Week3_Quiz

Anyi Guo

28/12/2018

Week 3 Quiz

Q1

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

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

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

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

```
library(rattle)
```

```
## Rattle: A free graphical interface for data science with R.
## Version 5.2.0 Copyright (c) 2006-2018 Togaware Pty Ltd.
## Entrez 'rattle()' pour secouer, faire vibrer, et faire défiler vos données.
```

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

```
set.seed(125)
inTrain<-createDataPartition(y=segmentationOriginal$Case,p=0.7,list=FALSE)
training<-segmentationOriginal[inTrain,]
testing<-segmentationOriginal[-inTrain,]
```

2. Set the seed to 125 and fit a CART model with the rpart method using all predictor variables and default caret settings.

```
modFit<-train(Class~.,method="rpart",data=training,tuneLength=10)
```

3. In the final model what would be the final model prediction for cases with the following variable values:

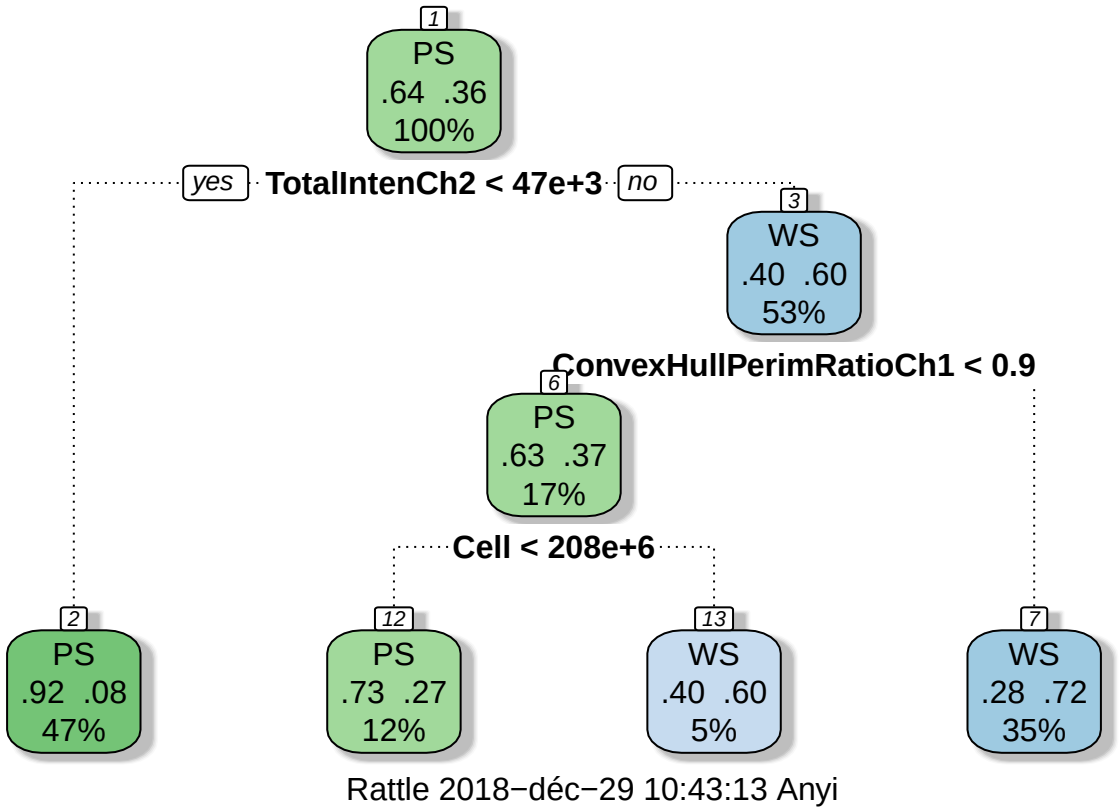
a. TotalIntench2 = 23,000; FiberWidthCh1 = 10; PerimStatusCh1=2

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

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

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

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



Answer

a. PS

b. Not possible to predict

c. PS

d. WS

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?

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

Q3

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. Then predict the value of area for the following data frame using the tree command with all defaults

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

```
## 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(modFit,newdata)
```

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

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

Answer 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

Q4

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,]
```

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<-train(chd~age+alcohol+obesity+tobacco+typea+ldl,data=trainSA,method="glm",family="binomial")
```

```
## 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.
```

```
pred<-predict(modFit,testSA)
pred2<-predict(modFit,trainSA)

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

# Misclassification on testing set
missClass(testSA$chd,pred)
```

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

```
# Misclassification on training set
missClass(trainSA$chd,pred2)
```

```
## [1] 0.2727273
```

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

Answer * Test Set Misclassification: 0.31 * Training Set: 0.27

Q5

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 by default the Gini importance.

```
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
```

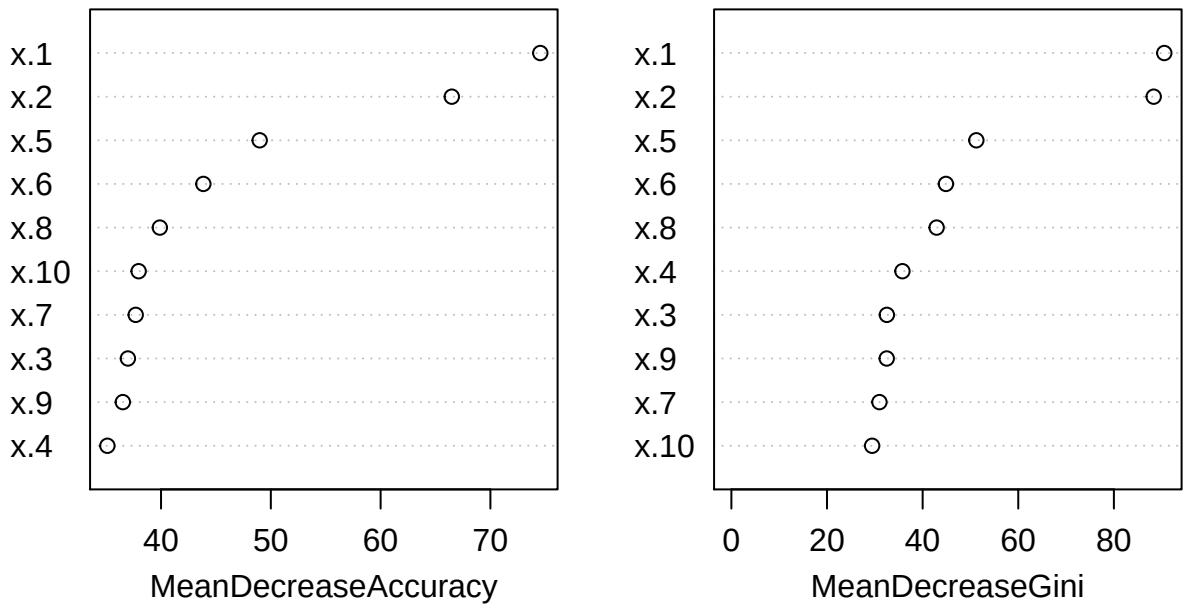
```
## The following object is masked from 'package:ggplot2':
##
##     margin
```

```r
vowel.train$y<-as.factor(vowel.train$y)
vowel.test$y<-as.factor(vowel.test$y)
set.seed(33833)

myForest<-randomForest(y~.,data=vowel.train,importance=TRUE)
importance(myForest)
```

```
##             1        2        3        4        5        6        7
## x.1  29.11895 29.44922 33.18261 48.42164 42.93955 35.09701 35.19359
## x.2  54.80527 42.99639 37.58728 31.51004 32.84094 24.78946 28.79998
## x.3  15.87398 15.59555 19.74431 19.93527 13.92606 14.86718 16.14087
## x.4  18.54148 16.73341 19.41532 20.31802 18.89242 18.34816 14.87841
## x.5  21.58627 26.27623 28.69133 27.81236 30.09398 21.83958 24.61554
## x.6  23.13863 27.33672 22.52596 27.69695 17.23596 23.47240 22.94971
## x.7  15.97967 18.02095 16.66341 13.52179 17.73142 15.97313 17.11697
## x.8  17.83590 20.32252 19.31248 23.83211 16.76017 19.56210 22.32204
## x.9  12.67408 17.92742 16.27996 16.02322 12.98966 15.46911 16.18884
## x.10 15.41072 14.56767 16.84268 15.16088 14.82269 11.24023 15.44459
##             8        9       10       11 MeanDecreaseAccuracy
## x.1  44.18992 43.62221 52.36179 40.32181              74.54433
## x.2  33.13660 27.27830 30.81691 36.33539              66.48362
## x.3  15.45219 18.05612 18.19880 17.66303              36.98281
## x.4  22.11390 19.67952 19.24777 19.18863              35.11464
## x.5  33.57581 24.45729 27.33320 26.67657              48.98575
## x.6  22.81997 21.13448 21.65237 25.87921              43.85545
## x.7  12.75196 10.53571 16.09083 15.85205              37.69351
## x.8  24.26107 25.56857 21.55164 21.04051              39.89072
## x.9  16.93566 14.77635 19.09838 18.87731              36.51970
## x.10 14.29647 13.12904 12.48161 16.67645              37.95938
##      MeanDecreaseGini
## x.1          90.53672
## x.2          88.32345
## x.3          32.51581
## x.4          35.77079
## x.5          51.22808
## x.6          44.87199
## x.7          30.96095
## x.8          42.92778
## x.9          32.48635
## x.10         29.43556
```

```r
varImpPlot(myForest)
```



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]

Answer

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