

COVID-19 DATA ANALYSIS

**Final Report**

Word Count: 4347

Higher Diploma in Science in Data Analytics

NICHOLAS UDOMBOSO

10547333@mydbs.ie

10/09/2021

Acknowledgements

First and foremost, I would like to acknowledge the support I have received from my Supervisor Dr. Shazia A. Afzal in providing me the professional advice and mentorship needed to submit this report.

# Contents

[Contents 1](#_Toc81369503)

[1. Introduction 2](#_Toc81369504)

[2. Background 2](#_Toc81369505)

[3. Literature Review 3](#_Toc81369506)

[4. Requirements Specification and Design 4](#_Toc81369507)

[5. Testing and Evaluation 18](#_Toc81369508)

[6. Conclusion 26](#_Toc81369509)

[7. Future Work 27](#_Toc81369510)

[References / Bibliography 27](#_Toc81369511)

[Appendices 29](#_Toc81369512)

# Introduction

Covid-19 virus is a global pandemic (WHO, 2020). It continues to spread. Many countries have turned to data to help them fight the virus as it spreads. As Carto (2020) puts it, ‘using COVID-19 data to fight & contain the pandemic with advanced analytics is critical to protect public health & save lives.’ This project is contributing to the efforts made so far to see the end of the disease.

* 1. What is this project trying to solve? This project aims to:
* Provide reality check of the current spread of the pandemic via dashboard visualization.
* Develop models to forecast future trend of the disease which could in turn support policies to strengthen effective control measures.
* Develop face mask detection system that could be deployed in crowded public places to ensure safety and control the spread of the disease.
  1. Scope of the project:
* To analyse Johns Hopkins University Covid-19, and World Data Bank datasets.
* To visualize the spread of the virus.
* To build timeseries forecasting models with the datasets
* To evaluate the accuracy of the developed forecasting models.
* To build convolution neural network (CNN) model for facemask detection with dataset from Kaggle to support the disease control.
  1. Implementation approach of the project:
* Download Johns Hopkins University Covid-19 dataset from GITHUB.
* Perform exploratory data analysis with Python on the dataset to understand, transform and prepare the data for analysis, modelling and forecasting.
* Build Dashboard visualization to depict the spread.
* Build Timeseries forecasting models with Holt’s Winters, Arima (Autoregressive integrated moving average), Sarima (Seasonal autoregressive integrated moving average), Linear Regression, Random Forest, and Facebook Prophet to compare performance of the models in forecasting covid-19 pandemic via error metrics assessment of the models.
* Forecast next 3-months (i.e., August 30 to November 29, 2021) of covid-19 pandemic using the best performed model.
* Build facemask detection model as a measure to support the control of spread of the virus with dataset from Kaggle.

# Background

Millions of people have been infected with Covid19 and lost their lives from it. It spreads when people are in proximity of about 1 metre range with each other (WHO, 2020). Crowded places encourage spread (WHO, 2020). Touching the eyes, mouth, or nose after touching contaminated surfaces without washing the hands is also getting people infected (WHO, 2020).

It is in public knowledge that countries across the globe have pursued measures like social distancing, face mask wearing, social gathering and business centres closures, travel restrictions, and most recently vaccinations among others to stop or lower the rate of spread of the disease. Efforts are continually being made globally to contain the pandemic and recover from its upsetting impacts. In the bid to revive the economy and social life many Governments have eased some control measures. This warrants continual efforts to be made to monitor the prevalent situation of the disease. I am analysing the Covid-19 publicly available dataset about the Novel Coronavirus provided by Johns Hopkins University to create visualizations, build forecast models and implement facemask detection system with dataset from Kaggle to support the control of spread of the virus.

2.1. Context of the project:

The results and inferences made are based on the datasets in use for this project.

2.2. The anticipated benefits of the system:

This is meant to inform the users of the system and the public the current prevalence of the covid-19 spread, forecast the future trend of the virus and to encourage the wearing of facemask in public places to ensure safety.

2.3. Typical users of the project product:

The Public

2.4. Used any relevant existing software/hardware?

Jupyter notebook, Google Colab, Tableau, Python libraries e.g., Scikit learn, Pandas, Keras API, Draw IO etc.

# Literature Review

Several literatures have reviewed different approaches that have been used in different countries to model the prevalence of the disease. Below are few examples:

Anastassopoulou projected Covid19 to end or slowdown in February 2020 in China with SIRD (susceptible, infected, recovered, and dead) model (Katoch et al, 2021). However, most early predictions had challenges for lack of enough data (Liu et al,2020). Kumar used Arima model to report that the spread would worsen in Iran and Europe (Katoch et al, 2021). Using linear and exponential growth models, Strohman (2020) reported projection in the number of cases that would demand intensive care unit (ICU) admission in Italy estimating it to 14,542 by March 2020. This helped to make Governments to start acting and allocate resources.

Ceylan (2020) formulated several timeseries models with ARIMA for Italy, Spain, and France with mean absolute percentage error (MAPE) values of 4.7520, 5.8486, and 5.6335 respectively.

Many countries have used Arima and several other models to predict the pandemic and continuously seek to improve performances of the models for better predictions. Following this, my strategy in this project is to implement various models and evaluate their performances. Then choose the best performed model to project the prevalence of the pandemic in Ireland by November 2021.

Previous predictions have helped governments to plan for resource allocations and distributions to fight and control the virus.

# Requirements Specification and Design

Below are the systems to be developed in this project:

I). System 1: Dashboard Visualization

II). System 2: Timeseries Forecast on Covid-19

III). System 3: Facemask detection.

Details below:

4.1. System 1: Dashboard Visualization

Purpose/specification of the System:

To provide visualization on the spread of covid19 pandemic which is a reality check on the prevalence of the disease to the public and the policy makers.

Work Roadmap:

Data acquisition from John Hopkins and World Data Bank -> Data Pre-processing and Exploration using Jupyter Notebook -> Dashboard Visualization

4.1.1. Data Acquisition:

I acquired data from Johns Hopkins University Center for Systems Science and Engineering (JHU CSSE), and the World Bank. JHU CSSE has a publicly hosted timeseries dataset on Github for Coronavirus. The data is updated daily and freely accessible to the public. It is a cumulative data on confirmed, death and recovered cases. The World Bank has the year 2020 country wise population data which I downloaded from the World Bank’s site.

4.1.2. Data Pre-processing and Exploration:

The data from JHU CSSE was not in a ready format for analysis. So, I went through the following steps to pre-process the data for visual analysis.

i). I imported the wget library on python and downloaded the respective CSV files from the urls the data are hosted. Exception to this method was the case of population data that I downloaded the CSV file manually from the World Bank site. Consequently, the data came in separate CSV files. So, it became inevitable to merge them to make it easy for visualization and analysis.

ii). Dates in the files are provided as column names. So, I changed the data shape from having the dates in column form to row form.

iii). Cleaned up missing values, corrected datatypes and took off cases from ships in the dataset.

iv). Introduced Active case column by subtracting deaths and recovered cases from confirmed cases.

v). Grouped the data by date and country fields to perform some aggregation for each group.

vi). Introduced new cases, new deaths, and new recoveries columns to capture daily figures.

4.1.3. Data Visualization:

I imported the formatted data into tableau. Below is the snapshot of the developed dashboard.

Graphical user interface, chart

Description automatically generated

Figure.1: Covid-19 cases dashboard.

The figure 1 above shows a dashboard of global confirmed cases of 379.9M, confirmed deaths of 7.44M, mortality rate of 1.96% and recovery rate of 38.52% in August. The dashboard can be filtered for any country, Ireland inclusive to understand the situation of the virus. See the below snapshot for Ireland in Figure 2.

Graphical user interface

Description automatically generated with medium confidence

Figure.2: Covid-19 cases dashboard for Ireland.

Ireland is not among the top ten countries impacted by the virus but in August had registered cumulatively 344,218 cases at 1.48% and 6.79% overall mortality and recovery rates from inception respectively.

The country had high mortality rates in April 2020 and January 2021. Decrease in trend has been observed since February 2021 but July and August are again showing the cases rising.

Graphical user interface, application

Description automatically generated

Figure.3: Highest number of confirmed cases in Ireland.

Ireland’s highest recorded cases stood at 104,768 in January 2021.

I published my dashboard to Tableau public. Please go to the reference section for the link to view the dashboard.

4.2. System 2: Timeseries Forecast on Covid-19

Purpose/specification of the System:

To develop a model to forecast 3-months trend of Covid-19 which could in turn support policies for effective control measures.

Work Roadmap:

Data acquisition from John Hopkins, and World Data Bank -> Data Pre-processing and Exploration using Jupyter Notebook -> Build the models -> Timeseries Forecasting

Steps to building the Covid-19 Forecasting model:

4.2.1. Data Acquisition:

I imported into Python the file ‘COVID-19-time-series-clean-data.csv’ which I saved after combining and cleaning up the JHU CSSE and World databank datasets on Jupyter notebook. Below is the snapshot of the cleaned data.

Graphical user interface, text, application

Description automatically generated

Figure.4: Snapshot of the dataset

4.2.2. Data Pre-processing and Exploration:

I performed further pre-processing to explore seasonal patterns and features in the data.

Below is a snapshot of more features extracted from the data.

Graphical user interface, application

Description automatically generated

Figure.5: snapshot of more features extracted in the dataset.

Figure 6 below illustrates the growing number of confirmed cases in Ireland.

Chart, line chart

Description automatically generated

Figure.6: Confirmed cases in Ireland from Jan.2020 to Aug.2021.

In figure 6 above an upward trend of confirmed cases is observed in the data for Ireland.

The below figure 7 shows a pattern repeating 4 times within a month. This suggests that the data has weekly seasonality.

Chart, line chart

Description automatically generated

Figure.7: weekly seasonality observed.

Figure 8 shows yearly confirmed covid cases and cases is observed to have increased in 2021 compared to 2020.

Chart, box and whisker chart

Description automatically generated

Figure.8: Confirmed cases yearly.

Chart

Description automatically generated

Figure.9: Monthly and quarterly trend.

Overall trend shows monthly and quarterly increasing trend in figure 9.

4.2.3. Building Time Series Forecasting Models on the data

‘Time series forecasting is the use of a model to predict future values based on previously observed values’ (Li, 2018). It is widely applied to predict weather, sales, stock price, business future etc. (Masui, 2021). In this project, I am modelling covid-19 cases with different algorithms to adopt the most suitable model based on performance to forecast the next 3 months of covid-19 trend in Ireland (i.e., August 30 to November 29, 2021).

I split the data into training and test sets to enable me to build the models with the training set and validate with the test set before performing the forecast.

Below is the structure of the datasets before and after splitting:

a) Complete dataset: 22-01-2020 to 29-08-2021 (contains actual data values of the entire dataset).

b) Train dataset: 22-01-2020 to 28-05-2021 (contains actual data values). I trained the models on this dataset.

c) Test dataset: 29-05-2021 to 29-08-2021 (contains actual data values). I compared the actual values of this dataset against the predicted values of the dataset, then finding the error metrics to validate performance of each models used.

On the 3-months forecasts, the date range is from 30-08-2021 to 29-11-2021.

4.2.3.1. Holt-Winters (aka triple exponential smoothing) Algorithm

Holt-Winters makes use of exponential smoothing to compute historical values and predict current and future values (or average) (Solarwind, 2019). Exponential smoothing is when we assign weights to observations in an exponentially increasing manner from oldest to newest observations (Stephanie, 2018). Holt-Winters considers value, trend, and seasonality as the three types of exponential smoothing, and hence, it is a triple exponential smoothing model.

I fit this model on the training set of my data as shown in figure 10.

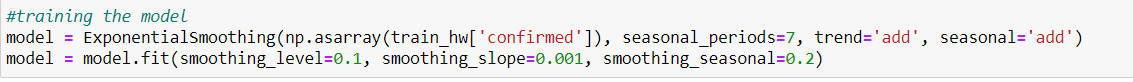


Figure.10: holt winter model training.

Below figure 11 is the plot showing the trend of the test(actual) data and the predicted values of the model.

Chart

Description automatically generated

Figure.11: Holt Winter model plot

Holt-Winter plot shows a linear trend for both the predicted values and test data. The plot also shows that the predicted values are very close to the validation data

4.2.3.2. Arima

ARIMA is a widely accepted model for timeseries forecasting. The model uses autoregression parameter p to represent the series past values, moving average parameter q to represent errors of past forecasting and parameter d to denote the order of difference taken when building its model for timeseries. Right values for the parameters p,d,q is required to build the most appropriate model for a timeseries. I used auto\_arima functionality in python’s pmdarima library to find the values automatically. The functionality auto tuned the parameters and suggested ARIMA(3,2,2) model as shown.

Graphical user interface, text, application

Description automatically generated

Figure.12: Arima model training

Table

Description automatically generated

Figure.13: Arima model

Figure 14 is the plot showing the trend of the test(actual) data and the predicted values of Arima model. Both test and predicted data are close.

Chart, line chart

Description automatically generated

Figure.14: Arima model plot

4.2.3.3. Sarima

SARIMA is Seasonal Arima. It has seasonal parameters (P, D, Q) in addition to Arima’s parameters. P is for seasonal autocorrelation, D for seasonal difference, and Q for seasonal moving average.

A snapshot below shows the model via auto\_arima functionality searching for the best parameter combinations to suggest the most suitable model for the series.

Text

Description automatically generated

….

Text, letter

Description automatically generated

Figure.15: Sarima model

The output suggests SARIMA (3,2,2) (0,0,0,7) model. Fitting the model on the timeseries

Chart, line chart

Description automatically generated

Figure.16: Sarima model plot

shows in figure 16 that the predicted values are also close to the actual values and the trend continues upward.

4.2.3.4. Linear Regression

Linear regression is designed as a statistical tool that can help predict future values from past historical values of a target variable (CHRON, 2020). In applying to my data, I extracted some valuable features from the data and used those features for the model. So, this can be regarded as a feature-based model. Below is a snapshot of the features.

A screenshot of a computer

Description automatically generated with medium confidence

Figure.17: snapshot of complete data.

Next, created x-train and y-train to separate the features and the target variable. Then used the x-train and y-train for model training and then making predictions on the test data.

Chart, line chart

Description automatically generated

Figure.18: Linear regression plot

Linear regression captures increasing trend as other models but also shows decreasing trend at some points.

4.2.3.5. Random Forest

A random forest is a technique for solving regression and classification tasks (Wikipedia). It uses ensemble learning, a method that utilizes several classifiers to solve complex tasks (Wikipedia).

I trained the model on the x-train and y-train data used earlier for linear regression model.

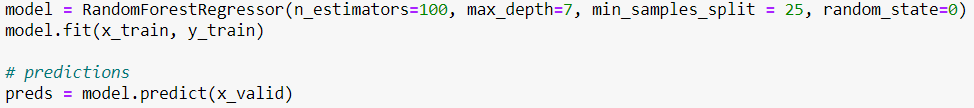


Figure.19: Random Forest model training

Chart

Description automatically generated

Figure.20: Random Forest plot

Random Forest is clearly not performing well on the data as shown in figure 20. It is unable to predict beyond its observation in the train data as seen on the plot above.

4.2.3.6. Facebook Prophet

Prophet Model for timeseries forecasting was introduced by Facebook in 2017(Masui, 2021). The algorithm considers: Trend, Seasonality, and Holiday. That is, Forecast = Trend + Seasonality + Holiday (Masui, 2021). In applying the model to the data, prophet was installed. Then, transformed the data to the specified format for prophet which is making the dataset to have two columns, ds and y. Column ds is for date and column y for the target variable. Next, trained the model on the data and made predictions with the trained data.

Text

Description automatically generated

Figure.21: Snapshot showing Prophet model column format.

Below in figure 22 visualizes predictions from the prophet model.

Chart, line chart

Description automatically generated

Figure.22: Prophet model plot(a)

The black dotted line represents the training data. The dark blue lines are the predicted values. The light blue line represents the trend. Hence the model is predicting that the confirmed cases is still rising exponentially.

Chart, line chart

Description automatically generated

Figure.23: Prophet model plot(b)

Figure 23 shows how the predicted and validation data of the model are close.

4.2.4. Three months forecast of Covid19

Based on the performances of the models above (please refer to section 5 to see the test and evaluation of the models), I used Arima and Prophet models to forecast the covid19 data for the next 3 months (Aug. 30 to 29 Nov. 2021).

4.2.4.1. Forecasting Covid19 for next 3months with Arima

As shown in below plot in figure 15, Arima model forecasts that covid19 virus cases will keep rising and approach approximately 380,000 confirmed cases in Ireland by November 2021.

Chart, line chart

Description automatically generated

Figure.24: Forecast with Arima

4.2.4.2. Forecasting Covid19 for next 3months with Prophet

Facebook prophet also forecast that the cases would rise above 350,000 by November 2021 as shown in the plot below in figure 16. It goes further to forecast the upper and lower ranges of the cases, that it could approximately fall to 93,000 or rise to 650,000 by November, 2021. Please see the below plot.

Chart, line chart

Description automatically generated

Figure.25: Forecast with Facebook prophet

From my observation the forecasts by the two models Arima, and Prophet are within the same range of values.

4.3. System 3: Facemask detection system

Purpose/specification of the System:

To implement a solution to encourage a more effective use of facemask in public spaces for effective control of the virus spread. This is a solution in high demand in crowded areas like malls, bus stations, in transportations, residential districts, enterprises etc. to promote safety.

Technical requirements:

These include Google Colab and imported libraries

Workflow:

Data from Kaggle -> Image Visualization -> Data Augmentation-> Build Convolutional Neural Network -> Compile & Train model -> Performance Evaluation & Test model -> Predict static and images with the model

4.3.1. Data Acquisition:

I acquired the data from Kaggle. The data consist of images of faces with mask and without mask.

Graphical user interface, website

Description automatically generated

Figure.26: Covid face mask detection dataset

Below snapshot shows the train, validation, and test sets on my google drive uploaded on Colab.

A picture containing graphical user interface

Description automatically generated

Figure.27: Snapshot of Train, Validation and Test datasets for facemask detection system.

Train set is used to train the model. The validation data is separate from the training data which is used to validate the model during training. With every epoch during training, The model trains on the training set and concurrently validates the data using the validation set with each epoch during training.

Validation is for preventing overfitting the train data, i.e., avoiding inability of the model to generalise data it has never seen. So, while training and validating, if the result for the validations is as good as the results for training data, it can be asserted that the model is not overfitting.

Test set helps to test the model after training and validation. The model will be used to predict the output of the data in the test set.

4.3.2. Data Augmentation:

This process creates new data based on modifications of existing data, i.e., the training set.

Why is augmentation needed? First, to help add more modified data to the training set. Second, to reduce overfitting.

Image data augmentation is supported in the keras deep learning library via ImageGenerator class.

So, I augmented the data with some zooming, rotation and pixel rescaling to normalize them for processing on convolutional neural network.

4.3.3. CNN Model Building:

Figure 28 is a diagram to show CNN model building flow.

Diagram

Description automatically generated

Figure. 28: CNN model building flow (Phung & Rhee, 2019).

Convolutional neural network (CNN) is a well-known artificial neural network used for analyzing images. It detects patterns in images and makes meaning of them. CNN has input layer, hidden layer, and output layer. Hidden layers are convolutional layers, which are the basis of CNN.

Below is the architecture of the model:

**INPUT** -> **Convolution** **Block1** (Convolution->RELU->MaxPooling->Dropout) ->

**Convolution** **Block2** (Convolution>RELU->MaxPooling->Dropout-Flatten)->

**FC1** (Dense->RELU->Dropout)-> **Output** (Dense->Sigmoid)

To summarize the architecture shown above, images are fed into the network as input. Feature extraction is performed in the convolution blocks (1 &2). The image is then down sampled, flattened up and made to pass through the fully connected layer (FC1) and a dense layer (Phung & Rhee, 2019).

4.3.4. Predicting images with the model:

At this point new sets of images to determine if a person is wearing mask or not can be used. The below shows successful prediction of two images that were uploaded. The model successfully labelled image with no mask and the image with mask on. Please see the snapshot below in figure 29.

Graphical user interface, text, application

Description automatically generated

Figure.29: Snapshot of labelled images as mask and nomask.

# Testing and Evaluation

To ensure good performance of the models built, it is necessary to test and evaluate their performances. First, tested the timeseries forecast models, then the CNN model built for facemask detection.

5.1. Testing Timeseries Forecast models:

I used error metrics, MAPE (mean absolute percentage error) and RMSLE (root mean squared logarithmic error) to evaluate the performance of the timeseries forecast models.

After splitting the data to train and test sets, I trained Holt-Winters, Arima, Sarima, Linear Regression, Random forests, and Facebook prophet models on the train set. Prediction was made based on the trained data and validated against the test data. Below show performances of each of the models.

5.1.1. Holt-Winters performance

Holt-Winter gave RMSLE value of 4.25 and MAPE value of 2.996 as shown.

Graphical user interface, text, application, email

Description automatically generated

Figure.30: Snapshot of Holt-Winter error metric.

Further tuning of the model parameters slightly reduced RMSLE to 4.196 but no improvement to MAPE, value 3.03. See figure 31 below.

Graphical user interface, text, application

Description automatically generated

Figure.31: Snapshot of Holt-Winter error metric after parameter tuning.

5.1.2. Arima performance

Arima model assessment shows the model performed very well with a low RMSLE value of 4.065 and MAPE 2.79 as shown.

Graphical user interface, text, application

Description automatically generated

Figure.32: Snapshot of Arima error metric.

5.1.3. Sarima performance

Sarima showed a very performance with the RMSLE of 4.623 and MAPE of 3.05. See figure 33.

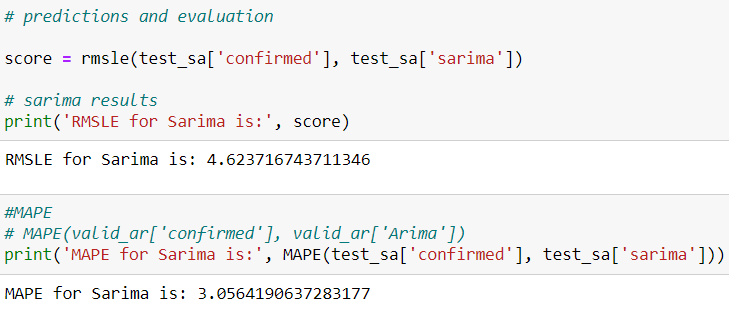


Figure.33: Snapshot of Sarima error metric.

5.1.4. Linear Regression performance

On assessing Linear regression, it resulted in RMSLE of 13.548 and MAPE of 11.73 for the regression as in figure 34.

Graphical user interface, text, application, email

Description automatically generated

Figure.34: Snapshot of Linear regression error metric.

Linear regression model did not show a very good performance on the data compared to the previous models.

Looking at the feature importance given by the model, month variable shows the highest coefficient value. In other words, the feature was the most relevant feature for giving the prediction. This is shown in figure 35 below.

Chart

Description automatically generated

Figure.35: Feature importance of Linear regression

5.1.5. Random Forest performance

Random forest model did not also show very good performance on the test data as can be seen in figure 36.

Graphical user interface, text, application, email

Description automatically generated

Figure.36: Snapshot of Random Forest error metric.

Here, a score of 15.495 for RMSLE value and 11.73 for MAPE was achieved. Looking at feature importance by the model in figure 37, shows year variable as the most useful feature in giving the predictions.

Graphical user interface

Description automatically generated

Figure.37: Feature importance of Random Forest

5.1.6. Facebook Prophet performance

The low RMSLE value of 4.889 and MAPE of 3.30 shows the model performed well on the data with minimal error. See the below figure 38.

Graphical user interface, text, application, email

Description automatically generated

Figure.38: Prophet model error metric

5.1.7. Comparing performances of the models

The table below table summarises and compares the error metrics obtained from the models based on the data.

Table

Description automatically generated

Figure.39: Error metric summary of the models

Except for Linear regression and Random Forest, the models’ performance values are within the same range. Prophet model performed best among the ML models (Linear regression, Random Forest, Prophet) but did not perform better than the classical models of Holt-Winters, Arima and Sarima on this data. However, their performances are close. Hence, on this basis I forecasted covid-19 timeseries for Ireland with Arima and Prophet.

5.2. Testing Face Mask Detection Model:

5.2.1. Evaluating performance of the model:

After training the model, I achieved 94% training accuracy and 93% validation accuracy which is a very good performance for the model. Please see the below figure 40.

A picture containing calendar

Description automatically generated………

A picture containing diagram

Description automatically generated

Figure.40: Face Mask Detection Model performance.

Below plot in figure 41 shows training and validation loss going down.

A picture containing graphical user interface

Description automatically generated

Figure.41: Training and validation loss of CNN.

Below plot in figure 42 shows 94% training accuracy and 93% validation accuracy achieved. Not much gap between them.

Chart

Description automatically generated

Figure.42: Training and validation accuracy of CNN.

# Conclusion

The covid19 dashboard built on this project shows that the virus is still spreading and have a growing trend globally and in Ireland. This makes it important to continuously study the prevalence of the pandemic and continue to forecast future trend of the disease for insights into how the health systems and policies need to be continuously strengthened and resources reallocated.

In this project I have been able to forecast covid19 pandemic trend from August 30 to November 29, 2021, using Arima and Prophet models with error metrics of 2.79 and 3.3 MAPE values respectively. This is a very good performance from the models. From the results, by end of November 2021 we can say that confirmed cases in Ireland would either be as high as 650,000 cases or see a decline as low as 93,000 cases.

Today timeseries forecasting is also applied in many applications like sales, weather, economic, healthcare, and business forecasts, etc. (Tableau).

To promote safety and support government initiatives to slow the spread of the disease, facemask detection model has also been implemented in this project to detect people wearing or not wearing mask. This is a measure to encourage mask wearing in public places. The model achieved 94% performance accuracy, a very significant performance.

# Future Work

While the project has explored many timeseries forecast models to forecast the spread of covid19 and implemented a facemask detection model to support the control of the disease spread, this work can be expanded further to more forecast models or have the models in use here optimized further to achieve possibly a better performance.

To deploy the facemask detection system in public places further work would be required to integrate the algorithm to smart devices e.g., an alarm is initiated if someone is not putting on a facemask. In addition, the algorithm can be further expanded to detect non-static images in cameras. However, for this project an algorithm has been implemented to be able to detect images wearing facemask and those without the mask.

# References / Bibliography

Agarwal,P. and Jhajharia, K. (2021). ‘Data analysis and modelling of COVID-19’, Journal of Statistics and Management Systems, Volume 24, 2021 - Issue 1: Prediction Analysis of COVID-19 using Data Science (Part-I). Available at: <https://www.tandfonline.com/doi/abs/10.1080/09720510.2020.1840076> (Accessed: 18 June 2021).

ANALYTICS VIDHYA (2018). ‘24 Ultimate Data Science (Machine Learning) Projects To Boost Your Knowledge and Skills (& can be accessed freely)’. Available at: https://www.analyticsvidhya.com/blog/2018/05/24-ultimate-data-science-projects-to-boost-your-knowledge-and-skills/ (Accessed: 18 June 2021).

Brownlee, J. How Do Convolutional Layers Work in Deep Learning Neural Networks? Available at: https://machinelearningmastery.com/convolutional-layers-for-deep-learning-neural-networks/ (Accessed: 15 August 2021).

Brownlee, J., (2020). ‘Time Series Forecasting With Prophet in Python’ Available at: https://machinelearningmastery.com/time-series-forecasting-with-prophet-in-python/

(Accessed 20 July 2021).

Carto., (2020) COVID-19 Data Analytics. Available at: <https://carto.com/covid-19/> (Accessed: 20 June 2021).

Ceylan, Z. (2020). Estimation of COVID-19 prevalence in Italy, Spain, and France. Available at: <https://www.sciencedirect.com/science/article/abs/pii/S0048969720323342> (Accessed: 15 August 2021).

Data Science Weekly. ‘How To Choose a Data Science Project For Your Data Science Portfolio’. Available at: https://www.datascienceweekly.org/articles/how-to-choose-a-data-science-project-for-your-data-science-portfolio (Accessed: 18 June 2021).

Elite Data Science. ‘8 Fun Machine Learning Projects for Beginners’. Available at: https://elitedatascience.com/machine-learning-projects-for-beginners (Accessed: 18 June 2021).

Github. (2021). COVID-19 Data Repository by the Center for Systems Science and Engineering (CSSE) at Johns Hopkins University. Available at: <https://github.com/CSSEGISandData/COVID-19> (Accessed: 29 August 2021).

K, G. (2020). COVID-19: Face Mask Detection using TensorFlow and OpenCV. Available at: <https://towardsdatascience.com/covid-19-face-mask-detection-using-tensorflow-and-opencv-702dd833515b> (Accessed: 28 July 2020).

Katoch, R and Sidhu, A. (2021). ‘An Application of ARIMA Model to Forecast the Dynamics of COVID-19 Epidemic in India’, Sage Journals. Available at: <https://journals.sagepub.com/doi/full/10.1177/0972150920988653> (Accessed: 19 June 2021).

Li, S. (2018). An End-to-End Project on Time Series Analysis and Forecasting with Python. Available at: <https://towardsdatascience.com/an-end-to-end-project-on-time-series-analysis-and-forecasting-with-python-4835e6bf050b> (Accessed: 24 August 2021).

Liu, L., Chen, Y., Wu, L. (2020). Forecasting Confirmed Cases, Deaths, and Recoveries from COVID-19 in China during the Early Stage. Available at: <https://www.hindawi.com/journals/mpe/2020/1405764/> (Accessed: 15 July, 2021).

Masui, T. (2021). Multi-step Time Series Forecasting with ARIMA, LightGBM, and Prophet. Available at: <https://towardsdatascience.com/multi-step-time-series-forecasting-with-arima-lightgbm-and-prophet-cc9e3f95dfb0> (Accessed: 24 August 2021).

Mitra, P. (2020). COVID Face Mask Detection Dataset. Available at: <https://www.kaggle.com/prithwirajmitra/covid-face-mask-detection-dataset> (Accessed: 5 August 2021)

Phung, & Rhee. (2019). A High-Accuracy Model Average Ensemble of Convolutional Neural Networks for Classification of Cloud Image Patches on Small Datasets. Available at: https://www.mdpi.com/2076-3417/9/21/4500/htm (Accessed: 12 August 2021).

Press G., (2016). ‘Cleaning Big Data: Most Time-Consuming, Least Enjoyable Data Science Task, Survey Says’. Available at: <https://www.forbes.com/sites/gilpress/2016/03/23/data-preparation-most-time-consuming-least-enjoyable-data-science-task-survey-says/?sh=447890b46f63> (Accessed 21 June 2021).

Rosebruck, A., (2020). ‘COVID-19: Face Mask Detector with OpenCV, Keras/TensorFlow, and Deep Learning‘ Available at: <https://www.pyimagesearch.com/2020/05/04/covid-19-face-mask-detector-with-opencv-keras-tensorflow-and-deep-learning/> (Accessed 25 July 2021).

Satrio, C., Darmawan, W., Nadia, B. and Hanafiah, N. (2020). Time series analysis and forecasting of coronavirus disease in Indonesia using ARIMA model and PROPHET. ScienceDirect.

Solarwinds. (2019). Holt-Winters Forecasting Simplified. Available at: <https://orangematter.solarwinds.com/2019/12/15/holt-winters-forecasting-simplified/>

(Accessed: 29 August 2021).

Strohman, A. (2020). What Italy Can Teach Us About Reacting to the Coronavirus. Available at: <https://www.americanactionforum.org/weekly-checkup/what-italy-can-teach-us-about-reacting-to-the-coronavirus/> (Accessed: 15 July 2021).

Tableau. Time Series Forecasting: Definition, Applications, and Examples. Available at: <https://www.tableau.com/learn/articles/time-series-forecasting> (Accessed: 5/09/2021).

Udomboso, N. (2021). Covid-19 Cases Dashboard. Available at: [covid19-DBS-project | Tableau Public](https://public.tableau.com/app/profile/nicholas7495/viz/covid19-DBS-project/CovidCasesbyCountry?publish=yes) (Accessed: 7 September 2021).

WHO(2020).’ Coronavirus disease (COVID-19): How is it transmitted?’ Available at: <https://www.who.int/emergencies/diseases/novel-coronavirus-2019/question-and-answers-hub/q-a-detail/coronavirus-disease-covid-19-how-is-it-transmitted> (Accessed 28 July 2021).

Wikipedia. COVID-19 pandemic. Available at: https://en.wikipedia.org/wiki/COVID-19\_pandemic (Accessed: 20 June 2021).

Wikipedia. Random forest. Available at: https://en.wikipedia.org/wiki/Random\_forest (Accessed: 29 August 2021).

# Appendices

The scripts of this project have been submitted as a separate file. Filenames of the scripts are covid covid19\_DBS\_project.ipynb and Face\_Mask\_Detection.ipynb.

Tableau dashboard file has also been submitted separately. Filename is covid19-DBS-project.twbx.

Please see my published Tableau public on: [covid19-DBS-project | Tableau Public](https://public.tableau.com/app/profile/nicholas7495/viz/covid19-DBS-project/CovidCasesbyCountry?publish=yes)