MachineLearningAssingment1

eujo007

February 1, 2016

Executive Summary

The following analysis examines the data captured from wearable accelerometers and try to predict if the participants are performing exercises correctly. For each record in the data set an outcome is assigned in the range of letters from A through E. I will use the provided training set and random forest to create a model to apply against the provided test set to make predictions on the test set

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

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

Data Pre Processing

In this section we examine the training and test data sets to understand the data composition and to determine what variables should be used for the model.

```
## '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 1323
084232 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 1.4
5 ...
##  $ pitch_belt              : num  8.07 8.07 8.07 8.05 8.07 8.06 8.09 8.13 8.16 8.1
7 ...
##  $ 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 -30
8 ...
##  $ roll_arm                : num  -128 -128 -128 -128 -128 -128 -128 -128 -128 -12
8 ...
##  $ 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 -16
1 ...
##  $ 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 -0.0
3 -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 -28
8 ...
##  $ 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 -12
4 ...
##  $ magnet_arm_x             : int  -368 -369 -368 -372 -374 -369 -373 -372 -369 -37
6 ...
##  $ 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]
```

Cleanup

From the str() output we see that there are 160 variables and 19622 records. To reduce the number of variables I will use nearZeroVar function to identify those variables that have little to zero contribution to the outcome. In addition I will remove X, user_name, all time related variables and variables that have NA values for the following reasons: - X, user_name: These variables help identify a record but are not measurements - timestamp data: I am not performing any type of time series analysis - NA variables: variables that are NA will cause the predict function to fail - Remove the factor variables since they can cause the model to explode in the number of terms. Each factor level, except the classe, could result in a new term in our model as learned in Regression. There are some factors with a low number of levels but most are over 10 with some being over 300

```
nearZeroAnalysis <- nearZeroVar(trainingSet, saveMetrics = TRUE)
print(head(nearZeroAnalysis,20))
```

```
##                         freqRatio percentUnique zeroVar   nzv
## X                        1.000000  100.00000000   FALSE FALSE
## user_name                1.100679    0.03057792   FALSE FALSE
## raw_timestamp_part_1     1.000000    4.26562022   FALSE FALSE
## raw_timestamp_part_2     1.000000   85.53154622   FALSE FALSE
## cvtd_timestamp           1.000668    0.10192641   FALSE FALSE
## new_window              47.330049    0.01019264   FALSE  TRUE
## num_window               1.000000    4.37264295   FALSE FALSE
## roll_belt                1.101904    6.77810621   FALSE FALSE
## pitch_belt               1.036082    9.37722964   FALSE FALSE
## yaw_belt                 1.058480    9.97349913   FALSE FALSE
## total_accel_belt         1.063160    0.14779329   FALSE FALSE
## kurtosis_roll_belt    1921.600000    2.02323922   FALSE  TRUE
## kurtosis_picth_belt    600.500000    1.61553358   FALSE  TRUE
## kurtosis_yaw_belt       47.330049    0.01019264   FALSE  TRUE
## skewness_roll_belt    2135.111111    2.01304658   FALSE  TRUE
## skewness_roll_belt.1   600.500000    1.72255631   FALSE  TRUE
## skewness_yaw_belt       47.330049    0.01019264   FALSE  TRUE
## max_roll_belt            1.000000    0.99378249   FALSE FALSE
## max_picth_belt           1.538462    0.11211905   FALSE FALSE
## max_yaw_belt           640.533333    0.34654979   FALSE  TRUE
```

```
print("Number of variables we can throw out: "); print(sum(nearZeroAnalysis$nzv))
```

```
## [1] "Number of variables we can throw out: "
```

```
## [1] 60
```

```
allVariables <- row.names(nearZeroAnalysis)
varsToKeep <- allVariables[!nearZeroAnalysis$nzv]
trainingSet2<- subset(trainingSet, select=  varsToKeep ) ## This data set represents
a training set with the least userful/variable variables removed
str(trainingSet2)
```

```
## 'data.frame':    19622 obs. of  100 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 1323
084232 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 ...
##  $ 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 1.4
```

```
5 ...
##  $ pitch_belt              : num  8.07 8.07 8.07 8.05 8.07 8.06 8.09 8.13 8.16 8.1
7 ...
##  $ 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 ...
##  $ 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 ...
##  $ 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 ...
##  $ 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 ...
##  $ 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 -30
8 ...
##  $ roll_arm                : num  -128 -128 -128 -128 -128 -128 -128 -128 -128 -12
8 ...
##  $ 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 -16
1 ...
##  $ 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 ...
##  $ 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 -0.0
3 -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 -28
8 ...
##  $ 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 -12
4 ...
```

```
##  $ magnet_arm_x             : int  -368 -369 -368 -372 -374 -369 -373 -372 -369 -37
6 ...
##  $ 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 ...
##  $ 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_yaw_arm              : int  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 ...
##  $ 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 ...
##  $ 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 ...
##  $ amplitude_roll_dumbbell  : num  NA NA NA NA NA NA NA NA NA NA ...
##  $ amplitude_pitch_dumbbell: num  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 ...
##  $ avg_roll_dumbbell        : num  NA NA NA NA NA NA NA NA NA NA ...
##  $ stddev_roll_dumbbell     : num  NA NA NA NA NA NA NA NA NA NA ...
##  $ var_roll_dumbbell        : num  NA NA NA NA NA NA NA NA NA NA ...
##  $ avg_pitch_dumbbell       : num  NA NA NA NA NA NA NA NA NA NA ...
##  $ stddev_pitch_dumbbell    : num  NA NA NA NA NA NA NA NA NA NA ...
##  $ var_pitch_dumbbell       : num  NA NA NA NA NA NA NA NA NA NA ...
##  $ avg_yaw_dumbbell         : num  NA NA NA NA NA NA NA NA NA NA ...
##  $ stddev_yaw_dumbbell      : num  NA NA NA NA NA NA NA NA NA NA ...
##  $ var_yaw_dumbbell         : num  NA NA NA NA NA NA NA NA NA NA ...
##  $ 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 -23
5 ...
##  $ 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 -27
0 ...
##  $ magnet_dumbbell_x        : int  -559 -555 -561 -552 -554 -558 -551 -555 -549 -55
8 ...
##  $ 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 -15
2 ...
##  $ max_picth_forearm        : num  NA NA NA NA NA NA NA NA NA NA ...
##  $ min_pitch_forearm        : num  NA NA NA NA NA NA NA NA NA NA ...
##  $ amplitude_pitch_forearm : num  NA NA NA NA NA NA NA NA NA NA ...
```

```
##  $ total_accel_forearm     : int  36 36 36 36 36 36 36 36 36 36 ...
##  $ var_accel_forearm       : num  NA NA NA NA NA NA NA NA NA NA ...
##  $ gyros_forearm_x         : num  0.03 0.02 0.03 0.02 0.02 0.02 0.02 0.02 0.03 0.0
2 ...
##  $ 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 -21
5 ...
##  $ 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 ...
##   [list output truncated]
```

```
## Remove X, user_name and the time related variables
trainingSet2 <- subset(trainingSet2, select=-c(X,user_name,cvtd_timestamp,raw_timesta
mp_part_2,raw_timestamp_part_1))

## Next I remove columns that are full of NAs since when we predict with the model th
ese columns will cause predict to fail. Note that I chose not to use impute because t
he columns with NA seem to be mostly if not all NA. Impute would try to calculate val
ues based on nearest neighboors of specific variable value but most of the other valu
es are also NA
trainingSet2 <- trainingSet2[,colSums(is.na(trainingSet2))==0]
mostlyNotNA <- trainingSet2[,colSums(is.na(trainingSet2))/19622<.3] ## Capture column
s/variables where at least 70% of the observations are not NA
print(length(names(trainingSet2)))
```

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

```
print(length(names(mostlyNotNA)))
```

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

Create the model and predict values based on the testSet

```
## Build the model using the caret train function.
## N.B. Based on slide 10 from the random forest lecture from week 3, Cross Validatio
n is handled by the caret train function
controls <- trainControl(method ="cv")
weightModelFit <- train(classe ~., data = trainingSet2, method="rf", ntree=100, trCon
trol = controls, na.action = na.omit)
```

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

```
weightModelFit$finalModel
```

```
##
## Call:
##  randomForest(x = x, y = y, ntree = 100, mtry = param$mtry)
##                Type of random forest: classification
##                      Number of trees: 100
## No. of variables tried at each split: 27
##
##         OOB estimate of  error rate: 0.16%
## Confusion matrix:
##      A    B    C    D    E  class.error
## A 5578    2    0    0    0 0.0003584229
## B    8 3787    2    0    0 0.0026336582
## C    0    5 3417    0    0 0.0014611338
## D    0    0   10 3205    1 0.0034203980
## E    0    1    0    3 3603 0.0011089548
```

```
pred <- predict(weightModelFit,testSet)
print(pred) ## The values I will put into the exercise quiz
```

```
##  [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
```

```
print("Estimated Out of Sample Error")
```

```
## [1] "Estimated Out of Sample Error"
```

```
print(1-weightModelFit$results[2,2])
```

```
## [1] 0.001477877
```

Summary

In summary I described above why I trimmed the original data set. I then used the caret train function with method random forest to generate the model. I passed the cross validation train control so that Cross Validation was handled internally therefore I did not explicitly write code to perform CV. The Out of Sample Error is estimated to be 0.13%. Based on the results I submitted to the quiz I would expect the out of sample error to be extremely low since I got 20/20 correct.