Quiz 3

Xi Fang

6/25/2020

Q1.

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

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

b. TotalIntench2 = 50,000; FiberWidthCh1 = 10;VarIntenCh4 = 100

c. TotalIntench2 = 57,000; FiberWidthCh1 = 8;VarIntenCh4 = 100

d. FiberWidthCh1 = 8;VarIntenCh4 = 100; PerimStatusCh1=2

TIP: Plot the resulting tree and to use the plot to answer this question.

```
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.
training <- subset(segmentationOriginal, Case =="Train")
testing <- subset(segmentationOriginal, Case == "Test")

## 2. Set the seed to 125 and fit a CART model to predict Class with the rpart method using all predict
set.seed(125)
modelfit <- train(Class ~., method= "rpart", data= training)

## 3. In the final model what would be the final model prediction for cases with the following variable
modelfit$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) *
```

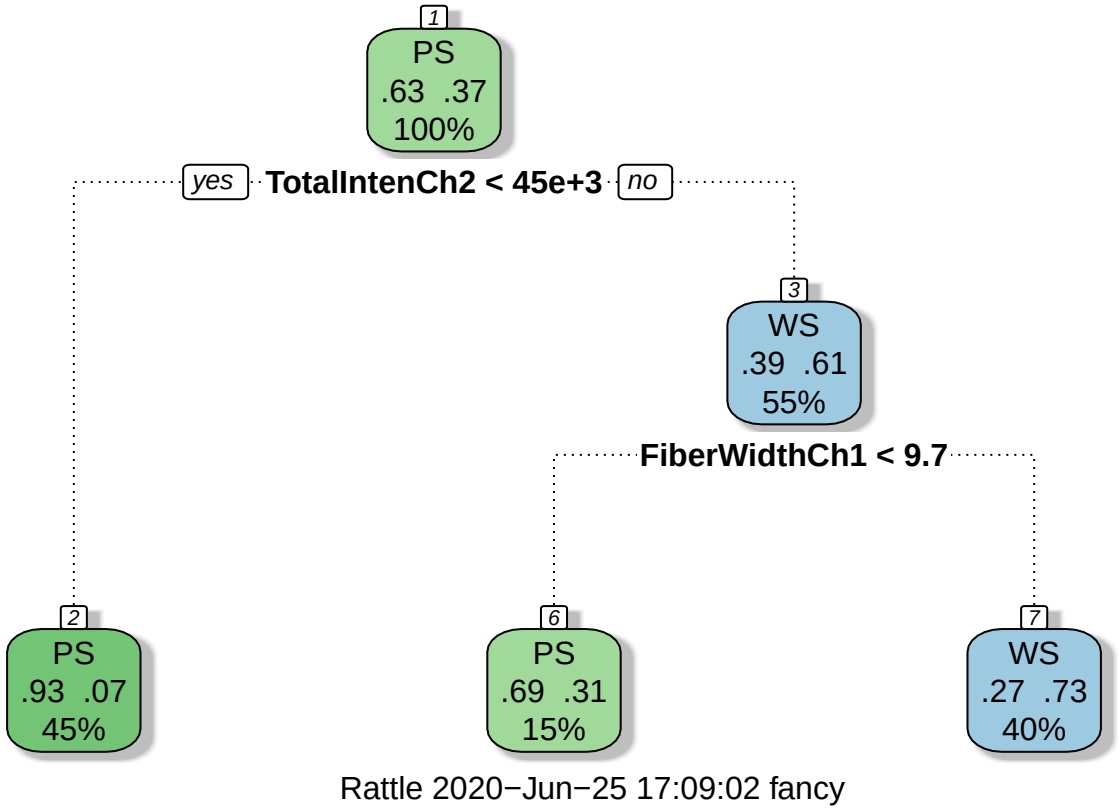
```
library(rattle)
```

```
## 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.
```

```
fancyRpartPlot(modelfit$finalModel)
```



```
#dev.copy(png, file="q3q1.png", width=500, height=470)
#dev.off()
```

Q2

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 larger and the variance is smaller. Under leave one out cross validation K is equal to th
```

Q3

Load the olive oil data using the commands:

(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

What is the resulting prediction? Is the resulting prediction strange? Why or why not?

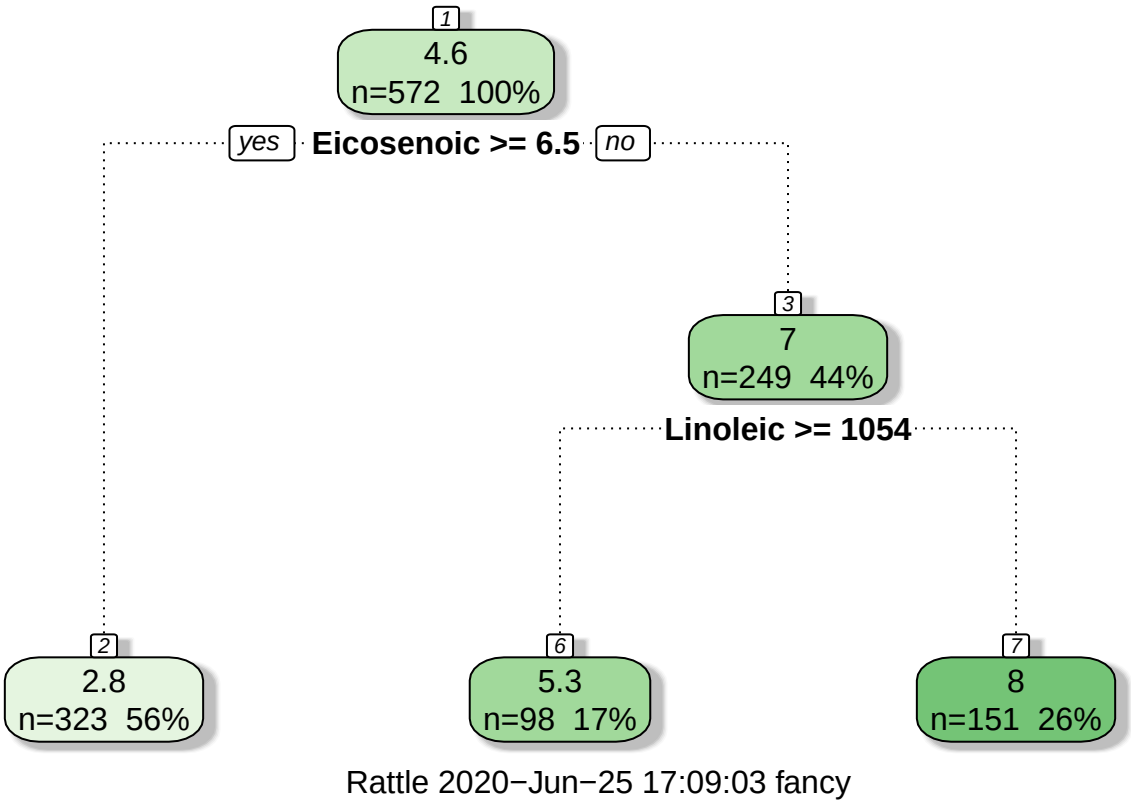
```
library(pgmm)
data(olive)
olive = olive[,-1]
modelfit <- train(Area ~., method = "rpart", data=olive)
```

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

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

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

```
library(rattle)
fancyRpartPlot(modelfit$finalModel)
```



Q4

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

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:

What is the misclassification rate on the training set? What is the misclassification rate on the test set?

```
library(remotes)
# install_version("ElemStatLearn", "2015.6.26.2")
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,]
# fit a logistic regression model (method="glm", be sure to specify family="binomial") with Coronary He
set.seed(13234)
modelfit <- train(chd ~ age+alcohol+obesity+tobacco+typea+ldl, method="glm", data=trainSA, family="bino
```

```
## 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.
```

```
# Calculate the misclassification rate for your model using this function and a prediction on the "resp
missClass = function(values,prediction){sum(((prediction > 0.5)*1) != values)/length(values)}
missClass(trainSA$chd, predict(modelfit, trainSA))
```

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

```
missClass(testSA$chd, predict(modelfit, testSA))
```

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

Q5

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

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.

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]

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

```
modelfit <- randomForest(y~., data=vowel.train)
library(caret)
order(varImp(modelfit), decreasing = T)
```

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