COMP0051 Algorithmic Trading

Coursework - 1

22092172

UCL

February 18, 2023

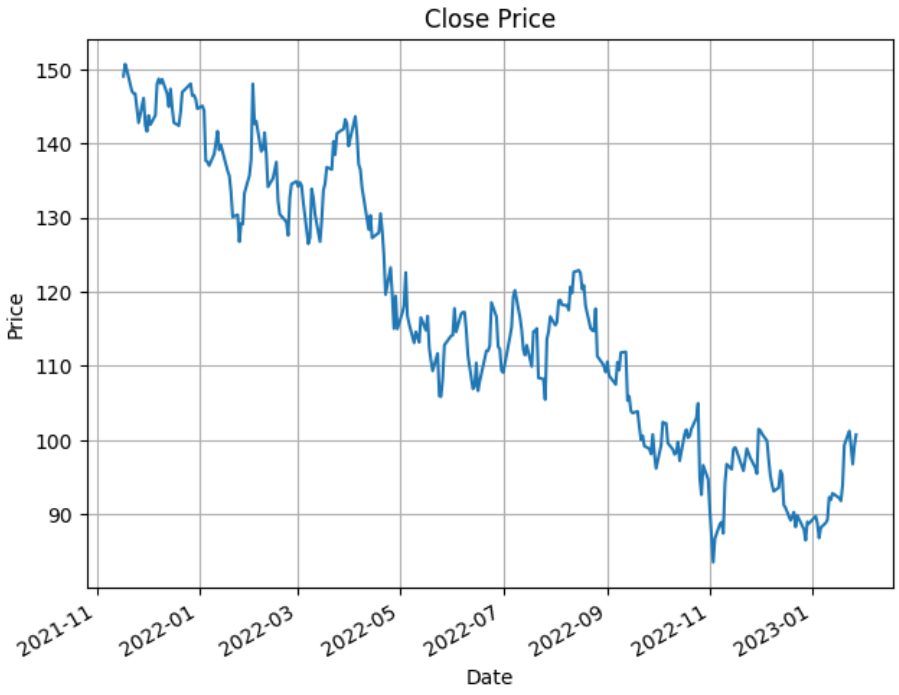
Time series

1. **Download a price time series using an API. The length of the time series T, with T=300. The resolution could be any, from tick data to months.**

   Ans. We retrieve the financial data for 300 trading days i.e. from 2021-11-17 to 2023-1-30 using the *yfinance* API of Google stock prices.

2. **Plot the price time series**

   Ans. The figure below shows the plot of *close prices* against time.



Moving averages

3. **Define mathematically the moving average of the price time series with an arbitrary time- window t**

   Ans. A series of averages computed from past data is called a moving average. Mathematically, the average is calculated by dividing the sum of the price data points over a specific time period, τ, by the number of data points. It is called a "moving" average because it is continually updated using the most recent price data.

$$y_t = \frac{\epsilon_t + \epsilon_{t-1} + \epsilon_{t-2} + \cdots + \epsilon_{t-\tau+1}}{\tau}$$

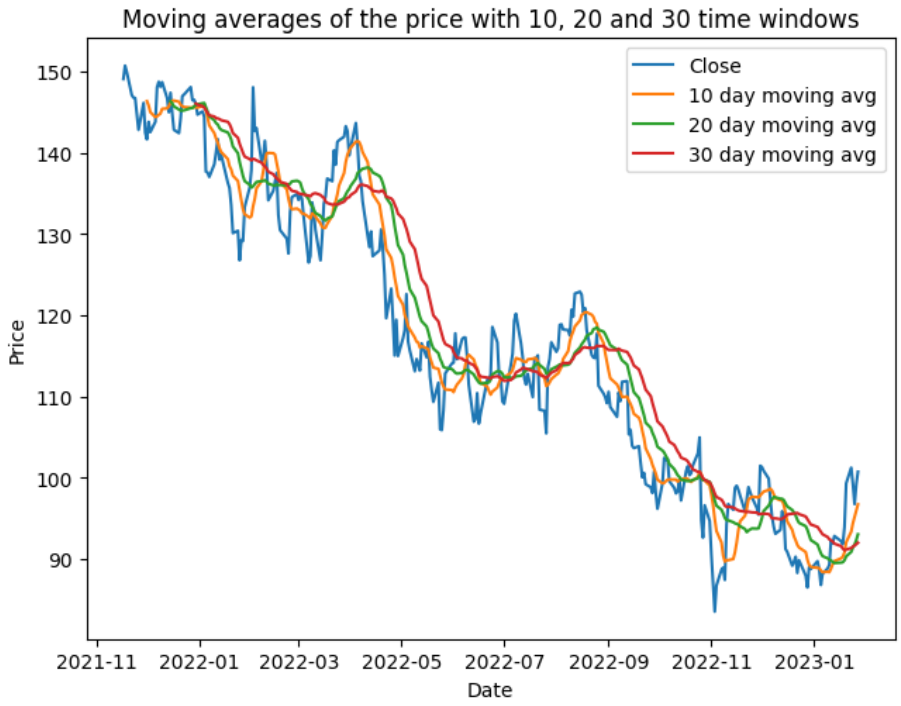
   where, ϵ_t is the close price at t and τ is the time window

4. **Compute three moving averages of the price time series, with time-windows t=10,20,30**

   Ans. We calculate simple moving averages for this analysis. For $\tau = 10$, we add the closing prices for the last ten trading days and divide the total by the number of days i.e. 10. Similarly, for $\tau = 20$ and $\tau = 30$ we sum up the closing prices over 20-day and 30-day periods and divide them by 20 and 30 respectively. We use the $.rolling(\tau)$ method in Python to compute the moving averages where the time window is passed as the argument.

5. **Plot the moving averages against the price time series**

   Ans. The figure below depicts the moving averages of the *Close prices* with time window $\tau = 10, 20$ and 30 days.



6. **Compute the linear and log-return of the price time series**

   Ans. Linear return is the relative change in the price of the time series over time t. It is given by :

$$linear\_ret_t = \frac{y_t - y_{t-1}}{y_{t-1}}$$

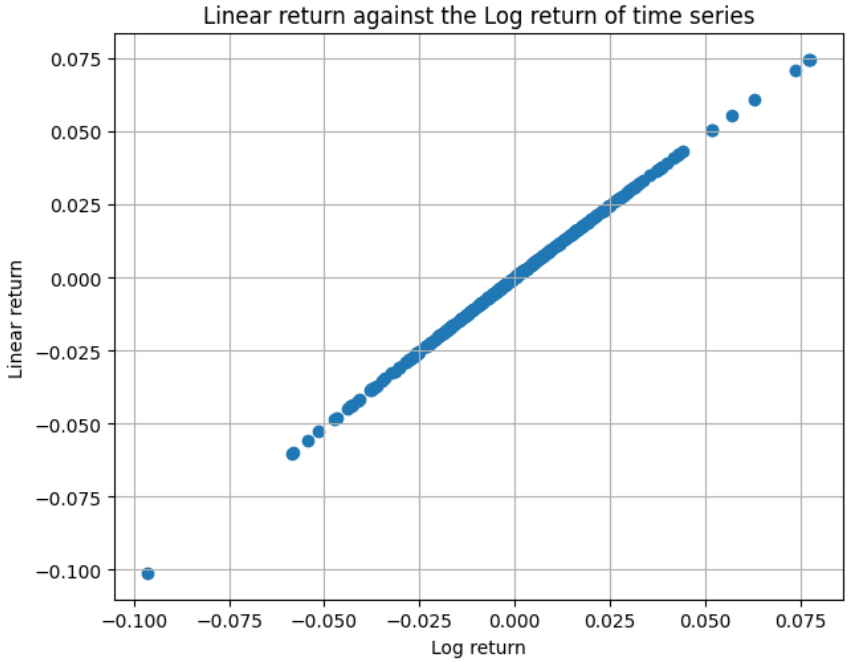
   where, y_t is the close price in time series at time $= t$

   Log return is calculated by taking the logarithm of two consecutive prices in the time series. It is given by :

$$log\_ret_t = \frac{y_t}{y_{t-1}}$$

7. **Plot the linear return against the log-return time series**

   Ans. We plot a scatter plot of linear return values against the log return values of close prices in the time series as shown below.



Time Series Analysis

8. **Define the auto-correlation function (for a stationary time-series)**

   Ans. Autocorrelation function is a mathematical representation of the degree of similarity between a given time series and a lagged version of itself over successive time intervals. For a stationary time series, the ACF will drop to zero relatively quickly, while the ACF of non-stationary data decreases slowly. Mathematically, the autocorrection function for a stationary time series between y_t and its values at lag k i.e. y_{t-k}, data points is given by:

$$\rho(y_t, y_{t-k}) = \frac{Covariance(y_t, y_{t-k})}{Std.Deviation(y_t) * Std.Deviation(y_{t-k})} = \frac{Covariance(y_t, y_{t-k})}{Variance(y_t)}$$

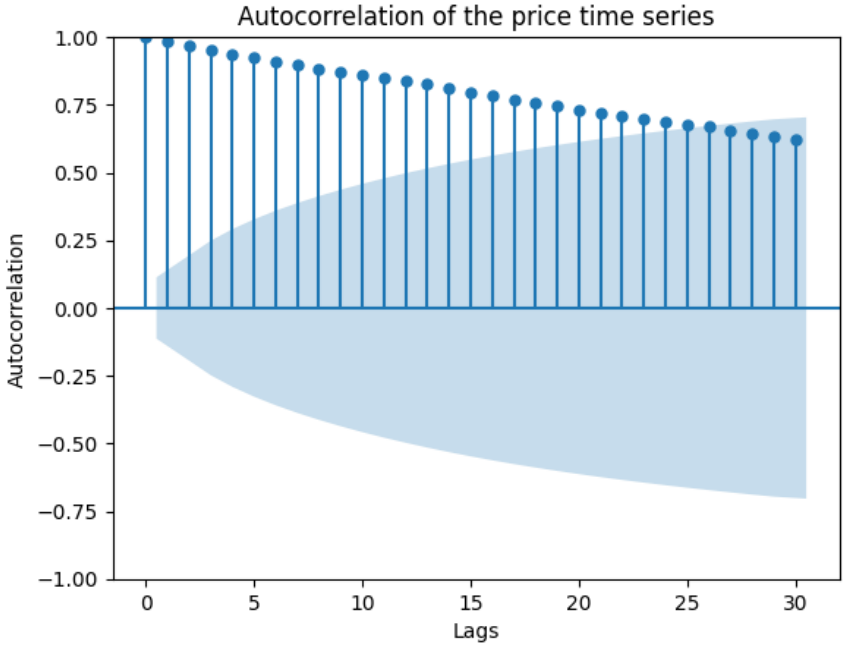
   here, since the standard deviation is constant for stationary data we change the denominator to get the second formula.

9. **Compute the auto-correlation function (ACF) of the price time series**

   Ans. The autocorrelation function of the close price is calculated using the acf function in the *statsmodels* module of Python with lag 0 by default.

10. **Plot the price ACF**

    Ans. The following figure used the $plot\_acf$ function to describe the auto-correlation function values applied over the close prices with $lags = 30$.

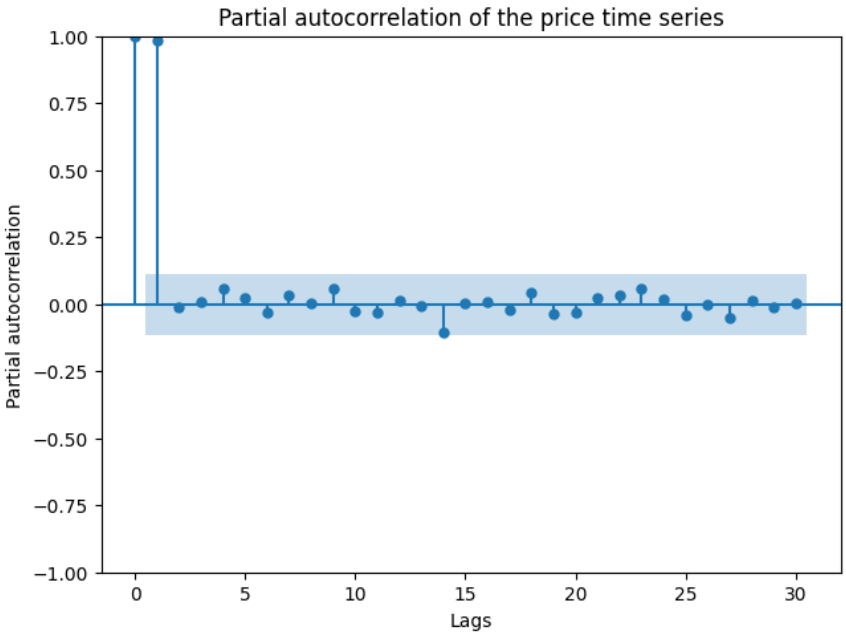


11. **Compute the partial auto-correlation function (PACF) of the price time series**

    Ans. The partial autocorrelation estimate of close price is calculated using the *pacf* function in the *statsmodels* module of Python with lag 0 by default.

12. **Plot the price PACF**

    Ans. The following figure used the *plot_pacf* function to describe the partial auto-correlation function values applied over the close prices with $lags = 30$.

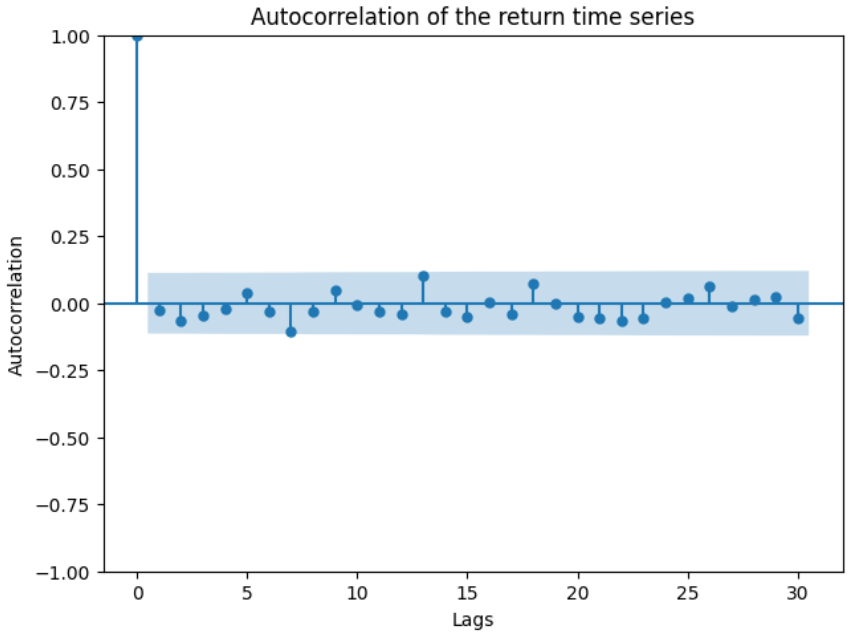


13. **Compute the auto-correlation function (ACF) of the return time series**

    Ans. As stated above, we use the *acf* function on linear return values to find their auto-correlation function.

14. **Plot the return ACF**

Ans. The following figure used the *plot_acf* function to describe the auto-correlation function values applied over the return close prices with $lags = 30$.

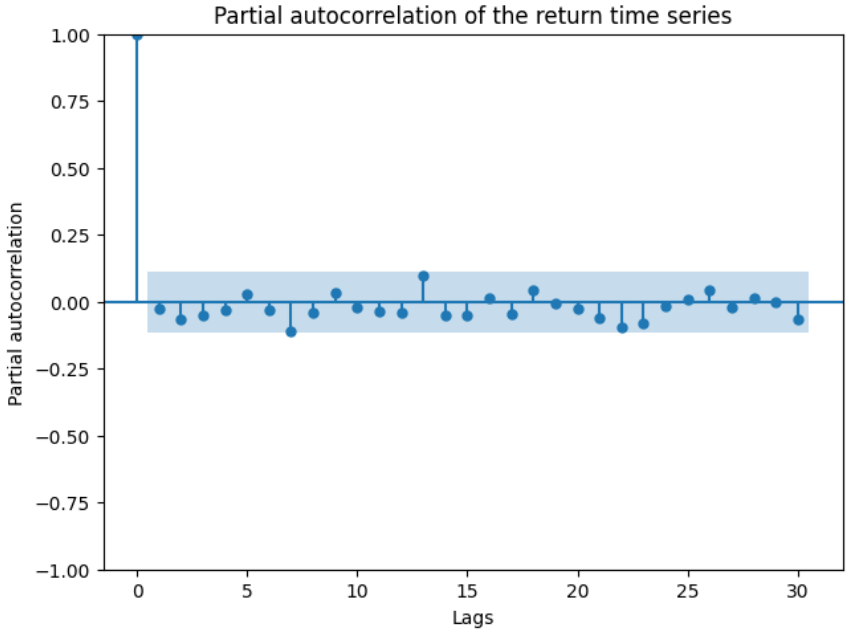


15. **Compute the partial auto-correlation function (PACF) of the return time series**

Ans. As stated above, we use the *pacf* function on linear return values to find their partial auto-correlation function.

16. **Plot the return PACF**

Ans. The following figure used the *plot_pacf* function to describe the partial auto-correlation function values applied over the return close prices with $lags = 30$.



ARMA models

17. **Define mathematically an ARMA(p,q) model**

Ans. An ARMA(p, q) or "Auto Regressive Moving Average" model is a linear combination of two linear models, where p denotes the number of lagged observations to be considered for auto regression and q is the size of the moving average window, used to do time series analysis of stationary time series. The equation for a time series ARMA(p, q) model is given by :

$$y_t = c + \phi_1 y_{t-1} + \phi_2 y_{t-2} + \cdots + \phi_p y_{t-p} + \epsilon_t + \theta_1 \epsilon_{t-1} + \theta_2 \epsilon_{t-2} + \cdots + \theta_q \epsilon_{t-q}$$

or

$$\phi_p(B) y_t = \theta_p(B) \epsilon_t$$

where, y_t is time series data at time $= t$, ϕ_i are the parameters of the autoregression model, θ_i are the parameters of the moving average model, c is a constant term, ϵ_t is white noise with zero expectation and σ_t^2 variance, and B is the backward shift operator.

18. **Define a training and test set and fit an ARMA model to the price time series**

Ans. We split the dataset in 60 : 40 ratio where 60% of the close price values are used for training and 40% of the values are used for testing of the ARMA model.

19. **Display the parameters of the model and its Mean Squared Error (MSE) in the training set and in the test set**

Ans. The parameters of the ARMA model with close prices are as follows:

```
==============================================================================
Dep. Variable:                      y   No. Observations:                  179
Model:                 ARIMA(1, 0, 1)   Log Likelihood                -443.288
Date:                Thu, 16 Feb 2023   AIC                            894.576
Time:                        02:02:01   BIC                            907.326
Sample:                             0   HQIC                           899.746
                                  - 179
Covariance Type:                  opg
==============================================================================
                 coef    std err          z      P>|z|      [0.025      0.975]
------------------------------------------------------------------------------
const        130.6591      7.519     17.377      0.000     115.922     145.397
ar.L1          0.9815      0.015     66.420      0.000       0.953       1.010
ma.L1         -0.0439      0.083     -0.531      0.595      -0.206       0.118
sigma2         8.1422      0.801     10.171      0.000       6.573       9.711
===================================================================================
Ljung-Box (L1) (Q):                   0.03   Jarque-Bera (JB):                 2.85
Prob(Q):                              0.86   Prob(JB):                         0.24
Heteroskedasticity (H):               1.16   Skew:                             0.24
Prob(H) (two-sided):                  0.56   Kurtosis:                         3.40
===================================================================================
```

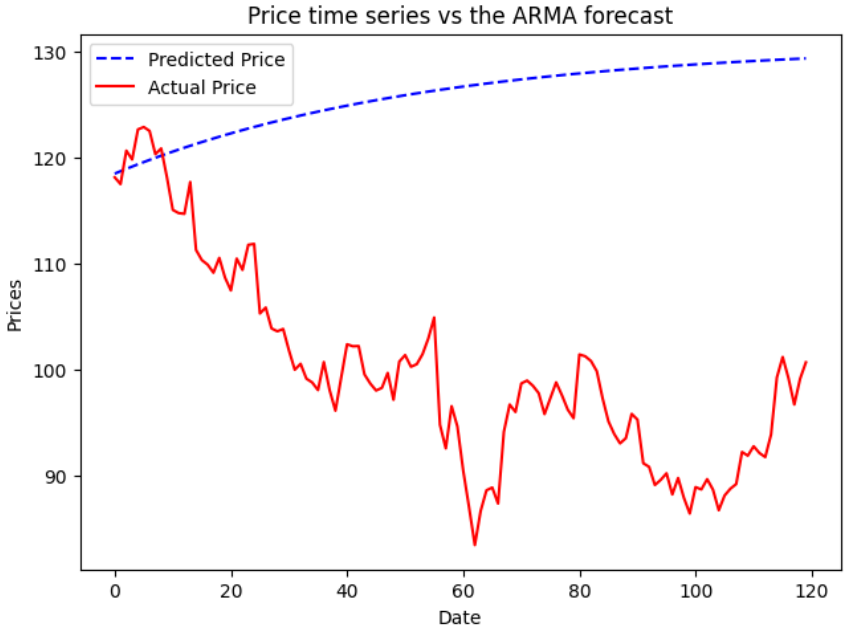
The mean square error (MSE) for the training and test set are shown below:

```
Mean squared error for train:  10.298216574830676
Mean squared error for test:  816.2554748382739
```

The reason for the very high values of MSE in the test set can be accounted for by the fact that an ARMA model is used on stationary data and the close price values were not stationary. We solve this problem in the next questions.

20. **Plot the price time series vs the ARMA forecast in the test set**

Ans. The following plot shows the ARMA forecast of close prices against the actual prices. We can observe that a "one-time forecast" method is not suitable for doing the time series analysis of this data. It is not able to identify the trend and hence, gives a very high value of MSE in the test set as well.



21. **Fit an ARMA model to the return time series**

Ans. We use the *ARIMA* function from the *statsmodel* module of python to build an ARMA model. To get an ARMA model, we use $\mathbf{d = 0}$ i.e. number of non-seasonal differences needed for stationarity, in the *ARIMA* function. We use $p = 1, d = 0, q = 1$ for building the ARMA model with linear return prices of the time series.

22. **Display the parameters of the model and its Mean Squared Error (MSE) in the training set and in the test set**

Ans. The parameters of the ARMA model for linear return of close prices are as follows:

```
==============================================================================
Dep. Variable:                      y   No. Observations:                  179
Model:                 ARIMA(1, 0, 1)   Log Likelihood                 421.661
Date:                Thu, 16 Feb 2023   AIC                           -835.322
Time:                        02:07:07   BIC                           -822.572
Sample:                             0   HQIC                          -830.152
                                  - 179
Covariance Type:                  opg
==============================================================================
                 coef    std err          z      P>|z|      [0.025      0.975]
------------------------------------------------------------------------------
const         -0.0011      0.002     -0.639      0.523      -0.004       0.002
ar.L1         -0.6992      0.475     -1.473      0.141      -1.629       0.231
ma.L1          0.6362      0.516      1.233      0.218      -0.375       1.648
sigma2         0.0005   5.23e-05     10.073      0.000       0.000       0.001
===================================================================================
Ljung-Box (L1) (Q):                   0.00   Jarque-Bera (JB):                 5.63
Prob(Q):                              0.99   Prob(JB):                         0.06
Heteroskedasticity (H):               1.86   Skew:                             0.36
Prob(H) (two-sided):                  0.02   Kurtosis:                         3.47
===================================================================================
```

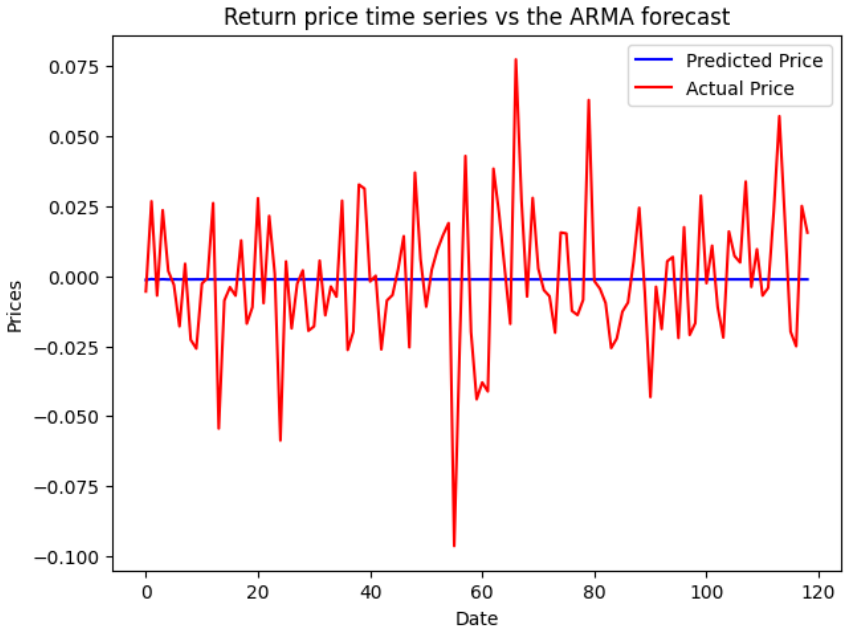
The mean square error (MSE) for the training and test set are shown below:

```
Mean squared error for train:  0.0005265163117657955
Mean squared error for test:  0.0005763401745449846
```

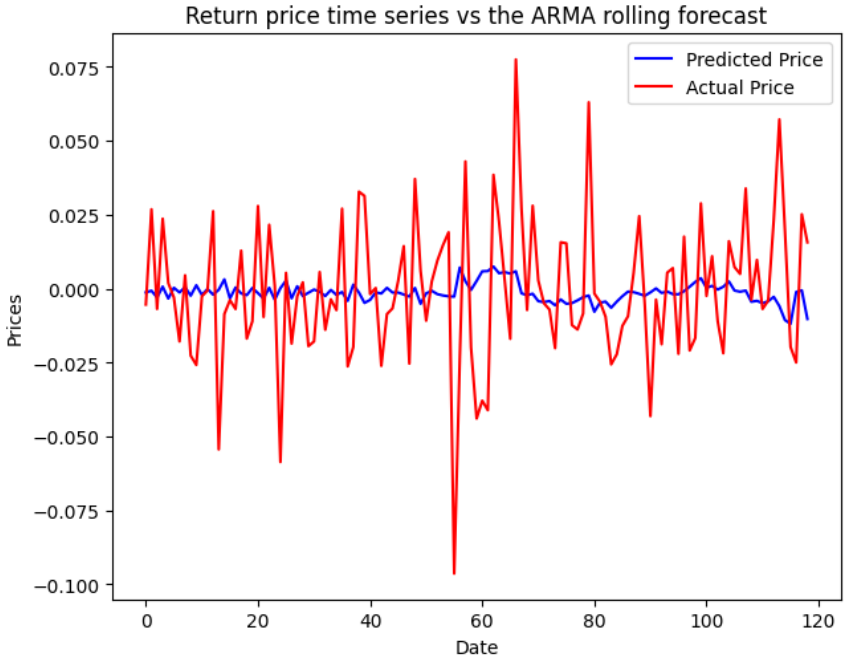
We can observe a significant drop in the mse because we are now applying an ARMA model to stationary time series.

23. **Plot the return time series vs the ARMA forecast in the test set**

Ans. The plot below represents the ARMA forecast against the test set close prices.



We can observe that even with very less MSE, they do not give optimal results with the used "one-time forecast" approach so we try "rolling forecast" method which is the process of continuously updating and refining the forecast of a time series by re-estimating the model parameters using the most recent data available. Although there is not much of a difference in MSE, our outcomes look more interpretable, as shown in the plot below.



Gaussianity and Stationarity test

24. **Introduce mathematically a Gaussianity test**

Ans. Mathematically, the gaussianity or normality test is used to determine if a data set is well-modeled by a normal distribution and to compute how likely it is for a random variable underlying the data set to be normally distributed. The Shapiro-Wilk test and the Kolmogorov-Smirnov test are two frequently used techniques to check the normality of time series data. We will discuss the mathematics behind the Shapiro-wilk test for this report.

The hypothesis testing used for this test is :

H_0 : the sample belongs to a normal distribution
H_1 : the sample doesn't belong to a normal distribution

The test statistic of the Shapiro-Wilk test is given by :

$$W = \frac{(\sum_i^n a_i r_{(i)})^2}{\sum_i^n (r_{(i)} - \mu)^2}$$

where, n is the number of observations, $r_{(i)}$ is the i^{th} smallest value in the sample, μ is the sample mean, coefficient values a_i is given by:

$$a_i = \frac{m^T V^{-1}}{\mathcal{N}}$$

where, $\mathcal{N}$ is the normalizing factor such that $\sum_i a_i^2 = 1$, m is the vector of expected values of all the order statistics in a Gaussian distribution, and V is the expected covariance of pairs of order statistics.

25. **Perform a Gaussianity test of the return time series**

Ans. We use the Shapiro-wilk test to test the gaussianity of the linear returns of the close prices in the time series. With a $p < 0.05$ where 5% is taken as a significance level, we reject the null hypothesis and conclude that the return time series is not normal (or gaussian).

```
Statistic = 0.986416, p = 0.006441
Sample does not look Gaussian (reject H0)
```

26. **Introduce mathematically a stationarity test**

Ans. A stationarity test determines if the statistical properties of a time series, such as mean, variance, standard deviation, and covariance, do not change with time. Two widely used stationarity tests used to examine stationarity in time-series data are the Augmented Dickey-Fuller (ADF) test and the Kwiatkowski-Phillips-Schmidt-Shin (KPSS) test. We have used the Augmented Dickey-Fuller test for our analysis in this report. The ADF test is an 'augmented' version of the Dickey-Fuller test, which tests whether a time series has a unit root indicating that it is non-stationary. The ADF test includes additional terms in the regression equation and handles more complex time series. Also, there are exactly as many unit roots in the series as differencing operations needed to make the series stationary. The hypothesis testing used by ADF test is:

H_0 : *the series is non-stationary/has a unit root*
H_1 : *the series is stationary/has no unit root*

The equation for the ADF test is :

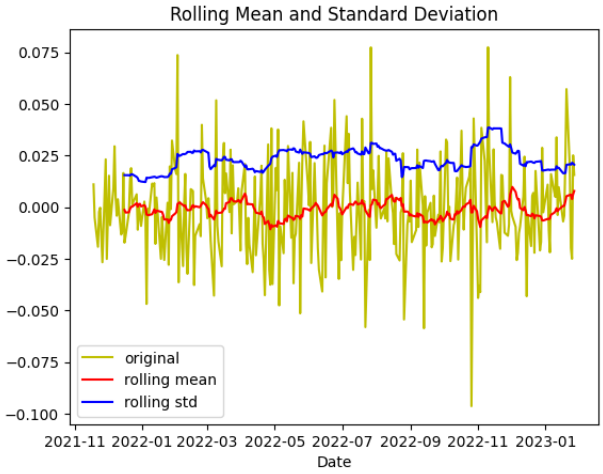
$$y_t = c + \beta t + \alpha y_{t-1} + \phi_1 \Delta Y_{t-1} + \phi_2 \Delta Y_{t-2} + \cdots + \phi_p \Delta Y_{t-p} + e_t$$

where, Y_{t-1} = lag 1 of time series and $\Delta\, Y_{t-1}$ is first difference of time series at time (t-1).

The unit root test is then carried out under the null hypothesis $\alpha = 0$ against the alternative hypothesis of $\alpha < 0$. t-test of H_0 is denoted as the adf test.

27. **Perform a stationarity test of the return time series**

Ans. The following figure shows the rolling mean and standard deviation of the return time series. We can observe that they do not change much with time. Thus the results of the Augmented Dickey fuller test, with a 5% significance level, show that the data is stationary.




```
Results of dickey fuller test
ADF Test Statistic : -17.646739051586813
p-value : 3.7366334439393935e-30
#Lags Used : 0
Number of Observations Used : 298
Strong evidence against the null hypothesis (Ho), reject the null hypothesis. Data is stationary
```

References

1. https://www.quantstart.com/articles/Autoregressive-Moving-Average-ARMA-p-q-Models-for-Time-Series-Analysis-Part-3/
2. https://towardsdatascience.com/time-series-from-scratch-autocorrelation-and-partial-autocorrelation-explained-1dd641e3076f
3. https://machinelearningmastery.com/a-gentle-introduction-to-normality-tests-in-python/
4. https://en.wikipedia.org/wiki/Augmented_Dickey%E2%80%93Fuller_test
5. https://medium.com/@cmukesh8688/why-is-augmented-dickey-fuller-test-adf-test-so-important-in-time-series-analysis-6fc97c6be2f0