Project: Practical Machine Learning, Prediction

Laura Snyder

March 31, 2019

Background

Using devices such as Jawbone Up, Nike FuelBand, and Fitbit it is now possible to collect a large amount of data about personal activity relatively inexpensively. These type of devices are part of the quantified self movement â□□ a group of enthusiasts who take measurements about themselves regularly to improve their health, to find patterns in their behavior, or because they are tech geeks. One thing that people regularly do is quantify how much of a particular activity they do, but they rarely quantify how well they do it. In this project, your goal will be to use data from accelerometers on the belt, forearm, arm, and dumbell of 6 participants. They were asked to perform barbell lifts correctly and incorrectly in 5 different ways.

Submission

The goal of the project is to predict the manner in which group members did the exercise, the “classe” variable in the training set. Description will include: 1) how model was built; 2) cross validation approach; 3) theexpected out of sample error; and 4) rationale for choices.

Prepare the Environment and Import the Data

Set working directory and load all anticipated library items for machine learning analysis.

```
setwd("~/R/Machine Learning/Project")

library(caret)
```

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

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

```
library(rpart)
library(rpart.plot)
```

```
## Warning: package 'rpart.plot' was built under R version 3.5.3
```

```
library(randomForest)
```

```
## Warning: package 'randomForest' was built under R version 3.5.3
```

```
## randomForest 4.6-14
```

```
## Type rfNews() to see new features/changes/bug fixes.
```

```
##
## Attaching package: 'randomForest'
```

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

```
library(rattle)
```

```
## Warning: package 'rattle' was built under R version 3.5.3
```

```
## Rattle: A free graphical interface for data science with R.
## Version 5.2.0 Copyright (c) 2006-2018 Togaware Pty Ltd.
## Type 'rattle()' to shake, rattle, and roll your data.
```

```
##
## Attaching package: 'rattle'
```

```
## The following object is masked from 'package:randomForest':
##
##     importance
```

```
library(RColorBrewer)
library(corrplot)
```

```
## Warning: package 'corrplot' was built under R version 3.5.3
```

```
## corrplot 0.84 loaded
```

```
library(gbm)
```

```
## Warning: package 'gbm' was built under R version 3.5.3
```

```
## Loaded gbm 2.1.5
```

Capture the csv training and testing sets from the URLs provided.

```
training <- read.csv(url("https://d396qusza40orc.cloudfront.net/predmachlearn/pml-training.csv"),header=TRUE)
testing <- read.csv(url("https://d396qusza40orc.cloudfront.net/predmachlearn/pml-testing.csv"),header=TRUE)
```

Explore, Clean and Reduce the Data

Take a look at the data.

```
dim(training)
```

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

```
dim(testing)
```

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

```
head(training)
```

```
##   X user_name raw_timestamp_part_1 raw_timestamp_part_2   cvtd_timestamp
## 1 1  carlitos           1323084231               788290 05/12/2011 11:23
## 2 2  carlitos           1323084231               808298 05/12/2011 11:23
## 3 3  carlitos           1323084231               820366 05/12/2011 11:23
## 4 4  carlitos           1323084232               120339 05/12/2011 11:23
## 5 5  carlitos           1323084232               196328 05/12/2011 11:23
## 6 6  carlitos           1323084232               304277 05/12/2011 11:23
##   new_window num_window roll_belt pitch_belt yaw_belt total_accel_belt
## 1         no         11      1.41       8.07    -94.4                3
## 2         no         11      1.41       8.07    -94.4                3
## 3         no         11      1.42       8.07    -94.4                3
## 4         no         12      1.48       8.05    -94.4                3
## 5         no         12      1.48       8.07    -94.4                3
## 6         no         12      1.45       8.06    -94.4                3
##   kurtosis_roll_belt kurtosis_picth_belt kurtosis_yaw_belt
## 1
## 2
## 3
## 4
## 5
## 6
##   skewness_roll_belt skewness_roll_belt.1 skewness_yaw_belt max_roll_belt
## 1                                                                      NA
## 2                                                                      NA
## 3                                                                      NA
## 4                                                                      NA
## 5                                                                      NA
## 6                                                                      NA
##   max_picth_belt max_yaw_belt min_roll_belt min_pitch_belt min_yaw_belt
## 1             NA                         NA             NA
## 2             NA                         NA             NA
## 3             NA                         NA             NA
## 4             NA                         NA             NA
## 5             NA                         NA             NA
## 6             NA                         NA             NA
##   amplitude_roll_belt amplitude_pitch_belt amplitude_yaw_belt
## 1                  NA                   NA
## 2                  NA                   NA
## 3                  NA                   NA
## 4                  NA                   NA
## 5                  NA                   NA
## 6                  NA                   NA
##   var_total_accel_belt avg_roll_belt stddev_roll_belt var_roll_belt
## 1                   NA            NA               NA            NA
## 2                   NA            NA               NA            NA
## 3                   NA            NA               NA            NA
## 4                   NA            NA               NA            NA
## 5                   NA            NA               NA            NA
## 6                   NA            NA               NA            NA
##   avg_pitch_belt stddev_pitch_belt var_pitch_belt avg_yaw_belt
## 1             NA                NA             NA           NA
## 2             NA                NA             NA           NA
## 3             NA                NA             NA           NA
## 4             NA                NA             NA           NA
## 5             NA                NA             NA           NA
## 6             NA                NA             NA           NA
##   stddev_yaw_belt var_yaw_belt gyros_belt_x gyros_belt_y gyros_belt_z
## 1              NA           NA         0.00         0.00        -0.02
## 2              NA           NA         0.02         0.00        -0.02
## 3              NA           NA         0.00         0.00        -0.02
```

```
## 4              NA           NA         0.02         0.00        -0.03
## 5              NA           NA         0.02         0.02        -0.02
## 6              NA           NA         0.02         0.00        -0.02
##   accel_belt_x accel_belt_y accel_belt_z magnet_belt_x magnet_belt_y
## 1          -21            4           22            -3           599
## 2          -22            4           22            -7           608
## 3          -20            5           23            -2           600
## 4          -22            3           21            -6           604
## 5          -21            2           24            -6           600
## 6          -21            4           21             0           603
##   magnet_belt_z roll_arm pitch_arm yaw_arm total_accel_arm var_accel_arm
## 1          -313     -128      22.5    -161              34            NA
## 2          -311     -128      22.5    -161              34            NA
## 3          -305     -128      22.5    -161              34            NA
## 4          -310     -128      22.1    -161              34            NA
## 5          -302     -128      22.1    -161              34            NA
## 6          -312     -128      22.0    -161              34            NA
##   avg_roll_arm stddev_roll_arm var_roll_arm avg_pitch_arm stddev_pitch_arm
## 1           NA              NA           NA            NA               NA
## 2           NA              NA           NA            NA               NA
## 3           NA              NA           NA            NA               NA
## 4           NA              NA           NA            NA               NA
## 5           NA              NA           NA            NA               NA
## 6           NA              NA           NA            NA               NA
##   var_pitch_arm avg_yaw_arm stddev_yaw_arm var_yaw_arm gyros_arm_x
## 1            NA          NA             NA          NA        0.00
## 2            NA          NA             NA          NA        0.02
## 3            NA          NA             NA          NA        0.02
## 4            NA          NA             NA          NA        0.02
## 5            NA          NA             NA          NA        0.00
## 6            NA          NA             NA          NA        0.02
##   gyros_arm_y gyros_arm_z accel_arm_x accel_arm_y accel_arm_z magnet_arm_x
## 1        0.00       -0.02        -288         109        -123         -368
## 2       -0.02       -0.02        -290         110        -125         -369
## 3       -0.02       -0.02        -289         110        -126         -368
## 4       -0.03        0.02        -289         111        -123         -372
## 5       -0.03        0.00        -289         111        -123         -374
## 6       -0.03        0.00        -289         111        -122         -369
##   magnet_arm_y magnet_arm_z kurtosis_roll_arm kurtosis_picth_arm
## 1          337          516
## 2          337          513
## 3          344          513
## 4          344          512
## 5          337          506
## 6          342          513
##   kurtosis_yaw_arm skewness_roll_arm skewness_pitch_arm skewness_yaw_arm
## 1
## 2
## 3
## 4
## 5
## 6
##   max_roll_arm max_picth_arm max_yaw_arm min_roll_arm min_pitch_arm
## 1           NA            NA          NA           NA            NA
## 2           NA            NA          NA           NA            NA
## 3           NA            NA          NA           NA            NA
## 4           NA            NA          NA           NA            NA
## 5           NA            NA          NA           NA            NA
## 6           NA            NA          NA           NA            NA
##   min_yaw_arm amplitude_roll_arm amplitude_pitch_arm amplitude_yaw_arm
## 1          NA                 NA                  NA                NA
```

```
## 2          NA                NA                 NA                NA
## 3          NA                NA                 NA                NA
## 4          NA                NA                 NA                NA
## 5          NA                NA                 NA                NA
## 6          NA                NA                 NA                NA
##   roll_dumbbell pitch_dumbbell yaw_dumbbell kurtosis_roll_dumbbell
## 1      13.05217      -70.49400    -84.87394
## 2      13.13074      -70.63751    -84.71065
## 3      12.85075      -70.27812    -85.14078
## 4      13.43120      -70.39379    -84.87363
## 5      13.37872      -70.42856    -84.85306
## 6      13.38246      -70.81759    -84.46500
##   kurtosis_picth_dumbbell kurtosis_yaw_dumbbell skewness_roll_dumbbell
## 1
## 2
## 3
## 4
## 5
## 6
##   skewness_pitch_dumbbell skewness_yaw_dumbbell max_roll_dumbbell
## 1                                                              NA
## 2                                                              NA
## 3                                                              NA
## 4                                                              NA
## 5                                                              NA
## 6                                                              NA
##   max_picth_dumbbell max_yaw_dumbbell min_roll_dumbbell min_pitch_dumbbell
## 1                 NA                                 NA                 NA
## 2                 NA                                 NA                 NA
## 3                 NA                                 NA                 NA
## 4                 NA                                 NA                 NA
## 5                 NA                                 NA                 NA
## 6                 NA                                 NA                 NA
##   min_yaw_dumbbell amplitude_roll_dumbbell amplitude_pitch_dumbbell
## 1                                       NA                      NA
## 2                                       NA                      NA
## 3                                       NA                      NA
## 4                                       NA                      NA
## 5                                       NA                      NA
## 6                                       NA                      NA
##   amplitude_yaw_dumbbell total_accel_dumbbell var_accel_dumbbell
## 1                                          37                 NA
## 2                                          37                 NA
## 3                                          37                 NA
## 4                                          37                 NA
## 5                                          37                 NA
## 6                                          37                 NA
##   avg_roll_dumbbell stddev_roll_dumbbell var_roll_dumbbell
## 1                NA                   NA                NA
## 2                NA                   NA                NA
## 3                NA                   NA                NA
## 4                NA                   NA                NA
## 5                NA                   NA                NA
## 6                NA                   NA                NA
##   avg_pitch_dumbbell stddev_pitch_dumbbell var_pitch_dumbbell
## 1                 NA                    NA                 NA
## 2                 NA                    NA                 NA
## 3                 NA                    NA                 NA
## 4                 NA                    NA                 NA
## 5                 NA                    NA                 NA
## 6                 NA                    NA                 NA
```

```
##   avg_yaw_dumbbell stddev_yaw_dumbbell var_yaw_dumbbell gyros_dumbbell_x
## 1               NA                  NA               NA                0
## 2               NA                  NA               NA                0
## 3               NA                  NA               NA                0
## 4               NA                  NA               NA                0
## 5               NA                  NA               NA                0
## 6               NA                  NA               NA                0
##   gyros_dumbbell_y gyros_dumbbell_z accel_dumbbell_x accel_dumbbell_y
## 1            -0.02             0.00             -234               47
## 2            -0.02             0.00             -233               47
## 3            -0.02             0.00             -232               46
## 4            -0.02            -0.02             -232               48
## 5            -0.02             0.00             -233               48
## 6            -0.02             0.00             -234               48
##   accel_dumbbell_z magnet_dumbbell_x magnet_dumbbell_y magnet_dumbbell_z
## 1             -271              -559               293               -65
## 2             -269              -555               296               -64
## 3             -270              -561               298               -63
## 4             -269              -552               303               -60
## 5             -270              -554               292               -68
## 6             -269              -558               294               -66
##   roll_forearm pitch_forearm yaw_forearm kurtosis_roll_forearm
## 1         28.4         -63.9        -153
## 2         28.3         -63.9        -153
## 3         28.3         -63.9        -152
## 4         28.1         -63.9        -152
## 5         28.0         -63.9        -152
## 6         27.9         -63.9        -152
##   kurtosis_picth_forearm kurtosis_yaw_forearm skewness_roll_forearm
## 1
## 2
## 3
## 4
## 5
## 6
##   skewness_pitch_forearm skewness_yaw_forearm max_roll_forearm
## 1                                                           NA
## 2                                                           NA
## 3                                                           NA
## 4                                                           NA
## 5                                                           NA
## 6                                                           NA
##   max_picth_forearm max_yaw_forearm min_roll_forearm min_pitch_forearm
## 1                NA                                NA                NA
## 2                NA                                NA                NA
## 3                NA                                NA                NA
## 4                NA                                NA                NA
## 5                NA                                NA                NA
## 6                NA                                NA                NA
##   min_yaw_forearm amplitude_roll_forearm amplitude_pitch_forearm
## 1                                      NA                     NA
## 2                                      NA                     NA
## 3                                      NA                     NA
## 4                                      NA                     NA
## 5                                      NA                     NA
## 6                                      NA                     NA
##   amplitude_yaw_forearm total_accel_forearm var_accel_forearm
## 1                                        36                NA
## 2                                        36                NA
## 3                                        36                NA
## 4                                        36                NA
```

```
## 5                                    36               NA
## 6                                    36               NA
##   avg_roll_forearm stddev_roll_forearm var_roll_forearm avg_pitch_forearm
## 1               NA                  NA               NA                NA
## 2               NA                  NA               NA                NA
## 3               NA                  NA               NA                NA
## 4               NA                  NA               NA                NA
## 5               NA                  NA               NA                NA
## 6               NA                  NA               NA                NA
##   stddev_pitch_forearm var_pitch_forearm avg_yaw_forearm
## 1                   NA                NA              NA
## 2                   NA                NA              NA
## 3                   NA                NA              NA
## 4                   NA                NA              NA
## 5                   NA                NA              NA
## 6                   NA                NA              NA
##   stddev_yaw_forearm var_yaw_forearm gyros_forearm_x gyros_forearm_y
## 1                 NA              NA            0.03            0.00
## 2                 NA              NA            0.02            0.00
## 3                 NA              NA            0.03           -0.02
## 4                 NA              NA            0.02           -0.02
## 5                 NA              NA            0.02            0.00
## 6                 NA              NA            0.02           -0.02
##   gyros_forearm_z accel_forearm_x accel_forearm_y accel_forearm_z
## 1           -0.02             192             203            -215
## 2           -0.02             192             203            -216
## 3            0.00             196             204            -213
## 4            0.00             189             206            -214
## 5           -0.02             189             206            -214
## 6           -0.03             193             203            -215
##   magnet_forearm_x magnet_forearm_y magnet_forearm_z classe
## 1              -17              654              476      A
## 2              -18              661              473      A
## 3              -18              658              469      A
## 4              -16              658              469      A
## 5              -17              655              473      A
## 6               -9              660              478      A
```

The training set is 19,622 rows of observations across 160 variables. The testing set is 20 rows of observactions across 160 variables. The numbers of variables is higher than is needed to build a predictive model. The first 7 columns are removed as they do not contain data that would be included in a predictive model (such as row number, name or time stamp). Further, some of the columns have many NAs. These should be removed to make the building of the model more manageable.

```
trainingred<- training[, colSums(is.na(training)) == 0]
testingred <- testing[, colSums(is.na(testing)) == 0]
trainingred<-trainingred[,-c(1:7)]
testingred<-testingred[,-c(1:7)]
```

```
dim(trainingred)
```

```
## [1] 19622    86
```

```
dim(testingred)
```

```
## [1] 20 53
```

The effect is that now the training set is reduced to 86 columns, and the testing set to 53 columns.

The goal of this project is to predict the manner in which exercise is done, as it relates to the “classe” variable in the training set. The training data will be split into 70% training data and 30% validation data (to determine out of sample errors) for the purpose of building the model. The original testing data will be reserved for the later set of 20 questions to test the final model.

Split the Training Data into Training and Validation Subsets

Splitting the data set into training and validation subsets will allow cross-validation. That is, to say, that a model built with the training data can later be used with the reserved validation data to ensure it performs as expected. A seed will be set to allow reproducibility.

```
set.seed(1234)
inTrain <- createDataPartition(trainingred$classe, p = 0.7, list = FALSE)
trainData <- trainingred[inTrain, ]
testData <- trainingred[-inTrain, ]
```

```
dim(trainData)
```

```
## [1] 13737    86
```

```
dim(testData)
```

```
## [1] 5885   86
```

The training set for the development of the model is 13,737 rows long, and the testing set for validating the model is 5,885 rows. Still, the possible column predictors can be reduced further. nearZeroVar diagnoses predictors that have one unique value (i.e. are zero variance predictors) or predictors that have very few unique values relative to the number of observations.

nearZeroVar diagnoses predictors that have one unique value (i.e. are zero variance predictors) or predictors that are have both of the following characteristics: they have very few unique values relative to the number of samples and the ratio of the frequency of the most common value to the frequency of the second most common value is large. This results in bringing down the number of columns to 53 in each set. Three models will be build from the test set and compared; the best model will be used.

```
r  zeros <- nearZeroVar(trainData)  trainData <- trainData[, -zeros]  testData  <- testData[, -zeros]  dim(trainData)
```

```
## [1] 13737    53
```

```
r  dim(testData)
```

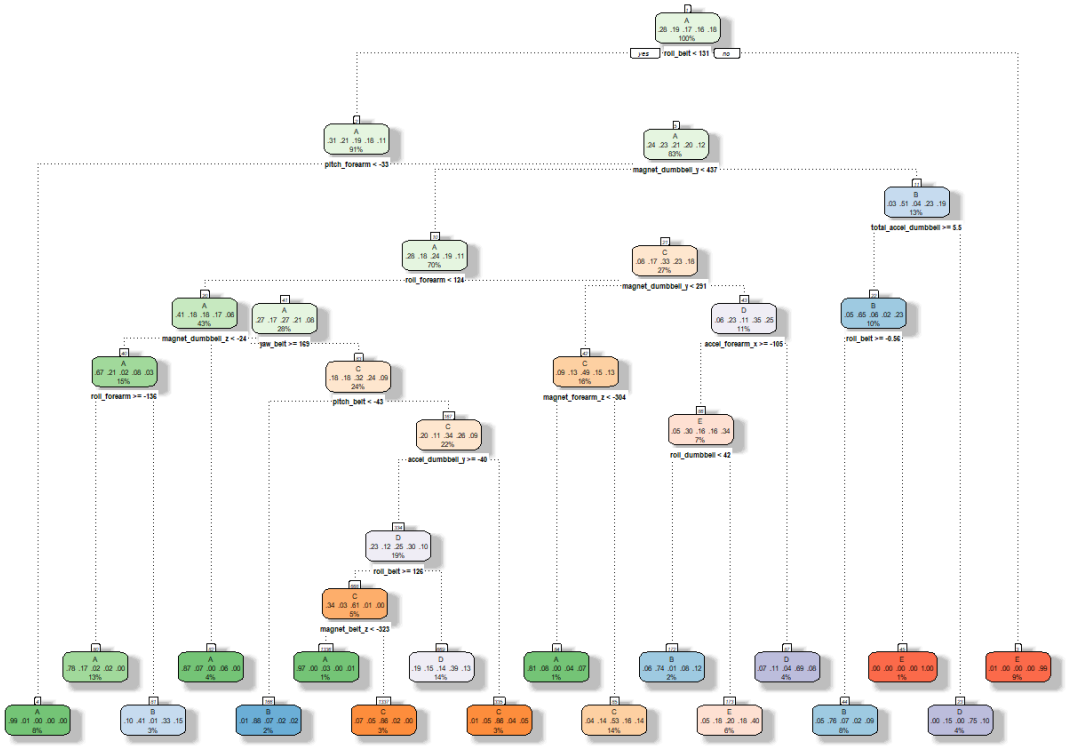
```
## [1] 5885   53
```

Build and Compare Prediction Models

Classification Tree

Start by building a classification tree.

```
set.seed(1234)
decisionTreeMod1 <- rpart(classe ~ ., data=trainData, method="class")
fancyRpartPlot(decisionTreeMod1)
```



Rattle 2019-Mar-31 14:28:54 Laura

Use the model on the test data to see how well it works.

```
predictTreeMod1 <- predict(decisionTreeMod1, testData, type = "class")
cm1 <- confusionMatrix(predictTreeMod1, testData$classe)
cm1
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction    A    B    C    D    E
##          A 1364  169   24   48   16
##          B   60  581   46   79   74
##          C   52  137  765  129  145
##          D  183  194  125  650  159
##          E   15   58   66   58  688
##
## Overall Statistics
##
##                Accuracy : 0.6879
##                  95% CI : (0.6758, 0.6997)
##     No Information Rate : 0.2845
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.6066
##  Mcnemar's Test P-Value : < 2.2e-16
##
## Statistics by Class:
##
##                      Class: A Class: B Class: C Class: D Class: E
## Sensitivity            0.8148  0.51010   0.7456   0.6743   0.6359
## Specificity            0.9390  0.94543   0.9047   0.8657   0.9590
## Pos Pred Value         0.8415  0.69167   0.6230   0.4958   0.7774
## Neg Pred Value         0.9273  0.88940   0.9440   0.9314   0.9212
## Prevalence             0.2845  0.19354   0.1743   0.1638   0.1839
## Detection Rate         0.2318  0.09873   0.1300   0.1105   0.1169
## Detection Prevalence   0.2754  0.14274   0.2087   0.2228   0.1504
## Balanced Accuracy      0.8769  0.72776   0.8252   0.7700   0.7974
```

The confustion matrix show a 68.79% accuracy rate. The out of sample error is about 31%. The model does not appear to be very accurate.

GBM

The next model attempted is Generalized Boosted Model (GBM). This model is also a supervised machine learning method, which basically compounds weak predictors together to form a better overall predictive model. Start with setting the seed, and then training the model. The GBM performs 150 iterations. There were 52 predictors of which 52 had non-zero influence (the 53rd item, classe, is the outcome being predicted and consequently is not used). The created model is then used to predict the outcome with the test data. A confusion matrix summarized the results, showing a 96.28% accuracy rate. The out of sample error is about 4%. This appears to be a very accurate model.

```
set.seed(1234)
controlGBM <- trainControl(method = "repeatedcv", number = 5, repeats = 1)
modGBM  <- train(classe ~ ., data=trainData, method = "gbm", trControl = controlGBM, verbose = FALSE)
modGBM$finalModel
```

```
## A gradient boosted model with multinomial loss function.
## 150 iterations were performed.
## There were 52 predictors of which 52 had non-zero influence.
```

```
predictGBM <- predict(modGBM, newdata=testData)
cmGBM <- confusionMatrix(predictGBM, testData$classe)
cmGBM
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction    A    B    C    D    E
##          A 1653   46    0    0    1
##          B   13 1064   35    5    7
##          C    4   26  976   19   14
##          D    3    2   12  931   18
##          E    1    1    3    9 1042
##
## Overall Statistics
##
##                Accuracy : 0.9628
##                  95% CI : (0.9576, 0.9675)
##     No Information Rate : 0.2845
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.9529
##  Mcnemar's Test P-Value : 2.944e-06
##
## Statistics by Class:
##
##                      Class: A Class: B Class: C Class: D Class: E
## Sensitivity            0.9875   0.9342   0.9513   0.9658   0.9630
## Specificity            0.9888   0.9874   0.9870   0.9929   0.9971
## Pos Pred Value         0.9724   0.9466   0.9394   0.9638   0.9867
## Neg Pred Value         0.9950   0.9842   0.9897   0.9933   0.9917
## Prevalence             0.2845   0.1935   0.1743   0.1638   0.1839
## Detection Rate         0.2809   0.1808   0.1658   0.1582   0.1771
## Detection Prevalence   0.2889   0.1910   0.1766   0.1641   0.1794
## Balanced Accuracy      0.9881   0.9608   0.9692   0.9793   0.9801
```

Random Forest

Finally, Random Forest is a supervised machine learning algorithm that builds multiple decision trees, then merges them together for an accurate and stable predicition. Start by training the model with trainControl. 500 trees are built with 27 variables tried at each split. A 56% error rate is found. Using the model to predit the outcome with the test data, the accuracy of the model is 99. 44%. The out of sample error is less than 1%. This suggests a very accurate model.

```
controlRF <- trainControl(method="cv", number=3, verboseIter=FALSE)
modRF <- train(classe ~ ., data=trainData, method="rf", trControl=controlRF)
modRF$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: 0.56%
## Confusion matrix:
##      A    B    C    D    E  class.error
## A 3903    3    0    0    0 0.0007680492
## B   16 2637    5    0    0 0.0079006772
## C    0    9 2381    6    0 0.0062604341
## D    0    1   23 2226    2 0.0115452931
## E    0    2    3    7 2513 0.0047524752
```

```
predictRF<- predict(modRF, newdata=testData)
cmRF <- confusionMatrix(predictRF, testData$classe)
cmRF
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction    A    B    C    D    E
##          A 1674   10    0    0    0
##          B    0 1128    4    1    1
##          C    0    1 1018    6    2
##          D    0    0    4  956    3
##          E    0    0    0    1 1076
##
## Overall Statistics
##
##                Accuracy : 0.9944
##                  95% CI : (0.9921, 0.9961)
##     No Information Rate : 0.2845
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.9929
##  Mcnemar's Test P-Value : NA
##
## Statistics by Class:
##
##                      Class: A Class: B Class: C Class: D Class: E
## Sensitivity            1.0000   0.9903   0.9922   0.9917   0.9945
## Specificity            0.9976   0.9987   0.9981   0.9986   0.9998
## Pos Pred Value         0.9941   0.9947   0.9912   0.9927   0.9991
## Neg Pred Value         1.0000   0.9977   0.9984   0.9984   0.9988
## Prevalence             0.2845   0.1935   0.1743   0.1638   0.1839
## Detection Rate         0.2845   0.1917   0.1730   0.1624   0.1828
## Detection Prevalence   0.2862   0.1927   0.1745   0.1636   0.1830
## Balanced Accuracy      0.9988   0.9945   0.9952   0.9951   0.9971
```

Choose Model and Run the Test Data

Both the GBM and RF models seems very accurate, so accurate that they could be overfitted. The RF model will be used. An additional drawback is the length of time that it takes to run. Using the RF model, the test data is run to prepare for the 20 question follow up quiz.

```
(predict(modRF, testing))
```

```
##  [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
```

[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