```
---
title: "Predict activity quality from activity monitors"
author: "Onur Akpolat"
date: "24. January 2015"
output:
  html_document:
    keep_md: yes
---

##Synopsis

Using devices such as Jawbone Up, Nike FuelBand, and Fitbit it is now possible to collect a large amount of data about personal activity relatively inexpensiv

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.

## Data description

The outcome variable is `classe`, a factor variable with 5 levels. For this data set, participants were asked to perform one set of 10 repetitions of the Unil

- 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)
- throwing the hips to the front (Class E)

## Initial configuration

The initial configuration consists of loading some required packages and initializing some variables.

```{r configuration, echo=TRUE, results='hide'}
#Data variables
training.file   <- './data/pml-training.csv'
test.cases.file <- './data/pml-testing.csv'
training.url    <- 'http://d396qusza40orc.cloudfront.net/predmachlearn/pml-training.csv'
test.cases.url  <- 'http://d396qusza40orc.cloudfront.net/predmachlearn/pml-testing.csv'
#Directories
if (!file.exists("data")){
  dir.create("data")
}
if (!file.exists("data/submission")){
  dir.create("data/submission")
}
#R-Packages
IscaretInstalled <- require("caret")
if(!IscaretInstalled){
    install.packages("caret")
    library("caret")
    }
IsrandomForestInstalled <- require("randomForest")
if(!IsrandomForestInstalled){
    install.packages("randomForest")
    library("randomForest")
    }
IsRpartInstalled <- require("rpart")
if(!IsRpartInstalled){
    install.packages("rpart")
    library("rpart")
    }
IsRpartPlotInstalled <- require("rpart.plot")
if(!IsRpartPlotInstalled){
    install.packages("rpart.plot")
    library("rpart.plot")
    }
# Set seed for reproducability
set.seed(9999)
```

Data processing
In this section the data is downloaded and processed. Some basic transformations and cleanup will be performed, so that `NA` values are omitted. Irrelevant co

The `pml-training.csv` data is used to devise training and testing sets.
The `pml-test.csv` data is used to predict and answer the 20 questions based on the trained model.
```

```
```{r dataprocessing, echo=TRUE, results='hide'}
# Download data
download.file(training.url, training.file)
download.file(test.cases.url,test.cases.file )
# Clean data
training   <-read.csv(training.file, na.strings=c("NA","#DIV/0!", ""))
testing <-read.csv(test.cases.file , na.strings=c("NA", "#DIV/0!", ""))
training<-training[,colSums(is.na(training)) == 0]
testing <-testing[,colSums(is.na(testing)) == 0]
# Subset data
training   <-training[,-c(1:7)]
testing <-testing[,-c(1:7)]
```

Cross-validation
In this section cross-validation will be performed by splitting the training data in training (75%) and testing (25%) data.

```{r datasplitting, echo=TRUE, results='hide'}
subSamples <- createDataPartition(y=training$classe, p=0.75, list=FALSE)
subTraining <- training[subSamples, ]
subTesting <- training[-subSamples, ]
```

Expected out-of-sample error
The expected out-of-sample error will correspond to the quantity: 1-accuracy in the cross-validation data. Accuracy is the proportion of correct classified ob

Exploratory analysis
The variable `classe` contains 5 levels. The plot of the outcome variable shows the frequency of each levels in the subTraining data.

```{r exploranalysis, echo=TRUE}
plot(subTraining$classe, col="orange", main="Levels of the variable classe", xlab="classe levels", ylab="Frequency")
```

The plot above shows that Level A is the most frequent classe. D appears to be the least frequent one.

Prediction models
In this section a decision tree and random forest will be applied to the data.

Decision tree
```{r decisiontree, echo=TRUE}
# Fit model
modFitDT <- rpart(classe ~ ., data=subTraining, method="class")
# Perform prediction
predictDT <- predict(modFitDT, subTesting, type = "class")
# Plot result
rpart.plot(modFitDT, main="Classification Tree", extra=102, under=TRUE, faclen=0)
```

Following confusion matrix shows the errors of the prediction algorithm.

```{r decisiontreecm, echo=TRUE}
confusionMatrix(predictDT, subTesting$classe)
```

Random forest
```{r randomforest, echo=TRUE}
# Fit model
modFitRF <- randomForest(classe ~ ., data=subTraining, method="class")
# Perform prediction
predictRF <- predict(modFitRF, subTesting, type = "class")
```

Following confusion matrix shows the errors of the prediction algorithm.

```{r randomforestcm, echo=TRUE}
confusionMatrix(predictRF, subTesting$classe)
```

Conclusion

Result

The confusion matrices show, that the Random Forest algorithm performens better than decision trees. The accuracy for the Random Forest model was 0.995 (95% C
```

```
### Expected out-of-sample error
The expected out-of-sample error is estimated at 0.005, or 0.5%. The expected out-of-sample error is calculated as 1 - accuracy for predictions made against t

## Submission
In this section the files for the project submission are generated using the random forest algorithm on the testing data.

```{r submission, echo=TRUE}
# Perform prediction
predictSubmission <- predict(modFitRF, testing, type="class")
predictSubmission
# Write files for submission
pml_write_files = function(x){
  n = length(x)
  for(i in 1:n){
    filename = paste0("./data/submission/problem_id_",i,".txt")
    write.table(x[i],file=filename,quote=FALSE,row.names=FALSE,col.names=FALSE)
  }
}
pml_write_files(predictSubmission)
```
```