Task B6: Machine Learning 3

Conducted by:

Group 1

Team Members	*Email*
Thanh Tam Vo	103487596@student.swin.edu.au
Phuc Khai Hoan Cao	103804739@student.swin.edu.au
Tai Minh Huy Nguyen	104220352@student.swin.edu.au

April 3, 2024

Table of Contents

1 Introduction

In task B4, we chose the parameter **LOOK_UP_DAYS** for no reasons. So, in this Task B6, we will try to apply some analystic techniques to find the approriate **LOOK_UP_DAYS**. Also, we will apply a simple ensemble learning technique to improve the model prediction: combining the output from Sequence Models (RNN, LSTM, GRU) with the output from Autoregressive Intergrated Moving Average (ARIMA) model. The dataset that will be used for the experiment is the Stock Record of Apple Inc; Long story shorts, task B6 is the extension based on task B4.

2 Importing Dependencies

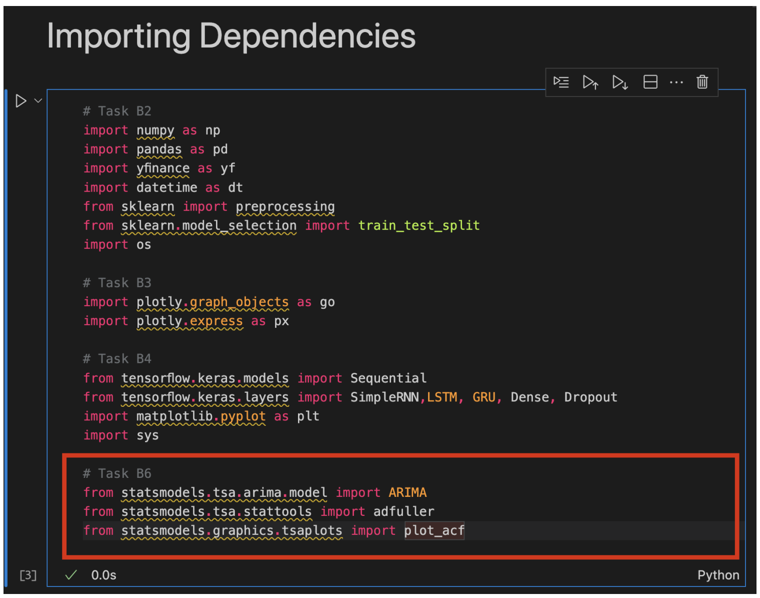

Figure 1: Importing Python Packages for Task B6

There are several new packages in task B6:

- **ARIMA**
- **adfuller**
- **plot_acf**

3 Autoregressive Intergrated Moving Average (ARIMA) model

3.1 Choosing `p` value for ARIMA Model (`LOOK_UP_DAYS`)

`p` is the number of past observation in order to predict the future record. In this case, **`p`** equals to the **`LOOK_UP_DAYS`**. But first, let's take a look at the entire dataset as Figure 2 below:

Figure 2: Training Set for Task B6

To identify the **`p`** value, we use the Partial Autocorrelation Function (PACF) as below:

```
plot_pacf(scaledStockData["Close"])
```

After executing, below is the result:

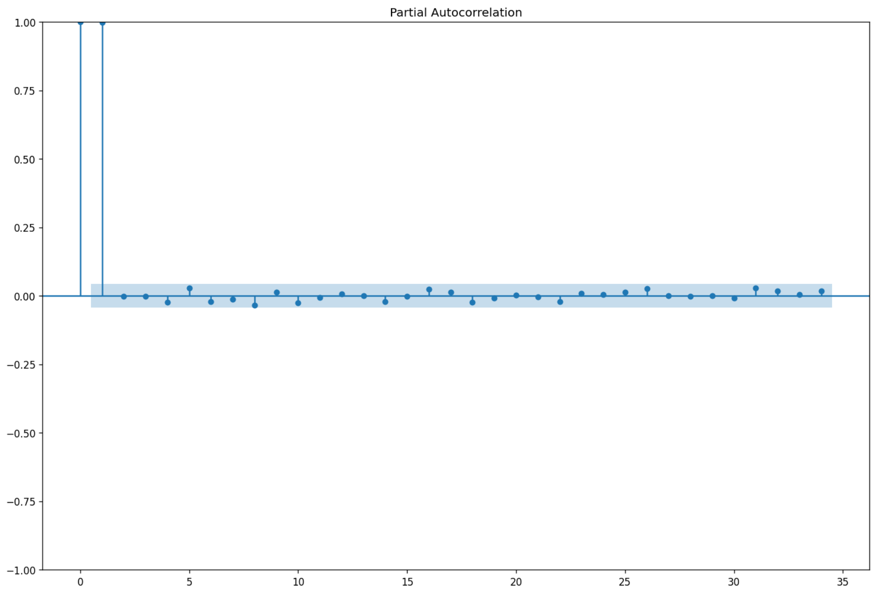

Figure 3: Partial Autocorrelation Function result

In Figure 3, it can be easy to conclude that the first two data points are highly correlated. In other words, the Stock records of the last two days strongly affect to the current Stock price. Hence, the final **`LOOK_UP_DAYS`** is set to 2.

3.2 Choosing `d` value for ARIMA model

In ARIMA Model, it requires the stationary in the series. In Figure 2, we can see that the stock price is non-stationary: Overall, the Stock Price of Apple Inc. follows the upward trend over the period of 8 years (from 2010 to 2018). To make the data more stationary, we need to take the differeciation until it becomes stationary. We need to plot the differeciation result and plot as Figure 4.

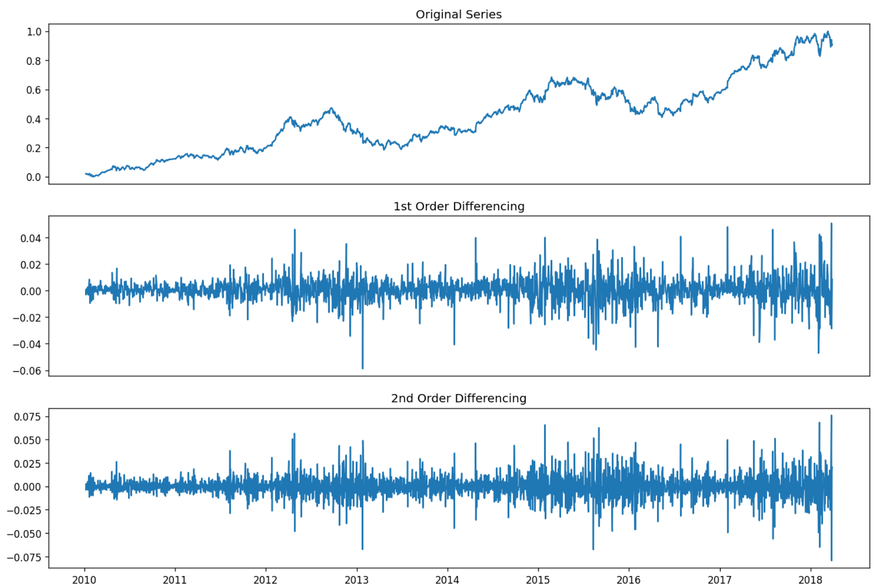

Figure 4: Differeciation result

In Figure 4, the first order and second order differencing is much more stationary. But we will choose first order for the experiment. So, **`d=1`**.

3.3 Choosing q for ARIMA model

After we make sure that our series data is stationary by taking the first order differencing. Then, we use the Autocorrelation Function to plot the series data:

```
# Taking the first order differeciation
plot_acf(scaledStockData["Close"].diff().dropna())
```

And the Figure 5 below is the result:

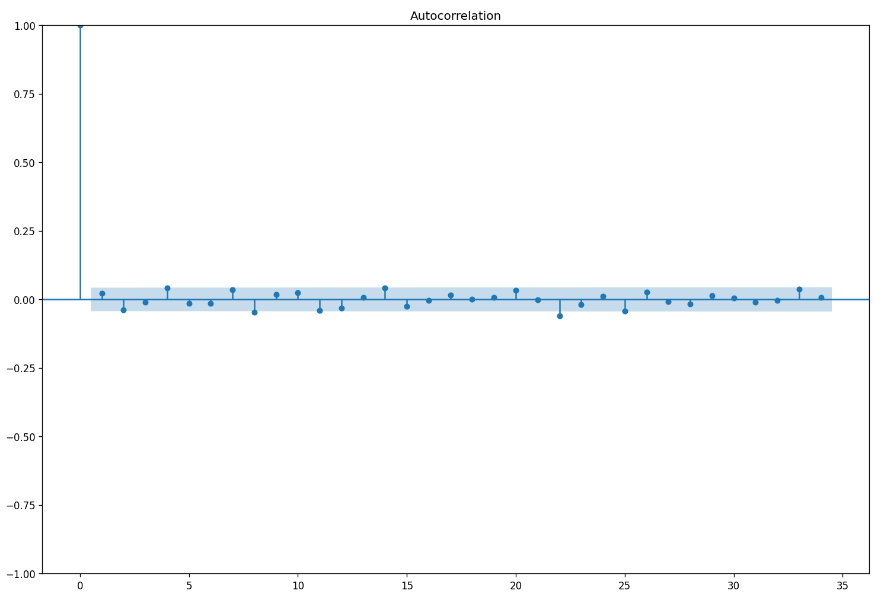

Figure 5: Autocorrelation result

We need to determine **q**: how much moving average (MA Model) we need to make sure that we can reduce the Autocorrelation level from the stationary series. In Figure 5 above, there is only one candle stays out of the blue area, so **q=1**.

4 Methodology

4.1 Training Methodology

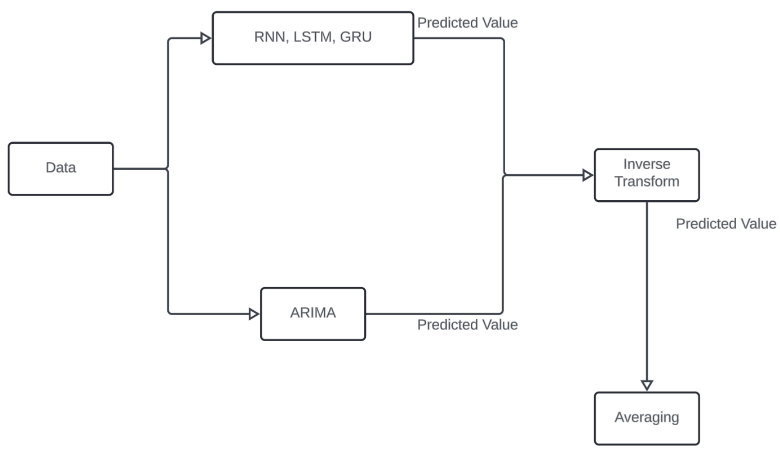

Figure 6: Training methodology for Task B6

In Figure 6, two models are trained separately with the same dataset (Apple Inc. Closed Price). Then, if it requires, we will inverse the predicted result to the original close price value. Finally, we will take the average values of two predicted results from ARIMA and one of these models: RNN, LSTM, GRU.

4.2 Model Configuration

There are 3 model: RNNs, LSTM, GRU; they must be evaluated with the same configuration (Same as Task B4):

- Two hidden layers
- 80 units per layer
- Fully connected (No dropout)
- Optimizer: Adam Optimizer Algorithm
- Number of epochs: 30
- All records are scaled using MinMax Scaler

The ARIMA Model must has the parameters of $(p = 2, d = 1, q = 1)$

5 Results

5.1 Ensemble Learning 1: LSTM and ARIMA

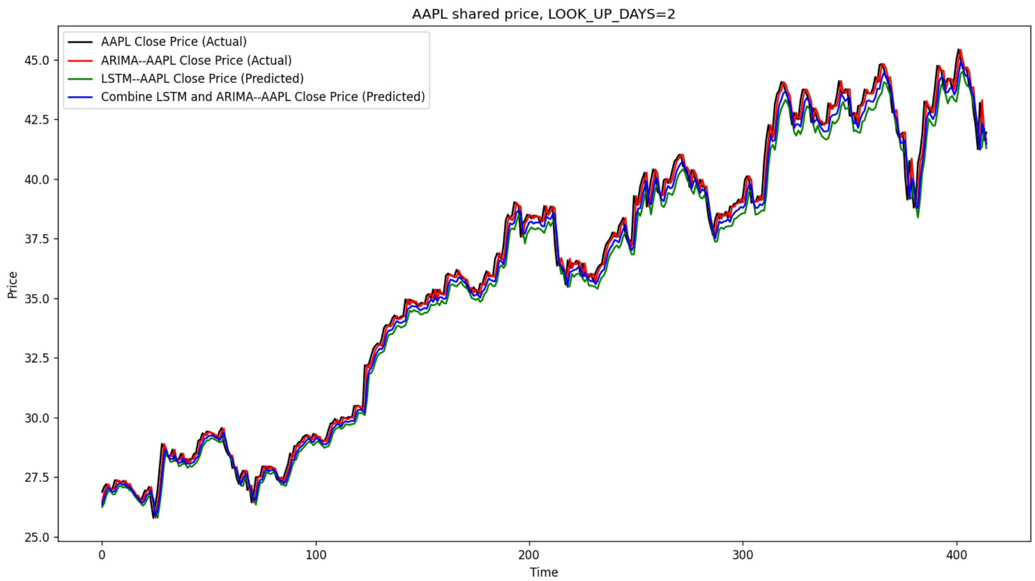

Figure 7: Training Model with ARIMA and LSTM

5.2 Ensemble Learning 2: RNN and ARIMA

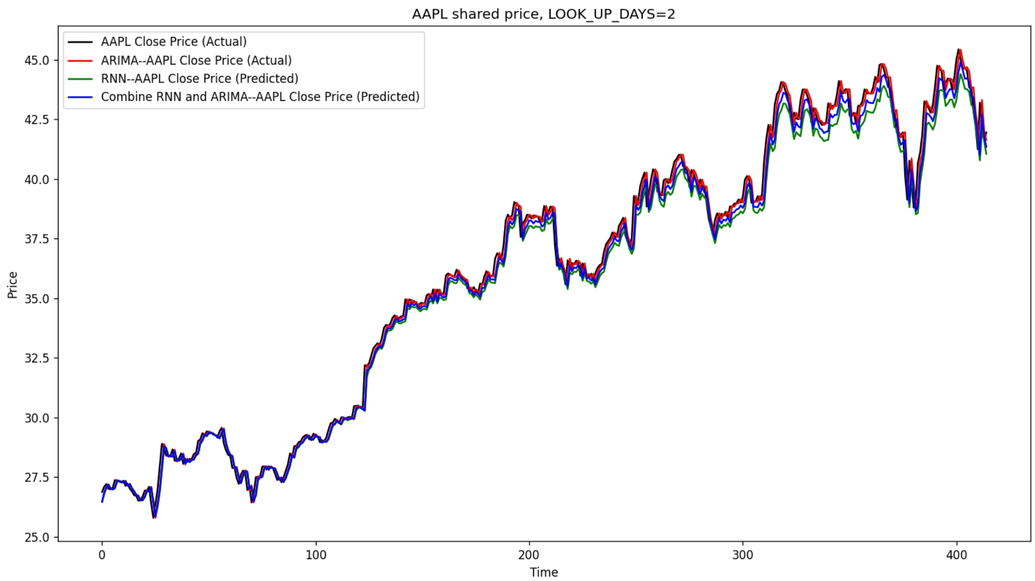

Figure 8: Training Model with ARIMA and RNN

5.3 Ensemble Learning 3: GRU and ARIMA

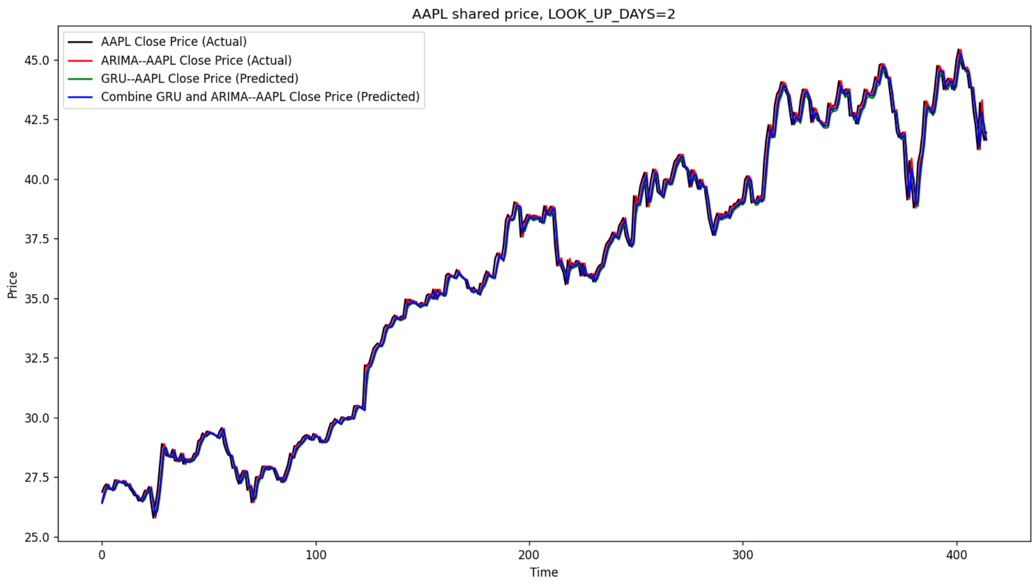

Figure 9: Training Model with ARIMA and RNN

5.4 Justifications

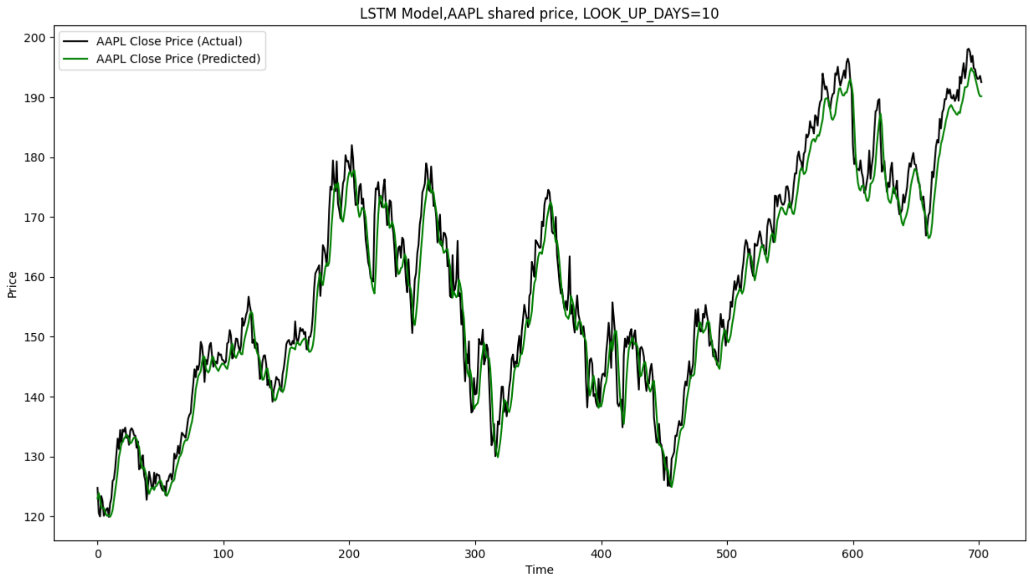

Figure 10: Prediction Result from Task B4

In Figure 10, we chose **`LOOK_UP_DAYS=10`** for no reasons. Compared to the result in Figure 9, our prediction result seems to get better. This can be explained by the analystic technique that we use: using Partial Autocorrelation to determine **`LOOK_UP_DAYS`**.

6 Conclusion

Overall, using the predicted output from the ARIMA model gives us a better result. But depend on the series (other Stock Tickets such as: IBM, etc.), it requires a data analystic stage to determine approriate parameter before training model. The trained model in this Task B6 can not be applied to predict other stock closed values.

7 Resources

- Quick way to find p, d and q values for ARIMA. Available at https://analyticsindiamag.com/quick-way-to-find-p-d-and-q-values-for-arima/
- Guide To AC and PAC Plots In Time Series. Available at https://analyticsindiamag.com/guide-to-ac-and-pac-plots-in-time-series/
- How to Create an ARIMA Model for Time Series Forecasting in Python. Available at https://machinelearningmastery.com/arima-for-time-series-forecasting-with-python/