Prediction and Analysis of New York Stock Exchange

Meghana & Keshav

Group 30

Table of Contents

First Name	Last Name	Online Students?	Monday or Tuesday	Shared with ITMD 525?
Meghana Ashwath	Reddy	Yes	Monday	Yes
Keshav	N R	Yes	Monday	Yes

1. Introduction and Motivations:

Stock market analysis is the foremost thing which is mandatory prior to any financial investment. To be defined in layman terms, stock market analysis refers to the entire procedure of monitoring and analyzing the stocks and thereby calculating the future trends. With the stock prices having the tendency to rise and fall, the whole scenario becomes volatile. However, since a defined pattern is followed by the stocks an insight can be procured after a thorough analysis. Stock market analysis is a process abided by most of the investors.

It is the process of investigating and studying data on existing stocks and trying to predict how they will do in the future market. This is used by most traders since stock prices can change from moment to moment, but they normally have a pattern of either going up or down that can be analyzed and followed. Some investors use what is called a technical analysis. This is mostly used to figure out the possible return the stock will provide its owners. After performing this out of analysis we will aid the traders gain insights on various stock listed in the stock market.

1.1. Tools\Languages Used:

WEKA - It is a widely-used toolkit for machine learning and data mining originally developed at the University of Waikato in New Zealand. It contains a large collection of state-of-the-art machine learning and data mining algorithms written in Java. WEKA contains tools for regression, classification, clustering, association rules, visualization, and data pre-processing.

R-Programming - R is a programming language and software environment for statistical computing and graphics supported by the R Foundation for Statistical Computing. The R language is widely used among statisticians and data miners for developing statistical software and data analysis

2. Data Description

New York stock exchange dataset provides information about the S&P 500 companies historical prices with fundamental data. Prices were fetched from Yahoo Finance, fundamentals are from Nasdaq Financials, extended by some fields from EDGAR SEC databases.

Source of Data: **https://www.kaggle.com/dgawlik/nyse**
Total no of rows**: 1704818 rows**
Total Number of Variables: **~100**
Time period**: 2010-2016**
Domain: **Financial Sector**

Data files description:

prices.csv: This file contains the daily prices of the stock from 2010 to the end of 2016. This data set is being use for the prediction and forecasting purpose using Time Series Model.

prices-split-adjusted.csv

fundamentals.csv: This file consists of metrics extracted from annual SEC 10K fillings over the range between 2012 to 2016 which is used to derive most of popular fundamental indicators.

fundamentals_Missing values

3. Research Problems and Solutions

1. To check the accuracy of the bankruptcy of around 250 companies for span of 2013 to 2016 using Naïve Bayes and Space Vector Machine classification algorithms on Current ratio factor.
2. Prediction and forecasting of Yahoo Stock shares using Time series models.

4. Model Learning:

- Data cleaning was done by eliminating the null values and filling the null values manually.
- Data was subjected to feature selection to get the essential features.
- Date field in the prices.csv was modified.

4.1 Data Processing:

- Data set had to be pre-processed by filling all the missing values manually.
- For data mining process, out of 78 attributes had to select 10 valuable attributes using feature selection using fundamental.csv file.
- Label type had to be renamed to 'Status' and all the 3 classes under the Category label where replaced with meaningful name, namely Healthy, Unhealthy and Not known where,
    i. healthy indicated that there was no possibility of bankruptcy.
    ii. unhealthy indicated possibility of bankruptcy.
    iii. Not known indicated that unpredictability of the outcome.
- For data analytics purpose, only data for Yahoo stock exchange was selected using the subset function in R using prices.csv file.

4.2 Data Mining/Analytics Tasks and Processes:

4.2.1 Data Mining Tasks and Processes:

The Classification Models used for classification are Naïve Bayes Algorithm, Space Vector Algorithm and K-Nearest Neighbor Classifier.

a) **Naïve Bayes Algorithm:**
    - Bayesian classification is based on Bayes' Theorem. Bayesian classifiers are the statistical classifiers. Bayesian classifiers can predict class membership probabilities such as the probability that a given tuple belongs to a particular class.
    - For our analysis, we have run the algorithm on the dataset using Weka with class labels as Healthy, Unhealthy and Not known.

b) **Support Vector Machine Classification Algorithm:**
    - Support vector machine (SVM) is a supervised learning methods used for classification and regression. But it is mainly used for classification problems.
    - In this algorithm, we plot each data item as a point in n-dimensional space and after performing classification, a hyper plane is created that differentiates the two classes.
    - For an efficient system recall and precession should be equal to or nearly equal to 1.

c) **K-Nearest Neighbor Classifier:**
    - K-Nearest neighbor (KNN) is a classification strategy that is an

example of a "lazy learner." Unlike all the other classification algorithms "lazy learners" do not require building a model with a training set before actual use. A lazy learner like k-nearest neighbors uses the training set directly to classify an input when an input is given.

- ➢ We run 3 iterations for different K-values (k=5, k=7, k=10) which yields the best accuracy on Weka.

Classification Model	Accuracy Obtained
Naïve Bayes	97.12%
Space Vector Machine	97.20%
KNN-5	99.22%
KNN-7	98.24%
KNN-10	98.80%

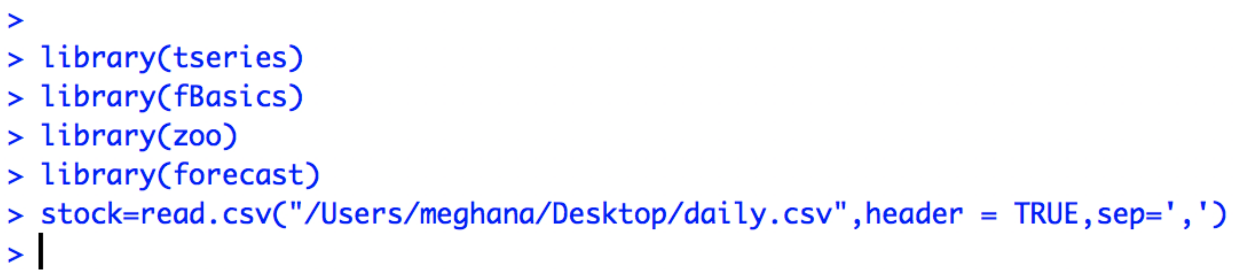
4.2.2 Data Analytics Tasks and Processes:

Time series model:

A time series is a series of data points indexed (or listed or graphed) in time order. Most commonly, a time series is a sequence taken at successive equally spaced points in time. Time series forecasting is the use of a model to predict future values based on previously observed values. In this project we have built AR, MA and ARIMA to predict and forecast the future values. And evaluated the best model based on MAE value of the model.

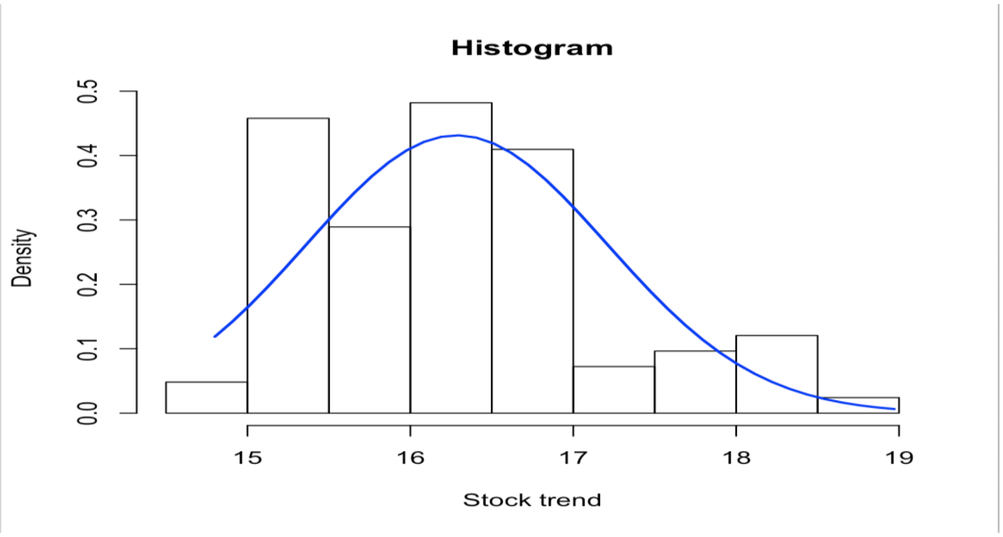
Time series model was built as per the following steps:

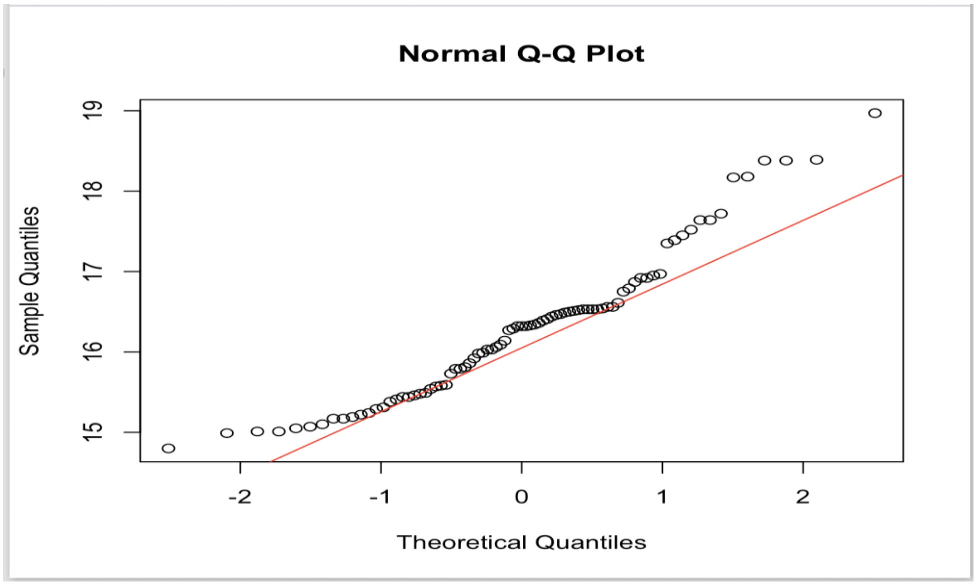
1. Data for last month was withheld for evaluation process.
2. Libraries for the time series was loaded
3. Remaining data was loaded in to R for the date range from 2010-2016.



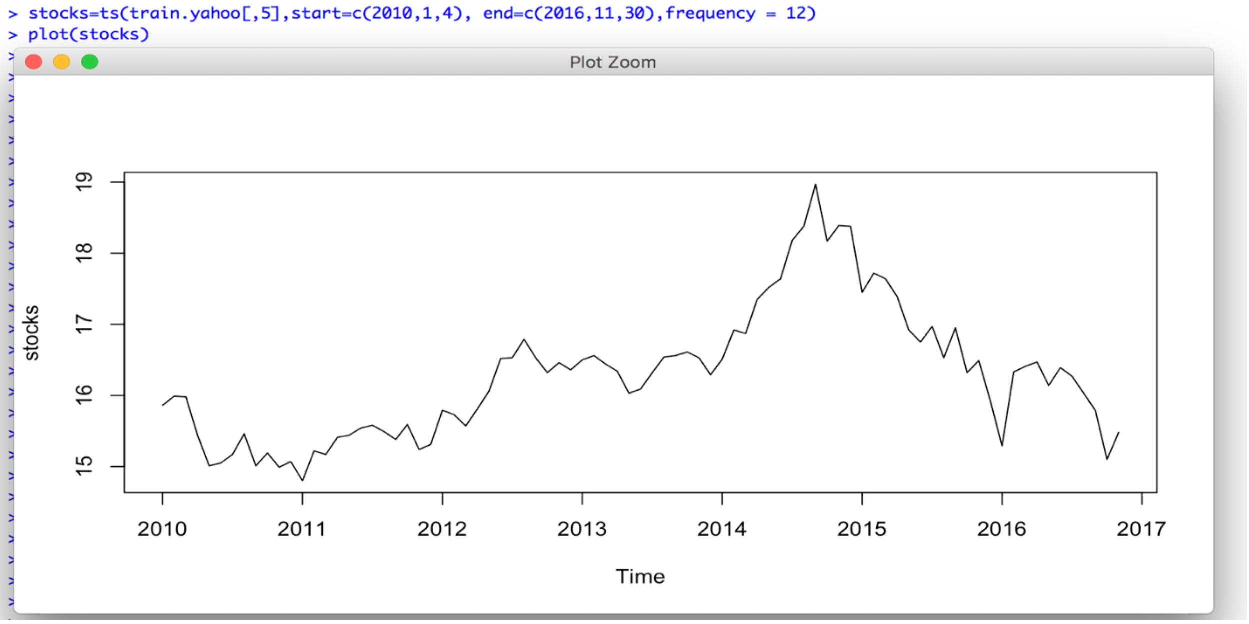
```
>
> library(tseries)
> library(fBasics)
> library(zoo)
> library(forecast)
> stock=read.csv("/Users/meghana/Desktop/daily.csv",header = TRUE,sep=',')
> |
```

4. Plotted the Histogram and QQ plot to check the distribution.





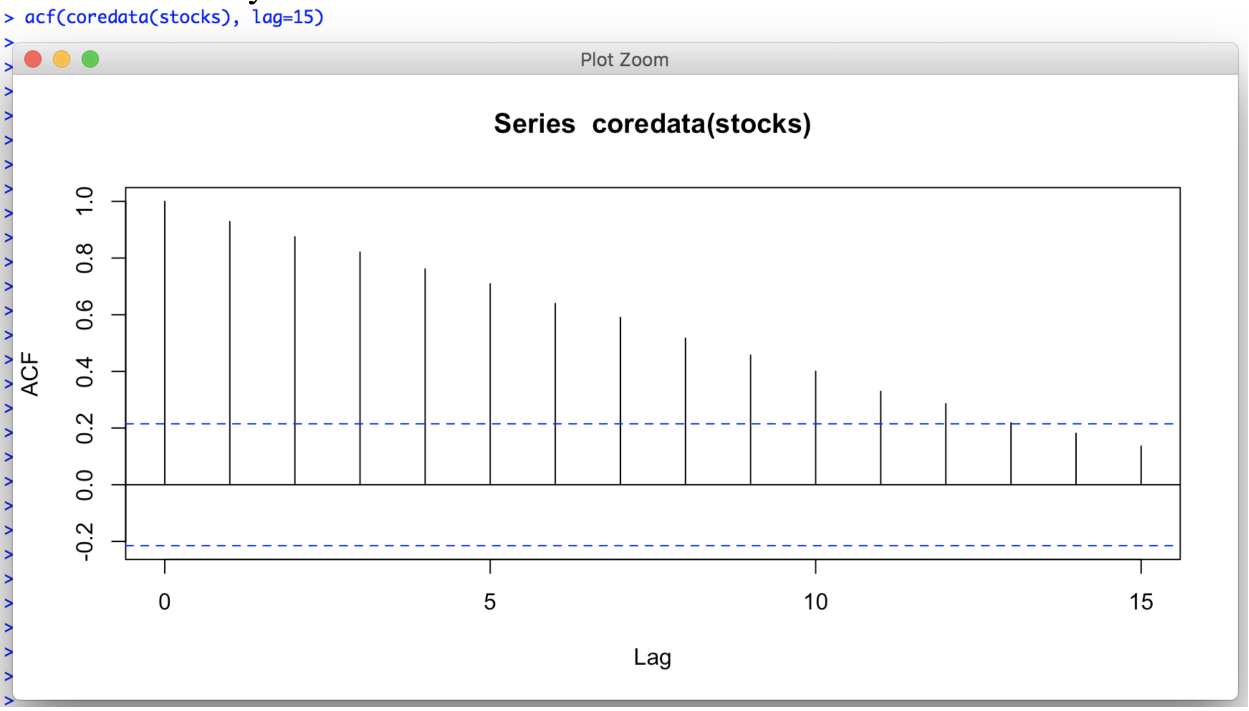
5. And plotted Time series plot and to check if it is stationary or not.



```
> stocks=ts(train.yahoo[,5],start=c(2010,1,4), end=c(2016,11,30),frequency = 12)
> plot(stocks)
```

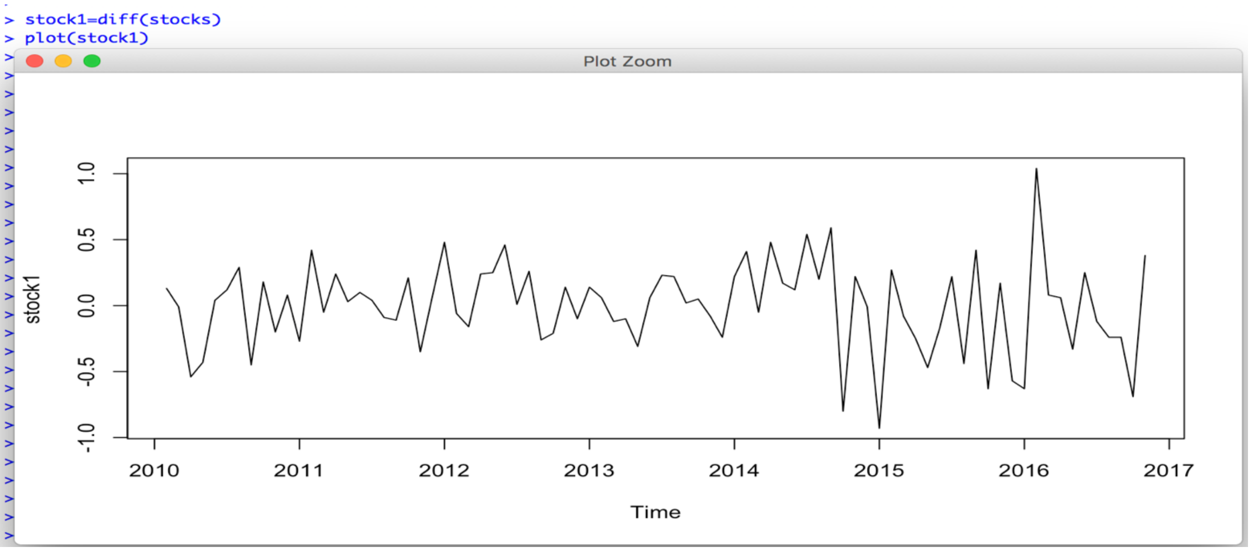
Since the mean and variance change over time and it shows trends, it can be concluded that the data is nonstationary.

6. Observed the serial correlation using ACF plot on ts object and found to be not serially correlated.



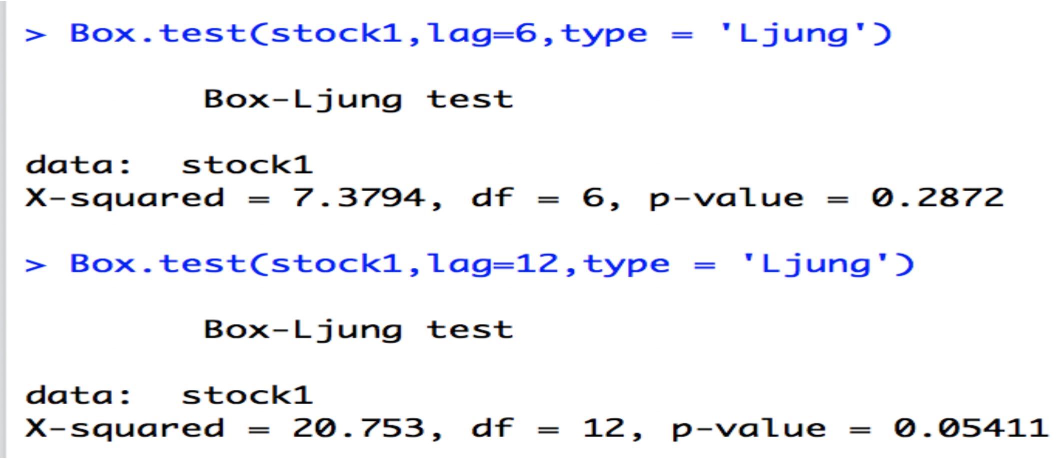
```
> acf(coredata(stocks), lag=15)
```

7. Applied the difference on the ts object and observed.



```
> stock1=diff(stocks)
> plot(stock1)
```

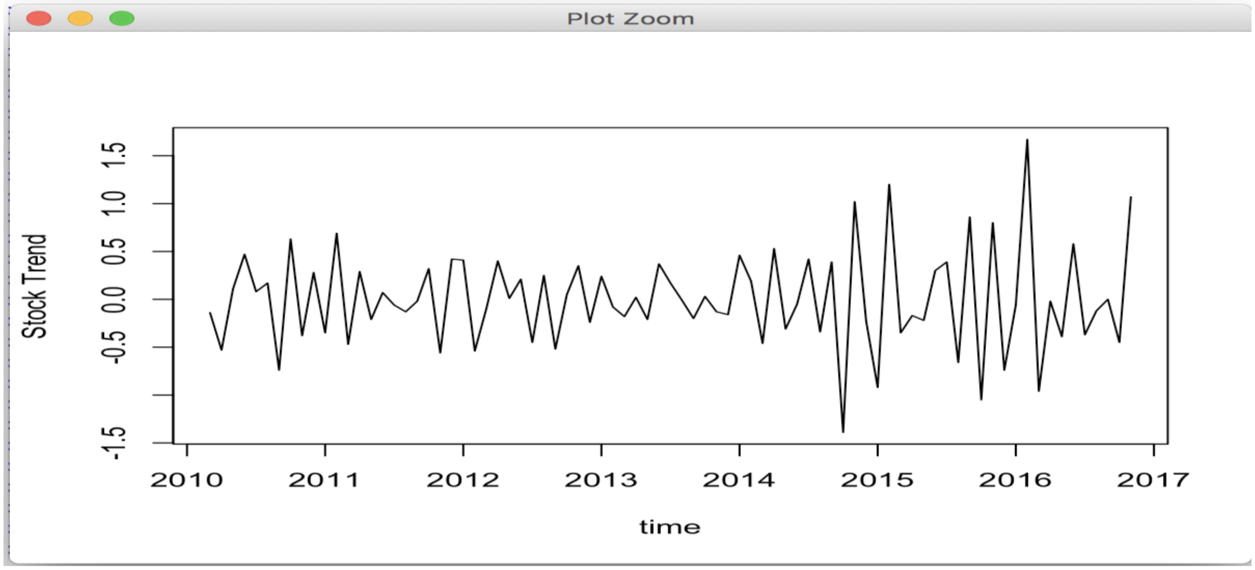
8. Ljung Test at lag 6 and lag 12 to evaluate if the stock closing rates are serially correlated.



```
> Box.test(stock1,lag=6,type = 'Ljung')

        Box-Ljung test

data:  stock1
X-squared = 7.3794, df = 6, p-value = 0.2872

> Box.test(stock1,lag=12,type = 'Ljung')

        Box-Ljung test

data:  stock1
X-squared = 20.753, df = 12, p-value = 0.05411
```

From the above test results, we can see that p value is greater than 0.05 hence it's not serially correlated so we go for 2nd difference.

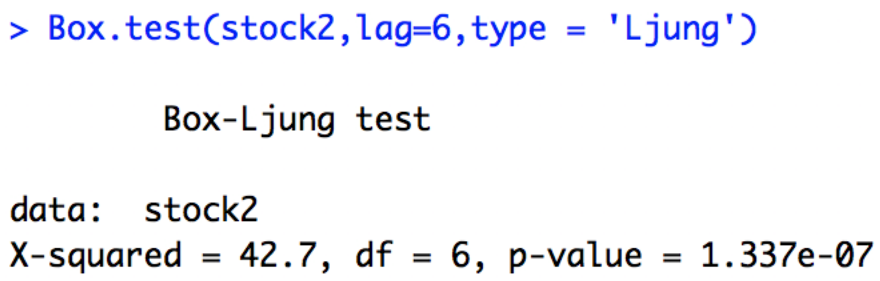
9. Applying the second difference and then plotting,



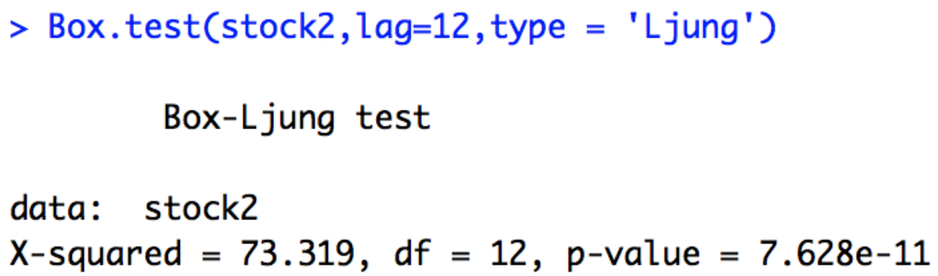
Since the mean and variance do not change much over the time it can be concluded that the data is stationary.

10.

Again, applying the Ljung test we see that p value is less than 0.05 hence it serially correlated.



```
> Box.test(stock2,lag=6,type = 'Ljung')

        Box-Ljung test

data:  stock2
X-squared = 42.7, df = 6, p-value = 1.337e-07
```

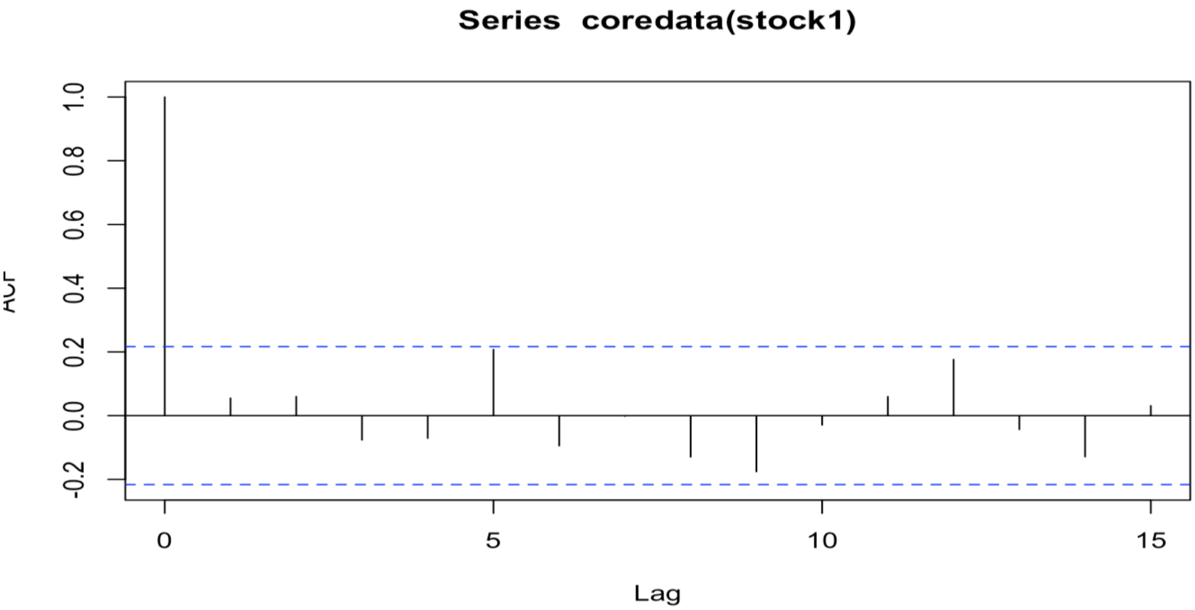


```
> Box.test(stock2,lag=12,type = 'Ljung')

        Box-Ljung test

data:  stock2
X-squared = 73.319, df = 12, p-value = 7.628e-11
```

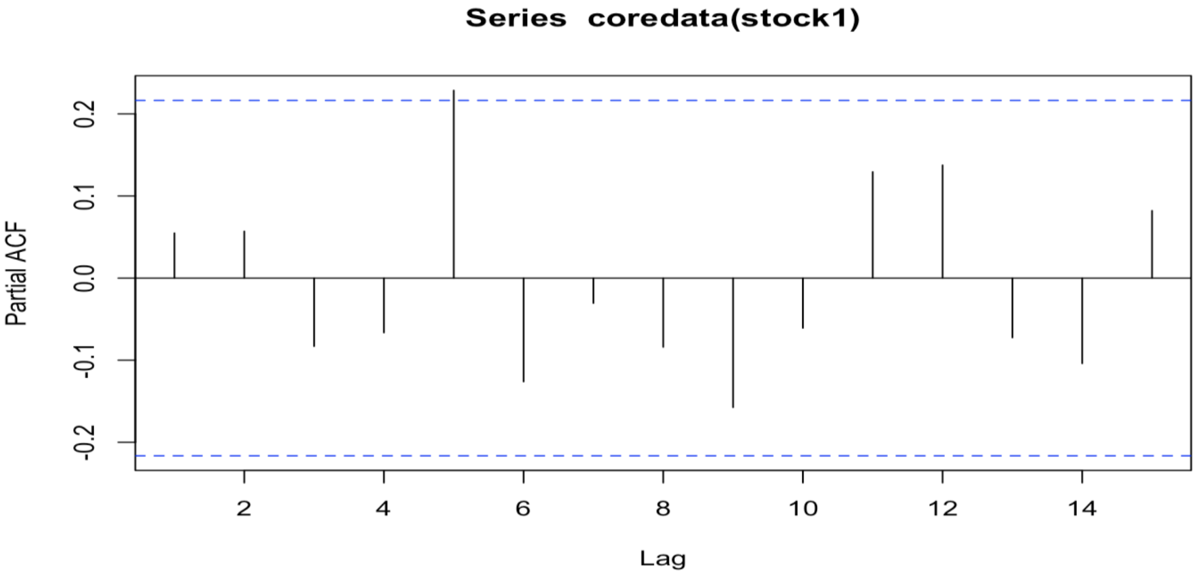
11.

Plot ACF and PACF to find the p and q values to build the model,

ACF plot,

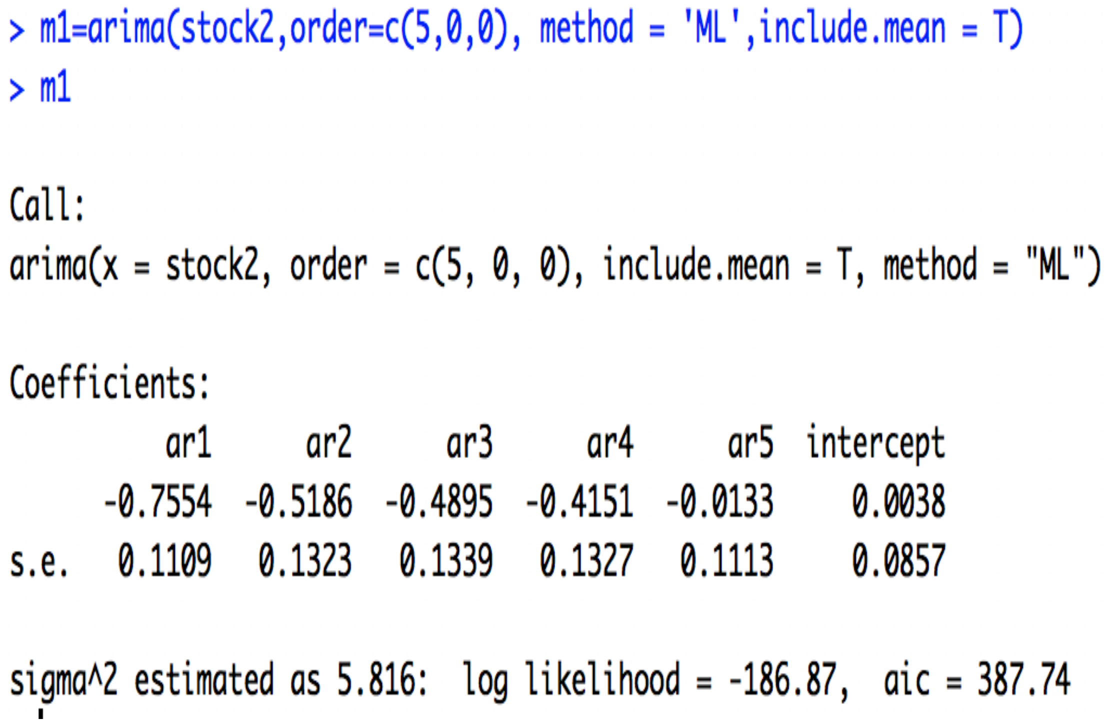


PACF plot,



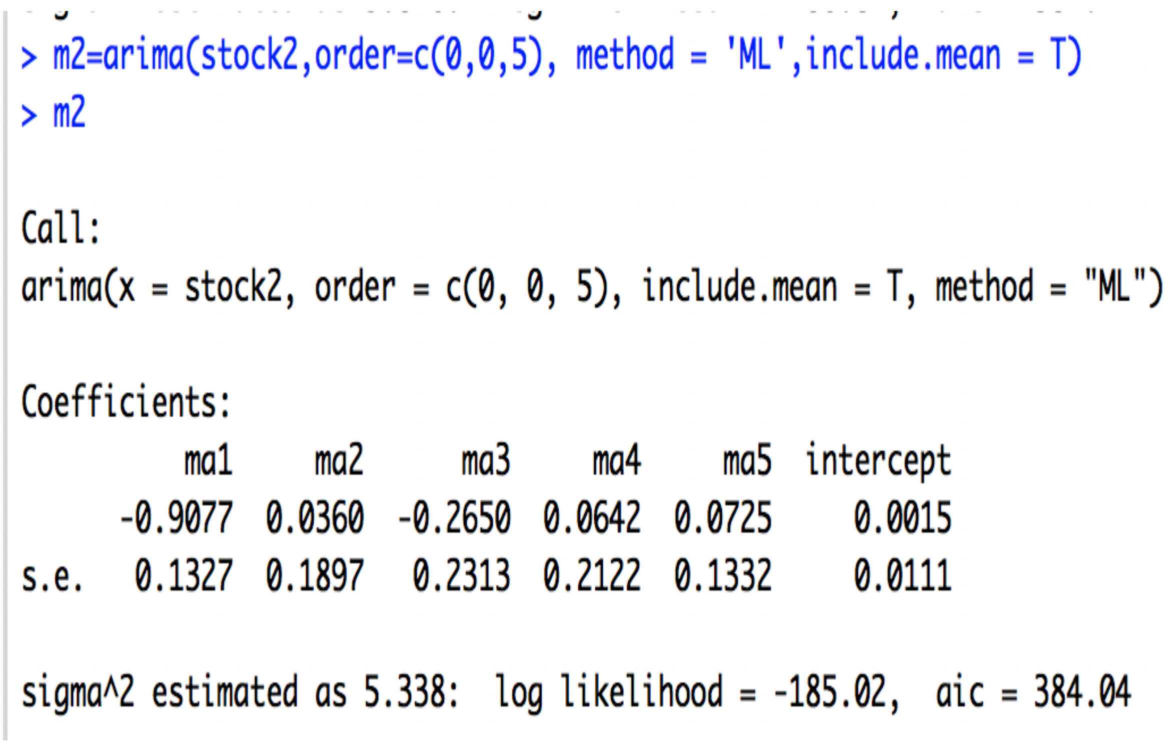
From the above plot, we can see that p value is 5 and q value is 5.

12. AR Model



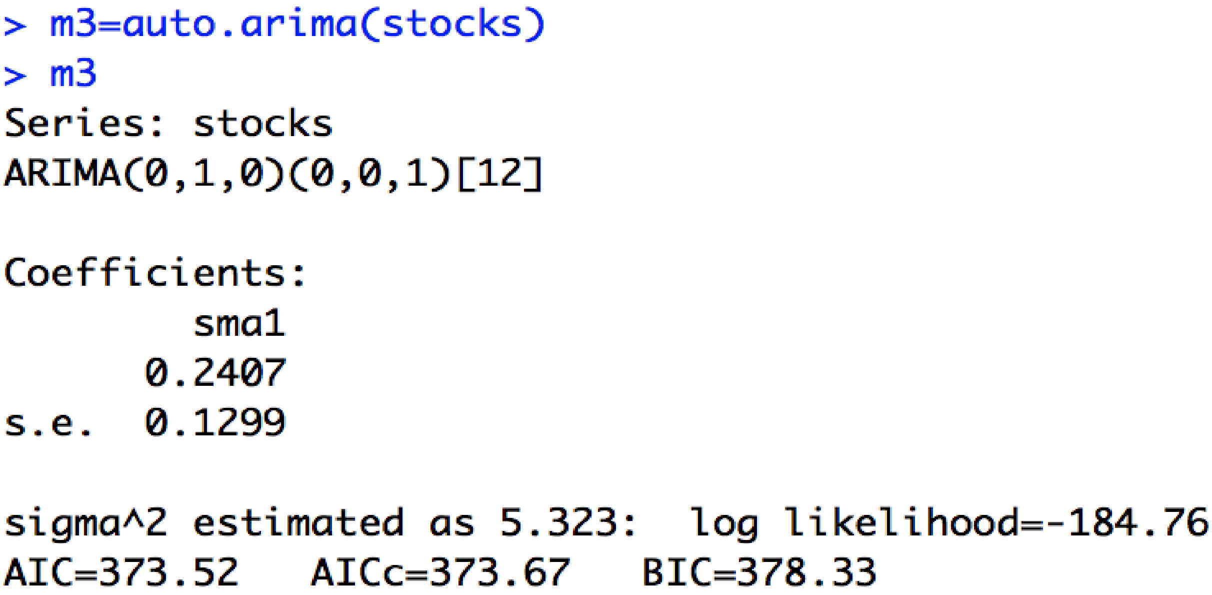
```
> m1=arima(stock2,order=c(5,0,0), method = 'ML',include.mean = T)
> m1

Call:
arima(x = stock2, order = c(5, 0, 0), include.mean = T, method = "ML")

Coefficients:
          ar1      ar2      ar3      ar4      ar5  intercept
      -0.7554  -0.5186  -0.4895  -0.4151  -0.0133     0.0038
s.e.   0.1109   0.1323   0.1339   0.1327   0.1113     0.0857

sigma^2 estimated as 5.816:  log likelihood = -186.87,  aic = 387.74
```

13. MA Model



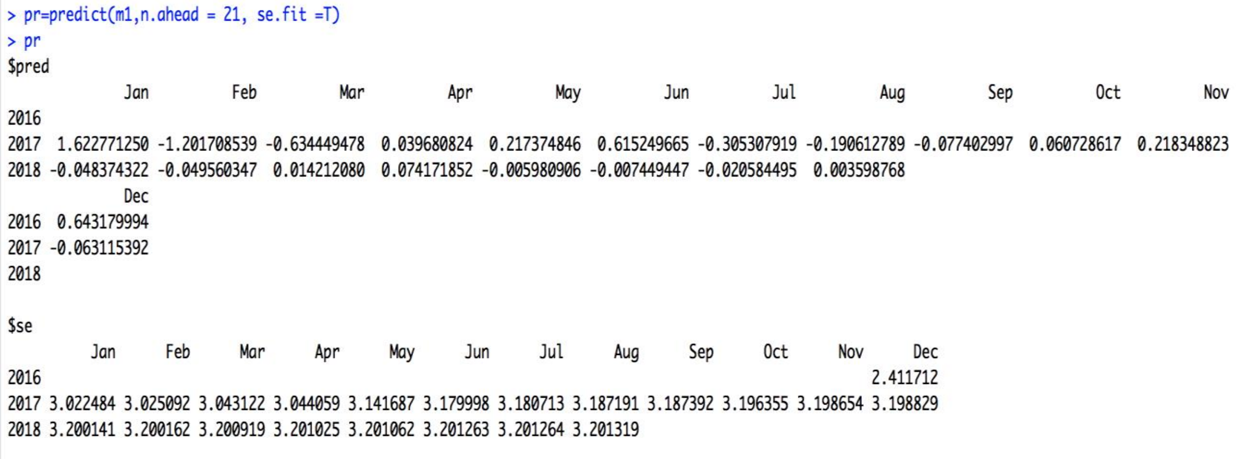
```
> m2=arima(stock2,order=c(0,0,5), method = 'ML',include.mean = T)
> m2

Call:
arima(x = stock2, order = c(0, 0, 5), include.mean = T, method = "ML")

Coefficients:
          ma1     ma2      ma3     ma4     ma5  intercept
      -0.9077  0.0360  -0.2650  0.0642  0.0725     0.0015
s.e.   0.1327  0.1897   0.2313  0.2122  0.1332     0.0111

sigma^2 estimated as 5.338:  log likelihood = -185.02,  aic = 384.04
```

14. ARIMA model

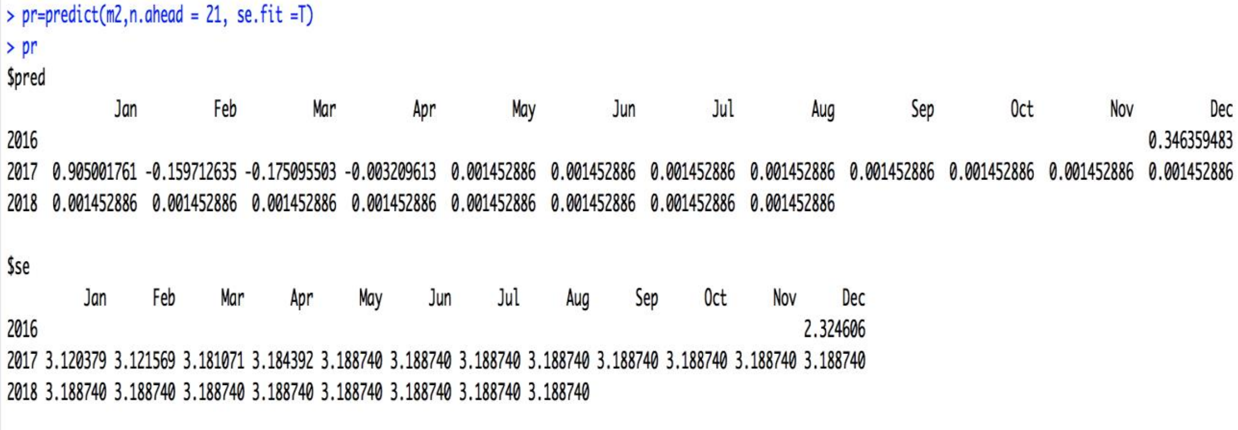


```
> m3=auto.arima(stocks)
> m3
Series: stocks
ARIMA(0,1,0)(0,0,1)[12]

Coefficients:
        sma1
      0.2407
s.e.  0.1299

sigma^2 estimated as 5.323:  log likelihood=-184.76
AIC=373.52   AICc=373.67   BIC=378.33
```

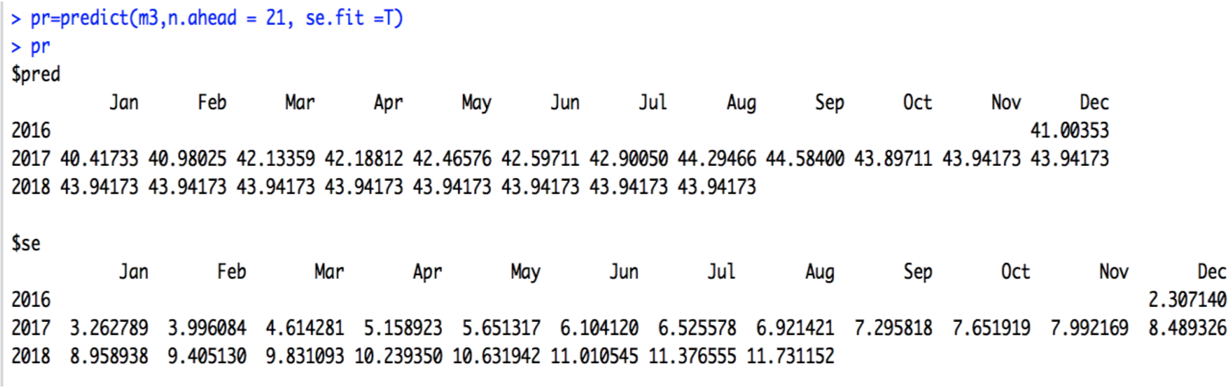
15. We build the prediction models for AR, MA and ARIMA,



```
> pr=predict(m1,n.ahead = 21, se.fit =T)
> pr
$pred
              Jan          Feb          Mar          Apr          May          Jun          Jul          Aug          Sep          Oct          Nov
2016
2017  1.622771250 -1.201708539 -0.634449478  0.039680824  0.217374846  0.615249665 -0.305307919 -0.190612789 -0.077402997  0.060728617  0.218348823
2018 -0.048374322 -0.049560347  0.014212080  0.074171852 -0.005980906 -0.007449447 -0.020584495  0.003598768
              Dec
2016  0.643179994
2017 -0.063115392
2018

$se
          Jan      Feb      Mar      Apr      May      Jun      Jul      Aug      Sep      Oct      Nov      Dec
2016                                                                                                    2.411712
2017 3.022484 3.025092 3.043122 3.044059 3.141687 3.179998 3.180713 3.187191 3.187392 3.196355 3.198654 3.198829
2018 3.200141 3.200162 3.200919 3.201025 3.201062 3.201263 3.201264 3.201319
```



```
> pr=predict(m2,n.ahead = 21, se.fit =T)
> pr
$pred
              Jan          Feb          Mar          Apr          May          Jun          Jul          Aug          Sep          Oct          Nov          Dec
2016                                                                                                                                                 0.346359483
2017  0.905001761 -0.159712635 -0.175095503 -0.003209613  0.001452886  0.001452886  0.001452886  0.001452886  0.001452886  0.001452886  0.001452886  0.001452886
2018  0.001452886  0.001452886  0.001452886  0.001452886  0.001452886  0.001452886  0.001452886  0.001452886

$se
          Jan      Feb      Mar      Apr      May      Jun      Jul      Aug      Sep      Oct      Nov      Dec
2016                                                                                                    2.324606
2017 3.120379 3.121569 3.181071 3.184392 3.188740 3.188740 3.188740 3.188740 3.188740 3.188740 3.188740 3.188740
2018 3.188740 3.188740 3.188740 3.188740 3.188740 3.188740 3.188740 3.188740
```



```
> pr=predict(m3,n.ahead = 21, se.fit =T)
> pr
$pred
          Jan      Feb      Mar      Apr      May      Jun      Jul      Aug      Sep      Oct      Nov      Dec
2016                                                                                                  41.00353
2017 40.41733 40.98025 42.13359 42.18812 42.46576 42.59711 42.90050 44.29466 44.58400 43.89711 43.94173 43.94173
2018 43.94173 43.94173 43.94173 43.94173 43.94173 43.94173 43.94173 43.94173

$se
           Jan       Feb       Mar       Apr       May       Jun       Jul       Aug       Sep       Oct       Nov       Dec
2016                                                                                                            2.307140
2017  3.262789  3.996084  4.614281  5.158923  5.651317  6.104120  6.525578  6.921421  7.295818  7.651919  7.992169  8.489326
2018  8.958938  9.405130  9.831093 10.239350 10.631942 11.010545 11.376555 11.731152
```

5. Evaluations and Results

5.1 Evaluation Methods for Data mining classification model

The Evaluation method used for the classification process is percentage split of the dataset, where 80% was used as Training data set and remaining 20% as Test data set.

5.1.1 Evaluation Metrics

Predictive (Classification) Accuracy:

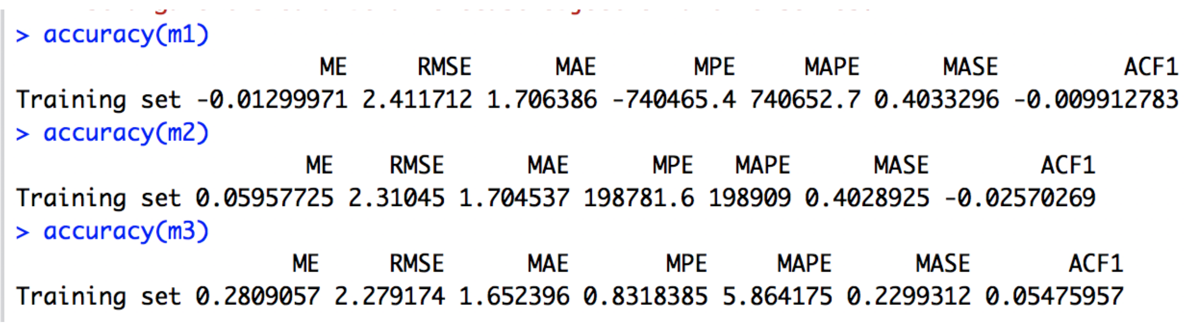
- This refers to the ability of the model to correctly predict the class label from the training dataset.
- Accuracy = Percentage (%) of testing set examples correctly classified by the classifier.

5.2 Time Series Model Evaluation

It can be evaluated by building the prediction models and then calculating the MAE value on test data set which was retained earlier and then performing the residual analysis.

5.2.1 Evaluation Metrics

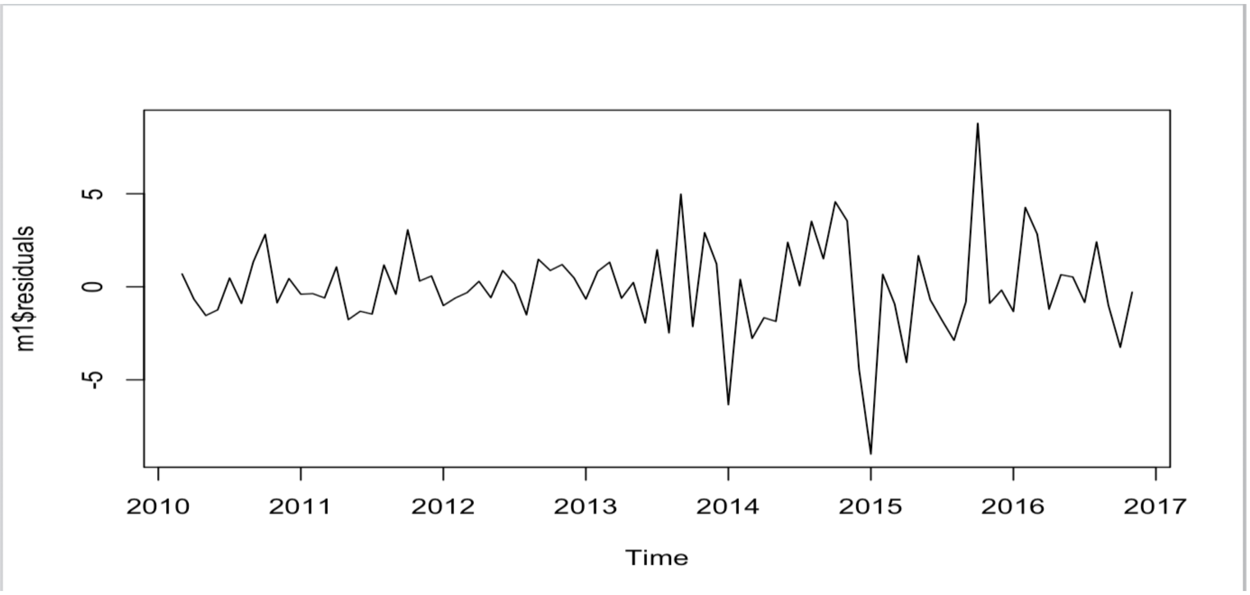
MAE value: By comparing the MAE value of 3 models we can say that m3 model which was create using auto ARIMA is best.

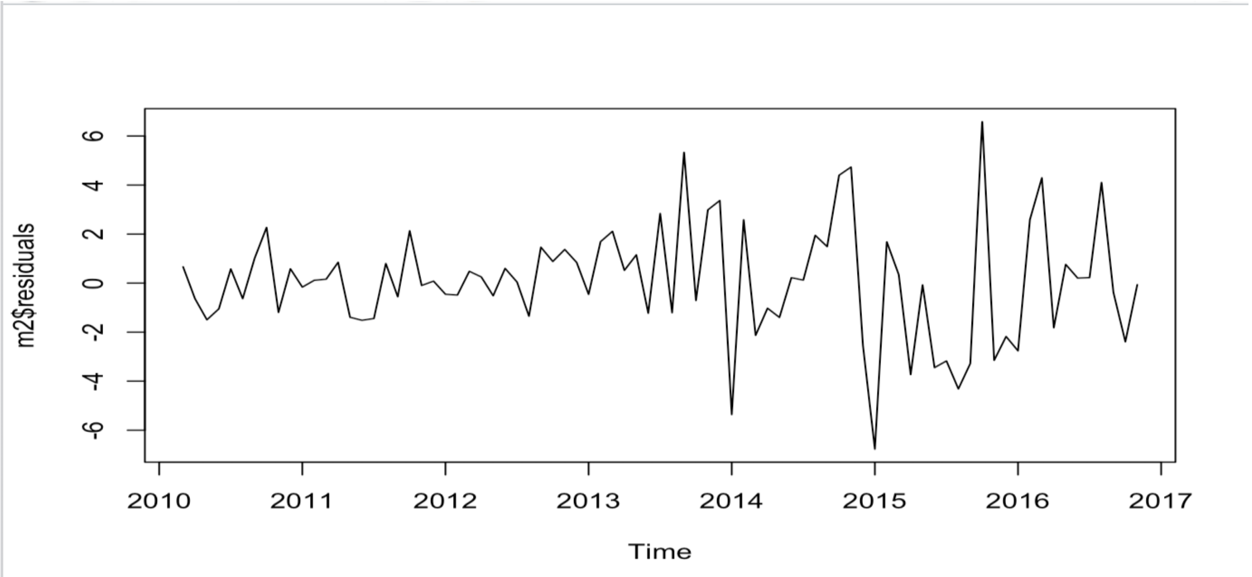


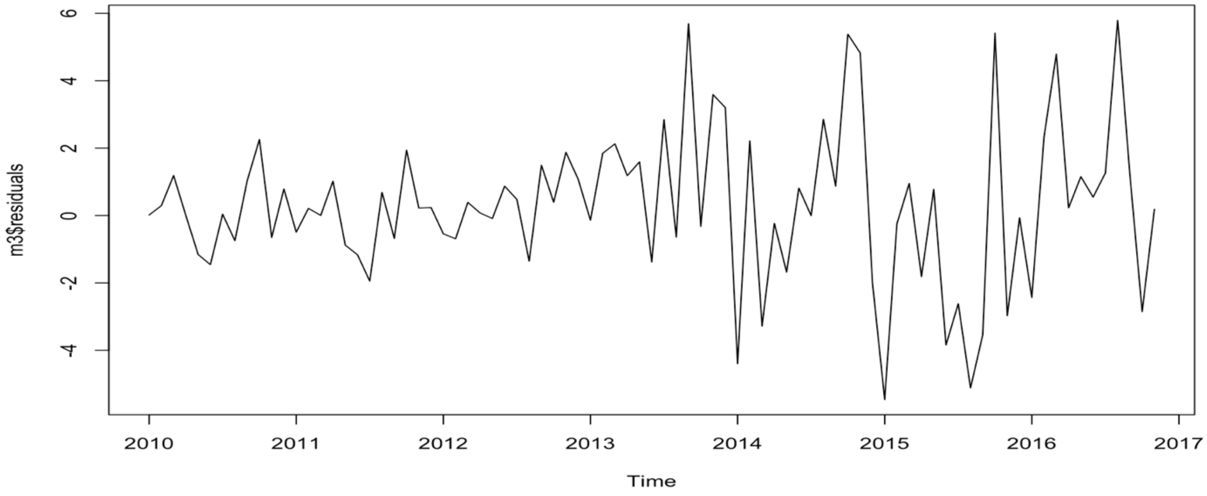
```
> accuracy(m1)
                      ME     RMSE      MAE       MPE     MAPE      MASE         ACF1
Training set -0.01299971 2.411712 1.706386 -740465.4 740652.7 0.4033296 -0.009912783
> accuracy(m2)
                     ME    RMSE      MAE      MPE   MAPE      MASE        ACF1
Training set 0.05957725 2.31045 1.704537 198781.6 198909 0.4028925 -0.02570269
> accuracy(m3)
                    ME     RMSE      MAE       MPE     MAPE      MASE       ACF1
Training set 0.2809057 2.279174 1.652396 0.8318385 5.864175 0.2299312 0.05475957
```

Residual Analysis:

a. By plotting the residuals for all models m1, m2 & m3.





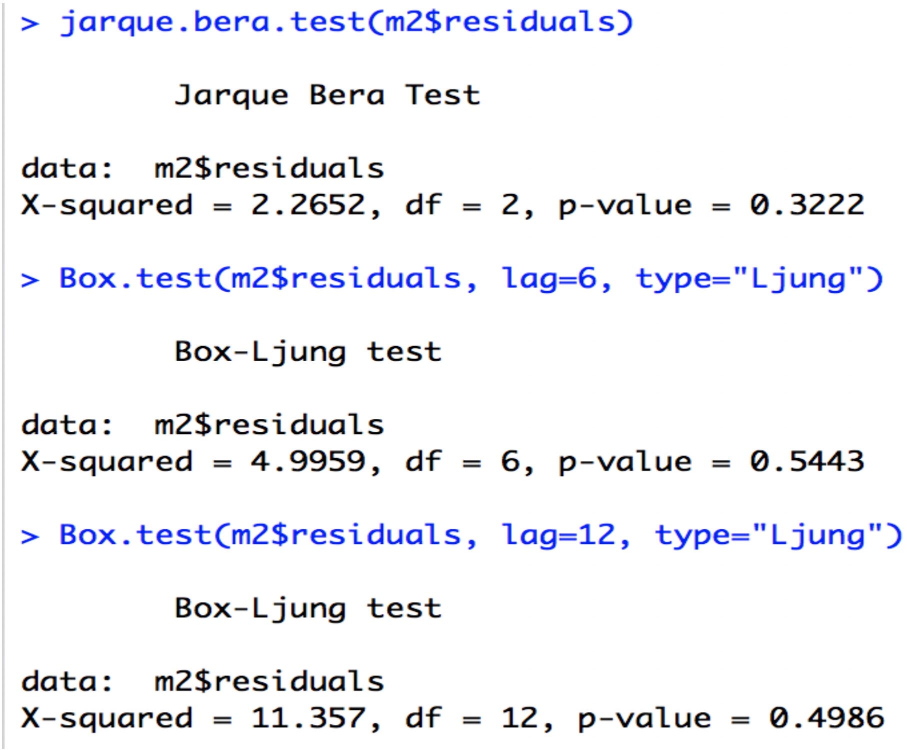


b. Using Jarque bera test to check normality and Ljung test to check the white noise.

Model m1:

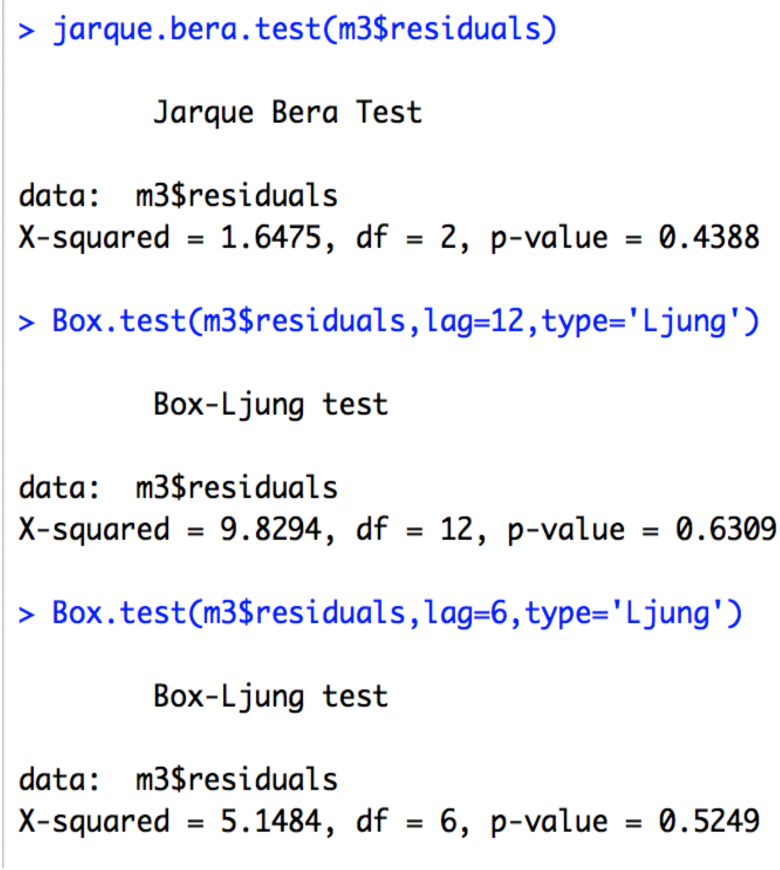
```
> jarque.bera.test(m1$residuals)

        Jarque Bera Test

data:  m1$residuals
X-squared = 36.704, df = 2, p-value = 1.071e-08

> Box.test(m1$residuals, lag=6, type="Ljung")

        Box-Ljung test

data:  m1$residuals
X-squared = 3.8808, df = 6, p-value = 0.6928

> Box.test(m1$residuals, lag=12, type="Ljung")

        Box-Ljung test

data:  m1$residuals
X-squared = 12.768, df = 12, p-value = 0.3862
```

Model m2:



```
> jarque.bera.test(m2$residuals)

        Jarque Bera Test

data:  m2$residuals
X-squared = 2.2652, df = 2, p-value = 0.3222

> Box.test(m2$residuals, lag=6, type="Ljung")

        Box-Ljung test

data:  m2$residuals
X-squared = 4.9959, df = 6, p-value = 0.5443

> Box.test(m2$residuals, lag=12, type="Ljung")

        Box-Ljung test

data:  m2$residuals
X-squared = 11.357, df = 12, p-value = 0.4986
```

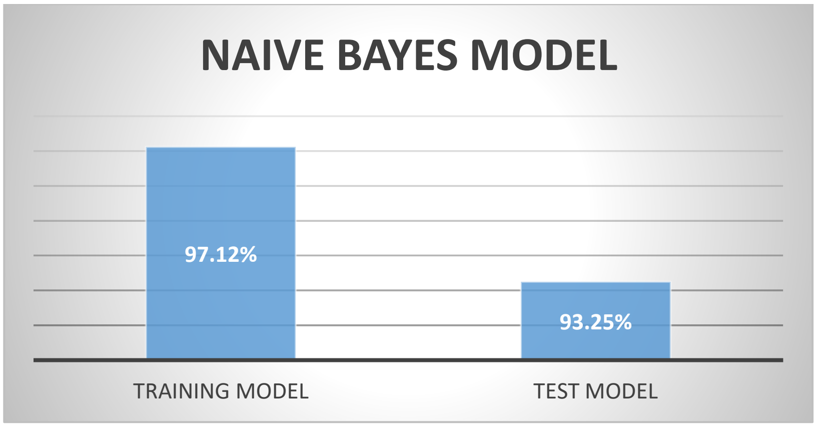
Model m3:

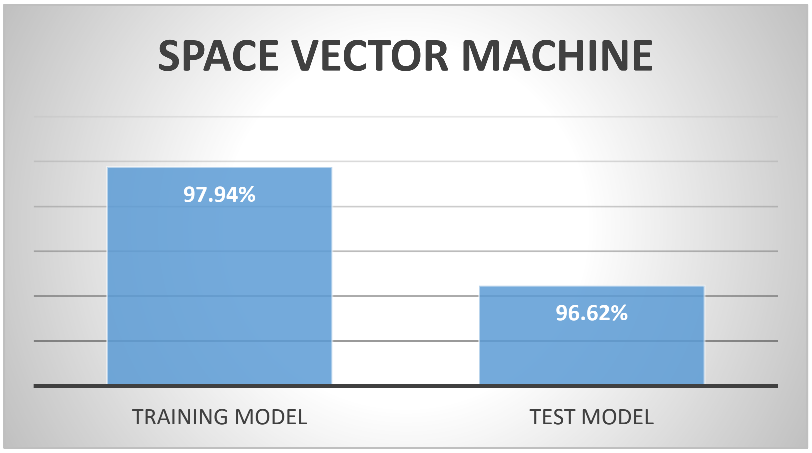


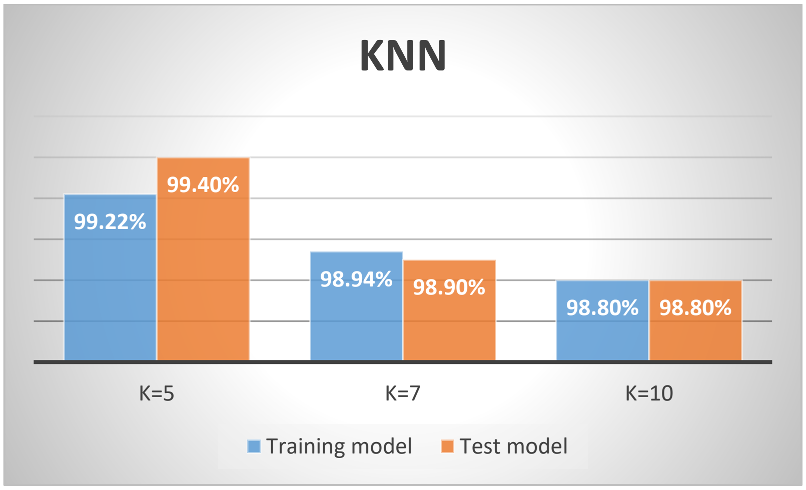
```
> jarque.bera.test(m3$residuals)

        Jarque Bera Test

data:  m3$residuals
X-squared = 1.6475, df = 2, p-value = 0.4388

> Box.test(m3$residuals,lag=12,type='Ljung')

        Box-Ljung test

data:  m3$residuals
X-squared = 9.8294, df = 12, p-value = 0.6309

> Box.test(m3$residuals,lag=6,type='Ljung')

        Box-Ljung test

data:  m3$residuals
X-squared = 5.1484, df = 6, p-value = 0.5249
```

5.3. Results and Findings

<u>Data Mining:</u>







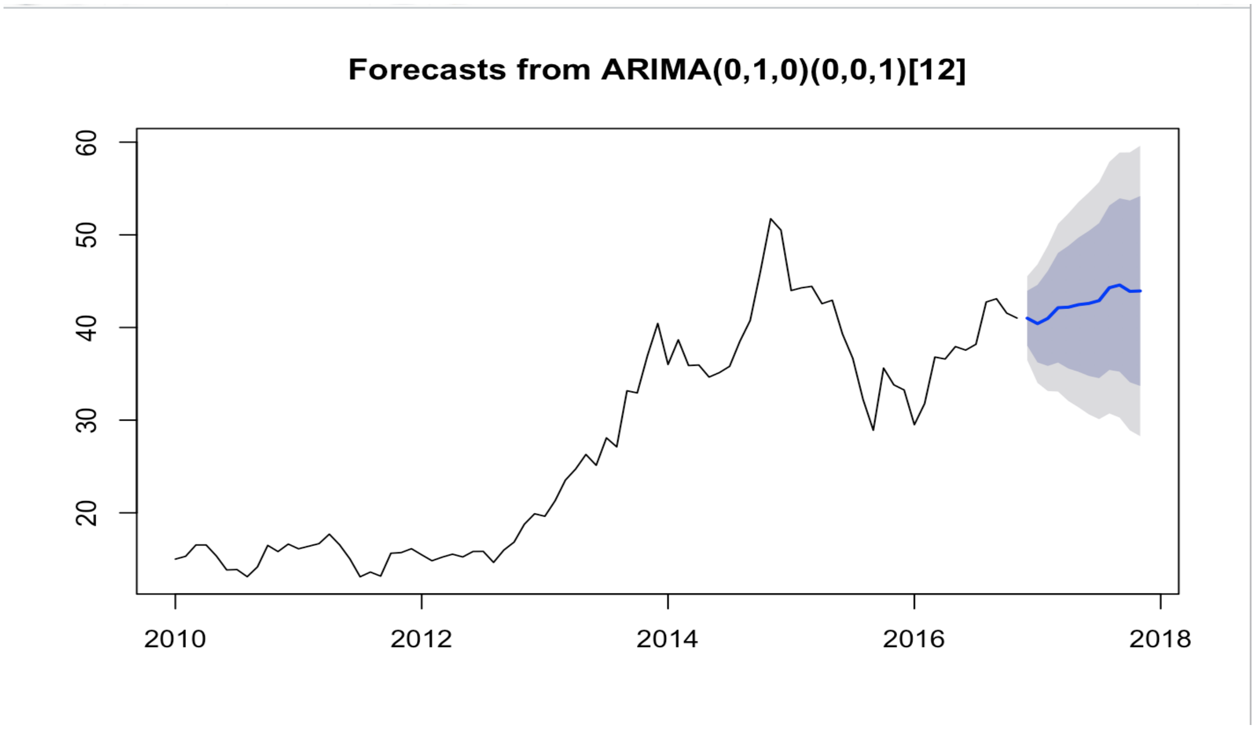
By looking at all the models output with respect to accuracy on both training and test data we can say that KNN out performs the other 2 classification algorithms in terms of accuracy. So, for this dataset we can say that KNN is much more applicable than other algorithms.

Data Analytics:

By observing the AIC and MAE values, we can see that ARIMA model is best because least is the value of MAE best is the model where MAE indicate the error with respect to the predicted and the actual values.

Since it is the best model, we have used ARIMA to forecast the future values. From the below forecasting plot, we can say that there is a trend in the future values.

By looking at residual plot we can see that data is stationary and with box test we can say that white noise is present since p value is greater than 0.05 hence ARIMA is an acceptable model.



6. Conclusions and Future Work

6.1. Conclusions

- For the data set looking at the accuracy level we can say that KNN is best based on the accuracy.
- By looking at MAE value we can tell that ARIMA model is the best since it has least MAE value.

6.2. Limitations

- Unlike other algorithms, KNN classifier does not predict the important features from the dataset
- KNN does not have ability to generate probabilities for each class.
- Historical data may not give a true picture of an underlying trend and the passage of time will inevitably introduce new variables in time series.

6.3. Potential Improvements or Future Work

- We have only used the classification algorithms to find the accuracy but these models can be further improved to predict the stock trends.
- In time series, we have studied only with respect to daily stock but in real time it is studied at much far granular level, on the order of minutes or even seconds. We can improve the time series model by looking at intraday trading data in additional to closing price data. By observing intra-day trends, we can create more robust models that capitalize on sudden changes in momentum during intra-day trading.