Practical Machine Learning-Prediction Assignment

AM

12/12/2018

## 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://web.archive.org/web/20161224072740/http:/groupware.les.inf.puc-rio.br/har> (see the section on the Weight Lifting Exercise Dataset).

## Data

The training data for this project are available here: <https://d396qusza40orc.cloudfront.net/predmachlearn/pml-training.csv> The test data are available here: <https://d396qusza40orc.cloudfront.net/predmachlearn/pml-testing.csv>

## Installing Packages

#install.packages("caret")  
#install.packages("randomForest")  
#install.packages("rpart")  
library(lattice)  
library(ggplot2)  
library(caret)  
library(randomForest)

## 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

library(rpart)  
library(rpart.plot)  
set.seed(1234)

## Loading data and cleaning

# After saving both data sets into my working directory  
# Some missing values are coded as string "#DIV/0!" or "" or "NA" - these will be changed to NA.  
# We notice that both data sets contain columns with all missing values - these will be deleted.   
  
# Loading the training data set into my R session replacing all missing with "NA"  
trainingset <- read.csv("pml-training.csv", na.strings=c("NA","#DIV/0!", ""))  
  
# Loading the testing data set   
testingset <- read.csv("pml-testing.csv", na.strings=c("NA","#DIV/0!", ""))  
  
# Check dimensions for number of variables and number of observations  
dim(trainingset)

## [1] 19622 160

dim(testingset)

## [1] 20 160

# Delete columns with all missing values  
trainingset<-trainingset[,colSums(is.na(trainingset)) == 0]  
testingset <-testingset[,colSums(is.na(testingset)) == 0]  
  
# Some variables are irrelevant to our current project: user\_name, raw\_timestamp\_part\_1, raw\_timestamp\_part\_,2 cvtd\_timestamp, new\_window, and num\_window (columns 1 to 7). We can delete these variables.  
trainingset<-trainingset[,-c(1:7)]  
testingset<-testingset[,-c(1:7)]  
# and have a look at our new datasets:  
dim(trainingset)

## [1] 19622 53

dim(testingset)

## [1] 20 53

head(trainingset)

## roll\_belt pitch\_belt yaw\_belt total\_accel\_belt gyros\_belt\_x gyros\_belt\_y  
## 1 1.41 8.07 -94.4 3 0.00 0.00  
## 2 1.41 8.07 -94.4 3 0.02 0.00  
## 3 1.42 8.07 -94.4 3 0.00 0.00  
## 4 1.48 8.05 -94.4 3 0.02 0.00  
## 5 1.48 8.07 -94.4 3 0.02 0.02  
## 6 1.45 8.06 -94.4 3 0.02 0.00  
## gyros\_belt\_z accel\_belt\_x accel\_belt\_y accel\_belt\_z magnet\_belt\_x  
## 1 -0.02 -21 4 22 -3  
## 2 -0.02 -22 4 22 -7  
## 3 -0.02 -20 5 23 -2  
## 4 -0.03 -22 3 21 -6  
## 5 -0.02 -21 2 24 -6  
## 6 -0.02 -21 4 21 0  
## magnet\_belt\_y magnet\_belt\_z roll\_arm pitch\_arm yaw\_arm total\_accel\_arm  
## 1 599 -313 -128 22.5 -161 34  
## 2 608 -311 -128 22.5 -161 34  
## 3 600 -305 -128 22.5 -161 34  
## 4 604 -310 -128 22.1 -161 34  
## 5 600 -302 -128 22.1 -161 34  
## 6 603 -312 -128 22.0 -161 34  
## gyros\_arm\_x gyros\_arm\_y gyros\_arm\_z accel\_arm\_x accel\_arm\_y accel\_arm\_z  
## 1 0.00 0.00 -0.02 -288 109 -123  
## 2 0.02 -0.02 -0.02 -290 110 -125  
## 3 0.02 -0.02 -0.02 -289 110 -126  
## 4 0.02 -0.03 0.02 -289 111 -123  
## 5 0.00 -0.03 0.00 -289 111 -123  
## 6 0.02 -0.03 0.00 -289 111 -122  
## magnet\_arm\_x magnet\_arm\_y magnet\_arm\_z roll\_dumbbell pitch\_dumbbell  
## 1 -368 337 516 13.05217 -70.49400  
## 2 -369 337 513 13.13074 -70.63751  
## 3 -368 344 513 12.85075 -70.27812  
## 4 -372 344 512 13.43120 -70.39379  
## 5 -374 337 506 13.37872 -70.42856  
## 6 -369 342 513 13.38246 -70.81759  
## yaw\_dumbbell total\_accel\_dumbbell gyros\_dumbbell\_x gyros\_dumbbell\_y  
## 1 -84.87394 37 0 -0.02  
## 2 -84.71065 37 0 -0.02  
## 3 -85.14078 37 0 -0.02  
## 4 -84.87363 37 0 -0.02  
## 5 -84.85306 37 0 -0.02  
## 6 -84.46500 37 0 -0.02  
## gyros\_dumbbell\_z accel\_dumbbell\_x accel\_dumbbell\_y accel\_dumbbell\_z  
## 1 0.00 -234 47 -271  
## 2 0.00 -233 47 -269  
## 3 0.00 -232 46 -270  
## 4 -0.02 -232 48 -269  
## 5 0.00 -233 48 -270  
## 6 0.00 -234 48 -269  
## magnet\_dumbbell\_x magnet\_dumbbell\_y magnet\_dumbbell\_z roll\_forearm  
## 1 -559 293 -65 28.4  
## 2 -555 296 -64 28.3  
## 3 -561 298 -63 28.3  
## 4 -552 303 -60 28.1  
## 5 -554 292 -68 28.0  
## 6 -558 294 -66 27.9  
## pitch\_forearm yaw\_forearm total\_accel\_forearm gyros\_forearm\_x  
## 1 -63.9 -153 36 0.03  
## 2 -63.9 -153 36 0.02  
## 3 -63.9 -152 36 0.03  
## 4 -63.9 -152 36 0.02  
## 5 -63.9 -152 36 0.02  
## 6 -63.9 -152 36 0.02  
## gyros\_forearm\_y gyros\_forearm\_z accel\_forearm\_x accel\_forearm\_y  
## 1 0.00 -0.02 192 203  
## 2 0.00 -0.02 192 203  
## 3 -0.02 0.00 196 204  
## 4 -0.02 0.00 189 206  
## 5 0.00 -0.02 189 206  
## 6 -0.02 -0.03 193 203  
## accel\_forearm\_z magnet\_forearm\_x magnet\_forearm\_y magnet\_forearm\_z  
## 1 -215 -17 654 476  
## 2 -216 -18 661 473  
## 3 -213 -18 658 469  
## 4 -214 -16 658 469  
## 5 -214 -17 655 473  
## 6 -215 -9 660 478  
## classe  
## 1 A  
## 2 A  
## 3 A  
## 4 A  
## 5 A  
## 6 A

head(testingset)

## roll\_belt pitch\_belt yaw\_belt total\_accel\_belt gyros\_belt\_x gyros\_belt\_y  
## 1 123.00 27.00 -4.75 20 -0.50 -0.02  
## 2 1.02 4.87 -88.90 4 -0.06 -0.02  
## 3 0.87 1.82 -88.50 5 0.05 0.02  
## 4 125.00 -41.60 162.00 17 0.11 0.11  
## 5 1.35 3.33 -88.60 3 0.03 0.02  
## 6 -5.92 1.59 -87.70 4 0.10 0.05  
## gyros\_belt\_z accel\_belt\_x accel\_belt\_y accel\_belt\_z magnet\_belt\_x  
## 1 -0.46 -38 69 -179 -13  
## 2 -0.07 -13 11 39 43  
## 3 0.03 1 -1 49 29  
## 4 -0.16 46 45 -156 169  
## 5 0.00 -8 4 27 33  
## 6 -0.13 -11 -16 38 31  
## magnet\_belt\_y magnet\_belt\_z roll\_arm pitch\_arm yaw\_arm total\_accel\_arm  
## 1 581 -382 40.7 -27.80 178 10  
## 2 636 -309 0.0 0.00 0 38  
## 3 631 -312 0.0 0.00 0 44  
## 4 608 -304 -109.0 55.00 -142 25  
## 5 566 -418 76.1 2.76 102 29  
## 6 638 -291 0.0 0.00 0 14  
## gyros\_arm\_x gyros\_arm\_y gyros\_arm\_z accel\_arm\_x accel\_arm\_y accel\_arm\_z  
## 1 -1.65 0.48 -0.18 16 38 93  
## 2 -1.17 0.85 -0.43 -290 215 -90  
## 3 2.10 -1.36 1.13 -341 245 -87  
## 4 0.22 -0.51 0.92 -238 -57 6  
## 5 -1.96 0.79 -0.54 -197 200 -30  
## 6 0.02 0.05 -0.07 -26 130 -19  
## magnet\_arm\_x magnet\_arm\_y magnet\_arm\_z roll\_dumbbell pitch\_dumbbell  
## 1 -326 385 481 -17.73748 24.96085  
## 2 -325 447 434 54.47761 -53.69758  
## 3 -264 474 413 57.07031 -51.37303  
## 4 -173 257 633 43.10927 -30.04885  
## 5 -170 275 617 -101.38396 -53.43952  
## 6 396 176 516 62.18750 -50.55595  
## yaw\_dumbbell total\_accel\_dumbbell gyros\_dumbbell\_x gyros\_dumbbell\_y  
## 1 126.23596 9 0.64 0.06  
## 2 -75.51480 31 0.34 0.05  
## 3 -75.20287 29 0.39 0.14  
## 4 -103.32003 18 0.10 -0.02  
## 5 -14.19542 4 0.29 -0.47  
## 6 -71.12063 29 -0.59 0.80  
## gyros\_dumbbell\_z accel\_dumbbell\_x accel\_dumbbell\_y accel\_dumbbell\_z  
## 1 -0.61 21 -15 81  
## 2 -0.71 -153 155 -205  
## 3 -0.34 -141 155 -196  
## 4 0.05 -51 72 -148  
## 5 -0.46 -18 -30 -5  
## 6 1.10 -138 166 -186  
## magnet\_dumbbell\_x magnet\_dumbbell\_y magnet\_dumbbell\_z roll\_forearm  
## 1 523 -528 -56 141  
## 2 -502 388 -36 109  
## 3 -506 349 41 131  
## 4 -576 238 53 0  
## 5 -424 252 312 -176  
## 6 -543 262 96 150  
## pitch\_forearm yaw\_forearm total\_accel\_forearm gyros\_forearm\_x  
## 1 49.30 156.0 33 0.74  
## 2 -17.60 106.0 39 1.12  
## 3 -32.60 93.0 34 0.18  
## 4 0.00 0.0 43 1.38  
## 5 -2.16 -47.9 24 -0.75  
## 6 1.46 89.7 43 -0.88  
## gyros\_forearm\_y gyros\_forearm\_z accel\_forearm\_x accel\_forearm\_y  
## 1 -3.34 -0.59 -110 267  
## 2 -2.78 -0.18 212 297  
## 3 -0.79 0.28 154 271  
## 4 0.69 1.80 -92 406  
## 5 3.10 0.80 131 -93  
## 6 4.26 1.35 230 322  
## accel\_forearm\_z magnet\_forearm\_x magnet\_forearm\_y magnet\_forearm\_z  
## 1 -149 -714 419 617  
## 2 -118 -237 791 873  
## 3 -129 -51 698 783  
## 4 -39 -233 783 521  
## 5 172 375 -787 91  
## 6 -144 -300 800 884  
## problem\_id  
## 1 1  
## 2 2  
## 3 3  
## 4 4  
## 5 5  
## 6 6

## Partitioning the training data set to allow cross-validation

The training data set contains 53 variables and 19622 obs. The testing data set contains 53 variables and 20 obs. In order to perform cross-validation, the training data set is partionned into 2 sets: subTraining (75%) and subTest (25%). This will be performed using random subsampling without replacement.

subsamples <- createDataPartition(y=trainingset$classe, p=0.75, list=FALSE)  
subTraining <- trainingset[subsamples, ]   
subTesting <- trainingset[-subsamples, ]  
dim(subTraining)

## [1] 14718 53

dim(subTesting)

## [1] 4904 53

head(subTraining)

## roll\_belt pitch\_belt yaw\_belt total\_accel\_belt gyros\_belt\_x gyros\_belt\_y  
## 2 1.41 8.07 -94.4 3 0.02 0.00  
## 3 1.42 8.07 -94.4 3 0.00 0.00  
## 4 1.48 8.05 -94.4 3 0.02 0.00  
## 5 1.48 8.07 -94.4 3 0.02 0.02  
## 6 1.45 8.06 -94.4 3 0.02 0.00  
## 7 1.42 8.09 -94.4 3 0.02 0.00  
## gyros\_belt\_z accel\_belt\_x accel\_belt\_y accel\_belt\_z magnet\_belt\_x  
## 2 -0.02 -22 4 22 -7  
## 3 -0.02 -20 5 23 -2  
## 4 -0.03 -22 3 21 -6  
## 5 -0.02 -21 2 24 -6  
## 6 -0.02 -21 4 21 0  
## 7 -0.02 -22 3 21 -4  
## magnet\_belt\_y magnet\_belt\_z roll\_arm pitch\_arm yaw\_arm total\_accel\_arm  
## 2 608 -311 -128 22.5 -161 34  
## 3 600 -305 -128 22.5 -161 34  
## 4 604 -310 -128 22.1 -161 34  
## 5 600 -302 -128 22.1 -161 34  
## 6 603 -312 -128 22.0 -161 34  
## 7 599 -311 -128 21.9 -161 34  
## gyros\_arm\_x gyros\_arm\_y gyros\_arm\_z accel\_arm\_x accel\_arm\_y accel\_arm\_z  
## 2 0.02 -0.02 -0.02 -290 110 -125  
## 3 0.02 -0.02 -0.02 -289 110 -126  
## 4 0.02 -0.03 0.02 -289 111 -123  
## 5 0.00 -0.03 0.00 -289 111 -123  
## 6 0.02 -0.03 0.00 -289 111 -122  
## 7 0.00 -0.03 0.00 -289 111 -125  
## magnet\_arm\_x magnet\_arm\_y magnet\_arm\_z roll\_dumbbell pitch\_dumbbell  
## 2 -369 337 513 13.13074 -70.63751  
## 3 -368 344 513 12.85075 -70.27812  
## 4 -372 344 512 13.43120 -70.39379  
## 5 -374 337 506 13.37872 -70.42856  
## 6 -369 342 513 13.38246 -70.81759  
## 7 -373 336 509 13.12695 -70.24757  
## yaw\_dumbbell total\_accel\_dumbbell gyros\_dumbbell\_x gyros\_dumbbell\_y  
## 2 -84.71065 37 0 -0.02  
## 3 -85.14078 37 0 -0.02  
## 4 -84.87363 37 0 -0.02  
## 5 -84.85306 37 0 -0.02  
## 6 -84.46500 37 0 -0.02  
## 7 -85.09961 37 0 -0.02  
## gyros\_dumbbell\_z accel\_dumbbell\_x accel\_dumbbell\_y accel\_dumbbell\_z  
## 2 0.00 -233 47 -269  
## 3 0.00 -232 46 -270  
## 4 -0.02 -232 48 -269  
## 5 0.00 -233 48 -270  
## 6 0.00 -234 48 -269  
## 7 0.00 -232 47 -270  
## magnet\_dumbbell\_x magnet\_dumbbell\_y magnet\_dumbbell\_z roll\_forearm  
## 2 -555 296 -64 28.3  
## 3 -561 298 -63 28.3  
## 4 -552 303 -60 28.1  
## 5 -554 292 -68 28.0  
## 6 -558 294 -66 27.9  
## 7 -551 295 -70 27.9  
## pitch\_forearm yaw\_forearm total\_accel\_forearm gyros\_forearm\_x  
## 2 -63.9 -153 36 0.02  
## 3 -63.9 -152 36 0.03  
## 4 -63.9 -152 36 0.02  
## 5 -63.9 -152 36 0.02  
## 6 -63.9 -152 36 0.02  
## 7 -63.9 -152 36 0.02  
## gyros\_forearm\_y gyros\_forearm\_z accel\_forearm\_x accel\_forearm\_y  
## 2 0.00 -0.02 192 203  
## 3 -0.02 0.00 196 204  
## 4 -0.02 0.00 189 206  
## 5 0.00 -0.02 189 206  
## 6 -0.02 -0.03 193 203  
## 7 0.00 -0.02 195 205  
## accel\_forearm\_z magnet\_forearm\_x magnet\_forearm\_y magnet\_forearm\_z  
## 2 -216 -18 661 473  
## 3 -213 -18 658 469  
## 4 -214 -16 658 469  
## 5 -214 -17 655 473  
## 6 -215 -9 660 478  
## 7 -215 -18 659 470  
## classe  
## 2 A  
## 3 A  
## 4 A  
## 5 A  
## 6 A  
## 7 A

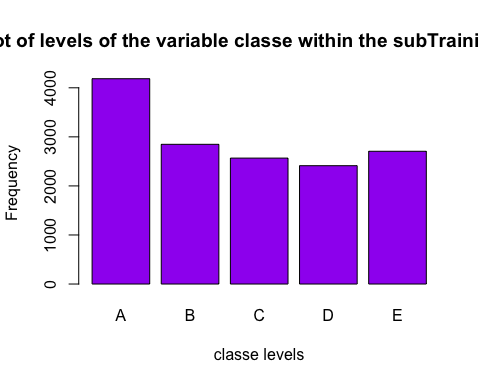
head(subTesting)

## roll\_belt pitch\_belt yaw\_belt total\_accel\_belt gyros\_belt\_x  
## 1 1.41 8.07 -94.4 3 0.00  
## 21 1.60 8.10 -94.4 3 0.02  
## 22 1.57 8.09 -94.4 3 0.02  
## 23 1.56 8.10 -94.3 3 0.02  
## 25 1.53 8.11 -94.4 3 0.03  
## 26 1.55 8.09 -94.4 3 0.02  
## gyros\_belt\_y gyros\_belt\_z accel\_belt\_x accel\_belt\_y accel\_belt\_z  
## 1 0.00 -0.02 -21 4 22  
## 21 0.00 -0.02 -20 1 20  
## 22 0.02 -0.02 -21 3 21  
## 23 0.00 -0.02 -21 4 21  
## 25 0.00 0.00 -19 4 21  
## 26 0.00 0.00 -21 3 22  
## magnet\_belt\_x magnet\_belt\_y magnet\_belt\_z roll\_arm pitch\_arm yaw\_arm  
## 1 -3 599 -313 -128 22.5 -161  
## 21 -10 607 -304 -129 20.9 -161  
## 22 -2 604 -313 -129 20.8 -161  
## 23 -4 606 -311 -129 20.7 -161  
## 25 -8 605 -319 -129 20.7 -161  
## 26 -10 601 -312 -129 20.7 -161  
## total\_accel\_arm gyros\_arm\_x gyros\_arm\_y gyros\_arm\_z accel\_arm\_x  
## 1 34 0.00 0.00 -0.02 -288  
## 21 34 0.03 -0.02 -0.02 -288  
## 22 34 0.03 -0.02 -0.02 -289  
## 23 34 0.02 -0.02 -0.02 -290  
## 25 34 -0.02 -0.02 0.00 -289  
## 26 34 -0.02 -0.02 -0.02 -290  
## accel\_arm\_y accel\_arm\_z magnet\_arm\_x magnet\_arm\_y magnet\_arm\_z  
## 1 109 -123 -368 337 516  
## 21 111 -124 -375 337 513  
## 22 111 -123 -372 338 510  
## 23 110 -123 -373 333 509  
## 25 109 -123 -370 340 512  
## 26 108 -123 -366 346 511  
## roll\_dumbbell pitch\_dumbbell yaw\_dumbbell total\_accel\_dumbbell  
## 1 13.05217 -70.49400 -84.87394 37  
## 21 13.38246 -70.81759 -84.46500 37  
## 22 13.37872 -70.42856 -84.85306 37  
## 23 13.35451 -70.63995 -84.64919 37  
## 25 13.05217 -70.49400 -84.87394 37  
## 26 12.80060 -70.31305 -85.11886 37  
## gyros\_dumbbell\_x gyros\_dumbbell\_y gyros\_dumbbell\_z accel\_dumbbell\_x  
## 1 0 -0.02 0.00 -234  
## 21 0 -0.02 0.00 -234  
## 22 0 -0.02 0.00 -233  
## 23 0 -0.02 0.00 -234  
## 25 0 -0.02 0.00 -234  
## 26 0 -0.02 -0.02 -233  
## accel\_dumbbell\_y accel\_dumbbell\_z magnet\_dumbbell\_x magnet\_dumbbell\_y  
## 1 47 -271 -559 293  
## 21 48 -269 -554 299  
## 22 48 -270 -554 301  
## 23 48 -270 -557 294  
## 25 47 -271 -555 290  
## 26 46 -271 -563 294  
## magnet\_dumbbell\_z roll\_forearm pitch\_forearm yaw\_forearm  
## 1 -65 28.4 -63.9 -153  
## 21 -72 26.9 -63.9 -151  
## 22 -65 27.0 -63.9 -151  
## 23 -69 26.9 -63.8 -151  
## 25 -68 27.1 -63.7 -151  
## 26 -72 27.0 -63.7 -151  
## total\_accel\_forearm gyros\_forearm\_x gyros\_forearm\_y gyros\_forearm\_z  
## 1 36 0.03 0.00 -0.02  
## 21 36 0.03 -0.03 -0.02  
## 22 36 0.02 -0.03 -0.02  
## 23 36 0.02 -0.02 -0.02  
## 25 36 0.05 -0.03 0.00  
## 26 36 0.03 0.00 0.00  
## accel\_forearm\_x accel\_forearm\_y accel\_forearm\_z magnet\_forearm\_x  
## 1 192 203 -215 -17  
## 21 194 208 -214 -11  
## 22 191 206 -213 -17  
## 23 194 206 -214 -10  
## 25 191 202 -214 -14  
## 26 190 203 -216 -16  
## magnet\_forearm\_y magnet\_forearm\_z classe  
## 1 654 476 A  
## 21 654 469 A  
## 22 654 478 A  
## 23 653 467 A  
## 25 667 470 A  
## 26 658 462 A

## A look at the data

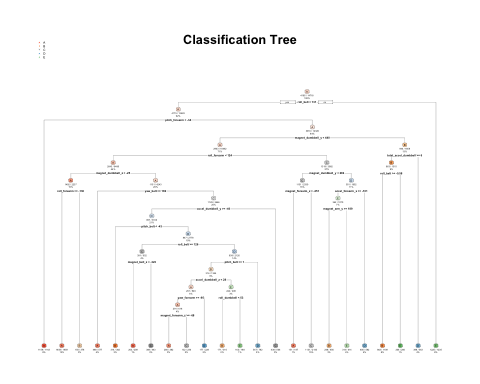
The variable “classe” contains 5 levels: A, B, C, D and E. A plot of the outcome variable will allow us to see the frequency of each levels in the subTraining data set and compare one another.

plot(subTraining$classe, col="purple", main="Bar Plot of levels of the variable classe within the subTraining data set", xlab="classe levels", ylab="Frequency")

 From the graph above, we can see that each level frequency is within the same order of magnitude of each other. Level A is the most frequent with more than 4000 occurrences while level D is the least frequent with about 2500 occurrences.

## First prediction model: Using Decision Tree

model1<-rpart(classe ~ ., data=subTraining, method="class")  
  
# Predicting:  
prediction1 <- predict(model1, subTesting, type = "class")  
  
# Plot of the Decision Tree  
rpart.plot(model1, main="Classification Tree", extra=102, under=TRUE, faclen=0)



# Test results on our subTesting data set:  
confusionMatrix(prediction1, subTesting$classe)

## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction A B C D E  
## A 1235 157 16 50 20  
## B 55 568 73 80 102  
## C 44 125 690 118 116  
## D 41 64 50 508 38  
## E 20 35 26 48 625  
##   
## Overall Statistics  
##   
## Accuracy : 0.7394   
## 95% CI : (0.7269, 0.7516)  
## No Information Rate : 0.2845   
## P-Value [Acc > NIR] : < 2.2e-16   
##   
## Kappa : 0.6697   
## Mcnemar's Test P-Value : < 2.2e-16   
##   
## Statistics by Class:  
##   
## Class: A Class: B Class: C Class: D Class: E  
## Sensitivity 0.8853 0.5985 0.8070 0.6318 0.6937  
## Specificity 0.9307 0.9216 0.9005 0.9529 0.9678  
## Pos Pred Value 0.8356 0.6469 0.6313 0.7247 0.8289  
## Neg Pred Value 0.9533 0.9054 0.9567 0.9296 0.9335  
## Prevalence 0.2845 0.1935 0.1743 0.1639 0.1837  
## Detection Rate 0.2518 0.1158 0.1407 0.1036 0.1274  
## Detection Prevalence 0.3014 0.1790 0.2229 0.1429 0.1538  
## Balanced Accuracy 0.9080 0.7601 0.8537 0.7924 0.8307

## Second prediction model: Using Random Forest

model2 <- randomForest(classe ~. , data=subTraining, method="class")  
  
# Predicting:  
prediction2 <- predict(model2, subTesting, type = "class")  
  
# Test results on subTesting data set:  
confusionMatrix(prediction2, subTesting$classe)

## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction A B C D E  
## A 1395 3 0 0 0  
## B 0 943 10 0 0  
## C 0 3 844 5 0  
## D 0 0 1 799 0  
## E 0 0 0 0 901  
##   
## Overall Statistics  
##   
## Accuracy : 0.9955   
## 95% CI : (0.9932, 0.9972)  
## No Information Rate : 0.2845   
## P-Value [Acc > NIR] : < 2.2e-16   
##   
## Kappa : 0.9943   
## Mcnemar's Test P-Value : NA   
##   
## Statistics by Class:  
##   
## Class: A Class: B Class: C Class: D Class: E  
## Sensitivity 1.0000 0.9937 0.9871 0.9938 1.0000  
## Specificity 0.9991 0.9975 0.9980 0.9998 1.0000  
## Pos Pred Value 0.9979 0.9895 0.9906 0.9988 1.0000  
## Neg Pred Value 1.0000 0.9985 0.9973 0.9988 1.0000  
## Prevalence 0.2845 0.1935 0.1743 0.1639 0.1837  
## Detection Rate 0.2845 0.1923 0.1721 0.1629 0.1837  
## Detection Prevalence 0.2851 0.1943 0.1737 0.1631 0.1837  
## Balanced Accuracy 0.9996 0.9956 0.9926 0.9968 1.0000

## Decision

As expected, Random Forest algorithm performed better than Decision Trees.

# predict outcome levels on the original Testing data set using Random Forest algorithm  
predictfinal <- predict(model2, testingset, type="class")  
predictfinal

## 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20   
## 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