**Covid-19 outbreak prediction using machine learning**

**Contents**

**1. Introduction to Covid-19**

**1.1 Problem statement**

**1.2 Analysing the present the condition in India**

**1.3 Reading the Datasets**

**1.4 Analysing the Covid-19 cases in India**

**1.5 Number of active Covid-19 cases in affected state/union territories.**

**1.6 Confirmed vs Recovered figures**

**1.7 How coronavirus cases are increasing in India?**

**2. Cumulative cases in India,Italy,South korea &Wuhan**

**2.1 Cumulative cases in India**

**2.2 Cumulative cases in Italy**

**2.3 Cumulative cases in South Korea**

**2.4 Cumulative cases in Wuhan**

**2.5 Comparison between the rise in cases in South korea,Italy and India**

**2.6 Trends after crossing 100 cases**

**3. Exploring the world wide data**

**3.1 Visualizing the worldwide covid-19 cases**

**4. Forecasting the values of worldwide confirmed,death & recovered cases using prophet**

**4.1 prophet**

**4.2 Forecasting the values of worldwide Confirmed cases**

**4.3 Forecasting the values of worldwide Death cases**

**4.4 Forecasting the values of worldwide Recovered cases**

**5.Conclusion & Inference**

**1.Intoduction to covid-19**

The coronavirus (COVID-19) emerged in Wuhan(china) in December 201 Jo 9, and has now become a pandemic. This pandemic has spread across 190 countries infecting 4.2 lakh people and killing 16,500 so far. In India, the first covid case was reported on jan 30th and now there are 300 COVID-19 cases have been reported so far on 22nd march 2020.

**1.1 Problem statement**

**As the corona virus outbreak continues to spread in the country , the question is**

**Will India be able to tackle this pandemic or are we going to witness another Italy/south korea/wuhan?**

We need a strong model that predicts how the virus could spread across different countries and regions. The goal of this project is to build a model that predicts the spread of the virus in the next 7 days.

**AIM: To build a model that predicts the spread of virus in the next seven days from 24th,March 2020.**

#Importing the required libraries

Import pandas as pd

*#visualization libraries*

*import matplotlib.pyplot as plt*

*%matplotlib inline #it instructs python not to produce visualization in different window*

*import seaborn as sns*

*import plotly.express as px*

*import plotly.graph\_objects as go*

*import folium # populate maps and geographical locations*

*from folium import plugins*

*#Manipulating the default plot size*

*plt.rcParams['figure.figsize']=10,12*

*# Disable warnings*

*import warnings*

*warnings.filterwarnings('ignore')*

**1.2 Analysing the present the condition in India**

As of march 25th the number of covid-19 cases crossed 550. Rise of pandemic is exponential.The pandemic grows extremely fast once it reaches a significant point but before that it is almost unnoticeable. This virus can not increase infinitely. At some point it stops finding the people to infect. After a sharp uprising the growth slows down.This curve is known as a sigmoid. Every point on the curve signifies the number of cases in a given day.Slope at any dip point gives the value of how many new cases are there in a day.

The government is worried about the health care resources.By flattening the curve we can delay the spread of the pandemic such that it does not exceed the health care resources. We can bring the curve of the spread of the disease to a minimum number of deaths.

**1.3 Reading the Datasets**

*#Reading a read a.xls file by creating a dataframe using pandas*

*#reading the datasets*

*df=pd.read\_excel('covid cases in India.xlsx')*

*df\_india = df.copy()*

*print(df)*

*x=open("op1.py",'a')*

*x.write(df.to\_string())*

*x.close()*

output 1

op1.py

Note: outputs are stored in a file and attached in the same folder

*#Day by Day data of India, korea, Italy and Wuhan*

*dbd\_India=pd.read\_excel('per\_day\_cases.xlsx',parse\_dates=True,sheet\_name='India')*

*dbd\_Italy=pd.read\_excel('per\_day\_cases.xlsx',parse\_dates=True,sheet\_name='Italy')*

*dbd\_korea=pd.read\_excel('per\_day\_cases.xlsx',parse\_dates=True,sheet\_name='Korea')*

*dbd\_Wuhan=pd.read\_excel('per\_day\_cases.xlsx',parse\_dates=True,sheet\_name='Wuhan')*

*print(dbd\_India)*

*x=open("op2.py",'a')*

*x.write(dbd\_India.to\_string())*

*x.close()*

Output2

op2.py

**1.4 Analysing the Covid-19 cases in India**

*#total case is the total number of confirmed cases (Indian national foreign national)*

*df.drop(['S. No.'],axis=1,inplace=True)*

*df['Total cases']=df['Total Confirmed cases (Indian National)']+ df['Total Confirmed cases ( Foreign National )']*

*total\_cases = df['Total cases'].sum()*

*print('Total number of confirmed COVID 2019 cases across India till date(22nd march,2020):',total\_cases)*

*Total number of confirmed COVID 2019 cases across India till date(22nd march,2020): 562*

*#highlighting dataframe*

#df.style.background\_gradient(cmap='Reds')

**1.5 Number of active Covid-19 cases in affected state/union territories.**

*# total active is the total cases - (Number of death + cured)*

*df['Total Active']=df['Total cases']-(df['Death']+df['Cured'])*

*total\_active = df['Total Active'].sum()*

*print('Total number of active covid 2019 cases across India:',total\_active)*

*Total number of active covid 2019 cases across India: 512*

*Tot\_Cases=df.groupby('Name of State / UT')['Total Active'].sum().sort\_values(ascending=False).to\_frame()*

*Tot\_Cases.style.background\_gradient(cmap='Reds')*

*print(Tot\_Cases)*

*x=open("op3.py",'a')*

*x.write(Tot-Cases.to\_string())*

*x.close()*

Output3

Op3,py

**1.6 Confirmed vs Recovered figures**

#coordinates of India states and union territories

India\_coord=pd.read\_excel('Indian Coordinates.xlsx')

df\_full = pd.merge(India\_coord,df,on='Name of State / UT')

# using Seaborn for visualization\n",

f, ax = plt.subplots(figsize=(12, 8))

data = df\_full[['Name of State / UT','Total cases','Cured','Death']]

data.sort\_values('Total cases',ascending=False,inplace=True)

sns.set\_color\_codes("pastel")

sns.barplot(x="Total cases", y="Name of State / UT", data=data,label="Total", color="r")

#plt.show()

sns.set\_color\_codes("muted")

sns.barplot(x="Cured", y="Name of State / UT", data=data, label="Cured", color="g")

plt.show()

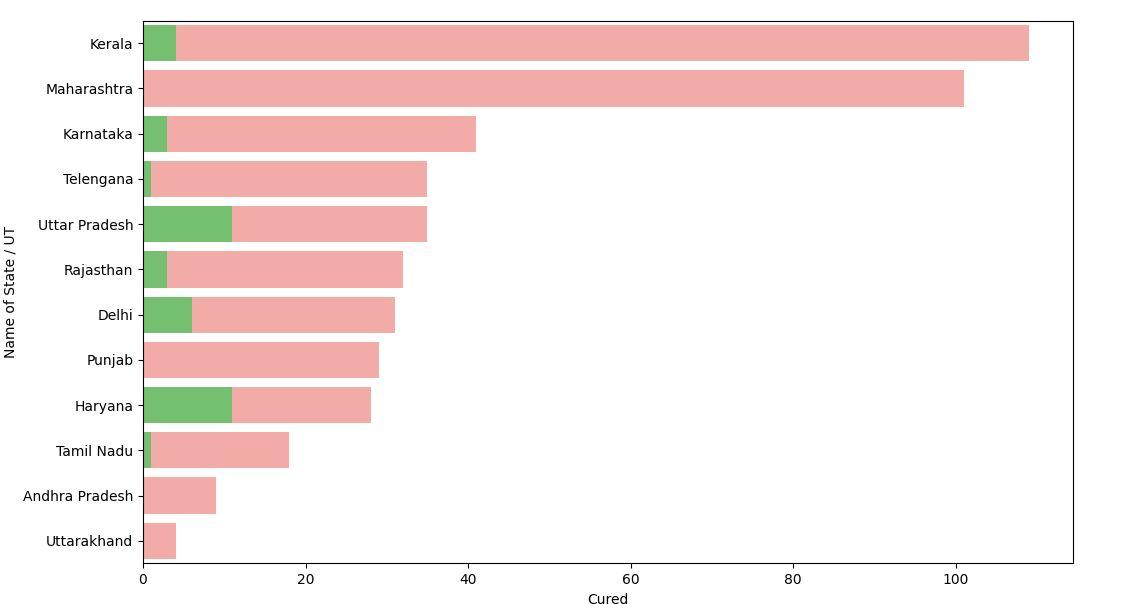
# Add a legend and informative axis label

ax.legend(ncol=2, loc="lower right", frameon=True)

ax.set(xlim=(0, 35), ylabel="",xlabel="Cases")

sns.despine(left=True, bottom=True)

figure\_cured

**Cured-green Total confirmed cases-pink**

**1.7 How coronavirus cases are increasing in India?**

*import plotly*

*import plotly.graph\_objects as go*

*plotly.io.renderers.default='colab'*

*#dbd\_Inida=pd.read\_excel('per\_day\_cases.xlsx',parse\_dates=True,sheet\_name='India')*

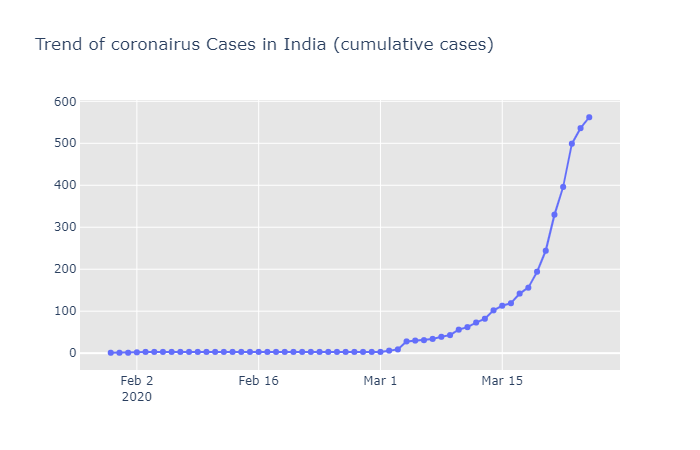
*fig = go.Figure()*

*fig.add\_trace(go.Scatter(x=dbd\_India['Date'],y= dbd\_India['Total Cases'],mode='lines+markers',name='Total Cases'))*

*fig.update\_layout(title\_text='Trend of coronavirus Cases in India (cumulative cases)',plot\_bgcolor='rgb(230,230,230)')*

*fig.write\_html('first\_figure.html',auto\_open=True)*

*fig.show*



first\_figure

*#new covid cases reported daily in India*

*import plotly.express as px*

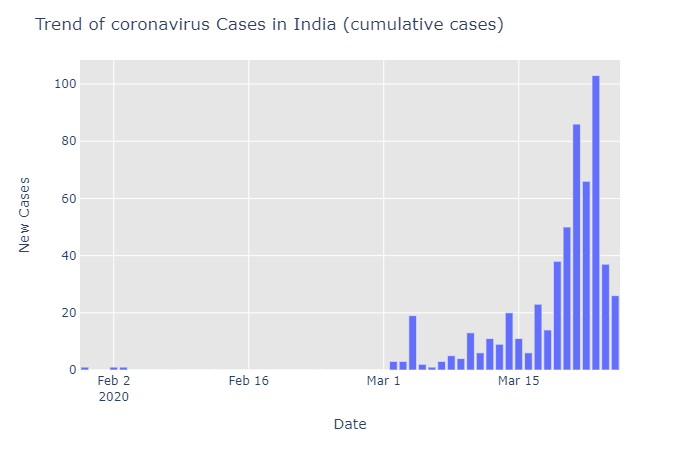
*fig=px.bar(dbd\_India,x="Date",y="New Cases", barmode='group',height=400)*

*fig.update\_layout(title\_text='Trend of coronavirus Cases in India (cumulative cases)',plot\_bgcolor='rgb(230,230,230)')*

*fig.write\_html('second\_figure.html',auto\_open=True)*

*fig.show()*

second\_figure



**2. Cumulative cases in India,Italy,South korea &Wuhan**

**2.1 Cumulative cases in India**

# import plotly.express as px

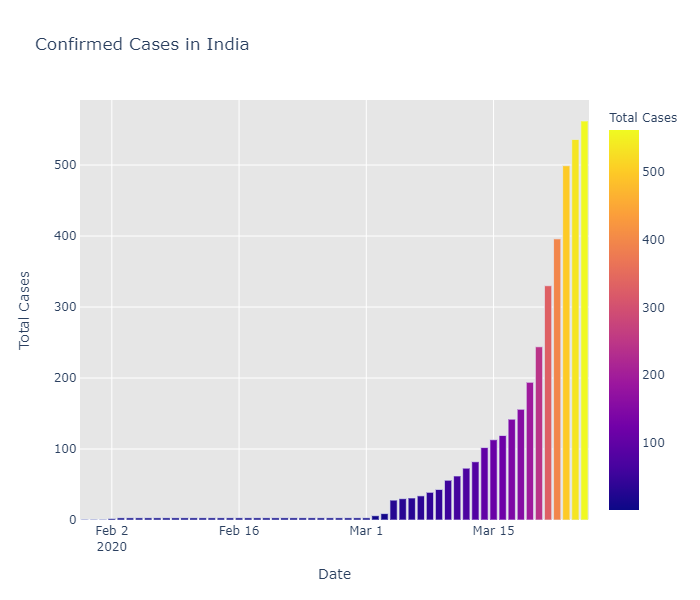
*fig = px.bar(dbd\_India, x="Date", y="Total Cases", color='Total Cases', orientation='v', height=600,title='Confirmed Cases in India', color\_discrete\_sequence = px.colors.cyclical.IceFire)*

*fig.update\_layout(plot\_bgcolor='rgb(230, 230, 230)')*

*fig.write\_html('third\_figure.html',auto\_open=True)*

*fig.show()*

third \_figure

**

**2.2 Cumulative cases in Italy**

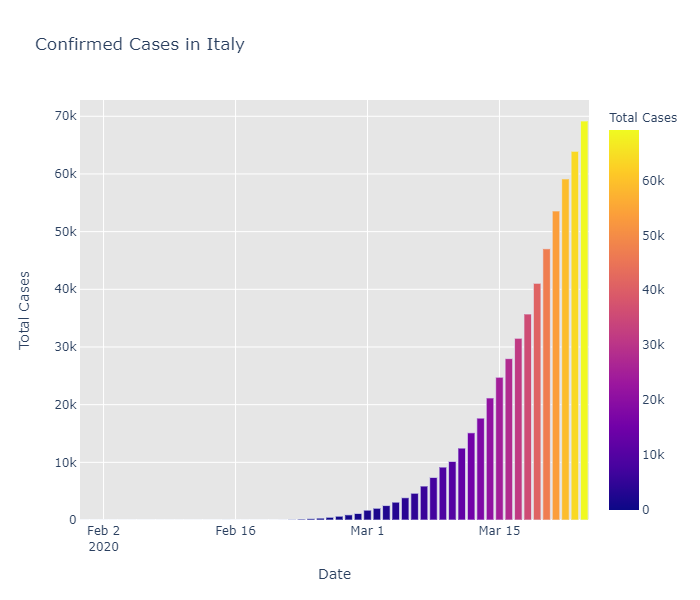
*fig = px.bar(dbd\_Italy, x="Date", y="Total Cases", color='Total Cases', orientation='v', height=600,title='Confirmed Cases in Italy', color\_discrete\_sequence = px.colors.cyclical.IceFire)*

*fig.update\_layout(plot\_bgcolor='rgb(230, 230, 230)')*

*fig.write\_html('fourth\_figure.html',auto\_open=True)*

*fig.show()*

fourth\_figure

**

**2.3 Cumulative cases inSouth korea**

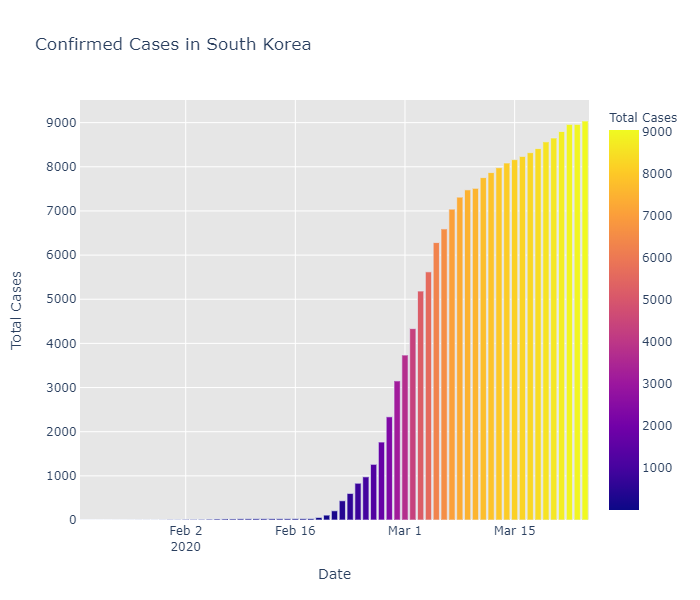
*fig = px.bar(dbd\_korea, x="Date", y="Total Cases", color='Total Cases', orientation='v', height=600, title='Confirmed Cases in South Korea', color\_discrete\_sequence = px.colors.cyclical.IceFire)*

*fig.update\_layout(plot\_bgcolor='rgb(230, 230, 230)')*

*fig.write\_html('fifth\_figure.html',auto\_open=True)*

*fig.show()*

*fifth\_figure*

**

**2.4 Cumulative cases in Wuhan**

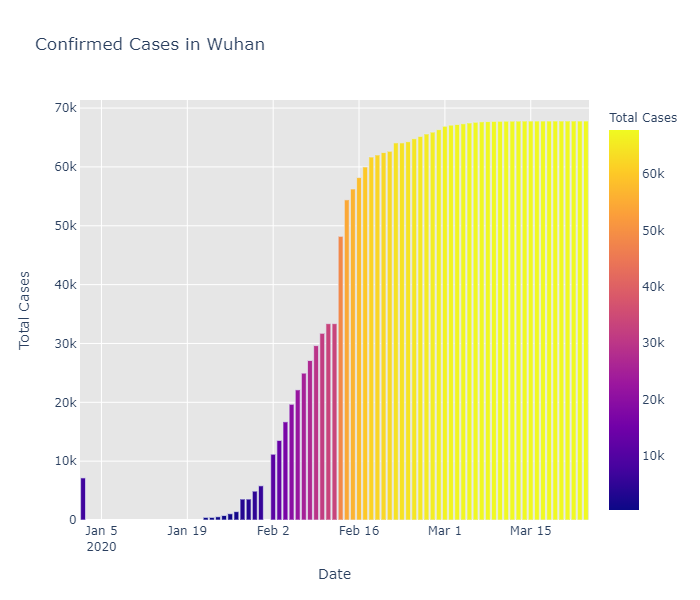
*fig = px.bar(dbd\_Wuhan, x="Date", y="Total Cases", color='Total Cases', orientation='v', height=600, title='Confirmed Cases in Wuhan', color\_discrete\_sequence = px.colors.cyclical.IceFire)*

*fig.update\_layout(plot\_bgcolor='rgb(230, 230, 230)')*

*fig.write\_html('sixth\_figure.html',auto\_open=True)*

*fig.show()*

*sixth\_figure*

**

**2.5 Comparison between the rise in cases in South korea,Italy and India**

*# import plotly.graph\_objects as go*

*from plotly.subplots import make\_subplots*

*fig = make\_subplots(rows=2, cols=2,specs=[[{}, {}], [{"colspan": 2}, None]],*

*subplot\_titles=("S.Korea","Italy", "India","Wuhan"))*

*fig.add\_trace(go.Bar(x=dbd\_korea['Date'], y=dbd\_korea['Total Cases'],*

*marker=dict(color=dbd\_korea['Total Cases'], coloraxis="coloraxis")),1, 1)*

*fig.add\_trace(go.Bar(x=dbd\_Italy['Date'], y=dbd\_Italy['Total Cases'],*

*marker=dict(color=dbd\_Italy['Total Cases'], coloraxis="coloraxis")),1, 2)*

*fig.add\_trace(go.Bar(x=dbd\_India['Date'], y=dbd\_India['Total Cases'],*

*marker=dict(color=dbd\_India['Total Cases'], coloraxis="coloraxis")),2, 1)*

*# fig.add\_trace(go.Bar(x=dbd\_Wuhan['Date'], y=dbd\_Wuhan['Total Cases'],*

*# marker=dict(color=dbd\_Wuhan['Total Cases'], coloraxis="coloraxis")),2, 2)*

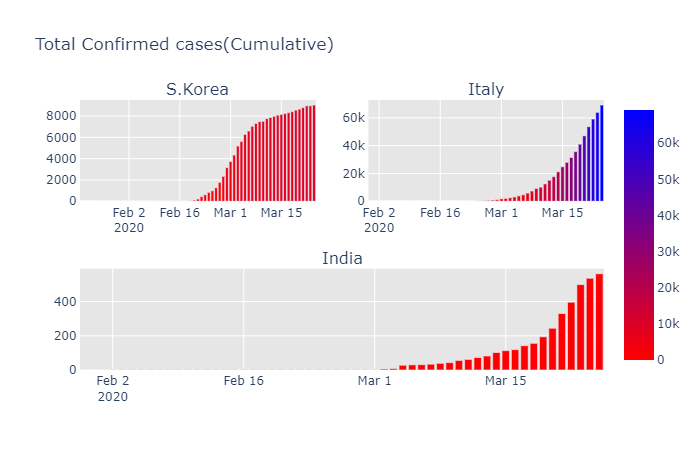
*fig.update\_layout(coloraxis=dict(colorscale='Bluered\_r'), showlegend=False,title\_text="Total Confirmed cases(Cumulative)")*

*fig.update\_layout(plot\_bgcolor='rgb(230, 230, 230)')*

*fig.write\_html('seventh\_figure.html',auto\_open=True)*

*fig.show()*

*seventh\_figure*

**

**2.6 Trends after crossing 100 cases**

*# import plotly.graph\_objects as go*

*title = 'Main Source for News'*

*labels = ['S.Korea', 'Italy', 'India']*

*colors = ['rgb(122,128,0)', 'rgb(255,0,0)', 'rgb(49,130,189)']*

*mode\_size = [10, 10, 12]*

*line\_size = [1, 1, 8]*

*fig = go.Figure()*

*fig.add\_trace(go.Scatter(x=dbd\_korea['Days after surpassing 100 cases'],*

*y=dbd\_korea['Total Cases'],mode='lines',*

*name=labels[0],*

*line=dict(color=colors[0], width=line\_size[0]),*

*connectgaps=True))*

*fig.add\_trace(go.Scatter(x=dbd\_Italy['Days after surpassing 100 cases'],*

*y=dbd\_Italy['Total Cases'],mode='lines',*

*name=labels[1],*

*line=dict(color=colors[1], width=line\_size[1]),*

*connectgaps=True))*

*fig.add\_trace(go.Scatter(x=dbd\_India['Days after surpassing 100 cases'],*

*y=dbd\_India['Total Cases'],mode='lines',*

*name=labels[2],*

*line=dict(color=colors[2], width=line\_size[2]),*

*connectgaps=True))*

*annotations = []*

*annotations.append(dict(xref='paper', yref='paper', x=0.5, y=-0.1,*

*xanchor='center', yanchor='top',*

*text='Days after crossing 100 cases ',*

*font=dict(family='Arial',*

*size=12,*

*color='rgb(150,150,150)'),*

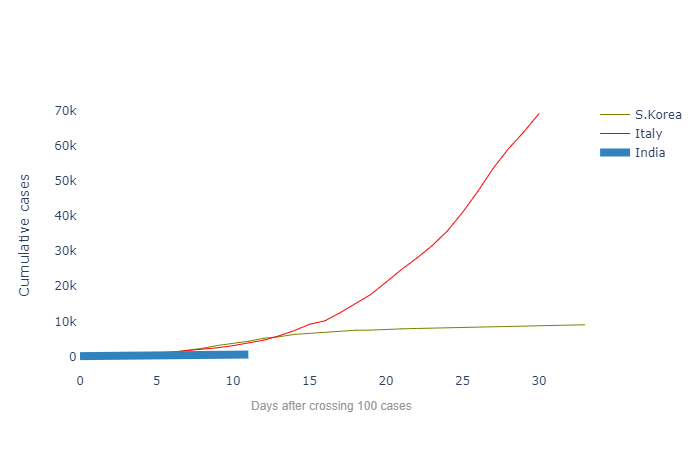
*showarrow=False))*

*fig.update\_layout(annotations=annotations,plot\_bgcolor='white',yaxis\_title='Cumulative cases')*

*fig.write\_html('eight\_figure.html',auto\_open=True)*

*fig.show()*

*eight\_figure*

**

**3. Exploring the world wide data**

*df = pd.read\_csv('covid\_19\_clean\_complete.csv',parse\_dates=['Date'])*

*df.rename(columns={'ObservationDate':'Date', 'Country/Region':'Country'}, inplace=True)*

*df\_confirmed = pd.read\_csv("time\_series\_covid19\_confirmed\_global.csv")*

*df\_recovered = pd.read\_csv("time\_series\_covid19\_recovered\_global.csv")*

*df\_deaths = pd.read\_csv("time\_series\_covid19\_deaths\_global.csv")*

*df\_confirmed.rename(columns={'Country/Region':'Country'}, inplace=True)*

*df\_recovered.rename(columns={'Country/Region':'Country'}, inplace=True)*

*df\_deaths.rename(columns={'Country/Region':'Country'}, inplace=True)*

*x=open("op4.py",'a')*

*x.write((df\_deaths).to\_string())*

*x.close()*

*print(df\_deaths)*

output4

op4.py

*y=open("op4.py")*

*n=7*

*for i in range(n):*

*line=y.readline()*

*z=open("op5.py",'a')*

*print(line)*

*z.write(line)*

*z.close()*

*print(df\_deaths.head)*

output 5

op5.py

*x=open("op6.py",'a')*

*x.write(df.to\_string())*

*x.close()*

*print(df)*

*output 6*

*op6.py*

*y=open("op6.py")*

*n=7*

*for i in range(n):*

*line=y.readline()*

*z=open("op7.py",'a')*

*print(line)*

*z.write(line)*

*z.close()*

*print(df.head())*

*output 7*

*op7.py*

*df2 = df.groupby(["Date", "Country", "Province/State"])[['Date', 'Province/State', 'Country', 'Confirmed', 'Deaths', 'Recovered']].sum().reset\_index()*

*x=open("op8.py",'w')*

*x.write(df2.to\_string())*

*x.close()*

*#print(df2)*

*y=open("op8.py")*

*n=7*

*for i in range(n):*

*line=y.readline()*

*z=open("op9.py",'a')*

*print(line)*

*z.write(line)*

*z.close()*

*print(df2.head())*

*#Check for India's data*

*import sys*

*x=open("op10.py",'w')*

*print(df.query('Country=="India"').groupby("Date")[['Confirmed', 'Deaths', 'Recovered']].sum().reset\_index(),file = x)*

*x.close()*

*output10*

*op10.py*

*#overall worldwide confirmed/Deaths/Recovered cases*

*import sys*

*x=open("op11.py",'w')*

*print(df.groupby('Date').sum().head(),file=x)*

*x.close()*

*output11*

*op11.py*

**3.1 Visualizing the worldwide covid-19 cases**

*confirmed = df.groupby('Date').sum()['Confirmed'].reset\_index()*

*deaths = df.groupby('Date').sum()['Deaths'].reset\_index()*

*recovered = df.groupby('Date').sum()['Recovered'].reset\_index()*

*fig = go.Figure()*

*#Plotting date wise confirmed cases*

*fig.add\_trace(go.Scatter(x=confirmed['Date'], y=confirmed['Confirmed'], mode='lines+markers', name='Confirmed',line=dic tot(color='blue', width=2)))*

*fig.add\_trace(go.Scatter(x=deaths['Date'], y=deaths['Deaths'], mode='lines+markers', name='Deaths', line=dict(color='Red', width=2)))*

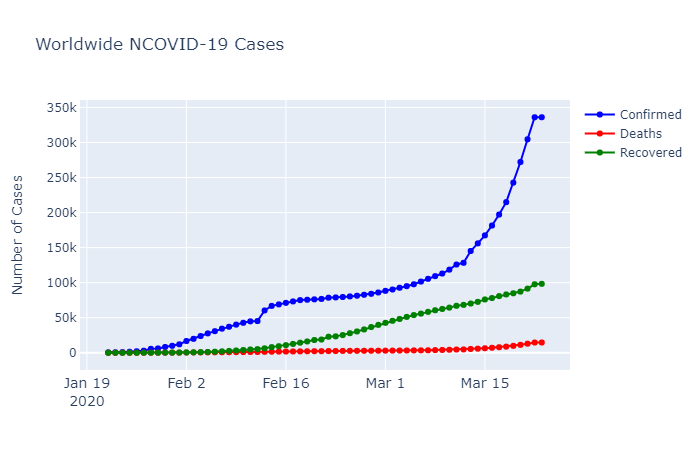
*fig.add\_trace(go.Scatter(x=recovered['Date'], y=recovered['Recovered'], mode='lines+markers', name='Recovered', line=dict(color='Green', width=2)))*

*fig.update\_layout(title='Worldwide NCOVID-19 Cases', xaxis\_tickfont\_size=14,yaxis=dict(title='Number of Cases'))*

*fig.write\_html('ninth\_figure.html',auto\_open=True)*

*fig.show()*

ninth\_figure

**

**4. Forecasting the values of worldwide confirmed,death & recovered cases using prophet**

**4.1 prophet**

*Prophet is an open source software released by Facebook’s Core Data Science team.*

*We use Prophet, a procedure for forecasting time series data based on an additive model where nonlinear 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.*

***Accurate and fast:*** *Prophet is used in many applications across Facebook for producing reliable forecasts for planning and goal setting. Facebook finds it to perform better than any other approach in the majority of cases. It fit models in [Stan](https://mc-stan.org/) so that we can get forecasts in just a few seconds.*

***Fully automatic****: Get a reasonable forecast on messy data with no manual effort. Prophet is robust to outliers, missing data, and dramatic changes in your time series.*

***Tunable forecasts:*** *The Prophet procedure includes many possibilities for users to tweak and adjust forecasts. You can use human-interpretable parameters to improve your forecast by adding your domain knowledge.*

***Available in R or Python****: Facebook has implemented the Prophet procedure in R and Python. Both of them share the same underlying Stan code for fitting. You can use whatever language you’re comfortable with to get forecasts.*

**4.2 Forecasting the values of worldwide confirmed cases**

*from fbprophet import Prophet*

*confirmed = df.groupby('Date').sum()['Confirmed'].reset\_index()*

*deaths = df.groupby('Date').sum()['Deaths'].reset\_index()*

*recovered = df.groupby('Date').sum()['Recovered'].reset\_index()*

*confirmed.columns = ['ds','y']*

*#confirmed['ds'] = confirmed['ds'].dt.date*

*confirmed['ds'] = pd.to\_datetime(confirmed['ds'])*

*x=open("op12.py",'w')*

*print(confirmed.tail())*

*x.close()*

Output 12

Op12.py

*m = Prophet(interval\_width=0.95)*

*m.fit(confirmed)*

*future = m.make\_future\_dataframe(periods=7)*

*x=open(“op13.py”,’w’)*

*print(future.tail())*

*x.close()*

*Output13*

*Op13.py*

*#predicting the future with date, and upper and lower limit of y value*

*forecast = m.predict(future)*

*x=open("op14.py",'w')*

*print(forecast[['ds', 'yhat', 'yhat\_lower', 'yhat\_upper']].tail(),file=x)*

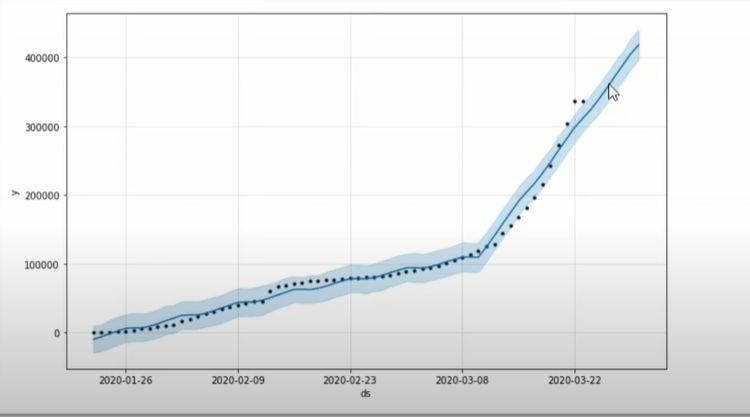
*x.close()*

Output 14

Op14.py

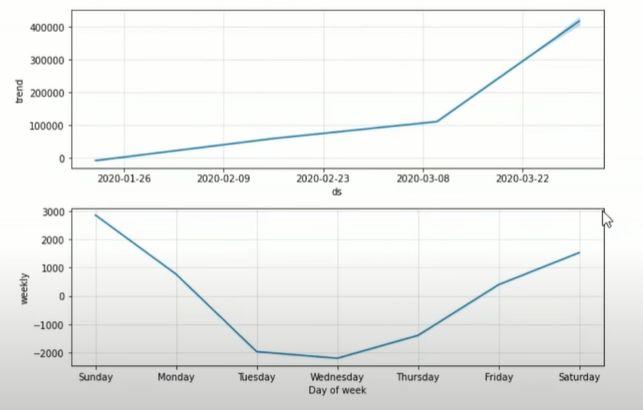
*confirmed\_forecast\_plot = m.plot(forecast)*

tenth\_figure



*confirmed\_forecast\_plot =m.plot\_components(forecast)*

eleventh\_figure



**4.3 Forecasting the worldwide Death using prophet (Base model)**

#Deaths

*deaths.columns = ['ds','y']*

*deaths['ds'] = pd.to\_datetime(deaths['ds'])*

*m = Prophet(interval\_width=0.95)*

*m.fit(deaths)*

*future = m.make\_future\_dataframe(periods=7)*

*print(future.tail())*

*forecast = m.predict(future)*

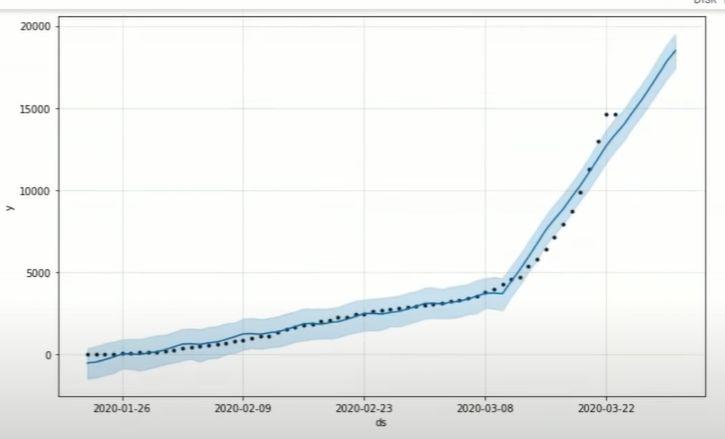
*forecast[['ds', 'yhat', 'yhat\_lower', 'yhat\_upper']].tail()*

Output 15

Op15.py

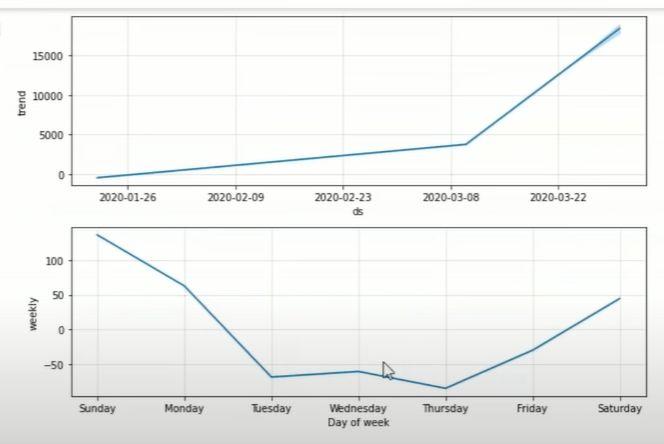
*deaths\_forecast\_plot = m.plot(forecast)*

twelvth\_figure



*deaths\_forecast\_plot = m.plot\_components(forecast)*

thirteenth\_figure



**4.3 Forecasting the worldwide Recovered cases using prophet (Base model)**

*recovered.columns = ['ds','y']*

*recovered['ds'] = pd.to\_datetime(recovered['ds'])*

*m = Prophet(interval\_width=0.95)*

*m.fit(recovered)*

*future = m.make\_future\_dataframe(periods=7)*

*future.tail())*

*forecast = m.predict(future)*

*x=open("op16.py",'w')*

*forecast[['ds', 'yhat', 'yhat\_lower', 'yhat\_upper']].tail()*

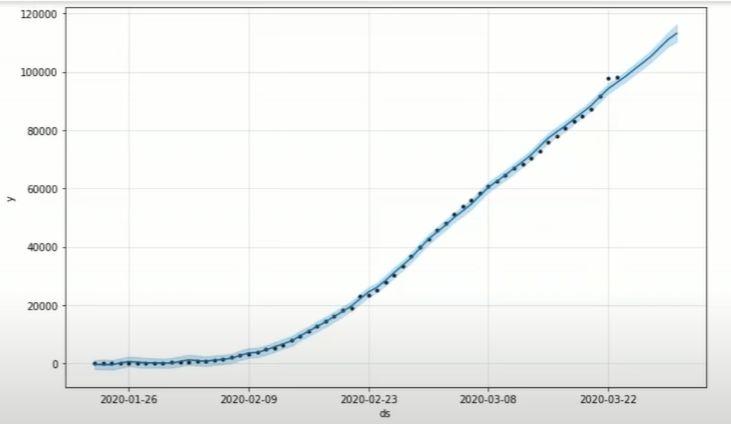
*x.close()*

Output 16

Op16.py

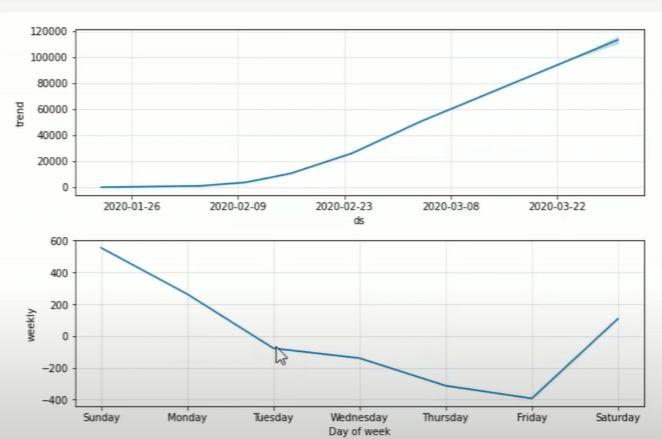
*recovered\_forecast\_plot = m.plot(forecast)*

*fourteenth\_figure*



*recovered\_forecast\_plot = m.plot\_components(forecast)*

fifteenth\_figure



**5.Conclusion & Inference**

As we see from the above graphs and comparing with the other nations like Italy,South korea and Wuhan spread of covid 19 cases in India follow a similar rise. Only from the lockdown the spread can be decreased.

**Inference from the above Analysis and Forecast**

Will Indian become the next Italy/ S.Korea/ Wuhan?

**Yes**, If we look at the world's data and its forecast we can say that India might face one of its worst days if we are not taking strict measures against COVID-19.