Contents

1. Introduction

COVID-19 have an increasingly important influence on modern society. Our team use six multiple-step-ahead predictive models to forecast the COVID-19 daily cases from 21/09/2020 to 05/10/2020 (15 days). The model we used contains Seasonal Naïve Method, Drift, Holt-winter model (Multiplicative), Seasonal ARIMA, Feed Forward Neural Networks (FNN) and forecast combination. Our report is written to do the exploratory data analysis first, then explain the six models and finally do the model selection.

2. Exploratory Data Analysis

2.1 Data Description

We will use a descriptive summary to present some descriptive analysis such as mean, standard deviation, maximum, minimum, skew, and kurtosis.

Table 1: Descriptive summary of data

Count	205
Mean	552.0146
Standard deviation	403.9527
Median	475
Mode	9
Minimum	1
Maximum	1685
Variation	163177.7498
Kurtosis	-0.0785
Skewness	0.8753

Table 1 above shows that 205 days of daily confirmed COVID-19 cases are used for our prediction. During these 205 days, the average cases happened each day is 552. The largest number of cases happened are 1685 for that county. It also has a large variation of 163177.7498 with no symmetric distribution which shows that the daily happened cases have huge fluctuations. The negative kurtosis (-0.0785) indicates a Platykurtic, which is less peaked or flatter than a normal distribution with thin tails. A normal distribution should have 3 sample kurtoses. The positive skewness (0.8753) indicates that there are many outliers on the right of the long upper tail. Mean is larger than median, and median is larger than mode (i.e. Mean>Median>Mode).

2.2 Data Visualization

The following Figure 1 shows the time series plot of COVID-19 daily cases from 29/02/2020 to 20/09/2020. It can be seen that the time series data have both upward and downward trends. During 03/2020 to 07/2020, there is no significant seasonal pattern can be seen.

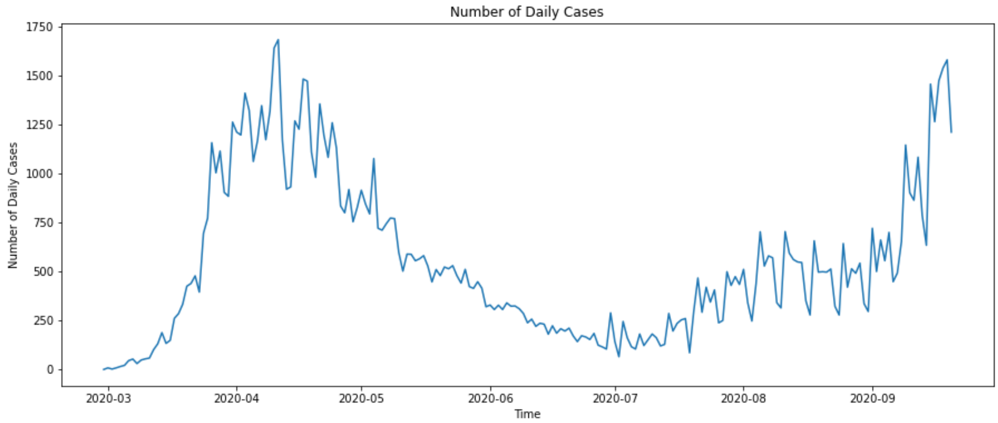

Figure 1: Number of daily cases

However, there might be a regular repeating pattern on a seasonal basis from 07/2020 to 09/2020. The seasonality seems to be seven days. To better justify our hypothesis, we use centred MA-7 to smooth the time series plot. Moving average is used to remove random variations and filter the noise. The formula for centred MA-7 is given by:

$$\widehat{T}_t = \frac{1}{7}\sum_{j=-3}^{3} y_{t+j}$$

According to Figure 2, the data has been perfectly smoothed by the moving average of order 7. This can be justified that the seasonality exists at 7 and we can apply seasonal naïve model as well as a seasonal ARIMA model in the following part.

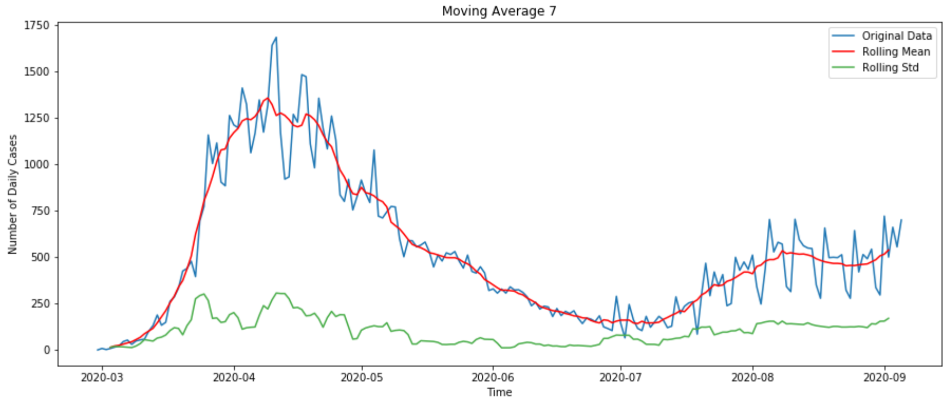

Figure 2: Visualization for MA-7

According to Figure 3, the number of accumulated daily cases is increasing smoothly from 29/02/2020 to 20/09/2020. There is a dramatic rise in the number of COVID-19 cases in the early stage (i.e. from 04/20 to 05/20). During the next 2 months, it increases slowly. Another significant growth in the number of COVID-19 cases from 07/20 to 09/20 might predict the next breakup.

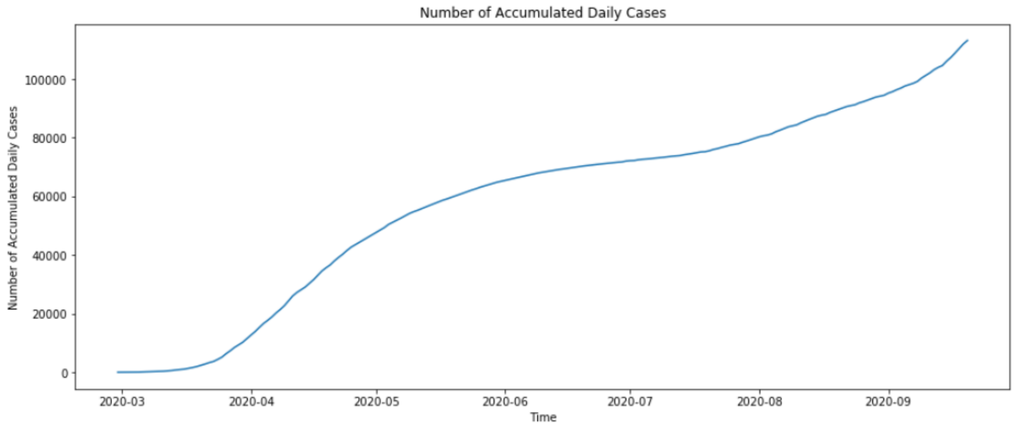

Figure 3: Number of accumulated daily cases

3. Methodology

Before applying the model, first, we need to split our train set into train set and validation set. The following model we apply is based on our validation set. The intuition is that we need to prevent our model from overfitting and evaluate our model accurately. We can report validation metrics during training, thus get a sense of which model perform best. In addition, we used MSE as our metrics, which formula is shown below.

$$\text{MSE} = \frac{1}{15}\sum_{h=1}^{15}(\hat{y}_{T+h|1:T} - y_{T+h})^2$$

3.1 Seasonal Naïve Method

3.1.1 Description

A seasonal naïve method is a naïve method for seasonal data. As the seasonality of our time series plot is seven justified by centred MA-7, we apply seasonal naïve method. Each forecast is set to be the last observation of each season (i.e. 7 days). The forecast for time T+h is given by:

$$\hat{y}_{t+h} = y_{t+h-7}$$

In this case, we repeat the last seven observations three times and then set the forecast to be the last observation of each season.

The advantage of seasonal naïve method is that they are simple and easy to apply with short previous observations (i.e. one observation). Thus, it is always used as a benchmark model to see how we can improve that via other intermediate methods. As it is a simple forecasting method, the disadvantage is quite obvious. The result is not accurate and the MSE might be quite large, which means we need to make improvement based on this simple forecasting method to make our MSE smaller.

3.1.2 Model Fitting and Forecasting

The seasonal naïve forecasting plot is shown below:

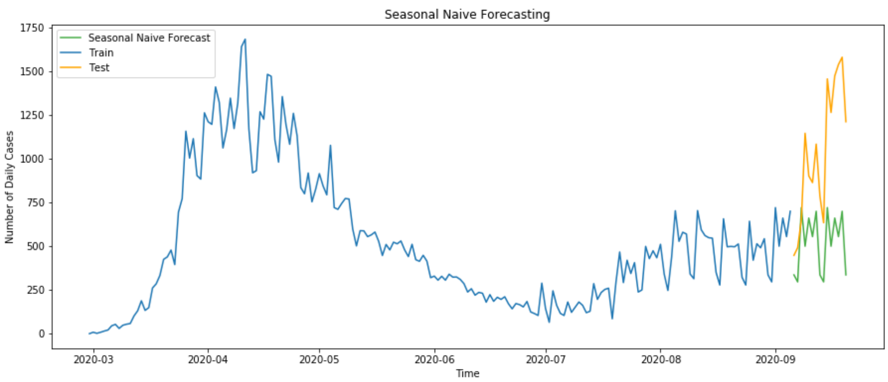

Figure 4: Seasonal naive forecasting

Figure 4 shows that the seasonal naïve model does not have a good predicted performance comparing to the validation set. The MSE shows a bad result at 359416.4000. Thus, the seasonal naïve model can be treated as our benchmark model but need to be improved.

3.2 Drift Method

3.2.1 Description

A drift method is a variation on the naïve method where the forecast is set to be the previous observations plus the average change (i.e. drift) seen in historical data. It allows forecasts to increase and decrease over time. The h-step forecasting is given as:

$$\hat{y}_{T+h} = y_t + \frac{h}{T-1}\sum_{t=2}^{T}(y_t - y_{t-1}) = y_t + h(\frac{y_t - y_1}{T-1})$$

In this case, we set to be the last observation of data set, and y_1 to be the first observation of data set.

As the drift method considering changes in forecasts, it is more accurate than a simple naïve method. It is also quite simple to calculate since we only need to set y_t and y_1. However, it is a variation on the naïve method, thus, compared to the intermediate method, the forecasts by drift method can be only treated as a benchmark model since the MSE might be large.

3.2.2 Model Fitting and Forecasting

The Visualization of prediction performance for Naïve Forecasting with Drift method is shown below:

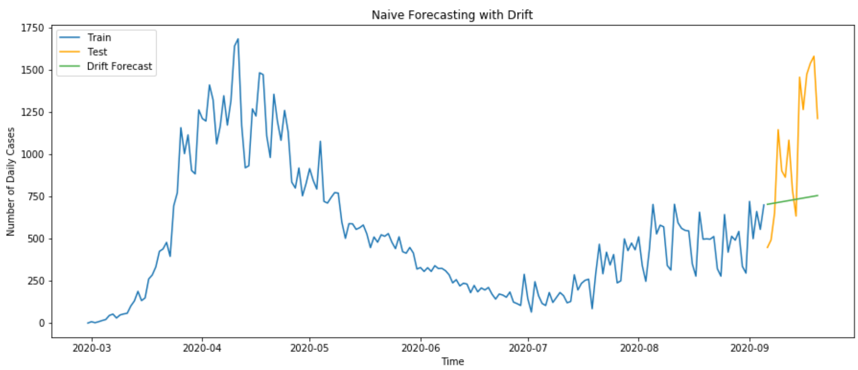

Figure 5: Naive forecasting with drift

We can see that the drift forecast is a straight line which is quite different from the validation set. The MSE can be reported as 222862.9858. Therefore, the performance of naïve forecasting with drift is not quite accurate thus can be treated as a benchmark model.

3.3 Holt-Winter Model (Multiplicative)

3.3.1 Description

There are three models of exponential smoothing: simple exponential smoothing (SES), trend corrected exponential smoothing (TCES), and holt-winters smoothing. Compared with others, holt-winters is the only model considering level (l_t), trend (b_t) and seasonality (S_t). In this case, as proved by MA-7, there exist seasonality and the seasonal period is 7 days. Hence, we choose to take holt-winter model into consideration.

Holt-winters smoothing is used for both additive seasonality and multiplicative seasonality. The former one is the constant seasonal variation along with the trend and the latter is useful when the seasonal pattern changes in a strong pattern and is proportional to the level of the series. In addition, the component forms of them are shown below:

$$Additive: l_t = \alpha(y_t - S_{t-M}) + (1-\alpha)(l_{t-1} + b_{t-1})$$

$$b_t = \beta(l_t - l_{t-1}) + (1-\beta)b_{t-1}$$

$$S_t = \gamma(y_t - l_t) + (1-\gamma)S_{t-M}$$

$$\hat{y}_{t+1|1:t} = (l_t + b_t) + S_{t+1-M}$$

$$Mutiplicative: l_t = \alpha\left(\frac{y_t}{S_{t-M}}\right) + (1-\alpha)(l_{t-1} + b_{t-1})$$

$$b_t = \beta(l_t - l_{t-1}) + (1-\beta)b_{t-1}$$

$$S_t = \gamma\left(\frac{y_t}{l_t}\right) + (1-\gamma)S_{t-M}$$

$$\hat{y}_{t+1|1:t} = (l_t + b_t) * S_{t+1-M}$$

Compared with other advanced models, holt-winters smoothing is easy to learn and apply (Connectus, 2016). In addition, it can handle lots of complicated seasonal patterns by simply finding the level, then adding in the effects of trend and seasonality. The most changeling part of this model is to find the right parameters (Orangematter, 2019). Furthermore, the lag is a side effect of the smoothing process because it neglects the ups and downs associated with random variation. However, it could help us to see the underlying phenomenon when presenting data and making a forecast of future values.

3.3.2 Model Fitting and Forecasting

In terms of MA-7, we set the seasonal period as 7 days. In addition, because there are two types of holt-winter models: additive and multiplicative, we made a judgement based on the corresponding training MSEs shown below.

Table 2: MSE for additive and multiplicative holt-winters on training data

	Additive	Multiplicative
MSE	13156.5239	9875.8733

It is obvious that multiplicative holt-winters have smaller training MSE, as a result, we use this model to do prediction on the validation set. By using the package in python, we can easily get the best parameters for multiplicative holt- winters as shown in table 3 below:

Table 3: Best parameters for multiplicative holt-winters model

α	β	γ	$l0$	$b0$
0.3662	0.1509	0.1325	565.7497	7.6723

By fitting the parameters to the model, we can get the performance of prediction MSE at 181809.4669 which is better than that of the former two benchmark models.

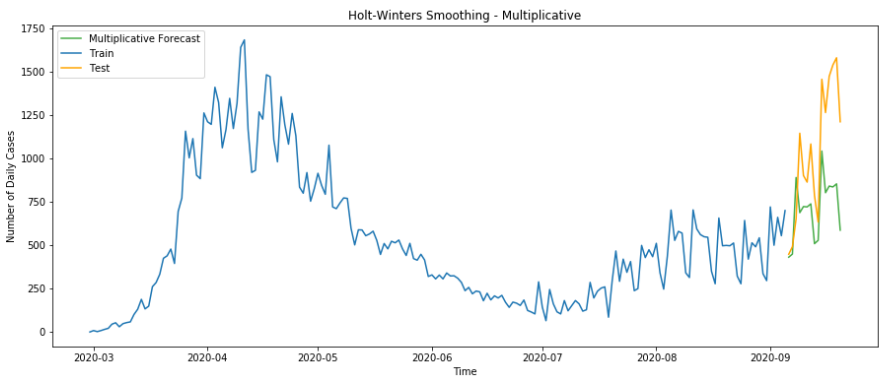

Figure 6: Holt-winters smoothing - multiplicative

Figure 6 shows our multiplicative forecasts basically consistent with the validation changes, although less than the validation dataset.

3.4 Seasonal ARIMA

3.4.1 Description

The Seasonal Autoregressive Integrated Moving Average (SARIMA) model are denoted as:

$$\mathrm{SARIMA}(p, d, q)(P, D, Q)_m$$

where p is the degree of the Non-seasonal AR model, d is the Non-seasonal integrated value, q is the degree of the Non-seasonal MA model, P is the degree of the Seasonal AR model, D is the Seasonal integrated value, Q is the degree of the Seasonal MA model, m is the seasonal frequency.

Formulation with backshift operators:

$$\left(1-\sum_{i=1}^{p}\varphi_i B^i\right)\left(1-\sum_{i=1}^{P}\varphi_i B^{im}\right)(1-B)^d(1-B^m)^D Y_t$$
$$= c + (1+\sum_{i=1}^{q}\theta_i\, B^i)(1+\sum_{i=1}^{Q}\Theta_i\, B^{im})\varepsilon_t$$

The SARIMA model consists of 6 major components:

The Non-seasonal AR(p) part of SARIMA shows that y_t is regressed on its own lagged $y_{t-1:t-p}$ values. The Seasonal AR(P) part represents Non-seasonal AR(p) with m (seasonal frequency). The Non-seasonal difference order(d).

The Seasonal difference order is represented by D. The Non-seasonal MA(q) part shows that $\varepsilon_{t-1:t-q}$ values occurred simultaneously at different times. The Seasonal MA(Q) part represents Non-seasonal MA(q) with m (seasonal frequency).

The main advantage of SARIMA is it support seasonal time series data. It can be used in stationary seasonal time series data without missing data (Zhang et al., 2013). Long-term trend and seasonal effects are considered in SARIMA (Wang, Feng, & Liu, 2013).

The disadvantage of SARIMA is that it is difficult for it to deal with nonlinear relationship in time series data. For time series forecasting method, SARIMA cannot interpret nonlinear relationship and complexity in the forecast (Zhang et al., 2013). For abnormal errors in time series data, SARIMA cannot make accurate and stationary forecast (Wang, Feng, & Liu, 2013).

3.4.2 Pre-processing Steps and Optimization

ACF plot is used to analyze the stationarity. For non-seasonal time series data, it can be considered stationary if the ACF plot dies down or cuts off reasonably quickly. Partial autocorrelations represent the correlation of two variables without the effects of other variables. Both the ACF and PACF plot is important for identifying Seasonal-ARIMA model.

According to Figure 7, the ACF plot for the original data dies down extremely slowly. The PACF plot, there is a large spike at lag 1 followed by a damped wave that alternates between positive and negative correlations.

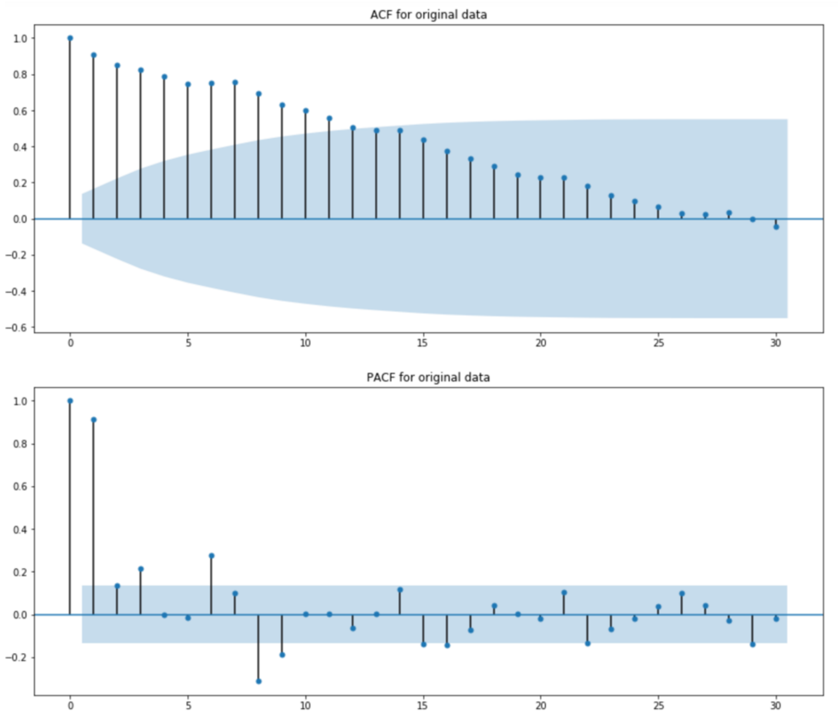

Figure 7: ACF and PACF plots for original data

Since the ACF plot dies down slowly, the data transformation is necessary. Firstly, we use log transformation. According to Figure 8, the ACF plot for logged data dies down quicker than that for original data. However, it is still nonstationary, and we need to do more transformation.

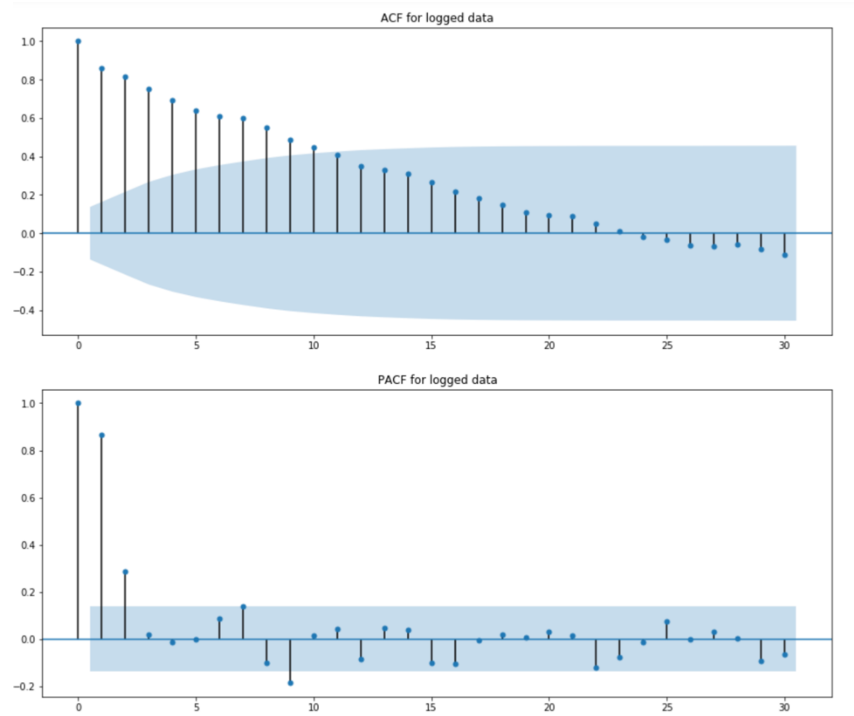

Figure 8: ACF and PACF plots for logged data

The further transformation we chose is the first order differencing on the logged data. According to Figure 8, the differenced logged data represents stationarity because it remains the same mean and a relatively constant variance and covariance.

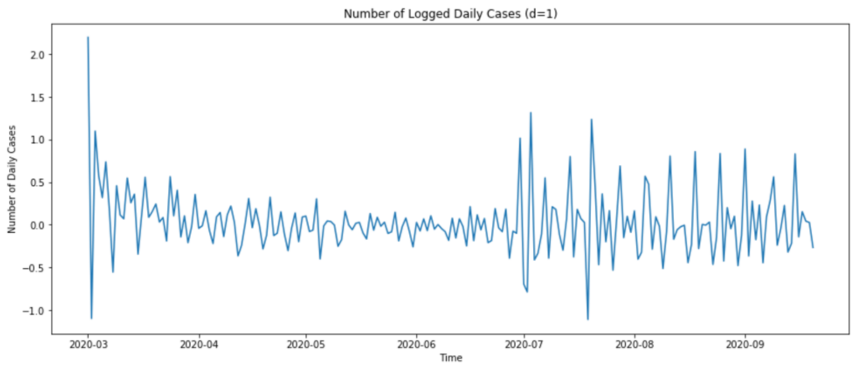

Figure 9: Visualization of logged data

The stationarity can be more accurately explained by the ACF and PACF plots. According to the ACF plot in Figure 10, it cuts off after a range between lag 2 to lag 4. From the PACF plot, we can see that there are significant correlations at the first, second and third lags, followed by some non-significant correlations.

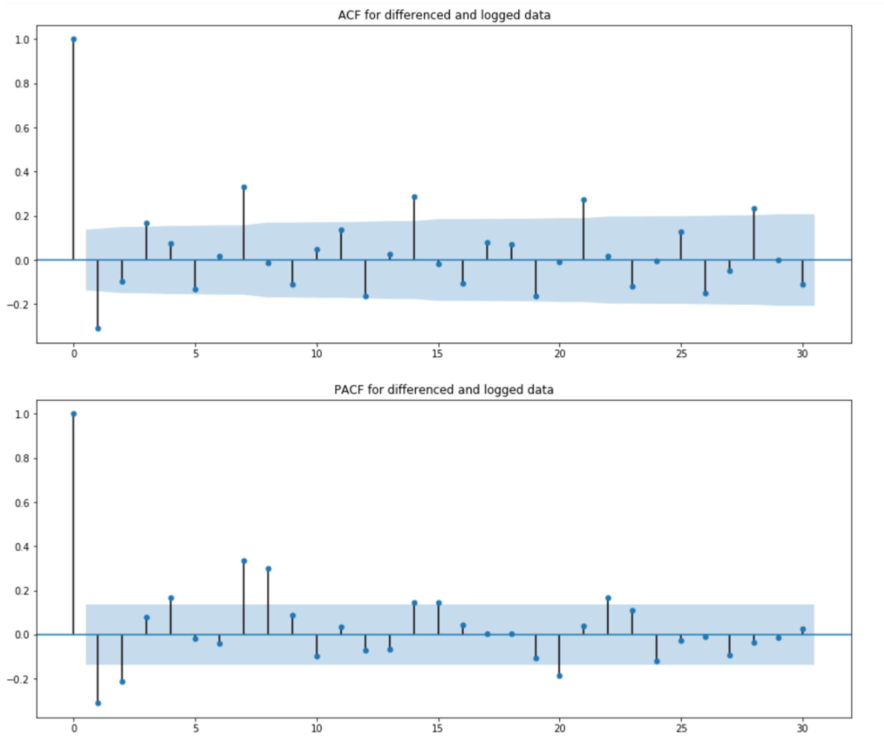

Figure 10: ACF and PACF for differenced and logged data

In order to get the best order of AR (p) and the best order of MA (q), we will use Akaike's information Criterion and set both the p and q between 1 to 4 to do the selection. After searching, the best fitting order is p=4, q=4.

For choosing the seasonal order P, Q and D, we will use the ACF and PACF plot to get the straightforward answer. For P we choose 1 because according to the PACF plot (Figure 10), it is positive at lag 7. For D we choose 0 since according to the PACF plot (Figure 10), there exists an unstable seasonal pattern over time. For Q we choose 0 since according to the ACF plot (Figure 10), it is positive at lag 7.

3.4.3 Model Fitting and Forecasting

As a result, SARIMA (4,1,4) (1,0,1,7) is used to fit a model on training data and compared to actual data.

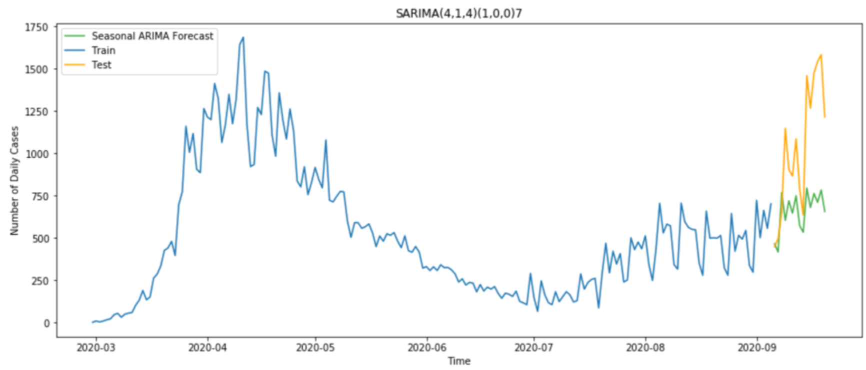

Figure 11: SARIMA forecast

According to Figure 11, it shows that our seasonal ARIMA forecast has a slight upward trend with a weekly seasonality. However, it still does not reach our validation set, and the MSE to represent the performance of prediction is 233220.6401.

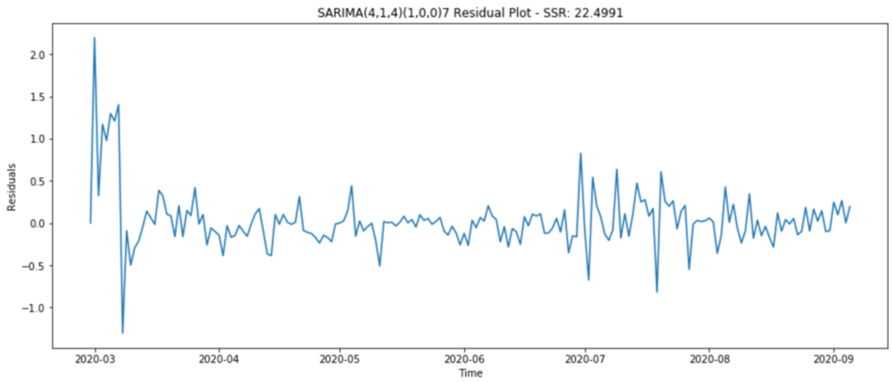

Figure 12: Residual plot for SARIMA

The visualization of residuals from the model is shown in Figure 12. It seems that the residuals of the model are relatively stationary instead of the first few days from 03/2020. What's more, according to Figure 13, the ACF plot for residuals show no significant autocorrelations which means that the model fits good for the prediction.

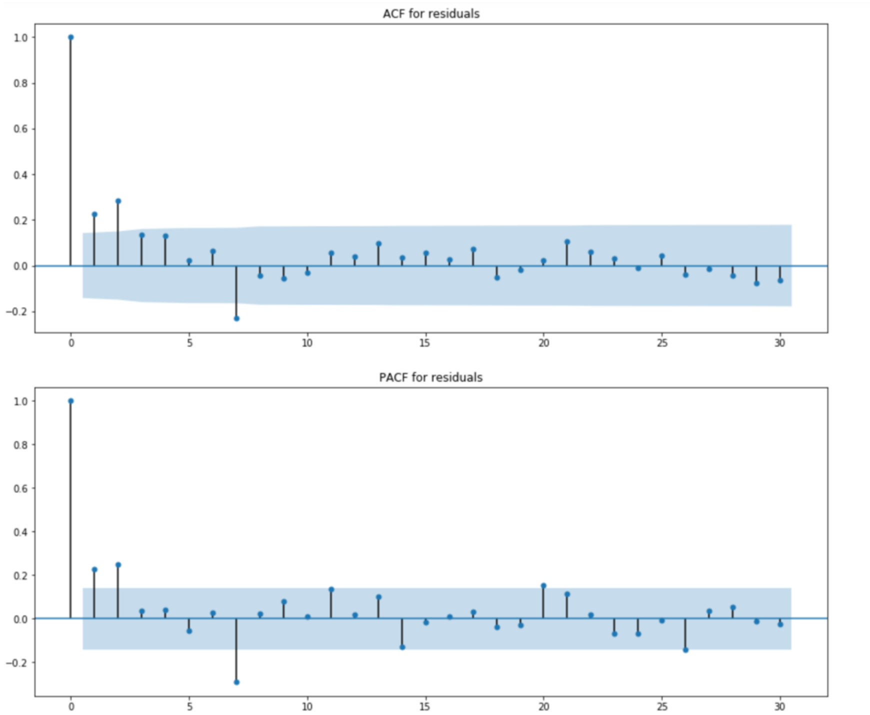

Figure 13: ACF and PACF for residuals

3.5 Feed Forward Neural Networks

3.5.1 Description

The neural network is a computational model inspired by the network of neurons in the human brain. It is made of an input layer, one or more hidden layers, and an output layer, which specific structure is shown in Figure 14. The input layer receives data, output layer export data, and hidden layers receive/process/send data within the network.

FNN model is suitable for cross-sectional data and time series data both. It is widely known for its high prediction accuracy and can perform well in many complex problems. However, it is hard to be interpreted. Additionally, a neural network has a risk of over-fitting when capturing a simple phenomenon. Since FNN is used for complex situations, it requires a longer time to run, more data and high-quality hardware or cloud computing services (Hoang, 2019).

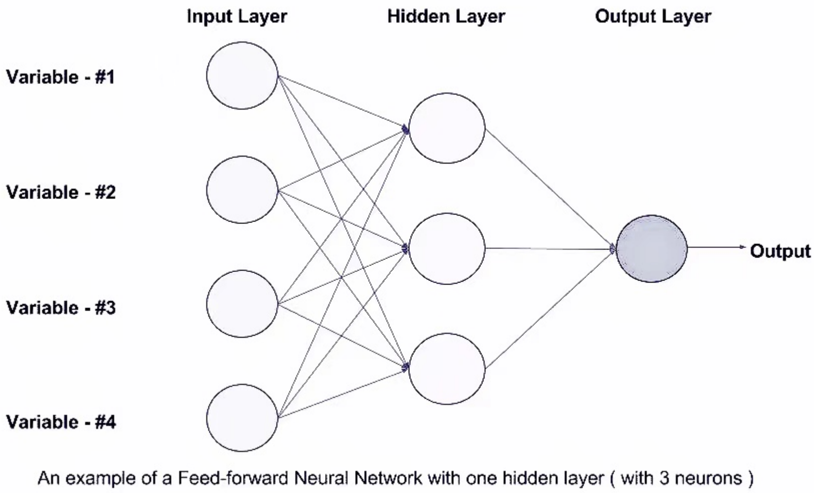

Figure 14: FNN neural network

We manually do hyper-parameter optimisation to build the optimized model.

Table 4: Best hyperparameters for FNN model

Time window	11
Hidden layer	1
Scaler	(0,1)
Neurons	20
Optimizer	RMSprop
Batch size	16
Epochs	100

For choosing the time window, since the NNAR$(p, P, k)_m$ means a model with inputs

$$(y_{t-1}, y_{t-2}, \dots, y_{t-p}, y_{t-m}, y_{t-2m}, \dots, y_{t-Pm})$$

Thus, time window equals p plus P*m. For p (4), P (1), m (7), we have a time window:

Time window = p + P * m = 4 + 1 * 7 = 11

The neural network we build includes one input layer with 11 neurons (time window), one hidden layer with 20 neurons and one output layer with a single neuron. The output neuron is a combination of 20 neurons in the hidden layer. This structure can be shown in Figure 15.

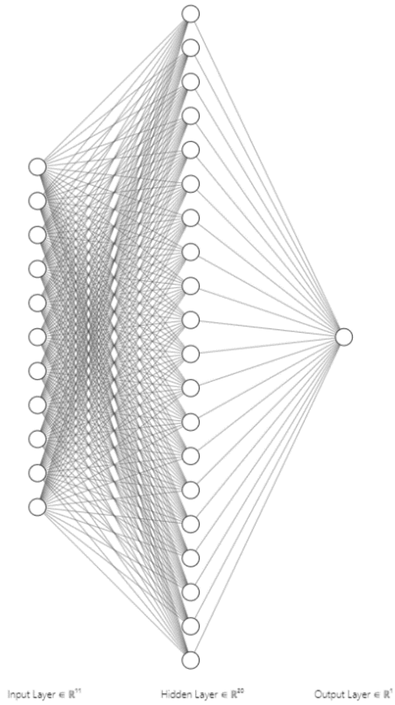

Figure 15: FNN neural network for COVID-19 data

The reason we scale the data from zero to one is that it is the best practice when we are using sigmoid or tanh activation function. Scale data range is set to be (0,1) when using Define optimizer as RMSprop to sustain a moving average of gradients square divided by moving average root (Keras Team, n.d.).

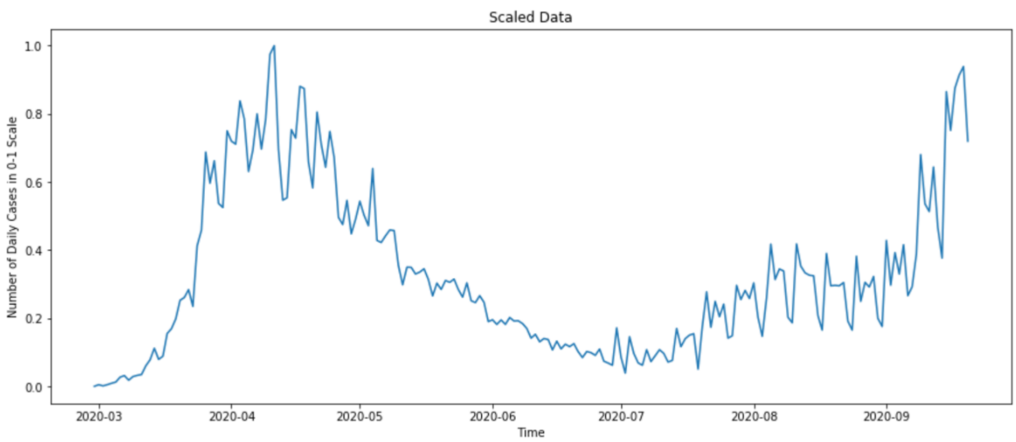

Figure 16: Scaled data

Figure 16 shows scaled data from 0 to 1 number of daily cases. It is consistent with original data using only training data to prevent information leakage.

Batch size is the number of samples to go through before updating the model. We set batch size as 16 because the size of our time series data is small. We do not need to save too much computational cost for setting a larger batch size. What' more, the smaller the batch size is, the better we prevent overfitting.

Epochs represent the number of times data passing through the neural network. We set epochs as 100 since it is consistent with data diversity to prevent underfitting and overfitting (SHARMA, 2019).

3.5.2 Model Fitting and Forecasting

Then we train the FNN model.

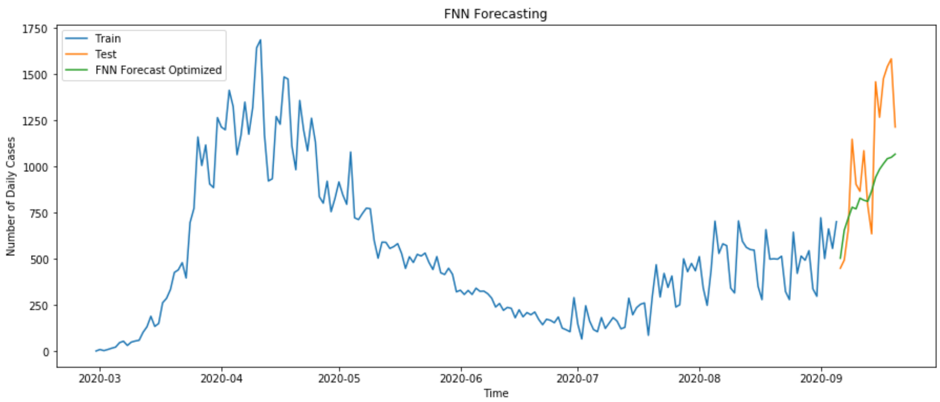

Figure 17: FNN forecasting

Figure 17 shows that our FNN forecasting model has a clear upward trend, thus being more consistent with the test changes. MSE can also be calculated to see the performance of our forecasting (95153.3882).

3.6 Forecasting Combinations

3.6.1 Description

It is possible to combine different models via using a simple average, which usually performs well. When there exist structural breaks in the data, it is plausible to combine models with different levels of adaptability, which might lead to better results than relying on a single model. In this case, we chose to set the same weight of different models because there is no obvious bias and for calculation convenience.

Model combination with stable, equal weights has generally been found to work well as hedging against the performance of the worst models and requiring the estimation of only a few parameters (Timmermann, 2004). However, the drawback is similar to its advantages. A combined model would shrinkage the performance of the best single models as well.

3.6.2 Model Fitting and Forecasting

Then we train the combined model and generated its plot (Figure 18) with MSE of 197677.8988.

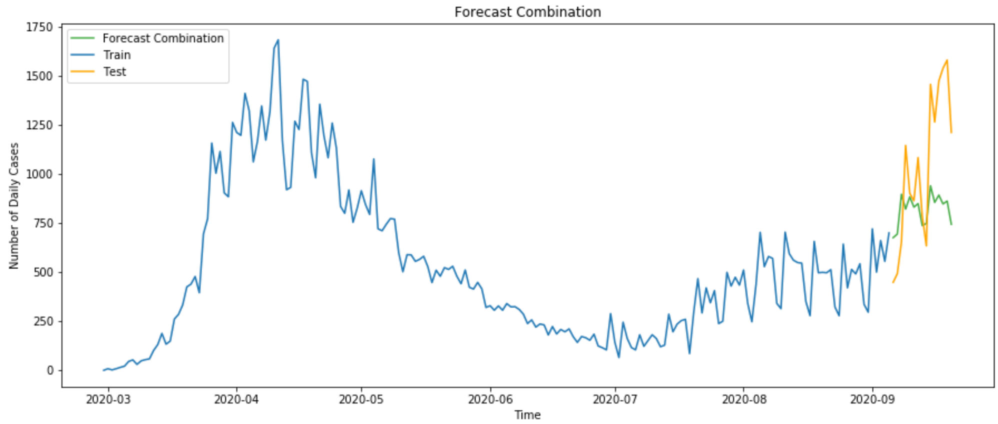

Figure 18: Forecast combination

The plot shows that our Forecast combination model has a clear trend and seasonality moving consistently with the test changes.

4. Model Selection

The performance of our selected models can be reflected by MSE. The comparison of MSE can be reported as:

Table 5: Model comparison

Model	MSE
FNN	95153.3882
Holt-Winters model -Multiplicative	181809.4669
Forecast combination	199792.6260
Drift	222862.9858
Seasonal ARIMA	233220.6401
Seasonal Naïve	359416.4000

According to the MSE chart, FNN performs best as the MSE of FNN is the smallest among all selected models.

The drift and seasonal naïve method are simple forecasting methods thus come out large MSE. Although the computations of seasonal naïve method and drift are quite simple, the models do not perform well as the forecasts are too naïve. We cannot simply predict the

COVID-19 cases as the last observation of each season (i.e. seven days). Although drift considers some changes, the changes are still the simple average changes. As the COVID-19 daily cases change irregularly without a fixed pattern, the drift forecasts are not accurate. Therefore, drift forecasts and seasonal naïve forecast can be only treated as benchmark models.

For intermediate models, the MSE of Holt-winter smoothing is 181809.4669 which is smaller than Seasonal ARIMA model. This means the Holt-winter smoothing performs better than Seasonal ARIMA model. To compare the forecasts of two models (Figure 19), Holt-Winters forecasts show a better trend than SARIMA forecasts.

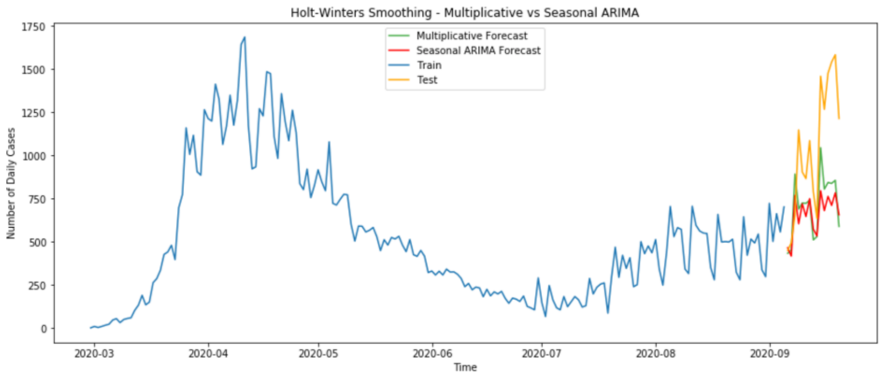

Figure 19: Holt-winters smoothing - multiplicative vs seasonal ARIMA

The reason might be that the COVID-19 daily cases change with uncertainty; therefore, it does not have a linear relationship and is complicated to forecast (Zhang et al., 2013). The SARIMA model may not be accurate due to the uncertainty of our target data. The Holt-Winter model, however, is suitable for data with a seasonal pattern. As the effects are added to the trend and seasonality, the trend performs better than SARIMA forecasts. This might be the reason that we have smaller MSE in Holt-Winter Smoothing model.

The MSE of Forecast Combination Model is 199792.6260 which is smaller than the MSE of SARIMA model but does not have a significant difference. The reason might be that Forecast Combination is useful for data with structural breaks. For data with structural breaks, it may perform significantly better than SARIMA model. However, we are analyzing COVID-19 daily cases. The COVID-19 daily cases do not pause thus do not have structural breaks. As a consequence, it does not perform much better than SARIMA model.

FNN performs best among all selected models as the MSE is the smallest. This is because our target is COVID-19 daily cases which is a cross-sectional data. Cross-sectional data is data collected by observing many kinds of subjects (White, 2020). The COVID-19 daily cases are typically cross-sectional data since we need to collect lots of data to ensure if it is a real COVID-19 case. Moreover, as the COVID-19 cases change every day, it provides us with a snapshot of the current cases. Therefore, the FNN model is perfectly fit for our target data, thus performs best among all selected models.

5. Conclusion

In conclusion, from the six predict models we use, Feed Forward Neural Network model has the best performance of prediction and it fits better for our training data which is cross-sectional. It is chosen to forecast the COVID-19 daily cases of the country from 21/09/2020 to 05/10/2020.

Reference

Connectus. (2016). 5 Advantages and Disadvantages of Exponential Smoothing. Retrieved from https://connectusfund.org/5-advantages-and-disadvantages-of-exponential-smoothing

Hoang, P. (2019). What are the advantages and disadvantages of making a feedforward neural network deeper? Retrieved from https://www.quora.com/What-are-the-advantages-and-disadvantages-of-making-a-feedforward-neural-network-deeper

Keras Team. (n.d.). *Keras documentation: RMSprop*. Retrieved from https://keras.io/api/optimizers/rmsprop/

Orangematter. (2019). Holt-Winters Forecasting Simplified. Retrieved from https://orangematter.solarwinds.com/2019/12/15/holt-winters-forecasting-simplified/

SHARMA, S. (2019). *Epoch vs batch size vs iterations*. Retrieved from https://towardsdatascience.com/epoch-vs-iterations-vs-batch-size-4dfb9c7ce9c9

Timmermann, A. (2004). Forecast combinations. Retrieved from https://econweb.ucsd.edu/~atimmerm/combine.pdf

Wang, S., Feng, J., & Liu, G. (2013). Application of seasonal time series model in the precipitation forecast. *Mathematical and Computer Modelling*, *58*(3-4), 677-683. doi:10.1016/j.mcm.2011.10.034

White, Susan. (2020, June 1). Mind Blowing Information on Cross sectional data. Retrieved from https://www.allassignmenthelp.com/blog/cross-sectional-data/.

Zhang, X., Liu, Y., Yang, M., Zhang, T., Young, A. A., & Li, X. (2013, May 1). *Comparative study of four time series methods in forecasting typhoid fever incidence in China*. Retrieved from https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3641111/#:~:text=In%20conclusion%2C%20the%20SARIMA%20model,properties%20and%20effective%20modeling%20process.&text=The%20disadvantage%20of%20the%20SARIMA,within%20the%20time%20series%20data