Prediction Assignment Write-Up

Background (from assignment)

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

Preparing the Data

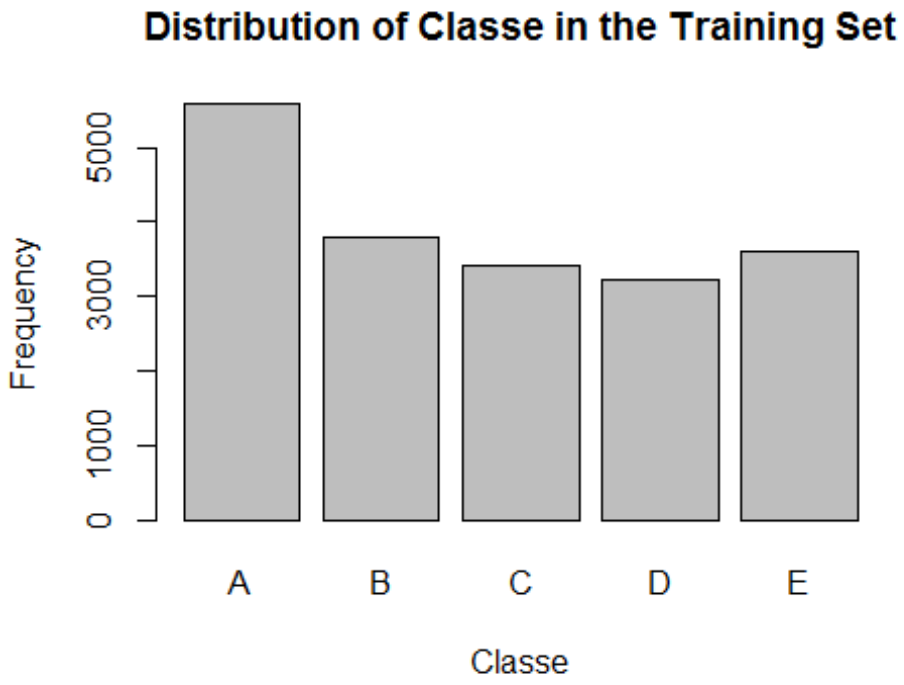
Begin by downloading the two data sets.

```
library(caret)

## Warning: package 'caret' was built under R version 3.2.4

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

library(randomForest)

## Warning: package 'randomForest' was built under R version 3.2.4

## randomForest 4.6-12
## Type rfNews() to see new features/changes/bug fixes.

trainURL <- "https://d396qusza40orc.cloudfront.net/predmachlearn/pml-
training.csv"
testURL <- "https://d396qusza40orc.cloudfront.net/predmachlearn/pml-
testing.csv"
download.file(trainURL, dest="training.csv")
download.file(testURL, dest="testing.csv")
training <- read.csv("training.csv", na.strings = c("", "NA", "#DIV/0!"))
testing <- read.csv("testing.csv", na.strings = c("", "NA", "#DIV/0!"))
```

A quick look at the dimensions confirms that the training set is quite large, and looking at head(training) reveals that some of the data may be incomplete or may not be necessary for our purposes. We will start by removing these.

```
training <- training[,-c(1:7)]
testing <- testing[,-c(1:7)]
bad <- nearZeroVar(training)
training <- training[, -bad]
good <- !apply(training, 2, function(x) sum(is.na(x)) || sum(x=="""))
training <- training[, good]
```

We're interested in how people do the exercise, housed in the "classe" variable. There are five levels: A ("exactly according to the specification"), B ("throwing the eblows to the front"), C ("lifting the dumbbell only halfway"), D ("lowering the dumbbell only halfway"), and E ("throwing the hips to the front"). We can visualize the frequency of each "classe" in the below plot.



Because we have such a large number, we can create subsets within the training set to allow us to validate. 25% of the training data set will be set aside for this purpose.

```
set.seed(12345)
training_set <- createDataPartition(training$classe, p = 0.75, list=FALSE)
training <- training[training_set,]
validation <- training[-training_set,]
```

Prediction Models

```
model <- randomForest(classe ~ ., data=training, method="class")
pred1 <- predict(model, training)
confusionMatrix(pred1, training$classe)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction    A    B    C    D    E
##          A 4185    0    0    0    0
##          B    0 2848    0    0    0
##          C    0    0 2567    0    0
##          D    0    0    0 2412    0
##          E    0    0    0    0 2706
##
## Overall Statistics
##
##                Accuracy : 1
##                  95% CI : (0.9997, 1)
##     No Information Rate : 0.2843
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 1
##  Mcnemar's Test P-Value : NA
##
## Statistics by Class:
##
##                      Class: A Class: B Class: C Class: D Class: E
## Sensitivity            1.0000   1.0000   1.0000   1.0000   1.0000
## Specificity            1.0000   1.0000   1.0000   1.0000   1.0000
## Pos Pred Value         1.0000   1.0000   1.0000   1.0000   1.0000
## Neg Pred Value         1.0000   1.0000   1.0000   1.0000   1.0000
## Prevalence             0.2843   0.1935   0.1744   0.1639   0.1839
## Detection Rate         0.2843   0.1935   0.1744   0.1639   0.1839
## Detection Prevalence   0.2843   0.1935   0.1744   0.1639   0.1839
## Balanced Accuracy      1.0000   1.0000   1.0000   1.0000   1.0000
```

Based on our training data, this model appears to be quite accurate. We will test on the validation set to confirm this.

```
pred2 <- predict(model, validation)
confusionMatrix(pred2, validation$classe)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction    A    B    C    D    E
##          A 1029    0    0    0    0
##          B    0  717    0    0    0
##          C    0    0  652    0    0
##          D    0    0    0  594    0
##          E    0    0    0    0  691
##
## Overall Statistics
##
##                Accuracy : 1
```

```
##                  95% CI : (0.999, 1)
##     No Information Rate : 0.2794
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 1
##  Mcnemar's Test P-Value : NA
##
## Statistics by Class:
##
##                      Class: A Class: B Class: C Class: D Class: E
## Sensitivity            1.0000   1.0000    1.000   1.0000   1.0000
## Specificity            1.0000   1.0000    1.000   1.0000   1.0000
## Pos Pred Value         1.0000   1.0000    1.000   1.0000   1.0000
## Neg Pred Value         1.0000   1.0000    1.000   1.0000   1.0000
## Prevalence             0.2794   0.1947    0.177   0.1613   0.1876
## Detection Rate         0.2794   0.1947    0.177   0.1613   0.1876
## Detection Prevalence   0.2794   0.1947    0.177   0.1613   0.1876
## Balanced Accuracy      1.0000   1.0000    1.000   1.0000   1.0000
```

Considering both the training and validation subsets, we can confirm that the accuracy of the model is very high. The accuracy of the model in both the training and validation sets was 100%.

```
print(model, digits=3)
```

```
##
## Call:
##  randomForest(formula = classe ~ ., data = training, method = "class")
##                Type of random forest: classification
##                      Number of trees: 500
## No. of variables tried at each split: 7
##
##         OOB estimate of  error rate: 0.46%
## Confusion matrix:
##      A    B    C    D    E  class.error
## A 4183    1    1    0    0 0.0004778973
## B   13 2834    1    0    0 0.0049157303
## C    0   12 2553    2    0 0.0054538372
## D    0    0   22 2387    3 0.0103648425
## E    0    0    2   10 2694 0.0044345898
```

Basing estimated out of sample error rate on the confusionMatrix alone would indicate that it is 0%. However, the OOB (out-of-bag) of error rate is 0.46%. Although not 0%, this nonetheless is a very reasonable error rate and so this model will be selected.

Quiz

```
answers <- predict(model, testing)
answers
```

```
##  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
```

References

Velloso, E.; Bulling, A.; Gellersen, H.; Ugulino, W.; Fuks, H. Qualitative Activity Recognition of Weight Lifting Exercises. Proceedings of 4th International Conference in Cooperation with SIGCHI (Augmented Human '13) . Stuttgart, Germany: ACM SIGCHI, 2013. Available at: http://groupware.les.inf.puc-rio.br/har