Practical Machine Learning Course Project - Prediction Assignment

Mishell Guerra

21 august, 2020

Introduction

The goal of your project is to predict the manner in which they did the exercise. This is the "classe" variable in the training set. You may use any of the other variables to predict with. You should create a report describing how you built your model, how you used cross validation, what you think the expected out of sample error is, and why you made the choices you did. You will also use your prediction model to predict 20 different test cases.

Model built

The expacted outcome variable is classe, a 5 level of factor variable. In this dataset, participants were asked to perform one set of 10 repetitions of the Unilateral Dumbbell Biceps Curl in 5 different fashion: Class A - exactly according to the specification, Class B - throwing the elbows to the front, Class c- lifting the dumbbell only halfway, Class D - lowering the dumbbell only halfway, Class E - throwing the hips to the front.

Class A correspons to the specified execution of the exercise, while the other 4 classes correpond to common mistakes. Decision tree will be used to create the model. After the model have been developed. Cross-validation will be performed. Two set of data will be created, original training data set (75%) and subtesting data set (25%).

Load library

```
## Loading required package: lattice

## Loading required package: ggplot2

## randomForest 4.6-14

## Type rfNews() to see new features/changes/bug fixes.

##
## Attaching package: 'randomForest'

## The following object is masked from 'package:ggplot2':
##
##     margin
```

Overview

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 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.

The objective of this report is to demonstrate the process employed to arrive at a prediction algorithm, which aims to classify the manner in which the participants employed certain exercises. The data comes from accelerometers attached on the belt, forearm and dumbells.

Load Dataset

```
trainingData <- read.csv("training.csv", na.strings = c("NA", "#DIV/0!", ""))
testingData <- read.csv("testing.csv", na.strings = c("NA", "#DIV/0!", ""))
trainingData <- trainingData[, colSums(is.na(trainingData)) == 0]
testingData <- testingData[, colSums(is.na(testingData)) == 0]
# Delete variables that are not related
trainingData <- trainingData[, -c(1:7)]
testingData <- testingData[, -c(1:7)]
# partioning the training set into two different dataset
traningPartitionData <- createDataPartition(trainingData$classe,  p = 0.7, list = F)
trainingDataSet <- trainingData[traningPartitionData, ]
testingDataSet <- trainingData[-traningPartitionData, ]
dim(trainingData); dim(testingDataSet)
```

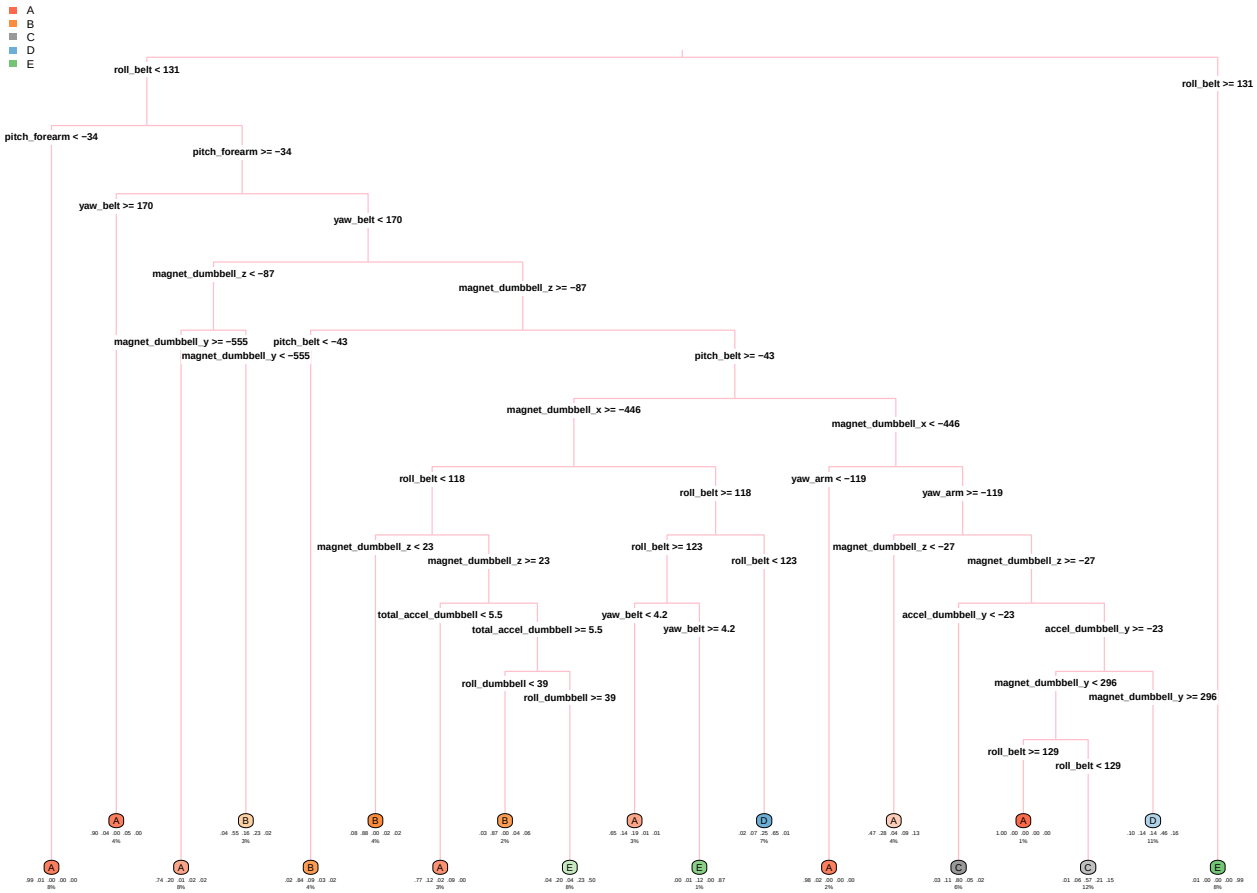
```
## [1] 19622    53
```

```
## [1] 5885   53
```

Prediction model 1 - decision tree

```
decisionTreeModel <- rpart(classe ~ ., data = trainingDataSet, method = "class")
decisionTreePrediction <- predict(decisionTreeModel, testingDataSet, type = "class")
rpart.plot(decisionTreeModel, main = "Decision Tree",cex=0.6, under = TRUE, faclen = 0, compress=TRUE,
```

Decision Tree




```
# Using confusion matrix to test results
confusionMatrix(factor(decisionTreePrediction), factor(testingDataSet$classe))
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction    A    B    C    D    E
##          A 1531  226   56   59   46
##          B   27  606   47   55   29
##          C   22   84  740  193  126
##          D   72  135  158  549   99
##          E   22   88   25  108  782
##
## Overall Statistics
##
##                Accuracy : 0.715
##                  95% CI : (0.7033, 0.7265)
##     No Information Rate : 0.2845
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.6381
##
##  Mcnemar's Test P-Value : < 2.2e-16
##
## Statistics by Class:
```

```
##
##                      Class: A Class: B Class: C Class: D Class: E
## Sensitivity            0.9146   0.5320   0.7212  0.56950   0.7227
## Specificity            0.9081   0.9667   0.9125  0.90571   0.9494
## Pos Pred Value         0.7982   0.7932   0.6352  0.54195   0.7629
## Neg Pred Value         0.9640   0.8959   0.9394  0.91482   0.9383
## Prevalence             0.2845   0.1935   0.1743  0.16381   0.1839
## Detection Rate         0.2602   0.1030   0.1257  0.09329   0.1329
## Detection Prevalence   0.3259   0.1298   0.1980  0.17213   0.1742
## Balanced Accuracy      0.9113   0.7494   0.8169  0.73761   0.8361
```

Prediction model 2 - random forest

```
randomForestModel <- randomForest(factor(classe) ~. , data = trainingDataSet, method = "class")
randomForestPrediction <- predict(randomForestModel, testingDataSet, type = "class")
confusionMatrix(factor(randomForestPrediction), factor(testingDataSet$classe))
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction    A    B    C    D    E
##          A 1674    6    0    0    0
##          B    0 1131    6    0    0
##          C    0    2 1019   12    0
##          D    0    0    1  949    2
##          E    0    0    0    3 1080
##
## Overall Statistics
##
##                Accuracy : 0.9946
##                  95% CI : (0.9923, 0.9963)
##     No Information Rate : 0.2845
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.9931
##
##  Mcnemar's Test P-Value : NA
##
## Statistics by Class:
##
##                      Class: A Class: B Class: C Class: D Class: E
## Sensitivity            1.0000   0.9930   0.9932   0.9844   0.9982
## Specificity            0.9986   0.9987   0.9971   0.9994   0.9994
## Pos Pred Value         0.9964   0.9947   0.9864   0.9968   0.9972
## Neg Pred Value         1.0000   0.9983   0.9986   0.9970   0.9996
## Prevalence             0.2845   0.1935   0.1743   0.1638   0.1839
## Detection Rate         0.2845   0.1922   0.1732   0.1613   0.1835
## Detection Prevalence   0.2855   0.1932   0.1755   0.1618   0.1840
## Balanced Accuracy      0.9993   0.9959   0.9951   0.9919   0.9988
```

Prediction model 2 - random forest

From the result, it show Random Forest accuracy is higher than Decision tree which is $0.9915 > 0.6644$. Therefore, we will use random forest to answer the assignment.

```r
FinalPrediction <- predict(randomForestModel, testingDataSet, type = "class")
```