Practical Machine Learning Course Project

M Tamaru Jones

June 10, 2016

Background

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 dumbell of 6 participants. They were asked to perform barbell lifts correctly and incorrectly in 5 different ways. 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).

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).

Read more: http://groupware.les.inf.puc-rio.br/har#weight_lifting_exercises#ixzz4BDzPUDMz

Read more: http://groupware.les.inf.puc-rio.br/har#weight_lifting_exercises#ixzz4BDzGANnF

Project Objective

In this project, the goal is to use data from accelerometers on the belt, forearm, arm, and dumbell of 6 participants to predict the manner in which they did the exercise.

The "classe" variable in the training set classifies each row of movement as A, B, C, D, or E. The testing set does not have a "classe" variable for the rows of movement, and this is what we are predicting.

Raw 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

The data for this project come from this source: http://groupware.les.inf.puc-rio.br/har.

```
# Data from the website (not stored to hard drive)
trainURL <- "https://d396qusza40orc.cloudfront.net/predmachlearn/pml-training.csv"
testURL <- "https://d396qusza40orc.cloudfront.net/predmachlearn/pml-testing.csv"
trainRaw <- read.csv(url(trainURL), na.strings=c("NA","#DIV/0!",""), header=TRUE, sep = ",")
testRaw <- read.csv(url(testURL), na.strings=c("NA","#DIV/0!",""), header=TRUE, sep = ",")
```

Data Cleanup

I called the following libraries to use with processing the data: caret, randomForest, and rpart.

```
## Loading required package: lattice
```

```
## Loading required package: ggplot2

## randomForest 4.6-12

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

##
## Attaching package: 'randomForest'

## The following object is masked from 'package:ggplot2':
##
##     margin
```

Take a look at the data by calling

```
head(training)
```

The first seven columns do not contain movement information, so I will delete these columns. There are also a number of columns that contain a number of NA, which I will filter out.

```
# Removes fist seven columns (time stamps (3), index, names, windows (2))
trainClean <- trainRaw[, 8:length(names(trainRaw))]
testClean <- testRaw[ ,8:length(names(testRaw))]

# Filtering out columns with NAs
trainClean <- trainClean[ , !apply(trainClean, 2, function(x) any(is.na(x)))]
testClean <- testClean[ , !apply(testClean, 2, function(x) any(is.na(x)))]

# make sure column names match
setdiff(names(trainClean), names(testClean))
```

```
## [1] "classe"
```

```
# differences are: training has "classe" and testing has "problem_id".


# make all the variables the same "type" - numeric (except classe in training)
trainClean[1:52] <- lapply(trainClean[1:52], as.numeric)
testClean[1:53] <- lapply(testClean[1:53], as.numeric)
trainClean$classe <- as.factor(trainClean$classe)
```

With the data now cleaned up - removing NAs and data not related to movement - I am now ready to build my model.

Model Building

To begin, I set the seed to ensure reproducibility. Then I separate the training set into two parts - one for training my model on and the other for cross-validation.

```r
set.seed(222324)

inTrain <- createDataPartition(trainClean$classe, p = 0.75, list = FALSE)
training <- trainClean[inTrain, ]
crossVal <- trainClean[-inTrain, ]
```

I choose to build my model with Random Forests, due to its high accuracy. One of the drawbacks to using this method is the speed, so with a data frame dimension of 14718 x 53, this will take a while. If you predict on the training set and check the accuracy,you will get an accuracy of 1, so I didn't bother.

```r
contRF <- trainControl(method="cv", number=3, verboseIter=FALSE)
modRF <- train(classe ~ . , data = training, method = "rf", trControl = contRF)
```

Cross Validation

Perform the prediction on the cross validation set to view the accuracy. This verifies that the chosen method of random forest has enough accuracy to justify persuing the prediction on the test set.

```r
predictRF <- predict(modRF, crossVal)
confusionMatrix(predictRF, crossVal$classe)$overall["Accuracy"]
```

```
##  Accuracy
## 0.9949021
```

With an accuracy of 99.5%, I am confident in moving forward to the test set with this method of model prediction.

Expected Sample Error

```r
modRF$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: 27
##
##         OOB estimate of  error rate: 0.69%
## Confusion matrix:
##      A    B    C    D    E class.error
## A 4179    4    1    0    1 0.001433692
## B   16 2825    7    0    0 0.008075843
## C    0   15 2545    7    0 0.008570316
## D    0    3   27 2380    2 0.013266998
## E    0    2    7    9 2688 0.006651885
```

The expected sample rate for the random tree method is 0.69% on the cross validation set, which is acceptable.

Test Set Conclusions

Now I will perform the prediction on the test set.

```
predictTest <- predict(modRF, newdata = testClean)
predictTest
```

```
##  [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
```

The final predictions above are what I would submit as classe data for the 20 movements from the test set.