Practical Machine Learning Peer Graded Assignment

David Li

16 June 2019

Executive Summary

This paper explores the use of machine learning techniques to predict the outcome of the five barbell lifting techniques based on accelerometer data. Using the training data, we designed an algorithm based on the random forest technique to accurately predict the outcomes. It is concluded the model possesses over 99% in predictive accuracy and 0.17% out of sample error, resulting in all predictions made in the testing dataset to be correct.

Introduction

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 information is available from the website here: http://groupware.les.inf.puc-rio.br/har (see the section on the Weight Lifting Exercise Dataset).

The training data for this project are downloaded from: https://d396qusza40orc.cloudfront.net/predmachlearn/pml-training.csv

The test data are downloaded from: 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.

It is worth citing Groupware@LES for being generous in allowing their data to be used for this assignment.

Objective

The goal of this report is to predict the manner the participants have performed the exercise. The “classe” variable is the outcome which categorically designates how the participants performed the exercise. All other variables are predictor variables. The report will also stipualte how the model was built, conduct cross-validation and expectation of the sample error rate. Finally, we will apply the model on the test set of 20 seperate cases.

Preparation

We will need to load the required packages to enable our analysis.

```
library(caret)
```

```
## Warning: package 'caret' was built under R version 3.5.3
```

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

```
## Loading required package: ggplot2
```

```
## Warning: package 'ggplot2' was built under R version 3.5.3
```

```
library(rpart)
library(knitr)
```

```
## Warning: package 'knitr' was built under R version 3.5.3
```

```
library(randomForest)
```

```
## Warning: package 'randomForest' was built under R version 3.5.3
```

```
## randomForest 4.6-14
```

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

```
##
## Attaching package: 'randomForest'
```

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

```
library(ElemStatLearn)
```

```
## Warning: package 'ElemStatLearn' was built under R version 3.5.3
```

```
library(corrplot)
```

```
## Warning: package 'corrplot' was built under R version 3.5.3
```

```
## corrplot 0.84 loaded
```

```
library(ggplot2)
set.seed(5849)#For reproducibility purposes
setwd("C:/Users/david/Desktop/Coursera/Module 8/Peer Graded Assignment/PracticalMachineLearning")
```

Now we will load in our training and test sets:

```
rawtrainset <- read.csv("./pml-training.csv",header = T, sep = ",",na.strings = c("NA",""))
rawtestset <- read.csv("./pml-testing.csv",header = T, sep = ",",na.strings = c( "NA",""))
```

The next step is to begin separating our dataset as a training and testing set. The dataset will also be cleaned before

```
rawtrainset <- rawtrainset[ ,-1]
inTrain = createDataPartition(rawtrainset$classe, p=0.60, list = F)
training = rawtrainset[inTrain,]
validating = rawtrainset[-inTrain,]
```

In terms of the model selection, we will be using the random forest method. The main reason for this is the benefit of prediction accuracy while noting the potential for overfitting and reducing interpretablity.

In building the algorithm, the dataset must be checked on the potential for columns to have missing data. We will set a rule that determines that columns with less than 60% of data are removed.

```
sum(colSums(!is.na(training[,-ncol(training)])) < 0.6*nrow(training))
```

```
## [1] 100
```

Next, we set a criteria to remove columns that do not meet the 60% data threshold before we apply the model.

```
Keep <- c((colSums(!is.na(training[,-ncol(training)])) >= 0.6*nrow(training)))
training   <-  training[,Keep]
validating <- validating[,Keep]
```

Since we have selected the random forest methodology, a cross-validation or separate test to attain an unbiased estimate of the test set error is not required. Instead this is estimated during execution of the model.

Modelling Execution

```
model <- randomForest(classe~.,data=training)
model
```

```
##
## Call:
##  randomForest(formula = classe ~ ., data = training)
##                Type of random forest: classification
##                      Number of trees: 500
## No. of variables tried at each split: 7
##
##         OOB estimate of  error rate: 0.14%
## Confusion matrix:
##      A    B    C    D    E  class.error
## A 3347    1    0    0    0 0.0002986858
## B    4 2274    1    0    0 0.0021939447
## C    0    3 2050    1    0 0.0019474197
## D    0    0    3 1927    0 0.0015544041
## E    0    0    0    4 2161 0.0018475751
```

Model Evaluation

The model will be evaluated through the confusion matrix function to get a good sense of the model accuracy.

```
confusionMatrix(predict(model,newdata=validating[,-ncol(validating)]),validating$classe)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction    A    B    C    D    E
##          A 2232    1    0    0    0
##          B    0 1517    2    0    0
##          C    0    0 1365    9    0
##          D    0    0    1 1275    1
##          E    0    0    0    2 1441
##
## Overall Statistics
##
##                Accuracy : 0.998
##                  95% CI : (0.9967, 0.9988)
##     No Information Rate : 0.2845
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.9974
##
##  Mcnemar's Test P-Value : NA
##
## Statistics by Class:
##
##                      Class: A Class: B Class: C Class: D Class: E
## Sensitivity            1.0000   0.9993   0.9978   0.9914   0.9993
## Specificity            0.9998   0.9997   0.9986   0.9997   0.9997
## Pos Pred Value         0.9996   0.9987   0.9934   0.9984   0.9986
## Neg Pred Value         1.0000   0.9998   0.9995   0.9983   0.9998
## Prevalence             0.2845   0.1935   0.1744   0.1639   0.1838
## Detection Rate         0.2845   0.1933   0.1740   0.1625   0.1837
## Detection Prevalence   0.2846   0.1936   0.1751   0.1628   0.1839
## Balanced Accuracy      0.9999   0.9995   0.9982   0.9956   0.9995
```

As we can see, for this particular model, we obtain a 99.73% accuracy over the validation set. While our out of sample error is 0.17%.

Model Testing

Now let's involve the testing dataset to test the predictive capacity of our model. We will need to conduct some cleaning on the testing dataset such that they are coerced for the same class as the training dataset.

```
rawtestset <- rawtestset[,-1] # Remove ID column which si the first column
rawtestset <- rawtestset[ , Keep] # Keep the same columns of testing dataset
rawtestset <- rawtestset[,-ncol(rawtestset)] # Remove the problem ID
testing <- rbind(training[100, -59] , rawtestset)  # Coerce testing dataset to the same structure as trainin
g dataset
row.names(testing) <- c(100, 1:20) # Apply the ID Row to row.names and 100 for the dummy row from the testin
g dataset
```

Predict with the testing dataset

```
predictions <- predict(model,newdata=testing[-1,])
predictions
```

```
##  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
```

Using the predictions made in this report, we submitted the answers in the quiz with 100% accuracy.