Coursera class on Practical Machine Learning: classification of physical exercises

The goal of the project is to use data on 6 participants who were were asked to perform barbell lifts correctly and incorrectly in 5 different ways. Measures from accelerometers on the belt, forearm, arm and dumbell of the participants are used to train a random forest model, in order to predict whether the lifts were performed correctly or incorrectly (4 distinct incorrect ways are used). The model achieved a high accuracy rate in and out of samples and correctly predicted the 20 instances of the test set.

Loading of the data

```
setInternet2(TRUE)
library(plyr); library(ggplot2);library(caret);library(randomForest)
```

```
## Loading required package: lattice
## randomForest 4.6-7
## Type rfNews() to see new features/changes/bug fixes.
```

```
setwd("C:/Users/USer/Documents/GitHub/CourseraPracticalMachineLearning/")
if (!file.exists("train.csv")) {
      train.path<-"https://d396qusza40orc.cloudfront.net/predmachlearn/pml-training.csv"
      download.file(train.path,"train.csv")
      }
train <- read.csv("train.csv",as.is=TRUE)
if (!file.exists("test.csv")) {
      test.path<-"https://d396qusza40orc.cloudfront.net/predmachlearn/pml-testing.csv"
      download.file(test.path,"test.csv")
      }
test <- read.csv("test.csv",as.is=TRUE)
```

The data for this project come from (http://groupware.les.inf.puc-rio.br/har). The testing data is already identified and segregated from the training data that we will use to select the model.

Preprocessing of the data

Let us remove the columns where over 50% of the instances are empty or missing and store the new data under the dataframe training. Then let us remove the first 7 columns which do not concern measurements per se but instead provide information relative to the identity of the person exercising and the conditions under which the measurements were made. Also classe is converted into a factor.

```
training<-train[,colnames(train)[sapply(1:ncol(train),function(.xyz) sum(is.na(train[,.xyz]))+sum(train
training<-training[,-c(1:7)];training$classe<-factor(training$classe)
print(table(training$classe))
```

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

We are left with 52 predictors for the variable Classe and 19622 instances. The 5 levels of the variable Classe are:
Class A: exactly according to the specification
Class B: throwing the elbows to the front
Class C: lifting the dumbbell only halfway
Class D: lowering the dumbbell only halfway
Class E: throwing the hips to the front

fitting of a random tree model

Let us split the training data in two same-size parts: the training set and the cross validation set.

```
inTrain = createDataPartition(training$classe, p = 1/2)[[1]]
training.t = training[ inTrain,];training.cv = training[-inTrain,]
modelFit<-train(classe~.,data=training.t, method="rf",allowParallel=T, trControl = trainControl(method =
print(modelFit$finalModel)
```

```
##
## Call:
##  randomForest(x = x, y = y, mtry = param$mtry, allowParallel = ..1)
##                Type of random forest: classification
##                      Number of trees: 500
## No. of variables tried at each split: 27
##
##         OOB estimate of  error rate: 1.05%
## Confusion matrix:
##      A    B    C    D    E class.error
## A 2784    4    1    0    1    0.002151
## B   16 1872    9    2    0    0.014218
## C    0   22 1683    6    0    0.016365
## D    0    1   25 1579    3    0.018035
## E    0    0    4    9 1791    0.007206
```

```
ConfusionMatrix<-confusionMatrix(training.cv$classe,predict(modelFit,training.cv))
print(ConfusionMatrix)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction    A    B    C    D    E
##          A 2789    0    1    0    0
##          B   25 1854   18    1    0
##          C    0   25 1677    9    0
##          D    0    1   28 1577    2
##          E    0    1    6    5 1791
##
## Overall Statistics
##
##                Accuracy : 0.988
##                  95% CI : (0.985, 0.99)
##     No Information Rate : 0.287
##     P-Value [Acc > NIR] : <2e-16
```

```
##
##                   Kappa : 0.984
##  Mcnemar's Test P-Value : NA
##
## Statistics by Class:
##
##                      Class: A Class: B Class: C Class: D Class: E
## Sensitivity             0.991    0.986    0.969    0.991    0.999
## Specificity             1.000    0.994    0.996    0.996    0.999
## Pos Pred Value          1.000    0.977    0.980    0.981    0.993
## Neg Pred Value          0.996    0.997    0.993    0.998    1.000
## Prevalence              0.287    0.192    0.176    0.162    0.183
## Detection Rate          0.284    0.189    0.171    0.161    0.183
## Detection Prevalence    0.284    0.193    0.174    0.164    0.184
## Balanced Accuracy       0.995    0.990    0.983    0.993    0.999
```

The Out Of Bags estimate of error rate on the training set is 1.03%.
The accuracy rate on the cross validation set is 0.9876 The model is performing quite well in and out of sample, which leads us to keep this model to predict the classe variable on the test set. The model achieved a perfect classification on the 20 instances of the test set.