Practical Machine Learning Project : Quality of Weight Lifting

Introduction

Using devices such as Jawbone Up, Nike FuelBand, and Fitbit it is now possible to collect a large amount of data about personal activity relatively inexpensively. One thing that people regularly do is quantify how much of a particular activity they do, but they rarely quantify how well they do it. In this project, we use data from accelerometers on the belt, forearm, arm, and dumbbell of 6 participants. They were asked to perform barbell lifts correctly and incorrectly in 5 different ways : 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). The goal of the project is to predict the manner in which they did the exercise (i.e. the Class, A to E) using the other variables in the data set.

Importing and cleaning the data

The data set is made of a training set and a test set.

An initial observation of the files via a text editor reveals that :

- the columns in the file are separated by commas
- there are many missing values, which are of 2 types : the usual "NA" and also "#DIV/0!"
- the first line gives the names of the variables
- the first column gives the row number, so is not a useful variable

Taking all this into account, we import the files into R as follows (assuming that they are located in the working directory) :

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

We now look at the imported training set :

```
fix(pml_training)
```

We can see that :

- there are 19622 rows (observations) and 160 columns (variables) in the training set
- columns 1 to 7 are not sensor readings and hence are not useful for the prediction
- columns 8 to 159 correspond to sensor readings for one the four sensors (belt, forearm, arm and dumbbell)
- column 160 is the Class variable (which is called "classe") which tells us how the exercise was done, i.e. the outcome variable that we want to predict

- some columns/variables have missing values (indicated as “NA” in the R object) for almost all observations

A similar look at the test set shows that it only has 20 rows/observations and has no Class variable. Therefore we won’t be able to know whether the predictions made on the test set are right or not.

We first remove the first 7 columns, which are not useful :

```
pml_training <- pml_training[,-c(1:7)]
pml_testing <- pml_testing[,-c(1:7)]
```

We then remove the columns/variables which have mostly missing values :

```
cols_vect <- (colSums(is.na(pml_training))==0)
pml_training <- pml_training[,cols_vect]
pml_testing <- pml_testing[,cols_vect]
dim(pml_training)
```

```
## [1] 19622    53
```

We can see that there are only 52 predictor variables left (plus the Class variable, which yields 53 columns/variables in total).

Choice and construction of the classifier

We chose the Random Forest (RF) as our classification algorithm for the following reasons :

1. the reference article of this study [1] explicitly mentions that it is well suited to the characteristic noise in the sensor data
2. it is likely that some of the 52 predictors are highly correlated with each other and this is not a problem for the Random Forest algorithm (whereas it can be an issue for other methods such as logistic regression)
3. the RF algorithm does some internal cross-validation which automatically provides an out-of-sample error rate (called the “Out-Of-Bag” or OOB error rate). Therefore there is no need to perform an external cross-validation

We use the default values of the parameters :

```
set.seed(55)
library(randomForest)
```

```
## randomForest 4.6-12
```

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

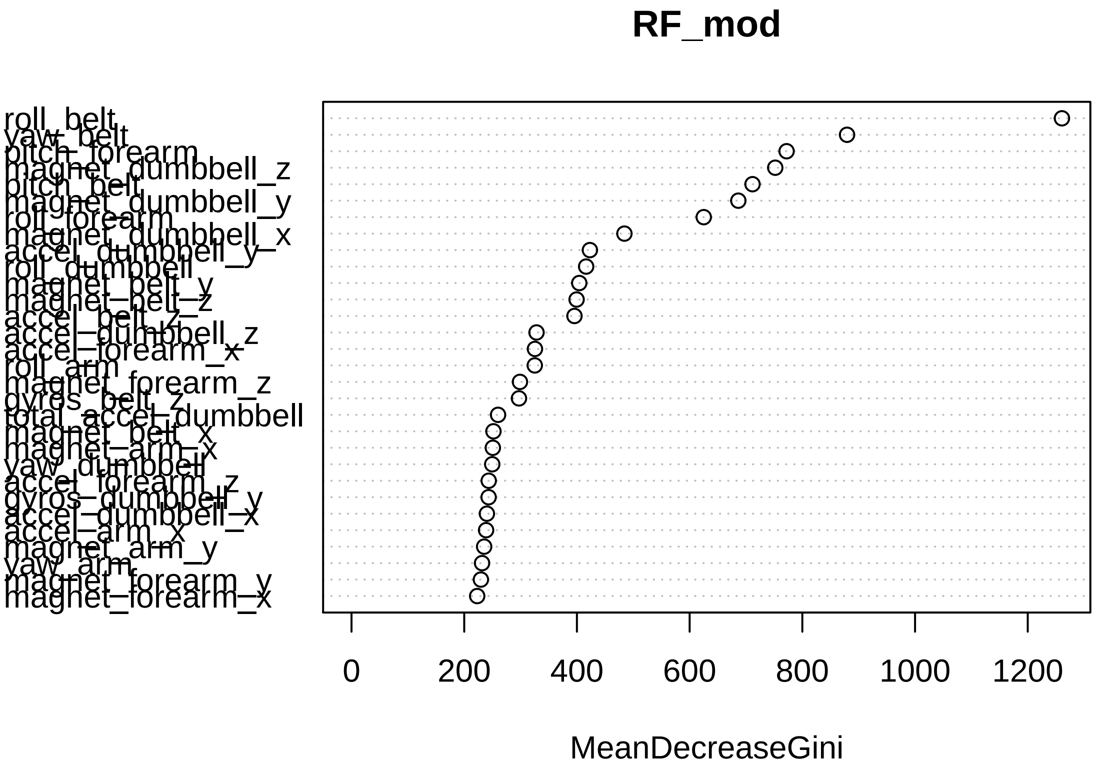
```
RF_mod <- randomForest(classe~., data = pml_training)
RF_mod
```

```
##
## Call:
##  randomForest(formula = classe ~ ., data = pml_training)
##                Type of random forest: classification
##                      Number of trees: 500
## No. of variables tried at each split: 7
##
##         OOB estimate of  error rate: 0.27%
## Confusion matrix:
##      A    B    C    D    E  class.error
## A 5578    1    0    0    1 0.0003584229
## B    9 3786    2    0    0 0.0028970240
## C    0   11 3410    1    0 0.0035067212
## D    0    0   19 3195    2 0.0065298507
## E    0    0    1    5 3601 0.0016634322
```

We can see that the OOB error rate is very low (around 0.3%), hence the classifier is very good.

Let's make a plot of the variables' respective importance :

```
varImpPlot(RF_mod)
```



We can see that the 3 most important variables/predictors are : roll_belt, yaw_belt and pitch_forearm.

Application of the classifier to the test set

We now apply the constructed classifier to the test set :

```
predTest <- predict(RF_mod, pml_testing)
predTest
```

```
##  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
```

As said above, since class labels are not provided with the test set, we can't know whether these predictions are right or not. However, given the very low OOB error rate above, we are quite confident abour our classifier.

References

1. Velloso, E.; Bulling, A.; Gellersen, H.; Ugulino, W.; Fuks, H. Qualitative Activity Recognition of Weight Lifting Exercises. Proceedings of 4th Augmented Human (AH) International Conference in cooperation with ACM SIGCHI (Augmented Human'13) . Stuttgart, Germany: ACM SIGCHI, 2013. Available at : http://groupware.les.inf.puc-rio.br/public/papers/2013.Velloso.QAR-WLE.pdf