DS-08-ASGN

Rick Hubbard

October 25, 2015

Practical Machine Learning Course Project

Rick Hubbard (25 Oct 2015)

Abstract

The purpose of this investigation is to apply a group of machine learning methods to a common problem set (exercise physiology) to determine which of the chosen methods has the greatest accuracy in predicting a style of exercising with respect to approximately 53 motion-related variables.

(Note: Housekeeping matters–such as loading R Packages–not shown in this report for readability and space saving purposes.)

Data Acquisition

The dataset(s) used in this investigation were compiled by the "HAR" (Human Activity Recognition) project, whose generous allowance of the use of their data and other findings is acknowledged and appreciated (see: http://groupware.les.inf.puc-rio.br/har).

HAR's "training" and "test" datasets were obtained and uploaded into RStudio (version 0.99.486); as shown here:

```r
# Create Data Directory
if(!file.exists("./Data")){
  dir.create("./Data")
}

# Acquire Training data:
fileURL <- "https://d396qusza40orc.cloudfront.net/predmachlearn/pml-training.csv"
if(!file.exists("./Data/pml-training.csv")){
  download.file(fileURL, destfile = "./Data/pml-training.csv", method = "curl")
}

# Acquire Test data
fileURL <- "https://d396qusza40orc.cloudfront.net/predmachlearn/pml-testing.csv"
if(!file.exists("./Data/pml-testing.csv")){
  download.file(fileURL, destfile = "./Data/pml-testing.csv", method = "curl")
}
```

After uploading the datasets into RStudio, they were examined using a combination of Exploratory Data Analysis methods (e.g., sums, averages), as well as visual inspection for values such as "NA" and "#DIV/0!". Based on these evaluations, both the training and test datasets were modified to facilitate subsequent analysis.

```r
# Load & Wrangle Data
pml.TrainR <- read.csv("./Data/pml-training.csv",na.strings=c("NA","","#DIV/0!"),header=TRUE)
pml.TestR <- read.csv("./Data/pml-testing.csv",na.strings=c("NA","","#DIV/0!"),header=TRUE)

# Exclude Superfluous Variables (X:num_window)
pml.TrainR <- pml.TrainR[,-c(1:7)]
pml.TestR <- pml.TestR[,-c(1:7)]

NA.zero.cutoff <- 0.93

# Shape Training Dataset to be useful
# Remove Columns which contains more than NA.zero.cutoff NAs
pml.TrainR <- pml.TrainR[,colSums(is.na(pml.TrainR)) < (nrow(pml.TrainR) * NA.zero.cutoff)]
# Remove Columns which contains more than NA.zero.cutoff Empty Cells
pml.TrainR <- pml.TrainR[,colSums(pml.TrainR == "") < (nrow(pml.TrainR) * NA.zero.cutoff)]
# Confirm Dataset Dimensions
dim(pml.TrainR)
```

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

```r
# Evaluate "TrainR" (Training) Dataset for Presence of Near Zero Variance Predictors
train.near.zero <- sum(nearZeroVar(pml.TrainR, saveMetrics = TRUE)[, 3])
```

Training and Test Datasets

Because there were no “NZV” (Near Zero Value) results, then the capabilities of R’s “caret” package were used and the HAR training dataset was partitioned into “train” and “test” subsets. (Apologies for the namespace overloading!)

```r
train.idx <- createDataPartition(y = pml.TrainR$classe, p = 0.7, list = FALSE)
train.part <- pml.TrainR[train.idx, ]
test.part <- pml.TrainR[-train.idx, ]
```

Generate Candidate Models

Three machine learning methods were chosen as candidates for this investigation; specifically: Decision Trees, Linear Discriminate Analysis, and Random Forest.

For each of the three methods, the “train” partition. of the HAR provided dataset was used to generate (“fit”) a model. Then the fitted model was applied to the “test” partition to gauge accuracy and Out-of-Sample error rates.

```r
# Generate Models
# Decision Trees
model.DTree <- train(classe ~ ., data = train.part, method = "rpart")
predict.DTree <- predict(model.DTree, newdata = test.part)
cM.DTree <- confusionMatrix(data = predict.DTree, test.part$classe)
cM.DTree.accuracy <- cM.DTree$overall[1]

# Linear Discriminate Analysis
```

```r
model.LDA <- train(classe ~ ., data = train.part, method = "lda")
predict.LDA <- predict(model.LDA, newdata = test.part)
cM.LDA <- confusionMatrix(data = predict.LDA, test.part$classe)
cM.LDA.accuracy <- cM.LDA$overall[1]

# Random Forest
model.RF <- randomForest(classe ~., data = train.part)
predict.RF <- predict(model.RF, newdata = test.part)
cM.RF <- confusionMatrix(data = predict.RF, test.part$classe)
cM.RF.accuracy <- cM.RF$overall[1]
```

Validity Assessment

Once the models were generated and tested, then their overall accuracy and Out-of-Sample error rates were evaluated.

```r
# Analysis of Model "Accuracy" Results
cM.DTree.accuracy
```

```
##  Accuracy
## 0.4943076
```

```r
cM.LDA.accuracy
```

```
##  Accuracy
## 0.6961767
```

```r
cM.RF.accuracy
```

```
##  Accuracy
## 0.9949023
```

```r
# Compute Out-of-Sample Error Rates
OOS.DTree <- 1 - cM.DTree.accuracy
OOS.LDA <- 1 - cM.LDA.accuracy
OOS.RF <- 1 - cM.RF.accuracy

OOS.DTree
```

```
##  Accuracy
## 0.5056924
```

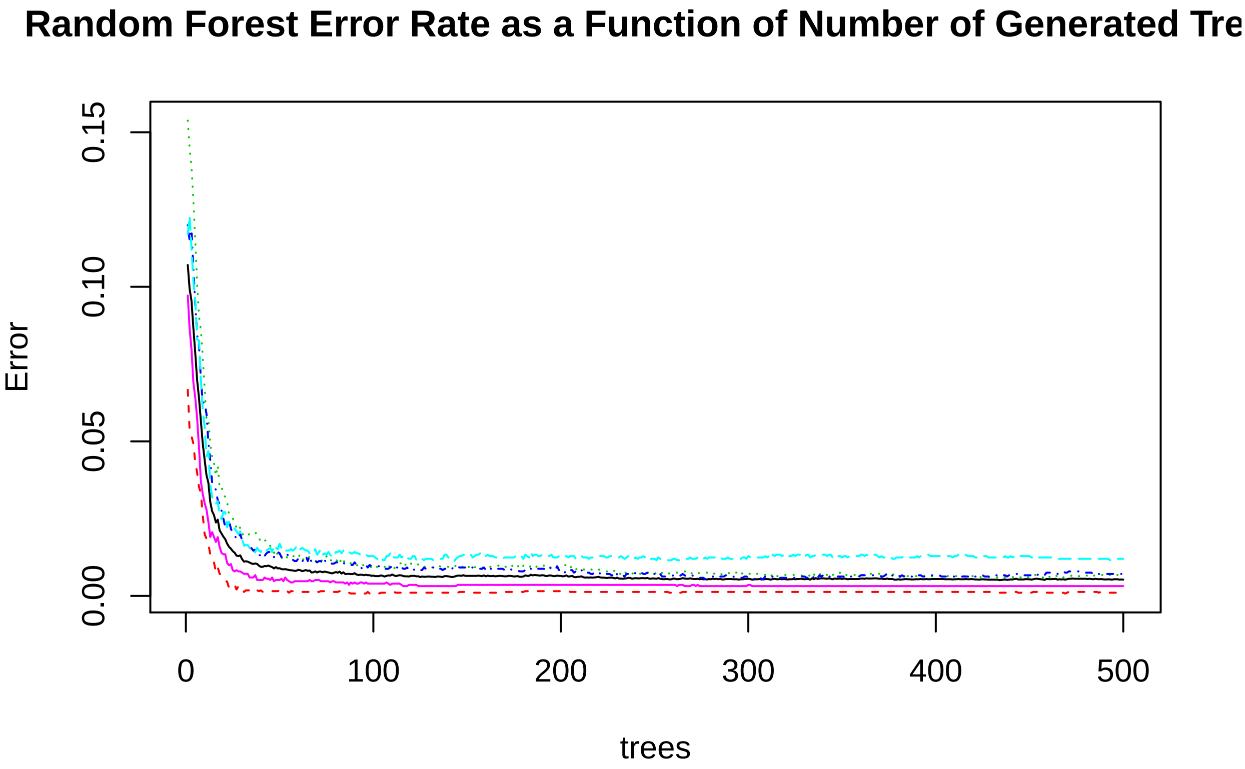
```r
OOS.LDA
```

```
##  Accuracy
## 0.3038233
```

```r
OOS.RF
```

```
##    Accuracy
## 0.005097706
```

Overall, the Random Forest approach had the highest degree of Accuracy (0.9949023)…and the lowest degree of Out-of-Sample error rate (0.0050977).



Final Prediction

The last step in this investigation was to apply the fitted models to the original HAR-provided "Test" dataset (again, apologies for namespace overloading!).

```r
# Analysis of RF on Test Data
final.result <- predict(model.RF, newdata = pml.TestR)
final.result
```

```
##  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
```

```r
# Write Prediction Files for Submission
fr.output <- as.character(final.result)
pml_write_files(fr.output)
```