Abstract

Starbucks is one of the top five companies in the coffee business, and there are over 43,000 shops worldwide, and it's considered one of the most popular coffee company. This project studies the stock market prices for Starbucks company, then predict the closing price in the next year, using the ARIMA time series model.

Data structure

The data was collected from MarketWatch, which is a stock market website. The collected data started from 1st Jan 2016 until 6th Dec 2021, with 1499 rows and 6 columns, adding them all into one data frame.

Date	Open	High	Low	Close	Volume
Date the data was collected	Opening price at a specific date	Highest price	Lowest price	Closing price	Volume
Object	Object	Object	Object	Object	Object

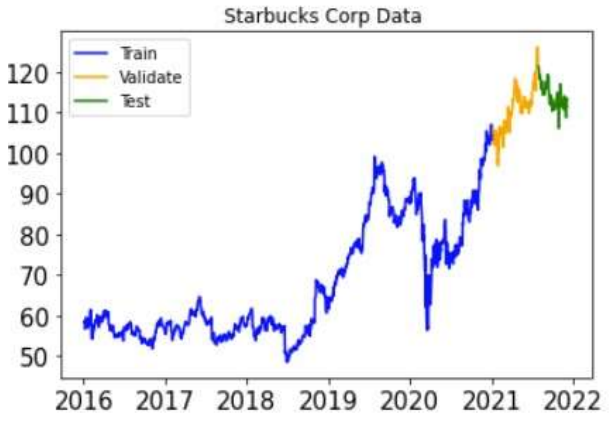
	0	1	2	3	4	5
0	Date	Open	High	Low	Close	Volume
1	12/30/2016	56.28	56.45	55.40	55.52	8,344,508
2	12/29/2016	56.35	56.47	56.14	56.32	3,781,721
3	12/28/2016	56.80	56.90	56.25	56.35	5,548,726
4	12/27/2016	56.99	57.39	56.81	56.86	4,186,157

Data cleaning:

The first step is to rename the columns, then dropping the unused columns in this analysis, dropping the rows which included names of the columns, dropping NA, set the index of the date frame to the date, and converting the date into date time, converting the close column into float, and sort the data frame. Last step is to split the data frame into: train for training the model which included 94 percent of the data, and from this training set taking the validation set which is 10 percent of the training set, and lastly is the testing set.

```
<class 'pandas.core.frame.DataFrame'>
DatetimeIndex: 1263 entries, 2016-01-04 to 2021-01-07
Data columns (total 1 columns):
 #   Column  Non-Null Count  Dtype
---  ------  --------------  -----
 0   CLOSE   1263 non-null   float64
dtypes: float64(1)
memory usage: 19.7 KB
None
##################################
<class 'pandas.core.frame.DataFrame'>
DatetimeIndex: 140 entries, 2021-01-08 to 2021-07-29
Data columns (total 1 columns):
 #   Column  Non-Null Count  Dtype
---  ------  --------------  -----
 0   CLOSE   140 non-null    float64
dtypes: float64(1)
memory usage: 2.2 KB
None
##################################
<class 'pandas.core.frame.DataFrame'>
DatetimeIndex: 90 entries, 2021-07-30 to 2021-12-06
Data columns (total 1 columns):
 #   Column  Non-Null Count  Dtype
---  ------  --------------  -----
 0   CLOSE   90 non-null     float64
dtypes: float64(1)
memory usage: 1.4 KB
None
```

DATE	CLOSE
2016-01-04	58.26
2016-01-05	58.65
2016-01-06	58.13
2016-01-07	56.69
2016-01-08	56.63
...	...
2021-11-30	109.64
2021-12-01	108.66
2021-12-02	111.42
2021-12-03	111.24
2021-12-06	113.36



Design and algorithm

After splitting the data into train, validation, and test, check the stationarity is a necessary step for the ARIMA time series model, by the dicky fuller and KPSS test. The result is that all of the data sets are not stationary, therefore use the ARIMA model with differencing once to make the data stationary.

The next step is to determine the best fit of the ARIMA model by trying different orders of the p and q values, starting from 0 to 5, based on the lowest root mean of squared errors. The table shows the best order of each value.

Value	Best order based on the RSME
1=> (1,1,0)	0.18177178481712283 6.650977012275106
2=> (2,1,1)	0.2288112984080849 6.559330841408596
3=> (1,1,3)	0.2381034677522918 6.534312738053918
4=> (4,1,2)	0.2583877561807341 6.487219747280541
5=> (0,1,5)	0.19787184703825342 6.571224612712827

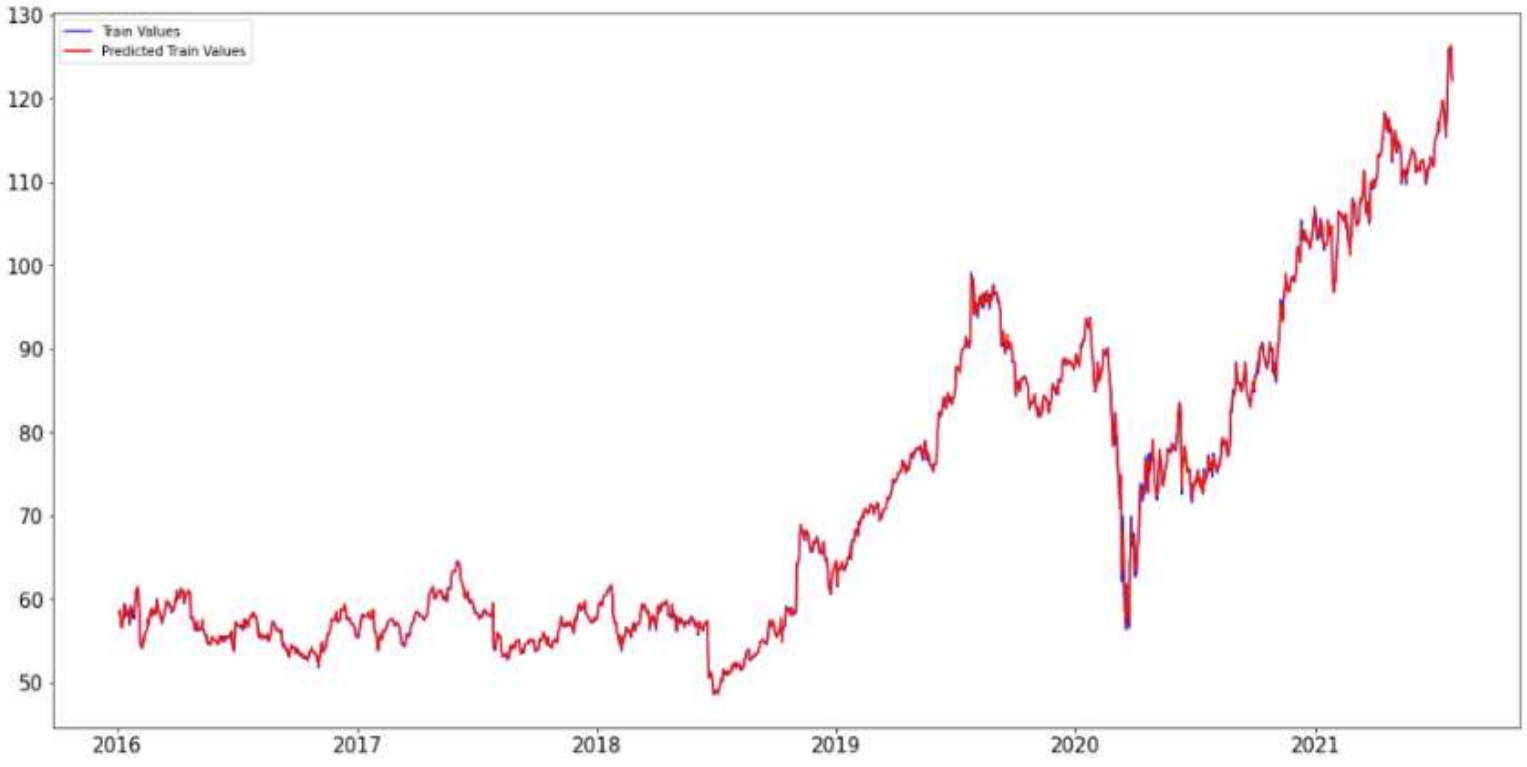
The **first approach** is to take the data from 2016-2021, and based on the results, the best fit of the train and the validation, which is the lowest values of RSME, is the (4,1,2). So, the final step is combining the train and the validation set into the final train, test the stationarity, fit the ARIME (4,1,2), and validate it with test set.

The result of applying the ARIMA (4,1,2):

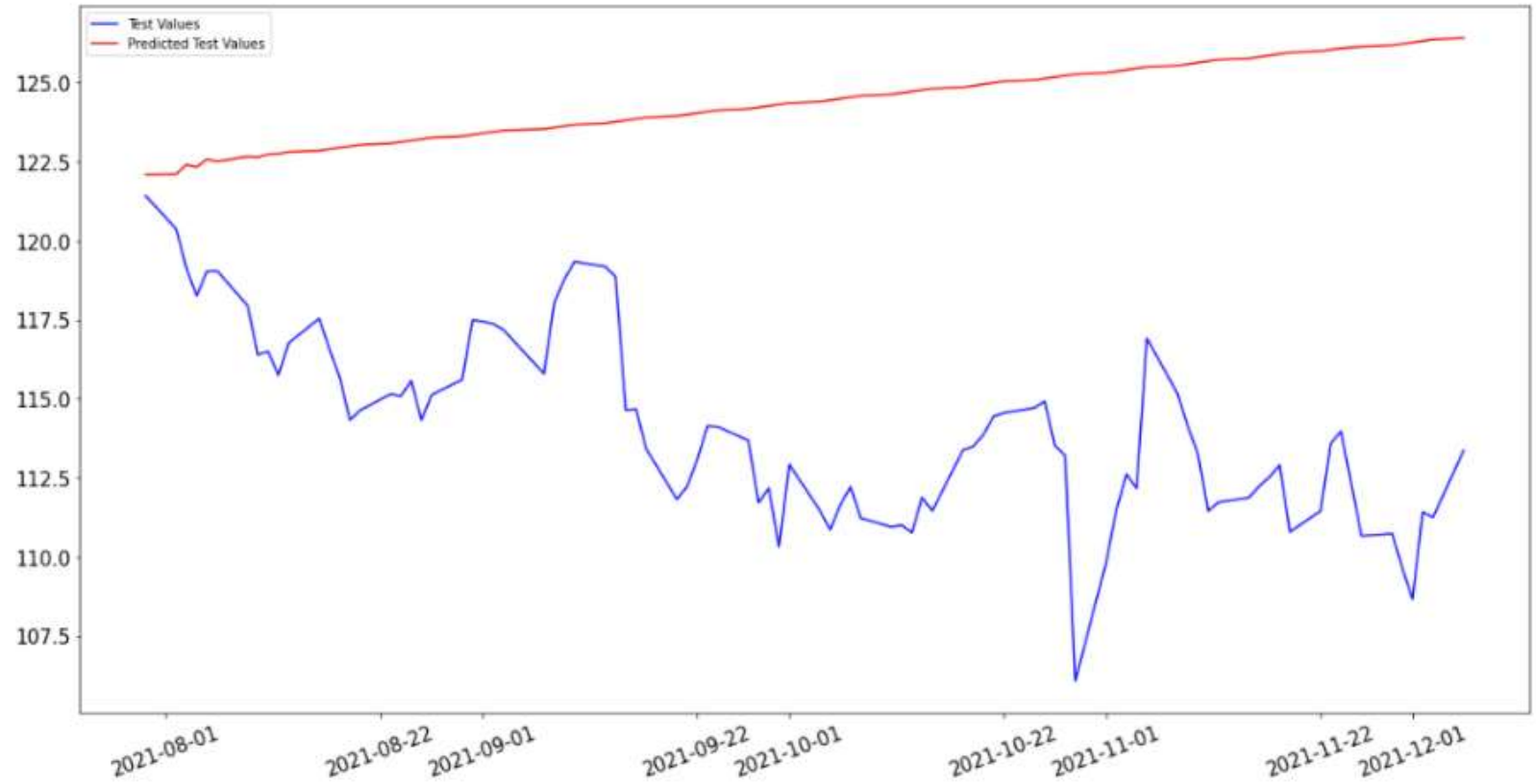
RMSE for the training set: 0.23596924955878643.
RMSE for the testing set: 10.993793405368965.
And the mean of the testing set is: 114.085778.

The equation of this model: $\hat{y}_t = c + \varphi_1 \hat{y}_{t-1} + \varphi_2 \hat{y}_{t-2} + \varphi_3 \hat{y}_{t-3} + \varphi_4 \hat{y}_{t-4} + \theta_1 \varepsilon_{t-1} + \theta_2 \varepsilon_{t-2} + \varepsilon_t$

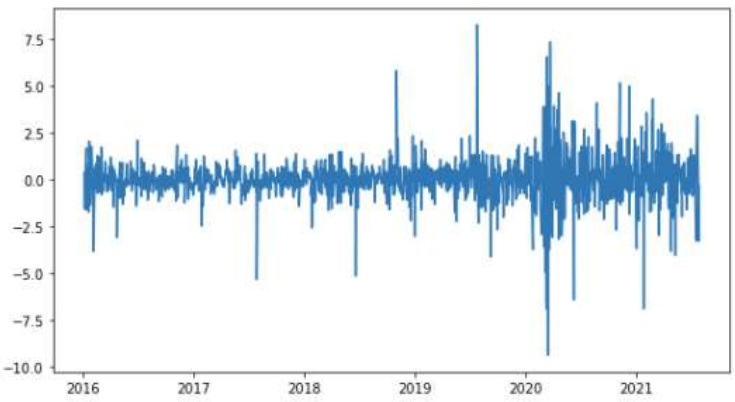
Train set:

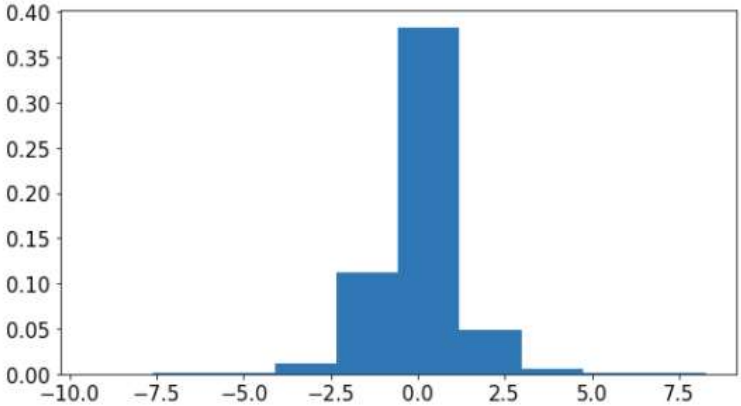


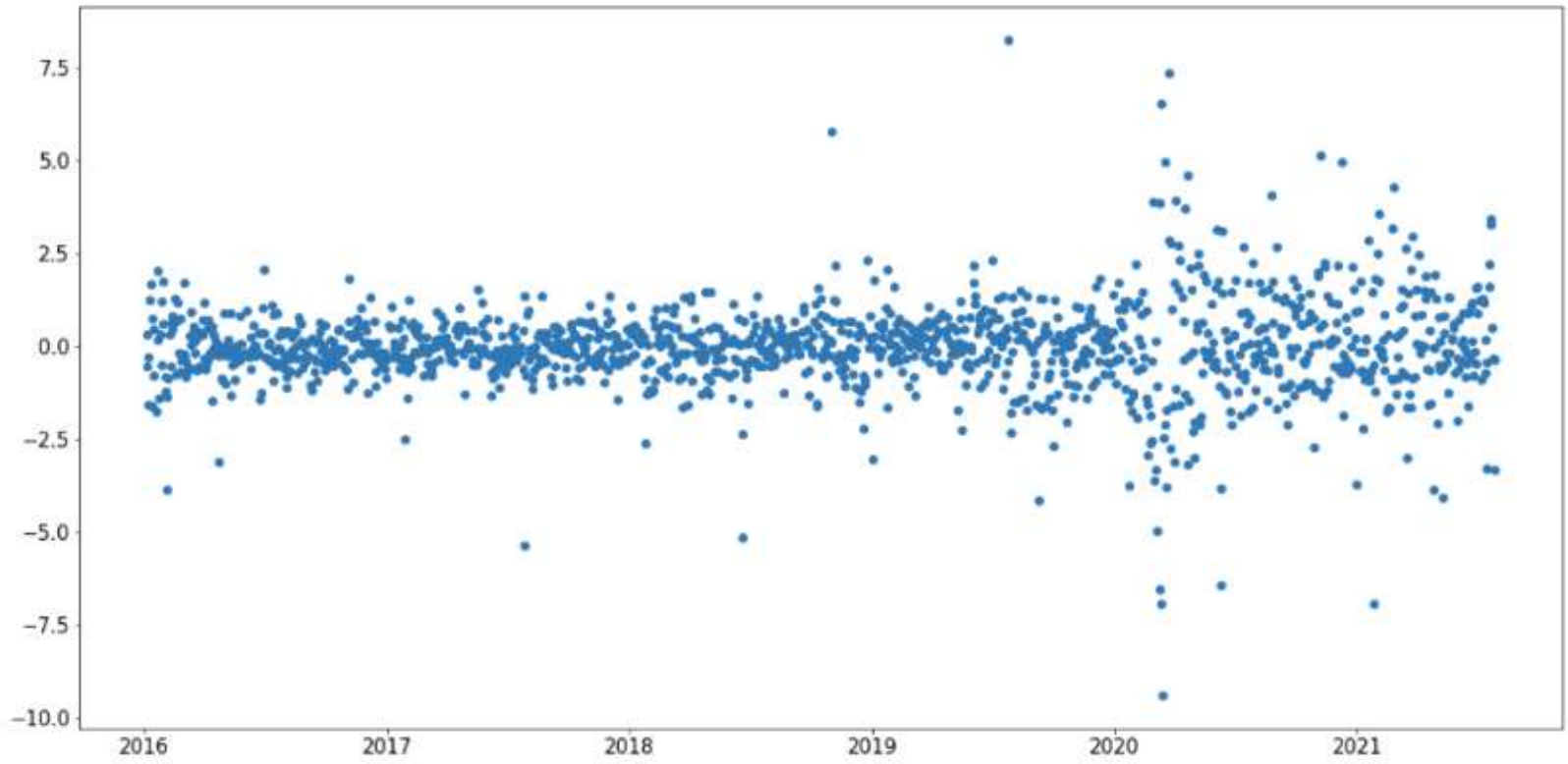
Test set:



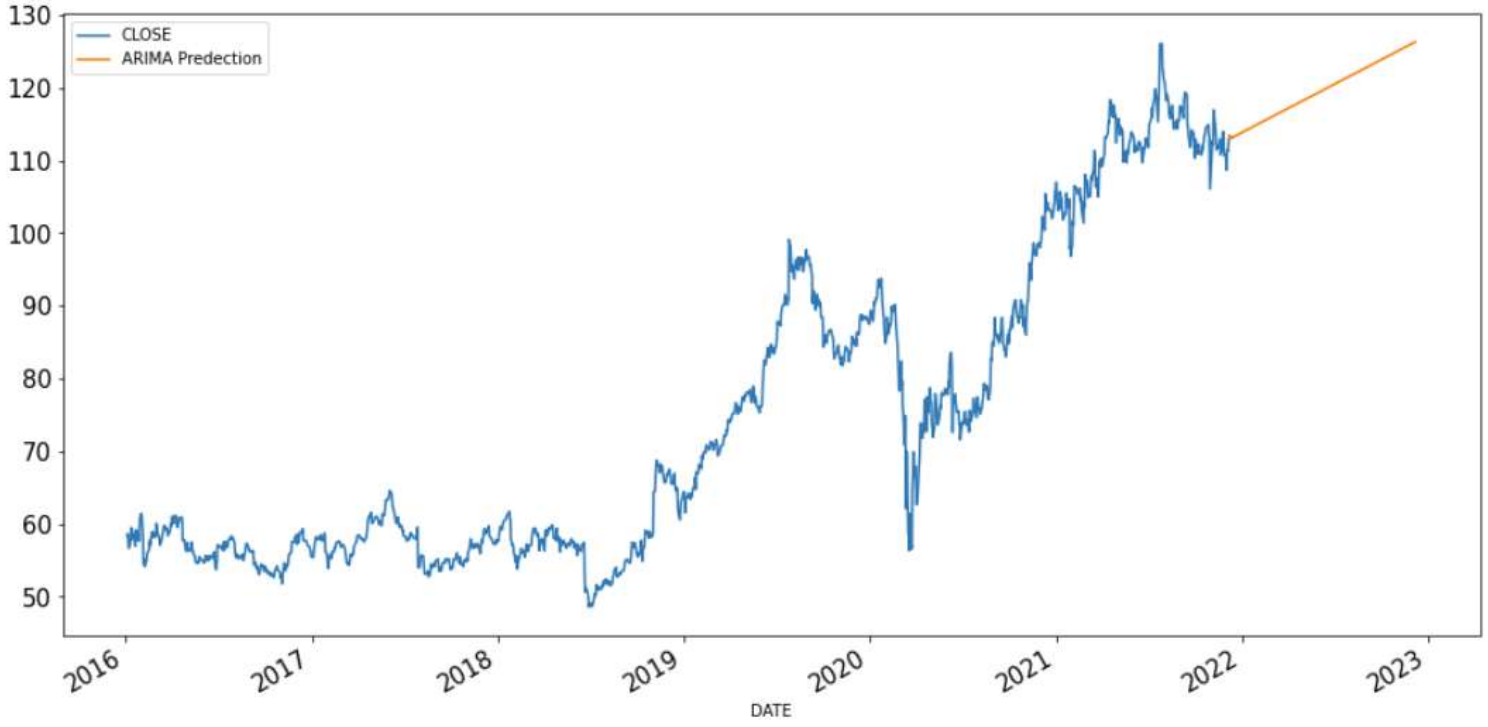
Residual error:

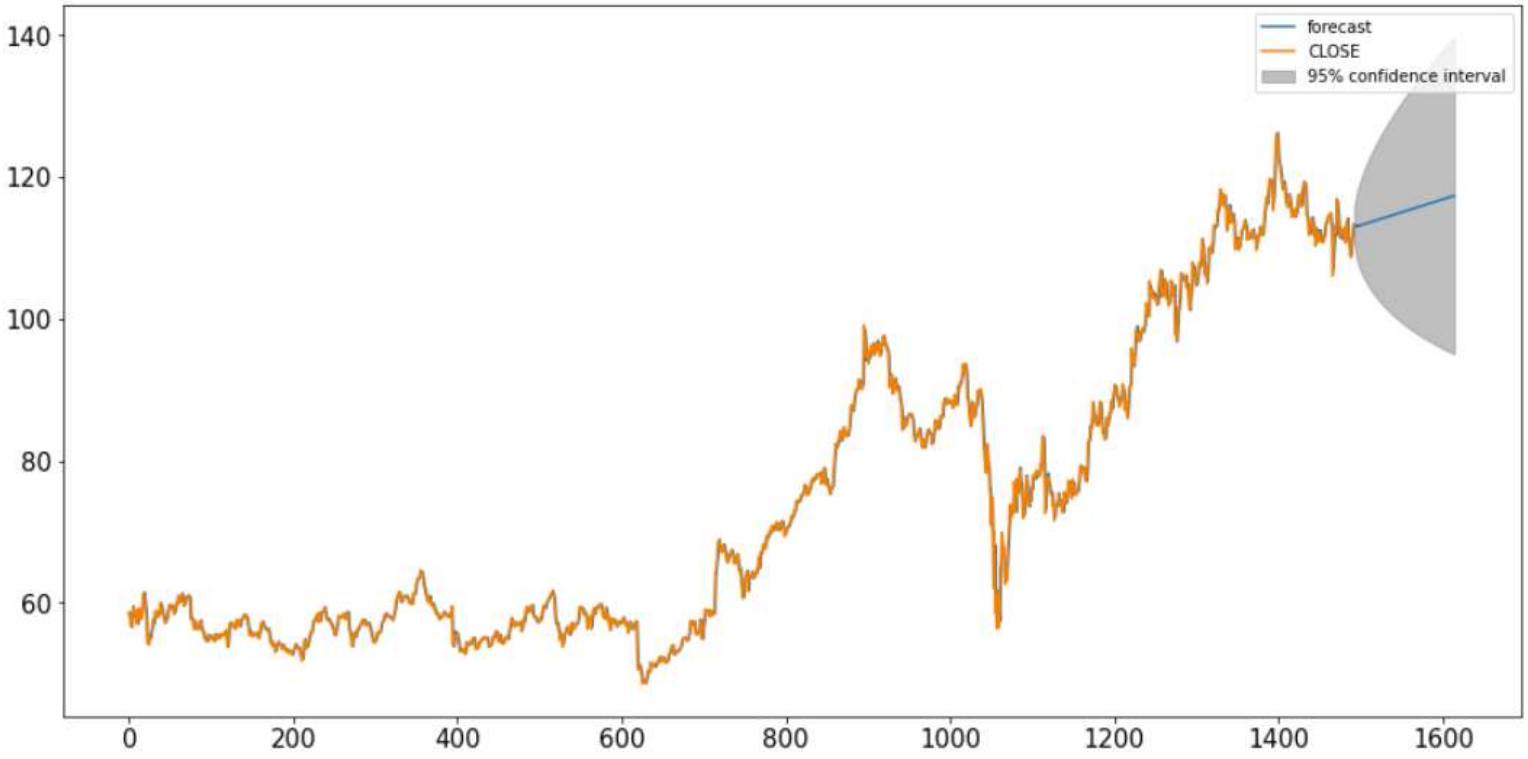






For the prediction





The **second approach**, is to not include the period from 2020-2021 due to covid-19 pandemic. As the table shows, the best fit of the ARIMA model is (2,1,3) with the lowest value in the RMSE.

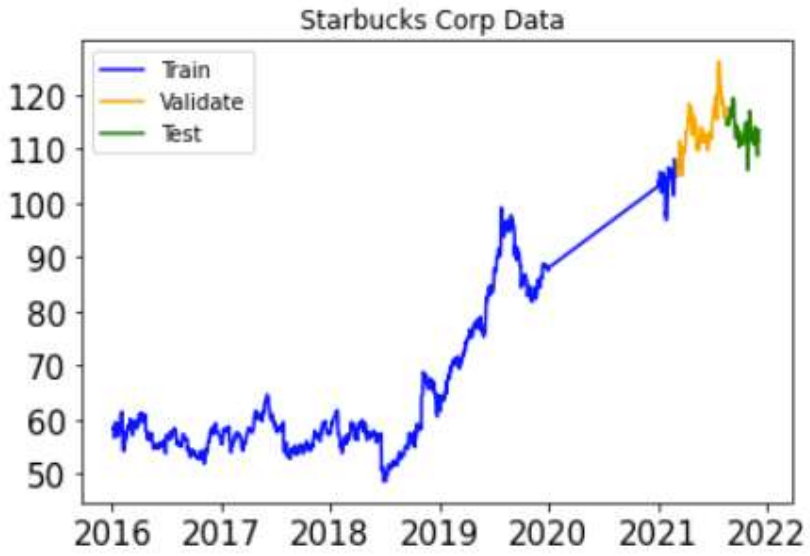
Value	Best order based on the RSME
1=> (1,1,1)	0.05600379482858372 7.043124768971345
2=> (2,1,2)	0.09343940612725077 7.041751447910158
3=> (2,1,3)	0.11438360582233079 6.385317308497374
4=> (2,1,4)	0.096605459925839 6.941709780979756
5=> (5,1,5)	0.15172715699321743 6.884117701137903

The result of applying the ARIMA (2,1,3):

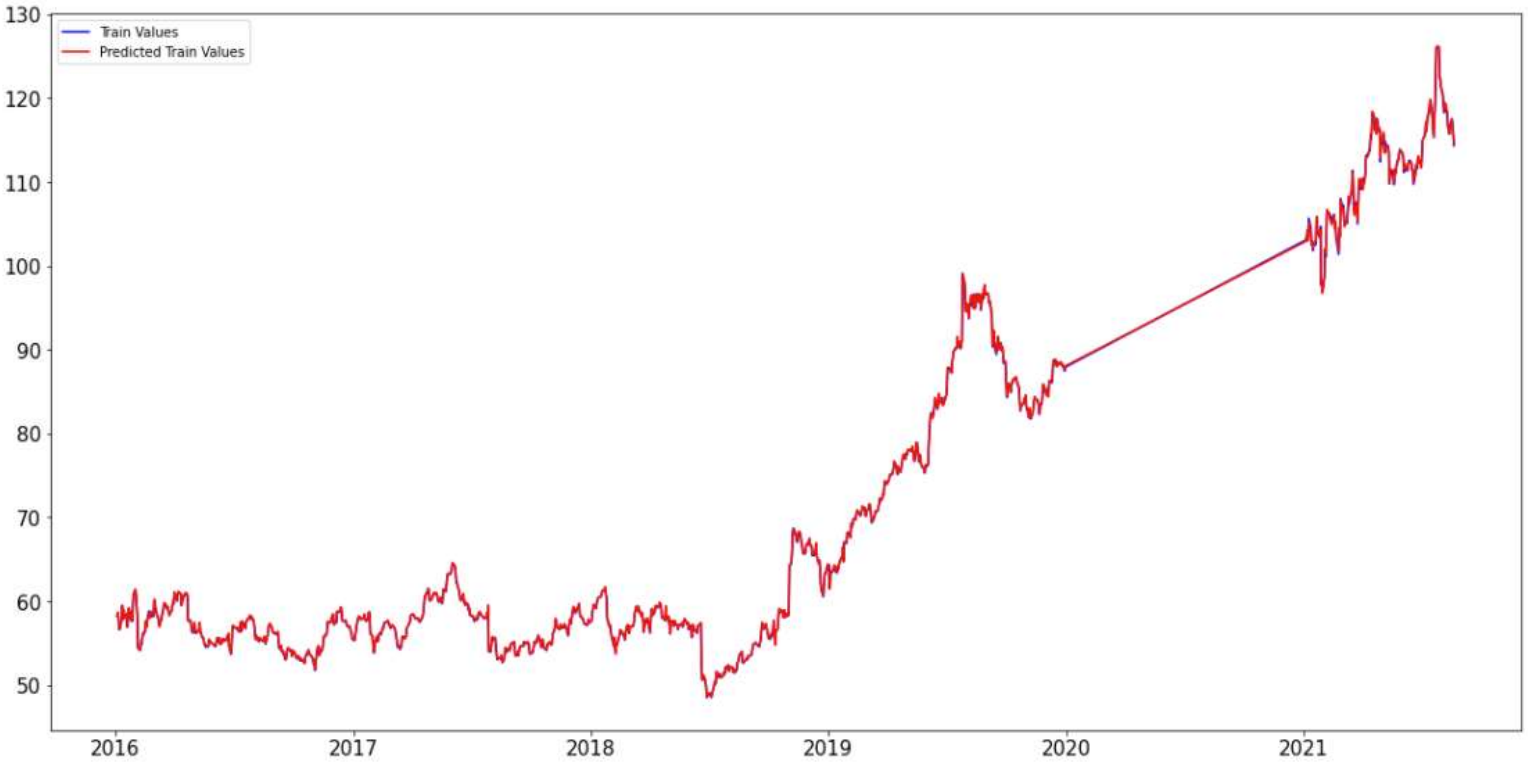
RMSE for the training set: 0.11975507325705646.
RMSE for the testing set: 4.6000970301995014.
And the mean of the testing set is: 113.373467.

The equation of this model: $\hat{y}_t = c + \varphi_1 \hat{y}_{t-1} + \varphi_2 \hat{y}_{t-2} + \theta_1 \varepsilon_{t-1} + \theta_2 \varepsilon_{t-2} + \theta_3 \varepsilon_{t-3} + \varepsilon_t$

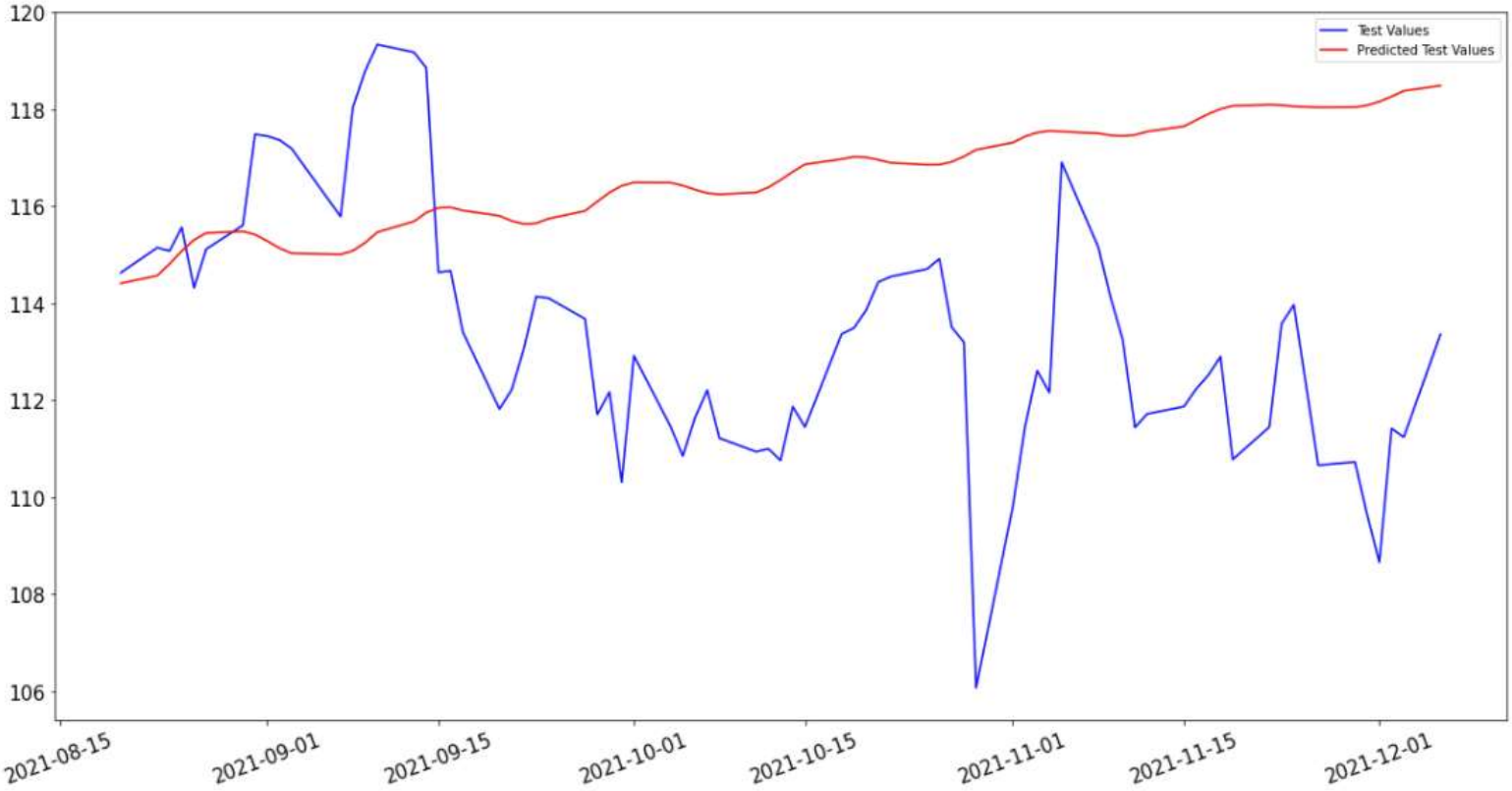
After removing the period between 2020-2021



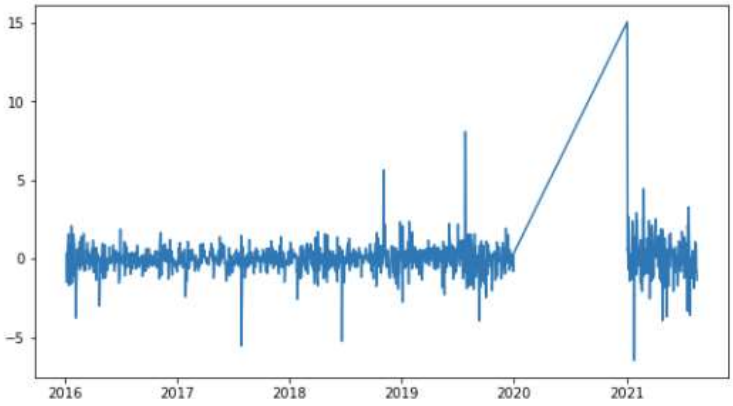
Train set:

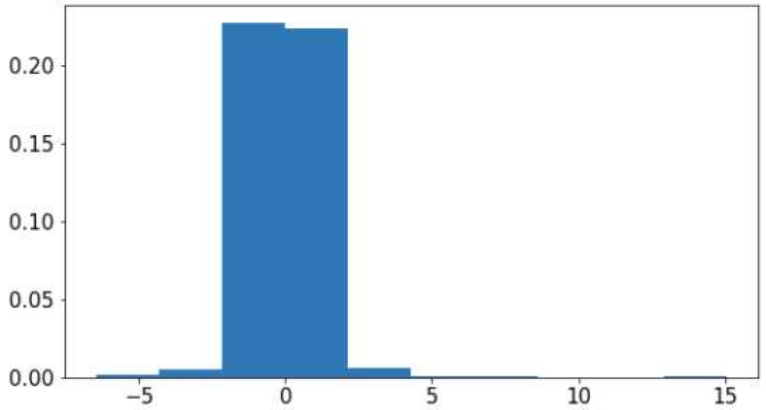


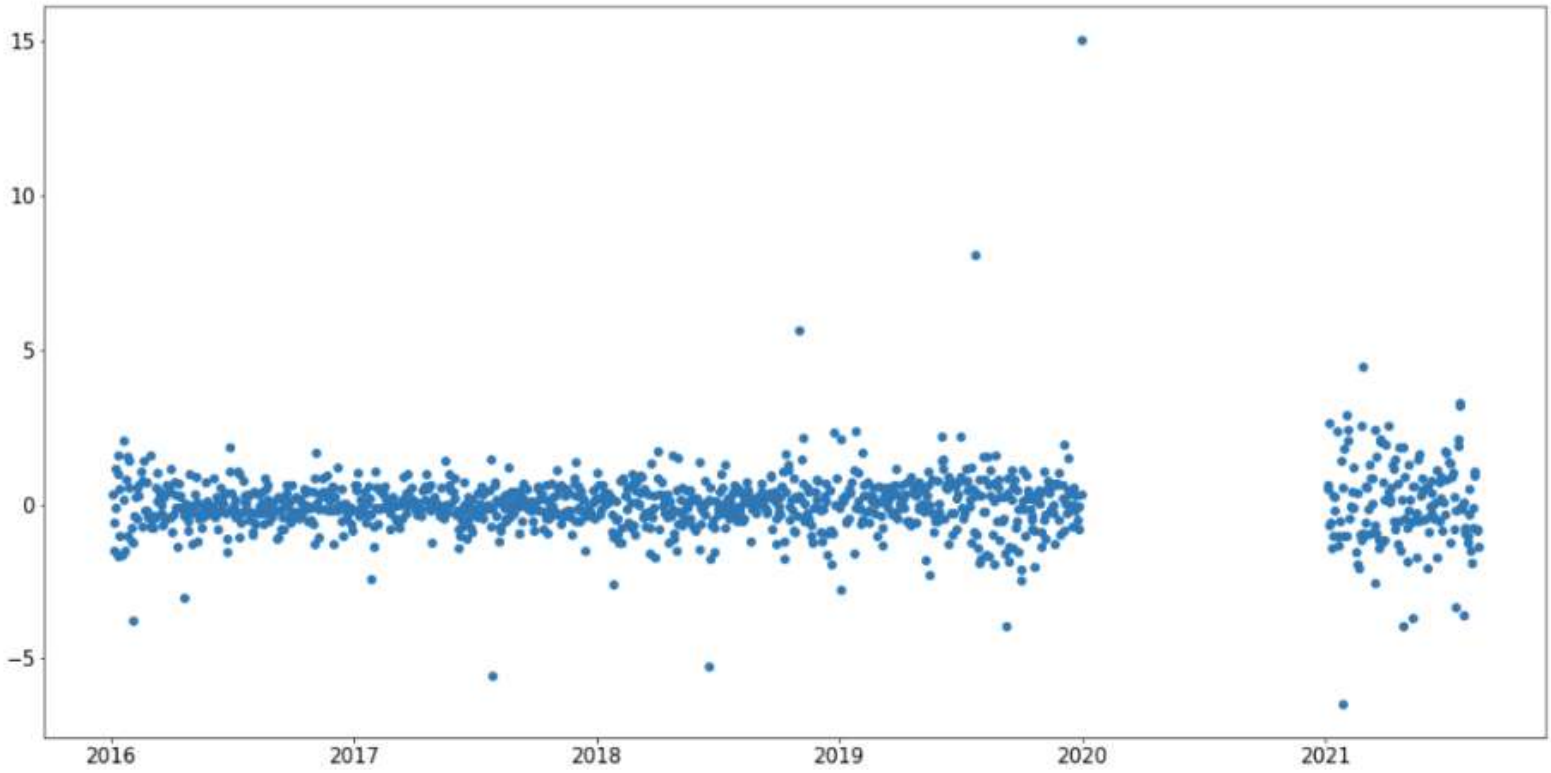
Test set:



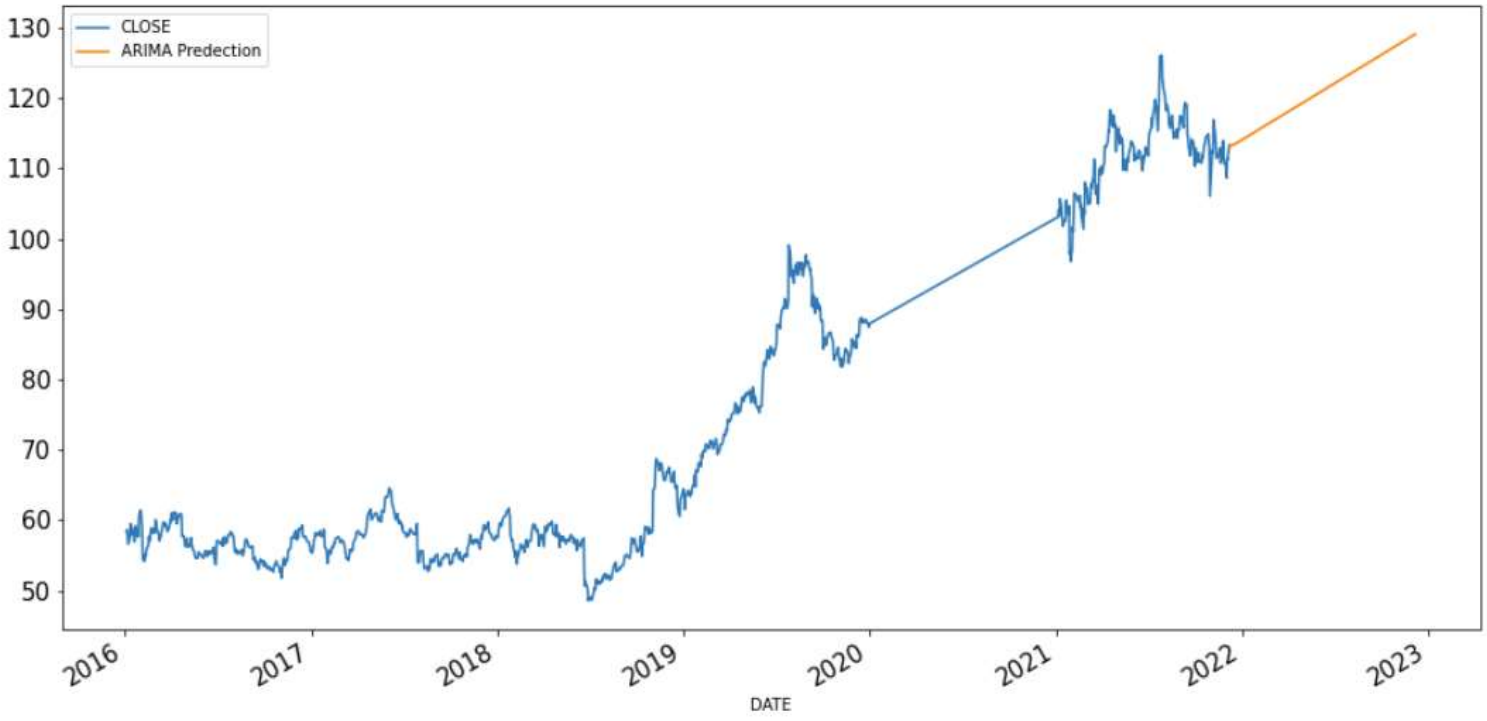
Residual error:

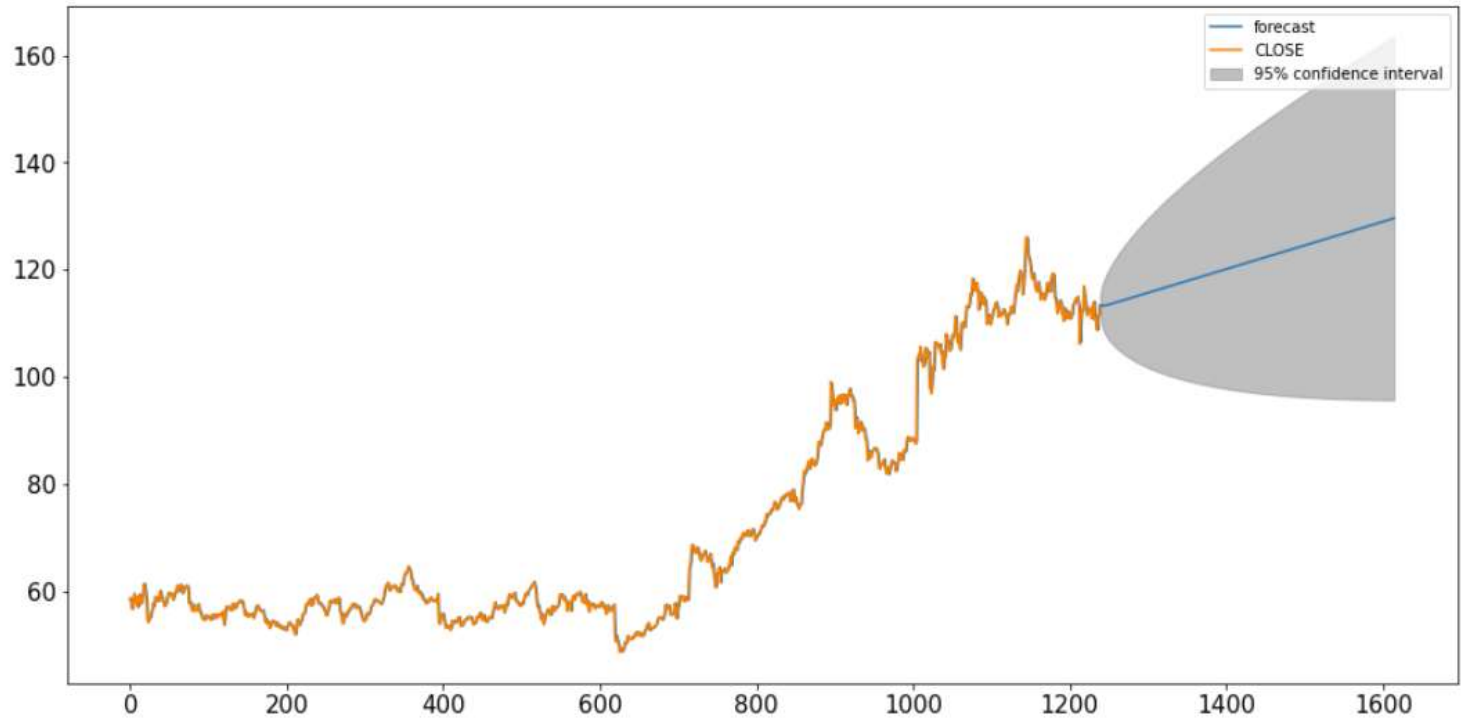






For the prediction:





Tools

- Jupyter.
- BeautifulSoup.
- Requests.
- Pandas.
- StringIO.
- CSV.
- NumPy.
- seaborn.
- matplotlib.pyplot.
- Adfuller.
- KPSS.
- ARIMA.
- Metrics

Conclusion and communication:

The results are inconclusive, due to the factors that can affect the results that was included in the ARIMA model. However, the first approach, which is including all the data from 2016 until 2021, included higher RMSE for the testing which is equal to 10.99 set than the second approach which is equal to 4.60 which is much better.

As a conclusion, ARIMA changes the best fit based on the data entered, and the above results shows and explains this.