Practical Machine Learning

GHyee

Sunday, February 08, 2015

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. In this project, the goal is to use data from accelerometers on the belt, forearm, arm, and dumbell of 6 participants to predict how well they perform barbell lifts. More information is available from the website here: <http://groupware.les.inf.puc-rio.br/har> (see the section on the Weight Lifting Exercise Dataset).

# Data Processing

The data source comes from <http://groupware.les.inf.puc-rio.br/har> and are already split into training and testing set beforehand. The training dataset can be downloaded from <https://d396qusza40orc.cloudfront.net/predmachlearn/pml-training.csv> while the testing dataset can be downloaded from <https://d396qusza40orc.cloudfront.net/predmachlearn/pml-testing.csv>. The data are processed after some exploratory analysis before models are fitted.

## Warning: package 'caret' was built under R version 3.1.2

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

## Warning: package 'rpart.plot' was built under R version 3.1.2  
## Warning: package 'rattle' was built under R version 3.1.2

## 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.

## Warning: package 'randomForest' was built under R version 3.1.2

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

## [1] 11776 160

## [1] 7846 160

## [1] 11776 100

## [1] 11776 58

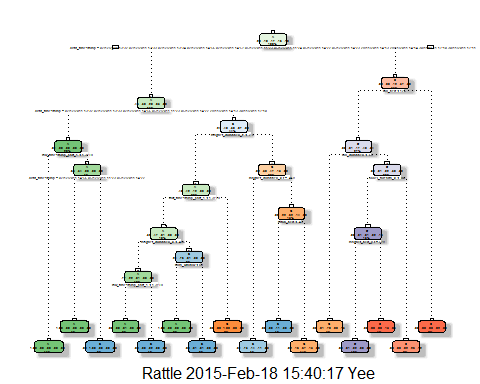
## [1] 7846 58

## [1] 20 57

# Model Fitting

Two machine learning method, namely the decision tree and Random forest, will be used to fit the model. The method with the better accuracy will be chosen. Decision tree is fitted using the rpart function from the rpart library. Random forest is fitted using the randomForest function from the randomForest package.

modFitA1 <- rpart(classe ~ ., data=myTraining, method="class")  
  
fancyRpartPlot(modFitA1)



predictionsA1 <- predict(modFitA1, myTesting, type = "class")  
  
confusionMatrix(predictionsA1, myTesting$classe)

## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction A B C D E  
## A 2152 70 8 2 0  
## B 59 1260 71 67 0  
## C 21 180 1275 198 62  
## D 0 8 10 801 78  
## E 0 0 4 218 1302  
##   
## Overall Statistics  
##   
## Accuracy : 0.865   
## 95% CI : (0.858, 0.873)  
## No Information Rate : 0.284   
## P-Value [Acc > NIR] : <2e-16   
##   
## Kappa : 0.83   
## Mcnemar's Test P-Value : NA   
##   
## Statistics by Class:  
##   
## Class: A Class: B Class: C Class: D Class: E  
## Sensitivity 0.964 0.830 0.932 0.623 0.903  
## Specificity 0.986 0.969 0.929 0.985 0.965  
## Pos Pred Value 0.964 0.865 0.734 0.893 0.854  
## Neg Pred Value 0.986 0.960 0.985 0.930 0.978  
## Prevalence 0.284 0.193 0.174 0.164 0.184  
## Detection Rate 0.274 0.161 0.163 0.102 0.166  
## Detection Prevalence 0.284 0.186 0.221 0.114 0.194  
## Balanced Accuracy 0.975 0.899 0.930 0.804 0.934

modFitB1 <- randomForest(classe ~. , data=myTraining)  
  
predictionsB1 <- predict(modFitB1, myTesting, type = "class")  
  
confusionMatrix(predictionsB1, myTesting$classe)

## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction A B C D E  
## A 2232 1 0 0 0  
## B 0 1517 1 0 0  
## C 0 0 1366 6 0  
## D 0 0 1 1280 4  
## E 0 0 0 0 1438  
##   
## Overall Statistics  
##   
## Accuracy : 0.998   
## 95% CI : (0.997, 0.999)  
## No Information Rate : 0.284   
## P-Value [Acc > NIR] : <2e-16   
##   
## Kappa : 0.998   
## Mcnemar's Test P-Value : NA   
##   
## Statistics by Class:  
##   
## Class: A Class: B Class: C Class: D Class: E  
## Sensitivity 1.000 0.999 0.999 0.995 0.997  
## Specificity 1.000 1.000 0.999 0.999 1.000  
## Pos Pred Value 1.000 0.999 0.996 0.996 1.000  
## Neg Pred Value 1.000 1.000 1.000 0.999 0.999  
## Prevalence 0.284 0.193 0.174 0.164 0.184  
## Detection Rate 0.284 0.193 0.174 0.163 0.183  
## Detection Prevalence 0.285 0.193 0.175 0.164 0.183  
## Balanced Accuracy 1.000 1.000 0.999 0.997 0.999

predictionsB2 <- predict(modFitB1, testing, type = "class")

The second model, Random Forest, will be chosen as it yields a much better accuracy compared to the first.

The test data set is used to validate the accuracy of the Random forest model.

pml\_write\_files = function(x){  
 n = length(x)  
 for(i in 1:n){  
 filename = paste0("problem\_id\_",i,".txt")  
 write.table(x[i],file=filename,quote=FALSE,row.names=FALSE,col.names=FALSE)  
 }  
}  
pml\_write\_files(predictionsB2)

s