Practical Machine Learning Course Project

K. Maxwell

May 6, 2020

## R Markdown

This is the course project for Practical Machine Learning. Devices such as Jawbone Up, Fitbit, and Nike FuelBand are now making it possible to easily track and obtain personal data on self movement. While it is easily able to obtain data on the quantity of activities performed it is rarely quantified how well the activity is performed. The goal of this project is the use data from accelerometers on the belt, forearm, arm, and dumbell of six participants who have been asked to perform barebell lifts correctly and incorrectly five different ways.

## Running Necessary Packages

library(e1071)  
library(caret)

## Loading required package: lattice

## Loading required package: ggplot2

library(rpart)  
library(rattle)

## Loading required package: tibble

## Loading required package: bitops

## Rattle: A free graphical interface for data science with R.  
## Version 5.4.0 Copyright (c) 2006-2020 Togaware Pty Ltd.  
## Type 'rattle()' to shake, rattle, and roll your data.

library(randomForest)

library(knitr)

## Loading the Data

The training and testing datasets are found at the following urls. The data is titled Assignment\_Training and Assignment\_Testing

train\_url <- "http://d396qusza40orc.cloudfront.net/predmachlearn/pml-training.csv"  
test\_url <- "http://d396qusza40orc.cloudfront.net/predmachlearn/pml-testing.csv"  
  
Assignment\_Training <- read.csv(url(train\_url))  
Assignment\_Testing <- read.csv(url(test\_url))  
colnames(Assignment\_Training)

## [1] "X" "user\_name"   
## [3] "raw\_timestamp\_part\_1" "raw\_timestamp\_part\_2"   
## [5] "cvtd\_timestamp" "new\_window"   
## [7] "num\_window" "roll\_belt"   
## [9] "pitch\_belt" "yaw\_belt"   
## [11] "total\_accel\_belt" "kurtosis\_roll\_belt"   
## [13] "kurtosis\_picth\_belt" "kurtosis\_yaw\_belt"   
## [15] "skewness\_roll\_belt" "skewness\_roll\_belt.1"   
## [17] "skewness\_yaw\_belt" "max\_roll\_belt"   
## [19] "max\_picth\_belt" "max\_yaw\_belt"   
## [21] "min\_roll\_belt" "min\_pitch\_belt"   
## [23] "min\_yaw\_belt" "amplitude\_roll\_belt"   
## [25] "amplitude\_pitch\_belt" "amplitude\_yaw\_belt"   
## [27] "var\_total\_accel\_belt" "avg\_roll\_belt"   
## [29] "stddev\_roll\_belt" "var\_roll\_belt"   
## [31] "avg\_pitch\_belt" "stddev\_pitch\_belt"   
## [33] "var\_pitch\_belt" "avg\_yaw\_belt"   
## [35] "stddev\_yaw\_belt" "var\_yaw\_belt"   
## [37] "gyros\_belt\_x" "gyros\_belt\_y"   
## [39] "gyros\_belt\_z" "accel\_belt\_x"   
## [41] "accel\_belt\_y" "accel\_belt\_z"   
## [43] "magnet\_belt\_x" "magnet\_belt\_y"   
## [45] "magnet\_belt\_z" "roll\_arm"   
## [47] "pitch\_arm" "yaw\_arm"   
## [49] "total\_accel\_arm" "var\_accel\_arm"   
## [51] "avg\_roll\_arm" "stddev\_roll\_arm"   
## [53] "var\_roll\_arm" "avg\_pitch\_arm"   
## [55] "stddev\_pitch\_arm" "var\_pitch\_arm"   
## [57] "avg\_yaw\_arm" "stddev\_yaw\_arm"   
## [59] "var\_yaw\_arm" "gyros\_arm\_x"   
## [61] "gyros\_arm\_y" "gyros\_arm\_z"   
## [63] "accel\_arm\_x" "accel\_arm\_y"   
## [65] "accel\_arm\_z" "magnet\_arm\_x"   
## [67] "magnet\_arm\_y" "magnet\_arm\_z"   
## [69] "kurtosis\_roll\_arm" "kurtosis\_picth\_arm"   
## [71] "kurtosis\_yaw\_arm" "skewness\_roll\_arm"   
## [73] "skewness\_pitch\_arm" "skewness\_yaw\_arm"   
## [75] "max\_roll\_arm" "max\_picth\_arm"   
## [77] "max\_yaw\_arm" "min\_roll\_arm"   
## [79] "min\_pitch\_arm" "min\_yaw\_arm"   
## [81] "amplitude\_roll\_arm" "amplitude\_pitch\_arm"   
## [83] "amplitude\_yaw\_arm" "roll\_dumbbell"   
## [85] "pitch\_dumbbell" "yaw\_dumbbell"   
## [87] "kurtosis\_roll\_dumbbell" "kurtosis\_picth\_dumbbell"   
## [89] "kurtosis\_yaw\_dumbbell" "skewness\_roll\_dumbbell"   
## [91] "skewness\_pitch\_dumbbell" "skewness\_yaw\_dumbbell"   
## [93] "max\_roll\_dumbbell" "max\_picth\_dumbbell"   
## [95] "max\_yaw\_dumbbell" "min\_roll\_dumbbell"   
## [97] "min\_pitch\_dumbbell" "min\_yaw\_dumbbell"   
## [99] "amplitude\_roll\_dumbbell" "amplitude\_pitch\_dumbbell"  
## [101] "amplitude\_yaw\_dumbbell" "total\_accel\_dumbbell"   
## [103] "var\_accel\_dumbbell" "avg\_roll\_dumbbell"   
## [105] "stddev\_roll\_dumbbell" "var\_roll\_dumbbell"   
## [107] "avg\_pitch\_dumbbell" "stddev\_pitch\_dumbbell"   
## [109] "var\_pitch\_dumbbell" "avg\_yaw\_dumbbell"   
## [111] "stddev\_yaw\_dumbbell" "var\_yaw\_dumbbell"   
## [113] "gyros\_dumbbell\_x" "gyros\_dumbbell\_y"   
## [115] "gyros\_dumbbell\_z" "accel\_dumbbell\_x"   
## [117] "accel\_dumbbell\_y" "accel\_dumbbell\_z"   
## [119] "magnet\_dumbbell\_x" "magnet\_dumbbell\_y"   
## [121] "magnet\_dumbbell\_z" "roll\_forearm"   
## [123] "pitch\_forearm" "yaw\_forearm"   
## [125] "kurtosis\_roll\_forearm" "kurtosis\_picth\_forearm"   
## [127] "kurtosis\_yaw\_forearm" "skewness\_roll\_forearm"   
## [129] "skewness\_pitch\_forearm" "skewness\_yaw\_forearm"   
## [131] "max\_roll\_forearm" "max\_picth\_forearm"   
## [133] "max\_yaw\_forearm" "min\_roll\_forearm"   
## [135] "min\_pitch\_forearm" "min\_yaw\_forearm"   
## [137] "amplitude\_roll\_forearm" "amplitude\_pitch\_forearm"   
## [139] "amplitude\_yaw\_forearm" "total\_accel\_forearm"   
## [141] "var\_accel\_forearm" "avg\_roll\_forearm"   
## [143] "stddev\_roll\_forearm" "var\_roll\_forearm"   
## [145] "avg\_pitch\_forearm" "stddev\_pitch\_forearm"   
## [147] "var\_pitch\_forearm" "avg\_yaw\_forearm"   
## [149] "stddev\_yaw\_forearm" "var\_yaw\_forearm"   
## [151] "gyros\_forearm\_x" "gyros\_forearm\_y"   
## [153] "gyros\_forearm\_z" "accel\_forearm\_x"   
## [155] "accel\_forearm\_y" "accel\_forearm\_z"   
## [157] "magnet\_forearm\_x" "magnet\_forearm\_y"   
## [159] "magnet\_forearm\_z" "classe"

dim(Assignment\_Training)

## [1] 19622 160

dim(Assignment\_Testing)

## [1] 20 160

## Data Preperation and Cleaning

Some necessary data preparation is required to clean the data and get it ready for use in the models. The first step is to remove the NAs. The second step is to remove the first 7 variables as they will provide no additional value to the models.

# Data prep: remove NAs  
nas <- sapply(Assignment\_Training, function(x) mean(is.na(x))) > 0.95  
Assignment\_Training <- Assignment\_Training[,nas == FALSE]  
Assignment\_Testing <- Assignment\_Testing[,nas == FALSE]  
dim(Assignment\_Training)

## [1] 19622 93

# Remove columns 1-7  
Assignment\_Training <- Assignment\_Training[,-c(1:7)]  
Assignment\_Testing <- Assignment\_Testing [,-c(1:7)]  
dim(Assignment\_Training)

## [1] 19622 86

## Partitioning the Data into Training and Cross-validation Sets

The training dataset is then partitioned (70%/30%) to create a training dataset and a cross-validation dataset.

intrain = createDataPartition(Assignment\_Training$classe, p=0.70, list = FALSE)  
Training\_final = Assignment\_Training[intrain,]  
Testing\_final = Assignment\_Training[-intrain,]  
  
colnames(Training\_final)

## [1] "roll\_belt" "pitch\_belt"   
## [3] "yaw\_belt" "total\_accel\_belt"   
## [5] "kurtosis\_roll\_belt" "kurtosis\_picth\_belt"   
## [7] "kurtosis\_yaw\_belt" "skewness\_roll\_belt"   
## [9] "skewness\_roll\_belt.1" "skewness\_yaw\_belt"   
## [11] "max\_yaw\_belt" "min\_yaw\_belt"   
## [13] "amplitude\_yaw\_belt" "gyros\_belt\_x"   
## [15] "gyros\_belt\_y" "gyros\_belt\_z"   
## [17] "accel\_belt\_x" "accel\_belt\_y"   
## [19] "accel\_belt\_z" "magnet\_belt\_x"   
## [21] "magnet\_belt\_y" "magnet\_belt\_z"   
## [23] "roll\_arm" "pitch\_arm"   
## [25] "yaw\_arm" "total\_accel\_arm"   
## [27] "gyros\_arm\_x" "gyros\_arm\_y"   
## [29] "gyros\_arm\_z" "accel\_arm\_x"   
## [31] "accel\_arm\_y" "accel\_arm\_z"   
## [33] "magnet\_arm\_x" "magnet\_arm\_y"   
## [35] "magnet\_arm\_z" "kurtosis\_roll\_arm"   
## [37] "kurtosis\_picth\_arm" "kurtosis\_yaw\_arm"   
## [39] "skewness\_roll\_arm" "skewness\_pitch\_arm"   
## [41] "skewness\_yaw\_arm" "roll\_dumbbell"   
## [43] "pitch\_dumbbell" "yaw\_dumbbell"   
## [45] "kurtosis\_roll\_dumbbell" "kurtosis\_picth\_dumbbell"  
## [47] "kurtosis\_yaw\_dumbbell" "skewness\_roll\_dumbbell"   
## [49] "skewness\_pitch\_dumbbell" "skewness\_yaw\_dumbbell"   
## [51] "max\_yaw\_dumbbell" "min\_yaw\_dumbbell"   
## [53] "amplitude\_yaw\_dumbbell" "total\_accel\_dumbbell"   
## [55] "gyros\_dumbbell\_x" "gyros\_dumbbell\_y"   
## [57] "gyros\_dumbbell\_z" "accel\_dumbbell\_x"   
## [59] "accel\_dumbbell\_y" "accel\_dumbbell\_z"   
## [61] "magnet\_dumbbell\_x" "magnet\_dumbbell\_y"   
## [63] "magnet\_dumbbell\_z" "roll\_forearm"   
## [65] "pitch\_forearm" "yaw\_forearm"   
## [67] "kurtosis\_roll\_forearm" "kurtosis\_picth\_forearm"   
## [69] "kurtosis\_yaw\_forearm" "skewness\_roll\_forearm"   
## [71] "skewness\_pitch\_forearm" "skewness\_yaw\_forearm"   
## [73] "max\_yaw\_forearm" "min\_yaw\_forearm"   
## [75] "amplitude\_yaw\_forearm" "total\_accel\_forearm"   
## [77] "gyros\_forearm\_x" "gyros\_forearm\_y"   
## [79] "gyros\_forearm\_z" "accel\_forearm\_x"   
## [81] "accel\_forearm\_y" "accel\_forearm\_z"   
## [83] "magnet\_forearm\_x" "magnet\_forearm\_y"   
## [85] "magnet\_forearm\_z" "classe"

dim(Training\_final)

## [1] 13737 86

dim(Testing\_final)

## [1] 5885 86

## Remove Near Zero Variance

The next step is to remove any variable with near zero variance. These variables have extremely low variances and as a result have very little information because they mostly consist of a single value (e.g. zero).

NZV <- nearZeroVar(Training\_final)  
 Training\_final <- Training\_final[, -NZV]  
 Testing\_final <- Testing\_final[, -NZV]  
 dim(Training\_final)

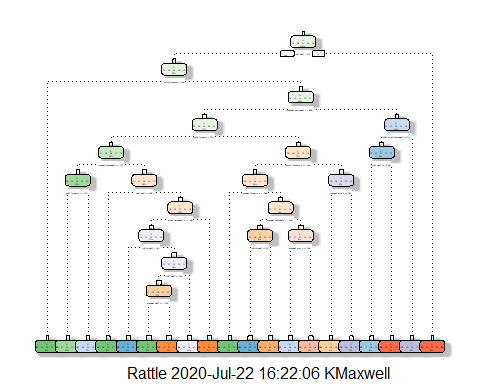
## [1] 13737 53

##Model Building

For this course project two different models are employed: Decision Tree and Random Forest. Evaluation will be done to determine the model with best accuracy. That model will then be used to detect the correction answers to the course project quiz.

## Predicting with a Decision Tree

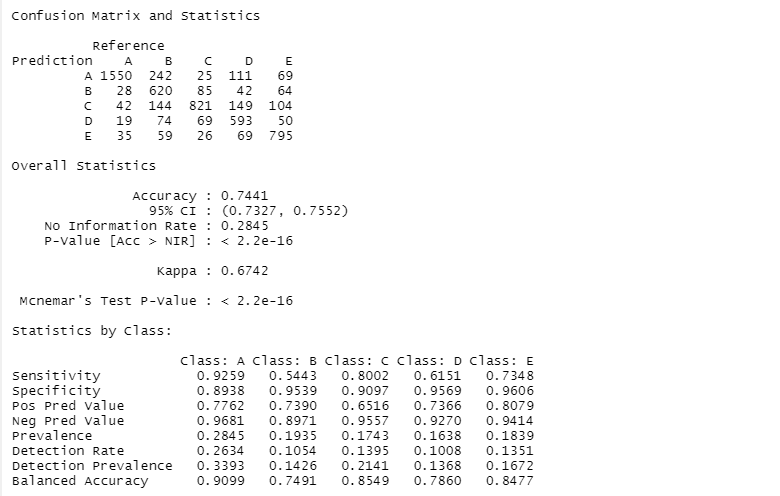
set.seed((12345))  
modelDT <- rpart(classe ~., data = Training\_final, method = "class")  
  
fancyRpartPlot(modelDT)



predictDT <- predict(modelDT, Testing\_final, type = “class”)

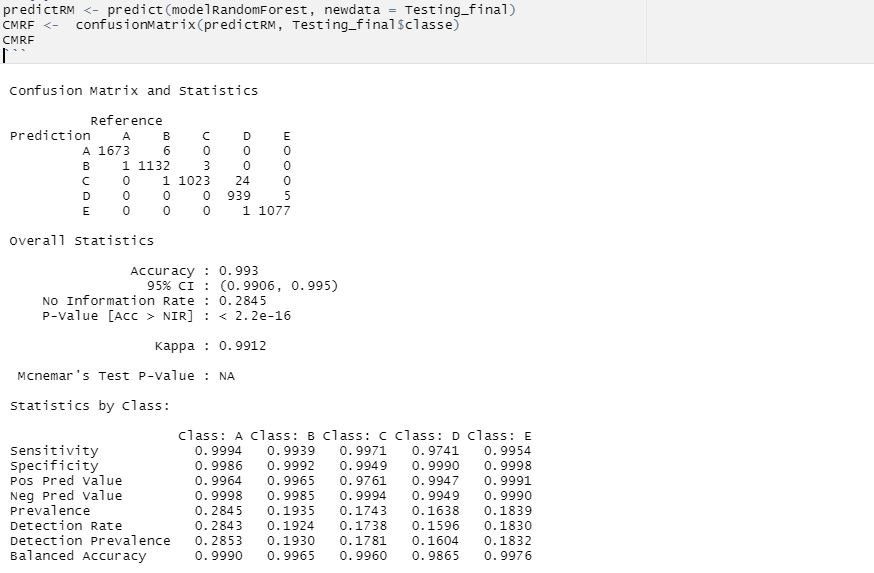
CMDT <- confusionMatrix(factor(predictDT), Testing\_final$classe)

CMDT



## Random Forest  
  
```r  
set.seed(12345)  
controlRF <- trainControl(method="cv", number=3, verboseIter=FALSE)  
modelRandomForest <- train(classe ~., method='rf', data = Training\_final, trControl=controlRF)  
modelRandomForest$finalModel

##   
## Call:  
## randomForest(x = x, y = y, mtry = param$mtry)   
## Type of random forest: classification  
## Number of trees: 500  
## No. of variables tried at each split: 2  
##   
## OOB estimate of error rate: 0.7%  
## Confusion matrix:  
## A B C D E class.error  
## A 3902 4 0 0 0 0.001024066  
## B 13 2640 5 0 0 0.006772009  
## C 0 22 2368 6 0 0.011686144  
## D 0 0 36 2215 1 0.016429840  
## E 0 0 1 8 2516 0.003564356



## Selecting the Best Model

The next step is to select the best model based on accuracy:

Decision Tree: 74% Accuracy

Random Forest: 99% Accuracy

The model with the best accuracy is the Random Forest model and that will be used to determine the answers to the final quiz.

BestModel <- predict(modelRandomForest, newdata = Assignment\_Testing)  
BestModel

## [1] 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