# Practical Machine Learning - Course Project Report

## Introduction

The aim of this project is to predict activity quality (the manner in which the participants did the exercise) from activity monitors (accelerometers on the belt, forearm, arm, and dumbell of 6 participants). The data source for this project and its description are available on this website: http://groupware.les.inf.puc-rio.br/har. Specifically, the training data (pml-training.csv) can be downloaded from: https://d396qusza40orc.cloudfront.net/predmachlearn/pml-training.csv, while the test data (pml-testing.csv) can be downloaded from: https://d396qusza40orc.cloudfront.net/predmachlearn/pml-testing.csv.

## Load Libraries

```r
library(caret)
library(randomForest)
```

### Load Raw Input Data

The data source is manually downloaded and placed inside this working directory. In this step, the orignial traning and test csv files are loaded into memory.

```r
rawTrain <- read.csv("pml-training.csv")
rawTest <- read.csv("pml-testing.csv")
```

### Clean Data

All columns from the original data containing missing values are discarded.

```r
rawTrain <- rawTrain[, colSums(is.na(rawTrain)) == 0]
rawTest <- rawTest[, colSums(is.na(rawTest)) == 0]
```

Columns that do not contribute much to the accelerometer measurements are also removed.

```r
classe <- rawTrain$classe
discardedTrain <- grepl("^X|timestamp|window", names(rawTrain))
rawTrain <- rawTrain[, !discardedTrain]
cleanTrain <- rawTrain[, sapply(rawTrain, is.numeric)]
cleanTrain$classe <- classe
discardedTest <- grepl("^X|timestamp|window", names(rawTest))
rawTest <- rawTest[, !discardedTest]
cleanTest <- rawTest[, sapply(rawTest, is.numeric)]
```

### Slice Data

In this step, a validation data set for future cross validation is created. Specifically, the clean training set obtained from the previous step is now divided into a pure training data set (70%) and a validation data set (30%).

```r
set.seed(20150523)
partitions <- createDataPartition(cleanTrain$classe, p=0.70, list=F)
trainData <- cleanTrain[partitions, ]
validationData <- cleanTrain[-partitions, ]
```

### Build Prediction Model using Random Forest

As random forest algrithm automatically chooses important variables and is robust to correlated covariates and outliers, it is used to fit a model for predicting activity quality from activity monitors. The algorihthm is configured to use 5-fold cross validation.

```r
control <- trainControl(method="cv", 5)
model <- train(classe ~ ., data=trainData, method="rf", trControl=control, ntree=250)
model
```

```
## Random Forest
##
## 13737 samples
##    52 predictor
##     5 classes: 'A', 'B', 'C', 'D', 'E'
##
## No pre-processing
## Resampling: Cross-Validated (5 fold)
##
## Summary of sample sizes: 10989, 10989, 10989, 10990, 10991
##
## Resampling results across tuning parameters:
##
##   mtry  Accuracy  Kappa   Accuracy SD  Kappa SD
##    2    0.9905    0.9879  0.001848     0.002340
##   27    0.9894    0.9866  0.002046     0.002590
##   52    0.9856    0.9818  0.003365     0.004262
##
## Accuracy was used to select the optimal model using  the largest value.
## The final value used for the model was mtry = 2.
```

The estimated accuracy and out-of-sample error of the model are computed based on cross-validation.

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

```r
accuracy <- postResample(predict, validationData$classe)
accuracy
```

```
## Accuracy    Kappa
##   0.9927   0.9908
```

```r
outOfSampleError <- 1 - as.numeric(confusionMatrix(validationData$classe, predict)$overall[1])
outOfSampleError
```

```
## [1] 0.007307
```

Overall, the estimated accuracy of the model is 99.27%, while the estimated out-of-sample error is 0.73%.

## Apply Prediction Model and Write Results to Files

Finally, the prediction model built previously is applied on the test data set. The predicted results are persisted into files.

```r
result <- predict(model, cleanTest[, -length(names(cleanTest))])
result
```

```
##  [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
```

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