Course Project

Andrey ABRAMOV

28 february 2016

# Introduction

The goal of our Course Project is to predict the same manner like [Groupware@LES](mailto:Groupware@LES) in which they did the Human Activity Recognition process. Human Activity Recognition - has emerged as a key research area in the last years and is gaining increasing attention by the pervasive computing research community.

[Groupware@LES](mailto:Groupware@LES) proposed a dataset with 5 classes (sitting-down, standing-up, standing, walking, and sitting) collected on some hours of activities of 6 healthy subjects. This human activity recognition research has traditionally focused on discriminating between different activities, i.e. to predict "which" activity was performed at a specific point in time. The approach we propose for the Weight Lifting Exercises dataset is to investigate "how (well)" an activity was performed by the wearer.

In this work we will try to make the same steps and received the same result as [Groupware@LES](mailto:Groupware@LES).

# Data Source

More information of [Groupware@LES](mailto:Groupware@LES) result is available from the website here: <http://groupware.les.inf.puc-rio.br/har>. 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>

# Limitation

According to the description of the Course Project complete R code should not be provided with data describing the results of the calculations.

So letТs start step by step:

## Formulation of idea

So, in order to achieve the same result, as colleagues from [Groupware@LES](mailto:Groupware@LES) group, clean received training Assembly from NA and null columns. All designation of a classes we will transform to numeric values.

Take 10 appropriate methods for training our model on the training Assembly.

As we result we will received 10 different predictions results. Assume that all 10 results will be not Уthe bestФ and we will need to proceed one more training operation on the 10 result set. It will help us to reduce the variation in results and to improve the accuracy of the forecast.

## So let's check the idea.

### Step 1

Choose 10 right methods from the existing 180, clean learning Assembly from the zero column and start the learning process.

Load all neede libraryes

library(caret)  
library(ggplot2)

Convert and clean data set for trainig

# clean the data set  
adelmo <- adelmo[,-c(1:7)]  
  
# convert to text format  
for (i in 1:ncol(adelmo)) {adelmo[,i] <- as.character(adelmo[,i])}  
# find all DIV/0! and replace by 0  
for (i in 1:ncol(adelmo)) {  
 adelmo[which(adelmo[,i]=="#DIV/0!"),i] <- 0  
 adelmo[which(adelmo[,i]==NA),i] <- 0  
}  
# all columns to number format  
for (i in 1:(ncol(adelmo)-1)) {adelmo[,i] <- as.numeric(adelmo[,i])}  
adelmo[,ncol(adelmo)] <- as.factor(adelmo[,ncol(adelmo)])  
  
for (i in 6:ncol(adelmo)-1) {  
 m <- is.na(adelmo[,i])  
 adelmo[m,i] <- 0  
}

Find and select 10

# find all possible methods for training  
possible.methods <-names(getModelInfo())  
method.numbers <- c(13,23,30,38,49,55,56,57,63,68)

I was surprised that on three PC sets of the methods are different.

### Step 2

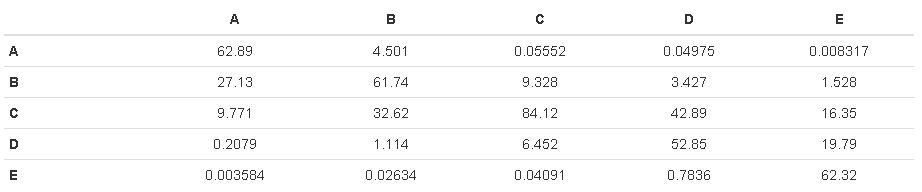
The dataset cleaned and ready for training

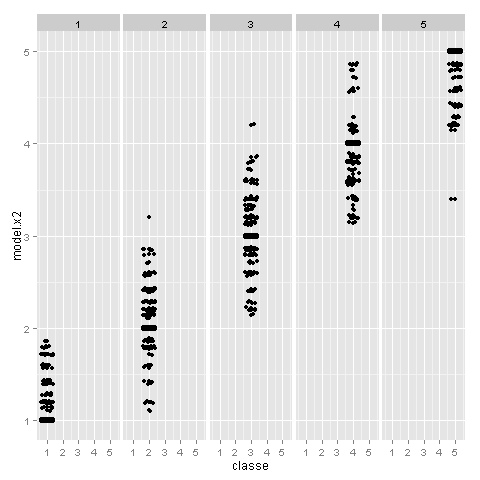
for (i in 1:6) {  
 cl <- makeCluster(detectCores())  
 registerDoParallel(cl) # Register workers (cores)  
 model.x <- train(classe~., data=adelmo, method=possible.methods[method.numbers[i]])  
 stopCluster(cl) # Closing parallel processing session  
 saveRDS(model.x, paste0("Model.", as.character(method.numbers[i]), "-", personal.name[j],".Rev.1.rds"))  
}

### Step 3

Lets find the prediction for the training dataset and compare it to existing classes

x <- predict(model.x, adelmo)  
 x <- as.data.frame(x)  
 names(x) <- c(Model.Name)  
 adelmo <- cbind(adelmo, x)

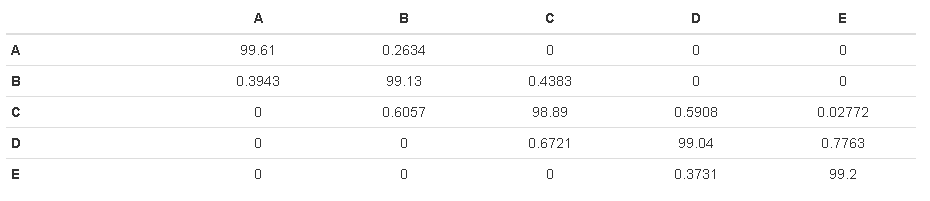
As the result we are receiving the next picture. 

All results are given in percent. And the distribution of the prediction versus the existing class showed below: 

### Step 4

As expected before, the range of models results is quite strong. Now apply the training results received by the method "knn" (as an option).

adelmo <- cbind(adelmo, classe)  
 model.x2 <- train(classe~Model.2+Model.4+Model.5+Model.6+Model.7+Model.8+Model.9+Model.10+  
 Model.11+Model.12, data =adelmo , method="knn")  
 #save common model for prediction result dataset  
 saveRDS(model.x2, paste0("Model.Common.", "-", personal.name[j],".Rev.1.rds"))  
 adelmo$model.x2 <- predict(model.x2, adelmo)  
 #save all data to file  
 saveRDS(adelmo, paste0("B.", "-", personal.name[j],".Rev.1.rds"))

And the result of our final model is the next. 

### Step 5

Lets check the result model on the testing dataset.

model.x <- readRDS(file = paste0("Model.",as.character(i),"-",personal.name[j],".Rev.1.rds"))  
 Model.Name <- paste0("Model.", as.character(i))  
 x <- predict(model.x, adelmo)  
 x <- as.data.frame(x)  
 names(x) <- c(Model.Name)  
 adelmo <- cbind(adelmo, x)

Comparing to the latest Course Quiz: B A B A A E D B A A B C B A E E A B B B 100% hit the target without deviation!

## Conclusions

As a result of applying re-training results improved significantly. I honestly didn't expect that a training example can reach a very good result with a probability of 99%++. Thus it was possible to follow by the same path as the [Groupware@LES](mailto:Groupware@LES) group, and get a decent result.

Special note is the using of multi-threaded calculations. Without using this package, the result would have been much later.

Thanks a lot for your time!