R Notebook

```r
packages.used=c("gbm", "caret","DMwR" ,"nnet","randomForest","e1071")

# check packages that need to be installed.
packages.needed=setdiff(packages.used,
                        intersect(installed.packages()[,1],
                                  packages.used))
# install additional packages
if(length(packages.needed)>0){
  install.packages(packages.needed, dependencies = TRUE)
}
```

Load and Process Data

```r
setwd("~/Spr2017-proj5-grp15/data/observation/")
short_after_selected <- read.csv("short_after20.csv")
long_after_selected <- read.csv("long_before20.csv")

colnames(long_after_selected)[1] <- "y"
long_after_selected$y <- ifelse(long_after_selected$RETURN >0, 1, 0)
long_after_selected <- long_after_selected[,c(-2,-12)]

colnames(short_after_selected)[1] <- "y"
short_after_selected$y <- ifelse(short_after_selected$RETURN >0, 1,0)
short_after_selected <- short_after_selected[,c(-2,-12)]

test.index <- sample(1:1500,300,replace = F)

test.sas <- short_after_selected[test.index,]
test.lbs <- long_after_selected[test.index,]
test.sas.x <- test.sas[,-1]
test.lbs.x <- test.lbs[,-1]
train.sas <- short_after_selected[-test.index,]
train.lbs <- long_after_selected[-test.index,]
```

```r
source("../lib/evaluation_measures.R")
source("../lib/train.R")
```

```
## Loading required package: survival
```

```
## Loading required package: lattice
```

```
## Loading required package: splines
```

```
## Loading required package: parallel
```

```
## Loaded gbm 2.1.3
```

```
## Loading required package: ggplot2
```

```
##
## Attaching package: 'caret'
## The following object is masked from 'package:survival':
##
##     cluster
```

```
## Loading required package: grid
```

```
## randomForest 4.6-12
```

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

```
##
## Attaching package: 'randomForest'
```

```
## The following object is masked from 'package:ggplot2':
##
##     margin
```

```
source("../lib/test.R")
source("../lib/cross_validation.R")
```

Short After Model

GBM

```
# GBM
start.time <- Sys.time()
res_gbm = train.gbm(train.sas)
pred.gbm = test.gbm(res_gbm,test.sas.x)
sas.gbm.sum = table(pred.gbm,test.sas$y)
end.time <- Sys.time()
gbm.sas.time <- end.time-start.time
performance_statistics(sas.gbm.sum)
```

```
## $precision
## [1] 0.6972112
##
## $recall
## [1] 0.9259259
##
## $f1
## [1] 0.7954545
##
## $accuracy
## [1] 0.7
```

SVM

```
# model.svm <- svm(y ~ ., data = train.sas, cost = 256, gamma = 0.3)
# Tune svm
start.time <- Sys.time()
model.svm.sas <- train.svm(train.sas)
pre.svm <- test.svm(model.svm.sas,test.sas.x)
```

```
svm.sas <- table(pre.svm,test.sas$y)
end.time <- Sys.time()
svm.sas.time <- end.time-start.time
performance_statistics(svm.sas)
```

```
## $precision
## [1] 0.744186
##
## $recall
## [1] 0.8465608
##
## $f1
## [1] 0.7920792
##
## $accuracy
## [1] 0.72
```

BPNN

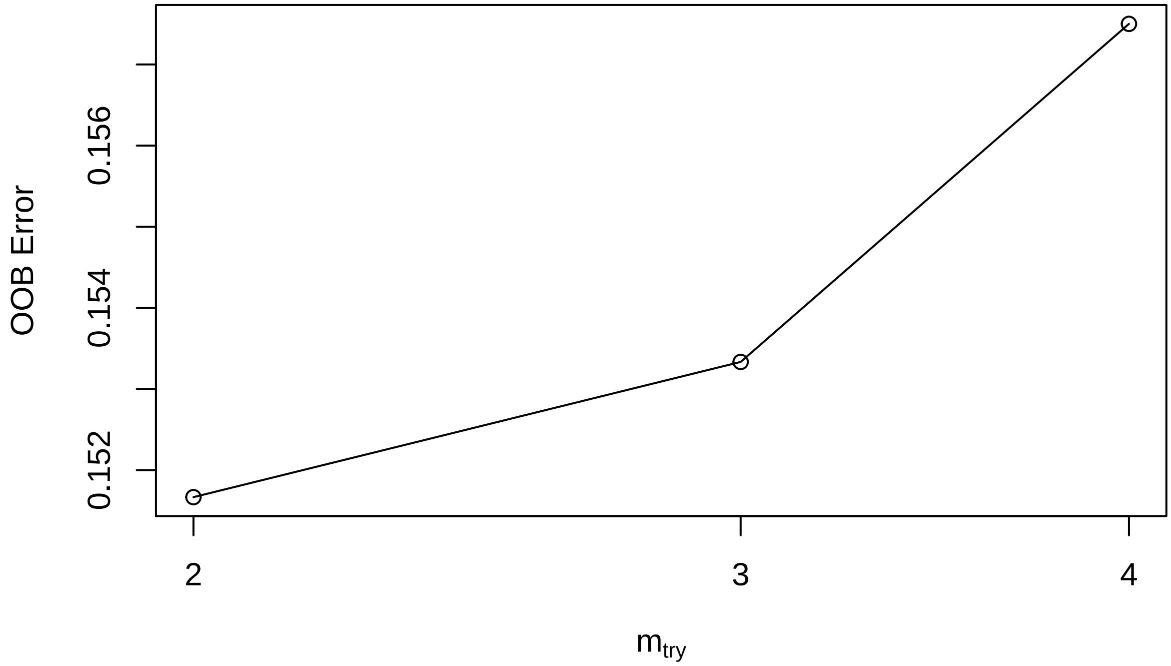
```
# netural network
start.time <- Sys.time()
# model.nnet <- nnet(y ~ ., data = train.sas, linout = F, size = 10, decay = 0.001, maxit = 200, trace =
# Tune bpnn
model.nnet <- train.bp(train.sas)
pre.nnet <- test.bp(model.nnet,test.sas.x)
nnet.sas <- table(pre.nnet,test.sas$y)
end.time <- Sys.time()
nnet.sas.time <- end.time-start.time
performance_statistics(nnet.sas)
```

```
## $precision
## [1] 0.7183099
##
## $recall
## [1] 0.8095238
##
## $f1
## [1] 0.761194
##
## $accuracy
## [1] 0.68
```

Random Forest

```
# Random Forest
start.time <- Sys.time()
model.rf <- train.rf(train.sas)
```

```
## mtry = 3  OOB error = 15.33%
## Searching left ...
## mtry = 2     OOB error = 15.17%
## 0.01086957 1e-05
## Searching right ...
## mtry = 4     OOB error = 15.75%
## -0.03846154 1e-05
```



```
pre.rf <- test.rf(model.rf,test.sas.x)
rf.sas <- table(pre.rf,test.sas$y)
performance_statistics(rf.sas)
```

```
## $precision
## [1] 0.8622449
##
## $recall
## [1] 0.8941799
##
## $f1
## [1] 0.8779221
##
## $accuracy
## [1] 0.8433333
```

```
end.time <- Sys.time()
rf.sas.time <- end.time-start.time
```

Logistic

```
start.time <- Sys.time()
res_logi = train.log(train.sas)
pred.logi = test.log(res_logi,test.sas.x)
log.sas <- table(pred.logi,test.sas$y)
end.time <- Sys.time()
```

```
log.sas.time <- end.time-start.time
performance_statistics(log.sas)
```

```
## $precision
## [1] 0.6342282
##
## $recall
## [1] 1
##
## $f1
## [1] 0.7761807
##
## $accuracy
## [1] 0.6366667
```

Majority Vote(Equal Weight)

```
# Majority Vote
pre=(as.numeric(as.character(pre.svm))+as.numeric(as.character(pred.gbm))+as.numeric(as.character(pre.r
pre=ifelse(pre>=2,1,0)
table(pre,test.sas$y)
```

```
##
## pre   0   1
##   0 176  54
##   1  13  57
```

Long - Before Model

SVM

```
# model.svm <- svm(y ~ ., data = train.lbs, cost = 256, gamma = 0.3)
# Tune svm
start.time <- Sys.time()
model.svm.lbs <- train.svm2(train.lbs)
pre.svm <- test.svm(model.svm.lbs,test.lbs.x)
svm.lbs <- table(pre.svm,test.lbs$y)
end.time <- Sys.time()
svm.time <- end.time-start.time
performance_statistics(svm.lbs)
```

```
## $precision
## [1] 0.6329114
##
## $recall
## [1] 0.7194245
##
## $f1
## [1] 0.6734007
##
## $accuracy
## [1] 0.6766667
```

BPNN

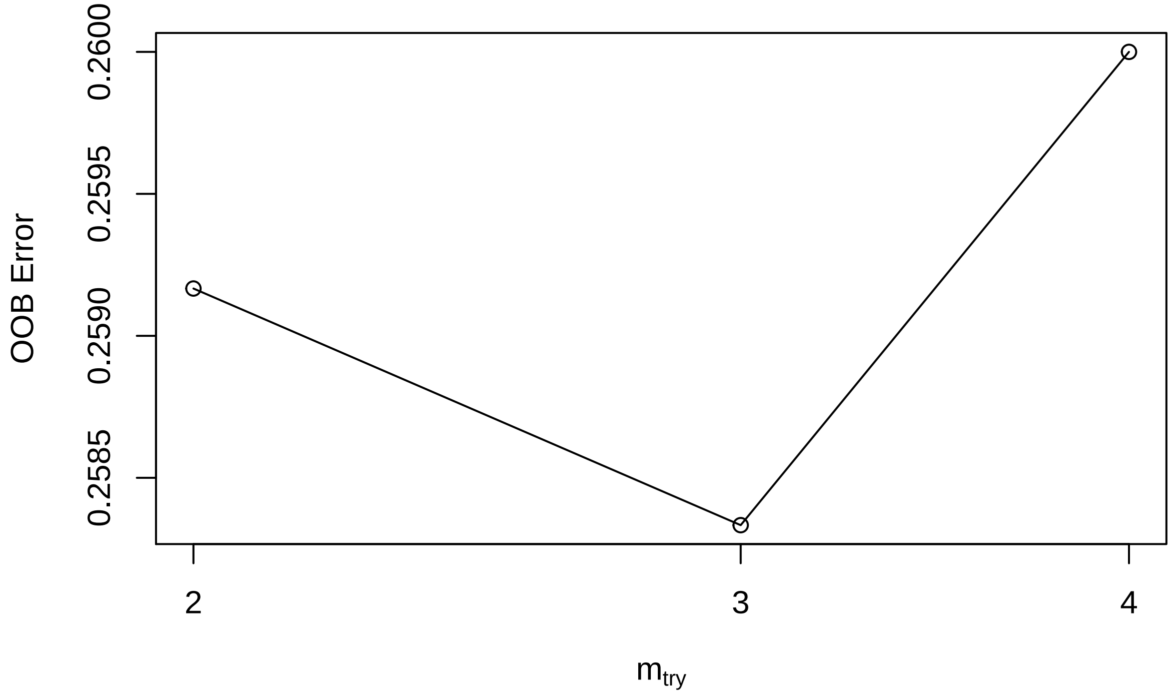
```
# netural network
start.time <- Sys.time()
# model.nnet <- nnet(y ~ ., data = train.sas, linout = F, size = 10, decay = 0.001, maxit = 200, trace
# Tune bpnn
model.nnet <- train.bp(train.lbs)
pre.nnet <- test.bp(model.nnet,test.lbs.x)
nnet.lbs <- table(pre.nnet,test.lbs$y)
end.time <- Sys.time()
nnet.lbs.time <- end.time-start.time
performance_statistics(nnet.lbs)
```

```
## $precision
## [1] 0.5280899
##
## $recall
## [1] 0.3381295
##
## $f1
## [1] 0.4122807
##
## $accuracy
## [1] 0.5533333
```

Random Forest

```
# Random Forest
start.time <- Sys.time()
model.rf <- train.rf(train.lbs)
```

```
## mtry = 3  OOB error = 25.83%
## Searching left ...
## mtry = 2     OOB error = 25.92%
## -0.003225806 1e-05
## Searching right ...
## mtry = 4     OOB error = 26%
## -0.006451613 1e-05
```



```
pre.rf <- test.rf(model.rf,test.lbs.x)
rf.lbs <- table(pre.rf,test.lbs$y)
end.time <- Sys.time()
rf.lbs.time <- end.time-start.time
performance_statistics(rf.lbs)
```

```
## $precision
## [1] 0.7278912
##
## $recall
## [1] 0.7697842
##
## $f1
## [1] 0.7482517
##
## $accuracy
## [1] 0.76
```

Logistic

```
start.time <- Sys.time()
res_logi = train.log(train.lbs)
pred.logi = test.log(res_logi,test.lbs.x)
log.lbs <- table(pred.logi,test.lbs$y)
end.time <- Sys.time()
log.lbs.time <- end.time-start.time
performance_statistics(log.lbs)
```

```
## $precision
## [1] 0.5087719
```

```
##
## $recall
## [1] 0.4172662
##
## $f1
## [1] 0.458498
##
## $accuracy
## [1] 0.5433333
```

GBM

```r
# GBM
start.time <- Sys.time()
res_gbm = train.gbm(train.lbs)
pred.gbm = test.gbm(res_gbm,test.lbs.x)
sas.gbm.sum = table(pred.gbm,test.lbs$y)
end.time <- Sys.time()
gbm.sas.time <- end.time-start.time
performance_statistics(sas.gbm.sum)
```

```
## $precision
## [1] 0.5912409
##
## $recall
## [1] 0.5827338
##
## $f1
## [1] 0.5869565
##
## $accuracy
## [1] 0.62
```