Using Machine Learning to Predict Physical Exercise Practice Quality

Project assignment of the Practical Machine Learning course from the Johns Hopkins University's Data Science Specialization on Coursera.

Background

Using devices such as Jawbone Up, Nike FuelBand, Fitbit, Garmin Vivofit, Microsoft Band, and Apple iWatch, among other products available in the market, 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, your 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 PUC-Rio website (http://groupware.les.inf.puc-rio.br/har) (see the section on the Weight Lifting Exercise Dataset).

Loading and Cleaning the Testing and Training Data

The data for this project come from the PUC Rio repository (http://groupware.les.inf.puc-rio.br/har). The two files in CSV format are: training data (https://d396qusza40orc.cloudfront.net/predmachlearn/pml-training.csv) and test data (https://d396qusza40orc.cloudfront.net/predmachlearn/pml-testing.csv). I would like to thank the PUC Rio team for allowing their data to be used for this kind of assignment.

```
library(caret)
```

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

```
RawTestData <- read.csv("pml-testing.csv",na.strings=c("NA",""))
RawTrainData <- read.csv("pml-training.csv",na.strings=c("NA",""))
NAs <- apply(RawTrainData,2,function(x) {sum(is.na(x))})
CleanTrainData <- RawTrainData[,which(NAs == 0)]
CleanTestData <- RawTestData[,which(NAs == 0)]
```

In addition to the datasets above, a seed value (i.e. 1027) was used in this assignment to assure its reproducibility.

```
set.seed(1027)
options(warn=-1)
```

Training and Cross Validation datasets

We used the code below to create the training and cross validation datasets. We used 80% of the file for training and the remaining 20% for cross validation.

```
trainIndex <- createDataPartition(y = CleanTrainData$classe, p=0.8,list=FALSE)
trainSet <- CleanTrainData[trainIndex,]
crossValidationSet <- CleanTrainData[-trainIndex,]
# Unnecessary variables were removed using the code below
removeIndex <- as.integer(c(1,2,3,4,5,6))
trainSet <- trainSet[,-removeIndex]
testSet <- CleanTestData[,-removeIndex]
```

Training code

The following training code was applied to generate the model.

```
mytrControl = trainControl(method = "cv", number = 4)
modelFit <- train(trainSet$classe ~.,data = trainSet, method="rf", trControl = mytrControl)
```

```
## Loading required package: randomForest
## randomForest 4.6-10
## Type rfNews() to see new features/changes/bug fixes.
```

```
modelFit
```

```
## Random Forest
##
## 15699 samples
##    53 predictor
##     5 classes: 'A', 'B', 'C', 'D', 'E'
##
## No pre-processing
## Resampling: Cross-Validated (4 fold)
##
## Summary of sample sizes: 11774, 11774, 11774, 11775
##
## Resampling results across tuning parameters:
##
##   mtry  Accuracy   Kappa      Accuracy SD   Kappa SD
##    2    0.9940123  0.9924257  0.0022835671  0.002889222
##   27    0.9972609  0.9965352  0.0009841666  0.001245005
##   53    0.9949041  0.9935538  0.0011952136  0.001511921
##
## Accuracy was used to select the optimal model using  the largest value.
## The final value used for the model was mtry = 27.
```

Errors Calculation using the Cross Validation dataset

The errors were calculated using the cross validation dataset.

```
predicted <- predict(modelFit, crossValidationSet)
SampleError <- sum(predicted == crossValidationSet$classe)/nrow(crossValidationSet)
```

Generating data for the prediction vector for the Assigment Submission

```
answers <- predict(modelFit, testSet)
```

Generating the Answers and Answers files

Using the code provided by the PML course instructors we were able to generate 20 files with the answers there were uploaded in the second portion of the assignment.

```
pml_write_files = function(x){
  n = length(x)
  for(i in 1:n){
    filename = paste0("problem_id_",i,".txt")
    write.table(x[i],file=filename,quote=FALSE,row.names=FALSE,col.names=FALSE)
  }
}

pml_write_files(answers)
```

The answers are presented below:

```
answers
```

```
##  [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
```

This model was able to predict all 20 answers correctly.