Practical Machine Learning Course Project

Lotte Sluyser

16 september 2017

# Introduction to the project

In this project the goal was to use data from accelerometers on the belt, forearm, arm and dumbell to quantify how well 6 participants do a particular activity.They were asked to perform barbell lifts correctly and incorrectly in 5 different ways. Goal was to predict the manner in which they did the exercise, the "classe" variable in the training set. This report describes how the model was built, how cross validation was used and the expected out of sample error. The model was used to predict 20 different test cases.

fileUrl<- "https://d396qusza40orc.cloudfront.net/predmachlearn/pml-training.csv"  
download.file(fileUrl,destfile="train.csv", method="curl")  
training = read.csv("~/Desktop/coursera/Datascience\_cursus\_8/train.csv")  
fileUrl2<- "https://d396qusza40orc.cloudfront.net/predmachlearn/pml-testing.csv"  
download.file(fileUrl2,destfile="test.csv", method="curl")  
testing = read.csv("~/Desktop/coursera/Datascience\_cursus\_8/test.csv")  
dim(training);dim(testing)

## [1] 19622 160

## [1] 20 160

## Data processing

Checking how many missing values (na's) there are per column:

sapply(training, function(x) sum(is.na(x)))

## X user\_name raw\_timestamp\_part\_1   
## 0 0 0   
## raw\_timestamp\_part\_2 cvtd\_timestamp new\_window   
## 0 0 0   
## num\_window roll\_belt pitch\_belt   
## 0 0 0   
## yaw\_belt total\_accel\_belt kurtosis\_roll\_belt   
## 0 0 0   
## kurtosis\_picth\_belt kurtosis\_yaw\_belt skewness\_roll\_belt   
## 0 0 0   
## skewness\_roll\_belt.1 skewness\_yaw\_belt max\_roll\_belt   
## 0 0 19216   
## max\_picth\_belt max\_yaw\_belt min\_roll\_belt   
## 19216 0 19216   
## min\_pitch\_belt min\_yaw\_belt amplitude\_roll\_belt   
## 19216 0 19216   
## amplitude\_pitch\_belt amplitude\_yaw\_belt var\_total\_accel\_belt   
## 19216 0 19216   
## avg\_roll\_belt stddev\_roll\_belt var\_roll\_belt   
## 19216 19216 19216   
## avg\_pitch\_belt stddev\_pitch\_belt var\_pitch\_belt   
## 19216 19216 19216   
## avg\_yaw\_belt stddev\_yaw\_belt var\_yaw\_belt   
## 19216 19216 19216   
## gyros\_belt\_x gyros\_belt\_y gyros\_belt\_z   
## 0 0 0   
## accel\_belt\_x accel\_belt\_y accel\_belt\_z   
## 0 0 0   
## magnet\_belt\_x magnet\_belt\_y magnet\_belt\_z   
## 0 0 0   
## roll\_arm pitch\_arm yaw\_arm   
## 0 0 0   
## total\_accel\_arm var\_accel\_arm avg\_roll\_arm   
## 0 19216 19216   
## stddev\_roll\_arm var\_roll\_arm avg\_pitch\_arm   
## 19216 19216 19216   
## stddev\_pitch\_arm var\_pitch\_arm avg\_yaw\_arm   
## 19216 19216 19216   
## stddev\_yaw\_arm var\_yaw\_arm gyros\_arm\_x   
## 19216 19216 0   
## gyros\_arm\_y gyros\_arm\_z accel\_arm\_x   
## 0 0 0   
## accel\_arm\_y accel\_arm\_z magnet\_arm\_x   
## 0 0 0   
## magnet\_arm\_y magnet\_arm\_z kurtosis\_roll\_arm   
## 0 0 0   
## kurtosis\_picth\_arm kurtosis\_yaw\_arm skewness\_roll\_arm   
## 0 0 0   
## skewness\_pitch\_arm skewness\_yaw\_arm max\_roll\_arm   
## 0 0 19216   
## max\_picth\_arm max\_yaw\_arm min\_roll\_arm   
## 19216 19216 19216   
## min\_pitch\_arm min\_yaw\_arm amplitude\_roll\_arm   
## 19216 19216 19216   
## amplitude\_pitch\_arm amplitude\_yaw\_arm roll\_dumbbell   
## 19216 19216 0   
## pitch\_dumbbell yaw\_dumbbell kurtosis\_roll\_dumbbell   
## 0 0 0   
## kurtosis\_picth\_dumbbell kurtosis\_yaw\_dumbbell skewness\_roll\_dumbbell   
## 0 0 0   
## skewness\_pitch\_dumbbell skewness\_yaw\_dumbbell max\_roll\_dumbbell   
## 0 0 19216   
## max\_picth\_dumbbell max\_yaw\_dumbbell min\_roll\_dumbbell   
## 19216 0 19216   
## min\_pitch\_dumbbell min\_yaw\_dumbbell amplitude\_roll\_dumbbell   
## 19216 0 19216   
## amplitude\_pitch\_dumbbell amplitude\_yaw\_dumbbell total\_accel\_dumbbell   
## 19216 0 0   
## var\_accel\_dumbbell avg\_roll\_dumbbell stddev\_roll\_dumbbell   
## 19216 19216 19216   
## var\_roll\_dumbbell avg\_pitch\_dumbbell stddev\_pitch\_dumbbell   
## 19216 19216 19216   
## var\_pitch\_dumbbell avg\_yaw\_dumbbell stddev\_yaw\_dumbbell   
## 19216 19216 19216   
## var\_yaw\_dumbbell gyros\_dumbbell\_x gyros\_dumbbell\_y   
## 19216 0 0   
## gyros\_dumbbell\_z accel\_dumbbell\_x accel\_dumbbell\_y   
## 0 0 0   
## accel\_dumbbell\_z magnet\_dumbbell\_x magnet\_dumbbell\_y   
## 0 0 0   
## magnet\_dumbbell\_z roll\_forearm pitch\_forearm   
## 0 0 0   
## yaw\_forearm kurtosis\_roll\_forearm kurtosis\_picth\_forearm   
## 0 0 0   
## kurtosis\_yaw\_forearm skewness\_roll\_forearm skewness\_pitch\_forearm   
## 0 0 0   
## skewness\_yaw\_forearm max\_roll\_forearm max\_picth\_forearm   
## 0 19216 19216   
## max\_yaw\_forearm min\_roll\_forearm min\_pitch\_forearm   
## 0 19216 19216   
## min\_yaw\_forearm amplitude\_roll\_forearm amplitude\_pitch\_forearm   
## 0 19216 19216   
## amplitude\_yaw\_forearm total\_accel\_forearm var\_accel\_forearm   
## 0 0 19216   
## avg\_roll\_forearm stddev\_roll\_forearm var\_roll\_forearm   
## 19216 19216 19216   
## avg\_pitch\_forearm stddev\_pitch\_forearm var\_pitch\_forearm   
## 19216 19216 19216   
## avg\_yaw\_forearm stddev\_yaw\_forearm var\_yaw\_forearm   
## 19216 19216 19216   
## gyros\_forearm\_x gyros\_forearm\_y gyros\_forearm\_z   
## 0 0 0   
## accel\_forearm\_x accel\_forearm\_y accel\_forearm\_z   
## 0 0 0   
## magnet\_forearm\_x magnet\_forearm\_y magnet\_forearm\_z   
## 0 0 0   
## classe   
## 0

There are many columns with 19216 na's. These are removed.

a<-Filter(function(x) sum(is.na(x)) < 19216, training)  
b<-Filter(function(x) sum(is.na(x)) < 19216, testing)

NearZerovar is applied to remove predictors with only 1 value

a\_nzv<- nearZeroVar(a)  
new\_a<- a[,-a\_nzv]  
b\_nzv<- nearZeroVar(b)  
new\_b<- b[,-b\_nzv]  
dim(new\_a);dim(new\_b)

## [1] 19622 59

## [1] 20 59

The first 6 variables are deleted as they are descriptive, no measures

trainnew<-new\_a[,-c(1:6)]  
testnew <-new\_b[,-c(1:6)]

30% of the trainingset is split off to function as validation set:

inTrain<- createDataPartition(y=trainnew$classe,p=0.7,list=FALSE)  
trainset<- trainnew[inTrain,]  
validationset<- trainnew[-inTrain,]

## Building the model

A random forest model is built.

library(randomForest)

## randomForest 4.6-12

## Type rfNews() to see new features/changes/bug fixes.

##   
## Attaching package: 'randomForest'

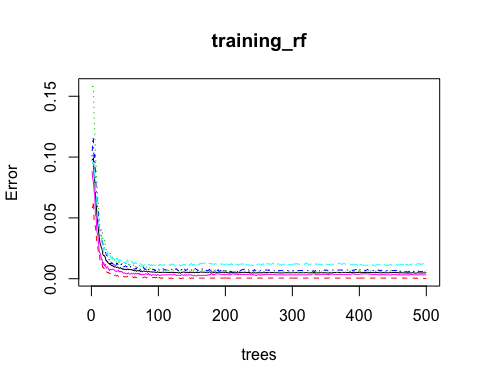
## The following object is masked from 'package:ggplot2':  
##   
## margin

set.seed(7)  
training\_rf <- randomForest(x=trainset[,1:(ncol(trainset)-1)], y=trainset[,"classe"], importance=TRUE, do.trace=100)

## ntree OOB 1 2 3 4 5  
## 100: 0.52% 0.08% 0.64% 0.79% 1.11% 0.28%  
## 200: 0.51% 0.05% 0.71% 0.63% 1.15% 0.32%  
## 300: 0.50% 0.05% 0.56% 0.67% 1.20% 0.32%  
## 400: 0.51% 0.05% 0.60% 0.71% 1.15% 0.36%  
## 500: 0.50% 0.03% 0.56% 0.67% 1.24% 0.32%

OOB = 0.50% Accuracy=1-OOB=99.5%

plot(training\_rf)

 Now testing the model on the validation set:

pred<-predict(training\_rf,validationset);validationset$predRight<-pred==validationset$classe  
table(pred,validationset$classe)

##   
## pred A B C D E  
## A 1674 11 0 0 0  
## B 0 1126 5 0 0  
## C 0 2 1018 4 0  
## D 0 0 3 960 4  
## E 0 0 0 0 1078

confusionMatrix(pred, validationset$classe)

## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction A B C D E  
## A 1674 11 0 0 0  
## B 0 1126 5 0 0  
## C 0 2 1018 4 0  
## D 0 0 3 960 4  
## E 0 0 0 0 1078  
##   
## Overall Statistics  
##   
## Accuracy : 0.9951   
## 95% CI : (0.9929, 0.9967)  
## No Information Rate : 0.2845   
## P-Value [Acc > NIR] : < 2.2e-16   
##   
## Kappa : 0.9938   
## Mcnemar's Test P-Value : NA   
##   
## Statistics by Class:  
##   
## Class: A Class: B Class: C Class: D Class: E  
## Sensitivity 1.0000 0.9886 0.9922 0.9959 0.9963  
## Specificity 0.9974 0.9989 0.9988 0.9986 1.0000  
## Pos Pred Value 0.9935 0.9956 0.9941 0.9928 1.0000  
## Neg Pred Value 1.0000 0.9973 0.9984 0.9992 0.9992  
## Prevalence 0.2845 0.1935 0.1743 0.1638 0.1839  
## Detection Rate 0.2845 0.1913 0.1730 0.1631 0.1832  
## Detection Prevalence 0.2863 0.1922 0.1740 0.1643 0.1832  
## Balanced Accuracy 0.9987 0.9938 0.9955 0.9972 0.9982

Accuracy : 0.9968 oob=1-0.9968=0.0032

## Use prediction model to predict 20 testcases

pred2<-predict(training\_rf,testnew)  
pred2

## 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