Practical Machine Learning - Week 3 Quiz

Ken Wood

9/17/2020

Question 1 Load the cell segmentation data from the AppliedPredictiveModeling package using the commands:

```
rm(list = ls())

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

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

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

1. Subset the data to a training set and testing set based on the Case variable in the data set.

```
train = segmentationOriginal[segmentationOriginal$Case=="Train",]
test = segmentationOriginal[segmentationOriginal$Case=="Test",]
```

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.

```
set.seed(125)
modFit <- train(Class~.,method="rpart",data=train)
modFit$finalModel
```

```
## n= 1009
##
## node), split, n, loss, yval, (yprob)
##       * denotes terminal node
##
## 1) root 1009 373 PS (0.63032706 0.36967294)
##   2) TotalIntenCh2< 45323.5 454  34 PS (0.92511013 0.07488987) *
##   3) TotalIntenCh2>=45323.5 555 216 WS (0.38918919 0.61081081)
##     6) FiberWidthCh1< 9.673245 154  47 PS (0.69480519 0.30519481) *
##     7) FiberWidthCh1>=9.673245 401 109 WS (0.27182045 0.72817955) *
```

```
library(rattle)
```

```
## Loading required package: tibble
```

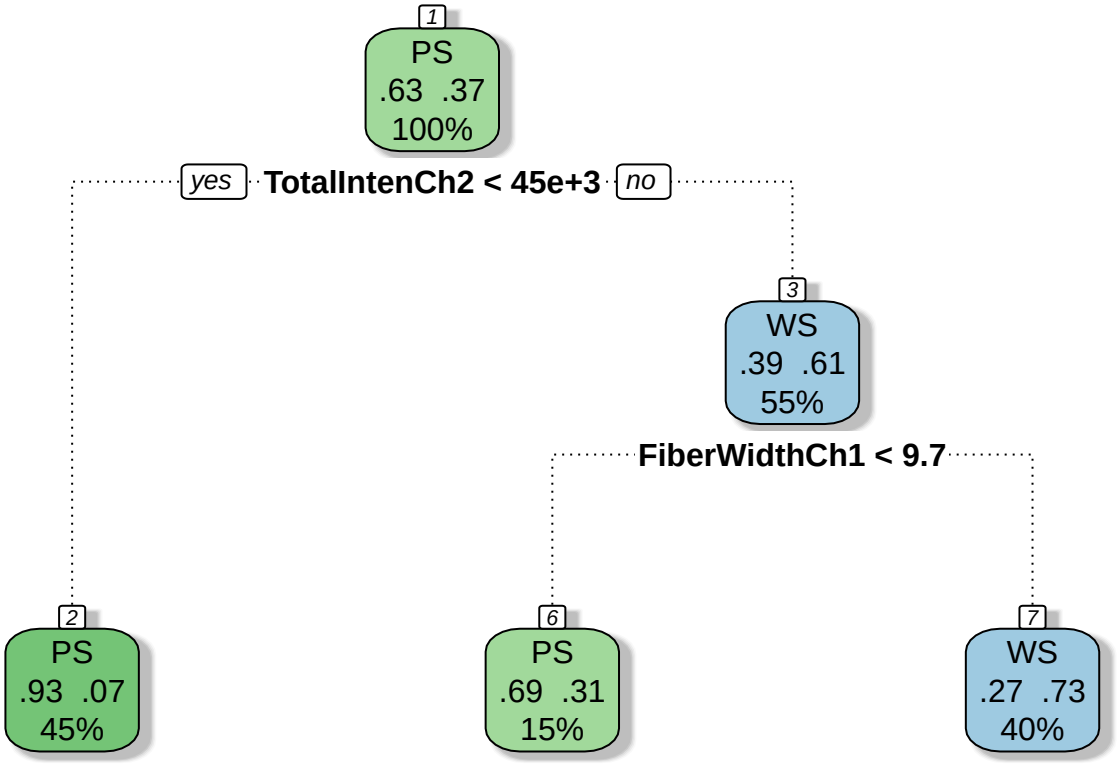
```
## Loading required package: bitops
```

```
## Rattle: A free graphical interface for data science with R.
## Version 5.4.0 Copyright (c) 2006-2020 Togaware Pty Ltd.
## Type 'rattle()' to shake, rattle, and roll your data.
```

```
library(rpart.plot)
```

```
## Loading required package: rpart
```

```
fancyRpartPlot(modFit$finalModel)
```



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

b. TotalIntench2 = 50,000; FiberWidthCh1 = 10;VarIntenCh4 = 100
c. TotalIntench2 = 57,000; FiberWidthCh1 = 8;VarIntenCh4 = 100
d. FiberWidthCh1 = 8;VarIntenCh4 = 100; PerimStatusCh1=2

Answers (determined by inspecting the tree diagram):

a. PS
b. WS
c. PS
d. Not possible to predict.

Question 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?

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

Question 3 Load the olive oil data using the commands:

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

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.

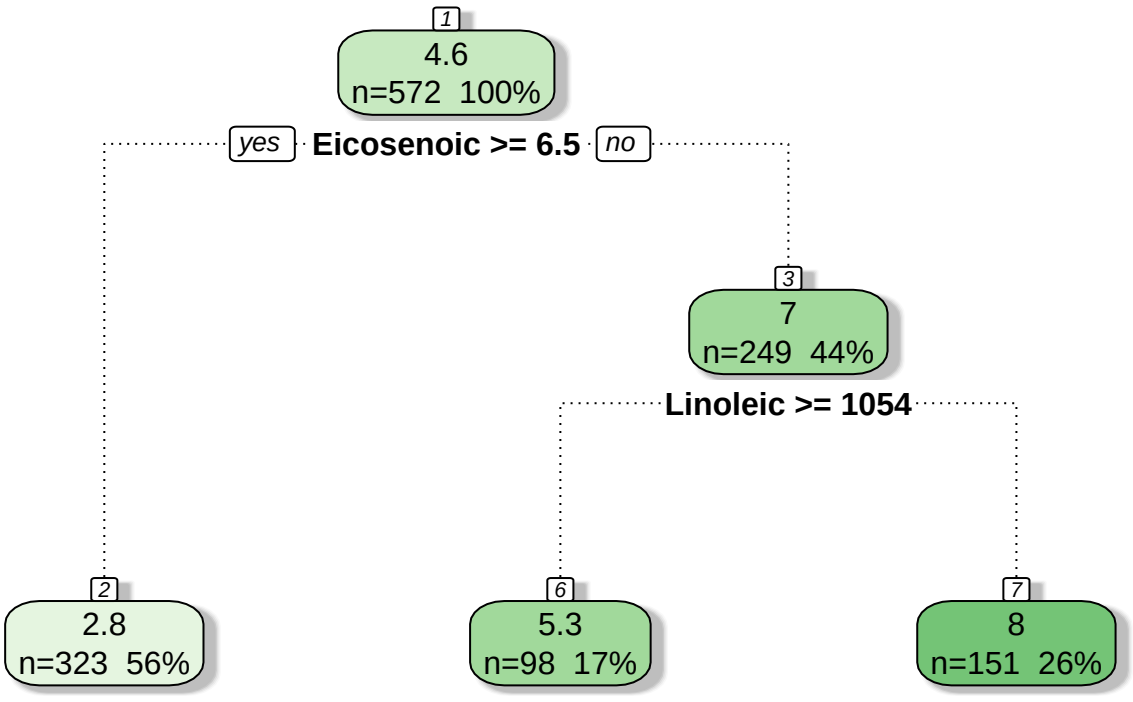
```
modFit1 <- train(Area~.,method="rpart",data=olive)
```

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

```
modFit1$finalModel
```

```
## n= 572
##
## node), split, n, deviance, yval
##       * denotes terminal node
##
## 1) root 572 3171.32000 4.599650
##   2) Eicosenoic>=6.5 323  176.82970 2.783282 *
##   3) Eicosenoic< 6.5 249  546.51410 6.955823
##     6) Linoleic>=1053.5 98   21.88776 5.336735 *
##     7) Linoleic< 1053.5 151  100.99340 8.006623 *
```

```
fancyRpartPlot(modFit1$finalModel)
```




```
newdata = as.data.frame(t(colMeans(olive)))
predict(modFit1,newdata = newdata)
```

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

The predicted value is 2.783. It is strange because Area should be a qualitative variable - but tree is reporting the average value of Area as a numeric variable in the leaf predicted for newdata.

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

```
library(ElemStatLearn)
data(SAheart)
set.seed(8484)
train = sample(1:dim(SAheart)[1],size=dim(SAheart)[1]/2,replace=F)
trainSA = SAheart[train,]
testSA = SAheart[-train,]
```

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.

```
set.seed(13234)
modFit2 <- train(chd ~ age + alcohol + obesity + tobacco + typea + ldl, data = trainSA, method = "glm",
```

```
## 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.
```

```
summary(modFit2)
```

```
##
## Call:
## NULL
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.8759  -0.8276  -0.5013   1.0320   2.2969
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept) -4.114985   1.397818  -2.944 0.003241 **
## age          0.052552   0.013804   3.807 0.000141 ***
## alcohol      0.004216   0.006293   0.670 0.502917
## obesity     -0.058587   0.040817  -1.435 0.151182
## tobacco      0.043858   0.036175   1.212 0.225375
## typea        0.027646   0.015904   1.738 0.082169 .
## ldl          0.160496   0.077954   2.059 0.039508 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 285.04  on 230  degrees of freedom
## Residual deviance: 245.54  on 224  degrees of freedom
## AIC: 259.54
##
## Number of Fisher Scoring iterations: 4
```

Calculate the misclassification rate for your model using this function and a prediction on the "response" scale:

```
missClass = function(values,prediction){sum(((prediction > 0.5)*1) != values)/length(values)}

# Calculate misclassification rate on 'trainSA' dataset
```

```
missClass(trainSA$chd,predict(modFit2, newdata = trainSA))
```

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

```
# Calculate misclassification rate on 'testSA' dataset
missClass(testSA$chd,predict(modFit2, newdata = testSA))
```

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

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

```
library(ElemStatLearn)
data(vowel.train)
data(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 the Gini importance be default. Calculate the variable importance using the varImp function in the caret package. What is the order of variable importance?

```
set.seed(33833)
library(randomForest)
```

```
## randomForest 4.6-14
```

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

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

```
## The following object is masked from 'package:rattle':
##
##     importance
```

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

```
vowel.train$y <- as.factor(vowel.train$y)
vowel.test$y <- as.factor(vowel.test$y)

modvowel <- randomForest(y ~ ., data = vowel.train)
order(varImp(modvowel), decreasing = TRUE)
```

```
##  [1]  1  2  5  6  8  4  3  9  7 10
```