Prediction Assignment Writeup

Background

The datasets is from the Human Activity Recognition (HAR) research. The goal of this research was to investigate “how well” an activity was performed by the wearer. The data is based on wearable accelerometers participants wore that measure activity and motion.

For the prediction of how welll individuals performed the assigned exercise six young health participants were asked to perform one set of 10 repetitions of the Unilateral Dumbbell Biceps Curl in five different fashions: exactly according to the specification (Class A), throwing the elbows to the front (Class B), lifting the dumbbell only halfway (Class C), lowering the dumbbell only halfway (Class D) and throwing the hips to the front (Class E).

The goal of this assignment is to predict the activity the individual was performing using the data on the test data set.

Loading Data

Loading the training and test data sets. The test set has no exercise activity classification. I will use the train data to create a model to predict which exercises were performed.

```
train <- read.csv("training.csv", header=T, na.strings = c("", "NA"))
test <- read.csv("testing.csv", header=T, na.strings= c("", "NA"))
```

Data Cleaning: Removing NA’s and Unnecessary columns

I’ll create a threshold for NA column sum then remove columns that exceed that threshold.

```
NAS <- apply(train, 2, function(x) sum(is.na(x)))
NAS <- as.numeric(NAS) ;NA_test <- NAS <= 1000
train <- train[,NA_test]
test <- test[, NA_test]
```

Remove columns X to num_windows. They won’t help model with prediction.

```
train <- select(train, -(X:num_window))
test <- select(test , -(X:num_window))
```

Data Partitioning

Divide the training data into a training and test set. We will train our data on the train set and test on the training_test data set.

```
inTrain <- createDataPartition(y=train$classe, p = 0.70, list = FALSE)
train <- train[inTrain,]
training_test <- train[-inTrain,]
```

Random Forest Model Creation

Used 3 fold cross validation to help pick better paramters in the prediction function.

```
set.seed(1983)
fitCross <-trainControl(method="cv", number=3, allowParallel=T, verbose=T)
RF_Model <-train(classe~.,data=train, method="rf", trControl=fitCross, verbose=F)
```

```
## Loading required package: randomForest
```

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

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

```
## + Fold1: mtry= 2
## - Fold1: mtry= 2
## + Fold1: mtry=27
## - Fold1: mtry=27
## + Fold1: mtry=52
## - Fold1: mtry=52
## + Fold2: mtry= 2
## - Fold2: mtry= 2
## + Fold2: mtry=27
## - Fold2: mtry=27
## + Fold2: mtry=52
## - Fold2: mtry=52
## + Fold3: mtry= 2
## - Fold3: mtry= 2
## + Fold3: mtry=27
## - Fold3: mtry=27
## + Fold3: mtry=52
## - Fold3: mtry=52
## Aggregating results
## Selecting tuning parameters
## Fitting mtry = 27 on full training set
```

Results

These are the results: 99.3% accucurate predicting correct activity(classe) on the training_test data.

```
results <- predict(RF_Model, newdata=training_test)
confusionMatrix(results, training_test$classe)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction    A    B    C    D    E
##          A 1163    0    0    0    0
##          B    0  800    0    0    0
##          C    0    0  724    0    0
##          D    0    0    0  653    0
##          E    0    0    0    0  796
##
## Overall Statistics
##
##                Accuracy : 1
##                  95% CI : (0.9991, 1)
##     No Information Rate : 0.2812
##     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             0.2812   0.1934    0.175   0.1579   0.1925
## Detection Rate         0.2812   0.1934    0.175   0.1579   0.1925
## Detection Prevalence   0.2812   0.1934    0.175   0.1579   0.1925
## Balanced Accuracy      1.0000   1.0000    1.000   1.0000   1.0000
```

Prediction on original test data

Activity(classe) prediction for 20 observations on the original test data.

```
answers <- predict(RF_Model, newdata=test)
answers <- as.character(answers)
answers
```

```
##  [1] "B" "A" "B" "A" "A" "E" "D" "B" "A" "A" "B" "C" "B" "A" "E" "E" "A"
## [18] "B" "B" "B"
```