Analysis of HAR Data to Predict Exercise Performance Quality Synopsis

The HAR (Human Activity Recognition) dataset is a collection of accelerometer measurements collected from a group of people performing certain activities. These measurements often are used to provide information on how much they do activity, but rarely on how well they do it. In this study users were asked to perform barbell lifts correctly and incorrectly in 5 different ways (classe)

- Class A - exactly according to the specification
- Class B - throwing the elbows to the front
- Class C - lifting the dumbbell only halfway
- Class D - lowering the dumbbell only halfway
- Class E - throwing the hips to the front

The goal of this project is to develop a prediction model using machine learning on a training data set and predict the manner the exercise was completed.

Loading Data

Load in the required libraries and the pml-training dataset

```r
library(caret)
```

```
## Warning: package 'caret' was built under R version 3.1.3
```

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

```r
library(randomForest)
```

```
## Warning: package 'randomForest' was built under R version 3.1.2
```

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

```r
pmlData<- read.csv("pml-training.csv", header = TRUE)
```

Cleaning and Preprocessing

A visual inspection of the pml training data shows columns that do not have accelerometer data and there are many columns that contain little or no data. In order to reduce the dataset I will remove the non-accelerometer columns and also remove the columns where the amount of NA data is greater than 10%

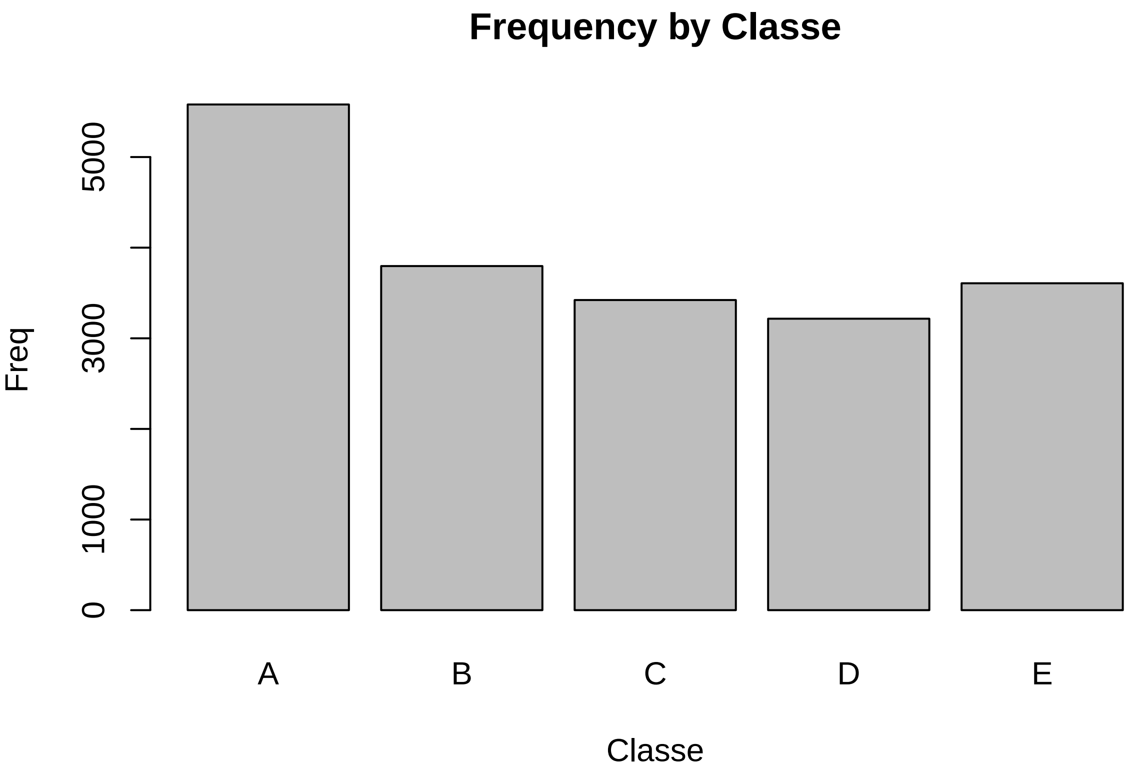
```r
##remove the non-accelerometer data
pmlDataClean <- pmlData[,-c(1:7)]

## convert empty records as NA and remove columns where the amount of NA is greater than 10%
pmlDataClean[pmlDataClean==""] <- NA
pmlDataClean <- pmlDataClean[ lapply( pmlDataClean, function(x) sum(is.na(x)) / length(x) ) < 0.1 ]
```

This reduced the number of variables from 160 to 53 (including the classe variable)

Exploratory Analysis

A look at the frequency of the class variable…

```
plot(pmlDataClean$classe, main="Frequency by Classe", xlab="Classe", ylab="Freq")
```



Setting Cross Validation

I will begin the modelling process by first splitting the training data into train and test subsets. I kept 75% in the training data and the remaining in the test set that I will use for cross-valdiating my model.

```
set.seed(1122)

inTrain <- createDataPartition(y=pmlDataClean$classe, p=0.75, list=FALSE)

train <-pmlDataClean[inTrain,]
test <-pmlDataClean[-inTrain,]
```

Model Selection

Many methods of classification were attempted, however ultimately I I chose to develop the final model using random forest due to its high degree of accuracy with a large number of variables where the interactions between the variance are not known.

```
fitControl <- trainControl(method = "none")
tgrid  <- expand.grid(mtry=c(6))
model  <- train(classe ~ ., data = train, method = "rf", trControl = fitControl, tuneGrid = tgrid)
model
```

```
## Random Forest
##
## 14718 samples
##    52 predictor
##     5 classes: 'A', 'B', 'C', 'D', 'E'
##
## No pre-processing
## Resampling: None
```

Performing Cross Validation

The model was then performed on the test set for cross validation. The accuracy on the test set was high at 0.995 and an expected out of sample error rate of 0.005 or 0.05%.

```r
predVal<- predict(model, test)
confusionMatrix(predVal, test$classe)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction    A    B    C    D    E
##          A 1391    2    0    0    0
##          B    3  946    6    0    0
##          C    0    1  848    7    0
##          D    0    0    1  797    2
##          E    1    0    0    0  899
##
## Overall Statistics
##
##                Accuracy : 0.995
##                  95% CI : (0.993, 0.997)
##     No Information Rate : 0.284
##     P-Value [Acc > NIR] : <2e-16
##
##                   Kappa : 0.994
##  Mcnemar's Test P-Value : NA
##
## Statistics by Class:
##
##                      Class: A Class: B Class: C Class: D Class: E
## Sensitivity             0.997    0.997    0.992    0.991    0.998
## Specificity             0.999    0.998    0.998    0.999    1.000
## Pos Pred Value          0.999    0.991    0.991    0.996    0.999
## Neg Pred Value          0.999    0.999    0.998    0.998    1.000
## Prevalence              0.284    0.194    0.174    0.164    0.184
## Detection Rate          0.284    0.193    0.173    0.163    0.183
## Detection Prevalence    0.284    0.195    0.175    0.163    0.184
## Balanced Accuracy       0.998    0.997    0.995    0.995    0.999
```

Run the final model on the test data set

```r
##Run model against Test Sample
pmlDataTest<- read.csv("pml-testing.csv", header = TRUE)
```

```
pmlDataTestClean <- pmlDataTest[,-c(1:7)]
predTest <- predict(model, pmlDataTestClean)
```

Results on Test Data

```
predTest
```

```
##  [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
```

```
table(predTest)
```

```
## predTest
## A B C D E
## 7 8 1 1 3
```