Course Project for the Practical Machine Learning Course

R script for the Practical Machine Learning 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.

Load the necessary libraries

```
library(caret)
library(ggplot2)
library(dplyr)
library(randomForest)
library(e1071)
library(rattle)
```

- The training data set has 19622 observations and 160 variables.
- The validation data set has 20 observations and 160 variables.

Before fitting the models it's necessary to clean the data.

```
## Remove the first seven columns since are the ones that don't work for the model
training <- training[,8:160]

## Remove all columns that have at least one empty row
training <- training[,colSums(is.na(training))==0]

## Remove all variables which its variance is small

ZeroVar <- nearZeroVar(training)
training <- training[,-ZeroVar]

## Split the data set into training & testing

inTrain <- createDataPartition(y=training$classe,p=0.7,list=FALSE)
trainData <- training[inTrain,]
testData <- training[-inTrain,]
```

Match the remaining variables to the validation dataset.

```
RemainingDF <- function(namesCol, newDF){

    x <- c()
    for(i in 1:159){
        for(j in 1:length(namesCol)){
            LogTest <- colnames(newDF)[i] == namesCol[j]
            if(LogTest == TRUE){
                x <- c(x,i)
                break
```

```
            }
        }
    }
    return(newDF[,c(x,160)])
}

RemCols <- colnames(training)
finalValidation <- RemainingDF(RemCols, validation)

## Due to the naturality of the problem I'll start with a random forest model

fit1 <- randomForest(classe~.,data=trainData)

confusionMatrix(predict(fit1,testData),testData$classe)$overall
```

```
##       Accuracy          Kappa  AccuracyLower  AccuracyUpper   AccuracyNull
##      0.9945624      0.9931217      0.9923324      0.9962778      0.2844520
## AccuracyPValue  McnemarPValue
##      0.0000000            NaN
```

Overall accuracy for the train set was 0.99. There doesn´t seem to be necessary to make another model.