Analysis, visualization, and prediction of Covid-19 cases in the USA using Data Mining

Manik Mahashabde

[10518579]

Dissertation submitted in partial fulfillment of the requirements for the degree of

[***Masters of Science in Information Systems with Computing***]

at

Supervisor: Hamidreza Khaleghzadeh

August 2020

Declaration

I Manik Mahashabde declare that this dissertation that I have submitted to Dublin Business School for the award of MSc. in Information Systems with Computing is the result of my investigations, except where otherwise stated, where it is acknowledged by references. Furthermore, this work has not been submitted for any other degree.

Signed: Manik Mahashabde

Student Number: 10518579

Date: 25-08-2020

Acknowledgment

I want to express my special thanks of gratitude to Dr. Hamidreza Khaleghzadeh, my teacher and research supervisor for his patient guidance from the beginning with constructive and valuable suggestions provided throughout the planning and improvement of my dissertation.

Secondly, I would like to thank my family and friends for their support and encouragement throughout my course.

Abstract

This study performed Exploratory Data Analysis(**EDA**), Tableau Visualization, and predictions for COVID-19 on a dataset of the USA. The dataset has attributes such as positive cases, recovered cases, death cases, states, ICU cases, etc. In the first part, EDA was used on the dataset for a better understanding of the existing data and identifying relationships between different attributes using python libraries such as Seaborn, Pandas, Numpy, etc. In the second part, Tableau was used for creating a COVID-19 dashboard on the same data. For predictions, different algorithms such as ARIMA, Prophet, Polynomial Regression were implemented. The dataset was split with 95% for model training and 5% for the testing. Finally, evaluation parameters such as MAE(Mean Absolute Error), MSE(Mean Squared Error), and RMSE(Root Mean Squared Error) were calculated for confirmed, recovered, and death cases. ARIMA gave the best results among the three models and it was later used for predicting cases from 25th August 2020 – 7th September 2020.

Table of Contents

List of Tables:

List of Figures:

CHAPTER 1: INTRODUCTION

1.1 About COVID 19

In December 2019 many people visited the local hospital in Wuhan China as they were having symptoms of flu such as a cold or a fever, or a cough, weakness, or difficulty in breathing or some patients had a combination of two or more symptoms. Initially, these all seemed to be symptoms of seasonal flu. But as the vaccines, medicines were ineffective in treating the patients. The investigation done by China's Center for Disease Control and Prevention found out that these people visited the local wet market in Wuhan. Upon research, a new kind of virus was discovered having some similarities related to the 2003 SARS-CoV coronavirus. This new virus was termed as Novel Coronavirus and the disease was named SARS-CoV2 or Covid19. However, it was found that the Novel coronavirus is much more contagious than SARS-CoV. As a result, the Chinese government took strict measures in stopping the spread of the virus by imposing complete lockdown and travel restrictions. But the virus was spreading initially to South Korea, Japan, and slowly to the rest of the world. On 30th January World Health Organization(WHO) declared COVID-19 as the public health emergency of international concerns and later on 11th March it was declared as a Pandemic. Since then the whole world has been following the measures of lockdown and containment of the diseases. Policies like social distancing and work from home have been adopted by the people. The top 3 countries affected by the Covid-19 as of today are the USA, Brazil, and Russia.

The first case in the USA was identified on 20th January while the first death happened on 29th February. Until 12th August 2020, the USA has the highest number of cases approximately 5.17 million with 157776 deaths. This research will help in determining the condition of COVID 19 in

the USA in the future. It will help in finding out the total number of COVID 19 cases in the future, the total number of deaths, the total number of recovery, active cases, etc. This research will also help in determining an optimum roadmap for the government. The government of the USA can use this research to check the prediction of COVID 19 in the future and then steps can be taken accordingly to reduce the cases.

1.2 Scope

- Finding an appropriate dataset of Coronavirus cases for the USA.
- Adding the dataset in Jupyter notebook.
- Using libraries like Numpy, Pandas for data cleaning.
- Determining correlations between different fields.
- Using Matplotlb, Seaborn for understanding data distribution and data analysis.
- Performing Encoding if required.
- Using Tableau for data visualization.
- Fitting the model for machine learning algorithms such as Linear Regression, Polynomial Regression, Prophet, ARIMA.
- Comparative analysis of machine learning algorithm results(calculating MAE, MSE, and RMSE)and determining the best machine learning algorithm.
- Predicting positive cases, recovered cases, and death cases using the most efficient algorithm.

1.3 Dissertation Roadmap

Following is the roadmap used for the implementation of the dissertation project as shown in figure 1.1

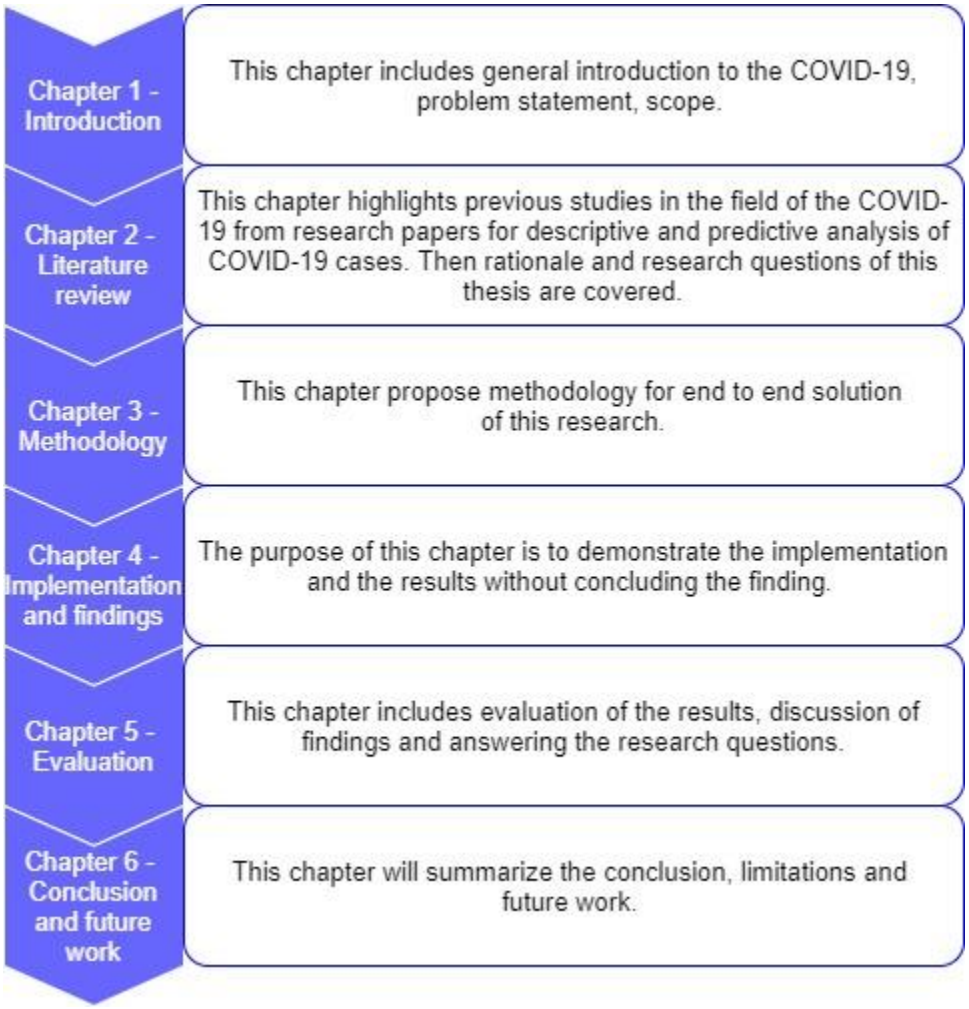

Figure 1.1: Dissertation Roadmap

CHAPTER 2: LITERATURE REVIEW

2.1 Previous Research Analysis

There have been various researches of Covid-19 up to date. They are explained below.

In (Dey, Rahman, Siddiqi, and Howlader, 2020) the outbreak information of Covid-19 from 22nd January 2020 until 15th February 2020 was analyzed for better visualization of the disease. Three open datasets provided by the Johns Hopkins University, World Health Organization, Chinese Center for Disease Control and Prevention, National Health Commission, and DXY were used in this study. The first dataset tracked the spread of Covid-19, the second datasets consist of the number of confirmed cases, recovered cases, and death cases. Finally, the third dataset consists of daily level cases of Covid-19. Visual exploratory data analysis techniques were used here using python libraries such as Pandas, Numpy, Plotly, Seaborn, Matplotlib, and Folium for different purposes such as data cleaning, data transformation, and finally data visualizations. For data visualization map view and treemap, view techniques were used. However, this study had limitations in terms of accuracy of data analysis and visualization for real-time analysis as the dataset was limited of only two months.

([The epidemiological characteristics of an outbreak of 2019 novel coronavirus diseases (COVID-19) in China], 2020) performed a descriptive exploratory analysis of 71,314 COVID-19 cases in China from December 2019 until 11th February 2020. The dataset was taken from China's Infectious Disease Information System as it contains a national identification number of infected Chinese citizens hence there are no chances of data duplications. Some of the significant analysis are as follow: 1) summary of patients characteristics in which demographic and clinical characteristics were summarized using descriptive statistics for confirmed cases; 2) examination

of age distribution and sex ratio in which age distribution graph was plotted for confirmed case in Wuhan along with that Male/Female sex ratio was also calculated; 3) calculation of fatalities rate and mortality rates were performed in which case of fatality is calculated as the percentage of the number of deaths over the total number of cases whereas mortality rate is calculated by the percentage of the number of deaths over total observed time; 4) geo-temporal analysis of virus spread in which ArcGIS Desktop Software was used for plotting color-coded maps of china for three different timelines between December and February of the COVID-19 cases in each province; 5) epidemiological curve construction in which a graph was plotted between the number of cases vs the date of symptoms appears for confirmed, suspected, diagnosed, asymptomatic cases, and 6) subgroup analysis in which again epidemiological curve were plotted for confirmed cases diagnosed outside of Hubei province and all cases diagnosed among healthcare workers. The main finding of this study was to show how quickly coronavirus spread all over China from one city in just 30 days.

In (Ruan et al., 2020) the identification of clinical predictors for mild and severe Covid-19 cases was performed for 150 patients(68 death cases and 72 discharged cases). The data was taken from Yin-tan hospital and Tongji hospital. Various graphs have been plotted such as 1) the number of cases vs age group for death and discharged cases; (2)key laboratory parameters such as white blood cell counts, lymphocytes counts, blood urea nitrogen, etc. between two groups of death and discharged cases; (3) various causes of death with respiratory failure having the highest percentage. This study aimed to effectively prioritize the hospital resources for patients with the highest risk.

In (Pandey, Chaudhary, Gupta, and Pal, 2020) the predictive analysis for confirmed Covid-19 cases was performed using the SEIR (Susceptible, Exposed, Infected, Recovered) model and Regression models such as linear regression, polynomial regression. The dataset was taken from John Hopkins University, USA. The period of the dataset was from 30/01/2020 to 30/03/2020. In the SEIR model, certain assumptions were taken such as the number of births and death remain the same. Parameters for the latent period, infectious period, susceptible people, average incubation period were also assumed. Based on these parameters R0(reproduction number) value was calculated and confirmed cases were predicted. In the case of the regression model, the training set and the test set were split and the model was trained and confirmed cases were predicted. To check the performance of both the model's root mean square log error value was calculated.

However, overfitting remained a major problem in this study. Along with that, it was difficult to train the model as the data was limited. Hence, death case predictions were not performed.

In (Chatterjee and Hassanien, 2020) the dataset was taken from the Kaggle. The data was from 22nd Jan 2020 until 10th April 2020. The data mining tools such as WEKA and Orange were used here. The algorithms used in the predictive analysis were Linear Regression, Multilayer Perceptron, and Vector Autoregression. Confidence Interval of 95% was set up for plotting the graph between predicted confirmed, predicted death, and predicted recovered with actual confirmed, actual death, and actual recovered cases using all the three models. Then the table

consisting of prediction data was also made for the next 69 days from 10th April i.e until 18th June 2020. It was observed that the MLP model gave the best result among the other two.

In (Han Lau et al., 2020) a website was created termed as CoronaTracker that provides visualizations, live count, predictions, and other Covid-19 related information. Data was extracted here from sources such as John Hopkins University, WHO, DingXiangYuan, and a website authorized by the Chinese government. The database was created in MySQL and the website was hosted in AWS. Python micro-service was used to fetch the data for the frontend dashboard. For predictive modeling, the SEIR model was used here. Values for some parameters such as R0(reproduction number), incubation rate, recovery rate, the total world population, etc. were used. Then the virus transmission was described using differential equations. Apart from case prediction, sentiment analysis was also performed using new articles stored in the CoronaTracker database. The description considered was a minimum of eight words and also in the English language. The library used for this was transformers.

In predictive modeling, the data gathered from JHU didn't have exposed individuals. As a workaround, an assumption was used here that all the current infected patients might be in the exposed category 6 days back as the incubation period assumed was 5.2 days. The forecast of 240 days from 20th January 2020 was calculated here.

(Jia et al., 2020) used three different kinds of mathematical models which were the Logistic model, Bertalanffy model, and Gompertz model. The models were initially fitted for the 2003 SARS outbreak and were then later validated in Covid-19 for Wuhan and other parts of China. It was observed that the fitting effect of the Logistic model is better as compared to the

other two models in terms of predicting confirmed cases and deaths. Whereas, Bertalanffy model performed poorly among three. The prediction in this model was for confirmed cases and deaths. The recovered cases prediction was not performed here. This model predicted that Covid-19 will probably end at the end of April in Wuhan and by March-end in China.

(Ranjan, 2020) did a Covid-19 case prediction for India using an exponential model and SEIR model. Comparison of cases in India was also done with other countries particularly the U.S. The data was taken from the John Hopkins University, World Health Organization, and Center for Disease Control and Prevention(CDC). For the exponential model, the data between 11th-13th March was used and for the SIR model, the data for 21 days was used starting from 10th March. It was observed that the exponential model gave better results for short term prediction whereas, the SIR model gave better predictions for the long term. The reproduction number(R0) value was in the expected range of 1.4-3.9. However, these predictions were only valid for Stage I and Stage II transmission in India. These models are invalid for stage III i.e community transmission. Apart from this, it was also assumed that all the exposed cases were symptomatic and didn't consider asymptomatic cases.

(Gupta, K. Pal, and Pandey, 2020) addressed several research questions related to Covid-19 in India. The first research question was about the impact of lockdown on the Covid-19 case in which a dataset of the data consisting from 20th January 2020 till 7th April 2020 was taken from the Indian database of Covid-19. The dataset was divided into three parts i.e 30th January 2020 - 4th March 2020, 5th March 2020 – 22nd March 2020, and 23rd March 2020 – 7th April 2020. The visualization of this data clearly stated that the lockdown by the Indian government was

successfully to stop the exponential increase in the Coronavirus cases. The second research question was about the short term prediction of the Coronavirus cases. For this purpose, the exponential model was used to predict cases for the next 3 weeks in India. Based on the growth of the exponential model, the polynomial regression line was drawn for degrees in range 2-6. Root Mean error was calculated for all the 5 degrees and it was observed that degree 4 gave the lowest RME value 237.58. It was later used for predictions. The third research question was about. Whether people are following social distancing or not. Google's mobility report visualizations indicate that there was a 77% decline in Retail and Recreational places like the movie theater, a restaurant which shows that people followed lockdown seriously. In the fourth research question, the Decision Tree classification model was used to determine whether an infected Covid-19 patient will decease or not based on three features which are the age of the patient, gender of the patient, and state/region of the patient.

In (Sayeed and Ayesha, 2020) data visualization and predictions were performed to help people and governmental bodies for a better understanding of Covid-19 cases. In the data visualization, various kinds of plots such as Bar Plots, Horizontal Bar Plots, Scatter plot, Pie chart, Bubble chart, Dot plots, Box plot, Gantt chart, TreeMaps were used for confirmed, recovered, and death cases of Covid-19. For the prediction of Covid-19 cases, the dataset was taken from the Kaggle. After cleaning and data transformation, three models i.e ARIMA(Auto Regressive Integrated Moving Averages) model, Holt winter model, and Fbprophet models were used for prediction. The accuracy results of all the 3 models were good. However, ARIMA is more complicated as compared to the other two models.

(Bhatnagar, 2020) developed a new mathematical model using the concepts of geometric progression and inequality. The dataset was taken from the worldometer website. This study mainly focused on the countries there were in stage-III transmission such as Italy, USA. It predicted the number of the days India will take to come in the category of community transmission. Apart from this, it also considered the effects of lockdown in slowing down the transmission by plotting the graphs between actual cases, predicted/proposed constrained environment cases, and proposed unconstrained environment cases. However, it didn't cover the number of recovered cases and death cases.

(N Roy et al., 2020) used a dataset from Kaggle for analysis, visualization, and prediction of confirmed cases. The dataset has cases from 22nd January 2020 – 26th April 2020. The analysis and visualization were done for various countries like Germany, USA, Italy, Spain. The python library that was used for data visualization is Plotly. For the prediction Prophet prediction model developed by Facebook has been used here. It is also based on the Additive Regression model. The prophet model predicted the confirmed cases from 27th April – 10th May 2020. However, this study didn't predict recovered cases and death cases.

In (Kucharski et al., 2020, pp. 553 - 558) a new stochastic transmission dynamic model was used for determining the rate of transmission of the virus in Wuhan from January 2020 until February 2020. Then using this the probability of the new cases causing an outbreak in other areas was calculated. The transmission model was fitted to four public datasets. These datasets were related to cases in Wuhan and internationally exported cases from Wuhan. The four datasets that were used are 1)the daily number of new international exported cases by date of

onset until 26th January 2020; 2)the daily number of new cases in Wuhan with no market exposure by date of onset between 01st December 2019 till 01st January 2020; 3)the daily number of new cases in China by date of onset between 29th December 2019 till 23rd January 2020; 4) proportion of infected passengers on evacuation flights between 29th January 2020 and 4th February 2020. Along with this, two additional datasets were also used for comparison with model outputs 1) the daily number of newly exported cases from Wuhan in countries with high connectivity with Wuhan until the date 10th February 2020; 2)data on newly confirmed cases in Wuhan between 16th January 2020 and 11th February 2020.

Individuals here were divided into four categories i.e susceptible, exposed, infectious, removed. It should be noted that to infer the transmission rate over time Monte Carlo simulation was used here. The basic reproduction number R0 was calculated here. It was observed that R0 in Wuhan declined from 2.35(by 23rd January 2020) to 1.05 one week after. Also, the probability is 1/2, or 50% chances of an outbreak if 4 or 5 new cases were discovered in areas outside Wuhan. The model estimated that there was 95% of the population was susceptible by 31st January 2020. However, the model couldn't predict the slow down of cases observed in early February and predicted the cases 10 times higher than what was observed due to the model fitting problem. The result suggested that there were 10 times more symptomatic cases in Wuhan than were reported as confirmed cases. During the model fitting the pattern of confirmed exported cases from Wuhan was not included in the model fitting. However, they were reproduced by the model. Lastly, there were more exported cases to France, the USA, and Australia as compared to the value that the model predicted.

(Chen et al., 2020) analyzed and visualized the publicly available datasets to understand the characteristics of the epidemic and finding out patterns from it. It answered a few of the questions such as 1) how the COVID-19 spread from Wuhan to the rest of the country? 2)To what extent Quarantine and lockdown helped in slowing down the spread of the disease. 3) Can we forecast the future if certain conditions change using a mathematical model? HeatMap has been plotted here to show how the virus transmitted from Wuhan to the rest of the country. It happened due to the Chinese New Year(25th January) as people traveled from in and out of the Wuhan to celebrate the new year with their families. The classic SEIR model doesn't consider the infected COVID-19 people who were quarantined to reduce the transmission of the disease. Along with that people who come in contact with an infected person and fall into the suspected zone were also quarantined. Hence, a new C-SEIR model was developed here by adding two new categories of suspected infection group(P) and quarantined diagnosed infected group(Q). The graph was plotted here that shows the effect of an increase or decrease or completely stopping of quarantine on the overall cases using the C-SEIR model.

(Gupta and K Pal, 2020) used the dataset from the John Hopkins University from the time-period of 30th January 2020 till 24th March 2020 for India. The study performed the exploratory data analysis of the current dataset and also performed predictive analysis using the ARIMA(Auto-Regressive Integrated Moving Average) model and exponential smoothing method. Firstly trend analysis of India was performed and various graphs such as the number of infected persons(cumulative) cases per day, reported cumulative deaths per day, infected new cases per day, and reported new deaths per day were plotted. Then to understand the impact of COVID-19 in each state graph was plotted showing the number of cases on the y-axis and states on the

x-axis and the trend was shown in decreasing order. India was also compared with other nations under SAARC(The South Asian Association for Regional Cooperation) as part of the third type of analysis. A graph was also plotted here to understand the trend of rising cases between different countries.

Then the prediction was performed using the ARIMA model in which the graph was plotted in log transformation. Parameters such as the first order auto-regressive component and the second-order moving average component gave significant results for the ARIMA model. After this graph was plotted between log-transformed data and predicted data with the 95% confidence interval. It gave good prediction results of the ARIMA model and prediction was calculated for the next 30 days. After this Exponential Smoothing method was used for forecasting using Holt's linear trend, Exponential trend, and Additive damped trend which also gave good results. However, it should be noted that both the models i.e ARIMA and exponential didn't predict the number of recovered cases and death cases and only focused on the confirmed cases prediction.

(Batista, 2020) used the SIR model for the final prediction of COVID-19 cases in China. The prediction for COVID-19 cases in China was 85000 cases. The differential equation for Susceptible, Exposed, and Recovered compartments were derived here. After this, R0(Reproduction Number) was calculated which was greater than 0.98 for all the data. Then, a graph of the SIR model was plotted in comparison to the Logistic model and actual cases. The results demonstrated that the predictions of both models were very close. The SIR model

predicted that the total number of cases in China will be 84100 cases and the logistics model predicted the 84000 cases.

In (T Wu, Leung, and M Leung, 2020, pp. 689 - 697) some datasets were used for analysis and predictions those are 1) number of cases exported in Wuhan internationally between 1st December 2019 and 25th January 2020; 2) The data of the monthly flight booking from the official Aviation guide; 3) the data on human mobility across 300 cities in mainland China from the Tencent database; 4) the data of the confirmed cases published at CDC(Chinese Center for Disease Control and Prevention); 5) the domestic passenger information from and to Wuhan during Chunyun 2020 festival. For predictions, the famous forecasting model SEIR(Susceptible, Exposed, Infected, Recovered) was used to calculate the reproduction number R0. It was calculated to be 2.68 with a 95% credible interval(CrI). Markov Chain Monte Carlo method was used for R0 calculations.

However, there were certain assumptions here 1) serial interval of SAR-CoV2 was the same as that of SARS-CoV of 2003 that was 8.4 days; 2)incubation period of SAR-CoV2 was similar to that of SARS and MERS that was 6 days. There were certain limitations to the study. 1) asymptomatic patients were not considered and every infected patient was assumed to be mildly symptomatic; 2) the epidemic forecast was based on inter-city mobility data of 2019 and 2020 was not considered; 3) the model didn't consider the seasonality effects in the forecast; 4) death and recovered cases were not evaluated by the model.

(Fanelli and Piazza, 2020) analyzed the COVID-19 cases in Italy, France, and China using the dataset from the John Hopkins University, USA. The dataset had data from time-period 22nd

January 2020 till 15th March 2020. Data analysis was done by plotting the recurrence graph plots for confirmed cases, infected cases, and death cases for all the three countries. For prediction SEID(susceptible, exposed, infected, death) model was used. The differential equation was derived for each category S, E, I, and D. The graph was plotted for Italy and China as there was the problem of model fitting for France due to its limited data. Furthermore, it was observed that the recovery rate does not seem to depend on the country and it was the same for Italy and China whereas, infected cases and death cases were different for both the countries. The model predicted the peak in Italy at around 21st March however it turned out to be false as Italy had a huge number of cases in April. It also predicted the mortality rate of 4%-8% for Italy and 1%-3% in China.

(Tomar and Gupta, 2020) used a dataset having data from 30th January 2020 until 4th April 2020. They used 80% of the data for the model training and rest 20% for forecasting and validating the model's performance. Two data-driven algorithms used in the COVID-19 prediction were LTSM(Long Short Term Memory) and Curve fitting. The prediction was done for the next 30 days and the effects of social isolation and lockdown were also considered in preventing the spread. The prediction results for both the algorithms were shown by plotting the graph of total positive cases, daily number of cases, total number of recovered cases, and the total number of death cases. The analysis of the effects of preventive measures like social-isolation and lockdown was done by plotting a graph of different values of the transmission rate(r) from r=0.001 to r=2.3 which showed positive results.

(Tang, Cao, Lan and Cao, 2020) built a SIR(Susceptible, Infective, Removal) on the data of infected cases, discharged cases, and discharged patients during the period of isolation in Wuhan. For the model's key parameter estimation the Least squares method was used. The model fitted pretty well with the data and the results were matching with the actual data. It predicted 22000 cases in Wuhan and the inflection point was at the end of February 2020. Various graphs were plotted here such as 1) the predicted vs observed values for the SIR model for the infective cases plotted on days(X-axis) vs Number of people(Y-axis); 2) the predicted vs observed values for the SIR model for the removal cases plotted on days(X-axis) vs Number of people(Y-axis); 3) the predictions for the SIR model for the infectives and removals plotted on days(X-axis) vs Number of people(Y-axis).

2.2 Rationale of the research

As per the previous studies various algorithms were used for the predictions. But the data available was limited in most of the research. Hence, questions arise about the accuracy of these predictions. The rationale of this research is performing the prediction by using various categories of algorithms such as:

1) Time series based models like ARIMA, Prophet model by Facebook.
2) Regressions based Machine learning algorithms such as Linear Regression and Polynomial Regression.

These algorithms were used for predicting the confirmed cases, recovered cases, and death cases from datasets with data from 22nd January 2020 – 12th August 2020. Results of the various machine learning algorithms were compared and the best algorithm was picked up.

2.3 Research Questions and Research Objectives

Based on the literature reviews and their limitations following are the research questions, aim, objective, and hypothesis of the thesis.

Research Question 1: Which is the best algorithm among Prophet, ARIMA, Polynomial Regression, and Linear Regression for COVID-19 predictions?

Research Question 2: What will be the total number of COVID 19 positive cases, death cases, and recovered cases in the USA by August end based on the current data?

Research Question 3: According to the future COVID-19 predictions is there an increase in active cases?

Aim: Using various machine learning algorithms for predicting Coronavirus cases.

Objective: The objective of this research is to implement data mining algorithms such as Linear Regression, Polynomial Regression, Prophet, and ARIMA model for a COVID-19 dataset containing data from 22nd January 2020 – 12th August 2020. Then, calculating metrics such as MAE(Mean Absolute Error), MSE(Mean Squared Error), and RMSE(Root Mean Squared Error) for these models. Based on the metrics finding the best algorithm and using it for predicting COVID-19 confirmed cases, recovered cases, and death cases.

Hypothesis: Coronavirus confirmed cases will reach up to 6 Million by 1st September 2020.

CHAPTER 3: METHODOLOGY

3.1 Importance of Methodology

Selecting an appropriate methodology plays an important role in any data analytics project especially because a data analyst or data scientist has to deal with large datasets with structured or un-unstructured or semi-structured data. Steps such as dataset selection, data cleaning, data understanding, business understanding, data visualization, exploratory data analysis, finding outliers or correlation, data transformation, applying prediction models, data evaluation needs to be executed sequentially for good results. If all the steps are executed appropriately then as a result predictions results are accurate. Businesses, governmental bodies, executives who are relied on the prediction result can then take appropriate actions according to the forecast. Hence, it is important to follow a methodology. There are three types of data mining methodologies:

a) CRISP-DM(Cross Industry Process for Data Mining).
b) KDD(Knowledge Discovery Databases).
c) SEMMA(Sample, Explore, Modify, Model, Assess).

3.2 Methodology used in the research

CRISP-DM was used here because it incorporates the Business Understanding stage and deployment stage which is not present in KDD and SEMMA methods. Although the deployment part of the CRISP-DM was not included in this thesis project. Another benefit of using CRISP-DM was that transition between stages can be reversed which is not present in the other two methods i.e KDD and SEMMA (Data Science project management methodologies, 2020).

3.2.1 About CRISP-DM(Wirth and Hipp, 2020) and (Crisp DM methodology - Smart Vision Europe, 2020)

CRISP-DM provides an overview of the life cycle of a data mining project as it contains different phases of a project, respective tasks, and task outputs. It provides a structured approach to planning the data mining project. It also provides the flexibility of backtracking to previous tasks and taking the necessary actions.

There are six steps in the CRISP-DM method as shown in figure 3.1:

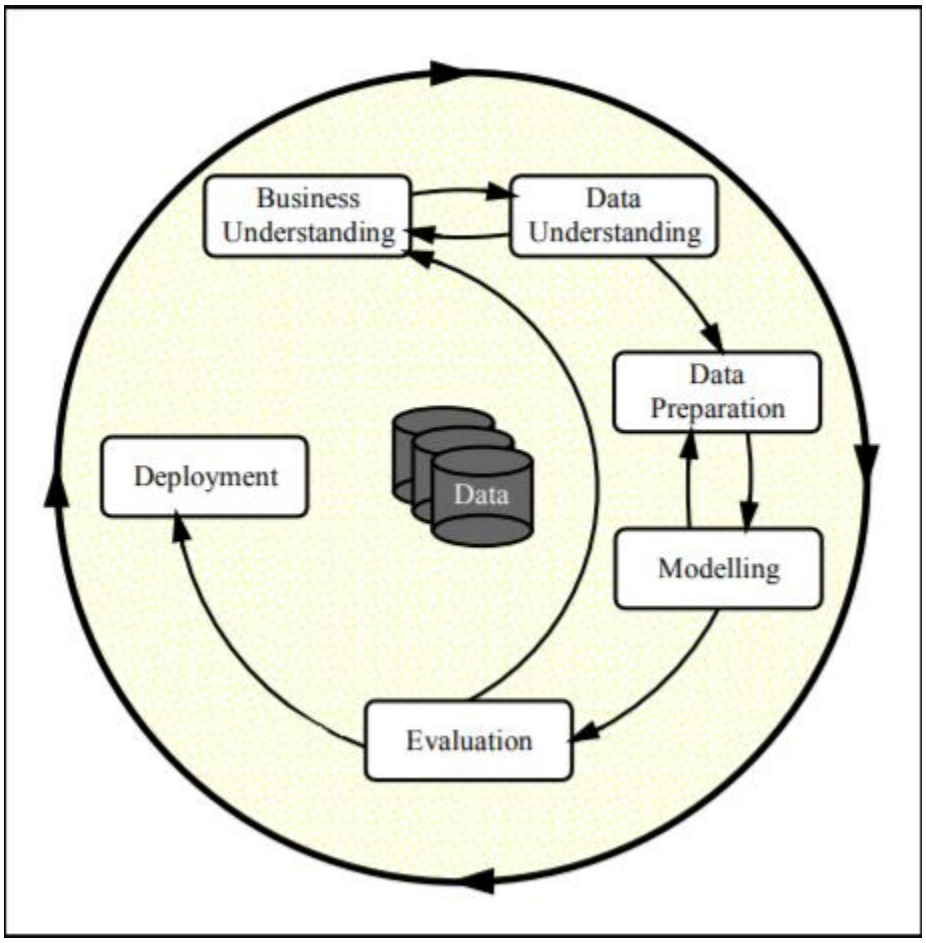

Figure 3.1: CRISP-DM flowchart

1) **Business Understanding**:- The first phase focuses on understanding the business objectives and it's requirements. This knowledge is later converted into a data mining problem definition

along with a preliminary plan of achieving the data mining and business goal. Along with this, business success criteria are also taken to determine whether the project is successful from the business point of view or not.

2) **Data Understanding**:- The phase starts with data collection or data loading or loading a tool for data understanding(if required). It also involves taking the steps to get familiar with the data identifying patterns, finding interesting subsets to form the hypothesis of the hidden information. Data exploration is done to address data mining questions using queries, visualization, and reporting tool. The data quality is also determined here to find out missing column values and checking the correctness of the data.

3) **Data Preparation**:- This phase covers all tasks in preparing the dataset from the initial raw data and data preparation tasks are performed multiple times to get the final dataset. Firstly, the following data is selected which is in the relevance of the data mining goals. Then, data cleaning is performed by removing/replacing null values. After this, data transformation is performed by inserting new records, making derived columns from existing attributes, encoding the values, etc.

4) **Modeling**:- In this phase, various data mining models are applied. The data models are selected as per the desired data mining goal. The parameter's value set for the model is decided in such a way to get optimum results. Then, the dataset is divided into a training set and a test set as per the requirements(usually 80-20 split or 70-30 is used). The model is built on the training set and its quality is evaluated on the test set. Finally, the model is built with the appropriate parameter settings.

5) **Evaluation**:- The models built in the previous step are evaluated here by checking if the models satisfy the business or data mining goals or not. If not, then the reasons for shortcomings are sought. The revision of the step for constructing the models is reviewed. The models that meet the business criteria become the approved models. At the end of this stage, the decisions to use the data mining results are also taken.

6) **Deployment**:- In the final stage, evaluation results are taken and a strategy is formed for their deployment. The deployment plan is formed in this stage has a summarization of the necessary steps and how they will be executed. Depending upon the business requirements, this step is as simple as generating a report or as complex as performing a repeatable data mining process. The deployment is generally done by the user by following the deployment steps given by data analysts. However, the deployment stage is not included as part of this thesis.

3.3 CRISP-DM in usage

In the business understanding phase, the business objective and goals were determined. The objective of this research is to determine the best machine learning algorithm for the prediction of confirmed COVID-19 cases, recovered cases, and death cases based on the current data and using it for predicting the future of the cases in the USA.

In the data understanding phase, the dataset was selected from the Kaggle. This dataset was also used in John Hopkins University and it contains information about COVID-19 statistics in the USA. The dataset mainly consists of attributes such as date, state, confirmed cases, pending cases, hospitalized cases, recovered cases, death cases, ICU cases. Then various python libraries such as Pandas, Numpy, Seaborn, etc were used for importing data, reading data, understanding

data distribution by plotting the scatter plots, relationships between different attributes, etc. The data understanding was achieved via EDA(Exploratory Data Analysis) technique.

In the data preparation phase, various functions were used in preparing the data for data modeling. The grouping was performing on the date attribute for creating a separate data frame, data scaling was also done to reshape the data for better performance of the model.

In the modeling phase, the dataset was divided into a training set and a test set with a 95%-5% ratio for all data mining algorithms. Libraries such as SK-learn, Fbprophet, stat-models, intertools were used. The model was fit on the training set to train and predictions were performed. Models used were Fbprophet, linear regression, polynomial regression, and ARIMA.

In the evaluation phase, the results of the models were compared with the test set. Plots were drawn between the actual data and predicted data to determine the accuracy and performance of the model. Various parameters such as MAE(Mean Absolute error), MSE(Mean Squared Error), RMSE(Root Mean Squared Error) were calculated, and the results were compared to determine the best data mining models that satisfy the business goals.

3.4 COVID-19 Forecasting Methods

Various kinds of machine learning algorithms and mathematical models were used for the prediction of COVID-19 confirmed cases, recovered cases, and death cases.

3.4.1 Prophet Model by Facebook (Taylor and Letham, 2020)

Prophet is a time series based model developed by Facebook in 2017. It is mainly used for forecasting. It is considered to be easy to use with just two parameters required which are **ds**(datestamp) and the value to be predicted denoted by **Y**. It is vigorous towards missing data

and identifying patterns in the data along with handling the anomalies pretty well. Prophet is a type of Additive Regression Model as shown in figure 3.2

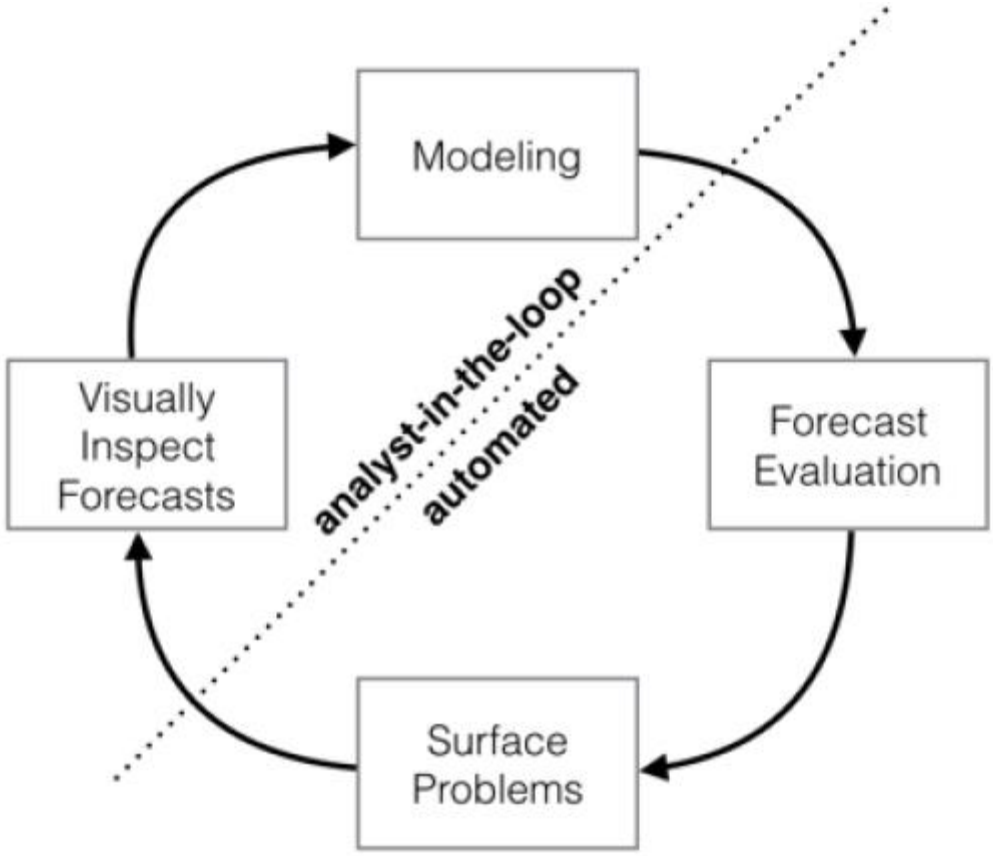

Figure 3.2: Prophet ARM

This model has three main components: trend, seasonality, and holidays. The equation is shown below:

$$y(t) = g(t) + s(t) + h(t) + Et$$

Here g(t) is the trend function accounted for modeling the trend changes in the value of the time-series. The s(t) represents periodic changes such as weekly or yearly seasonality. The h(t) represents the effect of holidays occurring which are on an irregular schedule over one or more days. The Et i.e error term represents idiosyncratic changes that are not accommodated by the model. Parametric assumptions are made later on that Et is normally distributed.

In the prophet model, there are several areas where an analyst can alter the model by applying his/her experience and knowledge. And the good thing is that an analyst doesn't need to understand the underlining statistics. For making those changes.

Capacities: If an analyst is having external data of the total market size and can apply the knowledge directly by specifying capacitors.

Changepoints: Known dates of change points, such as dates of product changes, can be specified without any complexity.

Holiday and Seasonality: Experienced analysts have experience in holiday impact growth in a particular region so that they can directly input the holiday dates and the applicable time scales of seasonality.

Smoothing parameters: By changing the value of **τ**, the analyst has the option of selecting from a global or local smoothing model. Parameter τ is responsible for controlling the flexibility of the model by altering its rate. If τ is increased model becomes more flexible in fitting the history hence training error is reduced. The seasonality and holiday smoothing parameters (σ, ν) help model gives an estimation to an analyst about how much historical variation is expected in the future.

3.4.2 Linear and Polynomial Regression

Regression is a type of predictive modeling technique that is used to find a relationship between a dependent and an independent variable. In this, the relationship between the variables is used to find the best fit line or the regression equation which is used for making the predictions (Introduction to Linear Regression and Polynomial Regression, 2020).

3.4.2.1 Linear Regression

In this method, the dependent variable is continuous and the relationship between a dependent and an independent variable is assumed to be linear. The linear regression is similar to the equation of the line in mathematics. The equation is represented below

$$Y = B(0) + B(1)*X + E$$

Where Y is the dependent variable, X is the independent variable, B(0) is the Y-intercept, and B(1) is the slope of the line and E is the error rate whose role is to add bais.

The equation for multiple linear regression is shown below

$$Y = B(0) + B(1)*X(1) + B(2)*X(2) + B(3)*X(3) +\ldots + B(K)*X(K) + E$$

Here a single dependent variable Y is dependent on the values of multiple independent variable X.

Following are the assumptions for obtaining good Linear Regression results (15 Types of Regression in Data Science, 2020):

1) There should be no outliers present in the data. An outlier is a value in the dataset that is either too high or too low as compared to the rest of the data. Due to the outlier, the results are distorted.
2) It assumes no Heteroscedasticity in the data. It means that the variation in the value of the dependent variable is not the same as that of an independent variable.

3) Absence of multicollinearity when the independent variables are highly correlated to each other as it causes difficulty in selecting the most important independent variable in case of multiple linear regression.
4) There should be no underfitting of data i.e when the model cannot even fit the training data. The data overfitting is also not good in linear regression. In overfitting, too many explanatory variables reduce the actual learning of the model. As a result, it performs well for the training data and poorly for the test data.

3.4.2.2 Polynomial Regression

Polynomial regression fits the higher degree of relations between a dependent variable and an independent variable. When the dataset cannot be fit with a linear regression then polynomial regression is used in that case. Figure 3.3 for polynomial regression is shown below (15 Types of Regression in Data Science, 2020)

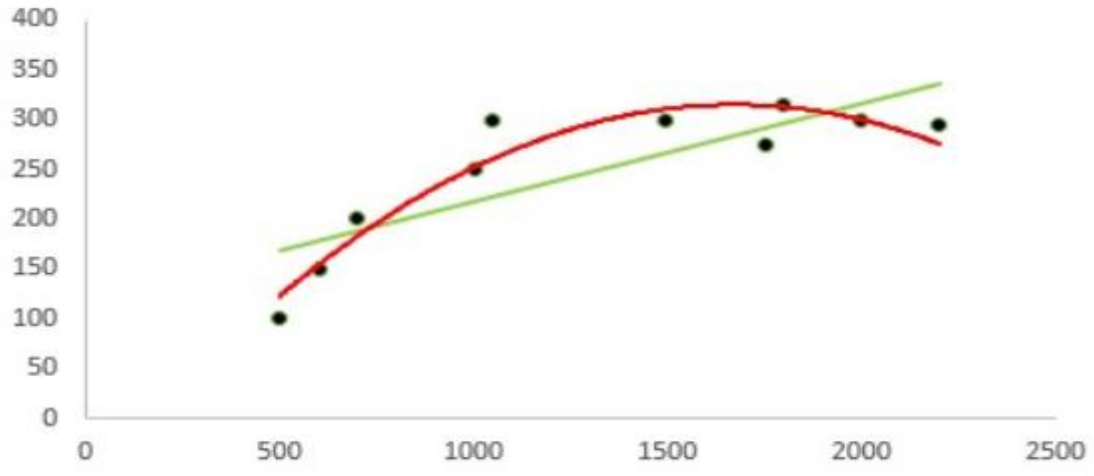

Figure 3.3: Linear Regression vs Polynomial Regression

If linear regression is used to fit the data of the above figure the error rate(E) would be very high. Hence, to have the lowest value of E, polynomial regression is used. If the relation between

dependent and independent variables is non-linear then polynomial regression is used. The equation is shown as

$$Y = B(0) + B(1)*X + B(2)*X^2 + B(3)*X^3 + \ldots + B(K)*X^K + E$$

3.4.3 ARIMA model

ARIMA stands for Autoregressive Integrated Moving Average. It is a generalization of the autoregressive moving average (ARMA)model. Both the models are fitted to time-series data. However, the ARIMA model is applied in cases where data shows the evidence of non-stationarity.

An ARMA model describes the conditional mean of y(t) as a function for both past observation Y(t-1), Y(t-2),.., Y(t-p) as well as past innovation E(t-1), E(t-2),.., E(t-p) (Devi, B.U., Sundar, D. and Alli, P., 2013). The AR degree is defined by the number of past observations in which Y(t) depends upon p and the number of past innovations in which Y(t) depends upon p is the MA degree.

Generally, these models are denoted by ARMA(p,q). The form of the ARMA(p,q) model is:

$$y(t) = C + \emptyset(1)\, y(t-1) + \emptyset(2)\, y(t-2) + \ldots + \emptyset(p)\, y(t-p) + \in(t) + \theta(1) \in(t-1) + \emptyset 2 \in t-2 + \ldots + \theta(q) \in(t\text{-q})$$

Where E(t) is an uncorrelated innovation process with mean zero. Y(t) is the actual value and E(t) is the random error at time t. The parameters used in the model are

C – Constant term

AR – Nonseasonal AR coefficients.

MA – Nonseasonal MA coefficients.

ARLags – Lags corresponding to nonzero, i.e no seasonal AR coefficients.

MALags - Lags corresponding to nonzero, i.e no seasonal MA coefficients.

D – Degree of non-seasonal differencing

Variance – Scaller variance of the innovation process.

Distribution – Distribution of the innovation process.

In the ARIMA model, the future value of the variable is the linear combination of the past value and past error. The ARIMA(1,0,1) is shown below:

$$(t) = (0) + \emptyset(1)\, y(t-1) + \varepsilon(t) - \theta(1)\varepsilon(t-1)$$

CHAPTER 4: IMPLEMENTATION AND FINDINGS

4.1 Introduction

In this section, the implementation of the artifact is explained in detail. Starting with the description of the software packages, tools, and the libraries used in the project. Then EDA(Exploratory Data Analysis) of the COVID-19 dataset is covered up followed by the data visualizations in Tableau. An interactive dashboard is also designed which helps in analyzing the Coronavirus trend in the USA for different states. Finally, the last sub-section covers the prediction for the COVID-19 situation in the USA by using different data mining algorithms.

4.2 Software packages, tools, and Libraries

Table 4.1: Tools and technologies used in the thesis.

Libraries	Programming language	Compiler	Visualization tools	Version Control
Numpy Pandas Seaborn Matplotlib SK-learn Math Statsmodels Itertools Fbprophet Plotly Warnings	Python	Jupyter Notebook	Tableau 2019.4.3	Github

4.3 Exploratory Data Analysis

The dataset has one CSV(Comma Separated Values) named “us_states_covid19_daily.csv”. It contains information about various COVID-19 statistics in the USA from the period 22nd January 2020 till 12th August 2020. This dataset was taken from the Kaggle and it was also used in John Hopkins University research. The file has 41 attributes. The most important attributes used in the thesis are explained below

date:- Contains date record in the format “YYYYMMDD”

state:- Representing 56 states of the USA and all assigned with a unique code.

positive:- Number of confirmed COVID-19 cases in every state each day.

negative:- Number of negative COVID-19 cases in every state each day.

pending:- Number of pending COVID-19 cases in every state each day.

total:- The total number of COVID-19 cases reported in every state each day.

hospitalizedCurrently:- Number of hospitalized COVID-19 cases in every state each day.

inICUCurrently:- Number of COVID-19 patients who are in ICU in every state each day.

onVentilatorCurrently:- Number of COVID-19 patients who are on a ventilator in every state each day.

recovered:- Number of recovered COVID-19 cases in every state on each day.

death:- Number of COVID-19 death cases in every state on each day.

The dataset has the data in a time-series manner with the values of the attributes increasing each day. The new data of all the attributes every day gets added to the previous day's data. Hence, according to the dataset 12th, August contains the most recent figures for COVID-19 containing the total number of positive, negative, total, recovered, death, hospitalized, ICU, ventilator cases.

The dataset was first read into a Jupyter notebook using the Pandas library and it was collected into a data frame named “data” as shown in figure 4.1

```
In [1]: import pandas as pd
        import numpy as np
        import seaborn as sns
        import matplotlib.pyplot as plt

In [3]: #reading the dataset
        data = pd.read_csv(r"C:\Users\Manik325\Desktop\Thesis Data\Thessis Dataset\USA_latest\us_states_covid19_daily.csv")
```

Figure 4.1: Reading the dataset

The rows and columns in the dataset are shown in figure 4.2

```
In [10]: data.shape
Out[10]: (8977, 41)
```

Figure 4.2: Shape of the dataset

The top 5 rows of the dataset are shown in figure 4.3

```
In [5]: data.head()
```

Out[5]:

	date	state	positive	negative	pending	hospitalizedCurrently	hospitalizedCumulative	inIcuCurrently	inIcuCumulative	onVentilatorCurrently	...	posNe
0	20200812	AK	4655.0	287927.0	NaN	39.0	NaN	NaN	NaN	3.0	...	2925
1	20200812	AL	104786.0	690985.0	NaN	1372.0	12292.0	NaN	1282.0	NaN	...	7957
2	20200812	AR	51114.0	522457.0	NaN	486.0	3472.0	NaN	NaN	113.0	...	5735
3	20200812	AS	0.0	1396.0	NaN	NaN	NaN	NaN	NaN	NaN	...	13
4	20200812	AZ	189443.0	854785.0	NaN	1469.0	19821.0	519.0	NaN	328.0	...	10442

5 rows × 41 columns

Figure 4.3: Top 5 rows of the dataset

As it can be seen that there were many fields where data was not available. All those fields were replaced with 0 using the fillna() function of Pandas as shown in figure 4.4

```
In [6]: data = data.fillna(0)

In [7]: data.head()
```

Out[7]:

	date	state	positive	negative	pending	hospitalizedCurrently	hospitalizedCumulative	inIcuCurrently	inIcuCumulative	onVentilatorCurrently	...	posNe
0	20200812	AK	4655.0	287927.0	0.0	39.0	0.0	0.0	0.0	3.0	...	2925
1	20200812	AL	104786.0	690985.0	0.0	1372.0	12292.0	0.0	1282.0	0.0	...	7957
2	20200812	AR	51114.0	522457.0	0.0	486.0	3472.0	0.0	0.0	113.0	...	5735
3	20200812	AS	0.0	1396.0	0.0	0.0	0.0	0.0	0.0	0.0	...	13
4	20200812	AZ	189443.0	854785.0	0.0	1469.0	19821.0	519.0	0.0	328.0	...	10442

5 rows × 41 columns

Figure 4.4: Top 5 rows after data cleaning

The description of the columns with the respective data types is shown in figure 4.5

```
In [16]: data.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 8977 entries, 0 to 8976
Data columns (total 41 columns):
date                        8977 non-null int64
state                       8977 non-null object
positive                    8977 non-null float64
negative                    8977 non-null float64
pending                     8977 non-null float64
hospitalizedCurrently       8977 non-null float64
hospitalizedCumulative      8977 non-null float64
inIcuCurrently              8977 non-null float64
inIcuCumulative             8977 non-null float64
onVentilatorCurrently       8977 non-null float64
onVentilatorCumulative      8977 non-null float64
recovered                   8977 non-null float64
dataQualityGrade            8977 non-null object
lastUpdateEt                8977 non-null object
dateModified                8977 non-null object
checkTimeEt                 8977 non-null object
death                       8977 non-null float64
hospitalized                8977 non-null float64
dateChecked                 8977 non-null object
totalTestsViral             8977 non-null float64
positiveTestsViral          8977 non-null float64
negativeTestsViral          8977 non-null float64
positiveCasesViral          8977 non-null float64
deathConfirmed              8977 non-null float64
deathProbable               8977 non-null float64
fips                        8977 non-null int64
positiveIncrease            8977 non-null int64
negativeIncrease            8977 non-null int64
total                       8977 non-null int64
totalTestResults            8977 non-null int64
totalTestResultsIncrease    8977 non-null int64
posNeg                      8977 non-null int64
deathIncrease               8977 non-null int64
hospitalizedIncrease        8977 non-null int64
hash                        8977 non-null object
commercialScore             8977 non-null int64
negativeRegularScore        8977 non-null int64
negativeScore               8977 non-null int64
positiveScore               8977 non-null int64
score                       8977 non-null int64
grade                       8977 non-null float64
dtypes: float64(19), int64(15), object(7)
memory usage: 2.8+ MB
```

Figure 4.5: Dataset datatype Information

It can be seen that there are 19 attributes with data type as float, 15 integer attributes, and 7 objects attributes. The columns in the dataset are shown in figure 4.6

```
In [8]: data.columns

Out[8]: Index(['date', 'state', 'positive', 'negative', 'pending',
               'hospitalizedCurrently', 'hospitalizedCumulative', 'inIcuCurrently',
               'inIcuCumulative', 'onVentilatorCurrently', 'onVentilatorCumulative',
               'recovered', 'dataQualityGrade', 'lastUpdateEt', 'dateModified',
               'checkTimeEt', 'death', 'hospitalized', 'dateChecked',
               'totalTestsViral', 'positiveTestsViral', 'negativeTestsViral',
               'positiveCasesViral', 'deathConfirmed', 'deathProbable', 'fips',
               'positiveIncrease', 'negativeIncrease', 'total', 'totalTestResults',
               'totalTestResultsIncrease', 'posNeg', 'deathIncrease',
               'hospitalizedIncrease', 'hash', 'commercialScore',
               'negativeRegularScore', 'negativeScore', 'positiveScore', 'score',
               'grade'],
              dtype='object')
```

Figure 4.6: Dataset column names

The describe() in the pandas was used to get information about count, mean, standard deviation, minimum, and maximum as shown in figure 4.7

```
In [14]: data.describe()
```

Out[14]:

	date	positive	negative	pending	hospitalizedCurrently	hospitalizedCumulative	inIcuCurrently	inIcuCumulative	onVentilatorCurr
count	8.977000e+03	8977.000000	8.977000e+03	8977.000000	8977.000000	8977.000000	8977.000000	8977.000000	8977.00
mean	2.020054e+07	33454.825109	3.313219e+05	139.468197	683.138911	3003.065055	136.127325	123.438899	54.07
std	1.543341e+02	69376.437079	7.218179e+05	2069.332898	1628.612930	10512.346282	429.883455	409.842098	171.33
min	2.020012e+07	0.000000	0.000000e+00	0.000000	0.000000	0.000000	0.000000	0.000000	0.00
25%	2.020041e+07	788.000000	1.313600e+04	0.000000	0.000000	0.000000	0.000000	0.000000	0.00
50%	2.020052e+07	7652.000000	9.352800e+04	0.000000	130.000000	71.000000	0.000000	0.000000	0.00
75%	2.020070e+07	34812.000000	3.395280e+05	0.000000	648.000000	1851.000000	81.000000	0.000000	22.00
max	2.020081e+07	586056.000000	8.717411e+06	64400.000000	18825.000000	89995.000000	5225.000000	3929.000000	2425.00

8 rows × 34 columns

Figure 4.7: Dataset description

The groupby() function of the Pandas was used on the date attribute to get COVID-19 positive, negative, death, hospitalized, recovered, etc. cases on each day from 22nd January 2020 till 12th August 2020 in all the states combined in the USA. The sum() function of the pandas was used on all the relevant attributes except the date.

It can be seen in figure 4.8

```
In [20]: df = data.groupby("date")["positive", "death", "recovered","total","hospitalizedCurrently","inIcuCurrently","onVentilatorCurre
         df
```

Out[20]:

	date	positive	death	recovered	total	hospitalizedCurrently	inIcuCurrently	onVentilatorCurrently
0	20200122	2.0	0.0	0.0	2	0.0	0.0	0.0
1	20200123	2.0	0.0	0.0	2	0.0	0.0	0.0
2	20200124	2.0	0.0	0.0	2	0.0	0.0	0.0
3	20200125	2.0	0.0	0.0	2	0.0	0.0	0.0
4	20200126	2.0	0.0	0.0	2	0.0	0.0	0.0
5	20200127	2.0	0.0	0.0	2	0.0	0.0	0.0
6	20200128	2.0	0.0	0.0	2	0.0	0.0	0.0
7	20200129	3.0	0.0	0.0	3	0.0	0.0	0.0
8	20200130	3.0	0.0	0.0	3	0.0	0.0	0.0
9	20200131	3.0	0.0	0.0	3	0.0	0.0	0.0
10	20200201	4.0	0.0	0.0	4	0.0	0.0	0.0
11	20200202	6.0	0.0	0.0	6	0.0	0.0	0.0
12	20200203	7.0	0.0	0.0	7	0.0	0.0	0.0
13	20200204	8.0	0.0	0.0	8	0.0	0.0	0.0
14	20200205	8.0	0.0	0.0	8	0.0	0.0	0.0
15	20200206	11.0	0.0	0.0	11	0.0	0.0	0.0
16	20200207	12.0	0.0	0.0	12	0.0	0.0	0.0
17	20200208	13.0	0.0	0.0	13	0.0	0.0	0.0

Figure 4.8: Groupby on Date

The date column was converted into a standard format using to_datetime() function as shown in figure 4.9

```
In [10]: date = pd.to_datetime(df["date"], format="%Y%m%d")
         date
```

```
Out[10]: 0     2020-01-22
         1     2020-01-23
         2     2020-01-24
         3     2020-01-25
         4     2020-01-26
         5     2020-01-27
         6     2020-01-28
         7     2020-01-29
         8     2020-01-30
         9     2020-01-31
         10    2020-02-01
         11    2020-02-02
         12    2020-02-03
         13    2020-02-04
         14    2020-02-05
         15    2020-02-06
         16    2020-02-07
         17    2020-02-08
         18    2020-02-09
         19    2020-02-10
         20    2020-02-11
```

Figure 4.9: Date converted to DateTime format

To understand the data distribution between different attributes the pairplot() of the seaborn library was used. The results are shown in figure 4.10

```
In [24]: sns.pairplot(df)
Out[24]: <seaborn.axisgrid.PairGrid at 0xa7be3296d8>
```

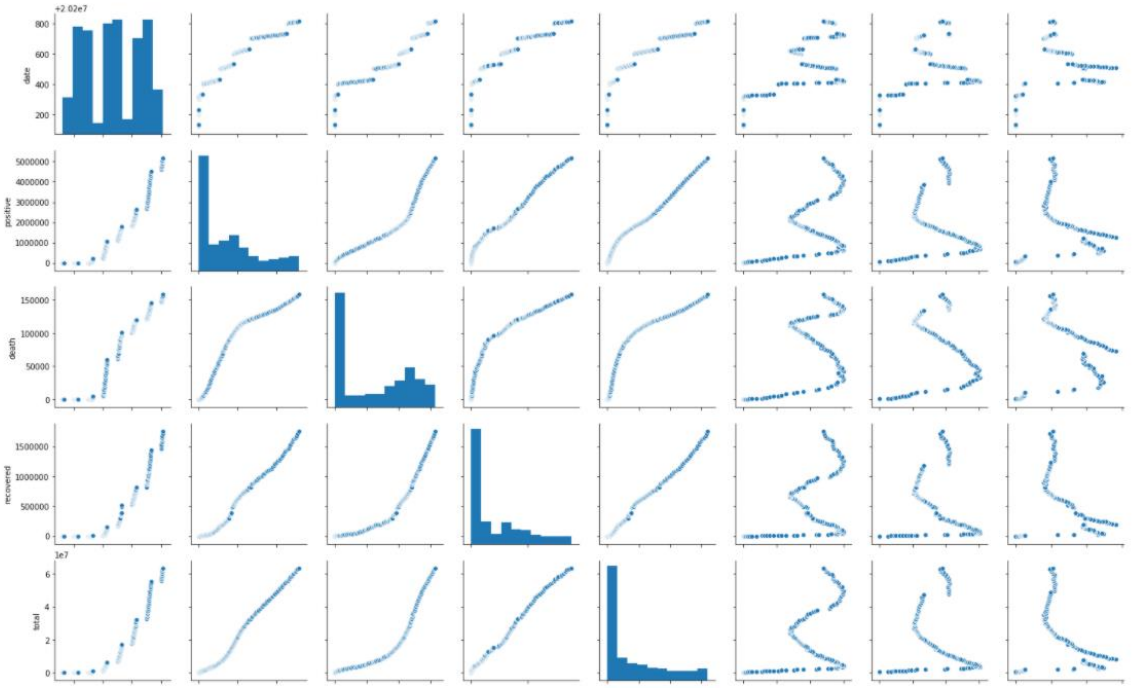

Figure 4.10: Dataset Pairplot

The data distribution on the graph is shown using the lmplot() function. The plot is between positive or confirmed cases(X-axis) and recovered cases(Y-axis) as shown in figure 4.11

```
In [16]: sns.lmplot(x="positive", y="recovered", data=df);
```

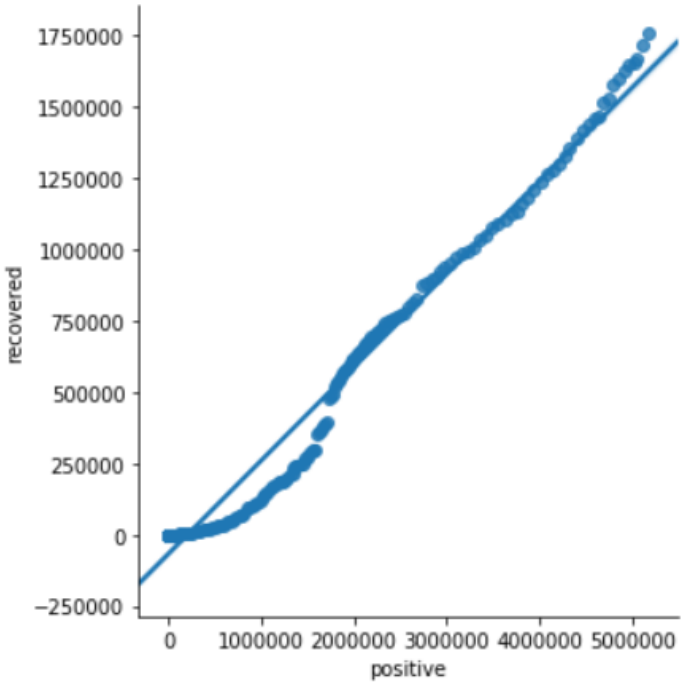

Figure 4.11: Positive vs Recovered lm-plot

It can be observed that although there are a high number of positive cases but the recovery rate is also good. To understand the relationship between the total number of cases and hospitalized cases scatterplot() function was used as shown in figure 4.12

```
In [30]: sns.scatterplot(x="total", y="hospitalizedCurrently", data=df);
```

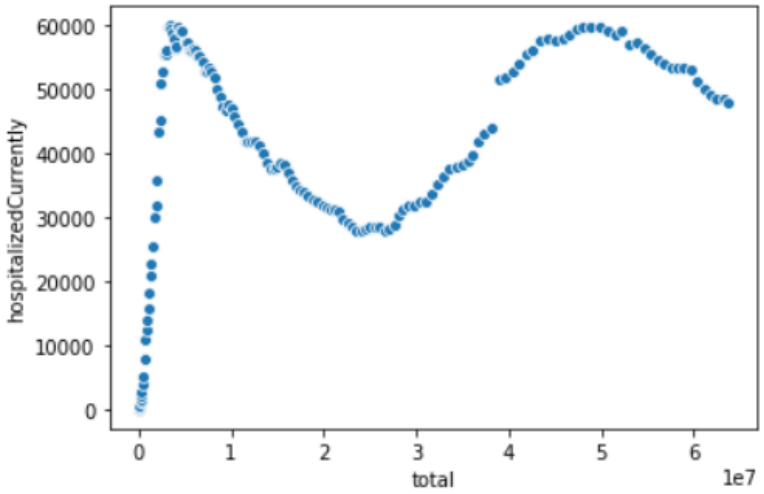

Figure 4.12: Total vs Currently Hospitalized Scatter-plot

From the above figure it can be observed that initially, the hospitalized cases increased exponentially as the government was not aware of the appropriate pandemic response(the same with all the countries around the world). As soon as the government and the people became more aware of the virus, the home-quarantine was adopted for most of the patients as they had mild symptoms. Hence, there was a fall in hospitalized cases thereafter. However, it can also be seen that hospitalized cases increased again in April-May 2020 it is because the COVID-19 is a highly contagious virus and it spreads exponentially. Moreover, the old age people(> 65) and people with underlining health-conditions such as diabetes, heart problems, etc. develop severe

symptoms and they are hospitalized. The same with the case of positive cases vs ICU cases as shown in figure 4.13

```
In [33]: sns.scatterplot(x="positive", y="inIcuCurrently", data=df);
```

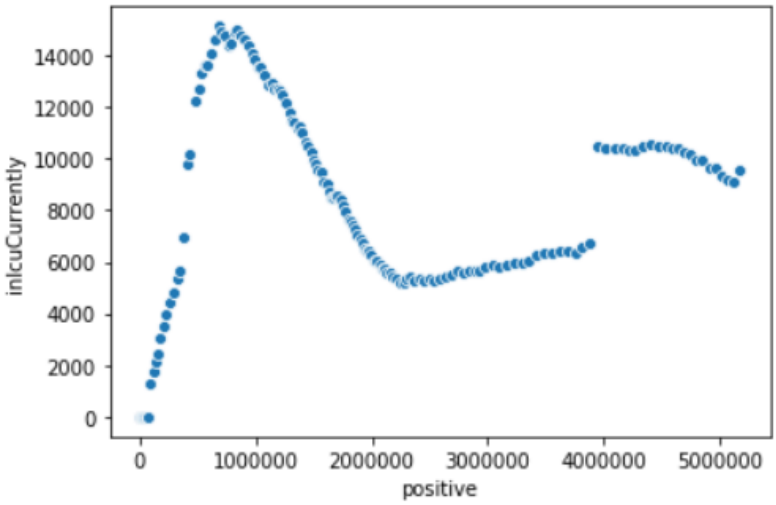

Figure 4.13: Positive vs ICU cases Scatter-plot

The graph between the date(X-axis) and the positive cases(Y-axis) was plotted with the barplot() function. The color contrast property was used for better visualization. There were a total of 5.17 million cases as shown in figure 4.14

```
In [37]: sns.set_style('darkgrid')
         plt.figure(figsize=(20,10))
         sns.barplot(x = date, y = df['positive'], palette='YlOrRd')
         plt.xticks(rotation = 90)
         plt.show()
```

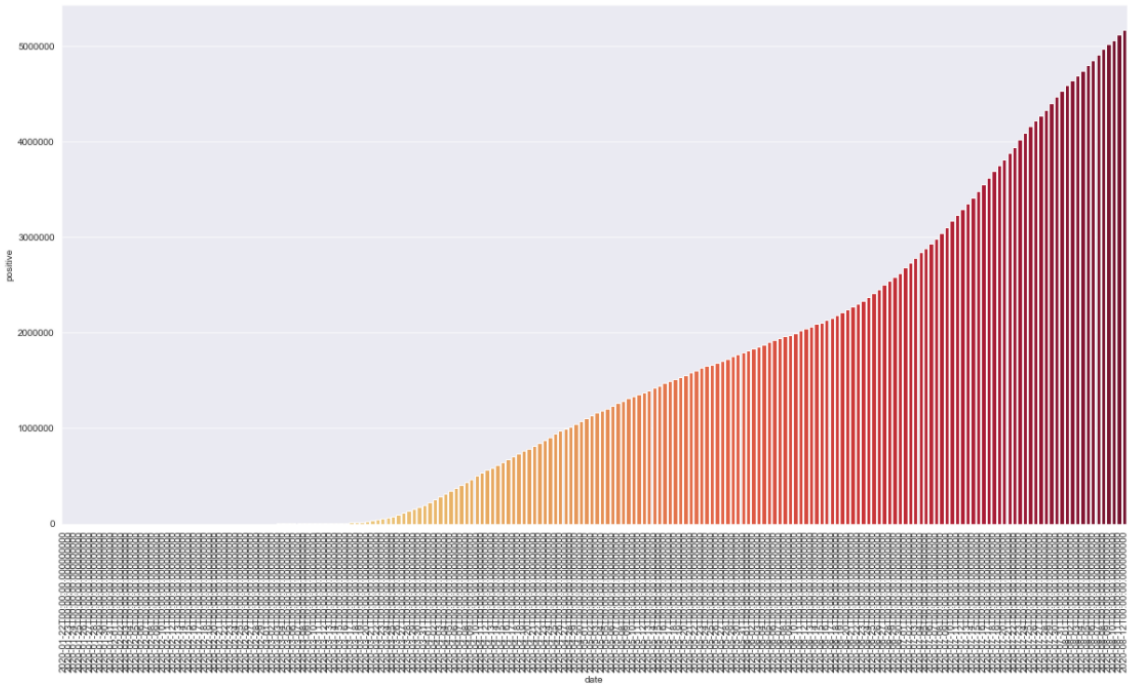

Figure 4.14: Date vs Positive Cases Bar-plot

Please note that X-axis values are not visible for figures 4.14, 4.15, and 4.16 due to a large number of dates from 22nd January 2020 – 12th August 2020.

The graph was also plotted between date and recovered cases using the same function. The total recovered cases until 12th August was around 1.75 million as shown in figure 4.15

```
In [38]: sns.set_style('darkgrid')
         plt.figure(figsize=(20,10))
         sns.barplot(x = date, y = df['recovered'], palette='Blues_d')
         plt.xticks(rotation = 90)
         plt.show()
```

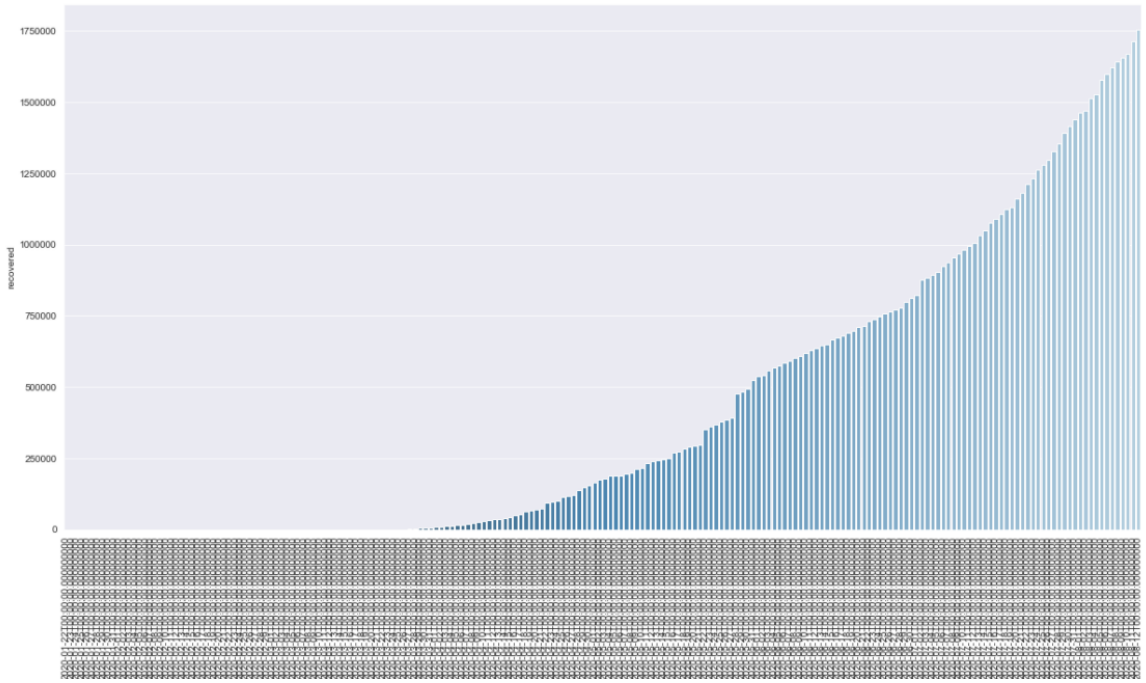

Figure 4.15: Date vs Recovered Cases Bar-plot

Finally, the graph between date and the death cases were also plotted, the total number of unfortunate death till 157K until 12th August. The graph is shown in figure 4.16

```
In [39]: sns.set_style('darkgrid')
         plt.figure(figsize=(20,10))
         sns.barplot(x = date, y = df['death'], palette='ch:.25')
         plt.xticks(rotation = 90)
         plt.show()
```

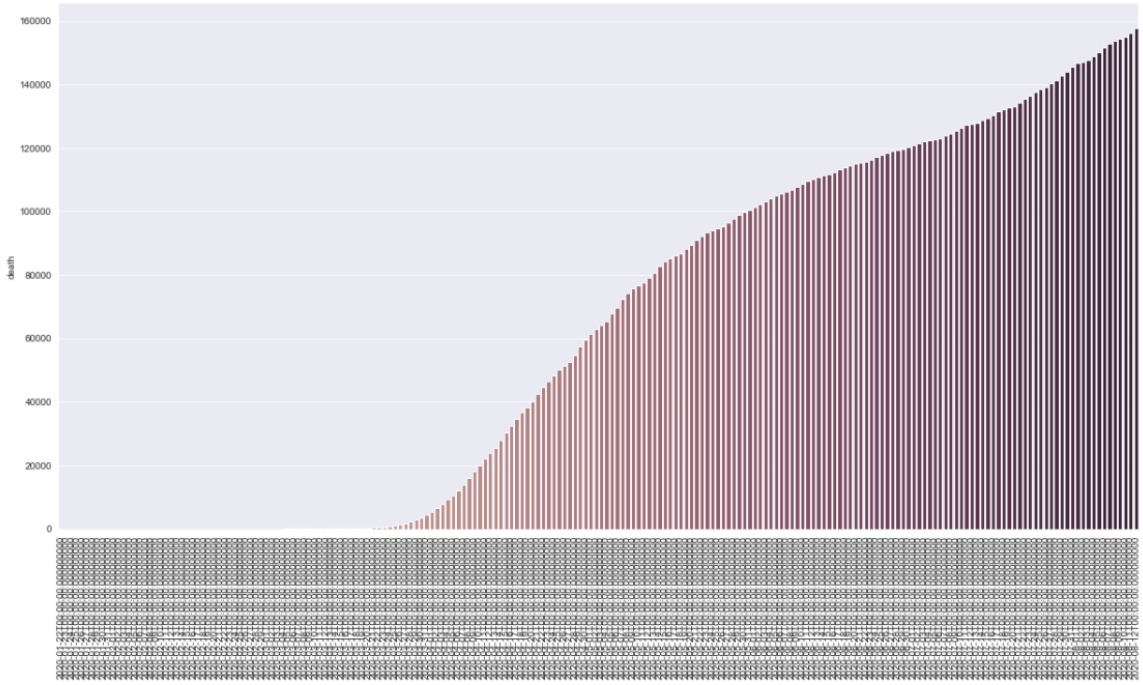

Figure 4.16: Date vs Death Cases Bar-plot

4.4 Visualization using Tableau

The dataset was loaded in Tableau and the visualizations were performed by creating 4 sheets and a dashboard using the same sheets. The time-series filter was created for all the sheets. It has a range of dates as that of the dataset i.e from 22nd January 2020 till 12th August 2020. While clicking the forward button it shows the increasing trend of Coronavirus in the USA based on different parameters and the decreasing trend is shown when the back button is clicked.

In the first sheet, the visualization was done on a map using the auto-generated attributes such as latitude and longitude. The map displays the geographical location of all the 56 states in the USA. By using the time-series filter with an increasing and decreasing trend of the COVID-19 in 56 states can be seen in figure 4.17

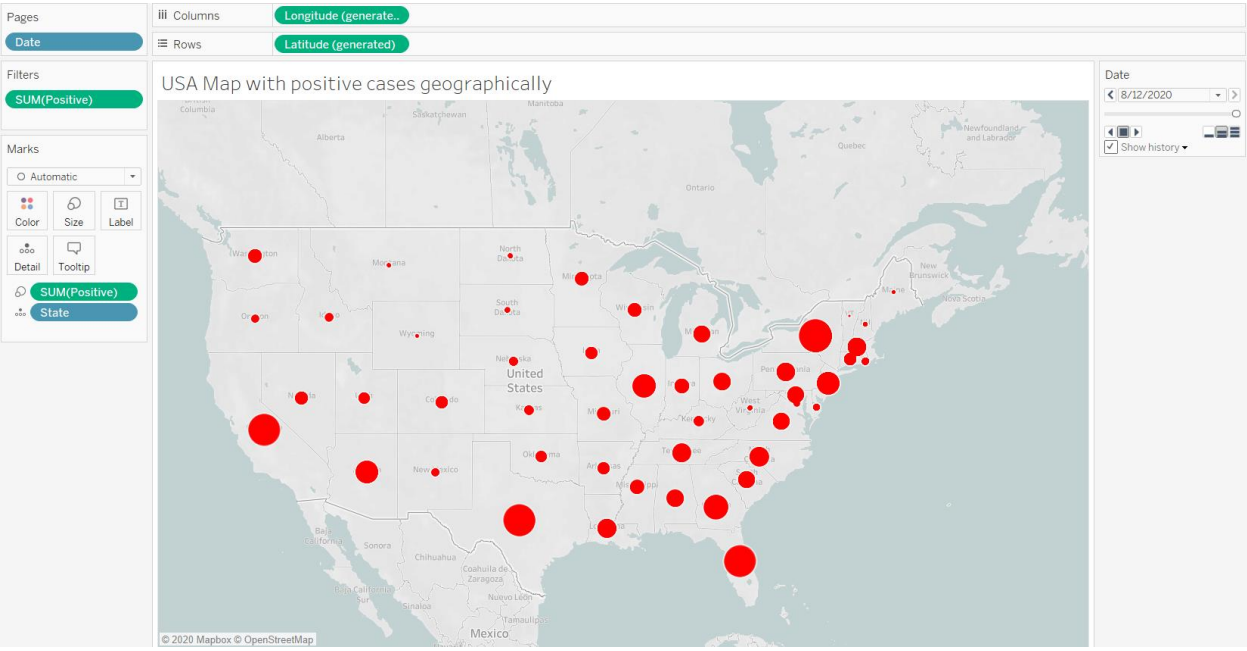

Figure 4.17: USA Map Visualization for Positive cases

The second sheet consists of the positive COVID-19 cases for all the 56 states in the USA. It can be observed that New York(NY), California(CA), Florida(FL), Texas(TX), and Georgia(GA) are the top 5 states in the USA with confirmed COVID-19 cases. The sum() function for positive Coronavirus cases was also used which shows the total case count in each state. The sheet is shown in figure 4.18

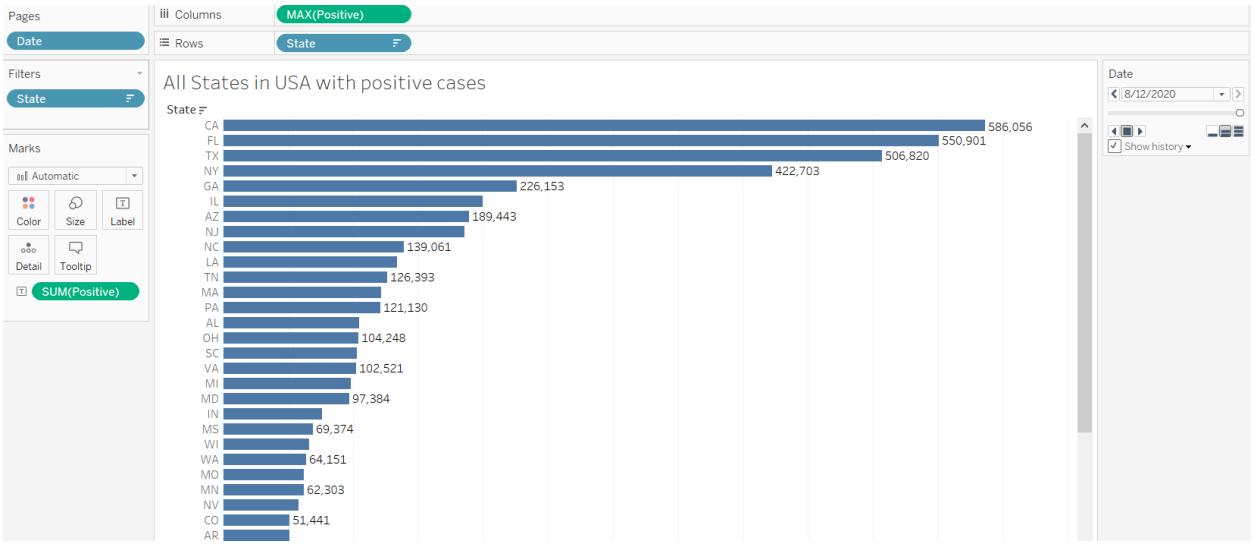

Figure 4.18: USA Positive cases for all states

In the third sheet, the positive, recovered, and death cases were visualized in an area chart design. All the states were plotted on the X-axis and the three types of cases on the Y-axis. It can be noted that the max() function was used as it shows the latest figure(12th August 2020) of COVID-19 in each state. The sheet is shown in figure 4.19

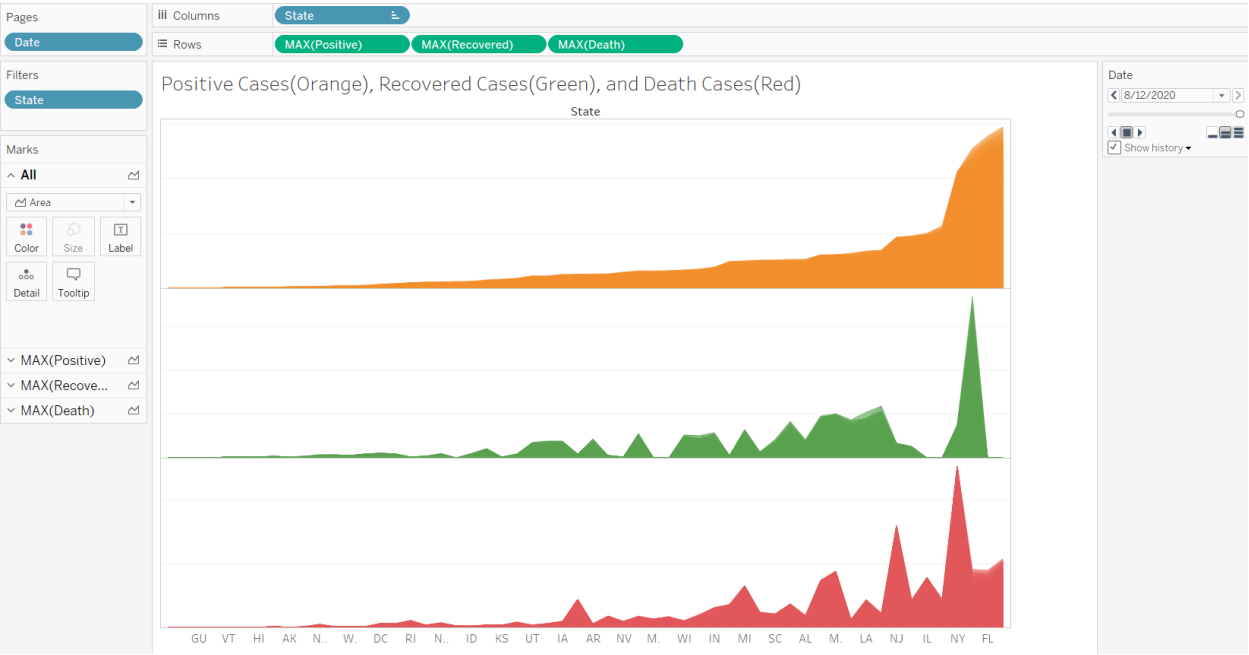

Figure 4.19: Positive, Recovered vs Death cases for all states

The fourth sheet visualizes the hospitalized, ICU, and ventilator cases for all the states. The design used in this sheet is a bar chart. It can be observed that New York(NY), California(CA), Florida(FL), Texas(TX), and New Jersey(NJ) are the top 5 states here. It is because these are the states with the highest number of COVID-19 confirmed cases.

The graph can be seen in figure 4.20

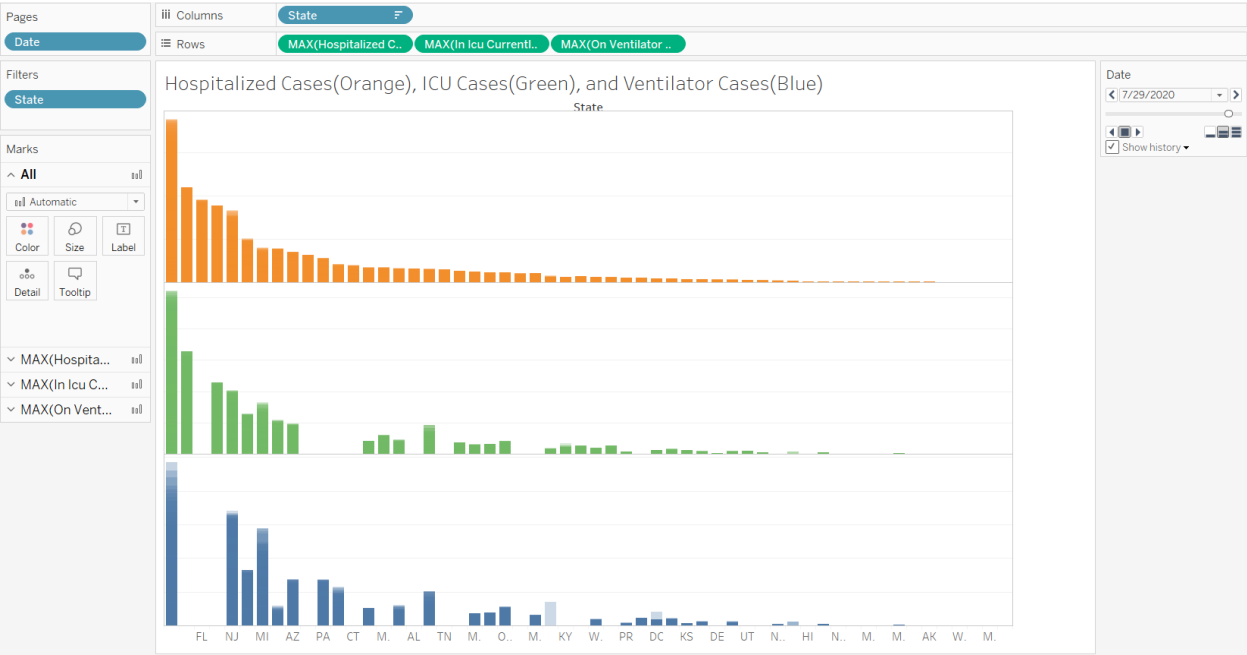

Figure 4.20: Hospitalized, ICU, and Ventilator cases for all states

Finally, the COVID-19 time-series dashboard consisting of all the four sheets is shown in figure 4.21

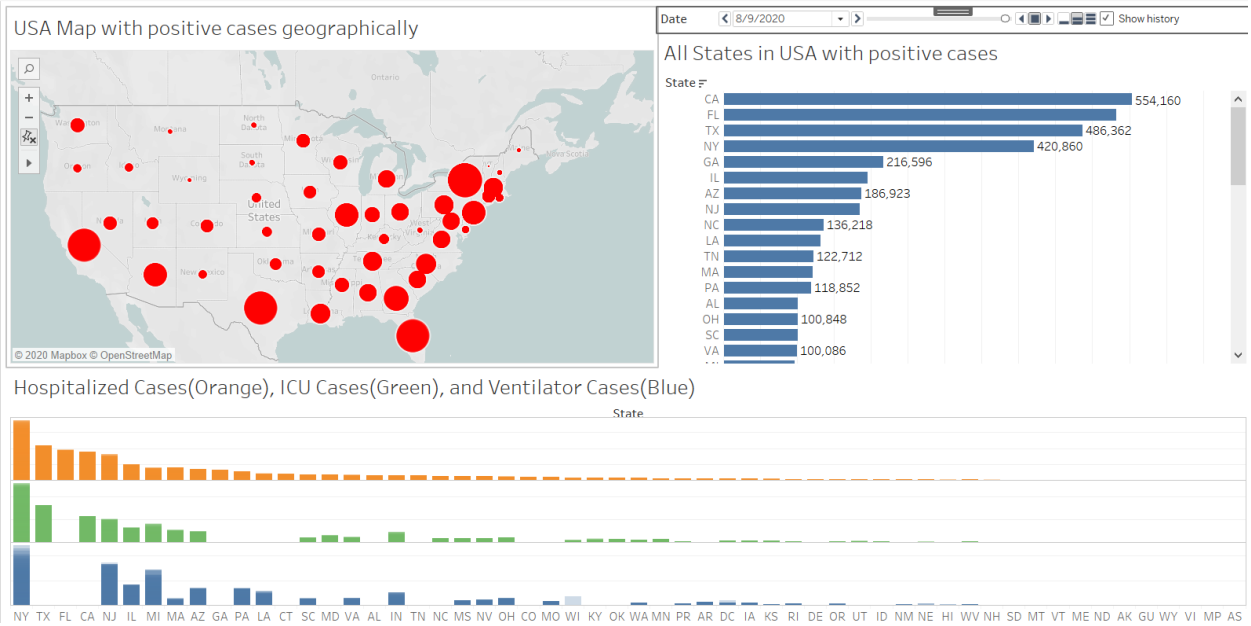

Figure 4.21: COVID-19 Tableau dashboard

4.5 Prediction for COVID-19

Implementation was done using fbprophet, polynomial regression, LSTM, and ARIMA model. The predictions were done until 31st August 2020. Parameters such as Mean Absolute Error(MAE), Mean Squared Error(MSE), and Root Mean Squared Error(RMSE) were calculated for confirmed cases, recovered cases, and death cases for comparison and performance evaluation.

4.5.1 Predictions by FBprophet Model

After reading the dataset and grouping in a data frame according to date using the groupby() function, the implementation for confirmed cases of COVID-19 was performed.

The fbprophet takes two parameters datestamp(ds) and the value to be predicted(y) as shown in figure 4.22

```
In [99]: Confirmed_Cases = df[["date","positive"]]

In [101]: Confirmed_Cases.rename(columns={"date":"ds","positive":"y"},inplace=True)

In [216]: Confirmed_Cases.tail()
```

Out[216]:

	ds	y
199	2020-08-08	4967754.0
200	2020-08-09	5019073.0
201	2020-08-10	5060880.0
202	2020-08-11	5116474.0
203	2020-08-12	5172509.0

Figure 4.22: Confirmed cases using Prophet

The variable Confirmed_cases was later split into a training set and a testing set with a 95%-5% ratio. In the next step, the new instance of the prophet model was created and the variable train_confirmed was fit in it as shown in figure 4.23

```
In [21]: m = Prophet()

In [22]: m.fit(train_confirmed)
INFO:fbprophet:Disabling yearly seasonality. Run prophet with yearly_seasonality=True to override this.
INFO:fbprophet:Disabling daily seasonality. Run prophet with daily_seasonality=True to override this.
Out[22]: <fbprophet.forecaster.Prophet at 0x533064e8d0>
```

Figure 4.23: Prophet data training

As there is no seasonality in the dataset so the warning was ignored. In the next step, the make_future_dataframe() function was used that takes one integer parameter “periods”. It tells about the number of days for which the prophet model should make predictions by taking into account the last date of the training set. The train_confirmed have the last day of 02nd August

hence the value for periods given was 29 to find the predictions until 31st August. The output was collected in a variable future_dates as shown in figure 4.24

```
In [23]: future_dates = m.make_future_dataframe(periods=29)

In [24]: future_dates.tail()
```

Out[24]:

	ds
218	2020-08-27
219	2020-08-28
220	2020-08-29
221	2020-08-30
222	2020-08-31

Figure 4.24: Future date list using Prophet

Then the predict() function of the prophet model was used on the variable future_dates. The output gave many parameters with the three parameters being the most important i.e yhat(predicted value), yhat_upper(higher range of the predicted value), and yhat_lower(lowest range of the predicted value) as shown in figure 4.25

```
In [25]: prediction = m.predict(future_dates)

In [26]: prediction[['ds', 'yhat', 'yhat_lower', 'yhat_upper']].tail()
```

Out[26]:

	ds	yhat	yhat_lower	yhat_upper
218	2020-08-27	6.076801e+06	5.926653e+06	6.217820e+06
219	2020-08-28	6.138825e+06	5.979048e+06	6.288770e+06
220	2020-08-29	6.199320e+06	6.028994e+06	6.351499e+06
221	2020-08-30	6.257192e+06	6.084451e+06	6.412594e+06
222	2020-08-31	6.312363e+06	6.130841e+06	6.487958e+06

Figure 4.25: Future date predictions by Prophet

It can be seen that the prophet model predicted 6.31 Million COVID-19 cases by August end. Then MAE, MSE, RMSE were calculated for actual data and the predicted data as shown in figure 4.26

```
In [39]: # Mean Absolute Error
         mean_absolute_error(test_confirmed['y'], new_prid)

Out[39]: 18775.671385538484

In [40]: # Mean Squared Error
         mse = mean_squared_error(test_confirmed['y'], new_prid)
         mse

Out[40]: 466672322.0160147

In [41]: # Root Mean Squared Error
         rmse = sqrt(mse)
         rmse

Out[41]: 21602.59989019874
```

Figure 4.26: MAE, MSE, and RMSE for Prophet

At last, the predicted value and actual value were plotted using the Plotly library as shown in figure 4.27

Figure 4.27: Plotly Confirmed cases graph for Prophet

The same steps were repeated for the recovered cases prediction and the death cases prediction. The prediction results for the confirmed, recovered, and death cases for 31st August are shown in Tables 4.2,4.3,4.4 respectively.

Table 4.2: Confirmed cases prediction by Prophet

Predicted Confirmed	Predicted Confirmed High	Predicted Confirmed Low	MAE	MSE	RMSE
6.31M	6.45M	6.14M	18775.67	466672322.01	21602.59

Table 4.3: Recovered cases prediction by Prophet

Predicted Recovered	Predicted Recovered High	Predicted Recovered Low	MAE	MSE	RMSE
1.94M	2.01M	1.87M	107000.25	11969016677.82	109403

Table 4.4: Death cases prediction by Prophet

Predicted Deaths	Predicted Deaths High	Predicted Deaths Low	MAE	MSE	RMSE
165157	173800	156330	4610.33	22233152.23	4715.20

4.5.2 Predictions by Polynomial Regression Model

This prediction was performed by linear regression initially and polynomial regression was also implemented later on as it was observed that the linear regression model was unable to fit the data points due to under-fitting.

The prediction using both the algorithms for positive or confirmed cases is discussed in this sub-section. After reading the dataset and performing data transformation the X(independent variable) and Y(dependent variable) were assigned which were later split into a training set and a testing set with a 95%-5% ratio.

The variable X was assigned the index value of days as shown in figure 4.28

```
In [11]: # Scaling the X
         X = np.array(df['Days']).reshape(-1,1)
```

```
In [12]: X_train = X[:194]
         X_train
```

```
Out[12]: array([[  0],
                [  1],
                [  2],
                [  3],
                [  4],
                [  5],
                [  6],
                [  7],
                [  8],
                [  9],
                [ 10],
                [ 11],
                [ 12],
                [ 13],
                [ 14],
```

Figure 4.28: Train-Test split for the independent variable X

The variable Y was assigned positive cases field as it depends on the value of X_train. It is shown in figure 4.29

```
In [14]: # Scaling the Y
         Y_Confirmed = np.array(df['positive']).reshape(-1,1)

In [15]: Y_Confirmed

         ...

In [16]: Y_Confirmed_train = Y_Confirmed[:194]
         Y_Confirmed_train

         ...

In [17]: Y_Confirmed_Test = Y_Confirmed[194:]
         Y_Confirmed_Test

Out[17]: array([[4694126.],
                [4745694.],
                [4797959.],
                [4852143.],
                [4913663.],
                [4967754.],
                [5019073.],
                [5060880.],
                [5116474.],
                [5172509.]])
```

Figure 4.29: Train-Test split for the dependent variable Y

After this, the model was fit using Linear Regression object and a NumPy array variable **X_pred** consisting of future dates was also created as shown in figure 4.30

```
In [36]: lin_reg = LinearRegression(normalize= True)
         lin_reg.fit(X_train, Y_Confirmed_train)

Out[36]: LinearRegression(normalize=True)

In [37]: list1 = [204,205,206,207,208,209,210,211,212,213,214,215,216,217,218,219,220,221,222,223]

         list1_array = np.array(list1).reshape(-1,1)
         X_pred = (np.append(X,list1_array)).reshape(-1,1)
```

Figure 4.30: Linear Regression model fit

Then, the predict() function was used and the predicted value was plotted with the actual value as shown in figure 4.31

```
In [45]: plt.figure(figsize=(12,10))
         plt.plot(df['positive'], label = 'Confirmed Cases')
         plt.plot( predicted_value[:204], label = 'Predicted Confirmed Cases')

         plt.legend()
         plt.show()
```

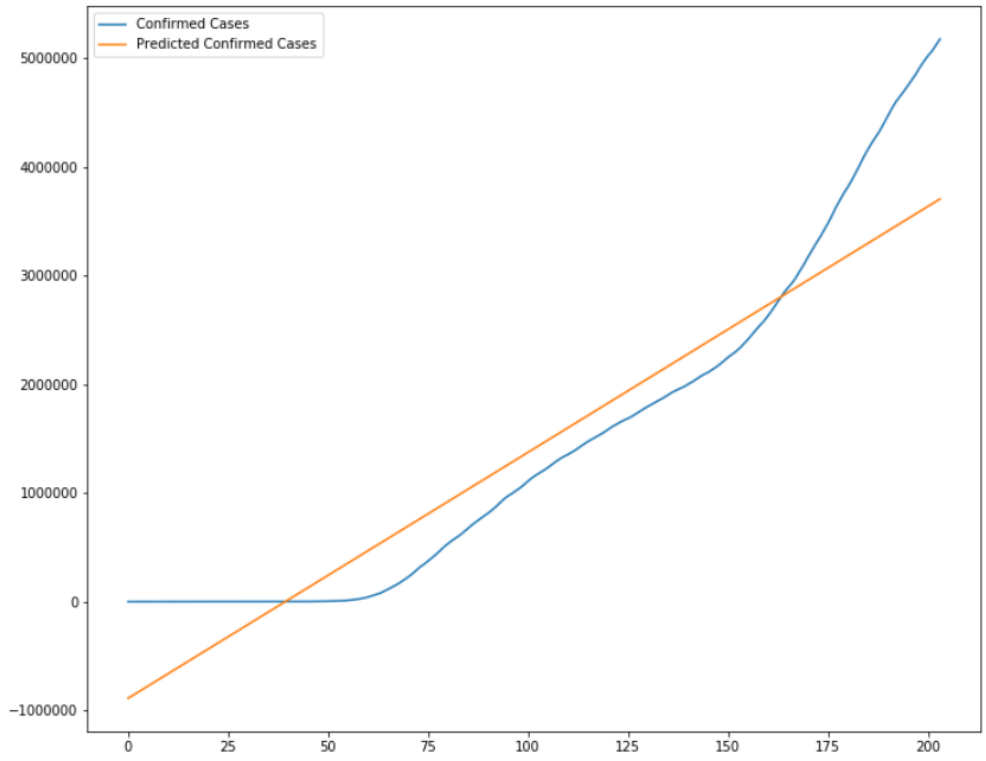

Figure 4.31: LR actual vs predicted graph

As can be observed from figure 4.31 that linear regression was unable to fit the data points due to under-fitting hence polynomial regression was required. In this, the variable X_train was transformed to a degree 3 cubic polynomial using fit_transform() function.

After transformation, it was later fit using linear regression as shown in figure 4.32

```
In [99]: # Polynomial Regression Starts
         poly = PolynomialFeatures(degree = 3)
         X_poly = poly.fit_transform(X_train)
         poly.fit(X_poly, Y_Confirmed_train)

In [100]: X_poly

In [101]:
         lin_reg2 = LinearRegression(normalize= True)
         lin_reg2.fit(X_poly, Y_Confirmed_train)
         pred_poly = poly.fit_transform(X_pred)
         poly_predicted_value_pred = lin_reg2.predict(pred_poly)
```

Figure 4.32: Polynomial Regression model fit

Figure 4.33 consists of MAE, MSE, and RMSE values for the PR model as shown.

```
In [103]: # Mean Absolute Error
          mean_absolute_error(Y_Confirmed_Test, poly_predicted_value_pred[194:204])
Out[103]: 210302.4888865863

In [106]: # Mean Squared Error
          mse_poly = mean_squared_error(Y_Confirmed_Test, poly_predicted_value_pred[194:204])
          mse_poly
Out[106]: 44302817495.90223

In [107]: # Root Mean Squared Error
          rmse_poly = sqrt(mse_poly)
          rmse_poly
Out[107]: 210482.34485557745
```

Figure 4.33: MAE, MSE, RMSE for PR

The graph between actual positive cases vs predicted positive cases is shown in figure 4.34

```
In [108]: #degree 3
plt.figure(figsize=(12,10))
plt.plot(df['positive'], label = 'Confirmed Cases')
plt.plot(poly_predicted_value_pred[:204], label = 'Predicted Confirmed Cases')

plt.legend()
plt.show()
```

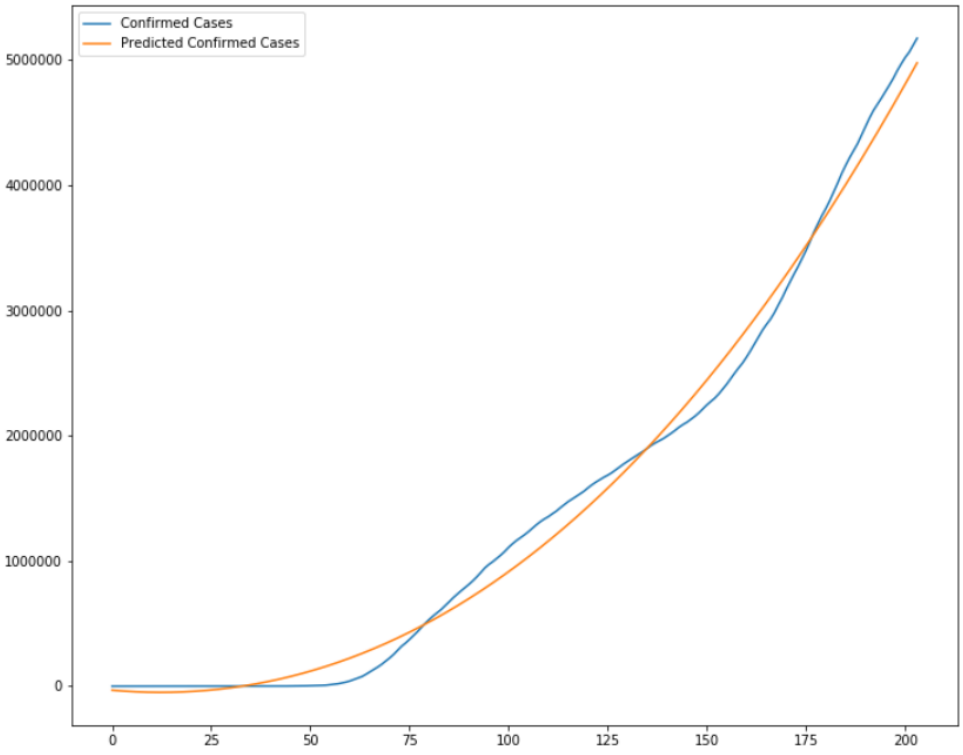

Figure 4.34: PR actual vs predicted graph

It can be observed that the polynomial regression model was fitting much better than LR. The prediction implementation steps were similar for recovered cases and death cases. The overall predictions result for positive, recovered, and death cases for 31st August 2020 are shown in tables 4.5,4.6,4.7 respectively.

Table 4.5: PR predicted confirmed cases

Predicted Confirmed	MAE	MSE	RMSE
6.19M	210302.48	44302817495.90	210482.34

Table 4.6: PR predicted recovered cases

Predicted Recovered	MAE	MSE	RMSE
2.16M	60028.35	3783847487.87	61512.98

Table 4.7: PR predicted death cases

Predicted Deaths	MAE	MSE	RMSE
204754	16349.90	268305449.94	16380.03

4.5.3 Predictions by ARIMA Model

In the ARIMA model, predicting the confirmed cases for 31st August, the date column was set as the index as shown in figure 4.35

```
In [27]: Confirmed_Cases = df[["date","positive"]]

In [28]: Confirmed_Cases.set_index('date', inplace=True)

In [29]: Confirmed_Cases.head()
```

Out[29]:

date	positive
2020-01-22	2.0
2020-01-23	2.0
2020-01-24	2.0
2020-01-25	2.0
2020-01-26	2.0

Figure 4.35: ARIMA confirmed cases

Then the dataset was split into a training set and testing set with a 95%-5% ratio as shown in figure 4.36.

```
In [148]: # 95% for train
          X_Confirmed = Confirmed_Cases.values[:194]

In [152]: X_Confirmed

In [160]: # 5% for test
          X_Test = Confirmed_Cases.values[194:]
          X_Test

Out[160]: array([[4694126.],
                 [4745694.],
                 [4797959.],
                 [4852143.],
                 [4913663.],
                 [4967754.],
                 [5019073.],
                 [5060880.],
                 [5116474.],
                 [5172509.]])
```

Figure 4.36: ARIMA train-test split

In the next step, a list was created that has all the combinations of p, d, and q pair values from 0 to 9 using the package **itertools** as shown in figure 4.37

```
In [50]: p=d=q=range(0,10)
         pdq = list(itertools.product(p,d,q))
         pdq
```

Figure 4.37: p,d,q combinations function

ARIMA model needs to have the least AIC(Akaike Information Criterion) value for the selected values of p,d, and q. As lower the AIC value, the higher will be the accuracy of the ARIMA model. Hence, the for-each loop was used for calculating AIC of each pair of p,d, and q to determine the least AIC value as shown in figure 4.38

```
In [153]: warnings.filterwarnings('ignore')
          for param in pdq:
              try:
                  model_arima = ARIMA(X_Confirmed,order=param)
                  model_arima_fit = model_arima.fit()
                  print(param,model_arima_fit.aic)
              except:
                  continue

(0, 0, 0) 6027.8052142671
(0, 0, 1) 5769.755460282155
(0, 0, 2) 5516.704919139463
(0, 0, 3) 5276.6090975572915
(0, 0, 4) 5060.292169042836
```

Figure 4.38: ARIMA AIC calculator function

The lowest AIC value observed was 3549.40 for the p,d,q combination (7, 2, 1). This combination was used in the next step for fitting the ARIMA model as shown in figure 4.39

```
In [167]: model_arima = ARIMA(X_Confirmed,order=(7, 2, 1))
          model_arima_fit = model_arima.fit()
          print(model_arima_fit.aic)

3549.4077013677133
```

Figure 4.39: ARIMA model fit

Then, the forecast() function was used to predicts the confirmed cases until 31st August as shown in figure 4.40

```
In [168]: prediction_confirmed = model_arima_fit.forecast(steps=29)[0]
          prediction_confirmed

Out[168]: array([4690581.00374555, 4740050.41337669, 4797791.19591519,
                 4859601.91738096, 4919542.44590526, 4971610.05786892,
                 5014244.86964386, 5054552.51451188, 5098964.17457962,
                 5150855.41433182, 5207011.01476973, 5261066.2931884 ,
                 5307513.8891167 , 5346136.61583642, 5382838.71797082,
                 5423871.77423255, 5471947.54172848, 5524015.21596768,
                 5573839.5668469 , 5616562.38015351, 5652527.517992  ,
                 5687071.6121232 , 5726033.1651951 , 5771638.09994239,
                 5820828.05362157, 5867697.31384401, 5907984.86397422,
                 5942342.47971672, 5975758.21583963])
```

Figure 4.40: ARIMA confirmed predicted cases

The graph was plotted between test data and predicted values as shown in figure 4.41

```
In [170]: plt.plot(X_Test)
          plt.plot(prediction_confirmed[:10],color='red')

Out[170]: [<matplotlib.lines.Line2D at 0x262bcee7b8>]
```

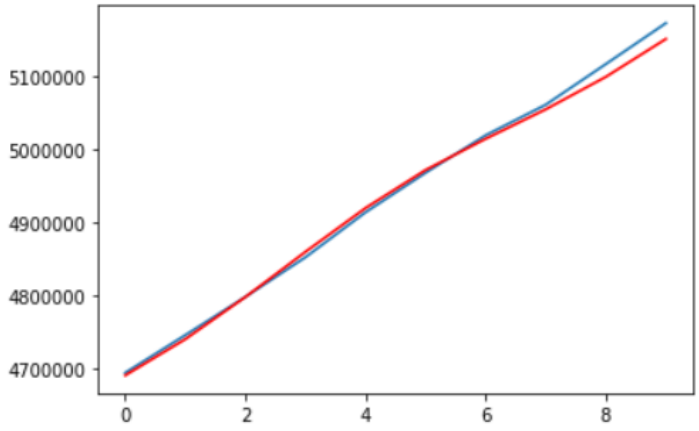

Figure 4.41: ARIMA actual vs predicted plot

For determining the accuracy of the ARIMA model various parameters such as MAE, MSE, and RMSE were calculated as shown in figure 4.42

```
In [172]: # MAE Confirmed Cases
          mean_absolute_error(X_Test, prediction_confirmed[:10])

Out[172]: 7686.983505055122

In [188]: # MSE Confirmed Cases
          mse_confirmed = mean_squared_error(X_Test, prediction_confirmed[:10])
          mse_confirmed

Out[188]: 98833741.54470842

In [189]: # RMSE Confirmed Cases
          rmse_confirmed = sqrt(mse_confirmed)
          rmse_confirmed

Out[189]: 9941.516058665722
```

Figure 4.42: MAE, MSE, RMSE for ARIMA

Similar steps were followed for predicting the recovered cases and the death cases. The lowest AIC value for recovered case prediction was 3998.66 with p,q, and q being (2, 2, 9) and for death case prediction the lowest AIC was 2567.84 with (7, 2, 4) as p,d, q values. The overall prediction results for confirmed cases, recovered cases, and death cases for 31st August using the ARIMA model are shown in tables 4.8,4.9, and 4.10 respectively.

Table 4.8: ARIMA predicted confirmed cases

Predicted Confirmed	**MAE**	**MSE**	**RMSE**
5.97M	7686.98	98833741.54	9941.51

Table 4.9: ARIMA predicted recovered cases

Predicted Recovered	MAE	MSE	RMSE
2.09M	42173.84	2096041581.21	45782.54

Table 4.10: ARIMA predicted death cases

Predicted Deaths	MAE	MSE	RMSE
178081	307.48	108587.32	329.52

4.6 Version Control and dataset source details

Github was used for version control. The complete source code for the implementation can be found at below URL:

https://github.com/mahas500/COVID-19-prediction-for-U.S.A

The dataset was taken from Kaggle. The source can be found at below URL:

https://www.kaggle.com/sudalairajkumar/covid19-in-usa

(Please note that this dataset gets updated every 10 days on Kaggle. However, as part of this dissertation data from 22nd January 2020 – 12th, August 2020 was used).

CHAPTER 5: EVALUATION

In this study, the predictions for COVID-19 confirmed, recovered, and death cases were done until 31st August using the Prophet model, ARIMA model, and Polynomial Regression. The model accuracy was based on MAE, MSE, and RMSE values. These values were computed by comparing the actual 5% test data with the prediction data for date range 03rd August 2020 – 12th August 2020. The results are shown in Table 5.1

Table 5.1: Three Models Evaluation

	MAE	MSE	RMSE
Prophet Confirmed	18775.67	466672322.01	21602.59
Prophet Recovered	107000.25	11969016677.82	109403.00
Prophet Deaths	4610.33	22233152.23	4715.20
ARIMA Confirmed	7686.98	98833741.54	9941.51
ARIMA Recovered	42173.84	2096041581.21	45782.54
ARIMA Deaths	307.48	108587.32	329.52
PR Confirmed	210302.48	44302817495.90	210482.34
PR Recovered	60028.35	3783847487.87	61512.98
PR Deaths	16349.90	268305449.94	16380.03

It can be observed from Table 5.1 that the ARIMA model gave the best results as compared to the Prophet model and PR model for all three(confirmed, recovered, death)types of predictions for all metrics. After ARIMA, the Prophet model gave better results for confirmed cases and death cases prediction as compared to the PR model. However, the PR model performed better in predicting the recovered cases than the Prophet model. Overall, it can be concluded that the time-series algorithms such as ARIMA, Prophet are better than Linear Regression and Polynomial Regression for predicting the COVID-19 cases as the dataset also has the data in time-series trend format. The COVID-19 predictions from 25th August 2020 – 07th September 2020 using the ARIMA model is shown in Table 5.2

Table 5.2: Covid-19 cases prediction using ARIMA

Date	Confirmed Cases	Recovered Cases	Death Cases	Active Cases
25-08-2020	5726033	1958161	171549	3596323
26-08-2020	5771638	1980990	172709	3617939
27-08-2020	5820828	2003354	173898	3643576
28-08-2020	5867697	2026224	175164	3666309
29-08-2020	5907984	2049215	176286	3682483
30-08-2020	5942342	2071999	177163	3693180
31-08-2020	5975758	2095358	178081	3702319
01-09-2020	6013601	2118519	179187	3715895
02-09-2020	6057673	2141843	180341	3735849
03-09-2020	6104902	2165519	181537	3757846
04-09-2020	6149768	2188976	182815	3777977
05-09-2020	6188537	2212822	183972	3791743
06-09-2020	6222077	2236732	184922	3800423
07-09-2020	6255100	2260614	185901	3808585

In table 5.2 it can be observed that COVID-19 confirmed cases will go up to 6.01 Million by 1st September which confirms the hypothesis of the thesis which states that "**Coronavirus confirmed cases will reach up to 6 Million by 1st September 2020**".

As per the research question 1 "**Which is the best algorithm among Prophet, ARIMA, Polynomial Regression, and Linear Regression for COVID-19 predictions**." It can be concluded that ARIMA is the best prediction model. As per research question 2 "**What will be the total number of COVID 19 positive cases, death cases, and recovered cases in the USA by August end based on the current data**". It can be noted that there will be 5.97 Million COVID-19 confirmed cases, 2.09 Million recovered cases, and 178K death cases by 31st August. The active cases are also calculated by the formula

Active cases = Confirmed Cases – (Recovered Cases + Death Cases)

The active cases seem to be increasing daily as per the data which answer the third research question "**According to the future COVID-19 predictions is there an increase in active cases**". This is a concern for the government.

CHAPTER 6: CONCLUSION AND FUTURE WORK

To conclude, this thesis followed the CRISP-DM methodology for analysis, visualization, and prediction of COVID-19 cases in the USA. In the implementation, EDA(Exploratory Data Analysis) was used for data analysis. Data cleaning was done using Pandas and Numpy. Different types of graphs were plot for a better understanding of the existing data using the Seaborn and Matplotlib libraries. Tableau was used for data visualization in which a time-series COVID-19 dashboard was created. The dashboard had 4 visualization sheets with different designs and it shows the increasing trends of COVID-19 in the USA based on different parameters.

For the prediction of COVID-19 cases ARIMA, Prophet, Polynomial Regression models were used. Best prediction results were given by the ARIMA model. It predicted approximately 6.01 Million confirmed cases by 1st September. Also, future predictions show that the rate of active cases will increase each day which is a concern for the U.S government.

However, there is a limitation of this study too as the ARIMA, Prophet, and PR models are not good options for long term predictions. The prediction values about the COVID-19 cases in the USA can also be calculated using these models for the next 6-8 months. But the predicted results will be very different from the actual results in the future due to many factors such as lockdown, travel restrictions, vaccine development, etc.

For future work, predictions can also be performed using deep learning algorithms such as MLP(Multilayer Perceptron), Recurrent Neural Network's LSTM(Long Short Term Memory), CNN(Convolutional Neural Network. Deep learning algorithms are better for long-term predictions and can also predict accurately with limited data as oppose to traditional time-series

algorithms such as ARIMA and Prophet which requires complete data for accurate predictions (Deep Learning for Time Series and why DEEP LEARNING? 2020). A website can also be created as part of future work for better information tracking of COVID-19 cases. It can also be deployed on a cloud platform for public sharing.

BIBLIOGRAPHY

Dey, S., Rahman, M., Siddiqi, U., and Howlader, A., 2020. *Analyzing the epidemiological outbreak of COVID-19: A visual exploratory data analysis approach*. *Journal of Medical Virology*, 92(6), pp.632-638.

PubMed. 2020. *[The Epidemiological Characteristics Of An Outbreak Of 2019 Novel Coronavirus Diseases (COVID-19) In China]*. [online] Available at: <https://pubmed.ncbi.nlm.nih.gov/32064853/> [Accessed 14 July 2020].

Ruan, Q., Yang, K., Wang, W., Jiang, L., and Song, J., 2020. *Clinical Predictors Of Mortality Due To COVID-19 Based On An Analysis Of Data Of 150 Patients From Wuhan, China*. 1st ed. [ebook] Springer. Available at: <https://link.springer.com/content/pdf/10.1007/s00134-020-05991-x.pdf>

[Accessed 14 July 2020].

Pandey, G., Chaudhary, P., Gupta, R., and Pal, S., 2020. *SEIR And Regression Model Based COVID-19 Outbreak Predictions In India*. 1st ed. [ebook] New Delhi: Cornell University. Available at: https://arxiv.org/ftp/arxiv/papers/2004/2004.00958.pdf

[Accessed 16 July 2020].

Chatterjee, J., and Hassanien, E., 2020. *A Machine Learning Forecasting Model For COVID-19 Pandemic In India*. 1st ed. [ebook] Springer. Available at: <https://link.springer.com/content/pdf/10.1007/s00477-020-01827-8.pdf>

[Accessed 20 July 2020].

Han Lau, C., Nazri, H., Vincent Ligot, D., Lee, G., and Liang Tan, C., 2020. *Coronatracker: World-Wide COVID-19 Outbreak Data Analysis And Prediction.* 1st ed. [ebook] Available at:<https://cdn.spotle.ai/projects/296083/10079/20_255695.pdf>

[Accessed 20 July 2020].

Jia, L., Li, K., Jiang, Y., Guo, X. and Zhao, T., 2020. *Prediction And Analysis Of Coronavirus Disease 2019*. [online] arXiv.org. Available at: <https://arxiv.org/abs/2003.05447>

[Accessed 20 July 2020].

Ranjan, R., 2020. *PREDICTIONS FOR COVID-19 OUTBREAK IN INDIA USING EPIDEMIOLOGICAL MODELS*. 1st ed. [ebook] medRxiv. Available at: <https://www.medrxiv.org/content/10.1101/2020.04.02.20051466v1.full.pdf>

[Accessed 22 July 2020].

Gupta, R., K. Pal, S., and Pandey, G., 2020. *A Comprehensive Analysis Of COVID-19 Outbreak Situation In India*. 1st ed. [ebook] New Delhi: medRxiv. Available at: <https://www.medrxiv.org/content/medrxiv/early/2020/04/16/2020.04.08.20058347.full.pdf>

[Accessed 21 July 2020].

Sayeed, M., and Ayesha, N., 2020. *Sayeed, M. And Ayesha, N., 2020. Tracking The Spread Of COVID-19 Cases In India Using Data Visualizing And Forecasting Techniques.* Available at: <https://www.researchgate.net/profile/Azam_Sayeed/publication/341297682_Tracking_the_Spread_of_COVID-19_Cases_in_India_using_Data_Visualizing_and_Forecasting_Techniques/links/5eb9add092851cd50dab3b40/Tracking-the-Spread-of-COVID-19-Cases-in-India-using-Data-Visualizing-and-Forecasting-Techniques.pdf>

[Accessed 21 July 2020].

Bhatnagar, M., 2020. *COVID-19: Mathematical Modeling And Predictions*. 1st ed. [ebook] New Delhi: IIT Delhi. Available at: <https://web.iitd.ac.in/~manav/COVID.pdf>

[Accessed 22 July 2020].

N Roy, A., Jose, J., Gautam, N., Nathalia, D. and Suresh, A., 2020. *Prediction And Spread Visualization Of Covid-19 Pandemic Using Machine Learning*. 1st ed. [ebook] Noida, Uttar Pradesh: preprints. Available at: <https://www.preprints.org/manuscript/202005.0147/v1>

[Accessed 22 July 2020].

Kucharski, A., Russell, T., Diamond, C., Liu, Y., Edmunds, J., Funk, S., and Eggo, R., 2020. *Early Dynamics Of Transmission And Control Of COVID-19: A Mathematical Modelling Study*. 1st ed. [ebook] London, UK: ScienceDirect, pp.553-558. Available at: <https://www.sciencedirect.com/science/article/pii/S1473309920301444>

[Accessed 28 July 2020].

Chen, B., Shi, M., Ni, X., Ruan, L., Jiang, H., Yao, H., Wang, M., Song, Z., Zhou, Q. and Ge, T., 2020. *Visual Data Analysis And Simulation Prediction For COVID-19*. [online] Arxiv.org. Available at: <https://arxiv.org/pdf/2002.07096.pdf>

[Accessed 28 July 2020].

Gupta, R., and K Pal, S., 2020. *Trend Analysis And Forecasting Of COVID-19 Outbreak In India*. 1st ed. [ebook] Delhi, India: medrxiv. Available at: <https://www.medrxiv.org/content/10.1101/2020.03.26.20044511v1.full.pdf>

[Accessed 28 July 2020].

Batista, M., 2020. *Estimation Of The Final Size Of The Coronavirus Epidemic By The SIR Model*. 1st ed. [ebook] Slovenia: Research Gate. Available at: <https://www.researchgate.net/profile/Milan_Batista/publication/339311383_Estimation_of_the_final_size_of_the_coronavirus_epidemic_by_the_SIR_model/links/5e767fca4585157b9a512f80/Estimation-of-the-final-size-of-the-coronavirus-epidemic-by-the-SIR-model.pdf>

[Accessed 28 July 2020].

T Wu, J., Leung, K., and M Leung, G., 2020. *Nowcasting And Forecasting The Potential Domestic And International Spread Of The 2019-Ncov Outbreak Originating In Wuhan, China: A Modelling Study*. 1st ed. [ebook] Hong Kong, China: Science Direct, pp.689-697. Available at: <https://reader.elsevier.com/reader/sd/pii/S0140673620302609?token=20071F31E3C264D257186639D83FDAD0F93F467382C513E6E279F73C26BF12F3620FEEBDE984A74DC3001E4DC6E12774>

[Accessed 29 July 2020].

Fanelli, D., and Piazza, F., 2020. *Analysis And Forecast Of COVID-19 Spreading In China, Italy, And France*. 1st ed. [ebook] France: Science Direct. Available at: <https://reader.elsevier.com/reader/sd/pii/S0960077920301636?token=46208C0C66DD765590F7922ED0C65B0E5C7C40327ACE27838618C620A60CF8E89D6F22432613EE46982FF745E4F8C8FC>

[Accessed 29 July 2020].

Tomar, A., and Gupta, N., 2020. *Prediction For The Spread Of COVID-19 In India And Effectiveness Of Preventive Measures*. 1st ed. [ebook] Srinagar, J & K, India: ScienceDirect. Available at: <https://reader.elsevier.com/reader/sd/pii/S0048969720322798?token=A83DD8A0847590D3F6A667C1848B34D913E3B2129A04DE1554B41558E6C43C2E265F0CE5B6A4B71C37E4E0B8291D9F0E> [Accessed 29 July 2020].

Tang, T., Cao, L., Lan, C., and Cao, L., 2020. *SIR Model For Novel Coronavirus-Infected Transmission Process And Its Application*. 1st ed. [ebook] Research Square. Available at: <https://europepmc.org/article/ppr/ppr122293>

[Accessed 30 July 2020].

Medium. 2020. *Data Science Project Management Methodologies*. [online] Available at: <https://medium.com/datadriveninvestor/data-science-project-management-methodologies-f6913c6b29eb>

[Accessed 30 July 2020].

Wirth, R., and Hipp, J., 2020. *CRISP-DM: Towards A Standard Process Model For Data Mining*. 1st ed. [ebook] Germany. Available at: <http://www.cs.unibo.it/~danilo.montesi/CBD/Beatriz/10.1.1.198.5133.pdf>

[Accessed 31 July 2020].

Smart Vision Europe. 2020. *Crisp DM Methodology - Smart Vision Europe*. [online] Available at: <https://www.sv-europe.com/crisp-dm-methodology/>

[Accessed 31 July 2020].

Taylor, S., and Letham, B., 2020. *Forecasting At Scale*. 1st ed. [ebook] California, United States: PeerJ Preprints. Available at: <https://peerj.com/preprints/3190.pdf>

[Accessed 10 August 2020].

Medium. 2020. *Introduction To Linear Regression And Polynomial Regression*. [online] Available at: <https://towardsdatascience.com/introduction-to-linear-regression-and-polynomial-regression-f8adc96f31cb>

[Accessed 10 August 2020].

ListenData. 2020. *15 Types Of Regression In Data Science*. [online] Available at: <https://www.listendata.com/2018/03/regression-analysis.html>

[Accessed 10 August 2020].

Sean J. Taylor & Benjamin Letham (2018) *Forecasting at Scale*, The American Statistician, 72:1, 37-45, DOI: 10.1080/00031305.2017.1380080

Devi B, U., D, S., and P, A., 2020. *An Effective Time Series Analysis For Stock Trend Prediction Using ARIMA Model For Nifty Midcap-50*. 1st ed. [ebook] Research Gate. Available at: <https://www.researchgate.net/publication/284323226_An_Effective_Time_Series_Analysis_for_Stock_Trend_Prediction_Using_ARIMA_Model_for_Nifty_Midcap-50>

[Accessed 11 August 2020].

Medium. 2020. *Deep Learning For Time Series And Why DEEP LEARNING?*. [online] Available at: <https://towardsdatascience.com/deep-learning-for-time-series-and-why-deep-learning-a6120b147d60>

[Accessed 22 August 2020].