National College of Ireland

Worldwide differences of Covid-19 on cases and deaths using time series forecasting models

MSc Research Project
Data Analytics

Ankish Kumar Chandani
Student ID: x18189245

School of Computing
National College of Ireland

Supervisor: Prof. Christian Horn

National College of Ireland
Project Submission Sheet
School of Computing

National College of Ireland

Student Name:	Ankish Kumar Chandani
Student ID:	x18189245
Programme:	MSc Data Analytics
Year:	2019-20
Module:	MSc Research Project
Supervisor:	Prof. Christian Horn
Submission Due Date:	28/9/2020
Project Title:	Worldwide differences of Covid-19 on cases and deaths using time series forecasting models
Word Count:	1097
Page Count:	14

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.

ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

Signature:	
Date:	27th September 2020

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST:

Attach a completed copy of this sheet to each project (including multiple copies).	☐
Attach a Moodle submission receipt of the online project submission, to each project (including multiple copies).	☐
You must ensure that you retain a HARD COPY of the project, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer.	☐

Assignments that are submitted to the Programme Coordinator office must be placed into the assignment box located outside the office.

Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

Configuration Manual

Ankish Kumar Chandani
x18189245

1 Introduction

Configuration Manual extensively explains the environmental setup of the research project. It includes the hardware specifications, programming languages used and various packages and libraries installed.

Configuration Manual also discusses about different results evaluated from the models implemented in the research. Also, the exploratory data analysis, pre-processing and other related information have been discussed in the report. The reason behind including these information in this report is because these were not part of the research document however they are worth discussion due to their relevant information.

2 Environment Specifications

2.1 Hardware Specifications

The research has been performed on the local machine with following configurations:

Operating System: Microsoft Windows 10 Pro, 64 bit
System Processor: Intel(R) Core(TM) i7-8550U CPU @1.80 GHz
Installed Memory (RAM): 16GB
Storage Capacity: 512GB SSD (Solid State Drive)

2.2 Technical Specifications

Following programming language and tools have been used in the research:

Python: Python software version 3.7 has been used for data collection, cleaning, pre-processing and exploratory analysis, modelling and evaluation.

Anaconda3 Jupyter: Python 3.7 is by default supported by Anaconda3 and it is a platform which is used for coding. Jupyter, a collaboration tool has been used in the anaconda platform for writing the code because of its ability to combine code and share them with anyone. It is easy in data cleaning and pre-processing.

Microsoft Excel 2016: Dataset was downloaded in csv file and MS Excel was used to open this file.

Following libraries and packages have been used in this research:

• import pandas as pd – used to calculate data and manipulations of the data
• import numpy as np – library for performing mathematical functions
• from datetime import date, datetime, timedelta - used for changing date time format
• import plotly.express as px - used for complex charts
• import plotly.graph_objs as go - used for graph objects
• import matplotlib.pyplot as plt - used in plotting visualization
• import warnings – library for implementing alerts
• import matplotlib.pyplot as plt – library used for plotting graphs
• from sklearn.preprocessing import PolynomialFeatures, StandardScaler - library used for implementing polynomial regression and data transformation respectively
• from sklearn.linear_model import LinearRegression - used for modelling linear regression
• import statsmodels.api as sm – library used to implement statistical models
• from sklearn.metrics import mean_absolute_error,mean_squared_error – used for calculating MAE and RMSE
• from fbprophet import Prophet - library used for applying prophet model
• import math import sqrt – library used for performing square root
• import seaborn as sns – library used for data visualization
• import matplotlib as mpl - used for scaling graph data
• from statsmodels.tsa.stattools import adfuller – library used to conduct adfuller test
• from scipy import stats – library used to apply statistical functions
• from statsmodels.tsa.api import Holt - used for implementing Holt's linear model
• from pmdarima.arima import auto_arima – library used in auto arima score calculation and implementing AR and ARIMA model

3 Dataset Source

Daily time series data from 31st Dec 2019 to 25th July 2020 for all the countries in the world which are impacted by novel Covid-19 has been collected.

Data is downloaded from Our World in Data website (`https://ourworldindata.org/coronavirus-source-data`) which collects all the data from ECDC (European Centre for Disease Prevention and Control) and WHO (World Health Organization) and then combines them into a proper structured format.

As the website is continuously updating data every day, dataset was downloaded till the date 25th July. In order to work on the real time data, below mentioned website can be accessed publicly. Dataset was downloaded into the local system, renamed for more understanding and then loaded into the data frame using python jupyter.
To reproduce the dataset loading into the data frame, user needs to change the dataset path as per the data located where it has been downloaded in order to load the dataset into data frame.
Dataset contains 32584 rows x 34 columns and below is the snapshot of 1st 4 rows with 1st few variables.

iso_code	continent	location	date	total_cases	new_cases	total_deaths	new_deaths	total_cases_per_million
AFG	Asia	Afghanistan	2019-12-31	0.0	0.0	0.0	0.0	0.0
AFG	Asia	Afghanistan	2020-01-01	0.0	0.0	0.0	0.0	0.0
AFG	Asia	Afghanistan	2020-01-02	0.0	0.0	0.0	0.0	0.0
AFG	Asia	Afghanistan	2020-01-03	0.0	0.0	0.0	0.0	0.0

Figure 1: Sample Dataset

4 Data Exploratory Analysis and Pre-Processing

Data was loaded into a data frame and variables which were not important for the analysis and which had more than 50% missing values were dropped from the data frame as shown in below figure 2:

```
# Dropping columns who have almost missing values 50% or more than that or not important for the analysis
df=df.drop(['total_tests_per_thousand','new_tests_per_thousand','new_tests_smoothed','new_tests_smoothed_per_thousand',
'tests_units','extreme_poverty','handwashing_facilities','population_density','median_age','aged_65_older','aged_70_older',
'gdp_per_capita','female_smokers','diabetes_prevalence','cardiovasc_death_rate','male_smokers','hospital_beds_per_thousand',
'life_expectancy'], axis = 1)
```

Figure 2: Variables Dropped

To make the visualization, tables and graphs more clear and easily understandable, variables were renamed as shown below:

```
# Renaming the column names
df=df.rename(columns={'iso_code':'ISO Code','continent':'Continent','location':'Country','date':'Date',
                      'total_cases':'Total Confirmed Cases','total_deaths':'Total Deaths','new_cases':'New Cases',
                      'new_deaths':'New Deaths','total_cases_per_million':'Total Cases/million',
                     'new_cases_per_million':'New Cases/million','total_deaths_per_million':'Total Deaths/million',
                      'new_deaths_per_million':'New Deaths/million','new_tests':'New Tests','total_tests':'Total Tests',
                      'population':'Population','stringency_index':'Stringency Index'})
```

Figure 3: Variables Renamed

Correlation between some important variables are shown as below:

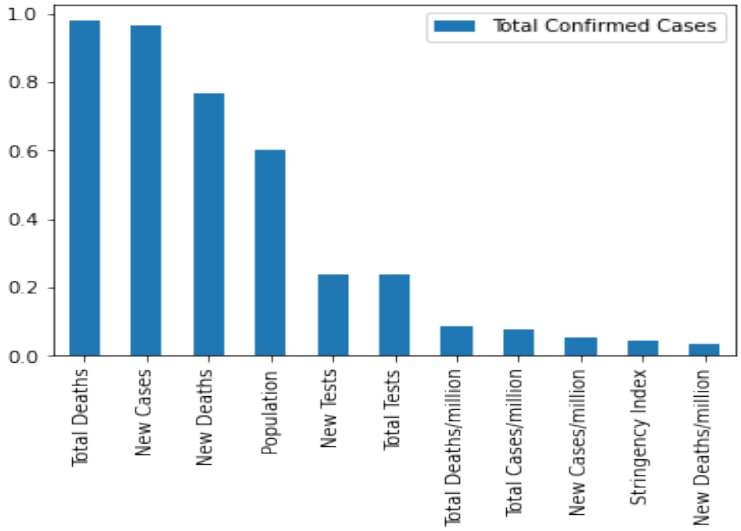

Figure 4: Total Confirmed Cases correlation

Above figure shows that total confirmed cases by Covid-19 was highly correlated with total deaths, new cases, new deaths and population.

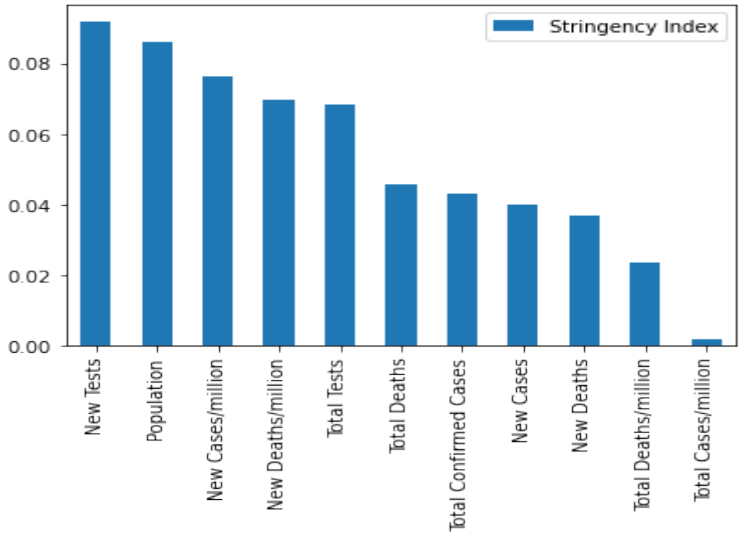

Figure 5: Stringency Index correlation

Above figure shows that stringency index during Covid-19 was highly correlated with new deaths, population, new cases/million, new deaths/million and total tests.

As shown in below figure 6, new dataframes were created out of the original one so that the analysis could be done on each continents separately as well as country specific

```
# Creating a new dataframe containing only data of location World
# It contains cumulative data of all countries together except for Spain's count for 25th July
df_world = df[df.Country == 'World']
# Creating a new dataframe containing only data of all countries
df_countries = df[df.Country != 'World']
# Creating a new dataframe containing only data of continent Europe
df_europe = df[df.Continent == 'Europe']
# Creating a new dataframe containing only data of continent Asia
df_asia = df[df.Continent == 'Asia']
# Creating a new dataframe containing only data of continent North America
df_na = df[df.Continent == 'North America']
# Creating a new dataframe containing only data of continent South America
df_sa = df[df.Continent == 'South America']
```

Figure 6: New Dataframes created

Below Europe continent map shows the number of Covid-19 confirmed cases in each country and we can see that Russia has the highest number of cases reported followed by Italy and least number of cases can be seen the countries with dark blue color.

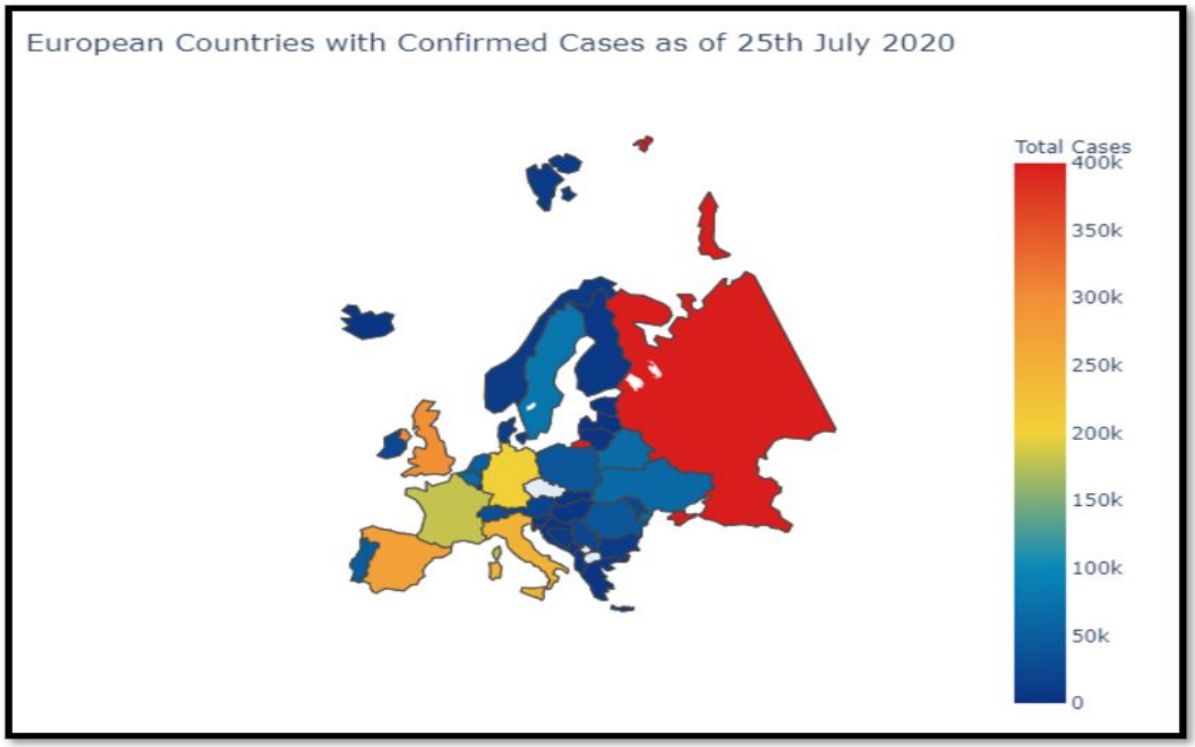

Figure 7: Confirmed Cases in Europe

Below Asia continent map shows the number of Covid-19 confirmed cases in each country and we can see that India has the highest number of cases reported.

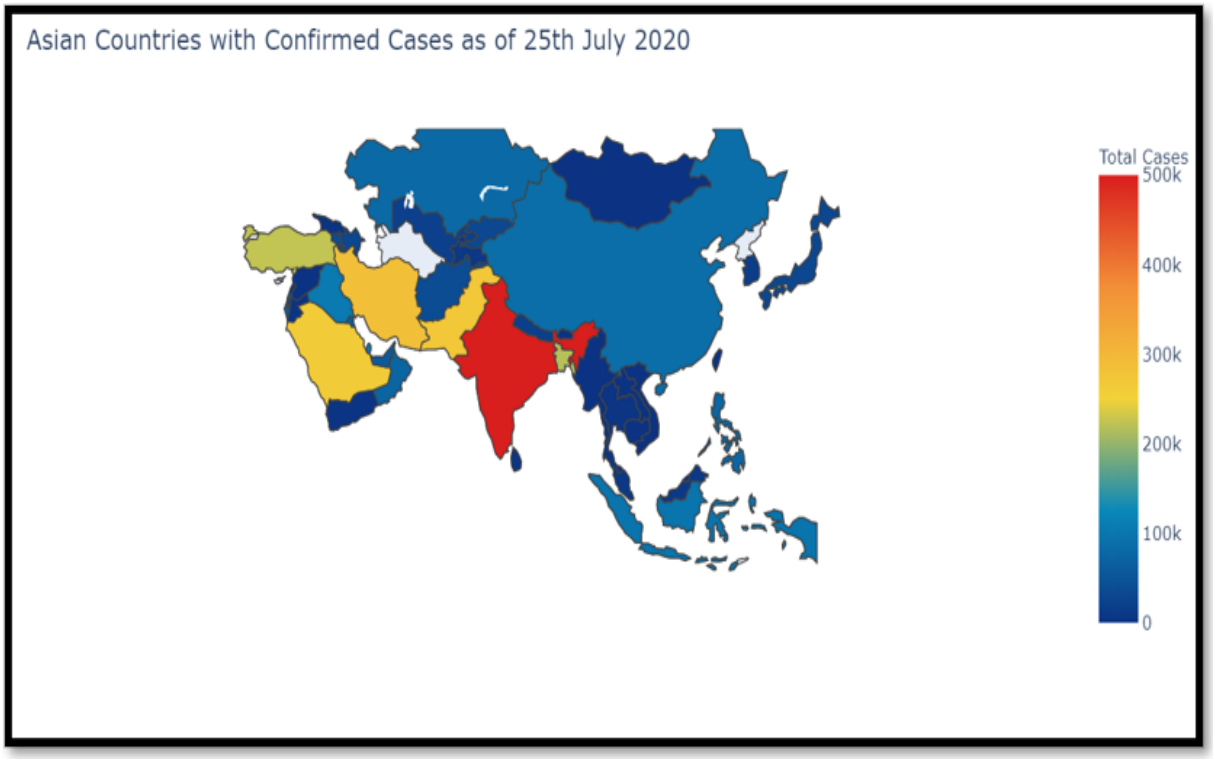

Figure 8: Confirmed Cases in Asia

Figure 9 shows the Covid-19 confirmed cases and deaths weekwise respectively for Ireland.

Figure 9: Ireland Analysis

Figure 10 as shown below provides the analysis of Covid-19 confirmed and total deaths worldwide as of 25th July 2020.

```
Total number of Confirmed Cases around the World:  15762053.0
Total number of Confirmed Deaths around the World:  639273.0
Approximate number of Confirmed Cases per Day around the World:  75779.0
Approximate number of Confirmed Deaths per Day around the World:  3073.0
Approximate number of Confirmed Cases per hour around the World:  3157.0
Approximate number of Confirmed Deaths per hour around the World:  128.0
Number of Confirmed Cases in last 24 hours:  281742.0
Number of Death Cases in last 24 hours:  6147.0
```

Figure 10: Worldwide Covid-19 analysis

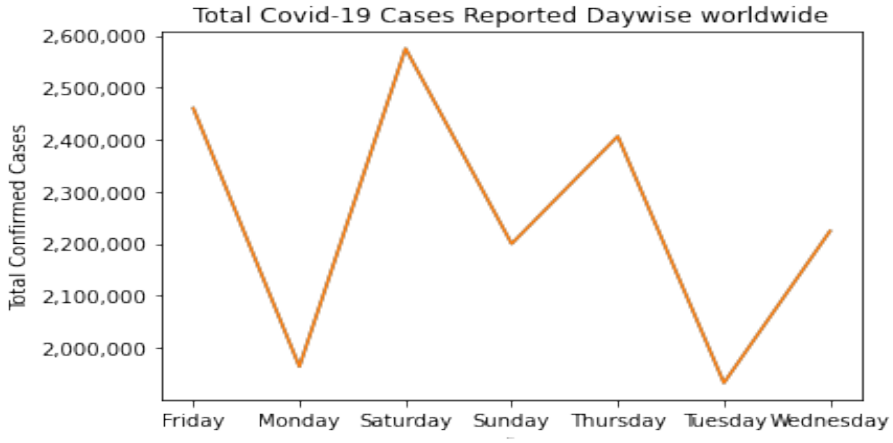

Figure 11: Worldwide reported cases by day

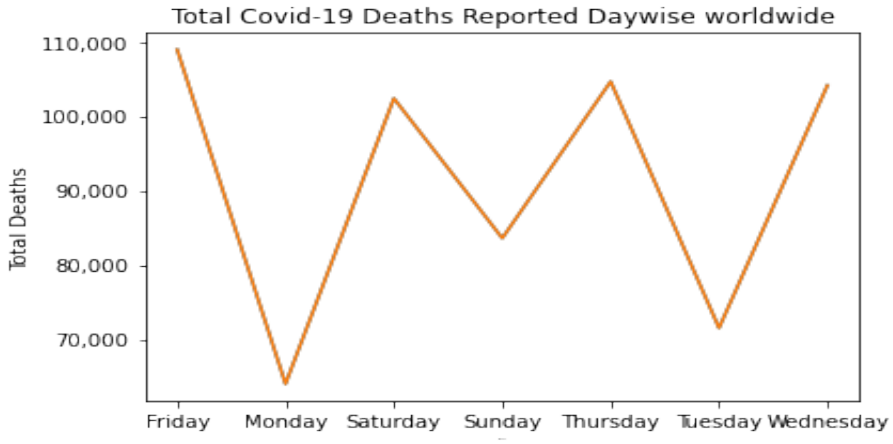

Figure 12: Worldwide reported deaths by day

As shown in figure 11 and 12, most of the cases have been reported on Saturday and deaths on Friday.
Figure 13 and 14 shows the trend of Covid-19 confirmed cases and deaths respectively for top 20 countries.

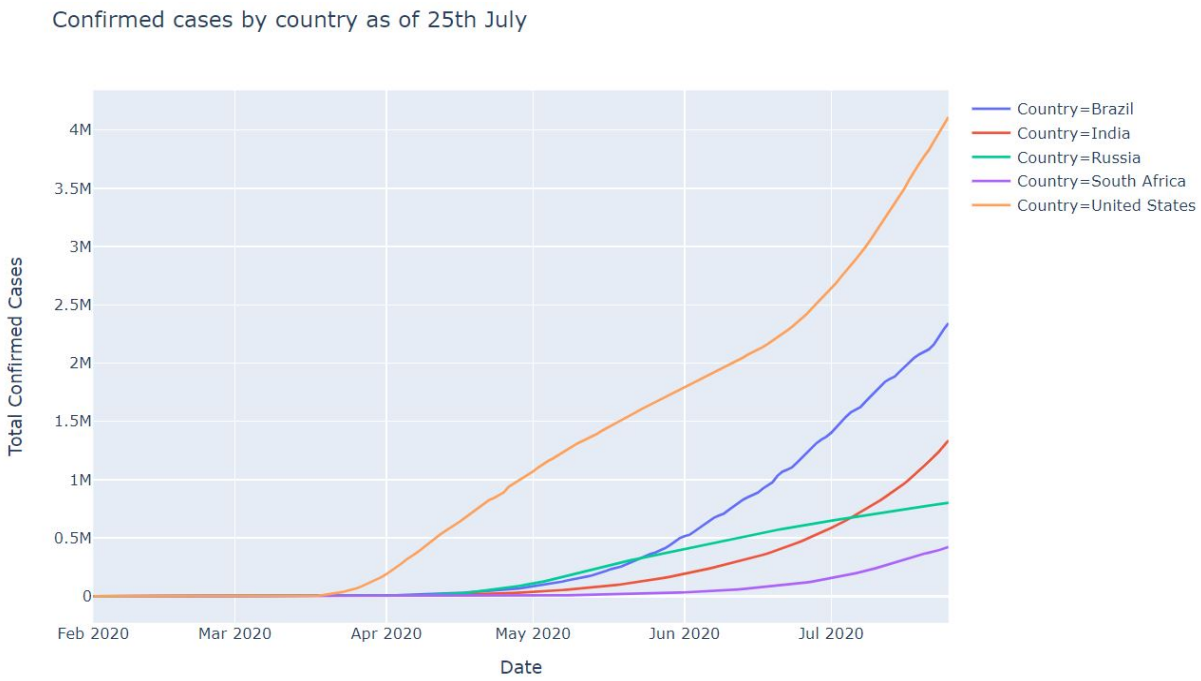

Figure 13: Trend of 5 countries with highest Covid-19 confirmed cases

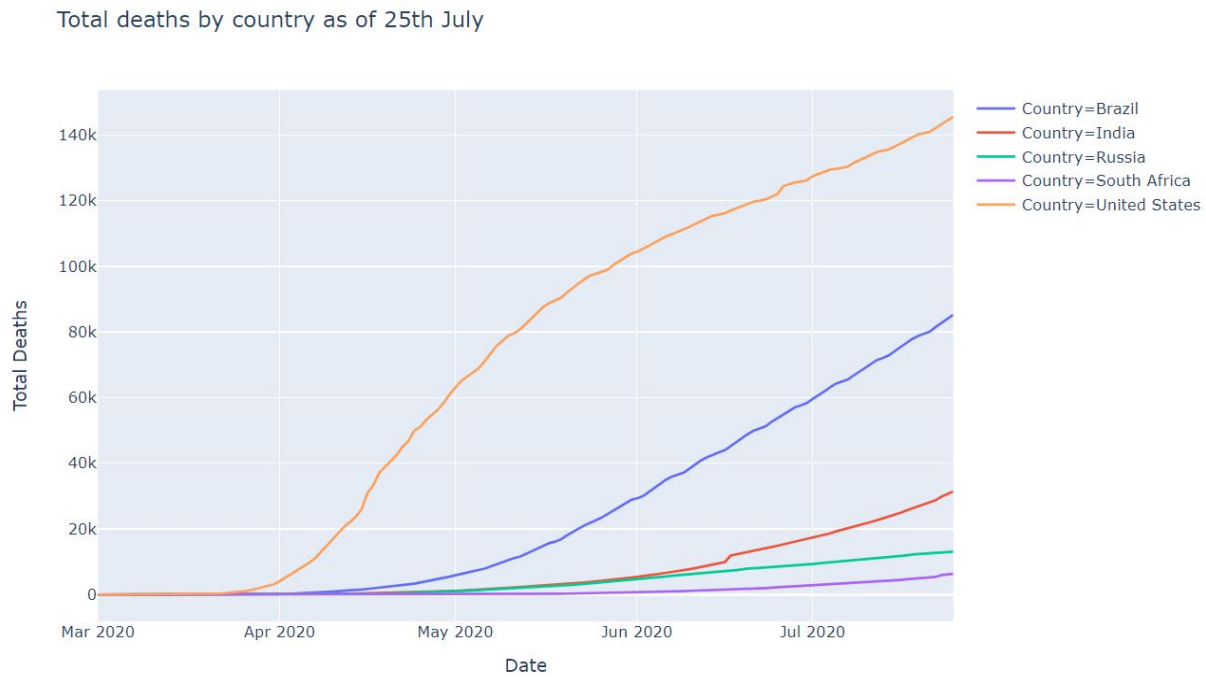

Figure 14: Trend of 5 countries with highest Covid-19 total deaths

Figure 15 shows the Covid-19 confirmed deaths per million population for top 5 countries.

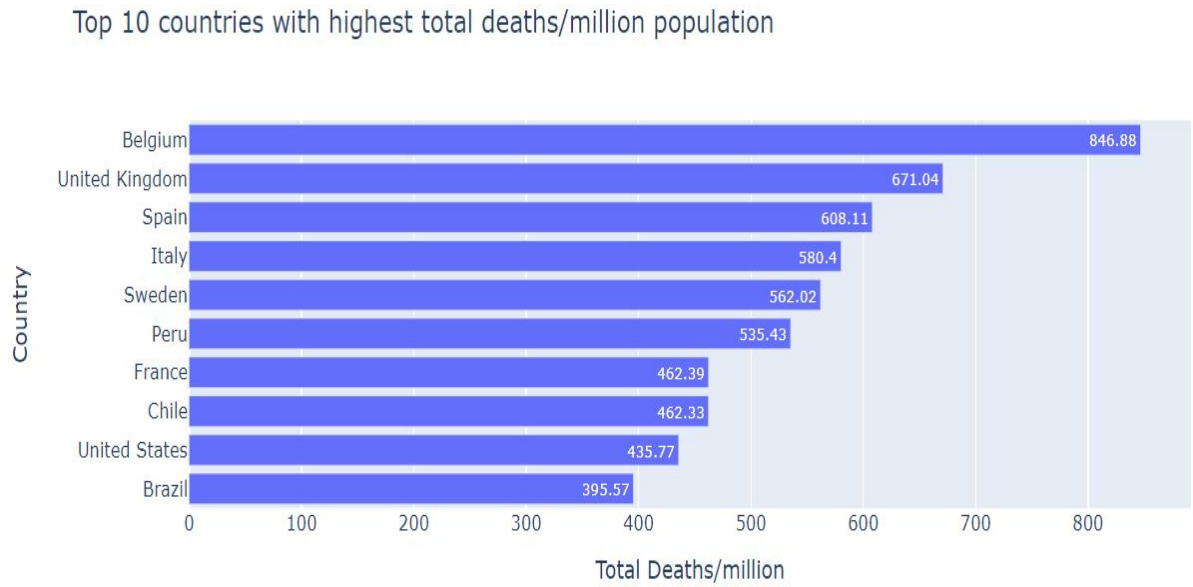

Figure 15: 20 Countries with the highest Covid-19 deaths per million population

Figure 16 shows the worldwide Covid-19 confirmed cases vs Moving average for 10 days.

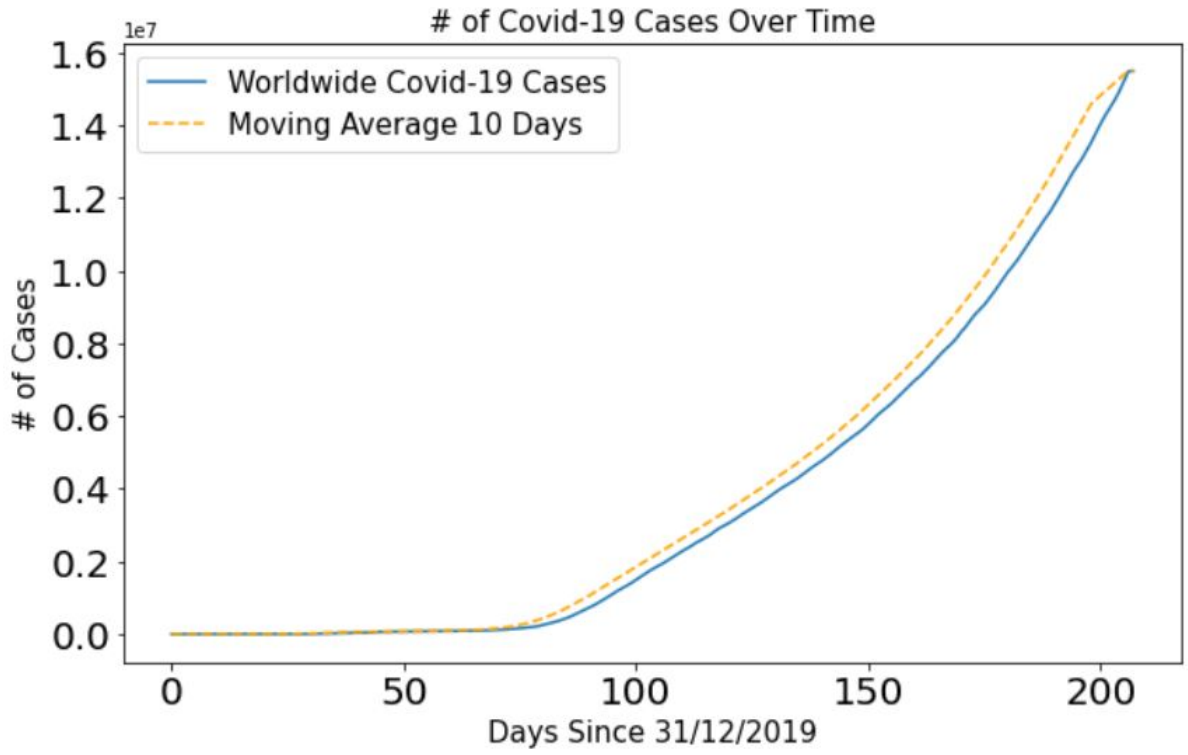

Figure 16: Worldwide Covid-19 confirmed cases vs Moving average for 10 days

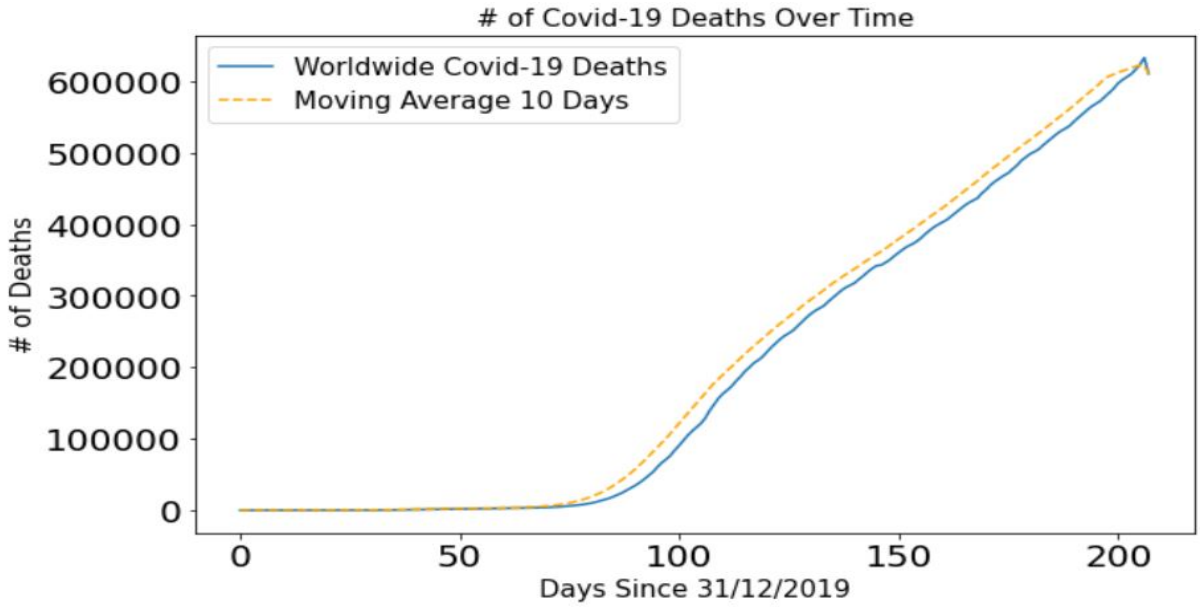

Figure 17: Worldwide Covid-19 total deaths vs Moving average for 10 days

Figure 17 shows the worldwide Covid-19 total deaths vs Moving average for 10 days.

5 Models Implementation and Evaluation

Confirmed Cases Prediction and Forecasting

5.1 Polynomial Regression Model

```
# Creating a new dataframe for model prediction and forecasting
datewise=df_countries.groupby(["Date"]).agg({"Total Confirmed Cases":'sum',"New Cases":'sum',"Total Deaths":'sum'})
datewise["Days Since"]=datewise.index-datewise.index.min()
# Converting the Day Since column datatype
datewise["Days Since"]=datewise.index-datewise.index[0]
datewise["Days Since"]=datewise["Days Since"].dt.days
# Splitting the dataset
train_ml=datewise.iloc[:int(datewise.shape[0]*0.95)]
test_ml=datewise.iloc[int(datewise.shape[0]*0.95):]
model_scores=[]
# Applying 4th degree of polynomial Regression
poly = PolynomialFeatures(degree = 4)
# Transforming the training and validation dataset
train_poly=poly.fit_transform(np.array(train_ml["Days Since"]).reshape(-1,1))
test_poly=poly.fit_transform(np.array(test_ml["Days Since"]).reshape(-1,1))
y=train_ml["Total Confirmed Cases"]
# Transforming and fitting the data
linreg=LinearRegression(normalize=True)
linreg.fit(train_poly,y)
# Predicting the model
prediction_poly=linreg.predict(test_poly)
# Forecasting the confirmed cases
new_date=[]
new_prediction_poly=[]
for i in range(1,60):
    new_date.append(datewise.index[-1]+timedelta(days=i))
    new_date_poly=poly.fit_transform(np.array(datewise["Days Since"].max()+i).reshape(-1,1))
    new_prediction_poly.append(linreg.predict(new_date_poly)[0])
```

Figure 18: Polynomial Regression Model

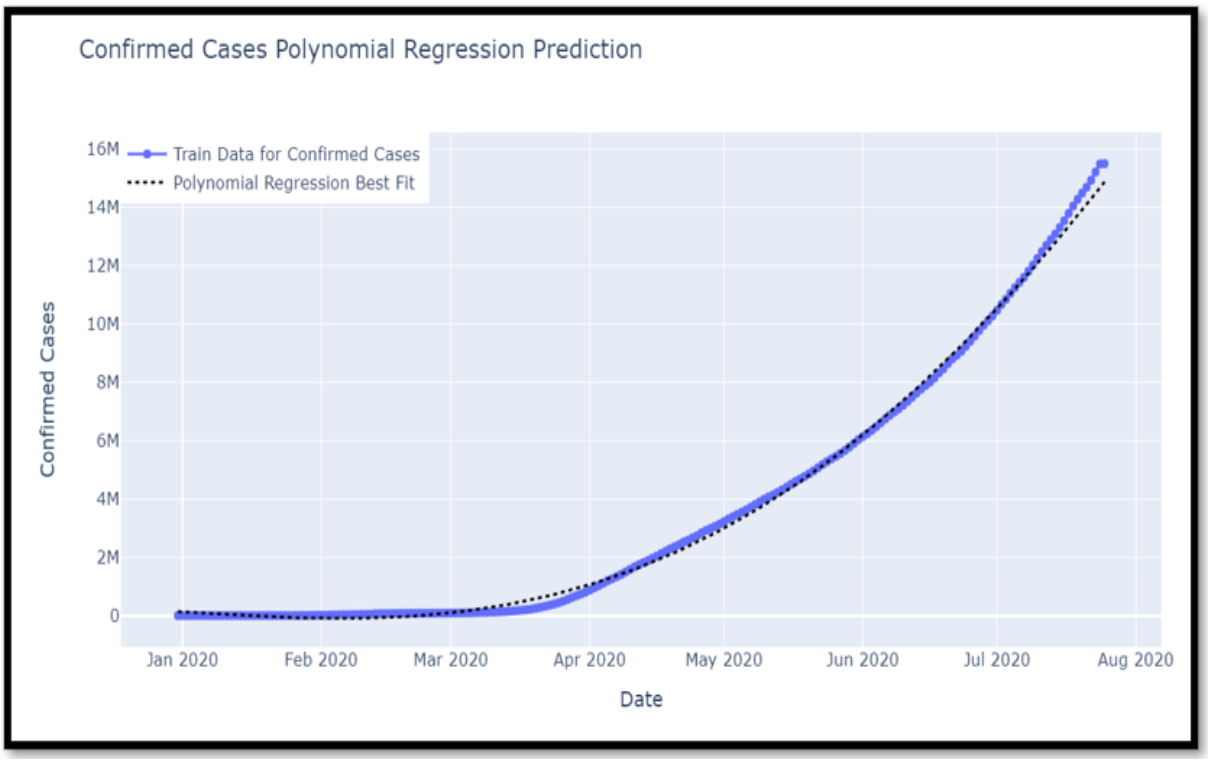

Figure 19: Polynomial Regression Model Prediction

5.2 Holt's Linear Model

```
# Dataset Splitting
model_train=datewise.iloc[:int(datewise.shape[0]*0.95)]
model_test=datewise.iloc[int(datewise.shape[0]*0.95):]
y_pred=model_test.copy()
# Fitting the model on the training set
holt=Holt(np.asarray(model_train["Total Confirmed Cases"])).fit(smoothing_level=0.2, smoothing_slope=1.8,optimized=False)
# Predicting the model on the test set
pred_holt=holt.forecast(len(model_test))
# Forecasting the confirmed cases
holt_new_date=[]
holt_new_prediction=[]
for i in range(1,60):
    holt_new_date.append(datewise.index[-1]+timedelta(days=i))
    holt_new_prediction.append(holt.forecast((len(model_test)+i))[-1])

model_predictions["Holt's Linear Model Prediction"]=holt_new_prediction
```

Figure 20: Holt's Linear Model

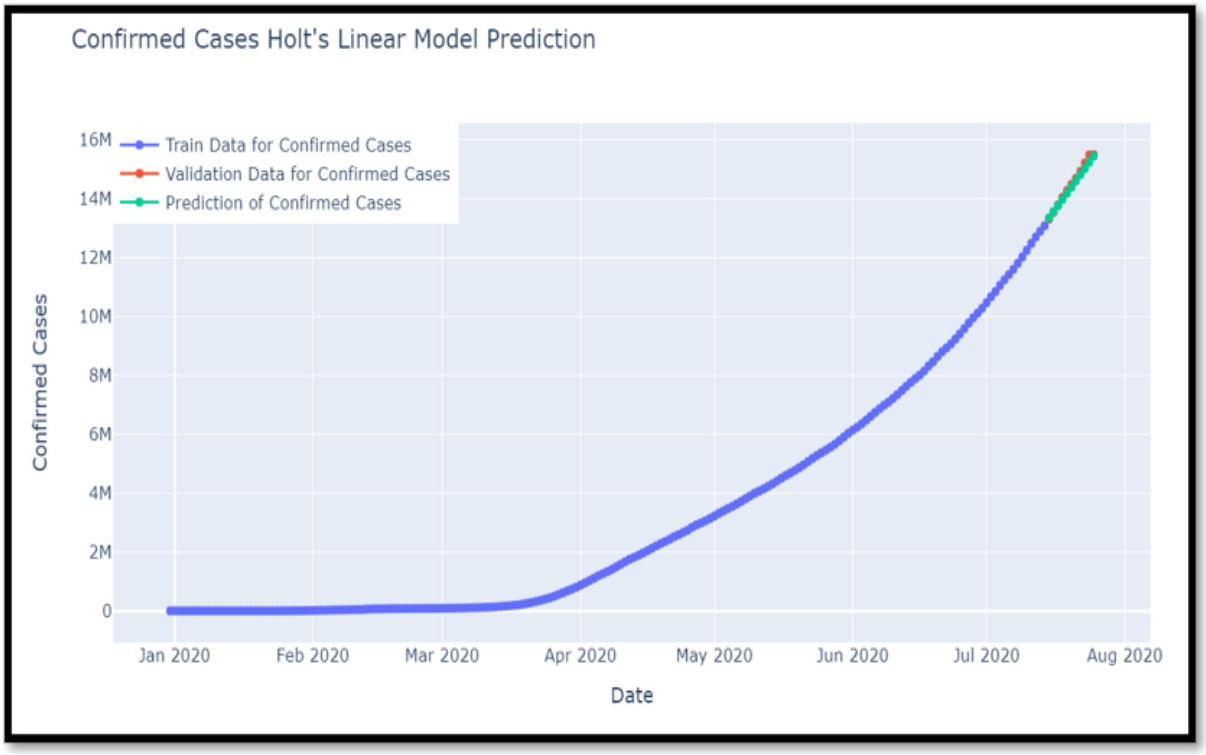

Figure 21: Holt's Linear Model Prediction

Predictions by AR Model, ARIMA Model and PROPHET Model has been already discussed in the research report. Although only 15 day forecasting is shown in the research report but in this document, 39 day forecasting will be presented.

Worlwide Covid-19 Confirmed Cases forecasting:

Date	Polynomial Regression Forecasting	Holt's Linear Model Forecasting	AR Model Forecasting	ARIMA Model Forecasting	Prophet Forecasting
15-08-2020	19.26M	19.86M	20.40M	21.37M	19.90M
16-08-2020	19.48M	20.07M	20.64M	21.67M	20.12M
17-08-2020	19.70M	20.28M	20.89M	21.96M	20.31M
18-08-2020	19.93M	20.49M	21.14M	22.26M	20.55M
19-08-2020	20.15M	20.70M	21.39M	22.58M	20.78M
20-08-2020	20.38M	20.91M	21.64M	22.92M	21.01M
21-08-2020	20.61M	21.12M	21.89M	23.26M	21.21M
22-08-2020	20.84M	21.33M	22.14M	23.59M	21.43M
23-08-2020	21.07M	21.54M	22.40M	23.91M	21.63M
24-08-2020	21.30M	21.75M	22.65M	24.23M	21.88M
25-08-2020	21.53M	21.96M	22.91M	24.55M	22.12M
26-08-2020	21.76M	22.17M	23.17M	24.89M	22.35M
27-08-2020	22.00M	22.38M	23.43M	25.25M	22.58M
28-08-2020	22.23M	22.59M	23.69M	25.61M	22.80M
29-08-2020	22.47M	22.80M	23.95M	25.96M	23.03M
30-08-2020	22.71M	23.01M	24.21M	26.30M	23.23M
31-08-2020	22.95M	23.22M	24.47M	26.64M	23.50M

01-09-2020	23.19M	23.43M	24.74M	26.98M	23.69M
02-09-2020	23.43M	23.64M	25.00M	27.35M	23.95M
03-09-2020	23.67M	23.85M	25.27M	27.72M	24.17M
04-09-2020	23.91M	24.06M	25.54M	28.10M	24.51M
05-09-2020	24.15M	24.27M	25.81M	28.48M	24.69M
06-09-2020	24.40M	24.48M	26.08M	28.84M	24.89M
07-09-2020	24.65M	24.69M	26.35M	29.20M	25.12M
08-09-2020	24.89M	24.90M	26.62M	29.57M	25.34M
09-09-2020	25.14M	25.11M	26.89M	29.95M	25.68M
10-09-2020	25.39M	25.32M	27.17M	30.35M	25.90M
11-09-2020	25.64M	25.53M	27.44M	30.75M	26.13M
12-09-2020	25.89M	25.74M	27.72M	31.15M	26.37M
13-09-2020	26.14M	25.95M	27.99M	31.53M	26.61M
14-09-2020	26.39M	26.16M	28.27M	31.91M	26.77M
15-09-2020	26.64M	26.37M	28.55M	32.30M	27.08M
16-09-2020	26.90M	26.58M	28.83M	32.71M	27.33M
17-09-2020	27.15M	26.79M	29.12M	33.13M	27.58M
18-09-2020	27.41M	27.00M	29.40M	33.55M	27.84M
19-09-2020	27.66M	27.21M	29.68M	33.96M	28.06M
20-09-2020	27.92M	27.42M	29.97M	34.37M	28.29M
21-09-2020	28.18M	27.63M	30.25M	34.77M	28.60M
22-09-2020	28.44M	27.84M	30.54M	35.18M	28.79M

Table 1: Confirmed Cases Forecasting

Confirmed Deaths Prediction and Forecasting

```
# Dataset Splitting
model_train=datewise.iloc[:int(datewise.shape[0]*0.95)]
model_test=datewise.iloc[int(datewise.shape[0]*0.95):]
y_pred=model_test.copy()
# Training the model
model_arima= auto_arima(model_train["Total Deaths"],trace=True, error_action='ignore', start_p=1,start_q=1,max_p=3,max_q=3,
                  suppress_warnings=True,stepwise=False,seasonal=False)
model_arima.fit(model_train["Total Deaths"])
# Predicting the model on the test set
prediction_arima=model_arima.predict(len(model_test))
y_pred["ARIMA Model Prediction"]=prediction_arima
# Forecasting the confirmed cases
new_date=[]
ARIMA_model_new_prediction=[]
for i in range(1,60):
    new_date.append(datewise.index[-1]+timedelta(days=i))
    ARIMA_model_new_prediction.append(model_arima.predict(len(model_test)+i)[-1])
model_predictions["ARIMA Model Prediction"]=ARIMA_model_new_prediction
```

Figure 22: ARIMA Model

```
# Dataset Splittng
prophet_pred=Prophet(interval_width=0.95,weekly_seasonality=True,)
prophet_deaths=pd.DataFrame(zip(list(datewise.index),list(datewise["Total Deaths"])),columns=['ds','y'])
# Training the data
prophet_pred.fit(prophet_deaths)
forecast_pred=prophet_pred.make_future_dataframe(periods=60)
forecast_confirmed=forecast_pred.copy()
# Predicting the model
confirmed_forecast=prophet_pred.predict(forecast_pred)
print(confirmed_forecast[['ds','yhat', 'yhat_lower', 'yhat_upper']])
```

Figure 23: PROPHET Model

Worlwide Covid-19 Confirmed Deaths forecasting:

Date	ARIMA Model Forecasting	Prophet Forecasting
15-08-2020	0.73M	0.76M
16-08-2020	0.74M	0.77M
17-08-2020	0.74M	0.77M
18-08-2020	0.75M	0.78M
19-08-2020	0.75M	0.79M
20-08-2020	0.76M	0.79M
21-08-2020	0.76M	0.80M
22-08-2020	0.77M	0.81M
23-08-2020	0.77M	0.81M
24-08-2020	0.78M	0.82M
25-08-2020	0.78M	0.83M
26-08-2020	0.79M	0.84M
27-08-2020	0.79M	0.84M
28-08-2020	0.80M	0.85M
29-08-2020	0.81M	0.86M
30-08-2020	0.81M	0.86M
31-08-2020	0.82M	0.87M
01-09-2020	0.82M	0.88M
02-09-2020	0.83M	0.88M
03-09-2020	0.83M	0.89M
04-09-2020	0.84M	0.90M
05-09-2020	0.84M	0.91M
06-09-2020	0.85M	0.91M
07-09-2020	0.85M	0.92M
08-09-2020	0.86M	0.93M
09-09-2020	0.86M	0.93M
10-09-2020	0.87M	0.94M
11-09-2020	0.88M	0.95M
12-09-2020	0.88M	0.95M
13-09-2020	0.89M	0.96M
14-09-2020	0.89M	0.97M
15-09-2020	0.90M	0.97M
16-09-2020	0.90M	0.98M
17-09-2020	0.91M	0.99M
18-09-2020	0.91M	0.99M
19-09-2020	0.92M	1.00M
20-09-2020	0.92M	1.01M
21-09-2020	0.93M	1.02M
22-09-2020	0.93M	1.02M

Table 2: Confirmed Deaths Forecasting