ML

Tom Reilly

Saturday, November 15, 2014

This is an R Markdown document. Markdown is a simple formatting syntax for authoring HTML, PDF, and MS Word documents. For more details on using R Markdown see http://rmarkdown.rstudio.com.

When you click the **Knit** button a document will be generated that includes both content as well as the output of any embedded R code chunks within the document. You can embed an R code chunk like this:

```
#Read files into R
Train <- read.csv("c:/rWork/pml-training.csv", header=T,
                  stringsAsFactors = FALSE)
Test <- read.csv("c:/rWork/pml-testing.csv", header=T,
                 stringsAsFactors = FALSE)
#Set options to no scientific notations and four digits
options(scipen=999, digits=4)
#Activate caret and rpart libraries
library("caret", lib.loc="~/R/win-library/3.1")
```

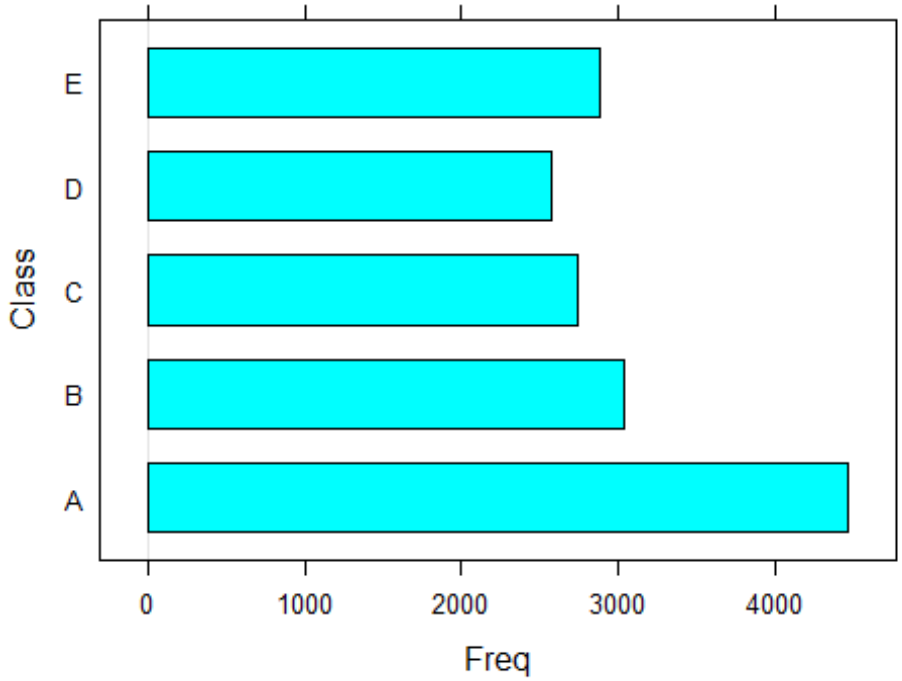
```
## Loading required package: lattice
## Loading required package: ggplot2
```

```
library("rpart", lib.loc="C:/Program Files/R/R-3.1.1/library")
#Separate Training and Test data
inTrain <- createDataPartition(y=Train$classe, p=.8, list=FALSE)
training <- Train[inTrain,]
trainingTest <- Train[-inTrain,]
dim(training); dim(trainingTest)
```

```
## [1] 15699   160
```

```
## [1] 3923  160
```

```
#display the classe variable graphically
barchart(training$classe, horiz=TRUE, ylab="Class")
```




```
#Fit classe to selected variables
```

You can also embed plots, for example:

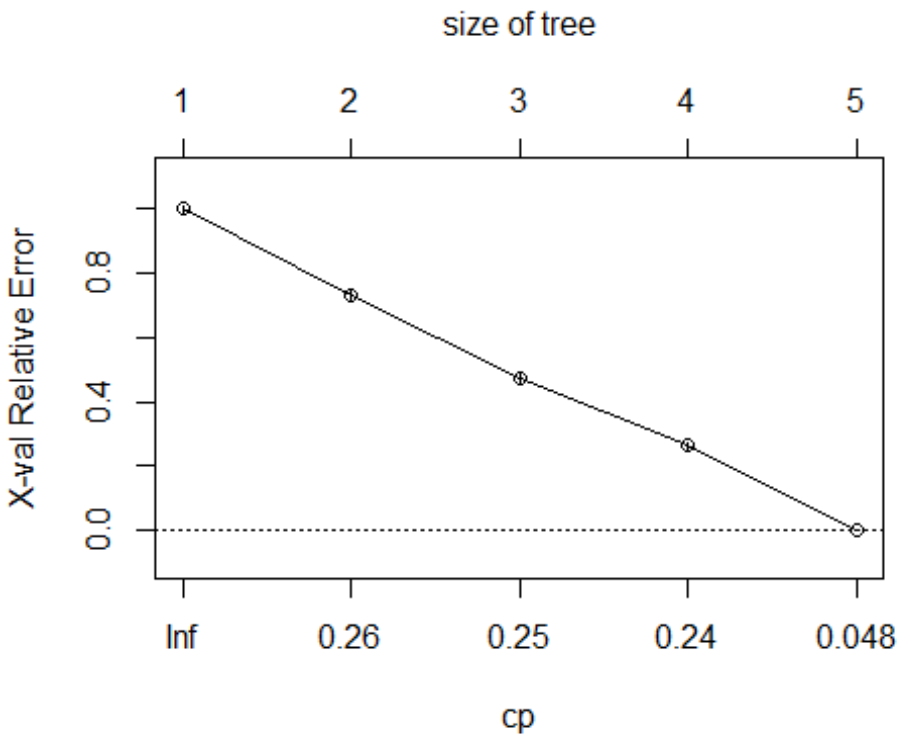
```
modFit <- rpart(classe~X+accel_arm_z+cvtd_timestamp+gyros_dumbbell_y+
                    magnet_belt_y+total_accel_forearm, method="class",
                data=trainingTest)
```

```r
#Predict the fitted rpart object
```

```r
pred <- predict(modFit, type = "prob")
# display the results
prcp <- printcp(modFit)
```

```
##
## Classification tree:
## rpart(formula = classe ~ X + accel_arm_z + cvtd_timestamp +
gyros_dumbbell_y +
##     magnet_belt_y + total_accel_forearm, data = trainingTest,
##     method = "class")
##
## Variables actually used in tree construction:
## [1] X
##
## Root node error: 2807/3923 = 0.72
##
## n= 3923
##
##     CP nsplit rel error  xerror    xstd
## 1 0.27      0      1.00 1.00000 0.01007
## 2 0.26      1      0.73 0.72960 0.01115
## 3 0.24      2      0.47 0.47275 0.01056
## 4 0.23      3      0.23 0.26185 0.00871
## 5 0.01      4      0.00 0.00036 0.00036
```

```r
# visualize cross-validation results
plcp <- plotcp(modFit)
```




```r
#detailed summary of splits
summ <- summary(modFit)
```

```
## Call:
## rpart(formula = classe ~ X + accel_arm_z + cvtd_timestamp +
gyros_dumbbell_y +
##     magnet_belt_y + total_accel_forearm, data = trainingTest,
##     method = "class")
##   n= 3923
##
##       CP nsplit rel error    xerror      xstd
## 1 0.2704      0    1.0000 1.0000000 0.0100670
## 2 0.2569      1    0.7296 0.7296046 0.0111459
## 3 0.2437      2    0.4727 0.4727467 0.0105569
## 4 0.2291      3    0.2291 0.2618454 0.0087067
## 5 0.0100      4    0.0000 0.0003563 0.0003562
##
## Variable importance
##                  X      cvtd_timestamp       magnet_belt_y
##                 58                  29                   6
##    gyros_dumbbell_y total_accel_forearm         accel_arm_z
##                  4                   2                   1
##
## Node number 1: 3923 observations,    complexity param=0.2704
```

```
##   predicted class=A  expected loss=0.7155  P(node) =1
##     class counts:  1116   759   684   643   721
##    probabilities: 0.284 0.193 0.174 0.164 0.184
##   left son=2 (1116 obs) right son=3 (2807 obs)
##   Primary splits:
##       X                < 5578   to the left,  improve=998.90, (0 missing)
##       cvtd_timestamp   splits as  LLLRLLRRLLRLLRLLRLLR, improve=606.10, (0
missing)
##       magnet_belt_y    < 554.5  to the right, improve=189.10, (0 missing)
##       gyros_dumbbell_y < 0.57   to the left,  improve= 94.19, (0 missing)
##       accel_arm_z      < -217.5 to the right, improve= 37.49, (0 missing)
##   Surrogate splits:
##       cvtd_timestamp      splits as  LLRRLLRRLRRLRRLRRLRR, agree=0.879,
adj=0.573, (0 split)
##       total_accel_forearm < 6      to the left,  agree=0.729, adj=0.047,
(0 split)
##
## Node number 2: 1116 observations
##   predicted class=A  expected loss=0  P(node) =0.2845
##     class counts:  1116     0     0     0     0
##    probabilities: 1.000 0.000 0.000 0.000 0.000
##
## Node number 3: 2807 observations,    complexity param=0.2569
##   predicted class=B  expected loss=0.7296  P(node) =0.7155
##     class counts:     0   759   684   643   721
##    probabilities: 0.000 0.270 0.244 0.229 0.257
##   left son=6 (759 obs) right son=7 (2048 obs)
##   Primary splits:
##       X                < 9380   to the left,  improve=738.80, (0 missing)
##       cvtd_timestamp   splits as  -LLR-LRR-LR-LR-LR-LR, improve=497.10, (0
missing)
##       magnet_belt_y    < 555    to the right, improve=161.40, (0 missing)
##       gyros_dumbbell_y < 0.57   to the left,  improve= 64.09, (0 missing)
##       accel_arm_z      < -262.5 to the left,  improve= 16.77, (0 missing)
##   Surrogate splits:
##       cvtd_timestamp      splits as  -LRR-LRR-LR-LR-RR-RR, agree=0.847,
adj=0.433, (0 split)
##       gyros_dumbbell_y    < -0.555 to the left,  agree=0.735, adj=0.021,
(0 split)
##       total_accel_forearm < 11.5   to the left,  agree=0.732, adj=0.008,
(0 split)
##       accel_arm_z         < -589   to the left,  agree=0.730, adj=0.001,
(0 split)
##
## Node number 6: 759 observations
##   predicted class=B  expected loss=0  P(node) =0.1935
##     class counts:     0   759     0     0     0
##    probabilities: 0.000 1.000 0.000 0.000 0.000
##
## Node number 7: 2048 observations,    complexity param=0.2437
```

```
##   predicted class=E  expected loss=0.6479  P(node) =0.522
##     class counts:     0     0   684   643   721
##    probabilities: 0.000 0.000 0.334 0.314 0.352
##   left son=14 (1327 obs) right son=15 (721 obs)
##   Primary splits:
##       X                < 16020  to the left,  improve=701.00, (0 missing)
##       cvtd_timestamp   splits as  --LR--LR-LR-LR-LR-LR, improve=373.30, (0
missing)
##       magnet_belt_y    < 580.5  to the right, improve=168.20, (0 missing)
##       gyros_dumbbell_y < 0.57   to the left,  improve= 86.72, (0 missing)
##       accel_arm_z      < -262.5 to the right, improve= 16.07, (0 missing)
##   Surrogate splits:
##       cvtd_timestamp      splits as  --LR--LR-LL-LR-LR-LL, agree=0.777,
adj=0.366, (0 split)
##       magnet_belt_y       < 580.5  to the right, agree=0.774, adj=0.358,
(0 split)
##       gyros_dumbbell_y    < 0.57   to the left,  agree=0.718, adj=0.198,
(0 split)
##       accel_arm_z         < -262.5 to the right, agree=0.662, adj=0.039,
(0 split)
##       total_accel_forearm < 52.5   to the left,  agree=0.659, adj=0.032,
(0 split)
##
## Node number 14: 1327 observations,    complexity param=0.2291
##   predicted class=C  expected loss=0.4846  P(node) =0.3383
##     class counts:     0     0   684   643     0
##    probabilities: 0.000 0.000 0.515 0.485 0.000
##   left son=28 (684 obs) right son=29 (643 obs)
##   Primary splits:
##       X                < 12800  to the left,  improve=662.90, (0 missing)
##       cvtd_timestamp   splits as  --LR--R--LR-LR-LR-LR, improve=223.90, (0
missing)
##       magnet_belt_y    < 555    to the right, improve= 23.61, (0 missing)
##       gyros_dumbbell_y < 0.49   to the left,  improve= 16.90, (0 missing)
##       accel_arm_z      < 157.5  to the left,  improve= 11.28, (0 missing)
##   Surrogate splits:
##       cvtd_timestamp      splits as  --LR--R--LR-LR-LR-LR, agree=0.787,
adj=0.561, (0 split)
##       gyros_dumbbell_y    < -0.025 to the right, agree=0.560, adj=0.092,
(0 split)
##       magnet_belt_y       < 602.5  to the left,  agree=0.556, adj=0.084,
(0 split)
##       total_accel_forearm < 46.5   to the left,  agree=0.545, adj=0.061,
(0 split)
##       accel_arm_z         < -105.5 to the right, agree=0.537, adj=0.044,
(0 split)
##
## Node number 15: 721 observations
##   predicted class=E  expected loss=0  P(node) =0.1838
##     class counts:     0     0     0     0   721
```

```
##    probabilities: 0.000 0.000 0.000 0.000 1.000
##
## Node number 28: 684 observations
##   predicted class=C  expected loss=0  P(node) =0.1744
##     class counts:     0     0   684     0     0
##    probabilities: 0.000 0.000 1.000 0.000 0.000
##
## Node number 29: 643 observations
##   predicted class=D  expected loss=0  P(node) =0.1639
##     class counts:     0     0     0   643     0
##    probabilities: 0.000 0.000 0.000 1.000 0.000
```

Note that the `echo = FALSE` parameter was added to the code chunk to prevent printing of the R code that generated the plot.