COVID-19 Cases Prediction Models And Important Features in California

A Project Report

Presented to

Data 228

Spring, 2022

By

Nghi Nguyen, Meiqing Zhen, and Edward Montoya

May 22, 2022

Table of contents

Abstract

The COVID-19 pandemic has been ongoing for a couple of years now and continues to disrupt nearly all aspects of everyday life. The purpose of this project is to find key features that have a correlation with predicting, and modeling future COVID-19 cases in the state of California. There currently exist models that attempt to tackle this question, but they are primarily focused on a country level and are limited in feature scope. The project approaches this challenge by adapting a diverse set of features with a particular focus on demographics for California. The project methods incorporate the use of five different datasets; Weather, COVID-19 Cases and deaths and tests, COVID-19 Hospital, COVID-19 Vaccination by Demographics, and COVID-19 Deaths by Demographics. This array of datasets allow for the use of implementing four different machine learning models Linear Regression, Random Forest, XGBoost, and Prophet for making predictions. The project utilizes MSE, RMSE, and R^2 as a metric to compare all of the models. The findings of this project conclude that XGBoost was the most accurate at making predictions with a R^2 score of 0.8906, and a MSE of 0.0014. Based upon the results of this project it is evident that there is a high level of difficulty in trying to accurately predict future COVID-19 cases in California, but as more features are incorporated such as demographic data and more advanced machine learning methods are implemented, then the ability to predict with a higher level of accuracy improves.

Keywords: COVID-19, Time Series Forecasting, Supervised Learning

1. Introduction

COVID-19 (or Coronavirus) began in China at the end of 2019 and rapidly spread around the world. The U.S. had the very first COVID-19 case at the beginning of 2020, and as of May 2022, the total number of COVID-19 cases in the U.S. was 82,820,565 and 101,130 of them were new cases on that same day (Centers for Disease Control and Prevention). The pandemic has led to many consequences for the economy, society, and public policies. For the economy, many businesses and organizations have shut down, people have been laid off, and the unemployment rate peaked for a long period. There was a time when necessary goods ran out, such as hand sanitizers, masks, toilet papers, etc. that created a crisis in society. The pressure from the pandemic has led to many social panic and mental issues. Many people who contracted the disease were dead or suffered. As of May 2022, the U.S. had a total of 998,512 deaths and 19,207 hospitalization cases because of COVID-19. Additionally, various COVID-19-related political policies have been enacted in the U.S. from 2020, such as shelter orders that constrained the number of people going outside except for necessary activities or the mask requirement rules when every person had to wear a mask when they are in public places. This project developed four machine learning prediction models: Linear Regression, Random Forest, XGBoost, and Prophet, and determined the best model among those four to predict the total number of COVID-19 in California based on a set of different features. The features focus more on demographic information, e.g., more demographic-related parameters. Another target is to identify the factors that are strongly correlated to the number of COVID-19 cases in California. The success of the project will help us prepare for the future; for example, if a specific age group is more likely to get COVID-19, the government can have helpful policies to encourage that group to get the vaccines.

There are two expected results for this project.

- A machine learning model that effectively predicts the total number of COVID-19 cases in California using a set of relevant parameters.
- Based on the model, identify the important features that are strongly correlated with the number of COVID-19 cases in California.

The deliverables of the project are as follows.

- A project plan that gives a general idea of the project and the general plan of the phases and necessary tools.
- A live presentation that explains the entire project as well as showing the resulting demo. The live presentation also takes all the questions that the audience might have.
- A project report that provides all the details of the project, including each phase and how the project processes.
- A code link from GitHub. All coding sources of the project are uploaded to a GitHub account and the GitHub link will be available for everyone to access.
- A Big Data Use Case report that describes the project from the perspective of an application provider with the assumption that a user wants to know about an application before he actually uses it.

COVID-19 prediction models have been developed in a variety of projects, and the products are not for commercial use. For example, the Institute for Health Metrics and Evaluation (IHME) publishes a free website called COVID-19 Projections that contains a lot of COVID-19-related-factor predictions. The following link leads to their website https://covid19.healthdata.org/global?view=cumulative-deaths&tab=trend

- Introduction: discuss the project's goal, objective, problem, motivations, impact, results, expected deliverables, market research, and structure.
- Project background and related work: give an idea of the project's background, technologies, and literature survey.
- System Requirements: describe the inputs, the behavior, and the outputs of the projects. This section also points out all the necessary technologies and tools.
- System Design: provide information on the system architecture design, the database design, the preparation process, design problems, solutions, and patterns.
- System Implementation: narrate the model development phase, including how the models are implemented.
- System Testing and Experiment: illustrates the evaluation metrics and shows the evaluation results.
- Conclusion and Future Work: summarizes the project and discusses the future direction.

2. Project Background and related work

Many works have been done on the COVID-19-related topic. Most of the papers we read work on worldwide, nationwide, or multi-statewide levels. Not many projects have built an effective model in California only. Furthermore, they mainly use the historical data of the pandemic to forecast future trends. Our project is a little different: we concentrate more on demographic aspects; for example, the vaccination data are categorized by age group, gender, race, and ethnicity. This project also analyzes what parameters strongly affect the number of COVID-19 cases in California. Amazon Web Services (AWS) such as AWS S3, ASW Glue,

AWS QuickSight, and Jupyter Notebook are the main tools to store and build machine learning models.

Tang et al. (2021) propose a combination model of Deep Deterministic Policy Gradient (DDPG) and Long Short-Term Memory (LSTM) models that can improve the prediction accuracy of COVID-19. The data used is the pandemic and economic statistical data in several states, such as California, New York, Texas, Washington, Iowa, and Florida. The pandemic data includes the number of deaths, new deaths, confirmed diagnoses, and new confirmed diagnoses. The economic data contains credit/debit card spending levels. The research points out how the consumer spending index affects the number of COVID-19 cases and uses the economic data to differentiate their approach and the traditional ones.

Zhou et al. (2020) build a spatiotemporal epidemiological forecasting model that can give the projection of the COVID-19 cases in 3109 counties in the U.S. Furthermore, it provides risk notifications for the residents in a specific area or for the travelers who have a planned trip in the U.S. The highlight of the project is that it counts on the state-level control measures such as social distancing and intercounty travel. This project focuses on biology data that includes factors such as self-immunization rate, basic disease transmission rate, etc. In the end, the authors mention that employing local information can significantly improve the precision of the prediction.

Pokkuluri et al. (2020) develop a preliminary classifier to predict COVID-19 deaths, the number of affected people, and the number of recovered people based on their historical data. According to their research paper, the classifier is called hybrid non-linear cellular automata (HNLCA) and is trained and tested with a total of 29,863 datasets. The authors also indicate that the model is compared to different machine learning methods of regression, Support Vector

Machines (SVM), Support Vector Regression (SVR), AdaBoost, and short-long term memory. The accuracy of HNLCA is 78.8%, which appears to be higher than the other methodologies.

Gupta et al. (2021) employ the data input of COVID-19 case counts per county per day and county populations in California, Indiana, and Iowa to build a prediction model for the number of COVID-19 cases. The model they propose is tested and returns the specificity and accuracy that is higher than 95% in general. The authors do not use machine learning in building the model, their model development phase includes five steps: define COVID-19 pandemic, filter important features, depict case count data over time, iterate case rates for population bands, and make ultimate model rules.

Santosh (2020) conducts a study that has a similar idea to our project: he points out the importance of unprecedented uncertainties to the COVID-19 situation, which are hospitalization, test rate, demographics, population density, vulnerable people, and poverty. The author also discusses different models that have been built and indicates the abundance of works that are built solely on the historical data of the parameters, e.g., using historical case data to predict the number of cases in the future. As this is a study paper, it does not propose a methodology to develop a COVID-19 prediction model; however, it hugely inspires our project as the parameters we use are somewhat similar to what Santosh mentions.

3. System Requirements

3.1 Raw datasets

A set of COVID-19-related parameters are required for this project. To fulfill this requirement, different datasets from different resources are collected.

3.1.1 California weather dataset.

The weather dataset includes daily data of 1,441 stations in California from January 1, 2020, to April 4, 2022. There are 50 attributes, some of which contain all null values. However, only four parameters will be used for the project: daily average temperature,daily maximum temperature, daily minimum temperature, and temperature at the observed time. The data is downloaded from the National Centers for Environmental Information (NOAA) and can run only three quarters at a time. Therefore, the raw weather data includes nine separate csv files, which are combined into one unique file named “weather_combined_csv.csv” using os and glob libraries in Jupyter Notebook. The csv file then is uploaded to AWS S3 for the cleaning and transformation process using AWS Glue.

3.1.2. California COVID-19 cases and deaths and tests dataset.

The dataset derives from the California Department of Public Health (CDPH) with 45,873 rows and 17 columns. The features chosen are date, daily cases, and daily total tests. The data is collected from 02/01/2020 to 02/21/2022.

3.1.3. California COVID-19 hospital dataset.

The dataset derives from the CDPH, has 39,590 rows and nine columns, and was collected from 03/09/2020 to 03/05/2022. All columns in the dataset are chosen for the project; however, they will be aggregated in the data processing phase. The parameters include:

- County where the hospital locates (in California)
- Date
- The number of patients hospitalized in an inpatient bed who have laboratory-confirmed COVID

- The number of patients hospitalized in an inpatient bed without a laboratory-confirmed COVID diagnosis who, in accordance with CDC's Interim Public Health Guidance for Evaluating Persons Under Investigation (PUIs), have signs and symptoms compatible with COVID
- The number of patients currently hospitalized in an inpatient bed who have suspected or confirmed COVID
- The total number of beds in the facility, including all surge beds, inpatient and outpatient post-surgical beds, labor and delivery unit beds, and observation beds.
- The number of laboratory-confirmed positive COVID patients that are in the ICU at the hospital
- The number of symptomatic patients, with tests for COVID pending laboratory confirmation, that are in the ICU at the hospital
- The number of ICU beds available at the hospital

3.1.4. California COVID-19 vaccination by demographics.

The dataset derives from the CDPH with 9338 rows and 19 columns. The parameters will be used are date, demographic values, demographic categories, and total doses. including Age Group, Gender, and Race/Ethnicity. The demographics category has age groups (0-17, 18-49, 50-64, 65+, and missing), gender (female, male, unknown), and race ethnicity (American Indian or Alaska Native, Asian, Black, Latino, Multi-Race, Native Hawaiian and other Pacific Islander, Other, and White). The data is collected from 01/05/2020 to 02/21/2022.

3.1.5. California COVID-19 deaths by demographics.

The dataset derives from the California COVID-19 State Dashboard, which is organized by CDPH. The data is updated every Tuesday and Friday, excluding holidays. The dataset has

12431 rows and 8 columns, including demographics category and values, total cases, percent of cases, total deaths, percent of deaths, percent of California population, and date. The demographic groups are similar to the information mentioned in section 4. The data is collected from 04/13/2020 to 04/13/2022.

3.2 Technology requirements

The technology requirements for this project include:

- Google Drive, Google Colab
- Jupyter Notebook
- Tableau Prep Builder
- AWS Glue, AWS S3, Amazon QuickSight

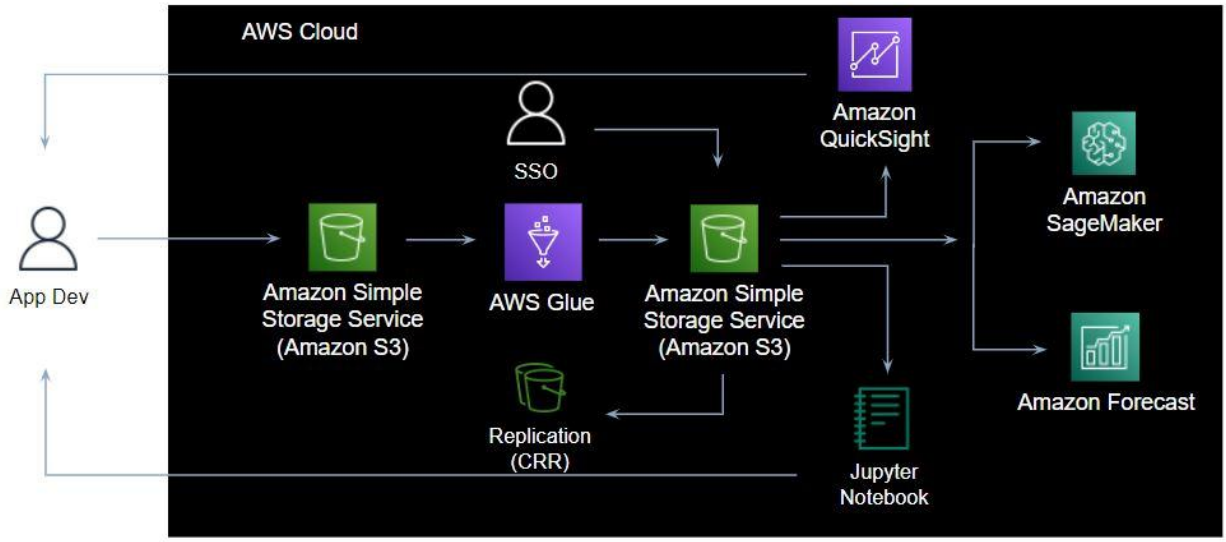
4. System Design/Data Preparation

The dataset design consists of working with primarily five separate datasets. For the Weather dataset it requires combining 9 separate (.CSV) files using a Jupyter Notebook. This allows for the filtering of unnecessary fields for the dataset, and keeping the desired five attributes including the date, precipitation, average temperature, max temperature, min temperature, and temperature at the time of observation. The COVID-19 Cases and Deaths and Tests, and COVID-19 Hospital datasets were both cleaned using a Jupyter Notebook. The COVID-19 Cases and Deaths dataset required the use of aggregation to combine all of the counties into a single value based on the correlated date. The COVID-19 Hospital dataset required the use of imputing null values with zero. This dataset also required the use of aggregation to sum all of the counties into a single correlated date. Finally, the COVID-19 Vaccination and Demographics, and COVID-19 Deaths by Demographics datasets were cleaned by using Tableau Prep Builder. In the Tableau environment both datasets were cleaned the exact

same way, which consisted of cleaning missing values by imputing with zero, dropping columns that don't intuitively make sense to have included, aggregating counties into a single correlated date, and performing pivots for rows into columns. Finally all of the datasets were normalized with min-max normalization, which places the values from zero to one. This was to prepare for the machine learning process. It allows for all of the features to have the same scale against one another, and it ensures that not a single feature would dominate the others in the training process.

Figure 1

Project Architecture Design



Note. This project is trying to employ big data applications to process data.

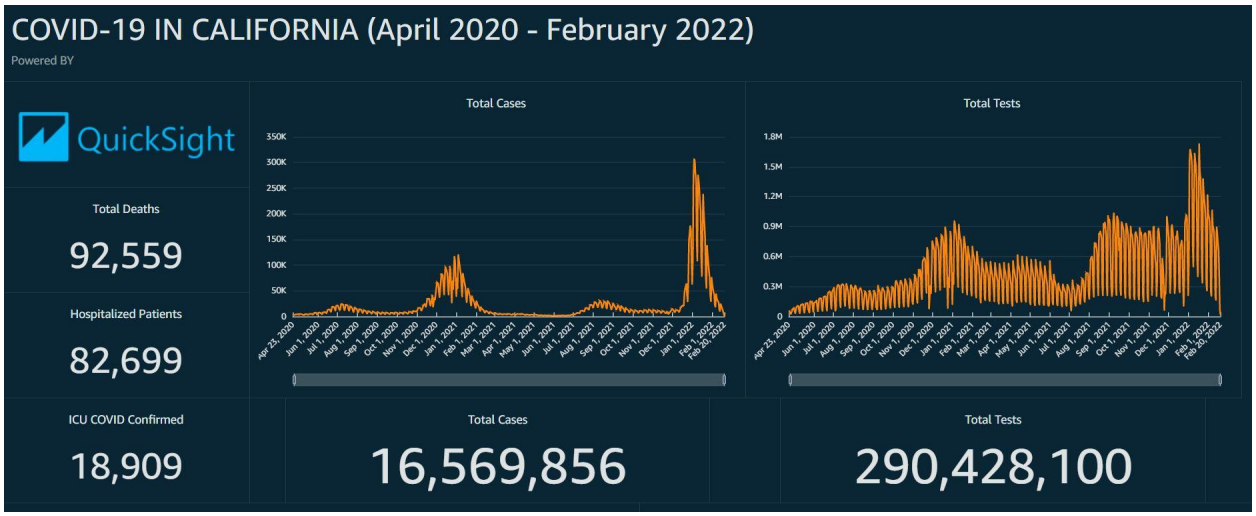
The architecture design (Figure 1) for the project primarily relies upon the AWS Cloud environment. The data was cleaned by using Jupyter Notebook, and Tableau Prep Builder. Once this process was done, the datasets were loaded into an Amazon S3 bucket. This allowed for the data to be utilized by AWS Glue for the ETL process. The data within Glue required minor transformations to achieve the combined table format that we desired. This was done by using joins to combine all the datasets by way of the date. The data was then placed back into a S3

bucket. This bucket implemented Cross-Region-Replication, which allowed for the data to be backed up. Additionally, Single-Sign-On (SSO) was implemented to allow the whole team to access the S3 bucket.

Since the data was now combined into a single dataset, it allowed for the team to visualize the dataset with Amazon QuickSight in hopes to gain some insights and understanding that could be applied to the machine learning models. This resulted in the implementation of a dashboard (Figure 2).

Figure 2

Data Insights



Note. The dashboard is developed using Amazon QuickSight.

Finally, the architecture design concluded by trying to make use of implementing Amazon SageMaker and Amazon Forecast. Unfortunately, due to time constraints, and the steep learning curve of each tool, this approach was not feasible. Instead the approach implemented the deployment of models by using Jupyter Notebook. Before training the models, the combined table is normalized using max-min normalization that scales all the parameters to the range zero to one. Normalization helps it easier to compare the variables and find out duplicate parameters.

5. System Implementation/Model Development

5.1 Linear Regression

First of all, we consider the covid 19 cases prediction as a supervised learning, so we split the dataset with shuffle into training and testing dataset with the ratio of 20%. Then, we start with a very basic machine learning model, linear regression, which is commonly used for predicting data. It models the linear relationship between independent variables and a dependent variable. To find the best linear regression model with training data, the sum of squared errors error function is commonly used. The error measures the difference between predictive values and actual data. In this project, we employ multiple Linear regression, which allows for two or more independent variables. It is defined in Equation 1.

$$y_i = \beta_0 + \beta_1 x_{i1} + \quad \cdots \quad + \beta_p x_{ip} + \epsilon_i = x_i^T \beta + \epsilon_i , \qquad i=1, \dots , n, \tag{1}$$

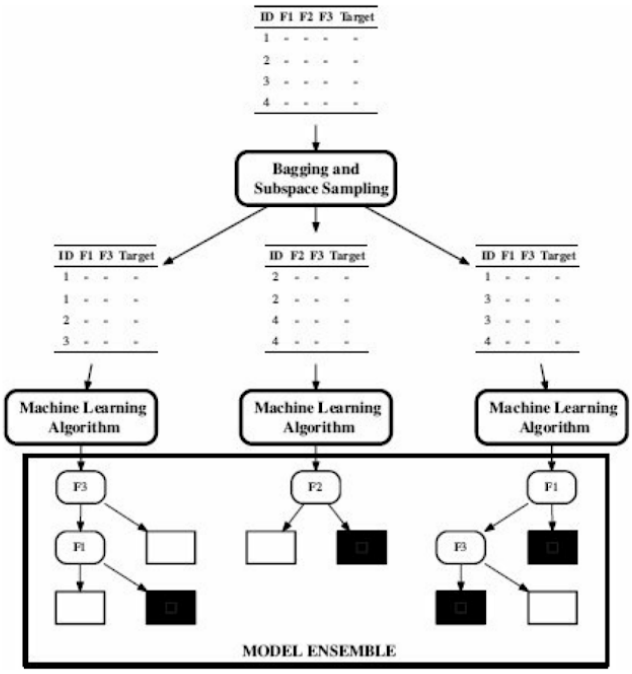
where the x is the feature values, and the beta is the weights of features (KELLEHER, 2020).

5.2 Random Forest

But we are not satisfied with one basic model, we want to try ensemble models to see if there is any improvement. So, we consider employing Random Forest. The architecture of a standard random forest shows in Figure 3. It combines bagging, subspace sampling, and decision trees. It models multiple decision trees based on the datasets created via bagging and subspace sampling. Bagging is taking multiple samples with replacement from the entire dataset while subspace sampling the subsets of independent variables in the model (KELLEHER, 2020).

Figure 3

Random Forest Architecture



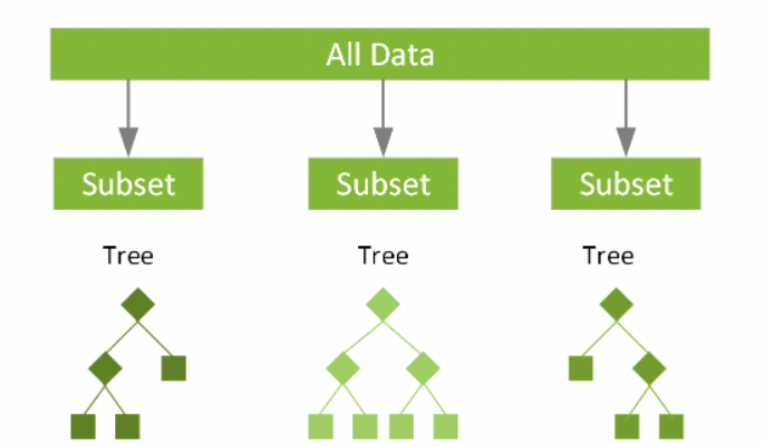
Note. A standard Random Forest architecture shows how it combines bagging, subspace sampling, and decision trees.

5.3 XGBoost

After that, we want to try another ensemble model. That is XGBoost, extreme gradient boosting. It is a model similar to random forest but with the gradient descent algorithm. It improves a single weak model by combining it with other weak models in order to generate a strong model. Gradient boosting makes the model learn from the error of prior decision trees. The final prediction is a weighted sum of all of the tree predictions. The architure of XGBoost is shown in Figure 4. (What is XGBoost?)

Figure 4

XGBoost Architecture



Note. A standard XGBoost which takes subsets from the entire dataset.

5.4 Prophet

Up to now, we only employ supervised machine learning. We want to know how a time series forecasting model would perform in this case. So, we introduce the Prophet. It is a model released by Facebook in early 2017. It works best with time series that have strong seasonal effects. It is robust to missing data and shifts in the trend, and typically handles outliers well. The basic function shows in Equation 2. As this model is time series forecasting, we include the time and exclude all features into the model without splitting the dataset and shuffling the dataset.

$$y(t) = g(t) + s(t) + h(t) + \epsilon_t \quad (2)$$

where g(t) is the growth term, s(t) is the seasonality term, h(t) is the holiday term, and e(t) is the error term.

6. System Testing and Experiment/Evaluation

6.1 Evaluation Metrics

The evaluation metrics to compare different models are Mean Square Error (MSE), Root Mean Square Error (RMSE), and the Coefficient of determination (R^2).

The MSE measures the average of the squares of the errors, which is defined as the Equation 3.

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(Y_i - \widehat{Y}_i)^2 \quad (3)$$

where n is the number of data points, Y_i is the observed values, and $\widehat{Y}_i$ is the predicted value.

The RMSE measures the square root of the average of squared errors. Equation 4 defines the formula.

$$\mathrm{R}MSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(Y_i - \widehat{Y}_i)^2} \quad (4)$$

where n is the number of data points, Y_i is the observed values, and $\widehat{Y}_i$ is the predicted value.

The R^2 measures the variance portion of the dependent variable predicted by the independent variable. The formula is defined in Equation 5.

$$R^2 = 1 - \frac{RSS}{TSS} \quad (5)$$

where RSS is the sum of squares of residuals, TSS is the total sum of squares.

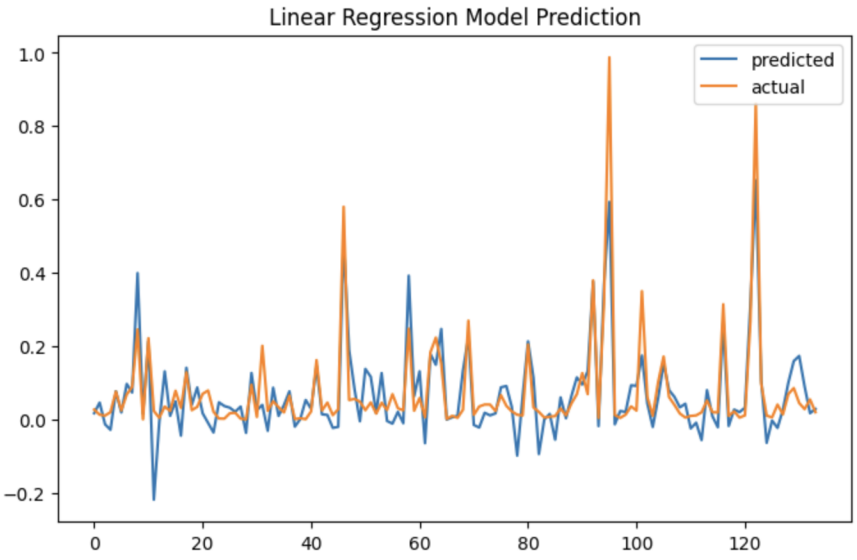
6.2 Result

The Figures 5, 6, and 7 are the graphs of the activity data and the predicted values by the Linear Regression, Random Forest, and XGBoost. The blue line represents the predicted values

while the orange line represents the actual values. The graph doesn't show covid 19 cases changing with time because we shuffle the dataset at the beginning. The overlap area means there is limited error between predicted values and actual data. Comparing three graphs, it is easy to notice that the predictions of Random Forest and XGBoost are more similar to the actual data.

Figure 5

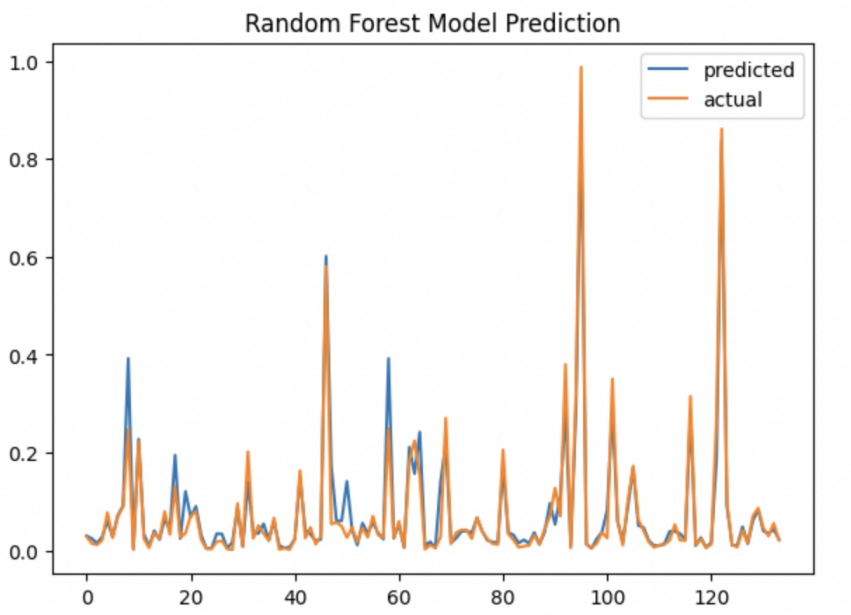
Linear Regression Model Prediction



Note. The x axis does not represent case time.

Figure 6

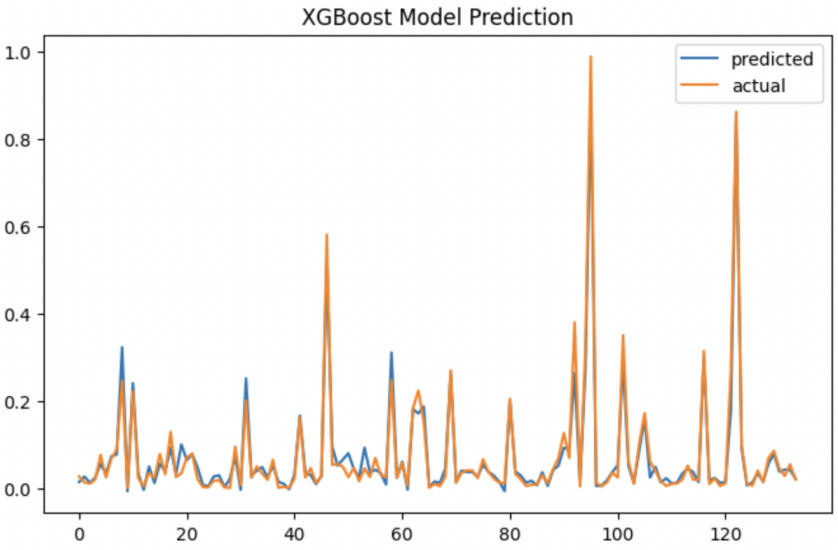
Random Forest Model Prediction



Note. The x axis does not represent case time.

Figure 7

XGBoost Model Prediction

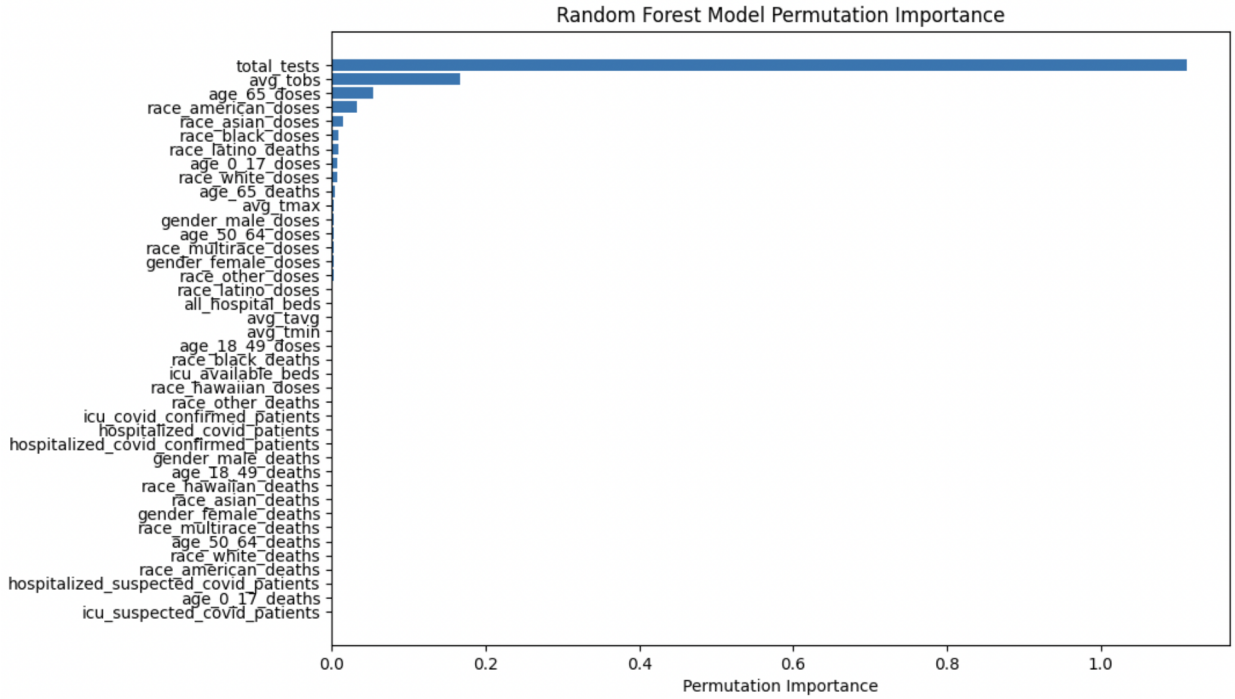


Note. The x axis does not represent case time.

The permutation importance of Random Forest and XGBoost demonstrates respectively in Figure 8 and 9. Analyzing the two graphs, it is easy to see that the shuffle of total test values increases model error the most, which means the two models rely on the total tests feature. Besides, average temperature at the observatory, elders' vaccination doses, and American vaccination doses also contribute a lot to two models.

Figure 8

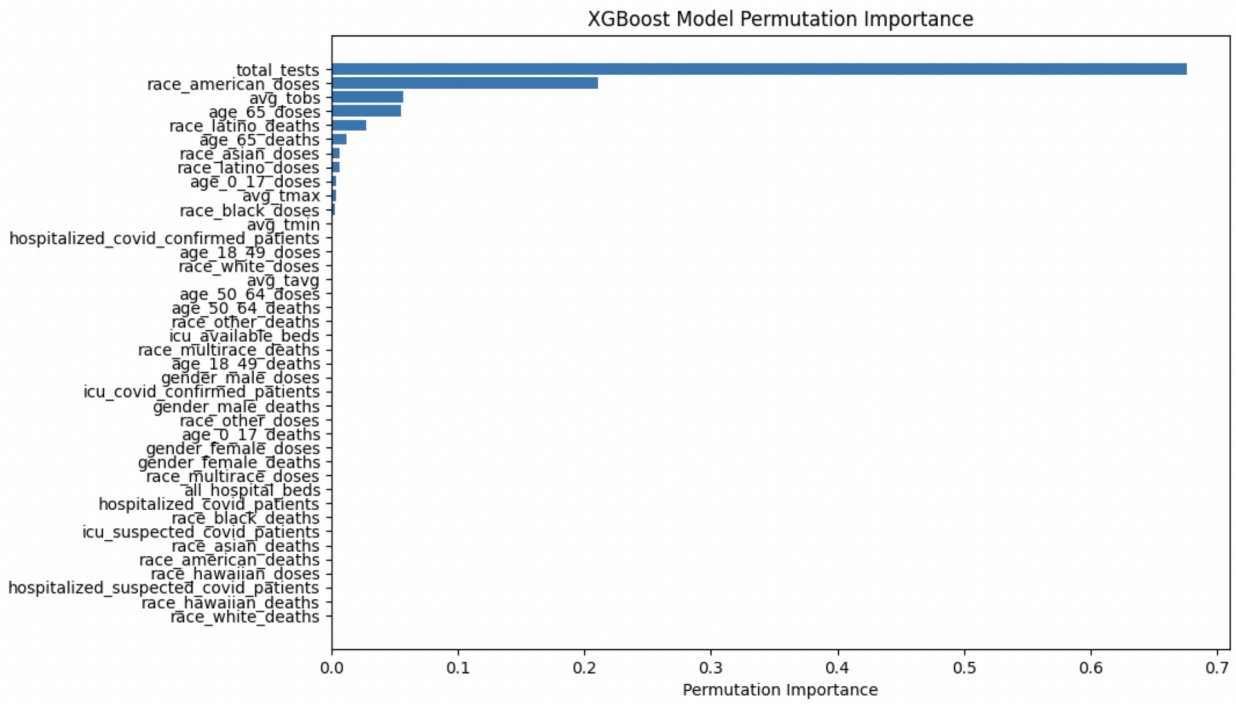
Random Forest Model Permutation Importance



Note. Not all features show permutation importance in the graph.

Figure 9

XGBoost Model Permutation Importance

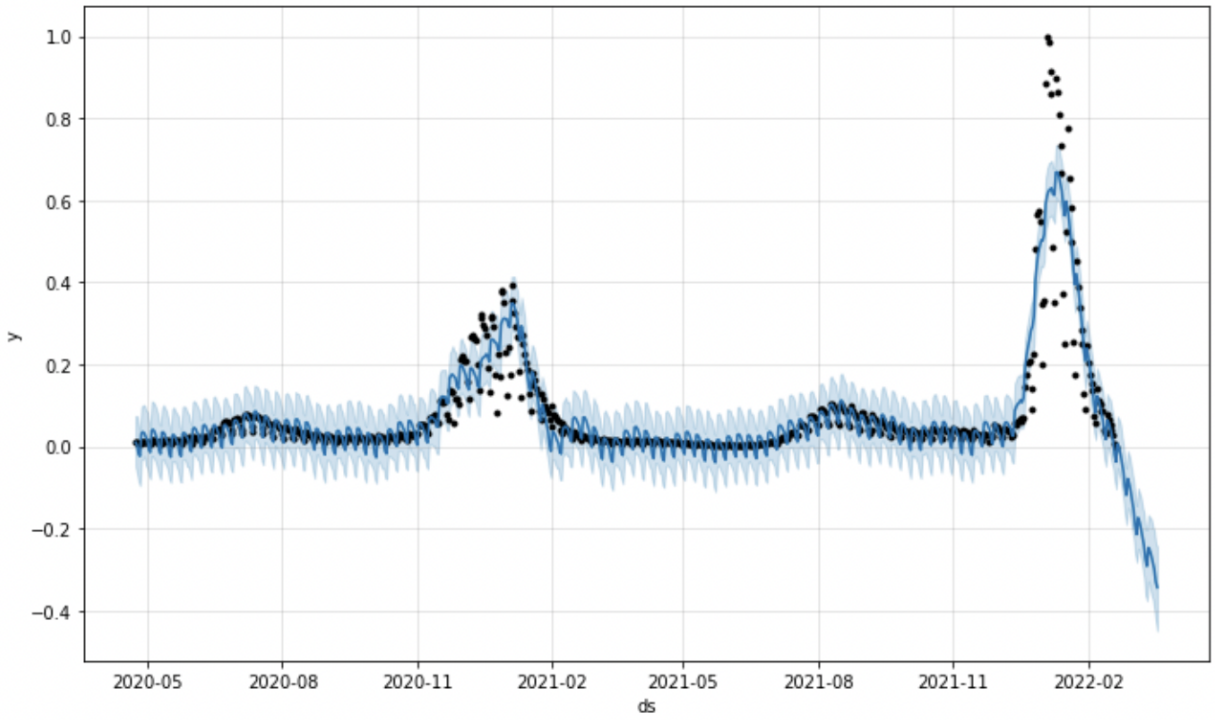


Note. Not all features show permutation importance in the graph.

Figure 10 demonstrates how the Prophet predicted the future cases and the comparison with actual data. It can be noticed that the model catches the trend of COVID-19 cases well most of the time, especially when the trend is steady. However, even though the model predicts there will be a large number of new cases in January, 2022, the model can't catch the outbreak at the highest point.

Figure 10

Prophet Model Time Series Prediction



Note. The black dots of the graph are the actual data while the darker blue line is the model prediction, and the lighter blue area is the error of the forecast.

Table 1 is the comparison of four models using the evaluation metrics discussed above - MSE, RMSE, and R^2. Random forest and XGBoost perform better than the others with lower RMSE and MSE and higher R^2 values. Not surprisingly, linear regression, as a baseline model, performs the worst. Prophet, as a time series forecasting model, doesn't perform the best. However, there is a limitation of this model. We only include the total cases in this model, and now it performs relatively well. If other features such as total tests are included in the model, the model might be better.

Table 1

Four Models Comparison

Model	Model Type	MSE	RMSE	R^2
Linear Regression	Supervised Learning	0.0048	0.0696	0.7435
Random Forest	Supervised Learning	0.0012	0.0352	0.9345
XGBoost	Supervised Learning	0.0006	0.0250	0.9668
Prophet	Time Series Forecasting	0.0026	0.0513	0.8614

7. Conclusion and Future Work

Predicting COVID-19 future cases is helpful for the government to prepare for the outbreak. To predict the outbreak well, the datasets and model selection are significant. Some papers which predict COVID-19 cases only focus on historical cases while we employ several features, such as weather, hospital data, and demographics of death and vaccination. Some features, such as total tests, American vaccination doses, and the elders vaccination doses, are verified as important in the permutation importance of Random Forest and XGBoost.

Overall, Random Forest and XGBoost provide a better prediction as supervised machine learning algorithms. Their permutation importance indicates that total tests have more influence

in the model prediction, which conform to our understanding. More COVID-19 tests in the community could find more COVID-19 cases, especially potential patients who do not have significant symptoms. Prophet, as a time series forecasting algorithm, only employs historial cases and predicts the future cases. It predicts future cases relatively well, except for the extreme point in the COVID-19 outbreak.

There will be some improvements that could be made in the future. Firstly, using grid search to tune the hyperparameters of Random Forest and XGBoost would be possible to further improve the model performance. Secondly, considering deep learning, such as LSTM, would be a good choice for memorizing short term and long term time series data. Finally, to shorten the training and testing time of the machine learning model, data mining, such as feature selection and reduction, should be implemented. The features selection could be based on the features in the permutation importance of Random Forest and XGBoost which contributes more to the model.

Appendix

GitHub source code link: https://github.com/hoangkhanhnghi01/Data-228

References

Centers for Disease Control and Prevention. (n.d.). *CDC Covid Data tracker*. Centers for Disease Control and Prevention. Retrieved May 19, 2022, from https://covid.cdc.gov/covid-data-tracker/#datatracker-home

COVID-19 Projections. Institute for Health Metrics and Evaluation. (n.d.). Retrieved May 19, 2022, from https://covid19.healthdata.org/global?view=cumulative-deaths&tab=trend

Gupta, A. K., Grannis, S. J., & Kasthurirathne, S. N. (2021). Evaluation of a parsimonious COVID-19 outbreak prediction model: Heuristic Modeling Approach using publicly available data sets. *Journal of Medical Internet Research*, *23*(7). https://doi.org/10.2196/28812

Kelleher, J. O. H. N. (2020). Fundamentals of machine learning for Predictive Data Analytics: Algorithms, worked examples, ... and case studies. MIT Press.

Santosh, K. C. (2020). Covid-19 prediction models and unexploited data. *Journal of Medical Systems*, *44*(9). https://doi.org/10.1007/s10916-020-01645-z

Tang, X., Li, Z., Hu, X., Xu, Z., & Peng, L. (2021). Self-correcting error-based prediction model for the COVID-19 pandemic and analysis of economic impacts. *Sustainable Cities and Society*, *74*, 103219. https://doi.org/10.1016/j.scs.2021.103219

Pokkuluri, K. S., & Devi Nedunuri, S. S. S. N. U. (2020). A novel cellular automata classifier for covid-19 prediction. *Journal of Health Sciences*. https://doi.org/10.17532/jhsci.2020.907

What is XGBoost? NVIDIA Data Science Glossary. (n.d.). Retrieved May 15, 2022, from https://www.nvidia.com/en-us/glossary/data-science/xgboost/

Zhou, Y., Wang, L., Zhang, L., Shi, L., Yang, K., He, J., Bangyao, Z., Overton, W., Purkayastha, S., & Song, P. (2020). A Spatiotemporal Epidemiological Prediction Model to Inform

County-Level COVID-19 Risk in the United States. *Harvard Data Science Review*, *Special Issue 1*. https://doi.org/10.1162/99608f92.79e1f45e