COVID Vaccination Progress Analysis

Bruce Jiang
Computer Engineering
San Jose State University
San Jose, U.S.
runfeng.jiang@sjsu.edu

Prasaanth Radhakrishnan
Computer Engineering
San Jose State University
San Jose, U.S.
prasaanth.radhakrishnan@sjsu.edu

Wai Yin Suen
Software Engineering
San Jose State University
San Jose, U.S.
waiyin.suen@sjsu.edu

Hanyu Hu
Software Engineering
San Jose State University
San Jose, U.S.
hanyu.hu@sjsu.edu

***Abstract*—COVID-19 is a contagious disease which is caused by a severe acute respiratory syndrome coronavirus 2(SARS-CoV-2) and is the reason for millions of deaths around the globe. This is no doubt one of the biggest pandemics the world has ever seen. This paper aims to study the effects of various vaccines on patients and their progress. Even though this paper also includes various datasets, it primarily studies the effects of the vaccine in California, U.S. The data starts from the 5th of January 2020 to the 9th of March 2022. The dataset would be expanded in the future for better analysis and results. It also includes the various steps involved in data mining which is pre-processing, encoding, scaling the data, and building an algorithm that would predict the best results for the datasets.**

***Keywords*—COVID-19, Vaccine, Data Mining, Trend Analysis**

I. Introduction

According to the Johns Hopkins data as of 31st March 2022, which is about 2 years since the outbreak began, there have been more than 487 million cases reported and more than 6 million deaths because of COVID-19 around the world. When it all seemed like there would be no end to the number of deaths, the vaccines were introduced and played a major part in reducing the effects and the deaths because of the virus. As of March 2022, there have been more than 10 billion vaccines administered. According to the WHO, this still poses a very big risk to humanity. Coronavirus belongs to a wide variety of viruses that cause various sorts of ailments and respiratory diseases. Their main effects are causing upper respiratory tract infections.

Human coronavirus was first characterized in the 1960s and they mainly caused respiratory tract infections in children. Over the years, at least five more variations have been identified and one of them includes the "Severe Acute Respiratory Syndrome" (SARS-CoV-2). The history of human coronaviruses began in 1965 when Tyrell and Bynoe found another virus called B814 which caused bronchial infections, but they were unable to cultivate them. Towards the late 1960s, while testing human and animal viruses they all seemed to have the same crown shaped appearance and they were named as coronaviruses denoting the crown-like appearance of the viruses [1].

During the 2002-2003 outbreak, about 8000 people were infected by SARS in 29 countries and there were 700+ fatalities reported to it. Now to think millions of people have succumbed to a different strain is terrifying. The availability of COVID vaccines to the general population greatly contributed to playing a role of reducing the symptoms and fatalities. Many countries and organizations introduced various vaccines with a different number of doses required or different side effects and so on. This was a long and arduous task and multiple companies worked on making a vaccine and put in months of effort but not all of them got approved [10].

There are various stages a vaccine needs to go through to get approved. First it needs to be clinically tested and approved for human testing which is the step in which most vaccines get removed. Out of the fraction of vaccines that do get approved, it gets tested through four phases at the end of which it could be given to the general population [10].

The timeline of the vaccine creation and delivery process is shown in Fig 1. The effectiveness of various approved vaccines is shown in Fig 2.

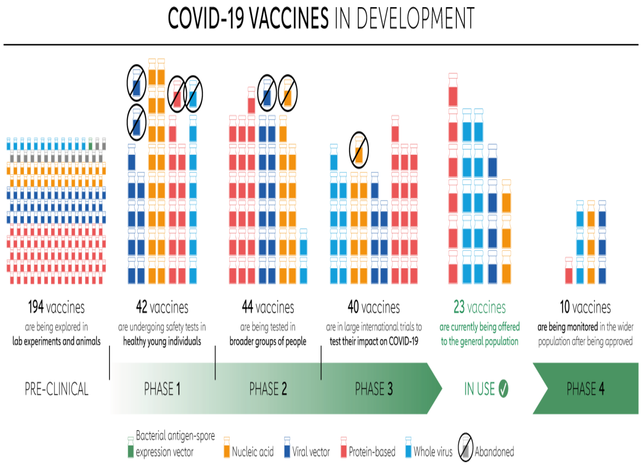

Fig. 1. Timeline of vaccine creation and delivery process [1]

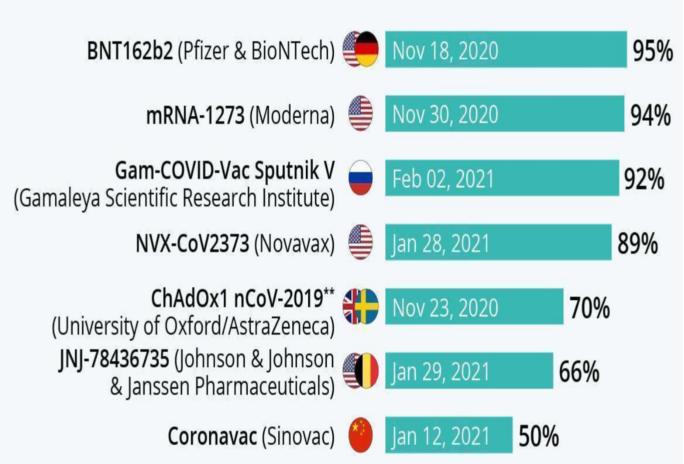

Fig. 2. Effectiveness of various approved vaccines [2]

A. Data Gathering and Usage

The dataset that we are currently using is the 'Statewide COVID-19 Vaccines Administered by County' which was provided by the 'California Open Data Portal'. It was created and last updated on the 9th of March. The main reason we decided to use this dataset was because of all the relevant features that were available like the total number of doses and each separate dose count for various vaccines. The vaccines that are used in this dataset are Pfizer, Moderna and Johnson & Johnson. We also chose this dataset because it has 38,130 entries which would help us to allocate enough data to train, test and validate the model when we use fully connected Neural networks [3].

It also provides us with data on the number of doses people have taken and the number of people getting vaccinated for the first time or booster. We plan to study the effectiveness of each individual vaccine and if the effects vary based on the number of doses.

We also plan to investigate datasets of other states and counties in the U.S and use our prediction model to test the accuracies of our predictions. There are also other datasets available in the California Open Portal which classify based on other demographics like gender, age group, race/ethnicity, VEM Quartile and so on. Analyzing these datasets could give more information on what the most influential factors are based on which we can determine the effectiveness of the vaccine [4].

B. Feature Selection

Feature Selection is the process of reducing the number of input variables or columns that we choose to make a predictive model. We know that the more the data, the better the prediction would be, but in some cases, extra columns would just be redundant or could even skew the data. In these cases, it would always be better to remove these columns.

Figure 3 below shows the feature selection in correlations among different columns of datasets. From this heat map, the target variable is a column named "California Flag".

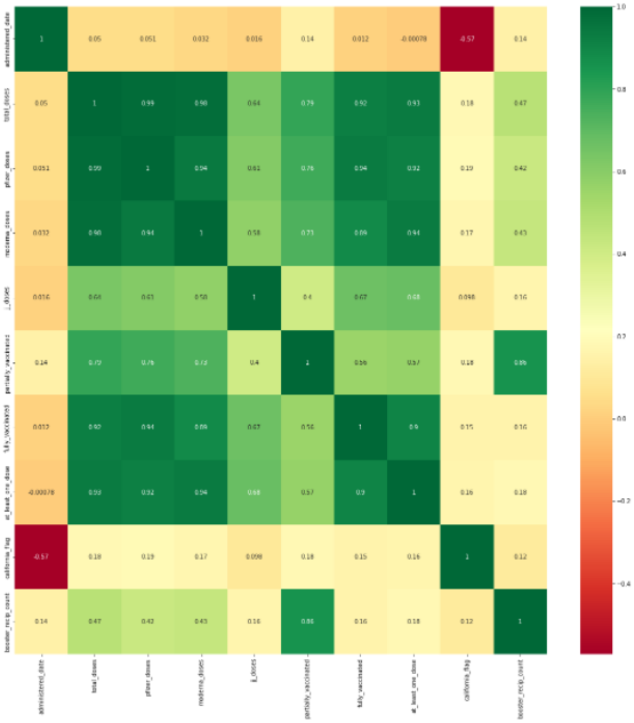

Fig.3. Feature Selection Heat Map

Feature Selection plays an important role in affecting the performance of the model. This can also include splitting a few columns to use as new features or adding in features from other datasets to improve the model. In our dataset, we removed the date column as there was no advantage in encoding a feature whose values were all distinct values. We would need to select the correct features from the 19 features.

C. Features in the Dataset

The dataset that was used from the 'California Open Data Portal' has 36580 entries and 19 features (columns). The features that are available are county, California flag (if the recipient is a resident of California) administered date, total doses, cumulative total doses totally as well for each vaccine individually (Pfizer, Moderna and Johnson and Johnson) which amount to eight features. It also includes other features like if the patient is partially vaccinated, fully vaccinated, at least one dose or the number of booster counts. While we would not be using each of these columns, the vaccine counts provide us invaluable information in building an appropriate model.

The feature that is county does not influence the effectiveness of the vaccines hence this feature was dropped due to the lack of influencing factors. The cumulative value features for the individual vaccines are redundant features that would not be necessary to the model as the actual dose count plays a better role in building a better model. Hence, we dropped a total of seven features.

II. Preprocessing

A. Dealing with the null values

Handling “NaN” values comes under the Feature Engineering and preprocessing part of Machine Learning. To deal with “NaN” values in a dataset, we need to understand how they came out to be and each way of handling “NaN” is based on the model being built. It also can be decided on other factors such as:

- If the dataset has multiple rows with multiple “NaN” values missing. In this case, dropping those rows would be the best solution
- If the dataset has a feature with multiple “NaNs”, then there could be multiple solutions to it such as replacing them with a completely new value, replacing them with the most occurring value, replacing them with zeroes or with the mean or median values.

This dataset that was chosen has about 615 missing or ‘NaN’ values which need to be handled. We noticed that the missing values were very random and low in number compared to the size of the dataset which has about 34000 rows. There were not many rows with a lot of missing values. We decided to replace the 615 missing values with the median values of that feature.

B. Encoding the dataset

All machine learning models require the data to be in a numerical form rather than a categorical form. This requires us to encode the categorical data to numbers so we can train and fit an appropriate model.

There are many techniques for encoding data and the three of the most popular techniques in this are Ordinal Encoding, Label Encoding and One-Hot Encoding. The technique that needs to be used depends on the variables that are present in the feature. There are two types of possible variables:

- Nominal Variable where the values have no relationship between themselves i.e., categorical variables.
- Ordinal variable where the values have a ranked ordering between the values.

We would be using Label Encoder to encode the two columns that are “administered_date” and the “california_flag” as the California flag has two distinct values that are either a California resident or not and they are encoded as 1 and 0 respectively. Each of the administered dates are distinct values hence they are Label Encoded as well.

C. Scaling the dataset

Feature Scaling in machine learning is one of the most important steps which is performed during the preprocessing stage of Machine Learning. The mostly used two techniques in scaling are Standardization and Normalization.

Standardization is used when we want to arrange our values with the mean as zero and a variance of 1. Normalization is used to arrange between two significant values such as [0,1] or [-1,1].

For our dataset and features, we decided to go with a Min - Max Scaling which is a normalization technique. It is one of the predefined methods that are available in the sklearn.preprocessing library. The transformation is given by:

$$(1) \qquad X\_std = (X - X.min) / (X.max - X.min)$$

$$(2) \qquad X\_scaled = X\_std * (max - min) + min$$

Where min and max are the range of the feature. This transformation is used as an alternative to zero mean, unit variance scaling. The X_scaled from (2) is the value that would be replaced in the feature column. If the min and max values are not specified, it would assign the values between the range of [0,1].

Fig 3. shows how the scaled columns look after plotting them using matplotlib.pyplot.

The same scaling method, min-max-scaler, is applied on the X train datasets and Y test datasets. In addition, we used the train test split from sklearn with 33% of the total datasets as test set, and it had 64 random states.

III. Training the model

A Machine Learning training process is defined as a process when a ML algorithm is given sufficient training data to learn from and that can be used to make correct predictions. This iterative process is called “Model fitting”.

There are two types of learning that are available. One is supervised learning where both the inputs and output values are available, and the model is used to predict outputs for which the inputs were not present in the training data set. The other method is unsupervised learning which mainly involves determining patterns in the data. The accuracy of the model in unsupervised learning improves accuracy based on the correlation to the expected patterns or clusters. There are various methods that are used to fit a model under a regression method or classification method, some of the most used ones are:

- Linear Regression
- Nonlinear Regression
- Support Vector Machine (SVM)
- Neural Network (Shallow and Deep)
- Decision Trees
- k Nearest Neighbor (kNN)
- Naive Bayes

We plan to use Linear Regression, SVM and Neural Networks to determine the best model for our data.

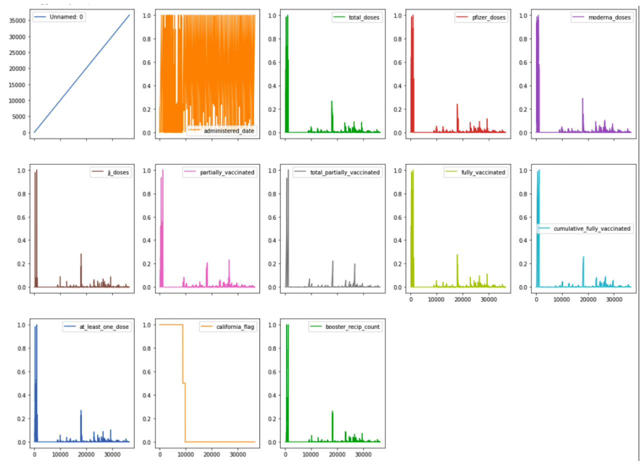

Fig. 4. Features of the dataset by plotting using matplotlib

A. Decision Tree Classifier

According to [6], the advantages of using decision tree classification are plenty. First, it is easy to be viewed and structured in a diagram. Each path of decision is logical and clear, determined by the information gained from entropy. Secondly, the cost of the decision tree is the tree depth times the number of samples splits. It is smaller compared to some large neural network models. Furthermore, it is capable of handling both categorical and numerical data. It can also handle problems that have multiple outputs.

The equations of calculating Entropy and Information Gain are listed below:

$$(3) \qquad Entropy(D) = \sum_{i=1}^{c} - p_i log_2(p_i)$$

$$(4) \quad Gain(D, A) Entropy(D) \sum_{j=1}^{v} 1 \frac{|Dj|}{|D|} Entropy(D_j)$$

B. Random Forest Classifier

Random Forest classification, from [6], is close to the CART algorithm. It can handle both classification and regression problems. The structure of the Random Forest is like a Decision Tree. It is a collection of simple Decision Trees where each tree produces a set of predictions given a set of inputs. Random Forest Classifier has its own advantage due to its robustness to handle different types of data. It is also accurate in classification results and estimating variables. The drawback of using Random Forest is that it is hard to be viewed graphically. It sometimes can overfit datasets with noise inside the classifications or regressions.

C. K Nearest Neighbors Classifier

K Nearest Neighbors Classifier is one of the most straightforward techniques where we classify a datapoint with respect to the points closest to it which are the neighbors and then use the neighbors to determine the datapoint that we want to identify. The neighbors are separated as classes and the examples will be classified based on their neighbors classes.

k-NN has many advantages such as:

- Its transparency makes it easier to program and debug.
- Various noise reduction techniques designed specifically for k-NN that improve its accuracy.
- It also can be applied to data where feature vectors are not available.

k-NN also has disadvantages such as:

- It computes the neighbors during run-time so having a large training set can reduce its performance.
- k-NN is extremely sensitive to outliers so special care has to be taken to handle them as they could greatly skew the data.

D. Neural Network Classifier

We also applied simple fully connected neural networks to train and predict the model. For the neural network, we picked 8 layers with relu as our activation function in each layer, and also used relu for our output layer. Relu should not drop any values except for negative values.

We expand the neurons first trying to get as much information as possible, then reduce down the neurons to get the most important and specific information. Since we already had relu applied, we did not add any dropouts to control the overfitting problem.

We choose mean_squre_error as the Loss function, which gives the specific error that we got, trying to reduce the loss so we can get a better prediction.

The result looks perfect to us, and if we are trying to train the model to predict fully_vaccinated, we are getting accuracy around 10-20%. But if we are trying to train and predict the booster_recip_count, we could get around 60-70% accuracy.

E. Support Vector Machine

The advantage of using a support vector machine is that it's efficient in classifying high dimensional data, particularly when the dimensions are bigger than the number of samples. However, the drawback of using SVM is long training time compared to other classifiers. And it does not provide estimates of probabilities directly to users.

In addition, the SVM performs like k-nearest neighbors but in the vector space rather than node coordinates. It's important for SVM to find the suitable range of vector wise classification. By check in Sklearn, SVM has many different models including support vector regression and support vector classifier.

F. Common Mistakes

The above algorithms are classifiers that can deal with either linear or non-linear data. However, because they are supervised learning, picking the relevant or more suitable labels in the training process can boost the accuracy by a considerable amount compared to less relevant labels.

The other common mistake is to use these models for large datasets. Compared to deep learning, they have less power in both prediction and capability in handling large inputs of data.

IV. Results

A. Figures and Tables

Cross validation results are shown below for Decision Tree Classifier and Random Forest Classifier targeting ‘fully_vaccinated’ columns:

TABLE I. Cross Validation For Decision Tree

Folds	Fold1	Fold2	Fold3	Fold4	Fold5
Accuracy	0.9627	0.9651	0.9592	0.9622	0.9614

TABLE II. Cross Validation For Random Forest

	Fold1	Fold2	Fold3	Fold4	Fold5
Accuracy	0.9851	0.9840	0.9798	0.9812	0.98184

The best parameters for Decision Tree Classifier are ‘max_depth’ of 5, and ‘min_samples_split’ of 10, and 30 ‘n_estimators’ for Random Forest Classifier.

TABLE III. K - Nearest Neighbors

	Accuracy	n_val
FNN	0.7234	4

TABLE IV. Neural Network Modules

	Accuracy	Error (mean squared error)
FNN	0.9168	1176816.0

TABLE V. Support Vector Machine

Kernel	Accuracy	Recall	Precision
Linear	0.9032	0.9032	0.8788

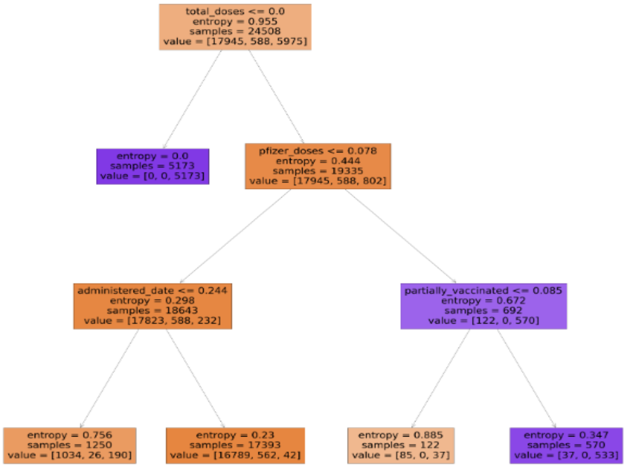

Fig 5. Decision Tree Classifier Plot

From figure 4, we can interpret the splitting process of the decision tree over X_train and y_train data. It is also a process of classification. The root node holds the pool of every data, and at second level, the pool splits when the information gain between the root node and the sum of its child is at maximum. Until the splitted pool becomes pure, which is completely uniform with entropy equal to zero, the split process will stop and leave the leaf node. Random Forest has similar criterions compared to decision tree process.

During training, as mentioned previously, the duration of training is a lot longer than decision tree based classifier. Below is the table for its performance:

The reason for not using 5 fold validation here is due to the long period of training. Perhaps with higher computational capabilities, this task can be achieved faster.

V. Data Visualization

This section will illustrate the data visualization of vaccinations during the COVID outbreak.

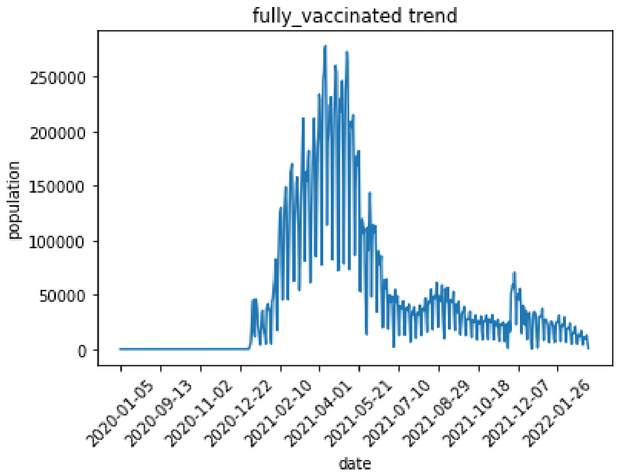

Fig 6. Date Vs. Fully_vaccinated Populations

The peak of taking vaccinations happened around the year end in 2020 to the start of 2021. This information told us the estimation time of where COVID started to spread.

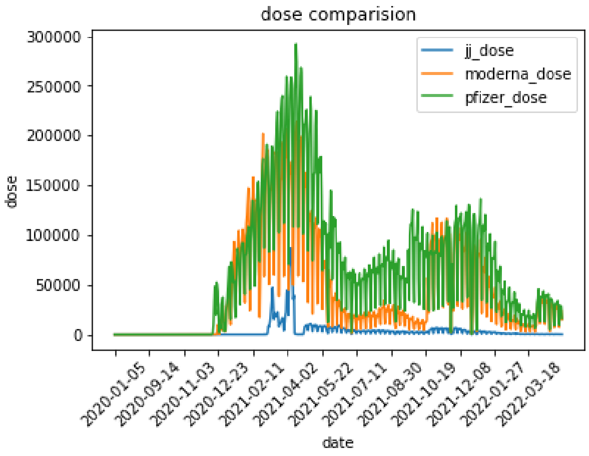

Fig 7. Date Vs. Multiple Vaccines Information

From figure 6 above, during the COVID outbreak, the Johnson & Johnson vaccines were taken the least and Pfizer were taken the most frequently.

VI. Conclusion

Although COVID situation has been alleviated by vaccines today, the virus is evolving with the development of vaccines. So, we are never too sure our health is secure from getting one. Therefore, this research of analyzing the impact of COVID is important. It's crucial for people to know when and where the outbreak will happen again. In this practice, we have only used a small size of data samples. For better predictions and forecasting, we plan on analyzing data with bigger size. In addition, to fit the large amount of data, the model will be changed to deep learning with neural networks.

VII. Authors And Affiliations

Prasaanth Radhakrishnan - Computer Engineering, San Jose State University

Bruce Jiang - Computer Engineering, San Jose State University

Wai Yin Suen - Software Engineering, San Jose State University

Hanyu Hu - Software Engineering, San Jose State University

References

[1] E. Gambhir, R. Jain, A. Gupta, and U. Tomer, "Regression Analysis of COVID-19 using Machine Learning Algorithms," 2020 International Conference on Smart Electronics and Communication (ICOSEC), 2020, pp. 65-71, doi: 10.1109/ICOSEC49089.2020.9215356. Link: https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=9215356

[2] S. Kodera, E. A. Rashed, A. Hirata, "Estimation of Real-World Vaccination Effectiveness of mRNA COVID-19 Vaccines against Delta and Omicron Variants in Japan", MDPI Vaccines Journal, 2022, doi: 10.3390/vaccines10030430. Link: https://www.mdpi.com/2076-393X/10/3/430/htm

[3] San. Kumar, Sum. Kumar, A, Singh, A, Raj, "COVID-19 Data Analysis and Prediction Using (Machine Learning) and Vaccination Update of India", SSRN, 17 May 2021, Available: Doi: 10.2139/ssrn.3847564, Link: https://papers.ssrn.com/sol3/papers.cfm?abstract_id=3847564

[4] K. Jeffrey, M. Kenneth, "History and Recent Advances in Coronavirus Discovery", The Pediatric Infectious Disease Journal, pp S223-S227, Available: Doi: 10.101097/01.inf.0000188166.17324.60, Link: https://journals.lww.com/pidj/fulltext/2005/11001/history_and_recent_adva
s_in_coronavirus.12.aspx

[5] Data Source, "Statewide COVID-19 Vaccines Administered By County", CA OPEN DATA PORTAL, URL: https://data.ca.gov/dataset/covid-19-vaccine-progress-dashboard-data/resource/c020ef6b-2116-4775-b11d-9df2875096ab

[6] B. Gupta, A. Rawat, A. Jain, A. Arora, N. Dhami, "Analysis of Various Decision Tree Algorithms for Classification in Data Mining", *International Journal of Computer Applications*, Vol. 163- No. 8, April 2017.

[7] Cunningham, Pádraig and Sarah Jane Delany. "k-Nearest Neighbour Classifiers: 2nd Edition (with Python examples)." *ArXiv* abs/2004.04523 (2020): n. pag. Link: https://arxiv.org/pdf/2004.04523.pdf

[8] Rish, Irina. (2001). An Empirical Study of the Naïve Bayes Classifier. IJCAI 2001 Work Empir Methods Artif Intell. 3.

[9] Dreiseitl, Stephan & Ohno-Machado, Lucila. (2002). Logistic regression and artificial neural network classification models: A methodology review. Journal of biomedical informatics. 35. 352-9. 10.1016/S1532-0464(03)00034-0. https://stemeducationjournal.springeropen.com/track/pdf/10.1186/s40594-018-0152-1.pdf

[10] G. Staff, "The COVID-19 vaccine race", Jan. 12, 2022, Access:

The COVID-19 vaccine race | Gavi, the Vaccine Alliance