Stock Index Forecasting Project Report

Sandra Nguemto

12/2/2018

Introduction

In this project, our aim was to learn more about machine learning and how it is implemented in R, as well as testing a forecasting model for Stock Indices. To achieve that, we implemented the model from the paper on S&P 500 data from 2017. The results of that implementation are presented here. To help with the implementation, several functions were created and regrouped in the `FMpackage` specifically created for this purpose.

Implementation of the forecasting model

```r
#Loading the data
data <- read_csv(here("Data", "sp500_working.csv"))
```

```
## Parsed with column specification:
## cols(
##   Date = col_character(),
##   AdjClose = col_double(),
##   Fluctuation = col_double()
## )
```

```r
#computing the whole mean of the Fluctuation Time Series (step 1)
len <- sum(abs(data$Fluctuation[2:251]))/(nrow(data)-1)
print(paste0("The whole mean for this dataset is: ", len))
```

```
## [1] "The whole mean for this dataset is: 7.279448236"
```

```r
#Fuzzify the fluctuation time series (step 1), using the data_fuzzification function
data <- data_fuzzification(data)
head(data, n = 3)
```

```
## # A tibble: 3 x 5
##   Date   AdjClose Fluctuation fuzzification St_bar
##   <chr>     <dbl>       <dbl>         <dbl>  <dbl>
## 1 1/3/17    2258.       NA                0   0
## 2 1/4/17    2271.       12.9              3   3.77
## 3 1/5/17    2269        -1.75             2   1.76
```

```r
#(step 2)Etablish 9th order FFLRs, using the function data_fflr function
data_lr <- data_fflr(N = 9, M = 15, data = data)
tail(data_lr, n = 3)
```

```
##     V1 V2 V3 V4 V5 V6 V7 V8 V9       V10
## 249  2  2  3  2  1  3  3  1  2 0.1527498
## 250  2  2  2  3  2  1  3  3  1 0.1783910
## 251  3  2  2  2  3  2  1  3  3 0.0000000
```

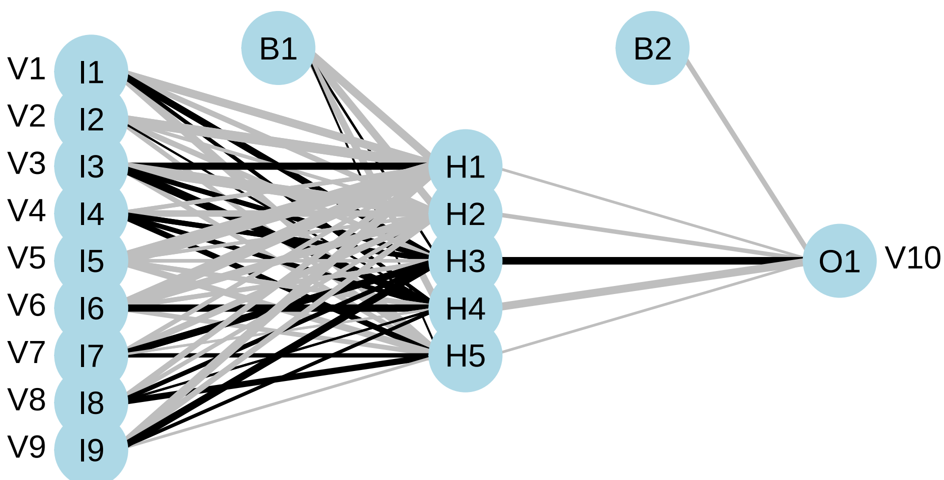
```r
#Determine the parameters for the forecasting model based on a
#Back Propagation Neural Network Machine Learning algorithm.(step 3)
```

```r
#Splitting the data into training and testing datasets.
#The training data set is from January to October and the testing dataset is from November to December.
index <- 210

training_2 <- data_lr[11:index,]
training_2 <- integer_conversion(training_2)

testing_2 <- data_lr[index+1:251,]
testing_2 <- integer_conversion(testing_2)

#implementing the BP neural network algorithm on the training data
bpnn <- neuralnet(V10 ~ V1+V2+V3+V4+V5+V6+V7+V8+V9, data = training_2, hidden=5,learningrate = 0.00008,
                  act.fct = "tanh", linear.output=T)
#head(bpnn$result.matrix)
#plot(bpnn)
plotnet(bpnn)
```




```r
#Using the FFLR obtained from the training data  to forecast the test dataset

temp_test <- subset(testing_2, select = c("V1","V2", "V3", "V4", "V5","V6","V7","V8","V9"))
bpnn.results <- compute(bpnn, temp_test)

#Predicted AdjClose price
ind <- seq(1:nrow(testing_2))
results_4 <-data.frame(date =data$Date[211:251],indexing = ind, actual_AdjClose = data$AdjClose[211:251]
                  predicted_AdjClose = (((bpnn.results$net.result*15) - 2))*len + data$AdjClose[210:250]
head(results_4, n = 3)
```

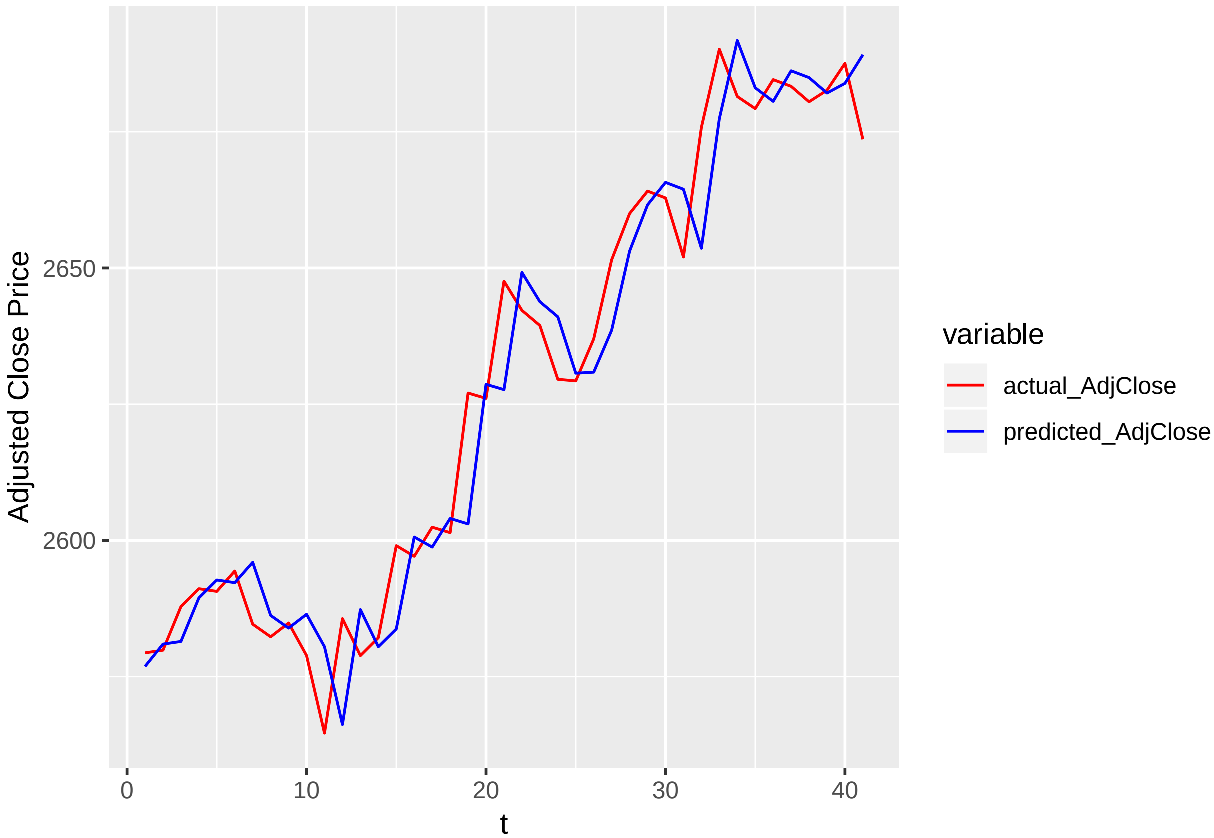
```
##        date indexing actual_AdjClose predicted_AdjClose
## 211 11/1/17        1     2579.360107        2576.860579
## 212 11/2/17        2     2579.850098        2580.964406
## 213 11/3/17        3     2587.840088        2581.449800
```

```r
#Evaluation of performance

#Plotting the results

results_sub = results_4[,c("indexing", "actual_AdjClose","predicted_AdjClose")]
```

```r
results_plot = melt(results_sub, id=c("indexing"))
ggplot(results_plot) + geom_line(aes(x= indexing, y=value, colour=variable)) +
  scale_colour_manual(values=c("red","blue")) + xlab("t") + ylab(label="Adjusted Close Price")
```




```r
# finding the MSE (mean squared error),
#RMSE (root of the mean squared error), MAE (mean absolute error), MPE (mean percentage error)
error_analysis(results_4)
```

```
## [1] "The mean squared error for this dataset is: 95.3433971792584"
## [1] "The root of the mean squared error for this dataset is: 9.76439435803667"
## [1] "The mean absolute error for this dataset is: 7.4317275487655"
## [1] "The mean percentage error for this dataset is: 0.00282499116019416"
```

Conclusion

The implementation of the model, and the plot of the result show that our results are in line with the results of the creators of the model. It performs well overall. Through this project, we were able to learn a lot more about the Back Propagation Neural Network Algorithm in particular, as well as the use of packages such as neuralnet and NeuralNetTools.