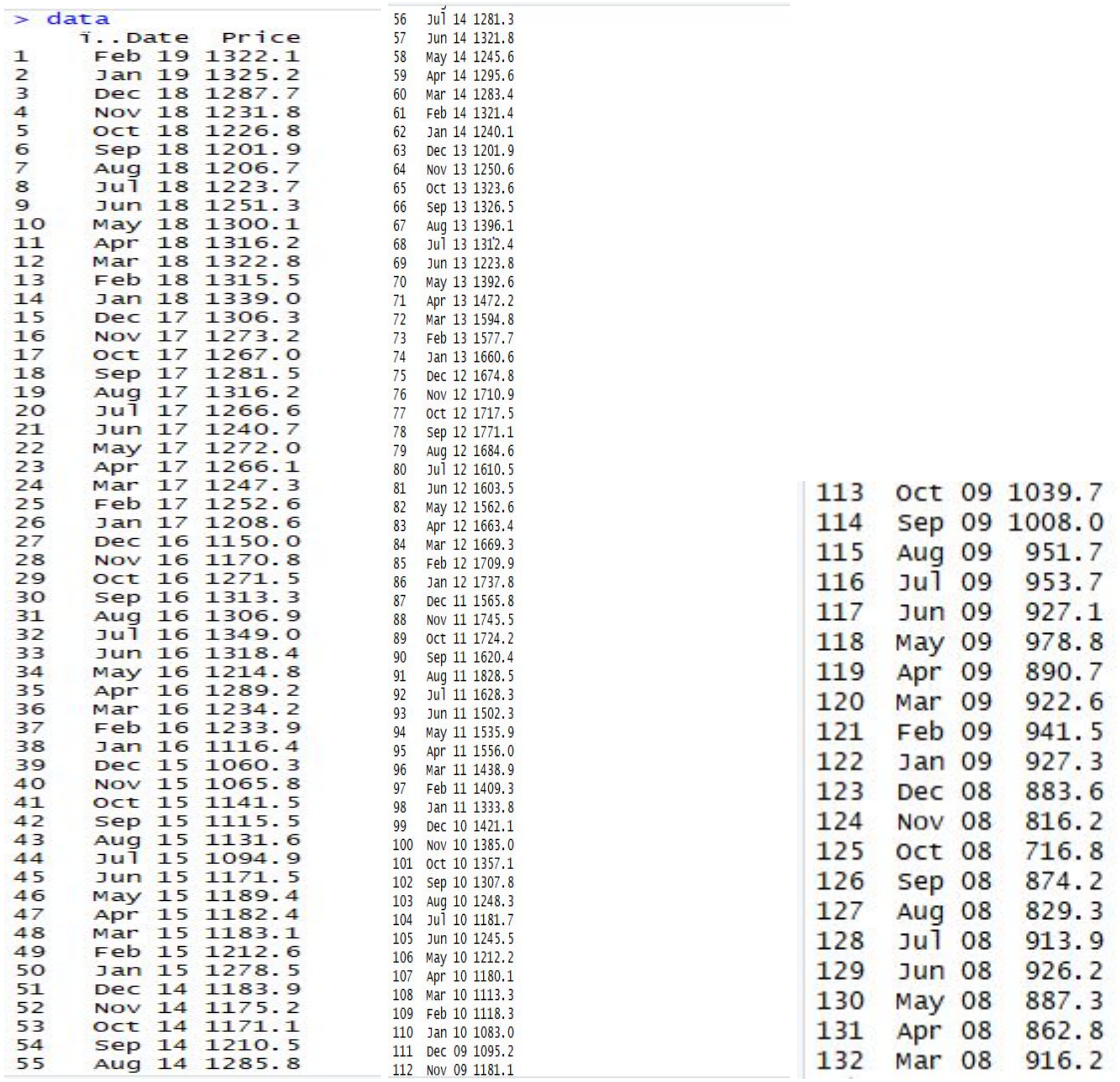
Analysing Gold Data using R and forecasting the future values.

Forecasting involves predicting values for a variable using its historical data points or it can also involve predicting the change in one variable given the change in the value of another variable.
I have used the ARIMA model in my approach to analyse the data and forecast the future values.ARIMA stands for Autoregressive Integrated Moving Average. ARIMA is also known as Box-Jenkins approach.
The ARIMA model combines three basic methods:

- AutoRegression (AR) – In auto-regression, the values of a given time series data are regressed on their own lagged values, which is indicated by the “p” value in the ARIMA model.
- Differencing (I-for Integrated) – This involves differencing the time series data to remove the trend and convert a non-stationary time series to a stationary one. This is indicated by the “d” value in the ARIMA model. If d = 1, it looks at the difference between two-time series entries, if d = 2 it looks at the differences of the differences obtained at d =1, and so forth.
- Moving Average (MA) – The moving average nature of the ARIMA model is represented by the “q” value which is the number of lagged values of the error term.

The value of p,d and q in forecasting our data will be obtained later.

Our dataset:-



```
> data
    ï..Date  Price
1     Feb 19 1322.1
2     Jan 19 1325.2
3     Dec 18 1287.7
4     Nov 18 1231.8
5     Oct 18 1226.8
6     Sep 18 1201.9
7     Aug 18 1206.7
8     Jul 18 1223.7
9     Jun 18 1251.3
10    May 18 1300.1
11    Apr 18 1316.2
12    Mar 18 1322.8
13    Feb 18 1315.5
14    Jan 18 1339.0
15    Dec 17 1306.3
16    Nov 17 1273.2
17    Oct 17 1267.0
18    Sep 17 1281.5
19    Aug 17 1316.2
20    Jul 17 1266.6
21    Jun 17 1240.7
22    May 17 1272.0
23    Apr 17 1266.1
24    Mar 17 1247.3
25    Feb 17 1252.6
26    Jan 17 1208.6
27    Dec 16 1150.0
28    Nov 16 1170.8
29    Oct 16 1271.5
30    Sep 16 1313.3
31    Aug 16 1306.9
32    Jul 16 1349.0
33    Jun 16 1318.4
34    May 16 1214.8
35    Apr 16 1289.2
36    Mar 16 1234.2
37    Feb 16 1233.9
38    Jan 16 1116.4
39    Dec 15 1060.3
40    Nov 15 1065.8
41    Oct 15 1141.5
42    Sep 15 1115.5
43    Aug 15 1131.6
44    Jul 15 1094.9
45    Jun 15 1171.5
46    May 15 1189.4
47    Apr 15 1182.4
48    Mar 15 1183.1
49    Feb 15 1212.6
50    Jan 15 1278.5
51    Dec 14 1183.9
52    Nov 14 1175.2
53    Oct 14 1171.1
54    Sep 14 1210.5
55    Aug 14 1285.8
56    Jul 14 1281.3
57    Jun 14 1321.8
58    May 14 1245.6
59    Apr 14 1295.6
60    Mar 14 1283.4
61    Feb 14 1321.4
62    Jan 14 1240.1
63    Dec 13 1201.9
64    Nov 13 1250.6
65    Oct 13 1323.6
66    Sep 13 1326.5
67    Aug 13 1396.1
68    Jul 13 1312.4
69    Jun 13 1223.8
70    May 13 1392.6
71    Apr 13 1472.2
72    Mar 13 1594.8
73    Feb 13 1577.7
74    Jan 13 1660.6
75    Dec 12 1674.8
76    Nov 12 1710.9
77    Oct 12 1717.5
78    Sep 12 1771.1
79    Aug 12 1684.6
80    Jul 12 1610.5
81    Jun 12 1603.5
82    May 12 1562.6
83    Apr 12 1663.4
84    Mar 12 1669.3
85    Feb 12 1709.9
86    Jan 12 1737.8
87    Dec 11 1565.8
88    Nov 11 1745.5
89    Oct 11 1724.2
90    Sep 11 1620.4
91    Aug 11 1828.5
92    Jul 11 1628.3
93    Jun 11 1502.3
94    May 11 1535.9
95    Apr 11 1556.0
96    Mar 11 1438.9
97    Feb 11 1409.3
98    Jan 11 1333.8
99    Dec 10 1421.1
100   Nov 10 1385.0
101   Oct 10 1357.1
102   Sep 10 1307.8
103   Aug 10 1248.3
104   Jul 10 1181.7
105   Jun 10 1245.5
106   May 10 1212.2
107   Apr 10 1180.1
108   Mar 10 1113.3
109   Feb 10 1118.3
110   Jan 10 1083.0
111   Dec 09 1095.2
112   Nov 09 1181.1
113   Oct 09 1039.7
114   Sep 09 1008.0
115   Aug 09  951.7
116   Jul 09  953.7
117   Jun 09  927.1
118   May 09  978.8
119   Apr 09  890.7
120   Mar 09  922.6
121   Feb 09  941.5
122   Jan 09  927.3
123   Dec 08  883.6
124   Nov 08  816.2
125   Oct 08  716.8
126   Sep 08  874.2
127   Aug 08  829.3
128   Jul 08  913.9
129   Jun 08  926.2
130   May 08  887.3
131   Apr 08  862.8
132   Mar 08  916.2
```

This dataset consist of 132 values with two columns:Date(monthly) and Price.The data consists of average price of gold every month from March 2008 to February 2019.Lets understand the code line by line.

Code:-Forecast.R

1. setwd("C:/Users/parek/Downloads")
2. mydata<-read.csv("Gold Futures Historical Data.csv")
3. mydata
4. library(MASS)

```
library(tseries)
library(forecast)
#These 3 libraries are used in my model.
col1 = 1;
col2 = 2;
data = mydata[c(col1, col2)];
#The original data consisted of many attributes, we just required the date and price.
data
data$Price <- as.numeric(gsub(",", "", data$Price))
#While converting the data to numeric form, we need to ignore the commas in 4 digit numbers to avoid any discrepancies, hence the gsub function is used and have replaced all commas with blank space.
data$Price
stock=log(data$Price[132:33])
#Taking 100 values as training dataset and will compare the remaining 32 values iwth the forecasted 32 values.
#Used log function for normalisation of data(scaling of data)
stock
plot(data)
pricearima=ts(stock,start=c(2008,12),frequency = 12)
#ts stands for tseries function in the tseries library.
fitstock=auto.arima(pricearima)
#the auto.arima function for obtaining p,d,q values to check if the data is suitable and stationary enough for forecasting.
fitstock
auto.arima(pricearima,ic='aic',trace=TRUE)
#all possible p,d,q values are displayed and best is choosen.
plot(pricearima,type='l')
```

29. title='Gold Price'
30. exp(stock)
31. **#exponenting the log function used to get the actual values**
32. myforecast=forecast(fitstock,h=33)
33. **#forecast fn used in forecast library and h=33 stands for 33 months.**
34. plot(myforecast)
35. myforecast
36. forecastedvalues=as.numeric(myforecast$mean)
37. finalforecastedvalues=exp(forecastedvalues)
38. **#to get the exponented forecasted values.**
39. finalforecastedvalues
40. dataerror=data.frame(data$Price[1:33],finalforecastedvalues)
41. col_headings=c("Actual Price","Forecasted Price")
42. names(dataerror)=col_headings
43. attach(dataerror)
44. dataerror
45. percentage_error=((dataerror$`Actual Price`-dataerror$`Forecasted Price`)/(dataerror$`Actual Price`))
46. percentage_error
47. mean(percentage_error)
48. **#Finding the percentage error.**

Output:-

Line 19:

> stock

[1] 6.820235 6.760183 6.788183 6.831090 6.817721 6.720582 6.773309 6.574797 6.704659 6.784004 6.832277 6.847474

[13] 6.827196 6.792008 6.886327 6.832061 6.860349 6.858250 6.915723 6.946687 7.074201 6.998692 6.987490 7.019565

[25] 7.015084 7.073354 7.100192 7.127292 7.074709 7.129538 7.176102 7.213105 7.233455 7.259186 7.195787 7.250848

[37] 7.271634 7.349874 7.336872 7.314753 7.395292 7.511251 7.390428 7.452518 7.464796 7.356152 7.460375 7.444190

[49] 7.420160 7.416619 7.354106 7.379944 7.384300 7.429283 7.479356 7.448625 7.444775 7.423449 7.414934 7.363723

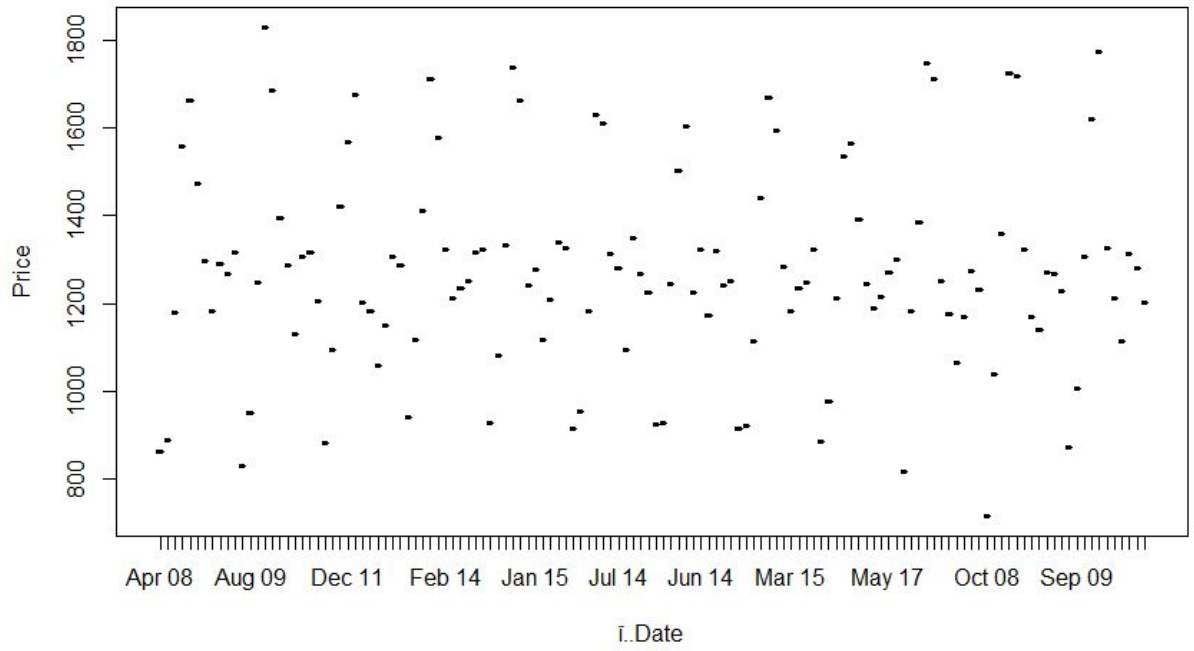
[61] 7.374504 7.294513 7.238928 7.109716 7.179613 7.241438 7.190299 7.188111 7.131379 7.091659 7.122947 7.186447

[73] 7.157268 7.166729 7.127373 7.186750 7.155630 7.159136 7.098789 7.065699 7.069194 7.076569 7.153443 7.100522

[85] 7.075893 7.075302 7.081204 7.066040 6.998418 7.031388 7.017058 7.040098 6.971481 6.966307 7.017865 7.117935

[97] 7.118178 7.161777 7.102335 7.184174

Line 20:plot(data)



Line 25:fitstock

Series: pricearima

ARIMA(1,1,0)

Coefficients:

ar1

-0.2242

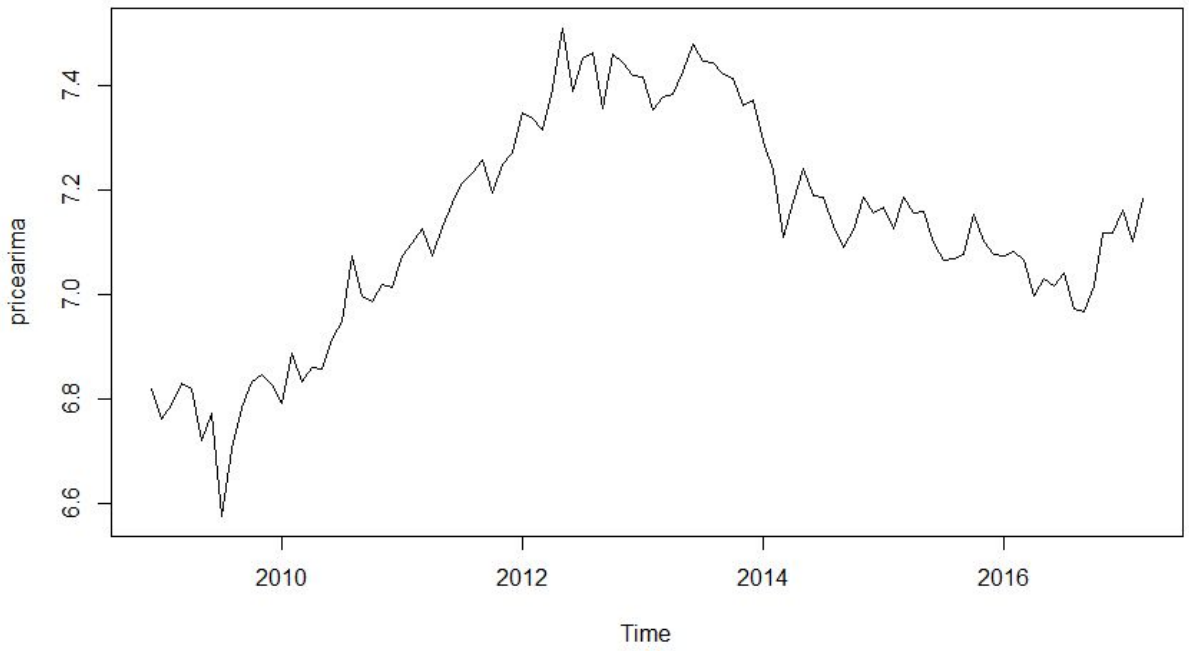
s.e. 0.0990

sigma^2 estimated as 0.003249: log likelihood=143.62

AIC=-283.24 AICc=-283.11 BIC=-278.05

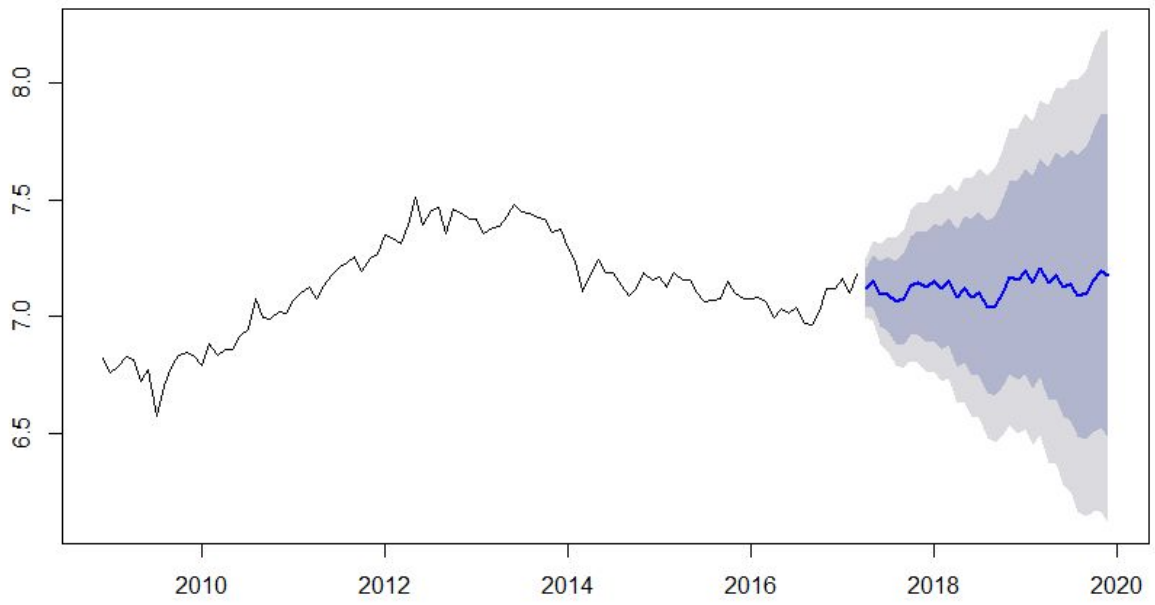
#The lower the aic value, the better accurate the forecasting will be, value of p,d and q are 1,1 and 0 respectively.

Line 28:plot(pricearima,type='l')



Line 34:plot(myforecast)

Forecasts from ARIMA[1,1,0]



Line 44:

> Actual Price Forecasted Price

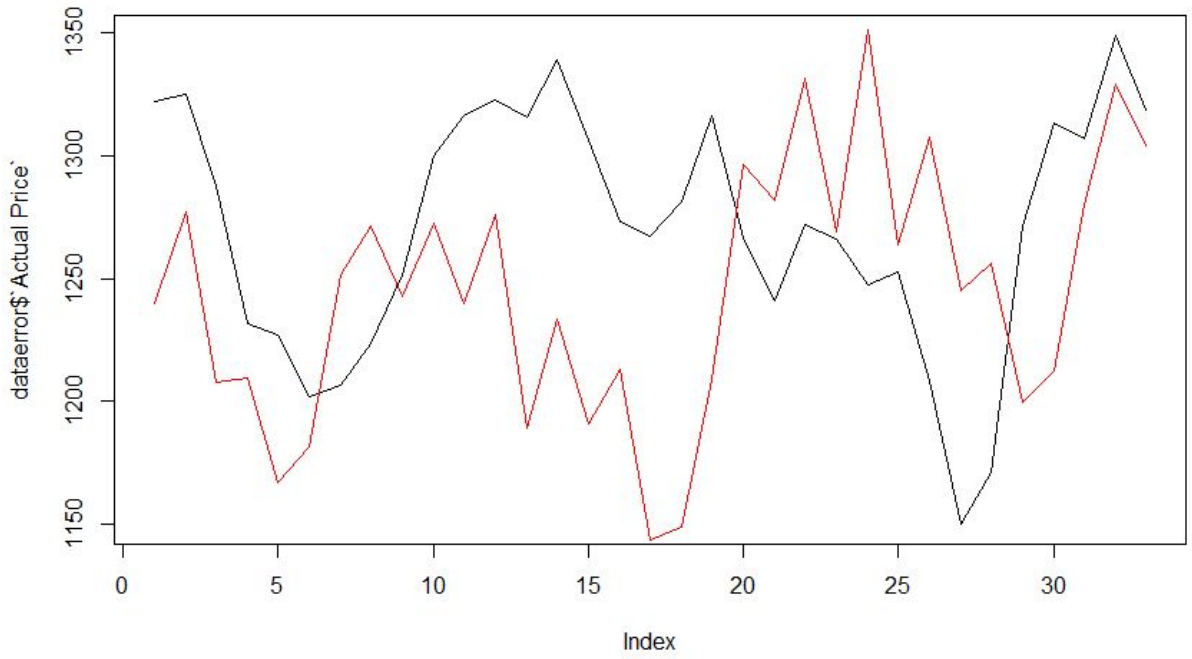
1	1322.1	1239.729
2	1325.2	1277.108
3	1287.7	1208.015
4	1231.8	1209.293
5	1226.8	1166.728
6	1201.9	1181.402
7	1206.7	1251.698
8	1223.7	1271.148
9	1251.3	1242.934
10	1300.1	1272.664
11	1316.2	1239.994
12	1322.8	1275.970
13	1315.5	1189.199
14	1339.0	1233.306
15	1306.3	1191.086
16	1273.2	1213.126
17	1267.0	1143.254
18	1281.5	1148.990
19	1316.2	1209.171
20	1266.6	1296.263
21	1240.7	1281.948
22	1272.0	1331.593

23	1266.1	1268.739
24	1247.3	1350.969
25	1252.6	1263.803
26	1208.6	1307.337
27	1150.0	1245.001
28	1170.8	1256.384
29	1271.5	1199.660
30	1313.3	1212.453
31	1306.9	1279.978
32	1349.0	1328.920
33	1318.4	1304.187

```
> percentage_error
 [1]  0.062302903  0.036290034  0.061881666  0.018271256  0.048966193  0.017054517
-0.037289791 -0.038774012  0.006685559
[10]  0.021102951  0.057898604  0.035401966  0.096010165  0.078934926  0.088198532
0.047183701  0.097668677  0.103402648
[19]  0.081316730 -0.023419070 -0.033245862 -0.046849597 -0.002084586 -0.083114684
-0.008943744 -0.081695577 -0.082609893
[28] -0.073098795  0.056500114  0.076789250  0.020599714  0.014885065  0.010780333
```

> mean(percentage_error)
[1] 0.009

```
> plot(dataerror$`Actual Price`, type="l")
> lines(dataerror$`Forecasted Price`, col="red")
```



Conclusion:-Hence, we used the ARIMA model to predict the future price of Gold using a training dataset of 100 values.We have achieved an error of 9% and this is because sample size was pretty less and to improve the accuracy we must consider a larger sample data so that there may not be any discrepancies in the future values.This entire model can be combined with shiny package in R to create a dashboard web application for multiple stocks and also for a better and a more efficient prediction.