STOCK MARKET PREDICTION USING TIME SERIES ANALYSIS

Pooja Radhakrishnan, Gokul Krishnaswamy, Gowtham Krishnaswamy

ABSTRACT

Stock markets are uniquely complicated. "Stock Market Prediction is very difficult, especially about the future". – "Neils Bohr". A successful prediction of stock's future price could yield significant profits. This paper presents extensive process of building stock price predictive model using the **ARIMA** model. The daily stock data obtained from New York Stock Exchange (NYSE) is used for stock price predictive model developed. Results obtained revealed that the **ARIMA** model has a strong potential for short-term prediction. We have used the closing points of the stock market index of a month as a factor to predict the future stock prices. Time series analysis using the **ARIMA** model is done to achieve this task.

RESEARCH QUESTION

To predict the monthly closing value of stock in the months to come in the near future, there are a few techniques which can predict the stock values accurately to certain extent in terms of boundaries, can we implement the same using the **ARIMA** model of Time Series Analysis?

DATA SET

The 'Stock Market' dataset was extracted from the service provider '*Yahoo.com*' using the "**^GSPC**" function of '**quantmod**' package in R for the time span 03-01-1999 to 03-01-2019 (20 years). It has daily data with the following attributes: Date, Opening value, Highest value, Lowest value, Closing Value, Volume and Adjusted value. A sample chunk of the data set is provided in the figure below:

date	open	high	low	close	volume	adjusted
1999-03-01	1238.33	1238.70	1221.88	1236.16	699500000	1236.16
1999-03-02	1236.16	1248.31	1221.87	1225.50	753600000	1225.50
1999-03-03	1225.50	1231.63	1216.03	1227.70	751700000	1227.70
1999-03-04	1227.70	1247.74	1227.70	1246.64	770900000	1246.64
1999-03-05	1246.64	1275.73	1246.64	1275.47	834900000	1275.47
1999-03-08	1275.47	1282.74	1271.58	1282.73	714600000	1282.73
1999-03-09	1282.73	1293.74	1275.11	1279.84	803700000	1279.84

Figure 1 a): Raw Stock Data Set with the attributes

The Raw data contains daily stock values for a span of 20 years. It is found that there are no missing values despite the data set being a bit large and there is a presence of a few outliers due to various factors. Since the dataset is huge, the monthly returns of the raw data is taken and plotted to explore the data and perform analysis on the same, to feed the model with this data as the training data and use the testing data to forecast for the desired period.

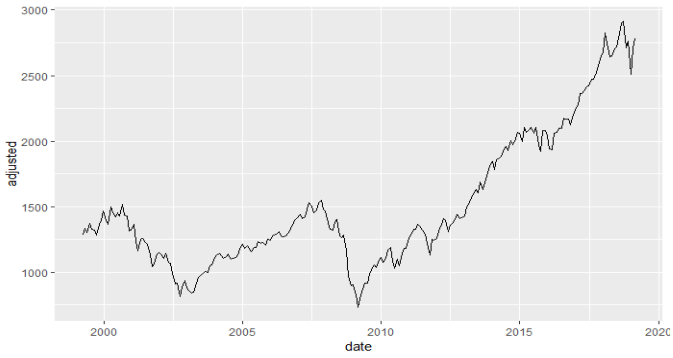

Figure 1b): Monthly Returns Plot with a few outliers for 20 years

The stock data is converted into a Time Series Data using the *ts()* function with the month and the adjusted, i.e. the closing value of the stock in the respective months. The data set is then split into Training and Testing Data sets. The Training data set has 18 years of data (03-01-1999 – 02-01-2017) and the Testing data set has 2 years of data (03-01-2017 – 03-01-2019).

```
stock_data_train <- ts(stock_ts, start=c(1999, 3), end=c(2017, 2), freq=12)
test_data <- ts(stock_ts, start=c(2017, 3), end=c(2019, 3), freq=12)
```

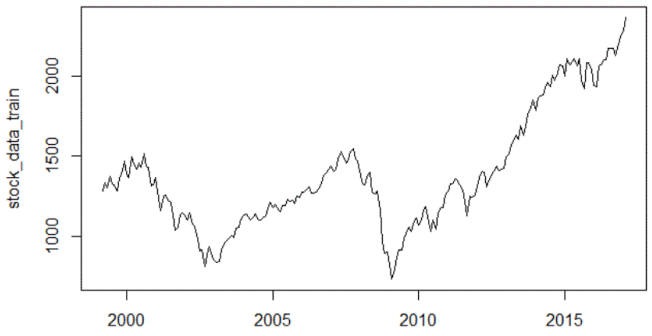

Figure 1 c): Training Data

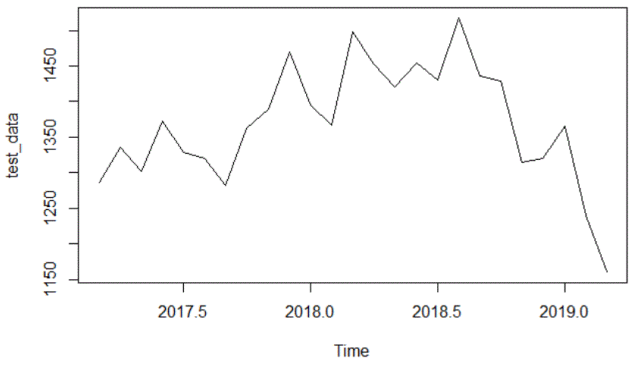

Figure 1 d): Testing Data

ARIMA MODEL

ARIMA stands for **Auto-Regressive Integrated Moving Average** and is specified by these three order parameters: ***(p, d, q)***. The process of fitting an **ARIMA** model is sometimes referred to as the Box-Jenkins method.

An *Auto Regressive* (AR(p)) component is referring to the use of past values in the regression equation for the series Y. The auto-regressive parameter p specifies the number of lags used in the model. For example, AR(2) or, equivalently, **ARIMA**(2,0,0), is represented as

$$Y(t) = c + \varphi 1.Y_{(t-1)} + \varphi 2.\ Y_{(t-2)} + e(t)$$

where $\varphi 1$, $\varphi 2$ are parameters for the model.

The d represents the degree of differencing in the *Integrated (*I(d)) component. Differencing a series involves simply subtracting its current and previous values d times. Often, differencing is used to stabilize the series when the stationarity assumption is not met, which we will discuss below.

A *Moving Average* (MA(q)) component represents the error of the model as a combination of previous error terms et. The order q determines the number of terms to include in the model

$$Y(t) = c + a1\ e(t-1) + a2\ e(t-2) + \ldots + aq\ e(t-q) + e(t)$$

Differencing, autoregressive, and moving average components make up a non-seasonal **ARIMA** model which can be written as a linear equation:

$$Y(t) = c + \varphi 1\ Yd_{(t-1)} + \varphi p\ yd(t-p) + \ldots + a1\ e(t-1) + aq\ e(t-q) + e(t)$$

where yd is Y differenced d times and c is a constant.

Note that the model above assumes non-seasonal series, which means we might need to de-seasonalize the series before modeling.

ARIMA models can be also specified through a seasonal structure. In this case, the model is specified by two sets of order parameters: (p, d, q) as described above and (P, D, Q) parameters describing the seasonal component of m periods.

METHODOLOGY

The approach used to address the research question involved the following steps:

- Exploratory Data Analysis
- Decomposing the data to check for the presence of the following non stationary components: *Trend*, *Seasonality*, *Cyclicity*, and *Randomness*.
- Check for Stationarity using ***Box-Ljung test*** and ***Augmented Dicky Fuller's test.***
- If not, making the data stationary by differencing.
- Extract the values of *p* and *q* using the ACF and PACF plots.
- Fit an **ARIMA** model both manually, and using the kernel.
- Forecast for the required period desired.

Upon Exploratory Data Analysis performed on the dataset, it is important to check for the impurities and presence of the non-stationary components of time series data. A stationary series is one in which the properties of mean, variance and co-variance are time independent.

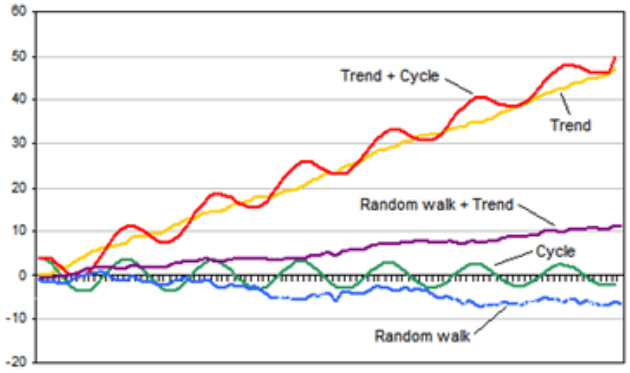

Figure 2: Non stationarity components - Theoretically

In order to make the data stationary, we have to remove these components and test the validity of stationarity using the Box-Ljung tests and Augmented Dickey Fuller's test. The ADF test has the Null hypothesis that the data is not stationary and the Alternative hypothesis stating that the data is stationary. The result of the ADF test yields the *p-value*, which will decide upon the acceptance or rejection of the Null hypothesis.

So, to make the non-stationary data stationary, we use a method called *differencing.* Simple *differencing* can remove the trend and coerce the data to stationarity. Differencing looks at the difference between the value of a time series at a certain point in time and its preceding value. That is, $X_t - X_{t-1}$ is computed. This is achieved using the *decompose()* or the *stl()* functions, which yield the plot of all the non-stationary components of the data to be visually interpreted.

The *AutoCorrelation Function* plot and *Partial AutoCorrelation Function* plots are analyzed upon this stationary data to determine if the data set falls under *Moving Average (MA), AutoRegression (AR) or* **ARIMA.** The ACF plot determines the '*q'* coefficient of the **ARIMA** model and the PACF plot determines the '*p'* coefficient of the **ARIMA** model.

The **ARIMA** model is built using these coefficients manually or with the kernel and is forecasted using the *forecast()* function for the desired period.

DATA SATISFACTION OF METHOD

The following are two conditions that the data must satisfy in order to develop an **ARIMA** model.

- Data must be time series in nature.
- Data must be stationary (*Mean and Variance of the data must be constant along time).*

'Stock Market' dataset was extracted from '**quantmod**' package in R for the time span 03-01-1999 to 03-01-2019 by following command.

stock_data <- tq_get(c("^GSPC"), get = "stock.prices", from = "1999-03-01", to = "2019-03-01")

It was explored and analyzed. It was found to contain no missing values.

The data had many attributes out of which the closing market index value was our dependent variable to predict it along time. The daily data which was extracted was converted into monthly data by taking the closing value of market index at the ending date of each month between the time periods specified while importing the data. The following command converts daily data to monthly data.

> stock_returns_monthly <- stock_data %>% group_by("^GSPC") %>% tq_transmute(select = adjusted, mutate_fun = to.period, period = "months")

It is then converted into a time series data by running the following command.

> stock_ts <- ts(stock_returns_monthly$adjusted, start = c(1999,3), freq = 12)

Upon Decomposition, the presence of the non-stationary components namely ***Trend***, ***Seasonality***, ***Cyclicity*** and ***Randomness*** were observed.

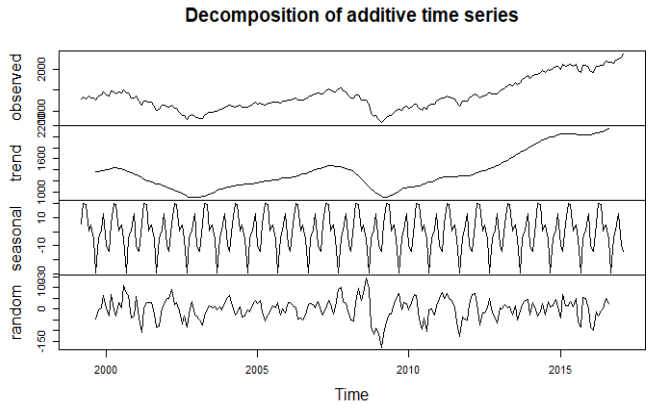

Figure 3: Data decomposition – trend, seasonality, randomness

This was confirmed by performing Box-Ljung tests and Augmented Dickey Fuller's test to the time series data.

```
        Box-Ljung test

data:  stock_ts
X-squared = 3183, df = 20, p-value < 2.2e-16


        Augmented Dickey-Fuller Test

data:  stock_ts
Dickey-Fuller = -1.1724, Lag order = 6, p-value = 0.9091
alternative hypothesis: stationary
```

p-value is greater than 0.05 => The Null hypothesis that the data is not stationary is accepted.

The ***ACF (AutoCorrelation Function)*** plot gives the differencing term *(d)* needed to make the data stationary. The data is then made stationary using differencing by running the following command.

> stock_diff <- diff(stock_data_train)

Hence, now, the dataset can be fitted into **ARIMA** model as it meets all the requirements.

After training the model with stationary data, we predict a result. In order to adjust to the future, we again add the initial components which made the data non stationary to the result.

APPLICATION OF METHOD

An **ARIMA** model is built using *Arima()* function in R. *Arima()* function takes 3 values as input namely *AutoRegressive* component*(p)*, *difference* term value*(d)* and *Moving Average* component*(q)* respectively.

The following is the representation of stationary data from Figure 1.b.

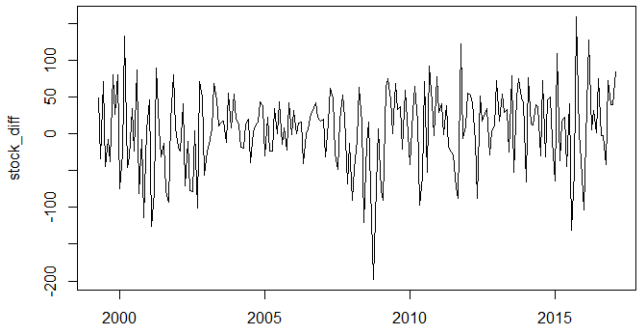

Figure 4: Stationary Data

Stationary data is represented using ***ACF*** *(AutoCorrelation Function)* plots and ***PACF*** *(Partial AutoCorrelation function)* plots to check for '*p*' and '*q*' values. The 'd' value was already found using ACF plot earlier.

The following are the ACF and PACF plots of the stationary data.

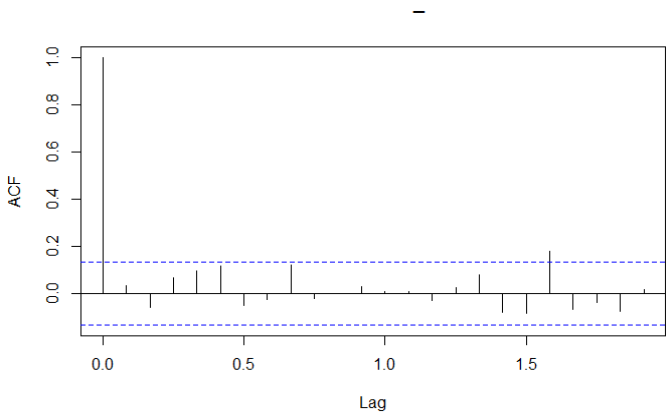

Figure 5: AutoCorrelation Function Plot

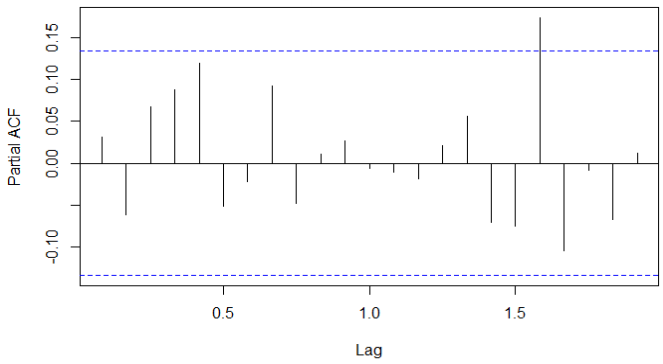

Figure 6: Partial AutoCorrelation Function Plot

In the ACF plot, the spike occurs at lag value '0'. So '0' is given as value of 'q' to the Arima() function. Similarly in PACF plot, the spike occurs at '1'.5' which is rounded to value '1'. So '1' is given as value of 'q' to the Arima() function. Previously we determined the 'd' value to be '1'(Differencing of order 1 need to be performed to make the original data stationary).

In R, auto.arima() function gives the value of the components (p,d,q). It also computes (P, D, Q) and m parameters (seasonal component of m periods of the data).

ARIMA model is built by either specifying the *p,d,q* parameters or by using *auto.arima()* function.

> manual_arima <- Arima(stock_data_train, order = c(1,1,0))

> auto_arima <- auto.arima(stock_data_train)

Using the model built, we can predict the future value of the stock.

FORECAST

MANUALLY IMPLEMENTED ARIMA MODEL:

The **ARIMA** model is now implemented both manually and by the kernel. Manually, the *p,d,*q values are extracted from the ACF and PACF plots and the time series data is monthly with a frequency of 12 months, and hence *period = 12.* Now the model is used to predict the values for the next 2 years using the *forecast ()* function. The following chunk of code shows the manual implementation of the **ARIMA** model:

```
#manually implementing ARIMA forecast

manual_arima <- Arima(stock_data_train, order = c(1,1,0), seasonal = list(order =
c(2,1,1), period = 12))
summary(manual_arima)
plot(forecast(manual_arima))
```

The following plot is the forecast using the *Arima()* function:

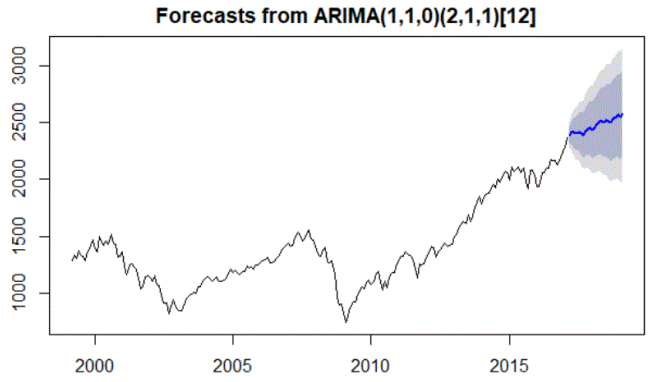

Figure 7: Manual ARIMA forecast

Now the same **ARIMA** model is made to run using the kernel without specifying the *SEASONAL DIFFERENCE PARAMETER (R),* the vector (2,0,1) which is shown in the following chunk of code:

> plot(forecast(auto.arima(stock_data_train, trace = TRUE), h = 24))

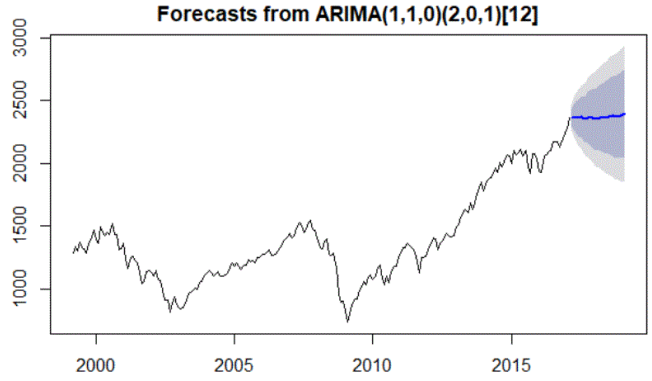

Figure 8: Auto ARIMA plot without seasonality

The **ARIMA** model is made to run using the kernel specifying the *SEASONAL DIFFERENCE PARAMETER (R),* the vector (2,1,1) which is shown in the following chunk of code:

*> plot(forecast(auto.arima(stock_data_train , **D = 1**, trace = TRUE), h = 24))*

The forecast for the stationary data is also obtained using the same **ARIMA** model , as shown in the following code snippet:

> plot(forecast(auto.arima(stock_diff , D = 1, trace = TRUE), h = 24))

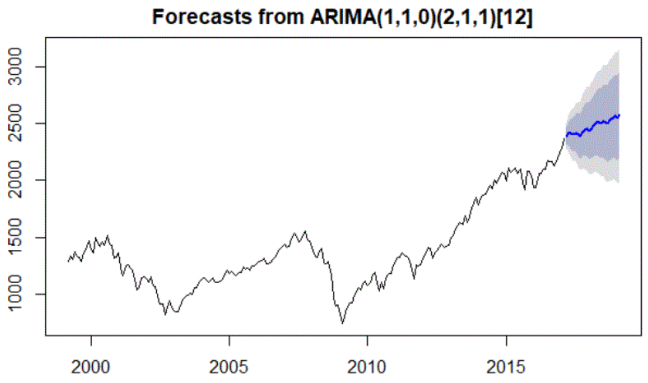

Figure 9: Auto ARIMA forecast with seasonality

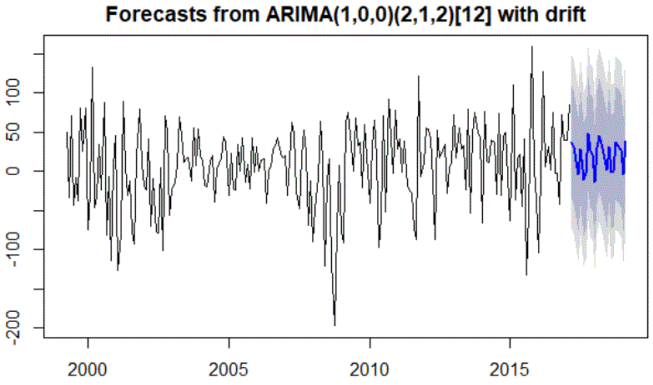

Figure : Auto ARIMA forecast for stationary data

RESULT

The resulting forecast plots shown above have the darker grey boundary to be the ***80% Confidence Interval*** and the lighter grey boundary to be the ***95% Confidence Interval.*** One interesting result that was observed from the plots was, the forecast plot without the seasonal difference parameter did not give approximate forecast value line, but it gave a tighter bound for the Confidence Intervals as observed in Figure 8. On the contrary, by including the seasonal difference parameter, the forecast plot gave a better approximation of the forecast value line, but it gave a loose bound for the Confidence Intervals as observed in Figure 9.

REFERENCES

- https://towardsdatascience.com/time-series-forecasting-arima-models-7f221e9eee06
- https://ademos.people.uic.edu/Chapter23.html
- https://www.analyticsvidhya.com/blog/2015/12/complete-tutorial-time-series-modeling/