Kaggle COVID19 Forecasting

Data Science Lab

Upender Shana Gonda
shanag01@ads.uni-passau.de

ABSTRACT

Currently millions of people throughout the world are getting affected by the corona virus. The percentage of confirmed, recovered and death cases are increasing daily and it's becoming a big challenging to the governments and welfare organizations to controlling the spreading of cases. So many organizations are providing the resources to the governments for preventing or controlling the corona cases allover the world. The government needs the appropriate statistics and the data prediction of future cases so that proper safety measures and awareness can be driven to controlling or preventing the corona cases(confirmed, recovered, and death cases). It is quite challenging to predict the number of death, confirmed, and recovered cases because of fluctuations in the data. In this paper, we are going to work with John Hopkins corona virus data, which will be analyzed by using the machine learning and data science techniques, We are focusing on implementing the two-time series models (ARIMA and SARIMA) on the John Hopkins dataset to predict the number of cases (confirmed, recovered and death) of different countries. We were able to compare the performance of the two models by using the evaluation Metrics.

INTRODUCTION

Coronavirus is a severe ongoing novel pandemic that emanated from Wuhan's capital of Hubei, China in December 2019, and has caused widespread devastation all over world. The World Health Organization (WHO[1]) named this coronavirus as "COVID19". On January 30 COVID-19 epidemic was declared to constitute a Public Health Emergency of International Concern by WHO, and on March 11, 2020, WHO declared it as a pandemic[22]. This disease causes mild to severe respiratory illness and cough, flu headache. Currently, it has affected more than sixty lakhs of people throughout the world, and reportedly around 379,941 people have lost their lives. As of now, Infected patients are treated with a supportive measure since there is no vaccine or antibiotic available[13, 20]. Due to the absence of a vaccine and a rapid potential transmission, the world has no other option but to prevent further transmission. In response to the prevention of deadly pandemic, the majority of countries across the globe have taken strict measures to ensure the social distancing, such as the closure of educational institutions, airports, business sectors, offices and banning all public events[13, 20].

But these measures have been taken on the expense of economic disruption, which has not only been devastated but also has spillover applications in human life [13, 20]. First and foremost, the ground level challenge is to forecast the cases of COVID19 means predicting for the occurrence of coronavirus disease in the specified area ahead of time and predictions of probable outbreaks or increase in the intensity of the disease so that the suitable control measures can be undertaken in advance to avoid loss of lives and economic crisis[13, 20].

Throughout this paper, we are going to work on the John Hopkins coronavirus data set, which relates to the time series representing the infected cases, recovered cases and, death cases. Firstly, we will describe our research statement and workflow, to direct our study in this way. Then, applying necessary pre-processing steps on the data, we will try to analyze components of time series. Afterwards, data science and machine learning approaches are being implemented to experiment with various forecasting methods and explain how different time series models can be used to predict the cases in the different regions of the world. In the last section, experimental results and suggestions for future directions will be presented.

1 PROBLEM STATEMENT AND DESCRIPTION

The time series forecasting is predicting future events based on past observations or values in time-series format data. Time series contains four components: trend, seasonality, irregularity, and cyclic. It is essential to bring the data in a stationary format for forecasting time series. We aim to analyze the multivariate time series which refers to changing values taken from multiple variables over time to forecast the spread of COVID19 cases of particular countries. Based on reliable information sources like WHO and Johns Hopkins University (JHU[2]) we are getting the daily updates on COVID19 cases regarding confirmed cases, recovery, and death among all over the countries. Firstly, the research work concentrates on comparing the coronavirus cases between different countries from the considered dataset. Furthermore, the work aims to check whether the dataset is stationary or not, visualizing and removing the trend and seasonality, next to selecting, training and implementing the models on a dataset and Finally, considering the past data into account, we try to predict/forecast the coronavirus cases among different countries or a particular country.

[1] https://www.who.int/

[2] https://www.jhu.edu/

2 RESEARCH GOAL AND QUESTIONS

In this paper, our main focus will be on the following questions:

2.1 Is there any similar trend among different countries?

We are going to analyze the dataset and trying to find any similar patterns or trends that are present, in the number of cases (death, recovered and confirmed) by visualizing for the considered countries or any particular country.

2.2 Which time series forecasting model is properly-suited, to predict future cases?

In this research work, we aim to implement and compare two-time series models namely ARIMA and SARIMA on the given dataset to predict confirmed, recovered, and death cases of a country. We try to identify the best-suited evaluation metric and keep that as the comparison baseline to compare the two models.

3 RELATED WORK

Related work for prediction of the spread of COVID19 has been done using different algorithms or techniques some of them are listed below:

One of the related work on COVID19 forecasting in, artificial intelligence and deep learning using Non-Linear Regressive Network (NAR)[1]. In this approach, they predicted the coronavirus cases for 9 countries between march and july[1]. All the necessary data is taken from the WHO[22]. The evaluation is done for both active cases and number of deaths. It is found that for global prediction, the performance of a network is 2.65[1].

One more related work on COVID-19 outbreak predictions is for India using SEIR and Regression Model[24].In this study, based on John Hopkins University collected data, COVID19 cases have been analysed for India from January to March and the predictions have been made for next 14 days using SEIR and Regression models[24]. By using RMSLE, the models performance was evaluated and observed that SEIR model achieved 1.52 and regression model achieved performance of 1.75. Between two models the error rate is found to be 2.01[24]. Also, calculated the spread of the disease value (R0) as 2.02. Finally, two models predicted that expected cases may fall between the range of 5000-6000 in the coming two weeks[24].

Another related Paper focused on, to forecast the number of deaths and active cases during the spread of virus in Italy, Spain and, Turkey from February 2, 2020, to March 27, 2020[6]. The model used for Forecasting is ARIMA which gave the prediction for number of cases in Italy,Turkey and, Spain was 76.1,81.7 and 95 percent respectively[6]. whereas, for the number of deaths in turkey , Italy and spain calculated as 71.4, 92.7 and 95.8 percent respectively and it was discovered that in all three predictions that the error terms are stationary as a outcome of Ljung-Box[6] .

Moreover, the model is considered as a good model and can be used for future predictions as MAPE, value is less than ten percent. In conclusion, expectation of cases are like, in Spain and Italy decreasing by July and in Turkey declined by September. While in the case of deaths, by July in Italy and Spain expectations of cases are lowest and for Turkey expected to be the highest[6].

On the other hand, china initially worked on covid19 real-time forecasting using existing models which are used during previous outbreaks to produce short-term predictions for corona cases in Hubei province[14]. They collected daily active cases for the COVID19 outbreak for each china area from the National Health Commission of China. Then, they started providing the forecasts of corona cases for every five consecutive days, with a logistic growth model, Richards growth model, and a sub-epidemic wave model[14].

The three models latest predictions are, on data until 9-2-2020, models estimated the cases like around range of 7409 to 7496 but, also some extra confirmed cases also registered in Hubei within the next five days, 1128–1929 additional cases also added in other places of china[14]. Models also predicted the cases for next 15 days by an average 37,415 to 38,028 in Hubei and 11,588 to 13,499 in other regions of China[14].

4 TIME SERIES MODELS

4.1 ARIMA

ARIMA - Autoregressive Integrated Moving Average [23] is a combination of AR - Autoregressive , MA - Moving Average , and the notion of Integration (I). Autoregression (AR): the output of the regression model depends on the linear combinations of input values. Integrated (I): In this to make time-series stationary, differencing is used on original observations. Moving Averages (MA): this technique is used to remove the variations between time steps(observations) of time series.

Each component in the model are indicated as a parameter(integer values) like ARIMA(p, d, q). These p,d,q parameters are defined as the lag value(p) – number of lag observations in the time series, Degree of difference(d) – calculates the number of differences of observation and Moving average(q) – deals with window size, known as the order of moving average [23].

4.2 SARIMA

SARIMA [9] is also known as seasonal ARIMA because that supports the seasonal components in time series. Whereas ARIMA does not support seasonal data [9], this is one of the important drawbacks of the ARIMA model. SARIMA needs the parameters of trend and seasonal components for model configuration. The trend parameters of the SARIMA model are similar to ARIMA but seasonal parameters are additionally required in SARIMA implementation. The four seasonal parameters are P,D,Q and M. where P,D,Q are auto-regressive, difference, moving average order respectively and M: Number of time steps required for a single seasonal period [9].

5 WORKFLOW

The following figure is the work flow of our proposed solution:

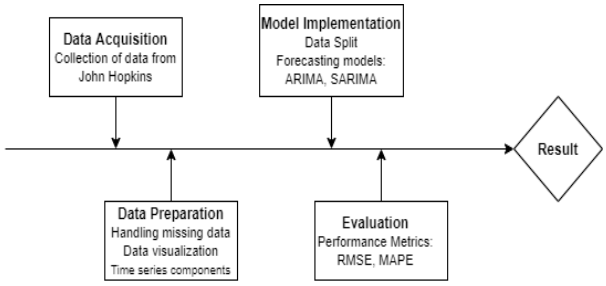

Figure 1: Workflow definition of the proposed solution

6 DATA ACQUISITION & PRE-PROCESSING

6.1 Data acquisition

For this project, data has been collected from john Hopkins University which is working on the global data of coronavirus cases from different reliable sources[5]. This dataset contains the daily time series summary tables, including infectious, recovered, and fatalities cases, about all the affected countries[4]. With some modifications, it will be suitable for developing time series models to better predict how the epidemic disease reacts in the future and it is given in CSV format.

6.2 Frameworks used

The following open-source frameworks and libraries were used as part of the project implementation. The data analysis and manipulation were achieved by using the Pandas[3] and Numpy[4]. Seaborn[5] and MatplotLib[6] functions were used for visualization of data. The ARIMA and SARIMA models were implemented using Statsmodels[7].

6.3 Data preprocessing

As mentioned in the workflow diagram (Figure 1), the preprocessing technique involves steps such as missing data, data handling, removing irrelevant features, analyzing the time series components, data visualization. We might need to change the datatype of categorical values. The Time series analysis technique has been used in this study to analyze the data. The purpose of using this technique is to forecast the behavior of variables in the near future based on previous behavior.

From the above figures 2, 3 and 4 we can observe that, U.S is registered with more confirmed, recovered and death cases compare to all other countries.

6.3.1 ***Removing Irrelevant Features.*** In this dataset, we will utilize 5 columns namely Country/Region, date, confirmed cases, recovered, and death columns. Along with this, some other columns

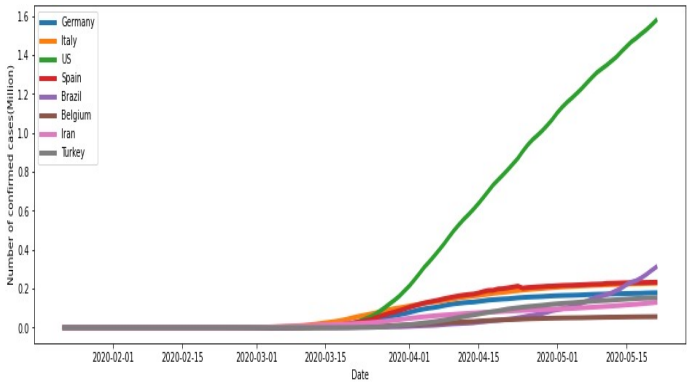

Figure 2: Number of confirmed cases of the Top countries from Jan to May

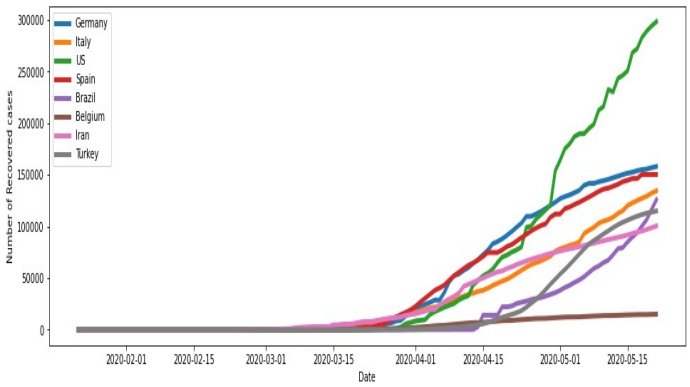

Figure 3: Number of recovered cases of the Top countries from Jan to May

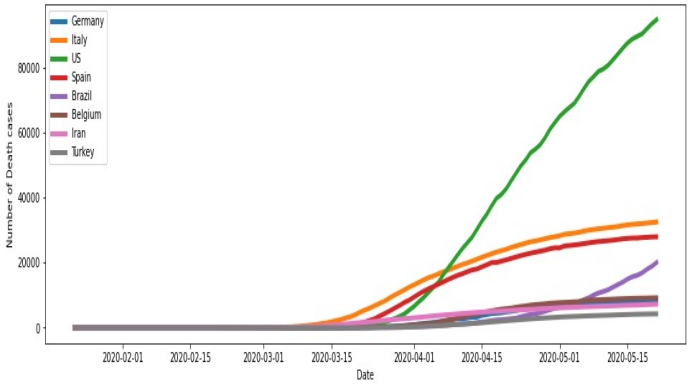

Figure 4: Number of Death cases of the Top countries from Jan to May

are presented which are irrelevant to our problem, So we are going to ignore them. For example, there are longitude and latitude columns in our dataset which may not be useful for our problem.

[3]https://pandas.pydata.org/
[4]https://numpy.org/
[5]https://seaborn.pydata.org/
[6]https://matplotlib.org/
[7]https://www.statsmodels.org/stable/index.html

6.3.2 ***Visualization.*** A visual summary of data is more appealing and gives meaningful insights to our data[21]. The goal of data analysis is to describe the characteristics of a dataset logically. Even if we can get important features without visualization, it will be tough to convey the meaning, Thus visualization makes it easier for us to find patterns and trends and provides a better understanding on information than going through hundreds of rows on spreadsheet[21]. One visualization example from our data in Figure 5, we have visualized increasing of deaths, confirmed, recovered cases of Germany over time(From Jan to May) using a line plot function. We can observe in the graph that, The rising trend of confirmed and recovered cases. Whereas, death cases are remains almost uniform.

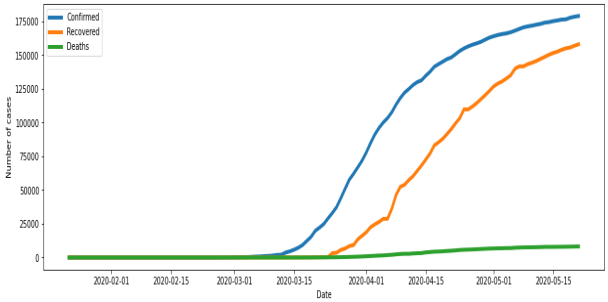

Figure 5: Cases of Germany

6.3.3 ***Missing values:*** After analyzing our data, In date column we observed the format of Date for all rows is not similar so using date-time formatting, made all dates format same also we found in 'Province/state' column, there are some missing values(NaN). For only a few countries, this feature is given and for most of them, this feature is missing. Anyway, we are not going to use this column for our implementation. Because we are focusing only on country/Region wise data So, we decided to remove or ignore that feature from our dataset.

7 COMPONENTS OF TIME SERIES

There are 4 components of time series: trend, seasonality, irregularity, and, cyclic. The patterns should also illuminate our functional understanding if they contain any annoying behavior it should be removed so that the intentions can be more clearly analyzed[12]. Because our data set span is only almost five months we cannot consider the cyclic component because the cyclic component is analyzed in long term oscillations, we will disregard the cyclic component in our analysis. In Figure 6, decomposition of time series components in confirmed cases in Germany can be seen.

7.1 Trend

The trend can be said as a long term movement which time series exhibits. It can be either an increasing or a decreasing long term movement. If a time series does not show any upward or downward trend, then we can say that time series is stationary [18]. In our datasets, we observed some upward trends as shown in Figure 6.

7.2 Seasonality

Time series is said to have seasonality component when it experiences regular seasonal fluctuations every year during same month or every week during a same day. In our dataset, we have observed seasonal factors as shown in Figure 6.

7.3 Irregularity

Random or irregular fluctuations occur in time series for a short time which is mainly caused by unpredictable influences. This component is unpredictable and non-repetitive. These short term fluctuations are also called as noise or residual in the statistical model .

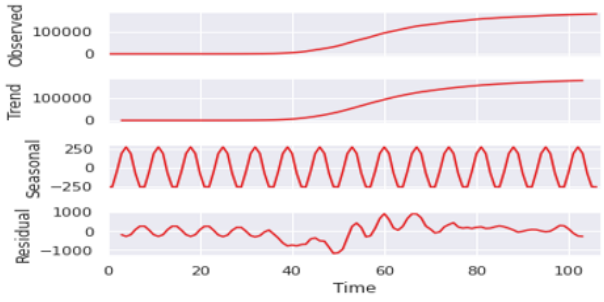

Figure 6: Decomposition of time series component in confirmed cases of Germany

8 DATA TRANSFORMATION

Most of the models require stationary data for forecasting, so data transformation is necessary to check for stationarity. Some of the tests for checking stationarity is Rolling statistics and Augmented Dickey-Fuller Test. By using this tests, we will check whether series data is stationary or non-stationary, if there is any non-stationary data is present then, we will transform it to stationary by applying some present methods, which will be discussed later.

8.1 Rolling Statistics

We will use the rolling- mean, standard deviation as metrics for computing rolling statistics. The general rule is that if rolling mean and standard deviation is constant that means time series is stationary but when we applied this technique to different countries in the data set for confirmed cases, recovered and deaths, we found that data for most of the countries was non-stationary.

For instance, as shown in Figure 7 we have considered Germany confirmed cases for rolling mean and standard and we observed that the series data is non-stationary and when we applied same to United kingdom we found that series data is almost stationary.

8.2 Augmented Dickey-Fuller Test

Augmented Dickey-Fuller (ADF)[11, 16]is one of the important tests for verifying or checking whether the considered time series from dataset is stationary or not. It is also helpful for checking the presence of unit root and which gives the result about the time series stationarity. Generally, there are two types of tests, one is a

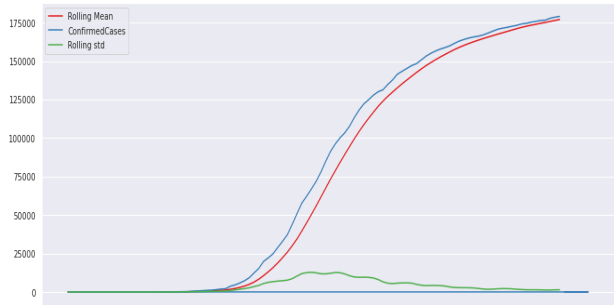

Figure 7: Applying rolling statistics with window size 7

null hypothesis and the other one is an alternate hypothesis. If a series with unit root = 1 then it is a null hypothesis, whereas the alternate hypothesis has or with no unit root value. ADF test results provide test statistics value and 3 critical values at 1%, 5% and 10%. By comparing the values between test statistics and three critical values, we can reject the null hypothesis if the test statistics is less than the critical value, which means the series is stationary[11, 16]. Test statistics value is higher or greater than critical value implies series is non-stationary.

After applying the ADF test on Germany confirmed COVID19 cases, we got values of test statistics as -0.842149 and critical values of 1%, 5% and 10% as -3.502705, -2.893158 and -2.583637 respectively. As the value of test statistics is higher or more than all critical values, we can assume that time series data is non- stationary. We have also applied the ADF test on Germany recovered cases and we observed the test values as -1.011717 for test statistics and critical values of 1%, 5% and 10% as -3.514869,-2.898409 and -2.586439 respectively, from this values of test statistics and critical, we can assume that the recovered cases time series is also non-stationary.

Table 1: Stationarity Test Result

	Cases		
Country	**Confirmed**	**Recovered**	**Deaths**
U.S.A	False	False	False
Germany	False	False	True
Italy	False	False	False
U. K	True	False	False
France	False	False	False
Spain	False	False	False
Brazil	False	False	False
Belgium	False	True	False
Iran	False	False	False
Mexico	False	False	False

But, for Germany deaths cases, we got values of test statistics as -4.526805 and critical values of 1%, 5% and 10% as -3.546395, -2.911939 and -2.593652 respectively. As the value of test statistics is less or lower than all critical values, we can assume that time series data for deaths cases is stationary.

In the same way, we have applied the ADF test for remaining all nine countries for Confirmed, Recovered and Deaths cases and our observations are listed in the table 1.

8.3 Handling Non-Stationary Time Series

Based on the ADF and Rolling statistic tests, the stationarity of the different time series has been recognized. For example, from table 1, it can be observed that 'Confirmed cases of Germany' is non-stationary. Using techniques like differencing, log transformation these non-stationary time series can be converted into stationary[7].

Differencing [11] is one of the most popular or important method in the time series to find the difference between consecutive observations [11]. From the time series it will try to calculate the deviating means. By applying the differencing technique, the differences between the consecutive observation are estimated to make a time series stationary [7]. Also, the ADF and Rolling statistic tests can also be used to check whether already differenced data is stationary or it requires any further differencing with a higher-order.

8.4 ACF: Autocorrelation Function and PACF: Partial Autocorrelation Function

From [2, 26] autocorrelation is helpful to find the similarity between two consecutive occurrences or observations and it is a way to calculate how one observation is linearly related to another observation. Autocorrelation mainly helps to find the appropriate log value(p) which we will use in the implementation of a time series model, and to analyze at what extent present values of time series related to previous values. ACF also is known as complete autocorrelation function, because ACF will consider all components which time series has like seasonality, trend, cyclic and residual while finding respective correlation [17].

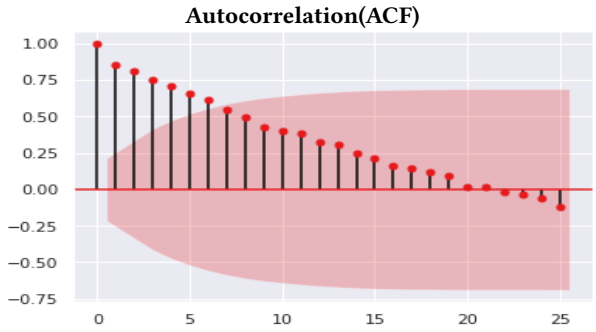

Figure 8: Auto correlation plot for 'confirmed cases of Germany' with lag values from 0 to 25

PACF is also the same as ACF. The only difference between both is that, it will find a correlation in residuals and tries to remove partially or not completely removed effects for providing a better version of correlation for the next lag to implement the series model. In other words simply, it will consider the intermediate values between two observations [17].

From the plot we can observe that in case of ACF, most of our values lies between 0 and +1 which represents positive correlation for 25 lags and that leads to need for differencing and autoregressive parameters.

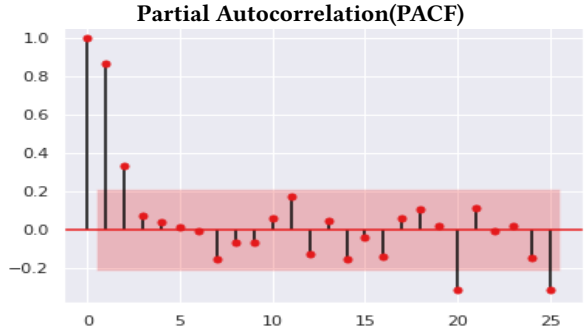

Figure 9: Partial auto correlation plot for 'confirmed cases of Germany' with lag values from 0 to 25

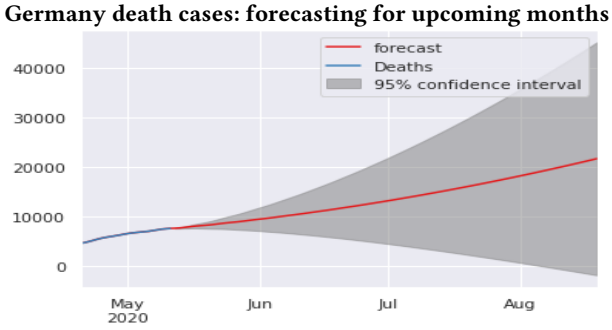

Figure 10: ARIMA: Actual Forecast

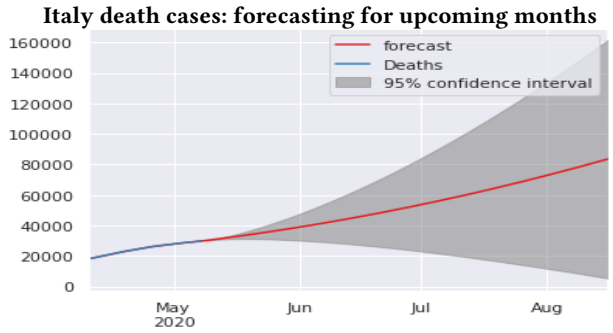

Figure 11: ARIMA: Actual Forecast

9 MODEL IMPLEMENTATION

To explain the different steps that are involving in implementing a forecasting model, we have considered the confirmed, recovered and death cases of 'Germany' and 'Italy' time-series data. In the implementation of time series models, we have used the statsmodels package of python. In the following sections, we will describe our models and results of our data set.

9.1 Dataset split : Train and Test

In order to develop a model, the first important step is splitting the data set into two train and test data. From [25], the model has to predict in the range of test data that initially we will build the model on train data. Then, the model's performance will be evaluated by comparing the test and predicted data [25]. Ideally, the confirmed, recovered and death cases of Germany and Italy time series dataset are split in chronological order like approximately 75% of the dataset is train data and the other 25% as test data.

9.2 ARIMA Model

For implementing the ARIMA model on the given dataset, firstly we have to obtain the ARIMA parameters(p, d, q). To get the optimal ARIMA parameters for confirmed, recovered and death cases of Germany and Italy we imported ARIMA function from statsmodels library and this function is used for entire time-series data. For instance, the function returned the optimal values(p, d, q) as (4, 2, 3) for death cases of Germany and (0, 2, 3) for death cases of Italy respectively. We have implemented the model to predict the death,confirmed, and recovered cases of Germany and Italy. From, these we considered the plots of death cases of both countries as an example to show how the prediction of cases in the future and to find how is the trend between both plots.

For evaluation of the performance of the model, the parameters are fitted on the train data then model forecasts the values in the test data range. Then, using the evaluation metrics we can evaluate the model's performance(it will be described in the evaluation section). The prediction graphs for death cases of Germany and Italy is shown in figure 10 and 11.

The confidence interval describes certain range of estimated values[3]. From figures 10 and 11, we can infer that the confidence interval represents 95% of the true estimated range of predicted cases.

By analyzing or comparing the above forecasting graphs of two countries, we can assume that there exists a similar trend(increase in cases) between death cases of Germany and Italy. As evidence, we are performing the Kullback-Leibler and Jensen-Shannon Divergence for both countries.

9.3 Kullback-Leibler and Jensen-Shannon Divergence

From [10], in order to compare two distributions there maybe a need of calculating the statistical distance between them. The better approach is to calculate the divergence between two distributions, divergence means finding how one distribution is different from others. Where in this case we are calculating the divergence for death cases of Germany and Italy. There are two commonly used divergence scores namely, Kullback- Leibler Divergence and Jensen-Shannon Divergence [10].

Kullback-Leibler Divergence score, shows how first distribution is different from second distribution.

Representation: KL(Germany || Italy), where "||" is divergence(Germany's divergence from Italy), basically the KL divergence score is not symmetrical. With reference to [10], we can formulate the equation of KL divergence as:

$$KL(Germany \parallel Italy) ! = KL(Italy \parallel Germany) \quad (1)$$

and it is also known as the “relative entropy”. By using the "rel_entr" function, we performed KL divergence for both countries. We yielded the following scores: KL(Germany||Italy) : -67479.016 nats and KL(Italy || Germany) : 267337.231 nats. Thus according to KL divergence, equation 1 satisfied .

Jensen-Shannon Divergence is another way to find out the similarity between two distributions. It uses KL divergence to calculate the score which is symmetrical. Therefore, the divergence of Germany from Italy should be the same as Italy from Germany. With reference to [10], we can formulate the equation of JS divergence as:

$$JS(Germany \parallel Italy) == JS(Italy \parallel Germany) \quad (2)$$

By using the "jensenshannon" function, we performed JS divergence for both countries. We yielded the following scores:
JS(Germany || Italy) = 0.001 nats and JS(Italy || Germany) = 0.001 nats. Thus, according to JS divergence, equation 2 satisfied.
So, now we can assume that there is a similarity between death cases of Germany and Italy. Therefore, we can conclude that there exists a similar trend between death cases of Germany and Italy forecasting.
In the same way, we have forecasted and performed the KL and JS divergence for the confirmed and recovered cases for both Germany and Italy and using these plots and divergence scores, we can conclude that there exists a similar trend between Germany and Italy.

9.4 SARIMA Model

We have implemented the model to forecast the confirmed, recovered, and death cases of Germany and Italy. From, these we considered the plots of death cases of both countries as an example to show how the prediction of cases in the future and to find how is the trend between both countries. The forecasting graphs for death cases of Germany and Italy is shown in Figures 12 and 13.

To evaluate the performance of the SARIMA model, we will be using the evaluation metrics which will be discussed in the evaluation section.

Germany death cases: forecasting for upcoming months

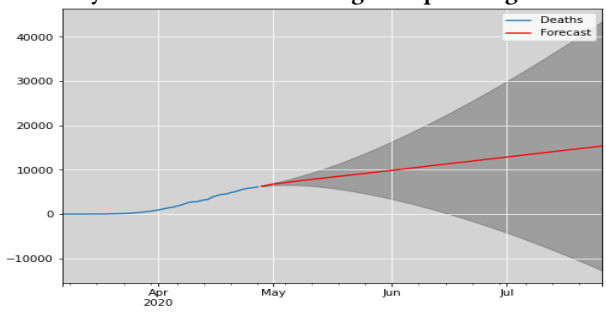

Figure 12: SARIMA: Actual Forecast

By analyzing and comparing both SARIMA model forecasting graphs of Germany and Italy and also after applying KL and JS divergence in both countries we can conclude that there exists a similar trend between them.

Italy death cases: forecasting for upcoming months

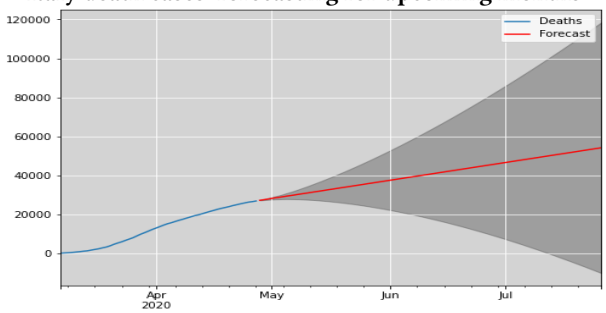

Figure 13: SARIMA: Actual Forecast

10 EVALUATION

The evaluation and performance are measure by comparing the predicted and actual output, the evaluation metrics namely RMSE and MAPE are used for evaluation of the performance of two models ARIMA and SARIMA.

10.1 Root Mean Squared Error (RMSE)

RMSE [15] is the standard way to measure error by differences between the actual points and the predicted points. These measured differences between these values are called residuals (Predicted error), where the residuals measures how far are these data points are from the regression line, these differences are calculated ones then squared and then the average of the squares and finally calculates the root. The purpose of squaring the value is to convert the negative values into positive values when the root means square results zero value that implies no error or perfect indication.

Furthermore, Root means the square error is commonly used in regression analysis, forecasting to verify the experimental results and the model performance. The RMSE is generalized mathematical equation is as follows

$$RMSE = \sqrt{\frac{\sum_{i=1}^{n}(\overline{y_i} - y_i)^2}{n}}$$

Mathematically for forecasting equation for RMSE is as follow:

$$RMSE = \sqrt{\overline{(f\ -o)^2}}$$

Where
f=forecast (expected values)
o=observed values (known results)

10.2 Mean Absolute Percentage Error (MAPE)

The MAPE [8] is commonly used for measuring the accuracy of the forecasting model, it performs an average of absolute percentage differences between the forecasted and actual values. Firstly it takes the actual values, Secondly takes the differences between the real and predicted, Thirdly the ratio of errors against the actual values, lastly the mean of these values, moreover it is also the most commonly used metrics for measuring the forecast error. Mathematically MAPE is represented as follows: [19]

$$MAPE = (\frac{1}{n}\sum\frac{|Actual - forecast|}{Actual}) \times 100$$

Table 2: Death Cases

Model	Region	Germany	Italy	US	UK	France	Spain	Brazil	Belgium	Iran	Mexico
ARIMA	RMSE	150.00	267.266	641.896	1146.85	1890.635	258.86	263.144	154.11	72.54	106.99
	MAPE	3.211	1.591	3.728	4.138	5.344	2.214	13.0111	2.821	3.76	13.239
SARIMA	RMSE	59.29	77.14	403.229	177.37	166.66	108.54	144.55	57.56	12.02	65.26
	MAPE	8.68	5.90	16.47	11.8	6.17	5.41	52.338	7.91	7.00	54.34

Table 3: Confirmed Cases

Model	Region	Germany	Italy	US	UK	France	Spain	Brazil	Belgium	Iran	Mexico
ARIMA	RMSE	665.05	209.86	1133.16	1889.62	1802.32	879.69	489.99	185.707	948.89	645.51
	MAPE	0.71	0.26	0.78	4.07	3.09	0.53	0.17	0.94	3.64	7.25
SARIMA	RMSE	332.1	250.03	2927.37	994.2	1238.90	713.97	2245.744	126.9	211.19	254.47
	MAPE	3.75	4.12	15.1	15.17	2.93	3.63	54.84	5.95	11.74	47.46

Table 4: Recovered cases

Model	Region	Germany	Italy	US	UK	France	Spain	Brazil	Belgium	Iran	Mexico
ARIMA	RMSE	693.7772	1283.145	1512.25	660.06	1663.11	563.60	1063.11	125.01	504.44	281.87
	MAPE	1.65	2.17	0.51	3.91	2.23	1.06	0.94	1.23	2.24	4.59
SARIMA	RMSE	820.6	1442.7	7186.85	8.58	390.1	1446.36	1362.95	83.8	202.03	839.8
	MAPE	9.76	52.8	30.4	9.8	10.13	13.02	50.26	10.36	10.09	54.32

11 CONCLUSION:

In this project, the forecasting methods ARIMA and SARIMA was implemented on the given John Hopkins COVID 19 data to forecast the number of deaths, recovered, and confirmed cases of top 10 countries. The performance of the two models was evaluated by using the results of RMSE and MAPE performance metrics. Based on the results of the performance metrics, it can be observed that, both the models gave a good prediction of cases in most of the time series.

REFERENCES

[1] [n.d.]. 2004.00958.pdf. https://arxiv.org/ftp/arxiv/papers/2004/2004.00958.pdf. (Accessed on 05/13/2020).

[2] [n.d.]. 2.8 Autocorrelation | Forecasting: Principles and Practice. https://otexts.com/fpp2/autocorrelation.html. (Accessed on 06/24/2020).

[3] [n.d.]. Confidence Intervals. https://sphweb.bumc.bu.edu/otlt/MPH-Modules/BS/BS704_Confidence_Intervals/BS704_Confidence_Intervals_print.html. (Accessed on 06/24/2020).

[4] [n.d.]. COVID-19 Dataset | Kaggle. https://www.kaggle.com/imdevskp/corona-virus-report/data. (Accessed on 06/06/2020).

[5] [n.d.]. COVID-19 Map - Johns Hopkins Coronavirus Resource Center. https://coronavirus.jhu.edu/map.html. (Accessed on 06/06/2020).

[6] [n.d.]. Forecasting of COVID-19 Cases and Deaths Using ARIMA Models | medRxiv. https://www.medrxiv.org/content/10.1101/2020.04.17.20069237v1. (Accessed on 05/14/2020).

[7] [n.d.]. Forecasting: Principles and Practice. https://otexts.com/fpp2/. (Accessed on 06/06/2020).

[8] [n.d.]. A Gentle Introduction to Backtesting for Evaluating the Prophet Forecasting Models | by Kan Nishida | learn data science. https://blog.exploratory.io/a-gentle-introduction-to-backtesting-for-evaluating-the-prophet-forecasting-/models-66c132adc37c. (Accessed on 07/18/2020).

[9] [n.d.]. A Gentle Introduction to SARIMA for Time Series Forecasting in Python. https://machinelearningmastery.com/sarima-for-time-series-forecasting-in-python/. (Accessed on 06/24/2020).

[10] [n.d.]. How to Calculate the KL Divergence for Machine Learning. https://machinelearningmastery.com/divergence-between-probability-distributions/. (Accessed on 06/24/2020).

[11] [n.d.]. An Introduction To Non Stationary Time Series In Python. https://www.analyticsvidhya.com/blog/2018/09/non-stationary-time-series-python/. (Accessed on 06/03/2020).

[12] [n.d.]. Introduction to Time Series Analysis for Organizational Research: Methods for Longitudinal Analyses - Andrew T. Jebb, Louis Tay, 2017. https://journals.sagepub.com/doi/abs/10.1177/1094428116668035. (Accessed on 06/06/2020).

[13] [n.d.]. (PDF) Spillover of COVID-19: impact on the Global Economy. https://www.researchgate.net/publication/340236487_Spillover_of_COVID-19_impact_on_the_Global_Economy. (Accessed on 05/14/2020).

[14] [n.d.]. Real-time forecasts of the COVID-19 epidemic in China from February 5th to February 24th, 2020 - ScienceDirect. https://www.sciencedirect.com/science/article/pii/S2468042720300051. (Accessed on 05/14/2020).

[15] [n.d.]. Root-mean-square deviation - Wikipedia. https://en.wikipedia.org/wiki/Root-mean-square_deviation. (Accessed on 07/18/2020).

[16] [n.d.]. seabold.pdf. https://conference.scipy.org/proceedings/scipy2010/pdfs/seabold.pdf. (Accessed on 06/24/2020).

[17] [n.d.]. Significance of ACF and PACF Plots In Time Series Analysis. https://towardsdatascience.com/significance-of-acf-and-pacf-plots-in-time-series-analysis-2fa11a5d10a8. (Accessed on 06/24/2020).

[18] [n.d.]. Trend, Seasonality, Moving Average, Auto Regressive Model : My Journey to Time Series Data with Interactive Code. https://towardsdatascience.com/trend-seasonality-moving-average-auto-regressive-model-my-journey-to-time-. (Accessed on 06/06/2020).

[19] [n.d.]. Welcome to Forecast Pro - Software for sales forecasting, inventory planning, demand planning, S&OP and collaborative planning. https://www.forecastpro.com/Trends/forecasting101August2011.html. (Accessed on 07/18/2020).

[20] [n.d.]. What Is Coronavirus? | Johns Hopkins Medicine. https://www.hopkinsmedicine.org/health/conditions-and-diseases/coronavirus. (Accessed on 05/14/2020).

[21] [n.d.]. What is data visualization? A definition, examples, and resources. https://www.tableau.com/learn/articles/data-visualization. (Accessed on 06/06/2020).

[22] [n.d.]. WHO | World Health Organization. https://www.who.int/. (Accessed on 05/14/2020).

[23] Jason Brownlee. 2017. How to create an ARIMA model for time series forecasting in Python. *Machine Learning Mastery. Saatavissa: https://machinelearningmastery.com/arima-for-time-series-forecasting-with-python/. Hakupäivä* 2 (2017), 2019.

[24] Nouby M Ghazaly, Muhammad A Abdel-Fattah, and AA Abd El-Aziz. [n.d.]. Novel Coronavirus Forecasting Model using Nonlinear Autoregressive Artificial Neural Network. ([n. d.]).

[25] Ramesh Medar, Vijay S Rajpurohit, and B Rashmi. 2017. Impact of Training and Testing Data Splits on Accuracy of Time Series Forecasting in Machine

Learning. In *2017 International Conference on Computing, Communication, Control and Automation (ICCUBEA)*. IEEE, 1–6.

[26] Robert Nau. 2015. Statistical forecasting: notes on regression and time series analysis. *Notes and materials for an advanced elective course on statistical forecasting that is taught at the Fuqua School of Business, Duke University* (2015).