COVID-19 DATA ANALYSIS https://github.com/ashukumar7/Covid19

Experiment Findings · June 2020

DOI: 10.13140/RG.2.2.23615.94880

1 author:

Ashutosh Kumar
National Institute of Technology, Jamshedpur
1 PUBLICATION **0** CITATIONS

SEE PROFILE

Documentation on

COVID-19

DATA ANALYSIS

By: Ashutosh Kumar

M. tech (1st Year), Computer Science and Engineering

National Institute of Technology, Jamshedpur

Mentor: Mr. Bipul Shahi

(Diginique TechLabs, in association with Cognizance'20 IIT ROORKEE)

TABLE OF CONTENT

Keyword

Virus, Pandemic, WHO, COVID-19, Machine Learning, Supervised, Unsupervised, Reinforcement.

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 spread over 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.

Dataset Pre-processing

This section include the parsing of date in a proper readable format[1], renaming some columns into short and descriptive names [2], adding new column ‘active cases’ with the help of other cases available in the dataset [3], creating a data frame that includes the latest cases up to date [4], grouping the data in terms of country and resetting the index [5][6].

[1]
```
df = pd.read_csv(r'...\covid_19_clean_complete.csv',parse_dates=['Date'])
```

[2]
```
df.rename(columns={"Country/Region":"country",'Province/State':'state'},inplace=True)
```

[3]
```
df['active'] = df['Confirmed'] - df['Deaths'] - df['Recovered']
```

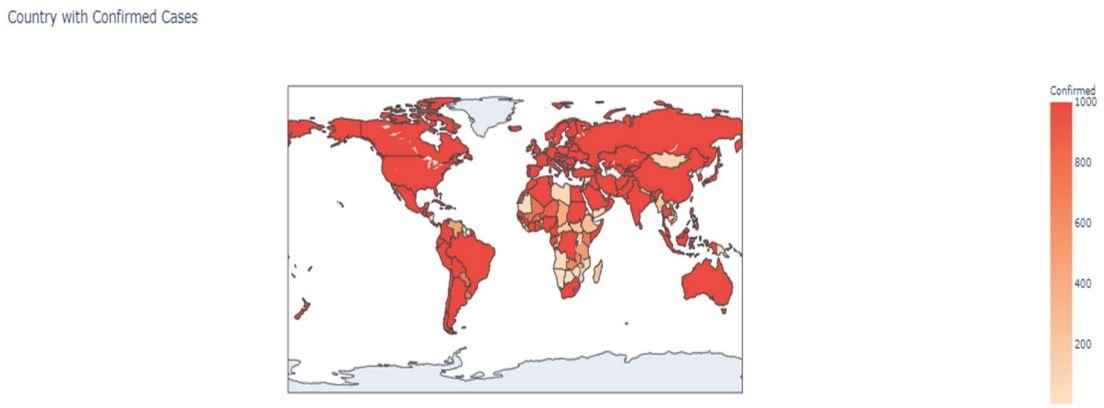
[4]
```
top = df[df['Date'] == df['Date'].max()]
```

[5]
```
world = top.groupby('country')['Confirmed','active','Deaths'].sum()
```

[6]
```
world = top.groupby('country')['Confirmed','active','Deaths'].sum().reset_index()
```

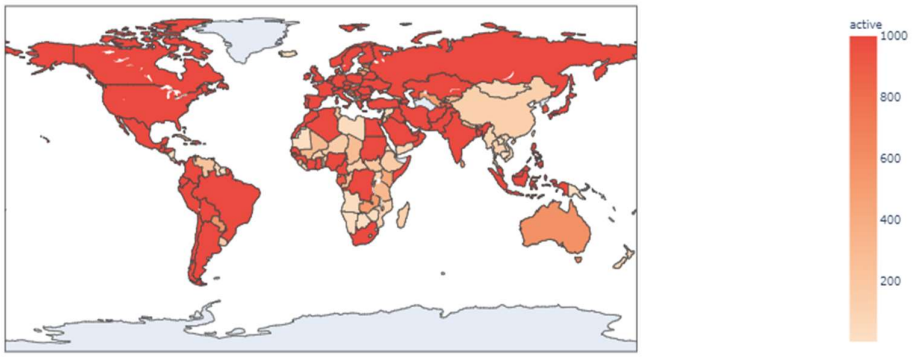
Plotting on world map

Confirmed Cases



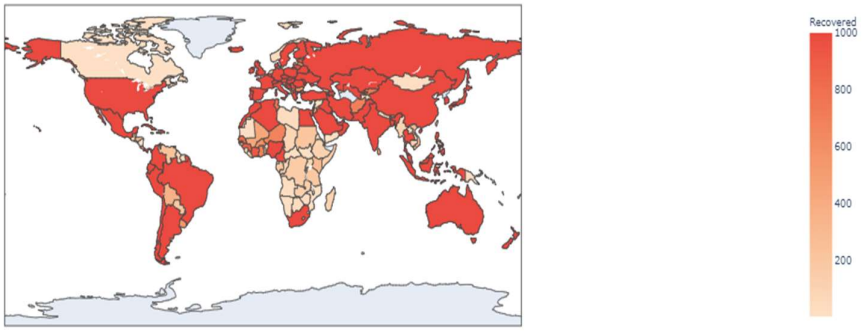
Active Cases

Country with Active Cases



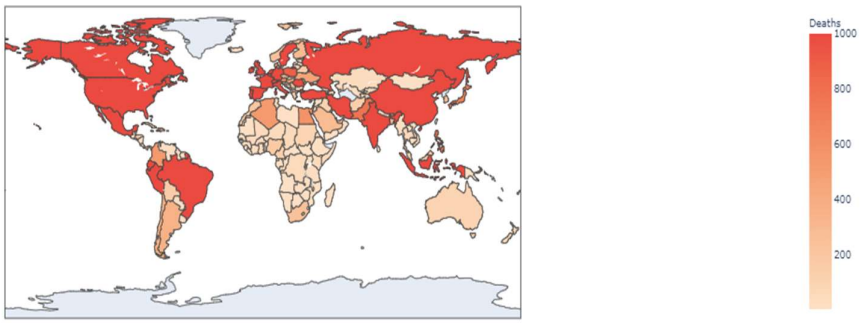
Recovered Cases

Country with Recovered Cases



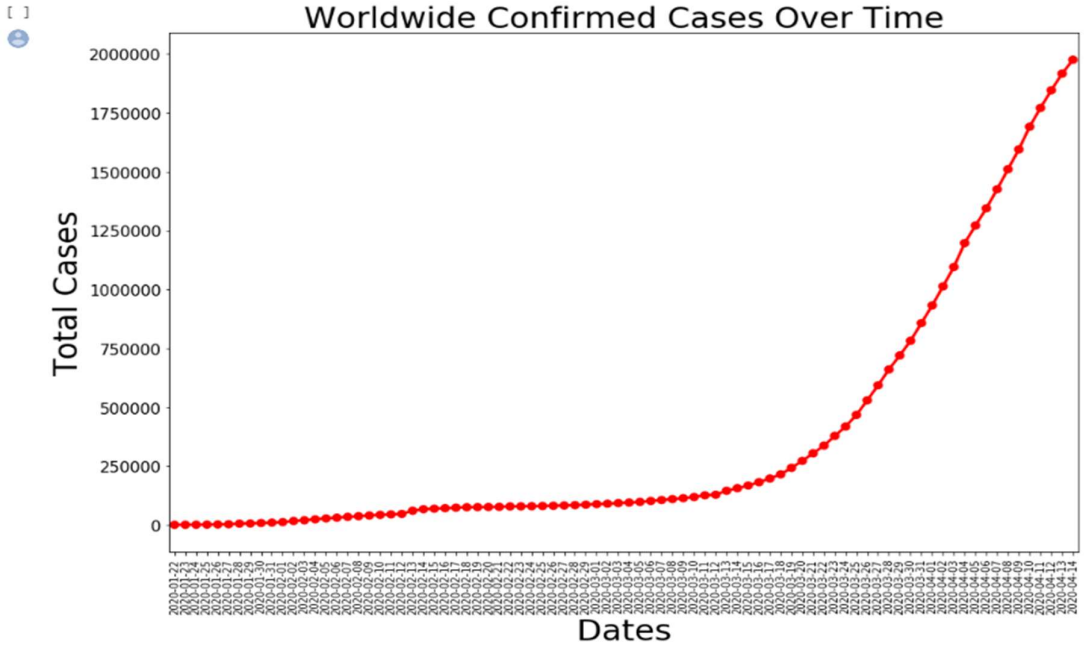
Death Cases

Country with death Cases

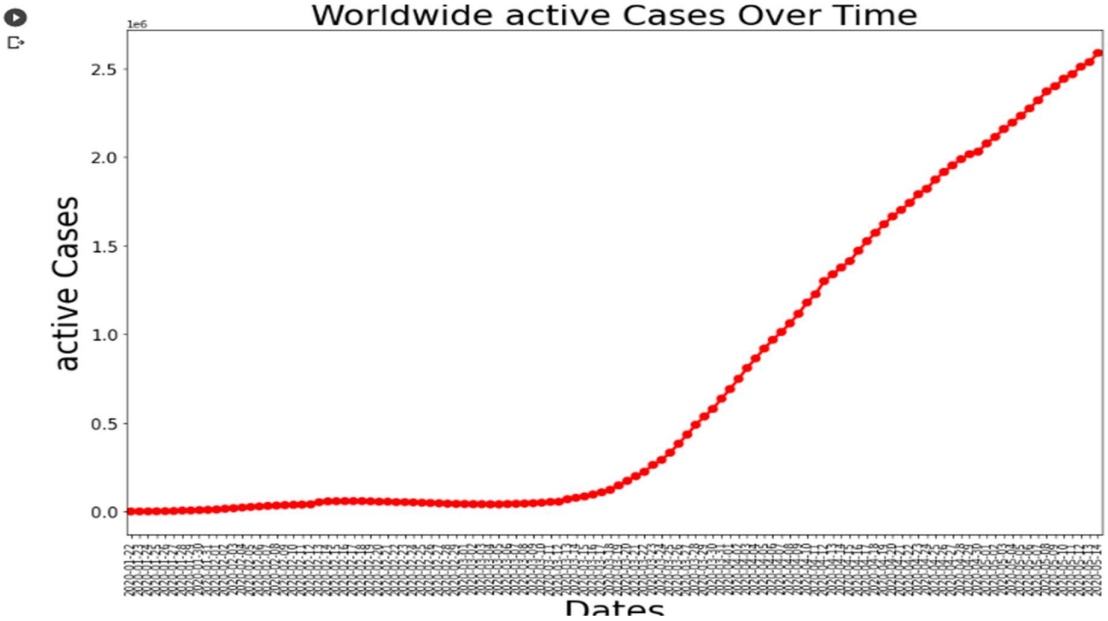


Cases over time

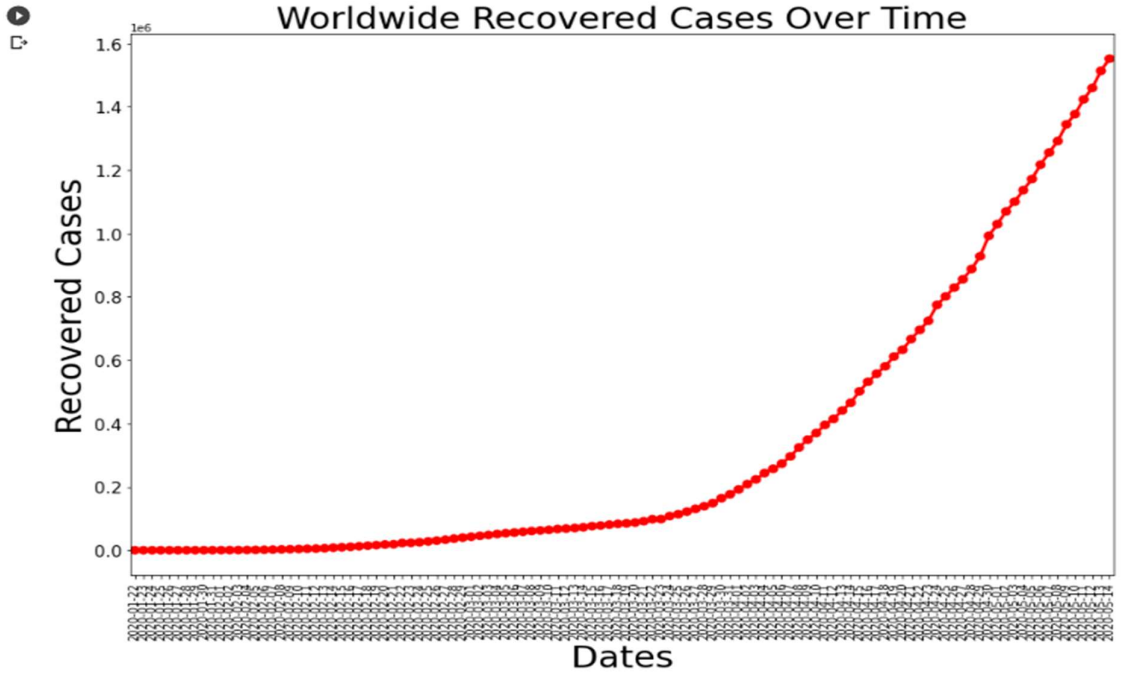
Confirmed Cases day-wise spread



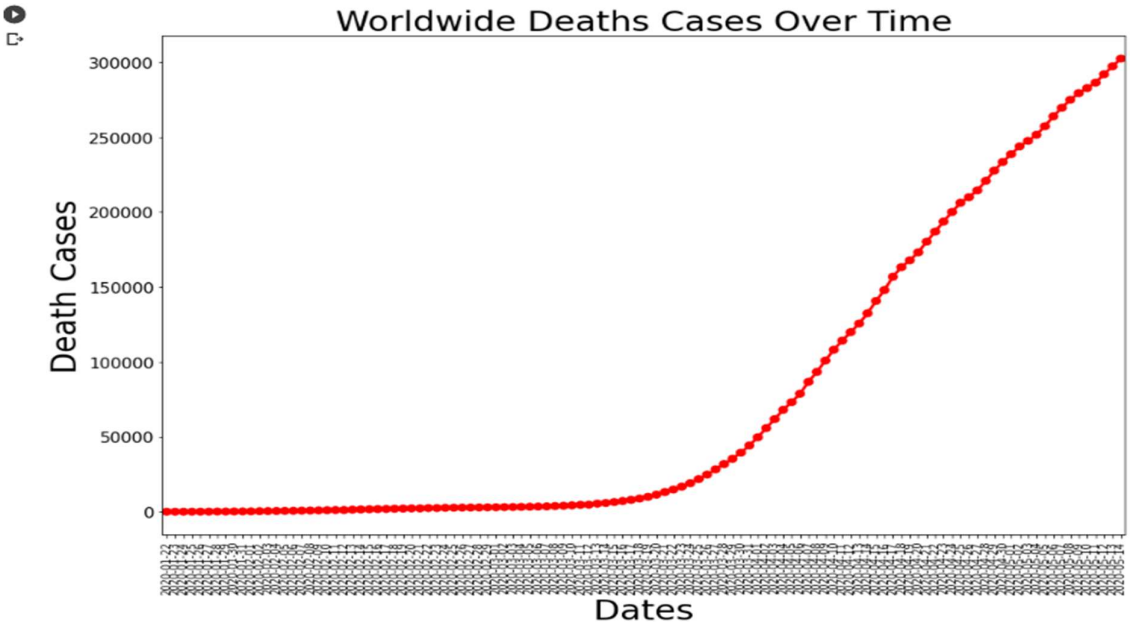
Active Cases day-wise spread



Recovered Cases day-wise



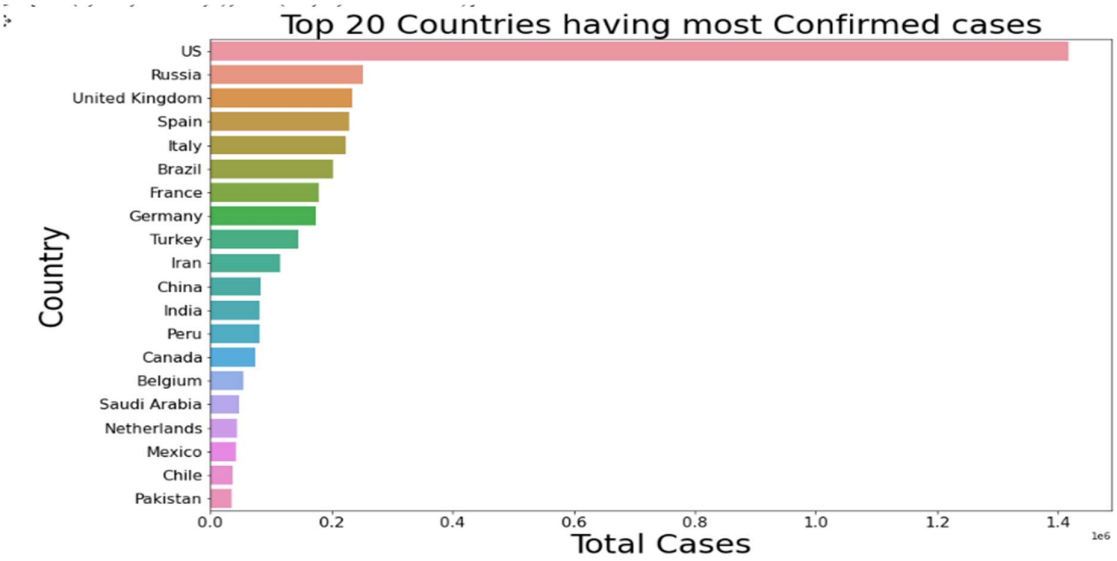
Death Cases day-wise



TOP 20 Countries

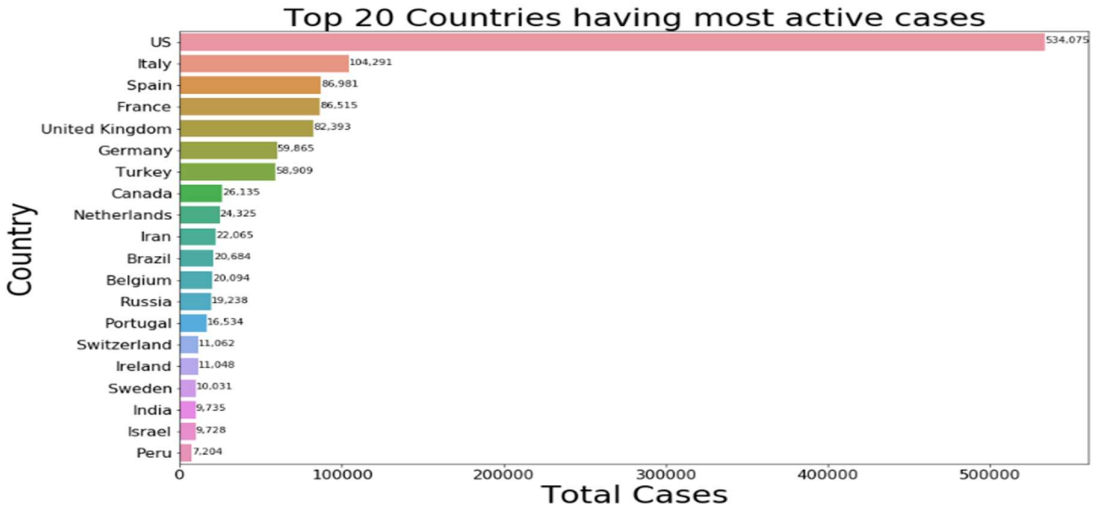
Confirmed

Most affected Country: US



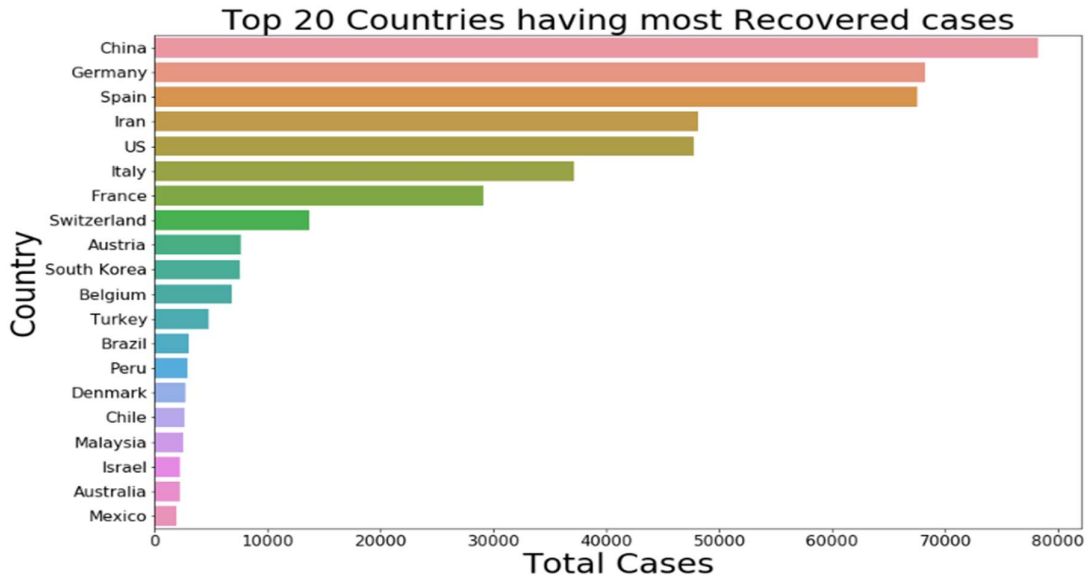
Active

Most affected country: US



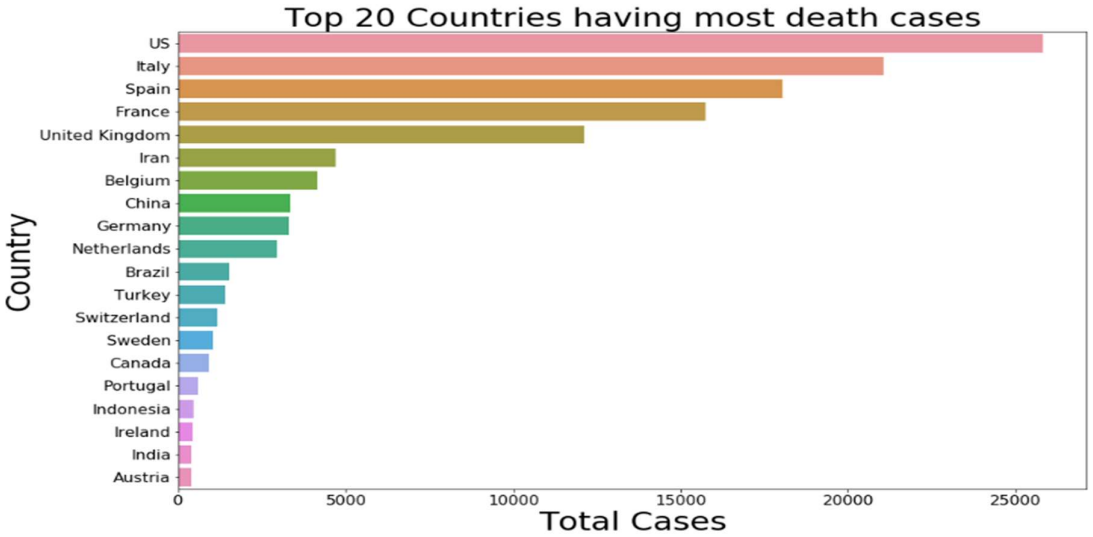
Recovered

Most recoveries: CHINA



Death

Most deaths in: US



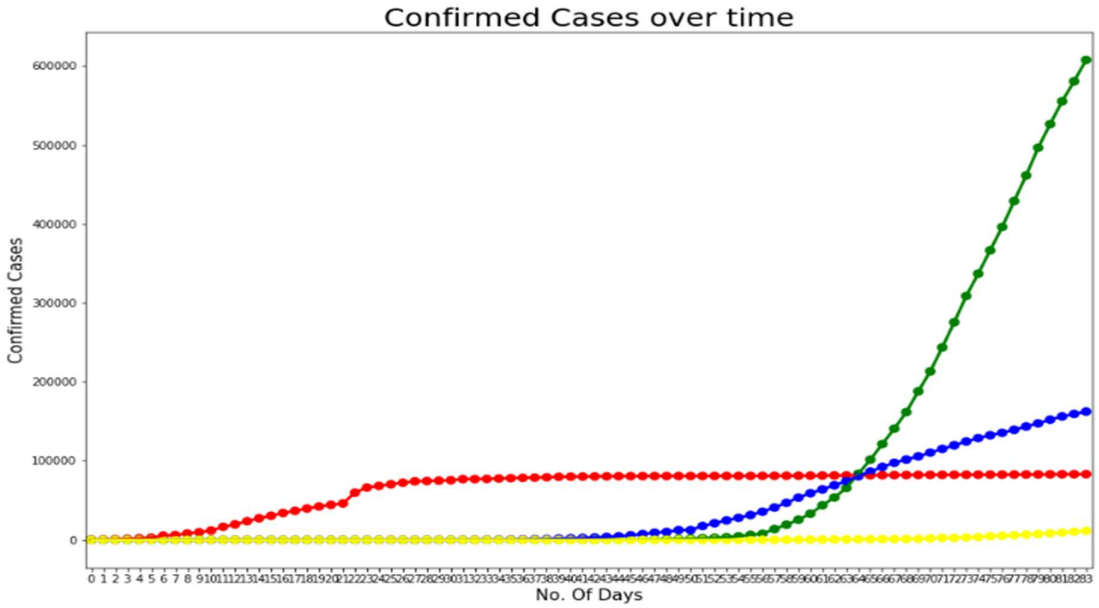
Analysis on some countries

Countries taken for analysis are: **CHINA**, **US**, **ITALY** AND **INDIA**.

Highest confirmed cases: US.

Lowest Confirmed cases: INDIA.

Confirmed

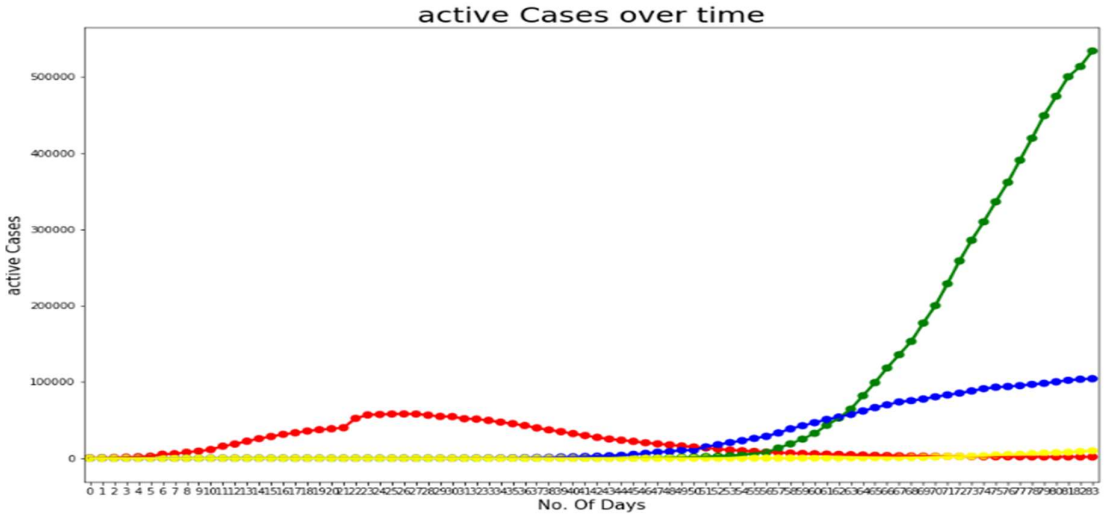


Active

Countries taken for analysis are: **CHINA**, **US**, **ITALY** AND **INDIA**.

Highest Active cases: US.

Lowest Active cases: CHINA.

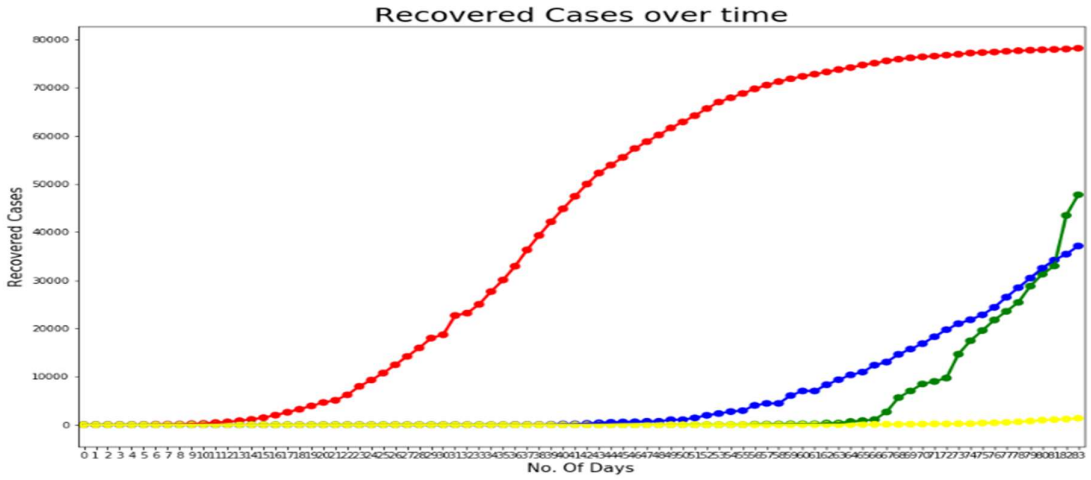


Recovered

Countries taken for analysis are: **CHINA**, **US**, **ITALY** AND **INDIA**.

Highest Recovered cases: CHINA.

Lowest Recovered cases: INDIA

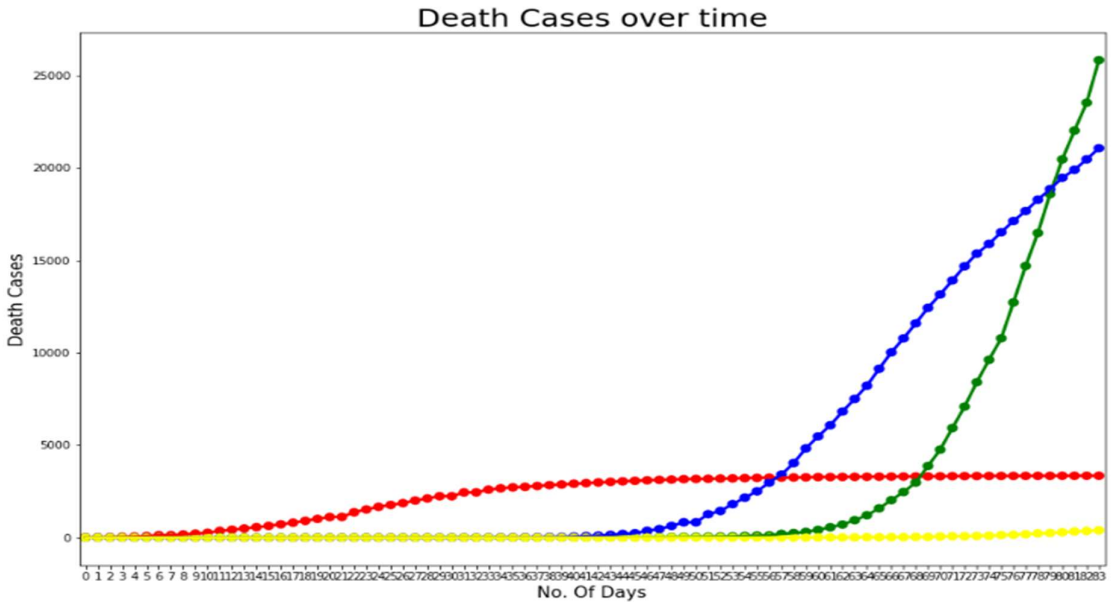


Death

Countries taken for analysis are: **CHINA**, **US**, **ITALY** AND **INDIA**.

Highest Death cases: US.

Lowest Death cases: INDIA



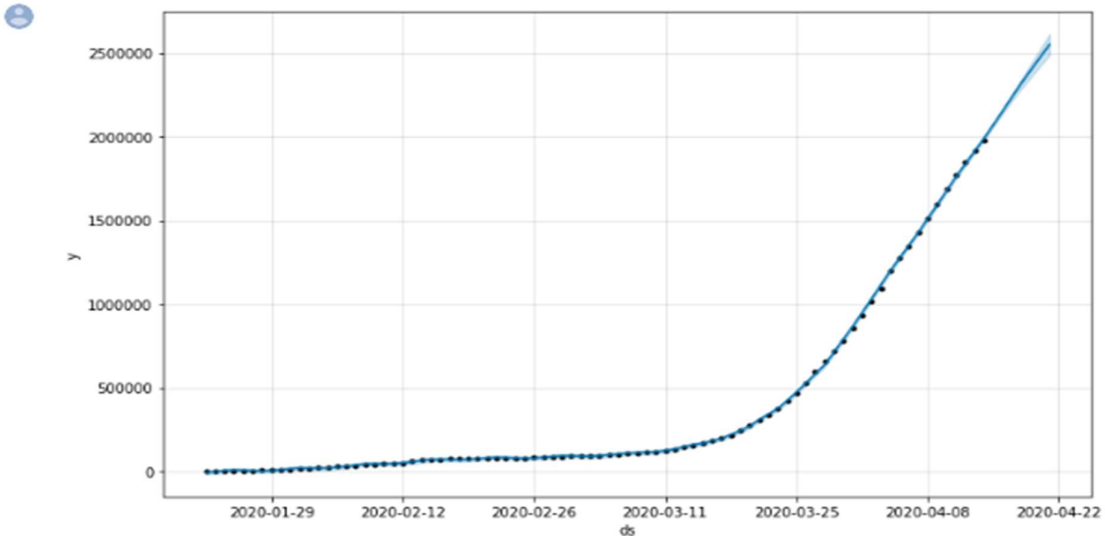
Prediction and Forecasting

Confirmed Cases (WORLD)

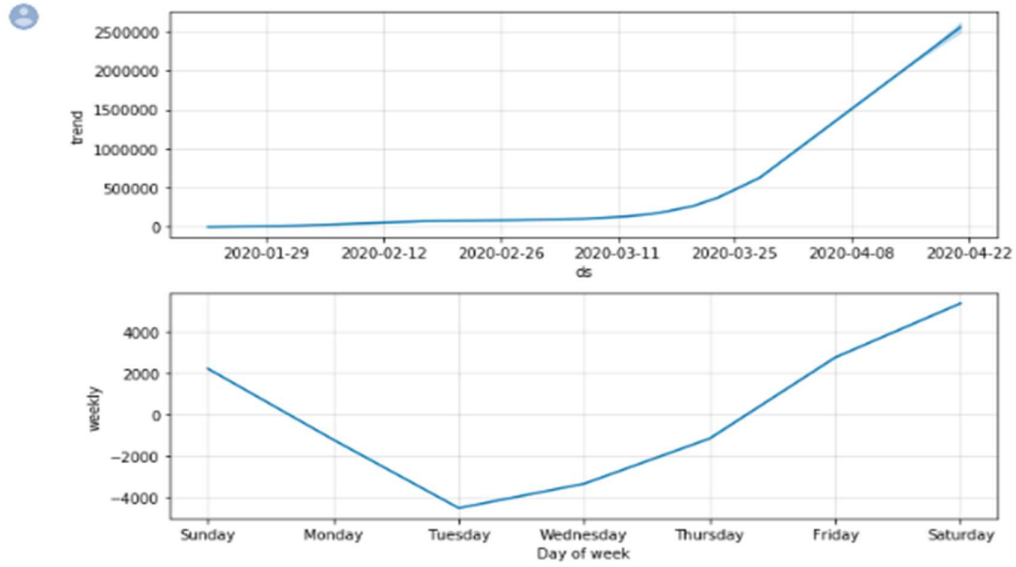
Predicted value : yhat || **Actual Value** :y

```
            ds          yhat          y
0   2020-01-22 -5.786925e+03      555.0
1   2020-01-23 -6.795861e+02      654.0
2   2020-01-24  3.362999e+03      941.0
3   2020-01-25  4.558362e+03     1434.0
4   2020-01-26  5.294289e+03     2118.0
..         ...           ...        ...
109 2020-05-10  4.095126e+06  4101693.0
110 2020-05-11  4.173778e+06  4177496.0
111 2020-05-12  4.252689e+06  4261741.0
112 2020-05-13  4.334106e+06  4347012.0
113 2020-05-14  4.419698e+06  4442157.0

[114 rows x 3 columns]
```

```
[ ] confirmed_forecast_plot = m.plot(forecast)  # ploting predicted value of confirmed cases
    # black dot - actual values
    # blue line = predicted values
```



```
[ ]  confirmed_forecast_plot = m.plot_components(forecast)
```



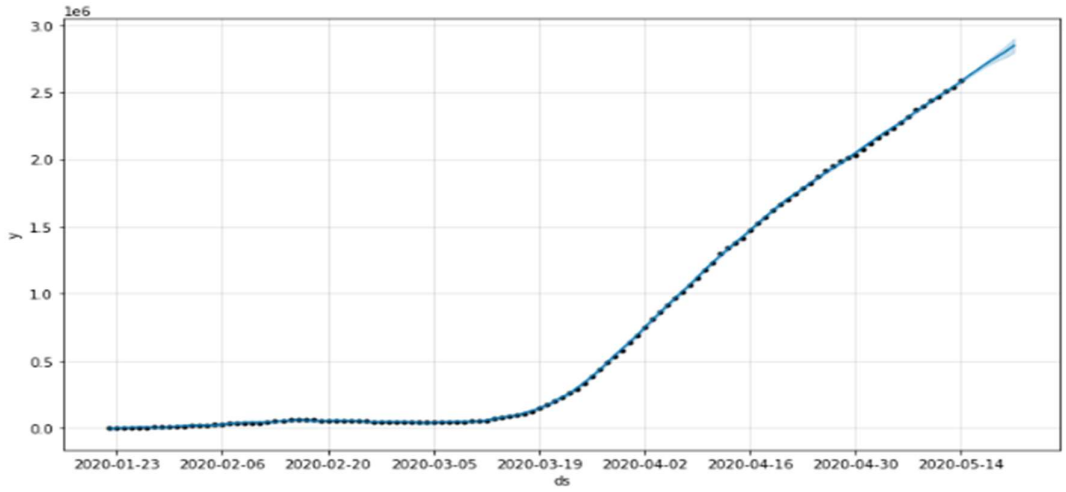
Active Cases (WORLD)

Predicted value : yhat || **Actual Value** :y

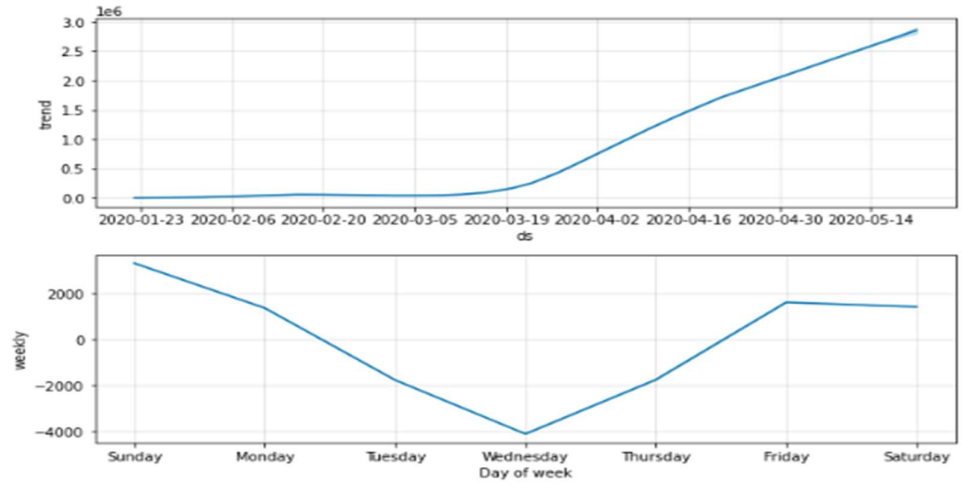
```
# ds - date
# yhat - prediction made
# yhat_lower - lower limit of prediction
# yhoat_upper - upper limit of prediction
metric_df.dropna(inplace=True)
print(metric_df)
```

```
            ds          yhat          y
0   2020-01-22 -5.659046e+03      510.0
1   2020-01-23 -2.181894e+03      606.0
2   2020-01-24  2.298928e+03      879.0
3   2020-01-25  3.230439e+03     1353.0
4   2020-01-26  6.247449e+03     2010.0
..         ...           ...        ...
109 2020-05-10  2.435660e+06  2442197.0
110 2020-05-11  2.471713e+06  2468048.0
111 2020-05-12  2.506587e+06  2510524.0
112 2020-05-13  2.542246e+06  2536535.0
113 2020-05-14  2.582596e+06  2588041.0
```

```
recovered_forecast_plot = m.plot(forecast)  # ploting predicted value of active cases

# black dot - actual values
# blue line = predicted values
```




```
recovered_forecast_plot = m.plot_components(forecast)
```

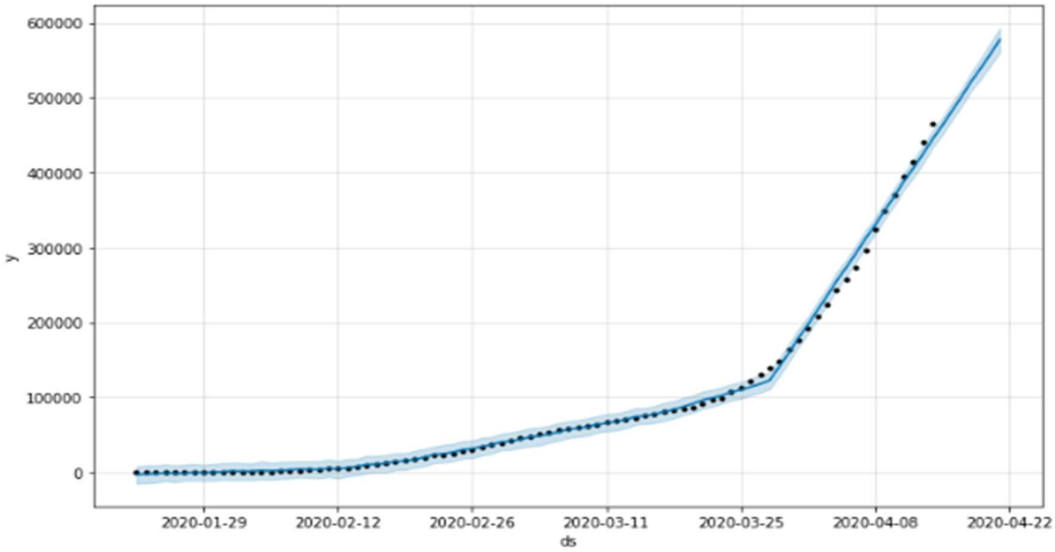


Recovered Cases (WORLD)

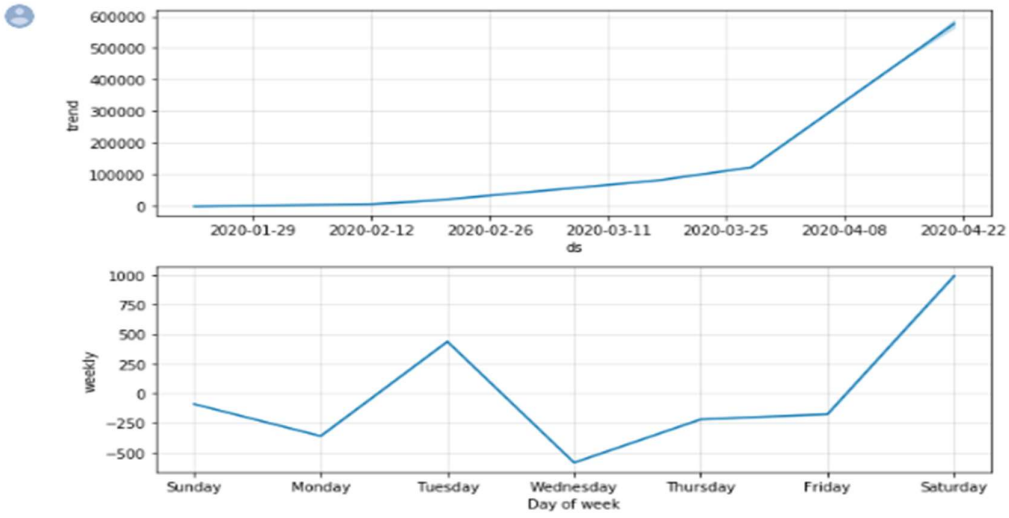
Predicted value : yhat || **Actual Value** :y

	ds	yhat	y
0	2020-01-22	-1.340404e+03	28.0
1	2020-01-23	3.015232e+02	30.0
2	2020-01-24	-2.619864e+02	36.0
3	2020-01-25	1.233584e+02	39.0
4	2020-01-26	-1.530887e+03	52.0
..	...	...	...
109	2020-05-10	1.375868e+06	1376787.0
110	2020-05-11	1.413454e+06	1423118.0
111	2020-05-12	1.451859e+06	1459275.0
112	2020-05-13	1.492197e+06	1513280.0
113	2020-05-14	1.531985e+06	1551698.0

[114 rows x 3 columns]




```
[ ]  recovered_forecast_plot = m.plot_components(forecast)
```



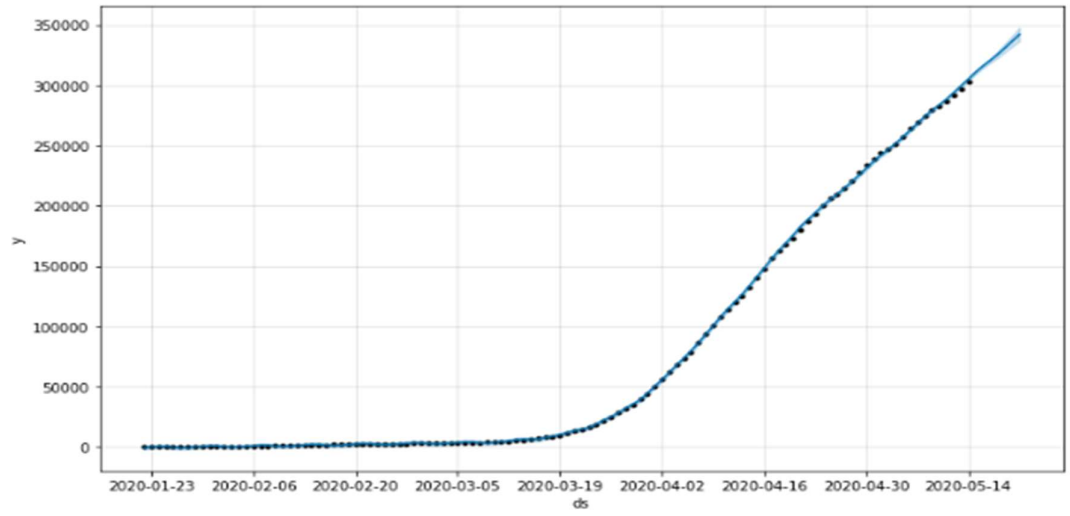
Death Cases (WORLD)

Predicted value : yhat || **Actual Value** :y

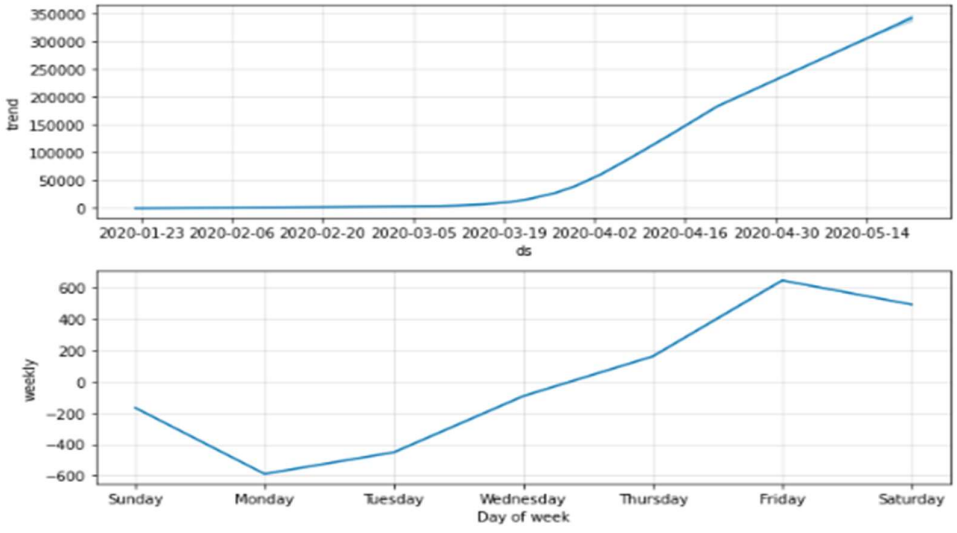
	ds	yhat	y
0	2020-01-22	-417.305727	555.0
1	2020-01-23	-94.405048	654.0
2	2020-01-24	460.126071	941.0
3	2020-01-25	375.335788	1434.0
4	2020-01-26	-214.777228	2118.0
..	...	...	...
109	2020-05-10	283459.176501	4101693.0
110	2020-05-11	288318.206060	4177496.0
111	2020-05-12	293738.505597	4261741.0
112	2020-05-13	299380.089793	4347012.0
113	2020-05-14	304915.809708	4442157.0

[114 rows x 3 columns]

```
deaths_forecast_plot = m.plot(forecast)  # ploting predicted value of death cases
```




```
deaths_forecast_plot = m.plot_components(forecast)
```



Prediction and Forecasting for India

Confirmed Cases (INDIA)

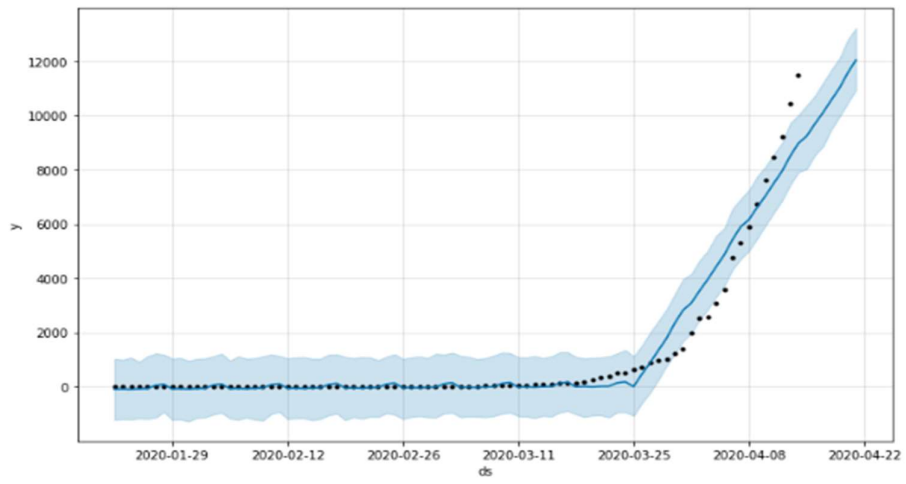
Predicted value : yhat || **Actual Value** :y

```
metric_df=forecast.set_index('ds')[['yhat']].join(india_confirmed.set_index('ds').y).reset_index()

# ds - date
# yhat - prediction made
# yhat_lower - lower limit of prediction
# yhoat_upper - upper limit of prediction
metric_df.dropna(inplace=True)
print(metric_df)
```

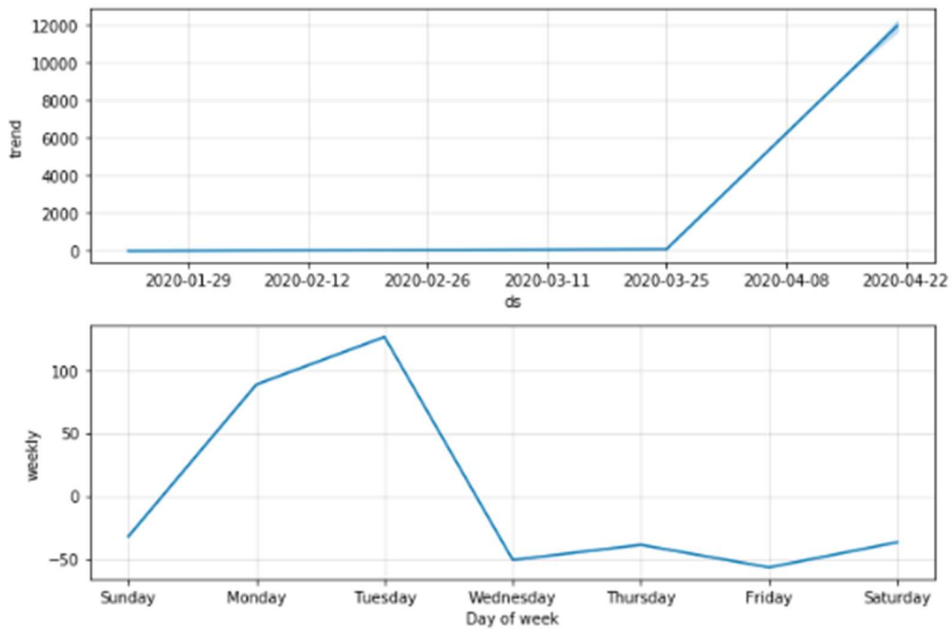
```
            ds          yhat        y
0   2020-01-22    -14.291047      0.0
1   2020-01-23     76.161076      0.0
2   2020-01-24   -376.987538      0.0
3   2020-01-25   -371.672414      0.0
4   2020-01-26   -254.250684      0.0
..         ...           ...      ...
109 2020-05-10  64565.901662  67161.0
110 2020-05-11  67217.116959  70768.0
111 2020-05-12  69781.644051  74292.0
112 2020-05-13  72313.937528  78055.0
113 2020-05-14  74890.626540  81997.0

[114 rows x 3 columns]
```

	ds	yhat	yhat_lower	yhat_upper
86	2020-04-17	10106.612051	8864.403004	11214.204411
87	2020-04-18	10565.888551	9482.179100	11686.294913
88	2020-04-19	11009.571306	9957.137959	12122.438495
89	2020-04-20	11569.586211	10462.555930	12808.011082
90	2020-04-21	12046.676495	10947.019012	13223.884052

```
india_confirmed_forecast_plot = m.plot(forecast)  # ploting predicted value of confirmed cases
```



```
[ ]  india_confirmed_forecast_plot = m.plot_components(forecast)
```



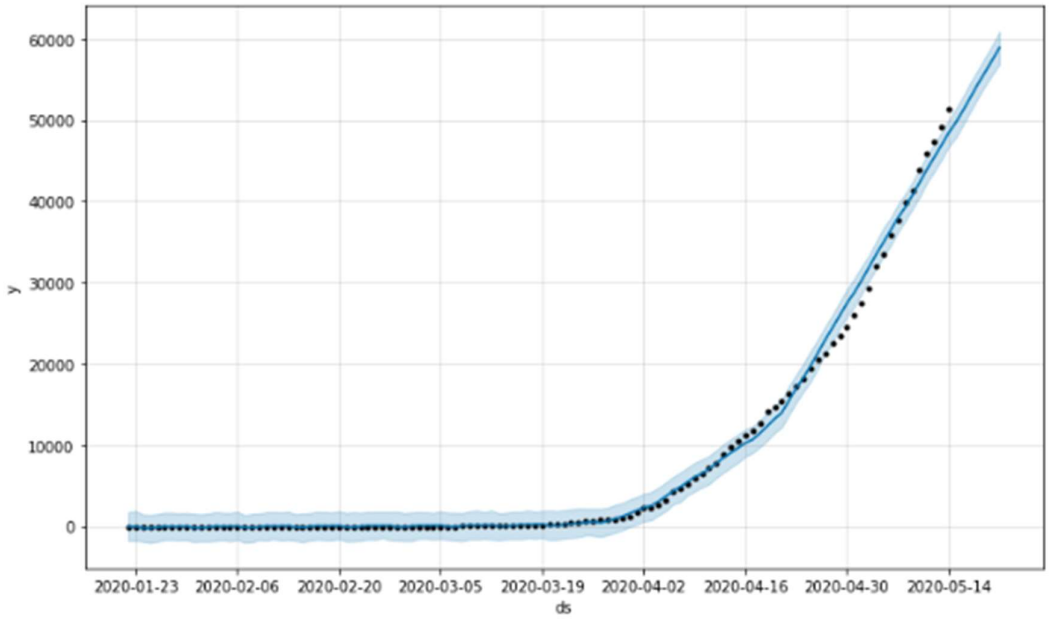
Active Cases (INDIA)

Predicted value : yhat || **Actual Value** :y

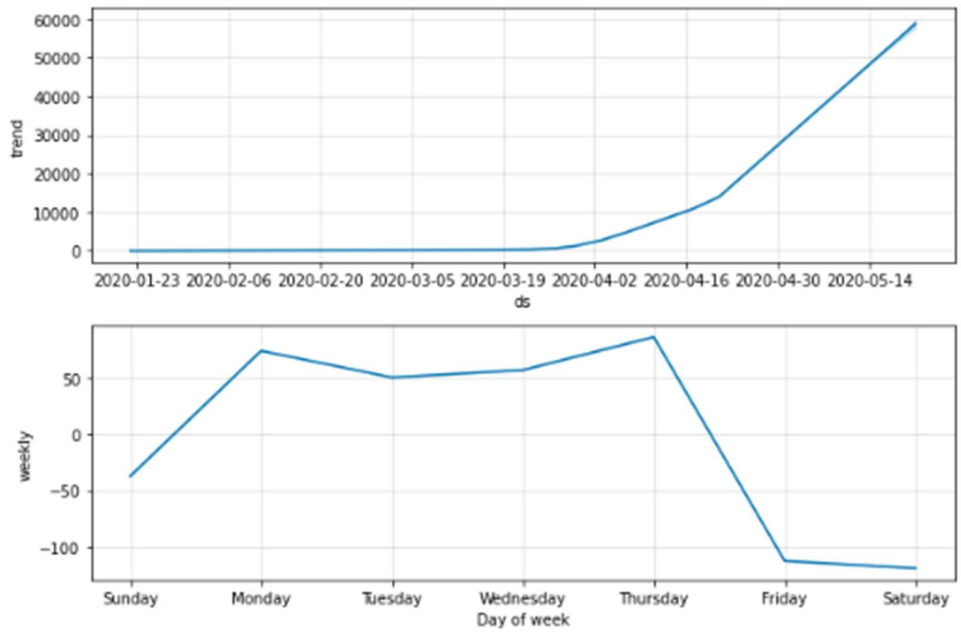
```
metric_df=forecast.set_index('ds')[['yhat']].join(india_active.set_index('ds').y).reset_index()

# ds - date
# yhat - prediction made
# yhat_lower - lower limit of prediction
# yhoat_upper - upper limit of prediction
metric_df.dropna(inplace=True)
print(metric_df)
```

```
            ds          yhat        y
0   2020-01-22    -38.880433      0.0
1   2020-01-23     -5.328541      0.0
2   2020-01-24   -200.042815      0.0
3   2020-01-25   -202.354706      0.0
4   2020-01-26   -116.416678      0.0
..         ...           ...      ...
109 2020-05-10  42327.947542  43980.0
110 2020-05-11  43935.575270  45925.0
111 2020-05-12  45408.209361  47457.0
112 2020-05-13  46910.992468  49104.0
113 2020-05-14  48436.613918  51379.0

[114 rows x 3 columns]
```

```
india_active_forecast_plot = m.plot(forecast)  # ploting predicted value of active cases
```




```
india_confirmed_forecast_plot = m.plot_components(forecast)
```



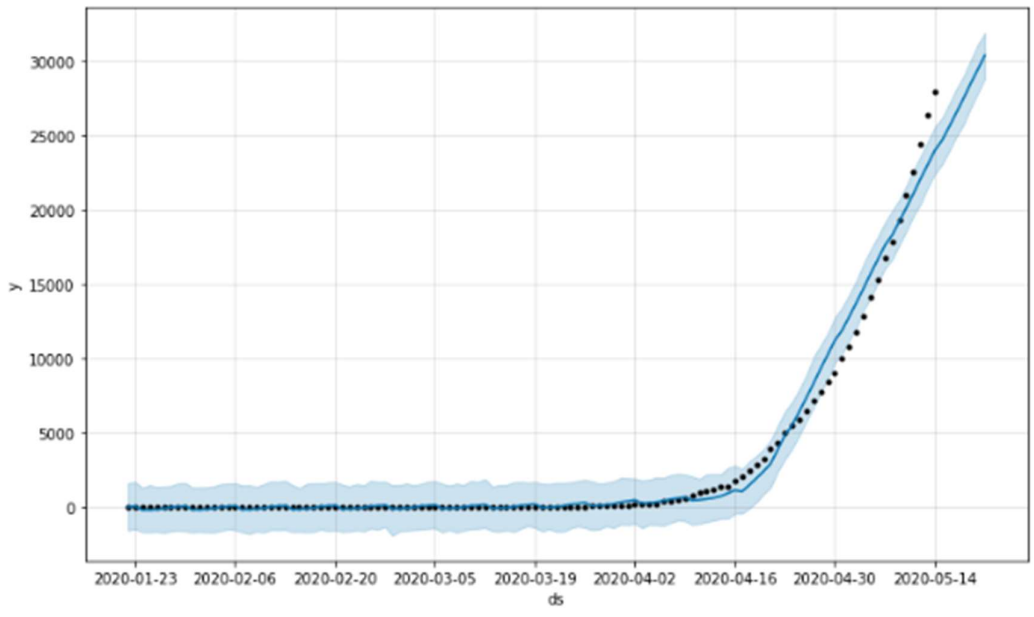
Recovered Cases (INDIA)

Predicted value : yhat || **Actual Value** :y

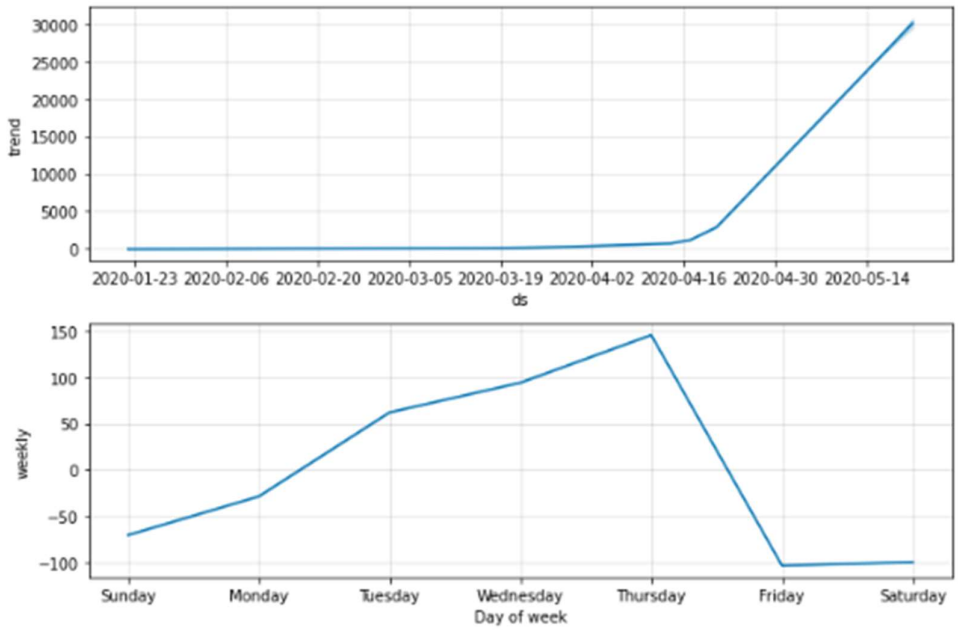
```
metric_df=forecast.set_index('ds')[['yhat']].join(india_recovered.set_index('ds').y).reset_index()

# ds - date
# yhat - prediction made
# yhat_lower - lower limit of prediction
# yhoat_upper - upper limit of prediction
metric_df.dropna(inplace=True)
print(metric_df)
```

```
            ds          yhat        y
0   2020-01-22     16.394136      0.0
1   2020-01-23     70.128292      0.0
2   2020-01-24   -176.388463      0.0
3   2020-01-25   -170.139918      0.0
4   2020-01-26   -138.764944      0.0
..         ...           ...      ...
109 2020-05-10  20108.956349  20969.0
110 2020-05-11  21062.125678  22549.0
111 2020-05-12  22064.224745  24420.0
112 2020-05-13  23007.667178  26400.0
113 2020-05-14  23970.515173  27969.0

[114 rows x 3 columns]
```

```
india_recover_forecast_plot = m.plot(forecast)  # ploting predicted value of recover cases
```



```
india_recover_forecast_plot = m.plot_components(forecast)
```

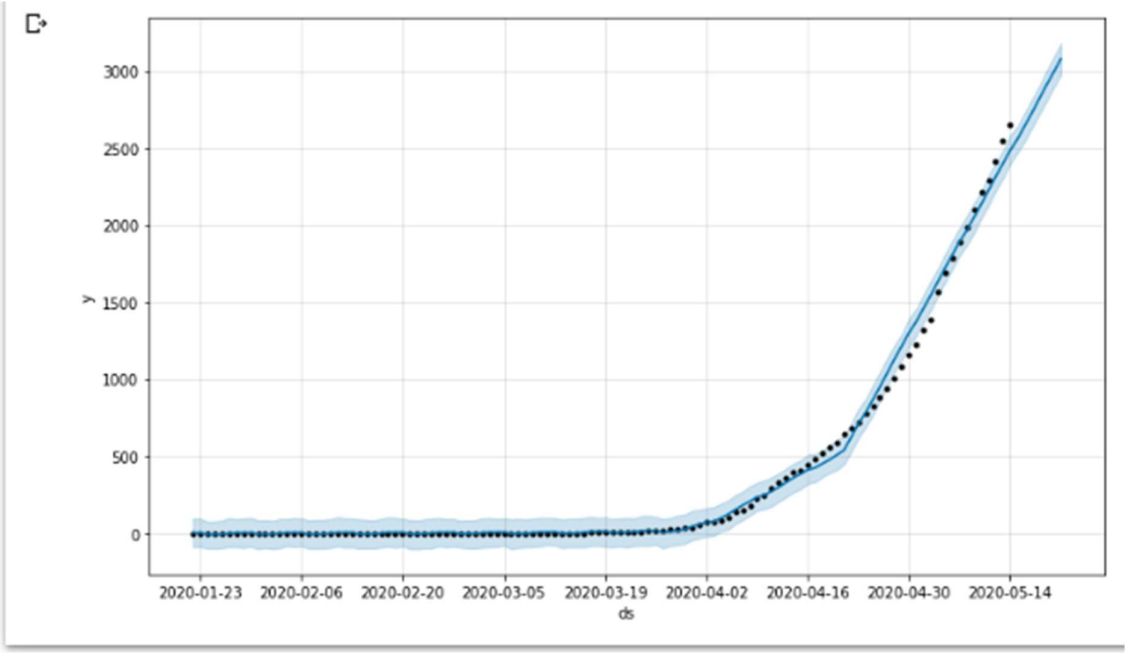


Death Cases (INDIA)

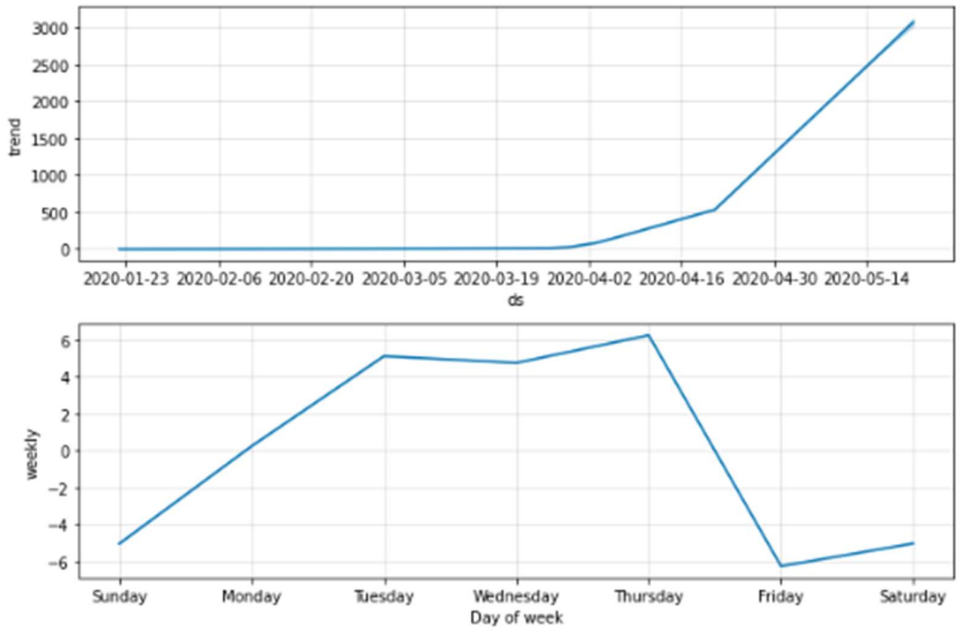
Predicted value : yhat || **Actual Value** :y

```
metric_df=forecast.set_index('ds')[['yhat']].join(india_deaths.set_index('ds').y).reset_index()

# ds - date
# yhat - prediction made
# yhat_lower - lower limit of prediction
# yhoat_upper - upper limit of prediction
metric_df.dropna(inplace=True)
print(metric_df)
```

```
            ds         yhat       y
0   2020-01-22     1.513042     0.0
1   2020-01-23     3.132882     0.0
2   2020-01-24    -9.289070     0.0
3   2020-01-25    -7.938807     0.0
4   2020-01-26    -7.838539     0.0
..         ...          ...     ...
109 2020-05-10  2135.318807  2212.0
110 2020-05-11  2225.249303  2294.0
111 2020-05-12  2314.767694  2415.0
112 2020-05-13  2399.031581  2551.0
113 2020-05-14  2485.169217  2649.0

[114 rows x 3 columns]
```




```
india_deaths_forecast_plot = m.plot_components(forecast)
```



Accuracy Check

R2 Score: R-squared values range from 0 to 1 and are commonly stated as percentages from 0% to 100%. An **R-squared** of 100% means that all movements of a security (or another dependent variable) are completely explained by movements in the index (or the independent variable(s) you are interested in).

Mean Square Error: In statistics, the mean squared error or mean squared deviation of an estimator measures the average of the squares of the errors—that is, the average squared difference between the estimated values and the actual value. MSE is a risk function, corresponding to the expected value of the squared error loss

Mean Absolute Error: In statistics, mean absolute error is a measure of errors between paired observations expressing the same phenomenon. Examples of Y versus X include comparisons of predicted versus observed, subsequent time versus initial time, and one technique of measurement versus an alternative technique of measurement.

On prediction of cases around world

	R2 score	Mean square	Mean absolute
Confirmed	0.9999772750477183	43988753.26062069	4776.939775171555
Active	0.999955817645072	32950433.474696837	4229.980832226832
Recovered	0.9997722347863156	43611232.23497777	603.0377561268665
Deaths	0.9999255700489449	724820.3995915962	3814.2269851803526

On prediction of cases in INDIA

	R2 score	Mean square	Mean absolute
Confirmed	0.9933119004036262	2704197.579317428	861.7053695219906
Active	0.9960804198909399	695454.5901662812	442.7155239532528
Recovered	0.9826537818355147	673471.1660193523	425.5148015583853
Deaths	0.9947698238765833	2291.270968093111 6	26.07155983531586 7

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.