library(rpart) #classification and regression trees

library(partykit) #treeplots

library(MASS) #breast and pima indian data

library(ElemStatLearn) #prostate data

library(randomForest) #random forests

library(gbm) #gradient boosting

library(caret) #tune hyper-parameters

###########CART first

data(prostate)

prostate$gleason = ifelse(prostate$gleason == 6, 0, 1)

pros.train = subset(prostate, train==TRUE)[,1:9]

pros.test = subset(prostate, train==FALSE)[,1:9]

set.seed(123)

tree.pros = rpart(lpsa~., data=pros.train)

print(tree.pros$cptable)

plotcp(tree.pros)

cp = min(tree.pros$cptable[5,])

prune.tree.pros = prune(tree.pros, cp = cp)

plot(as.party(tree.pros))

plot(as.party(prune.tree.pros))

party.pros.test = predict(prune.tree.pros, newdata=pros.test)

rpart.resid = party.pros.test - pros.test$lpsa #calculate residual

mean(rpart.resid^2)

########CART breast cancer

data(biopsy)

biopsy <- biopsy[,-1]

names(biopsy) = c("thick", "u.size", "u.shape", "adhsn", "s.size", "nucl", "chrom", "n.nuc", "mit", "class")

biopsy.v2 = na.omit(biopsy)

set.seed(123) #random number generator

ind = sample(2, nrow(biopsy.v2), replace=TRUE, prob=c(0.7, 0.3))

biop.train = biopsy.v2[ind==1,] #the training data set

biop.test = biopsy.v2[ind==2,] #the test data set

str(biop.test)

set.seed(123)

tree.biop = rpart(class~., data=biop.train)

print(tree.biop$cptable)

cp = min(tree.biop$cptable[3,])

prune.tree.biop = prune(tree.biop, cp = cp)

plot(as.party(tree.biop))

plot(as.party(prune.tree.biop))

rparty.test = predict(prune.tree.biop, newdata=biop.test, type="class")

table(rparty.test, biop.test$class)

(136+64)/209

################RF

set.seed(123)

rf.pros = randomForest(lpsa~., data=pros.train)

print(rf.pros)

plot(rf.pros)

which.min(rf.pros$mse)

set.seed(123)

rf.pros.2 = randomForest(lpsa~., data=pros.train, ntree=70)

print(rf.pros.2)

varImpPlot(rf.pros.2, scale=TRUE,main="Variable Importance Plot - PSA Score")

importance(rf.pros.2)

rf.pros.test = predict(rf.pros.2, newdata=pros.test)

#plot(rf.pros.test, pros.test$lpsa)

rf.resid = rf.pros.test - pros.test$lpsa #calculate residual

mean(rf.resid^2)

set.seed(123)

rf.biop = randomForest(class~., data=biop.train)

print(rf.biop)

plot(rf.biop)

which.min(rf.biop$err.rate[,1])

set.seed(123)

rf.biop.2 = randomForest(class~., data=biop.train, ntree=19)

#getTree(rf.biop,1)

print(rf.biop.2)

rf.biop.test = predict(rf.biop.2, newdata=biop.test, type="response")

table(rf.biop.test, biop.test$class)

(139+67)/209

varImpPlot(rf.biop.2)

data(Pima.tr)

data(Pima.te)

pima = rbind(Pima.tr, Pima.te)

set.seed(502)

ind = sample(2, nrow(pima), replace=TRUE, prob=c(0.7,0.3))

pima.train = pima[ind==1,]

pima.test = pima[ind==2,]

set.seed(321)

rf.pima = randomForest(type~., data=pima.train)

print(rf.pima)

plot(rf.pima)

which.min(rf.pima$err.rate[,1])

set.seed(321)

rf.pima.2 = randomForest(type~., data=pima.train, ntree=80)

print(rf.pima.2)

rf.pima.test = predict(rf.pima.2, newdata=pima.test, type="response")

table(rf.pima.test, pima.test$type)

(75+33)/147

#varImpPlot(rf.pima.2)

########gbm

grid = expand.grid(.n.trees=seq(100,500, by=200), .interaction.depth=seq(1,4, by=1), .shrinkage=c(.001,.01,.1), .n.minobsinnode=10)

control = trainControl(method="LOOCV")

set.seed(123)

gbm.pros.train = train(lpsa~., data=pros.train, method="gbm", trControl=control, tuneGrid=grid)

gbm.pros.train

gbm.pros = gbm(lpsa~.,data=pros.train, n.trees=300,interaction.depth=3,shrinkage=0.01, distribution="gaussian")

gbm.pros.test = predict(gbm.pros, newdata=pros.test, n.trees=300)

gbm.resid = gbm.pros.test - pros.test$lpsa

mean(gbm.resid^2)

plot(gbm.pros.test, pros.test$lpsa, main="Predicted versus Actuals")

##############breast cancer next

#biop.train$class = ifelse(biop.train$class=="benign",0,1)

#biop.train$class = as.factor(biop.train$class)

control = trainControl(method="CV", number=10)

set.seed(123)

gbm.biop.train = train(class~., data=biop.train, method="gbm", trControl=control, tuneGrid=grid)

gbm.biop.train

#require 0/1

biop.train$class = ifelse(biop.train$class=="benign",0,1)

gbm.biop = gbm(class~., distribution="bernoulli", data=biop.train, n.trees=100, interaction.depth=1, shrinkage=0.1)

gbm.biop.test = predict(gbm.biop, newdata=biop.test, type="response", n.trees=100)

gbm.class = ifelse(gbm.biop.test <0.5,"benign", "malignant")

table(gbm.class, biop.test$class)

(140+65)/209

set.seed(123)

gbm.pima.train = train(type~., data=pima.train, method="gbm", trControl=control, tuneGrid=grid)

gbm.pima.train

pima.train$type = ifelse(pima.train$type=="No",0,1)

gbm.pima = gbm(type~., distribution="bernoulli", data=pima.train, n.trees=500,interaction.depth=3,shrinkage=0.01)

gbm.pima.test = predict(gbm.pima, newdata=pima.test, type="response", n.trees=500)

gbm.type = ifelse(gbm.pima.test <0.5,"No", "Yes")

table(gbm.type, pima.test$type)

(77+32)/147

summary(gbm.pima)