Human Activity Recognition Project

Yann Claudel

22 dec 2016

Synopsis

The purpose of this analysis is to predict the manner in which people did weight lifting exercises, meaning quantify how well they do it.
The model is built with data from accelerometers on the belt, forearm, arm, and dumbbell of 6 participants.

> Six young health participants were asked to perform one set of 10 repetitions of the Unilateral Dumbbell Biceps Curl in five different fashions: exactly according to the specification (Class A), throwing the elbows to the front (Class B), lifting the dumbbell only halfway (Class C), lowering the dumbbell only halfway (Class D) and throwing the hips to the front (Class E).

These information come from the website: http://groupware.les.inf.puc-rio.br/har (http://groupware.les.inf.puc-rio.br/har)

Loading and preprocessing the data

Training data: https://d396qusza40orc.cloudfront.net/predmachlearn/pml-training.csv (https://d396qusza40orc.cloudfront.net/predmachlearn/pml-training.csv)

Testing data: https://d396qusza40orc.cloudfront.net/predmachlearn/pml-testing.csv (https://d396qusza40orc.cloudfront.net/predmachlearn/pml-testing.csv)

Data come from the website: http://groupware.les.inf.puc-rio.br/har (http://groupware.les.inf.puc-rio.br/har)

```
library(randomForest)
library(caret)
library(ggplot2)
library(corrplot)
```

Load data

```
train = read.csv("pml-training.csv", header = TRUE,na.strings=c("", "NA", "NU
LL"))
test = read.csv("pml-testing.csv", header = TRUE, na.strings=c("", "NA", "NUL
L"))
```

Remove unused columns

```
train <- subset(train, select =-c(X,user_name,raw_timestamp_part_1,raw_timest
amp_part_2,cvtd_timestamp,new_window,num_window))
test <- subset(test, select =-c(X,user_name,raw_timestamp_part_1,raw_timestam
p_part_2,cvtd_timestamp,new_window,num_window))
```

Remove variables that have more than 50% of NA in the train data

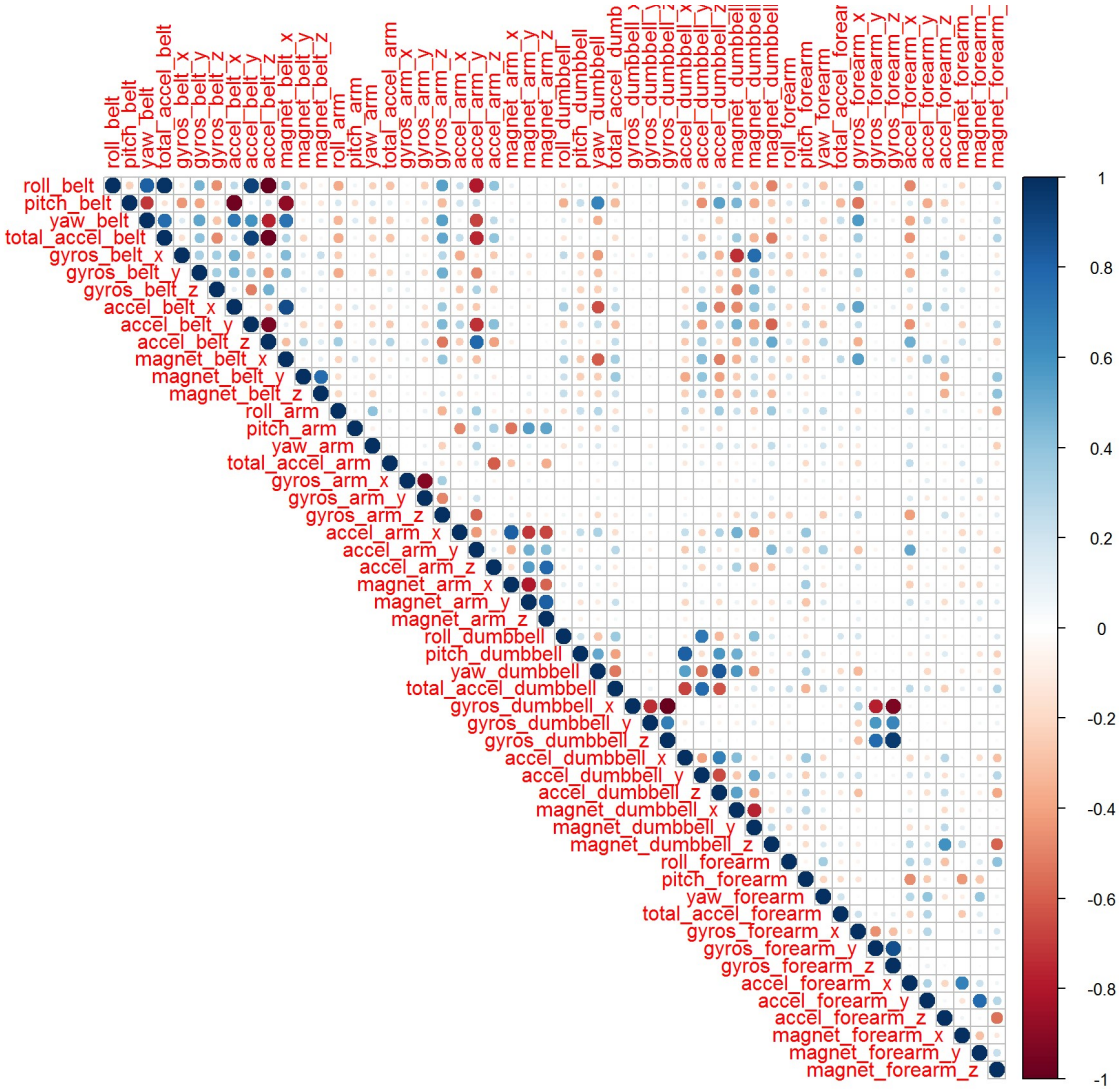
```
percentNA <- colSums(is.na(train))/19622
train<- train[,percentNA < 0.5]
test<- test[,percentNA < 0.5]
```

Split train data in order to do cross validation of the model

```
inTrain <- createDataPartition(y=train$classe,p=0.7, list=FALSE)
trainModel <- train[inTrain,]
testModel <- train[-inTrain,]
```

Exploratory data analyses - Correlation matrix

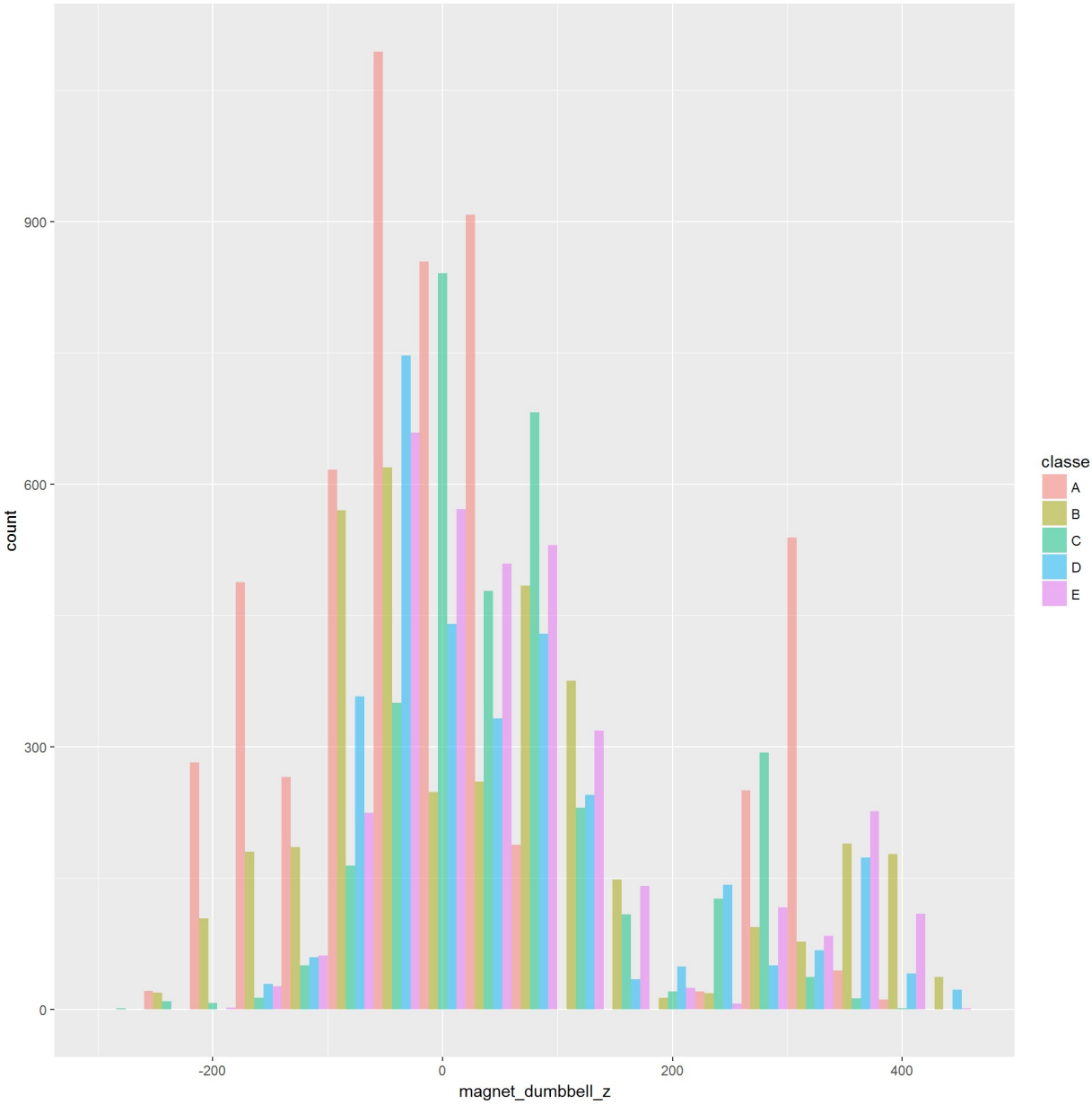
```
M <- cor(subset(trainModel, select =-c(classe)))
corrplot(M, type="upper")
```



Plot variable

I have plot several variables in order to see a pattern (for instance magnet_dumbbell_z hereafter) I would say that it's difficult to see a pattern in all these graphs. But a cycle A-B-C-D-E-A-B-.. seems to be present.

```
g <- ggplot(train, aes(x=magnet_dumbbell_z, fill=classe)) +  geom_histogram(b
inwidth=40, alpha=.5, position="dodge")
g
```



The problem is not linear so I select the Ramdow Forest Model.

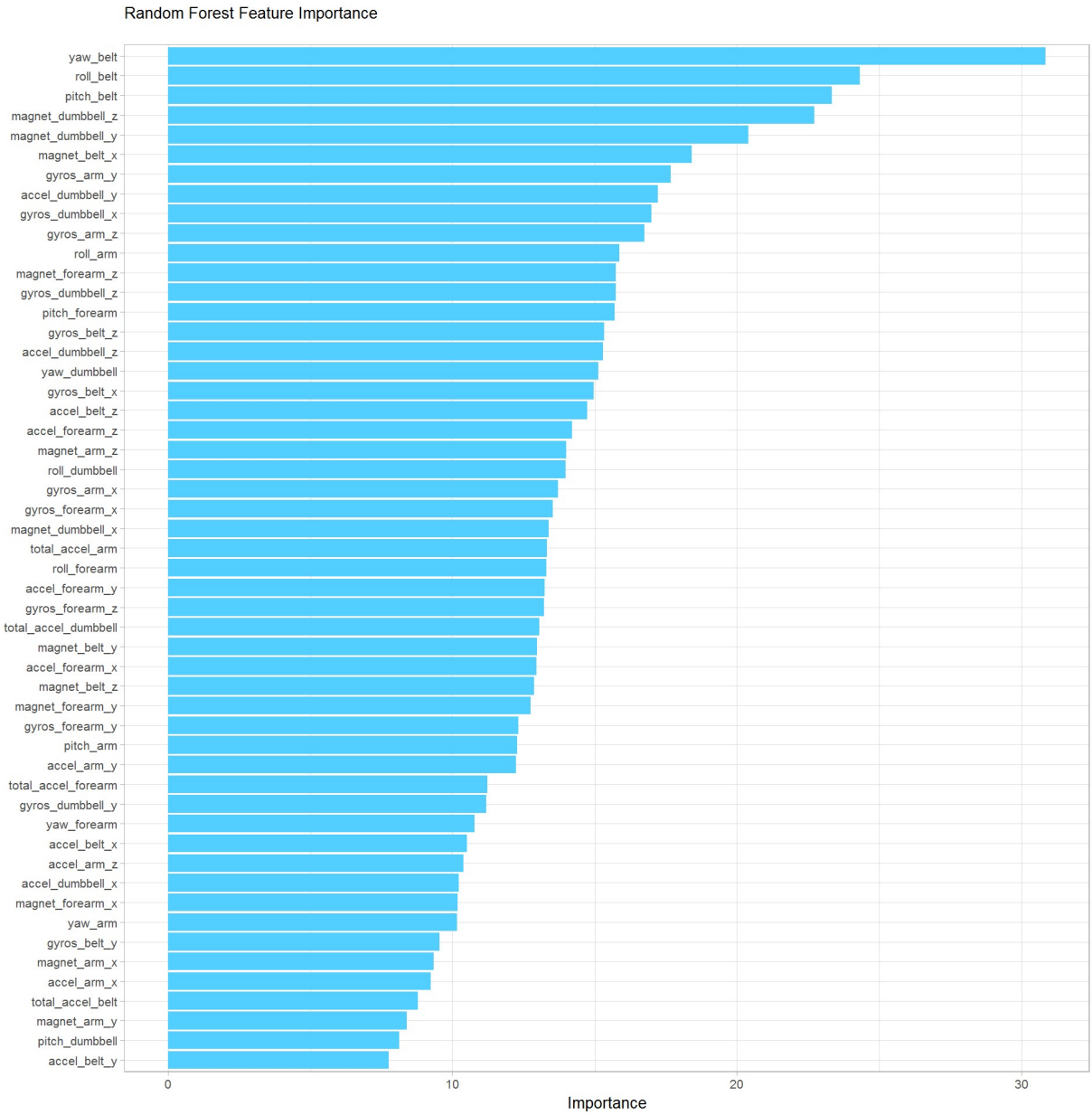
Build the model with randow forest

```
set.seed(12345)
mod_rf <- randomForest(subset(trainModel, select =-c(classe)), trainModel$cla
sse, ntree=100, importance=TRUE)
```

Build a model

```
imp <- importance(mod_rf, type=1)
featureImportance <- data.frame(Feature=row.names(imp), Importance=imp[,1])

ggplot(featureImportance, aes(x=reorder(Feature, Importance), y=Importance))
+
      geom_bar(stat="identity", fill="#53cfff") +
      coord_flip() +
      theme_light(base_size=10) +
      xlab("") +
      ylab("Importance") +
      ggtitle("Random Forest Feature Importance\n") +
      theme(plot.title=element_text(size=10))
```



Cross validation

```
pred_rf <- predict(mod_rf, subset(testModel, select =-c(classe)),type="clas
s")
confMat <- confusionMatrix(pred_rf, testModel$classe)
confMat$overall[1]
```

```
##  Accuracy
## 0.9952421
```

```
confMat
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction    A    B    C    D    E
##          A 1673    8    0    0    0
##          B    1 1131    5    0    0
##          C    0    0 1021   11    0
##          D    0    0    0  953    3
##          E    0    0    0    0 1079
##
## Overall Statistics
##
##                Accuracy : 0.9952
##                  95% CI : (0.9931, 0.9968)
##     No Information Rate : 0.2845
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.994
##  Mcnemar's Test P-Value : NA
##
## Statistics by Class:
##
##                      Class: A Class: B Class: C Class: D Class: E
## Sensitivity            0.9994   0.9930   0.9951   0.9886   0.9972
## Specificity            0.9981   0.9987   0.9977   0.9994   1.0000
## Pos Pred Value         0.9952   0.9947   0.9893   0.9969   1.0000
## Neg Pred Value         0.9998   0.9983   0.9990   0.9978   0.9994
## Prevalence             0.2845   0.1935   0.1743   0.1638   0.1839
## Detection Rate         0.2843   0.1922   0.1735   0.1619   0.1833
## Detection Prevalence   0.2856   0.1932   0.1754   0.1624   0.1833
## Balanced Accuracy      0.9988   0.9959   0.9964   0.9940   0.9986
```

Prediction

```
pred_rf <- predict(mod_rf, subset(test, select =-c(problem_id)))
pred_rf
```

```
##  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
```

HAPPY DATA MINING 2017