TimeSeriesForecast

Prashant Patil

February 12, 2019

Time Series Forecasting using R

Introduction

Burton Malkiel a Princeton University professor claimed in his book, a blindfolded monkey throwing darts at a newspaper's financial pages could select a portfolio that would do just as well as one carefully selected by experts. While the direction of stocks or market is dependent on multiple factors, in this analysis we try to apply different prediction methods/machine learning algorithms to see if we can make a better and informed judgement. We compare different Time serries prediction models based on the RMSE.

There are serveral other factors which can be explored for improving prediction e.g. Corelation between opening and closing price etc. This study just focuses on Time Series.

Overview:

The goal is perform a time series analysis of avaiable data and try to forecast. For this we use https://www.kaggle.com/szrlee/stock-time-series-20050101-to-20171231. It has historical data of DJIA 30 companies from Jan 2006 to Dec 2017. This dataset contains:

```
##         Date  Open  High   Low Close    Volume Name
## 1 2006-01-03 10.34 10.68 10.32 10.68 201853036 AAPL
## 2 2006-01-04 10.73 10.85 10.64 10.71 155225609 AAPL
## 3 2006-01-05 10.69 10.70 10.54 10.63 112396081 AAPL
## 4 2006-01-06 10.75 10.96 10.65 10.90 176139334 AAPL
## 5 2006-01-09 10.96 11.03 10.82 10.86 168861224 AAPL
## 6 2006-01-10 10.89 11.70 10.83 11.55 570088246 AAPL
```

Date - Date for which the data is collected Open - Price at which the stock trading opened High - Highest stock price for that day Low - Lowed stock price for that day Close - Price at which the stock trading closed Volume - Number of shares traded Name - Stock code AAPL - Apple Inc.

We will use the closing price of stock for this prediction.

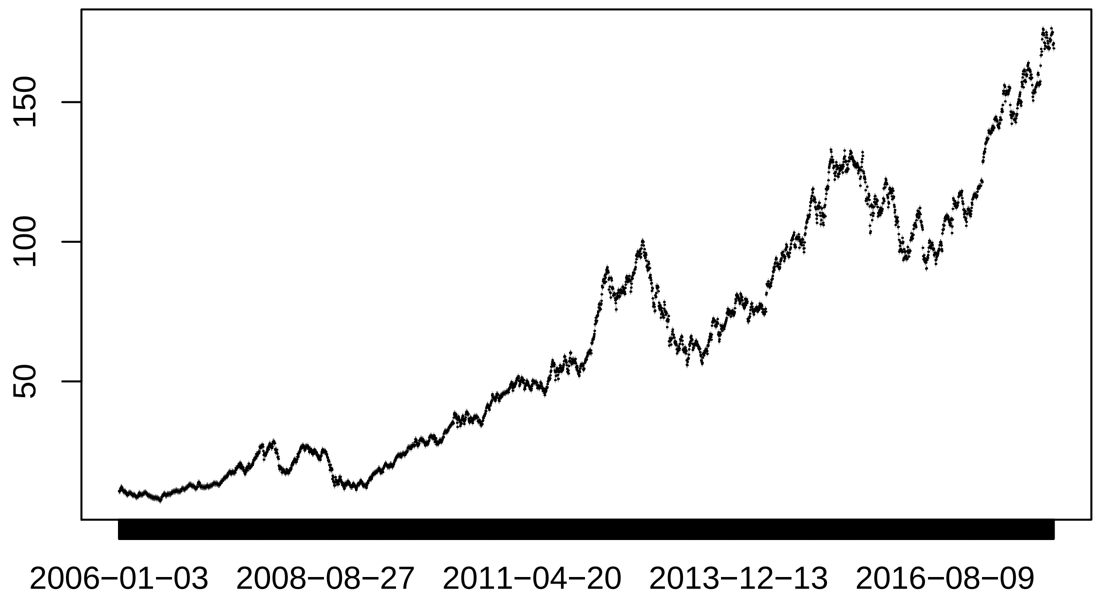
Analysis:

For dividing data in training and test parts, we can not use random splitting because we will lose the time sequence. So we divide the data by years. Data from 2006 to 2016 will be used for training and we will use 2017 data for testing our models. This same analysis can be applied to differnt time series forecasts like Sales, Call Volume etc.

There are 3019 rows available and 2016 data ends on index 2768.

For simplicity we will keep last 300 observations as our test data and rest of the data will be used for training the models.

We are using Closing price prediction, the following plot shows the current trend of closing price.



Methods:

There are serveral approaches to time series analysis. We use forecast package in R and compare the result of different methods based on RMSE of final prediction we will select the best approach.

1. Mean Forecast:

By this approach we are going to predict values based on simple mean of available train data.

```
fit_meanf <- meanf(trainData[,5], h = 300)
pred_meanf <- as.numeric(fit_meanf$mean)
RMSE(testData$Close,pred_meanf)
```

```
## [1] 90.60174
```

2. Naive Prediction:

This approach uses the forecast value to be same as previous available value.

```
fit_naive <- naive(trainData[,5], h = 300)
pred_naive <- as.numeric(fit_naive$mean)
RMSE(testData$Close,pred_naive)
```

```
## [1] 33.48592
```

3. Random Walk Forest - Drift Method:

This approach predicts the future values based on change in historical values over the time.

```
fit_rwf <- rwf(trainData[,5], h = 300, drift = TRUE)
pred_rwf <- as.numeric(fit_rwf$mean)
RMSE(testData$Close,pred_rwf)
```

```
## [1] 26.84666
```

4. Time series Linear Regression:

This approach predicts the value based on the trend and seasonality over the time in training data.

```
fit_lm <- tslm(as.ts(trainData[, 5]) ~ trend)
pred_lm <- as.numeric(forecast(fit_lm, h = 300)$mean)
RMSE(testData$Close,pred_lm)
```

```
## [1] 26.13019
```

5. Exponential Smoothing State Space model:

This model uses a state space framework for automatic forecasting using exponential smoothing methods.

```
fit_ets <- trainData[,5] %>% ets(model="MMZ")
pred_ets <- as.numeric(forecast(fit_ets, h = 300)$mean)
RMSE(testData$Close,pred_ets)
```

```
## [1] 9.290167
```

6. Neutral Networks:

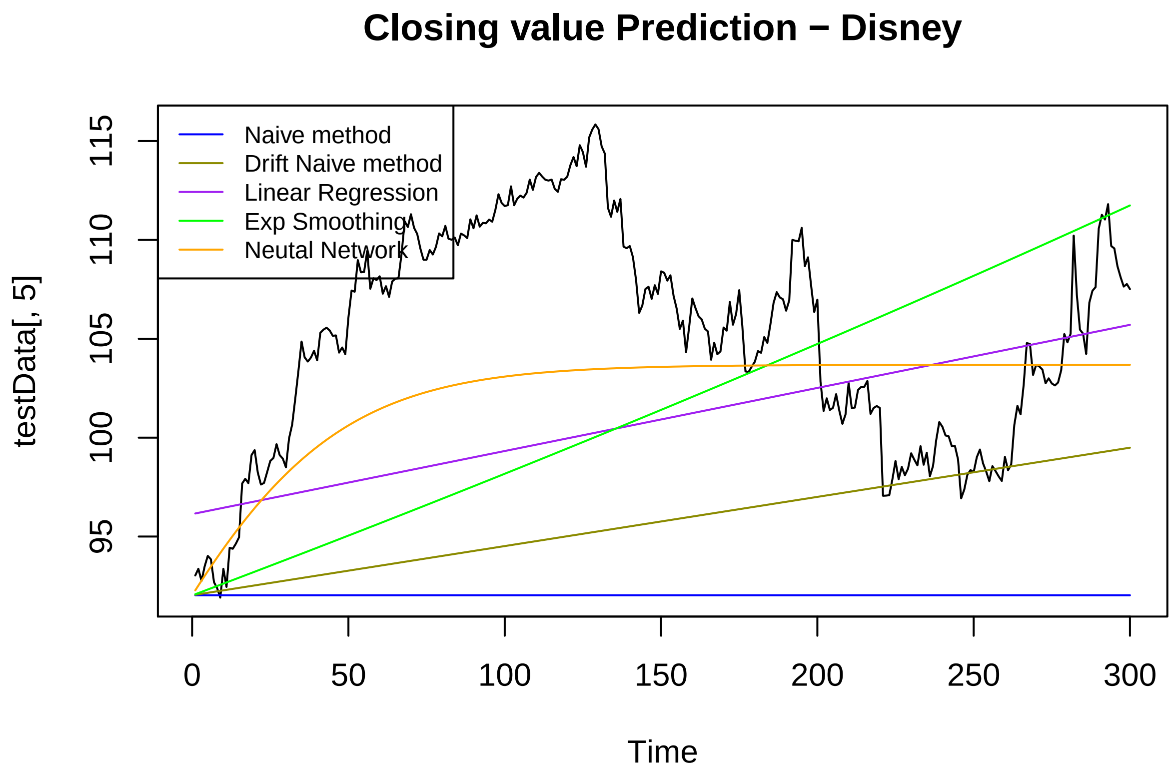
This approach uses forecast using neutral network models.

```
fit_NeuNet <- nnetar(as.ts(trainData[,5]))
pred_nn <- as.numeric(forecast(fit_NeuNet, h = 300)$mean)
RMSE(testData$Close,pred_nn)
```

```
## [1] 36.56519
```

Conclusion:

The following graph shows predictions for Disney Stock and also actual values. Mean forecast is excluded from plot to focus on predictions which are closer to actuals.



In this project we compared Time Series predictions using different models. Same models are applied to 3 different stocks. Following table shows the RMSEs obtained for different stocks.

```
## # A tibble: 6 x 4
##   Method                         RMSE_AAPL RMSE_DIS RMSE_WMT
##   <chr>                              <dbl>    <dbl>    <dbl>
## 1 Mean Forecast                      90.6     52.7     19.1
## 2 Naive Prediction                   33.5     14.3     12.5
## 3 Random Walk Forest                 26.8     11.2     11.1
## 4 Time series Linear Regression      26.1      7.60     8.19
## 5 Exponential Smoothing               9.29     8.95    10.2
## 6 Neutral Network                    36.6      5.62    10.3
```

Out of the methods compared, Exponential smoothing method appears to be most effective. There are several more methods like LSTM, RNN, GRU etc. which can be explored further for time series prediction.