Predicting “How well we do barbell lifts?” using data from Sports Devices

Sebastián Fuenzalida Garcés
August 2015

Executive Summary

Load Data

First we need to read the files (included in the repo). After an analysis of the data, to avoid problems with missing data I decided to consider as NA’s: [“NA”, “#DIV/0!”, “”], considering them as moments when the user didn’t do anything or problems with the devices gathering the data.

```
library(AppliedPredictiveModeling); library(caret); library(rattle); library(randomForest); library(doPa
registerDoParallel(cores=2)

train_data<-read.csv("pml-training.csv",header=TRUE,na.strings=c("NA","#DIV/0!",""))
test_data<-read.csv("pml-testing.csv",header=TRUE,na.strings=c("NA","#DIV/0!",""))
```

Relevant Data

Looking at the data we can classify the columns in 3 categories: -Useful Data: Variables that we will use in the model -Not Useful Data: Variables that don’t have any relation with the classe -Data with too many NA’s: variables where more than 80% of the rows are NA’s so we are not going to include them in the model

Considering that we modify the train and test data to have only the useful columns:

```
#First 7 columns aren't useful for the model (name, time, window, etc...)
train_data1<-train_data[,8:length(train_data)]
test_data1<-test_data[,8:length(test_data)]

#NearZeroVar gives us a first approach of columns that we don't need
nsv<-nearZeroVar(train_data1,saveMetrics=TRUE)

#We exclude the columns that NZV is TRUE (almost all of them are 0 or NA)
train_data2<-train_data1[,-which(names(train_data1) %in% row.names(nsv[nsv$nzv==TRUE,]))]
test_data2<-test_data1[,-which(names(test_data1) %in% row.names(nsv[nsv$nzv==TRUE,]))]

dim(train_data2)
```

```
## [1] 19622   118
```

We see that we still have 118 columns, so we are going to do a deeper selection of variables. We are going to find the variables that have some correlation between them using what we learn in Lecture “Preprocessing with PCA”, but using a low correlation (10%) to just find avoid the variables that have a lot of NA’s and weren’t find by the nearZeroVar function.

```r
#We exclude the Classe column
M<-abs(cor(train_data2[,-118])); diag(M)<-0
useful_var<-unique(row.names(which(M>0.1,arr.ind=T)))
useful_var<-c(useful_var,"classe")

#Now we subset the train and test data using this columns
vars<-names(train_data2) %in% useful_var
train_data3<-train_data2[vars]
test_data3<-test_data2[vars]

#Just to be sure we check if both data sets have the same columns
all.equal(names(train_data3),names(test_data3))
```

```
## [1] "1 string mismatch"
```

We finally get 53 columns that are going to be part of our model, and the only different column is “classe” in train and “problem_id” in test.

Model Building