

## Summary

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, the goal is to use data from accelerometers on the belt, forearm, arm, and dumbbell of 6 participants. They were asked to perform barbell lifts correctly and incorrectly in 5 different ways: \* Exactly according to the specification (Class A) \* Throwing the elbows to the front (Class B) \* Lifting the dumbbell only halfway (Class C) \* Lowering the dumbbell only halfway (Class D) \* Throwing the hips to the front (Class E)

Class A corresponds to the specified execution of the exercise, while the other 4 classes correspond to common mistakes.

In this project, the goal is to use data from accelerometers on the **belt, forearm, arm, and dumbbell** of 6 participants and predict the class type for the given testing data. In general this would help people to quantify how well they do their exercise.

## Loading Data

- Loading the training, test data:

```
setwd('C:/Shahrzad_Docs/PERSONAL_DOCUMENTS/COURSERA/Machine_learning')
training_dat<- read.csv(file="pml-training.csv", header=TRUE, sep=',', na.strings = c("NA", "#DIV/0!"))

testing_dat <- read.csv(file="pml-testing.csv", header=TRUE, sep=',', na.strings = c("NA", "#DIV/0!"))
str(training_dat)
```

```
## '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 ...
## $ raw_timestamp_part_2 : int 788290 808298 820366 120339 196328 304277 368296 440390 484323 484...
## $ 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 1.43 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 : num NA NA NA NA NA NA NA NA NA NA ...
## $ kurtosis_pitch_belt : num NA NA NA NA NA NA NA NA NA NA ...
## $ kurtosis_yaw_belt : logi NA NA NA NA NA NA NA ...
## $ skewness_roll_belt : num NA NA NA NA NA NA NA NA NA NA ...
## $ skewness_roll_belt.1 : num NA NA NA NA NA NA NA NA NA NA ...
## $ skewness_yaw_belt : logi NA NA NA NA NA NA NA ...
## $ max_roll_belt : num NA NA NA NA NA NA NA NA NA NA ...
## $ max_pitch_belt : int NA NA NA NA NA NA NA NA NA NA ...
## $ max_yaw_belt : num NA NA NA NA NA NA NA NA NA NA ...
## $ 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 NA ...
## $ min_yaw_belt        : num  NA NA NA NA NA NA NA NA NA NA NA ...
## $ amplitude_roll_belt : num  NA NA NA NA NA NA NA NA NA NA NA ...
## $ amplitude_pitch_belt : int  NA NA NA NA NA NA NA NA NA NA NA ...
## $ amplitude_yaw_belt   : num  NA NA NA NA NA NA NA NA NA NA NA ...
## $ var_total_accel_belt : num  NA NA NA NA NA NA NA NA NA NA NA ...
## $ avg_roll_belt       : num  NA NA NA NA NA NA NA NA NA NA NA ...
## $ stddev_roll_belt    : num  NA NA NA NA NA NA NA NA NA NA NA ...
## $ var_roll_belt       : num  NA NA NA NA NA NA NA NA NA NA NA ...
## $ avg_pitch_belt      : num  NA NA NA NA NA NA NA NA NA NA NA ...
## $ stddev_pitch_belt   : num  NA NA NA NA NA NA NA NA NA NA NA ...
## $ var_pitch_belt      : num  NA NA NA NA NA NA NA NA NA NA NA ...
## $ avg_yaw_belt        : num  NA NA NA NA NA NA NA NA NA NA NA ...
## $ stddev_yaw_belt     : num  NA NA NA NA NA NA NA NA NA NA NA ...
## $ var_yaw_belt        : num  NA 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 NA ...
## $ avg_roll_arm        : num  NA NA NA NA NA NA NA NA NA NA NA ...
## $ stddev_roll_arm     : num  NA NA NA NA NA NA NA NA NA NA NA ...
## $ var_roll_arm        : num  NA NA NA NA NA NA NA NA NA NA NA ...
## $ avg_pitch_arm       : num  NA NA NA NA NA NA NA NA NA NA NA ...
## $ stddev_pitch_arm    : num  NA NA NA NA NA NA NA NA NA NA NA ...
## $ var_pitch_arm       : num  NA NA NA NA NA NA NA NA NA NA NA ...
## $ avg_yaw_arm         : num  NA NA NA NA NA NA NA NA NA NA NA ...
## $ stddev_yaw_arm      : num  NA NA NA NA NA NA NA NA NA NA NA ...
## $ var_yaw_arm         : num  NA 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   : num  NA NA NA NA NA NA NA NA NA NA NA ...
## $ kurtosis_pitch_arm  : num  NA NA NA NA NA NA NA NA NA NA NA ...
## $ kurtosis_yaw_arm    : num  NA NA NA NA NA NA NA NA NA NA NA ...
## $ skewness_roll_arm   : num  NA NA NA NA NA NA NA NA NA NA NA ...
## $ skewness_pitch_arm  : num  NA NA NA NA NA NA NA NA NA NA NA ...
## $ skewness_yaw_arm    : num  NA NA NA NA NA NA NA NA NA NA NA ...
## $ max_roll_arm        : num  NA NA NA NA NA NA NA NA NA NA NA ...

```

```
## $ max_picth_arm      : num  NA NA NA NA NA NA NA NA NA NA NA ...
## $ max_yaw_arm        : int   NA NA NA NA NA NA NA NA NA NA NA ...
## $ min_roll_arm       : num  NA NA NA NA NA NA NA NA NA NA NA ...
## $ min_pitch_arm      : num  NA NA NA NA NA NA NA NA NA NA NA ...
## $ min_yaw_arm        : int   NA NA NA NA NA NA NA NA NA NA NA ...
## $ amplitude_roll_arm : num  NA NA NA NA NA NA NA NA NA NA NA ...
## $ amplitude_pitch_arm : num  NA NA NA NA NA NA NA NA NA NA NA ...
## $ amplitude_yaw_arm   : int   NA 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 : num  NA NA NA NA NA NA NA NA NA NA NA ...
## $ kurtosis_picth_dumbbell : num  NA NA NA NA NA NA NA NA NA NA NA ...
## $ kurtosis_yaw_dumbbell : logi  NA NA NA NA NA NA ...
## $ skewness_roll_dumbbell : num  NA NA NA NA NA NA NA NA NA NA NA ...
## $ skewness_pitch_dumbbell : num  NA NA NA NA NA NA NA NA NA NA NA ...
## $ skewness_yaw_dumbbell : logi  NA NA NA NA NA NA ...
## $ max_roll_dumbbell   : num  NA NA NA NA NA NA NA NA NA NA NA ...
## $ max_picth_dumbbell  : num  NA NA NA NA NA NA NA NA NA NA NA ...
## $ max_yaw_dumbbell    : num  NA NA NA NA NA NA NA NA NA NA NA ...
## $ min_roll_dumbbell   : num  NA NA NA NA NA NA NA NA NA NA NA ...
## $ min_pitch_dumbbell  : num  NA NA NA NA NA NA NA NA NA NA NA ...
## $ min_yaw_dumbbell    : num  NA NA NA NA NA NA NA NA NA NA NA ...
## $ amplitude_roll_dumbbell : num  NA NA NA NA NA NA NA NA NA NA NA ...
## [list output truncated]
```

## Data Cleansing

Keeping only the columns related to: belt, forearm, arm, and dumbbell so removing the first 7 columns.

```
training_dat1<-training_dat[,c(8:160)]
str(training_dat1)
```

```
## 'data.frame':    19622 obs. of  153 variables:
## $ 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  : num  NA NA NA NA NA NA NA NA NA NA NA ...
## $ kurtosis_picth_belt : num  NA NA NA NA NA NA NA NA NA NA NA ...
## $ kurtosis_yaw_belt   : logi  NA NA NA NA NA NA ...
## $ skewness_roll_belt  : num  NA NA NA NA NA NA NA NA NA NA NA ...
## $ skewness_roll_belt.1 : num  NA NA NA NA NA NA NA NA NA NA NA ...
## $ skewness_yaw_belt   : logi  NA NA NA NA NA NA ...
## $ max_roll_belt       : num  NA NA NA NA NA NA NA NA NA NA NA ...
## $ max_picth_belt      : int   NA NA NA NA NA NA NA NA NA NA NA ...
## $ max_yaw_belt        : num  NA NA NA NA NA NA NA NA NA NA NA ...
## $ min_roll_belt       : num  NA NA NA NA NA NA NA NA NA NA NA ...
## $ min_pitch_belt      : int   NA NA NA NA NA NA NA NA NA NA NA ...
## $ min_yaw_belt        : num  NA NA NA NA NA NA NA NA NA NA NA ...
## $ amplitude_roll_belt : num  NA NA NA NA NA NA NA NA NA NA NA ...
## $ amplitude_pitch_belt : int   NA NA NA NA NA NA NA NA NA NA NA ...
```

```

## $ amplitude_yaw_belt      : num  NA NA NA NA NA NA NA NA NA NA NA ...
## $ var_total_accel_belt    : num  NA NA NA NA NA NA NA NA NA NA NA ...
## $ avg_roll_belt           : num  NA NA NA NA NA NA NA NA NA NA NA ...
## $ stddev_roll_belt        : num  NA NA NA NA NA NA NA NA NA NA NA ...
## $ var_roll_belt           : num  NA NA NA NA NA NA NA NA NA NA NA ...
## $ avg_pitch_belt          : num  NA NA NA NA NA NA NA NA NA NA NA ...
## $ stddev_pitch_belt       : num  NA NA NA NA NA NA NA NA NA NA NA ...
## $ var_pitch_belt          : num  NA NA NA NA NA NA NA NA NA NA NA ...
## $ avg_yaw_belt            : num  NA NA NA NA NA NA NA NA NA NA NA ...
## $ stddev_yaw_belt         : num  NA NA NA NA NA NA NA NA NA NA NA ...
## $ var_yaw_belt            : num  NA 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 0.02 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 NA ...
## $ avg_roll_arm             : num  NA NA NA NA NA NA NA NA NA NA NA ...
## $ stddev_roll_arm          : num  NA NA NA NA NA NA NA NA NA NA NA ...
## $ var_roll_arm             : num  NA NA NA NA NA NA NA NA NA NA NA ...
## $ avg_pitch_arm            : num  NA NA NA NA NA NA NA NA NA NA NA ...
## $ stddev_pitch_arm         : num  NA NA NA NA NA NA NA NA NA NA NA ...
## $ var_pitch_arm            : num  NA NA NA NA NA NA NA NA NA NA NA ...
## $ avg_yaw_arm              : num  NA NA NA NA NA NA NA NA NA NA NA ...
## $ stddev_yaw_arm           : num  NA NA NA NA NA NA NA NA NA NA NA ...
## $ var_yaw_arm              : num  NA 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        : num  NA NA NA NA NA NA NA NA NA NA NA ...
## $ kurtosis_pitch_arm       : num  NA NA NA NA NA NA NA NA NA NA NA ...
## $ kurtosis_yaw_arm         : num  NA NA NA NA NA NA NA NA NA NA NA ...
## $ skewness_roll_arm        : num  NA NA NA NA NA NA NA NA NA NA NA ...
## $ skewness_pitch_arm       : num  NA NA NA NA NA NA NA NA NA NA NA ...
## $ skewness_yaw_arm         : num  NA NA NA NA NA NA NA NA NA NA NA ...
## $ max_roll_arm             : num  NA NA NA NA NA NA NA NA NA NA NA ...
## $ max_pitch_arm            : num  NA NA NA NA NA NA NA NA NA NA NA ...
## $ max_yaw_arm              : int   NA NA NA NA NA NA NA NA NA NA NA ...
## $ min_roll_arm             : num  NA NA NA NA NA NA NA NA NA NA NA ...
## $ min_pitch_arm            : num  NA NA NA NA NA NA NA NA NA NA NA ...

```

```
## $ min_yaw_arm          : int  NA NA NA NA NA NA NA NA NA NA NA ...
## $ amplitude_roll_arm   : num  NA NA NA NA NA NA NA NA NA NA NA ...
## $ amplitude_pitch_arm  : num  NA NA NA NA NA NA NA NA NA NA NA ...
## $ amplitude_yaw_arm    : int  NA 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 : num  NA NA NA NA NA NA NA NA NA NA NA ...
## $ kurtosis_pitch_dumbbell : num  NA NA NA NA NA NA NA NA NA NA NA ...
## $ kurtosis_yaw_dumbbell  : logi  NA NA NA NA NA NA ...
## $ skewness_roll_dumbbell : num  NA NA NA NA NA NA NA NA NA NA NA ...
## $ skewness_pitch_dumbbell : num  NA NA NA NA NA NA NA NA NA NA NA ...
## $ skewness_yaw_dumbbell  : logi  NA NA NA NA NA NA ...
## $ max_roll_dumbbell     : num  NA NA NA NA NA NA NA NA NA NA NA ...
## $ max_pitch_dumbbell    : num  NA NA NA NA NA NA NA NA NA NA NA ...
## $ max_yaw_dumbbell      : num  NA NA NA NA NA NA NA NA NA NA NA ...
## $ min_roll_dumbbell     : num  NA NA NA NA NA NA NA NA NA NA NA ...
## $ min_pitch_dumbbell    : num  NA NA NA NA NA NA NA NA NA NA NA ...
## $ min_yaw_dumbbell      : num  NA NA NA NA NA NA NA NA NA NA NA ...
## $ amplitude_roll_dumbbell : num  NA NA NA NA NA NA NA NA NA NA NA ...
## $ amplitude_pitch_dumbbell : num  NA NA NA NA NA NA NA NA NA NA NA ...
## $ amplitude_yaw_dumbbell : num  NA NA NA NA NA NA NA NA NA NA NA ...
## $ total_accel_dumbbell  : int   37 37 37 37 37 37 37 37 37 37 ...
## $ var_accel_dumbbell    : num  NA NA NA NA NA NA NA NA NA NA NA ...
## $ avg_roll_dumbbell     : num  NA NA NA NA NA NA NA NA NA NA NA ...
## $ stddev_roll_dumbbell  : num  NA NA NA NA NA NA NA NA NA NA NA ...
## $ var_roll_dumbbell     : num  NA NA NA NA NA NA NA NA NA NA NA ...
## [list output truncated]
```

```
testing_dat1<-testing_dat[,c(8:160)]
```

## Removing the columns that have more than 90% NA.

```
training_dat2 <- training_dat1[ lapply( training_dat1, function(x) sum(is.na(x)) / length(x) ) < 0.1 ]
str(training_dat2)
```

```
## 'data.frame':    19622 obs. of  53 variables:
## $ 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 0.02 0.03 ...
## $ 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 ...
```

```
testing_dat2 <- testing_dat1[ lapply( testing_dat1, function(x) sum(is.na(x)) / length(x) ) < 0.1 ]
```

## Training and Cross Validation sets

We allocate 75% in training set and 25% in cross validation set

```
set.seed(12345)
library(caret)
```

```
## Loading required package: lattice
```

```
## Loading required package: ggplot2
```

```
intrain <-createDataPartition(training_dat2$classe, p=0.75,list=FALSE)
training = training_dat2 [intrain,]
crossvalidation = training_dat2 [-intrain,]
```

## Prediction using Random Forests Model

Here we train the model using random forest function:

```
library(randomForest)
```

```
## randomForest 4.6-12
```

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

```
##
```

```
## Attaching package: 'randomForest'
```

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

```
##
```

```
##      margin
```

```
ModelFit = randomForest(classe~., data= training)
prediction <- predict(ModelFit, crossvalidation, type = "class")
confusionMatrix(prediction, crossvalidation $classe)
```

```
## Confusion Matrix and Statistics
```

```
##
```

```
##           Reference
```

```
## Prediction    A    B    C    D    E
```

```
##           A 1395     8     0     0     0
```

```
##           B     0   937     2     0     0
```

```
##           C     0     4   853     3     1
```

```
##           D     0     0     0   801     0
```

```
##           E     0     0     0     0   900
```

```
##
```

```
## Overall Statistics
```

```
##
```

```
##           Accuracy : 0.9963
```

```
##           95% CI : (0.9942, 0.9978)
```

```
## No Information Rate : 0.2845
```

```
## P-Value [Acc > NIR] : < 2.2e-16
```

```
##
```

```
##           Kappa : 0.9954
```

```
## Mcnemar's Test P-Value : NA
```

```
##
```

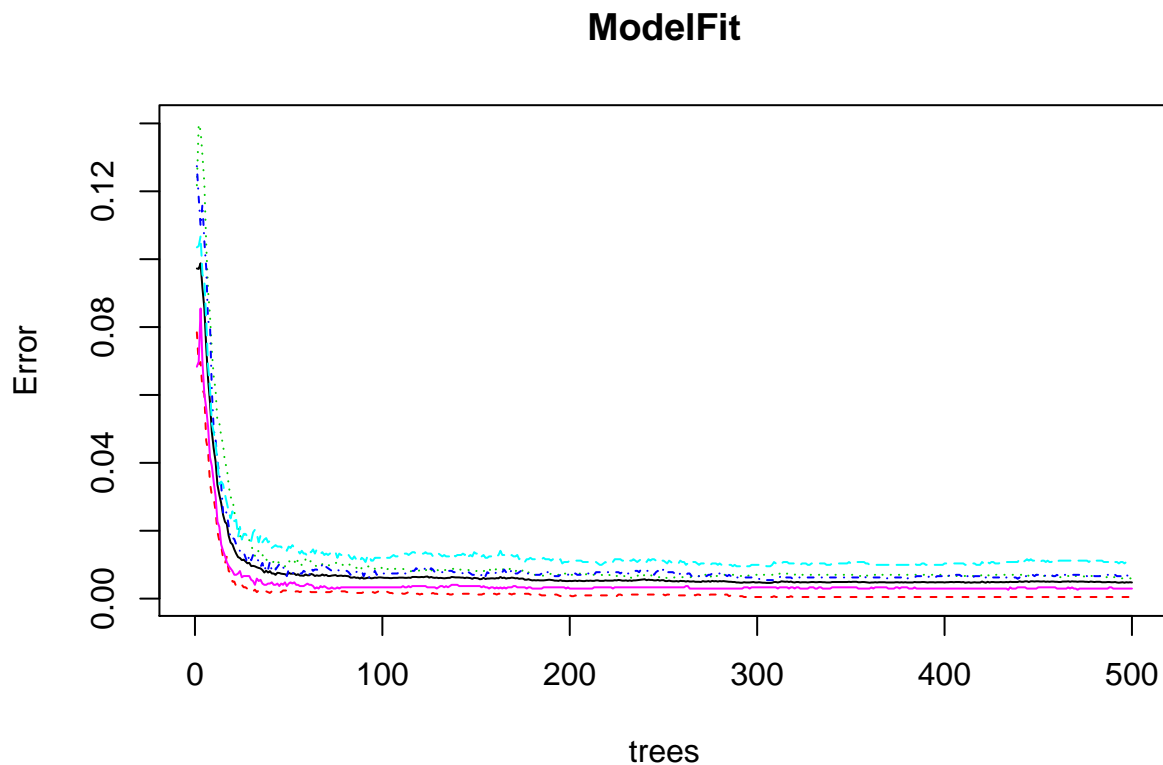
```
## Statistics by Class:
```

```
##
```

```
##           Class: A Class: B Class: C Class: D Class: E
```

## Sensitivity	1.0000	0.9874	0.9977	0.9963	0.9989
## Specificity	0.9977	0.9995	0.9980	1.0000	1.0000
## Pos Pred Value	0.9943	0.9979	0.9907	1.0000	1.0000
## Neg Pred Value	1.0000	0.9970	0.9995	0.9993	0.9998
## Prevalence	0.2845	0.1935	0.1743	0.1639	0.1837
## Detection Rate	0.2845	0.1911	0.1739	0.1633	0.1835
## Detection Prevalence	0.2861	0.1915	0.1756	0.1633	0.1835
## Balanced Accuracy	0.9989	0.9934	0.9978	0.9981	0.9994

This shows that predicting using random forest has 99.66% accuracy and the out-of-sample error is 100%-99.66%= 0.34%. Therefore this is a good Model.



The above graph also shows that for the default number of trees (500) the error is less than 1%.

**Applying the machine learning algorithm to the given test cases:**

```
prediction <- predict(ModelFit, testing_dat2, type = "class")
prediction
```

```
##  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
```