wk4Assignment.Rmd

#LOAD THE given data DATA. REINITIALIZE DIV/0 AS NA.

```
#load data
library(caret)
dataPml <- read.csv("pml-training.csv", na.strings=c("#DIV/0!"), row.names = 1)
testPml <- read.csv("pml-testing.csv", na.strings=c("#DIV/0!"), row.names = 1)
```

#Explore the data

```
str(dataPml)
```

```
## 'data.frame':    19622 obs. of  159 variables:
##  $ 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 132308423
2 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      : num  NA NA NA NA NA NA NA NA NA NA ...
##  $ kurtosis_picth_belt     : num  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 ...
##  $ 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 ...
##  $ max_roll_belt           : Factor w/ 196 levels "-0.2","-0.4",..: 196 196
196 196 196 196 196 196 196 196 ...
##  $ max_picth_belt          : Factor w/ 23 levels "10","11","17",..: 23 23 2
3 23 23 23 23 23 23 23 ...
##  $ max_yaw_belt            : num  NA NA NA NA NA NA NA NA NA NA ...
##  $ min_roll_belt           : Factor w/ 185 levels "-0.2","-0.5",..: 185 185
185 185 185 185 185 185 185 185 ...
##  $ min_pitch_belt          : Factor w/ 17 levels "0","1","15","16",..: 17 1
7 17 17 17 17 17 17 17 17 ...
##  $ min_yaw_belt            : num  NA NA NA NA NA NA NA NA NA NA ...
##  $ amplitude_roll_belt     : Factor w/ 149 levels "0","0.1","0.13",..: 149 1
49 149 149 149 149 149 149 149 149 ...
##  $ amplitude_pitch_belt    : Factor w/ 14 levels "0","1","10","11",..: 14 1
4 14 14 14 14 14 14 14 14 ...
##  $ amplitude_yaw_belt      : num  NA NA NA NA NA NA NA NA NA NA ...
##  $ var_total_accel_belt    : Factor w/ 66 levels "0","0.0217","0.0278",..: 6
6 66 66 66 66 66 66 66 66 66 ...
##  $ avg_roll_belt           : Factor w/ 192 levels "-0.2","-0.3",..: 192 192
192 192 192 192 192 192 192 192 ...
##  $ stddev_roll_belt        : Factor w/ 70 levels "0","0.091","0.0957",..: 7
0 70 70 70 70 70 70 70 70 70 ...
##  $ var_roll_belt           : Factor w/ 97 levels "0","0.0083","0.0092",..: 9
7 97 97 97 97 97 97 97 97 97 ...
```

```
##  $ avg_pitch_belt          : Factor w/ 215 levels "-0.2","-0.4",..: 215 215
215 215 215 215 215 215 215 215 ...
##  $ stddev_pitch_belt       : Factor w/ 44 levels "0","0.0571","0.1",..: 44 4
4 44 44 44 44 44 44 44 44 44 ...
##  $ var_pitch_belt          : Factor w/ 64 levels "0","0.0033","0.0393",..: 6
4 64 64 64 64 64 64 64 64 64 ...
##  $ avg_yaw_belt            : Factor w/ 241 levels "-0.1","-0.7",..: 241 241
241 241 241 241 241 241 241 241 ...
##  $ stddev_yaw_belt         : Factor w/ 59 levels "0","0.0407","0.0522",..: 5
9 59 59 59 59 59 59 59 59 59 ...
##  $ var_yaw_belt            : Factor w/ 146 levels "0","0.0017","0.0027",..:
146 146 146 146 146 146 146 146 146 146 ...
##  $ gyros_belt_x            : num  0 0.02 0 0.02 0.02 0.02 0.02 0.02 0.02 0.0
3 ...
##  $ 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 -2
1 ...
##  $ 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 60
9 ...
##  $ magnet_belt_z           : int  -313 -311 -305 -310 -302 -312 -311 -313 -3
12 -308 ...
##  $ roll_arm                : num  -128 -128 -128 -128 -128 -128 -128 -128 -1
28 -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 -1
61 -161 ...
##  $ total_accel_arm         : int  34 34 34 34 34 34 34 34 34 34 ...
##  $ var_accel_arm           : Factor w/ 396 levels "0","0.0179","0.02",..: 39
6 396 396 396 396 396 396 396 396 396 ...
##  $ avg_roll_arm            : Factor w/ 331 levels "-0.7853","-1.9176",..: 33
1 331 331 331 331 331 331 331 331 331 ...
##  $ stddev_roll_arm         : Factor w/ 331 levels "0","0.05","0.1081",..: 33
1 331 331 331 331 331 331 331 331 331 ...
##  $ var_roll_arm            : Factor w/ 331 levels "0","0.0025","0.0117",..:
331 331 331 331 331 331 331 331 331 331 ...
##  $ avg_pitch_arm           : Factor w/ 331 levels "-0.3724","-0.4137",..: 33
1 331 331 331 331 331 331 331 331 331 ...
##  $ stddev_pitch_arm        : Factor w/ 331 levels "0","0.0153","0.135",..: 3
31 331 331 331 331 331 331 331 331 331 ...
##  $ var_pitch_arm           : Factor w/ 331 levels "0","0.0182","0.0275",..:
331 331 331 331 331 331 331 331 331 331 ...
##  $ avg_yaw_arm             : Factor w/ 331 levels "-0.0188","-0.07",..: 331
331 331 331 331 331 331 331 331 331 ...
```

```
##  $ stddev_yaw_arm          : Factor w/ 328 levels "0","0.3471","0.3594",..:
328 328 328 328 328 328 328 328 328 328 ...
##  $ var_yaw_arm             : Factor w/ 328 levels "0","0.1205","0.1292",..:
328 328 328 328 328 328 328 328 328 328 ...
##  $ gyros_arm_x             : num  0 0.02 0.02 0.02 0 0.02 0 0.02 0.02 0.0
2 ...
##  $ gyros_arm_y             : num  0 -0.02 -0.02 -0.03 -0.03 -0.03 -0.03 -0.0
2 -0.03 -0.03 ...
##  $ gyros_arm_z             : num  -0.02 -0.02 -0.02 0.02 0 0 0 0 -0.02 -0.0
2 ...
##  $ accel_arm_x             : int  -288 -290 -289 -289 -289 -289 -289 -289 -2
88 -288 ...
##  $ accel_arm_y             : int  109 110 110 111 111 111 111 111 109 11
0 ...
##  $ accel_arm_z             : int  -123 -125 -126 -123 -123 -122 -125 -124 -1
22 -124 ...
##  $ magnet_arm_x            : int  -368 -369 -368 -372 -374 -369 -373 -372 -3
69 -376 ...
##  $ magnet_arm_y            : int  337 337 344 344 337 342 336 338 341 33
4 ...
##  $ magnet_arm_z            : int  516 513 513 512 506 513 509 510 518 51
6 ...
##  $ kurtosis_roll_arm       : num  NA NA NA NA NA NA NA NA NA NA ...
##  $ kurtosis_picth_arm      : num  NA NA NA NA NA NA NA NA NA NA ...
##  $ kurtosis_yaw_arm        : num  NA NA NA NA NA NA NA NA NA NA ...
##  $ skewness_roll_arm       : num  NA NA NA NA NA NA NA NA NA NA ...
##  $ skewness_pitch_arm      : num  NA NA NA NA NA NA NA NA NA NA ...
##  $ skewness_yaw_arm        : num  NA NA NA NA NA NA NA NA NA NA ...
##  $ max_roll_arm            : Factor w/ 291 levels "-0.1","-0.2",..: 291 291
291 291 291 291 291 291 291 291 ...
##  $ max_picth_arm           : Factor w/ 264 levels "-0.8","-1.8",..: 264 264
264 264 264 264 264 264 264 264 ...
##  $ max_yaw_arm             : Factor w/ 52 levels "13","14","15",..: 52 52 5
2 52 52 52 52 52 52 52 ...
##  $ min_roll_arm            : Factor w/ 279 levels "-0.2","-0.6",..: 279 279
279 279 279 279 279 279 279 279 ...
##  $ min_pitch_arm           : Factor w/ 291 levels "-1","-1.2","-10.4",..: 29
1 291 291 291 291 291 291 291 291 291 ...
##  $ min_yaw_arm             : Factor w/ 39 levels "1","10","11",..: 39 39 39
39 39 39 39 39 39 39 ...
##  $ amplitude_roll_arm      : Factor w/ 307 levels "0","0.03","0.5",..: 307 3
07 307 307 307 307 307 307 307 307 ...
##  $ amplitude_pitch_arm     : Factor w/ 295 levels "0","1","10","10.4",..: 29
5 295 295 295 295 295 295 295 295 295 ...
##  $ amplitude_yaw_arm       : Factor w/ 52 levels "0","1","10","11",..: 52 5
2 52 52 52 52 52 52 52 52 ...
##  $ 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 ...
##  $ kurtosis_picth_dumbbell : num  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 ...
##  $ skewness_pitch_dumbbell : num  NA NA NA NA NA NA NA NA NA NA ...
##  $ skewness_yaw_dumbbell   : logi  NA NA NA NA NA NA ...
##  $ max_roll_dumbbell       : Factor w/ 339 levels "-10.1","-11.1",..: 339 33
9 339 339 339 339 339 339 339 339 ...
##  $ max_picth_dumbbell      : Factor w/ 340 levels "-100.1","-100.3",..: 340
340 340 340 340 340 340 340 340 340 ...
##  $ max_yaw_dumbbell        : num  NA NA NA NA NA NA NA NA NA NA ...
##  $ min_roll_dumbbell       : Factor w/ 333 levels "-1","-1.2","-10.1",..: 33
3 333 333 333 333 333 333 333 333 333 ...
##  $ min_pitch_dumbbell      : Factor w/ 357 levels "-1.5","-10.5",..: 357 35
7 357 357 357 357 357 357 357 357 ...
##  $ min_yaw_dumbbell        : num  NA NA NA NA NA NA NA NA NA NA ...
##  $ amplitude_roll_dumbbell : Factor w/ 388 levels "0","0.96","1",..: 388 38
8 388 388 388 388 388 388 388 388 ...
##  $ amplitude_pitch_dumbbell: Factor w/ 384 levels "0","1.02","1.15",..: 384
384 384 384 384 384 384 384 384 384 ...
##   [list output truncated]
```

As you can see the first 5 columns mean nothing with respect to the machine learning features. Lets exclude them

```
dataPml <- dataPml[, 6:dim(dataPml)[2]]
```

#Feature selection The given problem is a classifier problem as at the end, we need to identify which classe a given datarow belongs. Hence for classifer machine learning we need to identify list of features.

We have 153 columns but not all columns are useful. lets find out which ones are not useful and eliminate from the feature selection process.

Lets remove unnecessary features by looking at the following types of data 1. Almost all the data in that column is NA. meaning it will not be useful to meaning predict / help predict an outcome.Lets have a threshold of 95% of rows if N/A, we will eliminate that column.

##Eliminate NA Columns

```
th_rows <- dim(dataPml)[1] * 0.95
naCols <- apply(dataPml, 2, function(x) sum(is.na(x)) > th_rows || sum(x=="")
> th_rows)

dataPml <- dataPml[, !naCols]
```

2. The column has same unique data for all rows. Again not an useful feature to predict.

3. The column has few unique values but the proporation frequence of occurence of highest to next highest is too high. Again this in our context we can eliminate as we dont have any binary value columns.

##Eliminate near zero & zero predicators

```
nzvCols <- nearZeroVar(dataPml, saveMetrics = TRUE)
dataPml <- dataPml[, nzvCols$nzv==FALSE]
```

#Make outcome as a factor variable dataPml$classe = factor(dataPml$classe)

#Model building

##Data Slicing Lets split the data into 70/30 for training and testing the model.

```
trainIndex <- createDataPartition(dataPml$classe, p = 0.7,
list = FALSE,
 times = 1)
training <- dataPml[trainIndex,]
testing <- dataPml[-trainIndex,]
dim(training)
```

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

```
dim(testing)
```

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

##Apply Classifier Models

1. Classifier Model- Recursive Partitioning And Regression Trees

```
###Train the model
modFit_RPART <- train(classe ~ ., data=training, method="rpart")
print(modFit_RPART$finalModel)
```

```
## n= 13737
##
## node), split, n, loss, yval, (yprob)
##       * denotes terminal node
##
##  1) root 13737 9831 A (0.28 0.19 0.17 0.16 0.18)
##    2) roll_belt< 130.5 12571 8676 A (0.31 0.21 0.19 0.18 0.11)
##      4) pitch_forearm< -34 1115    8 A (0.99 0.0072 0 0 0) *
##      5) pitch_forearm>=-34 11456 8668 A (0.24 0.23 0.21 0.2 0.12)
##       10) magnet_dumbbell_y< 439.5 9672 6940 A (0.28 0.18 0.24 0.19 0.1)
##         20) roll_forearm< 121.5 5913 3475 A (0.41 0.18 0.18 0.17 0.06) *
##         21) roll_forearm>=121.5 3759 2524 C (0.078 0.18 0.33 0.23 0.18) *
##       11) magnet_dumbbell_y>=439.5 1784  885 B (0.031 0.5 0.041 0.22 0.2) *
##    3) roll_belt>=130.5 1166   11 E (0.0094 0 0 0 0.99) *
```

```
###Test the model
predictClasse <- predict(modFit_RPART, newdata = testing)

###Evaluate the efficiency of the model
confusionMatrix(predictClasse, testing$classe)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction    A    B    C    D    E
##          A 1522  489  496  449  169
##          B   25  387   35  168  130
##          C  124  263  495  347  307
##          D    0    0    0    0    0
##          E    3    0    0    0  476
##
## Overall Statistics
##
##                Accuracy : 0.4894
##                  95% CI : (0.4765, 0.5022)
##     No Information Rate : 0.2845
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.3316
##
##  Mcnemar's Test P-Value : NA
##
## Statistics by Class:
##
##                      Class: A Class: B Class: C Class: D Class: E
## Sensitivity            0.9092  0.33977  0.48246   0.0000  0.43993
## Specificity            0.6193  0.92457  0.78576   1.0000  0.99938
## Pos Pred Value         0.4870  0.51946  0.32227      NaN  0.99374
## Neg Pred Value         0.9449  0.85370  0.87790   0.8362  0.88790
## Prevalence             0.2845  0.19354  0.17434   0.1638  0.18386
## Detection Rate         0.2586  0.06576  0.08411   0.0000  0.08088
## Detection Prevalence   0.5310  0.12659  0.26100   0.0000  0.08139
## Balanced Accuracy      0.7643  0.63217  0.63411   0.5000  0.71965
```

2. Classifier Model- Random Forest ###Train the model

```
modFit_RF <- train(classe ~ ., data=training, method="rf")
print(modFit_RF$finalModel)
```

```
##
## Call:
##  randomForest(x = x, y = y, mtry = param$mtry)
##                Type of random forest: classification
##                      Number of trees: 500
## No. of variables tried at each split: 27
##
##         OOB estimate of  error rate: 0.23%
## Confusion matrix:
##      A    B    C    D    E class.error
## A 3906    0    0    0    0 0.000000000
## B    9 2645    3    1    0 0.004890895
## C    0    4 2391    1    0 0.002086811
## D    0    0    5 2246    1 0.002664298
## E    0    1    0    7 2517 0.003168317
```

```
###Test the model
predictClasse <- predict(modFit_RF, newdata = testing)

###Evaluate the efficiency of the model
confusionMatrix(predictClasse, testing$classe)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction    A    B    C    D    E
##          A 1672    0    0    0    0
##          B    1 1138    1    0    0
##          C    0    1 1022    6    0
##          D    0    0    3  958    1
##          E    1    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            0.9988   0.9991   0.9961   0.9938   0.9991
## Specificity            1.0000   0.9996   0.9986   0.9992   0.9998
## Pos Pred Value         1.0000   0.9982   0.9932   0.9958   0.9991
## Neg Pred Value         0.9995   0.9998   0.9992   0.9988   0.9998
## Prevalence             0.2845   0.1935   0.1743   0.1638   0.1839
## Detection Rate         0.2841   0.1934   0.1737   0.1628   0.1837
## Detection Prevalence   0.2841   0.1937   0.1749   0.1635   0.1839
## Balanced Accuracy      0.9994   0.9994   0.9973   0.9965   0.9994
```

3. Classifier Model- Linear Discriminant Analysis

```
###Train the model
modFit_LDA <- train(classe ~ ., data=training, method="lda")
print(modFit_LDA$finalModel)
```

```
## Call:
## lda(x, grouping = y)
##
## Prior probabilities of groups:
##         A         B         C         D         E
## 0.2843416 0.1934920 0.1744195 0.1639368 0.1838101
##
## Group means:
##   num_window roll_belt pitch_belt   yaw_belt total_accel_belt gyros_belt_x
## A   383.0317  59.26880  0.4852483 -12.446354         10.66897 -0.005335381
## B   501.5515  62.95861  0.2532543 -16.172818         10.86305 -0.004571106
## C   483.7755  64.69875 -0.7516068  -8.163172         11.18823 -0.018484975
## D   427.7567  59.74787  2.0619671 -19.774996         11.08659 -0.014347247
## E   372.5251  75.44804  0.1972832  -3.793596         12.81386  0.009778218
##   gyros_belt_y gyros_belt_z accel_belt_x accel_belt_y accel_belt_z
## A   0.03979263   -0.1220942    -6.473630     28.92729    -62.60164
## B   0.04136945   -0.1306772    -5.304364     31.01881    -70.12904
## C   0.03825543   -0.1368280    -4.461603     31.16569    -70.83180
## D   0.03616785   -0.1316696    -8.586146     30.01332    -67.56172
## E   0.03957228   -0.1303050    -4.035248     29.39881    -93.00079
##   magnet_belt_x magnet_belt_y magnet_belt_z   roll_arm  pitch_arm
## A      57.32258      602.1933     -337.8810 -0.6898259   3.073912
## B      48.03725      599.4127     -336.8277 33.0079985  -7.050598
## C      56.65275      600.0104     -337.5392 25.3559432  -1.630401
## D      48.11723      593.9192     -341.3917 23.1299023 -10.451794
## E      63.41465      568.2689     -378.7485 18.9506772 -12.879853
##      yaw_arm total_accel_arm gyros_arm_x gyros_arm_y gyros_arm_z
## A -11.536485        27.29442  0.02472094  -0.2174245   0.2638300
## B   8.490433        26.52558  0.02879609  -0.2794582   0.2599248
## C   3.993844        24.29424  0.10169449  -0.2649290   0.2738147
## D   5.266097        23.54840  0.04187833  -0.2508837   0.2566918
## E  -1.364713        24.87842  0.05417426  -0.2884634   0.2863485
##   accel_arm_x accel_arm_y accel_arm_z magnet_arm_x magnet_arm_y
## A  -132.63364    47.29749   -74.38889    -20.86226    236.10420
## B   -44.18172    28.84349   -96.10986    234.35741    131.20617
## C   -76.07095    41.67195   -53.30801    156.71786    191.39524
## D    16.77309    25.96492   -50.17940    399.96536     95.02398
## E   -18.58970    14.08119   -78.12594    320.93901     82.07406
##   magnet_arm_z roll_dumbbell pitch_dumbbell yaw_dumbbell
## A     413.7512      21.21757     -19.019963     1.430322
## B     199.6554      36.77571       3.684815    15.390367
## C     363.5338     -12.77864     -24.912817   -15.139477
## D     293.2229      50.39964      -2.528837     1.256954
## E     211.9648      26.72273      -6.219765     5.429959
##   total_accel_dumbbell gyros_dumbbell_x gyros_dumbbell_y gyros_dumbbell_z
## A             14.63543        0.1669918       0.02199437       -0.1519432
## B             14.37208        0.1639767       0.01086155       -0.1456584
## C             13.01711        0.1948706       0.05174457       -0.1541027
```

```
## D             11.44183        0.2099556       0.01651421       -0.1318384
## E             14.39485        0.1311208       0.11204752       -0.1371762
##   accel_dumbbell_x accel_dumbbell_y accel_dumbbell_z magnet_dumbbell_x
## A      -50.5217614         52.21147        -56.35535         -385.5102
## B       -0.3239278         69.93040        -15.73213         -250.3691
## C      -40.7787980         31.07721        -52.50543         -362.9558
## D      -23.3587922         53.87389        -34.56794         -318.5160
## E      -16.8922772         55.73347        -23.61228         -288.4455
##   magnet_dumbbell_y magnet_dumbbell_z roll_forearm pitch_forearm
## A          216.3740          10.34716     26.78248     -7.112399
## B          272.0831          50.21783     33.87191     14.758093
## C          152.7504          62.34933     61.26069     12.051023
## D          219.3446          57.01155     16.96171     28.127815
## E          240.9727          71.08990     38.63727     16.552665
##   yaw_forearm total_accel_forearm gyros_forearm_x gyros_forearm_y
## A   25.382053            32.19355       0.1820251      0.04807988
## B   13.823883            35.26373       0.1349661      0.07073363
## C   40.333769            34.91152       0.1970576      0.10724124
## D    5.381559            36.09725       0.1165897     -0.02103908
## E    9.685323            36.74059       0.1327010      0.07451485
##   gyros_forearm_z accel_forearm_x accel_forearm_y accel_forearm_z
## A       0.1139222        0.515105        171.9470       -60.37583
## B       0.1756622      -74.079007        138.0369       -47.23702
## C       0.1413564      -44.438230        214.0797       -63.21285
## D       0.1111057     -153.218028        152.1150       -49.39565
## E       0.1596158      -73.481188        143.1442       -58.74931
##   magnet_forearm_x magnet_forearm_y magnet_forearm_z
## A        -198.6221         480.8630         409.9954
## B        -327.6802         282.8220         376.7904
## C        -332.7763         507.3022         463.6361
## D        -454.9361         314.3020         362.7726
## E        -324.1949         269.6194         355.5632
##
## Coefficients of linear discriminants:
##                                LD1           LD2           LD3
## num_window            4.343376e-04 -0.0006395982  0.0017020024
## roll_belt             5.893733e-02  0.0960811711  0.0055758058
## pitch_belt            3.135405e-02  0.0072742105 -0.0778486615
## yaw_belt             -9.192488e-03  0.0005723823 -0.0104171885
## total_accel_belt     -2.321639e-02 -0.0114513101 -0.2900855869
## gyros_belt_x          6.651845e-01  0.2059039332  0.6966091282
## gyros_belt_y         -1.456759e+00 -1.9019208921 -0.6796376054
## gyros_belt_z          5.910832e-01  0.4628473061  0.4487696023
## accel_belt_x         -5.030400e-04 -0.0002654717  0.0178995267
## accel_belt_y         -2.166487e-02 -0.0311458831  0.0564466702
## accel_belt_z          7.798576e-03  0.0290163199 -0.0078022035
## magnet_belt_x        -1.093625e-02  0.0026433428 -0.0221895251
## magnet_belt_y        -2.321152e-02 -0.0073683092 -0.0015758607
## magnet_belt_z         7.832073e-03 -0.0008284672  0.0111636305
```

```
## roll_arm              1.004891e-03  0.0003702368  0.0023999191
## pitch_arm            -3.126000e-03  0.0057242451  0.0050778576
## yaw_arm               1.123451e-03 -0.0007371541  0.0015401988
## total_accel_arm       6.473838e-03 -0.0232319678 -0.0239801993
## gyros_arm_x           1.308753e-01  0.0212458729 -0.0835096987
## gyros_arm_y           8.913822e-02 -0.0637689022 -0.1431624994
## gyros_arm_z          -1.507391e-01 -0.1241147020  0.0182537504
## accel_arm_x          -2.865373e-03 -0.0049750710 -0.0082202840
## accel_arm_y          -2.460761e-03  0.0153223995 -0.0011653361
## accel_arm_z           1.005804e-02 -0.0018509024  0.0016341199
## magnet_arm_x         -1.482936e-05 -0.0003810038  0.0022099844
## magnet_arm_y         -1.465270e-03 -0.0054967346  0.0057337236
## magnet_arm_z         -3.790982e-03 -0.0022370761 -0.0055868112
## roll_dumbbell         2.401313e-03 -0.0040536303 -0.0023468614
## pitch_dumbbell       -4.844018e-03 -0.0029913603 -0.0036669213
## yaw_dumbbell         -7.635042e-03  0.0068197680 -0.0030340302
## total_accel_dumbbell  6.622395e-02  0.0639794535  0.0004529643
## gyros_dumbbell_x      2.817121e-01 -0.5751623948  0.1609021870
## gyros_dumbbell_y      2.301455e-01 -0.2855396397  0.0049060571
## gyros_dumbbell_z      2.257550e-01 -0.3880496015 -0.0555247443
## accel_dumbbell_x      1.169651e-02  0.0087278567  0.0005817334
## accel_dumbbell_y      1.733474e-03  0.0030166292  0.0014498296
## accel_dumbbell_z      2.332750e-03  0.0023000639  0.0019143162
## magnet_dumbbell_x    -3.800819e-03 -0.0002297157  0.0033886011
## magnet_dumbbell_y    -8.476574e-04  0.0022439215 -0.0005034534
## magnet_dumbbell_z     1.267315e-02 -0.0100241379 -0.0012765950
## roll_forearm          1.502545e-03  0.0014037609  0.0002775173
## pitch_forearm         1.640068e-02 -0.0133375588  0.0057031006
## yaw_forearm          -2.781313e-04  0.0007178516  0.0007394127
## total_accel_forearm   3.279783e-02  0.0069195712 -0.0081087205
## gyros_forearm_x      -5.742705e-02 -0.0624950327  0.2115978895
## gyros_forearm_y      -1.736232e-02 -0.0398213099  0.0344738328
## gyros_forearm_z       1.074022e-01  0.1462865602 -0.0867288107
## accel_forearm_x       3.103457e-03  0.0106468600  0.0004427196
## accel_forearm_y       7.650028e-04 -0.0011056540 -0.0006671561
## accel_forearm_z      -7.050584e-03  0.0027734088  0.0036764344
## magnet_forearm_x     -1.609998e-03 -0.0035231119 -0.0000714468
## magnet_forearm_y     -9.023046e-04 -0.0014692634  0.0003532143
## magnet_forearm_z     -7.554407e-05 -0.0014121938 -0.0003687905
##                              LD4
## num_window            3.420531e-05
## roll_belt             7.176974e-02
## pitch_belt            9.574673e-03
## yaw_belt             -4.291728e-03
## total_accel_belt     -1.543406e-01
## gyros_belt_x          4.652755e-01
## gyros_belt_y          1.324599e+00
## gyros_belt_z         -6.714394e-01
## accel_belt_x          5.090641e-03
```

```
## accel_belt_y          6.157300e-03
## accel_belt_z          1.713903e-02
## magnet_belt_x        -3.113337e-03
## magnet_belt_y        -3.462085e-03
## magnet_belt_z         2.983269e-03
## roll_arm              6.698996e-04
## pitch_arm             1.641743e-03
## yaw_arm              -1.427452e-03
## total_accel_arm      -1.869443e-02
## gyros_arm_x           3.736227e-02
## gyros_arm_y           1.823914e-01
## gyros_arm_z           1.504701e-01
## accel_arm_x          -2.190513e-03
## accel_arm_y           3.905344e-03
## accel_arm_z          -7.044369e-03
## magnet_arm_x          1.197430e-03
## magnet_arm_y          4.147195e-04
## magnet_arm_z          1.924743e-03
## roll_dumbbell        -7.534340e-03
## pitch_dumbbell       -4.619588e-03
## yaw_dumbbell         -3.820748e-03
## total_accel_dumbbell  2.016328e-03
## gyros_dumbbell_x      4.004926e-03
## gyros_dumbbell_y      1.805559e-01
## gyros_dumbbell_z      4.208525e-02
## accel_dumbbell_x      5.997506e-03
## accel_dumbbell_y     -2.236858e-03
## accel_dumbbell_z      1.545604e-03
## magnet_dumbbell_x    -2.333334e-03
## magnet_dumbbell_y    -2.063882e-03
## magnet_dumbbell_z     9.697575e-03
## roll_forearm          1.319473e-03
## pitch_forearm         1.761540e-04
## yaw_forearm           1.043657e-03
## total_accel_forearm   3.692616e-03
## gyros_forearm_x       1.294736e-01
## gyros_forearm_y       1.987009e-02
## gyros_forearm_z      -5.827401e-02
## accel_forearm_x       4.048446e-03
## accel_forearm_y      -2.076424e-03
## accel_forearm_z      -4.967900e-03
## magnet_forearm_x     -1.225032e-03
## magnet_forearm_y      3.371937e-04
## magnet_forearm_z      1.102034e-03
##
## Proportion of trace:
##    LD1    LD2    LD3    LD4
## 0.4733 0.2434 0.1708 0.1126
```

```
###Test the model
predictClasse <- predict(modFit_LDA, newdata = testing)

###Evaluate the efficiency of the model
confusionMatrix(predictClasse, testing$classe)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction    A    B    C    D    E
##          A 1379  136   86   52   38
##          B   49  753  101   41  146
##          C  102  157  689  129  100
##          D  136   44  119  710  114
##          E    8   49   31   32  684
##
## Overall Statistics
##
##                Accuracy : 0.7162
##                  95% CI : (0.7045, 0.7277)
##     No Information Rate : 0.2845
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.6413
##
##  Mcnemar's Test P-Value : < 2.2e-16
##
## Statistics by Class:
##
##                      Class: A Class: B Class: C Class: D Class: E
## Sensitivity            0.8238   0.6611   0.6715   0.7365   0.6322
## Specificity            0.9259   0.9290   0.8996   0.9161   0.9750
## Pos Pred Value         0.8155   0.6908   0.5854   0.6322   0.8507
## Neg Pred Value         0.9297   0.9195   0.9284   0.9467   0.9217
## Prevalence             0.2845   0.1935   0.1743   0.1638   0.1839
## Detection Rate         0.2343   0.1280   0.1171   0.1206   0.1162
## Detection Prevalence   0.2873   0.1852   0.2000   0.1908   0.1366
## Balanced Accuracy      0.8748   0.7950   0.7856   0.8263   0.8036
```

#Conclusion

Of all the methods, Random Forest seems the best model with 99% accuracy. We will use that model alone and there is no need for ensemle of additional models.