COVID-19 Forecasting in Tennessee

Mathew Thomas

Western Governors University

**Table of Contents**

[**A1. LETTER OF TRANSMITTAL**](#_12yi81rktg6l) **3**

[**A2. PROJECT RECOMMENDATION**](#_yz01r5heobrc) **4**

[**Problem Summary**](#_wqzyx4hwr9ol) **4**

[**Application Benefits**](#_ooalcpfzrz32) **4**

[**Application Description**](#_rhja3cb2hwbs) **4**

[**Data Description**](#_3m30lzclwuog) **5**

[**Objective and Hypotheses**](#_rftr2xperqjn) **5**

[**Methodology**](#_ii6jajh6842p) **5**

[**Funding Requirements**](#_912fdvjc60ei) **6**

[**Stakeholders Impact**](#_1kojdpce3sff) **6**

[**Data Precautions**](#_mnuur12cl1ge) **6**

[**Developer’s Expertise**](#_z5w0ia1j2vzd) **7**

[**B. EXECUTIVE SUMMARY**](#_p2wkztwicgfj) **7**

[**Problem Statement**](#_lgfzxsj2j586) **7**

[**Customer Summary**](#_kp1rveqhzgt3) **8**

[**Existing System Analysis**](#_4uxkut1obxzj) **8**

[**Data**](#_2jzqhtwqw8ul) **8**

[**Project Methodology**](#_lrvphawttnxc) **9**

[**Project Outcomes**](#_rwrwklhqr8ff) **9**

[**Implementation Plan**](#_otvep3rl8lv4) **10**

[**Evaluation Plan**](#_cbg5iohwyan6) **10**

[**Resources and Costs**](#_6e01t74xgxq1) **11**

[**Timeline and Milestones**](#_ozkig5wjupbj) **11**

[**D. POST-IMPLEMENTATION REPORT**](#_gua6t7ipyywh) **12**

[**Project Purpose**](#_d6i50dddslla) **12**

[**Datasets**](#_2m60vls2ohbn) **12**

[**Data Product Code**](#_7bh7ow94qoa5) **14**

[**Hypothesis verification**](#_1ow0ited38n4) **15**

[**Effective Visualizations and Reporting**](#_wb0l0frjeh0) **15**

[**Accuracy analysis**](#_wjgo2je05m1d) **15**

[**Application Testing**](#_wds4ajvcb8xe) **16**

[**Application Files**](#_rk91no3a69nz) **16**

[**User’s Guide**](#_h08p6f3yu8sz) **17**

[**Summation of Learning Experience**](#_kqmka7s4all8) **17**

## A1. LETTER OF TRANSMITTAL

January 21st, 2021

Larry Fisherman, Project Manager

Various Vaccines

2005 Beechwood Blvd,

Pittsburgh, PA 15217

Dear Mr. Fisherman,

COVID-19 has been devastating the world since the beginning of 2020. With a potential vaccine rapidly approaching the market, a distribution solution needs to be developed. In the early stages, vaccines will be scarce. Various Vaccines need to ensure that doses are supplied to each state based on demand. Each state has a varying number of COVID-19 cases and deaths. Vaccine distribution takes weeks, so the company needs to know how many vaccines will be needed weeks in advance. Daily data will not be useful in determining a state’s demand two weeks out. Each state's data needs to be analyzed to forecast cases and deaths, thus ensuring every vaccine gets utilized.

MTK Designs is proposing a web application that analyzes all of a state’s COVID-19 data and predicts cases/deaths on future dates. This will benefit Various Vaccines by assisting in the distribution of their vaccines across the United States. We can determine how long doses will take to ship from Various Vaccines to each state. Then we can enter shipping times into our application to see home many cases/deaths there will be when the vaccine arrives. This will help keep up with the demand for doses. If an application is not put into place early on, then more people will continue to get ill and die from COVID-19. The object of this application is to forecast the number of cases and dates the state of Tennessee will have up to a year in advance.

Our team at MKT Designs has determined that it will take roughly 6 weeks to complete and launch the application. The developer assigned this project charges $100 an hour and works 40 hours a week. Which makes the total cost to develop the application $24,000. Routine maintenance on the application is paid for yearly and costs $20,000 a year. This includes 4 hours every week for changes and upkeep. The budget is subject to change as the project develops.

Our team is very invested into this project because of the number of people it could help.

We have the knowledge to complete this project in a timely manner. Our main developer has been studying computer science competencies for years and has the capability to create the best application for Various Vaccines. I believe that if we work together, we could change the course of this pandemic. By knowing where to send vaccines we can save thousands of lives. Our proposed application will prove to be beneficial for both Various Vaccines and the American people.

Thank you for your time and consideration.

CEO, MKT Designs

## A2. PROJECT RECOMMENDATION

### Problem Summary

Various Vaccines is on the verge of developing a COVID-19 vaccine. Once completed, the vaccine will need to be distributed. They have selected Tennessee as the first state to receive the vaccine. It will take 2-3 weeks to deliver the vaccine across Tennessee. If Various Vaccines uses current data to ship doses, there won’t be enough vaccines upon delivery. If there are 200,000 cases in Tennessee when the vaccine is shipped, by the time they would arrive 2 - 3 weeks later there could be 300,000 cases and not enough doses. This will cause Various Vaccines to constantly be behind on delivering vaccines. This delay will cause many people to get sick and die. In order to efficiently distribute the vaccine, an application is needed to predict COVID-19 cases and deaths weeks in advance.

The web application will be protected with a login feature to enhance security. Once logged in the user will be brought to the main application. From the application a date can, be selected and once submitted the application forecasts the number of cases and deaths on that particular day. The other two pages of the application will consist of charts and diagrams to help explain the data. One page will contain results from the tests used to determine accuracy and train the model. Such an application will ensure that Various Vaccines will deliver doses of their vaccine to people in need across Tennessee and then across the entire United States.

The application uses Tennessee’s COVID-19 data from 01/22/2020 -02/08/2021. Data can be updated in the future to create a more accurate prediction. All updates are automatic once linked to the URL of the new data. These updates are not included in the maintenance of the application.

### Application Benefits

The application will streamline the distribution process for Various Vaccines. The application provides Various Vaccines with the ability to accurately distribute their vaccine. By allowing the company to predict how many cases and deaths there will be once vaccines are delivered, they can prevent a shortage of vaccines. A hub will be created for the company to analyze COVID-19 data and trends. Numerous tests will be provided so the company can ensure the data is accurate. Having such an application allows Various Vaccines to focus on the testing and manufacturing of the vaccine.

### Application Description

The application is built upon Python. Python has many libraries that specify machine learning. Jupyter Notebooks will be used as our Python environment. Notebooks assist in managing and installing packages. They are also very useful for sharing code across multiple workstations. The main libraries that will be used in the notebook to analyze and train data are NumPy, Pandas, Seaborn, and Matplotlib. Once the functions are created and tested, a web app will be developed. All of the code within the Jupyter Notebook will be ported to Google Colab. Colab is very similar to Jupyter Notebook except Colab can run Python in the browser. This feature is necessary for hosting the application online. The web app will be created using Anvil, a dynamic web development tool. Anvil is used for front end web development, using JavaScript, HTML, and CSS. Anvil is Python based so our code can be reused on the website. Anvil allows for functions to be called from the Colab notebook. The functions created to Forecast COVID-19 cases and deaths, can be sent to the web application. Putting everything together creates a fully functional Time Series Forecasting web application.

### Data Description

The data that will be used will be read from a CSV file. The file will be retrieved from HeathData.gov. The datasheet is named “United States COVID-19 Cases and Deaths by State over Time''. The datasheet will then be modified to remove every state except Tennessee. The data columns date, cases, and deaths will remain and all other columns will be removed. After modification, the file will be saved as a CSV and imported into Python via the readcsv function. The date column is of the Date type. The cases and deaths columns will be Integers. Time will be the independent variable and be represented by the date column. The dependent variable will be the cases and deaths column. The only limitation of the dataset will be the amount of data recorded. COVID-19 is still a fairly new disease. This limits the amount of data collected to the dates 01/22/2020 - 02/08/2021.

### Objective and Hypotheses

The objective of this project is to Forecast the number of COVID-19 cases and deaths on any given date. The current hypothesis states that if previous COVID-19 data can be analyzed and trained then an application can be created to predict future cases and deaths on that trained data.

### Methodology

MKT Designs will use Agile Software Development methodology for this project. We believe that using Agile Methods quickly leads to quality projects. The many cycles of Agile allow us to work on individual features of the project, one at a time. The cycle will start with defining requirements and planning. Once defined, we will be able to start building each feature. Then after the feature is built, we begin testing the code. The cycle will end when the code is ready to be released. These cycles continue throughout the stages of the project. With every cycle, new features will be added, tested, then deployed. The benefit of doing many short iterations, is the ability to be adaptable. We are able to make changes as needed with every cycle. Constant testing also ensures that when the final product is released it will be fully functional. Our first cycle or sprint will be gathering and analyzing the data. The next sprint will be used to train our model on the dataset. Next, we will create an interactive website for our application. The final sprint will combine the trained data and the website. The Agile methodology is known for being a quick and adaptable approach to software development.

### Funding Requirements

The project will be built upon free software. The free software includes environments, such as Python, Jupyter Notebook, and Google Colab. The web development tool Anvil allows for free development and hosting. As stated in the letter of transmittal the cost of development will be as follows, a minimum of 6 weeks to complete and launch the application at $100 an hour, 40 hours a week. The total cost to develop the application is roughly $24,000. Routine maintenance on the application is paid for yearly and costs $20,000 a year. This includes 4 hours every week for changes and upkeep. The budget is still subject to change as the project develops.

### Stakeholders Impact

The stakeholders for Various Vaccines include both the stockholders and the members of the community. This project will assist Various Vaccines in the delivery of their vaccines. By using our application numerous problems will be avoided. With each problem that the company faces, the stockholders will be negatively affected. If deadlines are not met and vaccines are not distributed effectively, then the company will receive negative press across the country. Negative press leads to lower stock prices and unhappy stockholders. More importantly is the community and how they will be affected without our application. The citizens of the United States are counting on Various Vaccines to supply them with vaccines. Negative impact on distribution will affect people's health and, in some cases, even lead to death.

### Data Precautions

The dataset that will be used is neither sensitive nor protected. The data is retrieved from HealthData.gov and is stated that their creative works such as writing or images are copyright-free. The data will be modified to only include the necessary information. All other information will be saved in a backup file. Using only the needed information allows for smaller storage files and quicker searching. Precautions commonly used with sensitive data will not apply to our dataset. The Health Insurance Portability and Accountability Act (HIPAA) ensures that no sensitive data will be disclosed without patient’s consent. Our dataset does not use any names or personal information. Similarly, the Family Educational Rights and Privacy Act (FERPA) protects sensitive information within student education records to be released without consent. Our data will not use any cardholder data and will not violate the Payment Card Industry Data Security Standard (PCI DSS).

### Developer’s Expertise

The developer tasked with this project is a senior level student. The developer has experience in many of the sought-after technologies. Such languages as Python, Java, JavaScript, HTML, and CSS are known. Knowledge of database tools including SQL, MySQL, and Excel. The developer has completed 8 projects of similar scale. Each of those projects have been complete before or by the deadline. The developer is self-taught. Thus, allowing for the ability to learn on the job. Before this project the developer has spent over 100 hours learning new apis that will assist in developing this app. The developer is fully capable of producing a quality product that will adhere to the budget and deadline.

## B. EXECUTIVE SUMMARY

### Problem Statement

The problem is creating an accurate Time Series Forecasting application that will predict Tennessee’s COVID-19 cases and deaths. None of the commonly used machine learning algorithms will work for this application. Machine learning is used on data that is either numerical or categorical, and not on a time series. Because of this, the number of tools at our disposal are limited. Other problems that we will face include finding a way to host a notebook-based application to a web server. Most websites are not meant to utilize Python. Python is needed though because of its forecasting capabilities. The libraries such as Pandas, NumPy, and Matplotlib are also necessary to analyze and visualize the data.

Our company will have limited resources on this project. With only one developer working on the project, it could be a daunting task. The dataset only has about a year of data, roughly 365 entries. With more entries the model could be more accurate. We will have to determine if the given dataset will be sufficient to create a reliable model. These problems will need to be addressed and monitored to ensure a quality product is produced.

### Customer Summary

The main users of this application will be employees of Various Vaccines tasked for distribution. The user will be able to log on to the application with verified credentials. The admins or developers will be in control of given individuals’ specific permissions. The average user will only be able to access the website's environment. The front end is designed to be very simple and easy to navigate. The application will only require the user to select a date and click submit. Visualizations will be pre built upon the data and are not user created. The back end will only be accessible to the software engineers. Once the application is completed Various Vaccines can determine if the application is private or public.

### Existing System Analysis

We would start building the application in Jupyter Notebooks. To run Jupyter Notebooks on a desktop, Conda will have to be installed. Conda is an application that organizes libraries and packages. Python is included with the installation of Conda. In the Conda terminal we can install Jupyter Notebooks, NumPy, Pandas, and Matplotlib. From there we would export all of our code to Google Collaboration. This environment does not require anything from the user to run. The finished application will be a web application. This website will be capable of running on any browser on a computer, phone, or tablet.

### Data

Data will be collected from [HeathData.gov](https://healthdata.gov/dataset/united-states-covid-19-cases-and-deaths-state-over-time). The name of the database is *United States COVID-19 Cases and Deaths by State over Time.* The link will lead directly to the page that hosts the database. The database is updated every day with the statistics from the previous day. The data can be downloaded as a csv, rdf, json, or xsl file. For this project we would import the data into our app via the .csv file. The data will be modified to remove all unnecessary data and will only leave the date, cases, and deaths columns. We will also remove all other states besides Tennessee. This data is backed upon accuracy. Any anomalies are fixed upon discovery. The application will be able to forecast up to a year out with the data including. Anytime the dataset can be updated to increase accuracy and extend the forecasting range. The application will be able to forecast further that a year out but with only one years’ worth of data, the accuracy will diminish over time.

### Project Methodology

Throughout the development of this application, we will use the Agile methodology. This methodology is known for being simple and efficient. There are numerous cycles within the development of an application using the agile methodology. Each cycle follows the same 6 stages. The first stage of the agile cycle is requirements gathering and analysis. Many of the requirements are listed in this documentation. Any requirements not listed can be discussed with the stakeholders of Various Vaccines. After the needs of the company are identified, we will move to the next stage, planning. A plan needs to be developed to execute the objectives of each cycle. Here teams are formed and the tasks required to complete are discussed. Next the teams begin to design the software-based requirements. Implementation is next, here the cycle is added to the product of the previous cycle to test compatibility. In the next stage teams of testers create tests and troubleshoot the code. We will do unit testing on applications main function, which is the forecasting function. Integrating testing will ensure that the collaboration notebook will interact properly with the web application. Once everything is completed, system testing will be conducted to test the entire application as a whole. Various Vaccines will determine if the application meets their business needs with an acceptance test. After the majority of the test cases pass, the final product is ready for deployment.

### Project Outcomes

Project Deliverables will include a schedule that defines the start and end dates of each development cycle, a breakdown of the budget, and test plans. Once the project’s deliverables are agreed upon, a schedule can be created based on the deliverables. Each of the high-level deliverables will be placed into individual cycles. We will be able to determine an estimated time of completion for each of the cycles. The budget has been stated above. It should be noted that the budget is based upon the deliverables mentioned above and is subject to change. Testing has also been talked about in the above paragraph. We will be able to go into more detail when this project scope is finalized.

Project deliverables will include a modified database, notebooks, and a web-based GUI. The database that the application will be running, will be included in application’s documentation. This database is modified to only include the necessary information. We will deliver both a Jupyter Notebook and a Google Collaboration Notebook. Both will function the same with minor tweaks to the code. Having two notebooks will ensure easy access of data to your teams. The web application will be tied directly to the Google Collaboration Notebook. The final deliverable is a fully functional web-based application. This web application will include a forecasting function, data visualizations, and a login feature.

### Implementation Plan

The project will be implemented into notebooks for the developers and a web application. The notebooks allow for implementation across multiple workstations. The web environment will allow for all users to access the application. All that a user will need is an up-to-date web browser. They will need to sign up with Various Vaccines in order to access the site. That feature is at the discretion of the company, they can disable the login page and allow for unrestricted access to the site. Testing will occur in each cycle of development. This will ensure that once the project is complete, each individual feature has been tested thoroughly. Milestone can be set after the terms of the project is agreed upon. The deliverables are mentioned in the above paragraph. User testing can be done with Various Vaccines’ approval. User testing is not included in the current budget.

### Evaluation Plan

The developed application will verify its success based on the accuracy of the forecasting model and upon meeting the requirements of the company. The application will need to be accurate in order to be useful. There are a number of tests we can use to verify the accuracy of the model. We will use decomposition to break down the time series into trend, seasonality, and noise. These will help analyze the data before we make a prediction. There are four main tests that we will use to judge the accuracy of the prediction model. Most of the tests use residual error which is the difference between the predicted value and the observed value. Standardized residual error which is calculated as the expected outcome minus the forecast. Estimated density will show the mean value of the residual errors. Normal Q-Q will be used to check the normality of the distribution of residual errors. Correlation will calculate the strength of the relationship between an observation and observations at prior time steps. We will validate the needs of the client based on the agreed requirements. The project will come to closure after all testing has been completed and the stakeholders accept the application.

Software quality assurance will be used throughout the development process. By using the Agile methodology, we will constantly be testing the software throughout the lifecycle of the project. After each cycle our SQA team will check the quality of the activities. Towards the end of the project, we will execute a formal technical review to evaluate the quality of the web application.

This is a contract-based project and while under contract MKT Designs will adhere to the organizational policies of Various Vaccines.

### Resources and Costs

The programming environments for this project are free to use and easy to install. The entirety of the project will be written in Python. We will use Jupyter Notebook, Google Collaboration Notebook, and Anvil. Jupyter Notebook can be installed with the Anaconda application. This application is free to use as well and can be installed on Windows or Mac. Google Collaboration requires no installation and is used within the browser. Python is built into this notebook. Anvil is a free to use website that assists in the development of a Python web application.

The environment costs $0 as long as Various Vaccines have their own workstations. Our team is already equipped with the tools required to develop this application. We will work in our office so no additional real estate is required.

The human resource requirements are estimated as the number of development hours at a fixed rate. Our team has determined that it will take roughly 6 weeks to complete this project. The developer will be paid $100 an hour and will work 40 hours a week. The total cost of the project is estimated to be $24,000. Routine maintenance on the application is paid for yearly and costs $20,000 a year. This includes 4 hours every week for changes and upkeep. The budget is subject to change. This is only the estimated cost to complete the application.

### Timeline and Milestones

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Milestone | Start Date | End Date | Duration | Dependencies | Resources |
| Gather Requirements | 02/08/2021 | 02/12/2021 | 5 Days | Stakeholder’s Approval | Stakeholders & Developer |
| Create Forecasting Function | 02/15/2021 | 03/05/2021 | 15 Days | Gathered Requirements | Dataset & Developer |
| Train Model | 03/08/2021 | 03/12/2021 | 5 Days | Created Forecasting Function | Python & Developer |
| Testing | 03/15/2021 | 03/26/2021 | 10 Days | Trained Model | Test Cases & Developer |
| Develop Website | 03/29/2021 | 04/09/2021 | 10 Days | Testing Completed | Anvil & Developer |
| Project Finish | 02/08/2021 | 04/09/2021 | 45 Days | Website Developed | Stakeholders & Developer |

## D. POST-IMPLEMENTATION REPORT

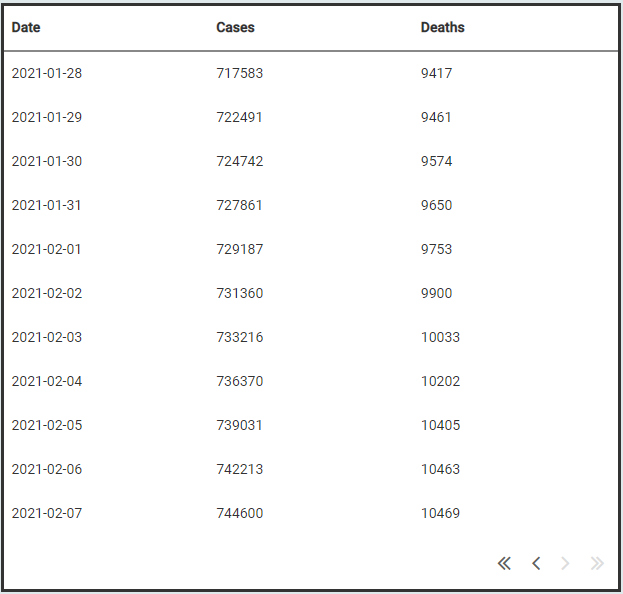
### Project Purpose

The purpose of this project was to create an application for Various Vaccines that would forecast COVID-19 cases and deaths on future dates. The application needed to assist in the distribution of the vaccine in Tennessee. We gathered information on current COVID-19 cases in Tennessee and created a forecasting model based on the collected data. The model was then trained to forecast COVID-19 cases and deaths. The forecasting function was uploaded to the web application and a website was formed. After testing was completed, we released the application to Various Vaccines. We monitored their usage for a week and optimized the performance based on their needs. The application has proven to solve the problem that Various Vaccines was facing. The application allows for their team to predict how many vaccines will be needed by the time the order has arrived in Tennessee. By using the application, Various Vaccines has been able to ensure that vaccines are distributed to those in need and no dose is wasted.

### Datasets

Our application used the dataset *United States COVID-19 Cases and Deaths by State over Time* from [HeathData.gov](https://healthdata.gov/dataset/united-states-covid-19-cases-and-deaths-state-over-time/resource/5157f946-8219-4e91-8e68#%7Bview-graph:%7BgraphOptions:%7Bhooks:%7BprocessOffset:%7B%7D,bindEvents:%7B%7D%7D%7D%7D,graphOptions:%7Bhooks:%7BprocessOffset:%7B%7D,bindEvents:%7B%7D%7D%7D%7D). The dataset was a csv file containing COVID-19 across the United States. The raw dataset can be downloaded by following this [link](https://data.cdc.gov/api/views/9mfq-cb36/rows.csv?accessType=DOWNLOAD).

The edited dataset only contains the columns date, number of cases, and number of deaths.



In order to modify the dataset, we first had to import it into our application. We saved the download link to the variable csv\_url. Then we created an array containing the names of the fields we needed. Both of these variables were passed through the read\_csv function and saved as a panda’s data frame.

|  |
| --- |
| csv\_url = "https://data.cdc.gov/api/views/9mfq-cb36/rows.csv?accessType=DOWNLOAD"  fields = ['submission\_date', 'state', 'tot\_cases', 'tot\_death']  csv\_full\_dataset = pd.read\_csv(csv\_url, usecols=fields) |

By searching for all rows where state equals TN, we are able to get rid of all other states. Then we drop the column containing TN to save on memory.

|  |
| --- |
| csv\_full\_dataset = csv\_full\_dataset[csv\_full\_dataset['state'] =='TN']  csv\_full\_dataset = csv\_full\_dataset.drop(csv\_full\_dataset.columns[1], axis=1) |

We changed the data type of the ‘submission\_date’ column to datetime. Then we are able to sort the database by date.

|  |
| --- |
| csv\_full\_dataset['submission\_date'] = pd.to\_datetime(csv\_full\_dataset.submission\_date)  csv\_full\_dataset = csv\_full\_dataset.sort\_values(by=['submission\_date']) |

Finally we can split the data into two trainable datasets. The first dataset named ‘dataset’ consists of the 'submission\_date' column and the ‘tot\_cases’ column. The second dataset named ‘death\_dataset’ contains the ‘submission\_date’ column and the ‘tot\_death’ column.

|  |
| --- |
| dataset = csv\_full\_dataset.drop(csv\_full\_dataset.columns[2], axis=1) death\_dataset = csv\_full\_dataset.drop(csv\_full\_dataset.columns[1], axis=1) |

### Data Product Code

The functionality of the data product was based on the ability to accurately predict COVID-19 cases and deaths. We had to create two different models, one for forecasting cases and one for forecasting deaths. Then each model was split into two groups, a training set and a test set. These sets were chosen at random to allow for an unbiased outcome. The training set is considered to be the descriptive data. Descriptive data is decomposed to further analyze the data. Once decomposed we study any trends, seasonality, and noise. Since there were trends in this dataset we use the function SARIMAX() to train the data. SARIMAX() stands for Seasonal AutoRegressive Integrated Moving Average with eXogenous regressors model. SARIMAX() uses Moving Average (MA) and Residual Errors to forecast the data. The data is then fitted onto the model. We then call the forecast function to provide us with predicted outputs. The next 400 days are predicted and saved into a data frame. The application uses a function that searches the data frame for the user selected date and retrieves the number of cases and deaths on that day. The entire source code can be found in this submission.

### Hypothesis verification

Our hypothesis stated that if previous COVID-19 data can be analyzed and trained then an application can be created to predict future cases and deaths on that trained data. With the completion of the project, we can determine that the predictions have sufficient accuracy. By comparing predictions to actual data Various Vaccines has also accepted the hypothesis. Over time the accuracy of the application will continue to increase. With the release of more data and the delivering of vaccines the data will begin to smooth out.

### Effective Visualizations and Reporting

The application contains three interactive visualizations and six data summary visualizations. The first two visualizations are bar graphs showing the number of cases and the number of deaths over time. Both bar graphs depict an upward trend of cases and deaths. These graphs have the ability to zoom in/out, compare data on hover, and save plot as png. In data preparation both graphs were analyzed to find trends. Following the bar graphs is a data table. This table contains all the information that the forecasting model is based upon. It contains a date column, a cases column, and a deaths column.

Prediction data is shown in two decomposition charts. The charts show trends, seasonality, and noise for both cases and deaths. Diagnostics tests are used to visualize the accuracy of the prediction model. The charts display the Standardized Residual Error, Histogram plus Estimated Density, Normal Q-Q, and Correlogram. Two graphs show the past data vs predicted data. The upper and lower bounds of the prediction data show how accurate the predictions are.

### Accuracy analysis

We used Standardized Residual Error, Histogram plus Estimated Density, Normal Q-Q, and Correlogram to determine the accuracy of our model. Because the data predicts future values it is impossible to put a percentage of accuracy to the model. Instead, we ran numerous tests to analyze the accuracy. The first test was the Standardized Residual Error, which is calculated as the expected outcome minus the forecast. This showed us the difference between the set of observed and predicted values. For this test we want the difference between the values to be zero. We see that for a six-month prediction the average is around zero but quickly worsens as time goes on. The next test used was a Histogram plus Estimated Density. Within the histogram kernel density estimation (KDE) is graphed, which shows the probability density of a random variable. This is compared to mean and standard deviation. In this test, accuracy is determined at values close to zero. Are data remains close to zero with almost all histogram values between negative one and one. Normal Q-Q (quantile-quantile) plot, showed the distribution of the data against the expected normal distribution. This test needs to show a linear line and the more curved the data the less accurate the model is. Our test shows a straight line throughout with curvature only on the ends. The last test used to analyze accuracy was a correlation test. The correlation measures the strength of the relationship between two variables on a scatter plot. The values are plotted between one and negative one. The values close to negative one shows a perfect negative linear relationship and the values close to positive one shows a perfect positive linear relationship. Our data averaged in the middle at zero which states no linear correlation. These four tests are proof of our application accuracy in forecasting COVID-19 cases and deaths.

### Application Testing

We used all four levels of application testing in this project. The first test conducted were unit tests on the forecasting function. Here we tested the ability of the function to take a date and output the number of cases and deaths. Once passed we move to integration testing. We tested the same function after we integrated it with the web application. Even though the function remained the same we had to ensure that it communicated with the web application correctly. During integration testing we discovered a bud that was caused by a data type mismatch. The web application was saving the date as a datetime type and the function required a date type. We were able to normalize the two data types and complete the integration testing. After the entire project was completed, we started system testing. We tested the login functionality first to make sure we could get into the web application. Then we tested the data visualization to ensure the data was being imported correctly. We had an issue with the data showing double the values but was fixed. We tested the website's usability and performance. The final level of testing was acceptance testing and was done by Various Vaccines. They showed the stakeholders the finished project and determined that the system was ready for release.

### Application Files

All of the files for this project can be found by following this GitHub repository [link](https://github.com/MattThomas95/Computer_Science_Capstone.git).

The repository includes the following files:

* README.md: Directions and information.
* United\_States\_COVID-19\_Cases\_and\_Deaths\_by\_State\_over\_Time.csv: Comma-Separated Values file containing all the COVID-19 data used in the application.
* Machine\_learning\_model.ipynb: Jupyter Notebook file containing the initial Python code.
* Capstone.ipynb: Google Colab file containing the final working application.
* Computer Science Capstone.yaml: Anvil saved data of web application.

### User’s Guide

An up-to-date browser and internet connection is needed to run this application.

1. Navigate to the project’s GitHub repository located [here](https://github.com/MattThomas95/Computer_Science_Capstone).
2. Click on the file named Capstone.ipynb
3. Select the Open in Colab button at the top of the code.
4. Must be signed into google account in order to run code.
5. Select Connect in the upper right corner of the page.
6. Under the Runtime tab in the top left of the page select Run All.
7. After all of the cells have executed you can follow [this](https://br2uqiaya7wd4p4i.anvil.app/MIS7MPLIOHOTJBI22QRFAO5N/) link to the website.
8. If the Google Colab session disconnects, the website will not function properly.
9. Follow steps 5 and 6 to reconnect the runtime.

The username for the web application is [app\_admin@wgu.edu](mailto:app_admin@wgu.edu) and the password is admin. Once logged on you will be able to use the application. From the date picker you can select any date within the next year. After a date is selected you can hit the submit button to see the results. The results show both the predicted number of cases and deaths on the selected date. Selecting Data Visualization in the navigation bar will take you to interactive data charts. Selecting Prediction Accuracy will take you to the testing data recorded to determine accuracy.

### Summation of Learning Experience

I came into this project with only a little experience in Python. I had no experience in machine learning. After I had an idea of what needed to be done, I set out to learn the core machine learning principles. I enrolled in the Udemy course, “Complete Machine Learning & Data Science Bootcamp 2021”. In this course I learned the fundamentals of Jupyter Notebook, Pandas, NumPy, Matplotlib and numerous other data science tools. About half way through the class I realized that I was only learning about Regression, which was not what my project would be based upon. I needed to learn Time Series Forecasting in order to predict values based on date data. Fortunately for me Time Series Forecasting used many of the same libraries as Regression. I was able to learn everything I needed to know to develop the Times Series Model. My next learning experience came in the form of a web application. I have created static web pages before but never in Python. I saw many examples of students using Binder or Voila to host their Jupyter Notebook but I was not satisfied with those methods. With a little research I came across Anvil. Anvil is a platform that allows for full stack web development with Python. It also has a feature that allows external Notebooks to communicate with the web application. This feature was exactly what I needed. After I read up on Anvil’s documentation, I was able to create a working prototype. The overhead of Jupyter Notebooks was becoming too much so I exported all of the code over to Google Collaboration. Google Colab has all of the same features but was hosted online and much easier to distribute the finished project. From there I was able to optimize the web application and finish up the interface. The final step was to upload everything to GitHub.

This project has reinforced the importance of self-learning for me. I came into this project knowing only a few of the necessary skills and now I have a functional data product. I now know that the information is out there and where to find it. I believe knowing where to find information is the most important tool in the concept of life-long learning. No matter the task, anything can be learned given the right time and resources. Knowledge is power and I will never stop learning.