Predicting Technique for Barbell Lifts Using Accelerometer Data and Random Forests

Robert Wexler

2/2/2017

Summary

I used random forests to predict technique (i.e. correctly and incorrect) for barbell lifts using data collected from accelerometers located on the belt, forearm, arm, and dumbell of six participants. I cleaned the data set, performed exploratory analyses, fit a number of different models, and evaluated different preprocessing and cross validation methods. The final model, which is a random forest with 10 trees, has an out sample error of 0.9845 to 0.9896 at the 95% confidence interval as predicted by cross validation with five folds.

Exploring and Cleaning Data

First, I wanted to get a sense of the size of the training and testing sets.

```
dim(testing)[1]/dim(training)[1]*100
```

```
## [1] 0.1019264
```

```
dim(testing)[1]+dim(training)[1]
```

```
## [1] 19642
```

The first value shows that the testing set is 0.10% the size of the training set. This is much too small for evaluating model performance. The second value shows that the total number of observations is 19642. This is a medium sample size and, therefore, I will split the data 60/40 into new training and validation sets.

```
set.seed(12345)
inTrain <- createDataPartition(y = training$classe, p = 0.60, list = FALSE)
training_new <- training[inTrain,]
validation_new <- training[-inTrain,]
dim(validation_new)[1]/dim(training_new)[1]*100
```

```
## [1] 66.62704
```

The printed value shows that 67% of the original training data is in the new training set whereas the rest of the data is in the new validation set. Now I will explore the values of the data frame to ensure that there are no missing or empty entries.

```
dim(training_new)
```

```
## [1] 11776   160
```

```
colNA <- names(which(colSums(is.na(training_new)) < 1))
training_new <- training_new[names(training_new) %in% colNA]
validation_new <- validation_new[names(validation_new) %in% colNA]
colEmpty <- names(which(colSums(training_new == "") < 1))
training_new <- training_new[names(training_new) %in% colEmpty]
validation_new <- validation_new[names(validation_new) %in% colEmpty]
```

The printed value shows that there are 11776 observations and 160 predictors. Looking at the first six entries of the new training set (code not shown), it is clear that some columns have many NA values or empty entries. I calculated the number of missing entries in each column and kept only those with zero. This reduced the number of predictors from 160 to 60 with the first seven columns corresponding to the participant and date/time and the other 53 comprising the design matrix. Finally, I looked for predictors that have one unique value.

```
nsv <- nearZeroVar(training_new, saveMetrics = TRUE)
sum(nsv$nzv)
```

```
## [1] 1
```

```
rownames(nsv)[which(nsv$nzv == TRUE)]
```

```
## [1] "new_window"
```

Only one predictor, new_window, had near zero variance but it is not part of the design matrix.

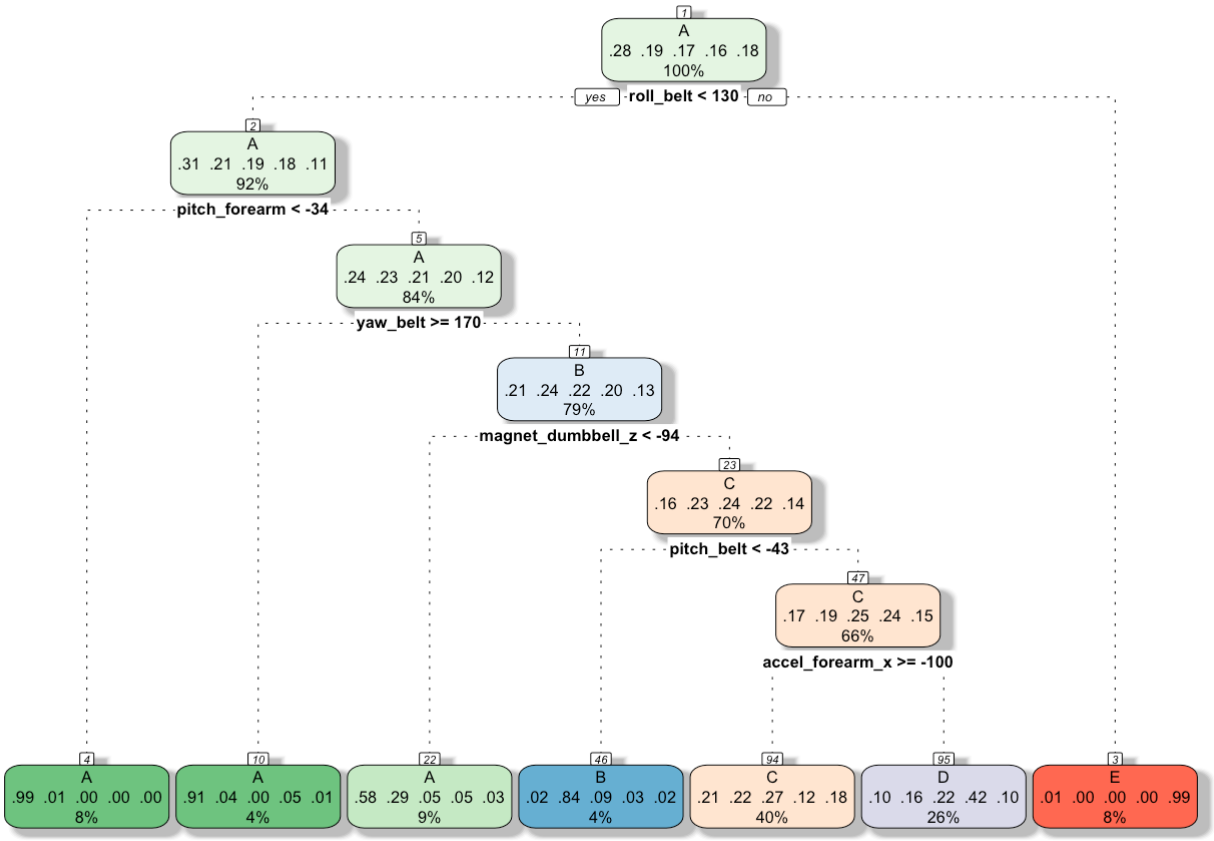
Model Selection

I tested models from three general categories: decision trees/random forests, boosting, and model based prediction (e.g. LDA and Naive Bayes).

Decision Trees and Random Forests

First, I trained a decision tree model with caret's default settings to predict classe using columns 8 to 59 of the new training set.

```
dcModel <- train(x = training_new[,8:59], y = training_new[,60], method = "rpart")
fancyRpartPlot(dcModel$finalModel)
```



Rattle 2017-Feb-03 23:13:17 robertwexler

The plot above shows that classe A is well-separated by pitch forearm < -34 and yaw belt >= 170, classe B by pitch belt < -43, and classe E by roll belt >= 130.

```
qplot(roll_belt, pitch_forearm, data = training_new, colour = classe)
```