Machine Learning - Prediction Assignment Writeup

CB

Sunday, November 23, 2014

1. Introduction

1.1. Background information

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.
These type of devices are part of the quantified self movement - a group of enthusiasts who take measurements about themselves regularly to improve their health, to find patterns in their behavior, or because they are tech geeks. 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, the goal will be to use data from accelerometers on the belt, forearm, arm, and dumbell of 6 participants.
They were asked to **perform barbell lifts correctly and incorrectly in 5 different ways**. More precisely, they 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**).
More information is available from the website here: http://groupware.les.inf.puc-rio.br/har (see the section on the Weight Lifting Exercise Dataset).

1.2. Data

The **training data** for this project are available here:
https://d396qusza40orc.cloudfront.net/predmachlearn/pml-training.csv
The **test data** are available here: https://d396qusza40orc.cloudfront.net/predmachlearn/pml-testing.csv

1.3. Goal of the project and submission

The goal of the project is to predict the manner in which they did the exercise. This is the **"classe"** variable in the training set. We may use any of the other variables to predict with. We should create a report describing how we **built your model**, how we used **cross validation**, what we think the **expected out of sample error is**, and **why we made the choices we did**. We will also **use our prediction model to predict 20 different test cases**.

2. Loading the data

We assume that the two datasets are in our working directory.

```
options(stringsAsFactors = FALSE)

originalTraining <- read.csv("pml-training.csv")
originalTesting  <- read.csv("pml-testing.csv")

dim(originalTraining)

## [1] 19622   160

dim(originalTesting)

## [1]  20 160
```

3. Selecting the features

The actual training set is huge (19622 observations of 160 variables). Consequently, it is important to reduce the size of the sets.
Since the data we use come from *accelerometers* on the belt, forearm, arm, and dumbell, we argue that it is important to **use the variables aceleration** as predictors for our model. Erring on the side of interpretability and simplicity, we choose to select only those twelve variables.

```
OkTraining <- originalTraining[ , union(grep("^accel_",
colnames(originalTraining)), grep("classe", colnames(originalTraining)) )]
OkTesting  <- originalTesting[ , union(grep("^accel_",
colnames(originalTesting)),   grep("classe", colnames(originalTesting))  )]

dim(OkTraining)

## [1] 19622    13

dim(OkTesting)

## [1] 20 12

names(OkTraining)

##  [1] "accel_belt_x"     "accel_belt_y"     "accel_belt_z"
##  [4] "accel_arm_x"      "accel_arm_y"      "accel_arm_z"
##  [7] "accel_dumbbell_x" "accel_dumbbell_y" "accel_dumbbell_z"
## [10] "accel_forearm_x"  "accel_forearm_y"  "accel_forearm_z"
## [13] "classe"

names(OkTesting)

##  [1] "accel_belt_x"     "accel_belt_y"     "accel_belt_z"
##  [4] "accel_arm_x"      "accel_arm_y"      "accel_arm_z"
##  [7] "accel_dumbbell_x" "accel_dumbbell_y" "accel_dumbbell_z"
## [10] "accel_forearm_x"  "accel_forearm_y"  "accel_forearm_z"
```

4. Histograms

In order to see the basic properties of three of our fearures, we perform their histrograms.

```
hist(OkTraining$accel_belt_x)

hist(OkTraining$accel_belt_y)


hist(OkTraining$accel_belt_z)
```

5. Splitting the data

We split our training set into **two subsets**: “sampleTraining” (75%) and “sampleTesting” (25%).

```
library(caret)

## Loading required package: lattice
## Loading required package: ggplot2

library(randomForest)

## randomForest 4.6-10
## Type rfNews() to see new features/changes/bug fixes.

inTrain <- createDataPartition(y = OkTraining$classe, p = 0.75, list = FALSE)
sampleTraining <- OkTraining[inTrain, ]
sampleTesting <-  OkTraining[-inTrain, ]
```

6. Fitting the model

We use a **random forest** because it is usually one of the top performing methods, along with boosting.
We also select the **cross-validation** option.

```
library(caret)
set.seed(651)

sampleTraining$classe <-as.factor(sampleTraining$classe)
sampleTraining[, 1:12] <- sapply(sampleTraining[, 1:12], as.numeric)

sampleTesting$classe <-as.factor(sampleTesting$classe)
sampleTesting[, 1:12] <- sapply(sampleTesting[, 1:12], as.numeric)

modFit <- train(classe~ ., data=sampleTraining , method="rf", trControl =
trainControl(method = "cv"))
```

7. Predicting on the sampleTesting dataset

Finally, we predict our model on the sampleTesting dataset.

```
predictions <- predict(modFit , newdata=sampleTesting)
confusionMatrix (predictions, sampleTesting$classe)

## Confusion Matrix and Statistics
```

```
##
##           Reference
## Prediction    A    B    C    D    E
##          A 1342   38    8   15    2
##          B   10  876   17    2    6
##          C   14   24  824   43    9
##          D   29    8    6  743    7
##          E    0    3    0    1  877
##
## Overall Statistics
##
##                Accuracy : 0.9507
##                  95% CI : (0.9442, 0.9565)
##     No Information Rate : 0.2845
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.9376
##  Mcnemar's Test P-Value : 2.119e-11
##
## Statistics by Class:
##
##                      Class: A Class: B Class: C Class: D Class: E
## Sensitivity            0.9620   0.9231   0.9637   0.9241   0.9734
## Specificity            0.9820   0.9912   0.9778   0.9878   0.9990
## Pos Pred Value         0.9552   0.9616   0.9015   0.9369   0.9955
## Neg Pred Value         0.9849   0.9817   0.9922   0.9852   0.9940
## Prevalence             0.2845   0.1935   0.1743   0.1639   0.1837
## Detection Rate         0.2737   0.1786   0.1680   0.1515   0.1788
## Detection Prevalence   0.2865   0.1858   0.1864   0.1617   0.1796
## Balanced Accuracy      0.9720   0.9571   0.9708   0.9560   0.9862
```

So, using cross-validation, we expect the **out of sample error** to be **about 5%**.
NB: It is important to keep in mind that usually, the following holds: *in sample* error < *out of sample* error.

8. Finding the 20 answers

Finally, we obtain the 20 answers of the “Course Project: Submission”.

```
Answers <- predict(modFit , newdata=originalTesting)
print(Answers)
```

```
##  [1] B A C A A E D B A A B C B A E E A B B B
## Levels: A B C D E
```