**Predicting Coronavirus Cases with Prophet**

by

Shubham Singh (17BCE1086)

Abhijit Pingle (17BCE1254)

A project report submitted to

**Dr. Shridevi K**

**SCHOOL OF COMPUTER SCIENCE AND ENGINEERING**

**in partial fulfilment of the requirements for the course of**

**CSE3024 – WEB MINING**

in

**B.Tech. (Computer Science Engineering)**

****

**VIT UNIVERSITY, CHENNAI**

**Vandalur – Kelambakkam Road**

**Chennai – 600127**

**APRIL 2020**

**CERTIFICATE**

Certified that this project report entitled “**Predicting Coronavirus Cases with Prophet”** is a bonafide work of Shubham Singh (17BCE1086) and Abhijit Pingle (17BCE1254) who carried out the “J”-Project work under my supervision and guidance for CSE3024 – WEB MINING.

**FACULTY:**

**Dr. Shridevi Krishna Kumar**

**SCHOOL OF COMPUTING SCIENCE AND ENGINEERING**

VIT University, Chennai

Chennai – 600127.

**ABSTRACT**

Globally, there is massive uptake and explosion of data and challenge is to address issues like scale, pace, velocity, variety, volume and complexity of the data. Considering the recent epidemic in China, modelling of COVID-19 epidemic for cumulative number of infected cases using data available in early phase was big challenge. Being COVID-19 pandemic during very short time span, it is very important to analyse the trend of these spread and infected cases. These predictive analytics can be empowered using Information, Communication and Technologies (ICT) services, tools and applications. This paper presents medical perspective of COVID-19 towards epidemiological triad and the study of state-of-the-art. The main aim this paper is to present different predictive analytics techniques available for trend analysis, different models and algorithms and their comparison. Finally, this paper concludes with prediction of COVID-19 using Prophet algorithm indicating more faster spread in short term. These predictions will be useful to government and healthcare communities to initiate appropriate measures to control this outbreak in time. Epidemic is a rapid and wide spread of infectious disease threatening many lives and economy damages. It is important to fore-tell the epidemic lifetime so to decide on timely and remedic actions. These measures include closing borders, schools, suspending community services and commuters. Resuming such curfews depends on the momentum of the outbreak and its rate of decay. Being able to accurately forecast the fate of an epidemic is an extremely important but difficult task.

**KEYWORDS**

COVID-19, Predictive Analytics, Forecasting models, Machine Learning Method, Prediction, Epidemic, Pandemic

1. **INTRODUCTION**

A novel human corona virus was originated from China in December 2019, causing a severe potentially fatal respiratory syndrome (COVID-19). The symptoms of COVID-19 may or may not be visual in infected individual hence spread rate can be faster as individual himself not aware of infection [1]. World Health Organization (WHO) declared COVID-19 as pandemic on 3oth January 2020 due to its faster spread [2]. Despite of the continuous efforts, the virus has managed to spread in most of the territories in the world; WHO has announced COVID-19 as Pandemic [3]. Countries throughout the world working cooperatively and openly with one another and coming together as an united front in regards of efforts to bring this situation under control using available Information, Communication and Technologies (ICT). ICT need to be critically used to bring the situation under control and predictive analytics can be empowered using ICT services, tools and applications. ICT can empower epidemiological study to find the determinants, occurrence, and distribution of health and disease in a defined population in terms of COVID-19.

As study [4] shows that, 5% to 80% of people are tested positive for SARSCOV-2 may be asymptomatic. Predictive analysis using ICT plays an important role as some asymptomatic cases will become symptomatic over a period of time. Artificial Intelligence (AI) can be beneficial tool to fight against pandemic like COVID-19. AI models can be used for estimating and predicting spread rate, AI also used in the past pandemics like Zika-virus in 2015. Due to accurate and fast predictions spread rate can be minimized by taking necessary precautionary action before the time.

In nutshell, taking into consideration the current scenario a sad reality of the COVID-19 pandemic is that many people have been infected. As per the daily situation report of WHO as on April 09, 2020 the COVID-19 transmission scenario reports 1436198 confirmed cases with 85522 deaths globally. In the view of above-mentioned related issues, we should promote ecumenical and interfaith collaboration and peaceful coexistence during the COVID-19 pandemic. Ensuring that accurate information is shared with communities; and misinformation in addressed through ICT.

1. **Medical Perspectives**

The emergence of SARS-CoV-2 is confirmed from Wuhan’s Huanan Seafood market, China, but specific animal source still remains uncertain. There is uncertainty regarding origin of SARS-CoV-2 [5]. The situation with SARS-CoV-2 is developing faster with the numbers of infected cases and death is increasing exponentially. The unprecedented control measures taken have been effective in preventing spreading of SARS CoV-2. Still, there is continued rise in number of cases with infection of SARS CoV-2. Hence, it is essential to identify that the increase is due to infected cases before lockdown, due to community transmission; hospital acquired infection or spread within family. This should be determined experimentally, which may help in revealing the actual numbers of infected patients and asymptomatic carriers.

Many studies have confirmed transmission amongst human of SARS-CoV-2 [6, 7], but mechanism of transmission and pathogenesis in spreading in humans remains to be fully explored? During transmission from human to human, whether the pathogenicity of this virus is decreased with the increase in rate of transmission? If the transmission of this virus is declined, the outbreak may eventually end. Nevertheless, if there is continuous and effective transmission, SARS-CoV-2 will develop into an additional human coronavirus which is community acquired. It is difficult to recognize and take further actions in patient with undefined and mild symptoms. Studying a group of asymptomatic infected cases, and follow them for their clinical presentation, titters of antibody and viral loads, will help in understanding about the number of subjects have symptoms later, whether viral shedding is actually less robust and how frequently asymptomatic carriers can transmit virus further. A study reported that asymptomatic infection is high (15.8%) in children under 10 years

Globally there has been lot of progress in monitoring and control of disease spread. It is evident that, there are lot of uncertainties and questions regarding transmission mechanism, asymptomatic or subclinical patient’s virus shedding, origin of virus, virus pathogenesis, treatment, symptoms, etc. This highlights the need of integrative approach of predictive analytics and mathematical modelling with biological science, which may help government to take appropriate measures and method for future preparedness in fighting against this outbreak. In spite of rapid progression in research towards this outbreak, most of the studies are unable to suggest and guide effective measures to control this current situation. However, more high-quality research in biomedical science along with predictive analytics and mathematical modelling is warranted to manage public health crisis in short and long term.

1. **Predictive Analysis**

Predictive analytics is specialized branch of data analytics for making better predictions using past data and using analysis techniques which includes statistical and learning methods. Discovery of patterns in input data and anticipating what is likely to happen is the man objective of predictive analytics. Statistical analysis, predictive modelling and machine learning are three main pillars of predictive analytics. The main capabilities of predictive analytics are statistical analysis, predictive modelling, linear regression and logistic model. Selection of appropriate predictive model and algorithm decide how efficiently we can make the better insights and useful decisions. Use case like hospital interested in prediction of number of patients likely to be admitted in intensive care unit in next seven days and prediction of fraud transaction for online banking provider might require different predictive analytics model than for predicting defaulter applicant for loan provider and predicting number of COVID-19 infected patient in next 10 days. Selection of appropriate predictive model is based on what predictive question would you like to address and how optimization can be carried out using predictive algorithms.

**Predictive Analytics Algorithm**

Predictive analytics algorithms are either based on machine learning or deep learning. Machine learning algorithms are used when there is a need of classification or clustering for prediction, decision or analysis. These algorithms are more suitable for structured data and can be linear or nonlinear in nature. Deep learning algorithms are subset of machine learning algorithms and more useful when there is a need of identification or to recognize something. These algorithms are more useful to bigger data like audio, video and images where machine learning algorithms start underperforming. The predictive analytics is mainly driven by learning techniques and there are wide ranges of applications for disease prediction in healthcare community [21, 22]. Random forest algorithm is based on decision trees and used for both classification and regression purposes. This algorithm is more suitable for big data and uses bagging to avoid the errors. This model can address over fitting more effectively. Gradient boosted model is ensemble model of decision tress and used for classification. This model uses incremental model by building one tree at each time by correcting errors made by previously trained tree. In contrast, in random forest there is no relation amongst trees. K-means algorithm works on unlabelled data and places new incoming data into logical groups based on some common feature. Consider the COVID-19 example where clusters are formed of various patients based on some severity of infection. Kmean model is useful to put new incoming patient into appropriate cluster. This method is extremely useful in this growing pandemic of COVID-19 due to large number of cases. Prediction of mortality and spread rate plays very important role in pandemic disease like COVID-19, as based on this prediction precautionary measures can be taken by public, government and heath care systems. WE have used FBProphet [23] algorithm for training the model and predicting number of infected cases in next three months. The key features of this algorithm are it works more accurate for time series data and mainly used for prediction and capacity planning. The time series dataset is referred from Kaggle where the statistics for this COVID-19 pandemic is give in the form of features like State/province, country, latitude, longitude, Date, confirmed infected, Deaths and Recovered. Out of these eight fields in the dataset and another feature of Profet is it does not require splitting of dataset wherein for fitting it takes whole dataset for accurate results.

**Note: The collected data is till 31st March 2020**

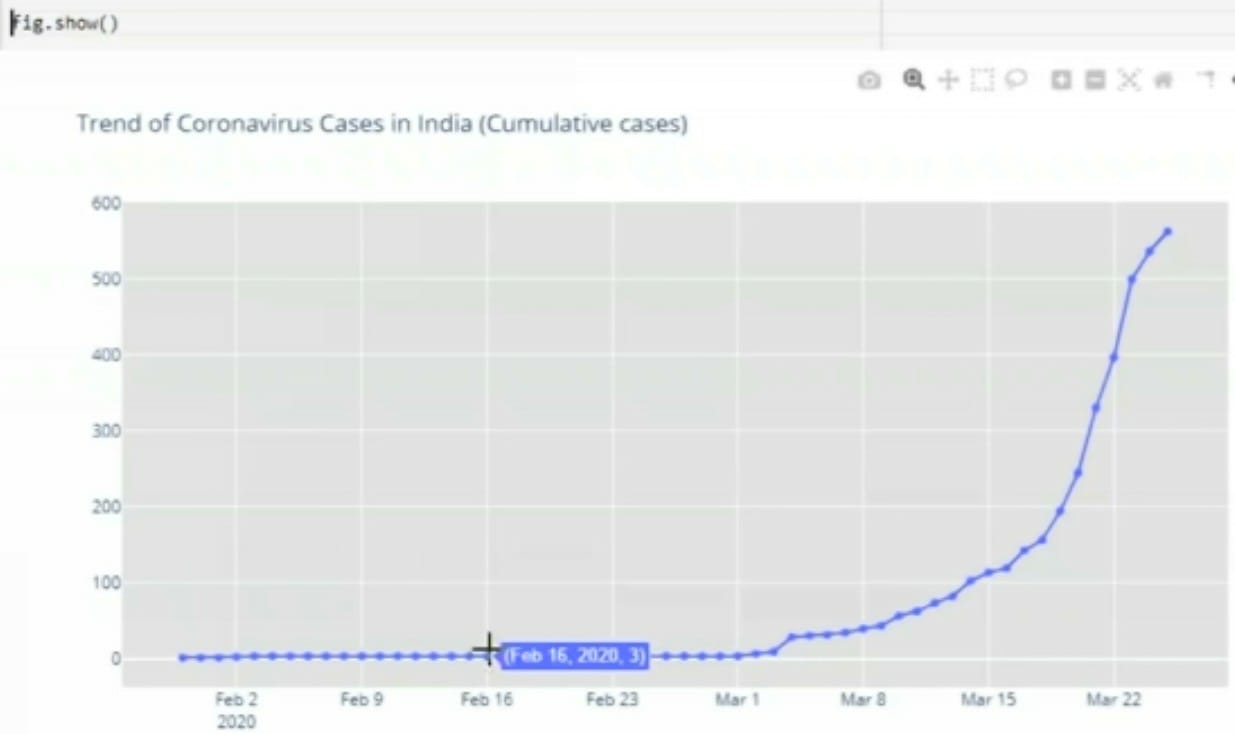
* Visualization of total confirmed cases of COVID-19 in India.



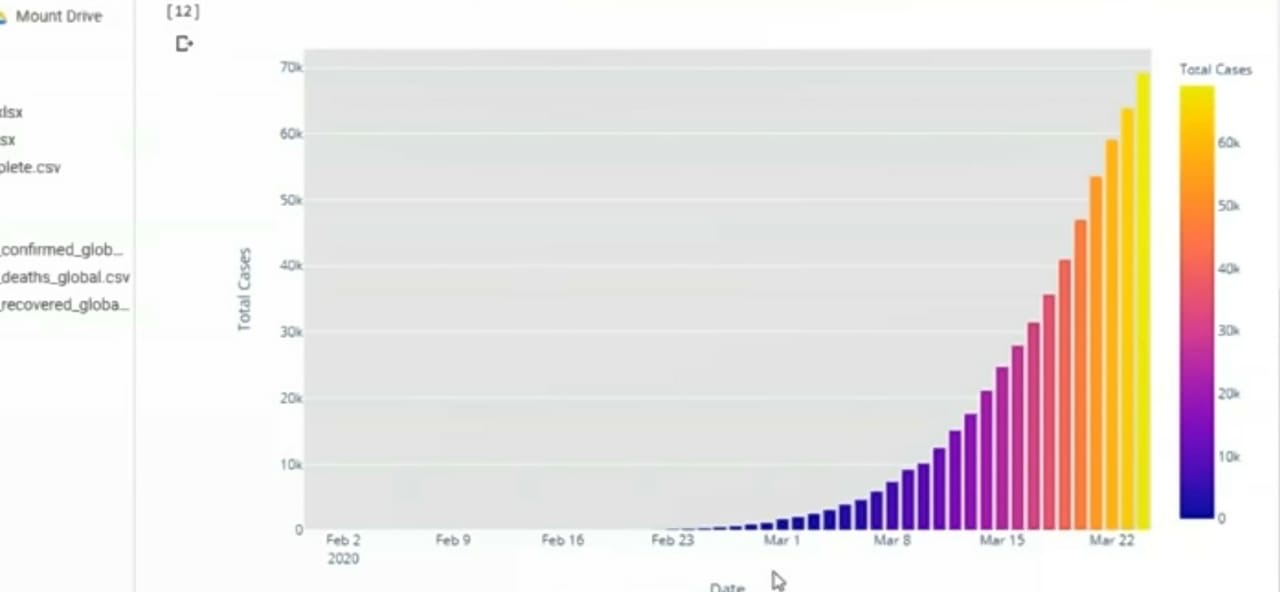
* State-Wise Visualization



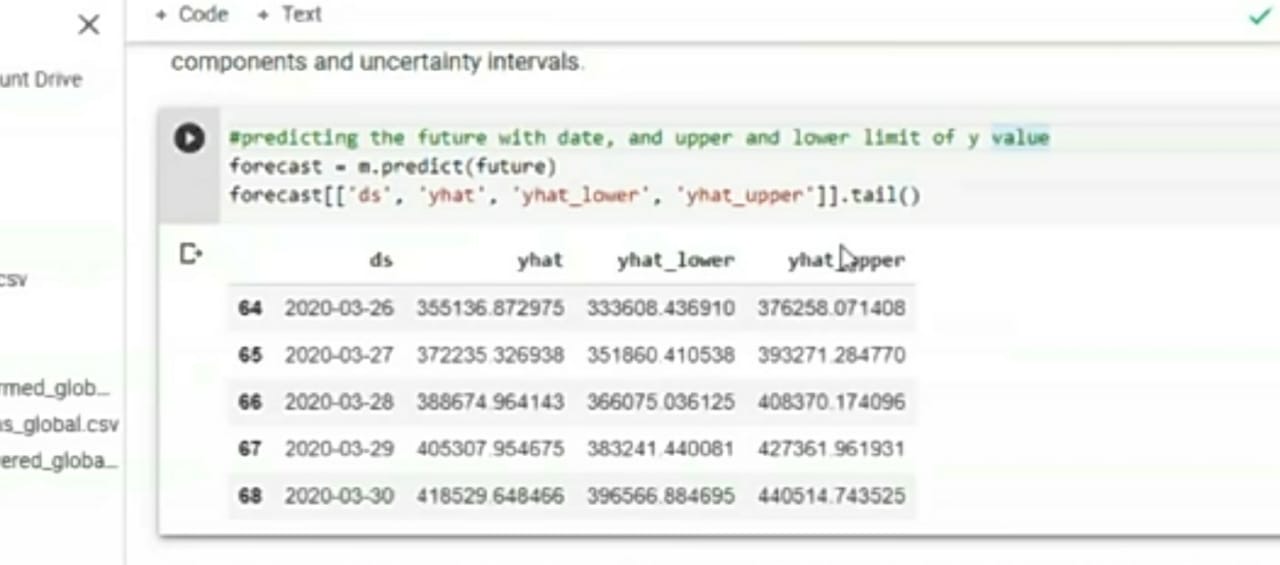
* Trend of COVID-19 in India



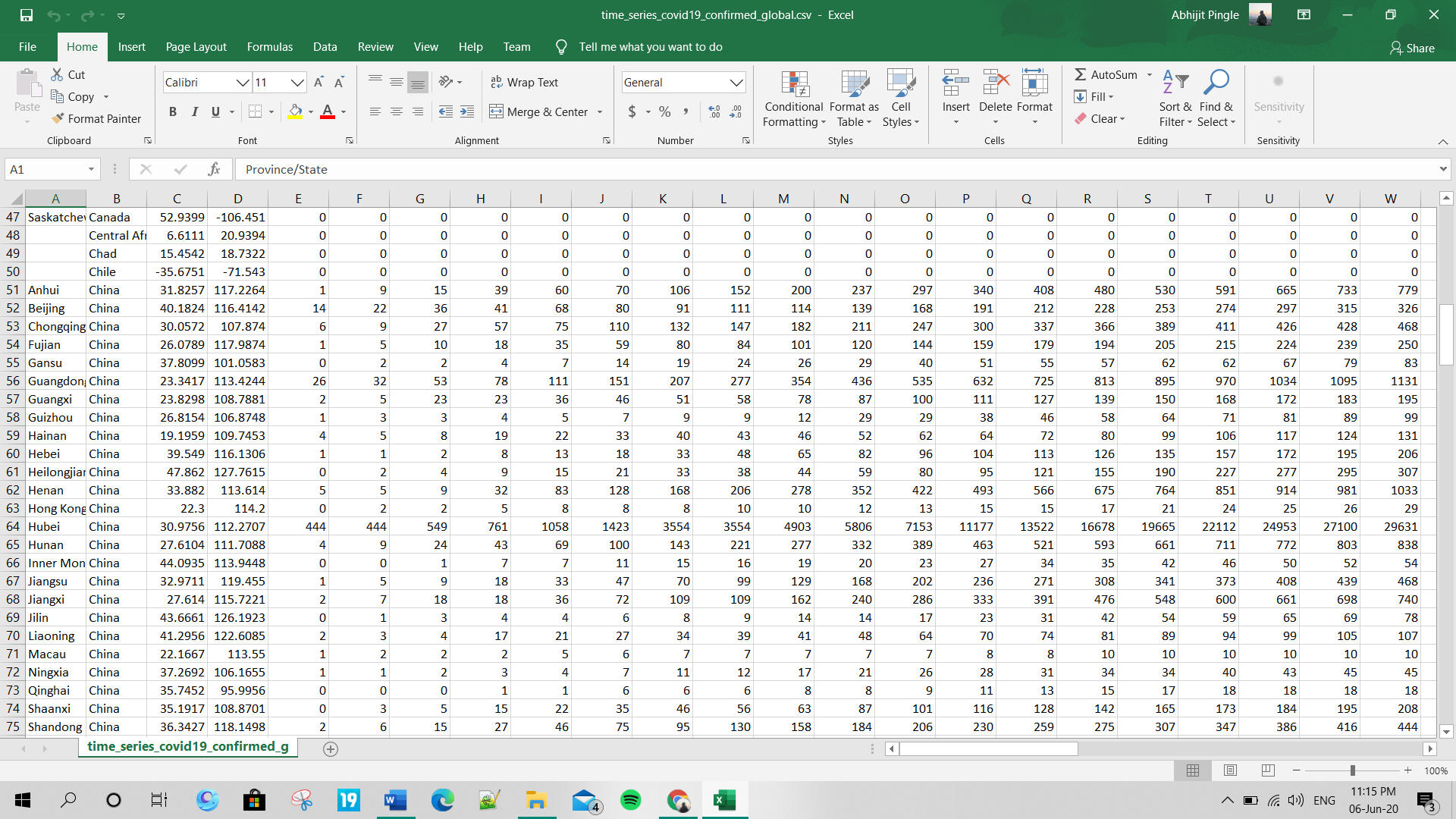
* Visualization of increment in total confirmed cases of COVID-19 in India (Exponential)



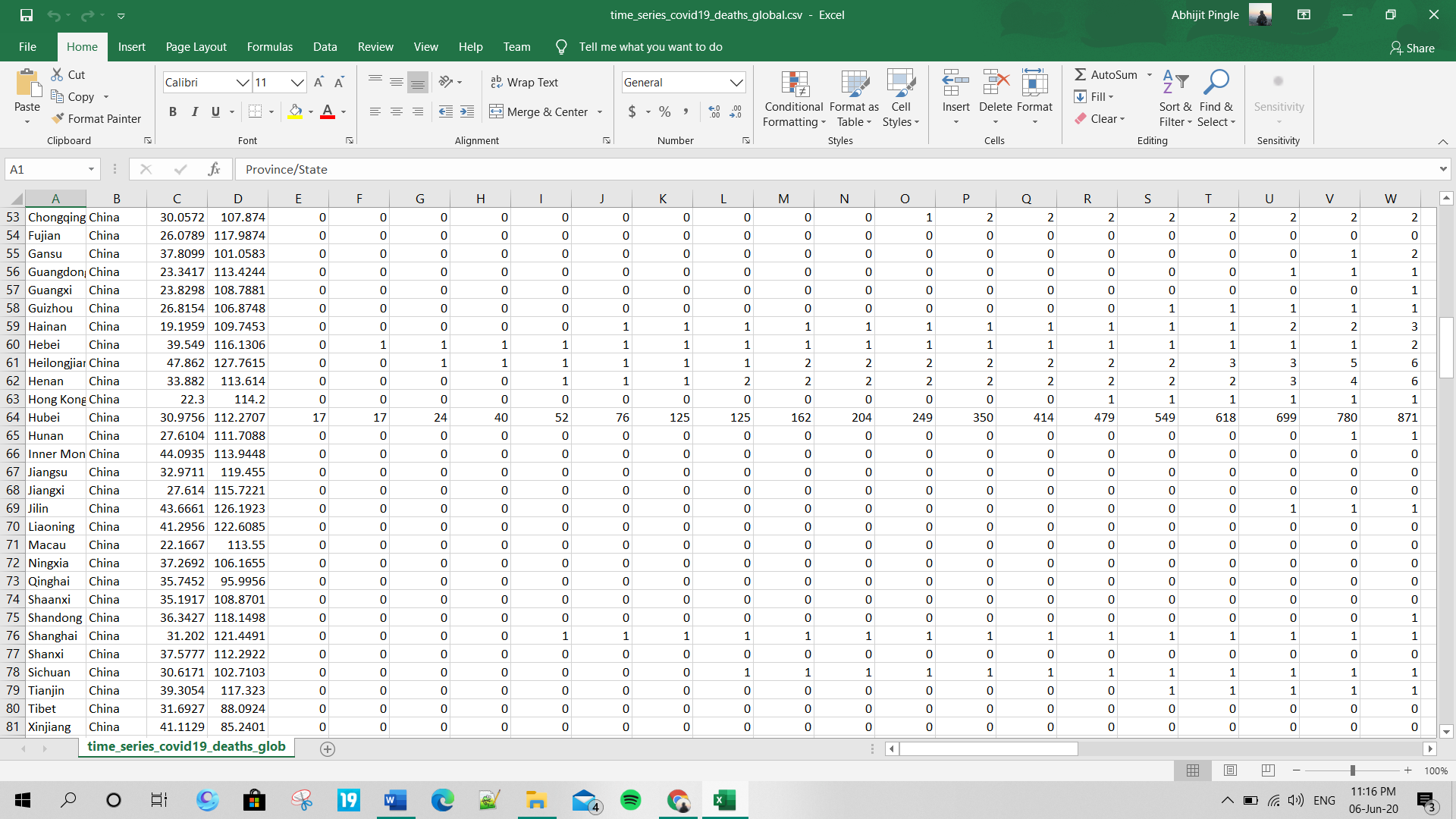
* Forecasting for a total case of COVID-19 a week ahead using open-source software Prophet. Here yhat\_upper is the highest case possibility and yhat\_lower is the lowest case possibility.



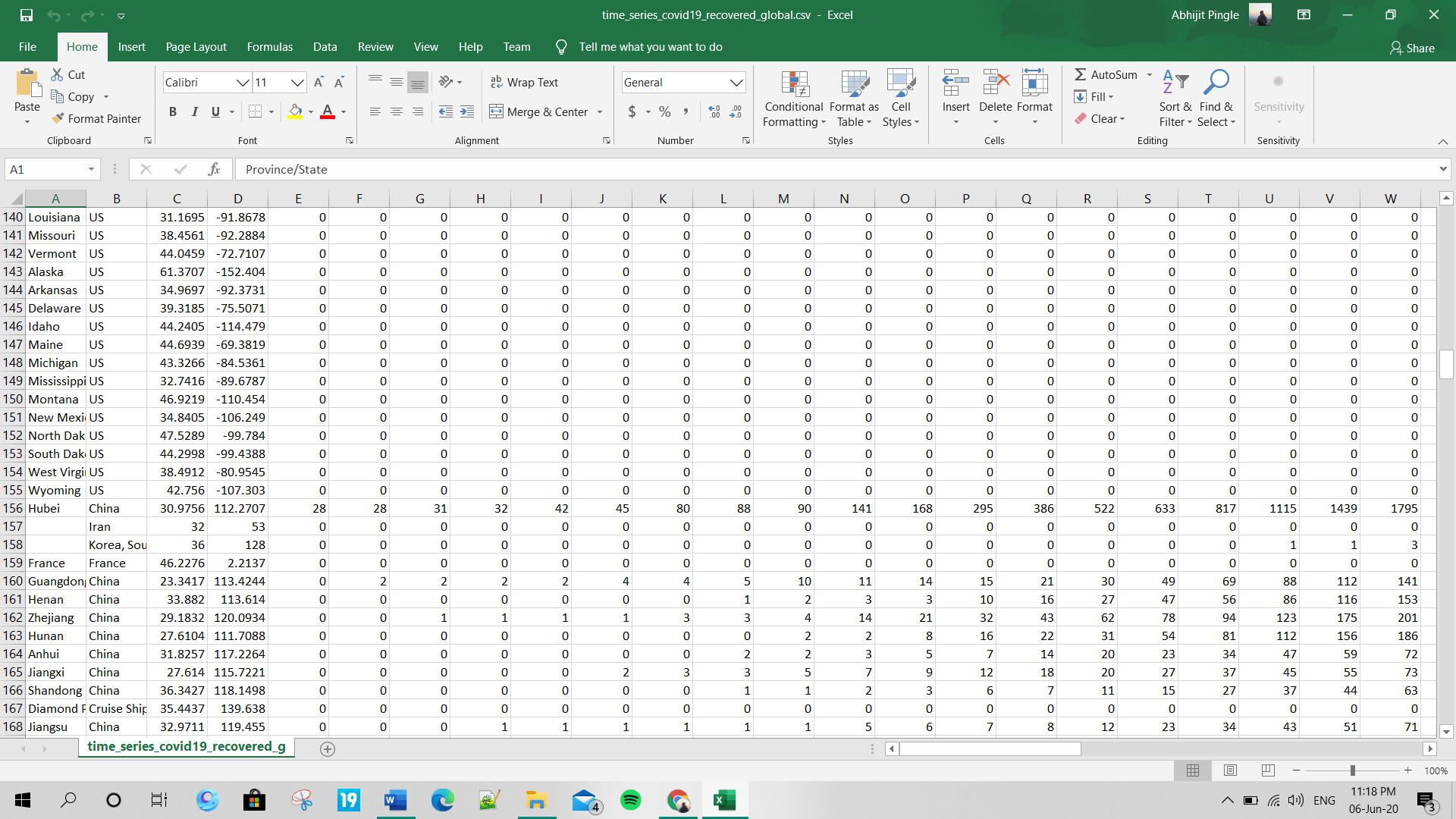
**Time\_Series – confirmed cases:**



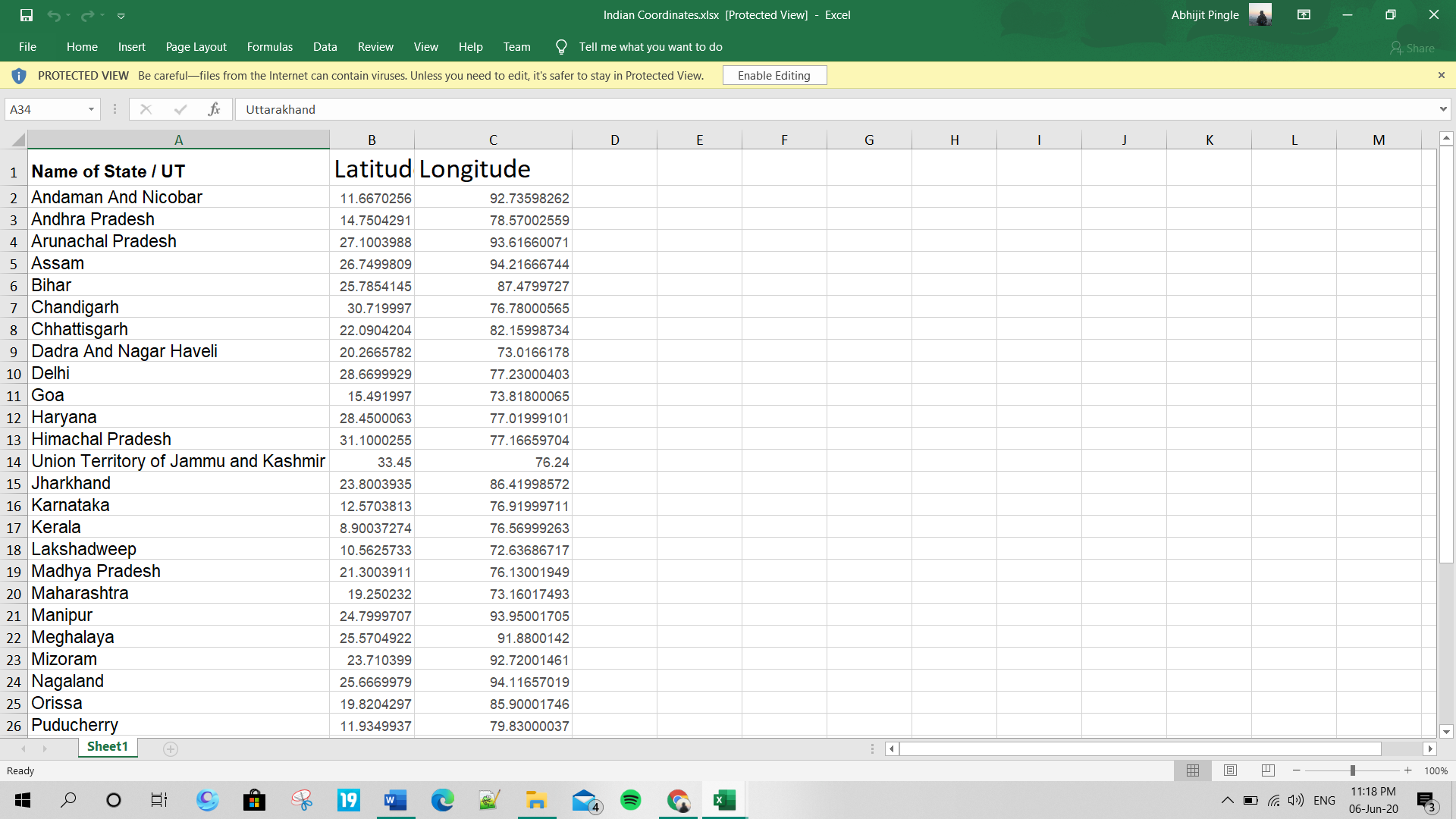
Time\_series – Deaths:



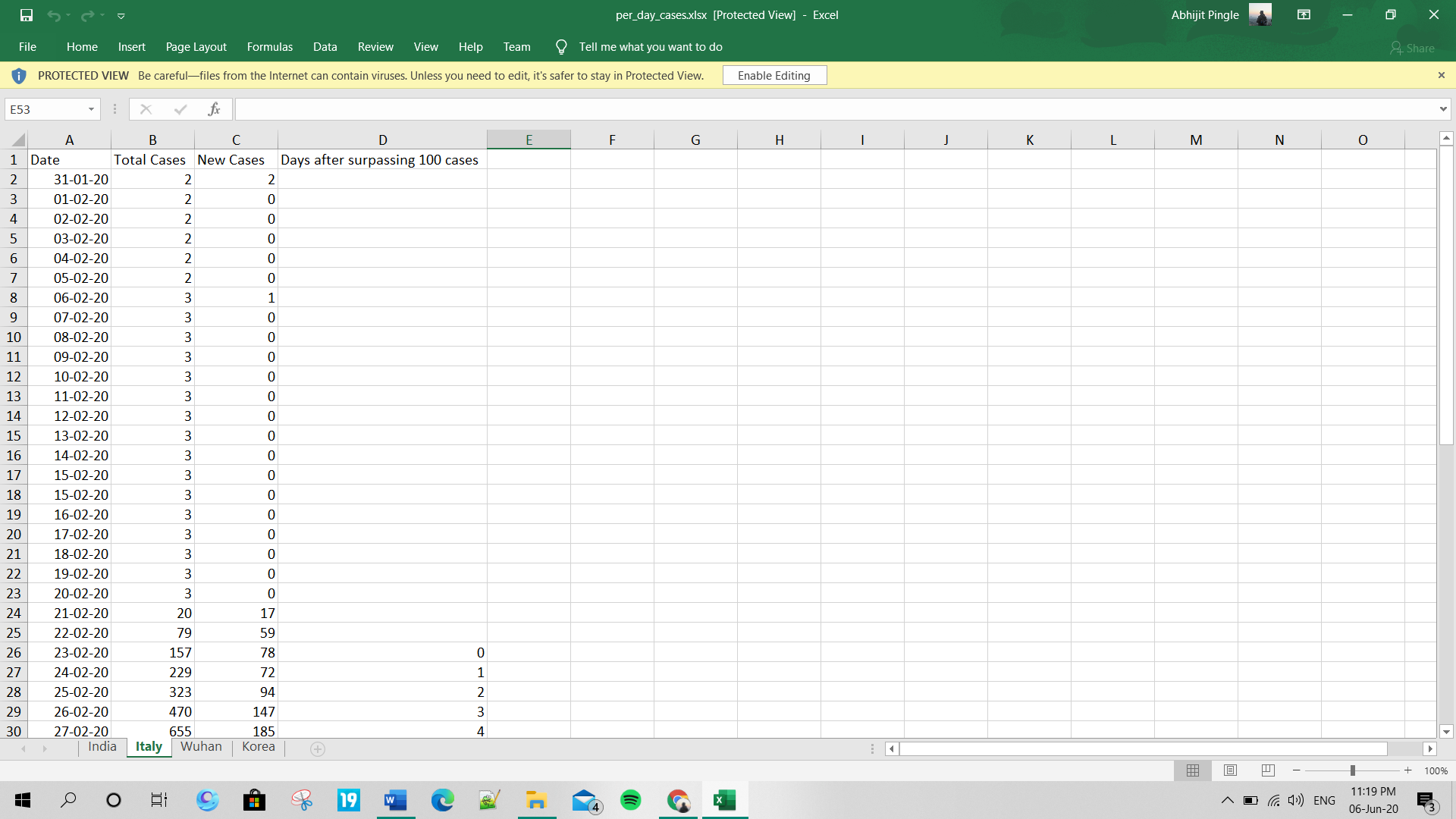
Time\_series- Recovered:



Indian coordinates:



Per day cases:



Code:

# 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')

#Learn how to read a .xls file by creating a dataframe using pandas

# Reading the datasets

df= pd.read\_excel('/content/Covid cases in India.xlsx')

df\_india = df.copy()

df

# 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")

#Learn how to play around with the dataframe and create a new attribute of 'Total Case'

#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)

#Learn how to highlight your dataframe

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

#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')

# Learn how to use folium to create a zoomable map

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

folium.CircleMarker([lat, lon], radius=value\*0.8, popup = ('<strong>State</strong>: ' + str(name).capitalize() + '<br>''<strong>Total Cases</strong>: ' + str(value) + '<br>'),color='red',fill\_color='red',fill\_opacity=0.3 ).add\_to(map)

#Learn how to use Seaborn for visualization

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)

#This cell's code is required when you are working with plotly on colab

import plotly

plotly.io.renderers.default = 'colab'

#Learn how to create interactive graphs using plotly

# import plotly.graph\_objects as go

# 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()

# New COVID-19 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='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

https://plot.ly/python/builtin-colorscales/

'''

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,

title='Confirmed Cases in Wuhan', color\_discrete\_sequence = px.colors.cyclical.IceFire)

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

fig.show()

#Learn how to create subplots using plotly

# 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.query('Country=="India"').groupby("Date")[['Confirmed', 'Deaths', 'Recovered']].sum().reset\_index()

df.groupby('Date').sum().head()

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 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()

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'])

m = Prophet(interval\_width=0.95)

m.fit(confirmed)

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

future.tail()

#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()

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

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