Practical Machine Learning Course Project

Executive Summary

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 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, which is the "classe" variable in the training set.

With the Random Forest model, we are able to predict how well a person is preforming an excercise with accuracy of 99.44%.

Data Processing

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

Download Training Data

```
pmlTraining <- "./pml-training.csv"
if (!file.exists(pmlTraining))
{
        fileUrl <- "https://d396qusza40orc.cloudfront.net/predmachlearn/pml-training.csv"
        download.file(fileUrl, destfile=pmlTraining)
        dateDownloaded <-date()
}
training <- read.csv(pmlTraining)
```

Cleanup Training Data

We first get all of the columns containing belt, forearm, arm, and dumbbell.

There are many columns with empty and NA values. So we need to remove the columns with empty or NA values and only keep the columns with valid numbers.

```
index <- grepl("belt|forearm|arm|dumbbell", names(training), ignore.case=TRUE)
isAnyMissing <- sapply(training, function (x) any(is.na(x) | x == ""))
newTraining <- training[, index & !isAnyMissing==TRUE]

# include the last column "classe" to the new training data set
newTraining <- cbind(training$classe, newTraining)

# rename the first column to classe
```

```
colnames(newTraining)[1] <- "classe"

# set the factor for the first column
newTraining$classe <- factor(newTraining$classe)

names(newTraining)
```

```
##  [1] "classe"               "roll_belt"            "pitch_belt"
##  [4] "yaw_belt"             "total_accel_belt"     "gyros_belt_x"
##  [7] "gyros_belt_y"         "gyros_belt_z"         "accel_belt_x"
## [10] "accel_belt_y"         "accel_belt_z"         "magnet_belt_x"
## [13] "magnet_belt_y"        "magnet_belt_z"        "roll_arm"
## [16] "pitch_arm"            "yaw_arm"              "total_accel_arm"
## [19] "gyros_arm_x"          "gyros_arm_y"          "gyros_arm_z"
## [22] "accel_arm_x"          "accel_arm_y"          "accel_arm_z"
## [25] "magnet_arm_x"         "magnet_arm_y"         "magnet_arm_z"
## [28] "roll_dumbbell"        "pitch_dumbbell"       "yaw_dumbbell"
## [31] "total_accel_dumbbell" "gyros_dumbbell_x"     "gyros_dumbbell_y"
## [34] "gyros_dumbbell_z"     "accel_dumbbell_x"     "accel_dumbbell_y"
## [37] "accel_dumbbell_z"     "magnet_dumbbell_x"    "magnet_dumbbell_y"
## [40] "magnet_dumbbell_z"    "roll_forearm"         "pitch_forearm"
## [43] "yaw_forearm"          "total_accel_forearm"  "gyros_forearm_x"
## [46] "gyros_forearm_y"      "gyros_forearm_z"      "accel_forearm_x"
## [49] "accel_forearm_y"      "accel_forearm_z"      "magnet_forearm_x"
## [52] "magnet_forearm_y"     "magnet_forearm_z"
```

After the cleanup, with exclusion of the outcome variable "classe", the new dataset contains 52 predictor variables (compared to 159 predictors before the cleanup).

Build the Model

We first split the dataset into a typical 60% training and 40% testing dataset.

```
library(caret)
library(randomForest)
set.seed(32343)
inTrain <- createDataPartition(y=newTraining$classe, p=0.60, list=FALSE)
splitTraining <- newTraining[ inTrain,]
splitTesting  <- newTraining[-inTrain,]
```

Then we use "rpart" (Recursive Partitioning and Regression Trees) method and "lda" (Linear Discriminant Analysis) to build 2 models and check the model accuracy using Confusion Matrix on the remaining 40% of test data.

```
# Use rpart: Recursive Partitioning and Regression Trees
# NOTE: It takes long time to generae training model with rpart. Be patient!
library(rpart)
modFit2 <- train(classe~., data=splitTraining, method="rpart")
pred2 <- predict(modFit2, newdata=splitTesting)
c2 <- confusionMatrix(pred2, splitTesting$classe)$overall
```

```
# Use lda: Linear Discriminant Analysis.
library(MASS)
modFit3 <- train(classe~., data=splitTraining, method="lda")
pred3 <- predict(modFit3, newdata=splitTesting)
c3 <- confusionMatrix(pred3, splitTesting$classe)$overall
# Model accuracy
accuracyrate <- cbind(c2[1], c3[1])
colnames(accuracyrate) <- c("rpart", "lda")
accuracyrate
```

```
##              rpart       lda
## Accuracy 0.5477951 0.7039256
```

The model accuracy of "rpart" is 54.78% and the model accuracy of lda is 70.39%, which is not high.

Random Forest is one of the two top performing algorithms along with bootsing in predictions contests. Although it is difficult to interpret, it is often very accurate, Thus we create the third training model with Random Forest.

```
modFit3 <- randomForest(splitTraining$classe ~ ., data = splitTraining)
modFit3
```

```
##
## Call:
##  randomForest(formula = splitTraining$classe ~ ., data = splitTraining)
##                Type of random forest: classification
##                      Number of trees: 500
## No. of variables tried at each split: 7
##
##         OOB estimate of  error rate: 0.66%
## Confusion matrix:
##      A    B    C    D    E  class.error
## A 3345    3    0    0    0 0.0008960573
## B   15 2259    5    0    0 0.0087757789
## C    0   15 2034    5    0 0.0097370983
## D    0    0   25 1904    1 0.0134715026
## E    0    0    4    5 2156 0.0041570439
```

Cross-Validation

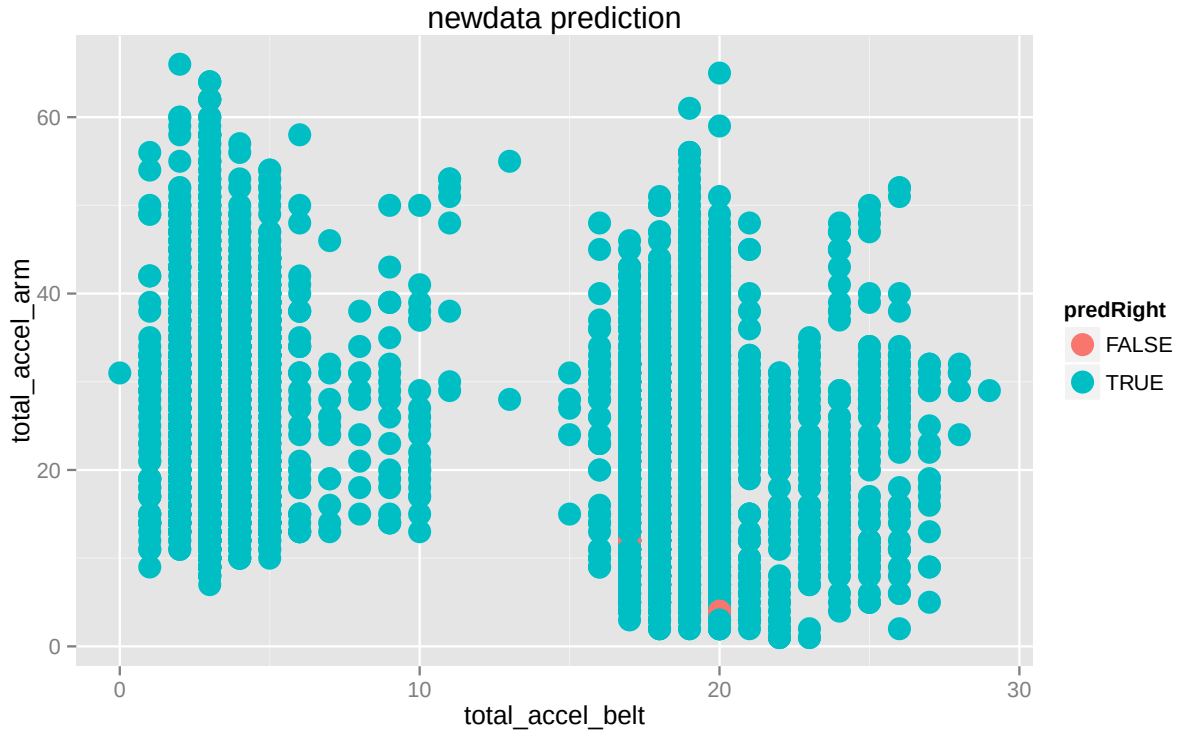
The Random Forest model is used to classify the remaining 40% of the data. A Confusion Matrix is created by passing the predictions from the model and the actual classifications, which determines the accuracy of the model.

```
predictions <- predict(modFit3, newdata=splitTesting)
confusionMatrix(predictions, splitTesting$classe)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction    A    B    C    D    E
```

```
##          A 2229    9    0    0    0
##          B    2 1508   14    0    0
##          C    0    1 1350   10    0
##          D    0    0    4 1275    3
##          E    1    0    0    1 1439
##
## Overall Statistics
##
##                Accuracy : 0.9943
##                  95% CI : (0.9923, 0.9958)
##     No Information Rate : 0.2845
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.9927
##  Mcnemar's Test P-Value : NA
##
## Statistics by Class:
##
##                      Class: A Class: B Class: C Class: D Class: E
## Sensitivity            0.9987   0.9934   0.9868   0.9914   0.9979
## Specificity            0.9984   0.9975   0.9983   0.9989   0.9997
## Pos Pred Value         0.9960   0.9895   0.9919   0.9945   0.9986
## Neg Pred Value         0.9995   0.9984   0.9972   0.9983   0.9995
## Prevalence             0.2845   0.1935   0.1744   0.1639   0.1838
## Detection Rate         0.2841   0.1922   0.1721   0.1625   0.1834
## Detection Prevalence   0.2852   0.1942   0.1735   0.1634   0.1837
## Balanced Accuracy      0.9985   0.9954   0.9926   0.9952   0.9988
```

```
library(ggplot2)
splitTesting$predRight <- predictions==splitTesting$classe
qplot(total_accel_belt, total_accel_arm, colour=predRight, data=splitTesting, main="newdata prediction"
```



The accuracy of the above model is 99.44% which is very high. By comparing with "rpart" and "lda", it turns out Random Forest is a great model to fit the given training dataset.

Predictions of 20 Test Cases

We load a new testing data set and perform the same data processing and cleanup as above. Then the random forest model is used to predict the classifications of the 20 results of this new testing data.

```
pmlTesting <- "./pml-testing.csv"
if (!file.exists(pmlTesting))
{
        fileUrl <- "https://d396qusza40orc.cloudfront.net/predmachlearn/pml-testing.csv"
        download.file(fileUrl, destfile=pmlTesting)
        dateDownloaded <-date()
}
testing <- read.csv(pmlTesting)

# Process and clean up the data to get the variables containing belt, forearm, arm, and dumbbell
index <- grepl("belt|forearm|arm|dumbbell", names(testing), ignore.case=TRUE)

# Find the columns with "NA" or empty values
isAnyMissing <- sapply(testing, function (x) any(is.na(x) | x == ""))

# Generate clean test data
clean_test_data <- testing[, index & !isAnyMissing==TRUE]
clean_test_data <- cbind(testing$problem_id, clean_test_data)
colnames(clean_test_data)[1] <- "classe"
```

```r
# predict the data
predictTest <- predict(modFit3, newdata=clean_test_data)
predictTest
```

```
##  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
```