Title: “Practical Machine Learning - Course Project”

Predicting Quality of Activity

Author: “Luís Adriano Domingues”

date: “27/11/2020”

Executive Summary

Considering that nowadays people regularly quantify how much of a particular activity they do, but they rarely quantify how well they do it. To explore this a group of people, using accelerometers, performed activities (lifting dumbells) in one correct and some incorrect ways - classified as mistakes. In this project, we will use data from accelerometers on the belt, forearm and arm of 6 participants in order to establish a prediction model.

The prediction model is used on a testing set in order to check its correctnness.

Loading and preliminary examination of data

```
## cleaning memory
rm(list = ls(all = TRUE))

## setting working directory
setwd('C:/Disco_D/LAMCD/Coursera/JHDS_course/Practical Machine
Learning/Project')

## loading related packages
library(caret)

## Loading required package: lattice

## Loading required package: ggplot2
```

```
library(knitr)
library(rpart)
library(rpart.plot)
library(RColorBrewer)
library(rattle)

## Loading required package: tibble

## Loading required package: bitops

## Rattle: A free graphical interface for data science with R.
## Version 5.4.0 Copyright (c) 2006-2020 Togaware Pty Ltd.
## Type 'rattle()' to shake, rattle, and roll your data.

library(randomForest)

## randomForest 4.6-14

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

##
## Attaching package: 'randomForest'

## The following object is masked from 'package:rattle':
##
##     importance

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

library(corrplot)

## corrplot 0.84 loaded

library(e1071)

## reading data files
trainingd <- read.csv(file="pml-training.csv")
testingd  <- read.csv(file="pml-testing.csv")

# create a partition on the training dataset and examine the data
inTrain  <- createDataPartition(trainingd$classe, p=0.7, list=FALSE)
dataTrain <- trainingd[inTrain, ]
dataTest  <- trainingd[-inTrain, ]
dim(dataTrain)

## [1] 13737   160

dim(dataTest)

## [1] 5885  160
## str(dataTrain)
## head(dataTrain)
## head (dataTest)
```

Cleaning the data

Examining the training data set, whith str(dataTrain) we find there are 160 variables. However most variables are not usefull for the prediction model: NAs, zeros, etc. So a cleaning process is required, where we will simply eliminate those useless variables.

```
nearzeroV <- nearZeroVar(dataTrain)
dataTrain <- dataTrain[, -nearzeroV]
dataTest  <- dataTest[, -nearzeroV]
## dim(dataTrain)
## dim(dataTest)
```

Eliminating NAs is also important.

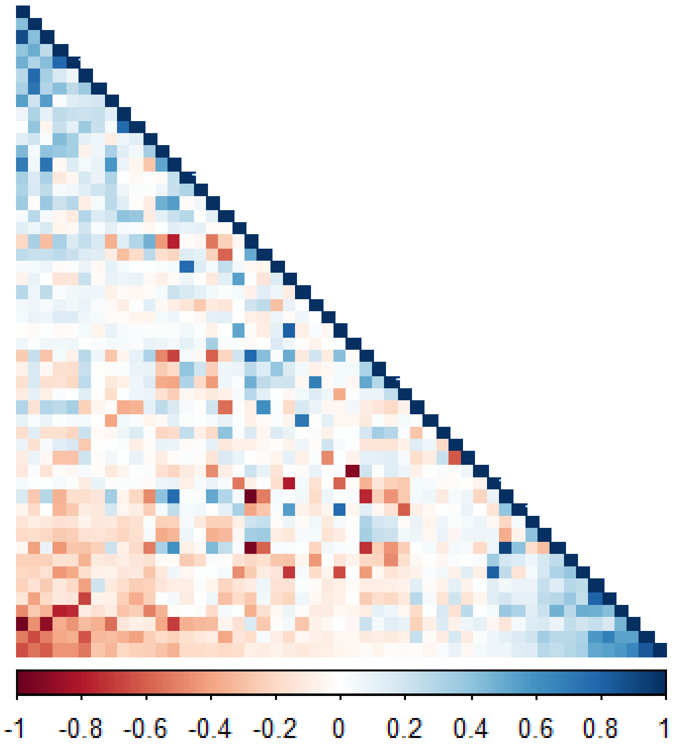
```
varNAs    <- sapply(dataTrain, function(x) mean(is.na(x))) > 0.95
dataTrain <- dataTrain[, varNAs == FALSE]
dataTest  <- dataTest[, varNAs == FALSE]
## dim(dataTrain)
## dim(dataTest)
```

The variables which are only identifications (1:7) are also not useful for prediction and will be removed - columns 1:7.

```
dataTrain <- dataTrain[, -(1:7)]
dataTest  <- dataTest[, -(1:7)]
dim(dataTrain)

## [1] 13737    52

dim(dataTest)

## [1] 5885   52

## head(dataTrain)
```

The modelling strategy is to build different prediction models, and them test to quantify their performances, finally select the best one to predict in the quizz test. We start by ploting a map of the correlation of the variables, to guide the selection of prediction models in the sequence.

```
M_corr <- cor(dataTrain[, -52])
corrplot(M_corr, order = "FPC", method = "color", type = "lower",
         tl.cex = 0.7, tl.col = rgb(1, 1, 1))
```



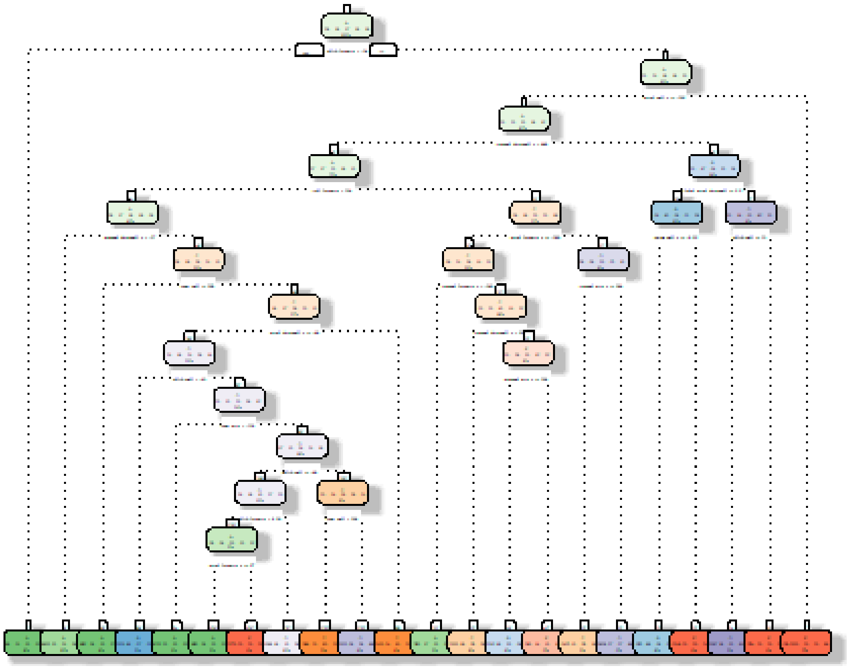
The higher correlations are shown in darker colours. While there are relatively few high correlations between variables, most variables have some strong correlation with some other variable, so we will keep all variables and move to build prediction models.

We will use 3 prediction models to compare their performances, which seems adequate - not too few, not too much. Since there are a considerable number of predictor variables we chose: Decision trees, Stochastic gradient boosting trees and Random forest decision trees.

```
## random forests
set.seed(271120)
c_RF <- trainControl(method="cv", number=3, verboseIter=FALSE)
m_F_RF <- train(classe ~ ., data=dataTrain, method="rf",
                          trControl=c_RF)
m_F_RF$finalModel

##
## Call:
##  randomForest(x = x, y = y, mtry = param$mtry)
##                Type of random forest: classification
##                      Number of trees: 500
## No. of variables tried at each split: 26
##
##         OOB estimate of  error rate: 0.71%
## Confusion matrix:
##      A    B    C    D    E  class.error
## A 3903    2    1    0    0 0.0007680492
## B   24 2627    6    0    1 0.0116629044
## C    0   19 2370    7    0 0.0108514190
## D    0    0   25 2224    3 0.0124333925
## E    0    0    3    7 2515 0.0039603960
```

```
## decision tree
set.seed(271120)
m_F_DT <- rpart(classe ~ ., data=dataTrain, method="class")
fancyRpartPlot(m_F_DT)
```

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



Rattle 2020-nov-27 17:28:59 luisa

```
## generalized boosted model
set.seed(271120)
c_GBM <- trainControl(method = "repeatedcv", number = 5, repeats = 1)
m_F_GBM  <- train(classe ~ ., data=dataTrain, method = "gbm",
                    trControl = c_GBM, verbose = FALSE)
m_F_GBM$finalModel
```

```
## A gradient boosted model with multinomial loss function.
## 150 iterations were performed.
## There were 51 predictors of which 51 had non-zero influence.
```

The Prediction Models have been created. The next step is to apply those models to the test section of the data partition, and then select the best model.

```
## prediction with random forest
p_RF <- predict(m_F_RF, newdata=dataTest)
## conf_M_RF <- confusionMatrix(p_RF, dataTest$classe)
## conf_M_RF

## plot(conf_M_RF$table, col = conf_M_RF$byClass,
##     main = paste("Random Forest - Accuracy =",
##                  round(conf_M_RF$overall['Accuracy'], 4)))

# prediction with decision tree
p_DT <- predict(m_F_DT, newdata=dataTest, type="class")
## conf_M_DT <- confusionMatrix(p_DT, dataTest$classe)
## conf_M_DT

## plot(conf_M_DT$table, col = conf_M_DT$byClass,
##     main = paste("Decision Tree - Accuracy =",
##                  round(conf_M_DT$overall['Accuracy'], 4)))

# prediction with generalized boosted model
p_GBM <- predict(m_F_GBM, newdata=dataTest)
## conf_M_GBM <- confusionMatrix(p_GBM, dataTest$classe)
## conf_M_GBM

## plot(conf_M_GBM$table, col = conf_M_GBM$byClass,
##     main = paste("GBM - Accuracy =",
round(conf_M_GBM$overall['Accuracy'], 4)))
```

The accuracy of the three prediction models was:

Random Forest: 0.996

Decision Tree: 0.737

GBM: 0.984

So we will use the Random Forest Predicition Model to answer the quizz (Testing DataSet: dataTest)

```
predictQuizz <- predict(m_F_RF, newdata=testingd)
predictQuizz

##  [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
```

LAMCD