Test3

Mao Soldevilla

2/10/2020

Loading Libraries

```r
library(caret)
```

```
FALSE Warning: package ’caret’ was built under R version 4.0.2
```

```
FALSE Loading required package: lattice
```

```
FALSE Loading required package: ggplot2
```

```
FALSE Warning: package ’ggplot2’ was built under R version 4.0.2
```

```r
library(ggplot2)
library(AppliedPredictiveModeling)
```

```
FALSE Warning: package ’AppliedPredictiveModeling’ was built under R version 4.0.2
```

```r
library(rattle)
```

```
FALSE Warning: package ’rattle’ was built under R version 4.0.2
```

```
FALSE Loading required package: tibble
```

```
FALSE Warning: package ’tibble’ was built under R version 4.0.2
```

```
FALSE Loading required package: bitops
```

```
FALSE Rattle: A free graphical interface for data science with R.
FALSE Versión 5.4.0 Copyright (c) 2006-2020 Togaware Pty Ltd.
FALSE Escriba ’rattle()’ para agitar, sacudir y  rotar sus datos.
```

```r
library(dplyr)
```

```
FALSE Warning: package ’dplyr’ was built under R version 4.0.2
```

```
FALSE
FALSE Attaching package: 'dplyr'

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

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

1. For this quiz we will be using several R packages. R package versions change over time, the right answers have been checked using the following versions of the packages.

AppliedPredictiveModeling: v1.1.6

caret: v6.0.47

ElemStatLearn: v2012.04-0

pgmm: v1.1

rpart: v4.1.8

If you aren't using these versions of the packages, your answers may not exactly match the right answer, but hopefully should be close.

Load the cell segmentation data from the AppliedPredictiveModeling package using the commands:

```
library(AppliedPredictiveModeling)
data(segmentationOriginal)
library(caret)
library(lattice)
library(ggplot2)
```

1. Subset the data to a training set and testing set based on the Case variable in the data set.
2. Set the seed to 125 and fit a CART model to predict Class with the rpart method using all predictor variables and default caret settings.
3. In the final model what would be the final model prediction for cases with the following variable values:

a. TotalIntench2 = 23,000; FiberWidthCh1 = 10; PerimStatusCh1=2 answer: PS
b. TotalIntench2 = 50,000; FiberWidthCh1 = 10;VarIntenCh4 = 100 answer: WS
c. TotalIntench2 = 57,000; FiberWidthCh1 = 8;VarIntenCh4 = 100 answer: PS
d. FiberWidthCh1 = 8;VarIntenCh4 = 100; PerimStatusCh1=2 answer: not possible

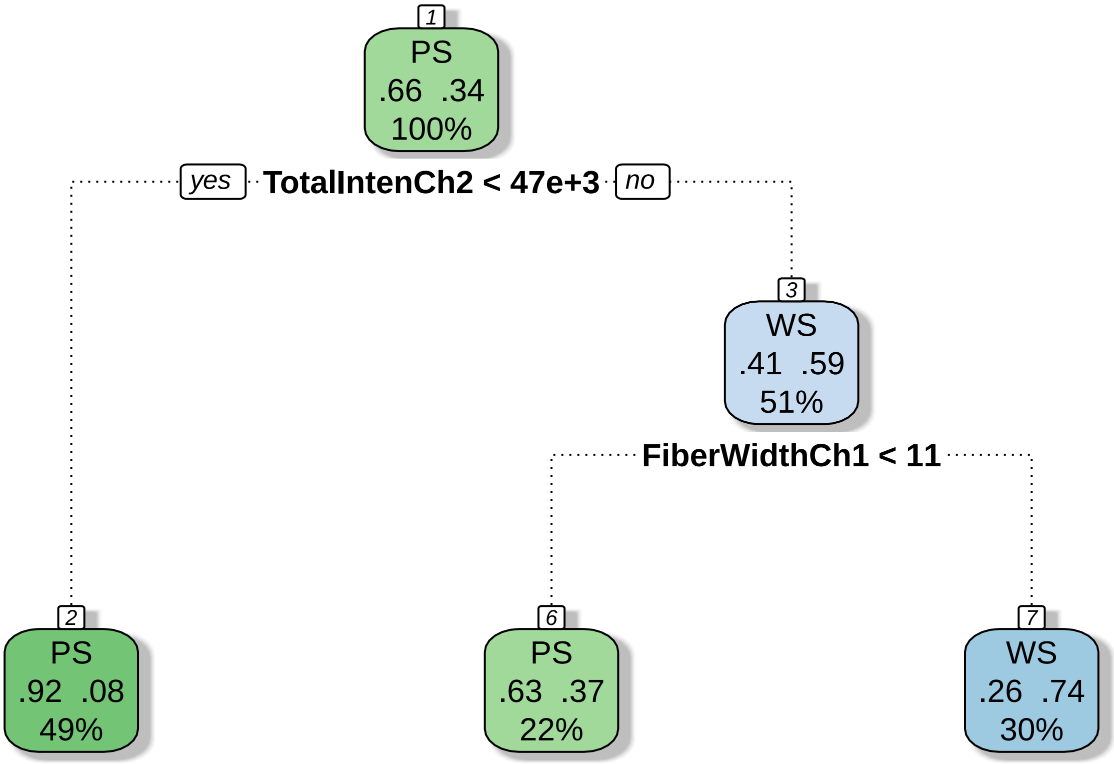
```
data <- segmentationOriginal
dim(data)
```

```
## [1] 2019  119
```

```r
inTrain <- createDataPartition(y = data$Case, p = 0.7, list = FALSE)
training <- data[inTrain, ]
testing <- data[-inTrain, ]
set.seed(125)
modFit <- train(Class ~ ., method = "rpart", data = training)
print(modFit$finalModel)
```

```
## n= 1414
##
## node), split, n, loss, yval, (yprob)
##       * denotes terminal node
##
## 1) root 1414 485 PS (0.6570014 0.3429986)
##   2) TotalIntenCh2< 47255.5 686  58 PS (0.9154519 0.0845481) *
##   3) TotalIntenCh2>=47255.5 728 301 WS (0.4134615 0.5865385)
##     6) FiberWidthCh1< 11.20335 310 116 PS (0.6258065 0.3741935) *
##     7) FiberWidthCh1>=11.20335 418 107 WS (0.2559809 0.7440191) *
```

```r
#plot(modFit$finalModel, uniform = TRUE, main= "Classification Tree")
#text(modFit$finalModel, use.n = TRUE, all = T, cex = .8)
#predict(modFit, testing)
fancyRpartPlot(modFit$finalModel)
```



Rattle 2020−Oct.−02 20:33:19 maole

```r
#predict(modFit, newdata = testing)
```

The answer above interlines by plot.

2. If K is small in a K-fold cross validation is the bias in the estimate of out-of-sample (test set) accuracy smaller or bigger? If K is small is the variance in the estimate of out-of-sample (test set) accuracy smaller or bigger. Is K large or small in leave one out cross validation?

The bias is smaller and the variance is smaller. Under leave one out cross validation K is equal to the sample size.

3. Load the olive oil data using the commands:

```r
library(pgmm)
data(olive)
olive = olive[,-1]
```

(NOTE: If you have trouble installing the pgmm package, you can download the -code-olive-/code- dataset here: olive_data.zip. After unzipping the archive, you can load the file using the -code-load()-/code- function in R.)

These data contain information on 572 different Italian olive oils from multiple regions in Italy. Fit a classification tree where Area is the outcome variable. Then predict the value of area for the following data frame using the tree command with all defaults

```r
#inTrain <- createDataPartition(y = olive$Area, p = 0.7, list = FALSE)
#training <- olive[inTrain, ]
newdata = as.data.frame(t(colMeans(olive)))
#testing <- olive[-inTrain, ]
set.seed(333)
modFit <- train(Area ~ ., method = "rpart", data = olive)
```

```
## Warning in nominalTrainWorkflow(x = x, y = y, wts = weights, info = trainInfo, :
## There were missing values in resampled performance measures.
```

```r
predict(modFit, newdata = newdata)
```

```
##        1
## 2.783282
```

4. Load the South Africa Heart Disease Data and create training and test sets with the following code:

```
#library(ElemStatLearn)
#data(SAheart)
SAheart <- read.csv("SAheart.csv")
#SAheart <- mutate(SAheart, chd = levels(chd, labels = c("Si", "No"), levels = c(1, 0)))
SAheart[SAheart$chd == "Si", ]$chd <- 1
SAheart[SAheart$chd == "No", ]$chd <- 0
SAheart$chd <- as.numeric(SAheart$chd)
set.seed(8484)
train = sample(1:dim(SAheart)[1],size=dim(SAheart)[1]/2,replace=F)
trainSA = SAheart[train,]
testSA = SAheart[-train,]
```

Then set the seed to 13234 and fit a logistic regression model (method="glm", be sure to specify family="binomial") with Coronary Heart Disease (chd) as the outcome and age at onset, current alcohol consumption, obesity levels, cumulative tabacco, type-A behavior, and low density lipoprotein cholesterol as predictors. Calculate the misclassification rate for your model using this function and a prediction on the "response" scale:

```
set.seed(13234)
#modFit <- glm(chd ~ age + alcohol + obesity + tobacco + typea + ldl, family = "binomial", data = testS
modFit <- train(chd ~ age + alcohol + obesity + tobacco + typea + ldl, method = "glm", data = trainSA, 
```

```
## Warning in train.default(x, y, weights = w, ...): You are trying to do
## regression and your outcome only has two possible values Are you trying to do
## classification? If so, use a 2 level factor as your outcome column.
```

```
#modFitTe <- train(chd ~ age + alcohol + obesity + tobacco + typea + ldl, method = "glm", data = trainS
#summary(modFit)
```

```
missClass = function(values,prediction){
        #values[values == "Si"] <- 1
        #values[values == "No"] <- 0
        #prediction[prediction == "Si"] <- 1
        #prediction[prediction == "No"] <- 0
        sum(((prediction > 0.5)*1) != values)/length(values)
        }
missClass(trainSA$chd, predict(modFit, newdata = trainSA))
```

```
## [1] 0.3116883
```

```
missClass(testSA$chd, predict(modFit, newdata = testSA))
```

```
## [1] 0.2813853
```

```
#missClass(olive, modFit2)
```

5. Load the vowel.train and vowel.test data sets:

```
# The library ElemStatLearn was retired from CRAN
#library(ElemStatLearn)
#data(vowel.train)
#data(vowel.test)
# Data obtainded from https://web.stanford.edu/~hastie/ElemStatLearn/data.html
#url <- "https://web.stanford.edu/~hastie/ElemStatLearn/datasets/vowel.train"
vowel.train <- read.csv("https://web.stanford.edu/~hastie/ElemStatLearn/datasets/vowel.train")
vowel.test <- read.csv("https://web.stanford.edu/~hastie/ElemStatLearn/datasets/vowel.test")
```

Set the variable y to be a factor variable in both the training and test set. Then set the seed to 33833. Fit a random forest predictor relating the factor variable y to the remaining variables. Read about variable importance in random forests here: http://www.stat.berkeley.edu/~breiman/RandomForests/cc_home.htm#ooberr The caret package uses by default the Gini importance.

```
set.seed(33833)
#modFit <- train(y ~ ., data = vowel.train, method = "rf", prox = TRUE)
myrf <- randomForest::randomForest(y ~ ., data = vowel.train)#, importance = TRUE, mtry = 3)
```

Calculate the variable importance using the varImp function in the caret package. What is the order of variable importance?

[NOTE: Use randomForest() specifically, not caret, as there's been some issues reported with that approach. 11/6/2016]

```
importance <- varImp(myrf)
importance
```

```
##              Overall
## row.names  147.6079
## x.1        938.2990
## x.2       1459.5662
## x.3        249.0419
## x.4        301.6438
## x.5        396.5347
## x.6        566.1163
## x.7        176.9082
## x.8        480.2494
## x.9        322.8302
## x.10       190.3514
```

```
importance$vars <- rownames(importance)
importance <- importance[order(importance$Overall, decreasing = TRUE),]
importance
```

```
##              Overall      vars
## x.2       1459.5662       x.2
## x.1        938.2990       x.1
## x.6        566.1163       x.6
## x.8        480.2494       x.8
## x.5        396.5347       x.5
## x.9        322.8302       x.9
## x.4        301.6438       x.4
## x.3        249.0419       x.3
## x.10       190.3514      x.10
## x.7        176.9082       x.7
## row.names  147.6079 row.names
```

The order of the variables is: x.2, x.1, x.5, x.6, x.8, x.4, x.9, x.3, x.7,x.10