# Human Activity Recognition

*M.P.*

*Tuesday, August 11, 2015*

## Human Activity Recognition

### using an on-body sensing approach

#### Executive Summary

this study investigates three aspects that pertain to qualitative activity recognition: the problem of specifying correct execution, the automatic and robust detection of execution mistakes, and how to provide feedback on the quality of execution to the user. An on-body sensing approach was used. The results of this study was the development of an machine learning algorithm that will provde feedback on quality of execution. The model had an accuracy of 97%.

#### Scope

For this human activity recognition research, six participants (male, between 20-28 years of age) 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 sensors were located on the particpants upper arm (bicep), forearm, belt, and dumbell. This human activity recognition research predicts "which" activity was performed at a specific point in time.

The goal of this project is to predict the manner in which they did the exercise. This is the "classe" variable in the training set. You may use any of the other variables to predict with. You should create a report describing how you built your model, how you used cross validation, what you think the expected out of sample error is, and why you made the choices you did. You will also use your prediction model to predict 20 different test cases.

The dataset used in this study investigates three aspects that pertain to qualitative activity recognition: the problem of specifying correct execution, the automatic and robust detection of execution mistakes, and how to provide feedback on the quality of execution to the user. An on-body sensing approach was used.

Class A corresponds to the specified execution of the exercise, while the other 4 classes correspond to common mistakes. Participants were supervised by an experienced weight lifter to make sure the execution complied to the manner they were supposed to simulate. The exercises were performed by six participants with little weight lifting experience. All participants simulated the mistakes in a safe/controlled manner using a dumbbell that weighed 1.25kg.

#### Download the data

```
options(warn=-1)##supress warnings.
library(caret)
```

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

```r
library(randomForest)
```

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

```r
if (!file.exists("MachineLearning")){
        dir.create("MachineLearning")
}
setwd("~/MachineLearning")
pmltrain <- read.csv("./MachineLearning/pml-training.csv")
pmltest <- read.csv("./MachineLearning/pml-testing.csv")
```

**Explore and Clean the data** Used *summary* and *str* to explore the data sets. There are some columns that have a large number of N/A's, or division by 0 errors. There are other columns that do not seem to be necessary as predictors for the prediciton model (columns 1 through 7). I will delete these colomuns.

```r
dim(pmltrain)
```

```
## [1] 19622   160
```

```r
dim(pmltest)
```

```
## [1]  20 160
```

The training data set has 19622 rows and 160 columns before cleaning.
The testing data set has 20 rows and 160 columns before cleaning.

Both the training and the test data sets were cleaned to remove the columns that were not needed as described above.

```
## [1] 19622    53
```

```
## [1] 20 52
```

The training data set has 19622 rows and 53 columns after cleaning.
The testing data set has 20 rows and 52 columns after cleaning.

```r
train1cor<-data.frame(round(cor(train1[1:52]), 2))
hightrain1cor <- which(row(train1cor) < col(train1cor) & train1cor>.8, arr.ind=TRUE)[,"col"]
```

**Testing for correlations** Serval predictors were found to be highly correlated.

```r
set.seed(88)
train2= createDataPartition(y=train1$classe, p = .70, list = FALSE)
training = train1[ train2,]
testing = train1[-train2,]
```

**Splitting the training data set into a training (70%), and test data sets.**

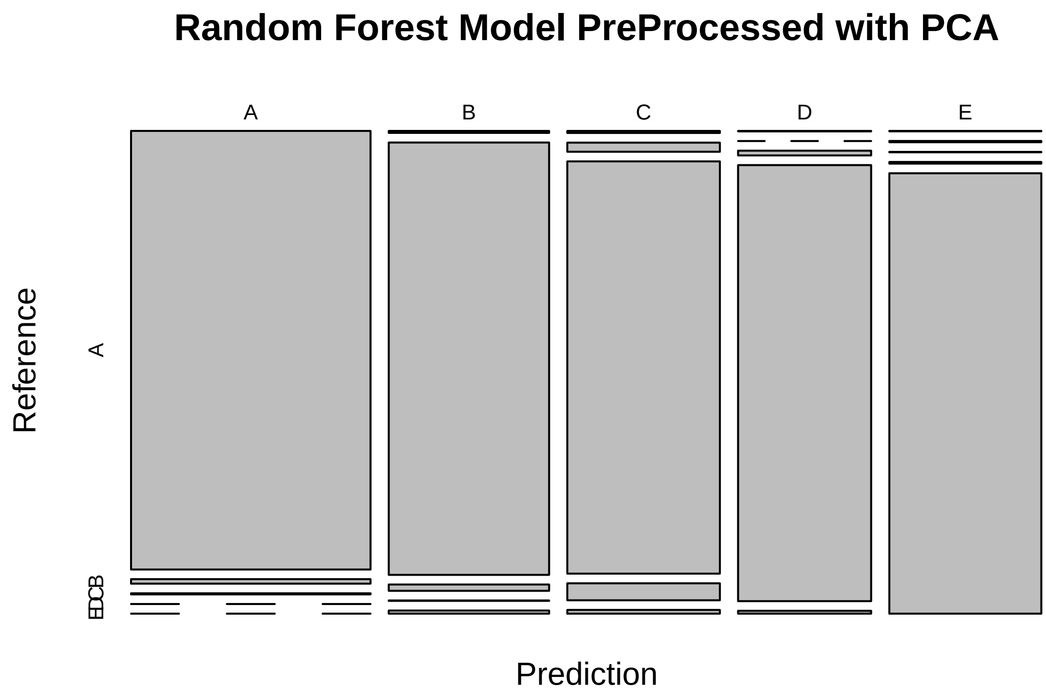
**Cross Validaton** Cross validation was determined by applying the fitmodelto the testing data. Additionally, Cross validation is accomplished automatically when using random forest, so independent cross validation is not required. Resampling of the data was done using backtesting.

```r
set.seed(889)
modelfit <- train(training$classe~.,data=training, preProcess="pca")##Preprocessing with PCA and traini
testing1 <- testing[,-53]##removeing classe from the data.
predicttesting<- predict(modelfit, testing1)
confusionmatrix <-confusionMatrix(predicttesting, testing$classe)
confusionmatrix
```

**Preprocess the training data set with PCA and fit a model using Random Forest.**

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction    A    B    C    D    E
##          A 1662   17    4    0    0
##          B    5 1097   16    1    8
##          C    5   22  995   41    9
##          D    1    0   10  918    6
##          E    1    3    1    4 1059
##
## Overall Statistics
##
##                Accuracy : 0.9738
##                  95% CI : (0.9694, 0.9778)
##     No Information Rate : 0.2845
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.9669
##  Mcnemar's Test P-Value : 3.08e-05
##
## Statistics by Class:
##
##                      Class: A Class: B Class: C Class: D Class: E
## Sensitivity            0.9928   0.9631   0.9698   0.9523   0.9787
## Specificity            0.9950   0.9937   0.9842   0.9965   0.9981
## Pos Pred Value         0.9875   0.9734   0.9282   0.9818   0.9916
## Neg Pred Value         0.9971   0.9912   0.9936   0.9907   0.9952
## Prevalence             0.2845   0.1935   0.1743   0.1638   0.1839
## Detection Rate         0.2824   0.1864   0.1691   0.1560   0.1799
## Detection Prevalence   0.2860   0.1915   0.1822   0.1589   0.1815
## Balanced Accuracy      0.9939   0.9784   0.9770   0.9744   0.9884
```

```r
plot(confusionmatrix[[2]], main="Random Forest Model PreProcessed with PCA")
```




The model had an Accuracy of 97%.

**Out-of-sample error** The out-of-sample error rate was .03. The out of sample error is determined from applying the prediction model to the training partitioned data set.

### 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.