CSE464: Advanced Database Systems
Fall 2021

Project Report
Group No. – 3 (Section 1)

Submitted by:

Student ID	Student Name	Contribution Percentage	Signature
2018-1-60-032	Fahim Faez Abir	35	
2017-3-60-058	Arnob Ghosh	25	
2017-3-60-068	SK Mohammad Asem	25	
2018-1-60-066	Abdullah Abdur Rahman	15	

1. Introduction

In December 2019 covid-19 was discovered and the overall scenario was very devastated still, we are struggling and dealing with the covid-19. So, from the dataset, we are trying to predict the future death case with time-series analysis. We used the ARIMA model of time-series analysis. We processed our data using different hypotheses to make it acceptable to the ARIMA model as time series required stationary data. We predicted the future range and we checked the accuracy of our model using RSS (Residual Sum Square) and it turned out very well.

2. Data Pre-processing

We mainly concentrated on date and death cases, so we dropped lab tests and confirmed cases from the dataset. Firstly we converted our day column from object to datetime64 [ns]. Then we selected day as an index because we are considering time as the index. Since time-series analysis requires stationary data, we converted our dataset to stationary data. How can we define data to be stationary?
At first, the mean of data should always be constant according to time to be stationary data. Variance should be equal at a different time interval. Variance is the distance from the mean. Each point distance from the mean should be equal. Another is auto-covariance which does not depend on time. These three conditions should be satisfied to be stationary data [5]. After pre-processing How we are sure if our data is stationary or not? There are two ways of checking if data is stationary or not. One is the rolling statistic which calculates moving average and moving variance. We have to calculate these two with a specific time window. Another is the Augmented Dickey-Fuller Test. The ADF Test (Augmented Dickey-Fuller Test) is a typical statistical test for determining whether or not a time series is stationary. When examining the stationary of a series, it is one of the most widely employed statistical tests [3]. ADF is mainly used for checking if there are any null-hypothesis in the data. If the test statistic is less than the critical value we can cut out the null hypothesis. If there is no null hypothesis then it is stationary data. By fulfilling the above requirements and checking all the parameters we confirmed our processed data to be stationary.

3. Dataset Characteristics and Exploratory Data Analysis (EDA)

The name of our dataset is a covid dataset, this is a dataset of the covid-19 situation of Bangladesh. It has 4 columns of date, death case, confirmed case and, test case. The data collection started from date April 4-2020, and we are working on the data till December 20-2021. There is a total of 626 rows in our dataset and 4 columns at first, after the pre-processing of data there are only two rows, date and death-case. As we are developing time-series date is our primary variable.

	Death Case
count	626.000000
mean	44.798722
std	54.528270
min	0.000000
25%	14.000000
50%	30.000000
75%	44.000000
max	264.000000

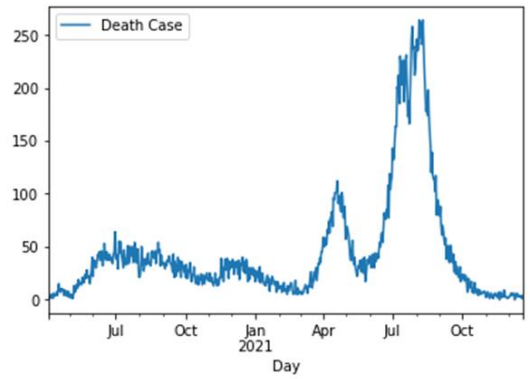

Figure: Calculative and Graphical Representation

From the above graph, we can see both upward and downward trends and seasonality in respect of death cases over time of our data.

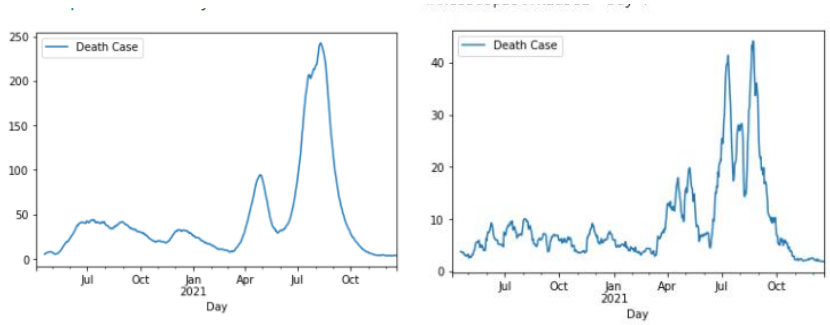

Figure: Standard deviation and Mean

We calculated standard deviation and mean over time to death case, and we can see that it is not stationary because we cannot find them constant with the time interval.

4. Machine Learning Models

Time series analysis is a method for studying a collection of data points over a period of time. Instead of capturing data points frequently or randomly, researchers use time series analysis to capture data points at regular intervals throughout a specified length of time [1].
What is the reason behind using time-series analysis? Where there are many other algorithms in machine learning. There is a fixed variable in the algorithm and that is time and in different datasets, it is represented differently. In other algorithms let's say regression there are two types of variables, one is dependent and the other is independent. Depending on these two types of variables we figure out the relativity of the variables. Then we analyze the whole system and get our desired conclusion. But in time series analysis we are only depending on one variable that is time. And on basis of the variable, we can predict the future by analyzing the past. In time series analysis time is plotted in X-axis and Y-Axis is our target value. Time-series analysis requires data on a daily, monthly, or yearly basis, it can also be weekly data with a regular interval, not anything with an irregular period gap. There is much use of time-series analysis in different fields i.e. it can be used to predict the situation of the stock market, sale analysis in the supermarket, etc.

But there are some exceptions where there is no time-series analysis needed like the case of constant data, for example in our country there is an increase in sales at a specific time of the year mostly on occasions and it is repeated every year and these do not need any prediction.

We cannot use time-series analysis in every dataset some conditions and requirements need to be fulfilled to get the desired result firstly the data should be stationary. Almost all the models of the time series analysis assume that the data is stationary. The benefit is that stationary data has a probability of repeating itself similarly in the future in a gap of time.

5. Description of Models and Associated Parameters

In time-series analysis there are many models, we chose ARIMA to do our project. ARIMA is one of the best models to work in time series analysis. ARIMA consists of three parts, The AR is autoregressive which is a correlation between previous times to current time, I is for Integration and MA is moving average which is the mean value over the time period. These three parts are considered three parameters to define the ARIMA model. AR is defined as a p-value which is called autoregressive lag, 'I' is defined as d value which is the order of differentiation, and MA is defined as q value which is moving average. To find p-value and q value we have to implement PACF (Partial autocorrelation) and ACF (Auto Correlation) graph [4].

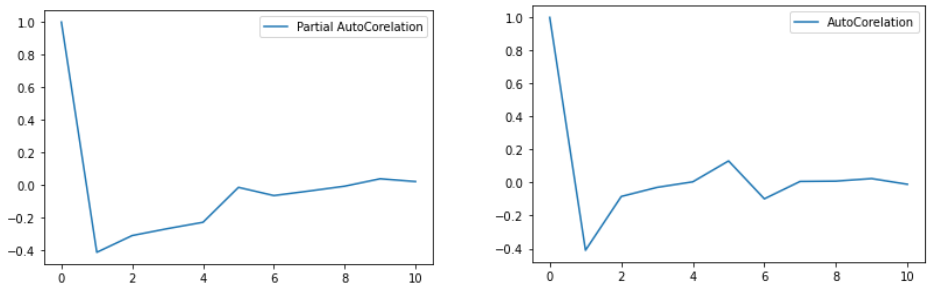

Figure: PACF and ACF graph

If we consider the vertical intersect between the 0.00 of the Y-axis and the curve and go down perpendicularly to X-axis then we will get the p-value and q value from PACF and ACF accordingly. Here from the graph, we can see our p-value is 1, and q values are also 1. For the d value part, we consider it as 0.

6. Performance Evaluation

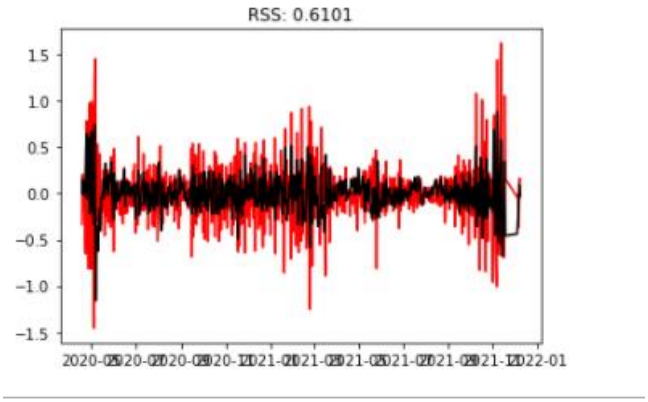

Figure: RSS graph

The residual sum of squares (RSS) is a statistical approach for determining how much variance in a data set is not explained by the regression model. Instead, it calculates the error term or variance in the residuals [2]. RSS is preferred the low the better. Our RSS value is 0.6101.

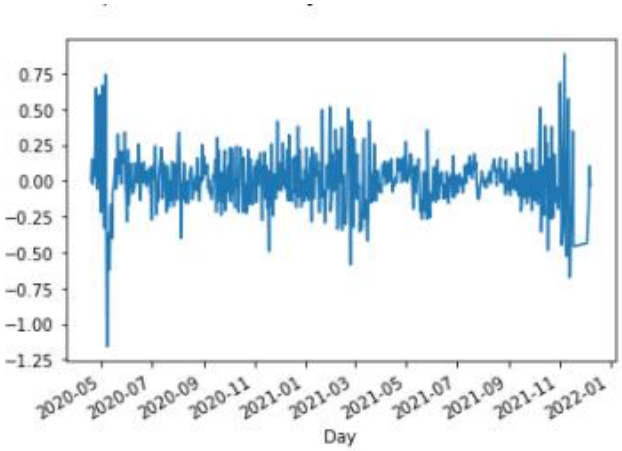

Figure: Prediction graph

Here we can see the shifted log-scale values daily of our data in Y-Axis and time in X-axis

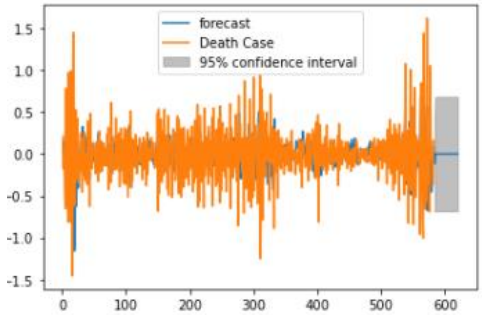

Figure: Forecast value

From our 590 days' worth of data, we are predicting the next 30 days of data, so the total is 620 data. Here the blue lines are the forecast value, grey lines are the confidence level and the forecast value cannot go outside of the confidence level.

7. Discussion

Since we are working with time series analysis our data must be stationary. So we converted our non-stationary to stationary to get better performance. Firstly we calculated our moving average with a window size of 15 as our dataset provides daily data so we considered a half month of data to be predicted. Then we plotted our rolling average to verify the stationarity of our data.

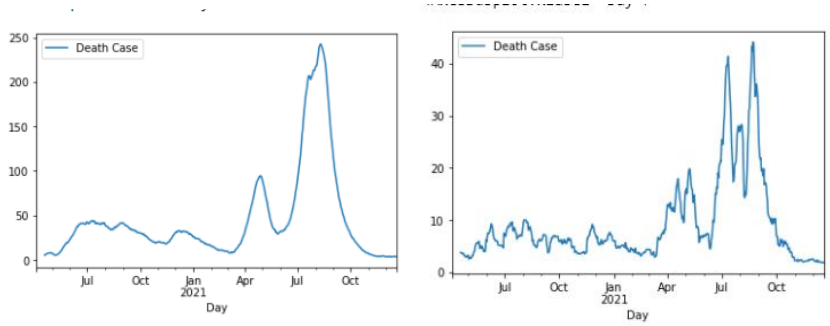

Figure: Mean and Standard Deviation

From the above graph, we can see that the square root of the variance is not constant so we concluded from this test that our primary data is not stationary.

Next, we tested our data with the Augmented Dickey-Fuller Test to check our data is stationary or not.

```
Test Statistic            -2.980075
p-value                    0.036797
#Lags Used                14.000000
number of observations    611.000000
critical Value(1%)         -3.441098
critical Value(5%)         -2.866282
critical Value(10%)        -2.569295
dtype: float64
```

Figure: a result of ADF Test.

The result shows that the P-value is less. And the less the p-value is the better it is for us. But we can see that our dataset has a null hypothesis. Because the test statistic is greater than the critical value. So, we concluded that our dataset is not stationary.
There are many ways to convert data to stationary. We can see the Y-Axis changing after the logarithm scaling.

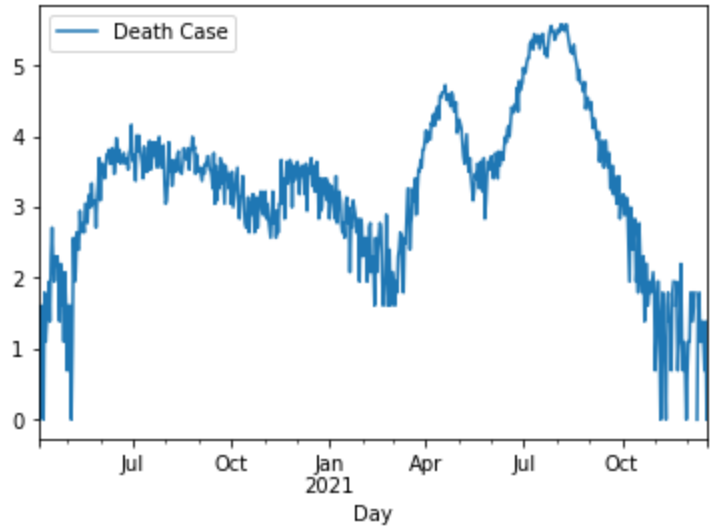

Figure: After Log Scaling

Then we calculated the moving average of the log scaled data and plotted the data but it did not convert into stationary yet but is better than the previous test.

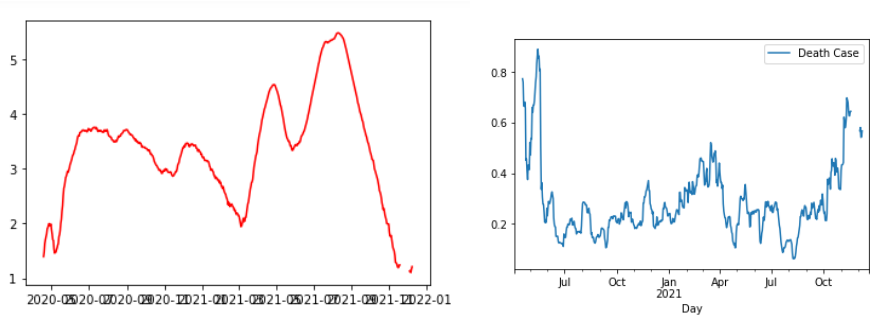

Figure: Moving Average and Standard Deviation

Next, we used the log scaled data and subtracted the moving average from the scaled data to make our data stationary. Then we got the difference from the subtraction and calculated mean and standard deviation from that, then we plotted the data and analyzed that the data is far better than before.

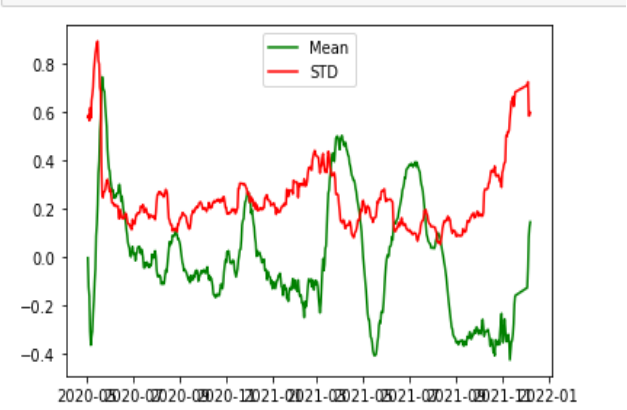

Figure: standard deviation and mean of the subtracted value

Then we performed ADF Testing again and we got to the conclusion that our data has now become stationary.

```
Test Statistic                -4.543439
p-value                        0.000164
#Lags Used                     4.000000
number of observations       580.000000
critical Value(1%)            -3.441675
critical Value(5%)            -2.866536
critical Value(10%)           -2.569431
dtype: float64
```

Figure: ADF Result

We can see that the p-value is less than before also test statistic is less than the critical value. We take the subtracted value and put it in a python built-in function called shift. After all the above hypotheses we trained the model to get our desired result. And our performance is very good as we mentioned in the performance evolution.

8. Conclusion

To perform this project, we gained some knowledge about time-series analysis and we found it very useful. It is quite difficult for us to implement such a new algorithm for us without supervision but there are many documentations on the internet [5] and we tried to do our best but there are still some laggings that we have not overcome. We implemented it in the covid-19 dataset which has covered the whole world with fear and panic, so if we can predict the future situation of covid-19 it will be very helpful to us. Because we can prevent the situations that might occur and there is a saying that prevention is better than cure.

9. References

[1]2022. [Online]. Available: https://www.tableau.com/learn/articles/time-series-analysis?fbclid=IwAR1YMXQCPHtBlE7XRx1GucasIAx3SqsnfnOUbf8uEqFfBiuPjYroYZs10bs. [Accessed: 23- Jan- 2022].
[2]"How the Residual Sum of Squares (RSS) Works", *Investopedia*, 2022. [Online]. Available: https://www.investopedia.com/terms/r/residual-sum-of-squares.asp. [Accessed: 23- Jan- 2022].

[3]"Augmented Dickey-Fuller (ADF) Test - Must Read Guide - ML+", *Machine Learning Plus*, 2022. [Online]. Available: https://www.machinelearningplus.com/time-series/augmented-dickey-fuller-test/?fbclid=IwAR25LKV2eoBJLKB1MftL8eITH081hxGrFNNIN12AFYhXpdSYr0DABh9gY5w. [Accessed: 23- Jan- 2022].
[4]*Youtu.be*, 2022. [Online]. Available: https://youtu.be/e8Yw4alG16Q. [Accessed: 23- Jan- 2022].
[5]*Youtu.be*, 2022. [Online]. Available: https://youtu.be/e8Yw4alG16Q. [Accessed: 23- Jan- 2022].

10. Appendix

```
import pandas as pd
import numpy as np

import matplotlib.pyplot as plt
%matplotlib inline

data = pd.read_csv(r"C:\Users\HP\Downloads\covid_dataset (1).csv")
data
data.dtypes

data.drop(['Lab Test', 'Confirmed case'], axis = 'columns', inplace = True)
data['Day'] = pd.to_datetime(data['Day'], infer_datetime_format = True)
data.dtypes
covid_data = data.set_index(['Day'])
covid_data.plot()
covid_data.describe()
covid_datamean = covid_data.rolling(window = 15).mean()
print(covid_datamean.head(15))
covid_datamean.plot()

covid_datastd = covid_data.rolling(window = 15).std()
print(covid_datastd)
covid_datastd.plot()
#graphical view
main_data = plt.plot(covid_data,color = 'blue')
mean = plt.plot(covid_datamean,color = 'green', label = 'Mean')
STD = plt.plot(covid_datastd,color = 'red', label = 'STD')
plt.legend(loc = 'best')
plt.show()

from statsmodels.tsa.stattools import adfuller
def addfuller_testing(covid_data):
datatest = adfuller(covid_data['Death Case'],autolag = 'AIC')
output = pd.Series(datatest[0:4],index = ['Test Statistic','p-value','#Lags
Used','number of observations'])
for key,value in datatest[4].items():
output['critical Value(%s)'%key]= value
print(output)
addfuller_testing(covid_data)

datalogscale = np.log(covid_data)
datalogscale.plot()
moving_average = datalogscale.rolling(window=15).mean()
moving_std = datalogscale.rolling(window=15).std()
#plt.plot(datalogscale)
plt.plot(moving_average, color = 'red')
##plt.plot(moving_std, color = 'black')
moving_std.plot()
logsminmovs =  datalogscale - moving_average
logsminmovs.dropna(inplace = True)
logsminmovs

movA = logsminmovs.rolling(window=15).mean()
stdA = logsminmovs.rolling(window=15).std()
```

```
#main_data = plt.plot(data,color = 'blue')
mean = plt.plot(movA,color = 'green', label = 'Mean')
STD = plt.plot(stdA,color = 'red', label = 'STD')
plt.legend(loc = 'best')
plt.show()

#adfuller testing
addfuller_testing(logsminmovs)

logsminmovs.plot()
shiftlogminmovs = logsminmovs - logsminmovs.shift()
shiftlogminmovs.plot()
shiftlogminmovs.dropna(inplace = True)
addfuller_testing(shiftlogminmovs)
shiftlogminmovs
from statsmodels.tsa.stattools import acf,pacf
Q_value = acf(shiftlogminmovs,nlags=10)
P_value = pacf(shiftlogminmovs,nlags=10, method = 'ols')
plt.plot(Q_value,label = 'AutoCorelation')
plt.legend(loc = 'best')
plt.plot(P_value,label = 'Partial AutoCorelation')
plt.legend(loc = 'best')
from statsmodels.tsa.arima_model import ARIMA
data_model = ARIMA(shiftlogminmovs, order = (1,0,1))
data_model_fit = data_model.fit()
plt.plot(shiftlogminmovs,color = 'red')
plt.plot(data_model_fit.fittedvalues,color = 'black')
plt.title('RSS: %.4f'% sum(data_model_fit.fittedvalues-shiftlogminmovs['Death
Case'])**2)
predictions = pd.Series(data_model_fit.fittedvalues,copy = True)
predictions.head()
predictions.plot()
plt.plot(covid_data)
data_model_fit.plot_predict(1,620)
prediction = data_model_fit.forecast(steps = 30 )
prediction[1]
```