# Classification of Weightlifting Technique

*A. Nugent, predmachlearn-035*

*Sunday, December 27, 2015*

## Purpose

To build a predictive model for assessing quality of weightlifting technique based on a training set that was classified by a subject matter expert.

## Methods

### Data

The dataset was generated by Velloso et al. in 2013, using 6 healthy adult males who were coached to perform dumbell curls in 5 different ways, graded in quality from A to E. There were 19622 observations of 160 variables in the raw training and test sets. The raw data included 7 columns of identifying factors (subjects, timestamps, and time windows) plus a number of columns containing NA or "DIV/0!".

For assessment purposes, the test set of 20 observations was provided blind.

### Preprocessing

The identifying columns named above were removed. All "skewness" and "kurtosis" columns (12 each) were removed, as these appeared to be junk (i.e. contained factors instead of continuous measurements, or "DIV/0!"). All remaining columns with NAs were removed. This left 52 variables to use as predictors, plus the output (classification) variable "classe".

Lacking sufficient domain knowledge, no transformations were applied to the remaining data.

The only nonstandard prerequisite package used was randomForest.

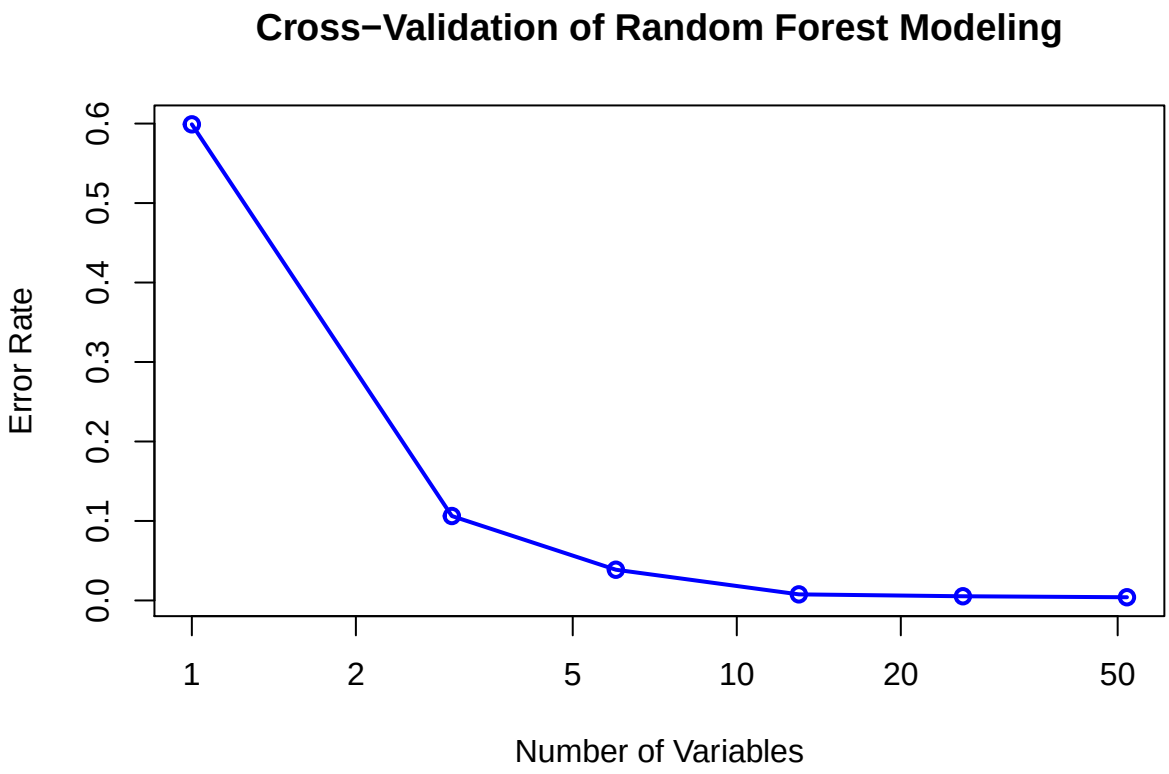
```
## randomForest 4.6-12
## Type rfNews() to see new features/changes/bug fixes.
```

### Model Selection

The random forest model was chosen for its demonstrated success in building classification models. To determine the optimal number of variables to use as predictors, cross-validation was invoked using the rfcv function from the randomForest package:

```
cv <- rfcv(data.train[ , -which(names(data.train) == "classe")], data.train$classe)
```

It was apparent that 26 variables would be sufficient for prediction, as the improvement in accuracy from 26 to 52 predictors was only 0.001 (0.1 percentage points). The cross- validation results are plotted below.




These same results are tabulated below.

```
##          52          26          13           6           3           1
## 0.004026093 0.005249210 0.007644481 0.038630109 0.106207318 0.599021506
```

Thus, the expected out-of-sample error rate would be 0.52% with 26 variables in the random forest model. This also yielded an acceptable computation duration (2.5 minutes using 64-bit Windows on a 2.83-GHz Intel Quad CPU with 8 GB RAM). Based on this result, the choice of model and parameters was vindicated.

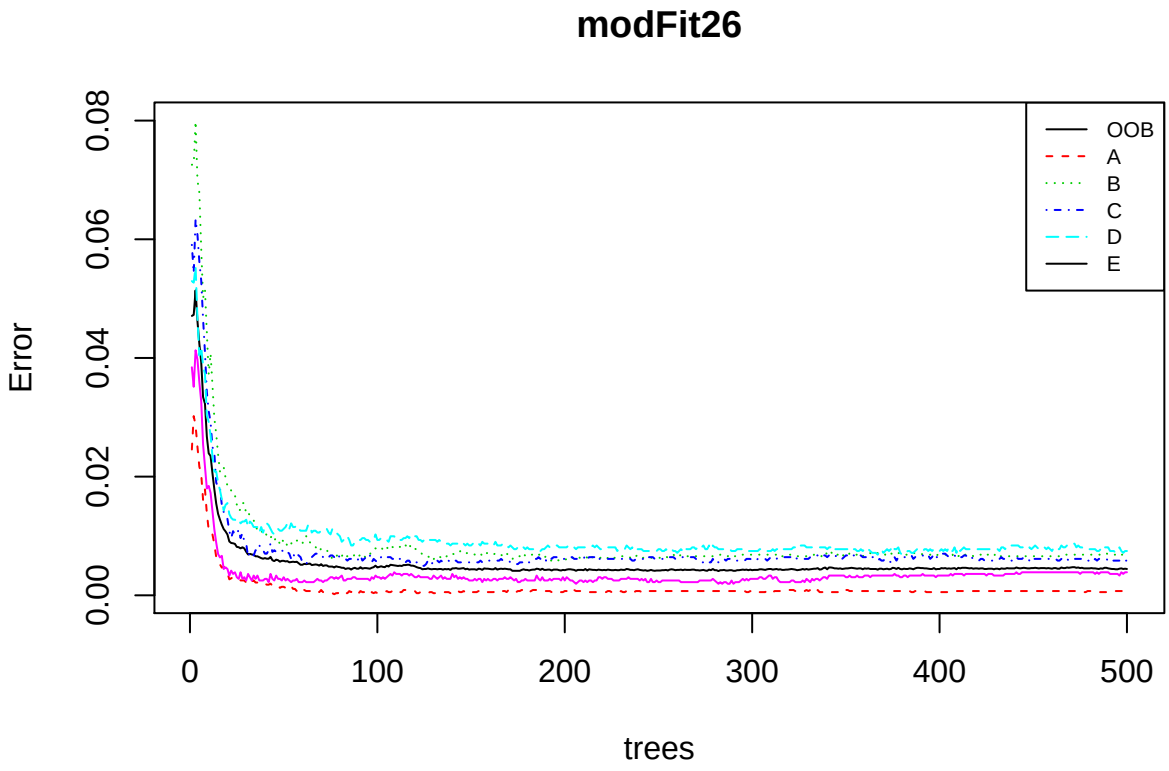
```
modFit26 <- randomForest(data.train[ , -which(names(data.train) == "classe")],
                         data.train$classe, mtry = 26, importance = TRUE)
```

# Results

## Model Evaluation

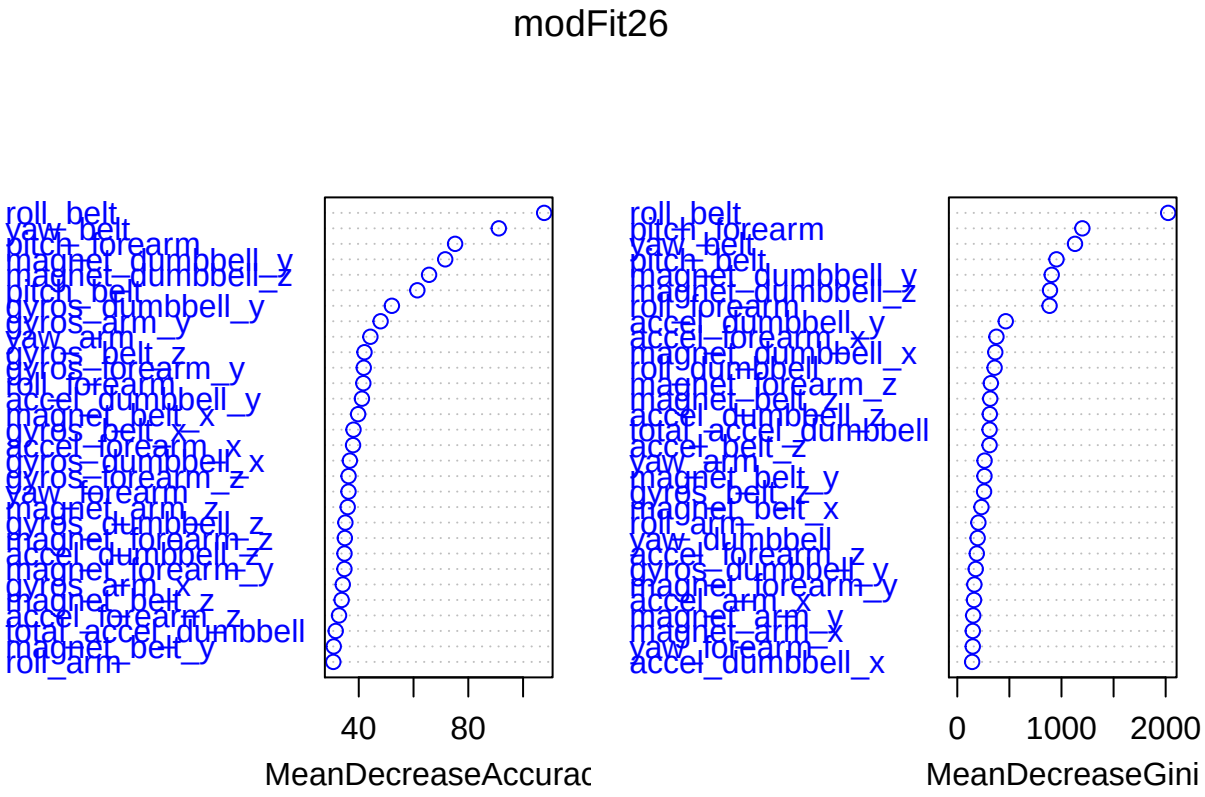
The following plot shows the rapid decline in error rate of each predicted class as the number of trees goes from zero to a little over 30. (OOB is the out-of-bag error.)

```
plot(modFit26)
modFit26.legend <- if (is.null(modFit26$test$err.rate)) {colnames(modFit26$err.rate)} else {colnames(mod
legend("topright", cex =0.7, legend=modFit26.legend, lty=c(1,2,3,4,5), col=c(1,2,3,4,5), horiz=FALSE)
```




The relative importance of each predictor in the model is illustrated in the following plot.

```
varImpPlot(modFit26, col="blue")
```




It is clear that the first 6 predictors are the most valuable; their order of importance varies only slightly between the two measures of accuracy.

The accuracy of the model as applied to the training set was perfect:

```
predictions.train <- predict(modFit26, newdata = data.train[ , -which(names(data.train) == "classe")])
data.train$correct <- predictions.train == data.train$classe
table(predictions.train, data.train$classe)
```

```
##
## predictions.train    A    B    C    D    E
##                 A 5580    0    0    0    0
##                 B    0 3797    0    0    0
##                 C    0    0 3422    0    0
##                 D    0    0    0 3216    0
##                 E    0    0    0    0 3607
```

### Predictions from Test Data

The resulting predictions for the test data were:

```
predictions <- predict(modFit26, newdata = data.test[ , -which(names(data.test) == "problem_id")])
predictions
```

```
##  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
```

Based on the out-of-sample error estimate and the perfect fit to the training data, and assuming that no bias was introduced when the training and test data were originally partitioned, the accuracy of the model should also be 100%.