Practical Machine Learning - Quiz3 - Data Science Specialization

Michael Karp

1/22/2015

```
# q1 - Load the cell segmentation data from the AppliedPredictiveModeling package using the
commands:
library(AppliedPredictiveModeling)
data(segmentationOriginal)
library(caret)
```

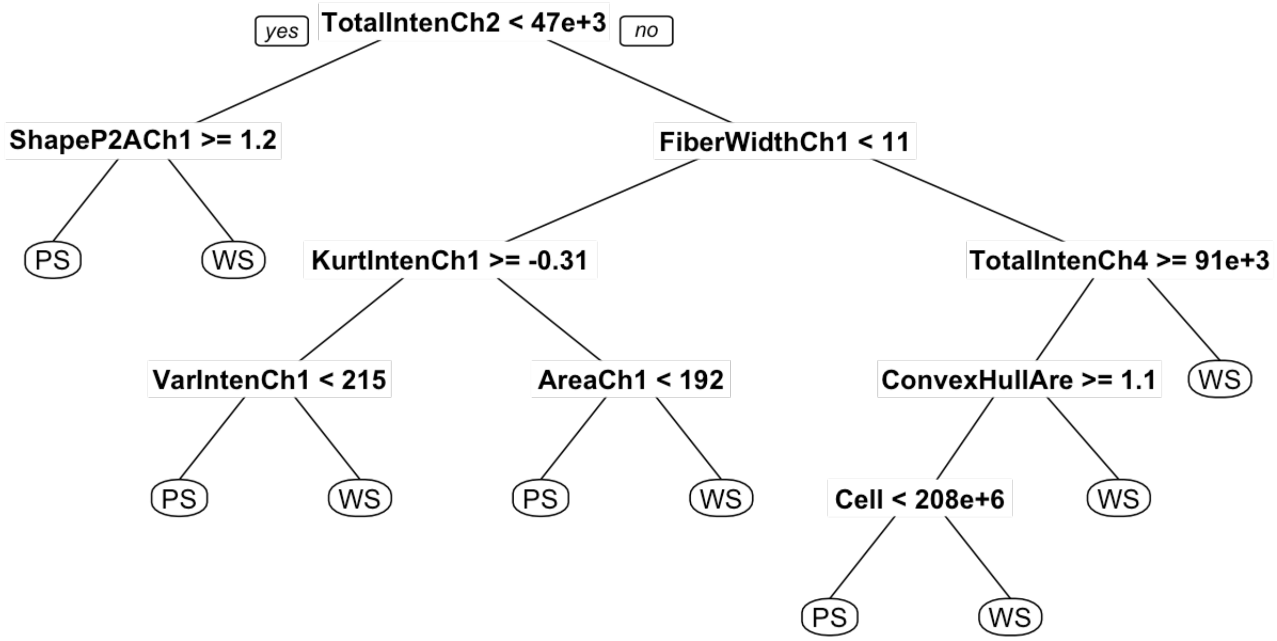
```
## Loading required package: lattice
## Loading required package: ggplot2
```

```
# 1. Subset the data to a training set and testing set based on the Case variable in the da
ta set.
inTrain <- createDataPartition(y=segmentationOriginal$Case, p=0.7, list=FALSE)
training <- segmentationOriginal[inTrain,]; testing <- segmentationOriginal[-inTrain,]
dim(training); dim(testing)
```

```
## [1] 1414  119
```

```
## [1] 605 119
```

```
# 2. Set the seed to 125 and fit a CART model with the rpart method using all predictor var
iables and default caret settings.
set.seed(125)
# fit classification tree to segmentation data
library(rpart)
segment_cart <- rpart(data=training, formula = Class~., method="class")
# print tree
# printcp(segment_cart)
# lotcp(segment_cart)
library(rpart.plot)
prp(segment_cart)
```

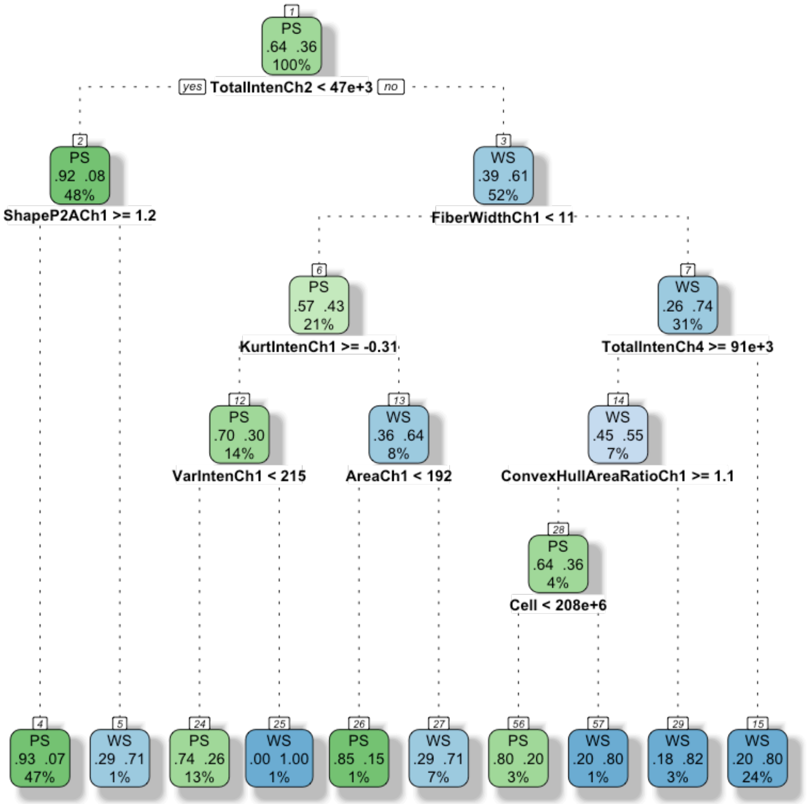


```
# summary(segment_cart)
```

```
library(rattle)
```

```
## Rattle: A free graphical interface for data mining with R.
## Version 3.4.1 Copyright (c) 2006-2014 Togaware Pty Ltd.
## Type 'rattle()' to shake, rattle, and roll your data.
```

```
fancyRpartPlot(segment_cart)
```



Rattle 2015-Jan-24 03:18:29 MKarp

```
# 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   PS
#  b. TotalIntench2 = 50,000; FiberWidthCh1 = 10;VarIntenCh4 = 100   WS
#  c. TotalIntench2 = 57,000; FiberWidthCh1 = 8;VarIntenCh4 = 100    PS
#  d. FiberWidthCh1 = 8;VarIntenCh4 = 100; PerimStatusCh1=2          No way to tell

# 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 the sample size.
# q3 - Load the olive oil data using the commands:
library(pgmm)
data(olive)
olive = olive[,-1]
# 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
olive_cart <- rpart(data= olive, formula = Area~.)
newdata = as.data.frame(t(colMeans(olive)))
# What is the resulting prediction? Is the resulting prediction strange? Why or why not?
olive_pred <- predict(olive_cart, newdata = newdata)
olive_pred
```

```
##     1
## 2.875
```

```
# 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 consum
ption, obesity levels,
# cumulative tabacco, type-A behavior, and low density lipoprotein cholesterol as predictor
s. Calculate the
# misclassification rate for your model using this function and a prediction on the "respon
se" scale:
# missClass = function(values,prediction){sum(((prediction > 0.5)*1) != values)/length(valu
es)}
# What is the misclassification rate on the training set? What is the misclassification rat
e on the test set?
set.seed(13234)
str(SAheart)
```

```
## 'data.frame':    462 obs. of  10 variables:
##  $ sbp      : int  160 144 118 170 134 132 142 114 114 132 ...
##  $ tobacco  : num  12 0.01 0.08 7.5 13.6 6.2 4.05 4.08 0 0 ...
##  $ ldl      : num  5.73 4.41 3.48 6.41 3.5 6.47 3.38 4.59 3.83 5.8 ...
##  $ adiposity: num  23.1 28.6 32.3 38 27.8 ...
##  $ famhist  : Factor w/ 2 levels "Absent","Present": 2 1 2 2 2 2 1 2 2 2 ...
##  $ typea    : int  49 55 52 51 60 62 59 62 49 69 ...
##  $ obesity  : num  25.3 28.9 29.1 32 26 ...
##  $ alcohol  : num  97.2 2.06 3.81 24.26 57.34 ...
##  $ age      : int  52 63 46 58 49 45 38 58 29 53 ...
##  $ chd      : int  1 1 0 1 1 0 0 1 0 1 ...
```

```
saheart_logistic <- train(data = trainSA, chd ~ age + alcohol + obesity + tobacco + typea +
ldl, method = "glm", family = "binomial")
missClass = function(values,prediction){sum(((prediction > 0.5)*1) != values)/length(values
)}
train_misclass <- missClass(trainSA$chd, predict(saheart_logistic, trainSA))
train_misclass
```

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

```
test_misclass <- missClass(testSA$chd, predict(saheart_logistic, testSA))
test_misclass
```

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

```
# 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 th
e seed to 33833. Fit a
# random forest predictor relating the factor variable y to the remaining variables. Read a
bout variable importance in
# random forests here: http://www.stat.berkeley.edu/~breiman/RandomForests/cc_home.htm#oobe
rr The caret package uses by
# defualt the Gini importance. Calculate the variable importance using the varImp function
in the caret package. What
# is the order of variable importance?
set.seed(33833)
vowel.train$y <- as.factor(vowel.train$y)
vowel.test$y <- as.factor(vowel.test$y)
str(vowel.train)
```

```
## 'data.frame':    528 obs. of  11 variables:
##  $ y   : Factor w/ 11 levels "1","2","3","4",..: 1 2 3 4 5 6 7 8 9 10 ...
##  $ x.1 : num  -3.64 -3.33 -2.12 -2.29 -2.6 ...
##  $ x.2 : num  0.418 0.496 0.894 1.809 1.938 ...
##  $ x.3 : num  -0.67 -0.694 -1.576 -1.498 -0.846 ...
##  $ x.4 : num  1.779 1.365 0.147 1.012 1.062 ...
##  $ x.5 : num  -0.168 -0.265 -0.707 -1.053 -1.633 ...
##  $ x.6 : num  1.627 1.933 1.559 1.06 0.764 ...
##  $ x.7 : num  -0.388 -0.363 -0.579 -0.567 0.394 0.217 0.322 -0.435 -0.512 -0.466 ...
##  $ x.8 : num  0.529 0.51 0.676 0.235 -0.15 -0.246 0.45 0.992 0.928 0.702 ...
##  $ x.9 : num  -0.874 -0.621 -0.809 -0.091 0.277 0.238 0.377 0.575 -0.167 0.06 ...
##  $ x.10: num  -0.814 -0.488 -0.049 -0.795 -0.396 -0.365 -0.366 -0.301 -0.434 -0.836 ...
```

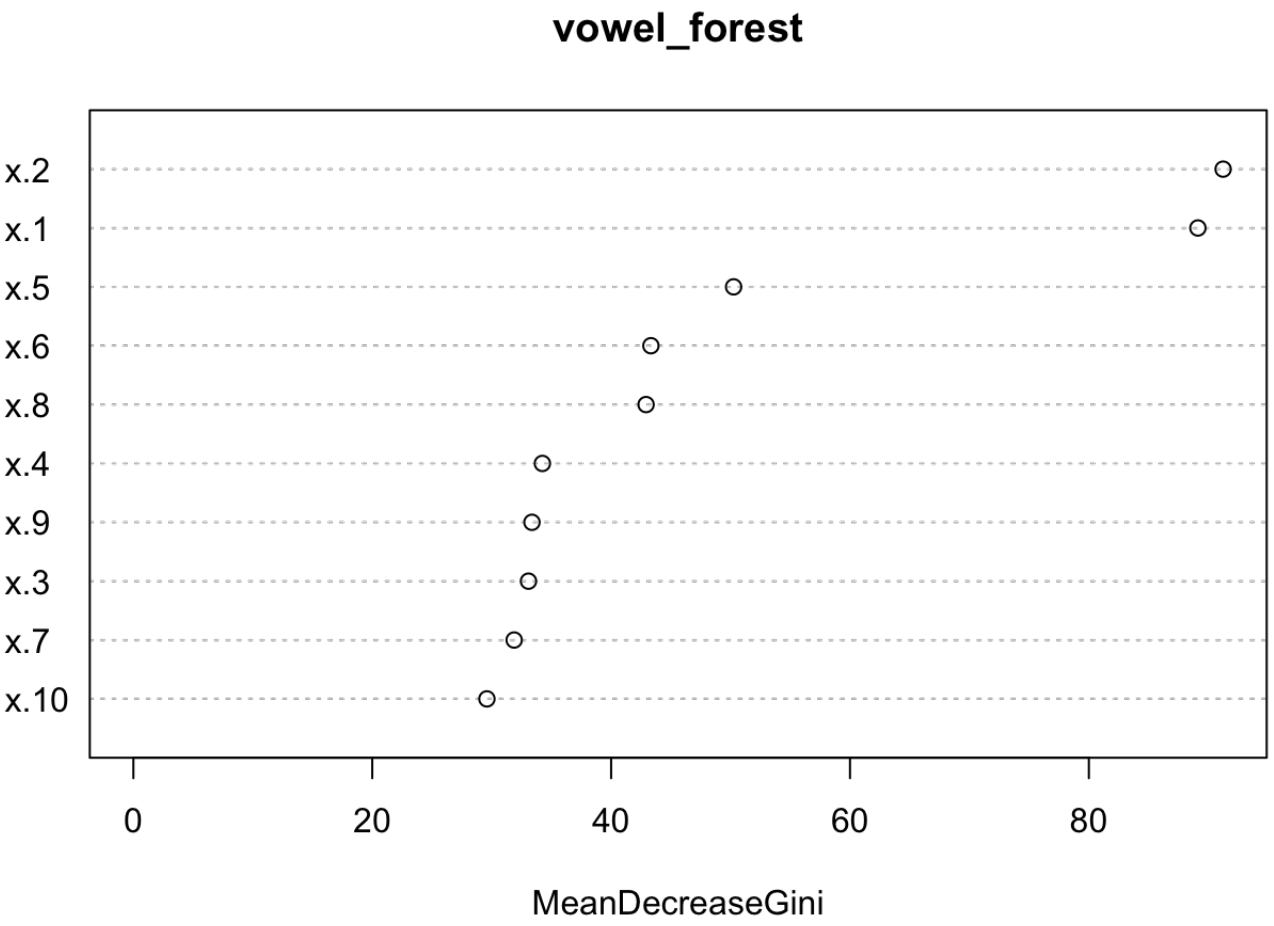
```
library(randomForest)
```

```
## randomForest 4.6-10
## Type rfNews() to see new features/changes/bug fixes.
```

```
vowel_forest <- randomForest(data = vowel.train, y~.)
varImp(vowel_forest)
```

```
##       Overall
## x.1  89.12864
## x.2  91.24009
## x.3  33.08111
## x.4  34.24433
## x.5  50.25539
## x.6  43.33148
## x.7  31.88132
## x.8  42.92470
## x.9  33.37031
## x.10 29.59956
```

```
varImpPlot(vowel_forest)
```


vowel_forest
x.2
x.1
x.5
x.6
x.8
x.4
x.9
x.3
x.7
x.10
MeanDecreaseGini