George Mason University
SEOR Department
OR/SYST 568 Applied Predictive Analytics
Fall 2015

Group Project Report

“PREDICTING THE UNITED STATES OIL PRICE”

Guided By, **By,**

Professor KC Chang

Anirudh Myakala

Rohit Vungarala

Prashanth Arasu

Ravi teja Konda

CONTENTS

1.Introduction

1.1 Study Purpose

1.2 USO

1.3 Scope

2. Approach

2.1 Data

2.2 Data Preprocessing

2.3 Models

3. Model's Description of R-Code

4. Results

5. Conclusions

6. Future work

7. References

1.Introduction

1.1 Study Purpose

Fluctuations in global crude oil prices have always been in the focus of economic and financial news. The higher crude oil prices rise, the more positive is the economic outlook for petroleum exporters. In contrast, those countries dependent on petroleum imports suffer to varying degrees from those same higher prices as import bills increase. Estimates for the price per barrel for crude oil from leading financial and multilateral institutions are thus closely monitored by governments, investors, and consumers alike. So we have decided to predict the oil prices of USO and make summary of results.

1.2 USO

The United States Oil Fund (USO) is an exchange-traded fund that attempts to track the price of Intermediate crude oil. It is a domestic exchange traded security designed to track the movements of Crude Oil. The investment objective of USO is to predict the future changes in percentage terms of its units' net asset value (NAV) to reflect the changes in percentage terms of spot price of crude oil. We are going to predict the future price of a company for 1 day using historical records of the respective company.

1.3 Scope

Considering the historical data and making them stationary (not heavily fluctuated) and predicting the future stock prices using the Linear Regression, SVM, Logistic & ARIMA models and taking the required inputs to the parameters using the historical data of the particular asset. Using these models, we can predict the values up to 1 year but the results might not be accurate after a period of 2 weeks(max). So finally the predictions for a week were only considered.

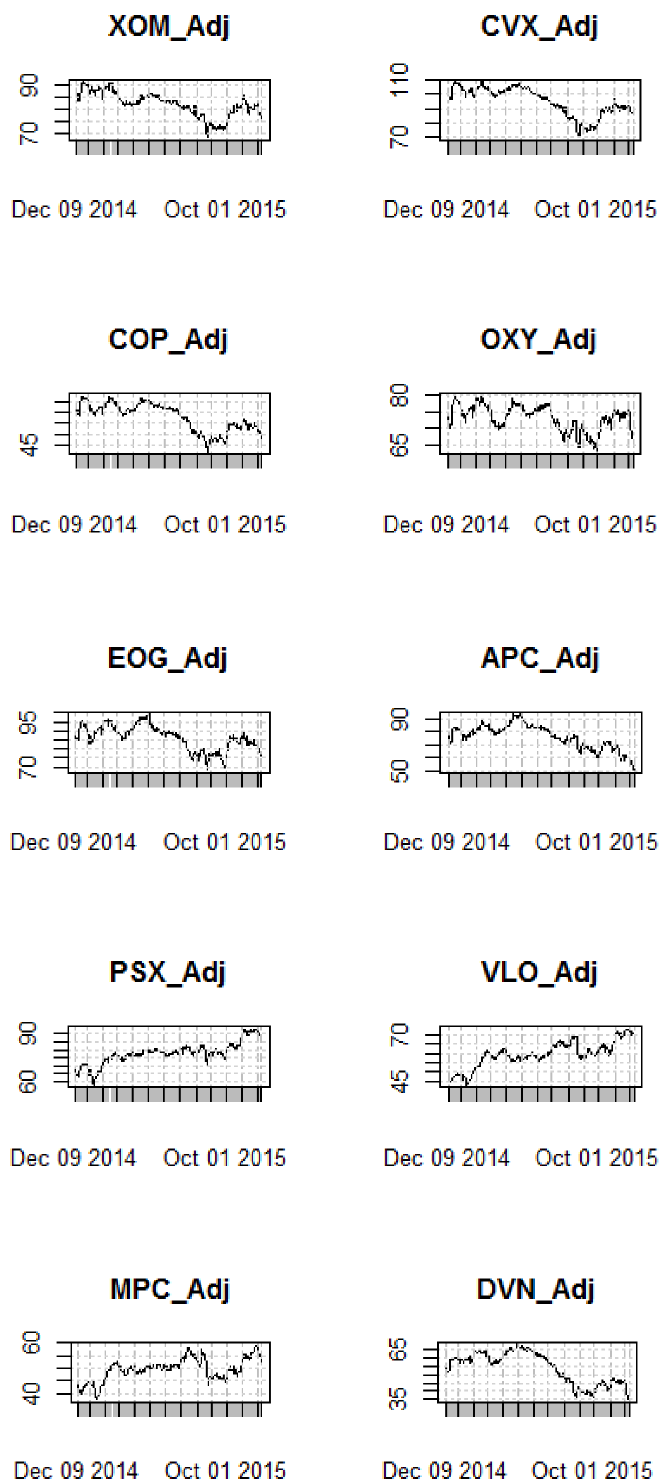
2. Approach

As discussed earlier in 1.3 we will be predicting the oil prices of the USO for the 1 week by considering the historical daily data of the individual assets for the past one year. The models used are Linear Regression and ARIMA.

2.1 Data

The stock prices of top 10 oil companies in US market are taken for the project. The historical data for 10 companies are taken into account for 1 whole year starting from Dec 2014 till Dec 2015. The following US top 10 oil companies' stocks were taken for a period of one year.

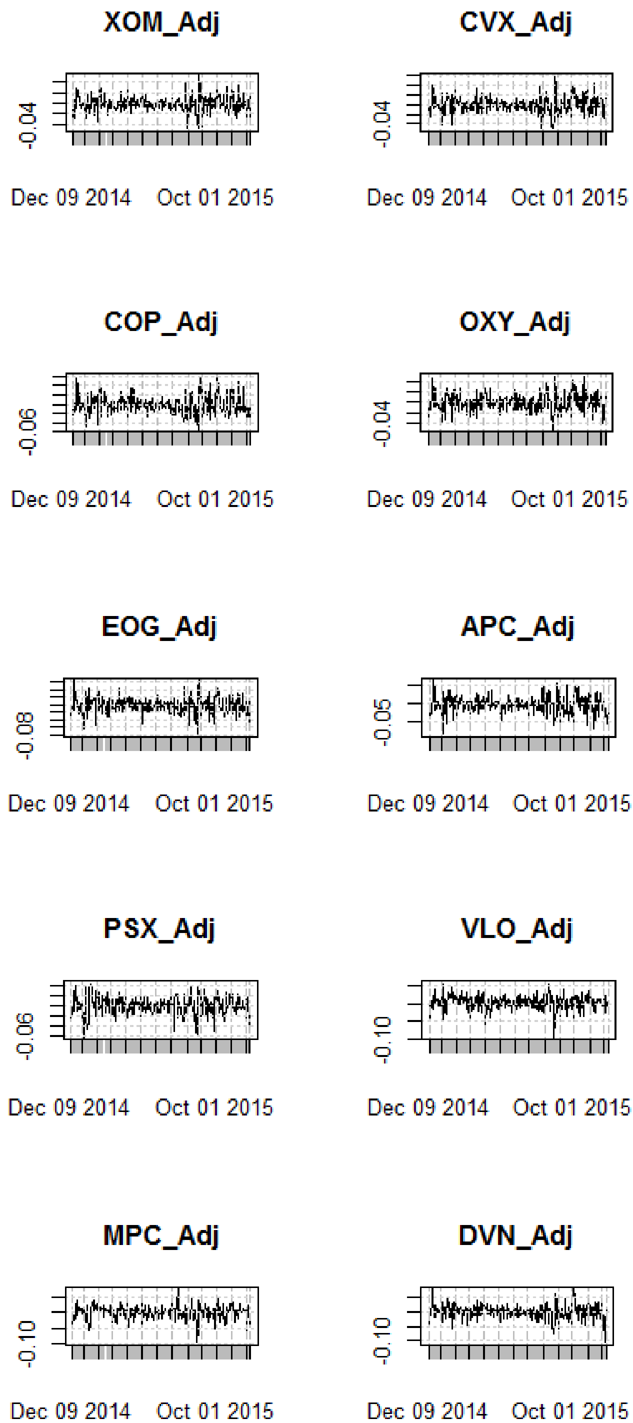
1.ExxonMobil (XOM)	2.Chevron (CVX)	3.ConocoPhillips (COP)	4.Occidental Petroleum (OXY)	5.Eog Resources (EOG)
6.AnadarkoPetrolem (APC)	7.Phillips66 (PSX)	8.Valero Energy (VLO)	9.Marathon Petroleum (MPC)	10.Devon Energy (DVN)



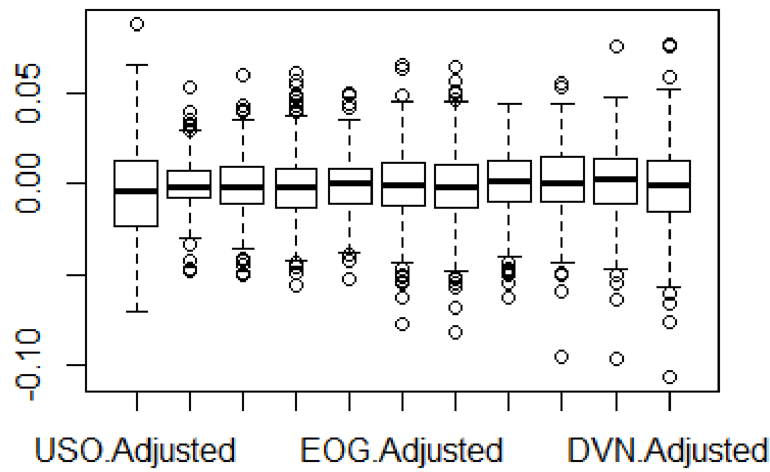
The adjusted close of the 10 Oil company's stocks from past one year

2.2 Data Preprocessing

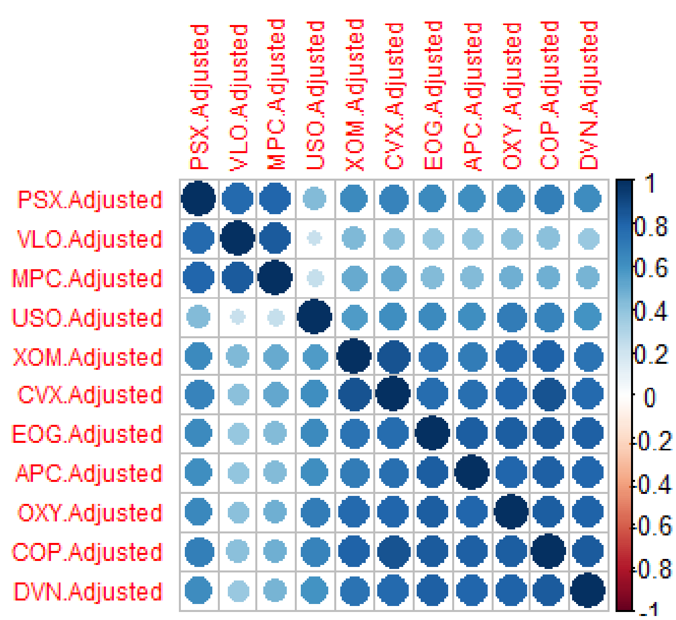
- To make the adjusted values stationary we use log difference so that the data will be stationary and the predictions can be done using the stationary data.



- Next to Check the distribution of the data the Shapiro Test for Normality was performed and found all p values to be below 0.05. Hence the Data is normally Distributed.
- Box- Plot of the data was made to check the mean and variance and outliers of the data.



- The results proved that there are no too many outliers and mean of all the variable are all most same.
- The next step was to find how correlated the data was. So we found out the correlations of the variables against Target variable.



The Corrplot of the 10 variables against the Target variable(USO)

- Data Partition was done using the Split function into training set and testing set.

  The division was 80% Training Data and 20% Test Data

CODE

```
data.set_Split <- sample(2, nrow(data.set), replace = TRUE, prob = c(0.8,0.2))

data.Train <- data.set[data.set_Split==1,]

data.Test <- data.set[data.set_Split==2,]
```

2.3 Models

The following four models were used for the predictions

- ✓ Linear Regression
- ✓ SVM
- ✓ Logistic (GLM)
- ✓ ARIMA

3. Description of R-Code the Models

- **Linear Regression Model**

CODE

```
set.seed(500)

lm <-
lm(USO.Adjusted~XOM.Adjusted+CVX.Adjusted+COP.Adjusted+OXY.Adjusted+EOG.Adjusted+A
PC.Adjusted+PSX.Adjusted+VLO.Adjusted+MPC.Adjusted+DVN.Adjusted,data.Train)

summary(lm)

predicted_Values <- predict(lm,data.Test)

predicted_Values
```

- The Lm function is used to fit the data and then predict function is applied to predict the future values trend
- Next the summary and the predicted Values are tabulated
- The ANOVA, AIC, BIC values are calculated
- Finally, all the pots are made for the model

Summary

```
Residuals:
     Min       1Q   Median       3Q      Max 
-0.05805 -0.01056 -0.00137  0.01129  0.05655 

Coefficients:
              Estimate Std. Error t value Pr(>|t|)    
(Intercept)  -0.001511   0.001354  -1.116    0.266    
XOM.Adjusted  0.001342   0.207621   0.006    0.995    
CVX.Adjusted  0.151199   0.201473   0.750    0.454    
COP.Adjusted  0.205119   0.196496   1.044    0.298    
OXY.Adjusted  0.746211   0.165936   4.497 1.22e-05 ***
EOG.Adjusted  0.201987   0.138866   1.455    0.148    
APC.Adjusted -0.024865   0.115633  -0.215    0.830    
PSX.Adjusted -0.057171   0.167550  -0.341    0.733    
VLO.Adjusted  0.129301   0.135067   0.957    0.340    
MPC.Adjusted -0.323440   0.131718  -2.456    0.015 *  
DVN.Adjusted  0.098719   0.116771   0.845    0.399    
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.01818 on 183 degrees of freedom
Multiple R-squared:  0.5598,	Adjusted R-squared:  0.5357 
F-statistic: 23.27 on 10 and 183 DF,  p-value: < 2.2e-16
```

Error = 0.01818, Adjusted R- Squared = 0.5357

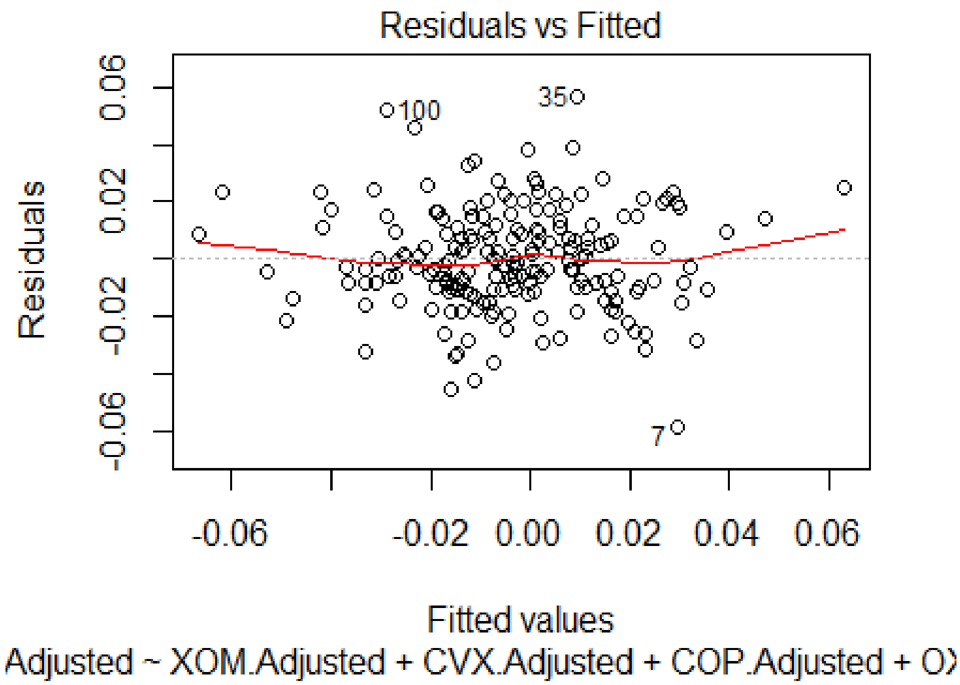
Predicted Values

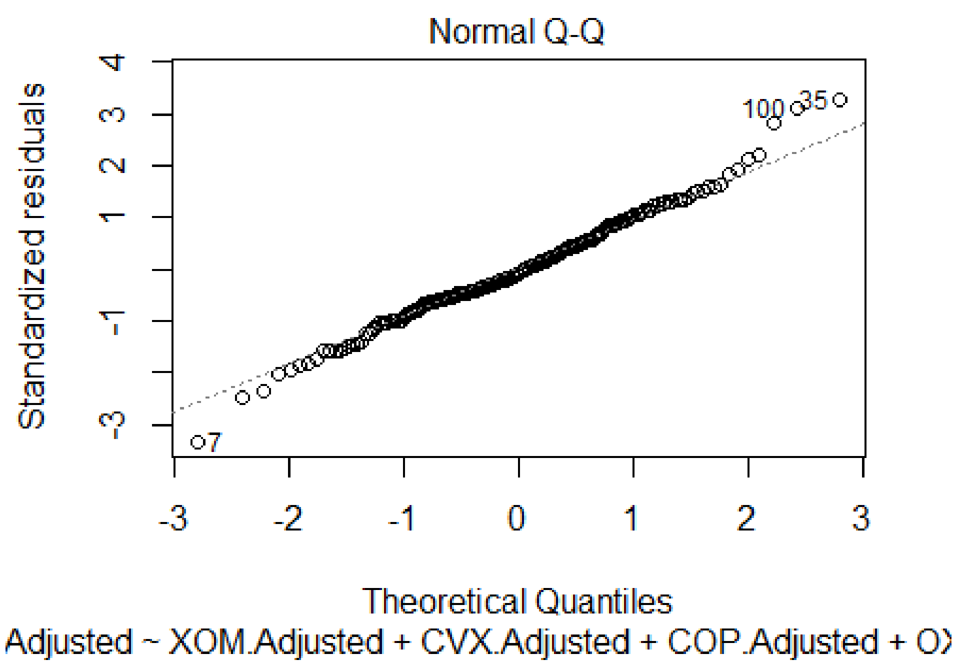
```
> predicted_values
            3             6             8            11            18            21            25 
-0.0175341453  0.0599831555  0.0259367617 -0.0168962693 -0.0210563359  0.0030998383 -0.0042954297 
           28            38            39            41            43            44            46 
 0.0193058734 -0.0236380835  0.0175416175 -0.0085765391 -0.0138046658  0.0277358840 -0.0098576292 
           53            58            62            65            69            80            88 
-0.0436500920 -0.0131300896  0.0048686196  0.0079746247  0.0103720041  0.0031583857 -0.0137004126 
           92           112           118           123           127           143           146 
 0.0071719095  0.0112588755  0.0018632974 -0.0137325880 -0.0143404601  0.0022848726 -0.0034388530 
          147           148           149           153           158           161           164 
-0.0259225586  0.0127859980 -0.0196662076 -0.0001148986  0.0413603430 -0.0349194034 -0.0121105757 
          165           166           167           171           175           179           181 
 0.0176385993 -0.0247107009  0.0354599066  0.0081547705 -0.0229517985  0.0264680099  0.0241384878 
          182           187           193           195           197           199           203 
 0.0237822530  0.0068174275  0.0413653844 -0.0346611331 -0.0191441088  0.0013882768  0.0332646006 
          205           208           217           225           227           228           230 
 0.0614294834  0.0056334896  0.0058920970  0.0013817941  0.0203821891 -0.0135665866 -0.0151257604 
          239           245 
-0.0182678003  0.0121840358 
```

ANOVA (Analysis of Variance Table)

```
Analysis of Variance Table

Response: USO.Adjusted
              Df   Sum Sq  Mean Sq  F value    Pr(>F)    
XOM.Adjusted   1 0.046854 0.046854 141.8149 < 2.2e-16 ***
CVX.Adjusted   1 0.006580 0.006580  19.9165 1.411e-05 ***
COP.Adjusted   1 0.006953 0.006953  21.0457 8.302e-06 ***
OXY.Adjusted   1 0.010892 0.010892  32.9668 3.837e-08 ***
EOG.Adjusted   1 0.001263 0.001263   3.8233   0.05207 .  
APC.Adjusted   1 0.000008 0.000008   0.0239   0.87742    
PSX.Adjusted   1 0.002101 0.002101   6.3580   0.01254 *  
VLO.Adjusted   1 0.000140 0.000140   0.4239   0.51581    
MPC.Adjusted   1 0.001848 0.001848   5.5925   0.01908 *  
DVN.Adjusted   1 0.000236 0.000236   0.7147   0.39899    
Residuals    183 0.060461 0.000330                       
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

Plots


Residuals vs Fitted
Residuals
Fitted values
Adjusted ~ XOM.Adjusted + CVX.Adjusted + COP.Adjusted + O)
100
35
7
0.06
0.02
-0.02
-0.06
-0.06
-0.02
0.00
0.02
0.04
0.06


Normal Q-Q
Standardized residuals
Theoretical Quantiles
Adjusted ~ XOM.Adjusted + CVX.Adjusted + COP.Adjusted + O)
100
35
7
4
3
2
1
-1
-3
-3
-2
-1
0
1
2
3

❖ **SVM**

CODE

set.seed(500)

svmFit <- train(USO.Adjusted ~ ., data = data.set, method = "svmRadial", preProc = c("center", "scale"), tuneLength = 10)

trControl = trainControl(method = "repeatedcv", repeats = 5)

svmFit

predict(svmFit)

plot(svmFit)

Summary

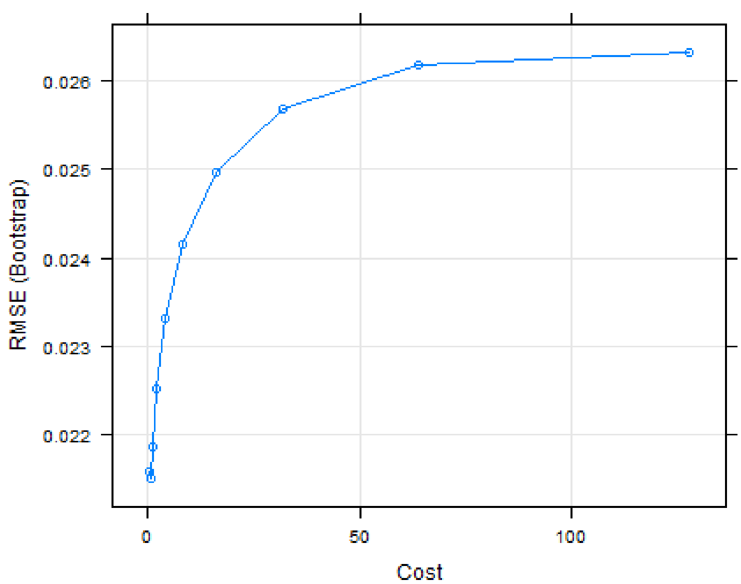
```
Support Vector Machines with Radial Basis Function Kernel

252 samples
 10 predictor

Pre-processing: centered (10), scaled (10)
Resampling: Bootstrapped (25 reps)
Summary of sample sizes: 252, 252, 252, 252, 252, 252, ...
Resampling results across tuning parameters:

  C       RMSE        Rsquared   RMSE SD      Rsquared SD
    0.25  0.02158266  0.3929266  0.001837875  0.07314575
    0.50  0.02150990  0.3837690  0.001772277  0.07017621
    1.00  0.02186705  0.3602334  0.001694570  0.06831423
    2.00  0.02252652  0.3271737  0.001578243  0.06500062
    4.00  0.02331672  0.2916866  0.001452176  0.05321063
    8.00  0.02415422  0.2630862  0.001613677  0.05426486
   16.00  0.02495993  0.2411763  0.001775473  0.05760972
   32.00  0.02567795  0.2222937  0.001985736  0.06048049
   64.00  0.02618674  0.2094688  0.001904732  0.05735402
  128.00  0.02631375  0.2065118  0.001899647  0.05711533

Tuning parameter 'sigma' was held constant at a value of 0.2020148
RMSE was used to select the optimal model using  the smallest value.
The final values used for the model were sigma = 0.2020148 and C = 0.5.
```

Therefore**, RMSE = 0.02158266   Rsquared = 0.3929266**

Predicted Values

```
> predict(svmFit)
  [1] -0.0360678692 -0.0002884381 -0.0246943361 -0.0229420199  0.0140219098  0.0045545330  0.0084334104
  [8]  0.0219322115 -0.0200769205  0.0262801409 -0.0212412861 -0.0035726734  0.0007811083 -0.0085707975
 [15] -0.0166753061  0.0009056176 -0.0297094363 -0.0269306037  0.0033992943  0.0124384254 -0.0077077437
 [22] -0.0384476459  0.0061489216  0.0110094920 -0.0239486214  0.0314691121 -0.0196011059  0.0191351679
 [29] -0.0125426885 -0.0211405484  0.0170416764 -0.0066037607 -0.0296683881  0.0084668094  0.0046632752
 [36]  0.0190020385  0.0213759394 -0.0290946085  0.0228398613 -0.0021404156  0.0009030922 -0.0093844366
 [43] -0.0152632553  0.0271204181  0.0180057583 -0.0079130444 -0.0244564593 -0.0223811572 -0.0120135136
 [50] -0.0103614892  0.0014922717  0.0106954358 -0.0311283063 -0.0101892933 -0.0115350314  0.0036295550
 [57]  0.0004636464 -0.0155593755 -0.0163850151 -0.0084134137 -0.0197397536  0.0126724993 -0.0093467763
 [64] -0.0199531251  0.0107815573 -0.0125246039  0.0280031756 -0.0267564066  0.0157653773 -0.0089947837
 [71] -0.0130656101  0.0221047468  0.0109659680 -0.0194582224  0.0138314981 -0.0255757549  0.0124362995
 [78] -0.0038289194  0.0167884505 -0.0008661523 -0.0195405569  0.0126591548  0.0091345606 -0.0136667175
 [85]  0.0198422883  0.0206807807 -0.0101570696 -0.0193526185  0.0093744271 -0.0216828239  0.0011366864
 [92]  0.0134137825 -0.0124042099 -0.0002676710  0.0067689805  0.0051833012 -0.0028397809 -0.0080020574
 [99] -0.0071721511 -0.0039545625  0.0061928592 -0.0300370691  0.0077737697 -0.0267618927  0.0046117004
[106] -0.0143075503 -0.0046182616  0.0057068431 -0.0012194100 -0.0291656287  0.0117260070  0.0164149299
[113] -0.0073114749 -0.0240683468  0.0001735620  0.0022283339  0.0011421365  0.0016909237  0.0129045345
[120] -0.0162373084 -0.0222099175  0.0148199169 -0.0132027003  0.0007801402  0.0183613071 -0.0099848697
[127] -0.0185407232 -0.0068112940  0.0158493898 -0.0039029955  0.0047157534 -0.0195334934  0.0096515649
[134]  0.0125406367 -0.0105891255 -0.0181840579  0.0103831745 -0.0264921355  0.0010902642 -0.0348548387
[141]  0.0008956776 -0.0353179240  0.0043795776 -0.0243187239  0.0163282797 -0.0023054835 -0.0117659298
[148]  0.0100742778 -0.0262360392 -0.0153210449 -0.0244560450 -0.0205048530  0.0056045839 -0.0200869475
[155] -0.0096197118 -0.0183075452 -0.0266738662  0.0153940895  0.0160717385 -0.0059667298 -0.0263652125
[162] -0.0199887074  0.0052225454 -0.0156000800  0.0231209994 -0.0225463689  0.0197158025 -0.0175643668
[169]  0.0030096021 -0.0218562929 -0.0044057061 -0.0044893137  0.0069248182 -0.0424415352 -0.0130346432
[176] -0.0087972950 -0.0257598150 -0.0070651898  0.0017498132  0.0092711043  0.0258357766  0.0196555731
[183] -0.0381945506  0.0142876489  0.0019731387 -0.0241834612 -0.0004643682 -0.0254860834  0.0100138228
[190] -0.0219741244 -0.0152258078  0.0151185747  0.0112604193  0.0038142712 -0.0369408661  0.0080915796
[197] -0.0199824727 -0.0250478940  0.0088757514  0.0055034539 -0.0240716234  0.0131759321  0.0309196917
[204] -0.0095208772  0.0102542238  0.0181424837  0.0169267248  0.0082949118  0.0270674175 -0.0046188899
[211] -0.0222438608 -0.0137782424  0.0053525314  0.0079931518  0.0005074698 -0.0262486509 -0.0040645927
[218] -0.0179004406  0.0184766234 -0.0122539898 -0.0255274009 -0.0163287871  0.0191613768 -0.0064793910
[225]  0.0095362465 -0.0013735498  0.0269896226 -0.0223066938 -0.0189547079 -0.0181115441 -0.0132877206
[232] -0.0019800152 -0.0210229680 -0.0329340478 -0.0013636586  0.0219828447 -0.0212214180  0.0117282059
[239] -0.0189813302 -0.0216328937  0.0068316239  0.0200913920 -0.0230171238 -0.0193182696 -0.0036372042
[246]  0.0185851278 -0.0342925949  0.0004730411 -0.0150887442 -0.0296570839 -0.0061041764 -0.0076437899
```

Plot



❖ Logistic (GLM)

CODE

```
set.seed(500)
logisticReg <- train(USO.Adjusted ~ ., data = data.set, method = "glm", trControl = trainControl(method = "repeatedcv", repeats = 5))
logisticReg
predict(logisticReg)
```

Summary

```
Generalized Linear Model

252 samples
 10 predictor

No pre-processing
Resampling: Cross-Validated (10 fold, repeated 5 times)
Summary of sample sizes: 226, 227, 226, 227, 227, 227, ...
Resampling results

  RMSE        Rsquared   RMSE SD      Rsquared SD
  0.01916584  0.4921758  0.002457392  0.1474752
```

RMSE = 0.01916584   Rsquared = 0.4921758

PREDICTIONS

```
> predict(logisticReg)
  [1] -2.589474e-02  1.442611e-03 -1.792926e-02 -1.662537e-02  2.147729e-02  5.378505e-02  3.073956e-02
  [8]  2.597979e-02 -1.810683e-02  2.503215e-02 -1.461124e-02 -7.967958e-03 -7.882721e-03 -6.988823e-03
 [15] -1.580753e-02 -5.024031e-03 -5.650321e-02 -2.475924e-02 -4.353465e-03  1.729675e-02 -2.069332e-03
 [22] -3.723578e-02  4.650540e-03  1.646583e-02 -5.793540e-03  3.248520e-02 -6.895851e-03  2.078821e-02
 [29] -9.852499e-03 -1.205866e-02  1.960658e-02 -1.281473e-02 -5.648279e-02  1.656711e-02  6.622856e-03
 [36]  1.975333e-02  2.814403e-02 -2.910154e-02  1.792340e-02 -1.066555e-02 -6.268283e-03 -4.648601e-03
 [43] -1.390054e-02  2.463431e-02  1.785094e-02 -1.131738e-02 -2.424107e-02 -1.515337e-02 -9.029138e-03
 [50] -5.729527e-03 -7.560625e-04  7.133862e-03 -3.701225e-02 -6.228847e-03 -5.400872e-03  2.784348e-03
 [57] -9.041663e-04 -1.463336e-02 -1.853164e-02 -9.000244e-03 -1.205433e-02  4.137808e-03  1.539075e-03
 [64] -2.014141e-02  5.827205e-03 -1.051578e-02  2.463807e-02 -2.955776e-02  1.379033e-02 -8.868105e-03
 [71] -1.094039e-02  1.478504e-02  6.051410e-03 -1.405003e-02  1.315891e-02 -2.483174e-02  1.669984e-02
 [78]  1.215586e-02  2.928474e-02  3.562047e-03 -1.142582e-02  2.004982e-02  3.698660e-04 -6.829257e-03
 [85]  2.508009e-02  2.696031e-02 -9.436508e-03 -1.589152e-02  1.551873e-03 -1.560561e-02 -6.433768e-03
 [92]  6.365723e-03 -1.286252e-02 -8.099019e-03  6.317230e-03  3.178638e-03  5.758300e-03 -1.577484e-02
 [99] -8.057250e-03 -2.041504e-02  7.549942e-03 -2.800336e-02 -1.458719e-03 -2.009536e-02  9.283663e-04
[106] -8.695093e-03 -5.248635e-03  8.399515e-03  2.331683e-04 -2.639913e-02  7.482543e-03  9.147852e-03
[113] -3.129496e-03 -9.604930e-03 -3.309266e-03  2.401723e-03 -1.612058e-03  3.438859e-03  7.709992e-03
[120] -1.383249e-02 -1.498727e-02  1.739325e-02 -1.309646e-02  1.299353e-03  1.346443e-02 -6.347420e-03
[127] -1.294277e-02 -8.240335e-03  9.262064e-03 -2.258255e-03 -3.007390e-03 -1.802225e-02  7.778785e-03
[134]  7.907261e-03 -4.760893e-03 -1.352219e-02  3.373886e-03 -2.528692e-02 -1.165586e-03 -3.388674e-02
[141] -4.448431e-03 -3.394212e-02  2.219244e-03 -2.533725e-02  1.139674e-02 -9.631111e-04 -2.182863e-02
[148]  1.080526e-02 -2.050257e-02 -1.338660e-02 -1.709646e-02 -2.128396e-02 -6.699519e-05 -1.587321e-02
[155] -4.541789e-03 -3.374482e-02 -1.748750e-02  3.678056e-02  1.274287e-02  6.079174e-03 -3.352079e-02
[162] -1.382797e-02  1.344365e-03 -1.045789e-02  1.620458e-02 -1.993333e-02  3.268741e-02 -7.780729e-03
[169]  2.297029e-02 -1.894455e-02  5.894815e-03 -5.876942e-03  2.575021e-03 -3.771157e-02 -1.991601e-02
[176] -1.478340e-02 -5.929639e-02 -2.984827e-03  1.794332e-02  6.064853e-02  2.499126e-02  3.101095e-02
[183] -4.204925e-02  9.932441e-03 -6.258598e-03 -2.231895e-02  9.512588e-03 -2.731196e-02  2.780152e-03
[190] -1.924010e-02 -1.139301e-02  8.898151e-03  3.940686e-02  2.094596e-03 -3.379347e-02 -8.752061e-04
[197] -1.641417e-02 -2.402674e-02 -1.615136e-03  2.583981e-03 -4.032616e-02  1.311656e-02  3.313271e-02
[204] -9.614107e-03  5.761462e-02  2.757171e-02  3.272199e-02  2.563831e-03  3.082114e-02 -1.842731e-03
[211] -1.627437e-02 -6.224675e-03  1.437238e-02  4.197784e-03 -1.311853e-02 -2.247828e-02  4.171173e-03
[218] -1.047768e-02  3.185300e-02 -1.396255e-02 -3.910599e-02 -3.027195e-02  4.840311e-02  1.270469e-03
[225] -1.889012e-03  1.942206e-02  1.888846e-02 -1.700075e-02 -1.015034e-02 -1.545072e-02 -2.899635e-03
[232]  6.343922e-03 -2.142551e-02 -4.300258e-02  6.837230e-03  3.545798e-02 -1.591219e-02  1.329499e-02
[239] -1.774775e-02 -2.156818e-02  1.362800e-03  2.864753e-02 -1.976878e-02 -1.475807e-02  6.041054e-03
[246]  1.170917e-02 -2.722809e-02 -2.086902e-02 -1.637222e-02 -3.886899e-02 -1.280249e-02  1.081198e-02
```

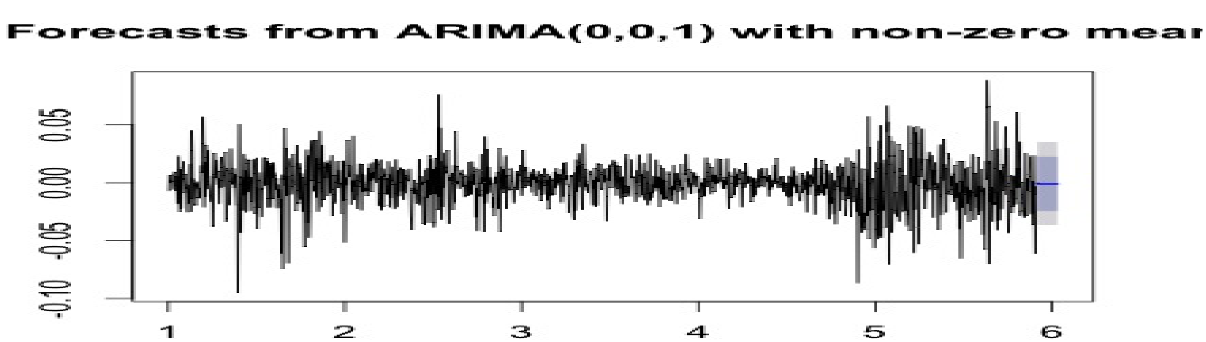
❖ ARIMA MODEL

Autogressive Intergrated Moving Average is a generalized version of autoregressive moving average. The model best fits for time series data to predict the future points in the series for forecasting. They are applied in some cases where data show evidence of non- stationary samples.

A time series Yt is said to be an ARIMA(p,d,q) process if ΔdYt is ARMA(p,q), where as ΔYt=Yt – Yt-1 is the differencing operator. If the log returns of an asset are Arima(p,q). Then the log price of the asset is ARIMA (p,1, q). The data is cleansed and NA are removed for a better value comparison. The package used is “quantmod” and “tseries”.

CODE

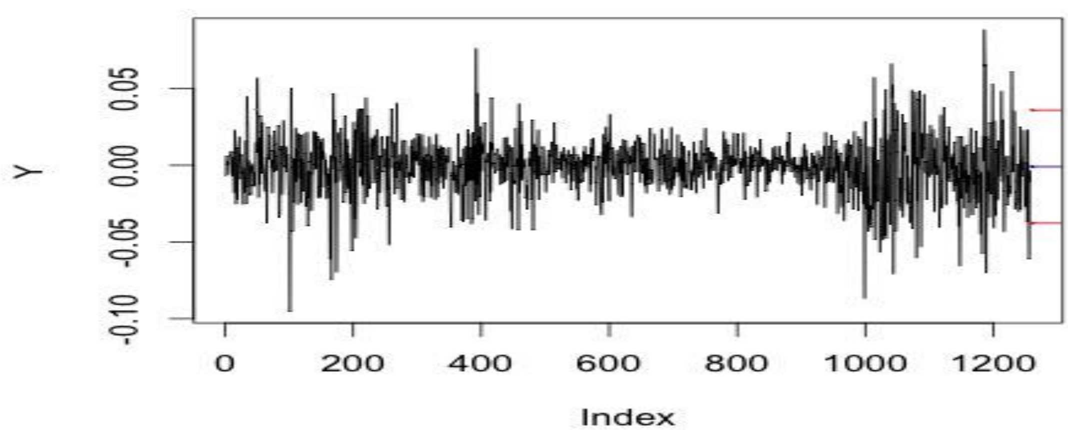
```
install.packages("tseries")
library(tseries)
library(forecast)
forecastArima <- function(x, n.ahead = 30) {
   myTs <- ts(oil$USO.Adjusted, start = 1, frequency = 256)
    fit.arima <- arima(myTs, order = c(0, 0, 1))
  fore <- forecast(fit.arima, h = n.ahead)
    plot(fore)
    upper <- fore$upper[, "95%"]
    lower <- fore$lower[, "95%"]
    trend <- as.numeric(fore$fitted)
    pred <- as.numeric(fore$mean)
    output <- data.frame(actual = c(oil$USO.Adjusted, rep(NA, n.ahead)),
trend = c(trend, rep(NA, n.ahead)), pred = c(rep(NA,
 nrow(oil)), pred), lower = c(rep(NA, nrow(oil)), lower),
                    upper = c(rep(NA, nrow(oil)), upper),
date = c(oil$Date, max(oil$Date) + (1:n.ahead)))
   return(output)
    }
forecastArima(oil)
```



Arima (1,0,1) Forecasting:

```
# ARIMA(1,0,1) forecasting
mydata.arima101 <- arima(Y, order = c(1,0,1))
mydata.pred1 <- predict(mydata.arima101, n.ahead=100)
plot (Y, type ="l")
lines(mydata.pred1$pred, col="blue")
lines(mydata.pred1$pred+2*mydata.pred1$se, col="red")
lines(mydata.pred1$pred-2*mydata.pred1$se, col="red")
predict(mydata.arima101,1)
```

Plot

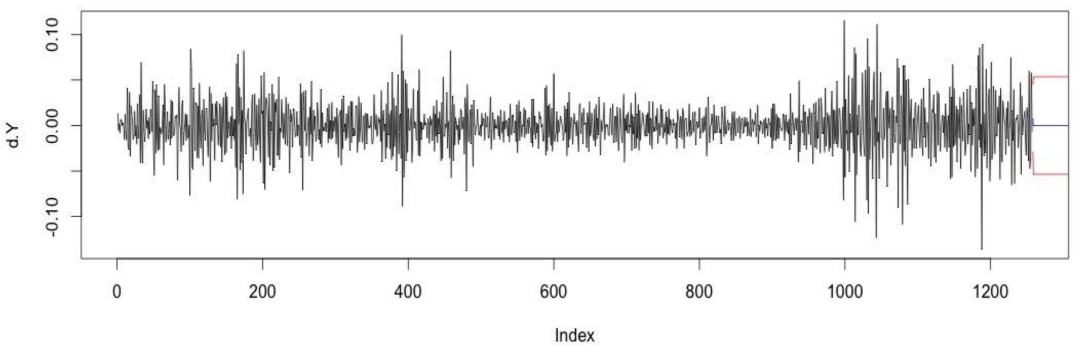



```
Call:
arima(x = Y, order = c(1, 0, 1))
Coefficients:
         ar1     ma1  intercept
     -0.6884  0.6299     -9e-04
s.e.   0.1749  0.1866      5e-04

sigma^2 estimated as 0.0003362:  log likelihood = 3245.54,  aic = -6483.08
>
Predicted value:
$pred
Time Series:
Start = 1259
End = 1259
Frequency = 1
[1] 0.01833645
```

ARIMA (1,1,1) Forecasting:

```
mydata.arima111 <- arima(d.Y, order = c(1,1,1))
mydata.pred1 <- predict(mydata.arima203, n.ahead=100)
plot (d.Y, type ="l")
lines(mydata.pred1$pred, col="blue")
lines(mydata.pred1$pred+2*mydata.pred1$se, col="red")
lines(mydata.pred1$pred-2*mydata.pred1$se, col="red")
predict(mydata.arima111,2)
View(mydata)
```

Plot




```
Call:
arima(x = d.Y, order = c(1, 1, 1))

Coefficients:
         ar1      ma1
     -0.5506  -1.0000
s.e.   0.0235   0.0021

sigma^2 estimated as 0.0004995:  log likelihood = 2987.59,  aic = -5969.18
```

Predicted value:

```
$pred
Time Series:
Start = 1258
End = 1259
Frequency = 1
[1]  0.003797393 -0.002097429
```

4. Results

VALUES→ MODELS	RMSE	Adjusted R-squared
LINEAR REGRESSION	**0.01825858**	**0.535758**
SVM	**0.02158266**	**0.3929266**
LOGISTIC REGRESSION	**0.01916584**	**0.4921759**

5. Conclusions

Thus, we predict the prices of the oil based on the historical data. Though Linear Regression model does not provide accurate results (No model provides accurate results) but it gives the values close to the real world stocks with least RMSE and high Adjusted Rsquared value.

6. FUTURE WORK

I would like to extend this work by considering various other factors that effect the oil prices and incorporate that into the code. I would also try to work with various other models like neural networks which is much advanced model and could predict values with more accuracy.

7.REFERENCES

Applied Predictive Modelling by Max Kuhn • Kjell Johnson
Statistics and Data Analysis For Financial Engineering by David Ruppert
Applied Predictive Analytics lectures by KC Chang