Homework #8

Mike Lehman

October 25, 2016

#1  
url = "https://archive.ics.uci.edu/ml/machine-learning-databases/pima-indians-diabetes/pima-indians-diabetes.data"  
diabetes <- read.table(url, sep = ",", header = FALSE)

#2  
names(diabetes) <- c("npregant", "plasma", "bp", "triceps", "insulin", "bmi", "pedigree", "age", "class")  
str(diabetes)

## 'data.frame': 768 obs. of 9 variables:  
## $ npregant: int 6 1 8 1 0 5 3 10 2 8 ...  
## $ plasma : int 148 85 183 89 137 116 78 115 197 125 ...  
## $ bp : int 72 66 64 66 40 74 50 0 70 96 ...  
## $ triceps : int 35 29 0 23 35 0 32 0 45 0 ...  
## $ insulin : int 0 0 0 94 168 0 88 0 543 0 ...  
## $ bmi : num 33.6 26.6 23.3 28.1 43.1 25.6 31 35.3 30.5 0 ...  
## $ pedigree: num 0.627 0.351 0.672 0.167 2.288 ...  
## $ age : int 50 31 32 21 33 30 26 29 53 54 ...  
## $ class : int 1 0 1 0 1 0 1 0 1 1 ...

#3  
diabetes$class <- factor(diabetes$class, levels = c(0,1), labels = c("normal","diabetic"))

#4  
set.seed(123)  
train\_sample <- sample(768, 538) #70% = 537.6  
train <- diabetes[train\_sample,]  
test <- diabetes[-train\_sample,]

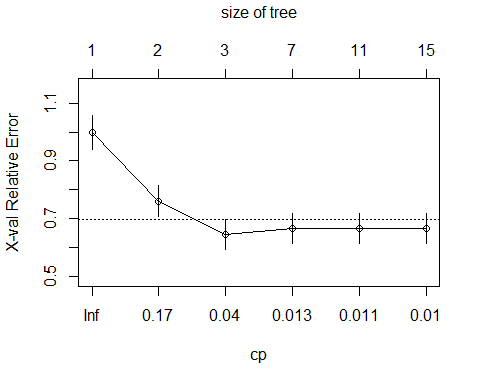
#5  
library(rpart)  
tree <- rpart(class ~ ., data = train, method="class", parms = list(split="information"))  
tree

## n= 538   
##   
## node), split, n, loss, yval, (yprob)  
## \* denotes terminal node  
##   
## 1) root 538 188 normal (0.65055762 0.34944238)   
## 2) plasma< 122.5 320 54 normal (0.83125000 0.16875000)   
## 4) bmi< 28.15 112 4 normal (0.96428571 0.03571429) \*  
## 5) bmi>=28.15 208 50 normal (0.75961538 0.24038462)   
## 10) npregant< 6.5 166 30 normal (0.81927711 0.18072289) \*  
## 11) npregant>=6.5 42 20 normal (0.52380952 0.47619048)   
## 22) pedigree< 0.776 34 12 normal (0.64705882 0.35294118)   
## 44) plasma< 99.5 16 1 normal (0.93750000 0.06250000) \*  
## 45) plasma>=99.5 18 7 diabetic (0.38888889 0.61111111) \*  
## 23) pedigree>=0.776 8 0 diabetic (0.00000000 1.00000000) \*  
## 3) plasma>=122.5 218 84 diabetic (0.38532110 0.61467890)   
## 6) bmi< 29.95 61 20 normal (0.67213115 0.32786885)   
## 12) age>=59.5 8 0 normal (1.00000000 0.00000000) \*  
## 13) age< 59.5 53 20 normal (0.62264151 0.37735849)   
## 26) age< 26 18 2 normal (0.88888889 0.11111111) \*  
## 27) age>=26 35 17 diabetic (0.48571429 0.51428571)   
## 54) plasma< 148 24 9 normal (0.62500000 0.37500000)   
## 108) plasma>=125.5 16 3 normal (0.81250000 0.18750000) \*  
## 109) plasma< 125.5 8 2 diabetic (0.25000000 0.75000000) \*  
## 55) plasma>=148 11 2 diabetic (0.18181818 0.81818182) \*  
## 7) bmi>=29.95 157 43 diabetic (0.27388535 0.72611465)   
## 14) plasma< 155.5 90 36 diabetic (0.40000000 0.60000000)   
## 28) bp>=43 83 36 diabetic (0.43373494 0.56626506)   
## 56) age< 34.5 40 17 normal (0.57500000 0.42500000)   
## 112) bmi< 41.8 31 10 normal (0.67741935 0.32258065) \*  
## 113) bmi>=41.8 9 2 diabetic (0.22222222 0.77777778) \*  
## 57) age>=34.5 43 13 diabetic (0.30232558 0.69767442) \*  
## 29) bp< 43 7 0 diabetic (0.00000000 1.00000000) \*  
## 15) plasma>=155.5 67 7 diabetic (0.10447761 0.89552239) \*

#6  
tree$cptable

## CP nsplit rel error xerror xstd  
## 1 0.26595745 0 1.0000000 1.0000000 0.05882527  
## 2 0.11170213 1 0.7340426 0.7606383 0.05450265  
## 3 0.01418440 2 0.6223404 0.6436170 0.05151239  
## 4 0.01241135 6 0.5585106 0.6648936 0.05210520  
## 5 0.01063830 10 0.5000000 0.6648936 0.05210520  
## 6 0.01000000 14 0.4414894 0.6648936 0.05210520

plotcp(tree)



min xerror = 0.6436170

xstd = 0.05151239

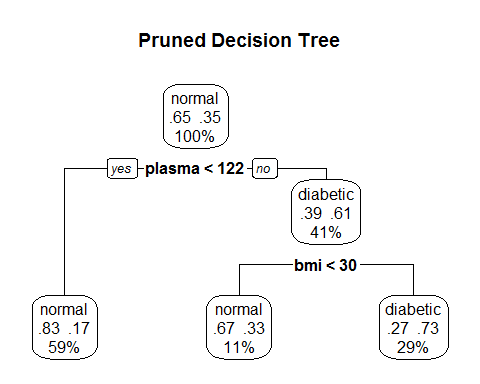
0.64 + 0.05 = 0.69

smallest tree (nsplit) with xerror below 0.69 = tree #3

cp = 0.01418440 ~ 0.0142

#7  
tree.pruned <- prune(tree, cp = .0142)

#8  
library(rpart.plot)  
prp(tree.pruned, type = 2, extra = 104, fallen.leaves = TRUE, main = "Pruned Decision Tree")



#9  
tree.prediction <- predict(tree.pruned, test, type = "class")  
tree.perf <- table(test$class, tree.prediction, dnn = c("Actual", "Predicted"))  
tree.perf

## Predicted  
## Actual normal diabetic  
## normal 115 35  
## diabetic 34 46

The pruned tree performed fairly well, correctly classifying 161 out of 230 test observations, for an accuracy of 70% and an error ate of 30%. The tree had nearly identical amounts of false negatives (34) and false positives (35).

#10  
library(randomForest)

## randomForest 4.6-12

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

set.seed(1234)  
tree.forest <- randomForest(class ~ ., data = train, na.action = na.roughfix, importance = TRUE)  
tree.forest

##   
## Call:  
## randomForest(formula = class ~ ., data = train, importance = TRUE, na.action = na.roughfix)   
## Type of random forest: classification  
## Number of trees: 500  
## No. of variables tried at each split: 2  
##   
## OOB estimate of error rate: 22.68%  
## Confusion matrix:  
## normal diabetic class.error  
## normal 302 48 0.1371429  
## diabetic 74 114 0.3936170

#11  
importance(tree.forest, type = 2)

## MeanDecreaseGini  
## npregant 20.22296  
## plasma 66.37462  
## bp 20.88104  
## triceps 18.14897  
## insulin 16.95255  
## bmi 41.61899  
## pedigree 29.18658  
## age 29.47099

#12  
forest.prediction <- predict(tree.forest, test, type = "class")  
forest.perf <- table(test$class, forest.prediction, dnn = c("Actual", "Predicted"))  
forest.perf

## Predicted  
## Actual normal diabetic  
## normal 119 31  
## diabetic 32 48

The random forest performed only slightly better than the pruned tree from question #9. The error rate improved by only two percent. Both false negatives and false positives decreased, and still remained close. The random forest performed slightly better against the training data (24% error rate) than against the test data (28% error rate), which is to be expected. However, the random forest did not perform much better than a single, pruned tree.