Machine Learning Project Assignment

Amit Kumar

2 August 2018

# load required packages  
  
library(caret)

## Warning: package 'caret' was built under R version 3.4.4

## Loading required package: lattice

## Loading required package: ggplot2

library(rattle)

## Warning: package 'rattle' was built under R version 3.4.4

## Rattle: A free graphical interface for data science with R.  
## Version 5.1.0 Copyright (c) 2006-2017 Togaware Pty Ltd.  
## Type 'rattle()' to shake, rattle, and roll your data.

### Load training and test data   
  
TrainData <- read.csv(url("https://d396qusza40orc.cloudfront.net/predmachlearn/pml-training.csv"),header=TRUE)  
dim(TrainData)

## [1] 19622 160

TestData <- read.csv(url("https://d396qusza40orc.cloudfront.net/predmachlearn/pml-testing.csv"),header=TRUE)  
dim(TestData)

## [1] 20 160

head(TestData)

## X user\_name raw\_timestamp\_part\_1 raw\_timestamp\_part\_2 cvtd\_timestamp  
## 1 1 pedro 1323095002 868349 05/12/2011 14:23  
## 2 2 jeremy 1322673067 778725 30/11/2011 17:11  
## 3 3 jeremy 1322673075 342967 30/11/2011 17:11  
## 4 4 adelmo 1322832789 560311 02/12/2011 13:33  
## 5 5 eurico 1322489635 814776 28/11/2011 14:13  
## 6 6 jeremy 1322673149 510661 30/11/2011 17:12  
## new\_window num\_window roll\_belt pitch\_belt yaw\_belt total\_accel\_belt  
## 1 no 74 123.00 27.00 -4.75 20  
## 2 no 431 1.02 4.87 -88.90 4  
## 3 no 439 0.87 1.82 -88.50 5  
## 4 no 194 125.00 -41.60 162.00 17  
## 5 no 235 1.35 3.33 -88.60 3  
## 6 no 504 -5.92 1.59 -87.70 4  
## kurtosis\_roll\_belt kurtosis\_picth\_belt kurtosis\_yaw\_belt  
## 1 NA NA NA  
## 2 NA NA NA  
## 3 NA NA NA  
## 4 NA NA NA  
## 5 NA NA NA  
## 6 NA NA NA  
## skewness\_roll\_belt skewness\_roll\_belt.1 skewness\_yaw\_belt max\_roll\_belt  
## 1 NA NA NA NA  
## 2 NA NA NA NA  
## 3 NA NA NA NA  
## 4 NA NA NA NA  
## 5 NA NA NA NA  
## 6 NA NA NA NA  
## max\_picth\_belt max\_yaw\_belt min\_roll\_belt min\_pitch\_belt min\_yaw\_belt  
## 1 NA NA NA NA NA  
## 2 NA NA NA NA NA  
## 3 NA NA NA NA NA  
## 4 NA NA NA NA NA  
## 5 NA NA NA NA NA  
## 6 NA NA NA NA NA  
## amplitude\_roll\_belt amplitude\_pitch\_belt amplitude\_yaw\_belt  
## 1 NA NA NA  
## 2 NA NA NA  
## 3 NA NA NA  
## 4 NA NA NA  
## 5 NA NA NA  
## 6 NA NA NA  
## var\_total\_accel\_belt avg\_roll\_belt stddev\_roll\_belt var\_roll\_belt  
## 1 NA NA NA NA  
## 2 NA NA NA NA  
## 3 NA NA NA NA  
## 4 NA NA NA NA  
## 5 NA NA NA NA  
## 6 NA NA NA NA  
## avg\_pitch\_belt stddev\_pitch\_belt var\_pitch\_belt avg\_yaw\_belt  
## 1 NA NA NA NA  
## 2 NA NA NA NA  
## 3 NA NA NA NA  
## 4 NA NA NA NA  
## 5 NA NA NA NA  
## 6 NA NA NA NA  
## stddev\_yaw\_belt var\_yaw\_belt gyros\_belt\_x gyros\_belt\_y gyros\_belt\_z  
## 1 NA NA -0.50 -0.02 -0.46  
## 2 NA NA -0.06 -0.02 -0.07  
## 3 NA NA 0.05 0.02 0.03  
## 4 NA NA 0.11 0.11 -0.16  
## 5 NA NA 0.03 0.02 0.00  
## 6 NA NA 0.10 0.05 -0.13  
## accel\_belt\_x accel\_belt\_y accel\_belt\_z magnet\_belt\_x magnet\_belt\_y  
## 1 -38 69 -179 -13 581  
## 2 -13 11 39 43 636  
## 3 1 -1 49 29 631  
## 4 46 45 -156 169 608  
## 5 -8 4 27 33 566  
## 6 -11 -16 38 31 638  
## magnet\_belt\_z roll\_arm pitch\_arm yaw\_arm total\_accel\_arm var\_accel\_arm  
## 1 -382 40.7 -27.80 178 10 NA  
## 2 -309 0.0 0.00 0 38 NA  
## 3 -312 0.0 0.00 0 44 NA  
## 4 -304 -109.0 55.00 -142 25 NA  
## 5 -418 76.1 2.76 102 29 NA  
## 6 -291 0.0 0.00 0 14 NA  
## avg\_roll\_arm stddev\_roll\_arm var\_roll\_arm avg\_pitch\_arm stddev\_pitch\_arm  
## 1 NA NA NA NA NA  
## 2 NA NA NA NA NA  
## 3 NA NA NA NA NA  
## 4 NA NA NA NA NA  
## 5 NA NA NA NA NA  
## 6 NA NA NA NA NA  
## var\_pitch\_arm avg\_yaw\_arm stddev\_yaw\_arm var\_yaw\_arm gyros\_arm\_x  
## 1 NA NA NA NA -1.65  
## 2 NA NA NA NA -1.17  
## 3 NA NA NA NA 2.10  
## 4 NA NA NA NA 0.22  
## 5 NA NA NA NA -1.96  
## 6 NA NA NA NA 0.02  
## gyros\_arm\_y gyros\_arm\_z accel\_arm\_x accel\_arm\_y accel\_arm\_z magnet\_arm\_x  
## 1 0.48 -0.18 16 38 93 -326  
## 2 0.85 -0.43 -290 215 -90 -325  
## 3 -1.36 1.13 -341 245 -87 -264  
## 4 -0.51 0.92 -238 -57 6 -173  
## 5 0.79 -0.54 -197 200 -30 -170  
## 6 0.05 -0.07 -26 130 -19 396  
## magnet\_arm\_y magnet\_arm\_z kurtosis\_roll\_arm kurtosis\_picth\_arm  
## 1 385 481 NA NA  
## 2 447 434 NA NA  
## 3 474 413 NA NA  
## 4 257 633 NA NA  
## 5 275 617 NA NA  
## 6 176 516 NA NA  
## kurtosis\_yaw\_arm skewness\_roll\_arm skewness\_pitch\_arm skewness\_yaw\_arm  
## 1 NA NA NA NA  
## 2 NA NA NA NA  
## 3 NA NA NA NA  
## 4 NA NA NA NA  
## 5 NA NA NA NA  
## 6 NA NA NA NA  
## max\_roll\_arm max\_picth\_arm max\_yaw\_arm min\_roll\_arm min\_pitch\_arm  
## 1 NA NA NA NA NA  
## 2 NA NA NA NA NA  
## 3 NA NA NA NA NA  
## 4 NA NA NA NA NA  
## 5 NA NA NA NA NA  
## 6 NA NA NA NA NA  
## min\_yaw\_arm amplitude\_roll\_arm amplitude\_pitch\_arm amplitude\_yaw\_arm  
## 1 NA NA NA NA  
## 2 NA NA NA NA  
## 3 NA NA NA NA  
## 4 NA NA NA NA  
## 5 NA NA NA NA  
## 6 NA NA NA NA  
## roll\_dumbbell pitch\_dumbbell yaw\_dumbbell kurtosis\_roll\_dumbbell  
## 1 -17.73748 24.96085 126.23596 NA  
## 2 54.47761 -53.69758 -75.51480 NA  
## 3 57.07031 -51.37303 -75.20287 NA  
## 4 43.10927 -30.04885 -103.32003 NA  
## 5 -101.38396 -53.43952 -14.19542 NA  
## 6 62.18750 -50.55595 -71.12063 NA  
## kurtosis\_picth\_dumbbell kurtosis\_yaw\_dumbbell skewness\_roll\_dumbbell  
## 1 NA NA NA  
## 2 NA NA NA  
## 3 NA NA NA  
## 4 NA NA NA  
## 5 NA NA NA  
## 6 NA NA NA  
## skewness\_pitch\_dumbbell skewness\_yaw\_dumbbell max\_roll\_dumbbell  
## 1 NA NA NA  
## 2 NA NA NA  
## 3 NA NA NA  
## 4 NA NA NA  
## 5 NA NA NA  
## 6 NA NA NA  
## max\_picth\_dumbbell max\_yaw\_dumbbell min\_roll\_dumbbell min\_pitch\_dumbbell  
## 1 NA NA NA NA  
## 2 NA NA NA NA  
## 3 NA NA NA NA  
## 4 NA NA NA NA  
## 5 NA NA NA NA  
## 6 NA NA NA NA  
## min\_yaw\_dumbbell amplitude\_roll\_dumbbell amplitude\_pitch\_dumbbell  
## 1 NA NA NA  
## 2 NA NA NA  
## 3 NA NA NA  
## 4 NA NA NA  
## 5 NA NA NA  
## 6 NA NA NA  
## amplitude\_yaw\_dumbbell total\_accel\_dumbbell var\_accel\_dumbbell  
## 1 NA 9 NA  
## 2 NA 31 NA  
## 3 NA 29 NA  
## 4 NA 18 NA  
## 5 NA 4 NA  
## 6 NA 29 NA  
## avg\_roll\_dumbbell stddev\_roll\_dumbbell var\_roll\_dumbbell  
## 1 NA NA NA  
## 2 NA NA NA  
## 3 NA NA NA  
## 4 NA NA NA  
## 5 NA NA NA  
## 6 NA NA NA  
## avg\_pitch\_dumbbell stddev\_pitch\_dumbbell var\_pitch\_dumbbell  
## 1 NA NA NA  
## 2 NA NA NA  
## 3 NA NA NA  
## 4 NA NA NA  
## 5 NA NA NA  
## 6 NA NA NA  
## avg\_yaw\_dumbbell stddev\_yaw\_dumbbell var\_yaw\_dumbbell gyros\_dumbbell\_x  
## 1 NA NA NA 0.64  
## 2 NA NA NA 0.34  
## 3 NA NA NA 0.39  
## 4 NA NA NA 0.10  
## 5 NA NA NA 0.29  
## 6 NA NA NA -0.59  
## gyros\_dumbbell\_y gyros\_dumbbell\_z accel\_dumbbell\_x accel\_dumbbell\_y  
## 1 0.06 -0.61 21 -15  
## 2 0.05 -0.71 -153 155  
## 3 0.14 -0.34 -141 155  
## 4 -0.02 0.05 -51 72  
## 5 -0.47 -0.46 -18 -30  
## 6 0.80 1.10 -138 166  
## accel\_dumbbell\_z magnet\_dumbbell\_x magnet\_dumbbell\_y magnet\_dumbbell\_z  
## 1 81 523 -528 -56  
## 2 -205 -502 388 -36  
## 3 -196 -506 349 41  
## 4 -148 -576 238 53  
## 5 -5 -424 252 312  
## 6 -186 -543 262 96  
## roll\_forearm pitch\_forearm yaw\_forearm kurtosis\_roll\_forearm  
## 1 141 49.30 156.0 NA  
## 2 109 -17.60 106.0 NA  
## 3 131 -32.60 93.0 NA  
## 4 0 0.00 0.0 NA  
## 5 -176 -2.16 -47.9 NA  
## 6 150 1.46 89.7 NA  
## kurtosis\_picth\_forearm kurtosis\_yaw\_forearm skewness\_roll\_forearm  
## 1 NA NA NA  
## 2 NA NA NA  
## 3 NA NA NA  
## 4 NA NA NA  
## 5 NA NA NA  
## 6 NA NA NA  
## skewness\_pitch\_forearm skewness\_yaw\_forearm max\_roll\_forearm  
## 1 NA NA NA  
## 2 NA NA NA  
## 3 NA NA NA  
## 4 NA NA NA  
## 5 NA NA NA  
## 6 NA NA NA  
## max\_picth\_forearm max\_yaw\_forearm min\_roll\_forearm min\_pitch\_forearm  
## 1 NA NA NA NA  
## 2 NA NA NA NA  
## 3 NA NA NA NA  
## 4 NA NA NA NA  
## 5 NA NA NA NA  
## 6 NA NA NA NA  
## min\_yaw\_forearm amplitude\_roll\_forearm amplitude\_pitch\_forearm  
## 1 NA NA NA  
## 2 NA NA NA  
## 3 NA NA NA  
## 4 NA NA NA  
## 5 NA NA NA  
## 6 NA NA NA  
## amplitude\_yaw\_forearm total\_accel\_forearm var\_accel\_forearm  
## 1 NA 33 NA  
## 2 NA 39 NA  
## 3 NA 34 NA  
## 4 NA 43 NA  
## 5 NA 24 NA  
## 6 NA 43 NA  
## avg\_roll\_forearm stddev\_roll\_forearm var\_roll\_forearm avg\_pitch\_forearm  
## 1 NA NA NA NA  
## 2 NA NA NA NA  
## 3 NA NA NA NA  
## 4 NA NA NA NA  
## 5 NA NA NA NA  
## 6 NA NA NA NA  
## stddev\_pitch\_forearm var\_pitch\_forearm avg\_yaw\_forearm  
## 1 NA NA NA  
## 2 NA NA NA  
## 3 NA NA NA  
## 4 NA NA NA  
## 5 NA NA NA  
## 6 NA NA NA  
## stddev\_yaw\_forearm var\_yaw\_forearm gyros\_forearm\_x gyros\_forearm\_y  
## 1 NA NA 0.74 -3.34  
## 2 NA NA 1.12 -2.78  
## 3 NA NA 0.18 -0.79  
## 4 NA NA 1.38 0.69  
## 5 NA NA -0.75 3.10  
## 6 NA NA -0.88 4.26  
## gyros\_forearm\_z accel\_forearm\_x accel\_forearm\_y accel\_forearm\_z  
## 1 -0.59 -110 267 -149  
## 2 -0.18 212 297 -118  
## 3 0.28 154 271 -129  
## 4 1.80 -92 406 -39  
## 5 0.80 131 -93 172  
## 6 1.35 230 322 -144  
## magnet\_forearm\_x magnet\_forearm\_y magnet\_forearm\_z problem\_id  
## 1 -714 419 617 1  
## 2 -237 791 873 2  
## 3 -51 698 783 3  
## 4 -233 783 521 4  
## 5 375 -787 91 5  
## 6 -300 800 884 6

str(TrainData)

## 'data.frame': 19622 obs. of 160 variables:  
## $ X : int 1 2 3 4 5 6 7 8 9 10 ...  
## $ user\_name : Factor w/ 6 levels "adelmo","carlitos",..: 2 2 2 2 2 2 2 2 2 2 ...  
## $ raw\_timestamp\_part\_1 : int 1323084231 1323084231 1323084231 1323084232 1323084232 1323084232 1323084232 1323084232 1323084232 1323084232 ...  
## $ raw\_timestamp\_part\_2 : int 788290 808298 820366 120339 196328 304277 368296 440390 484323 484434 ...  
## $ cvtd\_timestamp : Factor w/ 20 levels "02/12/2011 13:32",..: 9 9 9 9 9 9 9 9 9 9 ...  
## $ new\_window : Factor w/ 2 levels "no","yes": 1 1 1 1 1 1 1 1 1 1 ...  
## $ num\_window : int 11 11 11 12 12 12 12 12 12 12 ...  
## $ roll\_belt : num 1.41 1.41 1.42 1.48 1.48 1.45 1.42 1.42 1.43 1.45 ...  
## $ pitch\_belt : num 8.07 8.07 8.07 8.05 8.07 8.06 8.09 8.13 8.16 8.17 ...  
## $ yaw\_belt : num -94.4 -94.4 -94.4 -94.4 -94.4 -94.4 -94.4 -94.4 -94.4 -94.4 ...  
## $ total\_accel\_belt : int 3 3 3 3 3 3 3 3 3 3 ...  
## $ kurtosis\_roll\_belt : Factor w/ 397 levels "","-0.016850",..: 1 1 1 1 1 1 1 1 1 1 ...  
## $ kurtosis\_picth\_belt : Factor w/ 317 levels "","-0.021887",..: 1 1 1 1 1 1 1 1 1 1 ...  
## $ kurtosis\_yaw\_belt : Factor w/ 2 levels "","#DIV/0!": 1 1 1 1 1 1 1 1 1 1 ...  
## $ skewness\_roll\_belt : Factor w/ 395 levels "","-0.003095",..: 1 1 1 1 1 1 1 1 1 1 ...  
## $ skewness\_roll\_belt.1 : Factor w/ 338 levels "","-0.005928",..: 1 1 1 1 1 1 1 1 1 1 ...  
## $ skewness\_yaw\_belt : Factor w/ 2 levels "","#DIV/0!": 1 1 1 1 1 1 1 1 1 1 ...  
## $ max\_roll\_belt : num NA NA NA NA NA NA NA NA NA NA ...  
## $ max\_picth\_belt : int NA NA NA NA NA NA NA NA NA NA ...  
## $ max\_yaw\_belt : Factor w/ 68 levels "","-0.1","-0.2",..: 1 1 1 1 1 1 1 1 1 1 ...  
## $ min\_roll\_belt : num NA NA NA NA NA NA NA NA NA NA ...  
## $ min\_pitch\_belt : int NA NA NA NA NA NA NA NA NA NA ...  
## $ min\_yaw\_belt : Factor w/ 68 levels "","-0.1","-0.2",..: 1 1 1 1 1 1 1 1 1 1 ...  
## $ amplitude\_roll\_belt : num NA NA NA NA NA NA NA NA NA NA ...  
## $ amplitude\_pitch\_belt : int NA NA NA NA NA NA NA NA NA NA ...  
## $ amplitude\_yaw\_belt : Factor w/ 4 levels "","#DIV/0!","0.00",..: 1 1 1 1 1 1 1 1 1 1 ...  
## $ var\_total\_accel\_belt : num NA NA NA NA NA NA NA NA NA NA ...  
## $ avg\_roll\_belt : num NA NA NA NA NA NA NA NA NA NA ...  
## $ stddev\_roll\_belt : num NA NA NA NA NA NA NA NA NA NA ...  
## $ var\_roll\_belt : num NA NA NA NA NA NA NA NA NA NA ...  
## $ avg\_pitch\_belt : num NA NA NA NA NA NA NA NA NA NA ...  
## $ stddev\_pitch\_belt : num NA NA NA NA NA NA NA NA NA NA ...  
## $ var\_pitch\_belt : num NA NA NA NA NA NA NA NA NA NA ...  
## $ avg\_yaw\_belt : num NA NA NA NA NA NA NA NA NA NA ...  
## $ stddev\_yaw\_belt : num NA NA NA NA NA NA NA NA NA NA ...  
## $ var\_yaw\_belt : num NA NA NA NA NA NA NA NA NA NA ...  
## $ gyros\_belt\_x : num 0 0.02 0 0.02 0.02 0.02 0.02 0.02 0.02 0.03 ...  
## $ gyros\_belt\_y : num 0 0 0 0 0.02 0 0 0 0 0 ...  
## $ gyros\_belt\_z : num -0.02 -0.02 -0.02 -0.03 -0.02 -0.02 -0.02 -0.02 -0.02 0 ...  
## $ accel\_belt\_x : int -21 -22 -20 -22 -21 -21 -22 -22 -20 -21 ...  
## $ accel\_belt\_y : int 4 4 5 3 2 4 3 4 2 4 ...  
## $ accel\_belt\_z : int 22 22 23 21 24 21 21 21 24 22 ...  
## $ magnet\_belt\_x : int -3 -7 -2 -6 -6 0 -4 -2 1 -3 ...  
## $ magnet\_belt\_y : int 599 608 600 604 600 603 599 603 602 609 ...  
## $ magnet\_belt\_z : int -313 -311 -305 -310 -302 -312 -311 -313 -312 -308 ...  
## $ roll\_arm : num -128 -128 -128 -128 -128 -128 -128 -128 -128 -128 ...  
## $ pitch\_arm : num 22.5 22.5 22.5 22.1 22.1 22 21.9 21.8 21.7 21.6 ...  
## $ yaw\_arm : num -161 -161 -161 -161 -161 -161 -161 -161 -161 -161 ...  
## $ total\_accel\_arm : int 34 34 34 34 34 34 34 34 34 34 ...  
## $ var\_accel\_arm : num NA NA NA NA NA NA NA NA NA NA ...  
## $ avg\_roll\_arm : num NA NA NA NA NA NA NA NA NA NA ...  
## $ stddev\_roll\_arm : num NA NA NA NA NA NA NA NA NA NA ...  
## $ var\_roll\_arm : num NA NA NA NA NA NA NA NA NA NA ...  
## $ avg\_pitch\_arm : num NA NA NA NA NA NA NA NA NA NA ...  
## $ stddev\_pitch\_arm : num NA NA NA NA NA NA NA NA NA NA ...  
## $ var\_pitch\_arm : num NA NA NA NA NA NA NA NA NA NA ...  
## $ avg\_yaw\_arm : num NA NA NA NA NA NA NA NA NA NA ...  
## $ stddev\_yaw\_arm : num NA NA NA NA NA NA NA NA NA NA ...  
## $ var\_yaw\_arm : num NA NA NA NA NA NA NA NA NA NA ...  
## $ gyros\_arm\_x : num 0 0.02 0.02 0.02 0 0.02 0 0.02 0.02 0.02 ...  
## $ gyros\_arm\_y : num 0 -0.02 -0.02 -0.03 -0.03 -0.03 -0.03 -0.02 -0.03 -0.03 ...  
## $ gyros\_arm\_z : num -0.02 -0.02 -0.02 0.02 0 0 0 0 -0.02 -0.02 ...  
## $ accel\_arm\_x : int -288 -290 -289 -289 -289 -289 -289 -289 -288 -288 ...  
## $ accel\_arm\_y : int 109 110 110 111 111 111 111 111 109 110 ...  
## $ accel\_arm\_z : int -123 -125 -126 -123 -123 -122 -125 -124 -122 -124 ...  
## $ magnet\_arm\_x : int -368 -369 -368 -372 -374 -369 -373 -372 -369 -376 ...  
## $ magnet\_arm\_y : int 337 337 344 344 337 342 336 338 341 334 ...  
## $ magnet\_arm\_z : int 516 513 513 512 506 513 509 510 518 516 ...  
## $ kurtosis\_roll\_arm : Factor w/ 330 levels "","-0.02438",..: 1 1 1 1 1 1 1 1 1 1 ...  
## $ kurtosis\_picth\_arm : Factor w/ 328 levels "","-0.00484",..: 1 1 1 1 1 1 1 1 1 1 ...  
## $ kurtosis\_yaw\_arm : Factor w/ 395 levels "","-0.01548",..: 1 1 1 1 1 1 1 1 1 1 ...  
## $ skewness\_roll\_arm : Factor w/ 331 levels "","-0.00051",..: 1 1 1 1 1 1 1 1 1 1 ...  
## $ skewness\_pitch\_arm : Factor w/ 328 levels "","-0.00184",..: 1 1 1 1 1 1 1 1 1 1 ...  
## $ skewness\_yaw\_arm : Factor w/ 395 levels "","-0.00311",..: 1 1 1 1 1 1 1 1 1 1 ...  
## $ max\_roll\_arm : num NA NA NA NA NA NA NA NA NA NA ...  
## $ max\_picth\_arm : num NA NA NA NA NA NA NA NA NA NA ...  
## $ max\_yaw\_arm : int NA NA NA NA NA NA NA NA NA NA ...  
## $ min\_roll\_arm : num NA NA NA NA NA NA NA NA NA NA ...  
## $ min\_pitch\_arm : num NA NA NA NA NA NA NA NA NA NA ...  
## $ min\_yaw\_arm : int NA NA NA NA NA NA NA NA NA NA ...  
## $ amplitude\_roll\_arm : num NA NA NA NA NA NA NA NA NA NA ...  
## $ amplitude\_pitch\_arm : num NA NA NA NA NA NA NA NA NA NA ...  
## $ amplitude\_yaw\_arm : int NA NA NA NA NA NA NA NA NA NA ...  
## $ roll\_dumbbell : num 13.1 13.1 12.9 13.4 13.4 ...  
## $ pitch\_dumbbell : num -70.5 -70.6 -70.3 -70.4 -70.4 ...  
## $ yaw\_dumbbell : num -84.9 -84.7 -85.1 -84.9 -84.9 ...  
## $ kurtosis\_roll\_dumbbell : Factor w/ 398 levels "","-0.0035","-0.0073",..: 1 1 1 1 1 1 1 1 1 1 ...  
## $ kurtosis\_picth\_dumbbell : Factor w/ 401 levels "","-0.0163","-0.0233",..: 1 1 1 1 1 1 1 1 1 1 ...  
## $ kurtosis\_yaw\_dumbbell : Factor w/ 2 levels "","#DIV/0!": 1 1 1 1 1 1 1 1 1 1 ...  
## $ skewness\_roll\_dumbbell : Factor w/ 401 levels "","-0.0082","-0.0096",..: 1 1 1 1 1 1 1 1 1 1 ...  
## $ skewness\_pitch\_dumbbell : Factor w/ 402 levels "","-0.0053","-0.0084",..: 1 1 1 1 1 1 1 1 1 1 ...  
## $ skewness\_yaw\_dumbbell : Factor w/ 2 levels "","#DIV/0!": 1 1 1 1 1 1 1 1 1 1 ...  
## $ max\_roll\_dumbbell : num NA NA NA NA NA NA NA NA NA NA ...  
## $ max\_picth\_dumbbell : num NA NA NA NA NA NA NA NA NA NA ...  
## $ max\_yaw\_dumbbell : Factor w/ 73 levels "","-0.1","-0.2",..: 1 1 1 1 1 1 1 1 1 1 ...  
## $ min\_roll\_dumbbell : num NA NA NA NA NA NA NA NA NA NA ...  
## $ min\_pitch\_dumbbell : num NA NA NA NA NA NA NA NA NA NA ...  
## $ min\_yaw\_dumbbell : Factor w/ 73 levels "","-0.1","-0.2",..: 1 1 1 1 1 1 1 1 1 1 ...  
## $ amplitude\_roll\_dumbbell : num NA NA NA NA NA NA NA NA NA NA ...  
## [list output truncated]

# Observations

1.The training data set is made of 19622 observations on 160 columns. 2.And,We can notice that many columns have NA values or blank values on almost every observation. So we will remove them, because they will not produce any information. 3.The first seven columns give information about the people who did the test, and also timestamps. We will not take them in our model.

# Here we get the indexes of the columns having at least 90% of NA or blank values on the training dataset  
indColToRemove <- which(colSums(is.na(TrainData) |TrainData=="")>0.9\*dim(TrainData)[1])  
# remove these blank variables from both training datasets  
TrainDataClean <- TrainData[,-indColToRemove]  
##### removing first 7 columns  
TrainDataClean <- TrainDataClean[,-c(1:7)]  
dim(TrainDataClean)

## [1] 19622 53

# We do the same operation on test dataset  
indColToRemove <- which(colSums(is.na(TestData) |TestData=="")>0.9\*dim(TestData)[1])   
TestDataClean <- TestData[,-indColToRemove]  
TestDataClean <- TestDataClean[,-1]  
dim(TestDataClean)

## [1] 20 59

#Note: After cleaning, the new training data set has only 53 columns  
  
# Here we create a partition of the traning data set   
set.seed(12345)  
inTrain1 <- createDataPartition(TrainDataClean$classe, p=0.75, list=FALSE)  
Train1 <- TrainDataClean[inTrain1,]  
Test1 <- TrainDataClean[-inTrain1,]  
dim(Train1)

## [1] 14718 53

dim(Test1)

## [1] 4904 53

# Next Steps

Now, let us try following 3 different models: 1. Classification Trees 2. Random Forest 3. Gradient Boosting Method

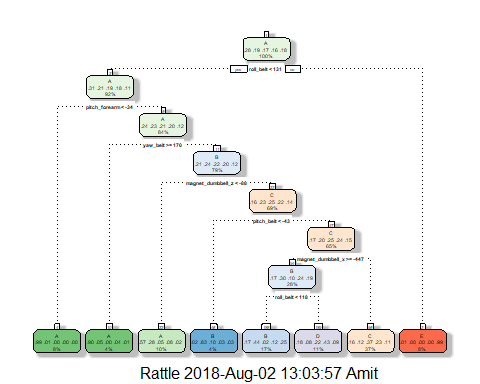
In order to limit the effects of overfitting, and improve the efficicency of the models, we will usecross-validation technique. We will use 5 folds (usually, 5 or 10 can be used, but 10 folds gives higher run times with no significant increase of the accuracy).

trControl <- trainControl(method="cv", number=5)  
model\_CT <- train(classe~., data=Train1, method="rpart", trControl=trControl)  
  
print(model\_CT)

## CART   
##   
## 14718 samples  
## 52 predictor  
## 5 classes: 'A', 'B', 'C', 'D', 'E'   
##   
## No pre-processing  
## Resampling: Cross-Validated (5 fold)   
## Summary of sample sizes: 11775, 11773, 11774, 11775, 11775   
## Resampling results across tuning parameters:  
##   
## cp Accuracy Kappa   
## 0.02430457 0.5758210 0.46200827  
## 0.04158359 0.4862738 0.32874504  
## 0.11706067 0.3167502 0.04962369  
##   
## Accuracy was used to select the optimal model using the largest value.  
## The final value used for the model was cp = 0.02430457.

fancyRpartPlot(model\_CT$finalModel)

## Warning: Bad 'data' field in model 'call'.  
## To silence this warning:  
## Call prp with roundint=FALSE,  
## or rebuild the rpart model with model=TRUE.



trainpred <- predict(model\_CT,newdata=Test1)  
  
confMatCT <- confusionMatrix(Test1$classe,trainpred)  
  
# display confusion matrix and model accuracy  
confMatCT$table

## Reference  
## Prediction A B C D E  
## A 870 159 273 88 5  
## B 162 530 214 43 0  
## C 29 36 674 116 0  
## D 46 136 429 193 0  
## E 16 221 224 51 389

confMatCT$overall[1]

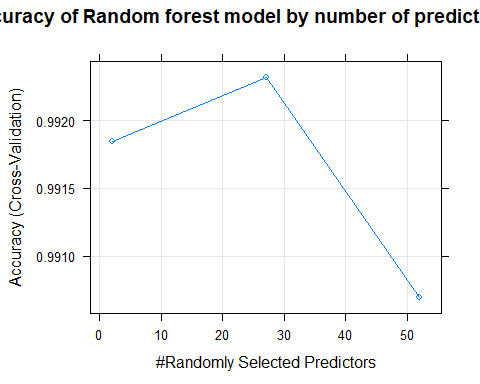
## Accuracy   
## 0.5415987

# Next, Let us try to fit Random Forest Model

model\_RF <- train(classe~., data=Train1, method="rf", trControl=trControl, verbose=FALSE)  
print(model\_RF)

## Random Forest   
##   
## 14718 samples  
## 52 predictor  
## 5 classes: 'A', 'B', 'C', 'D', 'E'   
##   
## No pre-processing  
## Resampling: Cross-Validated (5 fold)   
## Summary of sample sizes: 11776, 11775, 11773, 11774, 11774   
## Resampling results across tuning parameters:  
##   
## mtry Accuracy Kappa   
## 2 0.9918466 0.9896855  
## 27 0.9923226 0.9902884  
## 52 0.9906918 0.9882252  
##   
## Accuracy was used to select the optimal model using the largest value.  
## The final value used for the model was mtry = 27.

plot(model\_RF,main="Accuracy of Random forest model by number of predictors")



trainpred <- predict(model\_RF,newdata=Test1)  
  
confMatRF <- confusionMatrix(Test1$classe,trainpred)  
  
# display confusion matrix and model accuracy  
confMatRF$table

## Reference  
## Prediction A B C D E  
## A 1394 1 0 0 0  
## B 6 939 4 0 0  
## C 0 2 849 4 0  
## D 0 0 10 794 0  
## E 0 0 2 5 894

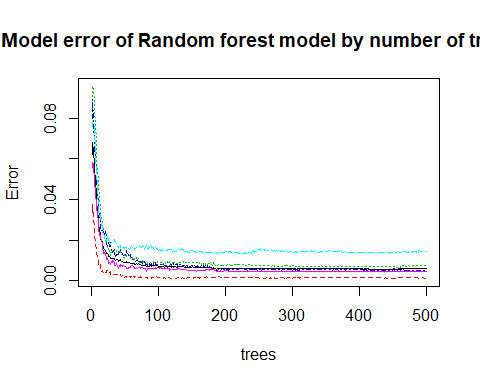
confMatRF$overall[1]

## Accuracy   
## 0.9930669

model\_RF$finalModel$classes

## [1] "A" "B" "C" "D" "E"

plot(model\_RF$finalModel,main="Model error of Random forest model by number of trees")



# Observation from Random Forest

With random forest, we reach an accuracy of 99.3% using cross-validation with 5 steps. This is very good. But let’s see what we can expect with Gradient boosting.

We can also notice that the optimal number of predictors, i.e. the number of predictors giving the highest accuracy, is 27. There is no significal increase of the accuracy with 2 predictors and 27, but the slope decreases more with more than 27 predictors (even if the accuracy is still very good). The fact that not all the accuracy is worse with all the available predictors lets us suggest that there may be some dependencies between them.

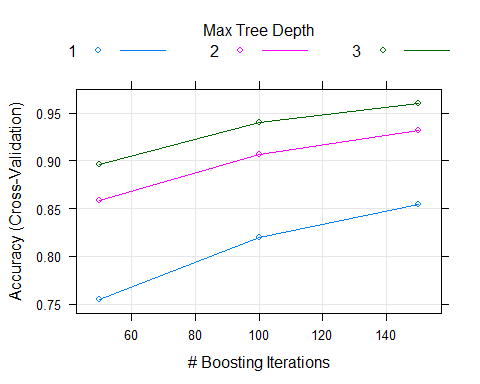
At last, using more than about 30 trees does not reduce the error significantly.

# Train with gradient boosting method

model\_GBM <- train(classe~., data=Train1, method="gbm", trControl=trControl, verbose=FALSE)  
print(model\_GBM)

## Stochastic Gradient Boosting   
##   
## 14718 samples  
## 52 predictor  
## 5 classes: 'A', 'B', 'C', 'D', 'E'   
##   
## No pre-processing  
## Resampling: Cross-Validated (5 fold)   
## Summary of sample sizes: 11775, 11775, 11774, 11773, 11775   
## Resampling results across tuning parameters:  
##   
## interaction.depth n.trees Accuracy Kappa   
## 1 50 0.7547203 0.6891712  
## 1 100 0.8194732 0.7714678  
## 1 150 0.8541930 0.8154916  
## 2 50 0.8585420 0.8207615  
## 2 100 0.9064419 0.8816012  
## 2 150 0.9317853 0.9136787  
## 3 50 0.8959101 0.8682235  
## 3 100 0.9406855 0.9249385  
## 3 150 0.9605249 0.9500546  
##   
## Tuning parameter 'shrinkage' was held constant at a value of 0.1  
##   
## Tuning parameter 'n.minobsinnode' was held constant at a value of 10  
## Accuracy was used to select the optimal model using the largest value.  
## The final values used for the model were n.trees = 150,  
## interaction.depth = 3, shrinkage = 0.1 and n.minobsinnode = 10.

plot(model\_GBM)



trainpred <- predict(model\_GBM,newdata=Test1)  
  
confMatGBM <- confusionMatrix(Test1$classe,trainpred)  
confMatGBM$table

## Reference  
## Prediction A B C D E  
## A 1378 14 3 0 0  
## B 39 874 34 1 1  
## C 0 20 824 10 1  
## D 0 1 21 776 6  
## E 3 7 8 16 867

confMatGBM$overall[1]

## Accuracy   
## 0.9622757

# Conclusion

This shows that the random forest model is the best one. We will then use it to predict the values of classe for the test data set.

FinalTestPred <- predict(model\_RF,newdata=TestDataClean)  
FinalTestPred

## [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

## R Markdown