# importing the required libraries

import pandas as pd

# Visualisation 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

# Manipulating the default plot size

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

# Disable warnings

import warnings

warnings.filterwarnings('ignore')

# Reading the datasets

# Coordinates of India States and Union Territories

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

#Day by day data of India, Korea, Italy and Wuhan

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

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

dbd\_Korea = pd.read\_excel('/content/per\_day\_cases.xlsx',parse\_dates=True, sheet\_name="Korea")

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

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 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)

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

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

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 inzip(df\_full['Latitude'], df\_full['Longitude'], df\_full['Total cases'], df\_full['Name of State / UT']):

 folium.CircleMarker([lat, lon], radius=value\*0.8, popup = ('<strong>State</strong>: ' + str(name).capitalize() + '

''<strong>Total Cases</strong>: ' + str(value) + '

'),color='red',fill\_color='red',fill\_opacity=0.3 ).add\_to(map)

Map

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")

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

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

# 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)

# Rise of COVID-19 cases in 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.show()

import plotly.express as px

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

fig.update\_layout(title\_text='Coronavirus Cases in India on daily basis',plot\_bgcolor='rgb(230, 230, 230)')

fig.show()

# 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)

'''Colour Scale for plotly

<a href="https://plot.ly/python/builtin-colorscales/">https://plot.ly/python/builtin-colorscales/</a>

'''

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

fig.show()

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.show()

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.show()

fig = px.bar(dbd\_Wuhan, x="Date", y="Total Cases", color='Total Cases', orientation='v', height=600,

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

fig.show()

# 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.show()

# 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.show()

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

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

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

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

df\_deaths = pd.read\_csv("/content/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()

fig = go.Figure()

#Plotting datewise confirmed cases

fig.add\_trace(go.Scatter(x=confirmed['Date'], y=confirmed['Confirmed'], mode='lines+markers', name='Confirmed',line=dict(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.show()

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)

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

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