Stock Price Trend Forecasting

Gulay Samatli-Pac

Agenda

- Introduction
- Features
- Modeling
- Linear Regression
- Logistic Regression
- The LASSO
- Random Forest Model
- Gradient Boosting Model
- XGBOOST
- Appendix

Introduction

- The underlying problem is interesting and appealing since even a slight improvement could make a significant increase in profit for the stock holder.
- It is a difficult problem:
  - particularly due to the volatility in prices caused by their sensitivity to financial and economical noises.
  - The traditional statistical techniques including time series and regression models are difficult to apply for prediction due to this volatility.
- Machine learning algorithms like support vector machine, random forest, decision tree learning are trending to predict stock prices since they are more robust.

Features

- AT&T (T) Stock from 1/1/2010 to 4/25/2017
- Features (total 31 features):
  - **Close**
  - **Volume**
  - **n-day price change (PCh)**, n in {1, 10, 20, 270}
  - **Kaufman's efficiency ratio (EffR)**, n in {10, 20, 270}
  - **n-day moving average (MA)**, n in {10, 20, 270}
  - **n-day momentum,rate-of-change, (ROC),** n in {1, 10, 20, 270}
  - **Stock moving direction (PrDir)**,i.e 1 if closing price that day is higher than the day before, 0 otherwise
  - **Calendar date (Action)** eg. 2 for earning record date, 1 for ex-dividend date, 0 for regular date
  - **Weekday(Day),** day of the week
  - **Shifted Volume (Volumelagged),** n in {1, 2, 5, 10, 20, 270}

Modeling

- The part of the data including NA values- caused due to creating new variables- are removed.
- Data is scaled in R with scale(ATT, center = TRUE, scale = TRUE) except PrDir, Action and Day features.
- Scaled data divided into train(2/1/2011 to 4/22/2016) and test(4/25/2016 to 4/25/2017) subsets.
- LASSO, Logistic Regression, Random Forest, Gradient Boosting and XGBoost are applied to model the variable Price Direction (PrDir).
- Linear Regression is applied to model the variable Close.
- Any fitted value less than 0.5 is assumed to 1 for prediction.
- Models are fitted in R

Logistic Regression

Coefficients	Estimate	Std Error	Pr(>|z|)	
MA10	**19.2139**	**2.5301**	**3.10E-14**	***
ROC20	**-5.35745**	**2.29738**	**0.0197**	*
PCh1	**-0.78296**	**0.18159**	**1.62E-05**	***
PCh20	**-5.48976**	**2.31805**	**0.0179**	*
Closelag5	**-4.37411**	**0.88023**	**6.72E-07**	***
Closelag10	**-3.59514**	**0.80939**	**8.92E-06**	***

Signif. codes: 0 ‘***’ 0.001 ‘**’ 0.01 ‘*’

Response : Price Direction

MSE = 0.4033

Accuracy = 0.5967

LASSO (Least Absolute Shrinkage and Selection Operator)

- A regression analysis method performing both variable selection and regularization to improve the prediction accuracy and interpretability of the statistical model it produces.
- The lasso uses the shrinkage method.
  - Fitting a model containing all predictors using a technique that constrains or regularizes the coefficient estimates, or equivalently, that shrinks the coefficient estimates towards zero.
  - By doing this, standard estimators can be improved, e.g. mean squared error (MSE).
- It produces robust models when the number of features is ‘large’.
  - Large number of features usually causes overfitting (10 or more features) and causes computational challenges.

LASSO

(Intercept)	1.530451e+01	EffR10	1.982895e+00
Volume	-7.535862e-09	EffR20	.
Day	2.055589e-02	EffR270	-6.401497e+00
Action	-4.268921e-01	Closelag1	-4.552047e-01
MA10	2.558529e+00	Closelag2	-8.980179e-01
MA20	.	Closelag5	-6.824091e-01
MA270	-1.317049e-02	Closelag10	-5.376649e-01
ROC1	-1.463223e+01	Closelag20	.
ROC10	.	Closelag270	.
ROC20	.	Volumelag1	-1.184589e-09
ROC270	.	Volumelag2	1.468090e-08
PCh1	-1.223365e+01	Volumelag5	-1.518731e-09
PCh10	6.919621e+00	Volumelag10	6.305931e-09
PCh20	-1.830732e+00	Volumelag20	-2.544176e-09
PCh270	9.959585e-01	Volumelag270	5.709783e-09

MSE = 0.4033

Accuracy = 0.5967

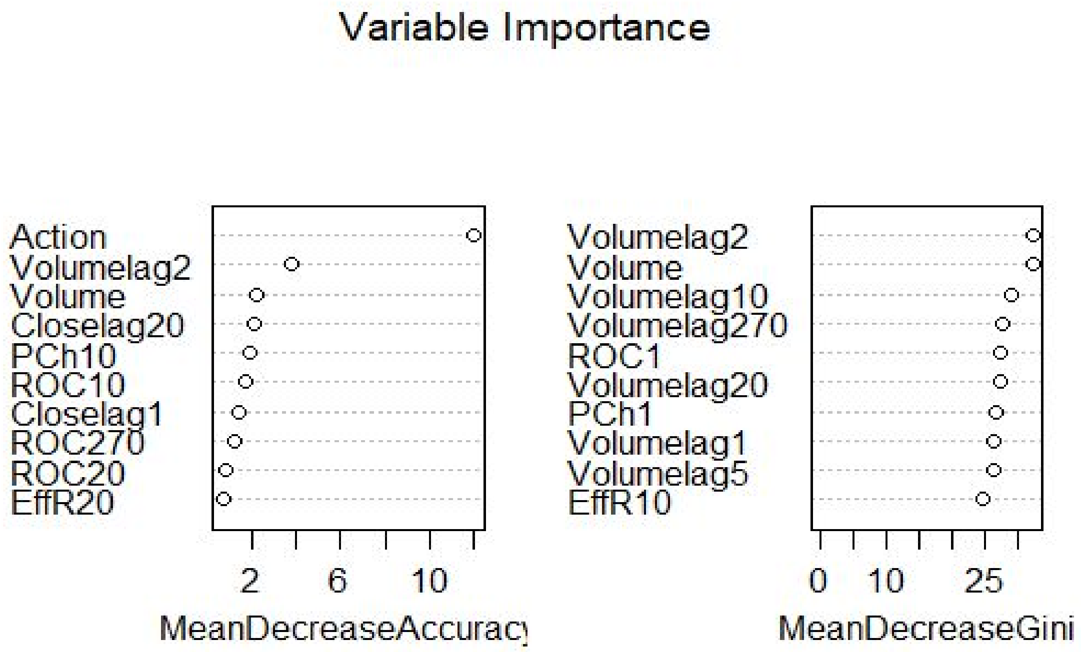
Random Forest

- A tree-based algorithm which involves building several decision trees using a random sample of m predictors from the full set of p predictors. Then it combines each tree output to improve the performance of the model.
- Advantages are as follows*:
    - It is robust to correlated predictors.
    - It is used to solve both regression and classification problems.
    - It can be also used to solve unsupervised ML problems.
    - It can handle thousands of input variables without variable selection.
    - It can be used as a feature selection tool using its variable importance plot.
    - It takes care of missing data internally in an effective manner.
- Disadvantages are as follows*:
    - The Random Forest model is difficult to interpret.
    - It tends to return erratic predictions for observations out of range of training data. For example, the training data contains two variable x and y. The range of x variable is 30 to 70. If the test data has x = 200, random forest would give an unreliable prediction.
    - It can take longer than expected time to computer a large number of trees.

(* : from https://www.hackerearth.com/practice/machine-learning/machine-learning-algorithms/tutorial-random-forest-parameter-tuning-r/tutorial/)

Random Forest

- Response : Price Direction



Random Forest

- Confusion Matrix and Statistics for number of tree = 2000 (ntree = 500)

	Reference	
Prediction	0	1
0	**107 (110)**	**105 (105)**
1	**10 (7)**	**21 (21)**

- **Accuracy = 0.5267** for ntree = 2000
- **Accuracy = 0.5391** for ntree = 500

Gradient Boosting Method

- Boosting sequentially fits multiple trees using the information from previously grown trees.
- In boosting, smaller trees (eg than random forest) usually sufficient since the growth of a particular tree takes into account the other trees that have already been grown.
- Response : Price Direction
- Confusion Matrix and Statistics for number of tree = 200, interaction depth =4

	Reference	
Prediction	0	1
0	**17**	**9**
1	**100**	**117**

- **Accuracy = 0.5514**

XGBOOST

- XGBoost is similar to the gradient boosting algorithms but mostly more efficient.
- It supports various objective functions, including regression, classification and ranking.
- Response : Price Direction
- Confusion Matrix and Statistics for nrounds = 100

	Reference	
Prediction	0	1
0	**77**	**71**
1	**40**	**55**

- **Accuracy = 0.5432**

Linear Regression

	Estimate	Std Error	t value	Pr(>|t|)	
Volume	**-6.70E-03**	**2.68E-03**	**-2.496**	**0.01268**	*****
Action	**-2.32E-02**	**8.74E-03**	**-2.655**	**0.00803**	******
MA10	**1.10E+00**	**9.81E-02**	**11.189**	**<2E-16**	*******
PCh1	**-5.19E-02**	**7.28E-03**	**-7.131**	**1.66E-12**	*******
EffR20	**1.04E-02**	**4.94E-03**	**2.11**	**0.03503**	*****
Closelag1	**9.17E-01**	**3.06E-01**	**2.998**	**0.00277**	******
Closelag2	**-6.56E-01**	**3.08E-01**	**-2.131**	**0.03329**	*****
Closelag5	**-2.25E-01**	**3.50E-02**	**-6.435**	**1.73E-10**	*******
Closelag10	**-1.91E-01**	**3.23E-02**	**-5.899**	**4.67E-09**	*******

Response variable : Close

Signif. codes: ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05

Conclusion

- Several models both statistical and machine learning are used to predict the stock price movement.
- The goal while building a model is to produce a higher prediction accuracy instead of exploring most exploratory features.
- Traditional statistical models like linear or logistic regression are easy to apply and interpret. However they are not so powerful for predictions, also sensitive to any volatility.
- Machine learning algorithms are powerful and robust for prediction but difficult to interpret the relation between response and features.
- For this data set, the XGBOOST model with proposed parameters yields the highest accuracy followed by Random Forest, LASSO and Gradient Boost.

Appendix

Logistic Regression

- fit.log1 <- glm(PrDir ~ ., family = binomial,control = list(maxit = 50), data= subset(train, select=-c(Close)))

```
> summary(fit.log1)

Call:
glm(formula = PrDir ~ ., family = binomial, data = subset(train,
    select = -c(Close)), control = list(maxit = 50))

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-2.4600  -1.1310   0.6796   1.0931   1.8795

Coefficients:
              Estimate Std. Error z value Pr(>|z|)
(Intercept)    0.02689    0.10738   0.250   0.8023
Volume        -0.09747    0.06758  -1.442   0.1493
Day            0.01359    0.04168   0.326   0.7444
Action        -0.43574    0.22469  -1.939   0.0525 .
MA10          19.21390    2.53010   7.594 3.10e-14 ***
MA20          -0.12082    1.60187  -0.075   0.9399
MA270         -0.04963    0.18618  -0.267   0.7898
ROC1          -0.67933    0.68384  -0.993   0.3205
ROC10          2.82766    2.22155   1.273   0.2031
ROC20         -5.35745    2.29738  -2.332   0.0197 *
ROC270         0.74691    0.81570   0.916   0.3598
PCh1          -0.78296    0.18159  -4.312 1.62e-05 ***
PCh10          3.11088    2.24138   1.388   0.1652
PCh20         -5.48976    2.31805  -2.368   0.0179 *
PCh270         0.53360    0.80133   0.666   0.5055
EffR10        -0.04792    0.10672  -0.449   0.6534
EffR20         0.22948    0.12736   1.802   0.0716 .
EffR270       -0.07462    0.11395  -0.655   0.5126
Closelag1     -0.55243    7.53131  -0.073   0.9415
Closelag2    -11.12615    7.60849  -1.462   0.1436
Closelag5     -4.37411    0.88023  -4.969 6.72e-07 ***
Closelag10    -3.59514    0.80939  -4.442 8.92e-06 ***
Closelag20     0.09821    0.73203   0.134   0.8933
Closelag270    0.29882    0.60576   0.493   0.6218
Volumelag1    -0.01917    0.07146  -0.268   0.7885
Volumelag2     0.17394    0.06980   2.492   0.0127 *
Volumelag5    -0.02432    0.05963  -0.408   0.6834
Volumelag10    0.10150    0.05979   1.698   0.0896 .
Volumelag20   -0.04033    0.05845  -0.690   0.4903
Volumelag270   0.06846    0.06032   1.135   0.2564
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 1821.2  on 1315  degrees of freedom
Residual deviance: 1710.9  on 1286  degrees of freedom
AIC: 1770.9

Number of Fisher Scoring iterations: 4
```

Logistic Regression

```
> fitted.log1 <- predict(fit.log1, newdata=subset(test,  select=-c(Close,PrDir)),type='response')

> fitted.log1 <- ifelse(fitted.log1 > 0.5,1,0)

> misClasificError <- mean(fitted.log1 != test$PrDir)

> print(paste('Accuracy for LogReg',1-misClasificError))
```

"Accuracy for LogReg 0.584362139917695"

> **MSE**

mean((fitted.log1 -test$PrDir)^2)

0.4156379

LASSO

- Lasso1=cv.glmnet(data.matrix(subset(ATT, select= - c(PrDir,Close))), ATT$PrDir, family='binomial', alpha=1, parallel=FALSE, standardize=TRUE, type.measure='auc',nfolds = 5)

```
> coef(Lasso1)
30 x 1 sparse Matrix of class "dgCMatrix"
                         1
(Intercept)   1.530451e+01
Volume       -7.535862e-09
Day           2.055589e-02
Action       -4.268921e-01
MA10          2.558529e+00
MA20          .
MA270        -1.317049e-02
ROC1         -1.463223e+01
ROC10         .
ROC20         .
ROC270        .
PCh1         -1.223365e+01
PCh10         6.919621e+00
PCh20        -1.830732e+00
PCh270        9.959585e-01
EffR10        1.982895e+00
EffR20        .
EffR270      -6.401497e+00
Closelag1    -4.552047e-01
Closelag2    -8.980179e-01
Closelag5    -6.824091e-01
Closelag10   -5.376649e-01
Closelag20    .
Closelag270   .
Volumelag1   -1.184589e-09
Volumelag2    1.468090e-08
Volumelag5   -1.518731e-09
Volumelag10   6.305931e-09
Volumelag20  -2.544176e-09
Volumelag270  5.709783e-09
```

LASSO

```
lasso.pred=predict(Lasso1 ,s=bestlam ,
newx=data.matrix(subset(ATT[1317:1559,], select= -
c(PrDir,Close) )),type='response')


fitted.lasso3 <- ifelse(lasso.pred3 > 0.5,1,0)
>MSE
mean((fitted.lasso3 -ATT[1317:1559,]$PrDir)^2)
>0.4567901



confusionMatrix(data=fitted.lasso,
+                reference=as.factor(test$PrDir))
Confusion Matrix and Statistics

          Reference
Prediction  0  1
         0 87 81
         1 30 45
```

```
  Accuracy : 0.5432
              95% CI : (0.4783, 0.607)
   No Information Rate : 0.5185
   P-Value [Acc > NIR] : 0.2402

                 Kappa : 0.0992
Mcnemar's Test P-Value : 2.077e-06

           Sensitivity : 0.7436
           Specificity : 0.3571
        Pos Pred Value : 0.5179
        Neg Pred Value : 0.6000
            Prevalence : 0.4815
        Detection Rate : 0.3580
  Detection Prevalence : 0.6914
     Balanced Accuracy : 0.5504

      'Positive' Class : 0
```

Random Forest (ntree = 2000)

```
> fit.RF1 <- randomForest(as.factor(PrDir) ~ . ,  data=subset(train,select= - Close), importance=TRUE, ntree=2000)
```

```
> var.imp.acc$Variables <- row.names(var.imp)
> var.imp.acc[order(var.imp.acc$MeanDecreaseAccuracy,decreasing = T),]
            MeanDecreaseAccuracy   Variables
Action              12.2344292      Action
Volumelag2           3.2476644  Volumelag2
EffR270              1.3959861     EffR270
ROC10                1.3455976       ROC10
Volume               1.1405526      Volume
PCh270               1.0168692      PCh270
ROC270               0.8563360      ROC270
PCh10                0.6316782       PCh10
Volumelag10          0.2440491 Volumelag10
Closelag20           0.1899665  Closelag20
Closelag1           -0.3093074   Closelag1
PCh20               -0.3741710       PCh20
EffR20              -0.6106158      EffR20
Closelag10          -0.8854782  Closelag10
Volumelag5          -1.3572260  Volumelag5
ROC20               -1.3968020       ROC20
Day                 -1.5182843         Day
MA10                -1.5893293        MA10
ROC1                -1.9424468        ROC1
Closelag2           -1.9748308   Closelag2
EffR10              -2.1411866      EffR10
Closelag270         -2.4875222 Closelag270
Closelag5           -2.6011779   Closelag5
Volumelag270        -3.0548864 Volumelag270
Volumelag20         -3.1341430 Volumelag20
PCh1                -3.1473682        PCh1
MA270               -3.8911259       MA270
MA20                -4.5365373        MA20
Volumelag1          -4.5831185  Volumelag1
```

Random Forest

```
> var.imp.gini$Variables <- row.names(var.imp)
> var.imp.gini[order(var.imp.gini$MeanDecreaseGini,decreasing =
T),]
             MeanDecreaseGini   Variables
Volumelag2         32.592224   Volumelag2
Volume             32.203402       Volume
Volumelag10        29.257658  Volumelag10
Volumelag270       27.684779 Volumelag270
ROC1               27.499856         ROC1
Volumelag20        27.413053  Volumelag20
Volumelag5         27.108157   Volumelag5
PCh1               26.339998         PCh1
Volumelag1         25.943577   Volumelag1
EffR10             24.459775       EffR10
EffR20             24.427091       EffR20
EffR270            23.166893      EffR270
PCh10              22.794556        PCh10
ROC10              22.762264        ROC10
Closelag270        22.458685  Closelag270
PCh270             22.301489       PCh270
ROC20              22.108730        ROC20
ROC270             22.095702       ROC270
MA270              22.074565        MA270
PCh20              22.032338        PCh20
Closelag20         21.925983   Closelag20
Closelag10         19.581243   Closelag10
Closelag1          19.433687    Closelag1
Closelag2          19.263077    Closelag2
Closelag5          18.798464    Closelag5
MA10               18.206973         MA10
MA20               17.509621         MA20
Day                10.481195          Day
Action              3.973274       Action
```

Random Forest - confusion matrix for test data

```
> confusionMatrix(data=test1$predicted.response,
+                 reference=test1$PrDir)
Confusion Matrix and Statistics

          Reference
Prediction   0   1
         0 107 105
         1  10  21
```

```
               Accuracy : 0.5267
                 95% CI : (0.4619, 0.5909)
    No Information Rate : 0.5185
    P-Value [Acc > NIR] : 0.424

                  Kappa : 0.0789
 Mcnemar's Test P-Value : <2e-16

            Sensitivity : 0.9145
            Specificity : 0.1667
         Pos Pred Value : 0.5047
         Neg Pred Value : 0.6774
             Prevalence : 0.4815
         Detection Rate : 0.4403
   Detection Prevalence : 0.8724
      Balanced Accuracy : 0.5406

       'Positive' Class : 0
```

Random Forest (ntree =500)

```
> var.imp.gini2[order(var.imp.gini2$MeanDecreaseGini,decreasing = T),]
```

	MeanDecreaseGini	Variables
Volumelag2	32.522905	Volumelag2
Volume	32.117955	Volume
Volumelag10	28.593796	Volumelag10
Volumelag270	28.107207	Volumelag270
ROC1	27.811562	ROC1
PCh1	27.413313	PCh1
Volumelag5	26.773434	Volumelag5
Volumelag20	26.666458	Volumelag20
Volumelag1	25.975683	Volumelag1
EffR10	25.302223	EffR10
EffR20	24.986024	EffR20
Closelag270	23.253635	Closelag270
PCh10	23.176206	PCh10
Closelag20	22.914681	Closelag20
EffR270	22.619160	EffR270
PCh20	22.483554	PCh20
ROC10	22.453218	ROC10
ROC270	22.229850	ROC270
PCh270	21.969093	PCh270
MA270	21.497584	MA270
ROC20	21.337090	ROC20
Closelag10	19.646474	Closelag10
Closelag1	19.231937	Closelag1
Closelag5	18.520719	Closelag5
Closelag2	18.430917	Closelag2
MA10	18.021254	MA10
MA20	17.657862	MA20
Day	10.047130	Day
Action	4.244718	Action

Random Forest (ntree =500)

```
> var.imp.acc2[order(var.imp.acc2$MeanDecreaseAccuracy,decreasing = T),]
            MeanDecreaseAccuracy    Variables
Action               6.62020318       Action
PCh10                2.19675698        PCh10
EffR10               1.93682786       EffR10
PCh270               1.87573275       PCh270
Volumelag2           1.60819580   Volumelag2
Closelag1            1.51859291    Closelag1
PCh20                0.90858326        PCh20
Volume               0.80390553       Volume
EffR20               0.76000522       EffR20
ROC270               0.73069498       ROC270
ROC20                0.60630525        ROC20
Closelag270          0.33934371  Closelag270
EffR270              0.33466283      EffR270
Day                  0.31187319          Day
Closelag20           0.27078840   Closelag20
Closelag10           0.01786889   Closelag10
ROC1                -0.01160016         ROC1
Closelag5           -0.46628376    Closelag5
Volumelag5          -0.62988688   Volumelag5
Closelag2           -0.63590281    Closelag2
MA10                -0.74609532         MA10
ROC10               -0.77720032        ROC10
Volumelag10         -1.03991167  Volumelag10
Volumelag270        -1.45973823 Volumelag270
Volumelag20         -1.68442445  Volumelag20
MA270               -1.82379313        MA270
PCh1                -1.91856303         PCh1
MA20                -3.25165584         MA20
Volumelag1          -3.84147394   Volumelag1
```

Random Forest - confusion matrix for test data

```
>confusionMatrix(data=test2$predicted.response,
+                              reference=test2$PrDir)
Confusion Matrix and Statistics

          Reference
Prediction   0   1
         0 110 105
         1   7  21
```

```
         Accuracy : 0.5391
           95% CI : (0.4742, 0.603)
No Information Rate : 0.5185
P-Value [Acc > NIR] : 0.2819

            Kappa : 0.1037
Mcnemar's Test P-Value : <2e-16

      Sensitivity : 0.9402
      Specificity : 0.1667
   Pos Pred Value : 0.5116
   Neg Pred Value : 0.7500
       Prevalence : 0.4815
   Detection Rate : 0.4527
Detection Prevalence : 0.8848
Balanced Accuracy : 0.5534

  'Positive' Class : 0
```

Gradient Boosting Method

```
fit.gbm1 <- gbm(PrDir ~ . ,
                data=subset(train,select= - Close),
                distribution="bernoulli",n.trees=200,
interaction.depth=4)
summary(fit.gbm1)
> summary(fit.gbm1)
                var           rel.inf
Volumelag2      Volumelag2    17.1579974
Volume          Volume        12.4125175
Volumelag10     Volumelag10    8.8226057
PCh1            PCh1           6.2801673
Action          Action         4.2204048
ROC1            ROC1           3.6741411
PCh20           PCh20          3.1384650
Closelag270     Closelag270    3.0802335
ROC270          ROC270         3.0699644
Closelag1       Closelag1      3.0242737
```

```
Closelag20      Closelag20     2.9523788
ROC10           ROC10          2.8811493
Volumelag1      Volumelag1     2.8384964
PCh10           PCh10          2.7675568
EffR10          EffR10         2.6818478
Volumelag270    Volumelag270   2.5437861
EffR20          EffR20         2.4452589
ROC20           ROC20          2.1837189
Volumelag20     Volumelag20    2.1329304
EffR270         EffR270        2.0420501
Volumelag5      Volumelag5     1.8831400
PCh270          PCh270         1.4679236
Closelag10      Closelag10     1.4101564
Closelag2       Closelag2      1.3709911
MA270           MA270          1.2971989
MA20            MA20           0.7853227
Closelag5       Closelag5      0.7539956
Day             Day            0.3815692
MA10            MA10           0.2997585
```

Gradient Boosting Method

```
> confusionMatrix(fitted.gbm1,test$PrDir)
Confusion Matrix and Statistics

          Reference
Prediction   0   1
         0  17   9
         1 100 117
```

```
               Accuracy : 0.5514
                 95% CI : (0.4865, 0.6151)
    No Information Rate : 0.5185
    P-Value [Acc > NIR] : 0.1678

                  Kappa : 0.076
 Mcnemar's Test P-Value : <2e-16

            Sensitivity : 0.14530
            Specificity : 0.92857
         Pos Pred Value : 0.65385
         Neg Pred Value : 0.53917
             Prevalence : 0.48148
         Detection Rate : 0.06996
   Detection Prevalence : 0.10700
      Balanced Accuracy : 0.53694

       'Positive' Class : 0
```

XGBOOST

```
> fit.xgboost1<-xgboost(data = data.matrix(subset(train,select= -
c(Close,PrDir))),  label = train$PrDir, nrounds = 100,
objective = "binary:logistic", eval_metric = "error", verbose = 1)

>predict.xgboost1 <- predict(fit.xgboost1,
data.matrix(subset(test,select= - Close)))

>fitted.xgboost1 <- ifelse(predict.xgboost1 > 0.5,1,0)

> confusionMatrix(fitted.xgboost1,test$PrDir)
Confusion Matrix and Statistics

          Reference
Prediction  0  1
         0 72 67
         1 45 59
```

```
               Accuracy : 0.5391
                 95% CI : (0.4742, 0.603)
    No Information Rate : 0.5185
    P-Value [Acc > NIR] : 0.28194

                  Kappa : 0.0831
 Mcnemar's Test P-Value : 0.04722

            Sensitivity : 0.6154
            Specificity : 0.4683
         Pos Pred Value : 0.5180
         Neg Pred Value : 0.5673
             Prevalence : 0.4815
         Detection Rate : 0.2963
   Detection Prevalence : 0.5720
      Balanced Accuracy : 0.5418

       'Positive' Class : 0
```

XGBOOST (nrounds =2000)

```
> confusionMatrix(fitted.xgboost2,test$PrDir)
Confusion Matrix and Statistics

          Reference
Prediction  0  1
         0 77 71
         1 40 55
```

```
      Accuracy : 0.5432
              95% CI : (0.4783, 0.607)
    No Information Rate : 0.5185
    P-Value [Acc > NIR] : 0.240214

                 Kappa : 0.0937
 Mcnemar's Test P-Value : 0.004407

            Sensitivity : 0.6581
            Specificity : 0.4365
         Pos Pred Value : 0.5203
         Neg Pred Value : 0.5789
             Prevalence : 0.4815
         Detection Rate : 0.3169
   Detection Prevalence : 0.6091
      Balanced Accuracy : 0.5473

       'Positive' Class : 0
```

Linear Regression

```
Call:
lm(formula = Close ~ ., data = subset(train, select = -c(PrDir)))

Residuals:
     Min       1Q   Median       3Q      Max
-0.44914 -0.05057  0.00169  0.05353  0.38322

Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept)  6.946e-04  4.415e-03   0.157  0.87501
Volume      -6.699e-03  2.684e-03  -2.496  0.01268 *
Day         -1.198e-03  1.713e-03  -0.699  0.48458
Action      -2.320e-02  8.740e-03  -2.655  0.00803 **
MA10         1.098e+00  9.810e-02  11.189  < 2e-16 ***
MA20         1.694e-02  6.553e-02   0.259  0.79603
MA270       -6.014e-03  7.650e-03  -0.786  0.43196
ROC1        -3.883e-02  2.765e-02  -1.404  0.16049
ROC10        9.068e-02  8.908e-02   1.018  0.30887
ROC20       -1.696e-01  9.100e-02  -1.864  0.06252 .
ROC270       1.038e-02  3.363e-02   0.309  0.75764
PCh1        -5.193e-02  7.282e-03  -7.131 1.66e-12 ***
PCh10        1.005e-01  8.988e-02   1.118  0.26357
PCh20       -1.744e-01  9.188e-02  -1.898  0.05790 .
PCh270      -1.941e-02  3.287e-02  -0.591  0.55494
EffR10       9.072e-04  4.120e-03   0.220  0.82576
EffR20       1.043e-02  4.944e-03   2.110  0.03503 *
EffR270      1.229e-04  4.657e-03   0.026  0.97894
Closelag1    9.168e-01  3.058e-01   2.998  0.00277 **
Closelag2   -6.562e-01  3.080e-01  -2.131  0.03329 *
Closelag5   -2.252e-01  3.499e-02  -6.435 1.73e-10 ***
Closelag10  -1.906e-01  3.231e-02  -5.899 4.67e-09 ***
Closelag20   9.975e-04  2.991e-02   0.033  0.97340
Closelag270  4.216e-02  2.475e-02   1.704  0.08866 .
Volumelag1  -1.833e-03  2.911e-03  -0.630  0.52907
Volumelag2   7.393e-03  2.705e-03   2.733  0.00635 **
Volumelag5  -5.129e-05  2.452e-03  -0.021  0.98331
Volumelag10  4.019e-03  2.393e-03   1.680  0.09321 .
Volumelag20 -1.967e-03  2.389e-03  -0.823  0.41041
Volumelag270 2.224e-04  2.455e-03   0.091  0.92782
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.08566 on 1286 degrees of freedom
Multiple R-squared:  0.9869,  Adjusted R-squared:  0.9866
F-statistic:  3329 on 29 and 1286 DF,  p-value: < 2.2e-16
```