CS815 Assignment 2 - Automatically Formulating Trading Rules

Group 11 Linu Roby/ Deepali Yadav

2024-03-30

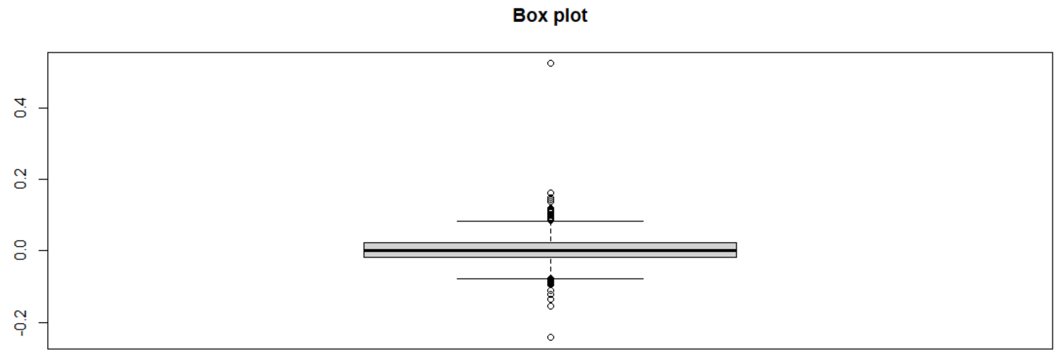
Background to the problem

Algorithmic Trading is the use of programs to assist in any aspect of trading financial assets and encompasses systems which may decide on timing, price or quantity of order. Algorithmic trading has been proved very efficient where the stocks can be traded in milliseconds but at the same time making sure the market movements are utilized at the right time. With algorithmic trading, trade orders can be executed without human intervention and trading outcomes can be optimized by avoiding human errors. In the year of 2019, 70% of traded shares on US stock exchanges were placed automatically (Folger,J.2020) Trade execution implements trades while ensuring low transaction costs(Artificial Intelligence in Asset Management - Söhnke M. Bartram,Jürgen Branke,Mehrshad Motahari,2020). The strategies used in algorithmic trading are based on technical analysis. Here past stock and market data are used to predict returns from future assets. In this coursework, we are using neural network for prediction of next day high and low prices of a stock. The neural network model is trained and used for predictions. A trading strategy is then formulated based on these predictions whether the stock would be bought, sold or put on hold for a day.
The article by Neven Pičuljan briefs about similar work done in hedge funds and finance using deep learning. While model selection, if the neural network is too complex, ie, if it has too many layers or too many neurons in each layer, the model can overfit the data. If the neural network is too simple, then it can underfit the data. In this article, a recurrent neural network is used for finance related task. The final component of the neural network is the linear layer and the final output of the network is the predicted Close price in the time period. During training, the predicted and actual Close prices are compared and the difference is reduced using back propogation algorithm and gradient descent optimization algorithm (Adam is used in this blog). After back testing, the training strategy is evaluated using Sharpe ratio. In reference to the article, the Sharpe ratio shows the ratio between the returns and the additional risk. Larger the Sharpe ratio, the better. In most equilibrium models, expected excess returns are positively related to risk as measured by the expected volatility of the market (Merton, 1980). When volatility is lower (higher) than expected, expected returns decrease (increase), and the corresponding upward (downward) revision of prices is picked up by the trading rules(Journal of Financial Economics,Franklin Allen & Risto Karjalainen,1999). In this project, we are prediting the high and low prices for next day using the values from previous days. We are using neural network which will be trained to get predictions of training and testing data. The training and testing error are

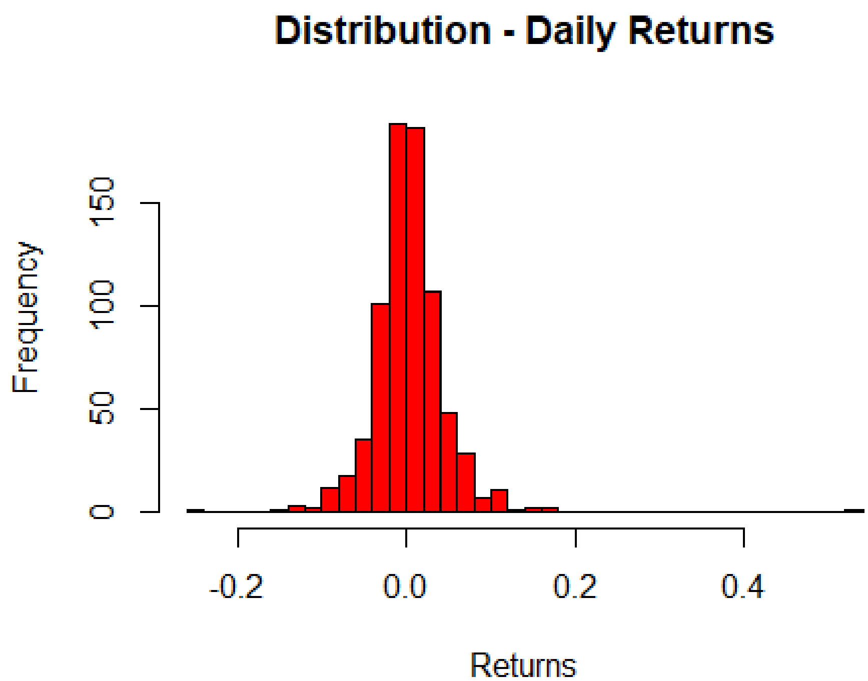
compared and visualised. These predictions will be then used to build a trading rule and to make decision whether and when the stock will be bought or sold.

Overview of Data

As we are applying trade rules, the data we select should ideally be a time series data. The data is taken from Yahoo Finance which will include relevant attributes and required technical indicators for the analysis. Data for a period of 3 years(2016-2018) is taken for training. For back testing, 25% of this data will be retained. The stock chosen is AMD(Advanced Micro Devices, Inc.) which has high trading volume making it suitable for trading strategies. AMD is a semiconductor company which develops computer processors. The price movements are stable since 2015 after hitting a low price. The box plot shows that there are some outliers. They indicate some large price movements which can be used in the trading strategy. However the stock is not highly volatile. The box is symmetric indicating the daily returns are randomly distributed.

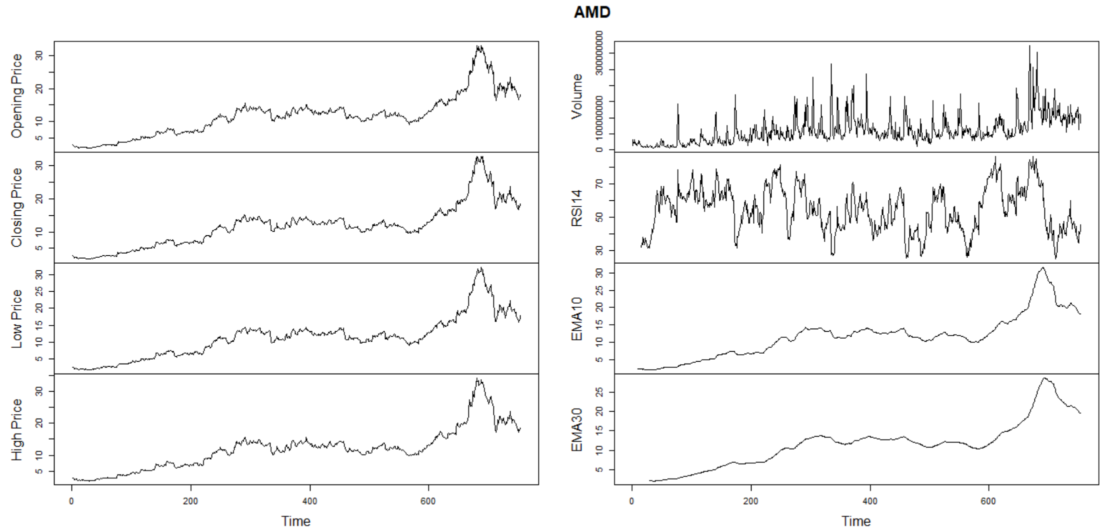


Box plot
0.4
0.2
0.0
-0.2



Distribution - Daily Returns
Frequency
150
100
50
0
-0.2
0.0
0.2
0.4
Returns

Data summary



Price prediction

To implement the trading strategy, the relevant attributes Opening/Closing prices, High/Low prices, Volume and Adjusted prices are extracted. The technical indicators used are EMA, SMA and RSI. The high and low prices are predicted from the lagged version of these inputs. The model is built to predict the price of future day.

Techincal Indicators

Based on the historical price and volume, the technical indicators identify trends and patterns for potential buy/sell. These can be leading or lagging technical indicators. Leading indicators predict future outcomes by looking ahead. This may be misleading and may not be always accurate.Lagging indicators are known after the occurrence of the event and can be useful to confirm the pattern that is occurring over time. In this work, the technical indicators used are SMA, EMA and RSI.

Simple Moving Avergae(SMA)

The aim of all moving averages is to find the direction in which the stock price is moving based on its past prices. SMA combines the price points over a period of time and divides by the total number of data points to give the average presented over a single trend line. If the stock is above the moving average, it denotes an upward trend. This indicates that its a BUY signal, when the stock crosses above the trend. When the stock crosses below the trend, its a SELL signal.

Exponential Moving Average(EMA)

Very similar to SMA but data is more responsive to new information as this places a greater weight on more recent data points.Exponential Moving Averages(EMA) identifies the trend direction to analyse if the price of the stock is showing an uptrend(price is above EMA) or downtrend(price is below EMA). This is one of the technical indicators used here to identify potential buy/sell opportunities. EMA allows traders to react quickly to changing market conditions (coinwofficial.medium.com). It is a usual practice to use EMA in conjunction with other technical indicators which in turn will help to confirm signals and also reduce false alarms.

Relative strength index (RSI)

RSI is a technical indicator which measures speed and change of price movements and give warning signals for dangerous price movements. The value oscillates between 0 and 100. If the value is above 70, the asset is traditionally considered as overbought. If the value is below 30, the asset is considered as oversold.

The relevant attributes are retrieved from the source. The SMA of low and high price are calculated over a period of 10 days. The EMA of adjusted prices over a period of 10 days and 30 days are calculated. RSI is calculated based on the adjusted price for over a typical period of 14 days.

Neural network model

The sequence of observations is transformed into an appropriate form using Lag. This gives a lagged version of the time series data set by shifting the time back by the specified number of observations. The model will be input with 1,2 and 3 days of lagged prices along with SMA,EMA and RSI which were lagged by 1 day. The out put of the model will have the high price and low price.

The inputs are scaled to normalize the data. The scaled parameters are saved for later to retrieve the real values. The trading data is split into training and testing data. 75% of data is allocated for training.

A neural network model is created with 2 hidden layers each having 15 neurons. All the other variables in training data is fed as inputs to formulate the target variables, high price and low price. The threshold is kept as 0.01. This means the weights are not updated if the partial derivative of error with respoect to weight is less than the threshold. This is to ensure that unnecessary weight updates are not happening for insignificant updates. This will prevent the model from slowing down. The stepmax values in kept hihg as 1e+06 to allow the model to run till this value if it fails to converge before that. The training performance of the neural network model stores the predictions on training data. This can be compared with the actual target to perform evaluation on training.

```
nn_model <- neuralnet(as.formula(high_price + low_price ~ .),
                      train_data,
                      hidden = c(15,15),
                      stepmax=1e+06,
                      threshold=0.01)
```

The picture can't be displayed.

Training the model

The scaled attributes which were stored earlier while scaling will be used to unscale the values to get the real values after training.

```
##   high_train_preds high_train_actuals low_train_preds low_train_actuals
## 1            1.976               1.94           1.839              1.85
## 2            1.965               1.98           1.895              1.90
## 3            1.979               1.93           1.910              1.83
##   ema_train_actuals train_day sma__train_actuals sma_hi_train_actuals
## 1             1.922         1              1.892                2.018
## 2             1.918         2              1.881                1.998
## 3             1.916         3              1.876                1.988
##   highprice_errors lowprice_errors
## 1          0.03641       -0.010825
## 2         -0.01527       -0.005208
## 3          0.04925        0.080405
```

The training error obtained is around 0.4 which is indicating that the model is working well on the train dataset.

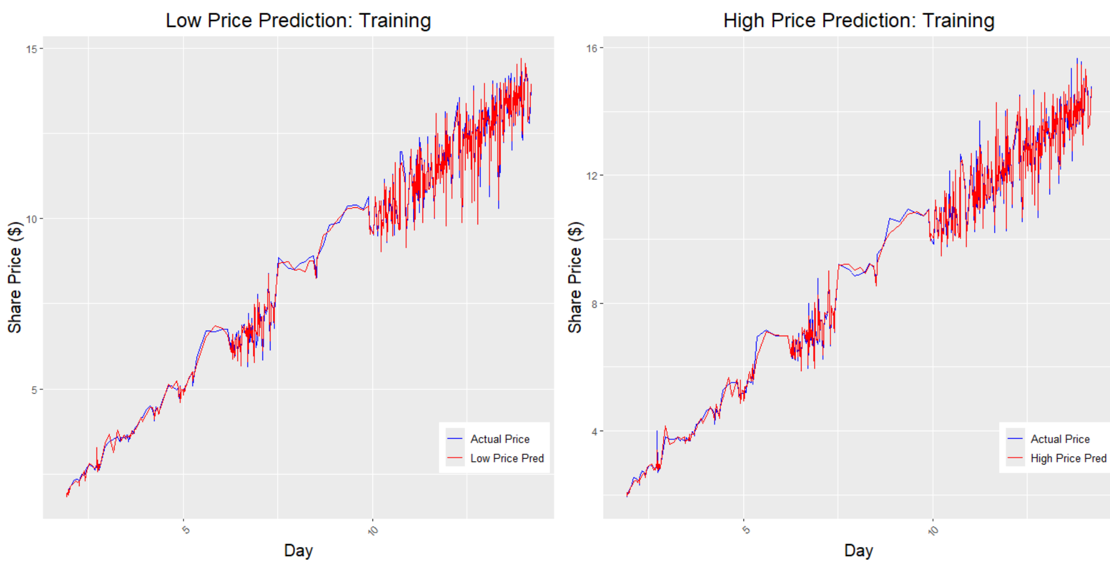
The actual values and predicted values from training data were compared. The model is performing well for training data with low RMSE values on high price and low price.

```
## High predictions error:
##  0.1823
```

```
##
## Low predictions error:
##  0.1688
```

Training performance plot

```
## Warning: A numeric `legend.position` argument in `theme()` was deprecated
in ggplot2
## 3.5.0.
## ℹ Please use the `legend.position.inside` argument of `theme()` instead.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was
## generated.
```



Testing the model

The 25% of data is fed to the model for predictions and the difference between actual values are compared.

```
##   high_train_preds high_train_actuals low_train_preds low_train_actuals
## 1            1.976                1.94           1.839              1.85
## 2            1.965                1.98           1.895              1.90
## 3            1.979                1.93           1.910              1.83
##   ema_train_actuals train_day sma__train_actuals sma_hi_train_actuals
## 1             1.922         1              1.892                2.018
## 2             1.918         2              1.881                1.998
## 3             1.916         3              1.876                1.988
##   highprice_testerrors lowprice_testerrors
## 1             0.284444             0.22764
## 2             0.096731            -0.04651
## 3             0.002541            -0.02894
```

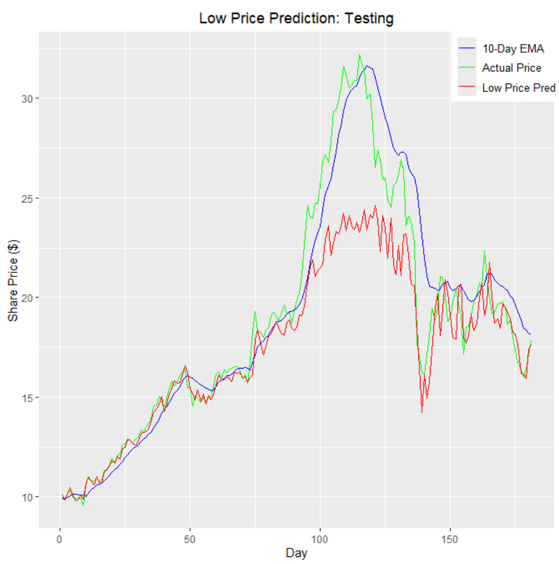
The model is performing not too well for testing data with higher RMSE values on high price and low price as compared to training data.

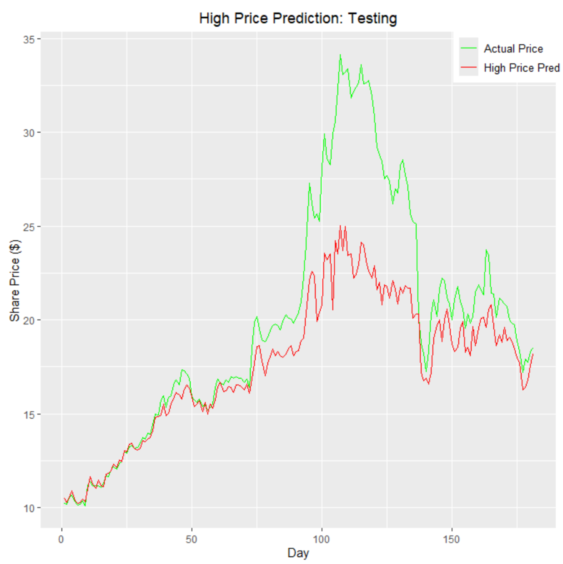
```
## High predictions error:
##  3.698

##
## Low predictions error:
##  2.593
```

Testing performance plot

It is clearly visible that the prediction lines follow the actual line which shows the accuracy of the model. The predicted values are showing significant deviation from the actual values in the testing data especially around day 75 - day 130. The actual values were considerably higher than the predicted values here. However for the significant deviations suggest that the model could be improved by training it on a more representative data. Both the graph shows the volatility of the actual prices.





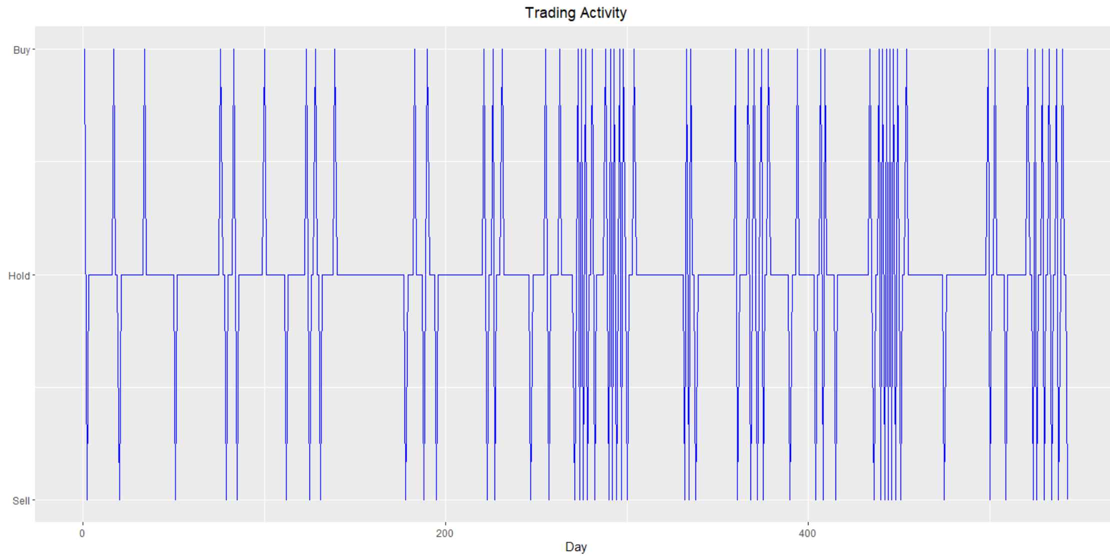
Trading Strategy

Here we are building a trading rule which allows to make decisions on the 3 possible actions - Buy, Sell or Hold. The rules are applied on training data. The variance will be set to 1% indicating the the prediction price will deviate only within 1% from the actual price during buying and selling. This is to ensure the predictions are not deviating much from the real market value. If the prediction is below the 10 day EMA, it is indicating a downward trend and is a buy signal. While buying, capital should not be 0, which means there should be money for the buying. If buying decision is made on the last day of training, there will not be any opportunity to further trading. The rule also makes sure that a buy can happen only after a sell. While selling, we should be having the stock in the portfolio to sell with capital 0. If its the last day of trading, then selling decision must be made without checking for other conditions. All the available stocks are sold at the closing price of that day.

A notional amount of $10,000 is set as investment. The earnings is calculated if the trading strategy was invoked on the training data.

Buy/Sell decisions plot

When the trading strategy is applied, a profit of $ 194091 profit is made after 114 trades: 57 buys and 57 sells. The buy/sell decisions are plotted as Trading activity.



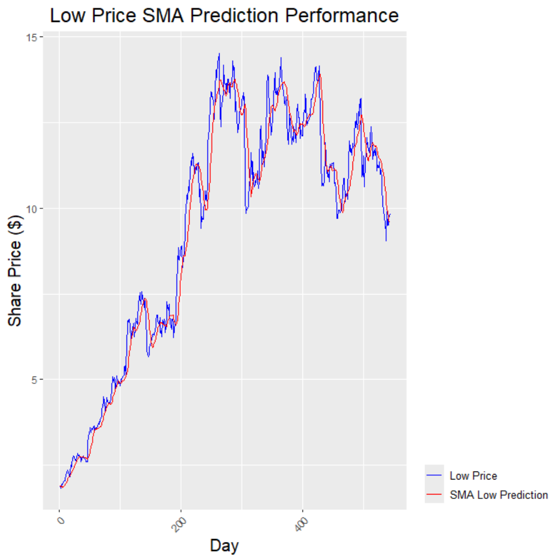
Comparison Performance - Mean value

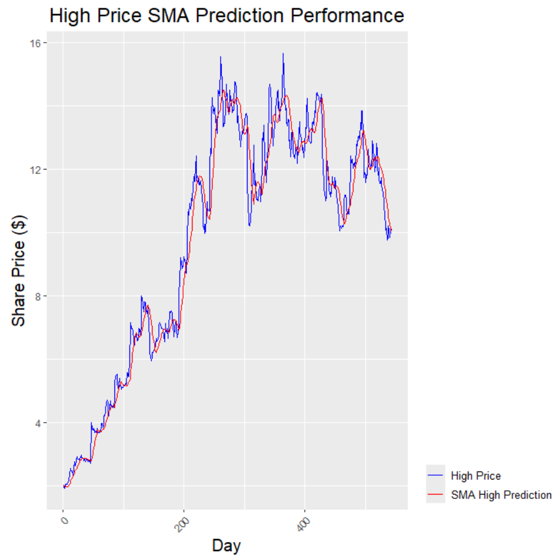
In order to further analyse the performance of our trading strategy, this is compared with another approach. Here we are comparing with the performance of the mean value. This is also invoked in the training data as done earlier. The trading strategy is compared against the performance of the mean value. The mean values of low and high prices for the past 7 days is used here instead of the predicted values done in neural network.

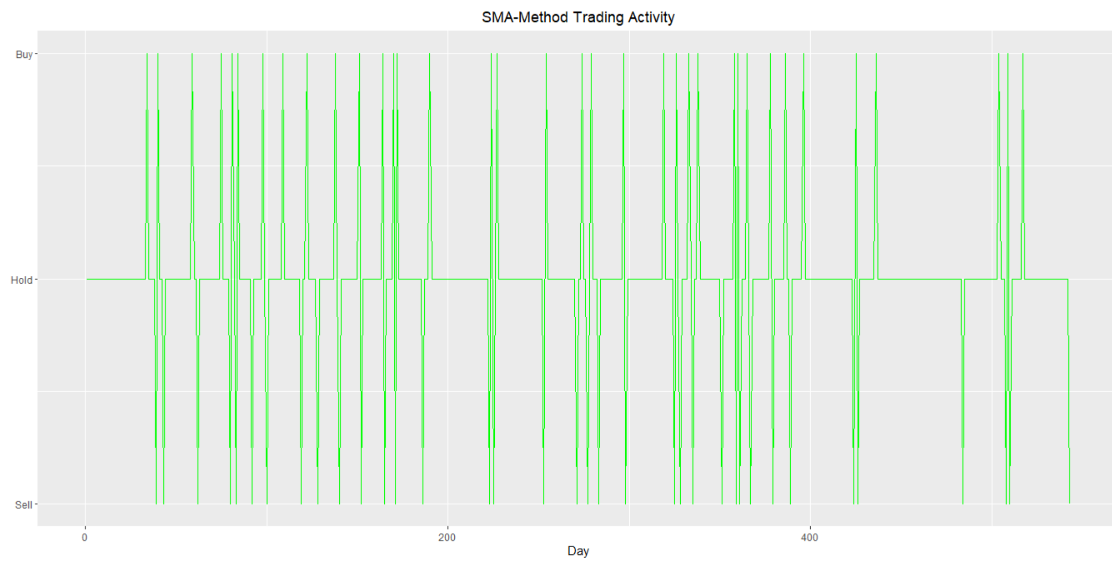
The same trading strategy is applied here and a profit of $61931 is made after 72 trades with 36 buys and 36 sells. Even though this is good profit and a positive value, it is not as high as the profit earned using neural network.

SMA Comparison - Plot

The actual line for both the graph high price and low price is consistently above the SMA line, which captures the trend of the stock prices and indicates a bullish signal .







Conclusion

This script evaluates the performance by comparing predicted prices with actual prices and making decisions. The neural network model has some degree of accuracy. It is performing well for training data with low RMSE values whereas it is opposite for the testing data. There are some significant deviations in the testing data, which is expected since no model can predict stock prices with complete accuracy. The model's performance can be improved by incorporating additional data. Whereas for the mean value the profit is not as good as the profit earned using neural network model which can be due to differences in market conditions during the periods. The trading strategy is simple which aims to buy low and sell high and the printed output provides insights. The model provides a framework for testing trading strategies and evaluated against market conditions.

References

https://www.linkedin.com/pulse/exciting-news-algorithmic-trading-revolutionizing-0a3of/

https://www.optionstocksmachines.com/post/2021-02-22-neural-nets/nothing-but-net/

https://www.sciencedirect.com/science/article/pii/S1877050917318252

https://www.sciencedirect.com/science/article/pii/S0304405X9800052X

Journal of Financial Economics 51 (1999) by Franklin Allen,Risto Karjalainen

https://www.logikfx.com/post/exponential-moving-average-101-how-to-use-ema-in-technical-analysis

https://www.ig.com/uk/trading-strategies/10-trading-indicators-every-trader-should-know-190604

Article:Trading with Exponential Moving Averages (EMA): A Comprehensive Guide, CoinW Exchange