

**A PROJECT REPORT  
ON  
Covid-19 Data Analysis using Machine Learning**



**Submitted in the partial fulfilment of award of  
BACHELOR OF TECHNOLOGY  
Degree In  
Computer Science and Engineering**

**Submitted To:**

**Mrs. SONALI BHATTACHARYA**

**Submitted By:**

**ANUPAM DUTTA  
BIROTTAM BISWAS  
RITESH SAHA  
SHUBHRANIL MAZUMDER**

# **COVID-19 DATA ANALYSIS USING MACHINE LEARNING**

B.Tech 4<sup>th</sup> Semester Project Report

Submitted towards the partial fulfillment of the requirements for the

B.Tech in Computer Science & Engineering

By

ANUPAM DUTTA	(123200803202)
BIROTTAM BISWAS	(123200803205)
RITESH SAHA	(123200803207)
SHUBHRANIL MAZUMDER	(123200803209)

Under the supervision

Of

SONALI BHATTACHARYA

Department of Computer Science & Engineering

JIS College of Engineering

Affiliated to

All India Council for Technical

University Grants Commission

Maulana Abul Kalam Azad University of Technology

NAAC, NBA, NIRF

BLOCK A, PHASE III, KALYANI, NADIA

2020 - 2021

## DECLARATION

We ANUPAM DUTTA, BIROTTAM BISWAS, RITESH SAHA, SHUBHRANIL MAZUMDAR, bearing College Roll No. 123200803202, 123200803205, 123200803207, 123200803209, Declare that the project work is an original work performed by us in the Department of Computer Science & Engineering, JIS College of Engineering, Kalyani. To complete the work, we have taken some references and are cited in the report.

Place:

Signature of student:

Date:

## CERTIFICATE OF APPROVAL

This is to certify that Anupam Dutta, College Roll No. 123200803202 has submitted the B.Tech project entitled **Covid-19 Data Analysis using Machine Learning** in partial fulfillment of the requirement for the 4<sup>th</sup> Semester B.Tech in Computer Science & Engineering of JIS COLLEGE OF ENGINEERING in the session 2020 – 2021. It is hereby approved and certified as creditable study of technological subject carried out and presented in a manner satisfactory to warrant its acceptance as a prerequisite to the B.Tech in Computer Science & Engineering for which it has been submitted.

It is understood that by the approval the undersigned does not necessarily endorse or approve any statement made. Opinion expressed or conclusion drawn therein, but approve the report only for the purpose for which has been submitted.

---

Name of Supervisor  
Designation

## **ACKNOWLEDGEMENT**

I wish to express my profound gratitude and indebtedness to Prof. SONALI BHATTACHARYA, Department of Computer Science & Engineering, JIS College of Engineering, Kalyani for introducing the present topic and for their inspiring guidance, constructive criticism and valuable suggestion throughout the project work.

Last but not least, my sincere thanks to all our friends who have patiently extended all sorts of help for accomplishing this undertaking.

ANUPAM DUTTA (123200803202)

BIROTTAM BISWAS (123200803205)

RITESH SAHA (123200803207)

SHUBHRANIL MAZUMDER (123200803209)

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

JIS COLLEGE OF ENGINEERING

BLOCK A, PHASE III, KALYANI, NADIA

## TABLE OF CONTENT

	Page no.
<i>Abstract</i>	7
Introduction to Covid-19	7
Introduction to Covid-19	7
Problem Statement	7
Technology and Concept	8
Dataset Info	10
Dataset Pre-processing	10
Building the Project	10
Confirmed Cases (INDIA)	11 - 13
Death Case (INDIA)	14 - 15
Analysis on Data & Visualization	16 – 21
Prediction and Forecasting	22 - 23
Conclusion	24

## **Abstract**

The aim of the project is to provide data analysis of covid-19 (a pandemic started in December 2019). Through plotting of data, various cases have been studied like most affected countries due to this pandemic. Study of data from various countries is combined to show the growth of cases and recovery graph. In this project, the predictions on various cases has been done and finally, the accuracy of the algorithm has been determined. Comparison graphs has also been plotted to analyse how much INDIA is getting affected/recover day by day.

## **Introduction to Covid-19**

On 31st December 2019, in the city of Wuhan (CHINA), a cluster of cases of pneumonia of unknown cause was reported to World Health organisation. In January 2020, a previously unknown new virus was identified, subsequently named 2019 novel corona virus. WHO has declared the COVID-19 as a pandemic. A pandemic is defined as disease spreadover a wide range of geographical area and that has affected high proportion of the population.

## **Problem Statement**

The pandemic has already taken grip over peoples' life. Since the start of the pandemic, some countries are facing problem of ever-increasing cases. Through the data analysis of cases one can analyse how countries all over the world are doing in terms of controlling the pandemic. Analysing data leads to adapt the prevention model of the countries that are doing great in terms of lowering the graph. Predictions are made with the dataset available to the individual/country/organisations, thus helping them to decide how far they are able to control the pandemic or up to how much extent they should guide preventive measures.

Through this project, a step towards helping people to understand the spread and predict the cases in their country is done. This project also gives an insight of how a country is doing in terms of limiting the spread.

# Technology and Concept

## Machine Learning

Machine learning is a field of study or process of teaching a computer to learn the fed data without being explicitly programmed. It makes computer make decisions similar to humans.

Now a days, it is actively being used in various field. E.g. Medical, Industries, Astronomy etc. The major types of Machine learning are Supervised Learning, Unsupervised Learning and Reinforcement Learning.

## Supervised Learning

The machine learning task of learning a function that can map an input data to output data and performs analysis based on that input-output pair.

## Unsupervised Learning

A type of machine learning that draw an inference from dataset consisting of input data without labelled responses. One of the common unsupervised learning methods called cluster analysis, is used find the hidden pattern or grouping of data.

## Reinforcement Learning

A type of machine learning that is bound to learn from experiences. There is no training dataset provided \*(such methods work in the absence of datasets). An agent in Reinforcement learning that rewards or penalise for actions done by the algorithm. The task is to find the best possible path to reach the goal.

## Some important terms

### Data frame

Pandas Data frame is 2D, mutable and heterogeneous tabular data structure with labelled axes. Data frame can be made of more than one series (series can only contain single list with index).

### Hypothesis

In Machine learning, Hypothesis is a model that is used to approximate the target function and performs mapping of input with output.

### Regression

Regression in Machine Learning is about predicting the continuous value-based learning gained by dataset. The correctness of the output can depend on the size of dataset, features, hypothesis used etc.



## **Classification**

The problem of identifying that in which sub-population a new example/observation belongs to, on the basis of learning obtained through training set containing observations along with the category they belong to.

# Building the Project

## Dataset Pre-processing

This section include the parsing of date in a proper readable format, renaming some columns into short and descriptive names , adding new column 'active cases' with the help of other cases available in the dataset , creating a data frame that includes the latest cases up to date , grouping the data in terms of country and resetting the index .

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

```
In [2]: #importing main dataset
        df = pd.read_csv('covid_19_india.csv', parse_dates=['Date'], dayfirst=True)
```

## Prediction and Forecasting for India

```
In [3]: df.head()
```

```
Out[3]:
```

	Sno	Date	Time	State/UnionTerritory	ConfirmedIndianNational	ConfirmedForeignNational	Cured	Deaths	Confirmed
0	1	2020-01-30	6:00 PM	Kerala	1	0	0	0	1
1	2	2020-01-31	6:00 PM	Kerala	1	0	0	0	1
2	3	2020-02-01	6:00 PM	Kerala	2	0	0	0	2
3	4	2020-02-02	6:00 PM	Kerala	3	0	0	0	3
4	5	2020-02-03	6:00 PM	Kerala	3	0	0	0	3

```
In [4]: #keeping only required columns
        df = df[['Date', 'State/UnionTerritory', 'Cured', 'Deaths', 'Confirmed']]
        #renaming column names
        df.columns = ['date', 'state', 'cured', 'deaths', 'confirmed']
```

```
In [5]: #Looking at the earlier dates
        df.head()
```

```
Out[5]:
```

	date	state	cured	deaths	confirmed
0	2020-01-30	Kerala	0	0	1
1	2020-01-31	Kerala	0	0	1
2	2020-02-01	Kerala	0	0	2
3	2020-02-02	Kerala	0	0	3
4	2020-02-03	Kerala	0	0	3

# Confirmed Cases (INDIA)

```
In [7]: #current date
today = df[df.date == '2021-07-05']
```

```
In [8]: today
```

```
Out[8]:
```

	date	state	cured	deaths	confirmed
16742	2021-07-05	Andaman and Nicobar Islands	7333	128	7482
16743	2021-07-05	Andhra Pradesh	1854754	12844	1902923
16744	2021-07-05	Arunachal Pradesh	33967	177	37105
16745	2021-07-05	Assam	489040	4652	517194
16746	2021-07-05	Bihar	711490	9601	722527
16747	2021-07-05	Chandigarh	60777	808	61728
16748	2021-07-05	Chhattisgarh	976917	13456	995718
16749	2021-07-05	Dadra and Nagar Haveli and Daman and Diu	10527	4	10569
16750	2021-07-05	Delhi	1408567	24995	1434554
16751	2021-07-05	Goa	162276	3073	167436
16752	2021-07-05	Gujarat	811297	10069	823833
16753	2021-07-05	Haryana	758231	9486	768903
16754	2021-07-05	Himachal Pradesh	197794	3483	202642
16755	2021-07-05	Jammu and Kashmir	308672	4337	316976
16756	2021-07-05	Jharkhand	340164	5115	345937
16757	2021-07-05	Karnataka	2773407	35367	2853643
16758	2021-07-05	Kerala	2855460	13716	2973684
16759	2021-07-05	Ladakh	19690	204	20120
16760	2021-07-05	Lakshadweep	9577	49	9900
16761	2021-07-05	Madhya Pradesh	780495	9009	789983
16762	2021-07-05	Maharashtra	5848693	123030	6098177
16763	2021-07-05	Manipur	64931	1196	72286
16764	2021-07-05	Meghalaya	46228	862	51524
16765	2021-07-05	Mizoram	17661	95	21337
16766	2021-07-05	Nagaland	23786	499	25519
16767	2021-07-05	Odisha	890778	4196	921896
16768	2021-07-05	Puducherry	114192	1761	117959
16769	2021-07-05	Punjab	577982	16110	596416
16770	2021-07-05	Rajasthan	942616	8938	952734
16771	2021-07-05	Sikkim	18722	308	21131
16772	2021-07-05	Tamil Nadu	2427988	33005	2496287
16773	2021-07-05	Telangana	611035	3691	626690
16774	2021-07-05	Tripura	63209	692	67677
16775	2021-07-05	Uttarakhand	331642	7333	340724
16776	2021-07-05	Uttar Pradesh	1681717	22640	1706621
16777	2021-07-05	West Bengal	1468815	17799	1505394

```
In [6]: #looking at the latest dates
df.tail()
```

```
Out[6]:
```

	date	state	cured	deaths	confirmed
16845	2021-07-07	Telangana	613124	3703	628282
16846	2021-07-07	Tripura	63964	701	68612
16847	2021-07-07	Uttarakhand	332006	7338	340882
16848	2021-07-07	Uttar Pradesh	1682130	22656	1706818
16849	2021-07-07	West Bengal	1472132	17834	1507241

```
In [9]: #Sorting data w.r.t number of confirmed cases
max_confirmed_cases= today.sort_values(by="confirmed",ascending=False)
max_confirmed_cases
```

```
Out[9]:
```

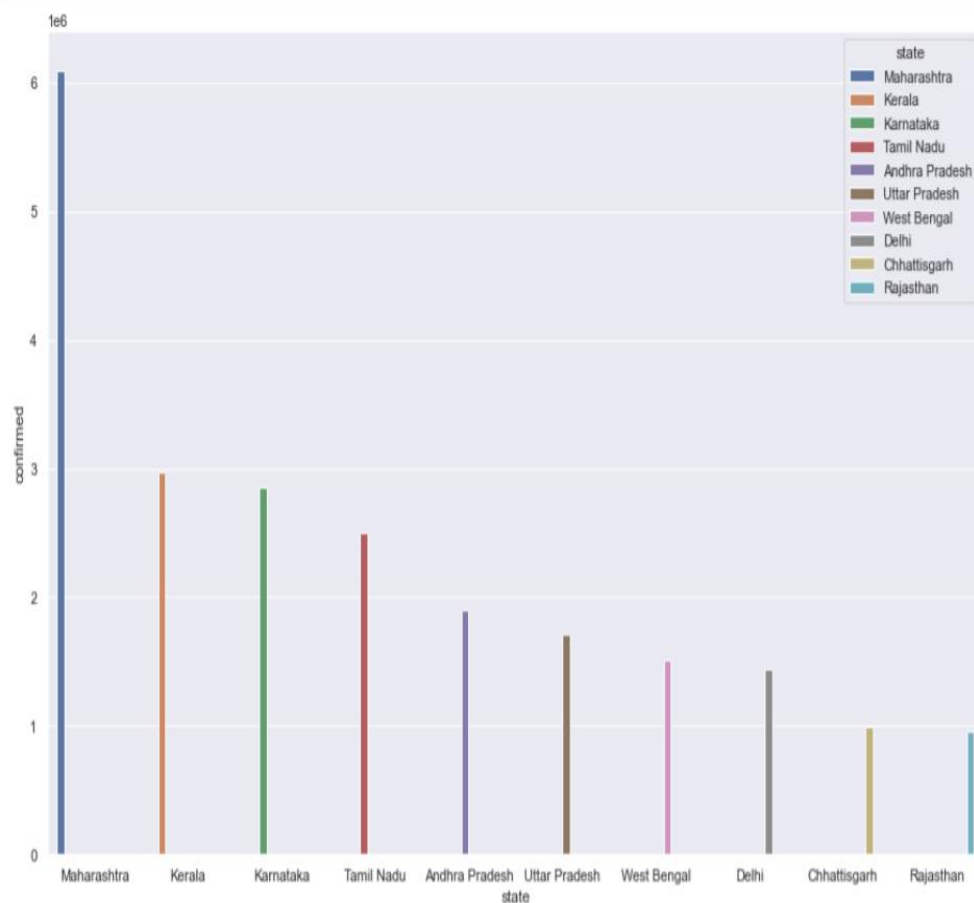
	date	state	cured	deaths	confirmed
16762	2021-07-05	Maharashtra	5848693	123030	6098177
16758	2021-07-05	Kerala	2855460	13716	2973684
16757	2021-07-05	Karnataka	2773407	35367	2853643
16772	2021-07-05	Tamil Nadu	2427988	33005	2496287
16743	2021-07-05	Andhra Pradesh	1854754	12844	1902923
16776	2021-07-05	Uttar Pradesh	1681717	22640	1706621
16777	2021-07-05	West Bengal	1468815	17799	1505394
16750	2021-07-05	Delhi	1408567	24995	1434554
16748	2021-07-05	Chhattisgarh	976917	13456	995718
16770	2021-07-05	Rajasthan	942616	8938	952734
16767	2021-07-05	Odisha	890778	4196	921896
16752	2021-07-05	Gujarat	811297	10069	823833
16761	2021-07-05	Madhya Pradesh	780495	9009	789983
16753	2021-07-05	Haryana	758231	9486	768903
16746	2021-07-05	Bihar	711490	9601	722527
16773	2021-07-05	Telangana	611035	3691	626690
16769	2021-07-05	Punjab	577982	16110	596416
16745	2021-07-05	Assam	489040	4652	517194
16756	2021-07-05	Jharkhand	340164	5115	345937
16775	2021-07-05	Uttarakhand	331642	7333	340724
16755	2021-07-05	Jammu and Kashmir	308672	4337	316976
16754	2021-07-05	Himachal Pradesh	197794	3483	202642
16751	2021-07-05	Goa	162276	3073	167436

16768	2021-07-05	Puducherry	114192	1761	117959
16763	2021-07-05	Manipur	64931	1196	72286
16774	2021-07-05	Tripura	63209	692	67677
16747	2021-07-05	Chandigarh	60777	808	61728
16764	2021-07-05	Meghalaya	46228	862	51524
16744	2021-07-05	Arunachal Pradesh	33967	177	37105
16766	2021-07-05	Nagaland	23786	499	25519
16765	2021-07-05	Mizoram	17661	95	21337
16771	2021-07-05	Sikkim	18722	308	21131
16759	2021-07-05	Ladakh	19690	204	20120
16749	2021-07-05	Dadra and Nagar Haveli and Daman and Diu	10527	4	10569
16760	2021-07-05	Lakshadweep	9577	49	9900
16742	2021-07-05	Andaman and Nicobar Islands	7333	128	7482

16768	2021-07-05	Puducherry	114192	1761	117959
16763	2021-07-05	Manipur	64931	1196	72286
16774	2021-07-05	Tripura	63209	692	67677
16747	2021-07-05	Chandigarh	60777	808	61728
16764	2021-07-05	Meghalaya	46228	862	51524
16744	2021-07-05	Arunachal Pradesh	33967	177	37105
16766	2021-07-05	Nagaland	23786	499	25519
16765	2021-07-05	Mizoram	17661	95	21337
16771	2021-07-05	Sikkim	18722	308	21131
16759	2021-07-05	Ladakh	19690	204	20120
16749	2021-07-05	Dadra and Nagar Haveli and Daman and Diu	10527	4	10569
16760	2021-07-05	Lakshadweep	9577	49	9900
16742	2021-07-05	Andaman and Nicobar Islands	7333	128	7482

```
In [10]: #Getting states with maximum number of confirmed cases
top_states_confirmed=max_confirmed_cases[0:10]
```

```
In [11]: #Making bar-plot for states with top confirmed cases
sns.set(rc={'figure.figsize':(15,10)})
sns.barplot(x="state",y="confirmed",data=top_states_confirmed,hue="state")
plt.show()
```



# Death Cases (INDIA)

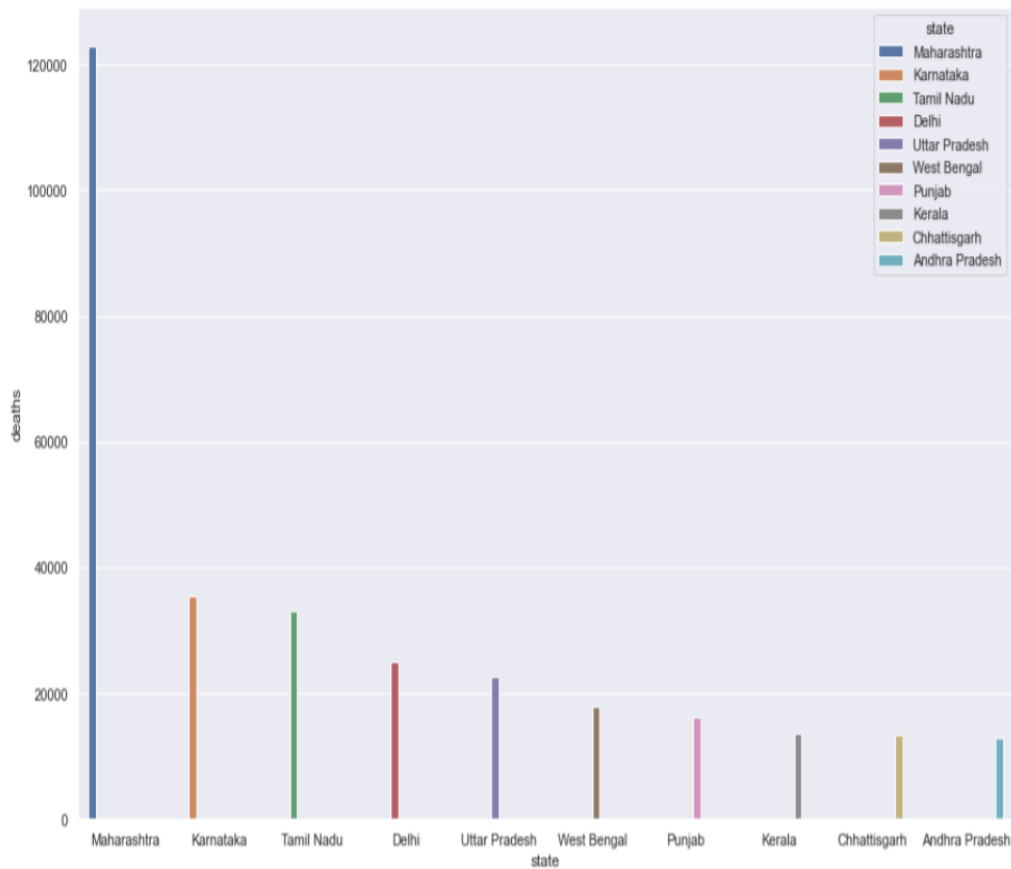
```
In [12]: #Sorting data w.r.t number of death cases
max_death_cases=today.sort_values(by="deaths", ascending=False)
max_death_cases
```

	date	state	cured	deaths	confirmed
16762	2021-07-05	Maharashtra	5848693	123030	6098177
16757	2021-07-05	Karnataka	2773407	35367	2853643
16772	2021-07-05	Tamil Nadu	2427988	33005	2496287
16750	2021-07-05	Delhi	1408567	24995	1434554
16776	2021-07-05	Uttar Pradesh	1681717	22640	1706621
16777	2021-07-05	West Bengal	1468815	17799	1505394
16769	2021-07-05	Punjab	577982	16110	596416
16758	2021-07-05	Kerala	2855460	13716	2973684
16748	2021-07-05	Chhattisgarh	976917	13456	995718
16743	2021-07-05	Andhra Pradesh	1854754	12844	1902923
16752	2021-07-05	Gujarat	811297	10069	823833
16746	2021-07-05	Bihar	711490	9601	722527
16753	2021-07-05	Haryana	758231	9486	768903
16761	2021-07-05	Madhya Pradesh	780495	9009	789983
16770	2021-07-05	Rajasthan	942616	8938	952734
16775	2021-07-05	Uttarakhand	331642	7333	340724
16756	2021-07-05	Jharkhand	340164	5115	345937
16745	2021-07-05	Assam	489040	4652	517194
16755	2021-07-05	Jammu and Kashmir	308672	4337	316976
16767	2021-07-05	Odisha	890778	4196	921896
16773	2021-07-05	Telangana	611035	3691	626690
16754	2021-07-05	Himachal Pradesh	197794	3483	202642
16751	2021-07-05	Goa	162276	3073	167436

16768	2021-07-05	Puducherry	114192	1761	117959
16763	2021-07-05	Manipur	64931	1196	72286
16764	2021-07-05	Meghalaya	46228	862	51524
16747	2021-07-05	Chandigarh	60777	808	61728
16774	2021-07-05	Tripura	63209	692	67677
16766	2021-07-05	Nagaland	23786	499	25519
16771	2021-07-05	Sikkim	18722	308	21131
16759	2021-07-05	Ladakh	19690	204	20120
16744	2021-07-05	Arunachal Pradesh	33967	177	37105
16742	2021-07-05	Andaman and Nicobar Islands	7333	128	7482
16765	2021-07-05	Mizoram	17661	95	21337
16760	2021-07-05	Lakshadweep	9577	49	9900
16749	2021-07-05	Dadra and Nagar Haveli and Daman and Diu	10527	4	10569

```
In [13]: #Getting states with maximum number of death cases
top_states_death=max_death_cases[0:10]
```

```
In [14]: #Making bar-plot for states with top death cases
sns.set(rc={'figure.figsize':(15,10)})
sns.barplot(x="state",y="deaths",data=top_states_death,hue="state")
plt.show()
```



```
In [15]: #Sorting data w.r.t number of cured cases
max_cured_cases=today.sort_values(by="cured",ascending=False)
max_cured_cases
```

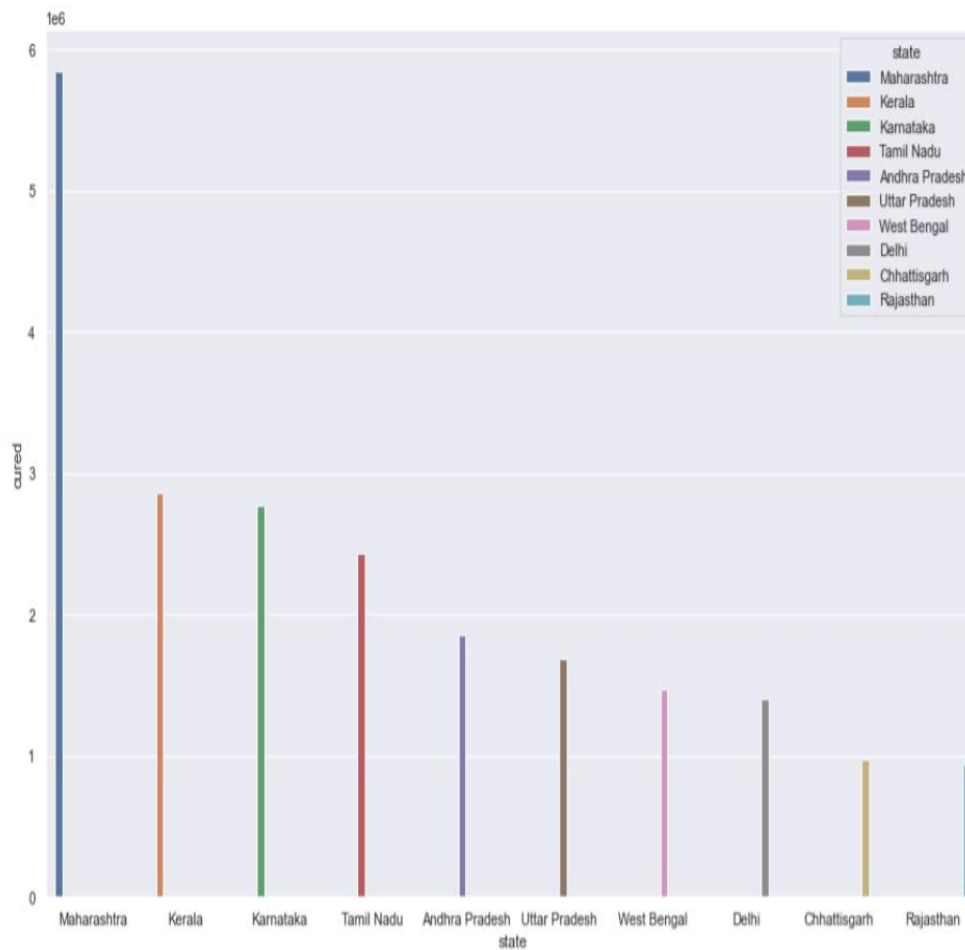
```
Out[15]:
```

	date	state	cured	deaths	confirmed
16762	2021-07-05	Maharashtra	5848693	123030	6098177
16758	2021-07-05	Kerala	2855460	13716	2973684
16757	2021-07-05	Karnataka	2773407	35367	2853643
16772	2021-07-05	Tamil Nadu	2427988	33005	2496287
16743	2021-07-05	Andhra Pradesh	1854754	12844	1902923
16776	2021-07-05	Uttar Pradesh	1681717	22640	1706621
16777	2021-07-05	West Bengal	1468815	17799	1505394
16750	2021-07-05	Delhi	1408567	24995	1434554
16748	2021-07-05	Chhattisgarh	976917	13456	995718
16770	2021-07-05	Rajasthan	942616	8938	952734
16767	2021-07-05	Odisha	890778	4196	921896
16752	2021-07-05	Gujarat	811297	10069	823833
16761	2021-07-05	Madhya Pradesh	780495	9009	789983
16753	2021-07-05	Haryana	758231	9486	768903
16746	2021-07-05	Bihar	711490	9601	722527
16773	2021-07-05	Telangana	611035	3691	626690
16769	2021-07-05	Punjab	577982	16110	596416
16745	2021-07-05	Assam	489040	4652	517194
16756	2021-07-05	Jharkhand	340164	5115	345937
16775	2021-07-05	Uttarakhand	331642	7333	340724
16755	2021-07-05	Jammu and Kashmir	308672	4337	316976
16754	2021-07-05	Himachal Pradesh	197794	3483	202642



```
In [16]: #Getting states with maximum number of cured cases
top_states_cured=max_cured_cases[0:10]
```

```
In [17]: #Making bar-plot for states with top death cases
sns.set(rc={'figure.figsize':(15,10)})
sns.barplot(x="state",y="cured",data=top_states_cured,hue="state")
plt.show()
```



16763	2021-07-05	Manipur	64931	1196	72286
16774	2021-07-05	Tripura	63209	692	67677
16747	2021-07-05	Chandigarh	60777	808	61728
16764	2021-07-05	Meghalaya	46228	862	51524
16744	2021-07-05	Arunachal Pradesh	33967	177	37105
16766	2021-07-05	Nagaland	23786	499	25519
16759	2021-07-05	Ladakh	19690	204	20120
16771	2021-07-05	Sikkim	18722	308	21131
16765	2021-07-05	Mizoram	17661	95	21337
16749	2021-07-05	Dadra and Nagar Haveli and Daman and Diu	10527	4	10569
16760	2021-07-05	Lakshadweep	9577	49	9900
16742	2021-07-05	Andaman and Nicobar Islands	7333	128	7482

```
In [18]: #Maharashtra
maha = df[df.state == 'Maharashtra']
```

```
In [19]: maha
```

```
Out[19]:
```

	date	state	cured	deaths	confirmed
76	2020-03-09	Maharashtra	0	0	2
91	2020-03-10	Maharashtra	0	0	5
97	2020-03-11	Maharashtra	0	0	2
120	2020-03-12	Maharashtra	0	0	11
133	2020-03-13	Maharashtra	0	0	14
...	...	...	...	...	...
16690	2021-07-03	Maharashtra	5836920	122353	6079352
16726	2021-07-04	Maharashtra	5845315	122724	6088841
16762	2021-07-05	Maharashtra	5848693	123030	6098177
16798	2021-07-06	Maharashtra	5861720	123136	6104917
16834	2021-07-07	Maharashtra	5872268	123531	6113335

Recovered Cases (INDIA)  
486 rows × 5 columns

```
In [20]: #Maharashtra
wb = df[df.state == 'West Bengal']
```

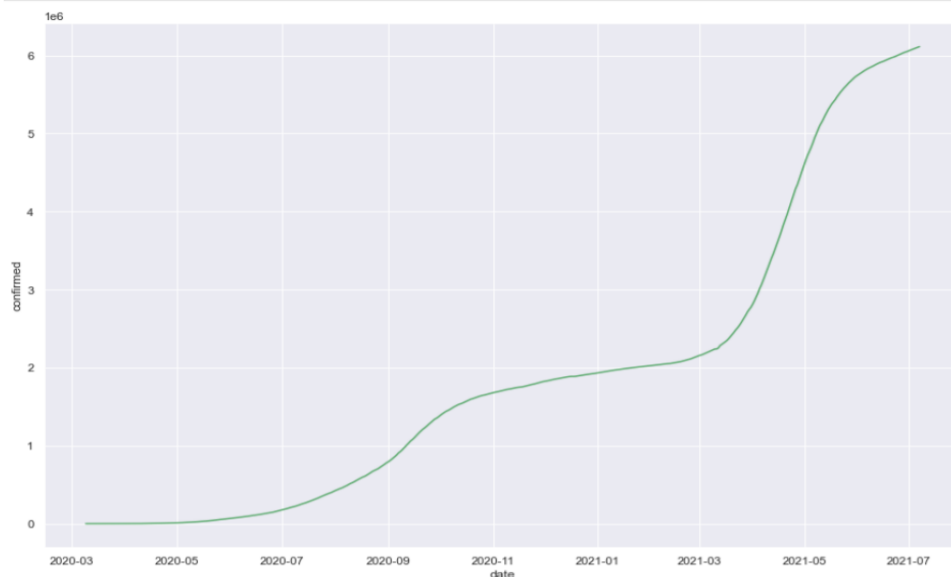
```
In [21]: wb
```

```
Out[21]:
```

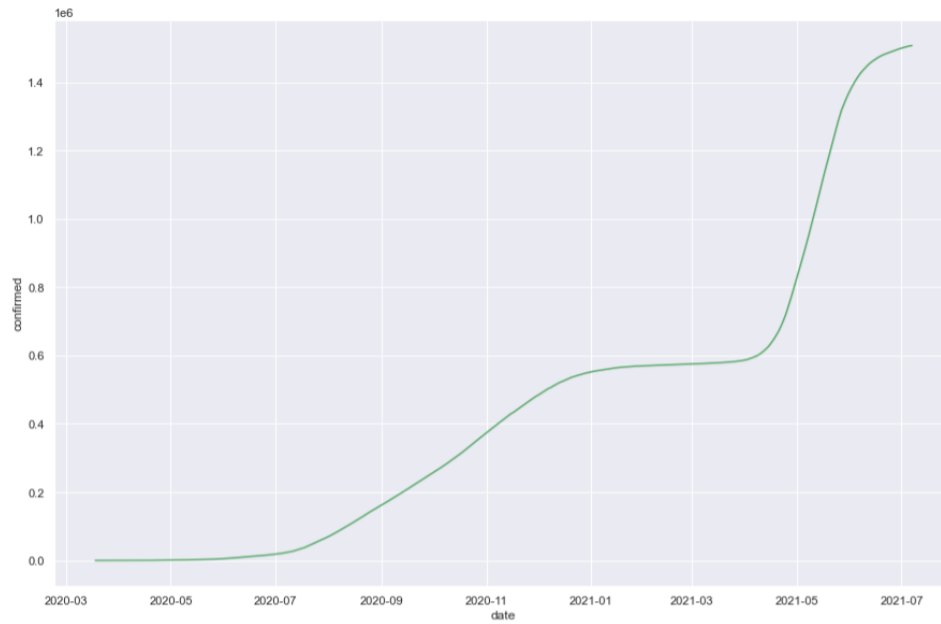
	date	state	cured	deaths	confirmed
208	2020-03-18	West Bengal	0	0	1
227	2020-03-19	West Bengal	0	0	1
247	2020-03-20	West Bengal	0	0	2
269	2020-03-21	West Bengal	0	0	3
292	2020-03-22	West Bengal	0	0	4
...	...	...	...	...	...
16705	2021-07-03	West Bengal	1465219	17758	1502706
16741	2021-07-04	West Bengal	1467038	17779	1504097
16777	2021-07-05	West Bengal	1468815	17799	1505394
16813	2021-07-06	West Bengal	1470512	17817	1506279
16849	2021-07-07	West Bengal	1472132	17834	1507241

477 rows × 5 columns

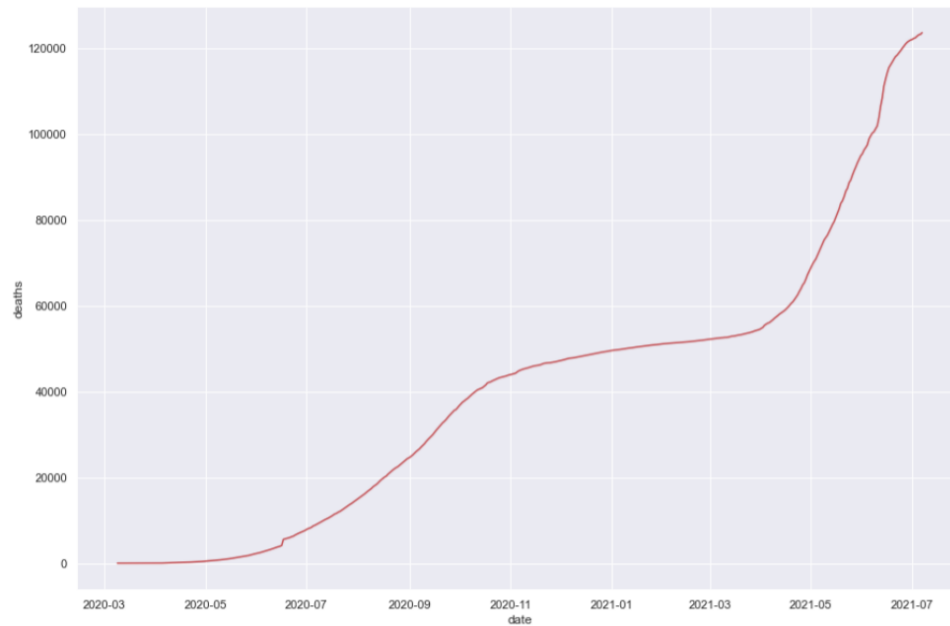
```
In [22]: #Visualizing confirmed cases in maharashtra
sns.set(rc={'figure.figsize':(15,10)})
sns.lineplot(x="date",y="confirmed",data=maha,color="g")
plt.show()
```



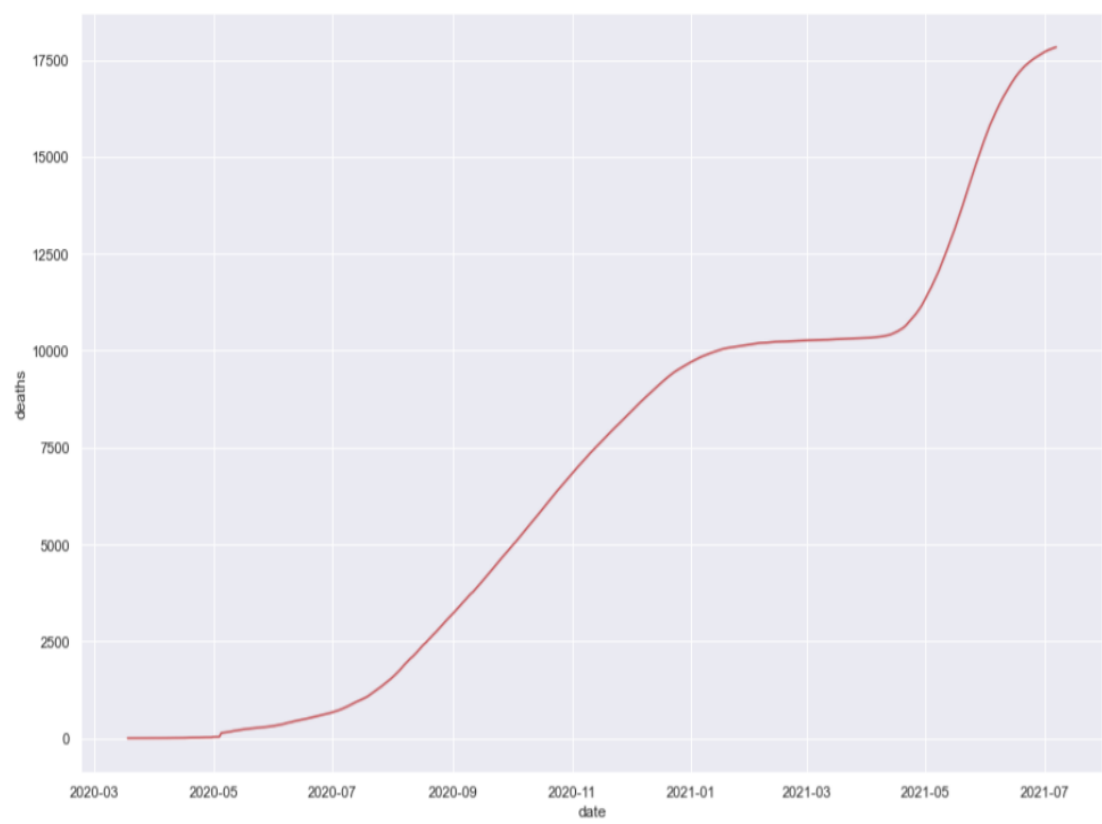
```
In [23]: #Visualizing confirmed cases in West Bengal
sns.set(rc={'figure.figsize':(15,10)})
sns.lineplot(x="date",y="confirmed",data=wb,color="g")
plt.show()
```



```
In [24]: #Visualizing death cases in maharashtra
sns.set(rc={'figure.figsize':(15,10)})
sns.lineplot(x="date",y="deaths",data=maha,color="r")
plt.show()
```



```
In [45]: #Visualizing death cases in West Bengal
sns.set(rc={'figure.figsize':(15,10)})
sns.lineplot(x="date",y="deaths",data=wb,color="r")
plt.show()
```



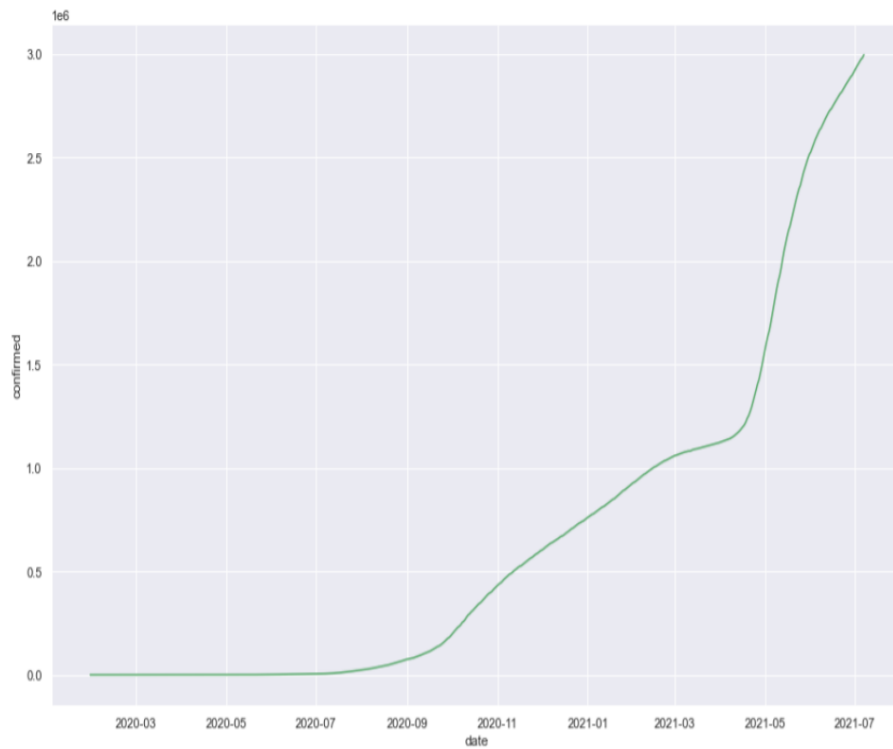
```
In [26]: #kerala
kerala= df[df.state == 'Kerala']
kerala
```

Out[26]:

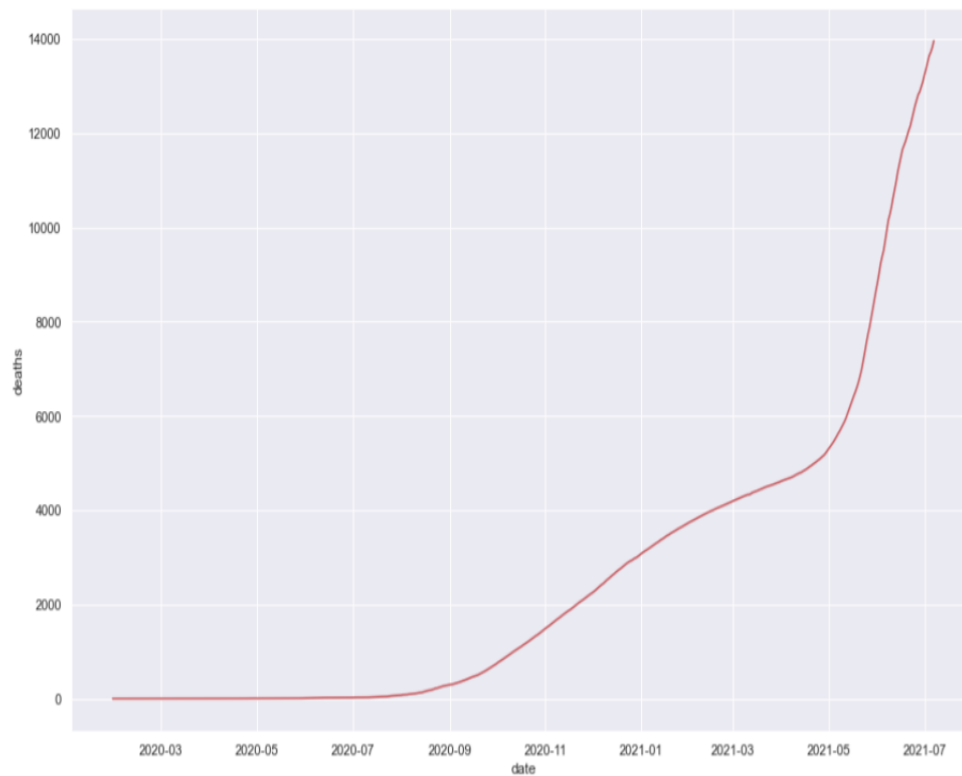
	date	state	cured	deaths	confirmed
0	2020-01-30	Kerala	0	0	1
1	2020-01-31	Kerala	0	0	1
2	2020-02-01	Kerala	0	0	2
3	2020-02-02	Kerala	0	0	3
4	2020-02-03	Kerala	0	0	3
...	...	...	...	...	...
16686	2021-07-03	Kerala	2831394	13505	2949128
16722	2021-07-04	Kerala	2843909	13640	2961584
16758	2021-07-05	Kerala	2855460	13716	2973684
16794	2021-07-06	Kerala	2866806	13818	2981721
16830	2021-07-07	Kerala	2877557	13960	2996094

525 rows × 5 columns

```
In [27]: #Visualizing confirmed cases in Kerala
sns.set(rc={'figure.figsize':(15,10)})
sns.lineplot(x="date",y="confirmed",data=kerala,color="g")
plt.show()
```



```
In [28]: #Visualizing death cases in Kerala
sns.set(rc={'figure.figsize':(15,10)})
sns.lineplot(x="date",y="deaths",data=kerala,color="r")
plt.show()
```



## ALL Cases (West Bengal)

In [49]:

```
wb
```

Out[49]:

	date	state	cured	deaths	confirmed
208	2020-03-18	West Bengal	0	0	1
227	2020-03-19	West Bengal	0	0	1
247	2020-03-20	West Bengal	0	0	2
269	2020-03-21	West Bengal	0	0	3
292	2020-03-22	West Bengal	0	0	4
...	...	...	...	...	...
16705	2021-07-03	West Bengal	1465219	17758	1502706
16741	2021-07-04	West Bengal	1467038	17779	1504097
16777	2021-07-05	West Bengal	1468815	17799	1505394
16813	2021-07-06	West Bengal	1470512	17817	1506279
16849	2021-07-07	West Bengal	1472132	17834	1507241

477 rows × 5 columns

In [50]:

```
#converting date-time to ordinal
wb['date']=wb['date'].map(dt.datetime.toordinal)
wb.head()
```

<ipython-input-50-01650ee450ec>:2: SettingWithCopyWarning:  
A value is trying to be set on a copy of a slice from a DataFrame.  
Try using .loc[row\_indexer,col\_indexer] = value instead

See the caveats in the documentation: [https://pandas.pydata.org/pandas-docs/stable/user\\_guide/indexing.html#returning-a-view-versus-a-copy](https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy)  
wb['date']=wb['date'].map(dt.datetime.toordinal)

Out[50]:

	date	state	cured	deaths	confirmed
208	737502	West Bengal	0	0	1
227	737503	West Bengal	0	0	1
247	737504	West Bengal	0	0	2
269	737505	West Bengal	0	0	3
292	737506	West Bengal	0	0	4

```
In [51]: #getting dependent variable and inpedent variable
x=maha['date']
y=maha['confirmed']
```

```
In [52]: x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.3)
```

```
In [53]: lr2 = LinearRegression()
```

```
In [58]: lr2.fit(np.array(x_train).reshape(-1,1),np.array(y_train).reshape(-1,1))
```

```
Out[58]: LinearRegression()
```

```
In [59]: wb.tail()
```

```
Out[59]:
```

	date	state	cured	deaths	confirmed
16705	737974	West Bengal	1465219	17758	1502706
16741	737975	West Bengal	1467038	17779	1504097
16777	737976	West Bengal	1468815	17799	1505394
16813	737977	West Bengal	1470512	17817	1506279
16849	737978	West Bengal	1472132	17834	1507241

```
In [60]: lr2.predict(np.array([[737985]]))
```

```
Out[60]: array([[4935452.70830536]])
```

## **Conclusion**

Through this project, the analysis on COVID-19 data has been performed successfully. The analysis on this pandemic spread has been done and compared between different countries. The analysis of confirmed cases, active cases, recovered cases and deaths are done separately to give a clear look on how the virus is spreading, which countries are getting affected mostly and how different countries are recovering. A separate analysis on cases of INDIA has been done and predictions of different cases both around the world and INDIA has been done. At last, the accuracy check using different metrics is performed over all the analysis done in this project.