Machine Learning Final Project

Emma Richard

March 8, 2019

Background

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 will be 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: http://web.archive.org/web/20161224072740/http:/groupware.les.inf.puc-rio.br/har (see the section on the Weight Lifting Exercise Dataset).

Data

The training data for this project are available here:

https://d396qusza40orc.cloudfront.net/predmachlearn/pml-training.csv

The test data are available here:

https://d396qusza40orc.cloudfront.net/predmachlearn/pml-testing.csv

The data for this project come from this source: http://web.archive.org/web/20161224072740/http:/groupware.les.inf.puc-rio.br/har. If you use the document you create for this class for any purpose please cite them as they have been very generous in allowing their data to be used for this kind of assignment.

Approach

The outcome variable is classe. Participants were asked to perform Unilateral Dumbbell Biceps Curl 5 ways: Class A:) According to specification Class B:) Throwing elbows to the front Class C:) Lifting dumbbell halfway Class D:) Lowering the dumbell halfway Class E:) Throwing hips to the front

In order to assess the data the following approach will be taken: 1.) Load and analyze the data 2.) Use cross-validation: 75% training set, 25% test set 3.) Apply decision tree method to build a model 4.) Apply random forest method to build a model 5.) Select the best model for prediction 6.) Report final outcome

Load libraries necessary for project

```
## Loading required package: lattice
```

```
## Loading required package: ggplot2
```

```
## randomForest 4.6-14
```

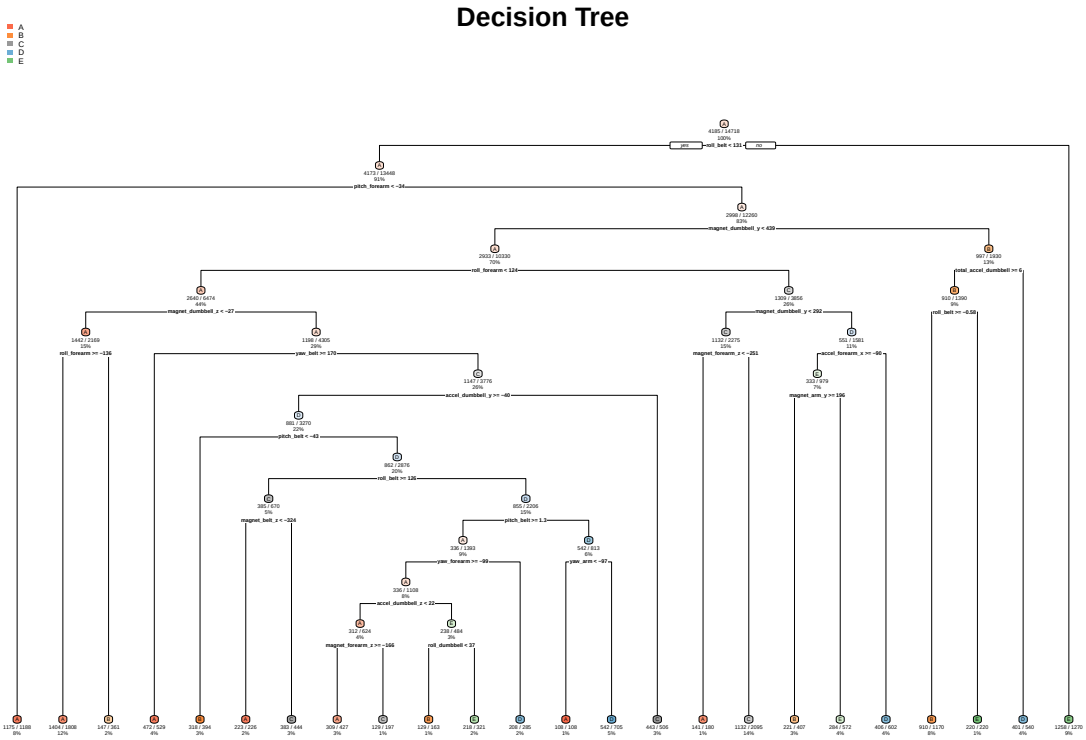
```
## Type rfNews() to see new features/changes/bug fixes.
```

```
##
## Attaching package: 'randomForest'
```

```
## The following object is masked from 'package:ggplot2':
##
##     margin
```

Get and clean data

Prediction Method 1: Decision Tree



Prediction Method 1 Results: Decision Tree

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction    A    B    C    D    E
##          A 1255  139   14   44   17
##          B   45  542   70   68   83
##          C   51  142  694  124  131
##          D   19   71   44  521   54
##          E   25   55   33   47  616
##
## Overall Statistics
```

```
##
##                Accuracy : 0.7398
##                  95% CI : (0.7273, 0.752)
##     No Information Rate : 0.2845
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.6704
##  Mcnemar's Test P-Value : < 2.2e-16
##
## Statistics by Class:
##
##                      Class: A Class: B Class: C Class: D Class: E
## Sensitivity            0.8996   0.5711   0.8117   0.6480   0.6837
## Specificity            0.9390   0.9327   0.8894   0.9541   0.9600
## Pos Pred Value         0.8543   0.6708   0.6077   0.7348   0.7938
## Neg Pred Value         0.9592   0.9006   0.9572   0.9325   0.9310
## Prevalence             0.2845   0.1935   0.1743   0.1639   0.1837
## Detection Rate         0.2559   0.1105   0.1415   0.1062   0.1256
## Detection Prevalence   0.2996   0.1648   0.2329   0.1446   0.1582
## Balanced Accuracy      0.9193   0.7519   0.8505   0.8011   0.8219
```

Prediction Method 2: Random Forest

Prediction Method 2 Results: Random Forest

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction    A    B    C    D    E
##          A 1394    3    0    0    0
##          B    1  944    2    0    0
##          C    0    2  853    9    1
##          D    0    0    0  794    1
##          E    0    0    0    1  899
##
## Overall Statistics
##
##                Accuracy : 0.9959
##                  95% CI : (0.9937, 0.9975)
##     No Information Rate : 0.2845
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.9948
##  Mcnemar's Test P-Value : NA
##
## Statistics by Class:
##
##                      Class: A Class: B Class: C Class: D Class: E
## Sensitivity            0.9993   0.9947   0.9977   0.9876   0.9978
## Specificity            0.9991   0.9992   0.9970   0.9998   0.9998
## Pos Pred Value         0.9979   0.9968   0.9861   0.9987   0.9989
## Neg Pred Value         0.9997   0.9987   0.9995   0.9976   0.9995
```

```
## Prevalence             0.2845   0.1935   0.1743   0.1639   0.1837
## Detection Rate         0.2843   0.1925   0.1739   0.1619   0.1833
## Detection Prevalence   0.2849   0.1931   0.1764   0.1621   0.1835
## Balanced Accuracy      0.9992   0.9970   0.9973   0.9937   0.9988
```

Model selection

The random forest model performed best. Random forest is used to get the final outcome.

Final outcome

```
##  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
```