Practical Machine Learning Project

Yunpeng Zhan

Augest 05, 2018

Scenario

Using devices such as Jawbone Up, Nike FuelBand, and Fitbit it is now possible to collect a large amount of data about personal activity relatively inexpensively. These type of devices are part of the quantified self movement - a group of enthusiasts who take measurements about themselves regularly to improve their health, to find patterns in their behavior, or because they are tech geeks. One thing that people regularly do is quantify how much of a particular activity they do, but they rarely quantify how well they do it. In this project, your goal will be to use data from accelerometers on the belt, forearm, arm, and dumbell of 6 participants. They were asked to perform barbell lifts correctly and incorrectly in 5 different ways. **The goal of the project is to predict the manner in which they did the exercise.**

Data

The training data for this project are available here: [https://d396qusza40orc.cloudfront.net/predmachlearn/pml-training.csv]

The test data are available here: [https://d396qusza40orc.cloudfront.net/predmachlearn/pml-testing.csv]

Code and resutls

Load required library and set seed.

```
library(caret)
```

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

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

```
library(rpart)
library(rpart.plot)
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:ggplot2':
##
##     margin
```

```
set.seed(1234)
```

Load datasets and premary cleaning

```
trainingurl <- "https://d396qusza40orc.cloudfront.net/predmachlearn/pml-training.csv"
testingurl <- "https://d396qusza40orc.cloudfront.net/predmachlearn/pml-testing.csv"
if(!file.exists("data")) {dir.create("data")}
download.file(trainingurl, destfile = "./data/pml-training.csv")
download.file (testingurl, destfile = "./data/pml-testing.csv")
trainingdata <- read.csv("./data/pml-training.csv", na.strings=c("NA","#DIV/0!", ""))
testdata <- read.csv("./data/pml-testing.csv", na.strings=c("NA","#DIV/0!", ""))

# Remove columns that are all Nulls

trainingdata<-trainingdata[,colSums(is.na(trainingdata)) == 0]
testdata <-testdata[,colSums(is.na(testdata)) == 0]

# Remove unrelated variables

trainingdata <- trainingdata[,-c(1:7)]
testdata <-testdata[,-c(1:7)]
```

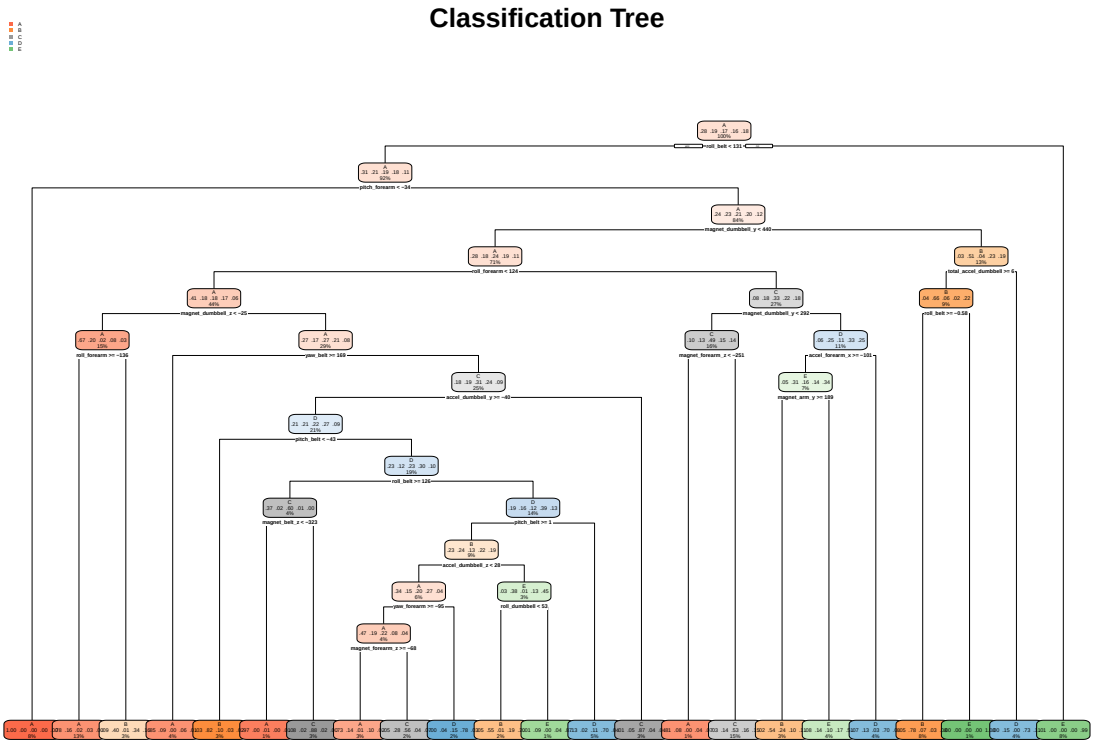
Partitioning the training data set to allow cross-validation

```
subs <- createDataPartition(y=trainingdata$classe, p=0.75, list=FALSE)
subtraining <- trainingdata[subs, ]
subtesting <- trainingdata[-subs, ]
```

First model – Decision Tree

```
model <- rpart(classe ~ ., data=subtraining, method="class")
rpart.plot(model, main="Classification Tree")
```

```
## Warning: labs do not fit even at cex 0.15, there may be some overplotting
```




```
prediction1 <- predict(model, subtesting, type = "class")
confusionMatrix(prediction1, subtesting$classe)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction    A    B    C    D    E
##          A 1235  157   16   50   20
##          B   55  568   73   80  102
##          C   44  125  690  118  116
##          D   41   64   50  508   38
##          E   20   35   26   48  625
##
## Overall Statistics
##
##                Accuracy : 0.7394
##                  95% CI : (0.7269, 0.7516)
##     No Information Rate : 0.2845
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.6697
##  Mcnemar's Test P-Value : < 2.2e-16
##
## Statistics by Class:
##
##                      Class: A Class: B Class: C Class: D Class: E
## Sensitivity            0.8853   0.5985   0.8070   0.6318   0.6937
## Specificity            0.9307   0.9216   0.9005   0.9529   0.9678
## Pos Pred Value         0.8356   0.6469   0.6313   0.7247   0.8289
## Neg Pred Value         0.9533   0.9054   0.9567   0.9296   0.9335
```

```
## Prevalence             0.2845   0.1935   0.1743   0.1639   0.1837
## Detection Rate         0.2518   0.1158   0.1407   0.1036   0.1274
## Detection Prevalence   0.3014   0.1790   0.2229   0.1429   0.1538
## Balanced Accuracy      0.9080   0.7601   0.8537   0.7924   0.8307
```

Second Model – Random Forest

```
model2 <- randomForest(classe ~. , data=subtraining, method="class")
prediction2 <- predict(model2, subtesting, type = "class")
confusionMatrix(prediction2, subtesting$classe)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction    A    B    C    D    E
##          A 1395    3    0    0    0
##          B    0  943   10    0    0
##          C    0    3  844    5    0
##          D    0    0    1  799    0
##          E    0    0    0    0  901
##
## Overall Statistics
##
##                Accuracy : 0.9955
##                  95% CI : (0.9932, 0.9972)
##     No Information Rate : 0.2845
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.9943
##  Mcnemar's Test P-Value : NA
##
## Statistics by Class:
##
##                      Class: A Class: B Class: C Class: D Class: E
## Sensitivity            1.0000   0.9937   0.9871   0.9938   1.0000
## Specificity            0.9991   0.9975   0.9980   0.9998   1.0000
## Pos Pred Value         0.9979   0.9895   0.9906   0.9988   1.0000
## Neg Pred Value         1.0000   0.9985   0.9973   0.9988   1.0000
## Prevalence             0.2845   0.1935   0.1743   0.1639   0.1837
## Detection Rate         0.2845   0.1923   0.1721   0.1629   0.1837
## Detection Prevalence   0.2851   0.1943   0.1737   0.1631   0.1837
## Balanced Accuracy      0.9996   0.9956   0.9926   0.9968   1.0000
```

From the results, it is obvious that Random Forest model performed better whose accuracy is 0.995 while the other one is 0,727.

Make prediction based on the Random Forest model

```
predictfinal <- predict(model2, testdata, type="class")
predictfinal
```

```
##  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20
```

```
##  B  A  B  A  A  E  D  B  A  A  B  C  B  A  E  E  A  B  B  B
## Levels: A B C D E
```