as far-reaching social and economic consequences. Several pharmaceutical companies, such as Pfizer BioNTech and Moderna, developed vaccines to prevent further spread of the virus. The CDC recommends that COVID 19 vaccines are a safe and effective way to develop immunity against the COVID-19 virus [4]. Achieving herd immunity by vaccinating the majority of the population against the disease is crucial for the society as a whole to recover from COVID-19 crisis.

However, the public has mixed opinions regarding COVID vaccination, as can be seen from many discussions online [5]. Although some were ecstatic to receive the vaccine and to get back to the "normal" pre-COVID lifestyle, some were concerned about the safety and efficacy of the vaccines. It is likely that the public's concern for COVID vaccines were due to the health misinformation circulating on online platforms, such as Twitter.

The objective of this project is to develop several Machine Learning models that can identify and classify the public sentiment toward COVID vaccines based on COVID vaccination tweets. We chose Twitter data for our analysis, due to its diverse users and its nature of real-time data. We are using Random Forest, Naive Bayes, Standard Vector Machine, and Logistic Regression models that can classify tweets into either positive or negative sentiments to achieve this purpose. By developing Machine Learning models that can classify public sentiments, we believe it can help public health decision makers understand the public's opinion and emotions regarding COVID-19 vaccines.

2. Related Work

Based on the research published by Sattar and Arifuzza man [9], we are planning to follow models that were implemented, such as Naives Bayes, Multinomial Naive Bayes, Support Vector Machine, and Logistic Regression. Villavicencio and Macrohon have chosen precision and recall to test the classifier model using the Naive Bayes classification algorithm. Based on this paper [13], we will follow similar evaluation methods for our models.

3. Motivation

The aim of this study is to investigate and track public sentiment and perception of COVID-19 vaccines. We are curious as to whether there is a salient change in public sentiment toward the vaccine over time as well as if the mainstream media is exaggerating or understating the public's concerns or enthusiasm of COVID vaccines.

Our study, combined with external data, such as COVID infection rate over time or major events regarding COVID 19 vaccines, will help the government better understand the public perceptions that can influence the

acceptance of vaccines. Our research will support public health officials and policy makers to understand the public's feelings toward vaccines so that they can design effective strategies to enhance vaccine uptake rates.

The acceptance of COVID vaccine is crucial to combat the coronavirus pandemic, and our research hopes to help government agencies and public health officials around the world to design successful campaigns that can enhance the public willingness to accept vaccines.

4. Proposed Method

4.1. Random Forest

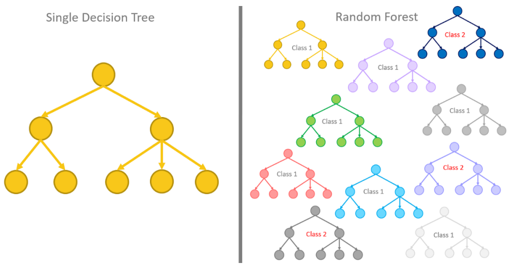


Figure 1. decision tree and random forest [10]

Random forest is an algorithm to solve problems such as classification and regression. It is an ensemble machine learning model that combines many classifiers to generate solutions. Many decision trees make up a single random forest algorithm. We can use bagging or bootstrap to train our forest. For our project, we train a weak tree from a bootstrap sample from the training set for each iteration. For splitting, a random collection of characteristics is picked for each node of this decision tree.

4.2. Naive Bayes

$$P(x_i \mid y) = P(i \mid y)x_i + (1 - P(i \mid y))(1 - x_i)$$

Figure 2. Bernoulli Naive Bayes [1]

The Naive Bayes Model is a widely used supervised learning algorithm that categorizes texts into several groups. The naive Bayes method in our model looks at all the tweet text to determine whether it is positive or negative, based on the output set. In our model, we choose Bernoulli Naive Bayes that implements the naive Bayes training and classification algorithms for data that is distributed according to multivariate Bernoulli distributions. In other words, there may be numerous features, but each one is considered to be a binary-valued variable. As a result, samples must be represented as binary-valued feature vectors; if given any other type of data, a Bernoulli Naive Bayes instance may binarize it.

4.3. Support Vector Machine

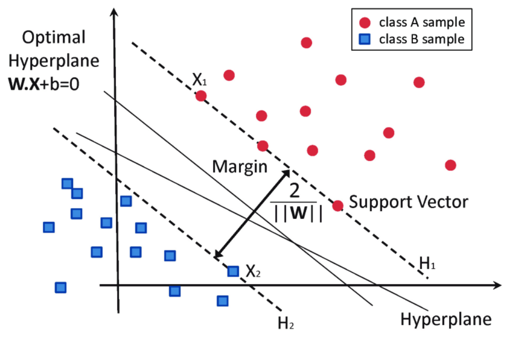


Figure 3. SVM [2]

Support Vector Machine (SVM) is another widely used binary classification technique. In an n-dimensional feature space, we wish to discover an (n-1) dimensional hyperplane that divides data points from two different classes, assuming that they may be separated by a line [11]. Assume that the hyperplane is w*x+b=0, which is the bound of the two classes, and that each class has at least one sample. In order to maximize the distance between the two hyperplanes, we must first maximize the distance between the two hyperplanes. The distance between the two classes is provided by: $2/\|w\|$. The prediction of classes in SVM makes use of the optimal hyperplane. Meanwhile, we use a linear kernel support vector machine in our model, because it matches the best with the text feature [6].

4.4. Logistic Regression

The method of modeling the probability of a discrete result given an input variable is known as logistic regression. Logistic regression models frequently model a binary result, which measures the relationship between the target variable and independent variable as either true or false. For our project, we train and test a logistic regression model that classifies tweets into either positive or negative tweets.

$$ln(\frac{p}{1-p}) = \beta_0 + \beta_0 x$$

In this formula, x is the feature vector, and beta is the coefficient vector, which is estimated from the training set while minimizing the negative log-likelihood.

5. Experiments

5.1. Data Set

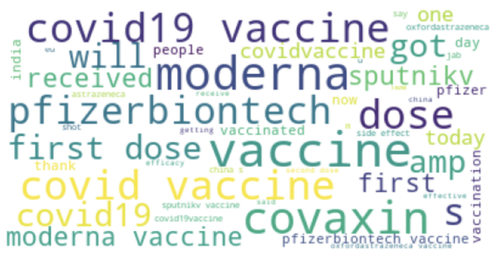


Figure 4. Word cloud of the data

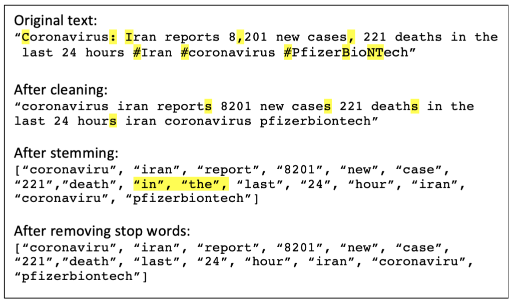
Our data set was obtained from Kaggle [7]. The original data was collected using the Python package, tweepy, and Tweet API. The data set contains 33,717 tweets related to seven types of vaccines: Pfizer/BioNTech; Sinopharm; Sinovac; Moderna; Oxford/AstraZeneca; Covaxin; Sputnik V. The tweets are collected from December 12th, 2020 to March 20th, 2021. There are 16 variables, but for our purpose of sentiment analysis based on tweet texts, we focus on the text column, which includes the text data of the original tweet and the link to the original tweet.

5.2. Data Preprocessing

5.2.1 Data Cleaning and Stopwords Removal

We have found that the text data from each tweet contains some unwanted components, such as emojis, URLs, punctuation marks, hashtags, and Twitter IDs. We eliminated these irrelevant details using the regex function. We also got rid of non-word characters, such as extra spaces and single characters, within the text. Additionally, we converted letters to lowercase and removed stop words from the text. Stop words are words like ”and”, ”but”, ”so” that are frequently used in the text but are not significant in the analysis [6]. We have downloaded a list of stop words from NLTK and removed some words, such as ”not” or ”isn’t” to prevent them from impacting the sentiment classification. By eliminating these low-level information from the text, we are hoping to improve the model performance and to correctly classify each text into either positive or negative sentiment.

5.2.2 Tokenization



Original text:
```
“Coronavirus: Iran reports 8,201 new cases, 221 deaths in the
 last 24 hours #Iran #coronavirus #PfizerBioNTech”
```

After cleaning:
```
“coronavirus iran reports 8201 new cases 221 deaths in the
last 24 hours iran coronavirus pfizerbiontech”
```

After stemming:
```
[“coronaviru”, “iran”, “report”, “8201”, “new”, “case”,
“221”,”death”, “in”, “the”, “last”, “24”, “hour”, “iran”,
“coronaviru”, “pfizerbiontech”]
```

After removing stop words:
```
[“coronaviru”, “iran”, “report”, “8201”, “new”, “case”,
“221”,”death”, “last”, “24”, “hour”, “iran”, “coronaviru”,
“pfizerbiontech”]
```

Figure 5. Data prepossessing process

We split a sentence that has been cleaned into tokens, a smaller unit of individual words or terms. We split the tweet that has been cleaned by splitting them based on the whitespace characters.

5.2.3 Data Normalization

After we have cleaned and tokenized each tweet, we normalize the text data using stemming. Word stemming normalizes each word by converting related words into their simpler root form. We use PorterStemmer from the NLTK library to accomplish this feat. Stemming words into their root form will truncate ”reports” into ”report” as shown in the figure 5. We are using stemming instead of lemmatization since lemmatization is more expensive in its computational cost.

5.2.4 Term Frequency-Inverse Document Frequency

Term frequency-inverse document frequency (tf-idf) is a value that encodes the importance of a word given a sentence. It can be computed by taking the product of raw term frequencies(df(d,t)), the number of times a term occurs in a document and the inverse document frequency (idf(t,d)), which gives a higher weight to more unique words and lower weight to commonly used terms in a document [8]. Hence, tf-idf can measure how important a specific word is in a set of tweets in our case. By down weighting frequently occurring words, we can improve the model performance similar to the effect of removing stopwords. We used TfidfVectorizer from scikit learn library to compute tf, idf, and tf-idf scores on documents of our training tweets.

$$idf(t,d) = \frac{log(n_d)}{(1 + df(d,t))}$$

$$tfidf(t,d) = tf(t,d) * idf(t,d)$$

5.3. Models

For each model, we compared the results between default parameters in the sci-kit learn and optimized models after using Grid search. Grid search is a tuning technique that is used to find the optimal values of hyperparameters. We apply grid search with 5-fold cross-validation for all the models. After applying grid search, we found that there are 3 hyperparameters that can increase the accuracy. The parameter "norm" means each output row will have a unit norm. The default of "norm" is "l2". The parameter "tokenizer" overrides the string tokenization step while preserving the preprocessing and n-grams generation steps and its default set is none. The parameter "use-idf" enables inverse-document-frequency reweighting and its default is set as true [3].

5.3.1 Random Forest

No Tuning : We choose 200 estimators and fix the random seed as 0 to fit the model. By the model default, we used gini criterion, 0 max depth, 2 minimum number of samples, and 1 leaf node for each sample. The test accuracy was 96.101 %.

Grid search: After grid search, we find the optimal hyperparameters for the random forest classifier is to set the norm as None, tokenizer with the function tokenizer-porter we defined in the data pre-processing step, and use-idf as False. We saw a slight improvement with the test accuracy of 96.764 %

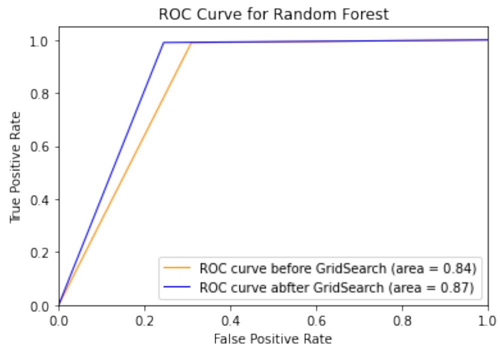


Figure 6. Random Forest ROC curve

We want to visualize the change of the Random Forest model before and after grid search with the ROC curves. The blue line that represents Random Forest after grid search lies above the orange line before grid search, therefore, grid search does improve the performance of Random Forest model in this case, though not too much.

5.3.2 SVM

No Tuning: Keeping the default linear SVM parameter settings in Sci-kit learn, we get the test accuracy of 97.044 %.

Grid Search: After grid search, we found that the optimal hyperparameters were norm set to None, tokenizer defined in the data preprocessing step, and use-idf set to false. We observe a slight improvement with the test accuracy of 97.324 %.

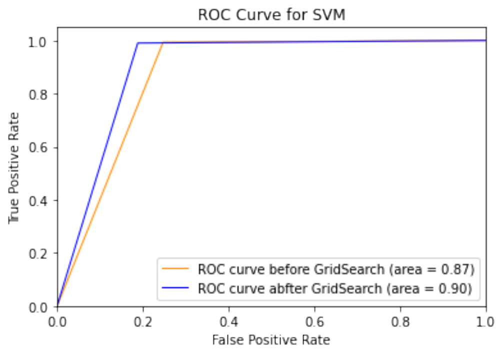


Figure 7. SVM ROC curve

By visualizing the ROC curves for the SVM model before and after grid search, the blue line that represents SVM after grid search lies above the orange line before grid search, therefore, grid search does slightly improve the performance of SVM model in this case.

5.3.3 Naive bayesian

No Tuning: In our model, we used the Bernoulli formula with the default parameter in sci-kit learn. We get a test accuracy of 91.233 %

Grid Search: We didn't apply grid search to Naive Bayes since there were no appropriate hyperparameters for us to tune.

5.3.4 Logistic regression

No Tuning: In our logistic regression model, we used the default parameter in sci-kit learn. We get a test accuracy of 92.406 %.

Grid Search: As a result of grid search, we found that the optimal hyperparameters are norm set to None, tokenizer defined in the preprocessing step, and use-idf set to False. There was an improvement in the test accuracy, which rose up to 95.515 %.

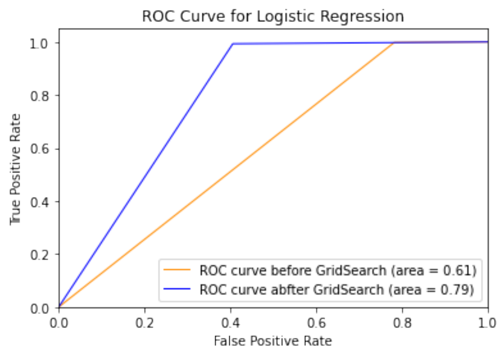


Figure 8. Logistic Regression ROC curve

By visualizing the ROC curves for the Logistic Regression model before and after grid search, the blue line that represents SVM after grid search lies above the orange line before grid search. According to the curve, grid search improves the performance of the Logistic Regression model a lot compared to the other two models we did grid search with.

We create the following table to better compare the effect of grid search between different models.

Method	Origin Accuracy	Tuned Accuracy
Random Forest	96.101 %	96.764 %
SVM	97.044 %	97.324 %
Naive bayesian	91.233 %	NA
Logistic Regression	92.406 %	95.515%

Table 1. Test Accuracy table

6. Results and Discussion

We decide to use precision, recall, and f1 score besides test accuracy to compare the performance of the models. Precision is the ratio of predicted observations with the same label that are predicted correctly to the total number of the predicted observations with that label. Recall is the ratio of the predicted observations with the same label that are predicted correctly to all of the actual cases. F1-score considers the case where there are both false positives and false negatives, and it combines both the precision and recall of the model [12]. The higher the F1-score is, the better the model is.

6.1. Random Forest

We get a precision of 0.89 for negative tweets and 0.97 for positive tweets. The recall of negative tweets is 0.76, and 0.99 for positive tweets. The random forest model has an f1 score of 0.82 for negative tweets and 0.98 score for positive tweets, and an overall f1-score of 0.97, with a test accuracy of 96.764 %.

6.2. SVM

We get a precision of 0.90 for negative tweets and 0.98 for positive tweets. The recall of negative tweets is 0.81, and 0.99 for positive tweets. The SVM has an f1 score of 0.85 for negative tweets and 0.99 score for positive tweets, and an overall f1-score of 0.97, with a test accuracy of 97.324 %.

6.3. Naive Bayesian

We get a precision of 0.87 for negative tweets and 0.91 for positive tweets. The recall of negative tweets is 0.09, and 1.00 for positive tweets. The Naive Bayesian model has an f1 score of 0.16 for negative tweets and 0.95 score for positive tweets, and an overall f1-score of 0.91, with a test accuracy of 91.233 %.

6.4. Logistic Regression

We get a precision of 0.90 for negative tweets and 0.96 for positive tweets. The recall of negative tweets is 0.59, and 0.99 for positive tweets. The Logistic Regression model has an f1 score of 0.72 for negative tweets and 0.98 score for positive tweets, and an overall f1-score of 0.92, with a test accuracy of 95.515 %.

The following two tables are the summary for the results for the positive and negative tweets.

6.5. Final Result

Method	Precision	Recall	F1 score
Random Forest	0.97	0.99	0.98
SVM	0.98	0.99	0.99
Naive bayesian	0.91	1.00	0.95
Logistic Regression	0.96	0.99	0.98

Table 2. Table for Positive Tweets

Method	Precision	Recall	F1 score
Random Forest	0.89	0.76	0.82
SVM	0.90	0.81	0.85
Naive bayesian	0.87	0.09	0.16
Logistic Regression	0.90	0.59	0.72

Table 3. Table for Negative Tweets

We have found the precision, recall, f1 score ,and the test accuracy of each model after performing grid search. Comparing test accuracies, we found that the model that performs best is SVM after grid search with the highest accuracy of 97.324 %, followed by Random Forest with an accuracy of 96.764 % on the best model chosen in grid search. The Logistic Regression model after grid search also performs well with an accuracy of 95.515%. The Naive Bayes model has the lowest test accuracy in four models with an accuracy of 91.233 %.

After comparing the precision, recall, and f1 score for both the positive and negative tweets, SVM model still performs best among the models. Therefore, we decide to choose the SVM model as our final result.

We want to further visualize the performance of the SVM model after grid search with a confusion matrix. The tweets with positive sentiment are labeled positive here and vice versa. According to the graph, the true positive rate is 89.63%, and the true negative rate is 7.70%. The false positive rate is 0.89% and the false negative rate is 1.78%. Our final model perform well for both prediction positive and negative tweets.

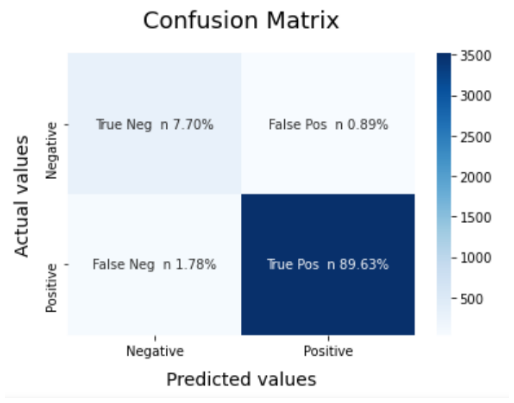


Figure 9. SVM Confusion Metrics

6.6. Discussion

We found that the dataset we've explored has a significantly greater number of tweets that are classified as positive. We project that the overall high positive sentiment in the dataset can be interpreted as the public's general optimism for COVID-19 vaccines. It is also likely that those who are against the COVID-19 vaccination are reluctant to express their opinions on the public social media platform like Twitter where your personal identity can be easily found by others. It should also be noted that our dataset is not perfectly representative of the entire population, since it doesn't account for opinions of those who are not actively tweeting on Twitter and those who are not on Twitter.

7. Conclusions

From our experiments, we can safely conclude that we were able to identify and classify public sentiments toward COVID-19 vaccines as either positive or negative, which was the goal of our project. We were able to obtain a generally high accuracy rate for all of our models.We found that the model with the best overall result was Support Vector Machine after applying Grid Search with an accuracy of 97.324%.

Following the models implemented in the previous works [9], we have also implemented Naive Bayes, Support Vector Machine, and Logistic Regression models. One thing to note is that their research classifies tweets into three sentiments, namely positive, negative and neutral. We have, however, focused on classifying tweets into binary sentiments, either positive or negative, to implement the Bernoulli Naive Bayes model. The classified sentiment of our study is very similar to that of Villavicencio and Macrohon's research [13], since they

also have a large number of tweets that are classified as positive.

For future research, we hope to obtain larger data sourced from multiple social media platforms to account for opinions of a greater variety of people. It should also be stressed that collecting more text data with negative sentiments and training our machine learning model based on it is one of the tasks that should be addressed to further improve the performance of the model. It will also be interesting to conduct a time series analysis along our machine learning model to analyze how significant events related to COVID-19 vaccines, such as FDA authorization of certain vaccines or accidents reported influence the public's sentiment over time. We are hopeful that our model can benefit public health decision-makers or government agencies in exploring the public's emotions on COVID-19 vaccines.

8. Acknowledgements

We appreciate the help of Professor Sebastian Raschka's help and his textbook [8] that guided us applying machine learning to sentiment analysis. We also appreciate Gabriel Preda who collected the original twitter data [7] that allowed us to analyze the sentiment of vaccination tweets.

9. Contributions

Hailey Park found the original dataset and was responsible for the preprocessing step of the data, which includes data cleaning and stopwords removal, tokenization, data normalization, and weighing the term frequency using tf-idf. Hailey wrote the abstract, introduction, related work, motivation, and data preprocessing, and conclusion section in the final documentation.

Yating Tian was responsible for developing machine learning models, including Random Forest, Naive Bayes, Support Vector Machine, Logistic Regression, as well as constructing ROC curve and confusion matrices. Yating wrote a proposed methods section which outlined each model we're implementing in this project and the models section in the final documentation.

Marissa Long was responsible for hyperparameter tuning using GridSearch on the Random Forest, Support Vector Machine, and Logistic Regression model as well as the evaluation of each model. Marissa wrote the models, results, discussion, and conclusion section, , and work on the references in the final documentation.

10. Code

All of the code for this project is available on the following GitHub link: `https://github.com/hparkailey/451_vaccination_tweets/tree/main`.

References

[1] 1.9. naive bayes.

[2] Classification of data by support vector machine (svm ...

[3] Sklearn.feature$_{e}xtraction.text.tfidfvectorizer$.

[4] Benefits of getting a covid-19 vaccine, Aug 2021.

[5] N. F. Johnson, N. Velásquez, N. J. Restrepo, R. Leahy, N. Gabriel, S. El Oud, M. Zheng, P. Manrique, S. Wuchty, Y. Lupu, and et al. The online competition between pro- and anti-vaccination views, May 2020.

[6] A. KOWALCZYK. Linear kernel: Why is it recommended for text classification ?, Jul 2020.

[7] G. Preda. Explore vaccines tweets, Mar 2021.

[8] S. Raschka and V. Mirjalili. *Applying Machine Learning to Sentiment Analysis*, page 259–283.

[9] N. S. Sattar and S. Arifuzzaman. Covid-19 vaccination awareness and aftermath: Public sentiment analysis on twitter data and vaccinated population prediction in the usa, Jun 2021.

[10] R. Silipo. From a single decision tree to a random forest, Oct 2019.

[11] K. Sirohi. Support vector machine (detailed explanation), Mar 2020.

[12] E. Solutions and . Name. Accuracy, precision, recall amp; f1 score: Interpretation of performance measures, Nov 2016.

[13] C. Villavicencio, J. J. Macrohon, X. A. Inbaraj, J.-H. Jeng, and J.-G. Hsieh. Twitter sentiment analysis towards covid-19 vaccines in the philippines using naïve bayes, May 2021.