# Machine Learning Project

*Manuel Mariscal*

*25/2/2020*

## SUMMARY

I 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 (A,B,C,D,E). More information is available from the website here: http://web.archive.org/web/20161224072740/http:/groupware.les.inf.puc-rio.br/har (see the section on the Weight Lifting Exercise Dataset).

The goal is to predict the manner in which different people did the exercise. This is the "classe" variable in the training set. In this report, I describe how I choose the best variables to train the model, I built a model using cross validation, and I calculate the accuracy of the prediction. See conclusions to check the results.

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

## LOAD AND PRE-PROCESS DATA

I will take only the columns with some information in the test set. Therefore, I will delete the variables in which all the observations are NAs. I am going to train the model using only those predictors.

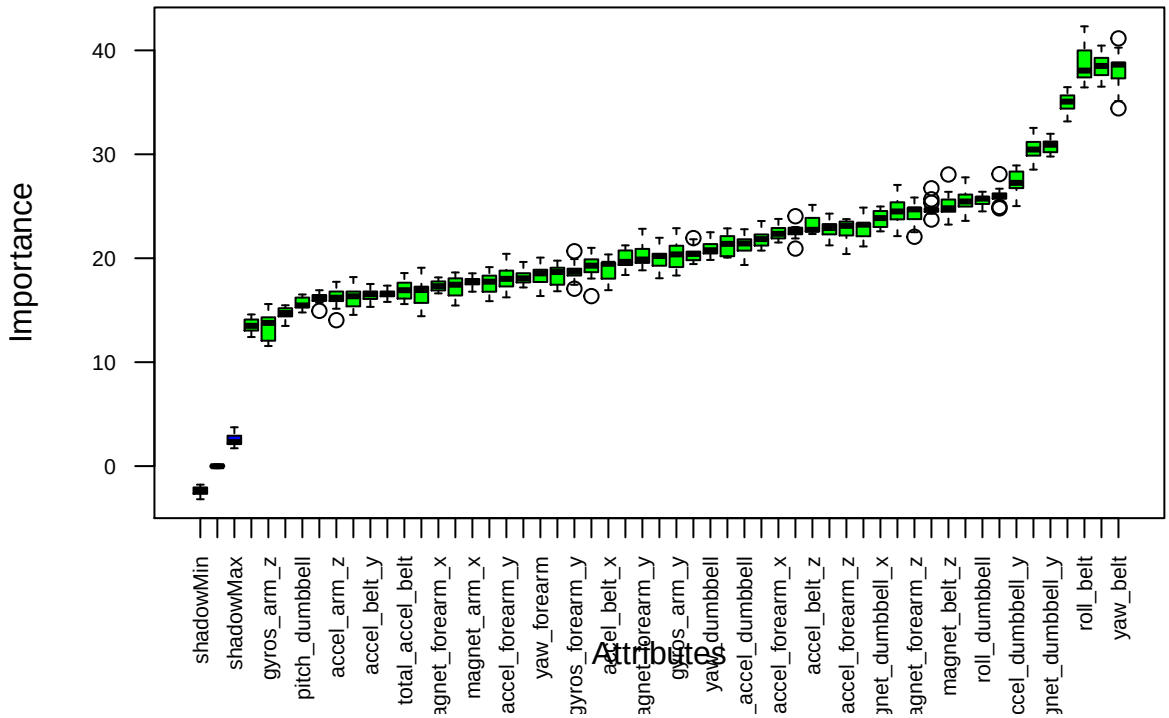
```
set.seed(222)
library(caret); library(kernlab); library(randomForest); library(dplyr)

setwd("C:/Users/mmariscal/Documents/3. Personal/JHU/8. Practical Machine Learning/Project/")
train <- read.csv("pml-training.csv")
test <- read.csv("pml-testing.csv")

index <- colSums(is.na(test)) == nrow(test)
train <- train[!index]
test <- test[!index]
# I am going to delete also the first columns with time and names information because
# in my opinion that could mislead the model
train <- train[-(1:7)]
test <- test[-(1:7)]
test <- select(test, -problem_id)
```

## FEATURE SELECTION

I am going to use different methods and mixing them. I will use *t_feat* as a small data sample only to study the best features and in order not to require a lot of computing time.

To train these models and also for the final one, I am using a k=5 fold cross validation method.

```r
inTrain <- createDataPartition(y=train$classe, p=0.8, list=FALSE)
t_feat <- train[-inTrain,]

# set parallel processing will speed up the computing
library(parallel)
library(doParallel)
cluster <- makeCluster(detectCores() - 1) # convention to leave 1 core for OS
registerDoParallel(cluster)

# Define train control for k fold cross validation and parallel processing
fitControl <- trainControl(method = "cv", number = 5, allowParallel = TRUE)

# RFE Method
subsets <- c(15, 20, 30) #the number of most important features the rfe should iterate
ctrl <- rfeControl(functions = rfFuncs, # for ramdon forest
                   method = "repeatedcv", # k-fold cross validation
                   repeats = 5, # I set K = 5
                   verbose = FALSE)

mod_rfe <- rfe(classe ~., data = t_feat, sizes = subsets,
               rfeControl = ctrl, trControl = fitControl)

names_rfe <- mod_rfe$optVariables

# varImp function
mod_rf <- train(classe ~., data = t_feat, method = "rf", trControl = fitControl)
imp <- varImp(mod_rf)
impDF<-data.frame(imp[1])
names_rf <- rownames(impDF)[order(impDF$Overall, decreasing=TRUE)[1:40]]

# Boruta
library(Boruta)
mod_bor <- Boruta(classe ~., data = t_feat, doTrace = 2, maxRuns = 300)
plot(mod_bor, las = 2, cex.axis = 0.7)
```




```
names_bor <- getSelectedAttributes(mod_bor)

# And finally, I choose the features that are common to the three of them.
features <- intersect(names_rf, intersect(names_bor, names_rfe))
features
```

```
##  [1] "roll_belt"            "pitch_forearm"        "yaw_belt"
##  [4] "magnet_dumbbell_z"    "magnet_dumbbell_y"    "pitch_belt"
##  [7] "roll_forearm"         "roll_dumbbell"        "magnet_belt_z"
## [10] "magnet_dumbbell_x"    "accel_forearm_x"      "total_accel_dumbbell"
## [13] "magnet_belt_y"        "accel_dumbbell_y"     "accel_belt_z"
## [16] "yaw_arm"              "accel_dumbbell_z"     "magnet_forearm_z"
## [19] "magnet_belt_x"        "gyros_belt_z"         "roll_arm"
## [22] "magnet_arm_x"         "accel_forearm_z"      "yaw_dumbbell"
## [25] "gyros_dumbbell_y"     "magnet_arm_y"         "magnet_arm_z"
## [28] "accel_dumbbell_x"     "gyros_dumbbell_x"     "magnet_forearm_y"
```

These features are the most important ones for training the model according to RFE, varIMP and Boruta methods. We can see also a picture representing the importance of the variables according to Boruta.

## MODEL SELECTION

I will use random forest with the list of features selected in the previous code. I am going to divide the data (because it is big enough) into a new training and testing data sets. That way, I will be able to test my model before predicting on the final test.

```
feat <- features[1]
for (i in 2:length(features))
{
        feat <- paste(feat, "+", features[i])
}
form <- as.formula(paste('classe ~', feat))

inTrain <- createDataPartition(y=train$classe, p=0.7, list=FALSE)
testing <- train[-inTrain,]
training <- train[inTrain,]

model <- train(form, data =training, method="rf", ntree=200, trControl = fitControl)
```

## TEST VALIDATION

```
pred <- predict(model, newdata = testing)

confusionMatrix(pred, testing$classe)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction    A    B    C    D    E
##          A 1673    9    0    0    0
##          B    0 1122    6    0    0
##          C    0    8 1019   13    1
##          D    1    0    1  950    2
##          E    0    0    0    1 1079
##
## Overall Statistics
##
##                Accuracy : 0.9929
##                  95% CI : (0.9904, 0.9949)
##     No Information Rate : 0.2845
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.991
##
##  Mcnemar's Test P-Value : NA
##
## Statistics by Class:
##
##                      Class: A Class: B Class: C Class: D Class: E
## Sensitivity            0.9994   0.9851   0.9932   0.9855   0.9972
## Specificity            0.9979   0.9987   0.9955   0.9992   0.9998
## Pos Pred Value         0.9946   0.9947   0.9789   0.9958   0.9991
## Neg Pred Value         0.9998   0.9964   0.9986   0.9972   0.9994
## Prevalence             0.2845   0.1935   0.1743   0.1638   0.1839
## Detection Rate         0.2843   0.1907   0.1732   0.1614   0.1833
## Detection Prevalence   0.2858   0.1917   0.1769   0.1621   0.1835
## Balanced Accuracy      0.9986   0.9919   0.9943   0.9923   0.9985
```

Because accuracy is good, I can train the final model with the complete data and predict the test set.

### FINAL MODEL

```
final_model <- train(form, data =train, method="rf", ntree=200, trControl = fitControl)

final_pred <- predict(model, newdata = test)
```

## Conclusion

The model has been trained using k-fold cross validation. The features were selected using Boruta, VarIMP and RFE methods:

```
features
```

```
##  [1] "roll_belt"            "pitch_forearm"        "yaw_belt"
##  [4] "magnet_dumbbell_z"    "magnet_dumbbell_y"    "pitch_belt"
##  [7] "roll_forearm"         "roll_dumbbell"        "magnet_belt_z"
## [10] "magnet_dumbbell_x"    "accel_forearm_x"      "total_accel_dumbbell"
## [13] "magnet_belt_y"        "accel_dumbbell_y"     "accel_belt_z"
## [16] "yaw_arm"              "accel_dumbbell_z"     "magnet_forearm_z"
## [19] "magnet_belt_x"        "gyros_belt_z"         "roll_arm"
## [22] "magnet_arm_x"         "accel_forearm_z"      "yaw_dumbbell"
## [25] "gyros_dumbbell_y"     "magnet_arm_y"         "magnet_arm_z"
## [28] "accel_dumbbell_x"     "gyros_dumbbell_x"     "magnet_forearm_y"
```

```
1 - confusionMatrix(pred, testing$classe)$overall[1]
```

```
##    Accuracy
## 0.007136788
```

```
final_pred
```

```
##  [1] 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
```

The accuracy of the model is bigger than 99.4% and has an out of sample error of only 0.0057 %. Probably, it would be neccessary to adjust better the prediction for class D.

The final prediction for the test set is also shown.