

Prediction Assignment Writeup

Author name: Jinendra Sontakke

Summary of 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 dumbbell 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>

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

Load library

```
library(data.table)
library(ggplot2)
library(caret)
```

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

```
library(dplyr)
```

```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:data.table':
##
##   between, first, last
```

```
## The following objects are masked from 'package:stats':
##
##   filter, lag
```

```
## The following objects are masked from 'package:base':  
##  
## intersect, setdiff, setequal, union
```

```
library(rpart)  
library(gbm)
```

```
## Loaded gbm 2.1.5
```

```
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:dplyr':  
##  
## combine
```

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

Load data

```
data_train <- read.csv("D:/Users/jinu/Documents/pml-training.csv")  
data_test <- read.csv("D:/Users/jinu/Documents/pml-training.csv")  
dim(data_train)
```

```
## [1] 19622 160
```

```
dim(data_test)
```

```
## [1] 20 160
```

```
summary(data_train$classe)
```

```
##      A      B      C      D      E
## 5580 3797 3422 3216 3607
```

Dataset partition for training data

```
data_train_part <- createDataPartition(data_train$classe, p=0.7, list=FALSE)
data_train_train <- data_train[data_train_part, ]
data_train_test  <- data_train[-data_train_part, ]
dim(data_train_train)
```

```
## [1] 13737    160
```

```
dim(data_train_test)
```

```
## [1] 5885    160
```

Remove useless variable for identification

```
data_train_train <- data_train_train[, -c(1:5)]
data_train_test  <- data_train_test[, -c(1:5)]
data_test        <- data_test[, -c(1:5)]
```

Near zero variance elimination

```
col_nzv <- nearZeroVar(data_train_train)
data_train_train <- data_train_train[, -col_nzv]
data_train_test  <- data_train_test[, -col_nzv]
data_test        <- data_test[, -col_nzv]
dim(data_train_train)
```

```
## [1] 13737    100
```

```
dim(data_train_test)
```

```
## [1] 5885    100
```

```
dim(data_test)
```

```
## [1] 20 100
```

Remove NA variable (include almost NA)

```
col_na <- sapply(data_train_train, function(x) mean(is.na(x))) >0.9
data_train_train <- data_train_train[, col_na==FALSE]
data_train_test <- data_train_test[, col_na==FALSE]
data_test <- data_test[, col_na==FALSE]
dim(data_train_train)
```

```
## [1] 13737 54
```

```
dim(data_train_test)
```

```
## [1] 5885 54
```

```
dim(data_test)
```

```
## [1] 20 54
```

```
summary(data_train_train)
```

```
##      num_window      roll_belt      pitch_belt      yaw_belt
## Min.   : 1.0      Min.   : -28.80      Min.   : -55.8000      Min.   : -180.00
## 1st Qu.:223.0      1st Qu.: 1.09      1st Qu.: 1.7900      1st Qu.: -88.30
## Median :425.0      Median :113.00      Median : 5.2800      Median : -14.30
## Mean   :430.9      Mean   : 64.01      Mean   : 0.3697      Mean   : -11.75
## 3rd Qu.:645.0      3rd Qu.:123.00      3rd Qu.:14.9000      3rd Qu.: 8.66
## Max.   :864.0      Max.   :162.00      Max.   : 60.3000      Max.   :179.00
## total_accel_belt  gyros_belt_x      gyros_belt_y      gyros_belt_z
## Min.   : 0.00      Min.   : -1.040000      Min.   : -0.53000      Min.   : -1.330
## 1st Qu.: 3.00      1st Qu.: -0.030000      1st Qu.: 0.00000      1st Qu.: -0.200
## Median :17.00      Median : 0.030000      Median : 0.02000      Median : -0.100
## Mean   :11.27      Mean   : -0.005739      Mean   : 0.03873      Mean   : -0.132
## 3rd Qu.:18.00      3rd Qu.: 0.110000      3rd Qu.: 0.11000      3rd Qu.: -0.020
## Max.   :28.00      Max.   : 2.220000      Max.   : 0.64000      Max.   : 1.610
## accel_belt_x      accel_belt_y      accel_belt_z      magnet_belt_x
## Min.   : -120.000      Min.   : -65.00      Min.   : -269.00      Min.   : -52.00
## 1st Qu.: -21.000      1st Qu.: 3.00      1st Qu.: -162.00      1st Qu.: 9.00
## Median : -15.000      Median : 32.00      Median : -150.00      Median : 35.00
## Mean   : -5.719      Mean   : 29.98      Mean   : -71.87      Mean   : 55.22
## 3rd Qu.: -5.000      3rd Qu.: 61.00      3rd Qu.: 28.00      3rd Qu.: 59.00
## Max.   : 79.000      Max.   :164.00      Max.   :103.00      Max.   :485.00
## magnet_belt_y      magnet_belt_z      roll_arm      pitch_arm
## Min.   :354.0      Min.   : -621      Min.   : -180.00      Min.   : -88.800
```

```

## 1st Qu.:582.0 1st Qu.: -374 1st Qu.: -30.90 1st Qu.: -25.800
## Median :601.0 Median : -319 Median : 0.00 Median : 0.000
## Mean :594.1 Mean : -345 Mean : 18.21 Mean : -4.726
## 3rd Qu.:610.0 3rd Qu.: -306 3rd Qu.: 77.40 3rd Qu.: 11.100
## Max. :669.0 Max. : 289 Max. : 180.00 Max. : 88.200
## yaw_arm total_accel_arm gyros_arm_x gyros_arm_y
## Min. : -180.0000 Min. : 1.00 Min. : -6.37000 Min. : -3.4400
## 1st Qu.: -42.2000 1st Qu.:17.00 1st Qu.: -1.33000 1st Qu.: -0.8000
## Median : 0.0000 Median :27.00 Median : 0.08000 Median : -0.2400
## Mean : -0.4126 Mean :25.48 Mean : 0.05367 Mean : -0.2596
## 3rd Qu.: 45.8000 3rd Qu.:33.00 3rd Qu.: 1.59000 3rd Qu.: 0.1600
## Max. : 180.0000 Max. :66.00 Max. : 4.87000 Max. : 2.8400
## gyros_arm_z accel_arm_x accel_arm_y accel_arm_z
## Min. : -2.3300 Min. : -383.00 Min. : -318.0 Min. : -636.00
## 1st Qu.: -0.0800 1st Qu.: -241.00 1st Qu.: -53.0 1st Qu.: -144.00
## Median : 0.2300 Median : -44.00 Median : 15.0 Median : -46.00
## Mean : 0.2664 Mean : -60.33 Mean : 33.5 Mean : -70.85
## 3rd Qu.: 0.7200 3rd Qu.: 83.00 3rd Qu.: 141.0 3rd Qu.: 23.00
## Max. : 2.9500 Max. : 435.00 Max. : 303.0 Max. : 292.00
## magnet_arm_x magnet_arm_y magnet_arm_z roll_dumbbell
## Min. : -584.0 Min. : -377.0 Min. : -595.0 Min. : -153.71
## 1st Qu.: -297.0 1st Qu.: -7.0 1st Qu.: 137.0 1st Qu.: -17.59
## Median : 291.0 Median : 202.0 Median : 444.0 Median : 48.28
## Mean : 192.7 Mean : 157.4 Mean : 308.1 Mean : 24.24
## 3rd Qu.: 636.0 3rd Qu.: 323.0 3rd Qu.: 545.0 3rd Qu.: 67.99
## Max. : 782.0 Max. : 583.0 Max. : 694.0 Max. : 153.55
## pitch_dumbbell yaw_dumbbell total_accel_dumbbell
## Min. : -148.50 Min. : -150.871 Min. : 0.00
## 1st Qu.: -40.94 1st Qu.: -77.579 1st Qu.: 4.00
## Median : -21.02 Median : -3.191 Median :11.00
## Mean : -10.77 Mean : 1.609 Mean :13.79
## 3rd Qu.: 17.58 3rd Qu.: 79.385 3rd Qu.:20.00
## Max. : 149.40 Max. : 154.952 Max. :42.00
## gyros_dumbbell_x gyros_dumbbell_y gyros_dumbbell_z accel_dumbbell_x
## Min. : -1.8500 Min. : -2.10000 Min. : -2.3800 Min. : -237.00
## 1st Qu.: -0.0300 1st Qu.: -0.14000 1st Qu.: -0.3100 1st Qu.: -51.00
## Median : 0.1400 Median : 0.03000 Median : -0.1300 Median : -8.00
## Mean : 0.1725 Mean : 0.04269 Mean : -0.1446 Mean : -28.71
## 3rd Qu.: 0.3500 3rd Qu.: 0.21000 3rd Qu.: 0.0300 3rd Qu.: 11.00
## Max. : 2.2000 Max. : 4.37000 Max. : 1.8700 Max. : 235.00
## accel_dumbbell_y accel_dumbbell_z magnet_dumbbell_x magnet_dumbbell_y
## Min. : -182.00 Min. : -334.00 Min. : -639.0 Min. : -3600
## 1st Qu.: -8.00 1st Qu.: -142.00 1st Qu.: -535.0 1st Qu.: 233
## Median : 43.00 Median : -1.00 Median : -479.0 Median : 311
## Mean : 53.37 Mean : -38.73 Mean : -327.4 Mean : 222
## 3rd Qu.: 114.00 3rd Qu.: 38.00 3rd Qu.: -297.0 3rd Qu.: 391
## Max. : 302.00 Max. : 318.00 Max. : 592.0 Max. : 633
## magnet_dumbbell_z roll_forearm pitch_forearm yaw_forearm
## Min. : -250.00 Min. : -180.00 Min. : -72.50 Min. : -180.00
## 1st Qu.: -45.00 1st Qu.: -0.68 1st Qu.: 0.00 1st Qu.: -68.10

```

```
## Median : 14.00 Median : 22.60 Median : 9.23 Median : 0.00
## Mean : 46.15 Mean : 34.39 Mean : 10.69 Mean : 20.24
## 3rd Qu.: 96.00 3rd Qu.: 140.00 3rd Qu.: 28.50 3rd Qu.: 110.00
## Max. : 451.00 Max. : 180.00 Max. : 89.80 Max. : 180.00
## total_accel_forearm gyros_forearm_x gyros_forearm_y
## Min. : 0.00 Min. : -2.9900 Min. : -6.65000
## 1st Qu.:29.00 1st Qu.: -0.2200 1st Qu.: -1.46000
## Median :36.00 Median : 0.0500 Median : 0.03000
## Mean :34.71 Mean : 0.1542 Mean : 0.06953
## 3rd Qu.:41.00 3rd Qu.: 0.5600 3rd Qu.: 1.62000
## Max. :78.00 Max. : 3.9700 Max. : 6.13000
## gyros_forearm_z accel_forearm_x accel_forearm_y accel_forearm_z
## Min. : -8.0900 Min. : -498.00 Min. : -632.0 Min. : -410.00
## 1st Qu.: -0.1800 1st Qu.: -178.00 1st Qu.: 53.0 1st Qu.: -182.00
## Median : 0.0800 Median : -57.00 Median : 199.0 Median : -43.00
## Mean : 0.1401 Mean : -61.71 Mean : 161.9 Mean : -56.58
## 3rd Qu.: 0.4900 3rd Qu.: 77.00 3rd Qu.: 311.0 3rd Qu.: 26.00
## Max. : 4.3100 Max. : 389.00 Max. : 591.0 Max. : 275.00
## magnet_forearm_x magnet_forearm_y magnet_forearm_z classe
## Min. : -1280.0 Min. : -892 Min. : -966.0 A:3906
## 1st Qu.: -615.0 1st Qu.: -3 1st Qu.: 192.0 B:2658
## Median : -377.0 Median : 583 Median : 513.0 C:2396
## Mean : -311.8 Mean : 375 Mean : 395.8 D:2252
## 3rd Qu.: -74.0 3rd Qu.: 736 3rd Qu.: 653.0 E:2525
## Max. : 672.0 Max. : 1480 Max. : 1090.0
```

```
summary(data_test)
```

```
## num_window roll_belt pitch_belt yaw_belt
## Min. : 48.0 Min. : -5.9200 Min. : -41.600 Min. : -93.70
## 1st Qu.:250.0 1st Qu.: 0.9075 1st Qu.: 3.013 1st Qu.: -88.62
## Median :384.5 Median : 1.1100 Median : 4.655 Median : -87.85
## Mean :379.6 Mean : 31.3055 Mean : 5.824 Mean : -59.30
## 3rd Qu.:467.0 3rd Qu.: 32.5050 3rd Qu.: 6.135 3rd Qu.: -63.50
## Max. :859.0 Max. :129.0000 Max. : 27.800 Max. :162.00
## total_accel_belt gyros_belt_x gyros_belt_y gyros_belt_z
## Min. : 2.00 Min. : -0.500 Min. : -0.050 Min. : -0.4800
## 1st Qu.: 3.00 1st Qu.: -0.070 1st Qu.: -0.005 1st Qu.: -0.1375
## Median : 4.00 Median : 0.020 Median : 0.000 Median : -0.0250
## Mean : 7.55 Mean : -0.045 Mean : 0.010 Mean : -0.1005
## 3rd Qu.: 8.00 3rd Qu.: 0.070 3rd Qu.: 0.020 3rd Qu.: 0.0000
## Max. :21.00 Max. : 0.240 Max. : 0.110 Max. : 0.0500
## accel_belt_x accel_belt_y accel_belt_z magnet_belt_x
## Min. : -48.00 Min. : -16.00 Min. : -187.00 Min. : -13.00
## 1st Qu.: -19.00 1st Qu.: 2.00 1st Qu.: -24.00 1st Qu.: 5.50
## Median : -13.00 Median : 4.50 Median : 27.00 Median : 33.50
## Mean : -13.50 Mean : 18.35 Mean : -17.60 Mean : 35.15
## 3rd Qu.: -8.75 3rd Qu.: 25.50 3rd Qu.: 38.25 3rd Qu.: 46.25
## Max. : 46.00 Max. : 72.00 Max. : 49.00 Max. :169.00
```

```

## magnet_belt_y magnet_belt_z roll_arm pitch_arm
## Min. :566.0 Min. : -426.0 Min. : -137.00 Min. : -63.800
## 1st Qu.:578.5 1st Qu.: -398.5 1st Qu.: 0.00 1st Qu.: -9.188
## Median :600.5 Median : -313.5 Median : 0.00 Median : 0.000
## Mean :601.5 Mean : -346.9 Mean : 16.42 Mean : -3.950
## 3rd Qu.:631.2 3rd Qu.: -305.0 3rd Qu.: 71.53 3rd Qu.: 3.465
## Max. :638.0 Max. : -291.0 Max. : 152.00 Max. : 55.000
## yaw_arm total_accel_arm gyros_arm_x gyros_arm_y
## Min. : -167.00 Min. : 3.00 Min. : -3.710 Min. : -2.0900
## 1st Qu.: -60.15 1st Qu.:20.25 1st Qu.: -0.645 1st Qu.: -0.6350
## Median : 0.00 Median :29.50 Median : 0.020 Median : -0.0400
## Mean : -2.80 Mean :26.40 Mean : 0.077 Mean : -0.1595
## 3rd Qu.: 25.50 3rd Qu.:33.25 3rd Qu.: 1.248 3rd Qu.: 0.2175
## Max. : 178.00 Max. :44.00 Max. : 3.660 Max. : 1.8500
## gyros_arm_z accel_arm_x accel_arm_y accel_arm_z
## Min. : -0.6900 Min. : -341.0 Min. : -65.00 Min. : -404.00
## 1st Qu.: -0.1800 1st Qu.: -277.0 1st Qu.: 52.25 1st Qu.: -128.50
## Median : -0.0250 Median : -194.5 Median :112.00 Median : -83.50
## Mean : 0.1205 Mean : -134.6 Mean :103.10 Mean : -87.85
## 3rd Qu.: 0.5650 3rd Qu.: 5.5 3rd Qu.:168.25 3rd Qu.: -27.25
## Max. : 1.1300 Max. : 106.0 Max. :245.00 Max. : 93.00
## magnet_arm_x magnet_arm_y magnet_arm_z roll_dumbbell
## Min. : -428.00 Min. : -307.0 Min. : -499.0 Min. : -111.118
## 1st Qu.: -373.75 1st Qu.: 205.2 1st Qu.: 403.0 1st Qu.: 7.494
## Median : -265.00 Median : 291.0 Median : 476.5 Median : 50.403
## Mean : -38.95 Mean : 239.4 Mean : 369.8 Mean : 33.760
## 3rd Qu.: 250.50 3rd Qu.: 358.8 3rd Qu.: 517.0 3rd Qu.: 58.129
## Max. : 750.00 Max. : 474.0 Max. : 633.0 Max. : 123.984
## pitch_dumbbell yaw_dumbbell total_accel_dumbbell
## Min. : -54.97 Min. : -103.3200 Min. : 1.0
## 1st Qu.: -51.89 1st Qu.: -75.2809 1st Qu.: 7.0
## Median : -40.81 Median : -8.2863 Median :15.5
## Mean : -19.47 Mean : -0.9385 Mean :17.2
## 3rd Qu.: 16.12 3rd Qu.: 55.8335 3rd Qu.:29.0
## Max. : 96.87 Max. : 132.2337 Max. :31.0
## gyros_dumbbell_x gyros_dumbbell_y gyros_dumbbell_z accel_dumbbell_x
## Min. : -1.0300 Min. : -1.1100 Min. : -1.180 Min. : -159.00
## 1st Qu.: 0.1600 1st Qu.: -0.2100 1st Qu.: -0.485 1st Qu.: -140.25
## Median : 0.3600 Median : 0.0150 Median : -0.280 Median : -19.00
## Mean : 0.2690 Mean : 0.0605 Mean : -0.266 Mean : -47.60
## 3rd Qu.: 0.4625 3rd Qu.: 0.1450 3rd Qu.: -0.165 3rd Qu.: 15.75
## Max. : 1.0600 Max. : 1.9100 Max. : 1.100 Max. : 185.00
## accel_dumbbell_y accel_dumbbell_z magnet_dumbbell_x magnet_dumbbell_y
## Min. : -30.00 Min. : -221.0 Min. : -576.0 Min. : -558.0
## 1st Qu.: 5.75 1st Qu.: -192.2 1st Qu.: -528.0 1st Qu.: 259.5
## Median : 71.50 Median : -3.0 Median : -508.5 Median : 316.0
## Mean : 70.55 Mean : -60.0 Mean : -304.2 Mean : 189.3
## 3rd Qu.:151.25 3rd Qu.: 76.5 3rd Qu.: -317.0 3rd Qu.: 348.2
## Max. :166.00 Max. : 100.0 Max. : 523.0 Max. : 403.0
## magnet_dumbbell_z roll_forearm pitch_forearm yaw_forearm

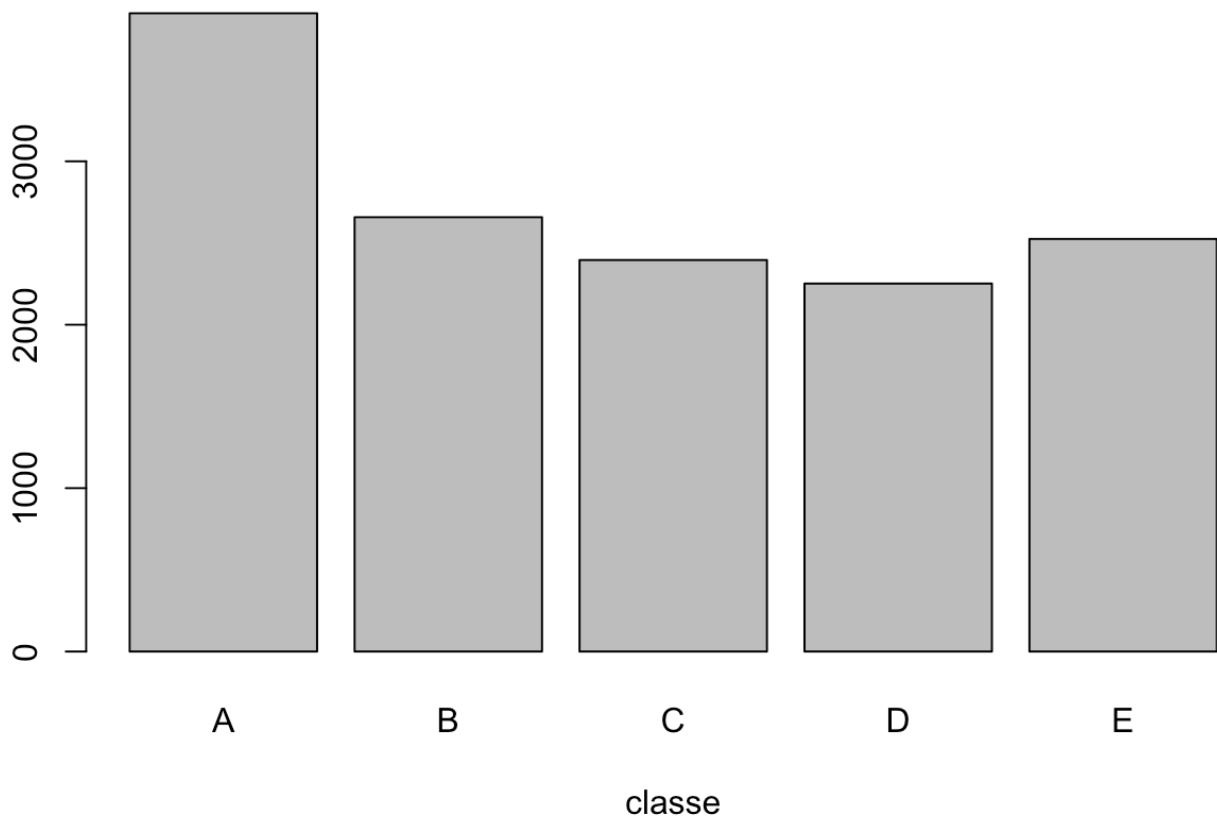
```

```
## Min.      :-164.00   Min.      :-176.00   Min.      :-63.500   Min.      :-168.000
## 1st Qu.: -33.00    1st Qu.: -40.25    1st Qu.: -11.457    1st Qu.: -93.375
## Median :  49.50    Median :  94.20    Median :   8.830    Median : -19.250
## Mean    :  71.40    Mean    :  38.66    Mean    :   7.099    Mean    :   2.195
## 3rd Qu.:  96.25    3rd Qu.: 143.25    3rd Qu.: 28.500    3rd Qu.: 104.500
## Max.     : 368.00    Max.     : 176.00    Max.     : 59.300    Max.     : 159.000
## total_accel_forearm gyros_forearm_x   gyros_forearm_y   gyros_forearm_z
## Min.      :21.00      Min.      :-1.0600   Min.      :-5.9700   Min.      :-1.2600
## 1st Qu.:24.00      1st Qu.: -0.5850   1st Qu.: -1.2875   1st Qu.: -0.0975
## Median :32.50      Median :  0.0200   Median :  0.0350   Median :  0.2300
## Mean    :32.05      Mean    : -0.0200   Mean    : -0.0415   Mean    :  0.2610
## 3rd Qu.:36.75      3rd Qu.:  0.2925   3rd Qu.:  2.0475   3rd Qu.:  0.7625
## Max.     :47.00      Max.     :  1.3800   Max.     :  4.2600   Max.     :  1.8000
## accel_forearm_x accel_forearm_y accel_forearm_z magnet_forearm_x
## Min.      :-212.0   Min.      :-331.0   Min.      :-282.0   Min.      :-714.0
## 1st Qu.: -114.8    1st Qu.:   8.5    1st Qu.: -199.0    1st Qu.: -427.2
## Median :  86.0     Median : 138.0    Median : -148.5    Median : -189.5
## Mean    :  38.8     Mean    : 125.3    Mean    : -93.7     Mean    : -159.2
## 3rd Qu.: 166.2     3rd Qu.: 268.0    3rd Qu.: -31.0     3rd Qu.:  41.5
## Max.     : 232.0     Max.     : 406.0    Max.     : 179.0     Max.     : 532.0
## magnet_forearm_y magnet_forearm_z   problem_id
## Min.      :-787.0   Min.      :-32.0    Min.      : 1.00
## 1st Qu.: -328.8    1st Qu.:275.2      1st Qu.: 5.75
## Median : 487.0     Median :491.5      Median :10.50
## Mean    : 191.8     Mean    :460.2      Mean    :10.50
## 3rd Qu.: 720.8     3rd Qu.:661.5      3rd Qu.:15.25
## Max.     : 800.0     Max.     :884.0      Max.     :20.00
```

##Frequency plot

```
plot(data_train_train$classe, main="Frequency of different levels", xlab="classe")
```


Frequency of different levels



Prediction models

1. Decision tree

```
# Fitting model
model_tree <- randomForest(classe ~., data=data_train_train, method="rpart")

# Predicting
predict_tree <- predict(model_tree, data_train_test, Type="rpart")

# Testing
confusionMatrix(predict_tree, data_train_test$classe)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction   A     B     C     D     E
##           A 1673     3     0     0     0
##           B   1 1136     2     0     0
##           C    0     0 1021     6     0
##           D    0     0   3  958     1
##           E    0     0   0   0 1081
##
## Overall Statistics
##
##           Accuracy : 0.9973
##           95% CI : (0.9956, 0.9984)
##           No Information Rate : 0.2845
##           P-Value [Acc > NIR] : < 2.2e-16
##
##           Kappa : 0.9966
##
## McNemar's Test P-Value : NA
##
## Statistics by Class:
##
##           Class: A Class: B Class: C Class: D Class: E
## Sensitivity      0.9994  0.9974  0.9951  0.9938  0.9991
## Specificity      0.9993  0.9994  0.9988  0.9992  1.0000
## Pos Pred Value   0.9982  0.9974  0.9942  0.9958  1.0000
## Neg Pred Value   0.9998  0.9994  0.9990  0.9988  0.9998
## Prevalence       0.2845  0.1935  0.1743  0.1638  0.1839
## Detection Rate   0.2843  0.1930  0.1735  0.1628  0.1837
## Detection Prevalence 0.2848  0.1935  0.1745  0.1635  0.1837
## Balanced Accuracy 0.9993  0.9984  0.9969  0.9965  0.9995
```

2. Boosting (gradient boosting)

```
# Fitting model
Control <- trainControl(method="repeatedcv", number=5, repeats=1)
model_rf <- randomForest(classe ~., data=data_train_train, method="gbm", trControl=
Control, verbose=FALSE)

# Predicting
predict_rf <- predict(model_rf, data_train_test, Type="gbm")

# Testing
confusionMatrix(predict_rf, data_train_test$classe)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction   A     B     C     D     E
##           A 1674     3     0     0     0
##           B    0 1136     2     0     0
##           C    0     0 1021     6     0
##           D    0     0    3  958     1
##           E    0     0    0    0 1081
##
## Overall Statistics
##
##           Accuracy : 0.9975
##           95% CI : (0.9958, 0.9986)
##           No Information Rate : 0.2845
##           P-Value [Acc > NIR] : < 2.2e-16
##
##           Kappa : 0.9968
##
## McNemar's Test P-Value : NA
##
## Statistics by Class:
##
##           Class: A Class: B Class: C Class: D Class: E
## Sensitivity       1.0000   0.9974   0.9951   0.9938   0.9991
## Specificity       0.9993   0.9996   0.9988   0.9992   1.0000
## Pos Pred Value    0.9982   0.9982   0.9942   0.9958   1.0000
## Neg Pred Value    1.0000   0.9994   0.9990   0.9988   0.9998
## Prevalence        0.2845   0.1935   0.1743   0.1638   0.1839
## Detection Rate    0.2845   0.1930   0.1735   0.1628   0.1837
## Detection Prevalence 0.2850   0.1934   0.1745   0.1635   0.1837
## Balanced Accuracy  0.9996   0.9985   0.9969   0.9965   0.9995
```

3. Random forest

```
# Fitting model
model_rf <- randomForest(classe ~., data=data_train_train, method="class")

# Predicting
predict_rf <- predict(model_rf, data_train_test, Type="class")

# Testing
confusionMatrix(predict_rf, data_train_test$classe)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction    A     B     C     D     E
##           A 1674     3     0     0     0
##           B    0 1136     1     0     0
##           C    0     0 1022     6     0
##           D    0     0     3  958     1
##           E    0     0     0     0 1081
##
## Overall Statistics
##
##           Accuracy : 0.9976
##           95% CI : (0.996, 0.9987)
##           No Information Rate : 0.2845
##           P-Value [Acc > NIR] : < 2.2e-16
##
##           Kappa : 0.997
##
## Mcnemar's Test P-Value : NA
##
## Statistics by Class:
##
##           Class: A Class: B Class: C Class: D Class: E
## Sensitivity      1.0000   0.9974   0.9961   0.9938   0.9991
## Specificity      0.9993   0.9998   0.9988   0.9992   1.0000
## Pos Pred Value   0.9982   0.9991   0.9942   0.9958   1.0000
## Neg Pred Value   1.0000   0.9994   0.9992   0.9988   0.9998
## Prevalence       0.2845   0.1935   0.1743   0.1638   0.1839
## Detection Rate   0.2845   0.1930   0.1737   0.1628   0.1837
## Detection Prevalence 0.2850   0.1932   0.1747   0.1635   0.1837
## Balanced Accuracy 0.9996   0.9986   0.9974   0.9965   0.9995
```

Apply to 20 test data

```
predict_test <- predict(model_rf, data_test)
predict_test
```

```
##  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
```