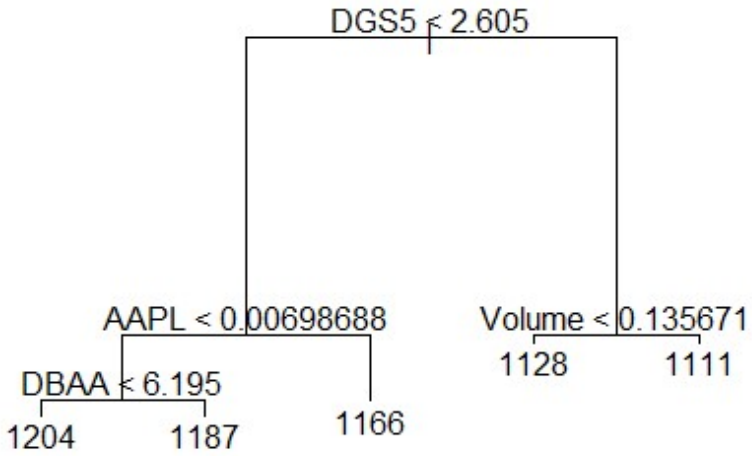
Decision Tree

```
#install.packages('tree')
library(tree)

## Warning: package 'tree' was built under R version 4.1.2

library(dplyr)

##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union

set.seed(1)
tree.train_set = tree(Close30~.,data=train_set, subset = folds )
summary(tree.train_set)

##
## Regression tree:
## tree(formula = Close30 ~ ., data = train_set, subset = folds)
## Variables actually used in tree construction:
## [1] "DGS5"   "AAPL"   "DBAA"   "Volume"
## Number of terminal nodes:  5
## Residual mean deviance:  13.76 = 20390 / 1482
## Distribution of residuals:
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
## -6.4920 -2.1830 -0.3522  0.0000  2.3370  7.3780

plot(tree.train_set)
text(tree.train_set,pretty=0)
```

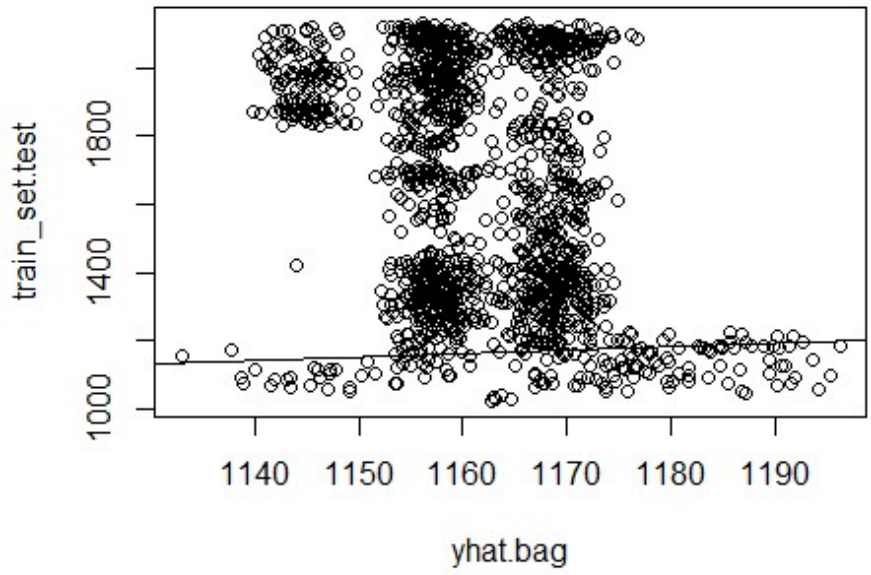



```
#cv.train_set = cv.tree(tree.train_set)
```

use Bagging and Random Forests to view variance explained

```
set.seed(1)
names(folds)

## NULL

library(randomForest)

## Warning: package 'randomForest' was built under R version 4.1.2

## randomForest 4.6-14

## Type rfNews() to see new features/changes/bug fixes.

##
## Attaching package: 'randomForest'

## The following object is masked from 'package:dplyr':
##
##     combine

#install.packages('randomForest')
train_set.test = train_set[-folds ,'Close30']
bag.train_set=randomForest(Close30~.,data=train_set, subset = folds,mtr
y=13,importance =TRUE)
bag.train_set
```

```
##
## Call:
##  randomForest(formula = Close30 ~ ., data = train_set, mtry = 13,
   importance = TRUE, subset = folds)
##                Type of random forest: regression
##                      Number of trees: 500
## No. of variables tried at each split: 13
##
##           Mean of squared residuals: 1.473552e-26
##                     % Var explained: 100

yhat.bag = predict(bag.train_set ,newdata=train_set[-folds ,])
plot(yhat.bag, train_set.test)
abline(0,1)
```

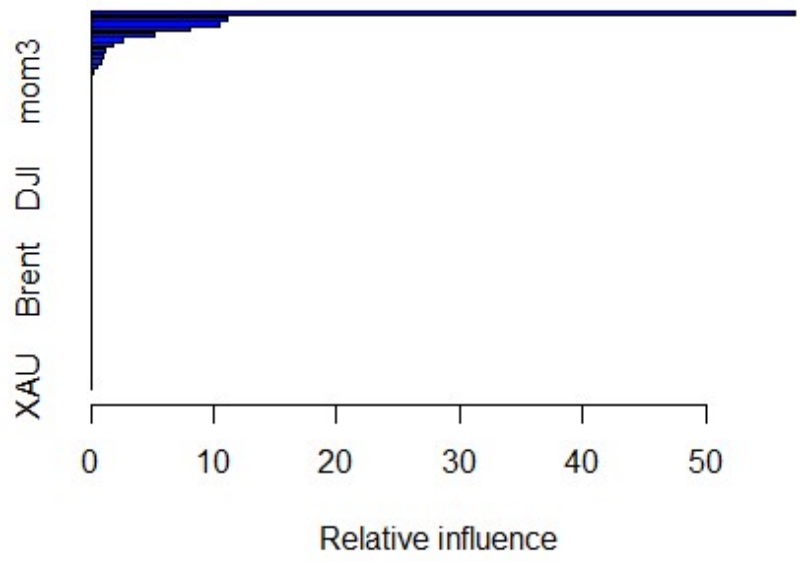



```
mean((yhat.bag - train_set.test)^2)

## [1] 331740.2
```

Boosting

```
#install.packages("gbm")
library(gbm)

## Warning: package 'gbm' was built under R version 4.1.2

## Loaded gbm 2.1.8
```

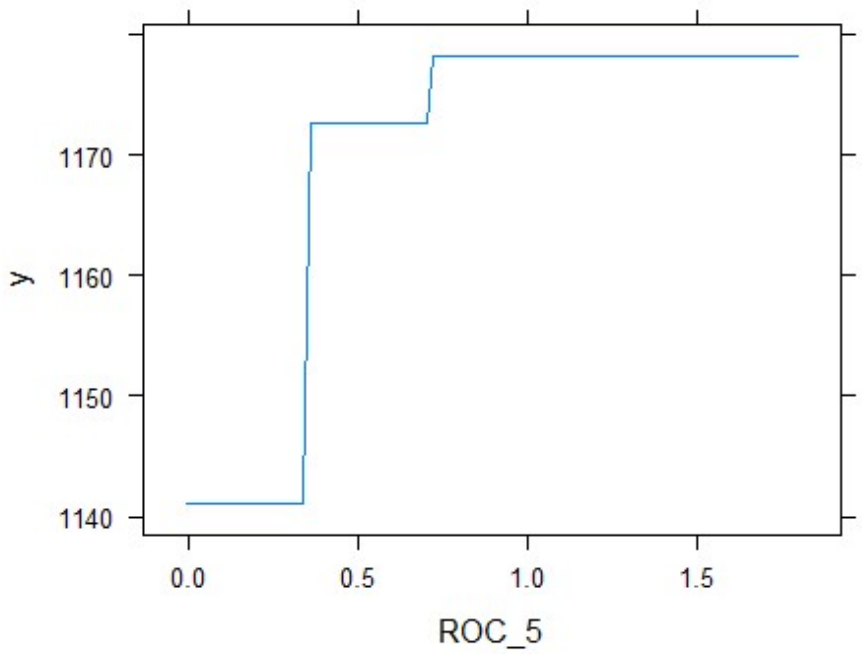
```
set.seed(1)
boost.train_set=gbm(Close30~.,data=train_set[folds,],distribution="gaus
sian",n.trees=5000, interaction.depth=4)
summary(boost.train_set)
```



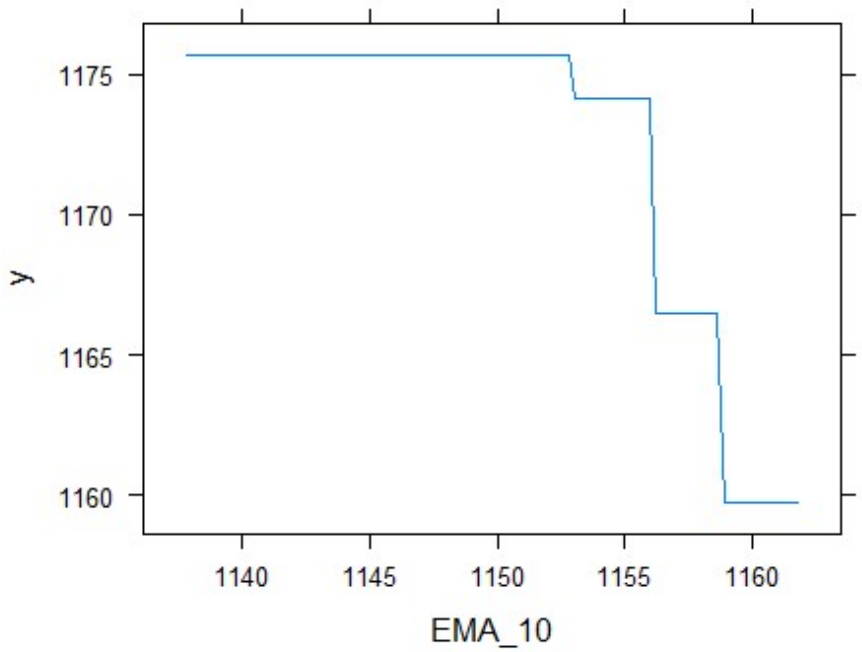

```
##                              var      rel.inf
## ROC_5                      ROC_5 5.731782e+01
## EMA_10                    EMA_10 1.115479e+01
## mom2                       mom2 1.038430e+01
## Volume                   Volume 8.029332e+00
## DGS5                       DGS5 5.085879e+00
## DE4                         DE4 2.678286e+00
## ROC_15                   ROC_15 1.777865e+00
## Close                     Close 1.172030e+00
## DBAA                       DBAA 9.372094e-01
## DTB3                       DTB3 7.809201e-01
## mom                         mom 4.813041e-01
## Nikkei.F               Nikkei.F 2.002663e-01
## mom1                       mom1 1.238899e-11
## mom3                       mom3 0.000000e+00
## ROC_10                   ROC_10 0.000000e+00
## ROC_20                   ROC_20 0.000000e+00
## EMA_20                   EMA_20 0.000000e+00
## EMA_50                   EMA_50 0.000000e+00
## DTB4WK                   DTB4WK 0.000000e+00
## DTB6                       DTB6 0.000000e+00
## DGS10                     DGS10 0.000000e+00
```

```
## Oil                       Oil 0.000000e+00
## Gold                     Gold 0.000000e+00
## DAAA                     DAAA 0.000000e+00
## AAPL                     AAPL 0.000000e+00
## AMZN                     AMZN 0.000000e+00
## GE                         GE 0.000000e+00
## JNJ                       JNJ 0.000000e+00
## JPM                       JPM 0.000000e+00
## MSFT                     MSFT 0.000000e+00
## WFC                       WFC 0.000000e+00
## XOM                       XOM 0.000000e+00
## FCHI                     FCHI 0.000000e+00
## DJI                       DJI 0.000000e+00
## IXIC                     IXIC 0.000000e+00
## RUT                       RUT 0.000000e+00
## NYSE                     NYSE 0.000000e+00
## TE1                       TE1 0.000000e+00
## TE2                       TE2 0.000000e+00
## TE3                       TE3 0.000000e+00
## TE5                       TE5 0.000000e+00
## TE6                       TE6 0.000000e+00
## DE1                       DE1 0.000000e+00
## DE2                       DE2 0.000000e+00
## DE5                       DE5 0.000000e+00
## DE6                       DE6 0.000000e+00
## CTB3M                   CTB3M 0.000000e+00
## CTB6M                   CTB6M 0.000000e+00
## CTB1Y                   CTB1Y 0.000000e+00
## AUD                       AUD 0.000000e+00
## Brent                   Brent 0.000000e+00
## CAC.F                   CAC.F 0.000000e+00
## copper.F             copper.F 0.000000e+00
## WIT.oil               WIT.oil 0.000000e+00
## DAX.F                   DAX.F 0.000000e+00
## DJI.F                   DJI.F 0.000000e+00
## EUR                       EUR 0.000000e+00
## FTSE.F                 FTSE.F 0.000000e+00
## gold.F                 gold.F 0.000000e+00
## NASDAQ.F             NASDAQ.F 0.000000e+00
## GAS.F                   GAS.F 0.000000e+00
## NZD                       NZD 0.000000e+00
## silver.F             silver.F 0.000000e+00
## RUSSELL.F           RUSSELL.F 0.000000e+00
## S.P.F                   S.P.F 0.000000e+00
## CHF                       CHF 0.000000e+00
## Dollar.index.F Dollar.index.F 0.000000e+00
## Dollar.index     Dollar.index 0.000000e+00
## wheat.F               wheat.F 0.000000e+00
## XAG                       XAG 0.000000e+00
## XAU                       XAU 0.000000e+00
```

```
par(mfrow=c(1,2))
plot(boost.train_set ,i="ROC_5")
```




```
plot(boost.train_set ,i="EMA_10")
```




```
yhat.boost=predict(boost.train_set,newdata=train_set[-folds,], n.trees=
5000)
mean((yhat.boost -train_set.test)^2)
```

```
## [1] 337231.9
```

```
set.seed(1)
boost.train_set=gbm(Close30~.,data=train_set[folds,],distribution="gaus
sian",n.trees=5000,interaction.depth=4,shrinkage =0.2, verbose =F)
yhat.boost=predict(boost.train_set,newdata=train_set[-folds,], n.trees=
5000)
mean((yhat.boost -train_set.test)^2)
```

```
## [1] 330866.8
```