Prediction Project (Machine Learning)

Introduction

This is an attempt to predict how well one of six people performed various barbell exercises. Data was collected using acceleromators on the belt, forearm, arm, and dumbell of the six subjects. The subjects were asked to perform the lifts correctly and incorrectly in 5 different ways. This is given by the “classe” variable in the data set and takes values A-E. I will attempt to predict what “grade” was given to the particular movement based on the accelerometer data provided.

This seemed to be a classification problem, and as such I chose a random forest algorithm for predicting the classe variable rather than a regression. I split the data into a training and test set, trained the algorithm on the training set, then tested it to gauge the out-of-sample performance.

Data Cleaning

For brevity, I won’t show the code to download the data and read.csv.

```
download.file("https://d396qusza40orc.cloudfront.net/predmachlearn/pml-training.csv", des
tfile = "training.csv")
train_data <- read.csv("training.csv")
library(caret)
```

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

I quickly checked the data using some simple summary functions:

```
str(train_data)
```

```
## '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 13230842
32 1323084232 1323084232 1323084232 1323084232 1323084232 ...
##  $ raw_timestamp_part_2    : int  788290 808298 820366 120339 196328 304277 368296 440
390 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 -9
4.4 -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 -0.0
2 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 ...
##  $ 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(train_data))
```

```
##                        X                user_name     raw_timestamp_part_1
##                        0                        0                        0
##     raw_timestamp_part_2          cvtd_timestamp               new_window
##                        0                        0                        0
##               num_window                roll_belt               pitch_belt
##                        0                        0                        0
##                yaw_belt         total_accel_belt       kurtosis_roll_belt
##                        0                        0                        0
##      kurtosis_picth_belt        kurtosis_yaw_belt       skewness_roll_belt
##                        0                        0                        0
##     skewness_roll_belt.1        skewness_yaw_belt            max_roll_belt
##                        0                        0                    19216
##           max_picth_belt             max_yaw_belt            min_roll_belt
##                    19216                        0                    19216
##           min_pitch_belt             min_yaw_belt       amplitude_roll_belt
##                    19216                        0                    19216
##     amplitude_pitch_belt       amplitude_yaw_belt     var_total_accel_belt
##                    19216                        0                    19216
##            avg_roll_belt         stddev_roll_belt            var_roll_belt
##                    19216                    19216                    19216
##           avg_pitch_belt        stddev_pitch_belt           var_pitch_belt
##                    19216                    19216                    19216
##             avg_yaw_belt          stddev_yaw_belt             var_yaw_belt
##                    19216                    19216                    19216
##             gyros_belt_x             gyros_belt_y             gyros_belt_z
##                        0                        0                        0
##             accel_belt_x             accel_belt_y             accel_belt_z
##                        0                        0                        0
##            magnet_belt_x            magnet_belt_y            magnet_belt_z
##                        0                        0                        0
##                roll_arm                 pitch_arm                   yaw_arm
##                        0                        0                        0
##          total_accel_arm            var_accel_arm             avg_roll_arm
##                        0                    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              gyros_arm_x
##                    19216                    19216                        0
##              gyros_arm_y              gyros_arm_z              accel_arm_x
##                        0                        0                        0
##              accel_arm_y              accel_arm_z             magnet_arm_x
##                        0                        0                        0
##              magnet_arm_y             magnet_arm_z        kurtosis_roll_arm
##                        0                        0                        0
##        kurtosis_picth_arm         kurtosis_yaw_arm        skewness_roll_arm
##                        0                        0                        0
##        skewness_pitch_arm         skewness_yaw_arm             max_roll_arm
##                        0                        0                    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            roll_dumbbell
##                    19216                    19216                        0
##           pitch_dumbbell             yaw_dumbbell   kurtosis_roll_dumbbell
##                        0                        0                        0
##  kurtosis_picth_dumbbell    kurtosis_yaw_dumbbell   skewness_roll_dumbbell
##                        0                        0                        0
##  skewness_pitch_dumbbell    skewness_yaw_dumbbell        max_roll_dumbbell
##                        0                        0                    19216
##       max_picth_dumbbell         max_yaw_dumbbell        min_roll_dumbbell
##                    19216                        0                    19216
##       min_pitch_dumbbell         min_yaw_dumbbell  amplitude_roll_dumbbell
##                    19216                        0                    19216
## amplitude_pitch_dumbbell   amplitude_yaw_dumbbell     total_accel_dumbbell
##                    19216                        0                        0
##       var_accel_dumbbell        avg_roll_dumbbell     stddev_roll_dumbbell
##                    19216                    19216                    19216
##        var_roll_dumbbell       avg_pitch_dumbbell    stddev_pitch_dumbbell
##                    19216                    19216                    19216
##       var_pitch_dumbbell         avg_yaw_dumbbell      stddev_yaw_dumbbell
##                    19216                    19216                    19216
##         var_yaw_dumbbell         gyros_dumbbell_x         gyros_dumbbell_y
##                    19216                        0                        0
##         gyros_dumbbell_z         accel_dumbbell_x         accel_dumbbell_y
##                        0                        0                        0
##         accel_dumbbell_z        magnet_dumbbell_x        magnet_dumbbell_y
##                        0                        0                        0
##        magnet_dumbbell_z             roll_forearm            pitch_forearm
##                        0                        0                        0
##              yaw_forearm    kurtosis_roll_forearm   kurtosis_picth_forearm
##                        0                        0                        0
##     kurtosis_yaw_forearm    skewness_roll_forearm   skewness_pitch_forearm
##                        0                        0                        0
##     skewness_yaw_forearm         max_roll_forearm        max_picth_forearm
##                        0                    19216                    19216
##          max_yaw_forearm         min_roll_forearm        min_pitch_forearm
##                        0                    19216                    19216
##          min_yaw_forearm   amplitude_roll_forearm  amplitude_pitch_forearm
##                        0                    19216                    19216
##    amplitude_yaw_forearm      total_accel_forearm        var_accel_forearm
##                        0                        0                    19216
##         avg_roll_forearm      stddev_roll_forearm         var_roll_forearm
##                    19216                    19216                    19216
##        avg_pitch_forearm     stddev_pitch_forearm        var_pitch_forearm
##                    19216                    19216                    19216
##          avg_yaw_forearm       stddev_yaw_forearm          var_yaw_forearm
##                    19216                    19216                    19216
```

```
##          gyros_forearm_x          gyros_forearm_y          gyros_forearm_z
##                        0                        0                        0
##          accel_forearm_x          accel_forearm_y          accel_forearm_z
##                        0                        0                        0
##         magnet_forearm_x         magnet_forearm_y         magnet_forearm_z
##                        0                        0                        0
##                   classe
##                        0
```

There seems to be a bunch of variables that are factor variables or have an excessive amount of NAs. Additionally, a few of the variables seem to be a lot of very similar values repeated (low variance).

How to deal with the very low variance predictors is not obvious. I elected to eliminate all of them for simplicity, althought this is not always optimal, given that even these variables can contain valuable information for the model to use. That said, after looking over the data I felt I'd still get a very accurate model without trying to figure out whether or not I could improve a little by retaining one or two of the low variance predictors.

The following steps address these issues by removing low variance predictors, any predictor with more than 5 NA's, and any non-numeric variables.

```
set.seed(1001)
inTrain <- createDataPartition(y=train_data$classe, p=.75, list=FALSE)
training <- train_data[inTrain,]
testing <- train_data[-inTrain,]

df <- training[,-nearZeroVar(training)]
na1 <- apply(df, 2, function(x)sum(is.na(x))<5)
clean_train <- df[,na1]
clean_train2 <- clean_train
clean_train2$user_name <- NULL
clean_train2$raw_timestamp_part_1 <- NULL
clean_train2$raw_timestamp_part_2 <- NULL
clean_train2$cvtd_timestamp <- NULL
clean_train2$new_window <- NULL
clean_train2$X <- NULL
clean_train2$num_window <- NULL
```

This leaves us with 52 predictors, all of which are movement characteristics that are either of class "integer" or "numeric", and none of which are low variance.

Model Training

I now train a random forest algorithm on the above cleaned and processed data set. I selected to preprocess the data using a Principle Component Analysis. Given that the remaining predictors are all different movement patterns, a weighted combination of them seemed to make sense, making a PCA appropriate.

```
rf10 <- train(classe~., preProcess="pca", method="rf", ntree=250, data=clean_train2)
```

```
## 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
```

```
## Warning in randomForest.default(x, y, mtry = param$mtry, ...): invalid
## mtry: reset to within valid range
```

```
## Warning in randomForest.default(x, y, mtry = param$mtry, ...): invalid
## mtry: reset to within valid range
```

```
## Warning in randomForest.default(x, y, mtry = param$mtry, ...): invalid
## mtry: reset to within valid range
```

```
## Warning in randomForest.default(x, y, mtry = param$mtry, ...): invalid
## mtry: reset to within valid range
```

```
## Warning in randomForest.default(x, y, mtry = param$mtry, ...): invalid
## mtry: reset to within valid range
```

```
## Warning in randomForest.default(x, y, mtry = param$mtry, ...): invalid
## mtry: reset to within valid range
```

```
## Warning in randomForest.default(x, y, mtry = param$mtry, ...): invalid
## mtry: reset to within valid range
```

```
## Warning in randomForest.default(x, y, mtry = param$mtry, ...): invalid
## mtry: reset to within valid range
```

```
## Warning in randomForest.default(x, y, mtry = param$mtry, ...): invalid
## mtry: reset to within valid range
```

```
## Warning in randomForest.default(x, y, mtry = param$mtry, ...): invalid
## mtry: reset to within valid range
```

```
## Warning in randomForest.default(x, y, mtry = param$mtry, ...): invalid
## mtry: reset to within valid range
```

```
## Warning in randomForest.default(x, y, mtry = param$mtry, ...): invalid
## mtry: reset to within valid range
```

```
## Warning in randomForest.default(x, y, mtry = param$mtry, ...): invalid
## mtry: reset to within valid range
```

```
## Warning in randomForest.default(x, y, mtry = param$mtry, ...): invalid
## mtry: reset to within valid range
```

```
## Warning in randomForest.default(x, y, mtry = param$mtry, ...): invalid
## mtry: reset to within valid range
```

```
## Warning in randomForest.default(x, y, mtry = param$mtry, ...): invalid
## mtry: reset to within valid range
```

```
## Warning in randomForest.default(x, y, mtry = param$mtry, ...): invalid
## mtry: reset to within valid range
```

```
## Warning in randomForest.default(x, y, mtry = param$mtry, ...): invalid
## mtry: reset to within valid range
```

```
## Warning in randomForest.default(x, y, mtry = param$mtry, ...): invalid
## mtry: reset to within valid range
```

```
## Warning in randomForest.default(x, y, mtry = param$mtry, ...): invalid
## mtry: reset to within valid range
```

```
## Warning in randomForest.default(x, y, mtry = param$mtry, ...): invalid
## mtry: reset to within valid range
```

```
## Warning in randomForest.default(x, y, mtry = param$mtry, ...): invalid
## mtry: reset to within valid range
```

```
## Warning in randomForest.default(x, y, mtry = param$mtry, ...): invalid
## mtry: reset to within valid range
```

```
## Warning in randomForest.default(x, y, mtry = param$mtry, ...): invalid
## mtry: reset to within valid range
```

```
## Warning in randomForest.default(x, y, mtry = param$mtry, ...): invalid
## mtry: reset to within valid range
```

```
## Warning in randomForest.default(x, y, mtry = param$mtry, ...): invalid
## mtry: reset to within valid range
```

```
## Warning in randomForest.default(x, y, mtry = param$mtry, ...): invalid
## mtry: reset to within valid range
```

```
## Warning in randomForest.default(x, y, mtry = param$mtry, ...): invalid
## mtry: reset to within valid range
```

```
## Warning in randomForest.default(x, y, mtry = param$mtry, ...): invalid
## mtry: reset to within valid range
```

```
## Warning in randomForest.default(x, y, mtry = param$mtry, ...): invalid
## mtry: reset to within valid range
```

```
## Warning in randomForest.default(x, y, mtry = param$mtry, ...): invalid
## mtry: reset to within valid range
```

```
## Warning in randomForest.default(x, y, mtry = param$mtry, ...): invalid
## mtry: reset to within valid range
```

```
## Warning in randomForest.default(x, y, mtry = param$mtry, ...): invalid
## mtry: reset to within valid range
```

```
## Warning in randomForest.default(x, y, mtry = param$mtry, ...): invalid
## mtry: reset to within valid range
```

```
## Warning in randomForest.default(x, y, mtry = param$mtry, ...): invalid
## mtry: reset to within valid range
```

```
## Warning in randomForest.default(x, y, mtry = param$mtry, ...): invalid
## mtry: reset to within valid range
```

```
## Warning in randomForest.default(x, y, mtry = param$mtry, ...): invalid
## mtry: reset to within valid range
```

```
## Warning in randomForest.default(x, y, mtry = param$mtry, ...): invalid
## mtry: reset to within valid range
```

```
## Warning in randomForest.default(x, y, mtry = param$mtry, ...): invalid
## mtry: reset to within valid range
```

```
## Warning in randomForest.default(x, y, mtry = param$mtry, ...): invalid
## mtry: reset to within valid range
```

```
## Warning in randomForest.default(x, y, mtry = param$mtry, ...): invalid
## mtry: reset to within valid range
```

```
## Warning in randomForest.default(x, y, mtry = param$mtry, ...): invalid
## mtry: reset to within valid range
```

```
## Warning in randomForest.default(x, y, mtry = param$mtry, ...): invalid
## mtry: reset to within valid range
```

```
## Warning in randomForest.default(x, y, mtry = param$mtry, ...): invalid
## mtry: reset to within valid range
```

```
## Warning in randomForest.default(x, y, mtry = param$mtry, ...): invalid
## mtry: reset to within valid range
```

```
## Warning in randomForest.default(x, y, mtry = param$mtry, ...): invalid
## mtry: reset to within valid range
```

```
## Warning in randomForest.default(x, y, mtry = param$mtry, ...): invalid
## mtry: reset to within valid range
```

```
## Warning in randomForest.default(x, y, mtry = param$mtry, ...): invalid
## mtry: reset to within valid range
```

```
print(rf10$finalModel)
```

```
##
## Call:
##  randomForest(x = x, y = y, ntree = 250, mtry = param$mtry)
##                Type of random forest: classification
##                      Number of trees: 250
## No. of variables tried at each split: 2
##
##         OOB estimate of  error rate: 2.58%
## Confusion matrix:
##      A    B    C    D    E class.error
## A 4146   18   10    9    2 0.009318996
## B   39 2755   48    3    3 0.032654494
## C    5   49 2474   33    6 0.036229061
## D    3    2   95 2308    4 0.043117745
## E    2   11   18   19 2656 0.018477458
```

I set ntree=250 to cut down on computing time, even though this will sacrifice a little accuracy over the default of 500 trees.

The OOB estimate of error is 2.45%. Thish is probably generous. Below is the confusion matrix for my testing data. The error rate is about .026 (2.6%), so just north of the estimate provided by R.

```
pred <- predict(rf10, newdata = testing)
testing$predright <- pred==testing$classe
table(pred, testing$classe)
```

```
##
## pred    A    B    C    D    E
##    A 1385   11    0    1    0
##    B    3  927    8    0    1
##    C    5    8  839   25   13
##    D    1    1    8  777    6
##    E    1    2    0    1  881
```

```
error_rate <- pred==testing$classe
table(error_rate)
```

```
## error_rate
## FALSE  TRUE
##    95  4809
```

```
127/4904
```

```
## [1] 0.02589723
```

In general, the model appears to do pretty well with the A and E values, but struggles a little more to separate between B/C/D.