# Course Project for Practical Machine Learning: The Write Up

*taroathirah*

*January 29, 2016*

Human activity recognition research has traditionally focused on discriminating between different activities. However, the ???how (well)??? investigation has only received little attention so far, even though it potentially provides useful information for a large variety of applications,such as sports training (http://groupware.les.inf.puc-rio.br/har (http://groupware.les.inf.puc-rio.br/har)).

For the prediction of how well 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).

This report aims to use machine learning algoritmhs to predict the class of exercise the individuals was performing by using meaurements available from devices such as Jawbone Up, Nike FuelBand, and Fitbit.

# Data Cleaning

The data for this project was obtained from http://groupware.les.inf.puc-rio.br/har (http://groupware.les.inf.puc-rio.br/har). Two data set are available; a training set and a test set for which 20 individuals data without any classification for the class of exercise are ready to be tested out.

```
#Data loading
setwd("~/Documents/R_programming_coursera")
pmlTrain<-read.csv("pml-training.csv", header=T, na.strings=c("NA", "#DIV/0!"))
pmlTest<-read.csv("pml-testing.csv", header=T, na.string=c("NA", "#DIV/0!"))
```

Training data was partitioned and preprocessed using the code described below. In brief, all variables with at least one ???NA??? were excluded from the analysis. Variables related to time and user information were excluded for a total of 51 variables and 19622 class measurements. Same variables were maintained in the test data set (Validation dataset) to be used for predicting the 20 test cases provided.

```
## NA exclusion for all available variables
noNApmlTrain<-pmlTrain[, apply(pmlTrain, 2, function(x) !any(is.na(x)))]
dim(noNApmlTrain)
```

```
## [1] 19622    60
```

```
## variables with user information, time and undefined
cleanpmlTrain<-noNApmlTrain[,-c(1:8)]
dim(cleanpmlTrain)
```

```
## [1] 19622    52
```

```
## 20 test cases provided clean info - Validation data set
cleanpmltest<-pmlTest[,names(cleanpmlTrain[,-52])]
dim(cleanpmltest)
```

```
## [1] 20 51
```

# Data Partitioning and Prediction Process

The cleaned downloaded data set was subset in order to generate a test set independent from the 20 cases provided set. Partitioning was performed to obtain a 75% training set and a 25% test set.

```
#data cleaning
library(caret)
inTrain<-createDataPartition(y=cleanpmlTrain$classe, p=0.75,list=F)
training<-cleanpmlTrain[inTrain,]
test<-cleanpmlTrain[-inTrain,]
#Training and test set dimensions
dim(training)
```

```
[1] 14718    52
```

```
dim(test)
```

```
[1] 4904   52
```

# Results and Conclusions

Random forest trees were generated for the training dataset using cross-validation. Then the generated algorithm was examnined under the partitioned training set to examine the accuracy and estimated error of prediction. By using 51 predictors for five classes using cross-validation at a 5-fold an accuracy of 99.2% with a 95% CI [0.989-0.994] was achieved accompanied by a Kappa value of 0.99.

```
library(caret)
set.seed(13333)
fitControl2<-trainControl(method="cv", number=5, allowParallel=T, verbose=T)
rffit<-train(classe~.,data=training, method="rf", trControl=fitControl2, verbos
e=F)
```

```
+ Fold1: mtry= 2
- Fold1: mtry= 2
+ Fold1: mtry=26
- Fold1: mtry=26
+ Fold1: mtry=51
- Fold1: mtry=51
+ Fold2: mtry= 2
- Fold2: mtry= 2
+ Fold2: mtry=26
- Fold2: mtry=26
+ Fold2: mtry=51
- Fold2: mtry=51
+ Fold3: mtry= 2
- Fold3: mtry= 2
+ Fold3: mtry=26
- Fold3: mtry=26
+ Fold3: mtry=51
- Fold3: mtry=51
+ Fold4: mtry= 2
- Fold4: mtry= 2
+ Fold4: mtry=26
- Fold4: mtry=26
+ Fold4: mtry=51
- Fold4: mtry=51
+ Fold5: mtry= 2
- Fold5: mtry= 2
+ Fold5: mtry=26
- Fold5: mtry=26
+ Fold5: mtry=51
- Fold5: mtry=51
Aggregating results
Selecting tuning parameters
Fitting mtry = 26 on full training set
```

```
predrf<-predict(rffit, newdata=test)
confusionMatrix(predrf, test$classe)
```

```
Confusion Matrix and Statistics

          Reference
Prediction    A    B    C    D    E
         A 1394    4    0    0    0
         B    1  943    8    0    0
         C    0    2  847    6    1
         D    0    0    0  798    3
         E    0    0    0    0  897

Overall Statistics

               Accuracy : 0.9949
                 95% CI : (0.9925, 0.9967)
    No Information Rate : 0.2845
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 0.9936
 Mcnemar's Test P-Value : NA

Statistics by Class:

                     Class: A Class: B Class: C Class: D Class: E
Sensitivity            0.9993   0.9937   0.9906   0.9925   0.9956
Specificity            0.9989   0.9977   0.9978   0.9993   1.0000
Pos Pred Value         0.9971   0.9905   0.9895   0.9963   1.0000
Neg Pred Value         0.9997   0.9985   0.9980   0.9985   0.9990
Prevalence             0.2845   0.1935   0.1743   0.1639   0.1837
Detection Rate         0.2843   0.1923   0.1727   0.1627   0.1829
Detection Prevalence   0.2851   0.1941   0.1746   0.1633   0.1829
Balanced Accuracy      0.9991   0.9957   0.9942   0.9959   0.9978
```

```
pred20<-predict(rffit, newdata=cleanpmltest)
pred20
```

```
[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
```

A boosting algorithm was also run to confirm and be able to compare predictions. Data is not shown but the boosting approach presented less accuracy **(96%)** (Data not shown). However, when the predictions for the 20 test cases were compared match was same for both ran algorimths.

```
fitControl2<-trainControl(method="cv", number=5, allowParallel=T, verbose=T)
gmbfit<-train(classe~.,data=training, method="gbm", trControl=fitControl2, verb
ose=F)
gmbfit$finalModel
class(gmbfit)
predgmb<-predict(gmbfit, newdata=test)
confusionMatrix(predgmb, test$classe)
predtrain<-predict(gmbfit, newdata=training)
confusionMatrix(predtrain, training$classe)
predtrain<-predict(gmbfit, newdata=training)
confusionMatrix(predtrain, training$classe)
```

Once, the predictions were obtained for the 20 test cases provided, the below shown script was used to obtain single text files to be uploaded to the courses web site to comply with the submission assigment. 20 out of 20 hits also confirmed the accuracy of the obtained models.

```
getwd()
pml_write_files = function(x){
  n = length(x)
  for(i in 1:n){
    filename = paste0("problem_id_",i,".txt")
    write.table(x[i],file=filename,quote=FALSE,row.names=FALSE,col.names=FALSE)
  }
}
pml_write_files(pred20)
```