Forecasting Stock Returns of Apple Inc. and Microsoft Corporation Using ARIMA Model

Alice Roberts, Zhi Yuh Ou Yang

December 3, 2018

Abstract

In this paper, we implement ARIMA Time Series forecasting methods to predict the stock returns for Apple Inc. and Microsoft Corporation. We discuss the accuracy of our predictions and compare the two predicted stock returns results. The accuracy of our computations for Apple Inc. and Microsoft Corporation are 51.19% and 54.37% respectively.

1 Introduction

1.1 Background

Apple Inc. founded by Steve Jobs, Steve Wozniak, and Ronald Wayne in April 1976, is an American multinational technology company. Apple Inc. is one of the most successful technology companies out there, with recent reports of their quarterly revenue averaging around $53.3 billion dollars.[INC18]

Microsoft Corporation founded by Bill Gates and Paul Allen on April 4$^{\text{th}}$, 1975, is also an American multinational technology company. Microsoft Corporation reported their quarterly revenue to be around $30 billion dollars. [Mic]

On November 30$^{\text{th}}$ of 2018, Microsoft Corporation's stock market value closed with higher value than Apple Inc.'s stock market value for the first time in eight years! [NS18] Apple Inc. has officially been dethroned as the world's most valuable company, with Microsoft Corporation taking its place.

The recent debate between the stock market values of Microsoft and Apple lead to our curiosity in predicting the stock returns of the two leading technological companies of 2018.

1.2 Data set

We are using data sets from Yahoo! Finance [Yah18] for both the Apple Inc and Microsoft Corporation. For Apple, we gathered the stock returns data from December 1$^{\text{st}}$, 2014 to December 1$^{\text{st}}$ of 2018. For Microsoft we gathered stock returns data from December 1$^{\text{st}}$, 2016 to December 1$^{\text{st}}$ 2018.

We split our data set into 2 parts: training and testing data sets. For the training data set we stopped at December 1st, 2017 for both companies. We used the last year (2017 - 2018) for both companies as our testing data, to predict the stock returns of each organizations.

1.3 Goal

Our goal for this project is to:

1. Test the extracted time series data for stationarity.
2. Identify the order of p, d and q for each of the ARIMA models.
3. Check the accuracy of each of the ARIMA models.

2 Methodology

For our project we wrote an R script [RY18] for all computations to manipulate our data. Our methods are based heavily off of an article from Rbloggers. [Par17]

2.1 Testing for Stationarity

Based on the Box-Jenkins approach, the time series has to be stationary. A stationary time series is a time series without a trend; one that has a constant mean and variance over time. The basic idea of stationarity is that the probability laws that govern the behavior of the process do not change over time. [CC11]

We test for stationarity using the Augmented Dickey-Fuller unit root test. The p-value obtaining from the ADF test has to be less than 0.05 for a time series to be stationary. If the p-value is greater than 0.05, we conclude that the time series has a unit root which means that it is a non-stationary process. In our analysis, we will use the adf.test() function to justify our results.

2.1.1 Differencing

A differecing method is applied if the process is not stationary. Differencing a time series means finding the differences between consecutive values of a time series data. The differenced values form a new time series data set which can be tested to uncover new correlations or other interesting statistical properties. i.e.

Define AR(1) model s.t.

$$Y_t = Y_t + e_t.$$

Then first difference of Y is noted as:

$$\nabla Y_t = Y_t - Y_{t-1}.$$

[CC11]

2.2 Identification of p order of AR Model

For a Auto Regressive model, we use the Partial Autocorrelation Function (PACF) to identify the order p of the AR model. We do this by checking if there is significant spike that passes the 95% confidence interval from the PACF plot. If there is, we have an AR model of at least order 1; AR(1). In addition, if we have significant spikes at lag 1, 2, and 3 on the PACF, then we have an AR model of the order 3 such as AR(3).

2.3 Identification of q order of MA Model

For a Moving Average model, we use the Auto Correlation Function (ACF) to identify the order q of the MA model. We find the order of q in the same fashion as above. We check for a significant spike that passes the confidence interval from the ACF plot. If there is a spike, then we have at least an MA model of order 1, MA(1). Furthermore, if we have significant spikes at lag 1, 2, and 3 on the ACF, then we have an MA model of the order 3, i.e. MA(3).

2.4 Estimation and Forecasting

Once we have determined the parameters p, d and q, we estimate the accuracy of the ARIMA model on a training data set and then use the fitted model to forecast the values of the test data set using a forecasting function. Moreover, our intention is to predict one year stock returns starting from 1st December 2017 to 30th November 2018 for both of the companies that we chose. We decided to use the past three years of stock returns data from Apple Inc. and the past one year stock returns data from Microsoft Corporation as our training data sets. In the end, we cross checked all of our predicted values to ensure that they are in line with the actual values. We do this by plotting the red line as predicted values and black line as actual values and by calculating the accuracy percentage. (see appendix for R Code).

3 Results

3.1 Apple Inc.

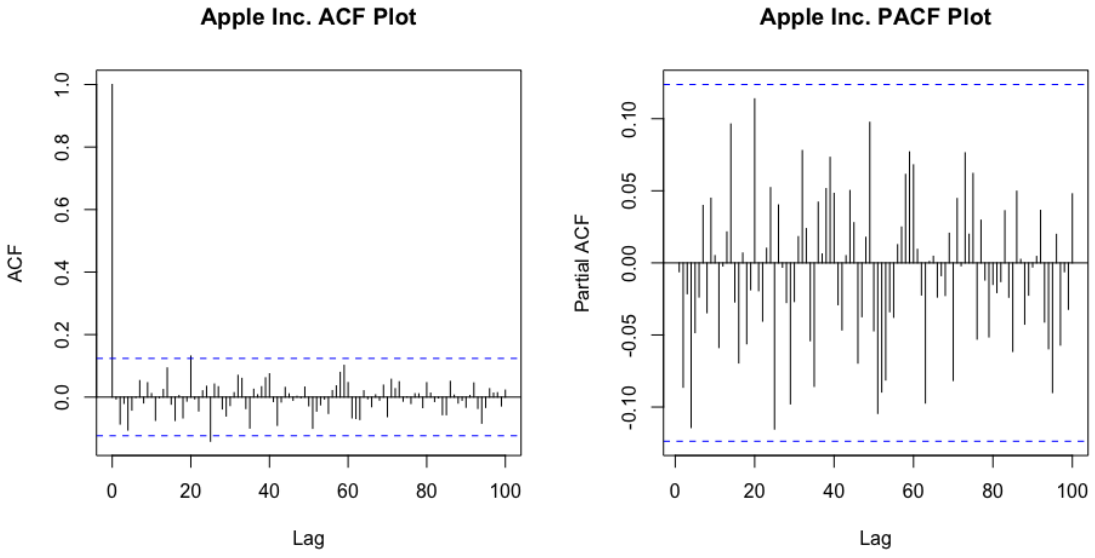
After implementing the Augmented Dickey-Fuller unit root test, we found that Apple Inc.'s stock returns data was stationary. It had a p-value less than 0.05, see Figure 1. Thus we did not have to consider the differencing method, since the data is stationary.

```
p-value smaller than printed p-value
        Augmented Dickey-Fuller Test

data:  stock
Dickey-Fuller = -9.2, Lag order = 10, p-value = 0.01
alternative hypothesis: stationary
```

Figure 1: Apple Inc. Stationarity Test

To identify the p, d and q order we chose to generate plots using the pacf() and acf() function in R, we did up to 100 lags for visualization purposes. From Figure 2, we see that there is an initial spike in the ACF plot but beyond lag one there is no immediate proceeding spike that passes the 95% confidence interval. Thus, we can conclude that the we have an MA model of q = 1. From the same figure, we see that there is no significant spike that passes through the confidence interval of the PACF plot. Thus, we conclude that there is p = 0; we do not have an AR model.



(a) ACF Plot

(b) PACF Plot

Figure 2: ACF and PACF plot for Apple Inc.

Once we found the order of the ARIMA model we use the function: forecast() in R to predict the stock returns for the last year (2017 - 2018). Figure 3 shows our predicted returns vs the actual returns.

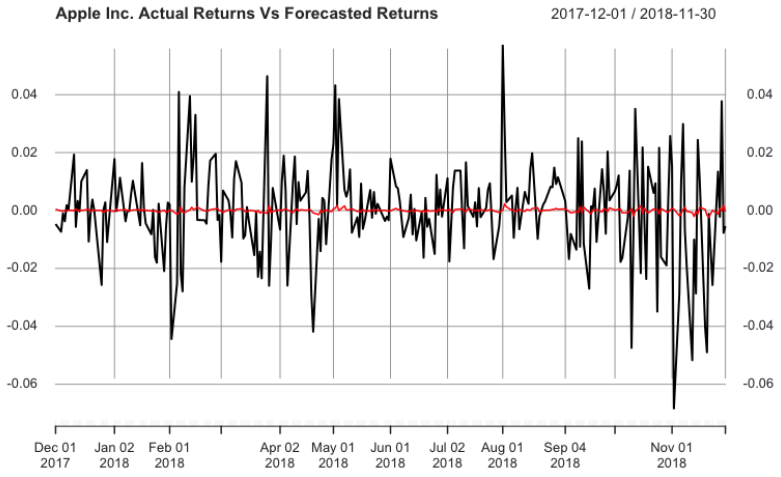


Figure 3: Predicted and Actual Stock Returns from 1st December 2017 to 30th November 2018 of Apple Inc.

```
Call:
arima(x = stock_train, order = c(0, 0, 1), include.mean = FALSE)

Coefficients:
         ma1
      0.0443
s.e.  0.0328

sigma^2 estimated as 0.0002269:  log likelihood = 2795.93,  aic = -5587.87
```

Figure 4: MA(1) Model Summary of Apple Inc.

Based on Figure 4, we can formulate the predicted MA(1) model for Apple Inc.'s stock returns to be: $\hat{Y}_t = e_t - 0.0443e_{t-1}$.

3.2 Microsoft Corporation

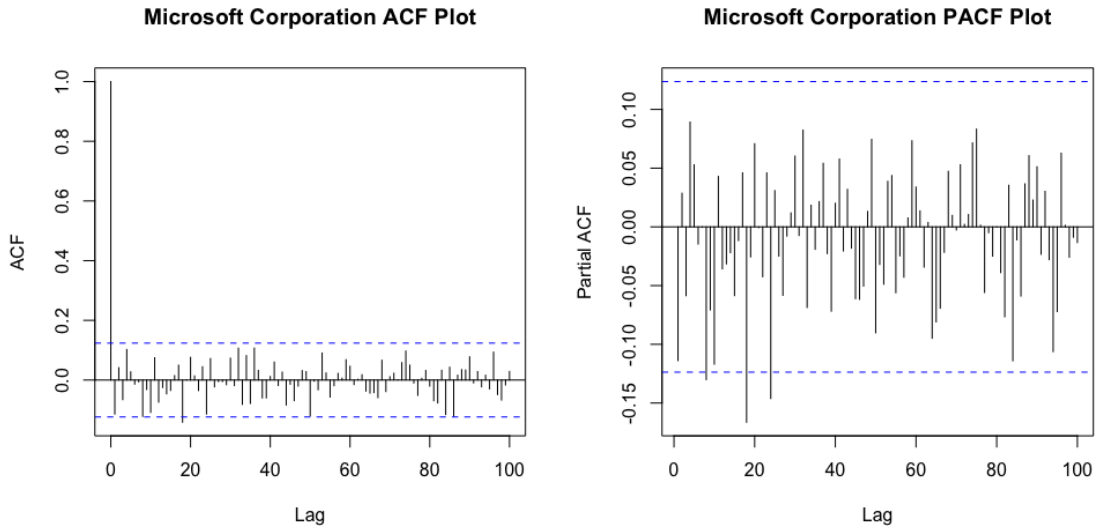
After implementing the Augmented Dickey-Fuller unit root test, we found that Microsoft Corporation's stock returns data was stationary. It had a p-value less than 0.05, see Figure 5. Thus we did not have to consider the differencing method, since the data is stationary.

```
p-value smaller than printed p-value
        Augmented Dickey-Fuller Test

data:  stock
Dickey-Fuller = -9.3762, Lag order = 7, p-value = 0.01
alternative hypothesis: stationary
```

Figure 5: Microsoft Corporation Stationarity Test

Microsoft's stock returns behave in a similar fashion to that of Apple's. To identify the p, d and q order we chose to generate plots using the pacf() and acf() function in R, we did up to 100 lags for visualization purposes. From Figure 6, we see that there is an initial spike in the ACF plot but beyond lag one there is no immediate proceeding spike that passes the 95% confidence interval. Thus, we can conclude that the we have an MA model of q = 1. From the same figure, we see that there is no significant spike that passes through the confidence interval of the PACF plot. Thus, we conclude that there is p = 0; we do not have an AR model (same case for Apple Inc.).



(a) ACF Plot

(b) PACF Plot

Figure 6: ACF and PACF plot for Microsoft Corporation

Once again, after we found the order of the ARIMA model we use the function: forecast() in R to predict the stock returns for the last year (2017 - 2018). Figure 7 shows our predicted returns vs the actual returns.

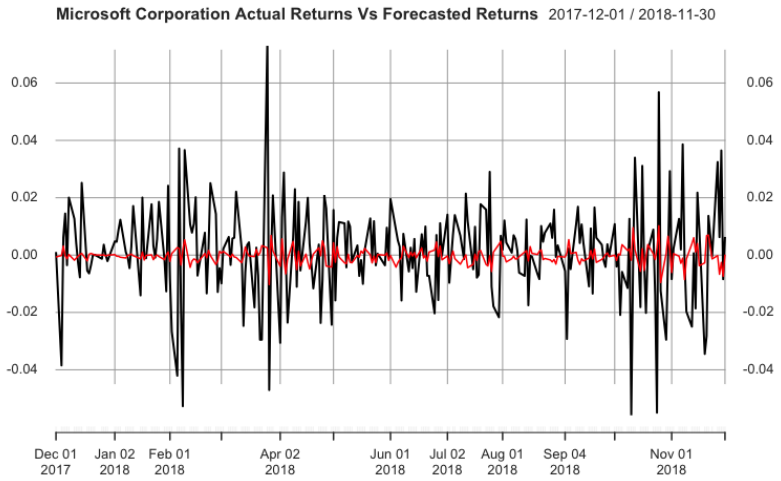


Figure 7: Predicted and Actual Stock Returns from 1st December 2017 to 30th November 2018 of Microsoft Corporation

```
Call:
arima(x = stock_train, order = c(0, 0, 1), include.mean = FALSE)

Coefficients:
          ma1
      -0.1909
s.e.   0.0456

sigma^2 estimated as 0.0001728:  log likelihood = 1462.16,  aic = -2920.32
```

Figure 8: MA(1) Model Summary of Microsoft Corporation

Based on Figure 8, we can formulate the predicted MA(1) model for Apple Inc.'s stock returns to be: $\hat{Y}_t = e_t + 0.1909e_{t-1}$.

4 Discussion

We see that our accuracy for the our Apple and Microsoft models were 51.19% and 54.37% respectively. During our implementation phase, we discovered that, in order to achieve optimum accuracy for the data sets we had to choose different time intervals for the training data sets.
For Apple's training data, we chose to include three years of stock returns data (2014 - 2017). We chose this time period because we found that this gives the best accuracy of 0.5119 when comparing the predicted values and actual values from 1st December 2017 to 1st December 2018.
For Microsoft's training data, we chose to use only one year of stock return data (2016 - 2017). We found that for Microsoft, this gives the best accuracy of 0.5437 when comparing the predicted values and actual values from 1st December 2017 to 1st December 2018.
We see that for this model, Microsoft Corporation's stock returns have a higher accuracy of prediction in comparison to Apple Inc.'s stock returns.
We hoped for an accuracy higher than 60% but due to the time constriction we were unable to achieve higher results. If provided with more time we would explore different ways of increasing our accuracy, some of which include:

- Implementing auto.arima function rather than choosing p, d and q parameters manually.
- Running the models with different manually chosen p, d and q parameters.
- Using another technique such as neural networks for forecasting.

We would also explore the normality of the two data sets and preform residual analysis on the data.

References

[CC11] Jonathan D. Cryer and Kung-sik Chan. *Time series analysis with applications in R*. Springer, 2011.

[INC18] Apple INC. Apple reports third quarter results, https://www.apple.com/ca/newsroom/2018/07/apple-reports-third-quarter-results/, 2018.

[Mic] Trended historical financials , https://www.microsoft.com/en-us/investor/earnings/trended/quarterly-income-statements.aspx.

[NS18] Jordan Novet and Sara Salinas. Apple loses spot as world's most valuable public company to microsoft, https://www.cnbc.com/2018/11/30/microsoft-apple-end-trading-market-cap.html, Nov 2018.

[Par17] Milind Paradkar. Forecasting stock returns using arima model, Mar 2017.

[RY18] Alice Roberts and Zhi Yuh Ou Yang. Time series analysis, https://github.com/aliceroberts10/time-series-anlaysis, 2018.

[Yah18] Yahoo. Yahoo finance - business finance, stock market, quotes, news, https://ca.finance.yahoo.com/, 2018.

5 Appendix

5.1 Estimation and Forecasting R Code

```
# Initialzing an xts object for Actual log returns
Actual_series <- xts(0,as.Date("2017-12-01","%Y-%m-%d"))

# Initialzing a dataframe for the forecasted return series
forecasted_series <- data.frame(Forecasted = numeric())
for (b in breakpoint:(nrow(stock)-1)) {
  stock_train <- stock[1:b, ]
  stock_test <- stock[(b+1):nrow(stock), ]
  # Summary of the ARIMA model using the determined (p,d,q) parameters
  fit <- arima(stock_train, order = c(0, 0, 1),include.mean = FALSE)
  summary(fit)
  # plotting a acf plot of the residuals
  acf(fit$residuals, main = "Residuals plot")
  # Forecasting the log returns
  ## Use forecast instead as the latest version does not include forecast.Arima
  arima.forecast <- forecast(fit, h = 1,level=99)
  summary(arima.forecast)
  # plotting the forecast
  par(mfrow = c(1,1))
```

```
  plot(arima.forecast, main = "ARIMA Forecast")
  # Creating a series of forecasted returns for the forecasted period
  forecasted_series <- rbind(forecasted_series, arima.forecast$mean[1])
  colnames(forecasted_series) <- c("Forecasted")
  # Creating a series of actual returns for the forecasted period
  Actual_return <- stock[(b+1),]
  Actual_series <- c(Actual_series,xts(Actual_return))
  rm(Actual_return)
  print(stock_prices[(b+1),])
  print(stock_prices[(b+2),])
}


arima(stock_train, order = c(0, 0, 1),include.mean = FALSE)



# Adjust the length of the Actual return series
Actual_series <- Actual_series[-1]
# Create a time series object of the forecasted series
forecasted_series <- xts(forecasted_series,index(Actual_series))
# Create a plot of the two return series - Actual versus Forecasted
plot(Actual_series, type = 'l', main = 'Actual Returns Vs Forecasted Returns')
lines(forecasted_series, lwd = 1.5, col = 'red')
legend('bottomright', c("Actual","Forecasted"), lty = c(1,1), lwd = c(1.5,1.5), col
    = c('black','red'))
# Create a table for the accuracy of the forecast
comparsion <- merge(Actual_series,forecasted_series)
comparsion$Accuracy <- sign(comparsion$Actual_series) == sign(comparsion$Forecasted
    )
print(comparsion)
# Compute the accuracy percentage metric
Accuracy_percentage <- sum(comparsion$Accuracy == 1)*100/length(comparsion$Accuracy
    )
print(Accuracy_percentage)
```