

```
In [1]: #Case Study: Analysing the Outbreak of COVID 19 using Machine Learning
#Problem Statement
#We need a strong model that predicts how the virus could spread across d
ifferent countries and regions.
#The goal of this task is to build a model that predicts the spread of th
e virus till 10th of June

#NOTE: The model was built on a test dataset updated till May 25th. But y
ou can access the
#source to these datasets at the 'John Hopkins University Coronavirus Res
ource Centre' which gets updated on a daily basis, so you can run this mo
del for the date you prefer.

#Tasks to be performed:
#Analysing the present condition in India
#Exploring the world wide data
#Forecasting the worldwide COVID-19 cases using Prophet for world and Ind
ia

#importinglibraries

import pandas as pd

#visualization libraries
import matplotlib.pyplot as plt
%matplotlib inline
import seaborn as sns
import plotly.express as px
import plotly.graph_objects as go
import folium
from folium import plugins

#plotsize manipulation
plt.rcParams['figure.figsize']=10,12

#disablewarnings
import warnings
warnings.filterwarnings('ignore')
```

In [2]: *#Reading the Datasets*

```
India_coord=pd.read_excel('Indian Coordinates.xlsx')
df=pd.read_excel('state_wise1.xlsx')
df_india= df.copy()
df
```

Out[2]:

	Name of State / UT	Confirmed	Recovered	Deaths	Active
0	Maharashtra	52667	15786	1695	35186
1	Tamil Nadu	17082	8731	119	8232
2	Gujarat	14468	6636	888	6944
3	Delhi	14053	6771	276	7006
4	Rajasthan	7376	4072	167	3137
5	Madhya Pradesh	6859	3571	300	2988
6	Uttar Pradesh	6497	3660	169	2668
7	West Bengal	3816	1414	278	2124
8	Andhra Pradesh	2983	1947	57	979
9	Bihar	2737	733	13	1991
10	Unassigned	2970	0	0	2970
11	Karnataka	2182	705	44	1431
12	Punjab	2081	1913	40	128
13	Telangana	1920	1164	56	700
14	Jammu and Kashmir	1668	809	23	836
15	Odisha	1438	649	7	782
16	Haryana	1213	802	16	395
17	Kerala	897	532	6	359
18	Assam	549	63	4	479
19	Jharkhand	405	148	4	253
20	Uttarakhand	349	58	4	284
21	Chhattisgarh	292	67	0	225
22	Chandigarh	266	187	4	75
23	Himachal Pradesh	223	63	4	153
24	Tripura	198	165	0	33
25	Goa	67	19	0	48
26	Ladakh	53	43	0	10
27	Puducherry	49	17	0	32
28	Manipur	36	4	0	32
29	Andaman and Nicobar Islands	33	33	0	0
30	Meghalaya	15	12	1	2
31	Nagaland	3	0	0	3
32	Dadra and Nagar Haveli and Daman and Diu	2	1	0	1
33	Arunachal Pradesh	2	1	0	1
34	Mizoram	1	1	0	0
35	Sikkim	1	0	0	1

	Name of State / UT	Confirmed	Recovered	Deaths	Active
36	Lakshadweep	0	0	0	0

In [3]: *#Analysing COVID19 Cases in India*

```
df=pd.read_excel('state_wise1.xlsx')
df_india= df.copy()
df['Total cases']=df['Confirmed']
total_cases= df['Total cases'].sum()
print('Total number of covid confirmed case till 25th may in India:',total_cases)
```

Total number of covid confirmed case till 25th may in India: 145451

In [4]: *#Number of Active COVID-19 cases in affected State/Union Territories*

```
df=pd.read_excel('state_wise1.xlsx')
df_india= df.copy()
df
df.style.background_gradient(cmap='Reds')
```

Out[4]:

	Name of State / UT	Confirmed	Recovered	Deaths	Active
0	Maharashtra	52667	15786	1695	35186
1	Tamil Nadu	17082	8731	119	8232
2	Gujarat	14468	6636	888	6944
3	Delhi	14053	6771	276	7006
4	Rajasthan	7376	4072	167	3137
5	Madhya Pradesh	6859	3571	300	2988
6	Uttar Pradesh	6497	3660	169	2668
7	West Bengal	3816	1414	278	2124
8	Andhra Pradesh	2983	1947	57	979
9	Bihar	2737	733	13	1991
10	Unassigned	2970	0	0	2970
11	Karnataka	2182	705	44	1431
12	Punjab	2081	1913	40	128
13	Telangana	1920	1164	56	700
14	Jammu and Kashmir	1668	809	23	836
15	Odisha	1438	649	7	782
16	Haryana	1213	802	16	395
17	Kerala	897	532	6	359
18	Assam	549	63	4	479
19	Jharkhand	405	148	4	253
20	Uttarakhand	349	58	4	284
21	Chhattisgarh	292	67	0	225
22	Chandigarh	266	187	4	75
23	Himachal Pradesh	223	63	4	153
24	Tripura	198	165	0	33
25	Goa	67	19	0	48
26	Ladakh	53	43	0	10
27	Puducherry	49	17	0	32
28	Manipur	36	4	0	32
29	Andaman and Nicobar Islands	33	33	0	0
30	Meghalaya	15	12	1	2
31	Nagaland	3	0	0	3
32	Dadra and Nagar Haveli and Daman and Diu	2	1	0	1
33	Arunachal Pradesh	2	1	0	1
34	Mizoram	1	1	0	0
35	Sikkim	1	0	0	1

	Name of State / UT	Confirmed	Recovered	Deaths	Active
36	Lakshadweep	0	0	0	0

In [5]: *#Visualising the spread geographically*

```
df_full = pd.merge(India_coord,df,on='Name of State / UT')
map = folium.Map(location=[20, 70], zoom_start=4,tiles='Stamenterrain')
for lat, lon, value, name in zip(df_full['Latitude'], df_full['Longitude'], df_full['Confirmed'], df_full['Name of State / UT']):
    folium.CircleMarker([lat, lon], radius=value*0.003, popup = ('<strong>State</strong>: ' + str(name).capitalize() + '),color='red',fill_color='red',fill_opacity=0.3 ).add_to(map)

map
```

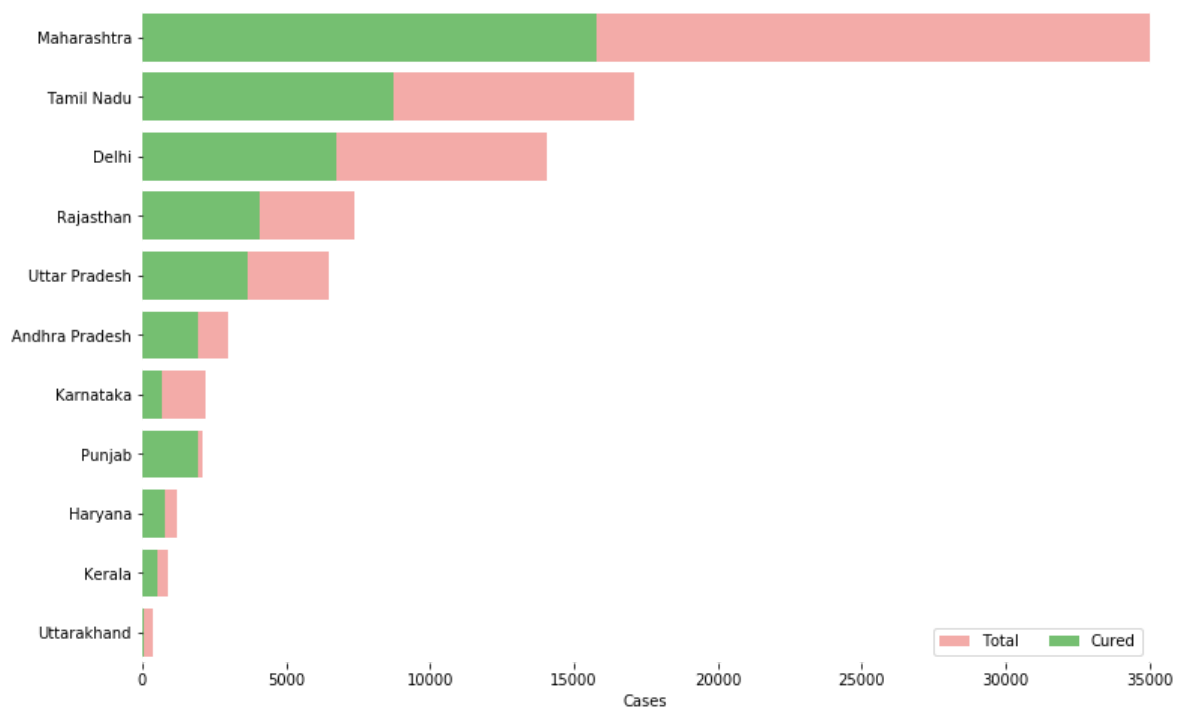
Out[5]: Make this Notebook Trusted to load map: File -> Trust Notebook

In [6]: *#Confirmed vs Recovered figures*

```
f, ax = plt.subplots(figsize=(12,8))
data= df_full[['Name of State / UT','Confirmed','Recovered','Deaths']]
data.sort_values('Confirmed',ascending=False,inplace=True)
sns.set_color_codes("pastel")
sns.barplot(x="Confirmed",y="Name of State / UT",data=data,label="Total",
color="r")

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

ax.legend(ncol=2, loc= "lower right", frameon=True)
ax.set(xlim=(0,35000), ylabel="", xlabel="Cases")
sns.despine(left=True,bottom=True)
```



In [7]: *#Exploring Worldwide 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)
df_deaths.head()
```

Out[7]:

	Province/State	Country	Lat	Long	1/22/20	1/23/20	1/24/20	1/25/20	:
0	NaN	Afghanistan	33.0000	65.0000	0	0	0	0	
1	NaN	Albania	41.1533	20.1683	0	0	0	0	
2	NaN	Algeria	28.0339	1.6596	0	0	0	0	
3	NaN	Andorra	42.5063	1.5218	0	0	0	0	
4	NaN	Angola	-11.2027	17.8739	0	0	0	0	

5 rows × 130 columns

```
df2 = df.groupby(["Date", "Country", "Province/State"])[['Date', 'Province/State', 'Country', 'Confirmed', 'Deaths', 'Recovered']].sum().reset_index()
df2.head()
```

Out[8]:

	Date	Country	Province/State	Confirmed	Deaths	Recovered
0	2020-01-22	Australia	Australian Capital Territory	0	0	0
1	2020-01-22	Australia	New South Wales	0	0	0
2	2020-01-22	Australia	Northern Territory	0	0	0
3	2020-01-22	Australia	Queensland	0	0	0
4	2020-01-22	Australia	South Australia	0	0	0

```
In [9]: df.query('Country=="India").groupby("Date")[['Confirmed', 'Deaths', 'Recovered']].sum().reset_index()
```

Out[9]:

	Date	Confirmed	Deaths	Recovered
0	2020-01-22	0	0	0
1	2020-01-23	0	0	0
2	2020-01-24	0	0	0
3	2020-01-25	0	0	0
4	2020-01-26	0	0	0
...
120	2020-05-21	118226	3584	48553
121	2020-05-22	124794	3726	51824
122	2020-05-23	131423	3868	54385
123	2020-05-24	138536	4024	57692
124	2020-05-25	144950	4172	60706

125 rows × 4 columns

```
In [10]: df.groupby("Date").sum().head()
```

Out[10]:

	Lat	Long	Confirmed	Deaths	Recovered
Date					
2020-01-22	5613.201163	6063.516762	555	17	28
2020-01-23	5613.201163	6063.516762	654	18	30
2020-01-24	5613.201163	6063.516762	941	26	35
2020-01-25	5613.201163	6063.516762	1434	42	38
2020-01-26	5613.201163	6063.516762	2118	56	51

```
In [11]: confirmed = df.groupby('Date').sum()['Confirmed'].reset_index()
deaths = df.groupby('Date').sum()['Deaths'].reset_index()
recovered = df.groupby('Date').sum()['Recovered'].reset_index()
```

```
In [12]: #Forecasting Total Number of Cases Worldwide
#In this segment, we're going to generate a week ahead forecast of
#confirmed cases of COVID-19 using Prophet, with specific prediction
#intervals by creating a base model both with and without tweaking of
#seasonality-related parameters and additional regressors.

from fbprophet import Prophet
```

```
In [13]: confirmed=df.groupby('Date').sum()['Confirmed'].reset_index()
deaths=df.groupby('Date').sum()['Deaths'].reset_index()
recovered=df.groupby('Date').sum()['Recovered'].reset_index()
```

```
In [14]: confirmed.columns=['ds','y']
confirmed['ds']=pd.to_datetime(confirmed['ds'])
```

```
In [15]: confirmed.tail()
```

```
Out[15]:
```

	ds	y
120	2020-05-21	5102418
121	2020-05-22	5210811
122	2020-05-23	5310356
123	2020-05-24	5407607
124	2020-05-25	5495055

```
In [17]: #Forecasting Confirmed COVID-19 Cases Worldwide with Prophet (Base model)
```

```
m = Prophet(interval_width=0.95)
m.fit(confirmed)
future = m.make_future_dataframe(periods=16)
future.tail()
```

INFO:fbprophet:Disabling yearly seasonality. Run prophet with yearly_seasonality=True to override this.
INFO:fbprophet:Disabling daily seasonality. Run prophet with daily_seasonality=True to override this.

```
Out[17]:
```

	ds
136	2020-06-06
137	2020-06-07
138	2020-06-08
139	2020-06-09
140	2020-06-10

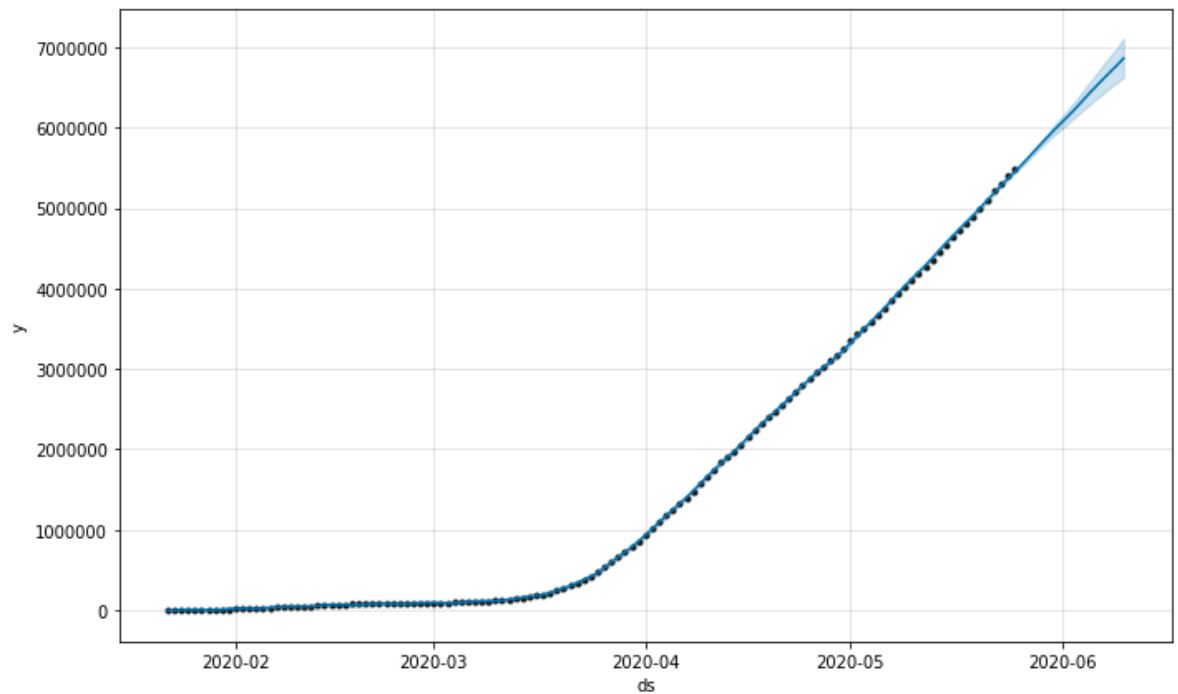
```
In [18]: #predicting the future with date, and upper and lower limit of y value
```

```
forecast = m.predict(future)
forecast[['ds', 'yhat', 'yhat_lower', 'yhat_upper']].tail()
```

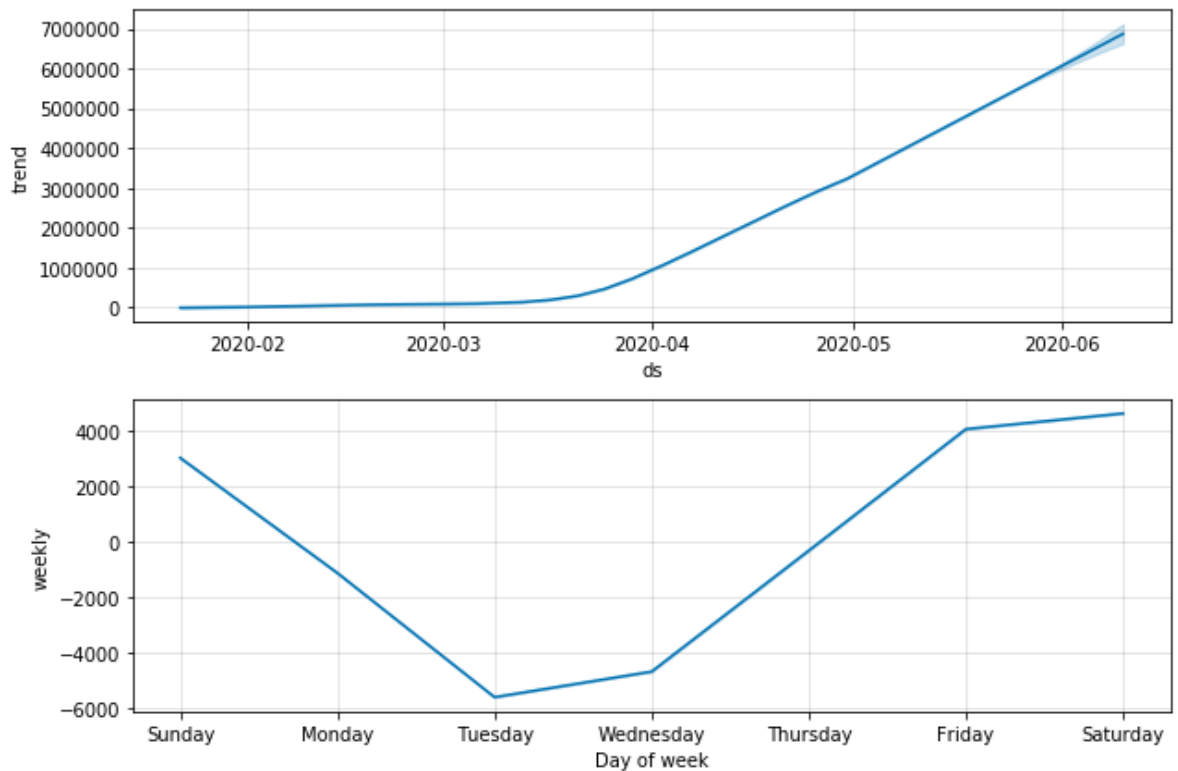
```
Out[18]:
```

	ds	yhat	yhat_lower	yhat_upper
136	2020-06-06	6.516870e+06	6.359295e+06	6.668509e+06
137	2020-06-07	6.603948e+06	6.423199e+06	6.778890e+06
138	2020-06-08	6.688484e+06	6.497919e+06	6.887997e+06
139	2020-06-09	6.772695e+06	6.558421e+06	6.988288e+06
140	2020-06-10	6.862291e+06	6.627281e+06	7.110045e+06

```
In [19]: confirmed_forecast_plot=m.plot(forecast)
```



```
In [20]: confirmed_forecast_plot =m.plot_components(forecast)
```



```
In [21]: #Forecasting Worldwide Recovered using Prophet (Base model)
```

```
recovered.columns=['ds','y']  
recovered['ds']=pd.to_datetime(confirmed['ds'])
```

```
In [22]: recovered.tail()
```

Out[22]:

	ds	y
120	2020-05-21	1895640
121	2020-05-22	2001920
122	2020-05-23	2056599
123	2020-05-24	2112135
124	2020-05-25	2174434

```
In [29]: m = Prophet(interval_width=0.95)
m.fit(recovered)
futurerecovered = m.make_future_dataframe(periods=16)
futurerecovered.tail()
```

INFO:fbprophet:Disabling yearly seasonality. Run prophet with yearly_seasonality=True to override this.

INFO:fbprophet:Disabling daily seasonality. Run prophet with daily_seasonality=True to override this.

Out[29]:

	ds
136	2020-06-06
137	2020-06-07
138	2020-06-08
139	2020-06-09
140	2020-06-10

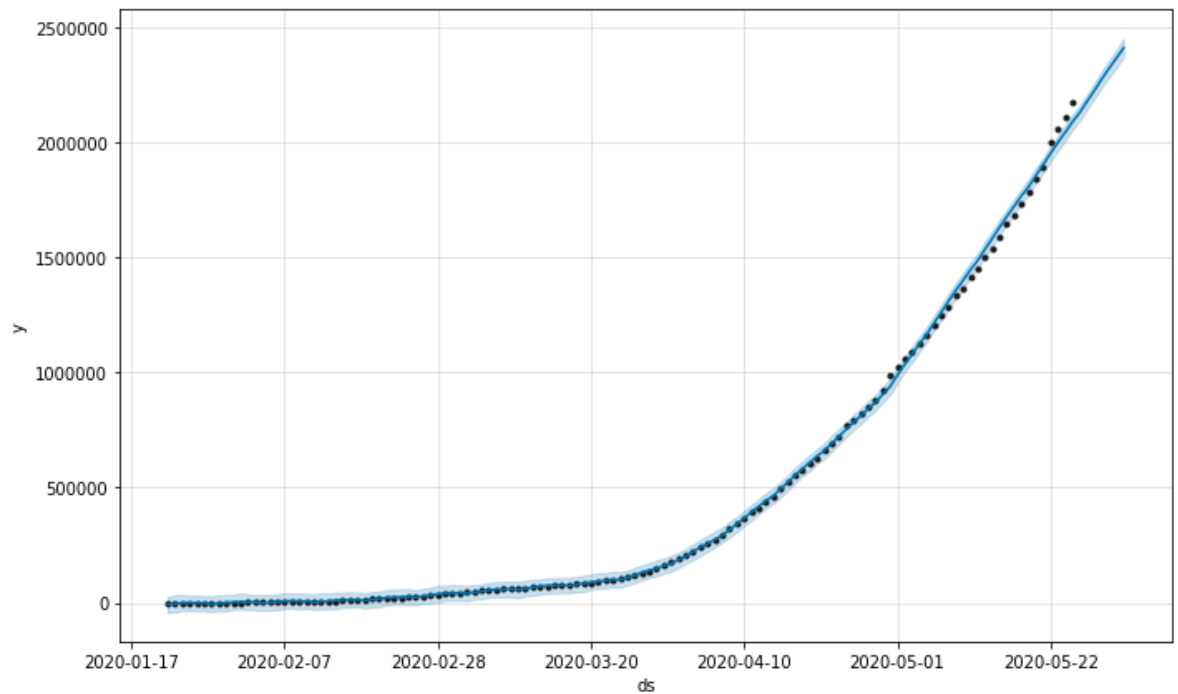
```
In [30]: ##predicting the future with date, and upper and lower limit of y value

forecastrecovered = m.predict(future)
forecastrecovered[['ds', 'yhat', 'yhat_lower', 'yhat_upper']].tail()
```

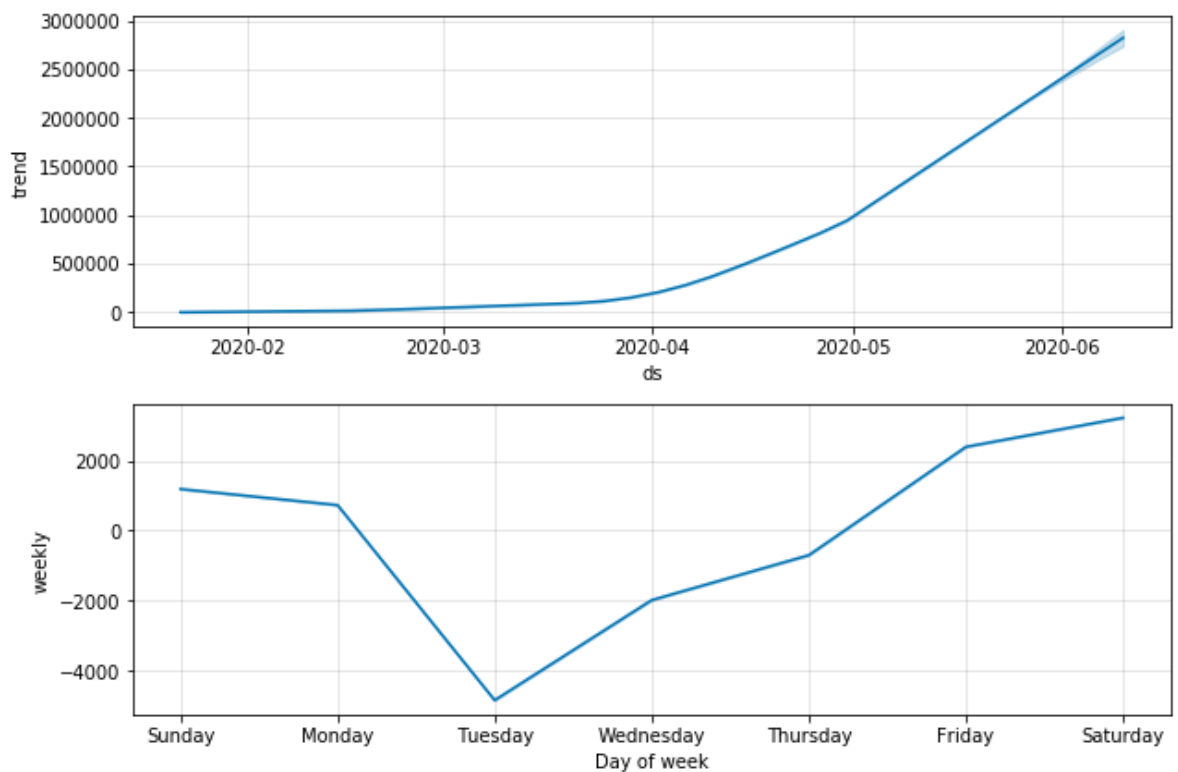
Out[30]:

	ds	yhat	yhat_lower	yhat_upper
127	2020-05-28	2.226445e+06	2.193217e+06	2.261252e+06
128	2020-05-29	2.275507e+06	2.240956e+06	2.311941e+06
129	2020-05-30	2.322304e+06	2.284575e+06	2.356949e+06
130	2020-05-31	2.366230e+06	2.326034e+06	2.404002e+06
131	2020-06-01	2.411735e+06	2.371639e+06	2.453192e+06

```
In [31]: confirmed_forecast_plot=m.plot(forecastrecovered)
```



```
In [46]: confirmed_forecastrecovered_plot =m.plot_components(forecastrecovered)
```



```
In [32]: #Forecasting Worldwide Deaths using Prophet (Base model)
```

```
deaths.columns=['ds','y']  
deaths['ds']=pd.to_datetime(confirmed['ds'])
```

```
In [33]: deaths.tail()
```

```
Out[33]:
```

	ds	y
120	2020-05-21	332924
121	2020-05-22	338160
122	2020-05-23	342097
123	2020-05-24	345059
124	2020-05-25	346232

```
In [34]: m = Prophet(interval_width=0.95)
m.fit(deaths)
futuredeaths = m.make_future_dataframe(periods=16)
futuredeaths.tail()
```

INFO:fbprophet:Disabling yearly seasonality. Run prophet with yearly_seasonality=True to override this.

INFO:fbprophet:Disabling daily seasonality. Run prophet with daily_seasonality=True to override this.

```
Out[34]:
```

	ds
136	2020-06-06
137	2020-06-07
138	2020-06-08
139	2020-06-09
140	2020-06-10

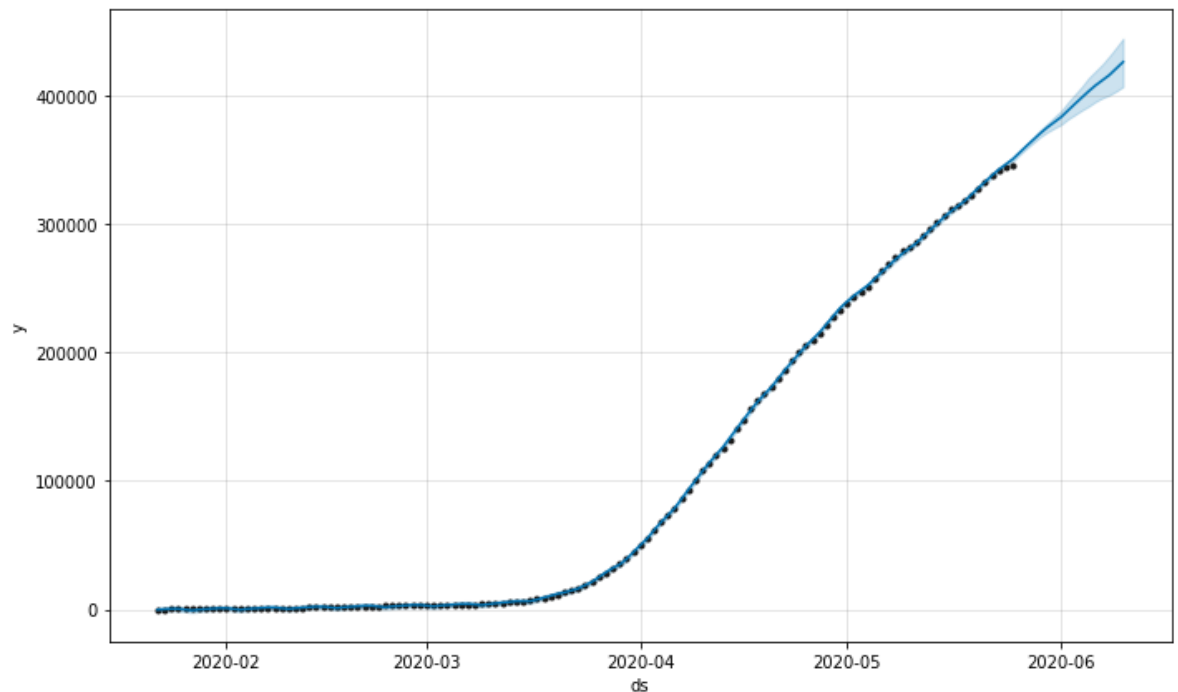
```
In [35]: #predicting the future with date, and upper and lower limit of y value
```

```
forecastdeaths = m.predict(future)
forecastdeaths[['ds', 'yhat', 'yhat_lower', 'yhat_upper']].tail()
```

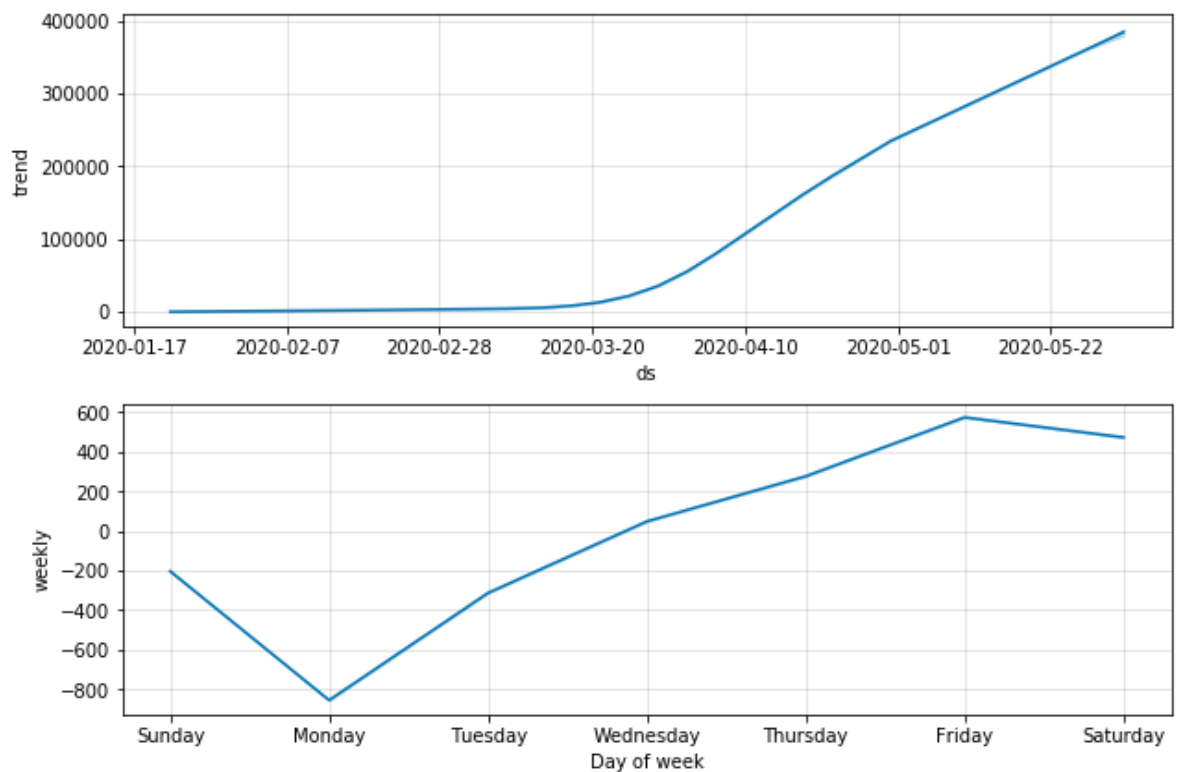
```
Out[35]:
```

	ds	yhat	yhat_lower	yhat_upper
127	2020-05-28	366151.876295	364034.426147	368129.886236
128	2020-05-29	371131.408469	368372.540975	373571.760777
129	2020-05-30	375711.904525	372103.179949	378662.981568
130	2020-05-31	379716.622592	375644.621934	383371.021921
131	2020-06-01	383746.307490	378572.823887	388627.164023

```
In [52]: death_forecast_plot=m.plot(forecastdeaths)
```



```
In [36]: confirmed_forecastdeath_plot =m.plot_components(forecastdeaths)
```



```
In [37]: #Forecasting Indian Corona numbers using Prophet model
```

```
df2=df.query('Country=="India"').groupby("Date")[['Confirmed','Deaths','R  
ecovered']].sum().reset_index()
```



```
In [38]: df2.groupby('Date').sum().tail()
```

Out[38]:

	Confirmed	Deaths	Recovered
Date			
2020-05-21	118226	3584	48553
2020-05-22	124794	3726	51824
2020-05-23	131423	3868	54385
2020-05-24	138536	4024	57692
2020-05-25	144950	4172	60706

```
In [39]: confirmedindia = df2.groupby('Date').sum()['Confirmed'].reset_index()  
deathsindia = df2.groupby('Date').sum()['Deaths'].reset_index()  
recoveredindia = df2.groupby('Date').sum()['Recovered'].reset_index()
```

```
In [40]: from fbprophet import Prophet
```

```
In [41]: confirmedindia = df2.groupby('Date').sum()['Confirmed'].reset_index()  
deathsindia = df2.groupby('Date').sum()['Deaths'].reset_index()  
recoveredindia = df2.groupby('Date').sum()['Recovered'].reset_index()
```

```
In [42]: #Forecasting Confirmed COVID-19 Cases in India with Prophet (Base model)  
  
confirmedindia.columns=['ds','y']  
confirmedindia['ds']=pd.to_datetime(confirmedindia['ds'])
```

```
In [43]: confirmedindia.tail()
```

Out[43]:

	ds	y
120	2020-05-21	118226
121	2020-05-22	124794
122	2020-05-23	131423
123	2020-05-24	138536
124	2020-05-25	144950

```
In [44]: m = Prophet(interval_width=0.95)
m.fit(confirmedindia)
future = m.make_future_dataframe(periods=16)
future.tail()
```

INFO:fbprophet:Disabling yearly seasonality. Run prophet with yearly_seasonality=True to override this.
INFO:fbprophet:Disabling daily seasonality. Run prophet with daily_seasonality=True to override this.

Out[44]:

	ds
136	2020-06-06
137	2020-06-07
138	2020-06-08
139	2020-06-09
140	2020-06-10

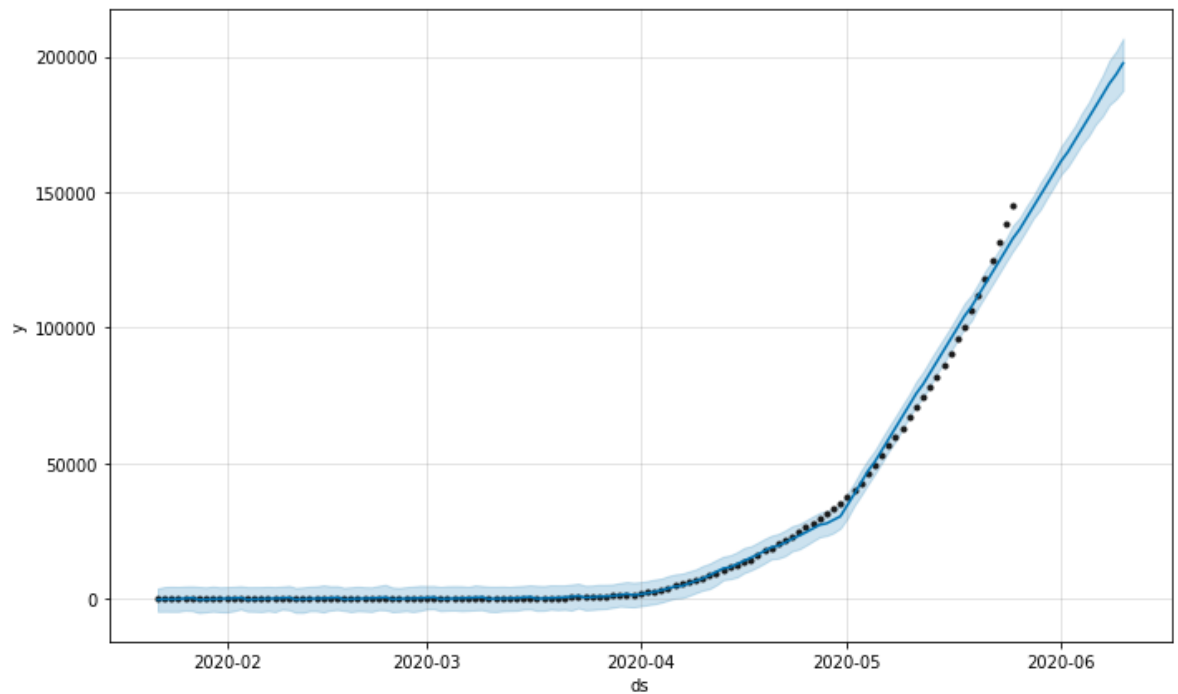
```
In [45]: ##predicting the future with date, and upper and lower limit of y value

forecastconfirmedindia = m.predict(future)
forecastconfirmedindia[['ds', 'yhat', 'yhat_lower', 'yhat_upper']].tail()
```

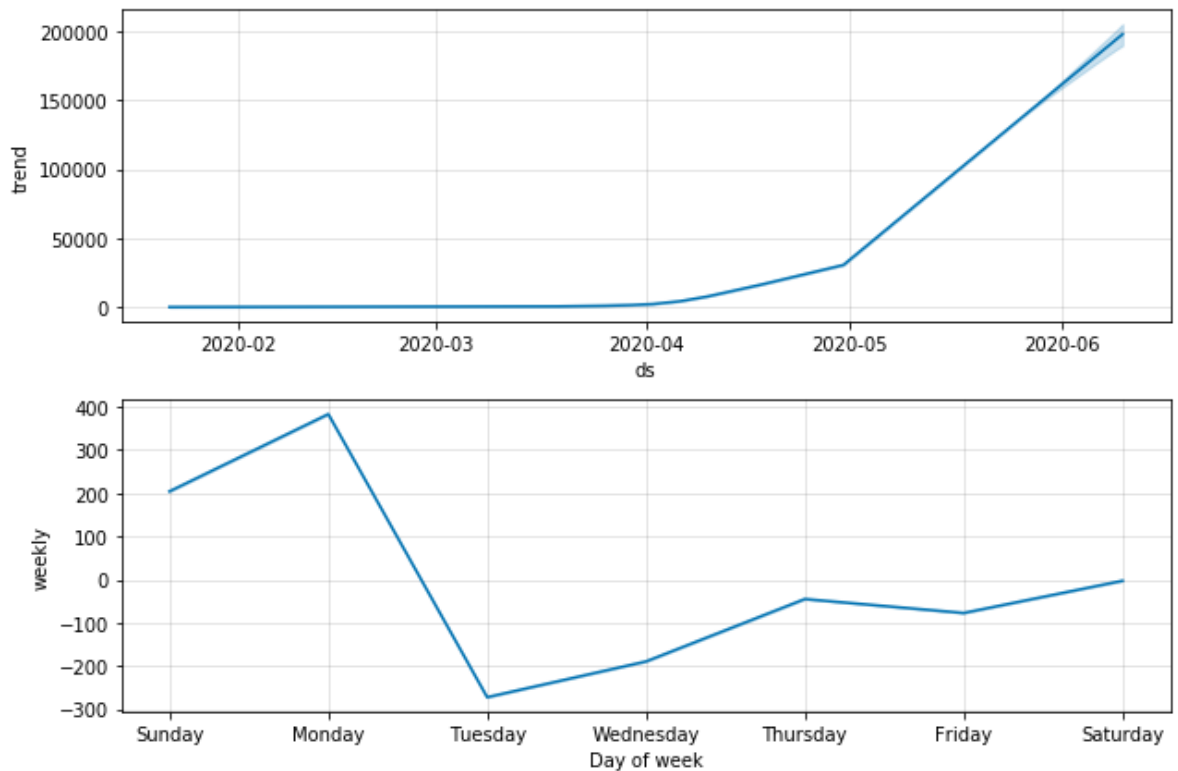
Out[45]:

	ds	yhat	yhat_lower	yhat_upper
136	2020-06-06	181503.092433	174578.872358	187961.995509
137	2020-06-07	185794.325700	178362.272098	192526.895954
138	2020-06-08	190057.525876	182501.560655	198250.414309
139	2020-06-09	193486.807298	185548.981870	201684.033971
140	2020-06-10	197653.979586	188492.450004	207277.623681

```
In [76]: confirmedindia_forecast_plot=m.plot(forecastconfirmedindia)
```



```
In [46]: confirmed_forecastindia_plot =m.plot_components(forecastconfirmedindia)
```



```
In [47]: #Forecasting Worldwide Recovered Cases with Prophet (Base model)
```

```
recoveredindia.columns=['ds','y']  
recoveredindia['ds']=pd.to_datetime(recoveredindia['ds'])
```

```
In [48]: recoveredindia.tail()
```

```
Out[48]:
```

	ds	y
120	2020-05-21	48553
121	2020-05-22	51824
122	2020-05-23	54385
123	2020-05-24	57692
124	2020-05-25	60706

```
In [49]: m = Prophet(interval_width=0.95)
m.fit(recoveredindia)
future = m.make_future_dataframe(periods=16)
future.tail()
```

INFO:fbprophet:Disabling yearly seasonality. Run prophet with yearly_seasonality=True to override this.

INFO:fbprophet:Disabling daily seasonality. Run prophet with daily_seasonality=True to override this.

```
Out[49]:
```

	ds
136	2020-06-06
137	2020-06-07
138	2020-06-08
139	2020-06-09
140	2020-06-10

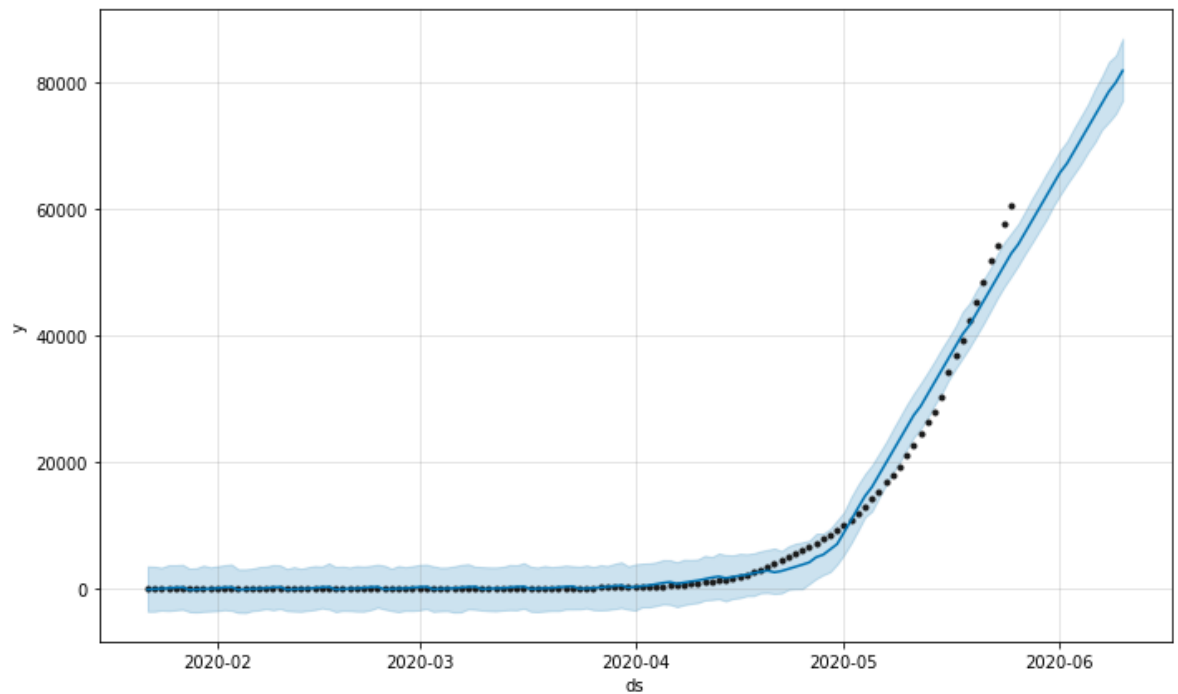
```
In [50]: ##predicting the future with date, and upper and lower limit of y value

forecastrecoveredindia = m.predict(future)
forecastrecoveredindia[['ds', 'yhat', 'yhat_lower', 'yhat_upper']].tail()
```

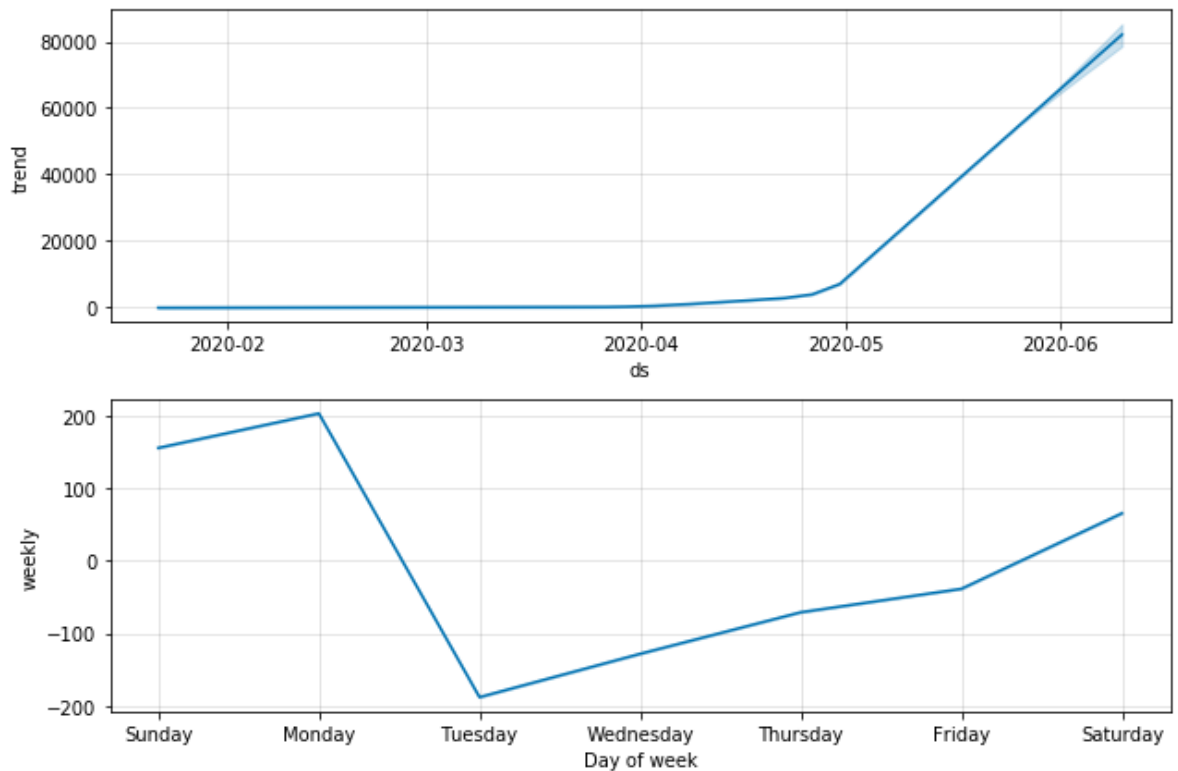
```
Out[50]:
```

	ds	yhat	yhat_lower	yhat_upper
136	2020-06-06	74807.689360	70568.960993	78920.599904
137	2020-06-07	76728.181421	72627.884754	81369.324345
138	2020-06-08	78606.138343	73968.091868	83249.060330
139	2020-06-09	80045.108980	75398.490725	84409.433249
140	2020-06-10	81935.291070	76889.107487	86434.574242

```
In [82]: recoveredindia_forecast_plot=m.plot(forecastrecoveredindia)
```



```
In [51]: recovered_forecastindia_plot =m.plot_components(forecastrecoveredindia)
```



```
In [52]: #Forecasting Deaths in India using Prophet (Base model)
```

```
deathsindia.columns=['ds','y']
deathsindia['ds']=pd.to_datetime(deathsindia['ds'])
```

```
In [53]: deathsindia.tail()
```

Out[53]:

	ds	y
120	2020-05-21	3584
121	2020-05-22	3726
122	2020-05-23	3868
123	2020-05-24	4024
124	2020-05-25	4172

```
In [54]: m = Prophet(interval_width=0.95)
m.fit(deathsindia)
future = m.make_future_dataframe(periods=16)
future.tail()
```

INFO:fbprophet:Disabling yearly seasonality. Run prophet with yearly_seasonality=True to override this.

INFO:fbprophet:Disabling daily seasonality. Run prophet with daily_seasonality=True to override this.

Out[54]:

	ds
136	2020-06-06
137	2020-06-07
138	2020-06-08
139	2020-06-09
140	2020-06-10

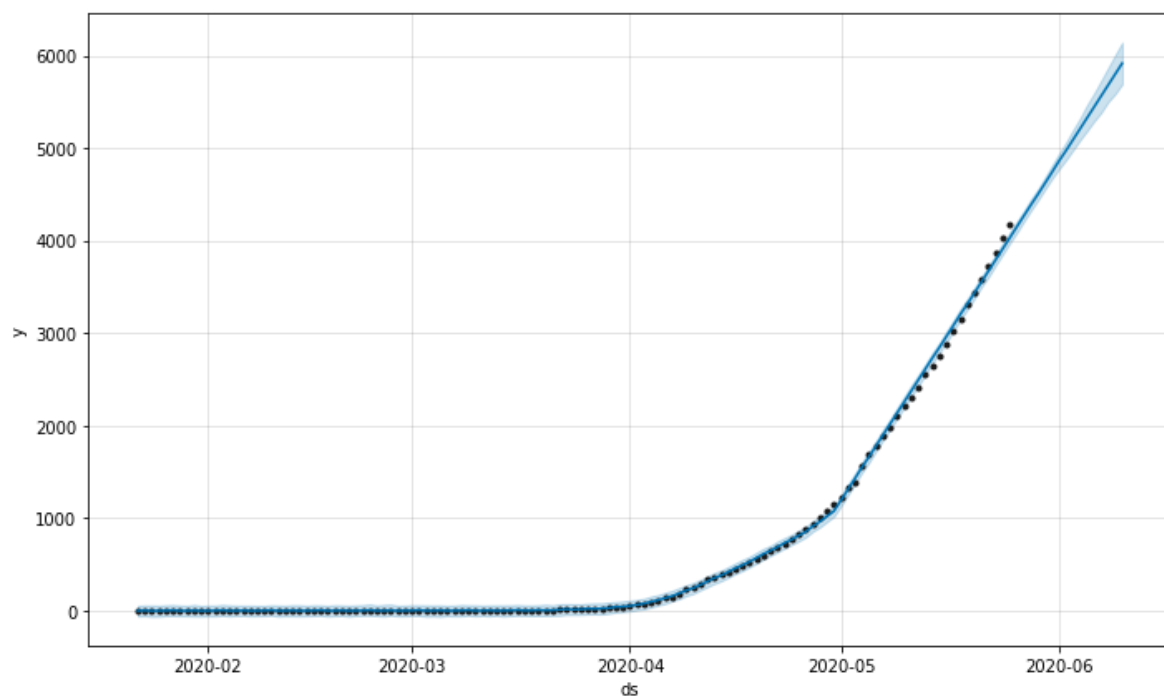
```
In [55]: #predicting the future with date, and upper and lower limit of y value

forecastdeathsindia = m.predict(future)
forecastdeathsindia[['ds', 'yhat', 'yhat_lower', 'yhat_upper']].tail()
```

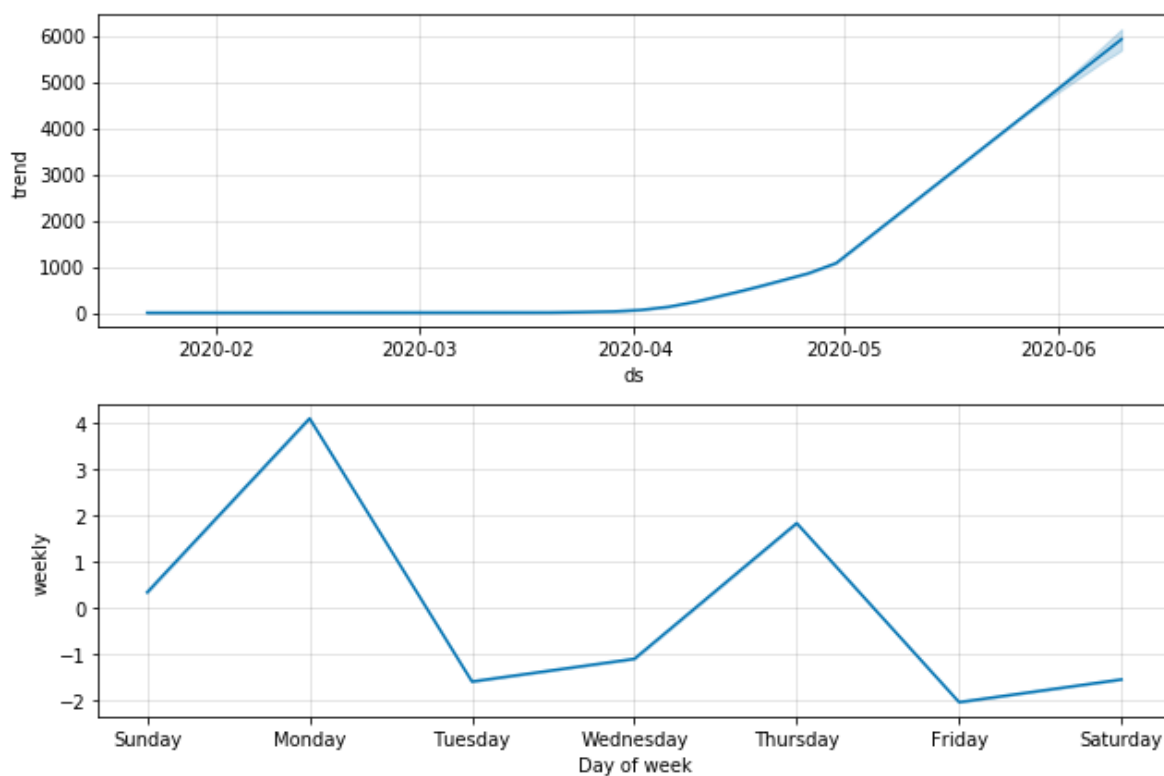
Out[55]:

	ds	yhat	yhat_lower	yhat_upper
136	2020-06-06	5442.095366	5289.232928	5592.854621
137	2020-06-07	5562.118480	5382.214108	5729.748718
138	2020-06-08	5684.020650	5498.165683	5868.632007
139	2020-06-09	5796.449597	5584.416726	6006.279326
140	2020-06-10	5915.074229	5690.410408	6141.559095

```
In [56]: deathstndia_forecast_plot=m.plot(forecastdeathstndia)
```



```
In [57]: deathstndia_forecast_plot =m.plot_components(forecastdeathstndia)
```



```
In [ ]: #Conclusion
        #This is a humble request to all our learners.
        #Don't take your cough and cold lightly as you would. If you look at the
        data, the number of cases in India is rising just like in USA. We will r
        each mark of 200,000 cases by 10th June. Don't let lower awareness and f
        ewer test numbers ruin the health of our world.
        #But the Best part here is Recovery rate is rising at the better pace as
        compared to confirmed cases it can reach mark of 82000 by 10th of June.
        #It shows us that if there are 100 patients getting admitted on 1st day o
        n the 14th day around 97 patients will receive discharge ie.2.8% fatality
        rate
        #Let's give a hand in fighting this pandemic at least by quarantining our
        selves by staying indoors and protecting ourselves and others around us.
        #Take precautions and stay indoors.
```