Course Project

Rick van Zijl

20 december 2017

## Course Project

This document describes the course project

## Background

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 inexpensively. These type of devices are part of the quantified self movement - a group of enthusiasts who take measurements about themselves regularly to improve their health, to find patterns in their behavior, or because they are tech geeks. One thing that people regularly do is quantify how much of a particular activity they do, but they rarely quantify how well they do it. In this project, your goal will be to use data from accelerometers on the belt, forearm, arm, and dumbell of 6 participants. They were asked to perform barbell lifts correctly and incorrectly in 5 different ways. More information is available from the website here: <http://groupware.les.inf.puc-rio.br/har> (see the section on the Weight Lifting Exercise Dataset)

setwd("C:/Users/rzij/Documents/Data Science Course/Course Project Machine Learning")  
  
library(caret)

## Warning: package 'caret' was built under R version 3.4.3

## Loading required package: lattice

## Loading required package: ggplot2

## Warning: package 'ggplot2' was built under R version 3.4.3

rm(list = ls())  
training <- read.csv("pml-training.csv")  
testing <- read.csv("pml-testing.csv")  
  
dim(training)

## [1] 19622 160

# Clear up for meta data, nearZeroVariables and Columns that contain mostly NA values.  
training <- training[,-c(1:7)]  
training <- training[,-c(nearZeroVar(training))]  
training <- training[, apply(training, 2, function(x) !any(is.na(x)))]  
  
# Repeat for the test data  
testing <- testing[,-c(1:7)]  
testing <- testing[,-c(nearZeroVar(testing))]  
testing <- testing[, apply(testing, 2, function(x) !any(is.na(x)))]  
  
# Split training set  
split <- createDataPartition(training$classe, p = 0.7, list = FALSE)  
training1 <- training[split,]  
training2 <- training[-split,]  
  
# Fit three predict models based on training1, test on training2, compare accuracy. The creation of the models   
# is now commented because they take long to run. Instead i load the environment  
load("ML algorithms.RData")  
trainctrl <- trainControl(verboseIter = TRUE)  
  
# gbmModel <- train(classe ~ ., data = training1, method = "gbm", trControl = trainctrl)  
gbmPredict <- confusionMatrix(predict(gbmModel, training2), training2$classe)  
gbmPredict$overall[1]

## Accuracy   
## 0.9566695

# rfModel <- train(classe ~ ., data = training1, method = "rf", trControl = trainctrl)  
rfPredict <- confusionMatrix(predict(rfModel, training2), training2$classe)  
rfPredict$overall[1]

## Accuracy   
## 0.9920136

# rpartModel <- train(classe ~ ., data = training1, method = "rpart", trControl = trainctrl)  
rpartPredict <- confusionMatrix(predict(rpartModel, training2), training2$classe)  
rpartPredict$overall[1]

## Accuracy   
## 0.4876805

# Apply the random forest model on the test set  
rfPredictTest <- predict(rfModel, testing)  
submission <- data.frame(id = testing$problem\_id, prediction = rfPredictTest)  
submission

## id prediction  
## 1 1 B  
## 2 2 A  
## 3 3 B  
## 4 4 A  
## 5 5 A  
## 6 6 E  
## 7 7 D  
## 8 8 B  
## 9 9 A  
## 10 10 A  
## 11 11 B  
## 12 12 C  
## 13 13 B  
## 14 14 A  
## 15 15 E  
## 16 16 E  
## 17 17 A  
## 18 18 B  
## 19 19 B  
## 20 20 B