Practical Machine Learning Course Project

Mark Zhang

July 18, 2018

Overview

This project concerns about classfying different type of human activities by using data collected from body activity tracking devices. Since the number of classes to be classified has more than 2 outcomes, so we will implement classification tree, random forest, and boosting classifiers to see which one performs the best, and then use the best performer to predict on the test set.

Setups

```
knitr::opts_chunk$set(message = F, fig.align = 'center')
require(caret)
require(dplyr)
require(rattle)
```

Read data

```
trainSet <- read.csv(url('https://d396qusza40orc.cloudfront.net/predmachlearn/pml-training.csv'))
testSetFinal <- read.csv(url('https://d396qusza40orc.cloudfront.net/predmachlearn/pml-testing.csv'))
```

Custom function

```
getCfMTX <- function (model, data) {
    pred <- predict(model, data)
    confusionMatrix(pred, data$classe)
}
```

Exploratory analysis

By looking at the structure of the dataset, we can see that there are a lot of columns with NA value or blank value. By examine more closely, we can see that all the columns with either NA or blank values, their number of NA or blank values are the same: 19216, which is accountable for about 97% of the data size, so we are going to drop these columns.

```
str(trainSet)
```

```
## '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 1
##  $ 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 1.42 1.43 1.45 ...
##  $ pitch_belt              : num  8.07 8.07 8.07 8.05 8.07 8.06 8.09 8.13 8.16 8.17 ...
##  $ yaw_belt                : num  -94.4 -94.4 -94.4 -94.4 -94.4 -94.4 -94.4 -94.4 -94.4 -94.4 ...
##  $ total_accel_belt        : int  3 3 3 3 3 3 3 3 3 3 ...
```

```
##  $ kurtosis_roll_belt      : Factor w/ 397 levels "","-0.016850",..: 1 1 1 1 1 1 1 1 1 1 ...
##  $ kurtosis_picth_belt     : Factor w/ 317 levels "","-0.021887",..: 1 1 1 1 1 1 1 1 1 1 ...
##  $ kurtosis_yaw_belt       : Factor w/ 2 levels "","#DIV/0!": 1 1 1 1 1 1 1 1 1 1 ...
##  $ skewness_roll_belt      : Factor w/ 395 levels "","-0.003095",..: 1 1 1 1 1 1 1 1 1 1 ...
##  $ skewness_roll_belt.1    : Factor w/ 338 levels "","-0.005928",..: 1 1 1 1 1 1 1 1 1 1 ...
##  $ skewness_yaw_belt       : Factor w/ 2 levels "","#DIV/0!": 1 1 1 1 1 1 1 1 1 1 ...
##  $ max_roll_belt           : num  NA NA NA NA NA NA NA NA NA NA ...
##  $ max_picth_belt          : int  NA NA NA NA NA NA NA NA NA NA ...
##  $ max_yaw_belt            : Factor w/ 68 levels "","-0.1","-0.2",..: 1 1 1 1 1 1 1 1 1 1 ...
##  $ min_roll_belt           : num  NA NA NA NA NA NA NA NA NA NA ...
##  $ min_pitch_belt          : int  NA NA NA NA NA NA NA NA NA NA ...
##  $ min_yaw_belt            : Factor w/ 68 levels "","-0.1","-0.2",..: 1 1 1 1 1 1 1 1 1 1 ...
##  $ amplitude_roll_belt     : num  NA NA NA NA NA NA NA NA NA NA ...
##  $ amplitude_pitch_belt    : int  NA NA NA NA NA NA NA NA NA NA ...
##  $ amplitude_yaw_belt      : Factor w/ 4 levels "","#DIV/0!","0.00",..: 1 1 1 1 1 1 1 1 1 1 ...
##  $ var_total_accel_belt    : num  NA NA NA NA NA NA NA NA NA NA ...
##  $ avg_roll_belt           : num  NA NA NA NA NA NA NA NA NA NA ...
##  $ stddev_roll_belt        : num  NA NA NA NA NA NA NA NA NA NA ...
##  $ var_roll_belt           : num  NA NA NA NA NA NA NA NA NA NA ...
##  $ avg_pitch_belt          : num  NA NA NA NA NA NA NA NA NA NA ...
##  $ stddev_pitch_belt       : num  NA NA NA NA NA NA NA NA NA NA ...
##  $ var_pitch_belt          : num  NA NA NA NA NA NA NA NA NA NA ...
##  $ avg_yaw_belt            : num  NA NA NA NA NA NA NA NA NA NA ...
##  $ stddev_yaw_belt         : num  NA NA NA NA NA NA NA NA NA NA ...
##  $ var_yaw_belt            : num  NA NA NA NA NA NA NA NA NA NA ...
##  $ gyros_belt_x            : num  0 0.02 0 0.02 0.02 0.02 0.02 0.02 0.02 0.03 ...
##  $ gyros_belt_y            : num  0 0 0 0 0.02 0 0 0 0 0 ...
##  $ gyros_belt_z            : num  -0.02 -0.02 -0.02 -0.03 -0.02 -0.02 -0.02 -0.02 -0.02 0 ...
##  $ accel_belt_x            : int  -21 -22 -20 -22 -21 -21 -22 -22 -20 -21 ...
##  $ accel_belt_y            : int  4 4 5 3 2 4 3 4 2 4 ...
##  $ accel_belt_z            : int  22 22 23 21 24 21 21 21 24 22 ...
##  $ magnet_belt_x           : int  -3 -7 -2 -6 -6 0 -4 -2 1 -3 ...
##  $ magnet_belt_y           : int  599 608 600 604 600 603 599 603 602 609 ...
##  $ magnet_belt_z           : int  -313 -311 -305 -310 -302 -312 -311 -313 -312 -308 ...
##  $ roll_arm                : num  -128 -128 -128 -128 -128 -128 -128 -128 -128 -128 ...
##  $ pitch_arm               : num  22.5 22.5 22.5 22.1 22.1 22 21.9 21.8 21.7 21.6 ...
##  $ yaw_arm                 : num  -161 -161 -161 -161 -161 -161 -161 -161 -161 -161 ...
##  $ total_accel_arm         : int  34 34 34 34 34 34 34 34 34 34 ...
##  $ var_accel_arm           : num  NA NA NA NA NA NA NA NA NA NA ...
##  $ avg_roll_arm            : num  NA NA NA NA NA NA NA NA NA NA ...
##  $ stddev_roll_arm         : num  NA NA NA NA NA NA NA NA NA NA ...
##  $ var_roll_arm            : num  NA NA NA NA NA NA NA NA NA NA ...
##  $ avg_pitch_arm           : num  NA NA NA NA NA NA NA NA NA NA ...
##  $ stddev_pitch_arm        : num  NA NA NA NA NA NA NA NA NA NA ...
##  $ var_pitch_arm           : num  NA NA NA NA NA NA NA NA NA NA ...
##  $ avg_yaw_arm             : num  NA NA NA NA NA NA NA NA NA NA ...
##  $ stddev_yaw_arm          : num  NA NA NA NA NA NA NA NA NA NA ...
##  $ var_yaw_arm             : num  NA NA NA NA NA NA NA NA NA NA ...
##  $ gyros_arm_x             : num  0 0.02 0.02 0.02 0 0.02 0 0.02 0.02 0.02 ...
##  $ gyros_arm_y             : num  0 -0.02 -0.02 -0.03 -0.03 -0.03 -0.03 -0.02 -0.03 -0.03 ...
##  $ gyros_arm_z             : num  -0.02 -0.02 -0.02 0.02 0 0 0 0 -0.02 -0.02 ...
##  $ accel_arm_x             : int  -288 -290 -289 -289 -289 -289 -289 -289 -288 -288 ...
##  $ accel_arm_y             : int  109 110 110 111 111 111 111 111 109 110 ...
##  $ accel_arm_z             : int  -123 -125 -126 -123 -123 -122 -125 -124 -122 -124 ...
```

```
##  $ magnet_arm_x           : int  -368 -369 -368 -372 -374 -369 -373 -372 -369 -376 ...
##  $ magnet_arm_y           : int  337 337 344 344 337 342 336 338 341 334 ...
##  $ magnet_arm_z           : int  516 513 513 512 506 513 509 510 518 516 ...
##  $ kurtosis_roll_arm      : Factor w/ 330 levels "","-0.02438",..: 1 1 1 1 1 1 1 1 1 1 ...
##  $ kurtosis_picth_arm     : Factor w/ 328 levels "","-0.00484",..: 1 1 1 1 1 1 1 1 1 1 ...
##  $ kurtosis_yaw_arm       : Factor w/ 395 levels "","-0.01548",..: 1 1 1 1 1 1 1 1 1 1 ...
##  $ skewness_roll_arm      : Factor w/ 331 levels "","-0.00051",..: 1 1 1 1 1 1 1 1 1 1 ...
##  $ skewness_pitch_arm     : Factor w/ 328 levels "","-0.00184",..: 1 1 1 1 1 1 1 1 1 1 ...
##  $ skewness_yaw_arm       : Factor w/ 395 levels "","-0.00311",..: 1 1 1 1 1 1 1 1 1 1 ...
##  $ max_roll_arm           : num  NA NA NA NA NA NA NA NA NA NA ...
##  $ max_picth_arm          : num  NA NA NA NA NA NA NA NA NA NA ...
##  $ max_yaw_arm            : int  NA NA NA NA NA NA NA NA NA NA ...
##  $ min_roll_arm           : num  NA NA NA NA NA NA NA NA NA NA ...
##  $ min_pitch_arm          : num  NA NA NA NA NA NA NA NA NA NA ...
##  $ min_yaw_arm            : int  NA NA NA NA NA NA NA NA NA NA ...
##  $ amplitude_roll_arm     : num  NA NA NA NA NA NA NA NA NA NA ...
##  $ amplitude_pitch_arm    : num  NA NA NA NA NA NA NA NA NA NA ...
##  $ amplitude_yaw_arm      : int  NA NA NA NA NA NA NA NA NA NA ...
##  $ roll_dumbbell          : num  13.1 13.1 12.9 13.4 13.4 ...
##  $ pitch_dumbbell         : num  -70.5 -70.6 -70.3 -70.4 -70.4 ...
##  $ yaw_dumbbell           : num  -84.9 -84.7 -85.1 -84.9 -84.9 ...
##  $ kurtosis_roll_dumbbell : Factor w/ 398 levels "","-0.0035","-0.0073",..: 1 1 1 1 1 1 1 1 1 1 ...
##  $ kurtosis_picth_dumbbell: Factor w/ 401 levels "","-0.0163","-0.0233",..: 1 1 1 1 1 1 1 1 1 1 ...
##  $ kurtosis_yaw_dumbbell  : Factor w/ 2 levels "","#DIV/0!": 1 1 1 1 1 1 1 1 1 1 ...
##  $ skewness_roll_dumbbell : Factor w/ 401 levels "","-0.0082","-0.0096",..: 1 1 1 1 1 1 1 1 1 1 ...
##  $ skewness_pitch_dumbbell: Factor w/ 402 levels "","-0.0053","-0.0084",..: 1 1 1 1 1 1 1 1 1 1 ...
##  $ skewness_yaw_dumbbell  : Factor w/ 2 levels "","#DIV/0!": 1 1 1 1 1 1 1 1 1 1 ...
##  $ max_roll_dumbbell      : num  NA NA NA NA NA NA NA NA NA NA ...
##  $ max_picth_dumbbell     : num  NA NA NA NA NA NA NA NA NA NA ...
##  $ max_yaw_dumbbell       : Factor w/ 73 levels "","-0.1","-0.2",..: 1 1 1 1 1 1 1 1 1 1 ...
##  $ min_roll_dumbbell      : num  NA NA NA NA NA NA NA NA NA NA ...
##  $ min_pitch_dumbbell     : num  NA NA NA NA NA NA NA NA NA NA ...
##  $ min_yaw_dumbbell       : Factor w/ 73 levels "","-0.1","-0.2",..: 1 1 1 1 1 1 1 1 1 1 ...
##  $ amplitude_roll_dumbbell: num  NA NA NA NA NA NA NA NA NA NA ...
##   [list output truncated]
```

```
colSums(is.na(trainSet) | trainSet == "")[colSums(is.na(trainSet) | trainSet == "") != 0]
```

```
##       kurtosis_roll_belt      kurtosis_picth_belt        kurtosis_yaw_belt
##                    19216                    19216                    19216
##       skewness_roll_belt     skewness_roll_belt.1        skewness_yaw_belt
##                    19216                    19216                    19216
##            max_roll_belt           max_picth_belt             max_yaw_belt
##                    19216                    19216                    19216
##            min_roll_belt           min_pitch_belt             min_yaw_belt
##                    19216                    19216                    19216
##      amplitude_roll_belt     amplitude_pitch_belt       amplitude_yaw_belt
##                    19216                    19216                    19216
##     var_total_accel_belt            avg_roll_belt         stddev_roll_belt
##                    19216                    19216                    19216
##            var_roll_belt           avg_pitch_belt        stddev_pitch_belt
##                    19216                    19216                    19216
##           var_pitch_belt             avg_yaw_belt          stddev_yaw_belt
##                    19216                    19216                    19216
##             var_yaw_belt            var_accel_arm             avg_roll_arm
```

```
##                   19216                   19216                   19216
##          stddev_roll_arm            var_roll_arm           avg_pitch_arm
##                   19216                   19216                   19216
##         stddev_pitch_arm           var_pitch_arm             avg_yaw_arm
##                   19216                   19216                   19216
##           stddev_yaw_arm             var_yaw_arm        kurtosis_roll_arm
##                   19216                   19216                   19216
##       kurtosis_picth_arm         kurtosis_yaw_arm        skewness_roll_arm
##                   19216                   19216                   19216
##       skewness_pitch_arm         skewness_yaw_arm             max_roll_arm
##                   19216                   19216                   19216
##            max_picth_arm             max_yaw_arm             min_roll_arm
##                   19216                   19216                   19216
##            min_pitch_arm             min_yaw_arm       amplitude_roll_arm
##                   19216                   19216                   19216
##      amplitude_pitch_arm         amplitude_yaw_arm   kurtosis_roll_dumbbell
##                   19216                   19216                   19216
##  kurtosis_picth_dumbbell    kurtosis_yaw_dumbbell   skewness_roll_dumbbell
##                   19216                   19216                   19216
##  skewness_pitch_dumbbell    skewness_yaw_dumbbell        max_roll_dumbbell
##                   19216                   19216                   19216
##       max_picth_dumbbell         max_yaw_dumbbell        min_roll_dumbbell
##                   19216                   19216                   19216
##       min_pitch_dumbbell         min_yaw_dumbbell  amplitude_roll_dumbbell
##                   19216                   19216                   19216
## amplitude_pitch_dumbbell   amplitude_yaw_dumbbell       var_accel_dumbbell
##                   19216                   19216                   19216
##        avg_roll_dumbbell     stddev_roll_dumbbell        var_roll_dumbbell
##                   19216                   19216                   19216
##       avg_pitch_dumbbell    stddev_pitch_dumbbell       var_pitch_dumbbell
##                   19216                   19216                   19216
##         avg_yaw_dumbbell      stddev_yaw_dumbbell         var_yaw_dumbbell
##                   19216                   19216                   19216
##    kurtosis_roll_forearm   kurtosis_picth_forearm     kurtosis_yaw_forearm
##                   19216                   19216                   19216
##    skewness_roll_forearm   skewness_pitch_forearm     skewness_yaw_forearm
##                   19216                   19216                   19216
##         max_roll_forearm        max_picth_forearm          max_yaw_forearm
##                   19216                   19216                   19216
##         min_roll_forearm        min_pitch_forearm          min_yaw_forearm
##                   19216                   19216                   19216
##   amplitude_roll_forearm  amplitude_pitch_forearm    amplitude_yaw_forearm
##                   19216                   19216                   19216
##        var_accel_forearm         avg_roll_forearm      stddev_roll_forearm
##                   19216                   19216                   19216
##         var_roll_forearm        avg_pitch_forearm     stddev_pitch_forearm
##                   19216                   19216                   19216
##        var_pitch_forearm          avg_yaw_forearm       stddev_yaw_forearm
##                   19216                   19216                   19216
##          var_yaw_forearm
##                   19216
```

```
19216/19622
```

```
## [1] 0.9793089
```

Data cleansing

By removing columns with NAs or blank values, and first 7 columns, which are not human activities tracking data, we have our dataset tidy. And then we subset 20% of data from the training set as our validation set to calculate the out of sample error rate.

```
columnsRemove <- names(colSums(is.na(trainSet) | trainSet == "")[colSums(is.na(trainSet) | trainSet ==
trainSet <- select(trainSet, -columnsRemove, -(X:num_window))

set.seed(0871)
sampledIdx <- createDataPartition(y = trainSet$classe, p = 0.8, list = F)
trainSubset <- trainSet[sampledIdx,]
valiSet <- trainSet[-sampledIdx,]
```

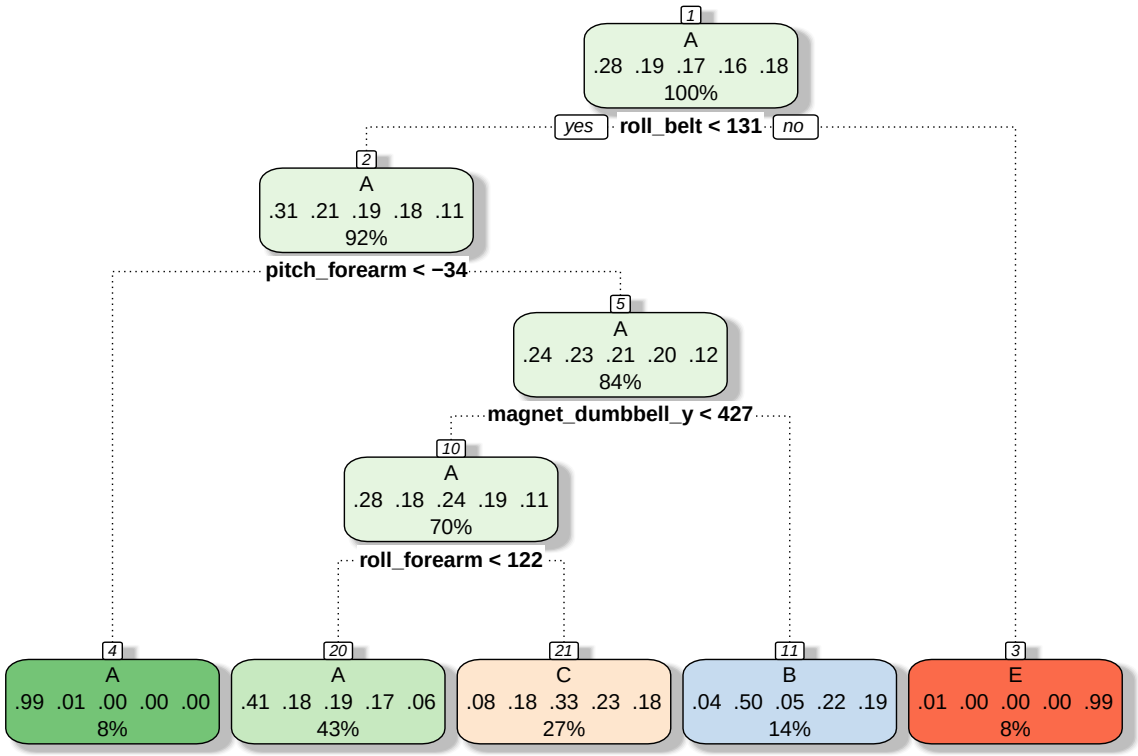
Modeling - rpart

We first train the rpart model with 3-folds cross-validation (same cv technique also used for the subsequent two models).

```
trCtrl <- trainControl(method = 'cv', number = 3)

fit_rpart <- train(classe ~ .,
                   method = 'rpart',
                   data = trainSubset,
                   trControl = trCtrl)
fancyRpartPlot(fit_rpart$finalModel)
```

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




```
getCfMTX(fit_rpart, valiSet)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction    A    B    C    D    E
##          A 1001  316  298  269   92
##          B   32  255   23  146  110
##          C   78  188  363  228  184
##          D    0    0    0    0    0
##          E    5    0    0    0  335
##
## Overall Statistics
##
##                Accuracy : 0.4981
##                  95% CI : (0.4823, 0.5139)
##     No Information Rate : 0.2845
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.3453
##  Mcnemar's Test P-Value : NA
##
## Statistics by Class:
##
##                      Class: A Class: B Class: C Class: D Class: E
## Sensitivity            0.8970   0.3360  0.53070   0.0000  0.46463
```

```
## Specificity            0.6527   0.9017  0.79068   1.0000  0.99844
## Pos Pred Value         0.5066   0.4505  0.34870      NaN  0.98529
## Neg Pred Value         0.9409   0.8499  0.88862   0.8361  0.89227
## Prevalence             0.2845   0.1935  0.17436   0.1639  0.18379
## Detection Rate         0.2552   0.0650  0.09253   0.0000  0.08539
## Detection Prevalence   0.5037   0.1443  0.26536   0.0000  0.08667
## Balanced Accuracy      0.7748   0.6188  0.66069   0.5000  0.73154
```

As the result shows, the accuracy is not quite satisfying.

Modeling - rf

Then let's take a look how random forest perform.

```
fit_rf <- train(classe ~ .,
                method = 'rf',
                data = trainSubset,
                trControl = trCtrl)
getCfMTX(fit_rf, valiSet)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction    A    B    C    D    E
##          A 1116    6    0    0    0
##          B    0  751    7    0    0
##          C    0    2  677   17    0
##          D    0    0    0  625    2
##          E    0    0    0    1  719
##
## Overall Statistics
##
##                Accuracy : 0.9911
##                  95% CI : (0.9876, 0.9938)
##     No Information Rate : 0.2845
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.9887
##  Mcnemar's Test P-Value : NA
##
## Statistics by Class:
##
##                      Class: A Class: B Class: C Class: D Class: E
## Sensitivity            1.0000   0.9895   0.9898   0.9720   0.9972
## Specificity            0.9979   0.9978   0.9941   0.9994   0.9997
## Pos Pred Value         0.9947   0.9908   0.9727   0.9968   0.9986
## Neg Pred Value         1.0000   0.9975   0.9978   0.9945   0.9994
## Prevalence             0.2845   0.1935   0.1744   0.1639   0.1838
## Detection Rate         0.2845   0.1914   0.1726   0.1593   0.1833
## Detection Prevalence   0.2860   0.1932   0.1774   0.1598   0.1835
## Balanced Accuracy      0.9989   0.9936   0.9920   0.9857   0.9985
```

```
varImp(fit_rf$finalModel)
```

```
##                       Overall
## roll_belt           616.18151
```

```
## pitch_belt           393.09201
## yaw_belt             477.91192
## total_accel_belt     211.67358
## gyros_belt_x         138.11424
## gyros_belt_y         136.15242
## gyros_belt_z         249.41933
## accel_belt_x         151.22556
## accel_belt_y         148.52043
## accel_belt_z         308.37120
## magnet_belt_x        210.92223
## magnet_belt_y        300.23194
## magnet_belt_z        302.04548
## roll_arm             265.20014
## pitch_arm            195.38831
## yaw_arm              224.02038
## total_accel_arm      143.75009
## gyros_arm_x          176.20701
## gyros_arm_y          164.87546
## gyros_arm_z           98.16639
## accel_arm_x          236.66383
## accel_arm_y          180.04778
## accel_arm_z          163.77056
## magnet_arm_x         214.30160
## magnet_arm_y         228.42321
## magnet_arm_z         199.05820
## roll_dumbbell        314.76599
## pitch_dumbbell       205.90013
## yaw_dumbbell         244.89868
## total_accel_dumbbell 237.36442
## gyros_dumbbell_x     149.96829
## gyros_dumbbell_y     237.85656
## gyros_dumbbell_z     114.71987
## accel_dumbbell_x     238.97004
## accel_dumbbell_y     319.88326
## accel_dumbbell_z     277.95707
## magnet_dumbbell_x    350.15773
## magnet_dumbbell_y    405.46133
## magnet_dumbbell_z    446.98878
## roll_forearm         337.57525
## pitch_forearm        366.62335
## yaw_forearm          187.42670
## total_accel_forearm  136.42627
## gyros_forearm_x      117.43498
## gyros_forearm_y      151.67912
## gyros_forearm_z      120.08777
## accel_forearm_x      258.20487
## accel_forearm_y      169.88082
## accel_forearm_z      207.64056
## magnet_forearm_x     225.47772
## magnet_forearm_y     214.95955
## magnet_forearm_z     225.48096
```

It looks like random forest is pretty good at handling this type of data!

Modeling - gbm

Finally, let's examine the performance of boosting.

```
fit_gbm <- train(classe ~ .,
                 method = 'gbm',
                 data = trainSubset,
                 trControl = trCtrl,
                 verbose = F)
getCfMTX(fit_gbm, valiSet)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction    A    B    C    D    E
##          A 1102   31    0    0    3
##          B   12  701   18    2    5
##          C    1   27  659   24    5
##          D    1    0    6  610    5
##          E    0    0    1    7  703
##
## Overall Statistics
##
##                Accuracy : 0.9623
##                  95% CI : (0.9558, 0.968)
##     No Information Rate : 0.2845
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.9523
##  Mcnemar's Test P-Value : 8.433e-05
##
## Statistics by Class:
##
##                      Class: A Class: B Class: C Class: D Class: E
## Sensitivity            0.9875   0.9236   0.9635   0.9487   0.9750
## Specificity            0.9879   0.9883   0.9824   0.9963   0.9975
## Pos Pred Value         0.9701   0.9499   0.9204   0.9807   0.9887
## Neg Pred Value         0.9950   0.9818   0.9922   0.9900   0.9944
## Prevalence             0.2845   0.1935   0.1744   0.1639   0.1838
## Detection Rate         0.2809   0.1787   0.1680   0.1555   0.1792
## Detection Prevalence   0.2896   0.1881   0.1825   0.1586   0.1812
## Balanced Accuracy      0.9877   0.9559   0.9729   0.9725   0.9863
```

Not bad, but still underperforms the random forest

Prediction

Since the random forest is the best performer, so we use it as our final model to make prediction on the test set.

```
predict(fit_rf, testSetFinal)
```

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