



Project : Analyzing the trends of COVID-19 with Python

Problem Statement:

Given data about COVID-19 patients, write code to visualize the impact and analyze the trend of rate of infection and recovery as well as make predictions about the number of cases expected a week in future based on the current trends.

Dataset:

CSV and Excel files containing data about the number of COVID-19 confirmed deaths and recovered patients both around the world and in India. Download Link

Guidelines:

- Use pandas to accumulate data from multiple data files.
- Use plotly (visualization library) to create interactive visualizations.
- Use Facebook prophet library to make time series models.
- Visualize the prediction by combining these technologies.

```
[17] ✓ 0s
#Import required libraries
import pandas as pd
import plotly.express as px
from prophet import Prophet
```

```
[18] ✓ 0s
#Load the CSV file
df = pd.read_csv('covid_19_clean_complete.csv')
df.head()
```

...	Province/State	Country/Region	Lat	Long	Date	Confirmed	Deaths	Recovered	Active	WHO Region
0	NaN	Afghanistan	33.93911	67.709953	2020-01-22	0	0	0	0	Eastern Mediterranean
1	NaN	Albania	41.15330	20.168300	2020-01-22	0	0	0	0	Europe
2	NaN	Algeria	28.03390	1.659600	2020-01-22	0	0	0	0	Africa
3	NaN	Andorra	42.50630	1.521800	2020-01-22	0	0	0	0	Europe
4	NaN	Angola	-11.20270	17.873900	2020-01-22	0	0	0	0	Africa

Next steps: [New interactive sheet](#)

FIG 1: IMPORTING THE REQUIRED LIBRARIES AND LOADING THE CSV FILE PROVIDED

```
[19] ✓ 0s
#Check columns
df.columns

Index(['Province/State', 'Country/Region', 'Lat', 'Long', 'Date', 'Confirmed',
       'Deaths', 'Recovered', 'Active', 'WHO Region'],
      dtype='object')
```

```
[20] ✓ 0s
#Convert Date column
df["Date"] = pd.to_datetime(df["Date"])
```

FIG 2: CHECKING THE COLUMNS AND CONVERTING THE DATE COLUMN

```
[21] ✓ 0s
#Create date-wise aggregated data
date_wise = df.groupby("Date")[["Confirmed", "Recovered", "Deaths"]].sum().reset_index()
date_wise.head()
```

...	Date	Confirmed	Recovered	Deaths
0	2020-01-22	555	28	17
1	2020-01-23	654	30	18
2	2020-01-24	941	36	26
3	2020-01-25	1434	39	42
4	2020-01-26	2118	52	56

Next steps: [New interactive sheet](#)

FIG 3: CREATING DATE – WISE AGGREGATED DATA

```
[22] ✓ 1s
#Visualizing infection trend
fig = px.line(
    date_wise,
    x="Date",
    y="Confirmed",
    title="COVID-19 Infection Trend (Confirmed Cases)"
)
fig.show()
```

FIG 4: VISUALIZING THE INFECTION TREND

COVID-19 Infection Trend (Confirmed Cases)

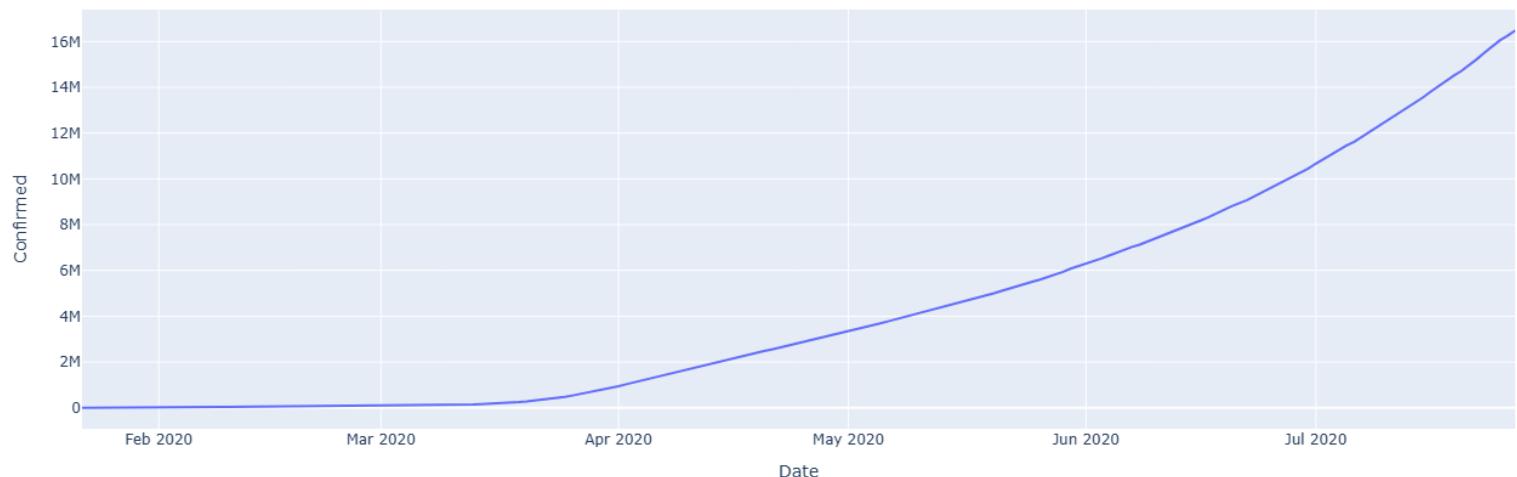


FIG 5: INFECTION TREND

[23]
✓ 0s

Visualizing the recovery trend
fig = px.line(
 date_wise,
 x="Date",
 y="Recovered",
 title="COVID-19 Recovery Trend"
)
fig.show()

FIG 6: VISUALISING THE RECOVERY TREND

COVID-19 Recovery Trend

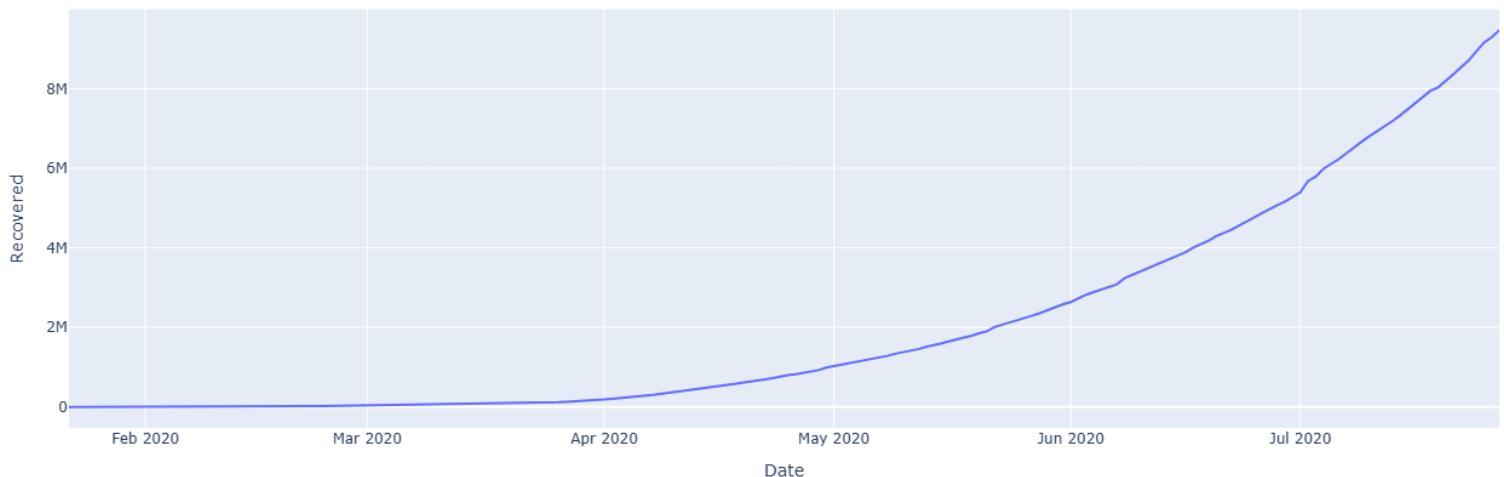


FIG 7: RECOVERY TREND

```
[24] ✓ 0s #Prepare data for prediction
prophet_df = date_wise[["Date", "Confirmed"]]
prophet_df.columns = ["ds", "y"]

[25] ✓ 0s #Train Prophet model
model = Prophet()
model.fit(prophet_df)

INFO:prophet:Disabling yearly seasonality. Run prophet with yearly_seasonality=True to override this.
INFO:prophet:Disabling daily seasonality. Run prophet with daily_seasonality=True to override this.
<prophet.forecaster.Prophet at 0x797a69900da0>

[26] ✓ 0s #Predict next 7 days
future = model.make_future_dataframe(periods=7)
forecast = model.predict(future)

[27] ✓ 0s #Visualizing the prediction
model.plot(forecast)
```

FIG 8: PREPARING, TRAINING, PREDICTING, VISUALISING THE PREDICTION

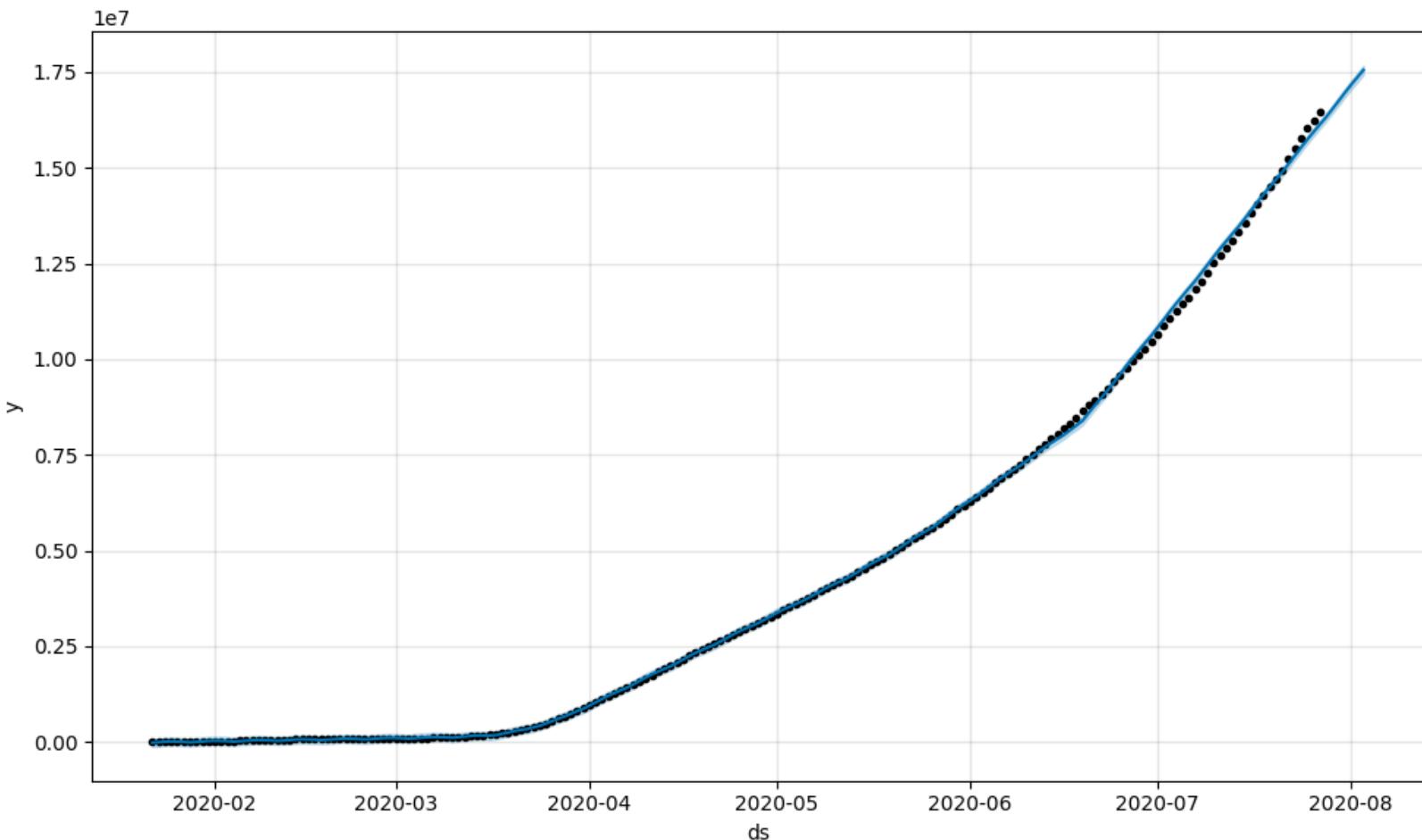


FIG 9: VISUALISING THE PREDICTION