Predicting Weight Lifting Techniques Using Machine Learning

Mikhail Romadanovskiy

25 Jul 2015

Summary

The weight-lifting data set[^website] consisted of almost 20K rows of continuous data recordings of six young men performing one weight-lifting activity using five different methods, one correct and four incorrect. The datasets had 160 variables, some raw data collected from sensors and many calculated. The best performing algorithm for predicting the classe of the 20-row prediction test data set was random forest using CV for resampling. Accuracy was very high, but the elapsed time was very long compared to other projects completed for projects in this series.

Introduction

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 – 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 is 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. More information is available from the website here.

Data Preprocessing

Training set: 19622 observations and 160 variables, Testing set: 20 observations and 160 variables.

Prepare data

The best process for developing the model was to partition the data into a training group and a testing group. Step 1 was to divide the data into two groups. The script split the data set into two equal sets. We will clean the data and leave only meaningful variables.

Dimensions reduced from 60 to 53. We don't want to use particular time/date.

Slice the data

Spliting the cleaned training set into a pure training data set (70%) and a validation data set (30%).

Correlation

[1] "accel_belt_z" "roll_belt"
[3] "accel_belt_y" "accel_dumbbell_z" [5] "accel_belt_x" "pitch_belt"
[7] "accel_arm_x" "accel_dumbbell_x" [9] "magnet_arm_y" "gyros_forearm_y" [11] "gyros_dumbbell_x" "gyros_dumbbell_z" [13] "gyros_arm_x"

Many variables are highly correlated. PCA will be used in the pre-processing.

Applying ML algorithm

Random Forest algorithm will be used to predict the results. It automatically selects important variables and is robust to correlated covariates. **5-fold cross validation** will be used when applying the algorithm.

Random Forest

13737 samples 52 predictor 5 classes: 'A', 'B', 'C', 'D', 'E'

No pre-processing Resampling: Cross-Validated (5 fold) Summary of sample sizes: 10990, 10989, 10989, 10991, 10989 Resampling results across tuning parameters:

mtry Accuracy Kappa Accuracy SD 2 0.9900997 0.9874743 0.003759171 27 0.9920648 0.9899616 0.001774433 52 0.9870416 0.9836058 0.002807343 Kappa SD

0.004756127 0.002245312 0.003554778

Accuracy was used to select the optimal model using the largest value. The final value used for the model was mtry = 27.

Performance estimation of the model on the validation data set.

Confusion Matrix and Statistics

```
    Reference
```

Prediction A B C D E A 1674 0 0 0 0 B 17 1116 5 1 0 C 0 5 1014 7 0 D 0 0 6 957 1 E 0 1 4 1 1076

Overall Statistics

```
          Accuracy : 0.9918
            95% CI : (0.9892, 0.994)
```

```
No Information Rate : 0.2873
P-Value [Acc > NIR] : < 2.2e-16

              Kappa : 0.9897
```

Mcnemar's Test P-Value : NA
Statistics by Class:

```
                     Class: A Class: B
```

Sensitivity 0.9899 0.9947 Specificity 1.0000 0.9952 Pos Pred Value 1.0000 0.9798 Neg Pred Value 0.9960 0.9987 Prevalence 0.2873 0.1907 Detection Rate 0.2845 0.1896 Detection Prevalence 0.2845 0.1935 Balanced Accuracy 0.9950 0.9949 Class: C Class: D Sensitivity 0.9854 0.9907 Specificity 0.9975 0.9986 Pos Pred Value 0.9883 0.9927 Neg Pred Value 0.9969 0.9982 Prevalence 0.1749 0.1641 Detection Rate 0.1723 0.1626 Detection Prevalence 0.1743 0.1638 Balanced Accuracy 0.9915 0.9946 Class: E Sensitivity 0.9991 Specificity 0.9988 Pos Pred Value 0.9945 Neg Pred Value 0.9998 Prevalence 0.1830 Detection Rate 0.1828 Detection Prevalence 0.1839 Balanced Accuracy 0.9989 So, the estimated accuracy of the model is 0.9918437, 0.9896801.

Predicting for Test Data Set

Validate our model with test dataset dowloaded above. [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

Appendix: Figures ##1. Correlation Matrix Visualization

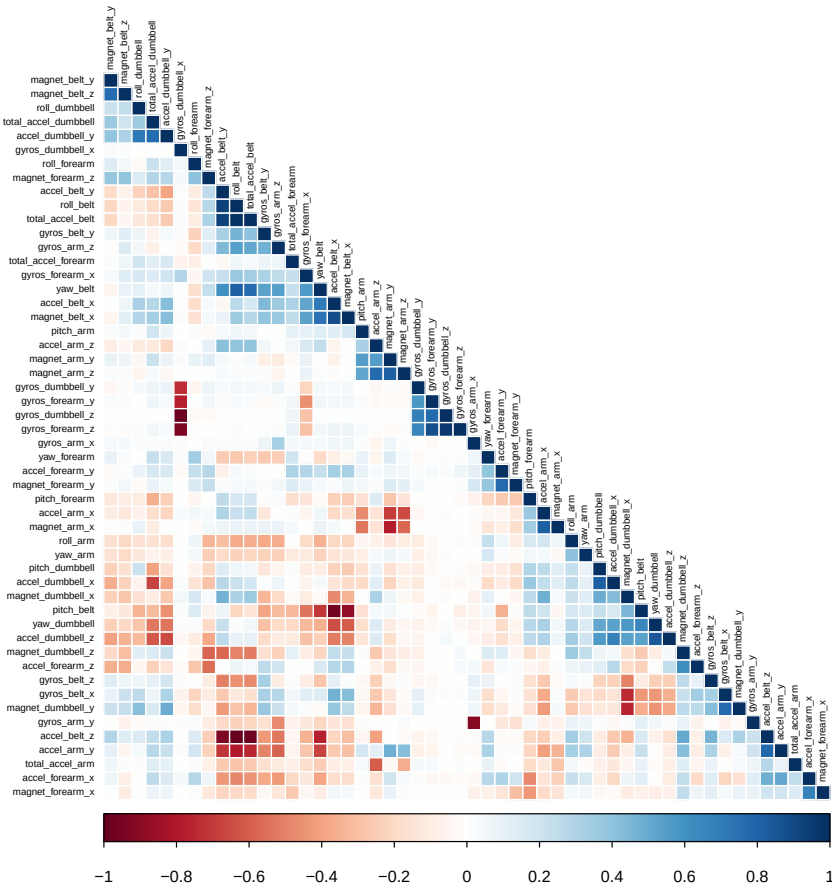

Figure 1: Correlation Matrix Visualization

2. *Decision Tree Visualization*

[^website] Source: `http://groupware.les.inf.puc-rio.br/har`

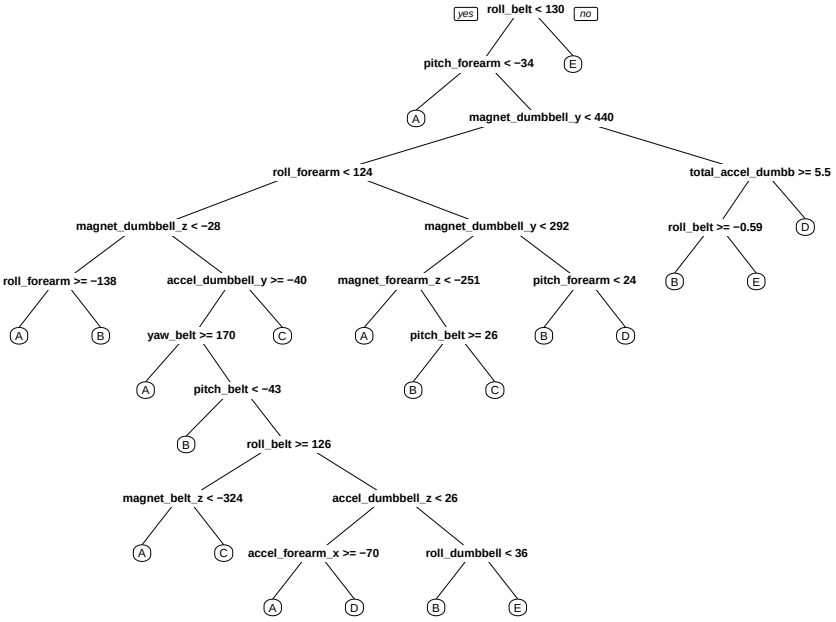

Figure 2: Decision Tree Visualization