PML Weight Lifting Quality Predictions

Derek Wilson

June 2, 2016

For the purpose of this exercise and running the scripts below for reproducability:

- it is assumed that one has downloaded the two required data sets, training (https://d396qusza40orc.cloudfront.net/predmachlearn/pml-training.csv) and testing (https://d396qusza40orc.cloudfront.net/predmachlearn/pml-testing.csv)
- the two files reside in the current working directory, set with *setwd()*
- the downloaded 'testing' data set is refer to as the 'predict' set given it contains no indicated classes and not suitable for testing models

Establish our envirnment.

- Load required packages.
- Set a date class for reading any data text fields.
- Define a function to read the csv files into data frames.

Data is loaded into memory. Columns with large amounts of missing data and those columns not consequential to our research are removed.

The traing data set is large enough to accomodate a testing data set.

Models

We will create and test three models. 1. Random Forest 2. Classification and Regression Tree (CART) 3. Gradient Boosting Machine (GBM)

After building each model we apply the testing data set to build a confusion matrix and assess accuracy and out-of-sample error.

We optimize performance, also in subseqent models, by setting the *trControl* parameters for utilizing any installed parallel backend and limiting cross-validation resampling to 4 iterations.

We then summarize the accuracy and out-of-sample error for comparison.

Model 1 - Random Forest

```
## Confusion Matrix and Statistics
## 
##           Reference
## Prediction    A    B    C    D    E
##          A 2231    0    0    1    0
##          B   10 1502    6    0    0
##          C    0   13 1353    2    0
##          D    0    0   19 1265    2
##          E    0    0    2    2 1438
## 
## Overall Statistics
##                                           
##                Accuracy : 0.9927          
##                  95% CI : (0.9906, 0.9945)
##     No Information Rate : 0.2856          
##     P-Value [Acc > NIR] : < 2.2e-16       
##                                           
##                   Kappa : 0.9908          
##  Mcnemar's Test P-Value : NA              
## 
## Statistics by Class:
## 
##                      Class: A Class: B Class: C Class: D Class: E
## Sensitivity            0.9955   0.9914   0.9804   0.9961   0.9986
## Specificity            0.9998   0.9975   0.9977   0.9968   0.9994
## Pos Pred Value         0.9996   0.9895   0.9890   0.9837   0.9972
## Neg Pred Value         0.9982   0.9979   0.9958   0.9992   0.9997
## Prevalence             0.2856   0.1931   0.1759   0.1619   0.1835
## Detection Rate         0.2843   0.1914   0.1724   0.1612   0.1833
## Detection Prevalence   0.2845   0.1935   0.1744   0.1639   0.1838
## Balanced Accuracy      0.9977   0.9944   0.9891   0.9964   0.9990
```

Model 2 - Classification and Regression Tree (CART)

```
## Confusion Matrix and Statistics
## 
##           Reference
## Prediction    A    B    C    D    E
##          A 2029   44  155    0    4
##          B  638  505  375    0    0
##          C  644   49  675    0    0
##          D  567  232  487    0    0
##          E  209  211  383    0  639
## 
## Overall Statistics
## 
##                Accuracy : 0.4904
##                  95% CI : (0.4793, 0.5016)
##     No Information Rate : 0.5209
##     P-Value [Acc > NIR] : 1
## 
##                   Kappa : 0.3339
##  Mcnemar's Test P-Value : NA
## 
## Statistics by Class:
## 
##                      Class: A Class: B Class: C Class: D Class: E
## Sensitivity            0.4965  0.48511  0.32530       NA  0.99378
## Specificity            0.9460  0.85114  0.87992   0.8361  0.88852
## Pos Pred Value         0.9091  0.33267  0.49342       NA  0.44313
## Neg Pred Value         0.6334  0.91530  0.78388       NA  0.99938
## Prevalence             0.5209  0.13268  0.26447   0.0000  0.08195
## Detection Rate         0.2586  0.06436  0.08603   0.0000  0.08144
## Detection Prevalence   0.2845  0.19347  0.17436   0.1639  0.18379
## Balanced Accuracy      0.7212  0.66812  0.60261       NA  0.94115
```

Model 3 - Gradient Boosting Machine (GBM)

```
## Confusion Matrix and Statistics
## 
##           Reference
## Prediction    A    B    C    D    E
##          A 2211   11    3    5    2
##          B   51 1424   41    1    1
##          C    0   57 1285   22    4
##          D    2    4   47 1229    4
##          E    1   25   10   17 1389
## 
## Overall Statistics
##                                           
##                Accuracy : 0.9607          
##                  95% CI : (0.9562, 0.9649)
##     No Information Rate : 0.2887          
##     P-Value [Acc > NIR] : < 2.2e-16       
##                                           
##                   Kappa : 0.9503          
##  Mcnemar's Test P-Value : 2.25e-12        
## 
## Statistics by Class:
## 
##                      Class: A Class: B Class: C Class: D Class: E
## Sensitivity            0.9762   0.9362   0.9271   0.9647   0.9921
## Specificity            0.9962   0.9851   0.9872   0.9913   0.9918
## Pos Pred Value         0.9906   0.9381   0.9393   0.9557   0.9632
## Neg Pred Value         0.9904   0.9847   0.9844   0.9931   0.9983
## Prevalence             0.2887   0.1939   0.1767   0.1624   0.1784
## Detection Rate         0.2818   0.1815   0.1638   0.1566   0.1770
## Detection Prevalence   0.2845   0.1935   0.1744   0.1639   0.1838
## Balanced Accuracy      0.9862   0.9607   0.9571   0.9780   0.9920
```

Model Accuracy and Out-of-Sampling Error

```
##                Accuracy   OoS.Error
## Random Forest 0.9927352 0.007264848
## CART          0.4904410 0.509559011
## GBM           0.9607443 0.039255672
```

The table above shows that the random forest model produces the greatest accuracy and lowest out-of-sample error. Therefore, we will use this model to create our predictions.

Predictions from Predict Data Set

```
##  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20 
##  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
```