DATA ANALYSIS AND FORECASTING OF COVID-19 USING AI

Mini Project Report (DSE (301) : Applications of Artificial Intelligence)

Submitted By,

Piyush Paliwal (17180)

Deep Pooja (17074)

Under guidance of

Dr Parthiban Srinivasan,

Department of Data Science and Engineering,
IISER Bhopal

JUNE 20, 2020

IISER BHOPAL

Contents

Preface

Aim of this project is to perform data analysis of COVID-19 cases in India and in highly affected countries with COVID-19. Study the different forecasting models is also done to predict the no of COVID-19 cases for next few days. We have compared results of different forecasting models of deep learning and machine learning. This project is submitted for course evaluation purpose.

Problem Statement:

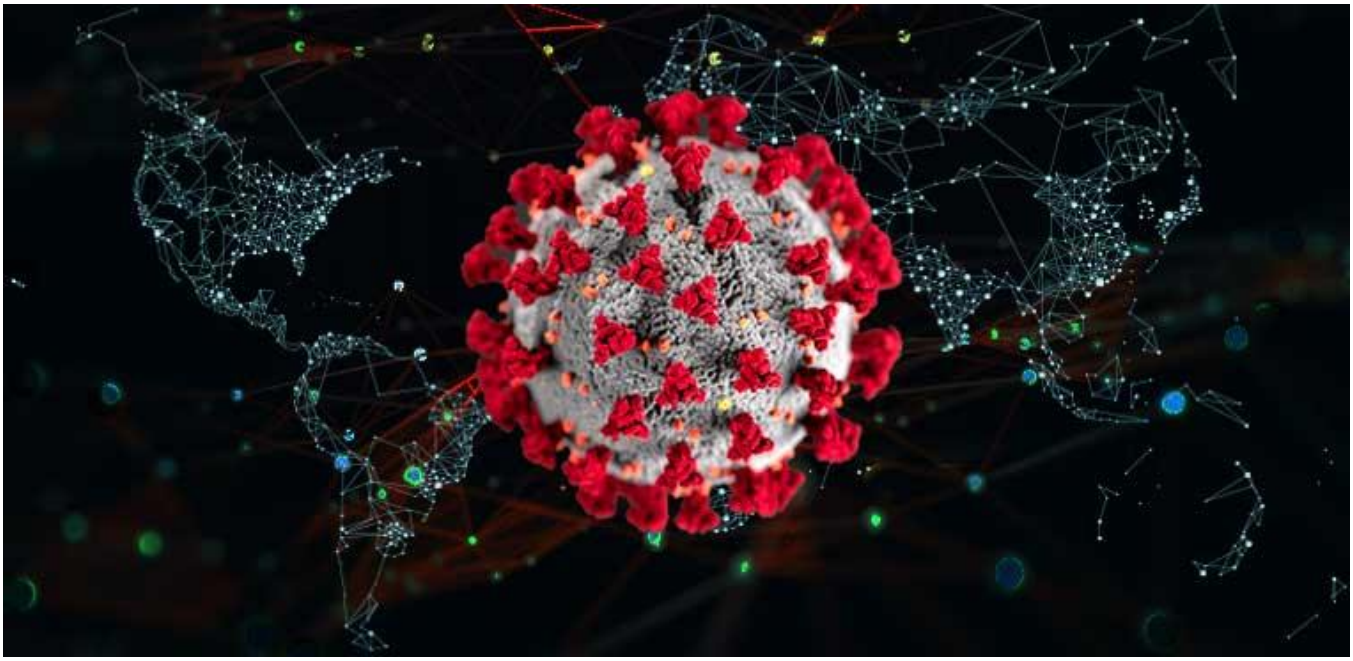
We have data of worldwide COVID-19 daily cases (from John Hopkins University). Our task is to forecast no. of COVID-19 cases for next week.

Our Strategy to approach the Problem:

1) First understand the mathematical epidemics model
2) Analyse the data worldwide as well for India
3) Use machine learning models to take data of rise in cases for last two months as input and train model on input data
4) Forecast the no of cases using trained models
5) Compare the results obtained from different model and check the results obtained with real after a week.
6) The summarizing results and conclusion on how we can do better.

Part: 1

Introduction



- What is COVID-19?

Coronavirus disease, code-named COVID-19, is an infectious disease caused by a virus, a member of the Beta coronavirus family named severe acute respiratory syndrome coronavirus 2 (SARS-CoV-2), previously referred to as 2019 novel coronavirus (2019-nCoV). It is thought that the virus outbreak has animal origins, and it was first transmitted to humans in Wuhan province China, in November/December 2019.

At present, no approved vaccines or specific antivirals are available for COVID-19. Previous SARS pandemic in 2002 and 2003 was controlled and finally stopped by conventional control measures, including travel restrictions and patient isolation. Currently, these measures are applied in almost all countries with the COVID-19 outbreak; however, their effectiveness depends on how rigorous they are. It follows that the methods enabling reliable prediction of spreading of COVID-19 would be of great benefit in persuading public

opinion why it is crucial to adhere to these measures in the past decade.

- How to perform forecasting of COVID-19?

According to data in real time, confirmed coronavirus disease 2019 (COVID-19) cases are growing exponentially in most countries around the world. Forecasting COVID-19 dissemination thus plays a key role. In the first place, to inform governments and healthcare professional what to expect and which measures to impose, and secondly, to motivate the wider public to adhere to the measures that were imposed to decelerate the spreading lest a regrettable scenario will unfold.

Modelling viral diseases such as COVID-19 is extremely important in determining their possible future impact. Modelling the spread and the effect of such a disease can be supremely important in understanding its impact. While traditional, statistical, modelling can offer precise models, artificial intelligence (AI) techniques could be the key to finding high-quality predictive models.

AI can in principle be used to track and to predict how the COVID-19 disease will spread over time and space. In fact, an AI-based model of HealthMap, at Boston Children's Hospital (USA), sounded one of the first alarms on 30 December 2019, around 30 minutes earlier than a scientist at the Program for Monitoring Emerging Diseases (PMED) issued an alert. The AI-inspired methods are a powerful tool for helping public health planning and policymaking.

- What will be the global impact of the novel coronavirus (COVID-19)?

Answering this question requires accurate forecasting the spread of confirmed cases as well as analysis of the number of deaths and

recoveries. Forecasting, however, requires ample historical data. At the same time, no prediction is certain as the future rarely repeats itself in the same way as the past. Moreover, forecasts are influenced by the reliability of the data, vested interests, and what variables are being predicted. Also, psychological factors play a significant role in how people perceive and react to the danger from the disease and the fear that it may affect them personally.

What is need of COVID-19 forecasting models?

The COVID-19 pandemic is a complex system involving biology, human behaviour, companies, and governments, and it's influenced by healthcare, economics, governance, and geopolitics concerning all nations. Sophisticated analytical methods could help improve economic, societal, and geopolitical stability. A massive amount of data about the COVID19 pandemic is generated every day. COVID19 is mutating faster than ever on account of its spread from one country to another country. So far, three types of COVID 19 strains have been identified depending on the fatality.

The spread of this pandemic depends upon several factors. For example, on a human to human interactions, population density at a particular geographic location, number of people following lockdown protocols, maintenance of hygiene at a particular location, health infrastructure at that location, numbers of active health workers and policemen, the prevalent weather conditions, medical history of patients (like people already suffering from diabetes or any other cardiac disease are more prone to infection and have a higher fatality rate), etc.

The framework for containing the outbreak depends on the following major categories:

1. Quarantine Efficiency
2. Government Management Efficiency
3. Monitoring and Detection
4. Emergency Treatment Readiness

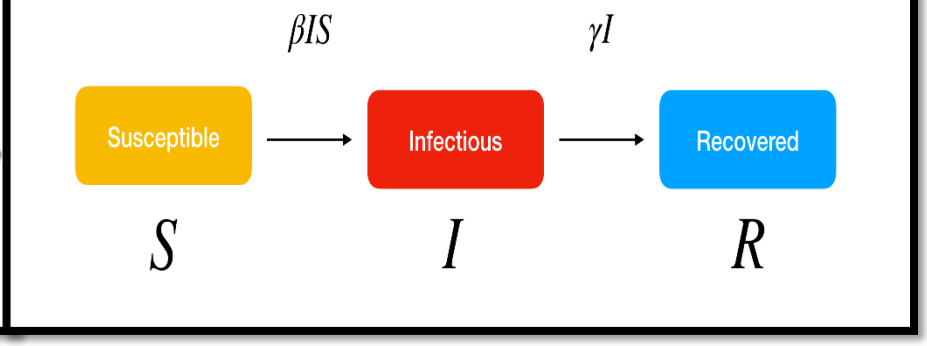
So, we have observed that all these parameters work together in order to stop or spread the pandemic and build better model. Our project is aimed to study all these parameters from data available to us in forecasting the COVID 19 situation at that location.

Understanding the Epidemic modelling: SEIR model

Predicting disease transmission on complex networks has attracted considerable recent attention in the epidemiology community. Infectious diseases spread over networks of contacts between susceptible and infectious individuals. Typical mathematical representation of an epidemic assumes that the host populations are fully mixed (mass-action approximation).

The SIR or SEIR (only difference is additional term E = exposed) model is a compartmental model which separates the population into several

$$\frac{dS}{dt} = -\frac{\beta IS}{N},$$
$$\frac{dI}{dt} = \frac{\beta IS}{N} - \gamma I,$$
$$\frac{dR}{dt} = \gamma I,$$



compartments, for example:

1.) **S**usceptible (can still be infected with considerable probability, but currently healthy)

2.) **E**xposed (mainly includes the Police personnel, the healthcare workers, lockdown offenders, and basic necessities shop vendors)

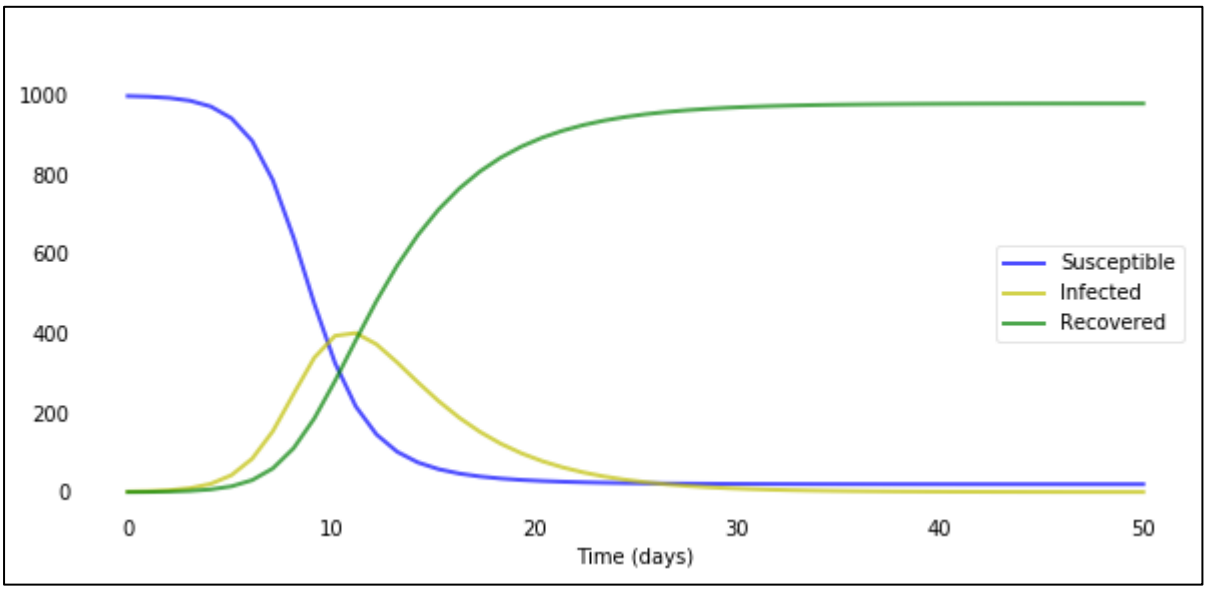
3.) **I**nfected (The COVID 19 patients as well as those quarantined by order)

4.) **R**ecovered (were already infected, cannot get infected again)

It helps in predicting number of cumulative cases and death rates for the upcoming 15 days. Taking all the factors into account this model

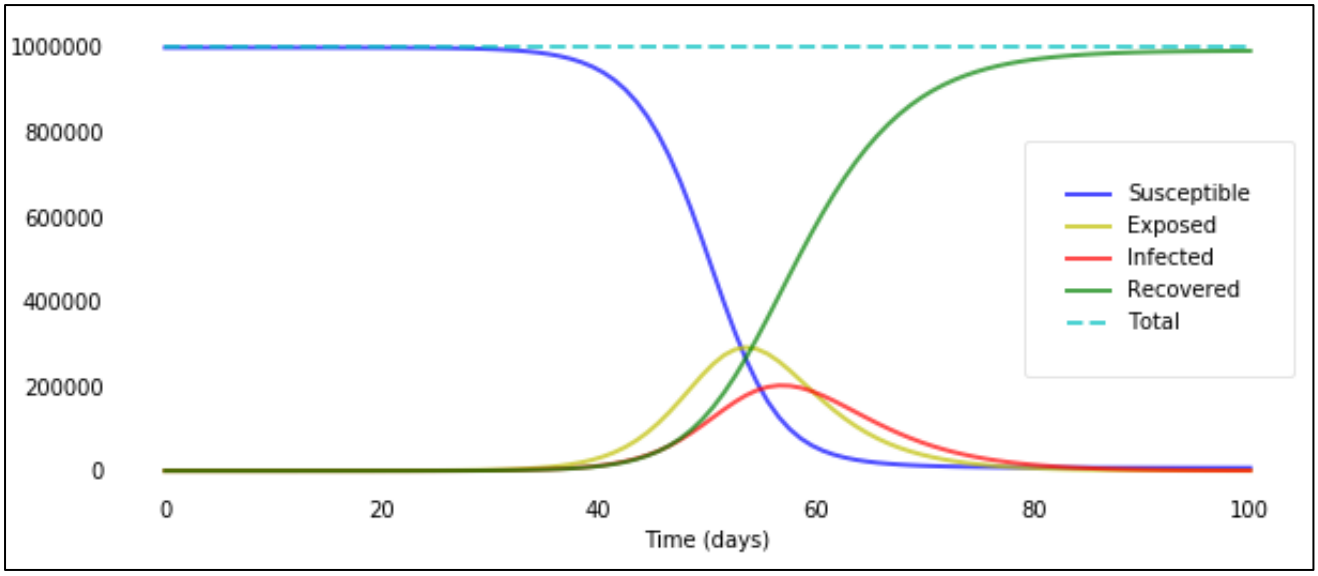
also helps us predicting the doubling rate given the current circumstances of the pandemic as well as the number of days with which the pandemic may end and no longer force a lockdown. This will be done by tracking the rate of increase in the number of cases vs recovery rate and death rate.

(all figures are obtained from codes on SIR model attached with drive)

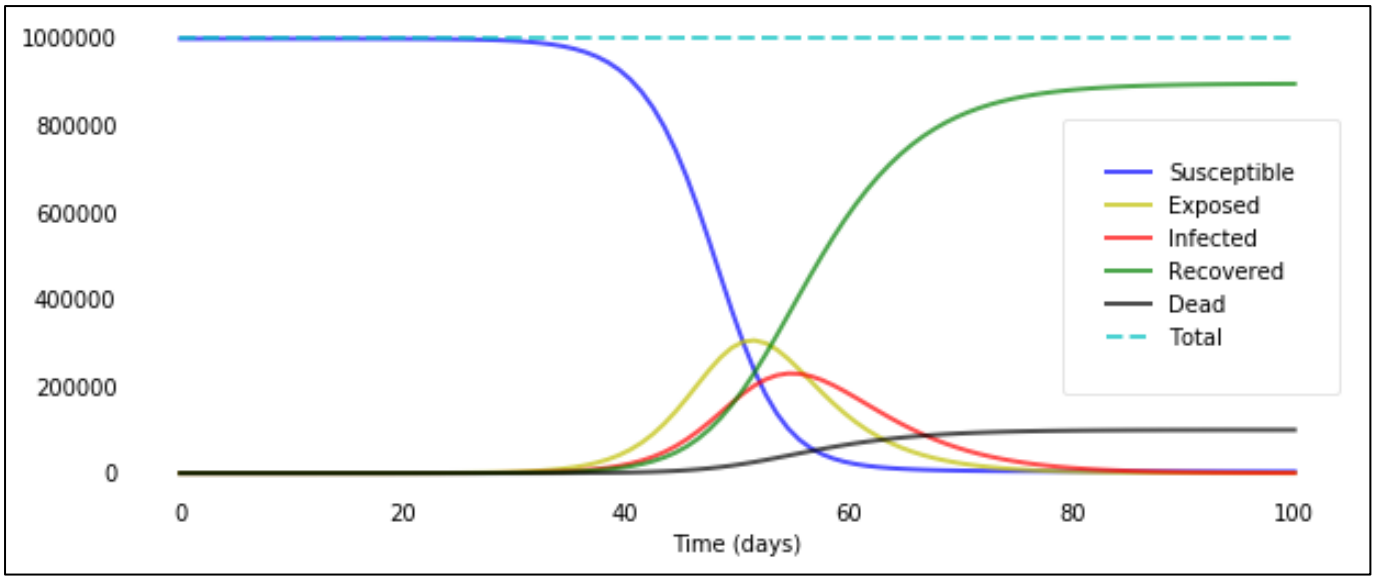


On plotting are three function, we get above plot. It shows that Susceptible people are getting infected and hence their no is decreasing and infected people no is increasing. No of death people is not considered. This is very simple epidemic model but in real case it is quite more complicated.

Let’s have a look at SEIR model:



1.) SEIR model (without death component)

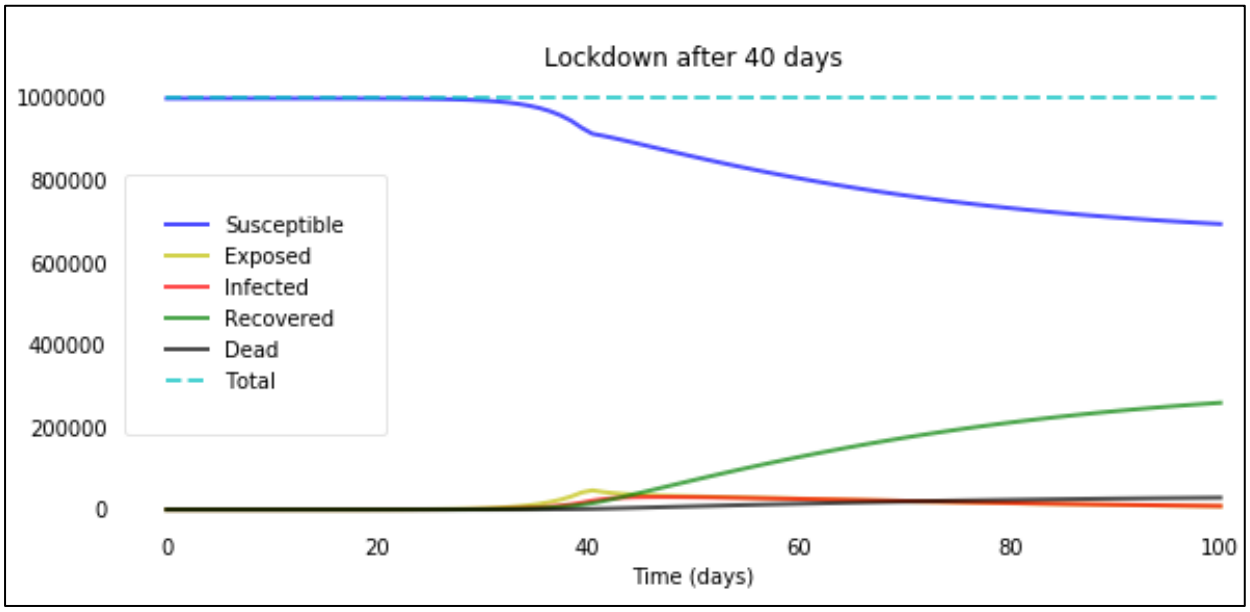


2.) SEIR model (with death component)

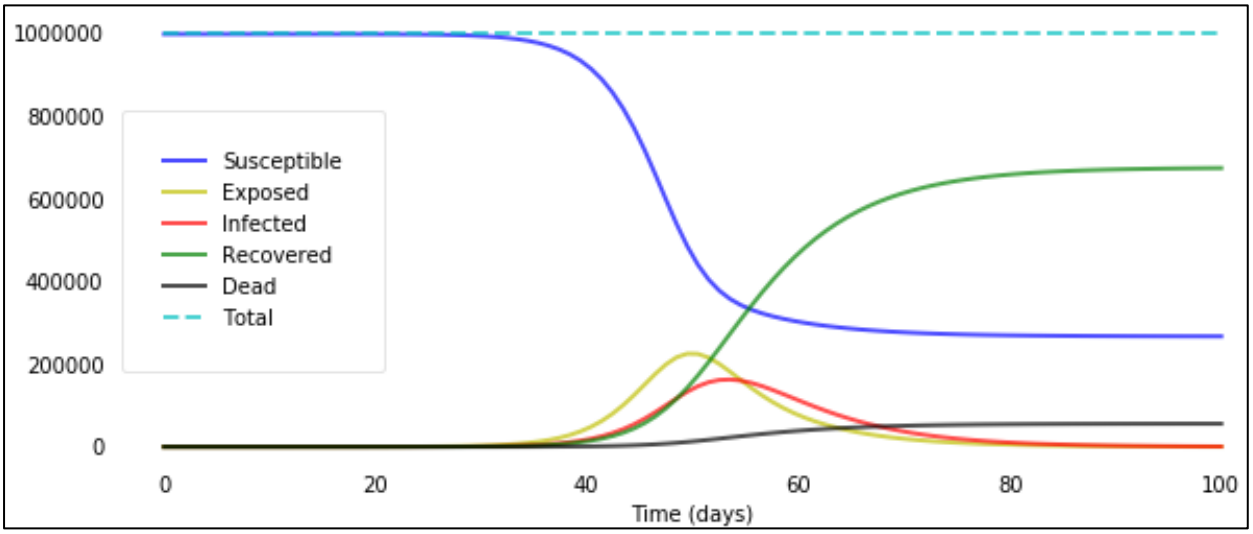
In SEIR model only additional term ‘Exposed’ is introduced.

Plot (1) corresponds to ideal case scenario where it is assumed that all people who have infected by decease can recover but it is not the case always. Those who has weak immune system can fight with decease. Plot (2) explains effect of dead people as well.

Let’s have a look at effects of lockdown on controlling the infection of decease. Here is a sample curve, plotted by reducing the rate of infection of susceptible people due to lock down.



SEIR plot during lockdown



Linear Regression approach in Pandemic modelling

Observation:

SEIR model are oversimplified model which helps in developing intuition about how pandemic spread data look like and it SEIR model is fitted onto real data then it may be used to forecast the end of pandemic or maximum people that can be affected in the pandemic.

Downside of SEIR model:

This model need lot of parameter tuning to find the best fit on the data. The only solution to this problem, is the take a model on which training and testing both possible. Such models are used in machine learning and deep learning. Before working on those models, let's have a look at analysis of COVID-19 cases in next part.

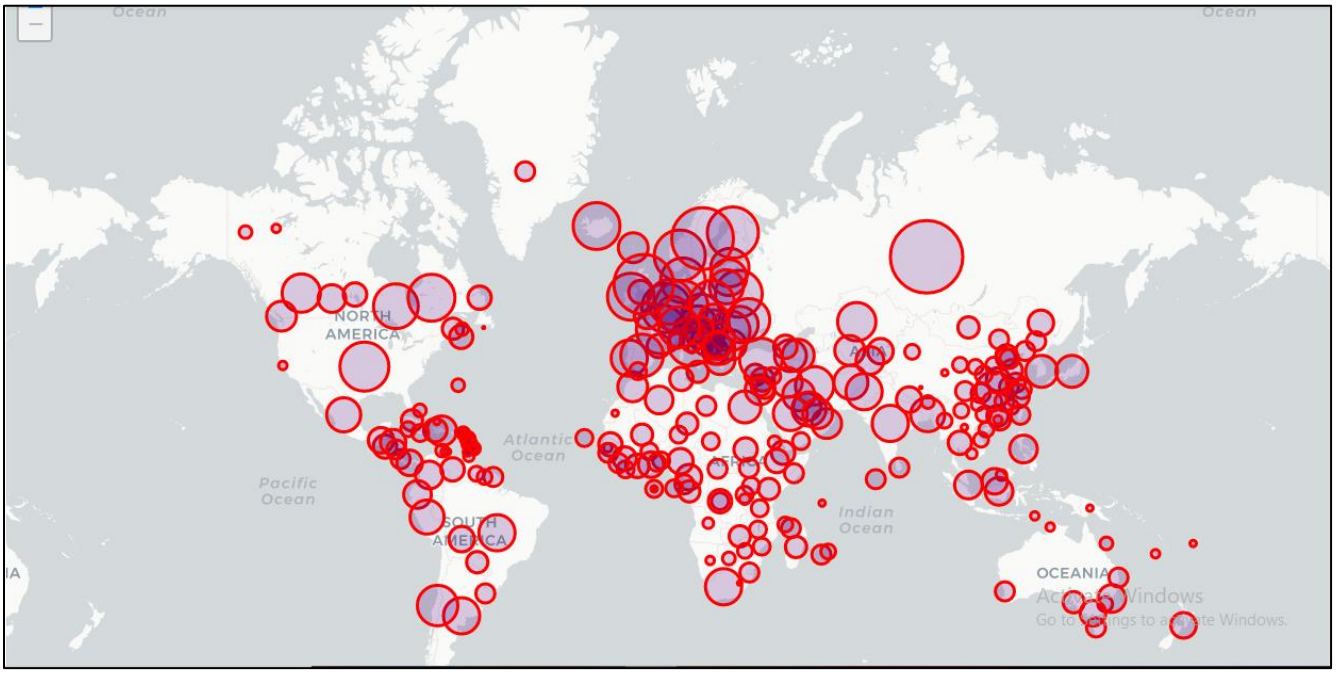
Part:2 - World Wide COVID-19 Analysis

In this section is focused on analysis of COVID-19 cases across the globe. We observe that for different countries the at which this pandemic is spreading is varying. At some part its spread is faster and some part its spread is slower.

Let's analyse the worldwide situation of this pandemic:

COVID-19 Interactive Analysis Dashboard

Details (codes and sources) of all generated figures/plots are in Jupiter Notebook



Dataset used for analysis: JHU CSSE COVID-19 Dataset

https://github.com/CSSEGISandData/COVID-19/tree/master/csse_covid_19_data

Current world scenario:

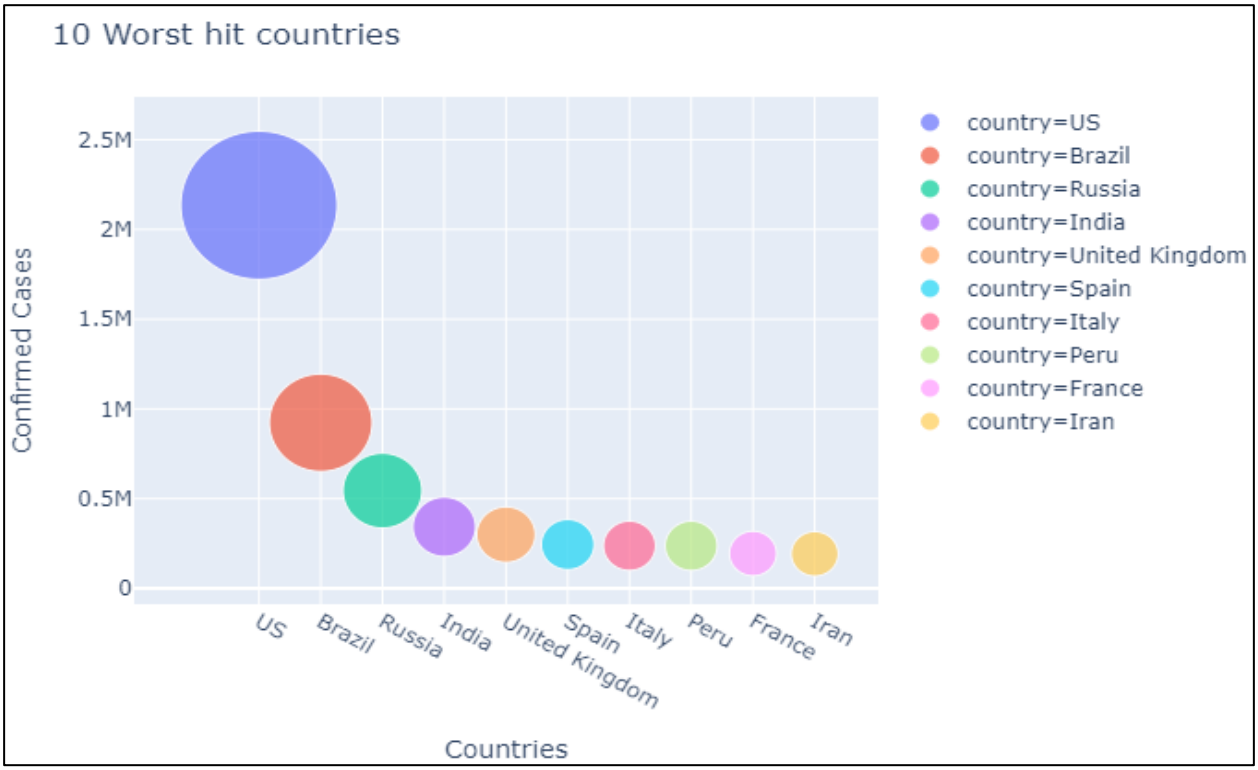
Confirmed: 8145047 Deaths: 440600 Recovered: 3939208

10 Worst hit countries

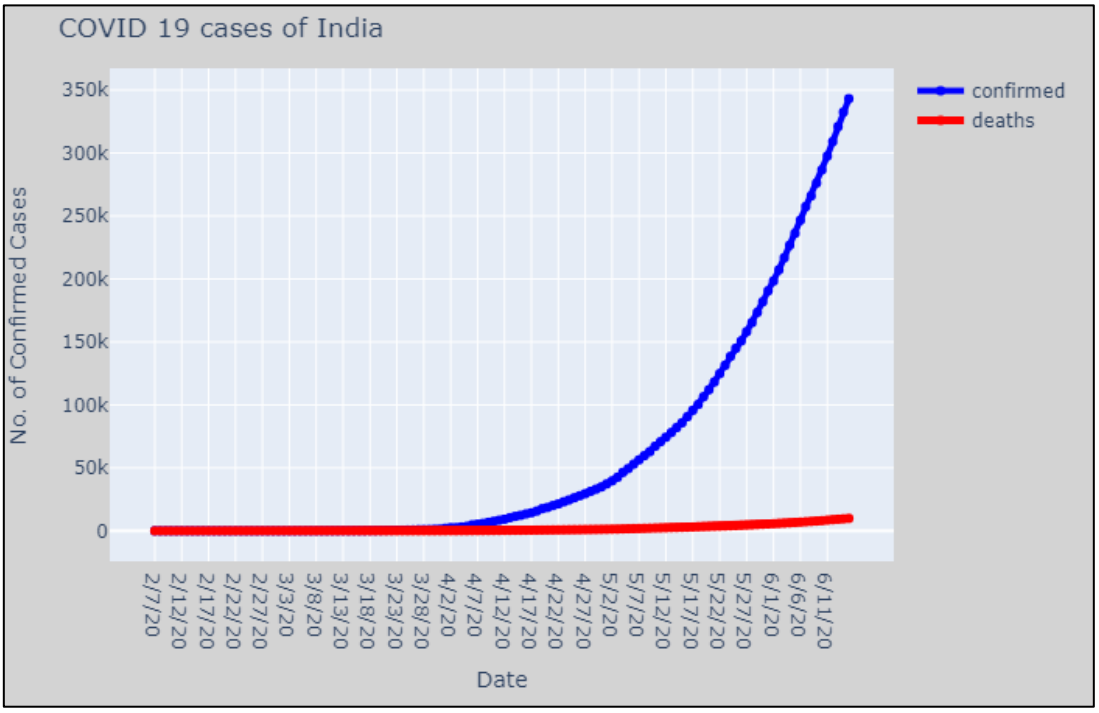
We have created bubble chart to visualize which countries are most affected by Covid19.

Covid19 Cases:

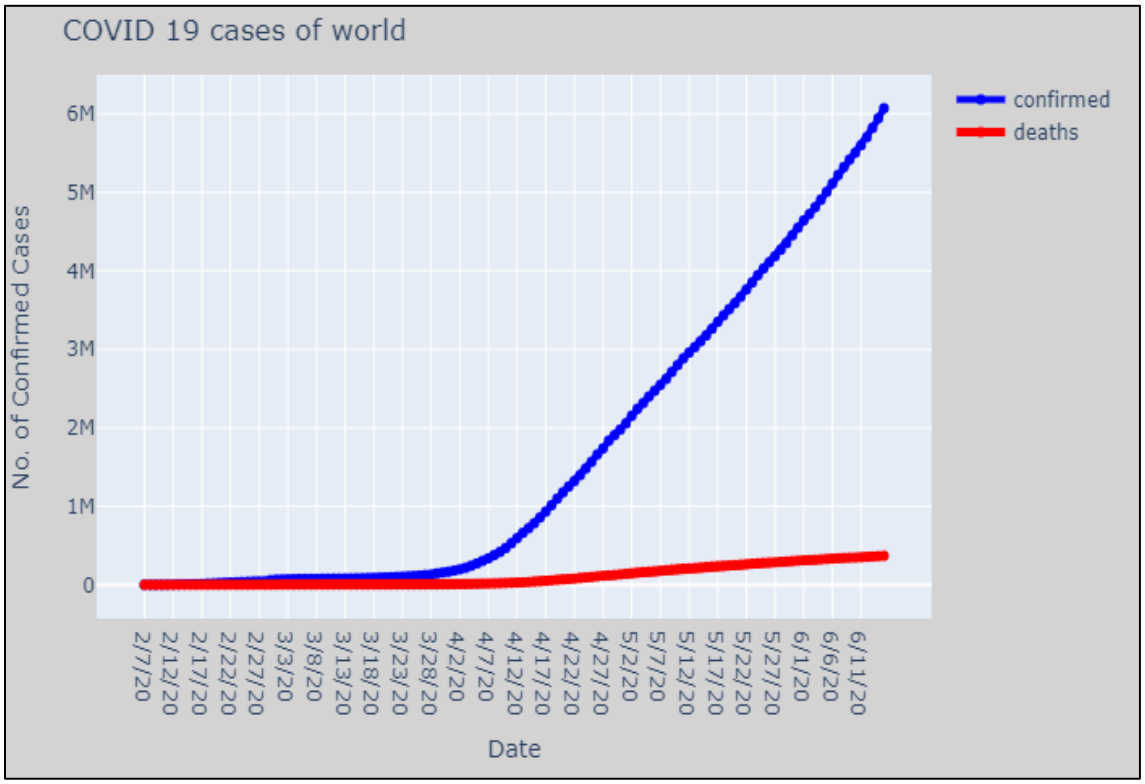
Let’s visualize how the number of cases is increasing in India and across the world.



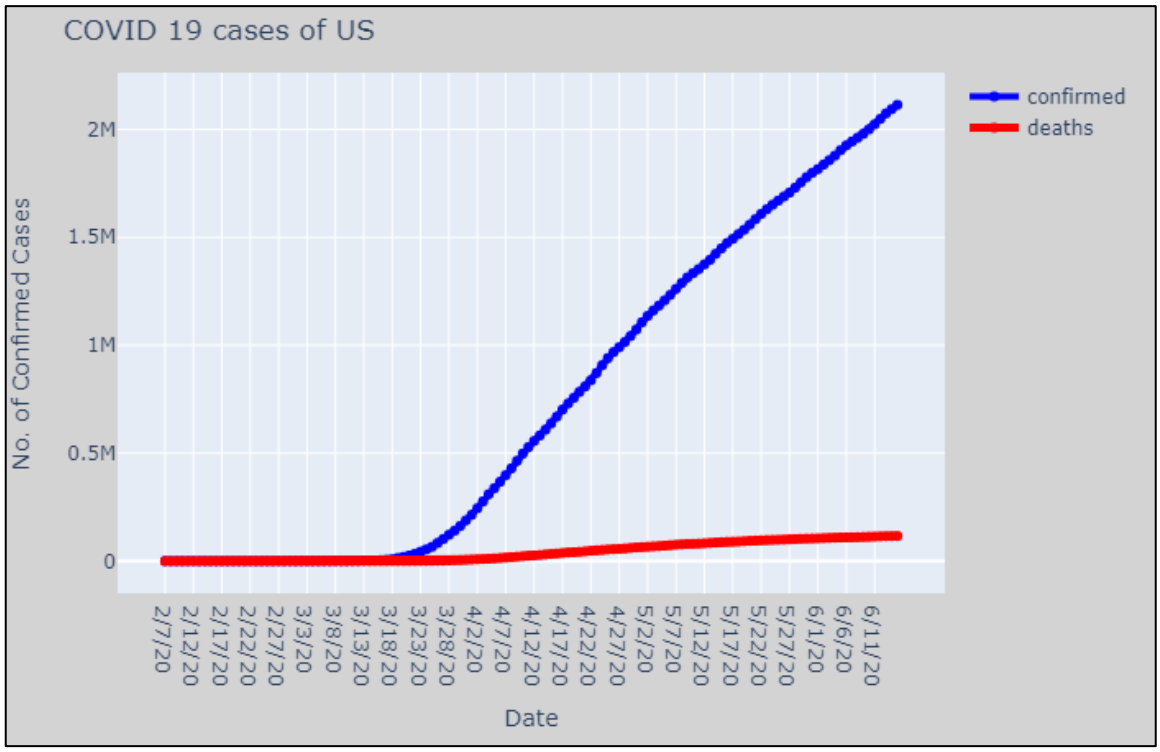
1. India



2. World

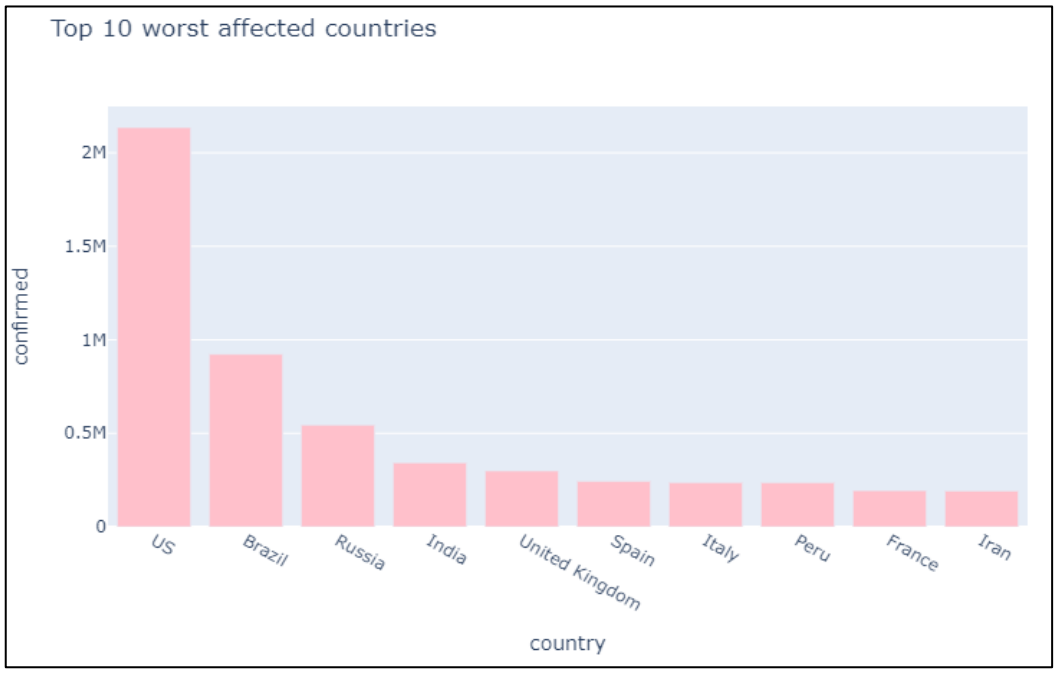


3. United States

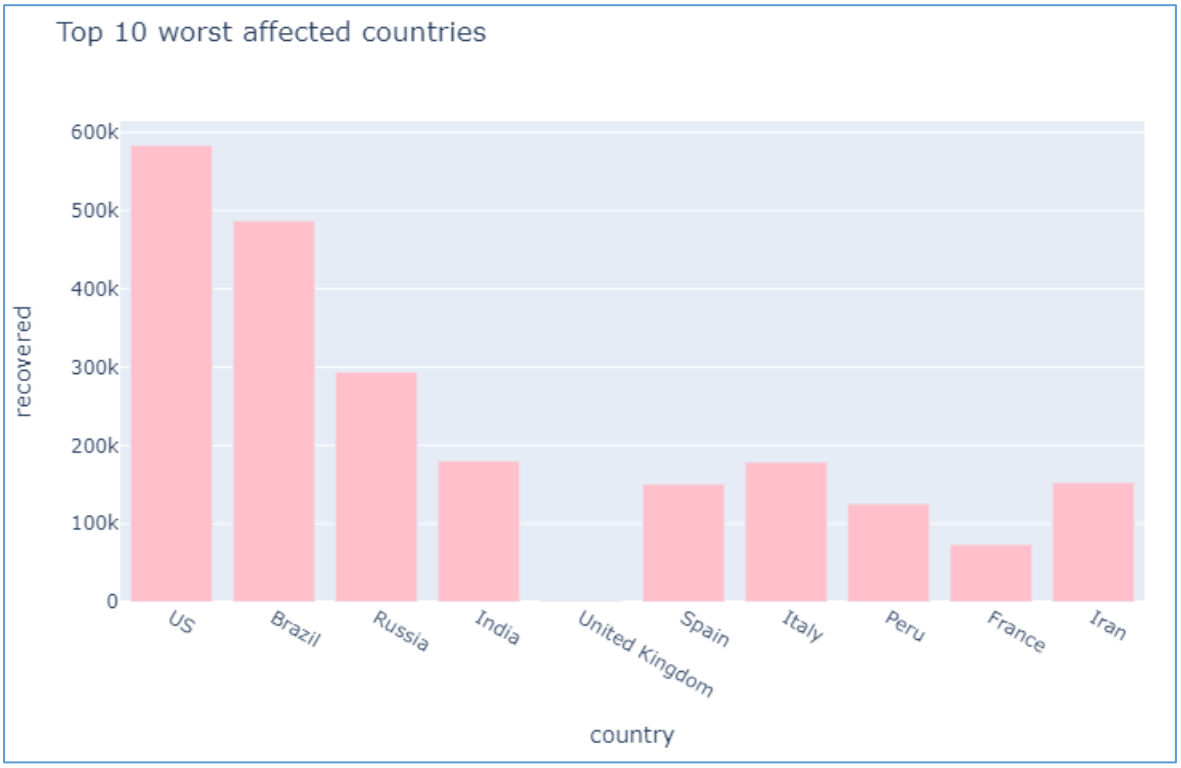


Histogram Plots of Confirmed cases, Deaths, Recovered cases

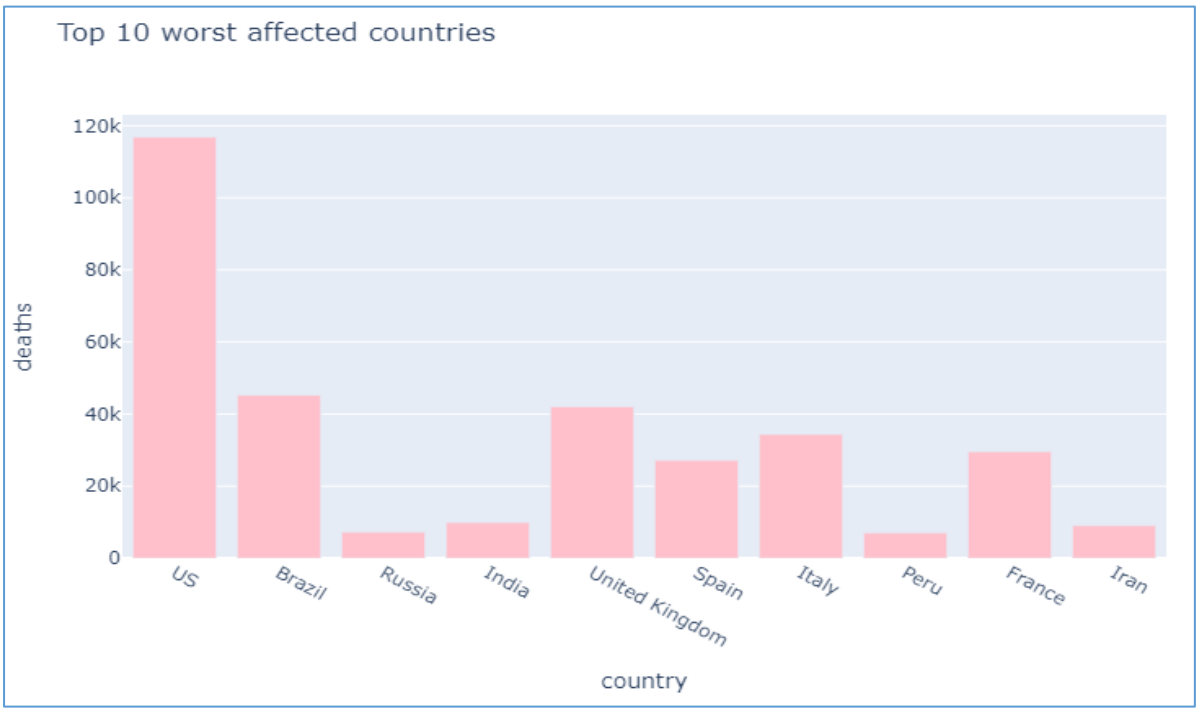
1. Confirmed cases



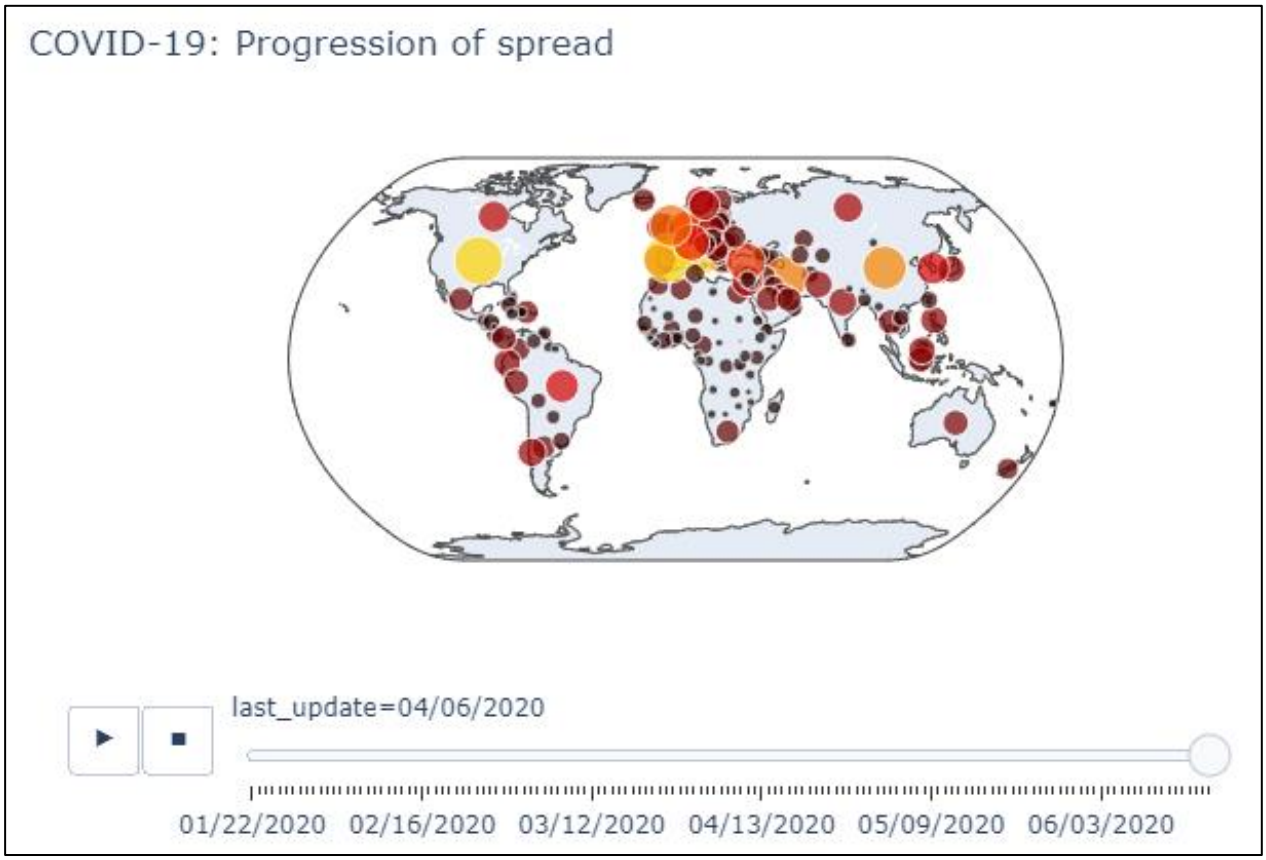
2. Deaths

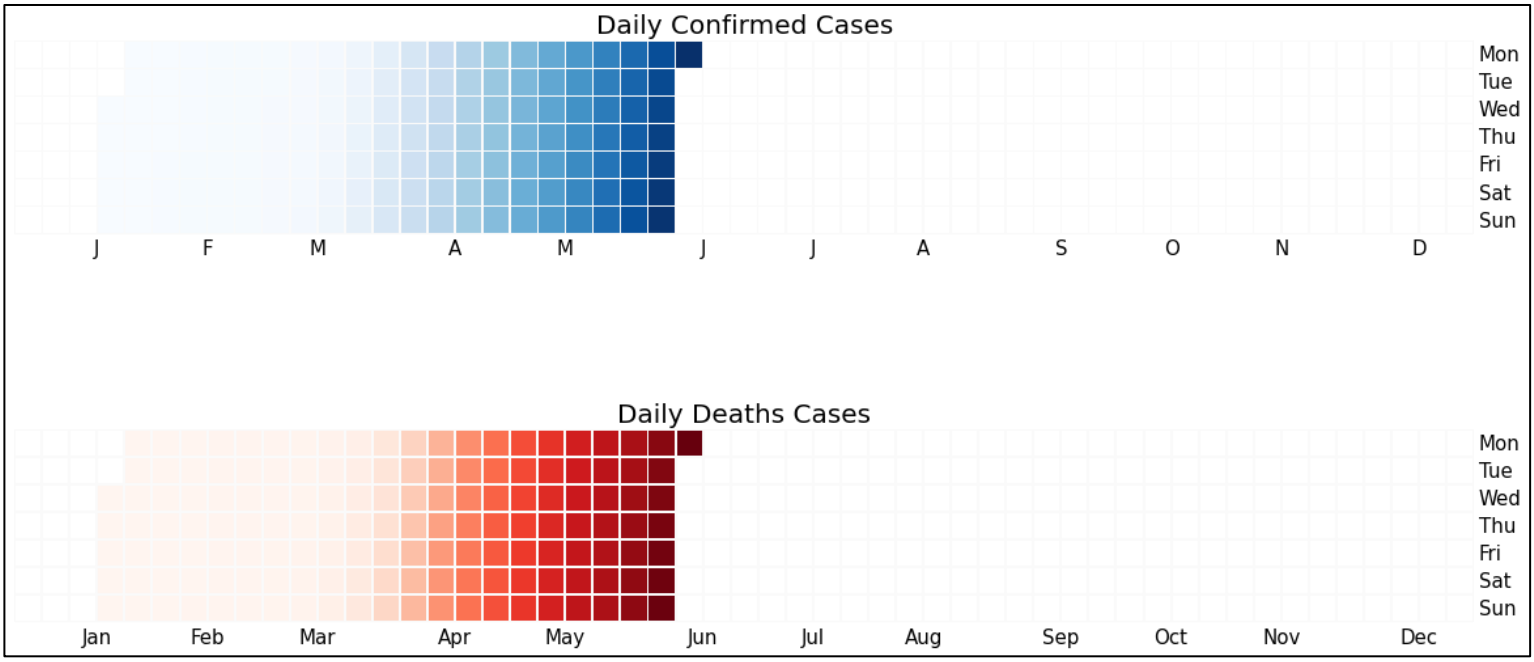


3. Recovered



Spread of COVID-19 across the Globe





Daily Growth in No. of COVID-19 cases

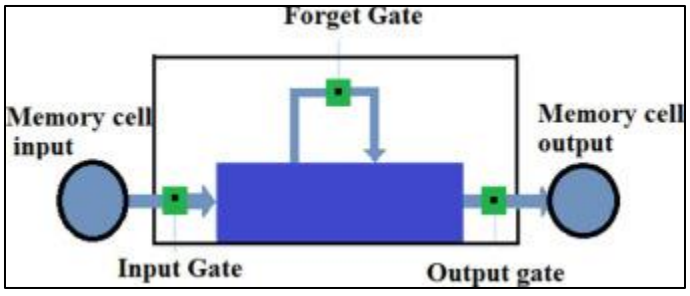
Part: 3) Covid19 forecasting with RNN

Even though epidemiological models are good at capturing vital components of an infectious disease, parameters of these models required several assumptions. Such hypothesized parameters would not fit the data perfectly and precision of such models will be low. Meanwhile, in engineering applications, model parameters are calculated with the help of real-time data.

In order to overcome the barriers of statistical approaches, Deep Learning based network can be used to predict the real-time transmission. This model could help public health care providers, policy makers to make necessary arrangements to tackle the rush of potential COVID-19 patients. A fully automated, real time forecasting model for COVID-19 transmission to help frontline health workers and government policy makers

Another approach to predict the no of cases is by using Recurrent Neural network. RNN is a deep learning model that is used for Time-series prediction, speech recognition, etc. Unlike traditional neural networks, recurrent networks use their memory (also called states) to predict sequence outputs. In simple words, RNN is used when we want to predict a future outcome based on the previous sequential inputs. For example, we can use RNN to predict the next word in a sentence by providing previous words.

LSTM model



The RNN will take as inputs:

- number of cases for 20 days
- number of fatalities for 20 days
- restrictions applied for the area in the past 20 days
- quarantine applied for the area in the past 20 days
- school opened or closed for the area in the past 20 days
- additional information related to the area (population, density, number of hospital beds, lung measurement, number of centenarian people)

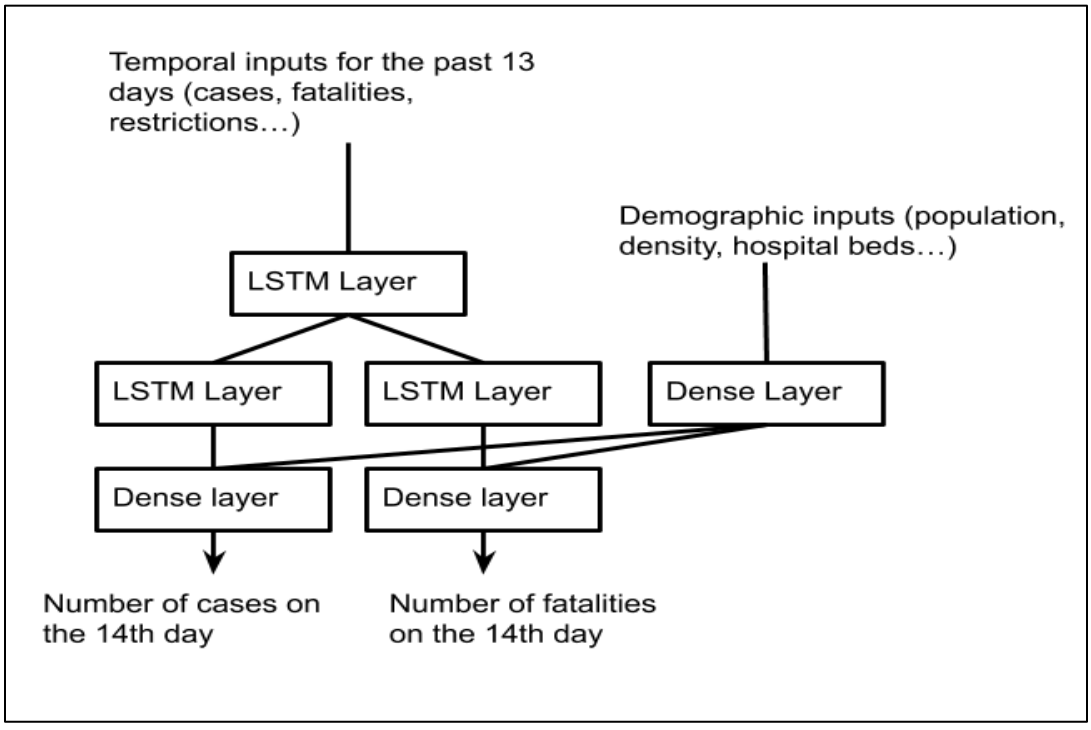
and as outputs:

- number of cases for the 21th day
- number of fatalities for the 21th day

Note:

Splitting the dataset with 90% for training and 10% for validation.

Model's Architecture

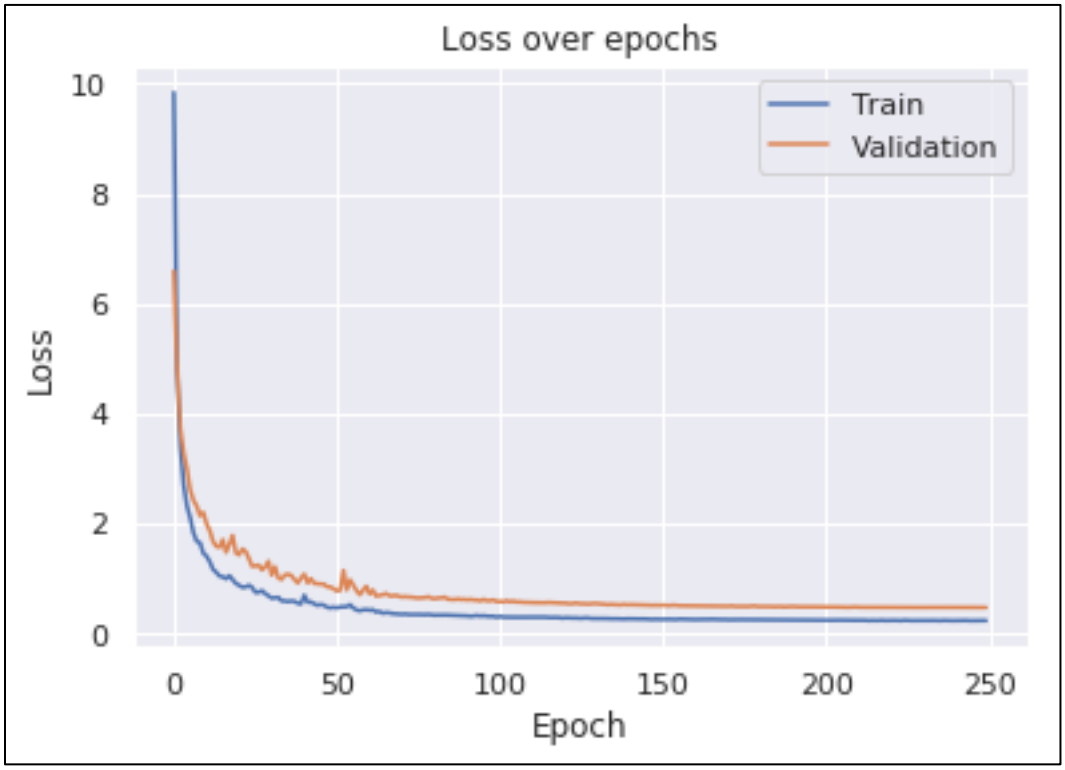


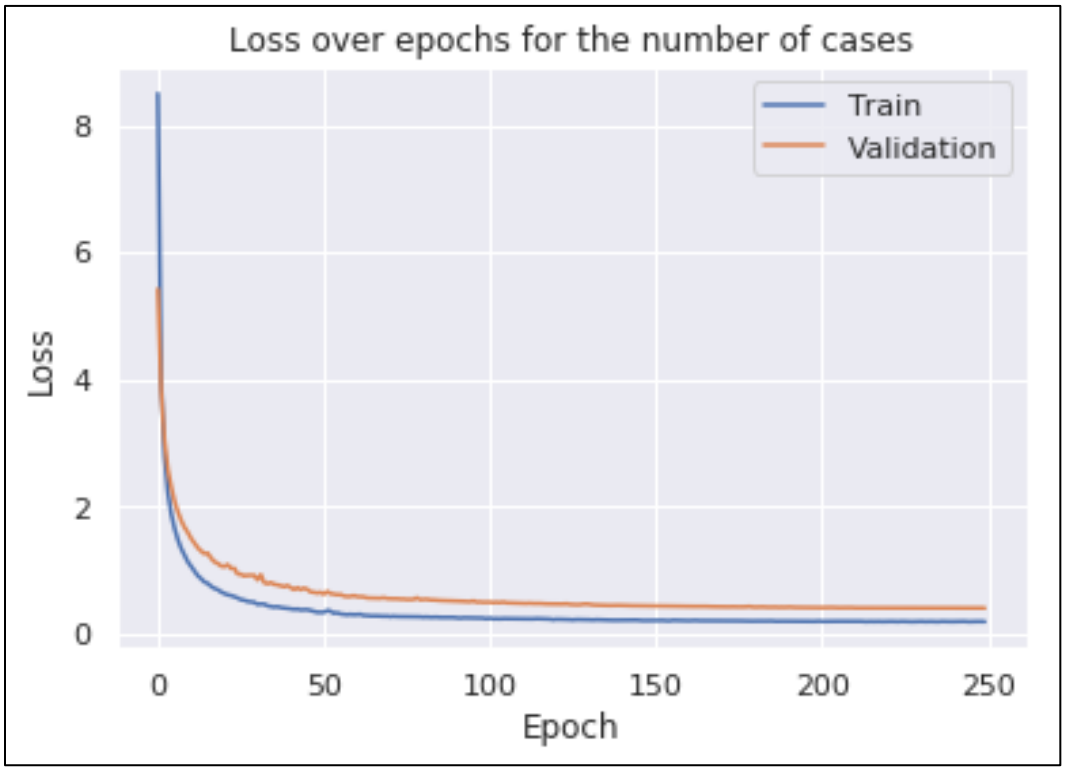
The model is very simple in terms of architecture. The only difference from what could traditionally be seen is that it has two outputs so we can have two different losses (one for the expected number of cases and for the expected number of fatalities).

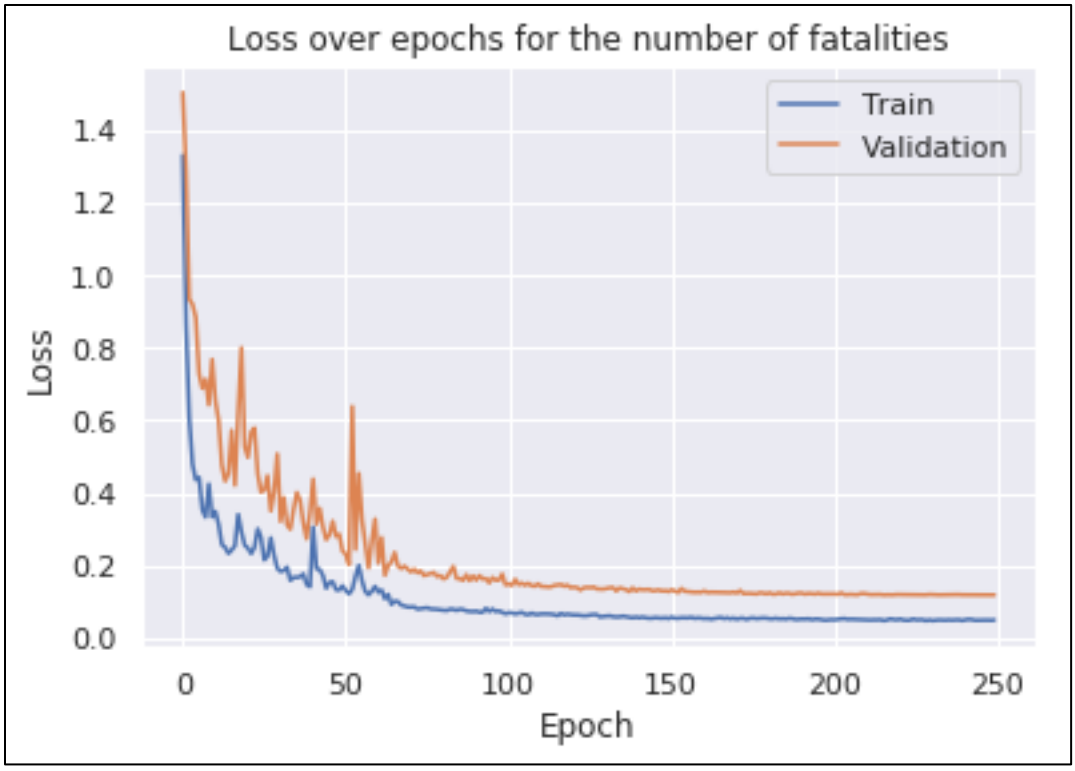
Total trainable parameters of model:

```
Model: "model"
__________________________________________________________________________________________________
Layer (type)                 Output Shape              Param #   Connected to
==================================================================================================
input_1 (InputLayer)         [(None, 20, 5)]           0
__________________________________________________________________________________________________
input_2 (InputLayer)         [(None, 5)]               0
__________________________________________________________________________________________________
lstm (LSTM)                  (None, 20, 64)            17920     input_1[0][0]
__________________________________________________________________________________________________
dense (Dense)                (None, 16)                96        input_2[0][0]
__________________________________________________________________________________________________
lstm_1 (LSTM)                (None, 32)                12416     lstm[0][0]
__________________________________________________________________________________________________
dropout (Dropout)            (None, 16)                0         dense[0][0]
__________________________________________________________________________________________________
lstm_2 (LSTM)                (None, 32)                12416     lstm[0][0]
__________________________________________________________________________________________________
concatenate (Concatenate)    (None, 48)                0         lstm_1[0][0]
                                                                 dropout[0][0]
__________________________________________________________________________________________________
concatenate_1 (Concatenate)  (None, 48)                0         lstm_2[0][0]
                                                                 dropout[0][0]
__________________________________________________________________________________________________
dense_1 (Dense)              (None, 128)               6272      concatenate[0][0]
__________________________________________________________________________________________________
dense_2 (Dense)              (None, 128)               6272      concatenate_1[0][0]
__________________________________________________________________________________________________
dropout_1 (Dropout)          (None, 128)               0         dense_1[0][0]
__________________________________________________________________________________________________
dropout_2 (Dropout)          (None, 128)               0         dense_2[0][0]
__________________________________________________________________________________________________
cases (Dense)                (None, 1)                 129       dropout_1[0][0]
__________________________________________________________________________________________________
fatalities (Dense)           (None, 1)                 129       dropout_2[0][0]
==================================================================================================
Total params: 55,650
Trainable params: 55,650
Non-trainable params: 0
__________________________________________________________________________________________________
```

Model Loss over Epochs:





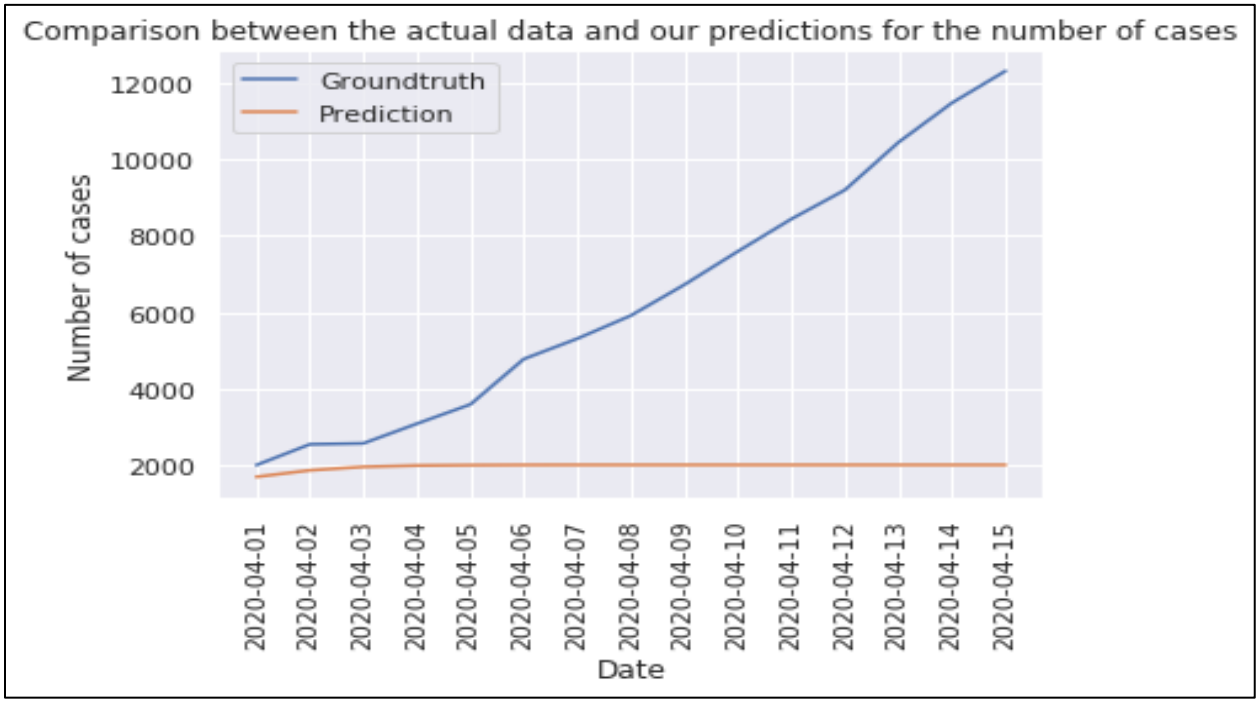


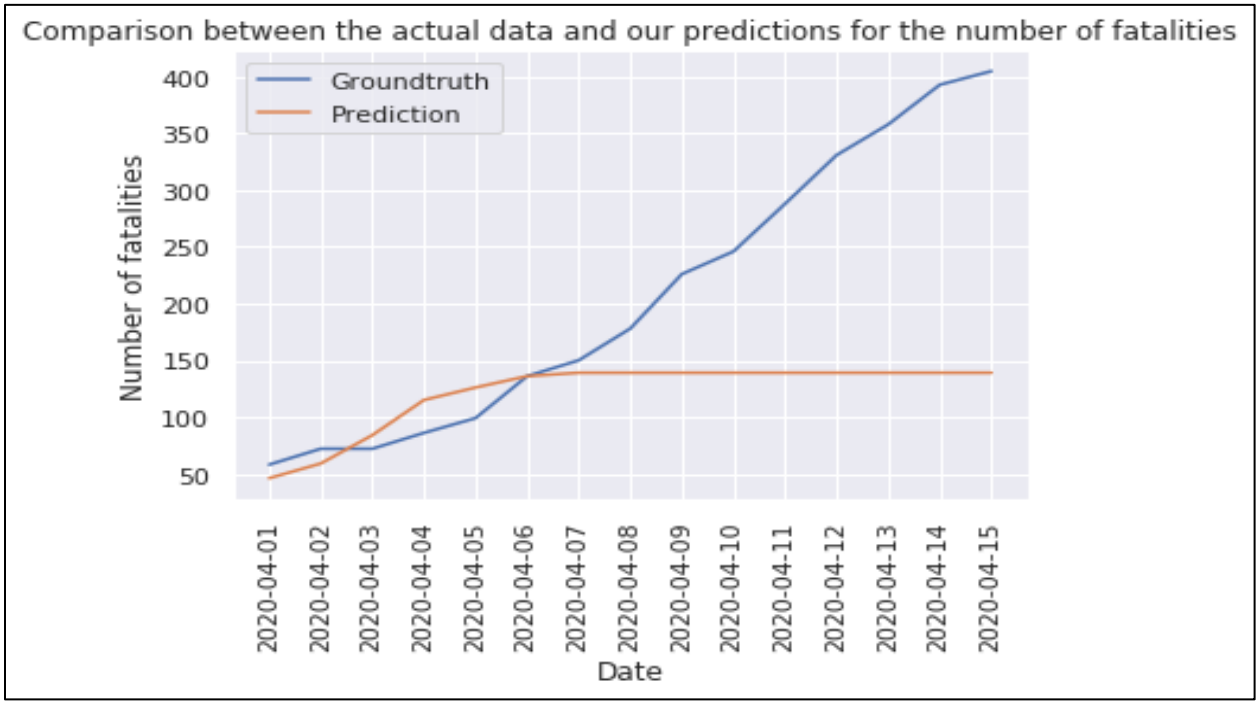
Errors: Below shown the root mean squared errors of cases as well as fatalities

```
France
RMSLE on cases: 0.9888926389687206
RMSLE on fatalities: 1.678121938079242
(44, 7) (44, 15)
Italy
RMSLE on cases: 0.5544086693861676
RMSLE on fatalities: 0.6825648180586751
(44, 7) (44, 15)
United Kingdom
RMSLE on cases: 1.6392286870476094
RMSLE on fatalities: 2.0536330823986924
(44, 7) (44, 15)
Spain
RMSLE on cases: 0.6947674189444768
RMSLE on fatalities: 0.9064676028292048
(44, 7) (44, 15)
Iran
RMSLE on cases: 0.6543418305070255
RMSLE on fatalities: 0.6179601431953902
(44, 7) (44, 15)
Germany
RMSLE on cases: 0.7046171533328783
RMSLE on fatalities: 1.811541832778929
(44, 7) (44, 15)
India
RMSLE on cases: 2.4620758241515195
RMSLE on fatalities: 1.799015053352249
```

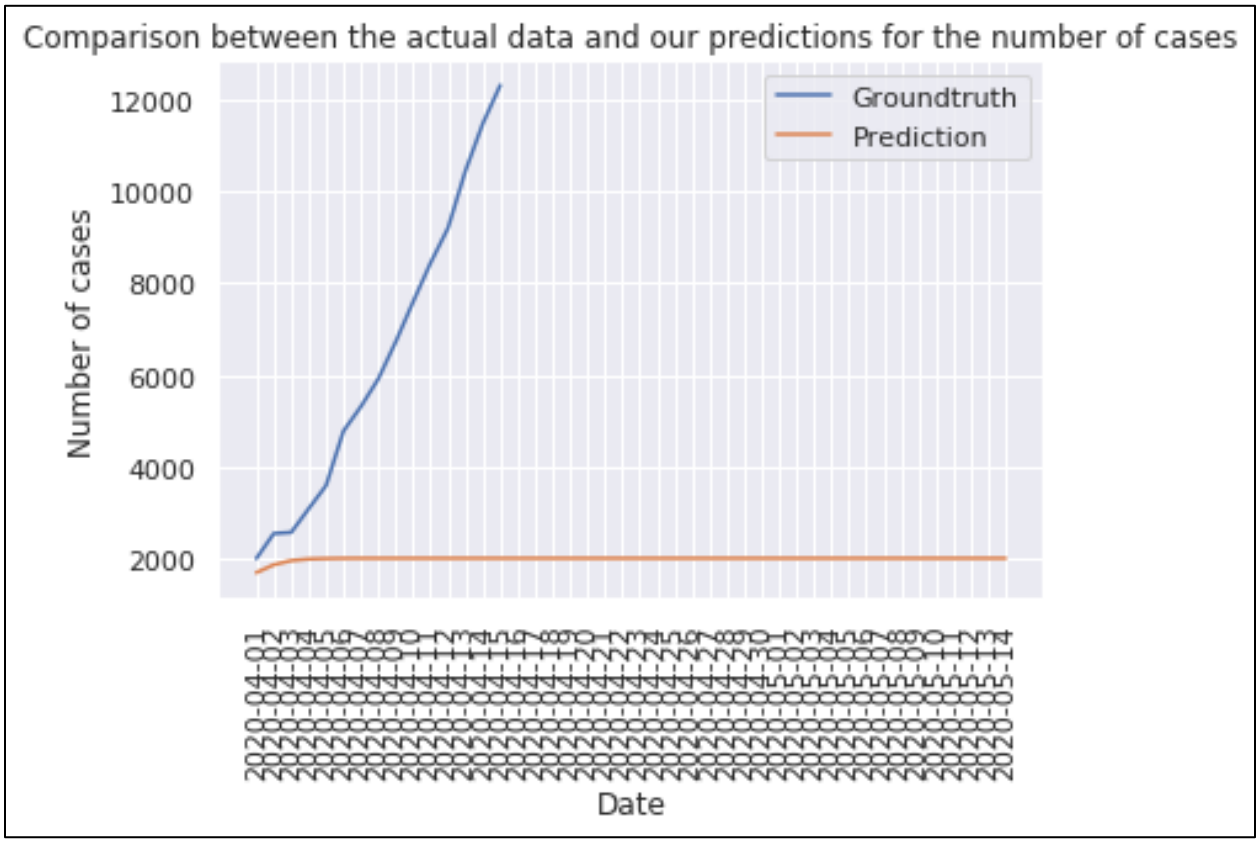
Outputs: Observing the curves

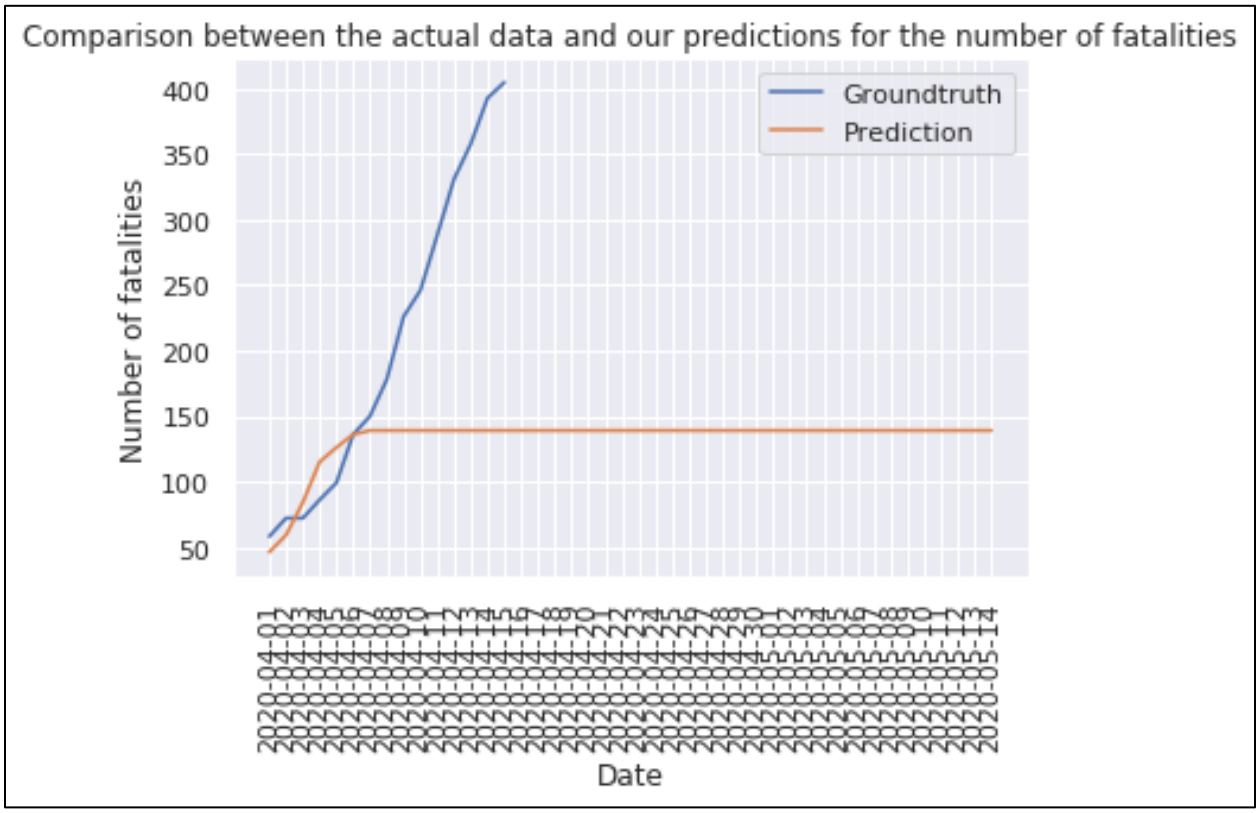
1.short-term prediction: These plots are deliberately generated for India. The codes in the notebook can be used to predict any country.





2. Long- Term prediction





Conclusion:

It is very intersting to note that pattern revealed by the RNN network is almost similar to situation which was arised after the prediction. This network was trained at the beginnig of pandemic. Benefit of using RNN is : We don’t have to worry about the no of parameters involved in COVID-19 transmission. Thanks to the Deep learning. We have’t trained this LSTM model on present day data. This is next task we are looking to work on.

Part :4 Analysis of COVID-19 Present situation in India

Exploratory Data analysis for India:

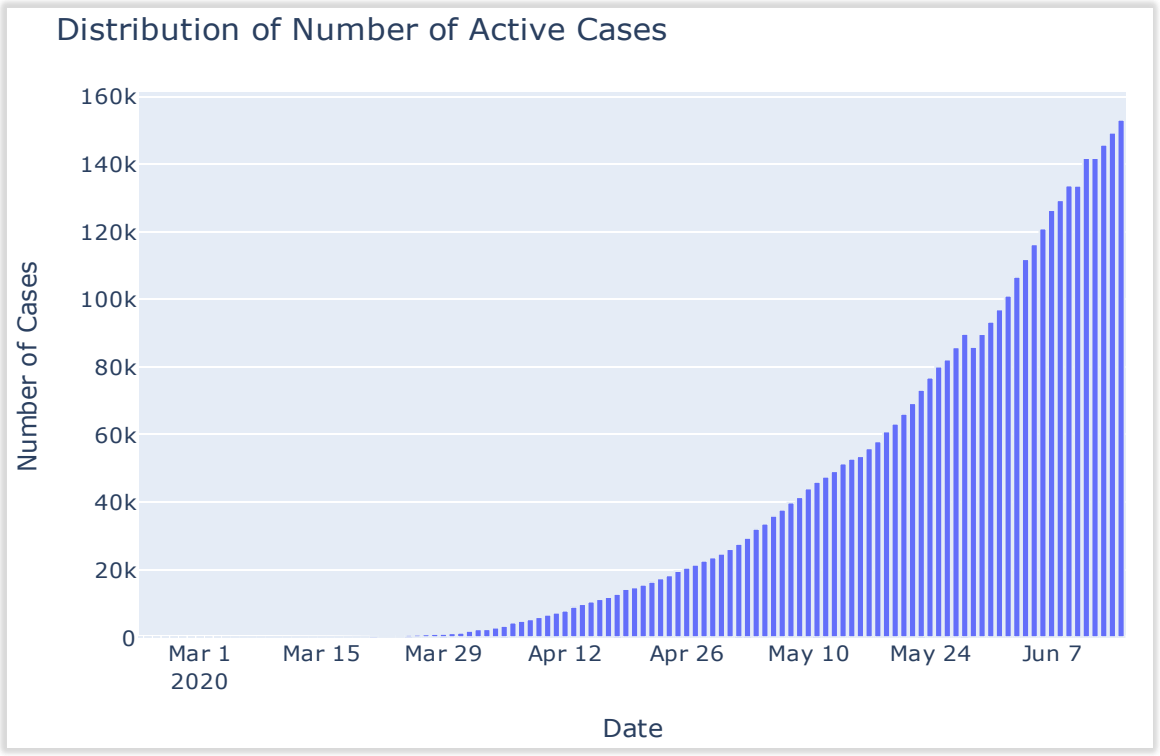
(All the figures used in this part are produced from covid-19-data-analysis-forecasting-for-india.ipynb)

In present situation, India no of COVID-19 cases are increasing continuously. In this section analysis of no of cases in lockdown is done. After the analysis, forecasting for India is done for next 5 days using Machine learning models and Time series forecasting models.

Analysis of Present situation in India is as follows:

Active Cases = Number of Confirmed Cases - Number of Recovered Cases - Number of Death Cases

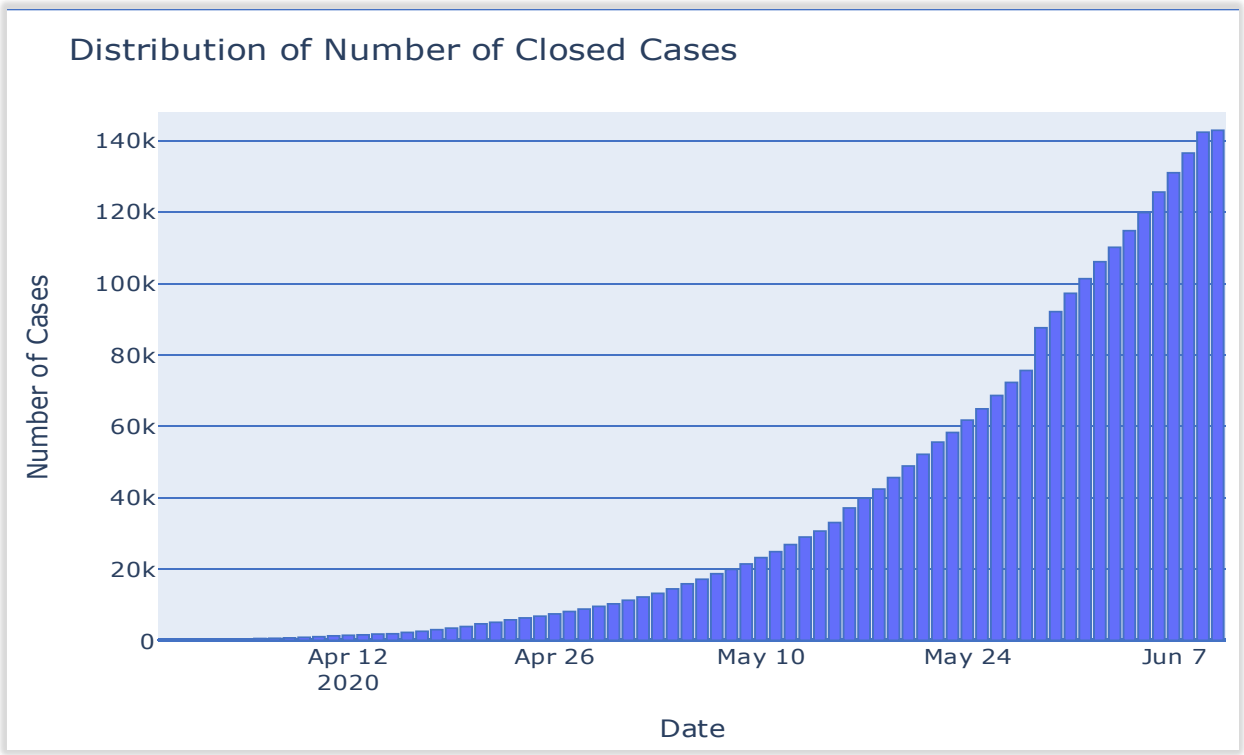
(Analysis done till date: 15-June-2020)



Number of Confirmed Cases 343091.0

Number of Recovered Cases 180013.0

Closed Cases = Number of Recovered Cases + Number of Death Cases



Number of Death Cases 9900.0

Number of Active Cases 153178.0

Number of Closed Cases 189913.0

Approximate Number of Confirmed Cases per day 2486.0

Approximate Number of Recovered Cases per day 1304.0

Approximate Number of Death Cases per day 72.0

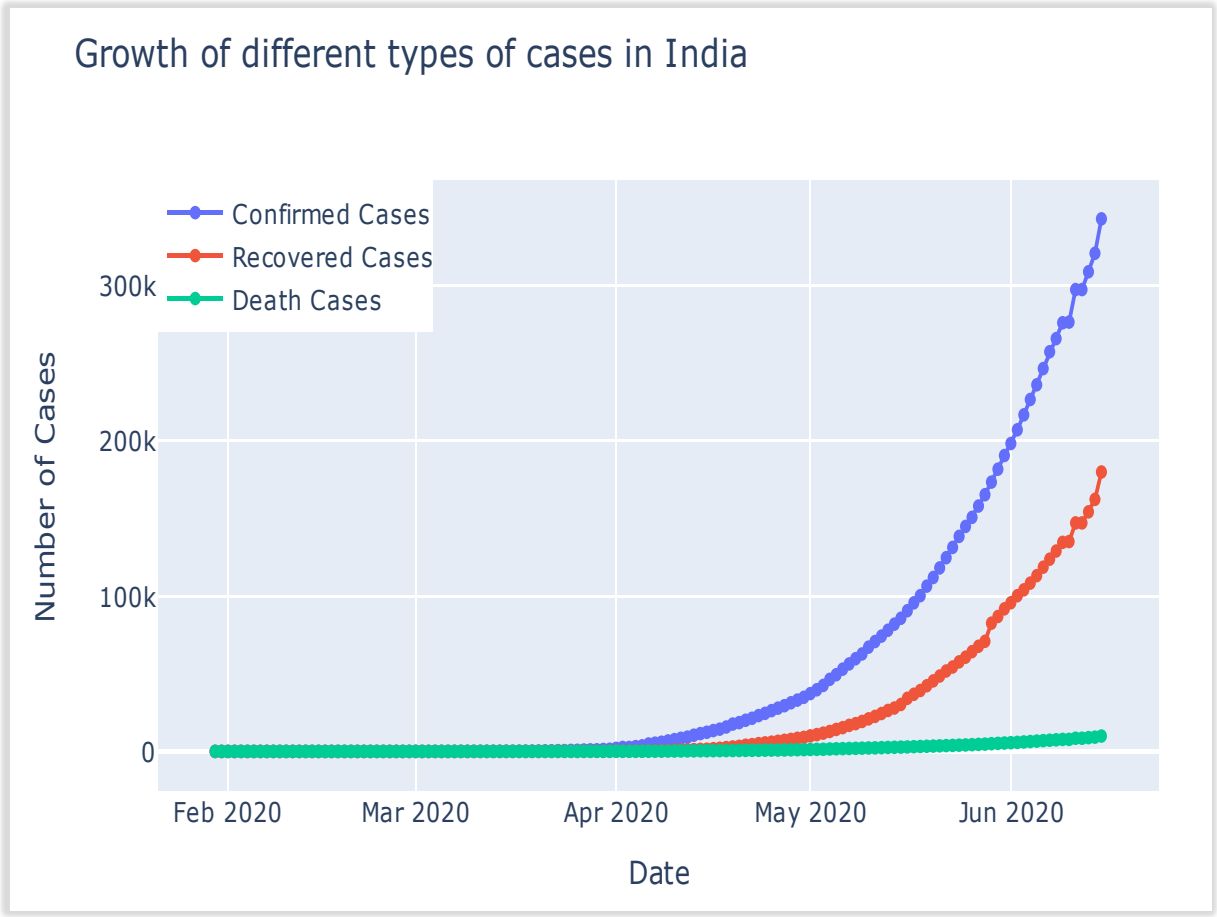
Number of New Confirmed Cases in last 24 hours are 22169.0

Number of New Recovered Cases in last 24 hours are 17634.0

Number of New Death Cases in last 24 hours are 705.0

Observation: Almost Exponential growth of Confirmed Cases in comparison to Recovered and Death Cases is a conclusive evidence why there is increase in number of Active Cases.

It is clear from the plot that growth rate of cases is similar to exponential.

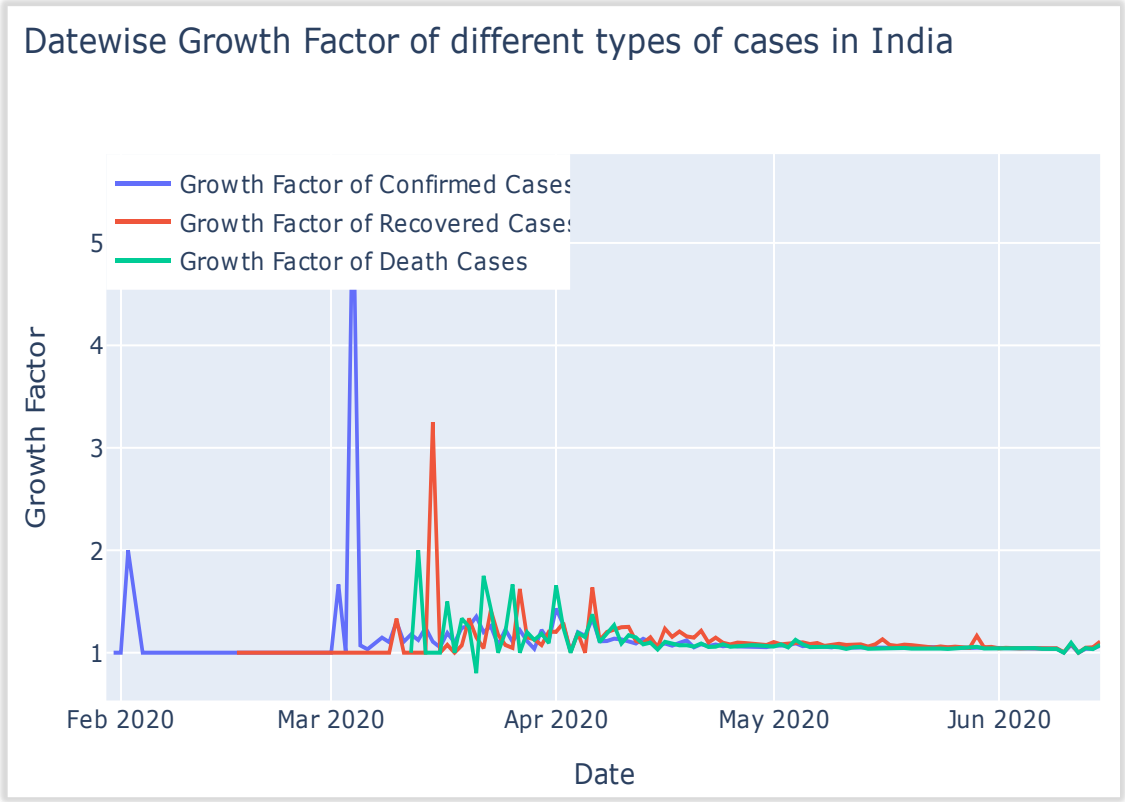


Mortality rate = (Number of Death Cases / Number of Confirmed Cases) x 100

Recovery Rate= (Number of Recovered Cases / Number of Confirmed Cases) x 100

Analysis shows the sudden rise in no of deaths as well as no of recoveries of COVID-19. From May 2020 to June 2020, Mortality rate is decreased while recovery rate is increased. it shows positive indication about improvement in

COVID-19 situations but later after unlocking lockdown situation is again tending towards outbreak.



Observation:

- Recovery Rate was initially very high when the number of positive (Confirmed) cases were low and showed a drastic drop with increasing number of cases. Increasing Mortality rate and dropped Recovery Rate is worrying sign for India.

- Increasing Mortality Rate and very slowly increasing Recovery Rate is conclusive evidence for increase in number of Closed Cases
- Recovery Rate is showing an upward trend which is a really good sign. Mortality Rate is showing a slight dip but with occasional upward trends.

Growth Factor analysis of Cases:

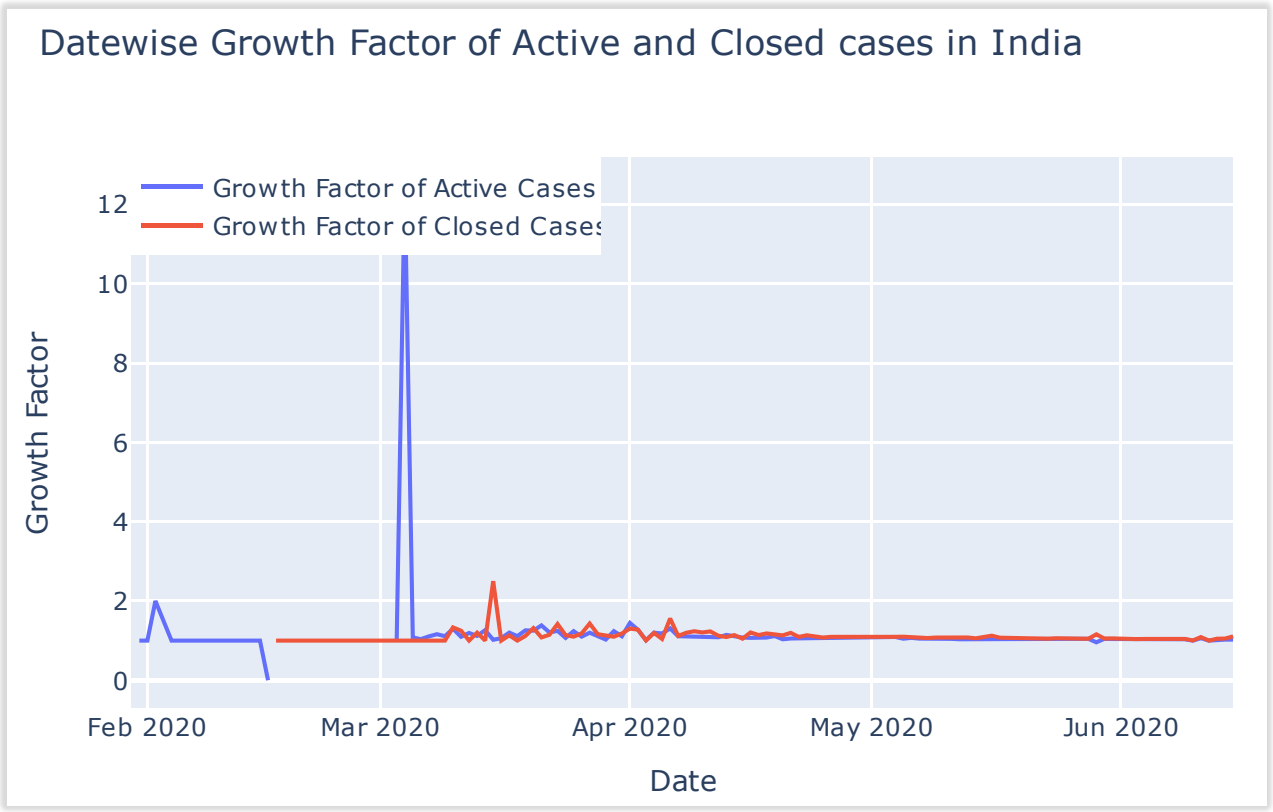
- Growth factor is the factor by which a quantity multiplies itself over time. The formula used is:

- Formula: Every day's new (Confirmed, Recovered, Deaths) / new (Confirmed, Recovered, Deaths) on the previous day.
- A growth factor above 1 indicates an increase corresponding case.
- A growth factor above 1 but trending downward is a positive sign, whereas a growth factor constantly above 1 is the sign of exponential growth.
- A growth factor constant at 1 indicates there is no change in any kind of cases.

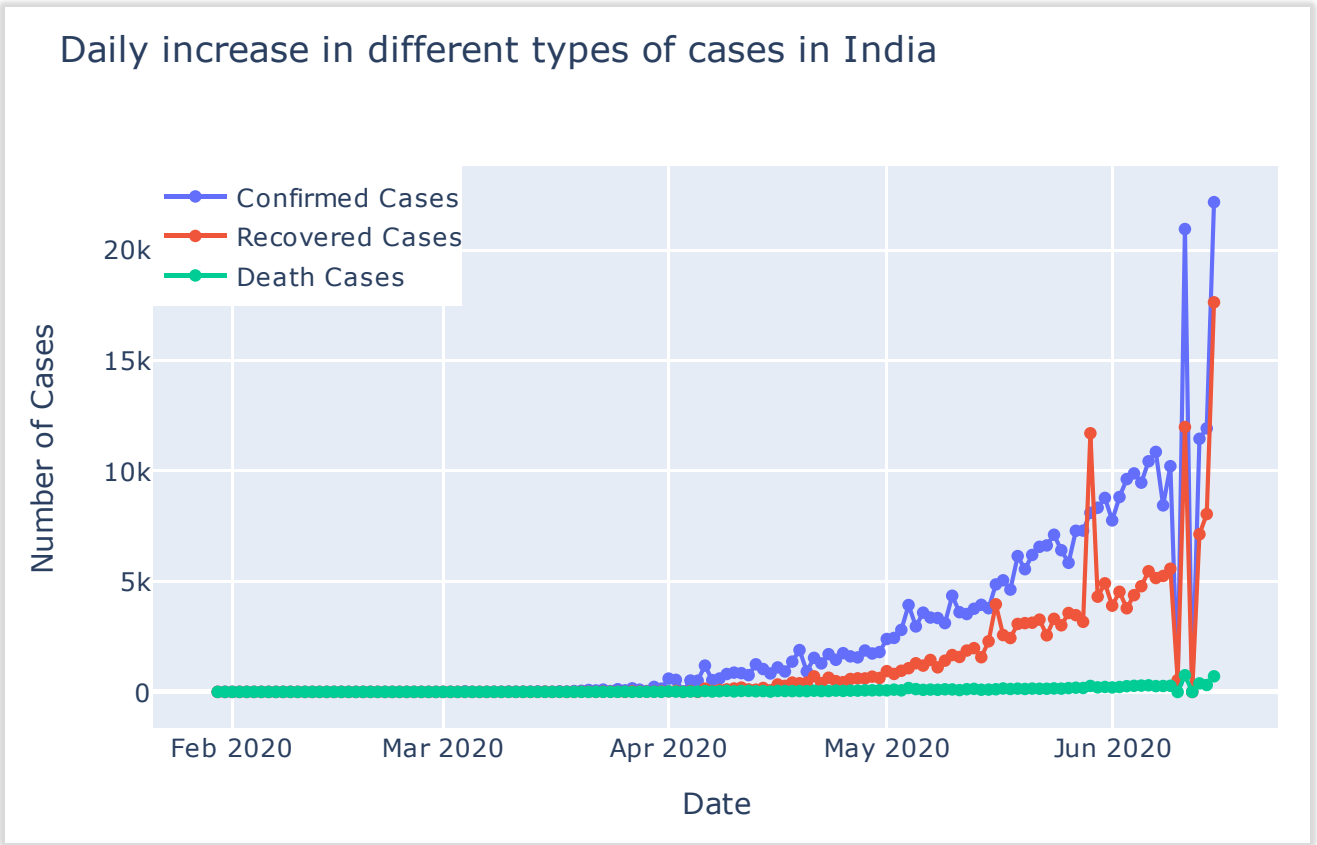
- Growth Factor of Recovered Cases is constantly very close to 1 indicating the Recovery Rate very low which was high initially as discussed earlier, with Growth Factor of Confirmed and Death Cases well above 1 is an indication of considerable growth in both types of Cases.

Growth factor for Active and Closed Cases

- Growth factor is the factor by which a quantity multiplies itself over time. The formula used is:
- Formula: Every day's new (Active and Closed Cases) / new (Active and Closed Cases) on the previous day.
- A growth factor above 1 indicates an increase corresponding case.
- A growth factor above 1 but trending downward is a positive sign.
- A growth factor constant at 1 indicates there is no change in any kind of cases.
- A growth factor below 1 indicates real positive sign implying more patients are getting recovered or dying as compared to the Confirmed Cases.



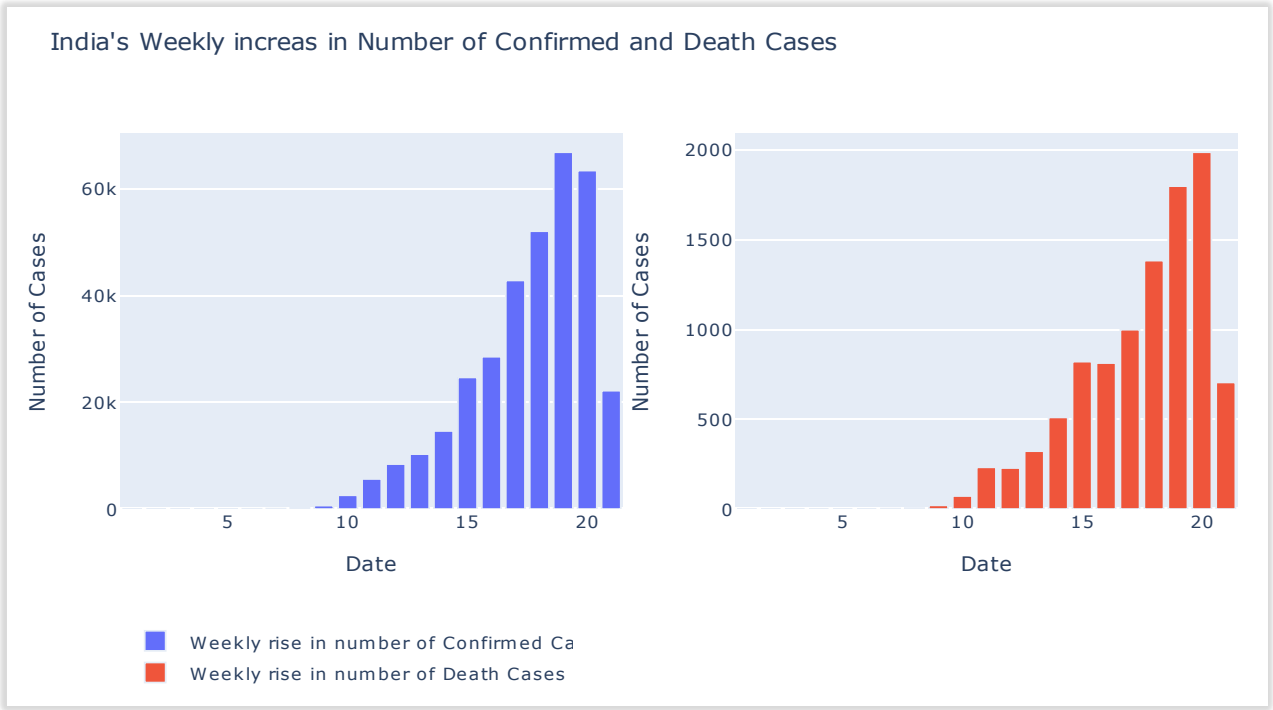
Daily increase in Cases

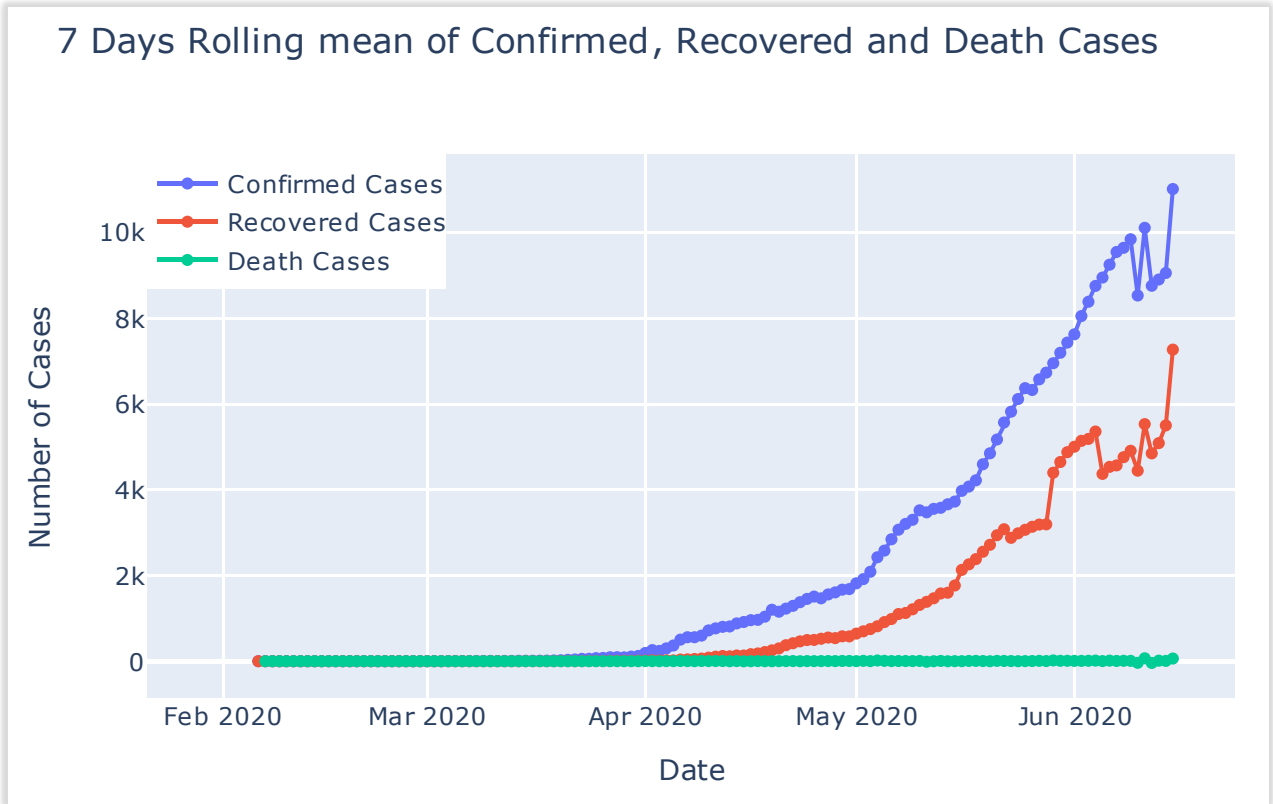


Average weekly increase in number of Confirmed Cases 16338

Average weekly increase in number of Recovered Cases 8572

Average weekly increase in number of Death Cases 471



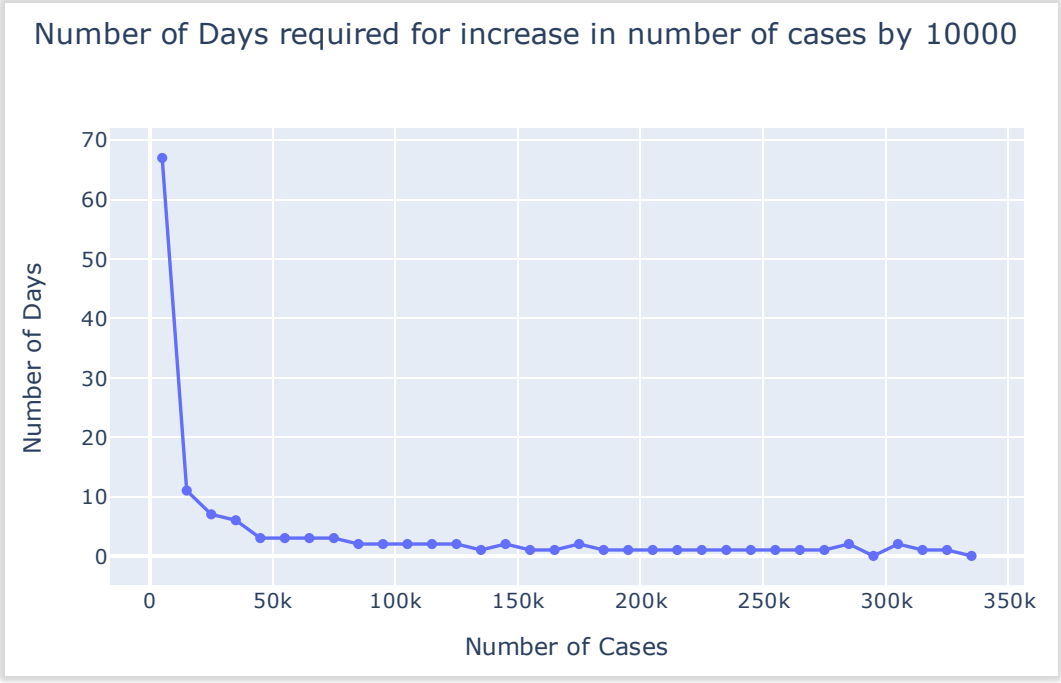


Week 21th week has just started….

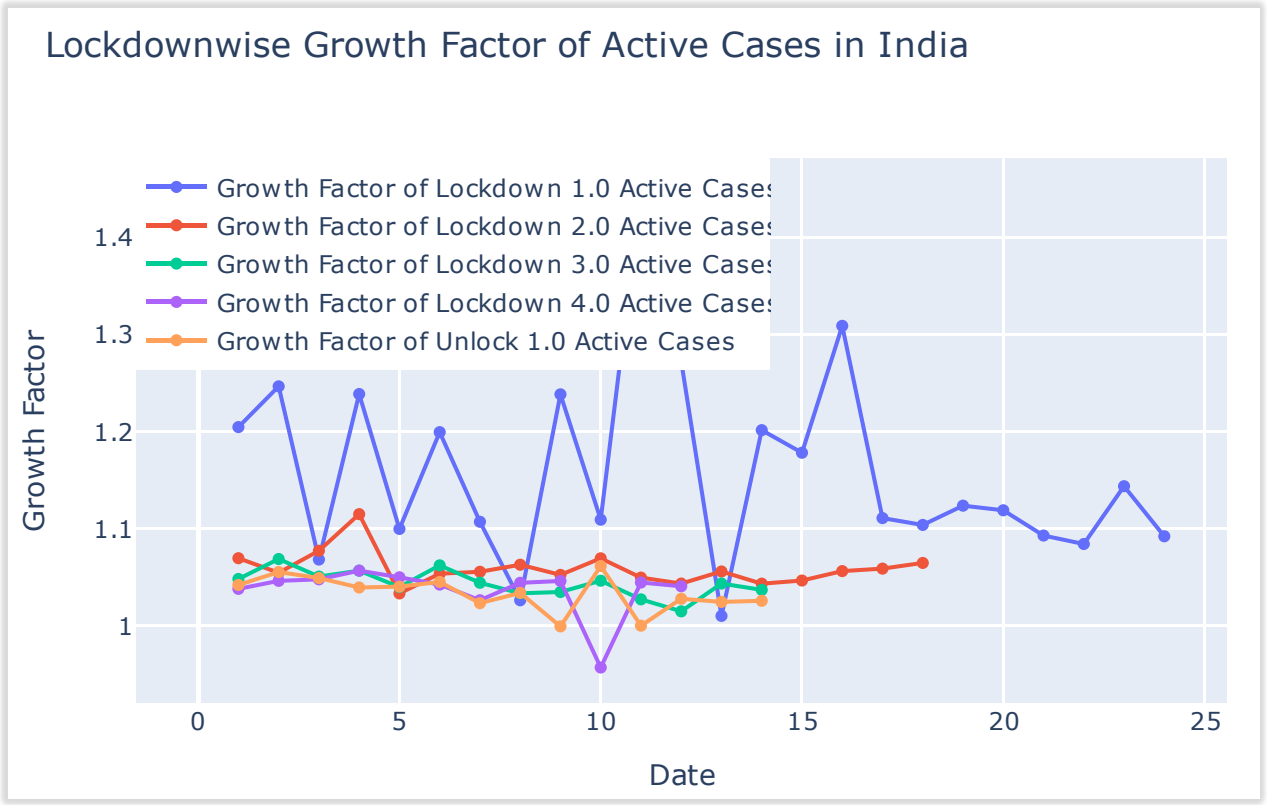
- Confirmed Cases are showing upward trend every week.
- Death cases showed a very slight dip in 16th week which is positive sign, but it followed a huge spike in 17th week and it following up in subsequent weeks which is alarming.

	No. of cases	Days since first case	Number of days required to Double the cases
0	65	41	41.000000
1	130	46	5.000000
2	260	50	4.000000
3	520	53	3.000000
4	1040	59	6.000000
5	2080	62	3.000000
6	4160	66	4.000000
7	8320	71	5.000000
8	16640	79	8.000000
9	33280	90	11.000000
10	66560	100	10.000000
11	133120	114	14.000000
12	266240	130	16.000000

Let's have a look at doubling rate of cases in India: It shows on increasing no of cases, days require to increase cases by 10000 are reducing.



Lockdown Analysis for India



Average Active Cases growth rate in Lockdown 1.0: 1.1593835916264263

Median Active Cases growth rate in Lockdown 1.0: 1.1212799256202945

Average Active Cases growth rate in Lockdown 2.0: 1.0589658468208336

Median Active Cases growth rate in Lockdown 2.0: 1.055711195842128

Average Active Cases growth rate in Lockdown 3.0: 1.0433479599958473

Median Active Cases growth rate in Lockdown 3.0: 1.0438197911523128

Average Active Cases growth rate in Lockdown 4.0: 1.0366005332298471

Median Active Cases growth rate in Lockdown 4.0: 1.0443448437313414

Average Active Cases growth rate in Unlock 1.0: 1.0333187386868092

Median Active Cases growth rate in Unlock 1.0: 1.0365447341500578

Let's analyse the effect of lockdown on doubling rate of cases.

Doubling rate in No lockdown period:

	No. of Cases	Days Since First Case	Days required for Doubling
0	1	2	2.000000
1	2	3	1.000000
2	4	31	28.000000
3	8	33	2.000000
4	16	33	0.000000
5	32	36	3.000000
6	64	41	5.000000
7	128	46	5.000000

Doubling rate in lockdown 1.0:

	No. of Cases	Days Since Lockdown 1.0	Days required for Doubling
0	330.000000	0	0.000000
1	660.000000	4	4.000000
2	1320.000000	9	5.000000
3	2640.000000	13	4.000000
4	5280.000000	16	3.000000
5	10560.000000	23	7.000000

Doubling rate in Lockdown 2.0

	No. of Cases	Days Since Lockdown 2.0	Days required for Doubling
0	12322.000000	0	0.000000
1	24644.000000	9	9.000000

Doubling rate in lockdown 3.0

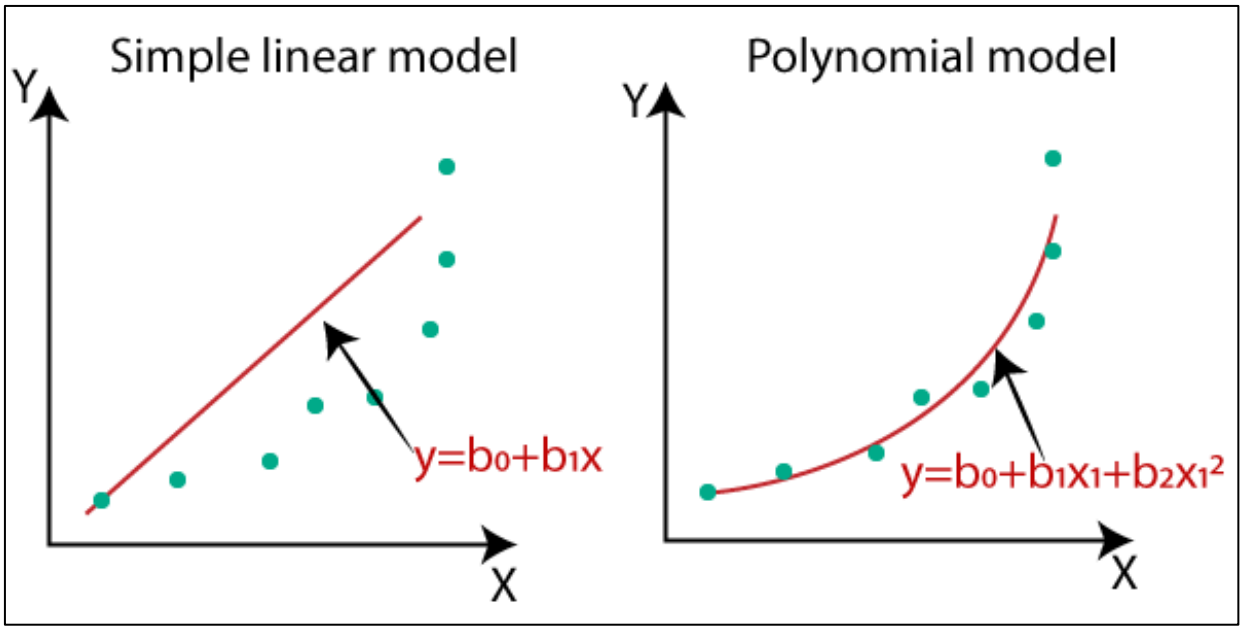
	No. of Cases	Days Since Lockdown 3.0	Days required for Doubling
0	46437.000000	0	0.000000
1	92874.000000	12	12.000000

- All Lockdowns seems to have shown a slight effect of the Growth Rate of Active Cases implying the COVID-19 controlling practices are working well but can be improved.
- The Growth rate of Active Cases has slowed down during each Lockdown.
- Doubling Rate of Cases seems to have improved significantly during each Lockdown period which is a good sign
- Growth of Active Cases is showing an increasing trend in Lockdown 4.0, probably because Lockdown 4.0 is much more lenient as compared to previous Lockdown versions.

Part: 5 - Machine Learning Predictions:

(Note: All the figure showing training of model and forecasting are taken form forecasting-india-COVID-19.ipynb notebook.)

1) Polynomial Regression:



Polynomial Regression is a form of linear regression in which the relationship between the independent variable x and dependent variable y is modelled as an *nth* degree polynomial. Polynomial regression fits a nonlinear relationship between the value of x and the corresponding conditional mean of y, denoted E(y |x)

Why Polynomial Regression:

- There are some relationships that a researcher will hypothesize is curvilinear. Clearly, such type of cases will include a polynomial term.
- Inspection of residuals. If we try to fit a linear model to curved data, a scatter plot of residuals (Y axis) on the predictor (X axis) will have patches of many positive residuals in the middle. Hence in such situation it is not appropriate.

- An assumption in usual multiple linear regression analysis is that all the independent variables are independent. In polynomial regression model, this assumption is not satisfied.

Uses of Polynomial Regression:

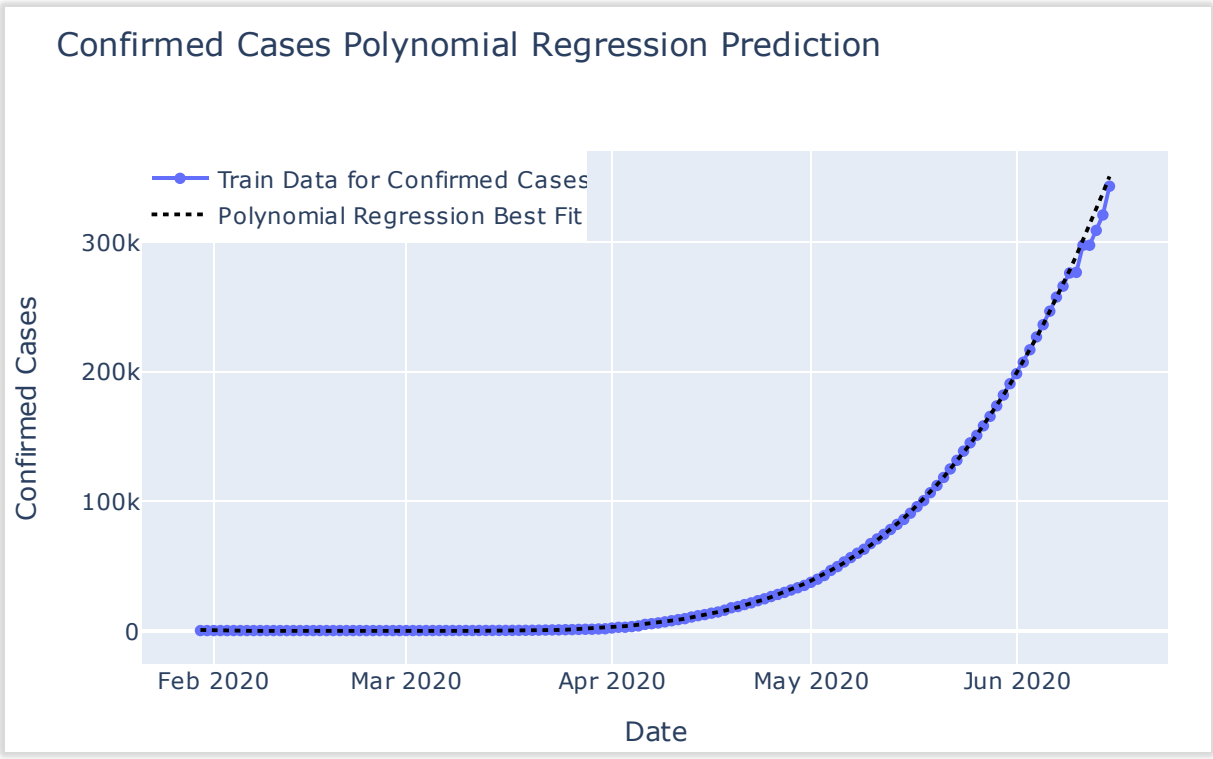
These are basically used to define or describe non-linear phenomenon such as:

- Growth rate of tissues.
- Progression of disease epidemics
- Distribution of carbon isotopes in lake sediments

The basic goal of regression analysis is to model the expected value of a dependent variable y in terms of the value of an independent variable x.

Forecasting using Polynomial Regression:

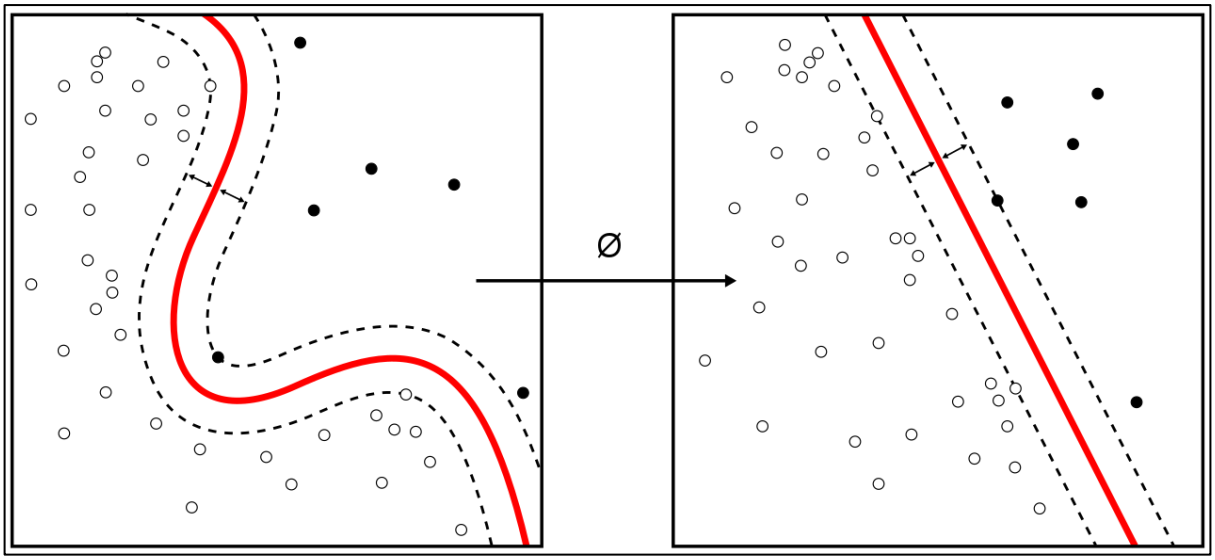
After Training this Model on COVID-19 data of India, forecasting for next 5 days is done that is as follows:



Root Mean Squared Error for Polynomial Regression: 12199.2466311633

	Date	Polynomial Regression Prediction
0	2020-06-16	363780.706971
1	2020-06-17	377183.628229
2	2020-06-18	390873.421254
3	2020-06-19	404844.389473
4	2020-06-20	419089.898330

2) Support Vector Machine Regressor:



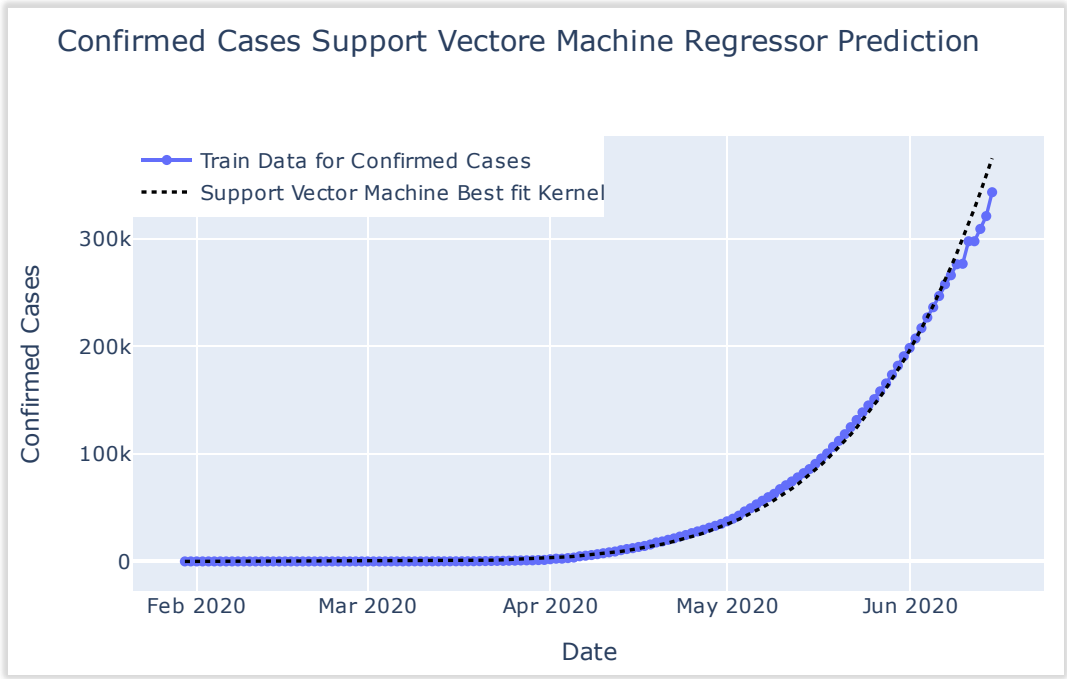
In machine learning, **support-vector machines** (**SVMs**, also **support-vector networks**[1]) are supervised learning models with associated learning algorithm that analyse data used for classification and **regression analysis**.

The Support Vector Machine (SVM) algorithm is a popular machine learning tool that offers solutions for both classification and regression problems. it presents one of the most robust prediction methods, based on the statistical

learning framework An SVM model is a representation of the examples as points in space, mapped so that the examples of the separate categories are divided by a clear gap that is as wide as possible. New examples are then mapped into that same space and predicted to belong to a category based on the side of the gap on which they fall. In addition to performing linear classification, SVMs can efficiently perform a non-linear classification using what is called the kernel trick, implicitly mapping their inputs into high-dimensional feature spaces.

Forecasting using SVM:

After Training this Model on COVID-19 data of India, forecasting for next 5 days is done that is as follows:



Root Mean Square Error for SVR Model: 27733.708602985276

Prediction of Model after training:

	Date	Polynomial Regression Prediction	SVM Prediction
0	2020-06-16	363780.706971	391314.245619
1	2020-06-17	377183.628229	408639.000287
2	2020-06-18	390873.421254	426598.283321
3	2020-06-19	404844.389473	445210.552046
4	2020-06-20	419089.898330	464494.663558

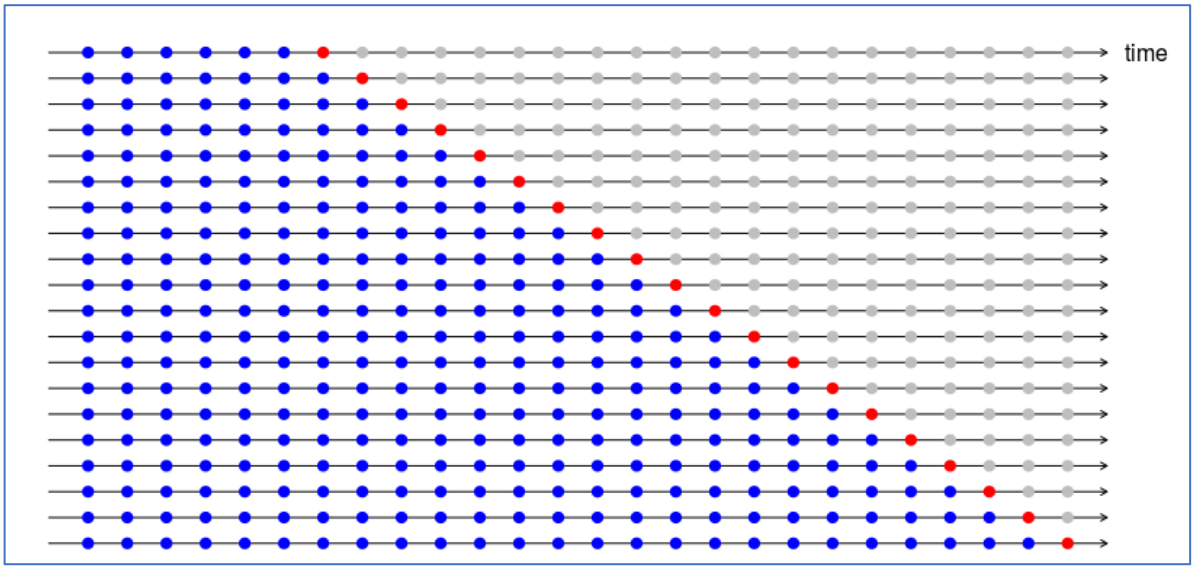
3) Time series forecasting models:

Time series forecasting is a hot topic which has many possible applications, such as stock prices forecasting, weather forecasting, business planning, resources allocation and many others. Even though forecasting can be considered as a subset of supervised regression problems, some specific tools are necessary due to the temporal nature of observations.

A time series is usually modelled through a stochastic process *Y(t),* i.e. a sequence of random variables. In a forecasting setting we find ourselves at time *t* and we are interested in estimating *Y(t+h),* using only information available at time *t.*

Due to the temporal dependencies in time series data, we cannot rely on usual validation techniques. To avoid biased evaluation, we must ensure that training sets contains observations that occurred prior to the ones in validation sets.

A possible way to overcome this problem is to use a sliding window. This procedure is called **time series cross validation** and it is summarised in the following picture, in which the blue points represents the training sets in each "fold" and the red points represent the corresponding validation sets.

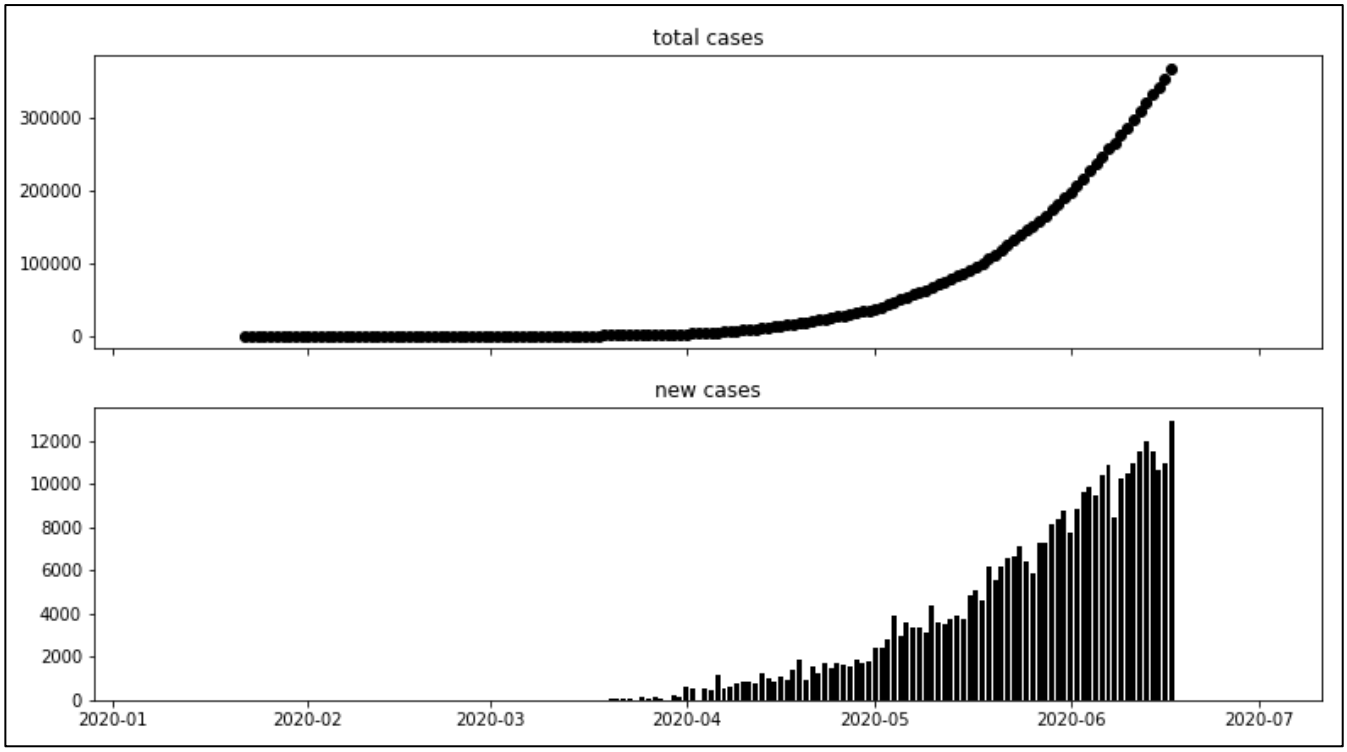


If we are interested in forecasting the next *n* time steps, we can apply the cross validation procedure for *1,2,…,n* steps ahead. In this way we can also compare the goodness of the forecasts for different time horizons.

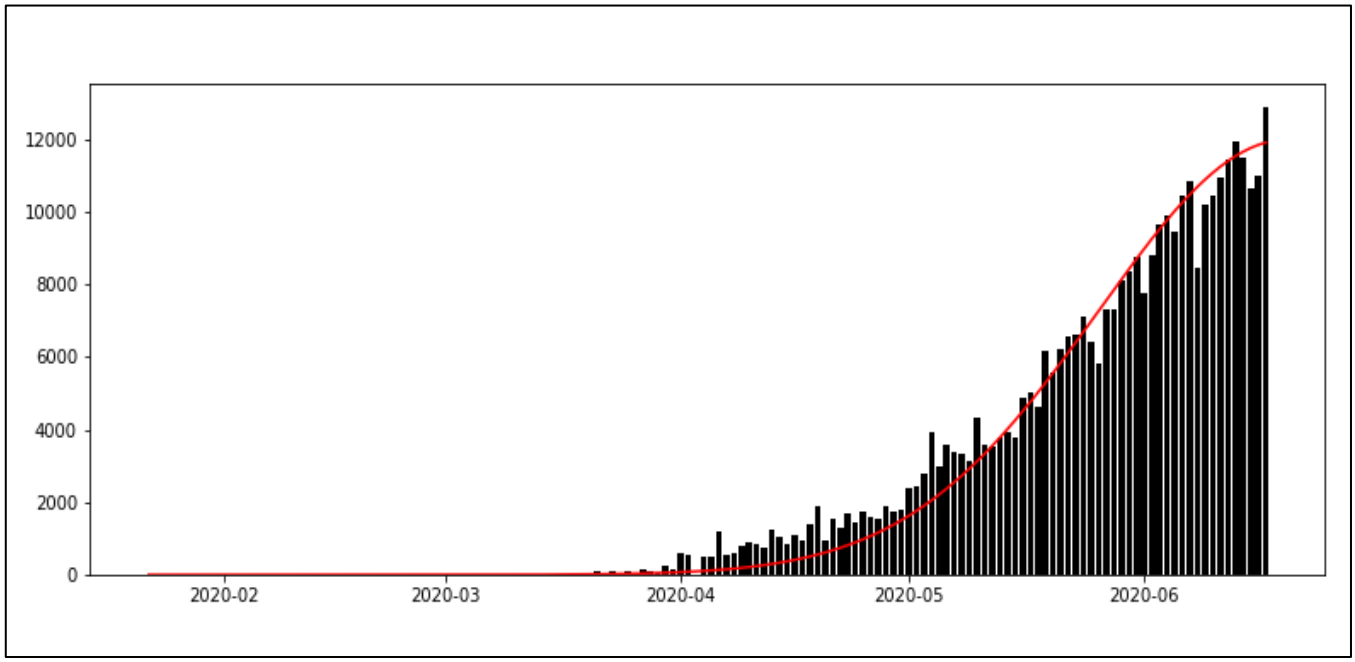
Once we have chosen the best model, we can fit it on the entire training set and evaluate its performance on a separate **test set subsequent in time**. The performance estimate can be done by using the same sliding window technique used for cross validation, but without re-estimating the model parameters.

Let's analyse the COVID-19 cases in India:

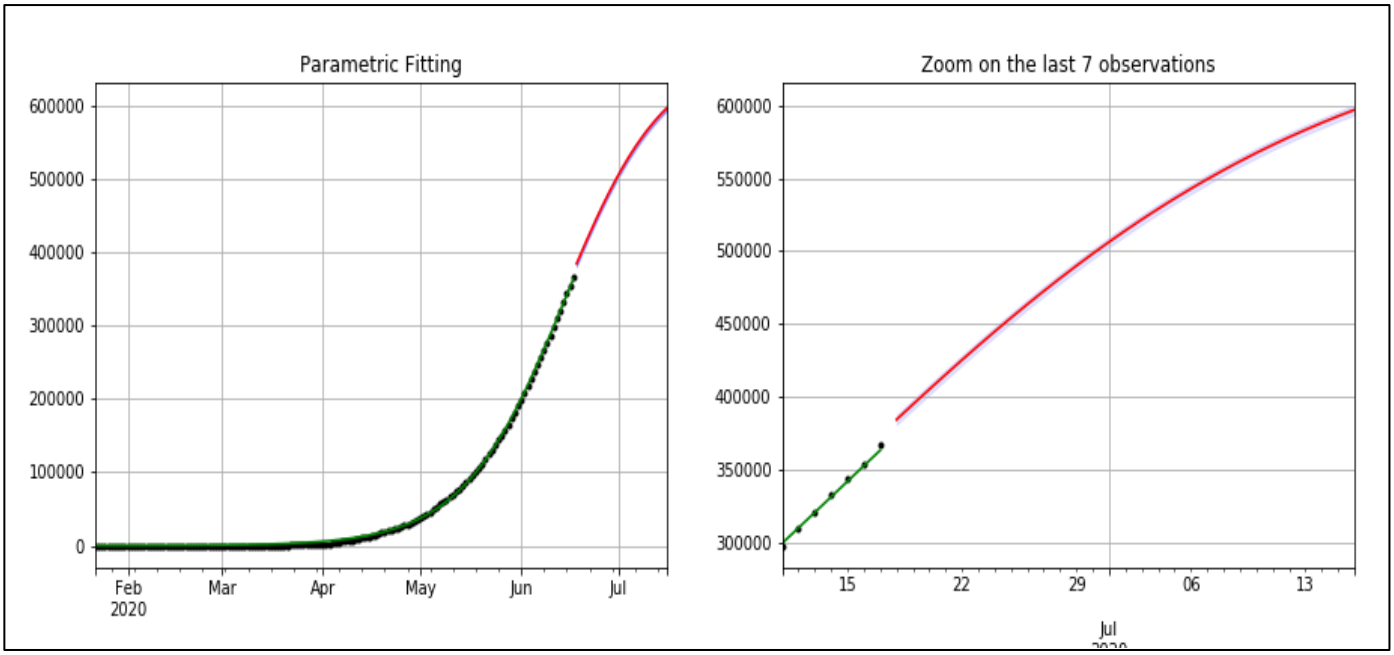
(refer Time_Series_Forecasting_with_Parametric_Curve_Fitting.ipynb)



Rise in of cases with time



Gaussian Curve fitting on no of cases with time



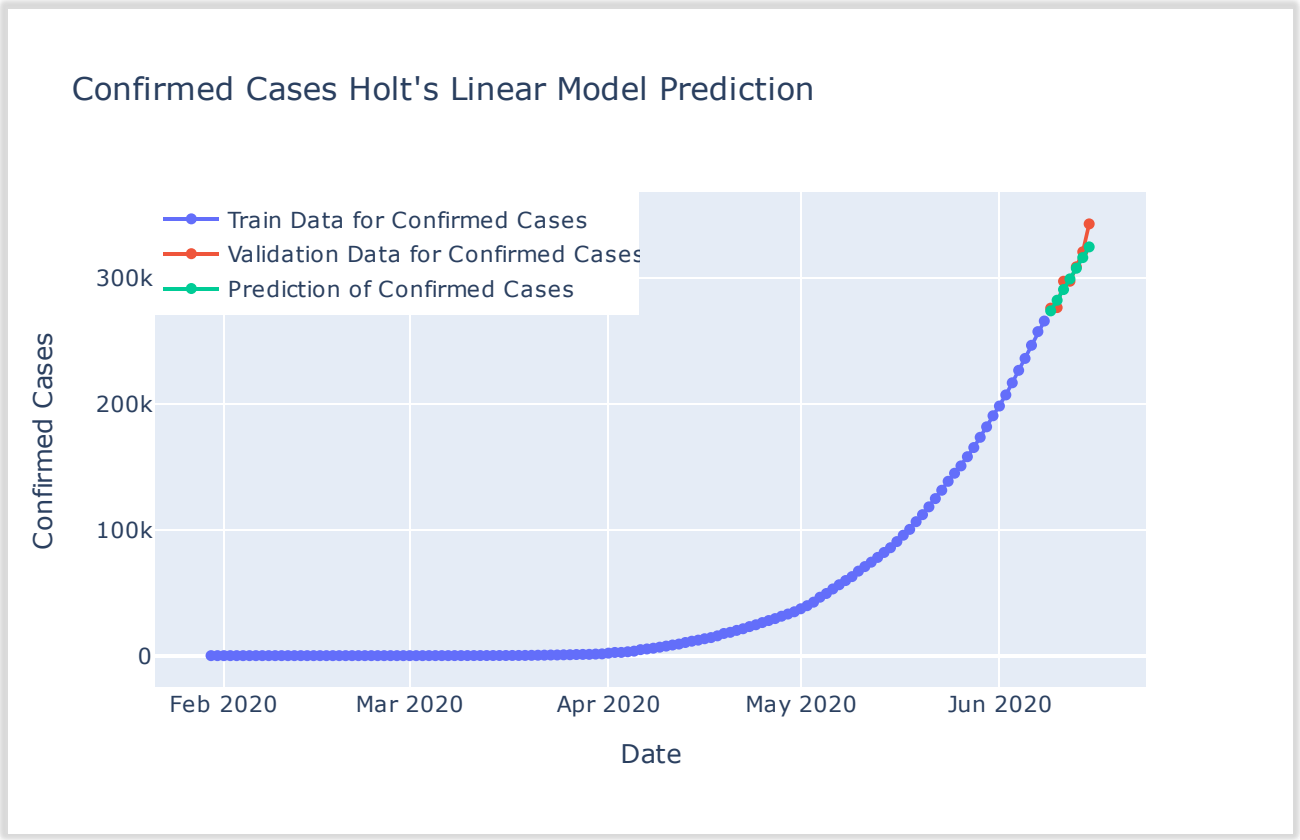
Forecasting Using Parametric fitting for India (Logistic curve Fitting)

Now we will do forecasting using Timeseries Models:

a) Holt's Linear Models:

Holt's two-parameter model, also known as linear exponential smoothing, is a popular smoothing model for forecasting data with trend. Holt's model has three separate equations that work together to generate a final forecast. The first is a basic smoothing equation that directly adjusts the last smoothed value for last period's trend. The trend itself is updated over time through the second equation, where the trend is expressed as the difference between the last two smoothed values. Finally, the third equation is used to generate the final forecast. Holt's model uses two parameters, one for the overall smoothing and the other for the trend smoothing equation. The method is also called double exponential smoothing or trend-enhanced exponential smoothing.

Model Training and fitting



Predictions using Model:

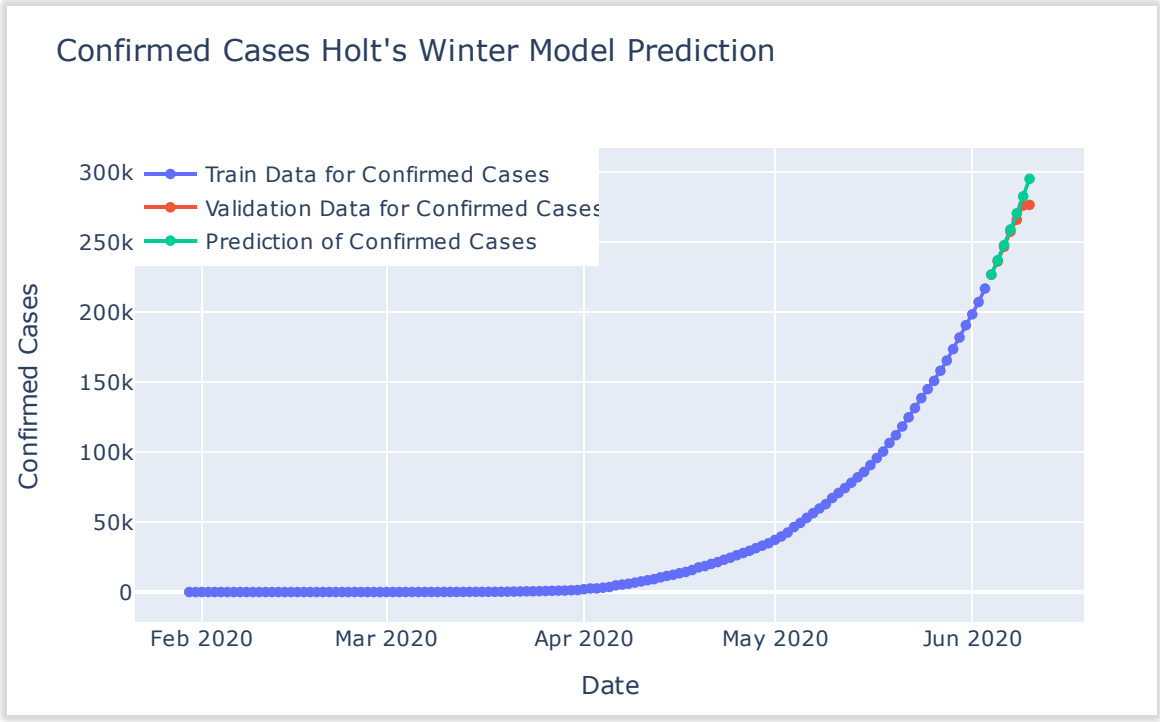
	Date	Polynomial Regression Prediction	SVM Prediction	Holt's Linear Model Prediction
0	2020-06-16	363780.706971	391314.245619	333274.975695
1	2020-06-17	377183.628229	408639.000287	341746.123005
2	2020-06-18	390873.421254	426598.283321	350217.270314
3	2020-06-19	404844.389473	445210.552046	358688.417624
4	2020-06-20	419089.898330	464494.663558	367159.564934

Root Mean Square Error Holt's Linear Model: 7955.928285231464

b) Holts-Winters model:

Holt-Winters is a model of time series behaviour. Forecasting always requires a model, and Holt-Winters is a way to model three aspects of the time series: a typical value (average), a slope (trend) over time, and a cyclical repeating pattern (seasonality). Holt-Winters uses exponential smoothing to encode lots of values from the past and use them to predict "typical" values for the present and future.

The three aspects of the time series behaviour—value, trend, and seasonality—are expressed as three types of exponential smoothing, so Holt-Winters is called triple exponential smoothing. The model predicts a current or future value by computing the combined effects of these three influences. The model requires several parameters: one for each smoothing (α, β, γ), the length of a season, and the number of periods in a season.



Cross validation of training set by Holts-Winter model

Predictions:

	Date	Holt's Winter Model Prediction
0	2020-06-16	345605.121672
1	2020-06-17	356917.271586
2	2020-06-18	368558.473047
3	2020-06-19	380540.570739
4	2020-06-20	392871.273986

Root Mean Square Error for Holt's Winter Model: 5488.957942007584

c) AR Model (Using AUTO ARIMA):

An autoregressive (AR) model **predicts future behaviour based on past behaviour**. It's used for forecasting when there is some correlation between values in a time series and the values that precede and succeed them.

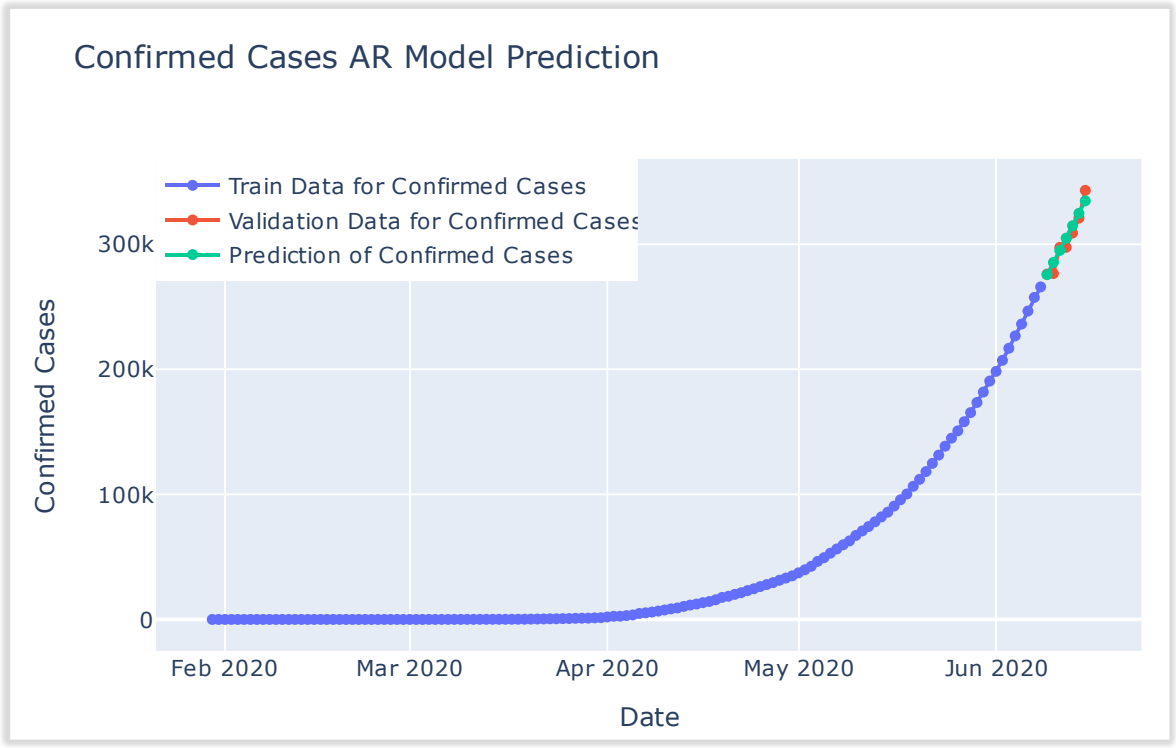
You *only* use past data to model the behaviour, hence the name *auto*regressive (the Greek prefix *auto–* means "self."). The process is basically a linear regression of the data in the current series against one or more past values in the same series.

In an AR model, the value of the outcome variable (Y) at some point *t* in time is — like "regular" linear regression — directly related to the predictor variable (X). Where simple linear regression and AR models differ is that Y is dependent on X **and previous values** for Y.

The AR process is an example of a stochastic process, which have degrees of uncertainty or randomness built in. The randomness means that you might be able to predict future trends pretty well with past data, but you're never going

to get 100 percent accuracy. Usually, the process gets “close enough” for it to be useful in most scenarios.

Training and Validation on AR model



Predictions

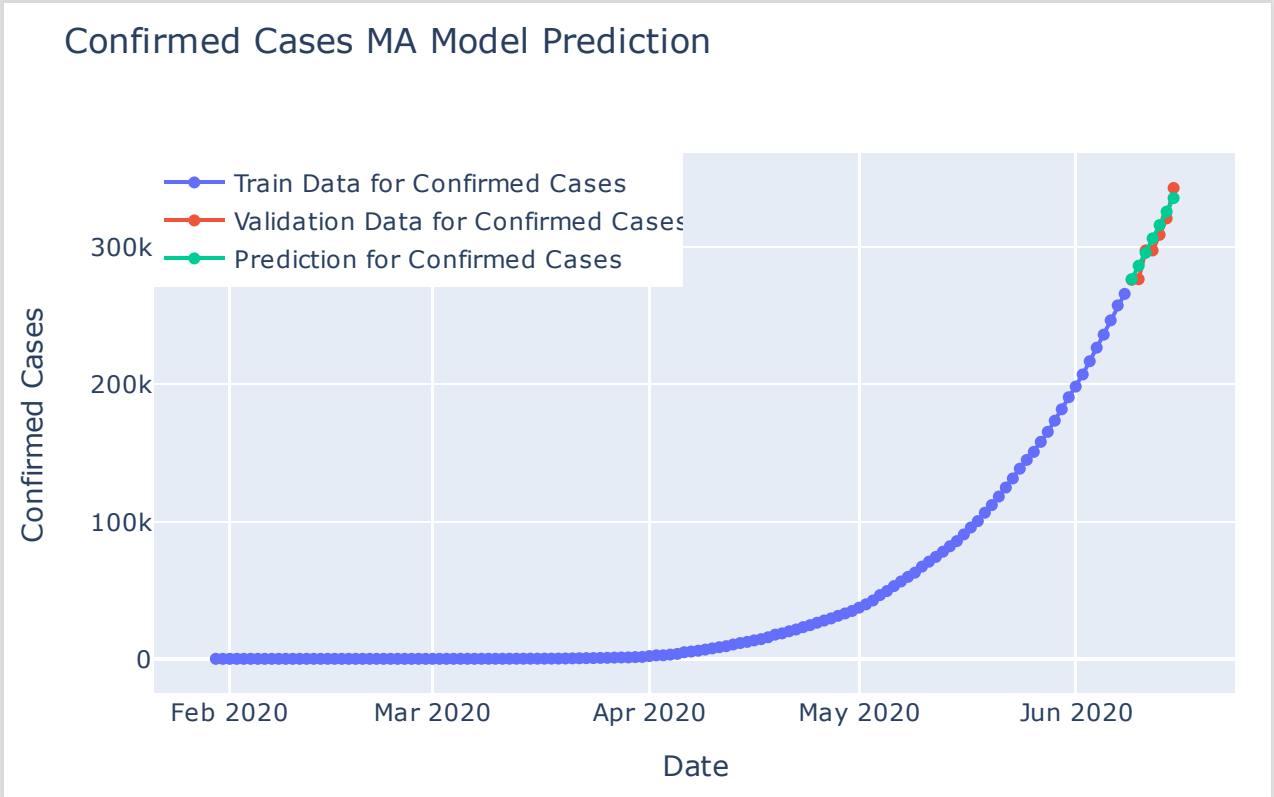
	Date	AR Model Prediction
0	2020-06-16	344827.245255
1	2020-06-17	354991.083473
2	2020-06-18	365232.696870
3	2020-06-19	375548.526331
4	2020-06-20	385938.517891

Root Mean Square Error for AR Model: 6068.65103112056

d) MA Model (Using AUTO ARIMA) or Moving Average Model:

In time series analysis, the **moving-average model** (**MA model**), also known as **moving-average process**, is a common approach for modelling univariate time series. The moving-average model specifies that the output variable depends linearly on the current and various past values of a stochastic (imperfectly predictable) term.

Together with the autoregressive (AR) model, the moving-average model is a special case and key component of the more general ARMA and ARIMA models of time series, which have a more complicated stochastic structure.



Training on MA Model

Predictions:

	Date	MA Model Prediction
0	2020-06-16	345666.160213
1	2020-06-17	355718.453139
2	2020-06-18	365857.834004
3	2020-06-19	376084.302806
4	2020-06-20	386397.859546

Root Mean Square Error for MA Model: 6605.118983074182

e) ARIMA model (Using Auto ARIMA)

ARIMA, short for 'Auto Regressive Integrated Moving Average' is actually a class of models that 'explains' a given time series based on its own past values, that is, its own lags and the lagged forecast errors, so that equation can be used to forecast future values.

Any 'non-seasonal' time series that exhibits patterns and is not a random white noise can be modelled with ARIMA models.

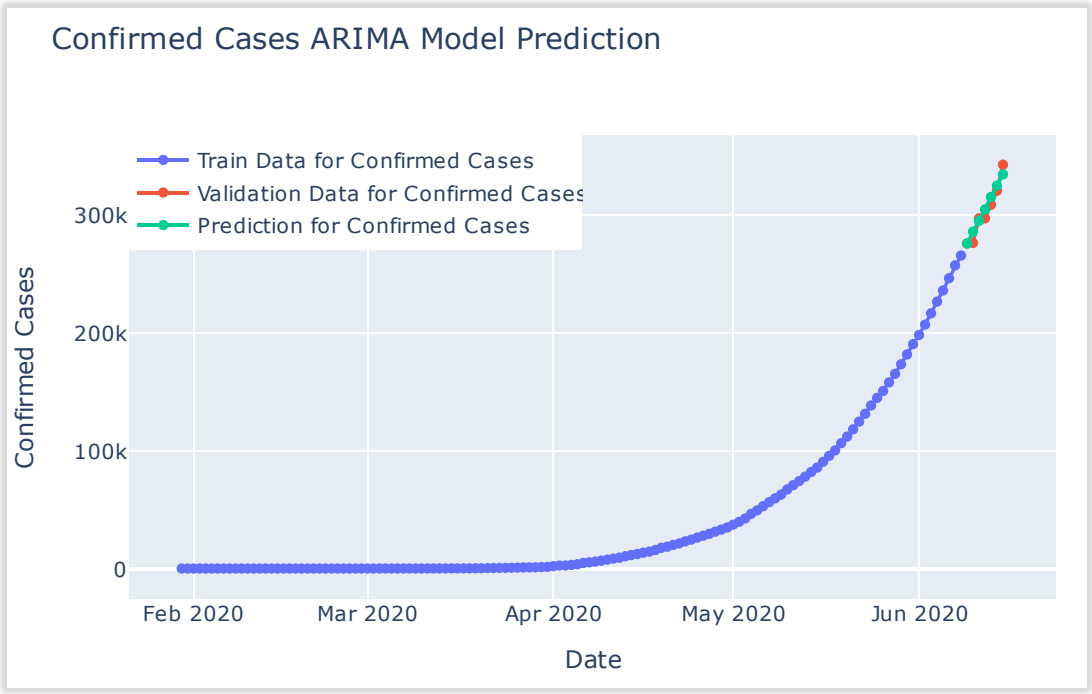
An ARIMA model is characterized by 3 terms: p, d, q where,

p is the order of the AR term
q is the order of the MA term

d is the number of differencing required to make the time series stationary

If a time series, has seasonal patterns, then you need to add seasonal terms and it becomes SARIMA, short for 'Seasonal ARIMA'. We will perform forecasting using SARIMA in Worldwide forecasting section.

Training and Cross-validation on ARIMA



Predictions:

Root Mean Square Error for ARIMA Model: 6390.2665580536295

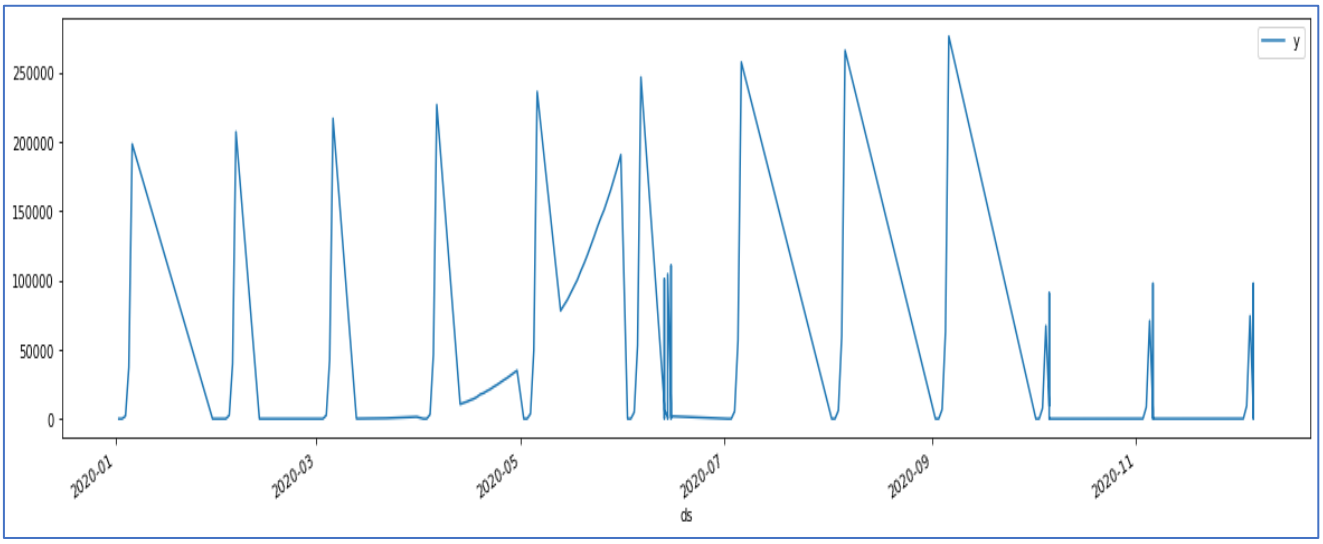
	Date	ARIMA Model Prediction
0	2020-06-16	345389.031409
1	2020-06-17	355788.715416
2	2020-06-18	365400.710446
3	2020-06-19	375690.668839
4	2020-06-20	386555.031446

f) SARIMA Model (Seasonal ARIMA)

(code for this model is taken from forecasting_with_SARIMS.ipynb. It is forecasted for India)

An extension to ARIMA that supports the direct modelling of the seasonal component of the series is called SARIMA.

It adds three new hyperparameters to specify the autoregression (AR), differencing (I) and moving average (MA) for the seasonal component of the series, as well as an additional parameter for the period of the seasonality.

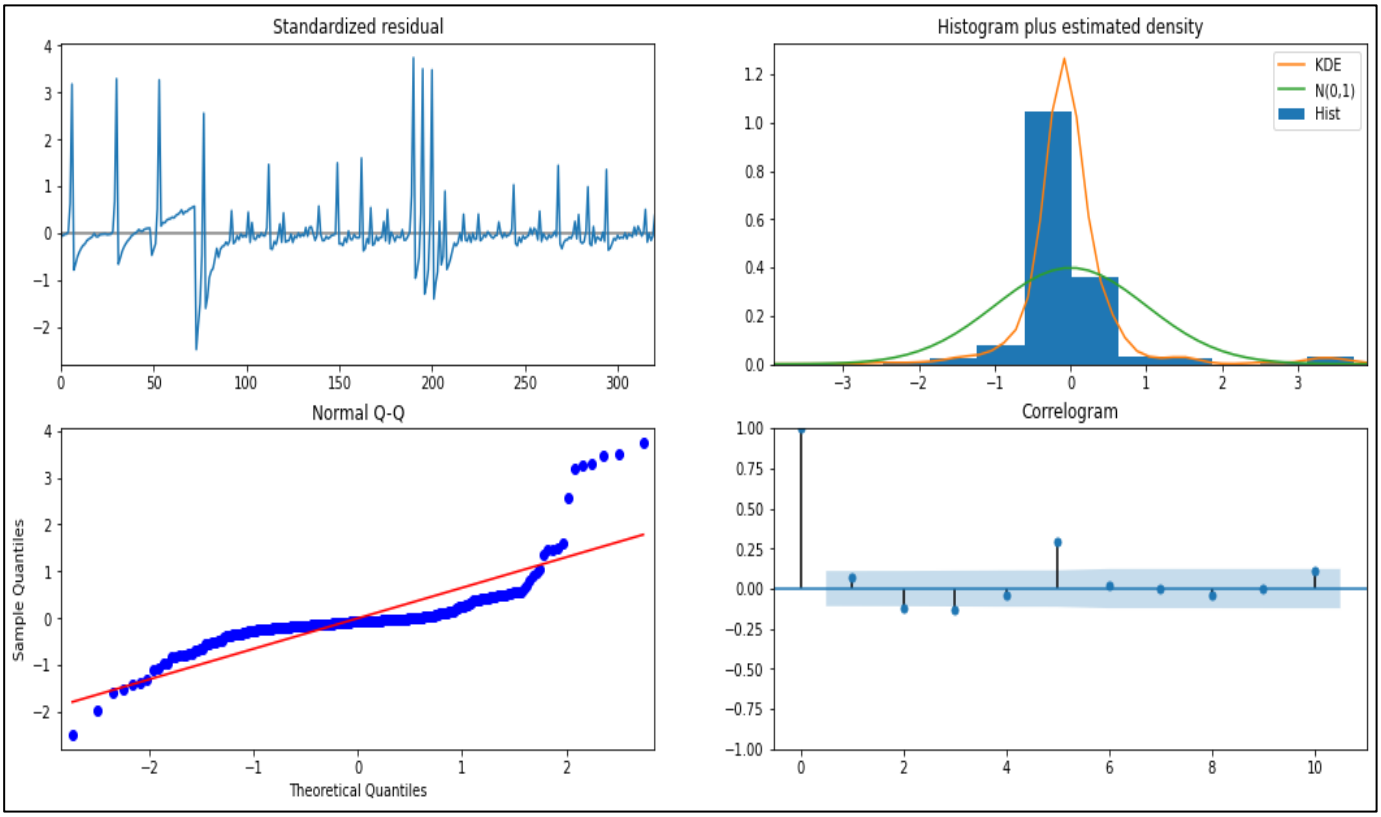


Dataset: covid_19_dataL.csv

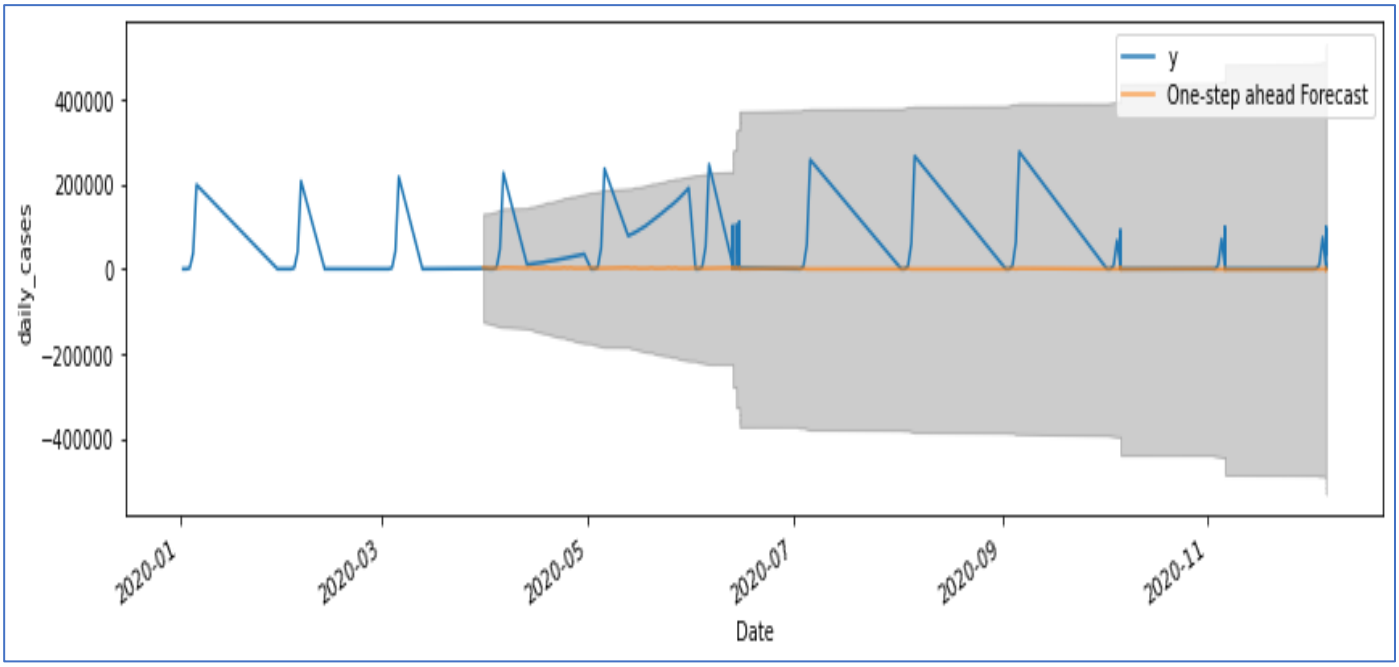
Learned parameters

```
==============================================================================
                 coef    std err          z      P>|z|      [0.025      0.975]
------------------------------------------------------------------------------
ma.L1         -0.8027      0.060    -13.454      0.000      -0.920      -0.686
ar.S.L12      -0.0255      0.156     -0.164      0.870      -0.331       0.280
ma.S.L12      -0.9929      0.034    -28.818      0.000      -1.060      -0.925
sigma2     4.249e+09    8.6e-12   4.94e+20      0.000    4.25e+09    4.25e+09
==============================================================================
```

Result Diagnostic



SARIMA Forecast



Error

MSE: 2805311490.076694
MAE: 22768.27357591518
MAEP 1562.863980757251 %

ds	Prediction (60 days)		
2020-03-31	3141.117958	2020-05-17	1223.950870
2020-04-02	1685.702578	2020-05-18	1226.515060
2020-04-03	1667.196936	2020-05-19	1488.843813
2020-04-04	1698.684872	2020-05-20	2637.609088
2020-04-05	2028.490380	2020-05-21	1219.682167
2020-04-06	3352.296978	2020-05-22	1203.778967
2020-04-13	2017.710122	2020-05-23	1236.126337
2020-04-14	3176.900263	2020-05-24	1560.283842
2020-04-15	1613.805341	2020-05-25	2852.045865
2020-04-16	1525.269760	2020-05-26	1553.706619
2020-04-17	1526.955120	2020-05-27	2685.773308
2020-04-18	1790.185492	2020-05-28	1165.587589
2020-04-19	2920.030814	2020-05-29	1081.746997
2020-04-20	1503.060631	2020-05-30	1084.311776
2020-04-21	1487.223848	2020-05-31	1346.639925
2020-04-22	1519.593151	2020-06-02	2495.417863
2020-04-23	1843.606514	2020-06-03	1077.490302
2020-04-24	3134.550732	2020-06-04	1061.587058
2020-04-25	1837.136556	2020-06-05	1093.934412
2020-04-26	2968.511031	2020-06-06	1418.092014
2020-04-27	1449.420393	2020-06-13	2709.854585
2020-04-28	1365.699621	2020-06-13	1411.514719
2020-04-29	1368.286843	2020-06-13	2543.581872
2020-04-30	1630.591966	2020-06-13	1023.395420
2020-05-02	2779.853081	2020-06-13	939.554748
2020-05-03	1361.901087	2020-06-13	942.119512
2020-05-04	1345.996146	2020-06-13	1204.447676
2020-05-05	1378.342941		
2020-05-06	1702.504224		
2020-05-13	2994.287679		
2020-05-14	1695.924189		
2020-05-15	2828.009019		
2020-05-16	1307.794602		

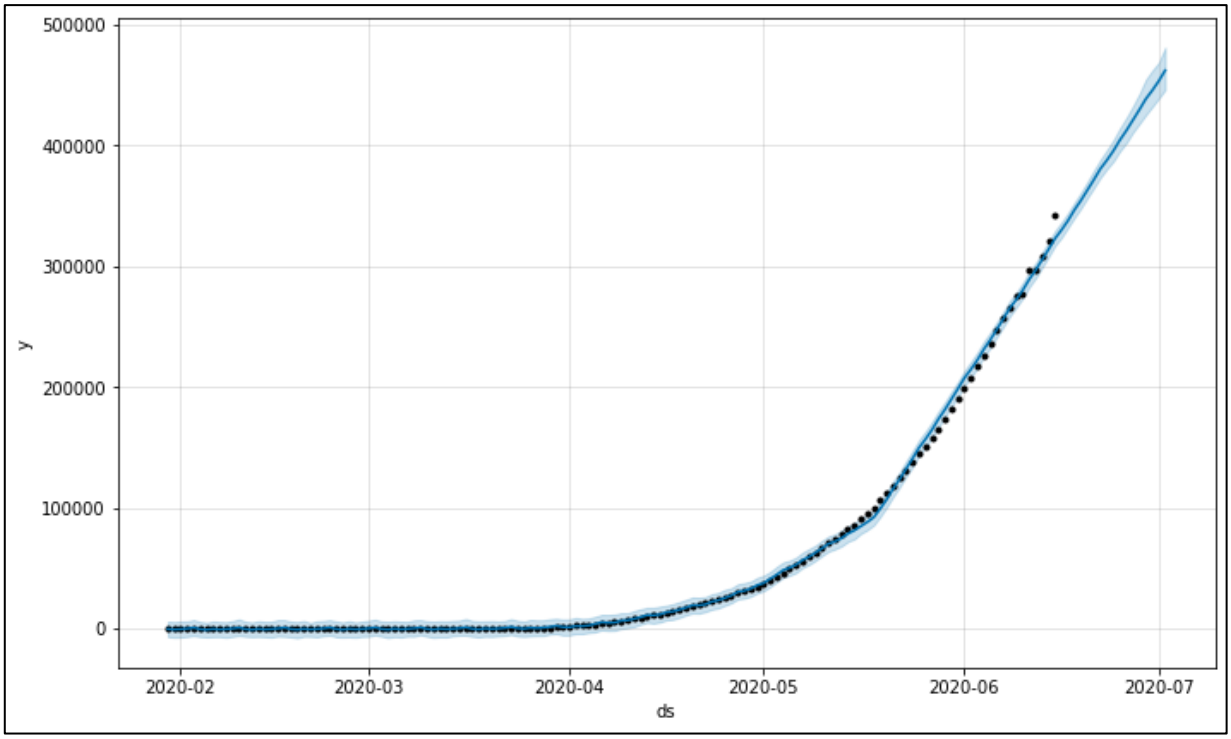
Conclusion:

It is observed after analysing situation worldwide that seasons are affecting less in transmission of COVID-19 so prediction of this model are not matching with present day situation. Hence, we can neglect the season effect of COVID-19 transmission.

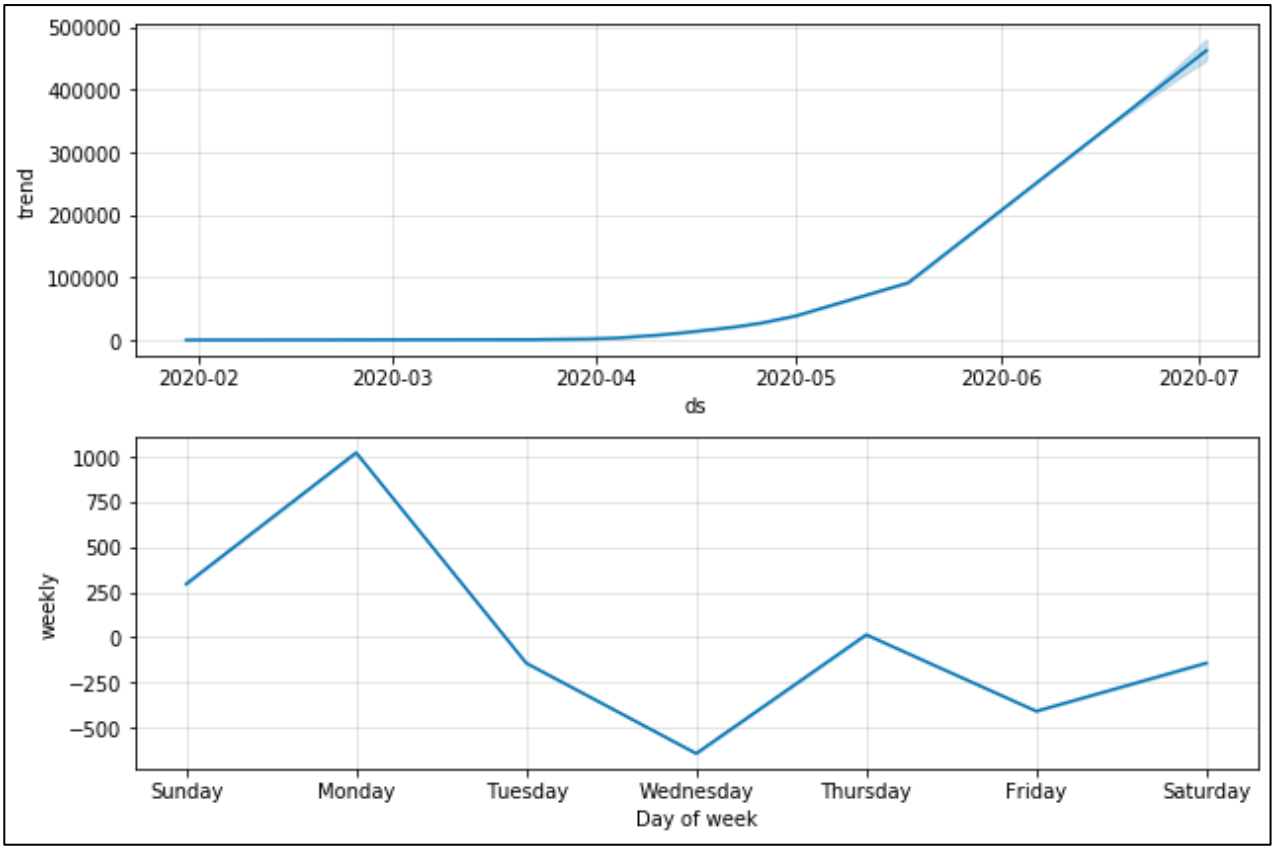
g) Facebook's Prophet Model:

Prophet is a procedure for forecasting time series data based on an additive model where non-linear trends are fit with yearly, weekly, and daily seasonality, plus holiday effects. It works best with time series that have strong seasonal effects and several seasons of historical data. Prophet is robust to missing data and shifts in the trend, and typically handles outliers well.

By default, Prophet uses a linear model for its forecast. When forecasting growth, there is usually some maximum achievable point: total market size, total population size, etc. This is called the carrying capacity, and the forecast should saturate at this point.



Training and Cross Validation on Prophet model



Forecasting using Prophet Model

Predictions:

	Date	Prophet's Prediction	Prophet's Upper Bound
0	2020-06-16	330151.749186	336702.439675
1	2020-06-17	337896.797530	344403.946543
2	2020-06-18	346797.902207	353023.507781
3	2020-06-19	354620.145485	361290.947388
4	2020-06-20	363131.008584	370294.136478

Root Mean Squared Error for Prophet Model: 3258.703268330808

Summarizing Results:

	Model Name	Root Mean Squared Error
5	Facebook's Prophet Model	3179.297315
3	Holt's Winter Model	4302.613782
0	Polynomial Regression	5569.343307
1	Support Vector Machine Regresssor	5569.343307
4	Auto Regressive Model (AR)	7697.623466
2	Holt's Linear Model	15762.561564

	Date	Polynomial Regression Prediction	SVM Prediction	Holt's Linear Model Prediction	Holt's Winter Model Prediction	AR Model Prediction
0	2020-06-16	302693.847525	321585.794049	286942.162358	308249.088535	293319.919258
1	2020-06-17	314915.647693	336368.784459	295738.189669	321820.149144	303235.590846
2	2020-06-18	327483.251333	351713.791340	304534.216979	335933.008064	313226.380958
3	2020-06-19	340399.122073	367637.779562	313330.244290	350607.616160	323292.413433
4	2020-06-20	353665.295081	384158.095203	322126.271601	365868.012620	333433.596876

	Date	MA Model Prediction	ARIMA Model Prediction	Prophet's Prediction	Prophet's Upper Bound	Average of Predictions Models
0	2020-06-16	296229.527076	299172.565004	276095.304806	282709.241958	296333.050063
1	2020-06-17	306634.607165	310599.552668	283281.491020	290065.179471	306962.132459
2	2020-06-18	317125.176408	322277.004160	290626.946583	297634.553975	317839.369978
3	2020-06-19	327701.234803	334204.501839	298147.978132	305016.974936	328926.429470
4	2020-06-20	338362.782352	346381.629625	305447.032691	312258.208331	340188.991598

Error table shows that Facebook's Prophet model has less root mean square error while Holtz linear model has high mean square error.

Comparison of obtained results with Actual Result:

	Date	Average of Predictions Models
0	2020-06-16	296333.050063
1	2020-06-17	306962.132459
2	2020-06-18	317839.369978
3	2020-06-19	328926.429470
4	2020-06-20	340188.991598

Date	Real Data (Actual No of cases in India after forecasting & data is taken from covid19-india.org)
2020-06-16	354157
2020-06-17	367265
2020-06-18	381091
2020-06-19	395831
2020-06-20	--

Conclusion:

Our prediction is close to actual number of cases. But still this table shows that actual number is quite higher than average prediction. Why does this happen? It is because COVID-19 spread depends on lot of other factors (its complex biology, population density, Health infrastructure, government management, lock down rules followed people to decrease the spread of decease) which haven't considered in these models. When no of parameters becomes large more complicate the modelling becomes. This problem can be solved by LSTM model (RNN) because this learns parameter tuning by itself. We have discussed it earlier. We can only work with RNN when we have large number of data points.

Summary:

The population and bad hygiene practices among majority of country's population are probably the most worrying concerns in accordance to COVID-19 for India. Another concern that might haunt India over and over again is lack of Medical Equipment, non-upgraded Medical technology, negligence towards Medical facilities and that might play vital role in this pandemic. Less number of Testing, unavailability of Medical Hospitals might just add up things to those worries.

The number still looks good right now considering the population and India. There is also a silver lining to it, India has been able to tackle some serious disease like Plague, Chickenpox, tuberculosis, HIV over a course, mind you which has higher Mortality Rate in comparison to COVID-19. Also, India enforced a Nationwide Lockdown at the right moment. "The unity among diversity" is another positive to take away, where people are working to help people below poverty line, people donating money to Government to fight against this pandemic which might play a significant role in this pandemic.

The course of this pandemic will be decided by the people of this country, the forecasts might look decent in comparison to other countries but that picture can change in just span of few days. It all depends on people how strictly they follow the rules and regulations imposed by the Government of India.

It will take 12-18 months for vaccine to be available on COVID-19 as per experts. Till then the only possible and effective vaccine on COVID-19 is Social Distancing at public places, Self-Isolation in case if you see any symptoms of COVID-19, Quarantine of COVID positive patients, Lockdown and TESTING ,TESTING AND TESTING!

References:

Source Code for Project:

https://www.kaggle.com/neelkudu28/covid-19-visualizations-predictions-forecasting/data?select=covid_19_data.csv

https://towardsdatascience.com/infectious-disease-modelling-beyond-the-basic-sir-model-216369c584c4

https://www.kaggle.com/frlemarchand/covid-19-forecasting-with-an-rnn/data?select=test.csv

https://medium.com/analytics-vidhya/how-to-predict-when-the-covid-19-pandemic-will-stop-in-your-country-with-python-d6fbb2425a9f

https://www.kaggle.com/deeppooja17074/covid-19-case-study-analysis-viz-comparisons/edit

https://towardsdatascience.com/building-covid-19-analysis-dashboard-using-python-and-voila-ee091f65dcbb

Other Sources:

https://www.frontiersin.org/articles/10.3389/fphy.2020.00127/full#B8

https://www.hindawi.com/journals/cmmm/2020/5714714/#data-availability

https://journals.plos.org/plosone/article?id=10.1371/journal.pone.0231236

http://jtd.amegroups.com/article/view/36385/html

https://www.kaggle.com/lisphilar/covid-19-data-with-sir-model

http://jtd.amegroups.com/article/view/36385/html

https://github.com/hf2000510/infectious_disease_modelling

geeksforgeeks.com

https://towardsdatascience.com/an-overview-of-time-series-forecasting-models-a2fa7a358fcb

https://orangematter.solarwinds.com/2019/12/15/holt-winters-forecasting-simplified/#:~:text=A%20Model%20Citizen,cyclical%20repeating%20pattern%20(seasonality).

https://www.statisticshowto.com/autoregressive-model/#:~:text=An%20autoregressive%20(AR)%20model%20predicts,that%20precede%20and%20succeed%20them.

https://www.machinelearningplus.com/time-series/arima-model-time-series-forecasting-python/#:~:text=ARIMA%2C%20short%20for%20'Auto%20Regressive,used%20to%20forecast%20future%20values.

https://www.sciencedirect.com/science/article/pii/S0960077920302642