Prediction Assignment

Gitta Borghart 28-04-2020

## Introduction

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. 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).

The project can be found on github:

## Source of 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>

The data for this project come from this source: <http://groupware.les.inf.puc-rio.br/har>. If you use the document you create for this class for any purpose please cite them as they have been very generous in allowing their data to be used for this kind of assignment.

setwd("~/")  
  
# Load the training dataset  
Training <- read.csv("pml-training.csv", na.strings=c("NA",""), strip.white=T)  
  
# Load the testing dataset  
Testing <- read.csv("pml-testing.csv", na.strings=c("NA",""), strip.white=T)  
  
#Data cleaning  
features <- names(Testing[,colSums(is.na(Testing)) == 0])[8:59]  
  
# Only use features used in testing cases.  
Training <- Training[,c(features,"classe")]  
Testing <- Testing[,c(features,"problem\_id")]  
  
dim(Training)

## [1] 19622 53

dim(Testing)

## [1] 20 53

library(caret)

## Loading required package: lattice

## Loading required package: ggplot2

set.seed(12345)  
inTrain <- createDataPartition(Training$classe, p=0.60, list=FALSE)  
train1 <- Training[inTrain,]  
test1 <- Training[-inTrain,]  
  
dim(train1)

## [1] 11776 53

dim(test1)

## [1] 7846 53

#The model  
library(rpart)  
library(rpart.plot)

## Warning: package 'rpart.plot' was built under R version 3.6.2

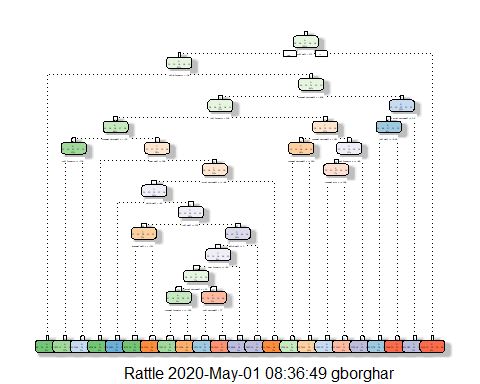
library(rattle)

## Warning: package 'rattle' was built under R version 3.6.3

## Rattle: A free graphical interface for data science with R.  
## Version 5.3.0 Copyright (c) 2006-2018 Togaware Pty Ltd.  
## Type 'rattle()' to shake, rattle, and roll your data.

modeltree <- rpart(classe ~ ., data = train1, method="class")  
fancyRpartPlot(modeltree)

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



#Predicting  
set.seed(12345)  
  
prediction <- predict(modeltree, test1, type = "class")  
confusionMatrix(prediction, test1$class)

## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction A B C D E  
## A 1995 246 49 83 51  
## B 75 890 111 119 115  
## C 44 198 1094 153 143  
## D 74 112 79 840 94  
## E 44 72 35 91 1039  
##   
## Overall Statistics  
##   
## Accuracy : 0.7466   
## 95% CI : (0.7368, 0.7562)  
## No Information Rate : 0.2845   
## P-Value [Acc > NIR] : < 2.2e-16   
##   
## Kappa : 0.6787   
##   
## Mcnemar's Test P-Value : < 2.2e-16   
##   
## Statistics by Class:  
##   
## Class: A Class: B Class: C Class: D Class: E  
## Sensitivity 0.8938 0.5863 0.7997 0.6532 0.7205  
## Specificity 0.9236 0.9336 0.9169 0.9453 0.9622  
## Pos Pred Value 0.8230 0.6794 0.6703 0.7006 0.8111  
## Neg Pred Value 0.9563 0.9039 0.9559 0.9329 0.9386  
## Prevalence 0.2845 0.1935 0.1744 0.1639 0.1838  
## Detection Rate 0.2543 0.1134 0.1394 0.1071 0.1324  
## Detection Prevalence 0.3089 0.1670 0.2080 0.1528 0.1633  
## Balanced Accuracy 0.9087 0.7600 0.8583 0.7992 0.8414

prediction2 <- predict(modeltree, Testing, type = "class")  
print(prediction2)

## 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20   
## C A B D A C D A A A C B A A E E A B B B   
## Levels: A B C D E

#Check for the Out of Sample Error calculation  
Check <- function(values, predicted) {  
 sum(predicted != values) / length(values)  
}  
OutofSample\_ErrorRate <- Check(test1$classe, prediction)  
print(OutofSample\_ErrorRate)

## [1] 0.2533775

Conclusion: Accuracy of prediction with the above data set is around 75%, which is quite alright. The out of sample error is 25%.