Journal of Emerging Strategies in New Economics

The Netherlands Press

Article

STOCK PRICE PREDICTION USING TIME SERIES FORECASTING BY MACHINE LEARNING MODELS

***[1]Sajaratuddur, [2]Lelya Hilda**

[1]Faculty of Chemical Engineering,
Universitas Islam Negeri Sumatera Utara, India

Orchid ID: https://orcid.org/0000-0002-4820-8631

Email: sajaratuddur@uinsu.ac.id

[2]Faculty of Chemical Engineering,
Institute Agama Islam Negeri Padangsidimpuan, INDONESIA,

Orchid ID: https://orcid.org/0000-0002-0607-1761

Email: lelyahilda@iain-psp.ac.id

Abstract.

It is a highly obvious fact that, the stock market is a fickle beast, and making forecasts may be difficult. Stock prices are impacted by both economic and non-economic variables. Refers to several essential physical, psychological, rational, and so on factors. The stock price is predicted using the autoregressive integrated migration Average (ARIMA) model in this research article. have a model for predicting stock prices. Create and disseminate obtained inventory data from Yahoo Finance on a regular basis. The experimental findings show that ARIMA models may be used to accurately estimate inventory levels and short-term pricing.

Keywords: ARIMA, Root mean square error, Finance, Time series, Short term pricing, seasonal decomposition, log decomposition.

Journal of Emerging Strategies in New Economics Volume 1 Issue 1

Received Date: 10 July 2022
Accepted Date: 14 August 2022
Published Date: 07 September 2022

1. Introduction

Stock movement prediction has long been a fascinating issue that has been thoroughly researched by experts from numerous disciplines. Machine learning, a well-established technique in a number of applications, has been actively researched for its ability to anticipate financial markets. Popular algorithms like Support Vector Machines (SVM) and Reinforcement Learning are said to be extremely successful at following stock markets and optimising profits on stock option acquisitions while minimising risks [1]. However, in many of these cases, the qualities used as input to machine learning algorithms are generated mostly from data from the same impacted industry. Because of this isolation, critical information conveyed by other entities is lost, and prediction results are subject to local influence.

External disturbances to financial markets are no longer limited to national borders as globalisation deepens links across international economies. Every day, new data is generated as capital markets change. The intrinsic value of a company's stock is calculated by assessing the stock's projected future cash flows and discounting them to their present value, also known as book value. This is distinct from the stock's market value, which is calculated by the share price of the corporation [6]. This stock's market value may change for causes unrelated to the company's core activities, such as: B. Market sentiment deviates from intrinsic value. The stock market is volatile, and there are several sophisticated financial measures to consider. Few people have the comprehensive expertise and understanding to grasp the indications and use them to create accurate forecasts in order to generate riches. To earn money trading stocks, most people must rely purely on chance. However, technological improvements provide prospects for consistent profits in the stock market and assist professionals in identifying the most profitable indications to create better forecasts. is critical to maximise income while limiting risk from stock option acquisitions[4].

ARIMA is a time series forecasting model that uses autocorrelation measurements to characterise the temporal structure of time series data and anticipate future values. The autoregressive component of the model calculates a sample's reliance on past data. These differences are assessed and aggregated to make data patterns stable or to eliminate apparent connections with earlier data (a core assumption of linear regression models is linear independence rather than collinearity). Moving averages can then be used to summarise and emphasise essential aspects of the data[7].

Autoregressive models are conceptually similar to linear regression models, and the latter assumptions apply here as well. To eliminate visible correlations and collinearity with preceding data, time series data should be steady. The features or values of sample observations in stationary time series data do not depend on the observed timestamps. For example, if you look at a hypothetical data set of a region's annual population and see that it doubles or rises by a constant amount each year, the data is not stable. Certain observations are strongly year-dependent, since population figures are affected by their proximity to preceding years. When training models using time series data, this reliance might lead to erroneous biases[3].

Because of the ARIMA model's capacity to create a lengthy sequence of predictions and projections from a single trained model, it may be used widely to anticipate the following 7 days' weather or market performance. You may also observe sales swings and calculate how many new clients to sign up for the year.

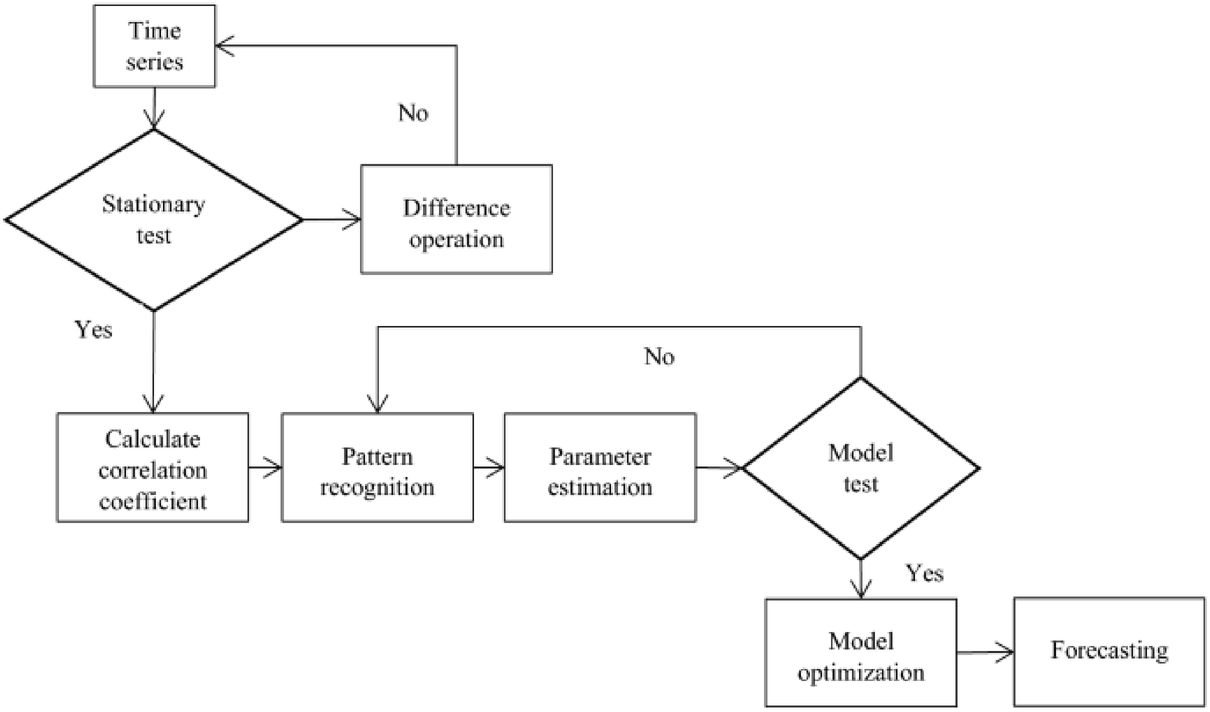
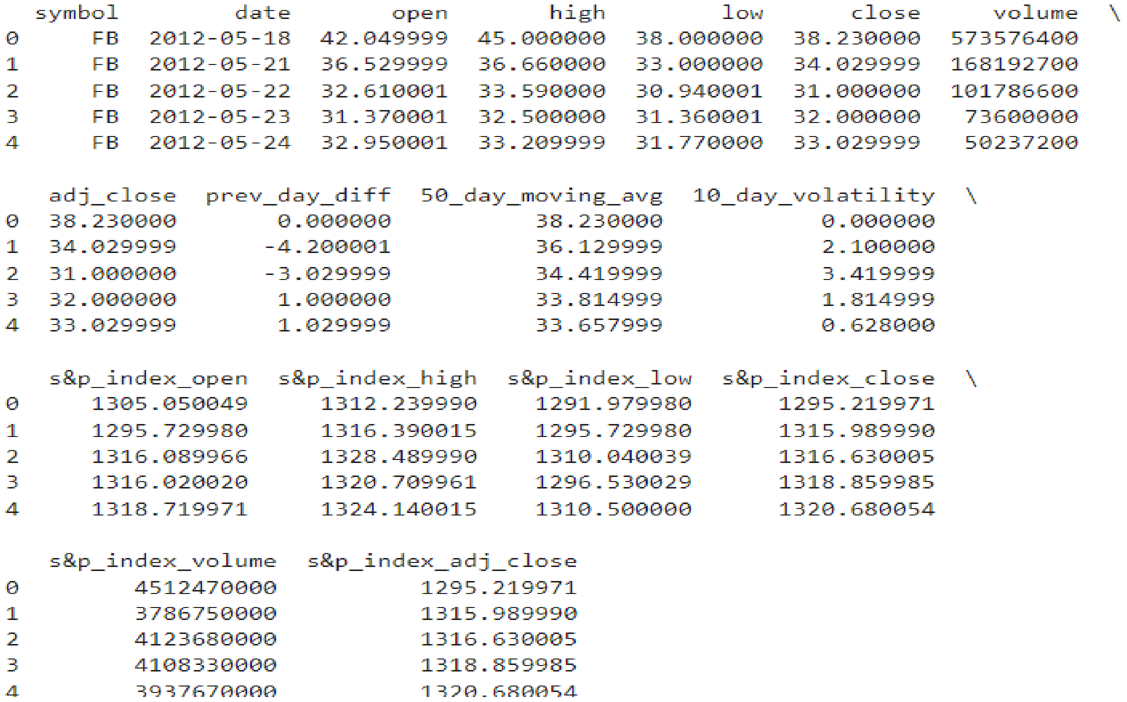


Figure 1:Flow chart of prediction process[4]

2. IMPLEMENTATION STOCK PREDICTION:

2.1. DATA HANDLING:

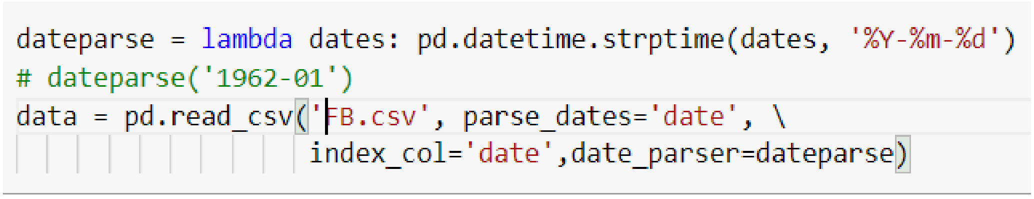
First and foremost, for Supervised learning features and labels are important for the prediction. However, model is trained using data and label. For this problem company's stock data is taken from the resource[3]. The features in that data are date, open, high, low, close, volume, adj_close, prev_day_diff , 50_day_morning_avg, 10_day_volatility and some columns are present as given below.



```
  symbol        date       open       high        low      close     volume  \
0     FB  2012-05-18  42.049999  45.000000  38.000000  38.230000  573576400
1     FB  2012-05-21  36.529999  36.660000  33.000000  34.029999  168192700
2     FB  2012-05-22  32.610001  33.590000  30.940001  31.000000  101786600
3     FB  2012-05-23  31.370001  32.500000  31.360001  32.000000   73600000
4     FB  2012-05-24  32.950001  33.209999  31.770000  33.029999   50237200

   adj_close  prev_day_diff  50_day_moving_avg  10_day_volatility  \
0  38.230000       0.000000          38.230000           0.000000
1  34.029999      -4.200001          36.129999           2.100000
2  31.000000      -3.029999          34.419999           3.419999
3  32.000000       1.000000          33.814999           1.814999
4  33.029999       1.029999          33.657999           0.628000

   s&p_index_open  s&p_index_high  s&p_index_low  s&p_index_close  \
0     1305.050049     1312.239990    1291.979980      1295.219971
1     1295.729980     1316.390015    1295.729980      1315.989990
2     1316.089966     1328.489990    1310.040039      1316.630005
3     1316.020020     1320.709961    1296.530029      1318.859985
4     1318.719971     1324.140015    1310.500000      1320.680054

   s&p_index_volume  s&p_index_adj_close
0        4512470000          1295.219971
1        3786750000          1315.989990
2        4123680000          1316.630005
3        4108330000          1318.859985
4        3937670000          1320.680054
```

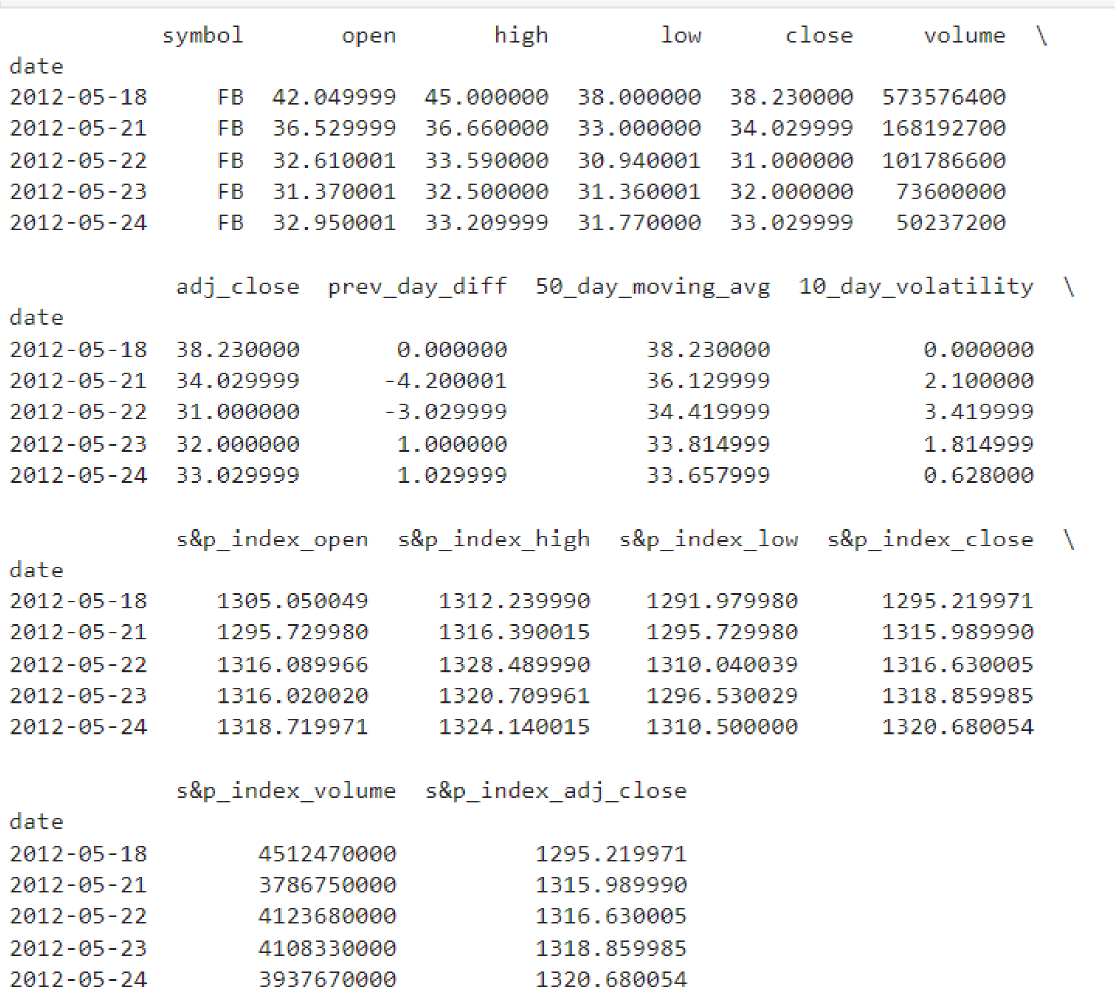
Figure 2: stock data

Data Handling is done according to the date of stock and the dates are parsed into the data for time series forecasting. By sing pandas date parsing is done as shown in the code snippet.



```
dateparse = lambda dates: pd.datetime.strptime(dates, '%Y-%m-%d')
# dateparse('1962-01')
data = pd.read_csv('FB.csv', parse_dates='date', \
                   index_col='date',date_parser=dateparse)
```

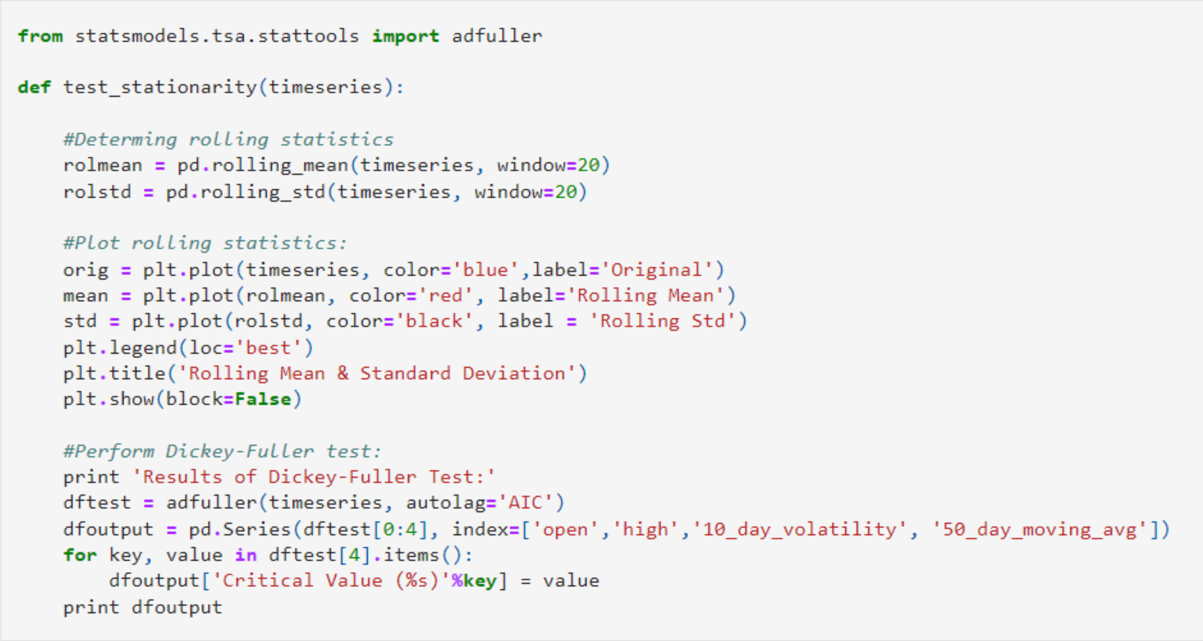
And the resulting output data is formed as shown below the first column in the data is date column and the remaining columns are formed as in the dataset.



```
           symbol       open       high        low      close     volume  \
date
2012-05-18     FB  42.049999  45.000000  38.000000  38.230000  573576400
2012-05-21     FB  36.529999  36.660000  33.000000  34.029999  168192700
2012-05-22     FB  32.610001  33.590000  30.940001  31.000000  101786600
2012-05-23     FB  31.370001  32.500000  31.360001  32.000000   73600000
2012-05-24     FB  32.950001  33.209999  31.770000  33.029999   50237200

            adj_close  prev_day_diff  50_day_moving_avg  10_day_volatility  \
date
2012-05-18  38.230000       0.000000          38.230000           0.000000
2012-05-21  34.029999      -4.200001          36.129999           2.100000
2012-05-22  31.000000      -3.029999          34.419999           3.419999
2012-05-23  32.000000       1.000000          33.814999           1.814999
2012-05-24  33.029999       1.029999          33.657999           0.628000

            s&p_index_open  s&p_index_high  s&p_index_low  s&p_index_close  \
date
2012-05-18     1305.050049     1312.239990    1291.979980      1295.219971
2012-05-21     1295.729980     1316.390015    1295.729980      1315.989990
2012-05-22     1316.089966     1328.489990    1310.040039      1316.630005
2012-05-23     1316.020020     1320.709961    1296.530029      1318.859985
2012-05-24     1318.719971     1324.140015    1310.500000      1320.680054

            s&p_index_volume  s&p_index_adj_close
date
2012-05-18        4512470000          1295.219971
2012-05-21        3786750000          1315.989990
2012-05-22        4123680000          1316.630005
2012-05-23        4108330000          1318.859985
2012-05-24        3937670000          1320.680054
```

2.2. Stationarity check:

Matplotlib is the library I used to test the stationarity of time series data. For each date in the data, this library generates adj near data. A statistical model that tests for stationarity is used to build this function. The pandas library uses rolling mean to calculate the rolling mean and rolling standard deviation of your data. The Dickey-Fuller test in statistics then evaluates the null hypothesis that the autoregressive time series model has a root of one for these three (original, moving average, moving standard deviation, and Dickey-Fuller test). The alternative is different, however it is generally stationarity or trend stationarity, depending on the version of the test employed as a result.



```
from statsmodels.tsa.stattools import adfuller

def test_stationarity(timeseries):

    #Determing rolling statistics
    rolmean = pd.rolling_mean(timeseries, window=20)
    rolstd = pd.rolling_std(timeseries, window=20)

    #Plot rolling statistics:
    orig = plt.plot(timeseries, color='blue',label='Original')
    mean = plt.plot(rolmean, color='red', label='Rolling Mean')
    std = plt.plot(rolstd, color='black', label = 'Rolling Std')
    plt.legend(loc='best')
    plt.title('Rolling Mean & Standard Deviation')
    plt.show(block=False)

    #Perform Dickey-Fuller test:
    print 'Results of Dickey-Fuller Test:'
    dftest = adfuller(timeseries, autolag='AIC')
    dfoutput = pd.Series(dftest[0:4], index=['open','high','10_day_volatility', '50_day_moving_avg'])
    for key, value in dftest[4].items():
        dfoutput['Critical Value (%s)'%key] = value
    print dfoutput
```

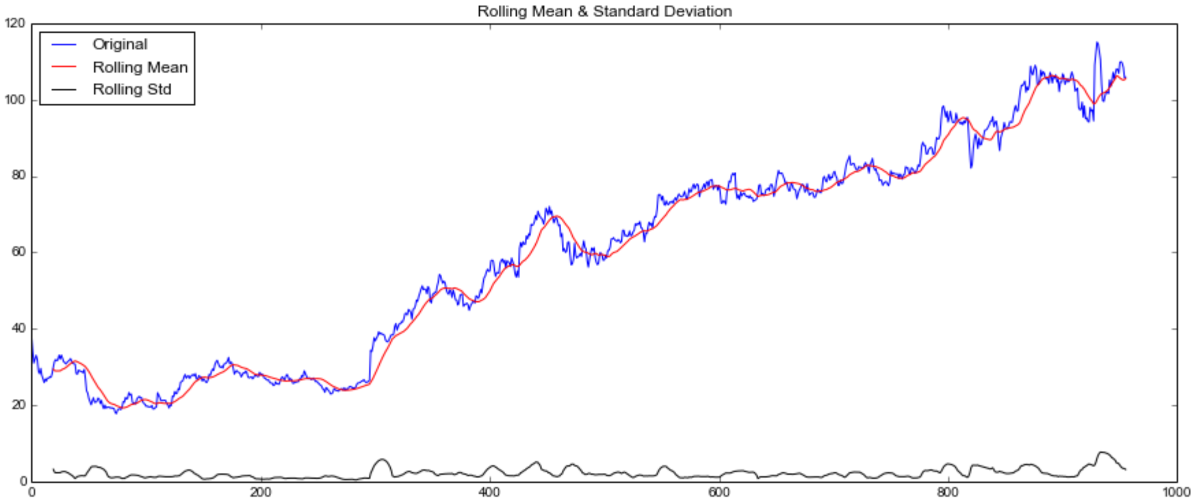
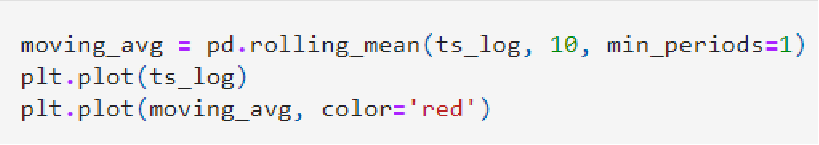
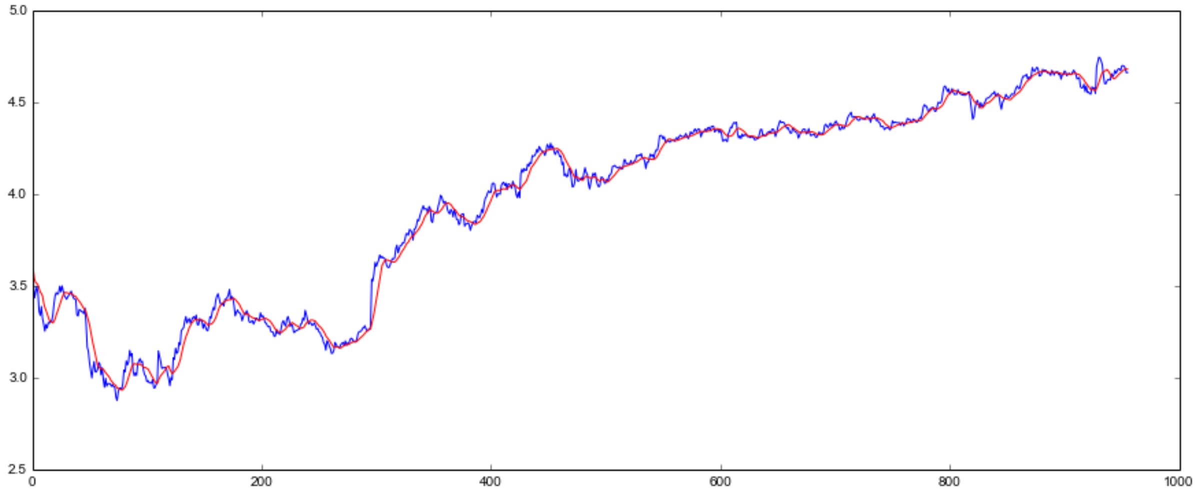


Figure 3: smoothness test using Dickey test

To smooth the curve, must eliminate a portion of the trend. Because this section is perplexing for the series. After determining the first moving average, the remaining values are subtracted from the original value. This occurs during the smoothing process.



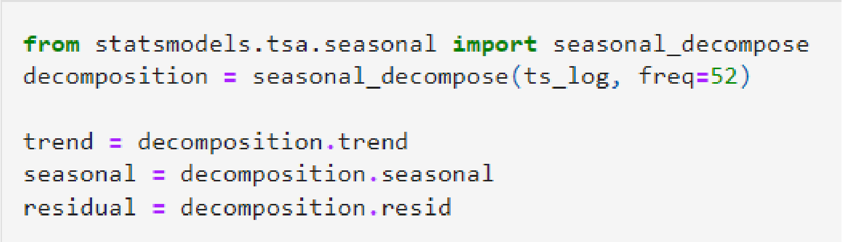
```
moving_avg = pd.rolling_mean(ts_log, 10, min_periods=1)
plt.plot(ts_log)
plt.plot(moving_avg, color='red')
```



3. SEASONAL DECOMPOSITION:

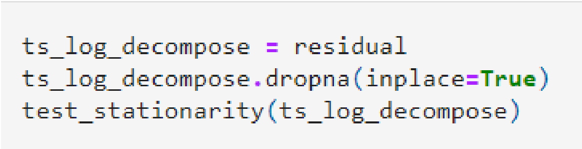
A result object is returned by the seasonal decompose() method. The result object includes arrays that allow you to access the four data pieces from the decomposition. For example, using an additive model, the following sample explains how to deconstruct a data series into trend, seasonal, and residual components. This is known as detrending. Time series data are frequently seen to have several components, including long-term patterns, seasonal changes, and erratic fluctuations. The initial step is to calculate the trend.

The second step is to remove any trends from the series. After that, the detrended dataset is utilized to estimate seasonal variables. The last step is to find the random (irregular) component.



```
from statsmodels.tsa.seasonal import seasonal_decompose
decomposition = seasonal_decompose(ts_log, freq=52)

trend = decomposition.trend
seasonal = decomposition.seasonal
residual = decomposition.resid
```

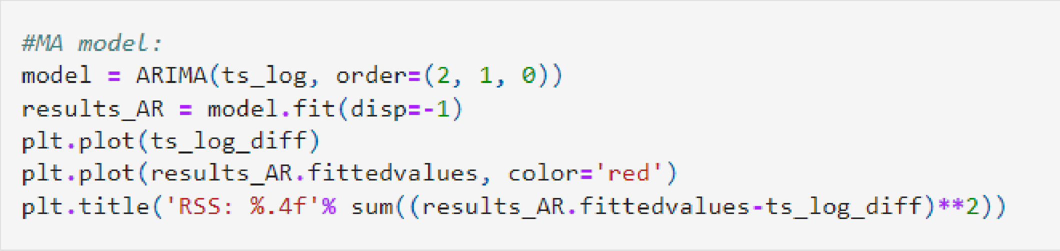
3.1. Log decomposition:



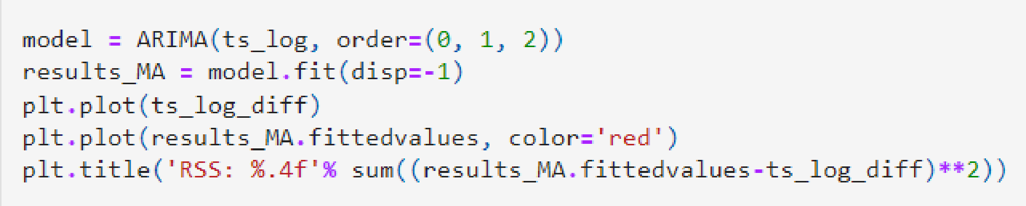
```
ts_log_decompose = residual
ts_log_decompose.dropna(inplace=True)
test_stationarity(ts_log_decompose)
```

3.2. Using ARIMA model for forecasting the time series data:

The word ARIMA stands for "Autoregressive Integrated Moving Average." It is a model used in statistics and econometrics to quantify events that occur over time. Models are used to interpret previous data or to forecast a collection of future data. ARIMA models are a type of statistical model that is used to analyze and forecast time series data. It clearly conforms to a set of time series data standard structures and provides a simple and powerful technique to generate advanced time series forecasts. The ARMA model is a constant model. If the model is not stationary, a sequence of differencing can be used to attain stationarity. The ARIMA model's "I" stands for integration. This is the number of nonseasonal fluctuations needed to ensure stationarity.



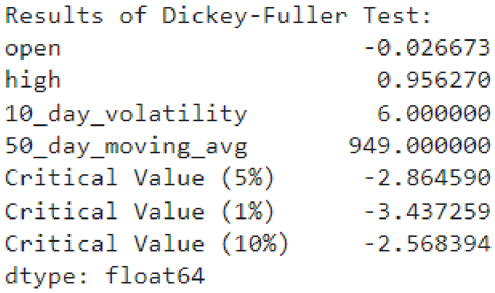
```
#MA model:
model = ARIMA(ts_log, order=(2, 1, 0))
results_AR = model.fit(disp=-1)
plt.plot(ts_log_diff)
plt.plot(results_AR.fittedvalues, color='red')
plt.title('RSS: %.4f'% sum((results_AR.fittedvalues-ts_log_diff)**2))
```



```
model = ARIMA(ts_log, order=(0, 1, 2))
results_MA = model.fit(disp=-1)
plt.plot(ts_log_diff)
plt.plot(results_MA.fittedvalues, color='red')
plt.title('RSS: %.4f'% sum((results_MA.fittedvalues-ts_log_diff)**2))
```

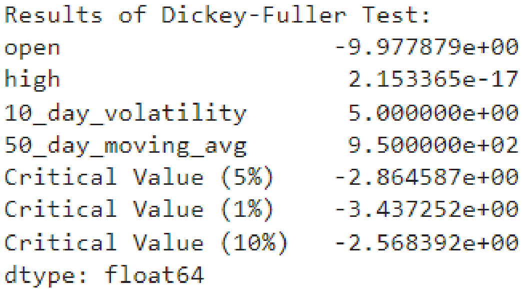
4. RESULTS AND DISCUSSIONS

First for the data Dickey fuller test is done and the results obtained are shown below



```
Results of Dickey-Fuller Test:
open                    -0.026673
high                     0.956270
10_day_volatility        6.000000
50_day_moving_avg      949.000000
Critical Value (5%)     -2.864590
Critical Value (1%)     -3.437259
Critical Value (10%)    -2.568394
dtype: float64
```

After removing the value during smoothing, repeat the steady state test. The original moving average and standard deviation of the Dickey-Fuller test are sung again in this test, and the results are acquired; graphs are provided below.



```
Results of Dickey-Fuller Test:
open                  -9.977879e+00
high                   2.153365e-17
10_day_volatility      5.000000e+00
50_day_moving_avg      9.500000e+02
Critical Value (5%)   -2.864587e+00
Critical Value (1%)   -3.437252e+00
Critical Value (10%)  -2.568392e+00
dtype: float64
```

The test results are now getting in the way. The test's Extended Dickey-Fuller Statistic (ADF) is a negative value. At a certain confidence level, the hypothesis that there is a root of unity is rejected the more negative it is. The data is plotted after seasonal decomposition, and the resulting plot is presented below, along with seasonality residuals.

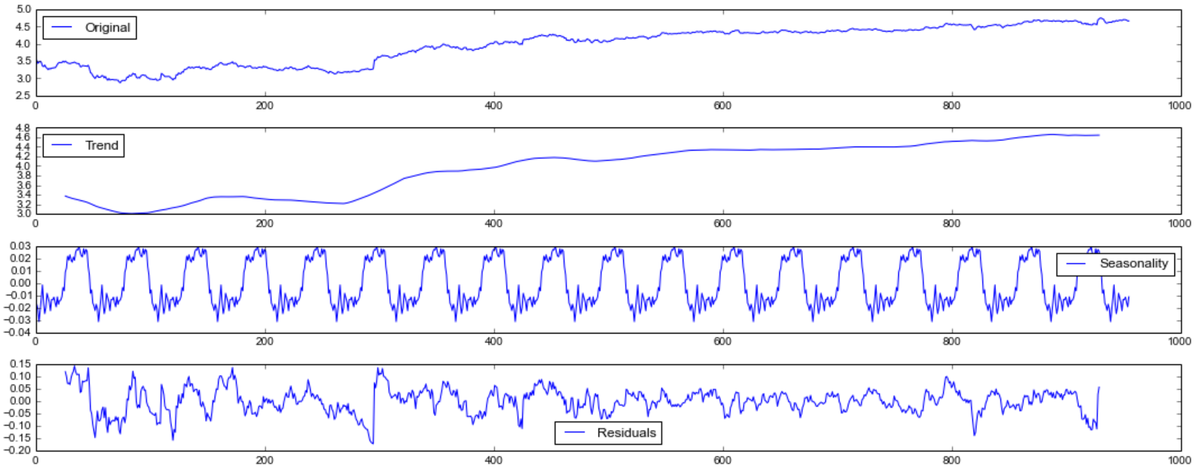


Figure 4: Seasonal Decomposition

The final plot is displayed in the image below after logarithmic decomposition of the residuals.

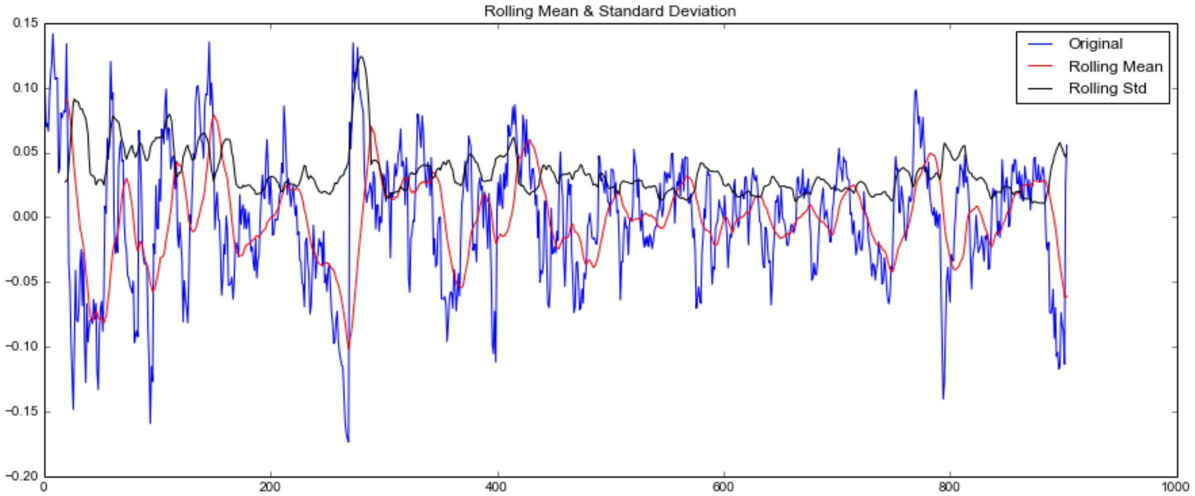


Figure 5: Log Decomposition

Plotting the time series data using the Arima model and then predicting with the model produced the result with the least difference between the original and forecasted. As a result, this graph demonstrates that this approach produces accurate forecasts for time series issues. RSS is used as an assessment statistic in model evaluation. It is the square of the difference between the difference's logarithm and its estimated value. This number is extremely low; the obtained value is 0.69, which is on the order of (2,1,0).

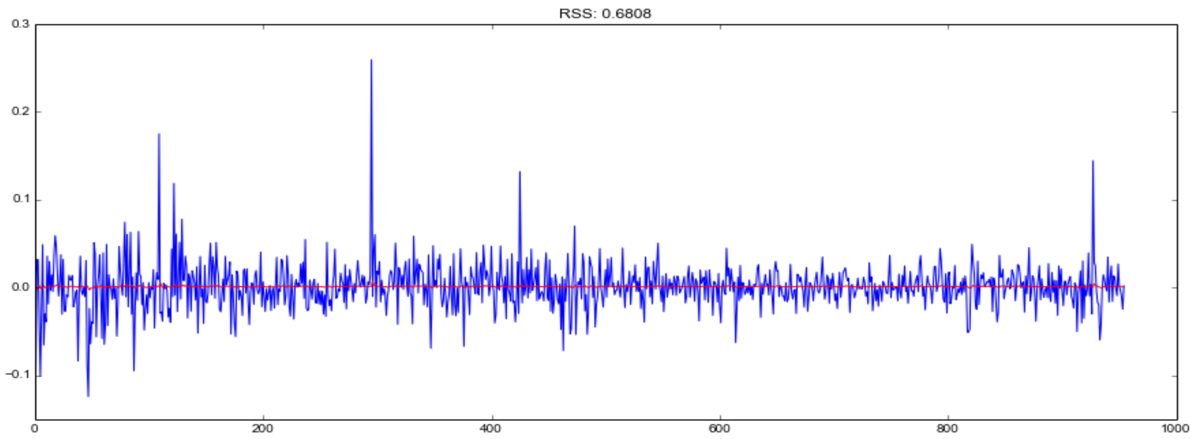


Figure 6: Arima model(2,1,0)

This is obtained for other predictions like (0,1,2) the obtained graph is shown below and the value of RSS is obtained as 0.68

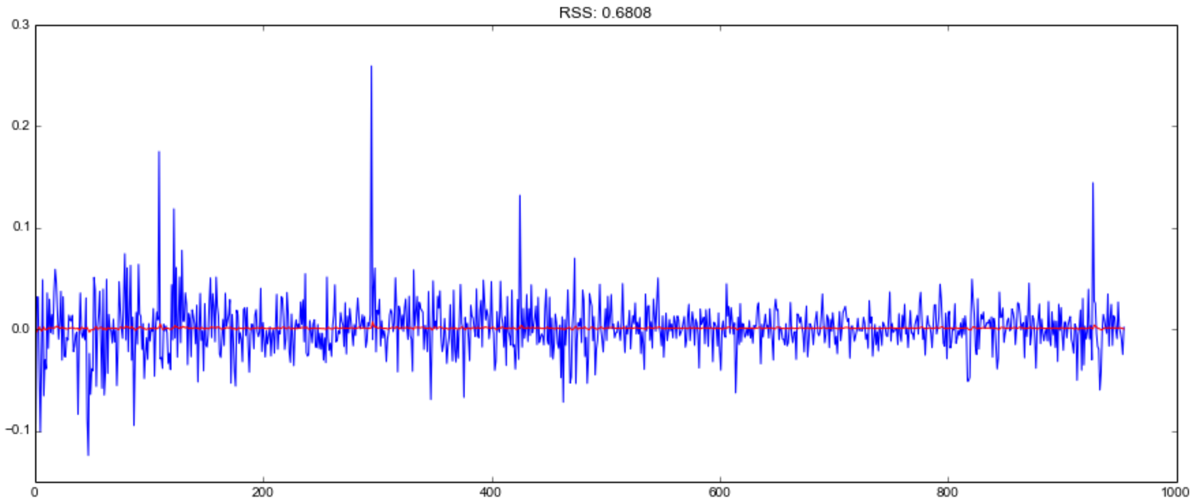


Figure 7: ARIMA model(0,1,2)

This is obtained for other predictions like (2,1,2) the obtained graph is shown below and the value of RSS is obtained as 0.67

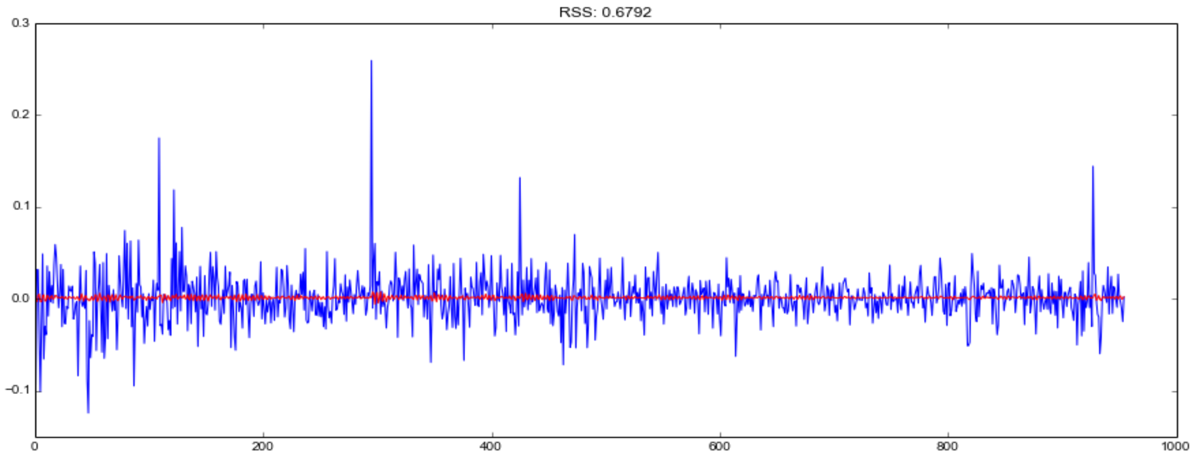


Figure 8: Arima model (2,1,2)

Using the RMSE to assess the error after converting the logarithmic scale back to the original scale. The standard deviation of the residuals is defined as the root mean square error (RMSE) (prediction errors). The distance between the data points and the regression line is measured by residuals. The RMSE is a measure of the distribution of these residuals. In other words, it indicates the degree to which the data is crowded around the line of best fit.

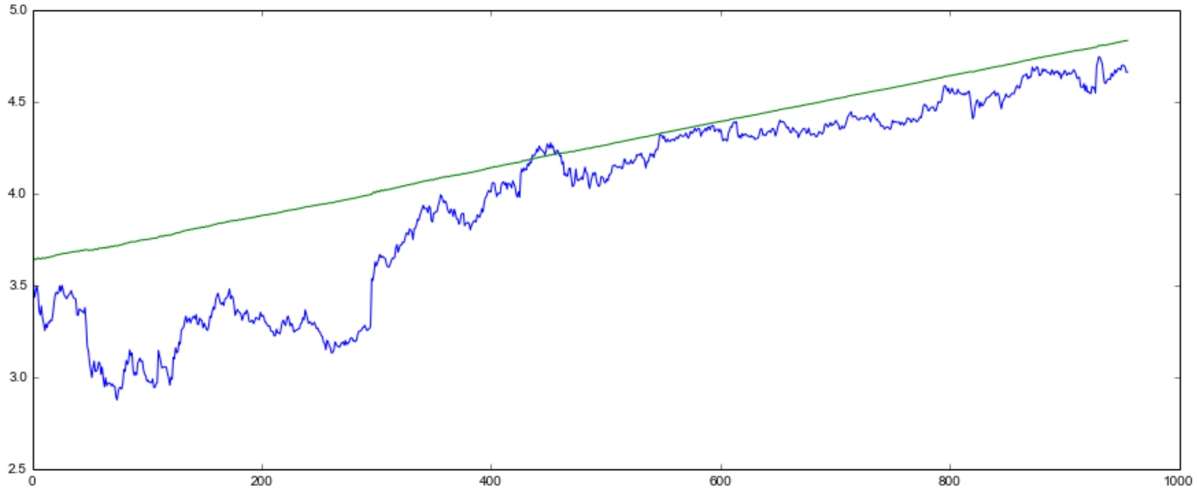


Figure 9: Final prediction evaluation

For the prediction the RMSE value obtained is 15.13 and it is low for these kind of data and hence , this machine learning model can be used for prediction of data.

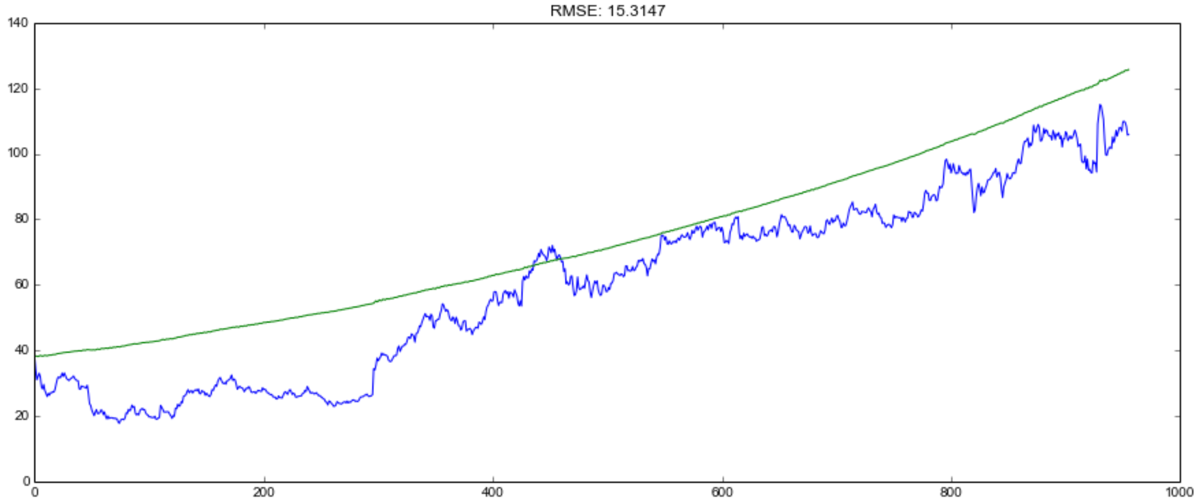


Figure 10: RMSE values for prediction

5. CONCLUSION

Stock price volatility makes forecasting harder. The experimental research in this work demonstrates that predictive models, particularly his ARIMA model, may be utilised to accurately anticipate future stock values. To test this premise, examined particular examples from ICICI Bank and Reliance Industries. The sole disadvantage of this investigation is that his ARIMA model performs better in terms of short-term predicting accuracy.

References

[1] SHEIKH MOHAMMAD IDREES, M. AFSHAR ALAM, PARUL AGARWAL," A Prediction Approach for Stock Market Volatility Based on Time Series Data", IEEE Access, vol. 7, pp. 17287-17298, 2019.

[2] Wint Nyein Chan," Time Series Data Mining: Comparative Study of ARIMA and Prophet Methods for Forecasting Closing Prices of Myanmar Stock Exchange", Journal of Computer Applications and Research, vol. 1, pp.75-80, 2020.R. Nicole, "Title of paper with only first word capitalized," J. Name Stand. Abbrev., in press.

[3] Reliance Industries Historical Stock Data, https://finance.yahoo.com/quote/RELIANCE.BO?p=RELIANCE.BO&.tsrc=fin-srch , [Online; Accessed March 2020]

[4] ICICI Bank Historical Stock Data, https://finance.yahoo.com/quote/ICICIBANK.BO/history?p=ICICIBANK.BO , [Online; Accessed March 2020]

[5] Closing Price, https://www.investor.gov/introductioninvesting/investing-basics/glossary/closing-price, [Online; Accessed March 2020]

[6] A. A. Ariyo, A. O. Adewumi and C. K. Ayo, "Stock Price Prediction Using the ARIMA Model," 2014 UKSim-AMSS 16th International Conference on Computer Modelling and Simulation, Cambridge, UK, 2014, pp. 106-112, doi: 10.1109/UKSim.2014.67.

[7] Leybourne, Stephen, Tae‐Hwan Kim, and Paul Newbold. "Examination of some more powerful modifications of the Dickey‐Fuller test."