Practical Machine Learning: Assignment 1

Vivek Verma

27 March 2016

Machine Learning: An Analysis of the Weight Lifting Exercises Dataset

Executive Summary

This project, the goal is to analyze data from accelerometers on the belt, forearm, arm, and dumbell of six participants. They were asked to perform barbell lifts correctly and incorrectly in five different ways. Specifically, the goal of this machine learning exercise is to predict the manner in which the participants did the exercise-that is, to predict the "classe" variable found in the training set. The prediction model will then be used to predict twenty different test cases, as provided in the testing dataset.

Data Processing and Analysis

We begin by loading the required libraries and reading in the training and testing datasets, assigning missing values to entries that are currently 'NA' or blank.

```
library(corrplot)
library(caret)
```

```
wm <- read.csv("pml-training.csv", header = TRUE, na.strings = c("NA", ""))
wm_test <- read.csv("pml-testing.csv", header = TRUE, na.strings = c("NA", ""))
```

Columns in the orignal training and testing datasets that are mostly filled with missing values are then removed. To do this, count the number of missing values in each column of the full training dataset. We use those sums to create a logical variable for each column of the dataset. The logical variable's value is 'TRUE' if a column has no missing values (i.e. if the colSums = 0). If there are missing values in the column, the logical variable's value corresponding to that column will be 'FALSE'.

Applying the logical variable to the columns of the training and testing datasets will only keep those columns that are complete.

Training dataset has fewer variables to review. Further, final testing dataset has consistent columns in it when compared with those in our slimmed-down training dataset. This will allow the fitted model (based on our training data) to be applied to the testing dataset.

```
csums <- colSums(is.na(wm))
csums_log <- (csums == 0)
training_fewer_cols <- wm[, (colSums(is.na(wm)) == 0)]
wm_test <- wm_test[, (colSums(is.na(wm)) == 0)]
```

Lnother logical vector in order to delete additional unnecessary columns from the training and testing datasets. Column names in the dataset containing the entries shown in the 'grepl' function will have a value of 'TRUE' in the logical vector. Since these are the columns we want to remove, we apply the negation of the logical vector against the columns of our dataset.

```
del_cols_log <- grepl("X|user_name|timestamp|new_window", colnames(training_fewer_cols))
training_fewer_cols <- training_fewer_cols[, !del_cols_log]
wm_test_final <- wm_test[, !del_cols_log]
```

Updated training dataset into a training dataset (70% of the observations) and a validation dataset (30% of the observations). This validation dataset will allow us to perform cross validation when developing model.

```
inTrain = createDataPartition(y = training_fewer_cols$classe, p = 0.7, list = FALSE)
small_train = training_fewer_cols[inTrain, ]
small_valid = training_fewer_cols[-inTrain, ]
```

Dataset contains 54 variables, with the last column containing the 'classe' variable we are trying to predict. Begin by looking at the correlations between the variables in our dataset. WWant to remove highly correlated predictors from our analysis and replace them with weighted combinations of predictors. This may allow a more complete capture of the information available.

```
corMat <- cor(small_train[,-54])
corrplot(corMat, order = "FPC", method = "color", type = "lower", tl.cex = 0.8, tl.col = rgb(0,0,0))
```