Evaluating public options of COVID-19 vaccine using natural language processing

**Problem description**

2020 is an unprecedented time and the whole world were almost shut down during the pandemic. Recently, two types of vaccines have been approved by FDA and they are like the light in the dark. Hopefully, vaccine can help end the pandemic and we can go back to normal life in 2021. However, there are different options about vaccines. People may welcome, oppose, or be skeptical about Vaccinations. It is important to find what people think. According to WHO, Community immunity requires at least 80% of the population to be vaccinated. As a result, knowing percentage of people who welcome, oppose, be skeptical is important. It would help government/companies to decide how many vaccines are needed. It would also help the government to decide whether more effort is needed to persuade people to get vaccination.

**Data sets**

My dataset is obtained from twitter API using stream filter method. The objective is to obtain the tweets related to covid vaccine. However, if I directly search ‘covid vaccine’, the obtained tweets may be misleading and large portion of tweets will be omitted because covid vaccine related tweets can have different keywords other than ‘covid vaccine’. To tackle this issue, I will utilize two filters: the first filter is to filter vaccine related tweets and after that, utilize the second filter to filter the covid related tweets based on the tweets obtained from the first filter. The keywords utilized in the vaccine-filter is ‘vaccine’, ‘vaccinated’ and ‘vaccination’; the keywords of the covid-filter is 'covid', 'covid-19', 'coronavirus', 'pandemic', 'covid19', 'social distance', 'cov2', 'quarantine', 'moderna', 'oxford', 'pfizer', 'reopen', 'social distance'. By filtering the tweets twice, I am able to obtain the datasets related to covid vaccine. The details about data import, data cleaning is described below.

1. Data import

As mentioned above, I utilized twitter API to obtain the tweets. After obtaining the API, I utilized the tweepy.Cursor function to obtain twitters. This function obtains the tweets that contain the key words in the “SearchKeyWords”. As mentioned above, as the first filter, the SearchKeyWords I utilized is ‘vaccine’, ‘vaccinated’ and ‘vaccination’. After filtering the tweets, the tweet text is saved into a datafame.

However, the tweepy.Cursor function using Api.search only allows to search the tweets in the past week. In addition, I can specify start and end date in this function to search the tweets in a specific date. In order to obtain the tweets in the full past weeks., I called tweepy.cursor function 7 times for each date and obtained the tweets in the full week.

After obtained all the tweets, I combined all the tweets into a new dataframe. The number of the total tweets obtained in this step is 2400. However, as shown in the figure below, the raw tweets contain punctuation, web addresses, retweets, and etc.. These features need to be cleaned before data analysis.

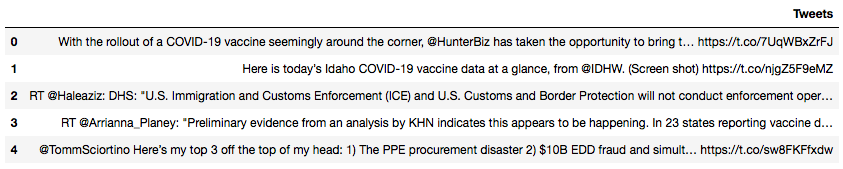


Figure 1: raw tweets

1. Data cleaning

After importing the raw tweets, I utilized functions to clean the tweets by removing @, retweets, removing punctuation, formatting in the lower case, splitting sentences and joining words.

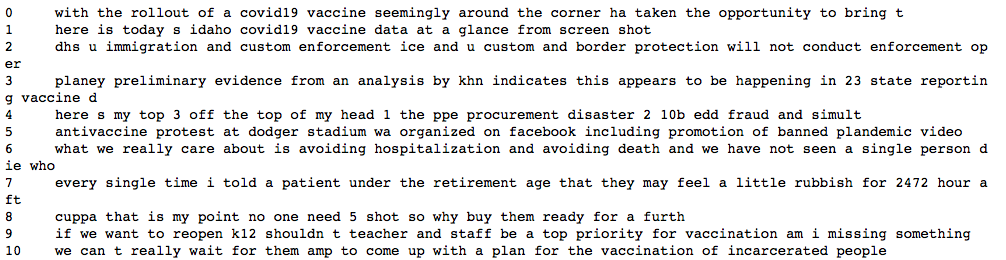


Figure 2: the tweet text after cleaning

After cleaning, the tweets may contain duplicate tweets, and thus the duplicated tweets are removed after tweet cleaning. After removing the duplicated tweets, the total number of remaining tweets is 1852. However, the tweets after all the previous data wrangling is the tweets related to vaccine. Thus, in the next step, I applied a second filter to filter the tweets related to covid, by searching the key words of 'covid', 'covid-19', 'coronavirus', 'pandemic', 'covid19', 'social distance', 'cov2', 'quarantine', 'moderna', 'oxford', 'pfizer', 'reopen', and 'social distance'. The final number of tweets after all the data cleaning and filtering is 548. Figure 3 shows the tweets related to covid vaccine after second filtering.

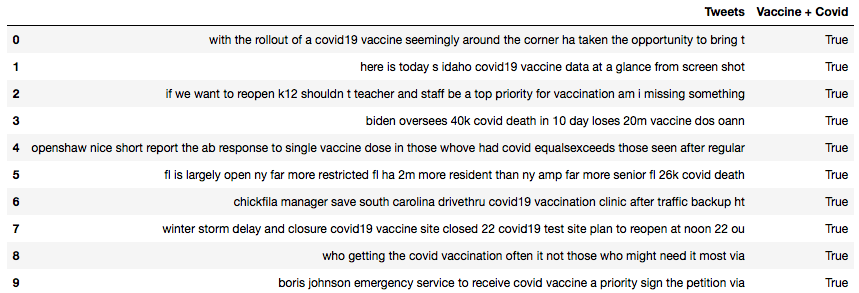


Figure 3: covid vaccine related tweets after cleaning and filters

**Data Analysis**

The topic in this project is related to twitter sediment. In the sediment analysis, it returns polarity and subjectivity. Subjectivity ranges from 0 to 1, where 0 is very objective and 1 is very subjective. Polarity ranges from -1 and 1 where -1 means very negative statement and 1 means very positive statement. People’s options about covid vaccine can be evaluated by evaluating the polarities. Figure 6 shows the code for twitter sediment analysis. Figure 4 shows the results of sediment analysis. The result shows that 44.34% of the tweets is positive about the covid vaccine, while 37.23% is neutral about the covid vaccine and 18.43% is negative about the covid vaccine.

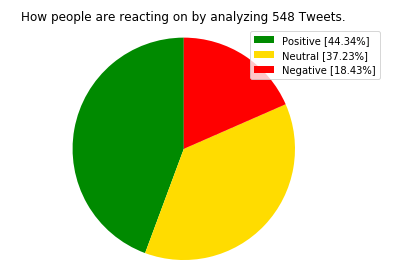


Figure 4: Pie chat of the sediment result

The subjectivity vs. polarity is also analyzed and the results are shown in Figure 8. The results show that there is more people with objective options about covid vaccine than people with negative options. It also shows that if the tweet is more objective, the option is more neutral. However, if tweet is more subjective, there is more possibilities to see negative or positive options. In addition, the percentage of tweets whose options are strongly negative (<-0.7) or positive (>0.7) is relative low.

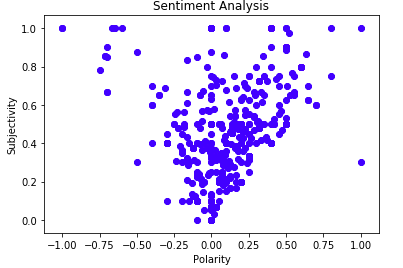


Figure 5: subjectivity vs. polarity

A word cloud chart is also generated to help visualize the text data and importance of each tag. In this chart, the size of each word is proportional to its frequency. The results show that the most frequent texts in the tweet dataset are covid19 vaccine and covid vaccine, which is consistent to our expectation.

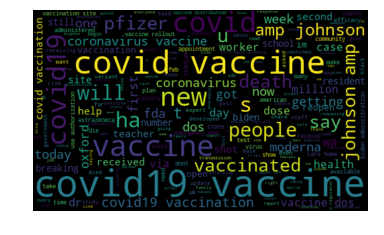


Figure 6: word cloud chart of covid vaccine related tweets

**Machine Learning**

In the machine learning part of this project, I utilized two kinds of machine learning methods to analyze our data. One is to use support vector machine to build own twitter sediment model and the other is to utilize unsupervised ML with clustering algorithm to divide the tweet datasets into 3 clusters and the purpose is to validate the sediment model & results.

1. support vector machine to build own twitter sediment model

In the previous analysis, each tweet has been labeled as positive, neutral, negative from the tweet sediment analysis by using the built-in TextBlob function. The main idea for the support vector classifier (SVC) analysis in this study is to build own sediment model by training the data with the existing labels. The steps are divided as follows:

* 1. Assign the labels to each tweet. The labels are obtained by the TextBlob function: the value is defined as 1 if the tweet is positive, 0 if the tweet is neutral, and -1 if the tweet is negative. The reason why label is added to each tweet is that the labels in the training data set can be utilized to train the model and the labels in the testing data can be utilized for model verification.
  2. Split the datasets into training data and testing data. In this step, I utilized sklearn’s train\_test\_split function to split the training and testing data. In this study, 30% of the data is considered as testing data and 70% of the data is considered as the training data.
  3. Vectorize all the tweet. Textual data cannot be directly recognized by machine learning models and the text needs to be encoded as integers for use as inputs in machine learning analysis. In this step, I utilized scikit-learn’s CountVecorizer function to vectorize the tweets. This function convert a collection of text documents to a vector of token counts.
  4. Remove the stopwords. Tweets may contains words such as ‘a’, ‘the’, ‘is’ and etc. These are common words but do not add any information in the content. We can improve our machine learning models by removing these stopwords. In addition, I also removed the word which only appears one time in the tweet dataset.
  5. Train the model using support vector machine and predict the labels of the testing data. I utilized sklearn’s SVC model for the machine learning analysis and tested various model fitting algorithms, such as linear, polynomial, sigmoid, and etc. The results show that the linear model gives the best fit, thus I utilized the linear algorithm to fit the training data and then utilized the model to predict the new labels of the testing data.
  6. Estimate the accuracy of the model. The model accuracy is evaluated by comparing the predicted labels using SVC and the previous labels using TextBlob function in the testing data. The calculated accuracy score for the SVC is 62.42%, which indicates that SVC predicts the same labels as the TextBlob function for 62.42% of the testing data. The reason of the discrepancy might be due to the limited training data in this study. In the current analysis, the training datasets contains 383 tweets, of which the size of datasets is too small to build a reliable machine learning model. Increasing the total number of tweets might help improve model accuracy.

1. utilize unsupervised ML with clustering algorithm to divide into 3 clusters

In the second machine learning analysis, I applied unsupervised machine learning approach to divide the dataset into 3 clusters. The main purpose is to compare whether the cluster analysis can provide the similar results to group the tweets compared to the tweet sediment analysis using TextBlob function.

The clustering method I utilized is KMeans and the input data is the vectorized tweet data from the previous analysis. The following figures shows that clustering results: the data is divided into three clusters: the first cluster contains 13.5% of the total tweets, the second cluster contains the 81.02% of the total tweets, and the third cluster contains 5.47% of the total tweets.

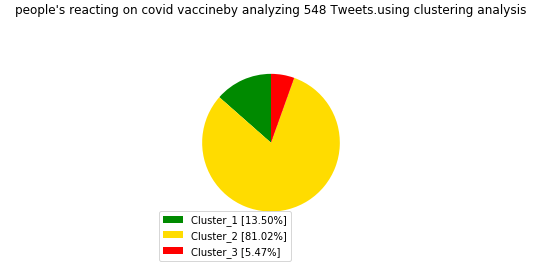


Figure 7: Pie chat of the sediment result using clustering analysis

I also obtained the top feature words in each of the clusters. According to the feature words, it doesn’t indicate the polarities, whether people welcome or oppose the covid vaccine. It indicates that the tweets are not clustered into groups based on the polarities but based on the frequencies of feature words in each group. This could probably explain why the results obtained by unsupervised machine learning are quite different from that obtained from the TextBlob function. The results indicate that the unsupervised clustering approach might be not suitable to perform tweet sediment analysis.

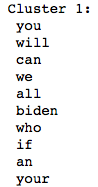
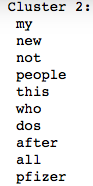
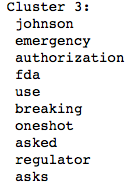
  

Figure 7: Top feature words in three clusters

**Summaries**

In this study, support vector classifier and unsupervised clustering analysis are performed to analyze the tweets to evaluate the options about the covid vaccine and the results are compared with tweet sediment analysis using TextBlob functions. One of the challenging problems in this study is to data wrangling as tweets not only contains text but also a lot of other features such as retweets and web addresses. I was able to build functions to clean the tweets and obtain the main tweet text. The other challenging issue in this project is filtering tweets. Directly filtering ‘covid vaccine’ will omit a significant number of tweets related to covid vaccine. To tackle this issue, I utilized two filters to filter covid and vaccine independently.

After data cleaning and filtering, sediment analysis was performed by using TextBlob function and the polarities are utilized to evaluate the tweet options. The results show that 44.34% of the tweets is positive about the covid vaccine, while 37.23% is neutral about the covid vaccine and 18.43% is negative about the covid vaccine. After tweet sediment analysis, support vector classifier analysis was performed to build own sediment model. The result shows an accuracy of 62.42% compared to TextBlob function. One possible reason for the discrepancy is the limited training data in this study. The second machine learning analysis is unsupervised clustering. The data set has been clustered into three groups and the top feature words are illustrated. The large differences between the results of sediment analysis and unsupervised clustering analysis might indicate unsupervised clustering analysis might not be suitable for sediment clustering.