Practical Machine Learning Course Project

YZZHENG

Saturday, March 21, 2015

### Get and set to the working directory

getwd()

## [1] "C:/GitH/Practical Machine Learning/PML-Course-Project"

setwd("C:/GitH/Practical Machine Learning/PML-Course-Project")  
dir()

## [1] "Course\_Project.xps" "CP1.html" "CP1.Rmd"   
## [4] "index.html" "pml-testing.csv" "pml-training.csv"   
## [7] "ReadMe.md" "results"

### Read files into R and define the NA strings

training <- read.csv("pml-training.csv", na.strings=c("NA","#DIV/0!", " "))  
testing <- read.csv("pml-testing.csv", na.strings=c("NA","#DIV/0!", " "))  
dim(training); dim(testing)

## [1] 19622 160

## [1] 20 160

### Find the columns which contain the "NA"

na\_cols1 <- sapply(training, function(x)any(is.na(x)))  
na\_cols2 <- sapply(testing, function(x)any(is.na(x)))  
table(na\_cols1)

## na\_cols1  
## FALSE TRUE   
## 60 100

table(na\_cols2)

## na\_cols2  
## FALSE TRUE   
## 60 100

### Remove the "NA" containing columns

training1 <- training[, !(na\_cols1)]  
testing1 <- testing[, !(na\_cols2)]  
dim(training1); dim(testing1)

## [1] 19622 60

## [1] 20 60

str(training1)

## 'data.frame': 19622 obs. of 60 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 ...  
## $ 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 ...  
## $ 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 ...  
## $ 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 ...  
## $ total\_accel\_dumbbell: int 37 37 37 37 37 37 37 37 37 37 ...  
## $ gyros\_dumbbell\_x : num 0 0 0 0 0 0 0 0 0 0 ...  
## $ gyros\_dumbbell\_y : num -0.02 -0.02 -0.02 -0.02 -0.02 -0.02 -0.02 -0.02 -0.02 -0.02 ...  
## $ gyros\_dumbbell\_z : num 0 0 0 -0.02 0 0 0 0 0 0 ...  
## $ accel\_dumbbell\_x : int -234 -233 -232 -232 -233 -234 -232 -234 -232 -235 ...  
## $ accel\_dumbbell\_y : int 47 47 46 48 48 48 47 46 47 48 ...  
## $ accel\_dumbbell\_z : int -271 -269 -270 -269 -270 -269 -270 -272 -269 -270 ...  
## $ magnet\_dumbbell\_x : int -559 -555 -561 -552 -554 -558 -551 -555 -549 -558 ...  
## $ magnet\_dumbbell\_y : int 293 296 298 303 292 294 295 300 292 291 ...  
## $ magnet\_dumbbell\_z : num -65 -64 -63 -60 -68 -66 -70 -74 -65 -69 ...  
## $ roll\_forearm : num 28.4 28.3 28.3 28.1 28 27.9 27.9 27.8 27.7 27.7 ...  
## $ pitch\_forearm : num -63.9 -63.9 -63.9 -63.9 -63.9 -63.9 -63.9 -63.8 -63.8 -63.8 ...  
## $ yaw\_forearm : num -153 -153 -152 -152 -152 -152 -152 -152 -152 -152 ...  
## $ total\_accel\_forearm : int 36 36 36 36 36 36 36 36 36 36 ...  
## $ gyros\_forearm\_x : num 0.03 0.02 0.03 0.02 0.02 0.02 0.02 0.02 0.03 0.02 ...  
## $ gyros\_forearm\_y : num 0 0 -0.02 -0.02 0 -0.02 0 -0.02 0 0 ...  
## $ gyros\_forearm\_z : num -0.02 -0.02 0 0 -0.02 -0.03 -0.02 0 -0.02 -0.02 ...  
## $ accel\_forearm\_x : int 192 192 196 189 189 193 195 193 193 190 ...  
## $ accel\_forearm\_y : int 203 203 204 206 206 203 205 205 204 205 ...  
## $ accel\_forearm\_z : int -215 -216 -213 -214 -214 -215 -215 -213 -214 -215 ...  
## $ magnet\_forearm\_x : int -17 -18 -18 -16 -17 -9 -18 -9 -16 -22 ...  
## $ magnet\_forearm\_y : num 654 661 658 658 655 660 659 660 653 656 ...  
## $ magnet\_forearm\_z : num 476 473 469 469 473 478 470 474 476 473 ...  
## $ classe : Factor w/ 5 levels "A","B","C","D",..: 1 1 1 1 1 1 1 1 1 1 ...

str(testing1)

## 'data.frame': 20 obs. of 60 variables:  
## $ X : int 1 2 3 4 5 6 7 8 9 10 ...  
## $ user\_name : Factor w/ 6 levels "adelmo","carlitos",..: 6 5 5 1 4 5 5 5 2 3 ...  
## $ raw\_timestamp\_part\_1: int 1323095002 1322673067 1322673075 1322832789 1322489635 1322673149 1322673128 1322673076 1323084240 1322837822 ...  
## $ raw\_timestamp\_part\_2: int 868349 778725 342967 560311 814776 510661 766645 54671 916313 384285 ...  
## $ cvtd\_timestamp : Factor w/ 11 levels "02/12/2011 13:33",..: 5 10 10 1 6 11 11 10 3 2 ...  
## $ new\_window : Factor w/ 1 level "no": 1 1 1 1 1 1 1 1 1 1 ...  
## $ num\_window : int 74 431 439 194 235 504 485 440 323 664 ...  
## $ roll\_belt : num 123 1.02 0.87 125 1.35 -5.92 1.2 0.43 0.93 114 ...  
## $ pitch\_belt : num 27 4.87 1.82 -41.6 3.33 1.59 4.44 4.15 6.72 22.4 ...  
## $ yaw\_belt : num -4.75 -88.9 -88.5 162 -88.6 -87.7 -87.3 -88.5 -93.7 -13.1 ...  
## $ total\_accel\_belt : int 20 4 5 17 3 4 4 4 4 18 ...  
## $ gyros\_belt\_x : num -0.5 -0.06 0.05 0.11 0.03 0.1 -0.06 -0.18 0.1 0.14 ...  
## $ gyros\_belt\_y : num -0.02 -0.02 0.02 0.11 0.02 0.05 0 -0.02 0 0.11 ...  
## $ gyros\_belt\_z : num -0.46 -0.07 0.03 -0.16 0 -0.13 0 -0.03 -0.02 -0.16 ...  
## $ accel\_belt\_x : int -38 -13 1 46 -8 -11 -14 -10 -15 -25 ...  
## $ accel\_belt\_y : int 69 11 -1 45 4 -16 2 -2 1 63 ...  
## $ accel\_belt\_z : int -179 39 49 -156 27 38 35 42 32 -158 ...  
## $ magnet\_belt\_x : int -13 43 29 169 33 31 50 39 -6 10 ...  
## $ magnet\_belt\_y : int 581 636 631 608 566 638 622 635 600 601 ...  
## $ magnet\_belt\_z : int -382 -309 -312 -304 -418 -291 -315 -305 -302 -330 ...  
## $ roll\_arm : num 40.7 0 0 -109 76.1 0 0 0 -137 -82.4 ...  
## $ pitch\_arm : num -27.8 0 0 55 2.76 0 0 0 11.2 -63.8 ...  
## $ yaw\_arm : num 178 0 0 -142 102 0 0 0 -167 -75.3 ...  
## $ total\_accel\_arm : int 10 38 44 25 29 14 15 22 34 32 ...  
## $ gyros\_arm\_x : num -1.65 -1.17 2.1 0.22 -1.96 0.02 2.36 -3.71 0.03 0.26 ...  
## $ gyros\_arm\_y : num 0.48 0.85 -1.36 -0.51 0.79 0.05 -1.01 1.85 -0.02 -0.5 ...  
## $ gyros\_arm\_z : num -0.18 -0.43 1.13 0.92 -0.54 -0.07 0.89 -0.69 -0.02 0.79 ...  
## $ accel\_arm\_x : int 16 -290 -341 -238 -197 -26 99 -98 -287 -301 ...  
## $ accel\_arm\_y : int 38 215 245 -57 200 130 79 175 111 -42 ...  
## $ accel\_arm\_z : int 93 -90 -87 6 -30 -19 -67 -78 -122 -80 ...  
## $ magnet\_arm\_x : int -326 -325 -264 -173 -170 396 702 535 -367 -420 ...  
## $ magnet\_arm\_y : int 385 447 474 257 275 176 15 215 335 294 ...  
## $ magnet\_arm\_z : int 481 434 413 633 617 516 217 385 520 493 ...  
## $ roll\_dumbbell : num -17.7 54.5 57.1 43.1 -101.4 ...  
## $ pitch\_dumbbell : num 25 -53.7 -51.4 -30 -53.4 ...  
## $ yaw\_dumbbell : num 126.2 -75.5 -75.2 -103.3 -14.2 ...  
## $ total\_accel\_dumbbell: int 9 31 29 18 4 29 29 29 3 2 ...  
## $ gyros\_dumbbell\_x : num 0.64 0.34 0.39 0.1 0.29 -0.59 0.34 0.37 0.03 0.42 ...  
## $ gyros\_dumbbell\_y : num 0.06 0.05 0.14 -0.02 -0.47 0.8 0.16 0.14 -0.21 0.51 ...  
## $ gyros\_dumbbell\_z : num -0.61 -0.71 -0.34 0.05 -0.46 1.1 -0.23 -0.39 -0.21 -0.03 ...  
## $ accel\_dumbbell\_x : int 21 -153 -141 -51 -18 -138 -145 -140 0 -7 ...  
## $ accel\_dumbbell\_y : int -15 155 155 72 -30 166 150 159 25 -20 ...  
## $ accel\_dumbbell\_z : int 81 -205 -196 -148 -5 -186 -190 -191 9 7 ...  
## $ magnet\_dumbbell\_x : int 523 -502 -506 -576 -424 -543 -484 -515 -519 -531 ...  
## $ magnet\_dumbbell\_y : int -528 388 349 238 252 262 354 350 348 321 ...  
## $ magnet\_dumbbell\_z : int -56 -36 41 53 312 96 97 53 -32 -164 ...  
## $ roll\_forearm : num 141 109 131 0 -176 150 155 -161 15.5 13.2 ...  
## $ pitch\_forearm : num 49.3 -17.6 -32.6 0 -2.16 1.46 34.5 43.6 -63.5 19.4 ...  
## $ yaw\_forearm : num 156 106 93 0 -47.9 89.7 152 -89.5 -139 -105 ...  
## $ total\_accel\_forearm : int 33 39 34 43 24 43 32 47 36 24 ...  
## $ gyros\_forearm\_x : num 0.74 1.12 0.18 1.38 -0.75 -0.88 -0.53 0.63 0.03 0.02 ...  
## $ gyros\_forearm\_y : num -3.34 -2.78 -0.79 0.69 3.1 4.26 1.8 -0.74 0.02 0.13 ...  
## $ gyros\_forearm\_z : num -0.59 -0.18 0.28 1.8 0.8 1.35 0.75 0.49 -0.02 -0.07 ...  
## $ accel\_forearm\_x : int -110 212 154 -92 131 230 -192 -151 195 -212 ...  
## $ accel\_forearm\_y : int 267 297 271 406 -93 322 170 -331 204 98 ...  
## $ accel\_forearm\_z : int -149 -118 -129 -39 172 -144 -175 -282 -217 -7 ...  
## $ magnet\_forearm\_x : int -714 -237 -51 -233 375 -300 -678 -109 0 -403 ...  
## $ magnet\_forearm\_y : int 419 791 698 783 -787 800 284 -619 652 723 ...  
## $ magnet\_forearm\_z : int 617 873 783 521 91 884 585 -32 469 512 ...  
## $ problem\_id : int 1 2 3 4 5 6 7 8 9 10 ...

### Load caret package and further remove the first column from the training data, the first and last columns from the test dataset

library(caret)

## Loading required package: lattice  
## Loading required package: ggplot2

training1 <- training1[, -1]  
testing1 <- testing1[, -c(1, 60)]

### Remove the near zero variance predictors

nzv1 <- nearZeroVar(training1, saveMetrics=TRUE)  
nzv2 <- nearZeroVar(testing1, saveMetrics=TRUE)  
nzv1; nzv2

## freqRatio percentUnique zeroVar nzv  
## user\_name 1.100679 0.03057792 FALSE FALSE  
## raw\_timestamp\_part\_1 1.000000 4.26562022 FALSE FALSE  
## raw\_timestamp\_part\_2 1.000000 85.53154622 FALSE FALSE  
## cvtd\_timestamp 1.000668 0.10192641 FALSE FALSE  
## new\_window 47.330049 0.01019264 FALSE TRUE  
## num\_window 1.000000 4.37264295 FALSE FALSE  
## roll\_belt 1.101904 6.77810621 FALSE FALSE  
## pitch\_belt 1.036082 9.37722964 FALSE FALSE  
## yaw\_belt 1.058480 9.97349913 FALSE FALSE  
## total\_accel\_belt 1.063160 0.14779329 FALSE FALSE  
## gyros\_belt\_x 1.058651 0.71348486 FALSE FALSE  
## gyros\_belt\_y 1.144000 0.35164611 FALSE FALSE  
## gyros\_belt\_z 1.066214 0.86127816 FALSE FALSE  
## accel\_belt\_x 1.055412 0.83579655 FALSE FALSE  
## accel\_belt\_y 1.113725 0.72877383 FALSE FALSE  
## accel\_belt\_z 1.078767 1.52379982 FALSE FALSE  
## magnet\_belt\_x 1.090141 1.66649679 FALSE FALSE  
## magnet\_belt\_y 1.099688 1.51870350 FALSE FALSE  
## magnet\_belt\_z 1.006369 2.32901845 FALSE FALSE  
## roll\_arm 52.338462 13.52563449 FALSE FALSE  
## pitch\_arm 87.256410 15.73234125 FALSE FALSE  
## yaw\_arm 33.029126 14.65701763 FALSE FALSE  
## total\_accel\_arm 1.024526 0.33635715 FALSE FALSE  
## gyros\_arm\_x 1.015504 3.27693405 FALSE FALSE  
## gyros\_arm\_y 1.454369 1.91621649 FALSE FALSE  
## gyros\_arm\_z 1.110687 1.26388747 FALSE FALSE  
## accel\_arm\_x 1.017341 3.95984099 FALSE FALSE  
## accel\_arm\_y 1.140187 2.73672409 FALSE FALSE  
## accel\_arm\_z 1.128000 4.03628580 FALSE FALSE  
## magnet\_arm\_x 1.000000 6.82397309 FALSE FALSE  
## magnet\_arm\_y 1.056818 4.44399144 FALSE FALSE  
## magnet\_arm\_z 1.036364 6.44684538 FALSE FALSE  
## roll\_dumbbell 1.022388 84.20650290 FALSE FALSE  
## pitch\_dumbbell 2.277372 81.74498012 FALSE FALSE  
## yaw\_dumbbell 1.132231 83.48282540 FALSE FALSE  
## total\_accel\_dumbbell 1.072634 0.21914178 FALSE FALSE  
## gyros\_dumbbell\_x 1.003268 1.22821323 FALSE FALSE  
## gyros\_dumbbell\_y 1.264957 1.41677709 FALSE FALSE  
## gyros\_dumbbell\_z 1.060100 1.04984201 FALSE FALSE  
## accel\_dumbbell\_x 1.018018 2.16593619 FALSE FALSE  
## accel\_dumbbell\_y 1.053061 2.37488533 FALSE FALSE  
## accel\_dumbbell\_z 1.133333 2.08949139 FALSE FALSE  
## magnet\_dumbbell\_x 1.098266 5.74864948 FALSE FALSE  
## magnet\_dumbbell\_y 1.197740 4.30129447 FALSE FALSE  
## magnet\_dumbbell\_z 1.020833 3.44511263 FALSE FALSE  
## roll\_forearm 11.589286 11.08959331 FALSE FALSE  
## pitch\_forearm 65.983051 14.85577413 FALSE FALSE  
## yaw\_forearm 15.322835 10.14677403 FALSE FALSE  
## total\_accel\_forearm 1.128928 0.35674243 FALSE FALSE  
## gyros\_forearm\_x 1.059273 1.51870350 FALSE FALSE  
## gyros\_forearm\_y 1.036554 3.77637346 FALSE FALSE  
## gyros\_forearm\_z 1.122917 1.56457038 FALSE FALSE  
## accel\_forearm\_x 1.126437 4.04647844 FALSE FALSE  
## accel\_forearm\_y 1.059406 5.11160942 FALSE FALSE  
## accel\_forearm\_z 1.006250 2.95586586 FALSE FALSE  
## magnet\_forearm\_x 1.012346 7.76679238 FALSE FALSE  
## magnet\_forearm\_y 1.246914 9.54031189 FALSE FALSE  
## magnet\_forearm\_z 1.000000 8.57710733 FALSE FALSE  
## classe 1.469581 0.02548160 FALSE FALSE

## freqRatio percentUnique zeroVar nzv  
## user\_name 2.000000 30 FALSE FALSE  
## raw\_timestamp\_part\_1 1.000000 100 FALSE FALSE  
## raw\_timestamp\_part\_2 1.000000 100 FALSE FALSE  
## cvtd\_timestamp 1.333333 55 FALSE FALSE  
## new\_window 0.000000 5 TRUE TRUE  
## num\_window 1.000000 100 FALSE FALSE  
## roll\_belt 1.000000 90 FALSE FALSE  
## pitch\_belt 1.000000 100 FALSE FALSE  
## yaw\_belt 1.000000 90 FALSE FALSE  
## total\_accel\_belt 1.200000 45 FALSE FALSE  
## gyros\_belt\_x 1.500000 70 FALSE FALSE  
## gyros\_belt\_y 2.250000 30 FALSE FALSE  
## gyros\_belt\_z 2.500000 65 FALSE FALSE  
## accel\_belt\_x 1.000000 85 FALSE FALSE  
## accel\_belt\_y 1.500000 75 FALSE FALSE  
## accel\_belt\_z 1.000000 85 FALSE FALSE  
## magnet\_belt\_x 1.000000 100 FALSE FALSE  
## magnet\_belt\_y 1.500000 80 FALSE FALSE  
## magnet\_belt\_z 1.500000 80 FALSE FALSE  
## roll\_arm 8.000000 65 FALSE FALSE  
## pitch\_arm 8.000000 65 FALSE FALSE  
## yaw\_arm 8.000000 65 FALSE FALSE  
## total\_accel\_arm 1.000000 85 FALSE FALSE  
## gyros\_arm\_x 2.000000 95 FALSE FALSE  
## gyros\_arm\_y 2.000000 95 FALSE FALSE  
## gyros\_arm\_z 1.000000 90 FALSE FALSE  
## accel\_arm\_x 2.000000 95 FALSE FALSE  
## accel\_arm\_y 1.000000 100 FALSE FALSE  
## accel\_arm\_z 1.000000 100 FALSE FALSE  
## magnet\_arm\_x 1.000000 100 FALSE FALSE  
## magnet\_arm\_y 2.000000 95 FALSE FALSE  
## magnet\_arm\_z 1.000000 100 FALSE FALSE  
## roll\_dumbbell 1.000000 100 FALSE FALSE  
## pitch\_dumbbell 1.000000 100 FALSE FALSE  
## yaw\_dumbbell 1.000000 100 FALSE FALSE  
## total\_accel\_dumbbell 2.000000 70 FALSE FALSE  
## gyros\_dumbbell\_x 1.000000 90 FALSE FALSE  
## gyros\_dumbbell\_y 1.000000 80 FALSE FALSE  
## gyros\_dumbbell\_z 4.000000 85 FALSE FALSE  
## accel\_dumbbell\_x 1.000000 100 FALSE FALSE  
## accel\_dumbbell\_y 3.000000 90 FALSE FALSE  
## accel\_dumbbell\_z 2.000000 95 FALSE FALSE  
## magnet\_dumbbell\_x 1.000000 100 FALSE FALSE  
## magnet\_dumbbell\_y 1.000000 100 FALSE FALSE  
## magnet\_dumbbell\_z 2.000000 95 FALSE FALSE  
## roll\_forearm 2.000000 95 FALSE FALSE  
## pitch\_forearm 1.000000 100 FALSE FALSE  
## yaw\_forearm 1.000000 100 FALSE FALSE  
## total\_accel\_forearm 1.500000 65 FALSE FALSE  
## gyros\_forearm\_x 1.000000 90 FALSE FALSE  
## gyros\_forearm\_y 1.000000 100 FALSE FALSE  
## gyros\_forearm\_z 1.000000 100 FALSE FALSE  
## accel\_forearm\_x 1.000000 100 FALSE FALSE  
## accel\_forearm\_y 1.000000 100 FALSE FALSE  
## accel\_forearm\_z 1.000000 100 FALSE FALSE  
## magnet\_forearm\_x 1.000000 100 FALSE FALSE  
## magnet\_forearm\_y 1.000000 100 FALSE FALSE  
## magnet\_forearm\_z 1.000000 100 FALSE FALSE

training1 <- training1[, -5]  
testing1 <- testing1[, -5]

### Create training and validation sets in training data for cross-validation

inTrain <- createDataPartition(y=training1$classe, p=0.7, list=FALSE)  
training1a <- training1[inTrain, ]  
training1b <- training1[-inTrain, ]  
dim(training1a); dim(training1b)

## [1] 13737 58

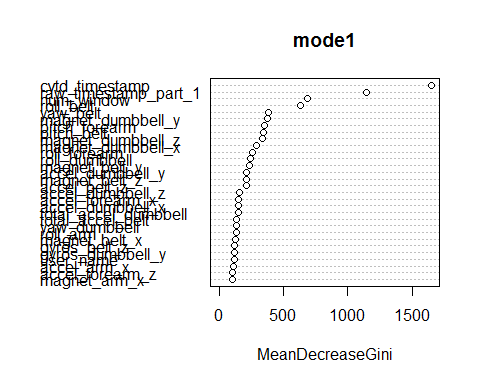
## [1] 5885 58

### Build models, first try random forest model algorithm

r library(randomForest)

## randomForest 4.6-10 ## Type rfNews() to see new features/changes/bug fixes.

r mode1 <- randomForest(classe ~ ., data=training1a) varImpPlot(mode1)



### Cross validate using the training validation data set

pred1 <- predict(mode1, training1b)  
confusionMatrix(training1b$classe, pred1)

## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction A B C D E  
## A 1674 0 0 0 0  
## B 0 1139 0 0 0  
## C 0 2 1019 5 0  
## D 0 0 3 960 1  
## E 0 0 0 0 1082  
##   
## Overall Statistics  
##   
## Accuracy : 0.9981   
## 95% CI : (0.9967, 0.9991)  
## No Information Rate : 0.2845   
## P-Value [Acc > NIR] : < 2.2e-16   
##   
## Kappa : 0.9976   
## Mcnemar's Test P-Value : NA   
##   
## Statistics by Class:  
##   
## Class: A Class: B Class: C Class: D Class: E  
## Sensitivity 1.0000 0.9982 0.9971 0.9948 0.9991  
## Specificity 1.0000 1.0000 0.9986 0.9992 1.0000  
## Pos Pred Value 1.0000 1.0000 0.9932 0.9959 1.0000  
## Neg Pred Value 1.0000 0.9996 0.9994 0.9990 0.9998  
## Prevalence 0.2845 0.1939 0.1737 0.1640 0.1840  
## Detection Rate 0.2845 0.1935 0.1732 0.1631 0.1839  
## Detection Prevalence 0.2845 0.1935 0.1743 0.1638 0.1839  
## Balanced Accuracy 1.0000 0.9991 0.9978 0.9970 0.9995

#### The model 1 give accuracy 99.99%, indicating very small expecting out of sample error. But "predict(mode1, testing1)" model cannot predict the real testing data samples due to the different column types between training and testing datasets.

### Try to find the different types of training and testing data sets

table(sapply(training1, class))

##   
## factor integer numeric   
## 3 28 27

table(sapply(testing1, class))

##   
## factor integer numeric   
## 2 31 24

### Fix the column class difference between training and testing data sets by applying the column head from training data to testing data set.

newFrame <- head(training1, 1)  
newFrame <- newFrame[, -length(colnames(newFrame))]  
newTesting1 <- rbind(newFrame, testing1)  
str(newTesting1)

## 'data.frame': 21 obs. of 57 variables:  
## $ user\_name : Factor w/ 6 levels "adelmo","carlitos",..: 2 6 5 5 1 4 5 5 5 2 ...  
## $ raw\_timestamp\_part\_1: int 1323084231 1323095002 1322673067 1322673075 1322832789 1322489635 1322673149 1322673128 1322673076 1323084240 ...  
## $ raw\_timestamp\_part\_2: int 788290 868349 778725 342967 560311 814776 510661 766645 54671 916313 ...  
## $ cvtd\_timestamp : Factor w/ 20 levels "02/12/2011 13:32",..: 9 13 19 19 2 15 20 20 19 10 ...  
## $ num\_window : int 11 74 431 439 194 235 504 485 440 323 ...  
## $ roll\_belt : num 1.41 123 1.02 0.87 125 1.35 -5.92 1.2 0.43 0.93 ...  
## $ pitch\_belt : num 8.07 27 4.87 1.82 -41.6 3.33 1.59 4.44 4.15 6.72 ...  
## $ yaw\_belt : num -94.4 -4.75 -88.9 -88.5 162 -88.6 -87.7 -87.3 -88.5 -93.7 ...  
## $ total\_accel\_belt : int 3 20 4 5 17 3 4 4 4 4 ...  
## $ gyros\_belt\_x : num 0 -0.5 -0.06 0.05 0.11 0.03 0.1 -0.06 -0.18 0.1 ...  
## $ gyros\_belt\_y : num 0 -0.02 -0.02 0.02 0.11 0.02 0.05 0 -0.02 0 ...  
## $ gyros\_belt\_z : num -0.02 -0.46 -0.07 0.03 -0.16 0 -0.13 0 -0.03 -0.02 ...  
## $ accel\_belt\_x : int -21 -38 -13 1 46 -8 -11 -14 -10 -15 ...  
## $ accel\_belt\_y : int 4 69 11 -1 45 4 -16 2 -2 1 ...  
## $ accel\_belt\_z : int 22 -179 39 49 -156 27 38 35 42 32 ...  
## $ magnet\_belt\_x : int -3 -13 43 29 169 33 31 50 39 -6 ...  
## $ magnet\_belt\_y : int 599 581 636 631 608 566 638 622 635 600 ...  
## $ magnet\_belt\_z : int -313 -382 -309 -312 -304 -418 -291 -315 -305 -302 ...  
## $ roll\_arm : num -128 40.7 0 0 -109 76.1 0 0 0 -137 ...  
## $ pitch\_arm : num 22.5 -27.8 0 0 55 2.76 0 0 0 11.2 ...  
## $ yaw\_arm : num -161 178 0 0 -142 102 0 0 0 -167 ...  
## $ total\_accel\_arm : int 34 10 38 44 25 29 14 15 22 34 ...  
## $ gyros\_arm\_x : num 0 -1.65 -1.17 2.1 0.22 -1.96 0.02 2.36 -3.71 0.03 ...  
## $ gyros\_arm\_y : num 0 0.48 0.85 -1.36 -0.51 0.79 0.05 -1.01 1.85 -0.02 ...  
## $ gyros\_arm\_z : num -0.02 -0.18 -0.43 1.13 0.92 -0.54 -0.07 0.89 -0.69 -0.02 ...  
## $ accel\_arm\_x : int -288 16 -290 -341 -238 -197 -26 99 -98 -287 ...  
## $ accel\_arm\_y : int 109 38 215 245 -57 200 130 79 175 111 ...  
## $ accel\_arm\_z : int -123 93 -90 -87 6 -30 -19 -67 -78 -122 ...  
## $ magnet\_arm\_x : int -368 -326 -325 -264 -173 -170 396 702 535 -367 ...  
## $ magnet\_arm\_y : int 337 385 447 474 257 275 176 15 215 335 ...  
## $ magnet\_arm\_z : int 516 481 434 413 633 617 516 217 385 520 ...  
## $ roll\_dumbbell : num 13.1 -17.7 54.5 57.1 43.1 ...  
## $ pitch\_dumbbell : num -70.5 25 -53.7 -51.4 -30 ...  
## $ yaw\_dumbbell : num -84.9 126.2 -75.5 -75.2 -103.3 ...  
## $ total\_accel\_dumbbell: int 37 9 31 29 18 4 29 29 29 3 ...  
## $ gyros\_dumbbell\_x : num 0 0.64 0.34 0.39 0.1 0.29 -0.59 0.34 0.37 0.03 ...  
## $ gyros\_dumbbell\_y : num -0.02 0.06 0.05 0.14 -0.02 -0.47 0.8 0.16 0.14 -0.21 ...  
## $ gyros\_dumbbell\_z : num 0 -0.61 -0.71 -0.34 0.05 -0.46 1.1 -0.23 -0.39 -0.21 ...  
## $ accel\_dumbbell\_x : int -234 21 -153 -141 -51 -18 -138 -145 -140 0 ...  
## $ accel\_dumbbell\_y : int 47 -15 155 155 72 -30 166 150 159 25 ...  
## $ accel\_dumbbell\_z : int -271 81 -205 -196 -148 -5 -186 -190 -191 9 ...  
## $ magnet\_dumbbell\_x : int -559 523 -502 -506 -576 -424 -543 -484 -515 -519 ...  
## $ magnet\_dumbbell\_y : int 293 -528 388 349 238 252 262 354 350 348 ...  
## $ magnet\_dumbbell\_z : num -65 -56 -36 41 53 312 96 97 53 -32 ...  
## $ roll\_forearm : num 28.4 141 109 131 0 -176 150 155 -161 15.5 ...  
## $ pitch\_forearm : num -63.9 49.3 -17.6 -32.6 0 -2.16 1.46 34.5 43.6 -63.5 ...  
## $ yaw\_forearm : num -153 156 106 93 0 -47.9 89.7 152 -89.5 -139 ...  
## $ total\_accel\_forearm : int 36 33 39 34 43 24 43 32 47 36 ...  
## $ gyros\_forearm\_x : num 0.03 0.74 1.12 0.18 1.38 -0.75 -0.88 -0.53 0.63 0.03 ...  
## $ gyros\_forearm\_y : num 0 -3.34 -2.78 -0.79 0.69 3.1 4.26 1.8 -0.74 0.02 ...  
## $ gyros\_forearm\_z : num -0.02 -0.59 -0.18 0.28 1.8 0.8 1.35 0.75 0.49 -0.02 ...  
## $ accel\_forearm\_x : int 192 -110 212 154 -92 131 230 -192 -151 195 ...  
## $ accel\_forearm\_y : int 203 267 297 271 406 -93 322 170 -331 204 ...  
## $ accel\_forearm\_z : int -215 -149 -118 -129 -39 172 -144 -175 -282 -217 ...  
## $ magnet\_forearm\_x : int -17 -714 -237 -51 -233 375 -300 -678 -109 0 ...  
## $ magnet\_forearm\_y : num 654 419 791 698 783 -787 800 284 -619 652 ...  
## $ magnet\_forearm\_z : num 476 617 873 783 521 91 884 585 -32 469 ...

dim(training1); dim(testing1); dim(newTesting1)

## [1] 19622 58

## [1] 20 57

## [1] 21 57

newTesting1 <- newTesting1[-1, ]

### Predict the testing data

predFinal1 <- predict(mode1, newTesting1)  
predFinal1

## 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21   
## 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

### Try the second model by linear discrination analysis algorithm

library(MASS)  
mode2 <- train(classe ~ ., method="lda", data=training1a)  
pred2 <- predict(mode2, training1b)  
confusionMatrix(training1b$classe, pred2)

## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction A B C D E  
## A 1527 129 18 0 0  
## B 135 824 170 10 0  
## C 1 125 877 22 1  
## D 0 0 115 801 48  
## E 0 0 10 110 962  
##   
## Overall Statistics  
##   
## Accuracy : 0.8481   
## 95% CI : (0.8387, 0.8572)  
## No Information Rate : 0.2826   
## P-Value [Acc > NIR] : < 2.2e-16   
##   
## Kappa : 0.808   
## Mcnemar's Test P-Value : NA   
##   
## Statistics by Class:  
##   
## Class: A Class: B Class: C Class: D Class: E  
## Sensitivity 0.9182 0.7644 0.7370 0.8494 0.9515  
## Specificity 0.9652 0.9345 0.9683 0.9670 0.9754  
## Pos Pred Value 0.9122 0.7234 0.8548 0.8309 0.8891  
## Neg Pred Value 0.9677 0.9465 0.9356 0.9711 0.9898  
## Prevalence 0.2826 0.1832 0.2022 0.1602 0.1718  
## Detection Rate 0.2595 0.1400 0.1490 0.1361 0.1635  
## Detection Prevalence 0.2845 0.1935 0.1743 0.1638 0.1839  
## Balanced Accuracy 0.9417 0.8494 0.8526 0.9082 0.9635

predFinal2 <- predict(mode2, newTesting1)  
predFinal2

## [1] B B B A A E D C A A B C B A E E A B B B  
## Levels: A B C D E

#### The model 2 give accuracy 86.2%, indicating ~14% expecting out of sample error.

### Find how agree each other of the two models

table(predFinal1, predFinal2)

## predFinal2  
## predFinal1 A B C D E  
## A 6 1 0 0 0  
## B 0 7 1 0 0  
## C 0 0 1 0 0  
## D 0 0 0 1 0  
## E 0 0 0 0 3

### Write each answer for each sample in the testing dataset using mode1 for submission

predFinal1 <- as.character(predFinal1)  
str(predFinal1)

## chr [1:20] "B" "A" "B" "A" "A" "E" "D" "B" "A" ...

pml\_write\_files = function(x){  
 n = length(x)  
 for(i in 1:n){  
 filename = paste0("problem\_id\_",i,".txt")  
 write.table(x[i],file=filename,quote=FALSE,row.names=FALSE,col.names=FALSE)  
 }  
}  
dir.create("results")

## Warning in dir.create("results"): 'results' already exists

setwd("./results")  
pml\_write\_files(predFinal1)  
dir()

## [1] "problem\_id\_1.txt" "problem\_id\_10.txt" "problem\_id\_11.txt"  
## [4] "problem\_id\_12.txt" "problem\_id\_13.txt" "problem\_id\_14.txt"  
## [7] "problem\_id\_15.txt" "problem\_id\_16.txt" "problem\_id\_17.txt"  
## [10] "problem\_id\_18.txt" "problem\_id\_19.txt" "problem\_id\_2.txt"   
## [13] "problem\_id\_20.txt" "problem\_id\_3.txt" "problem\_id\_4.txt"   
## [16] "problem\_id\_5.txt" "problem\_id\_6.txt" "problem\_id\_7.txt"   
## [19] "problem\_id\_8.txt" "problem\_id\_9.txt"