ADVANCED ANALYTICS

ASSIGNMENT 2

LIM JIA LOK, DARREN, ALDALTON

Contents

Presentation Link:

https://youtu.be/SEzRakG5A3U

Introduction

In this assignment, the team was tasked to create a machine learning (ML) model to forecast the stock share price of the top 5 shares, starting from May 1st 2020 to May 1st 2021, in order to purchase a Mercedes-AMG GT, costing at minimum RM 1.1 million.

The top 5 stock shares in Malaysia are found via comparing the average growth percentage over a period of 335 days, specifically from 1st June 2019 to 1st May 2020.

Lastly, the ML model is fed the chosen stock shares to forecast the stock share price 1 year into the future, from 1st May 2020 to 1st May 2021. It will then be used to determine whether it is enough to purchase the car.

Methodology

a. Description of data (description of data frame)

Stock Ticker

- Stock ticker shows how individual stock prices and stock market rises and fall.
  Stock ticker changes as the prices vary when the stock market is open. The ticker will then show the final results from that trading day once the market closes. Stock ticker commonly contains ticker symbol, open, high, low, close, adjustment close and volume

Ticker Symbol

- Each publicly traded company has a ticker symbol that uniquely identifies the companies. This is because it simplifies the naming of the company and makes it easier to look up the stock ticker. For example, Amazon's ticker symbol is AMZN.

Open

- Open is the price of a stock that started trading when the stock market opens. The price might be the same as when the market closes the night before. However, during after hours trading, events that happen then might change the stock price overnight. Events such as company earnings reports could happen in that period.

High

- High is the highest price the stock achieved between the opening and closing of the stock market.

Low

- Low is the lowest price the stock achieved between the opening and closing of the stock market

Close

- Close is the raw price of a stock when the stock market closed for that trading day. This closing price is normally used to determine how that particular share performed during that trading day. Thus, closing prices are normally fed into machine learning to calculate and predict the performance of that particular stock.

Adj_close (Adjusted Close)

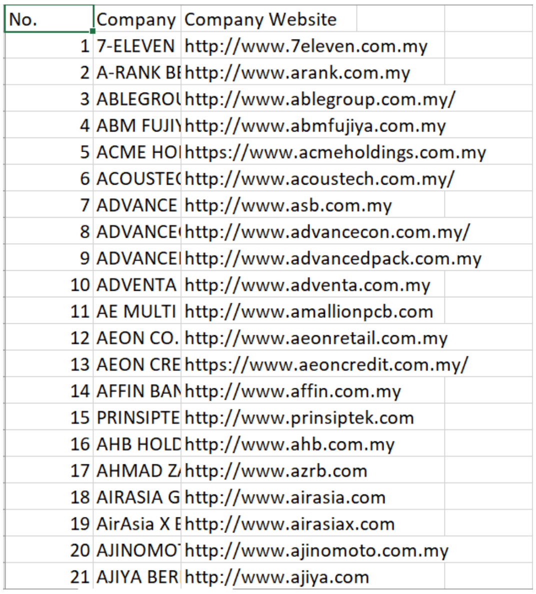
- Adjusted closing price basically adjusts or modifies the stock closing price due to corporate actions which are dividends, stock splits and rights offerings. This adjustment to the stock closing price is to take into account these actions in order to properly reflect the accuracy of the closing price. For example, in the event the share price closed at RM50 and there is a corporate action such as posting dividends for RM10 per share. The newly adjusted closing price would be RM40 to account for that dividend. Thus, adjusted closing price is more accurate than raw closing price. Thus, this parameter is used to feed data into our ML model as it more accurately reflects the closing price.

Volume

- Volume is the number of shares bought or sold. It varies between opening and closing of the stock market.

b. Exploratory data analysis (graphical plots of data)

A web extension, Table Capture, was used to gather Malaysian companies that have a stake in the Malaysian stock market. The data was saved in an Excel file. The results are as seen below.



No.	Company	Company Website
1	7-ELEVEN	http://www.7eleven.com.my
2	A-RANK BE	http://www.arank.com.my
3	ABLEGROU	http://www.ablegroup.com.my/
4	ABM FUJIY	http://www.abmfujiya.com.my
5	ACME HOL	https://www.acmeholdings.com.my
6	ACOUSTEC	http://www.acoustech.com.my/
7	ADVANCE	http://www.asb.com.my
8	ADVANCEC	http://www.advancecon.com.my/
9	ADVANCED	http://www.advancedpack.com.my
10	ADVENTA	http://www.adventa.com.my
11	AE MULTI	http://www.amallionpcb.com
12	AEON CO.	http://www.aeonretail.com.my
13	AEON CRE	https://www.aeoncredit.com.my/
14	AFFIN BAN	http://www.affin.com.my
15	PRINSIPTE	http://www.prinsiptek.com
16	AHB HOLD	http://www.ahb.com.my
17	AHMAD ZA	http://www.azrb.com
18	AIRASIA G	http://www.airasia.com
19	AirAsia X B	http://www.airasiax.com
20	AJINOMOT	http://www.ajinomoto.com.my
21	AJIYA BER	http://www.ajiya.com

Figure 1 : Excel table of company names

After that, a web scraping program written in Python code was used to gather the company's ticker symbols. From the scrapped ticker symbols, Python code was written to calculate the average growth percentage of the company stock over a 335 day period, from June 1st 2019 to May 1st 2020. Companies with no ticker symbols are filtered out. The below figure describes the top 5 companies with the highest average growth percentage in descending order.

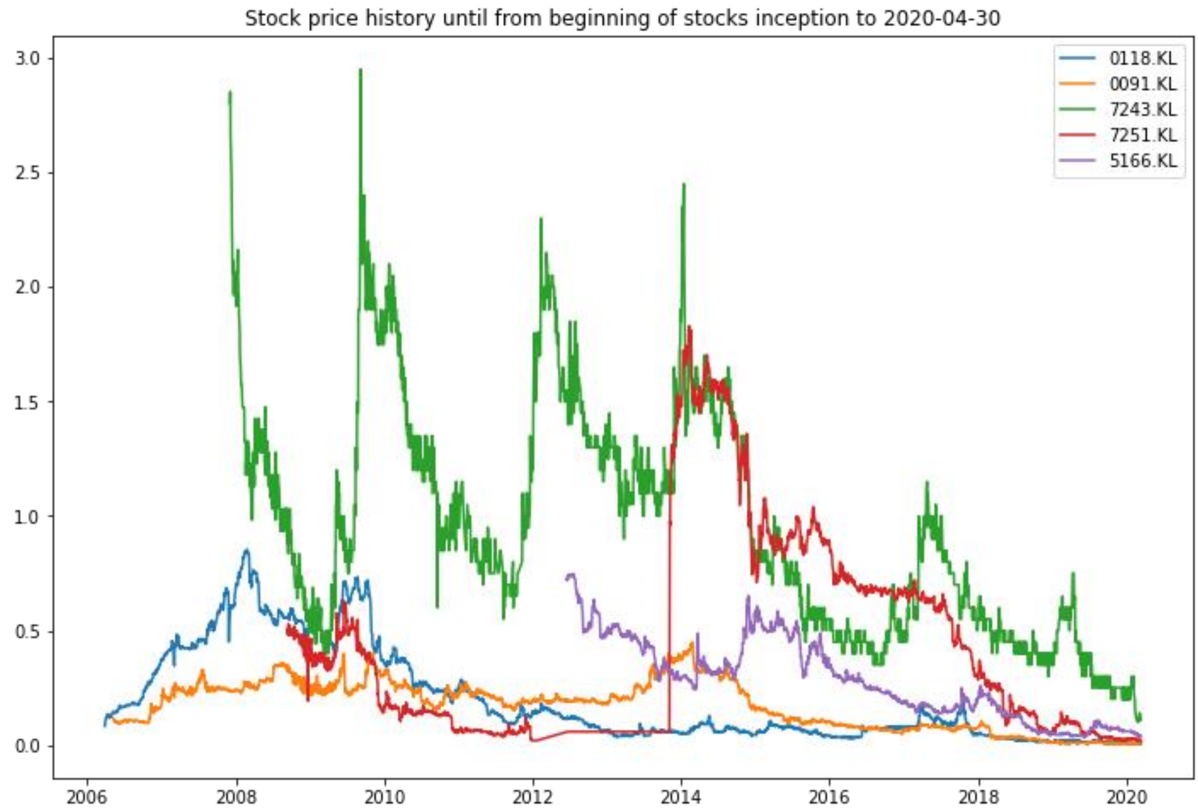
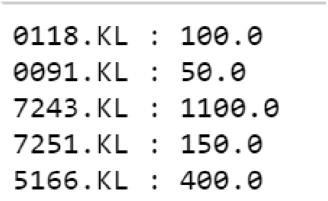


Figure 2: Top five stock shares visualized

Ticker	Average Growth Percentage
0118.KL (TRIVE)	11.79%
0091.KL (DAYA)	7.01%
7243.KL (IMPIANA)	3.31%
7251.KL (BARAKAH)	1.65%
5166.KL (MINDA)	1.63%

Table 1: Average growth percentage of top five stock shares.

It can be concluded that the stock share, 0118.KL, had the highest growth, followed by 0091.KL with 7.01%. The stock with the lowest growth percentage is 5166.KL with a growth of 1.63% over a 335 day period. Furthermore, the total capital invested on the 1st May 2020 on market open is RM1800.



```
0118.KL : 100.0
0091.KL : 50.0
7243.KL : 1100.0
7251.KL : 150.0
5166.KL : 400.0
```

Figure 3: Distribution of amount invested across the stocks

7343.KL makes up the majority investment at RM1100 followed by 5166.KL at RM400. The least invested stock is 0091.KL at RM50.

c. Approach (description of models used and why)

The model chosen to forecast the stock price is the SARIMA model, but we did initially use the ARIMA model. SARIMA is short for Seasonal Autoregressive Integrated Moving Average. It is used to predict future time series data based on its past values with a seasonal component (Brownlee 2018). Whereas, ARIMA is the non-seasonal version (Malik 2018).

Time series data is data that is indexed in a timely order. For example, today's date can be used as a unique index. This format fits the stock data, given that it has a timestamp as an index and a closing price as a feature.

We are specifically building a univariate model since there is only one feature, the adjusted close price of the stock share. The model will try to learn a pattern in the trend of the data and use it to forecast the stock price in the future. It can be seen that the predicted data (green line) fits the test data (orange line) in the figure below. It is able to capture the trend of the data based on the previous data.

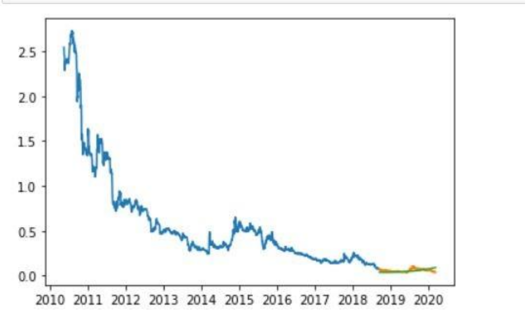


Figure 4: Prediction of the stock 0118.KL using ARIMA

Experiments

Training and Test

SARIMA

For SARIMA, we used up all the dataset as a training set. Thus, we do not have a test set. The reason for this is because we are using Akaike's Information Criterion (AIC) to evaluate our model and the seasonal nature of SARIMA. For a time-series analysis especially for SARIMA, AIC is very useful as the time series most valuable data is mostly the most recent data, Therefore, the most recent data is always stuck in the validation set and test set. Thus, our SARIMA model is using all the dataset as a training set and using AIC will improve the model selection when compared to the traditional train/test model selection methods. SARIMA performs better than ARIMA. Thus, to test the forecasting on the data set, we had to set the time period on when to perform forecasting, the start date and end date.

ARIMA

The test data was prioritized. The dataset for the test data is 366 data points.
The training data is the total amount of data minus the test data size, this varies stock to stock. The test data is the last 366 data points in the dataset, this is constant for all stocks.

Configuring Hyperparameters

SARIMA

When it comes to configuring SARIMA, there is a need to select hyperparameters for the trend and seasonal elements of the series. (Brownlee, 2018)

Trend Elements:

Parameters	Description	How to interpret
p	Trend autoregression order	If p = 1 in a monthly cycle, helps to understand whether January data will have influence on February.
d	Trend difference order	It is the level of integration for the process. It means how many times the time series operator should be applied to have the time series to become stationary
q	Trend moving average order	The number of prior noises (errors) that affect the current value

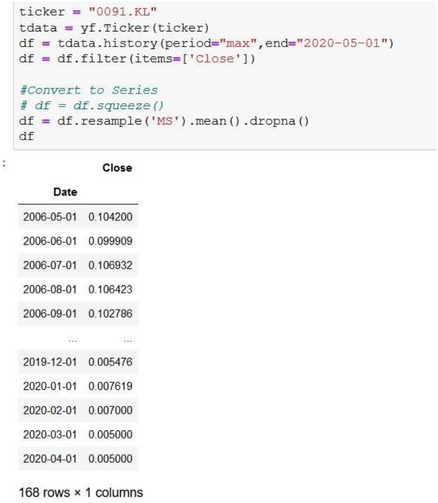
Table 2: The trend or non-seasonal elements

Seasonal Elements:

Parameters	Description
P	Seasonal autoregressive order
D	Seasonal difference order
Q	Seasonal moving average
m	The number of time steps for a single seasonal period

Table 3: The seasonal elements

The number m that is determined will affect the seasonal elements P, D, Q. The values for m can be 12 for monthly data is monthly seasonal cycle, 52 for weekly seasonal cycle, and 365 for daily seasonal cycle. For our SARIMA modelling, we used m = 12. As our stock ticker data is daily, we had to resample it to monthly so that it can handle variations in the data so that the model can train on the various patterns in the dataset. This is because when we tried m = 365 for a daily seasonal cycle, the changes in the patterns is too minute in which it causes overfitting. Thus, we want to get the monthly predicted adjusted closing price for a year starting from 2020-05-01 to 2021-05-01.

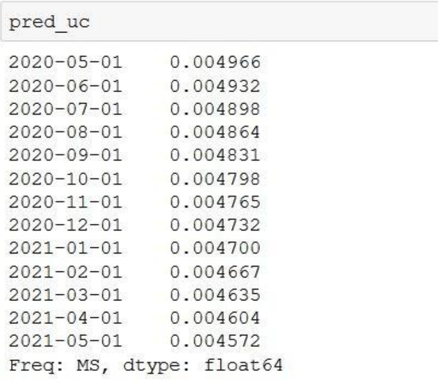


```
ticker = "0091.KL"
tdata = yf.Ticker(ticker)
df = tdata.history(period="max",end="2020-05-01")
df = df.filter(items=['Close'])

#Convert to Series
# df = df.squeeze()
df = df.resample('MS').mean().dropna()
df
```

	Close
Date	
2006-05-01	0.104200
2006-06-01	0.099909
2006-07-01	0.106932
2006-08-01	0.106423
2006-09-01	0.102786
...	...
2019-12-01	0.005476
2020-01-01	0.007619
2020-02-01	0.007000
2020-03-01	0.005000
2020-04-01	0.005000

168 rows × 1 columns

Figure 5: Resampling of the daily stock prices into monthly.



```
pred_uc
```

```
2020-05-01    0.004966
2020-06-01    0.004932
2020-07-01    0.004898
2020-08-01    0.004864
2020-09-01    0.004831
2020-10-01    0.004798
2020-11-01    0.004765
2020-12-01    0.004732
2021-01-01    0.004700
2021-02-01    0.004667
2021-03-01    0.004635
2021-04-01    0.004604
2021-05-01    0.004572
Freq: MS, dtype: float64
```

Figure 6: The monthly predicted adjusted closing price from 2020-05-01 to 2021-05-01.

ARIMA

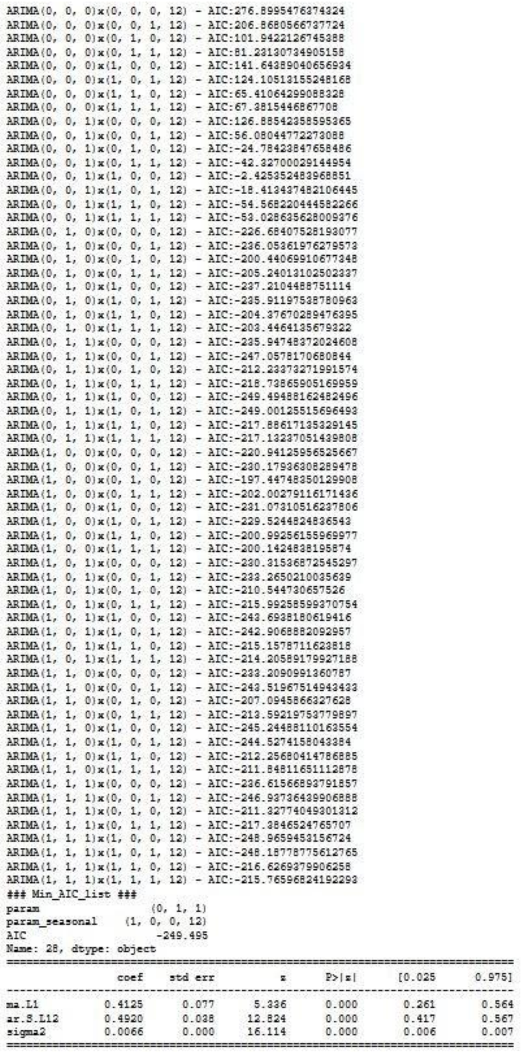
ARIMA is the same as SARIMA, except it does not have a seasonal component attached to it. It only has the trend component. Hyperparameters are automatically found using auto_arima which was provided by a library called pmdarima. This was used to save time finding the p,d,q combo that provides us the lowest AIC value. Instead of doing a monthly forecast, ARIMA was tasked with doing a forecast based on the daily adjusted close prices of the stock. The results were not as good as the SARIMA variant and therefore discarded in favour of SARIMA.

Assessment and Validation

Finding the best combinations of the hyperparameters for ARIMA (trend elements only) and SARIMA (trend and seasonal elements) using AIC

In order to find the best combinations of the hyperparameters, we used Akaike's Information Criterion (AIC) to evaluate the quality of each model to fit on the training data in relation to each other and tried to find the models with the lowest AIC values. This is to select the best performing model. One thing to note about AIC is the assumption that the same data is used between all the models.

Basically, in our SARIMA model we set the range of p = d = q to (0, 2) so it will try out all the possible combinations for both the trend elements and the seasonal elements. Thus, during the evaluation phase, all possible combinations of pdq are used to find the models with the lowest AIC value as the lowest AIC is equivalent to the best model.



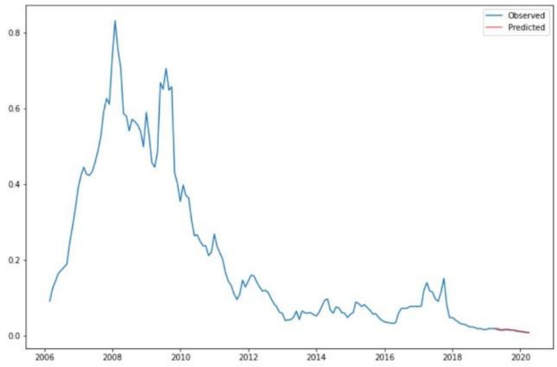
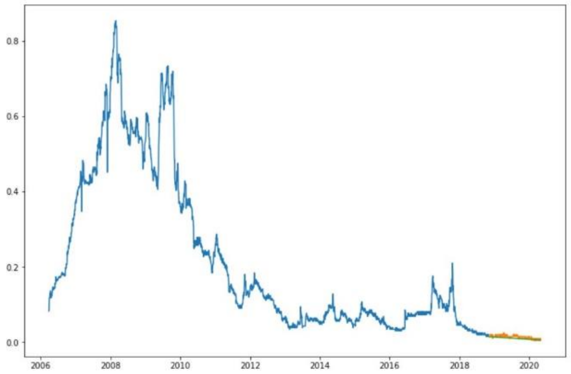
```
ARIMA(0, 0, 0)x(0, 0, 0, 12) - AIC:276.8995476374324
ARIMA(0, 0, 0)x(0, 0, 1, 12) - AIC:206.8680566737724
ARIMA(0, 0, 0)x(0, 1, 0, 12) - AIC:101.9422126745388
ARIMA(0, 0, 0)x(0, 1, 1, 12) - AIC:81.23130734905158
ARIMA(0, 0, 0)x(1, 0, 0, 12) - AIC:141.64389040656934
ARIMA(0, 0, 0)x(1, 0, 1, 12) - AIC:124.10513155248168
ARIMA(0, 0, 0)x(1, 1, 0, 12) - AIC:65.41064299088328
ARIMA(0, 0, 0)x(1, 1, 1, 12) - AIC:67.3815446867708
ARIMA(0, 0, 1)x(0, 0, 0, 12) - AIC:126.88542358595365
ARIMA(0, 0, 1)x(0, 0, 1, 12) - AIC:56.08044772273088
ARIMA(0, 0, 1)x(0, 1, 0, 12) - AIC:-24.78423847658486
ARIMA(0, 0, 1)x(0, 1, 1, 12) - AIC:-42.32700029144954
ARIMA(0, 0, 1)x(1, 0, 0, 12) - AIC:-2.425352483968851
ARIMA(0, 0, 1)x(1, 0, 1, 12) - AIC:-18.413437482106445
ARIMA(0, 0, 1)x(1, 1, 0, 12) - AIC:-54.568220444582266
ARIMA(0, 0, 1)x(1, 1, 1, 12) - AIC:-53.028635628009376
ARIMA(0, 1, 0)x(0, 0, 0, 12) - AIC:-226.68407528193077
ARIMA(0, 1, 0)x(0, 0, 1, 12) - AIC:-236.05361976279573
ARIMA(0, 1, 0)x(0, 1, 0, 12) - AIC:-200.44069910677348
ARIMA(0, 1, 0)x(0, 1, 1, 12) - AIC:-205.24013102502337
ARIMA(0, 1, 0)x(1, 0, 0, 12) - AIC:-237.2104488751114
ARIMA(0, 1, 0)x(1, 0, 1, 12) - AIC:-235.91197538780963
ARIMA(0, 1, 0)x(1, 1, 0, 12) - AIC:-204.37670289476395
ARIMA(0, 1, 0)x(1, 1, 1, 12) - AIC:-203.4464135679322
ARIMA(0, 1, 1)x(0, 0, 0, 12) - AIC:-235.94748372024608
ARIMA(0, 1, 1)x(0, 0, 1, 12) - AIC:-247.0578170680844
ARIMA(0, 1, 1)x(0, 1, 0, 12) - AIC:-212.23373271991574
ARIMA(0, 1, 1)x(0, 1, 1, 12) - AIC:-218.73865905169959
ARIMA(0, 1, 1)x(1, 0, 0, 12) - AIC:-249.49488162482496
ARIMA(0, 1, 1)x(1, 0, 1, 12) - AIC:-249.00125515696493
ARIMA(0, 1, 1)x(1, 1, 0, 12) - AIC:-217.88617135329145
ARIMA(0, 1, 1)x(1, 1, 1, 12) - AIC:-217.13237051439808
ARIMA(1, 0, 0)x(0, 0, 0, 12) - AIC:-220.94125956525667
ARIMA(1, 0, 0)x(0, 0, 1, 12) - AIC:-230.17936308289478
ARIMA(1, 0, 0)x(0, 1, 0, 12) - AIC:-197.44748350129908
ARIMA(1, 0, 0)x(0, 1, 1, 12) - AIC:-202.00279116171436
ARIMA(1, 0, 0)x(1, 0, 0, 12) - AIC:-231.07310516237806
ARIMA(1, 0, 0)x(1, 0, 1, 12) - AIC:-229.5244824836543
ARIMA(1, 0, 0)x(1, 1, 0, 12) - AIC:-200.99256155969977
ARIMA(1, 0, 0)x(1, 1, 1, 12) - AIC:-200.1424838195874
ARIMA(1, 0, 1)x(0, 0, 0, 12) - AIC:-230.31536872545297
ARIMA(1, 0, 1)x(0, 0, 1, 12) - AIC:-233.2650210035639
ARIMA(1, 0, 1)x(0, 1, 0, 12) - AIC:-210.544730657526
ARIMA(1, 0, 1)x(0, 1, 1, 12) - AIC:-215.99258599370754
ARIMA(1, 0, 1)x(1, 0, 0, 12) - AIC:-243.6938180619416
ARIMA(1, 0, 1)x(1, 0, 1, 12) - AIC:-242.9068882092957
ARIMA(1, 0, 1)x(1, 1, 0, 12) - AIC:-215.1578711623818
ARIMA(1, 0, 1)x(1, 1, 1, 12) - AIC:-214.20589179927188
ARIMA(1, 1, 0)x(0, 0, 0, 12) - AIC:-233.2090991360787
ARIMA(1, 1, 0)x(0, 0, 1, 12) - AIC:-243.51967514943433
ARIMA(1, 1, 0)x(0, 1, 0, 12) - AIC:-207.0945866327628
ARIMA(1, 1, 0)x(0, 1, 1, 12) - AIC:-213.59219753779897
ARIMA(1, 1, 0)x(1, 0, 0, 12) - AIC:-245.24488110163554
ARIMA(1, 1, 0)x(1, 0, 1, 12) - AIC:-244.5274158043384
ARIMA(1, 1, 0)x(1, 1, 0, 12) - AIC:-212.25680414786885
ARIMA(1, 1, 0)x(1, 1, 1, 12) - AIC:-211.84811651112878
ARIMA(1, 1, 1)x(0, 0, 0, 12) - AIC:-236.61566893791857
ARIMA(1, 1, 1)x(0, 0, 1, 12) - AIC:-246.93736439906888
ARIMA(1, 1, 1)x(0, 1, 0, 12) - AIC:-211.32774049301312
ARIMA(1, 1, 1)x(0, 1, 1, 12) - AIC:-217.3846524765707
ARIMA(1, 1, 1)x(1, 0, 0, 12) - AIC:-248.9659453156724
ARIMA(1, 1, 1)x(1, 0, 1, 12) - AIC:-248.18778775612765
ARIMA(1, 1, 1)x(1, 1, 0, 12) - AIC:-216.6269379906258
ARIMA(1, 1, 1)x(1, 1, 1, 12) - AIC:-215.76596824192293
### Min_AIC_list ###
param                    (0, 1, 1)
param_seasonal       (1, 0, 0, 12)
AIC                       -249.495
Name: 28, dtype: object
==============================================================================
                 coef    std err          z      P>|z|      [0.025      0.975]
------------------------------------------------------------------------------
ma.L1          0.4125      0.077      5.336      0.000       0.261       0.564
ar.S.L12       0.4920      0.038     12.824      0.000       0.417       0.567
sigma2         0.0066      0.000     16.114      0.000       0.006       0.007
==============================================================================
```

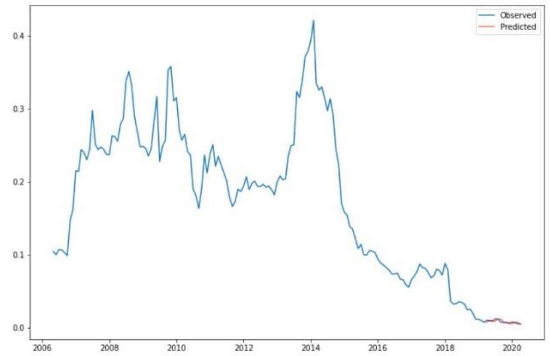
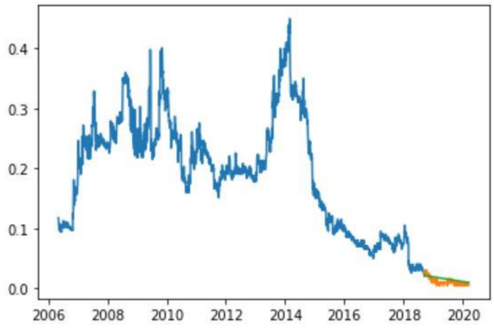
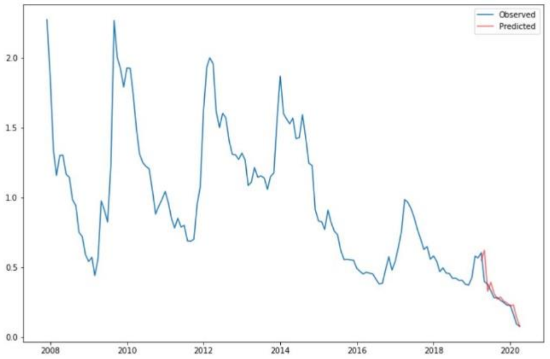
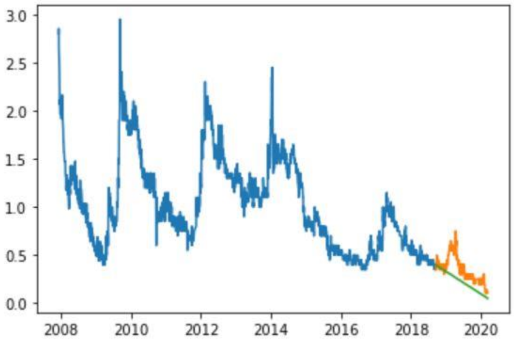
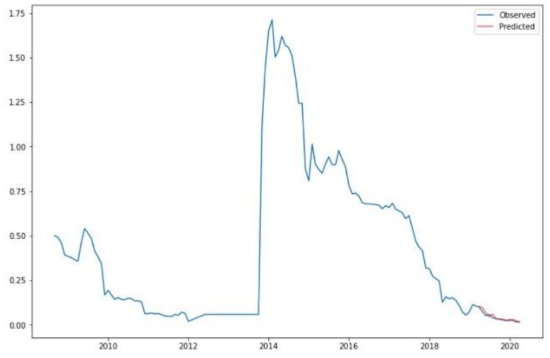
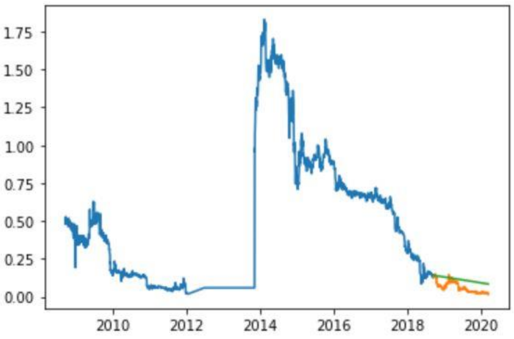
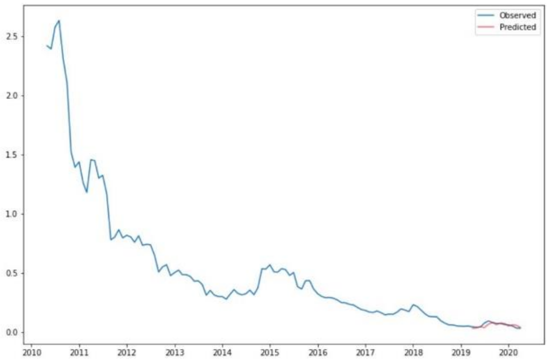
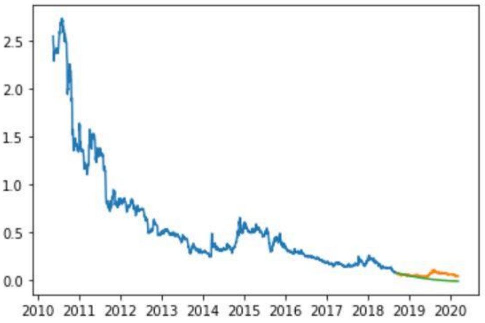
Figure 7: Best Combinations that produce lowest, AIC Order = (1, 0, 0) Seasonal Order = (0, 0, 0, 12)

MSE (Mean Squared Error)

MSE is used to evaluate the quality of the model’s performance or predictive ability. It measures the average of the square of errors which is the average squared differences of the predicted values and actual values. Ideally, the smaller the value of MSE, the better the performance or predictive ability of the model as the predicted values are closer to the actual values.

Ticker	MSE (SARIMA)	MSE (ARIMA)
0118.KL	2.295629745743182e-06	2.4450253737399626e-05
0091.KL	3.7359765977477315e-06	3.74948581999268e-05
7243.KL	0.005269762069368535	0.026308348334562633
7251.KL	0.0001232976159704131	0.0032625956321725275
5166.KL	0.0002554418795987717	0.0025258323757062374

SARIMA	**ARIMA** (Green is predicted value)
 0118.KL	

 0091.KL	
 7243.KL	
 7251.KL	
 5166.KL	

Based on the evaluation of MSE for both SARIMA and ARIMA, we can evaluate that SARIMA models outperforms the ARIMA models, so we decided to utilise SARIMA model as the final model to calculate the predicted adjusted closing price.

Dynamic (ARIMA) Vs One Step Forecasting (SARIMA)

For our ARIMA model, we used dynamic forecasting whereas the SARIMA model uses one step forecasting. Dynamic forecasting uses the value of the previous forecasted value of the dependent variable to compute the next one. One step forecasting uses the actual value for each subsequent forecast. Below is the forecasting.

Examples of dynamic forecasting and one step forecasting:

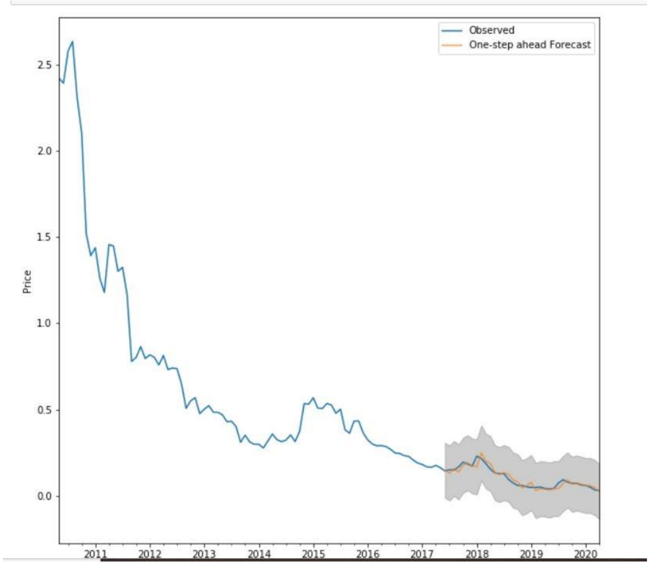


Figure 8: One Step Forecasting in SARIMA model

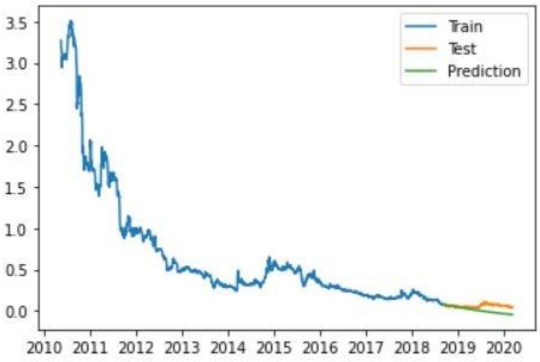


Figure 9: Dynamic Forecasting in the ARIMA model

In terms of prediction accuracy, one step accuracy has a better accuracy as seen in an example graph below that shows the prediction error graph between both types of forecasting.

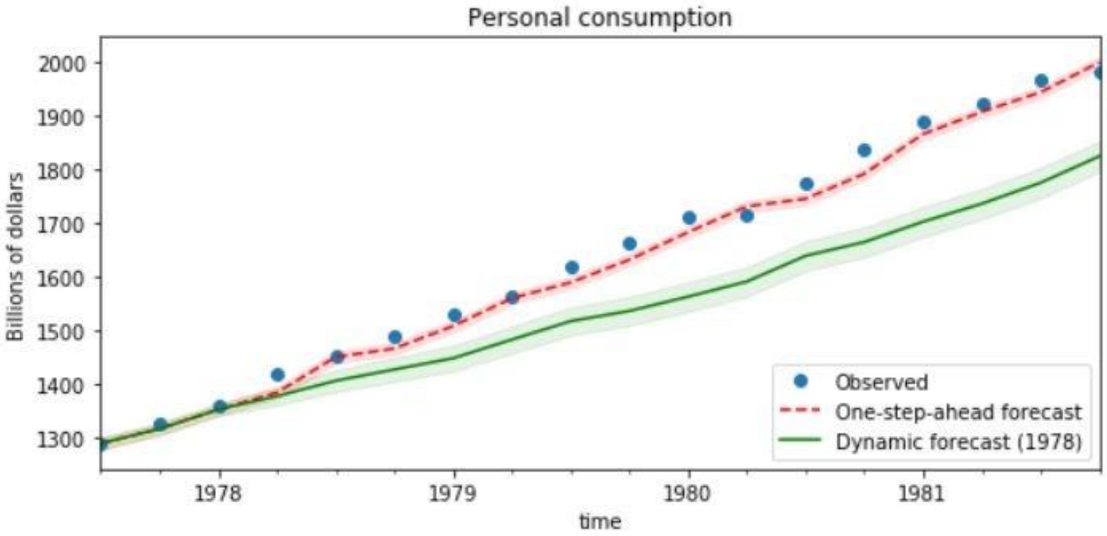


Figure 10: Comparison between one-step ahead and dynamic forecasting in prediction

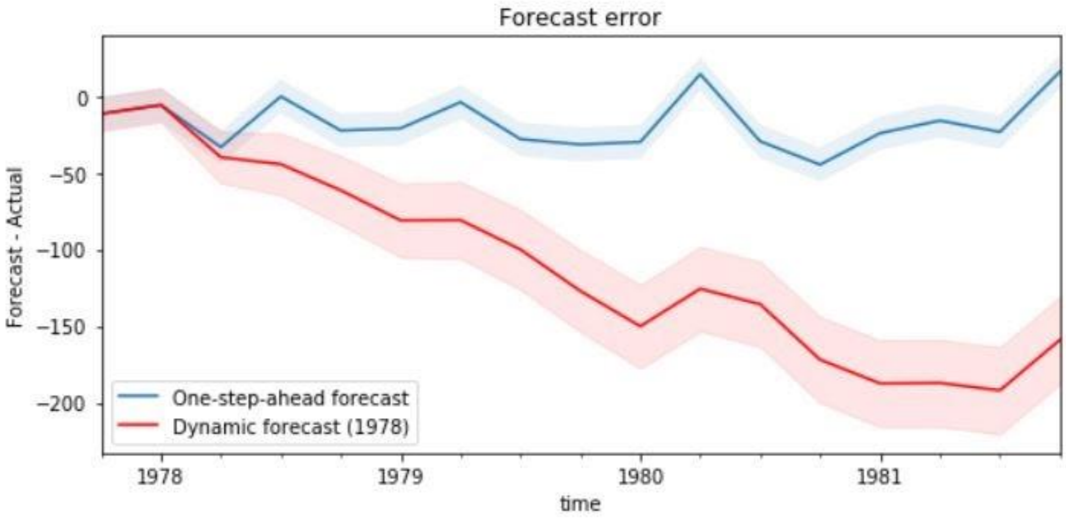
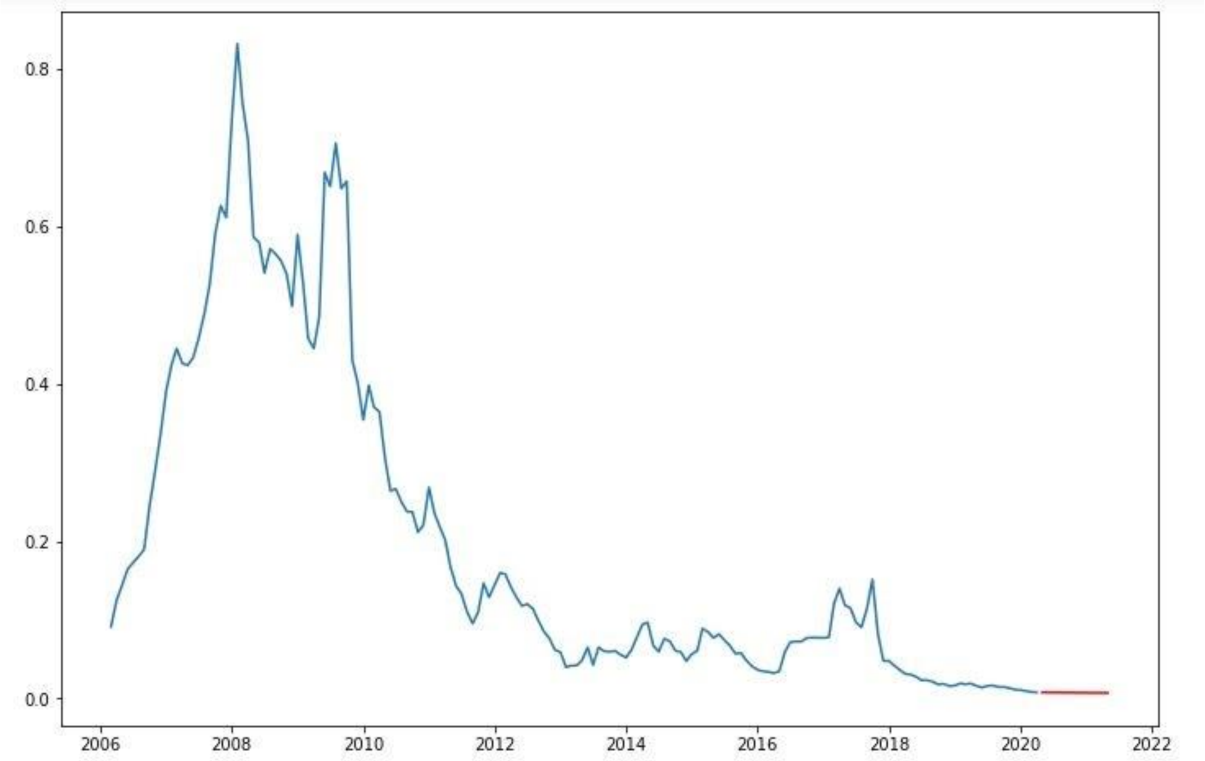


Figure 11: Forecast error between one-step ahead and dynamic forecasting

As seen here, one step forecasting performs better than dynamic forecasting.

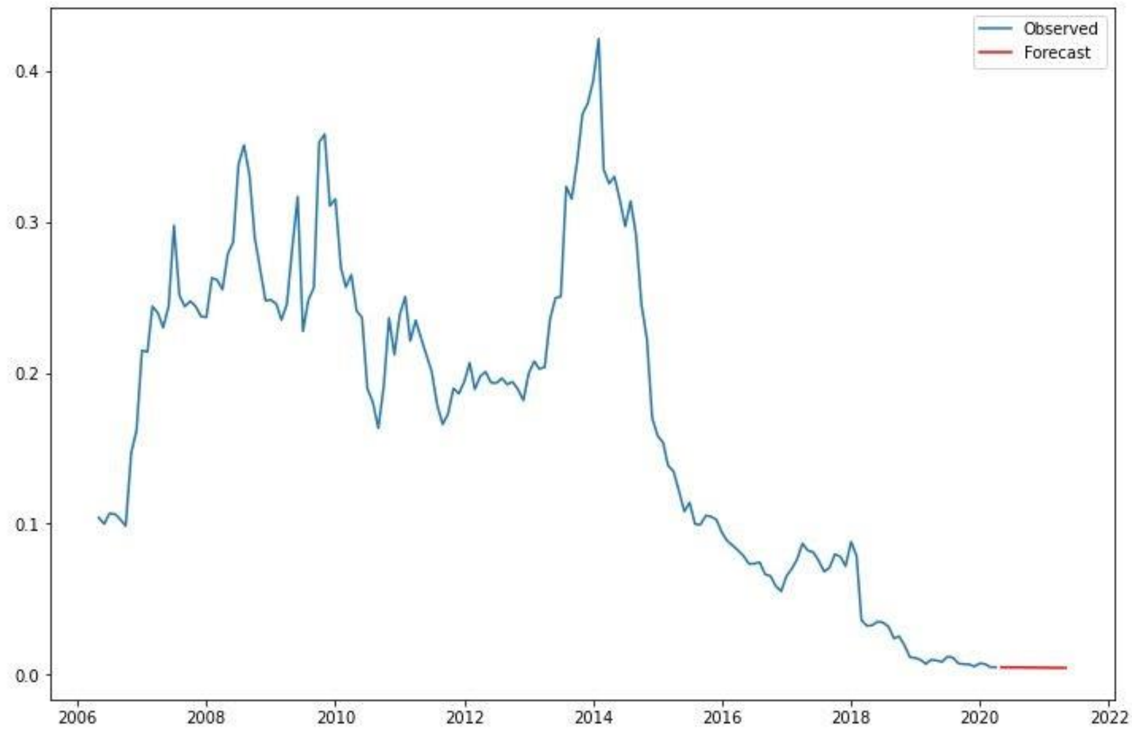
Results for SARIMA (Chosen as Final Model)

0118.KL



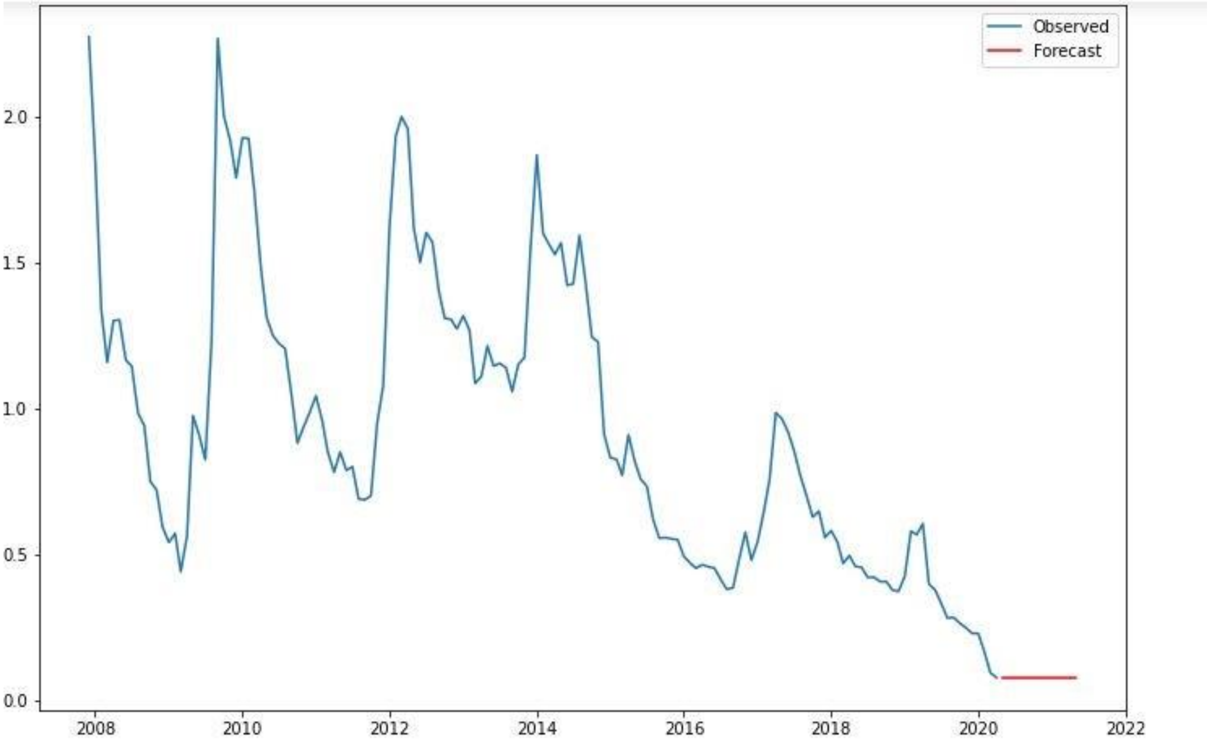
2020-05-01	0.007662
2020-06-01	0.007598
2020-07-01	0.007534
2020-08-01	0.007470
2020-09-01	0.007407
2020-10-01	0.007345
2020-11-01	0.007283
2020-12-01	0.007222
2021-01-01	0.007161
2021-02-01	0.007101
2021-03-01	0.007041
2021-04-01	0.006982
2021-05-01	0.006923

0091.KL



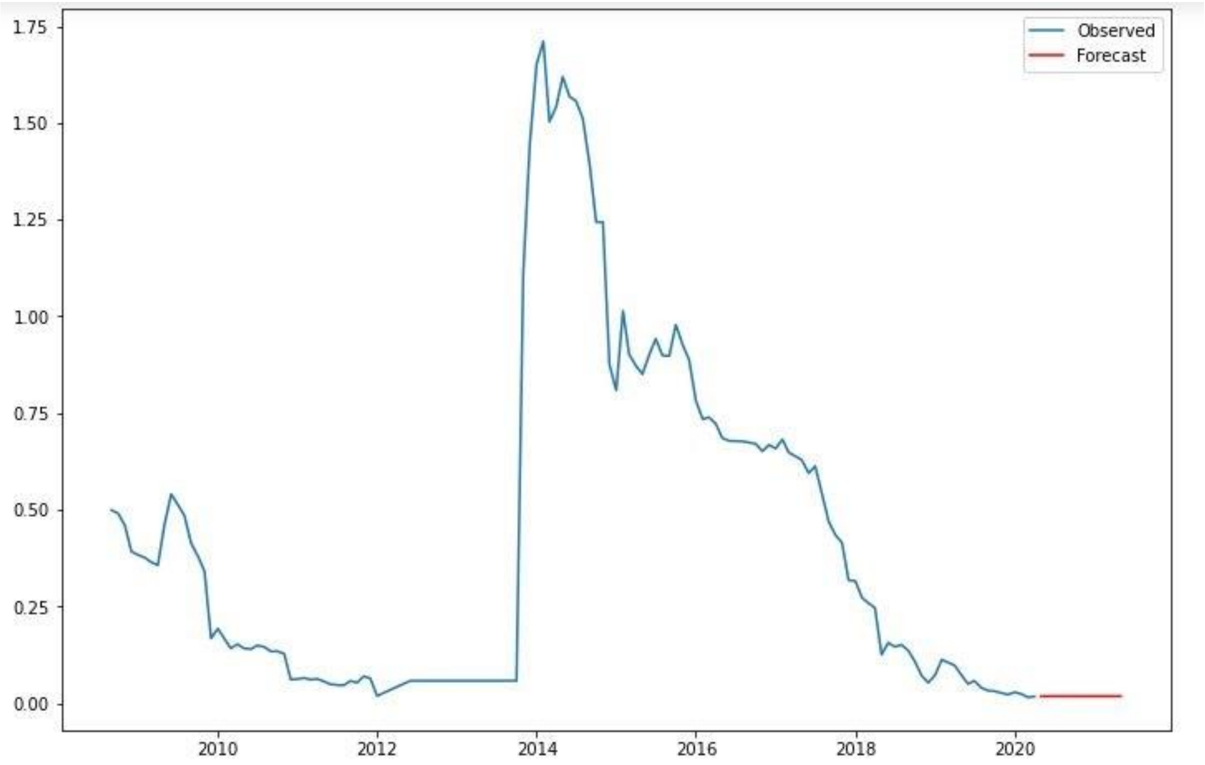
2020-05-01	0.004966
2020-06-01	0.004932
2020-07-01	0.004898
2020-08-01	0.004864
2020-09-01	0.004831
2020-10-01	0.004798
2020-11-01	0.004765
2020-12-01	0.004732
2021-01-01	0.004700
2021-02-01	0.004667
2021-03-01	0.004635
2021-04-01	0.004604
2021-05-01	0.004572

7243.KL



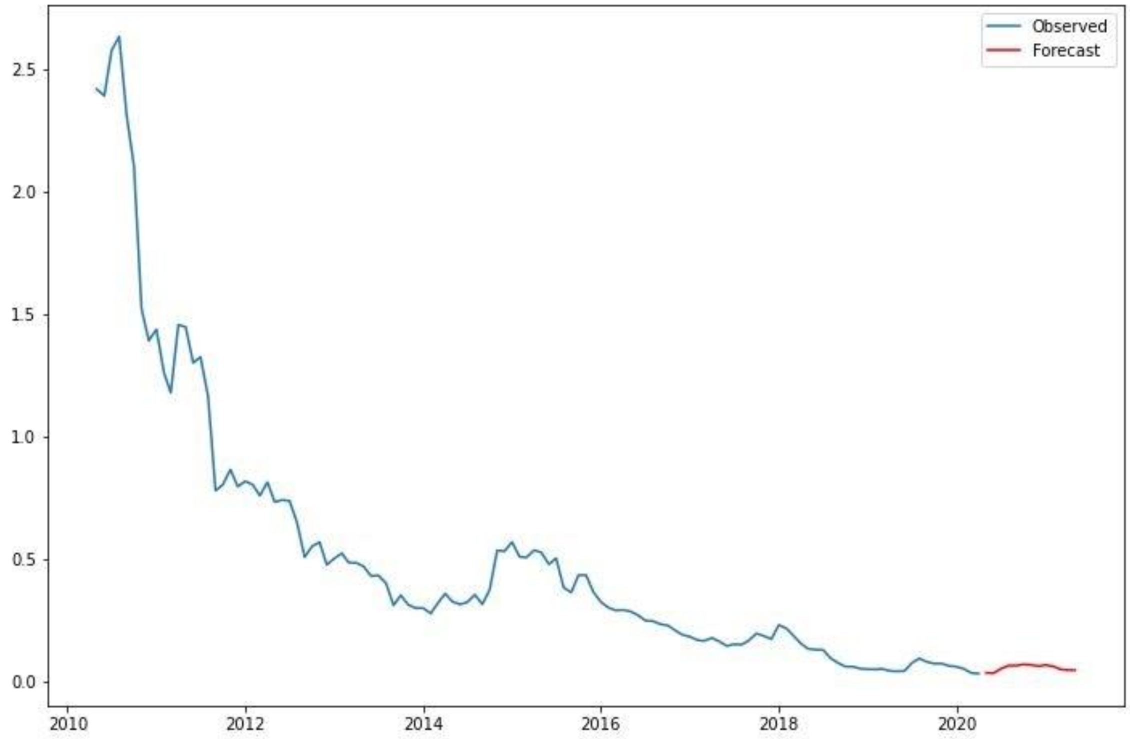
2020-05-01	0.076104
2020-06-01	0.076104
2020-07-01	0.076104
2020-08-01	0.076104
2020-09-01	0.076104
2020-10-01	0.076104
2020-11-01	0.076104
2020-12-01	0.076104
2021-01-01	0.076104
2021-02-01	0.076104
2021-03-01	0.076104
2021-04-01	0.076104
2021-05-01	0.076104

7251.KL



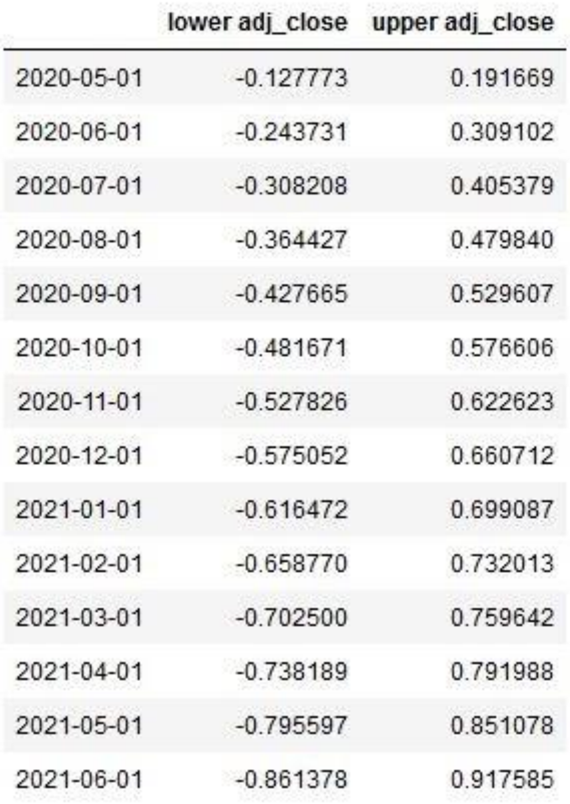
2020-05-01	0.018723
2020-06-01	0.018866
2020-07-01	0.018904
2020-08-01	0.018914
2020-09-01	0.018917
2020-10-01	0.018918
2020-11-01	0.018918
2020-12-01	0.018918
2021-01-01	0.018918
2021-02-01	0.018918
2021-03-01	0.018918
2021-04-01	0.018918
2021-05-01	0.018918

5166.KL



2020-05-01	0.034108
2020-06-01	0.032822
2020-07-01	0.050769
2020-08-01	0.063683
2020-09-01	0.063834
2020-10-01	0.068307
2020-11-01	0.066428
2020-12-01	0.061676
2021-01-01	0.065957
2021-02-01	0.060662
2021-03-01	0.049224
2021-04-01	0.045350
2021-05-01	0.045222

The prediction values are calculated from the average of the predicted lower adjusted close price and upper adjusted close price. Thus, the predicted values will not go lower than 0. As seen in the graphs, we can see the stock prices has a downward trend.



	lower adj_close	upper adj_close
2020-05-01	-0.127773	0.191669
2020-06-01	-0.243731	0.309102
2020-07-01	-0.308208	0.405379
2020-08-01	-0.364427	0.479840
2020-09-01	-0.427665	0.529607
2020-10-01	-0.481671	0.576606
2020-11-01	-0.527826	0.622623
2020-12-01	-0.575052	0.660712
2021-01-01	-0.616472	0.699087
2021-02-01	-0.658770	0.732013
2021-03-01	-0.702500	0.759642
2021-04-01	-0.738189	0.791988
2021-05-01	-0.795597	0.851078
2021-06-01	-0.861378	0.917585

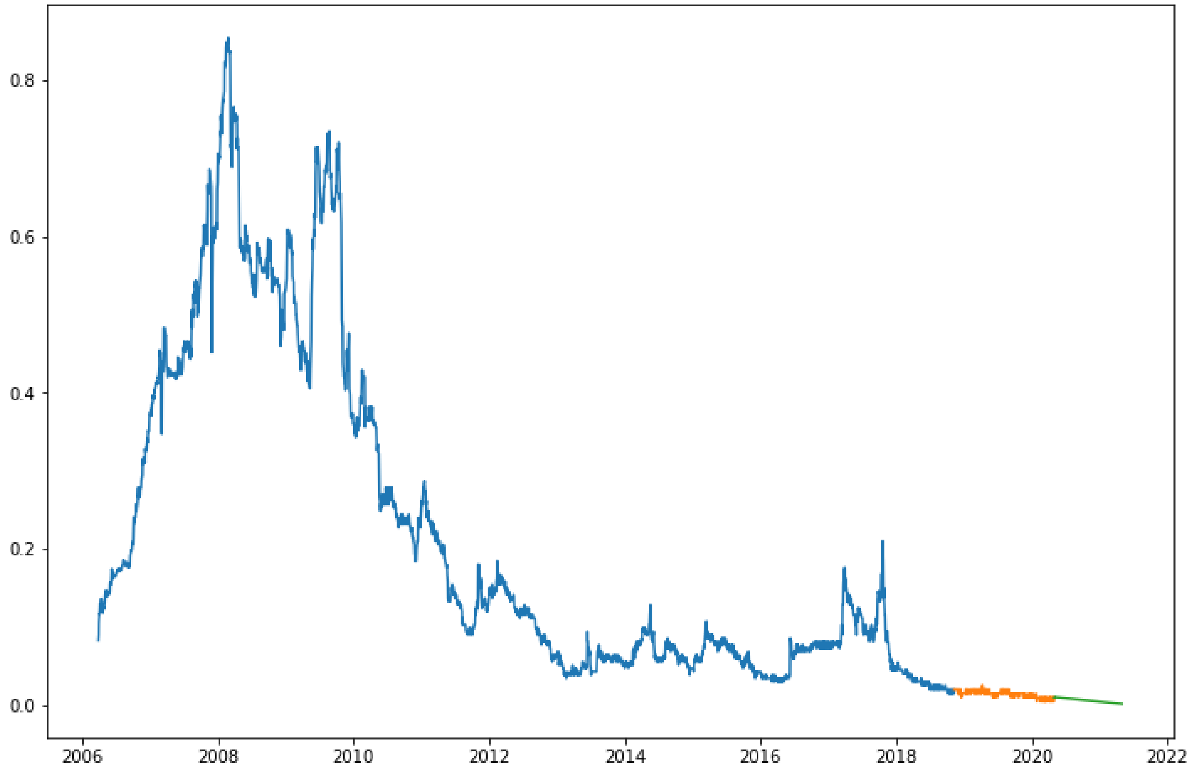
SARIMA model predictions for final adjustment close

Ticker	Final Price (Ringgit)
0118.KL	0.006923
0091.KL	0.004572
7243.KL	0.076104
7251.KL	0.018918
5166.KL	0.045222

Table 4: Final price for the stocks using SARIMA

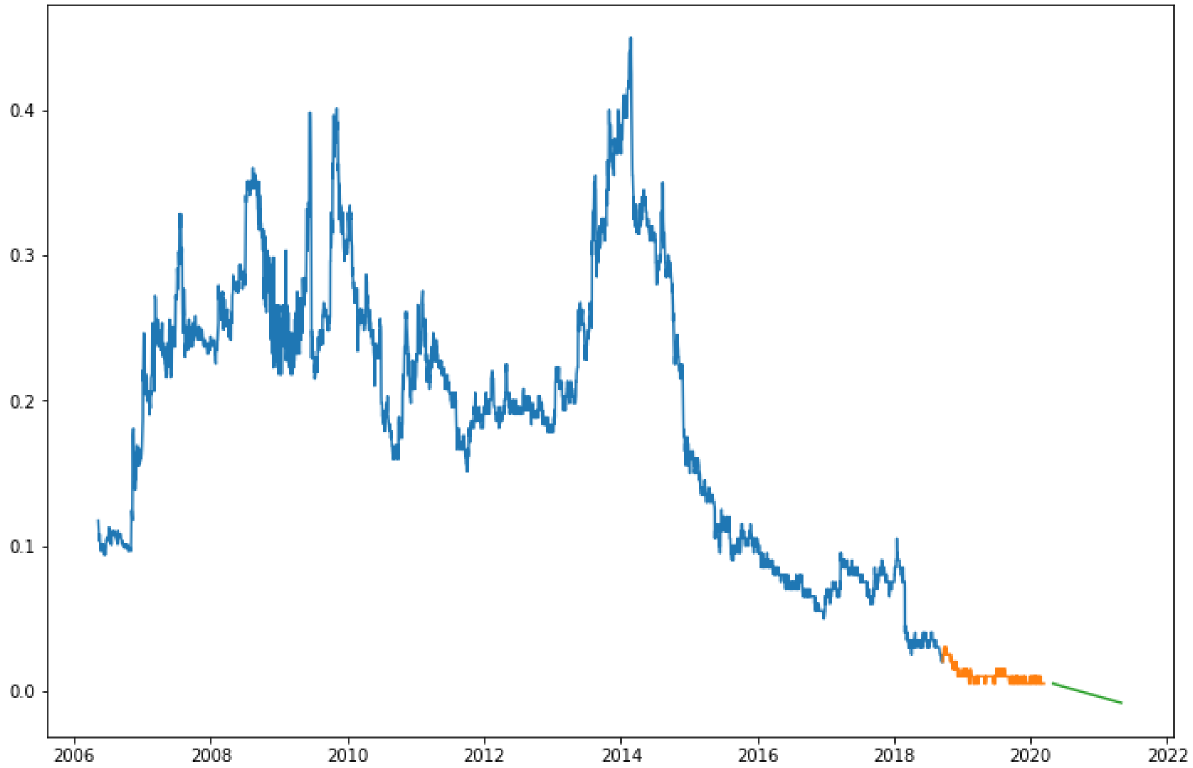
Results for ARIMA (Initial Model)

0118.KL



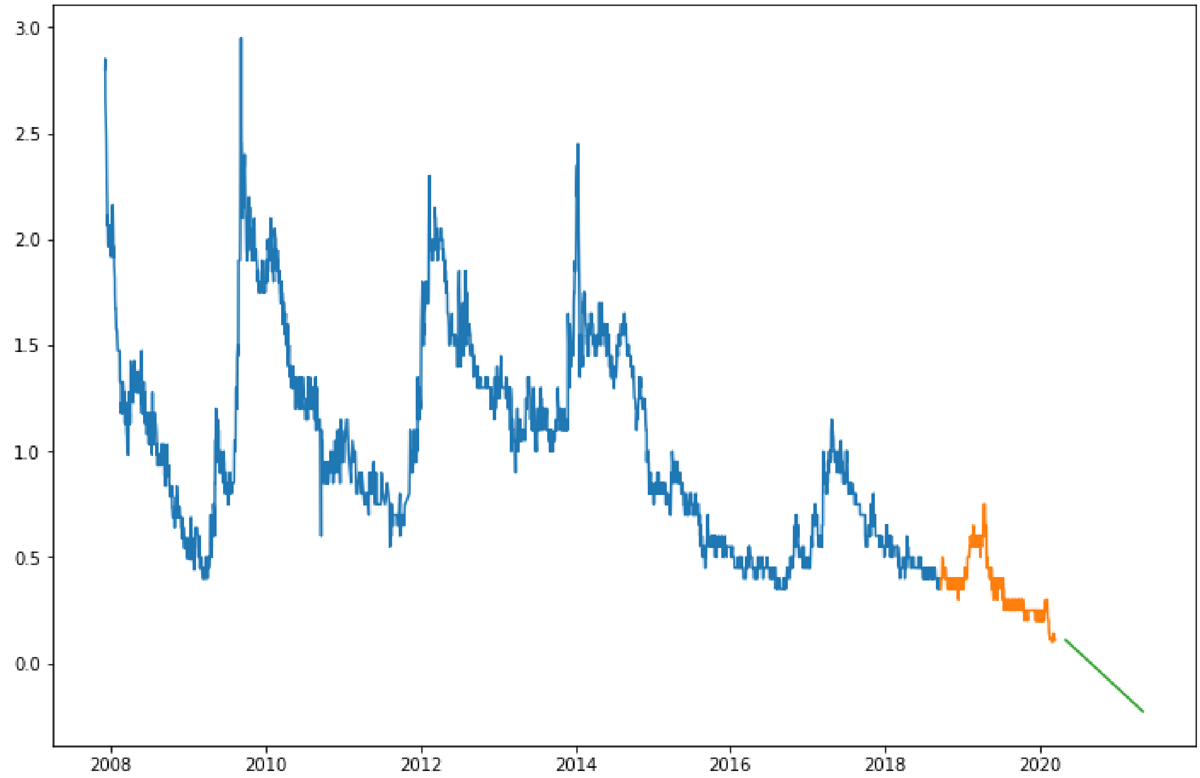
2021-04-28	0.001581
2021-04-29	0.001557
2021-04-30	0.001532
2021-05-01	0.001507
2021-05-02	0.001483

0091.KL



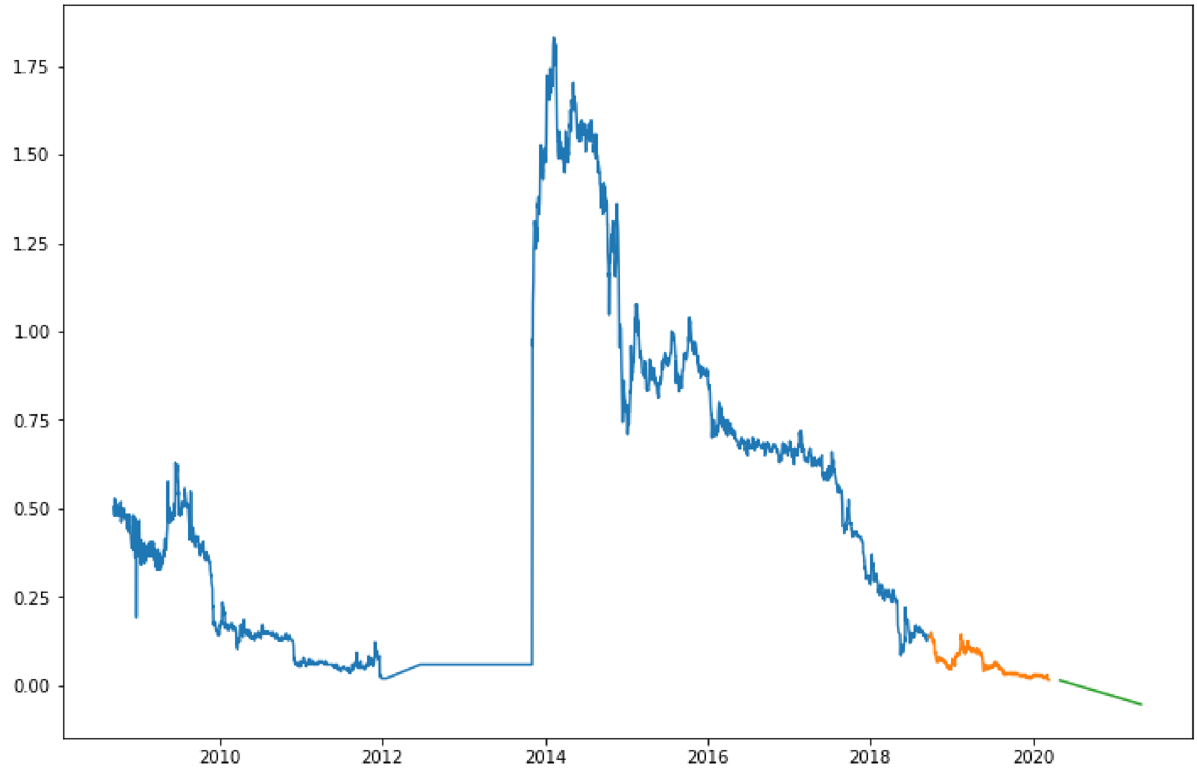
2021-04-27	-0.008198
2021-04-28	-0.008234
2021-04-29	-0.008271
2021-04-30	-0.008307
2021-05-01	-0.008344

7243.KL



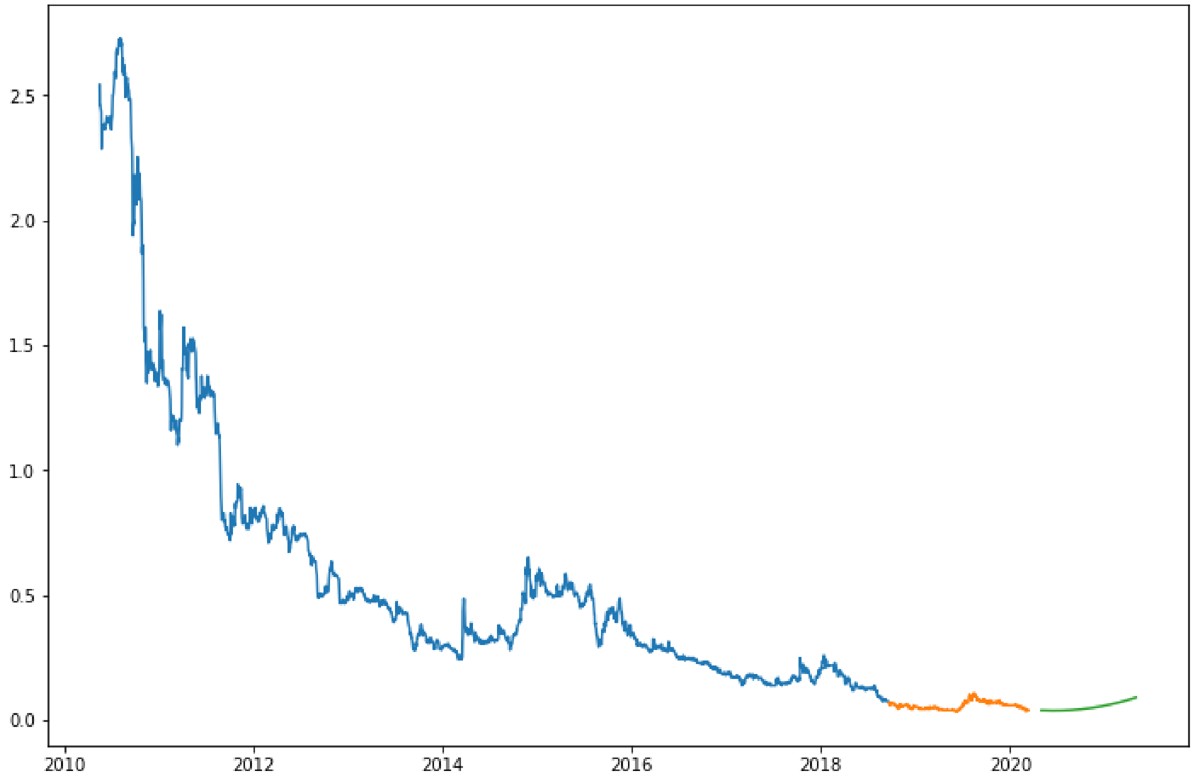
2021-04-27	-0.224594
2021-04-28	-0.225521
2021-04-29	-0.226447
2021-04-30	-0.227373
2021-05-01	-0.228300

7251.KL



2021-04-27	-0.052892
2021-04-28	-0.053078
2021-04-29	-0.053264
2021-04-30	-0.053450
2021-05-01	-0.053636

5166.KL



2021-04-27	0.089757
2021-04-28	0.090086
2021-04-29	0.090416
2021-04-30	0.090747
2021-05-01	0.091079

As seen in the graph there is a downward trend for the following stocks 0118.KL, 0091.KL, 7243.KL, and 7251.KL. However, for 5166.KL, there is a rebound as it trends upward slightly. However, the other downward stocks go below zero. In reality, stock prices cannot go below zero so the ARIMA models has a difficulty in trying to capture the various patterns on the training data and is rather erratic which results in a bigger MSE value. This can be seen in ARIMA graphs in which the green line which is the predicted value does not really capture the trend of the test set. As seen in the MSE table above. One of the possible problems is also the ARIMA model using the dynamic forecasting which has a larger prediction error rate.

ARIMA model predictions for final adjustment close

Ticker	Final Price (Ringgit)
0118.KL	0.001507
0091.KL	-0.008344
7243.KL	-0.228300
7251.KL	-0.053636
5166.KL	0.091079

Table 5: Final price for the stocks using ARIMA

Discussions

Assumptions

- The data for each stock share is complete and has no missing data.

- The stock performance was measured using its average growth percentage
- We do not have the full understanding to build our own models and require other peoples' written and produced models. Therefore, our focus is on the comprehension of how the data is trained, forecasted and the parameters that govern how well the model will perform.
- SARIMA uses all the observable data available as a training set and thus there is no true train and test split.
- The stocks that we used are Malaysian stocks, so all currency used in the stocks is under the Malaysian Ringgit.

Weakness

- ARIMA model requires p,d,q parameters (Pietro 2018 & ahajib 2016).
    - This is solved using auto_arima

- ARIMA is worst for large datasets, but since our datasets are small, number in the thousands. The model will not be affected (Priya 2015).

- SARIMA requires p,d,q parameters as well as P,D,Q,s parameters. Time and processing power were spent deriving the optimal parameters for the SARIMA for every stock share.

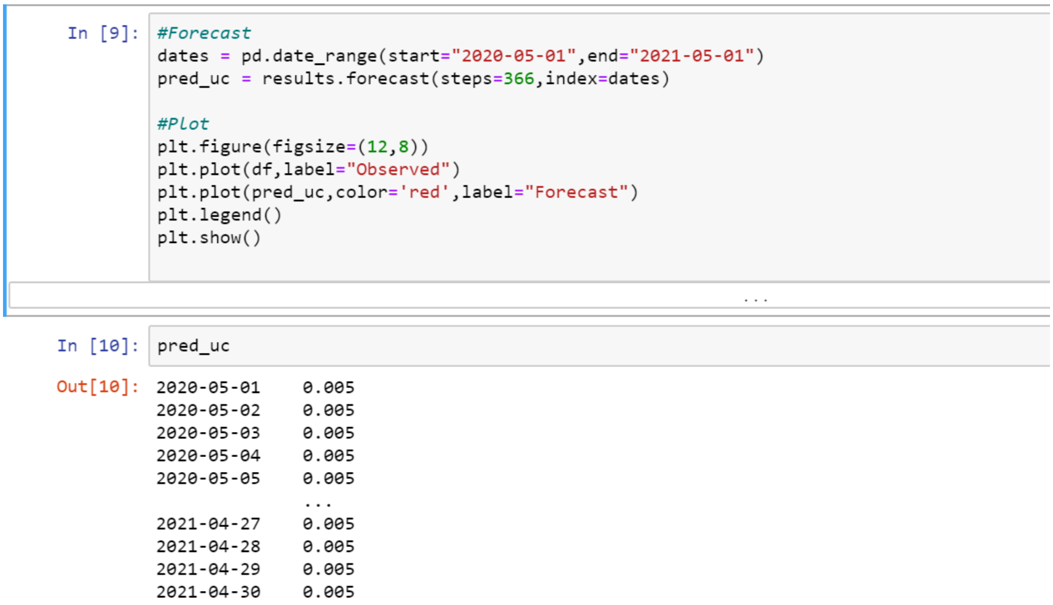
- In the ARIMA model we were able to split the dataset into training, testing and validating but in the SARIMA model all the dataset is used the training process and we tried to split the dataset for testing purposes but the model would not run as the model is built to function like that.

Shortcomings

- We did not go ahead with an LSTM model because of the processing power that it uses when training a dataset. When Eddie was testing out the LSTM model, the average wait time for the LSTM model to train a model is around 3-4 hours but when compared to the SARIMA model we use, it only took us about 15 minutes to train the model.

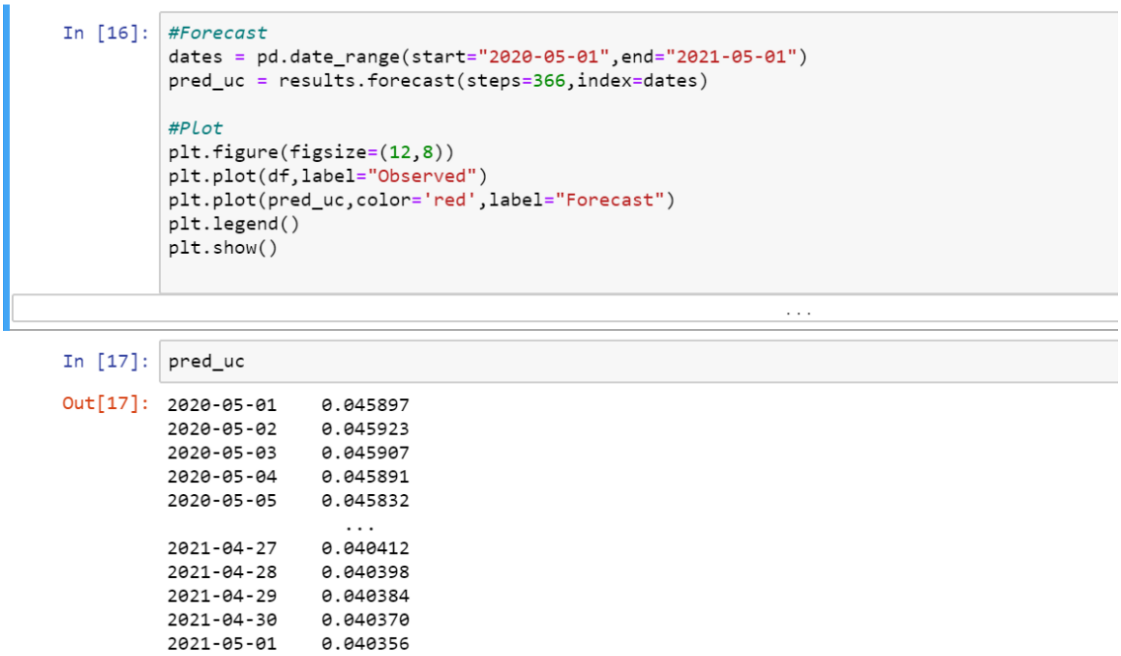
- Instead of using an LSTM model we went ahead with an SARIMA model. As stated above, SARIMA uses less processing power and uses a much shorter time to train the model. Another reason why we went with an SARIMA model is because it was much easier to understand when compared to LSTM as we do not have the time due to the current pandemic.

- The SARIMA model takes a longer processing power and time when we tried to test for daily results. The daily results that we got did not have much variance between them so we decided it was not worth the processing power and time and instead we stuck with just the monthly result but the variances of results are not the same for each stock. An example can be seen from the two figures below showing the daily predicted results for stock 0091 and stock 5166.



```
In [9]:  #Forecast
          dates = pd.date_range(start="2020-05-01",end="2021-05-01")
          pred_uc = results.forecast(steps=366,index=dates)

          #Plot
          plt.figure(figsize=(12,8))
          plt.plot(df,label="Observed")
          plt.plot(pred_uc,color='red',label="Forecast")
          plt.legend()
          plt.show()

                                                             ...

In [10]:  pred_uc

Out[10]:  2020-05-01    0.005
          2020-05-02    0.005
          2020-05-03    0.005
          2020-05-04    0.005
          2020-05-05    0.005
                         ...
          2021-04-27    0.005
          2021-04-28    0.005
          2021-04-29    0.005
          2021-04-30    0.005
```

Figure 12: Daily predicted results for stock 0091



```
In [16]: #Forecast
         dates = pd.date_range(start="2020-05-01",end="2021-05-01")
         pred_uc = results.forecast(steps=366,index=dates)

         #Plot
         plt.figure(figsize=(12,8))
         plt.plot(df,label="Observed")
         plt.plot(pred_uc,color='red',label="Forecast")
         plt.legend()
         plt.show()
```

```
In [17]: pred_uc

Out[17]: 2020-05-01    0.045897
         2020-05-02    0.045923
         2020-05-03    0.045907
         2020-05-04    0.045891
         2020-05-05    0.045832
                         ...
         2021-04-27    0.040412
         2021-04-28    0.040398
         2021-04-29    0.040384
         2021-04-30    0.040370
         2021-05-01    0.040356
```

Figure 13: Daily predicted results for stock 5561

Recommendation

- Final total value is RM1517.39.
- The car we can realistically recommend is Mercedes Benz Kids Battery Operated Electric Ride On Car, retail price at RM1299, before discount. Link to car

Conclusion & Summary

In conclusion, the stocks did not perform well as expected as the returns are not enough to warrant the purchase of the Mercedes AMG-GT car. The investment resulted in a loss in these five stocks. In terms of selecting the machine learning model for predicting stocks using time series, we had the option of LSTM and ARIMA models. We decided to go for ARIMA model after testing both models as in terms of computational power, LSTM takes too long to train (3-4 hours) and takes up a huge computational power while ARIMA (20-30m minutes) is faster and takes less computational power so it allows us to play around with the ARIMA model more and to understand parameters and how it functions. After performing various training on the ARIMA model, we came across the SARIMA model, an extension of ARIMA but with seasonality. From our testing of normal ARIMA vs SARIMA, we found out that the SARIMA performs better in terms of prediction error rate and fits the training data better. Thus, we went for SARIMA model, but we still included the results of normal ARIMA model for comparison. In our opinion working on the stock data is an arduous process as we are newcomers to the stock market prediction scene and thus, we had issues trying to find and fit the models to the training data. However, from our findings, we found out that despite having the top 5 average growth for the selected stocks, all the stocks prediction 1 year ahead has a downward trend and thus the prior investments suffer in the end. We found out that stocks are difficult to predict as for this assignment as its very volatile.

Sources

Brownlee, J 2018. *A gentle introduction to SARIMA for time series forecasting in python*, Machine Learning Mastery, viewed 13 June 2020, < https://machinelearningmastery.com/sarima-for-time-series-forecasting-in-python >.

Malik, F 2018, *Understanding Auto Regressive Moving Average Model — ARIMA*, Medium, viewed on 5 June 2020, < https://medium.com/fintechexplained/understanding-auto-regressive-model-arima-4bd463b7a1bb >

ahajib 2016, *Time series prediction using ARIMA vs LSTM,* StackExchange, viewed 10 June 2020, < https://datascience.stackexchange.com/questions/12721/time-series-prediction-using-arima-vs-lstm >

Pietro, M D 2018?, *Time Series Forecasting: ARIMA vs LSTM vs PROPHET*, Medium, viewed on 8 June 2020, < https://medium.com/analytics-vidhya/time-series-forecasting-arima-vs-lstm-vs-prophet-62241c203a3b >

Khandelwal, R 2019, *Time Series Prediction using SARIMAX,* Medium, viewed 8 June 2020 < https://medium.com/datadriveninvestor/time-series-prediction-using-sarimax-a6604f258c56 >

tktktk0711 2017, *statespace.SARIMAX model: why the model use all the data to train mode, and predict the a range of train model,* StackOveflow, viewed 9 June 2020, < https://stackoverflow.com/questions/44235558/statespace-sarimax-model-why-the-model-use-all-the-data-to-train-mode-and-pred >

Ng, Y 2019, *Machine Learning Techniques applied to Stock Price Prediction*, TowardsDataScience, viewed on 12 May 2020, < https://towardsdatascience.com/machine-learning-techniques-applied-to-stock-price-prediction-6c1994da8001 >

Felton, I 2019, *A Quick Example of Time-Series Prediction Using Long Short-Term Memory (LSTM) Networks*, Medium, viewed 12 May 2020 < https://medium.com/swlh/a-quick-example-of-time-series-forecasting-using-long-short-term-memory-lstm-networks-ddc10dc1467d >

Singh, A 2018, *Stock Prices Prediction Using Machine Learning and Deep Learning Techniques (with Python codes)*, Analytics Vidhya, viewed 18 May 2020, < https://www.analyticsvidhya.com/blog/2018/10/predicting-stock-price-machine-learningnd-deep-learning-techniques-python/ >

Brownlee, J 2016, *Time Series Forecasting as Supervised Learning*, Machine Learning Mastery, viewed 19 May 2020, < https://machinelearningmastery.com/time-series-forecasting-supervised-learning/ >

Brownlee, J 2019, *How to Save and Load Your Keras Deep Learning Model*, Machine Learning Mastery, viewed 19 May 2020, < https://machinelearningmastery.com/save-load-keras-deep-learning-models/ >

Priya, V 2015, *How much data is considered to be small data/Large data in data mining?*, ResearchGate, viewed 19 May 2020,< https://www.researchgate.net/post/How_much_data_is_considered_to_be_small_data_Large_data_in_data_mining2 >.

Zajic, A 2019, *Introduction to AIC — Akaike Information Criterion*, TowardsDataScience, viewed 13 June 2020, < https://towardsdatascience.com/introduction-to-aic-akaike-information-criterion-9c9ba1c96ced >

SARIMA model is adapted from:
https://github.com/adsung/Stock-Price-trend-classification-and-price-forecast-using-LSTM-and-ARIMA-model/blob/master/ARIMA_model_price_forecast.ipynb