How do people exercise?

David Parra

July 26, 2017

Summary

This document contains a simple machine learning analysis of the Weight Lifting Exercise Dataset associating the classe variable (Technique of WWeigh lifting) as a function of body measurements performed during the exercise. First some exploratory data analysis is made, followed by some preprocessing of the data, and finally some machine learning algorithms were used in order to find the best predictor for the type of weight lifting exercise and to determine the out of sample error when applying this algorithms.

Loading and Exploratory analisis

The datasets for both the pml_training and the pml_testing part of the exercise were uploaded. It was determined that there were two types of missing values; NA and "". Once this values were accounted for it was noted that most of the columns in both datasets were mostly missing values, therefore this columns were removed from the pml_training dataset.

```r
pml_training <- read.csv("pml-training.csv", na.strings = c("",
    "NA"))
pml_testing <- read.csv("pml-testing.csv", na.strings = c("",
    "NA"))

dim(pml_training)
```

```
## [1] 19622   160
```

```r
summary(pml_training$classe)
```

```
##    A    B    C    D    E
## 5580 3797 3422 3216 3607
```

```r
subtrain <- pml_training[, colSums(is.na(pml_training)) == 0][8:60]
print(paste("Reduced dimension: ", dim(pml_training)))
```

```
## [1] "Reduced dimension:  19622" "Reduced dimension:  160"
```

After preprocessing the pml_training dataset, the number of variables was reduced to 60, however given the nature of this problem, time stamp, user_name data and window data was removed from the dataset, i.e. first 7 columns, because these data corresponded to characteristics of each sample rather than measured values of motion. Finally the data set was reduced to 53 variables; 1 outcome plus 52 numeric/integer covariates.

Prediction Algorithms

Cross validation was done by splitting the pml_training dataset into a test and a training dataset, the latter was used to train 2 different algorithms of machine learning: Linear discriminant (LDA) analysis and Random Forest (RF). LDA was selected as a first approach given its usefulness for data whose outcome is a factor and whose covariates are numeric. On the other hand RF is a method that can help us discriminate which covariates make the best decision tree as a classifier. In this particular exercise since there is no prior knowledge in the characteristics of the experiment, and the covariates most likely will not have a linear dependence with the outcome, RF is an appropiate approach to model the data.

```r
# create partition for testing
set.seed(132)
inTrain <- createDataPartition(y = subtrain$classe, p = 0.75,
    list = FALSE)
training <- subtrain[inTrain, ]
testing <- subtrain[-inTrain, ]

set.seed(332)
modelFitlda <- train(classe ~ ., method = "lda", data = training)
modelFitrf <- train(classe ~ ., method = "rf", data = training,
    trControl = trainControl(method = "cv", number = 3))

predlda <- predict(modelFitlda, testing)
predrf <- predict(modelFitrf, testing)
```

Results

Accuracies and the estimation of the out of sample errors from the two prediction models are the following

```r
print(paste0("Accuracy LDA: ", confusionMatrix(testing$classe,
    predlda)$overall[1]))
```

```
## [1] "Accuracy LDA: 0.708197389885807"
```

```r
print(paste0("Accuracy RF: ", confusionMatrix(testing$classe,
    predrf)$overall[1]))
```

```
## [1] "Accuracy RF: 0.993270799347471"
```

```r
print(paste0("Out of sample error LDA: ", round(1 - confusionMatrix(testing$classe,
    predlda)$overall[1], digits = 4) * 100, "%"))
```

```
## [1] "Out of sample error LDA: 29.18%"
```

```r
print(paste0("Out of sample error RF: ", round(1 - confusionMatrix(testing$classe,
    predrf)$overall[1], digits = 4) * 100, "%"))
```

```
## [1] "Out of sample error RF: 0.67%"
```

The best model that predicts the type of weight lifting is the random forest one. Given that it's accuracy is close to 1, there was no need to test other classifiers or combine predictors to increase accuracy.

```r
modelFitrf$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: 2
##
##         OOB estimate of  error rate: 0.67%
## Confusion matrix:
##      A    B    C    D    E  class.error
## A 4183    1    0    0    1 0.0004778973
## B   21 2820    7    0    0 0.0098314607
## C    0   20 2544    3    0 0.0089598753
## D    0    0   39 2372    1 0.0165837479
## E    0    0    3    2 2701 0.0018477458
```

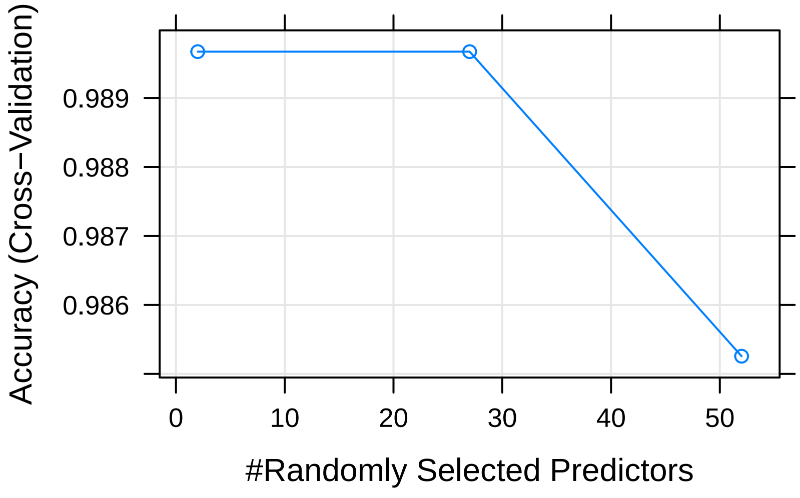

Figure 1: Accuracy as a function of randomly selected covariates

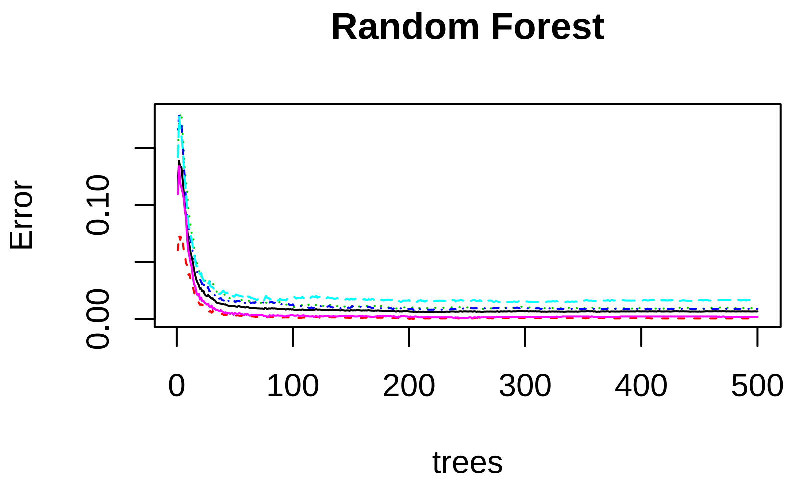

Figure 2: Error as a function of bootstrapped trees

Applying this model to the pml_testing dataset we can predict the outcome for each of the 20 test samples:

```
predict(modelFitrf, pml_testing)
```

```
##  [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
```