Predicting Exercise Manner

Jeff Olson

3/29/2021

Assignment

Predict the manner in which people exercise using data from accelerometers on the belt, forearm, arm, and dumbbell of 6 participants. Six healthy participants (20-28 years old, with little weight lifting experience) were asked to perform one set of 10 repetitions of the Unilateral Dumbbell Biceps Curl in five different fashions: exactly according to the specification (Class A), throwing the elbows to the front (Class B), lifting the dumbbell only halfway (Class C), lowering the dumbbell only halfway (Class D) and throwing the hips to the front (Class E). (See section on Weight Lifting Exercise Dataset, Velloso, E.; Bulling, A.; Gellersen, H.; Ugulino, W.; Fuks, H., http://groupware.les.inf.puc-rio.br/har).

Results

Using a random forest approach in the caret package of R, I was able to develop an algorithm that predicted almost 100% of the exercise manners in six cross-validations of the training data, using the random forest method, with cross-validation. I applied it to a validation set that was almost as accurate and then to the test data, with 100% accuracy.

How the Model Was Built

The random forest method was chosen, because it has shown itself to be effective in predicting classifications between more than two qualitative outcomes, as in this case. It tries multiple combinations of variables and selects combinations based on predictive value.

Variable Selection

The initial choice of variables was based on eliminating those with no variance. With only one value, they could not provide any discrimination (although, perhaps, NA observations might have been used as a second value, a possibility for future consideration). These variables were identified and removed from both the training and testing data sets, reducing the number of variables from 160 to 59. Caret provides for pre-processing methods that address this type of issue, but applicable ones for eliminating variables with little or no variance, such as “zv” and “nzv”, only apply to numeric variables. Most of the problematic variables were not numeric, so they needed to be eliminated in a pre-process that was not possible through the function preProcess. The remaining variables were almost entirely variables that were direct measurements of the movements.

```
## rf variable importance
##
##   only 20 most important variables shown (out of 80)
##
##                      Overall
## X                    100.000
## raw_timestamp_part_1  32.729
```

```
## roll_belt             23.186
## num_window            22.045
## yaw_belt              15.866
## magnet_dumbbell_y     13.410
## pitch_belt            13.301
## pitch_forearm         13.287
## magnet_dumbbell_z     13.031
## magnet_belt_y         11.446
## magnet_dumbbell_x     11.283
## accel_belt_z          10.816
## roll_forearm          10.758
## roll_dumbbell         10.409
## accel_dumbbell_y       9.934
## magnet_belt_z          9.080
## accel_dumbbell_z       8.041
## roll_arm               7.817
## gyros_belt_z           7.699
## accel_forearm_x        7.460
```

To avoid overfitting, mtry was reduced sequentially from the square root of the number of predictors, the default, until the accuracy started to diminish.

Final Model

```
##
## Call:
##  randomForest(x = x, y = y, mtry = param$mtry, trContol = ..1)
##                Type of random forest: classification
##                      Number of trees: 500
## No. of variables tried at each split: 4
##
##         OOB estimate of  error rate: 0.14%
## Confusion matrix:
##      A    B    C    D    E class.error
## A 2790    0    0    0    0 0.000000000
## B    2 1896    0    0    0 0.001053741
## C    0    3 1708    0    0 0.001753361
## D    0    0    6 1601    1 0.004353234
## E    0    0    0    2 1801 0.001109262
```

Cross Validation

Cross validation is at the heart of the random forest method. According to the creators, "In random forests, there is no need for cross-validation or a separate test set to get an unbiased estimate of the test set error. It is estimated internally, during the run, as follows: Each tree is constructed using a different bootstrap sample from the original data. About one-third of the cases are left out of the bootstrap sample and not used in the construction of the kth tree."(Breiman and Cutler, Random Forests, https://www.stat.berkeley.edu/~breiman/RandomForests/cc_home.htm). The confusion matrix provides a cross-validated (5 fold) set of results.

Confusion Matrix

```
## Bootstrapped (25 reps) Confusion Matrix
##
## (entries are percentual average cell counts across resamples)
##
```

```
##           Reference
## Prediction    A    B    C    D    E
##          A 28.4  0.0  0.0  0.0  0.0
##          B  0.0 19.4  0.0  0.0  0.0
##          C  0.0  0.0 17.4  0.1  0.0
##          D  0.0  0.0  0.0 16.3  0.0
##          E  0.0  0.0  0.0  0.0 18.3
##
##  Accuracy (average) : 0.9985
```

Half of the training set was set aside as a validation set. Here are the results associated with the confustion matrix of the validation set:

Confusion matrix of the validation set

```
## predValid
##    A    B    C    D    E
## 2790 1900 1715 1606 1801
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction    A    B    C    D    E
##          A 2790    0    0    0    0
##          B    0 1899    1    0    0
##          C    0    0 1710    5    0
##          D    0    0    0 1603    3
##          E    0    0    0    0 1801
##
## Overall Statistics
##
##                Accuracy : 0.9991
##                  95% CI : (0.9983, 0.9996)
##     No Information Rate : 0.2843
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.9988
##
##  Mcnemar's Test P-Value : NA
##
## Statistics by Class:
##
##                      Class: A Class: B Class: C Class: D Class: E
## Sensitivity            1.0000   1.0000   0.9994   0.9969   0.9983
## Specificity            1.0000   0.9999   0.9994   0.9996   1.0000
## Pos Pred Value         1.0000   0.9995   0.9971   0.9981   1.0000
## Neg Pred Value         1.0000   1.0000   0.9999   0.9994   0.9996
## Prevalence             0.2843   0.1935   0.1744   0.1639   0.1839
## Detection Rate         0.2843   0.1935   0.1743   0.1634   0.1836
## Detection Prevalence   0.2843   0.1936   0.1748   0.1637   0.1836
## Balanced Accuracy      1.0000   0.9999   0.9994   0.9983   0.9992
```

Out-of-Sample Error

The out-of-sample error is the error associated with the testing data. There was no error associated with the testing data. All 20 predictions matched the testing outcomes.