Practical Machine Learning - Final Project

Fred Smith

Tuesday, May 03, 2016

Executive Summary

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, the goal is to use data from accelerometers on the belt, forearm, arm, and dumbell of 6 participants.

The goal of this project is to predict the manner in which they did the exercise. This is the “classe” variable in the training set. Any of the other variables may be used to predict with.

This Report describes a prediction model, how it was built, and validated. An estimate of the expected error is provided along with rational for choices that influenced the model design. Lastly, the model was used to predict exercises for a small (20 observation) test set, that was independend of the training and validation data sets.

Background and Data Source

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 types 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. This project uses 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://groupware.les.inf.puc-rio.br/har (http://groupware.les.inf.puc-rio.br/har) (see the section on the Weight Lifting Exercise Dataset).

Exploratory Data Analysis

Read the complete dataset and perform basic analysis.

```
setwd("C:/Users/Fred/Documents/Academic/DataScience")
data <- read.csv("./MachineLearning/Project/pml-training.csv")
dim(data)
```

```
## [1] 19622   160
```

```
sum(head(complete.cases(data)))
```

```
## [1] 0
```

Since there are 160 variables, and none of the 19,622 observations are complete, I first wish to gain a more complete picture of the nature of the data.

```
names(data)
```

```
##   [1] "X"                        "user_name"               
##   [3] "raw_timestamp_part_1"     "raw_timestamp_part_2"    
##   [5] "cvtd_timestamp"           "new_window"              
##   [7] "num_window"               "roll_belt"               
##   [9] "pitch_belt"               "yaw_belt"                
##  [11] "total_accel_belt"         "kurtosis_roll_belt"      
##  [13] "kurtosis_picth_belt"      "kurtosis_yaw_belt"       
##  [15] "skewness_roll_belt"       "skewness_roll_belt.1"    
##  [17] "skewness_yaw_belt"        "max_roll_belt"           
##  [19] "max_picth_belt"           "max_yaw_belt"            
##  [21] "min_roll_belt"            "min_pitch_belt"          
##  [23] "min_yaw_belt"             "amplitude_roll_belt"     
##  [25] "amplitude_pitch_belt"     "amplitude_yaw_belt"      
##  [27] "var_total_accel_belt"     "avg_roll_belt"           
##  [29] "stddev_roll_belt"         "var_roll_belt"           
##  [31] "avg_pitch_belt"           "stddev_pitch_belt"       
##  [33] "var_pitch_belt"           "avg_yaw_belt"            
##  [35] "stddev_yaw_belt"          "var_yaw_belt"            
##  [37] "gyros_belt_x"             "gyros_belt_y"            
##  [39] "gyros_belt_z"             "accel_belt_x"            
##  [41] "accel_belt_y"             "accel_belt_z"            
##  [43] "magnet_belt_x"            "magnet_belt_y"           
##  [45] "magnet_belt_z"            "roll_arm"                
##  [47] "pitch_arm"                "yaw_arm"                 
##  [49] "total_accel_arm"          "var_accel_arm"           
##  [51] "avg_roll_arm"             "stddev_roll_arm"         
##  [53] "var_roll_arm"             "avg_pitch_arm"           
##  [55] "stddev_pitch_arm"         "var_pitch_arm"           
##  [57] "avg_yaw_arm"              "stddev_yaw_arm"          
##  [59] "var_yaw_arm"              "gyros_arm_x"             
##  [61] "gyros_arm_y"              "gyros_arm_z"             
##  [63] "accel_arm_x"              "accel_arm_y"             
##  [65] "accel_arm_z"              "magnet_arm_x"            
##  [67] "magnet_arm_y"             "magnet_arm_z"            
##  [69] "kurtosis_roll_arm"        "kurtosis_picth_arm"      
##  [71] "kurtosis_yaw_arm"         "skewness_roll_arm"       
##  [73] "skewness_pitch_arm"       "skewness_yaw_arm"        
##  [75] "max_roll_arm"             "max_picth_arm"           
##  [77] "max_yaw_arm"              "min_roll_arm"            
##  [79] "min_pitch_arm"            "min_yaw_arm"             
##  [81] "amplitude_roll_arm"       "amplitude_pitch_arm"     
##  [83] "amplitude_yaw_arm"        "roll_dumbbell"           
##  [85] "pitch_dumbbell"           "yaw_dumbbell"            
##  [87] "kurtosis_roll_dumbbell"   "kurtosis_picth_dumbbell" 
##  [89] "kurtosis_yaw_dumbbell"    "skewness_roll_dumbbell"  
##  [91] "skewness_pitch_dumbbell"  "skewness_yaw_dumbbell"   
##  [93] "max_roll_dumbbell"        "max_picth_dumbbell"      
##  [95] "max_yaw_dumbbell"         "min_roll_dumbbell"       
##  [97] "min_pitch_dumbbell"       "min_yaw_dumbbell"        
##  [99] "amplitude_roll_dumbbell"  "amplitude_pitch_dumbbell"
## [101] "amplitude_yaw_dumbbell"   "total_accel_dumbbell"    
## [103] "var_accel_dumbbell"       "avg_roll_dumbbell"       
## [105] "stddev_roll_dumbbell"     "var_roll_dumbbell"       
## [107] "avg_pitch_dumbbell"       "stddev_pitch_dumbbell"   
## [109] "var_pitch_dumbbell"       "avg_yaw_dumbbell"        
## [111] "stddev_yaw_dumbbell"      "var_yaw_dumbbell"        
## [113] "gyros_dumbbell_x"         "gyros_dumbbell_y"        
## [115] "gyros_dumbbell_z"         "accel_dumbbell_x"        
## [117] "accel_dumbbell_y"         "accel_dumbbell_z"        
## [119] "magnet_dumbbell_x"        "magnet_dumbbell_y"       
## [121] "magnet_dumbbell_z"        "roll_forearm"            
## [123] "pitch_forearm"            "yaw_forearm"             
## [125] "kurtosis_roll_forearm"    "kurtosis_picth_forearm"  
## [127] "kurtosis_yaw_forearm"     "skewness_roll_forearm"   
## [129] "skewness_pitch_forearm"   "skewness_yaw_forearm"    
## [131] "max_roll_forearm"         "max_picth_forearm"       
## [133] "max_yaw_forearm"          "min_roll_forearm"        
## [135] "min_pitch_forearm"        "min_yaw_forearm"         
## [137] "amplitude_roll_forearm"   "amplitude_pitch_forearm" 
## [139] "amplitude_yaw_forearm"    "total_accel_forearm"     
## [141] "var_accel_forearm"        "avg_roll_forearm"        
## [143] "stddev_roll_forearm"      "var_roll_forearm"        
## [145] "avg_pitch_forearm"        "stddev_pitch_forearm"    
## [147] "var_pitch_forearm"        "avg_yaw_forearm"         
## [149] "stddev_yaw_forearm"       "var_yaw_forearm"         
## [151] "gyros_forearm_x"          "gyros_forearm_y"         
## [153] "gyros_forearm_z"          "accel_forearm_x"         
## [155] "accel_forearm_y"          "accel_forearm_z"         
## [157] "magnet_forearm_x"         "magnet_forearm_y"        
## [159] "magnet_forearm_z"         "classe"
```

Data Cleansing

Observation Outcomes

The “classe” variable indicates the outcome of each observed exercise: * A - Correctly performed * B - Throwing elbows to the front * C - Lifting the dumbell only half way * D - Lowering the dumbell only half way * E - Throwing the hips to the front

Observation Variables

The data is described in this paper: http://groupware.les.inf.puc-rio.br/public/papers/2013.Velloso.QAR-WLE.pdf (http://groupware.les.inf.puc-rio.br/public/papers/2013.Velloso.QAR-WLE.pdf). Section 5.1 describes the author’s rational for selecting the following features: * Belt - mean roll * Belt - variance roll

* Belt - acceleration (paper called for maximum, range, and variance, but these were apparently not in the dataset) * Belt - variance acceleration (included because of lack of others) Variances of the gyro and magnetometer were not in the dataset. * Arm - variance acceleration. Maximum and minimum of magnetometer were not in the dataset. * Dumbell - None of the variables from the paper were in the dataset (maximum acceleration, variance gyro, and maximum and minimum magnetometer) * Glove - The paper calls out glove data, but non is included in the dataset. Forearm variables were substituted instead due to proximity of the wrist to the glove. * Forearm - Pitch. Maximum and minimum gyro were not in the dataset.

Because many of the variables that the paper refers to are not included in the dataset, the classifier that is described in the paper cannot be duplicated exactly. The following variables were selected to augment those variables that are available, based on subjective similarity to the missing variables, and intuitions of which variables would provide mechanical differentiation between the failure modes: * Arm - total acceleration * Forearm - minimum, maximum, and variance of pitch * Dumbell - total and variance of acceleration, and pitch

```
select.features <- c(
        "classe",                       # Observations
        "avg_roll_belt",                # Variables selected in paper
        "var_roll_belt",
        "total_accel_belt",
        "var_total_accel_belt",
        "var_accel_arm",
        "amplitude_pitch_forearm",
        "total_accel_arm",              # Variables added in place of missing v
ariables
        "min_pitch_forearm",
        "max_picth_forearm",
        "var_pitch_forearm",
        "total_accel_dumbbell",
        "var_accel_dumbbell",
        "var_pitch_dumbbell"
        )
select.features
```

```
##  [1] "classe"                  "avg_roll_belt"
##  [3] "var_roll_belt"           "total_accel_belt"
##  [5] "var_total_accel_belt"    "var_accel_arm"
##  [7] "amplitude_pitch_forearm" "total_accel_arm"
##  [9] "min_pitch_forearm"       "max_picth_forearm"
## [11] "var_pitch_forearm"       "total_accel_dumbbell"
## [13] "var_accel_dumbbell"      "var_pitch_dumbbell"
```

```
select.data <- data[,select.features]
dim(select.data)
```

```
## [1] 19622    14
```

```
sum(complete.cases(select.data))
```

```
## [1] 406
```

The above selections resulted in a very sparse matrix. Therefore, the following variables were chosen based on availability of data values, and mechanical intuitions about the movements being measured.

```
select.features <- c(
        "classe",                         # Observations
        "roll_belt",
        "pitch_belt",
        "total_accel_belt",
        "roll_arm",
        "pitch_arm",
        "yaw_arm",
        "total_accel_arm",
        "roll_dumbbell",
        "pitch_dumbbell",
        "yaw_dumbbell"
        )
select.features
```

```
##  [1] "classe"           "roll_belt"        "pitch_belt"
##  [4] "total_accel_belt" "roll_arm"         "pitch_arm"
##  [7] "yaw_arm"          "total_accel_arm"  "roll_dumbbell"
## [10] "pitch_dumbbell"   "yaw_dumbbell"
```

```
select.data <- data[,select.features]
dim(select.data)
```

```
## [1] 19622    11
```

```
sum(complete.cases(select.data))
```

```
## [1] 19622
```

Training and Test Datasets

Now that we have a better understanding of the full dataset, we can randomly partition the full set into training and test sets used to calculate and validate (respectively) any proposed models. Per best practices, 70% of the records, partitioned on the outcome “classe” variable, will be randomly selected for the training dataset, and the remaining 30% will be used to validate any models. In order to make these results repeatable, the required libraries are loaded, and the random seed is set to start the process.

```r
library(lattice); library(ggplot2); library(caret)
set.seed(5309)
inTrain  <- createDataPartition(y=select.data$classe,p=0.7,list=FALSE)
training <- select.data[inTrain,]
testing  <- select.data[-inTrain,]
```

Recursive Partitioning (rpart) Classification Tree

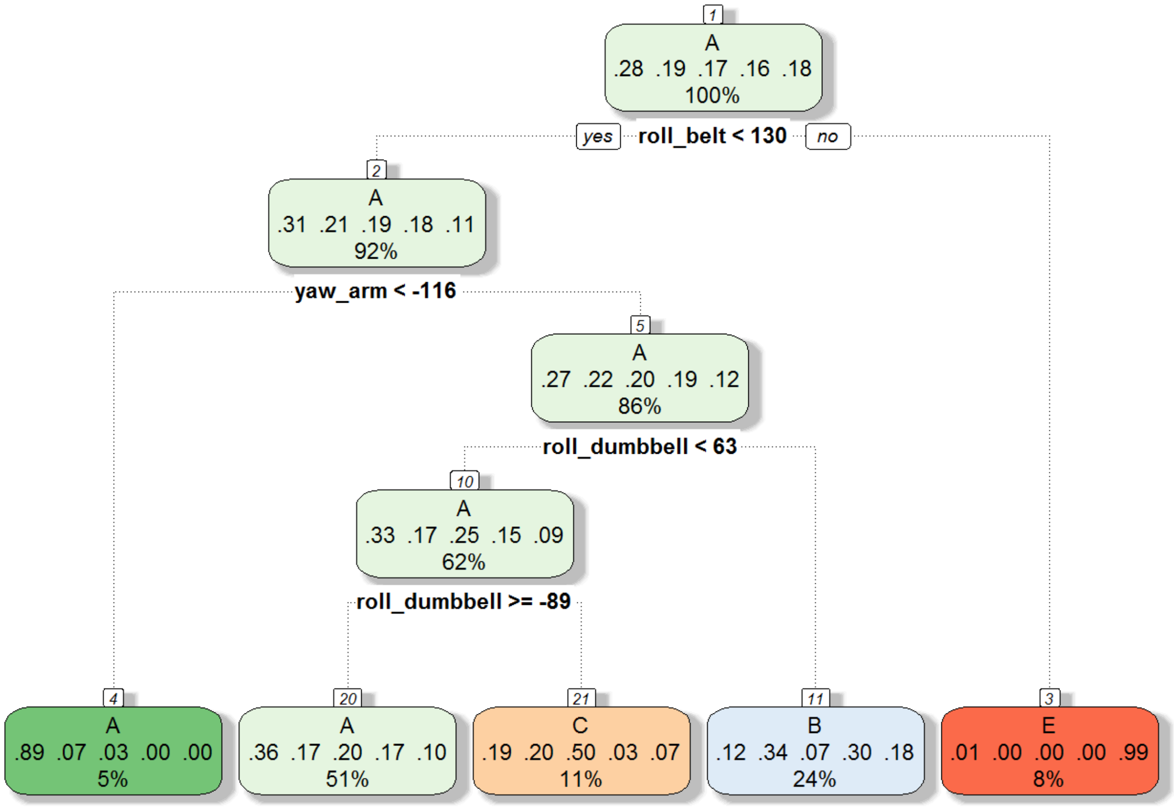
```r
library(rpart); library(rattle)
```

```
## Rattle: A free graphical interface for data mining with R.
## Version 4.1.0 Copyright (c) 2006-2015 Togaware Pty Ltd.
## Type 'rattle()' to shake, rattle, and roll your data.
```

```r
set.seed(867)
mod.rpart <- train(classe~.,method="rpart",data=training)
pred.rpart <- predict(mod.rpart,newdata=testing)
confusionMatrix(pred.rpart,testing$classe)
```

```
## Confusion Matrix and Statistics
## 
##           Reference
## Prediction    A    B    C    D    E
##          A 1360  528  610  541  298
##          B  192  488   96  401  235
##          C  120  123  320   22   55
##          D    0    0    0    0    0
##          E    2    0    0    0  494
## 
## Overall Statistics
##                                           
##                Accuracy : 0.4523          
##                  95% CI : (0.4396, 0.4652)
##     No Information Rate : 0.2845          
##     P-Value [Acc > NIR] : < 2.2e-16       
##                                           
##                   Kappa : 0.2773          
##  Mcnemar's Test P-Value : NA              
## 
## Statistics by Class:
## 
##                      Class: A Class: B Class: C Class: D Class: E
## Sensitivity            0.8124  0.42845  0.31189   0.0000  0.45656
## Specificity            0.5305  0.80531  0.93414   1.0000  0.99958
## Pos Pred Value         0.4076  0.34561  0.50000      NaN  0.99597
## Neg Pred Value         0.8768  0.85446  0.86540   0.8362  0.89089
## Prevalence             0.2845  0.19354  0.17434   0.1638  0.18386
## Detection Rate         0.2311  0.08292  0.05438   0.0000  0.08394
## Detection Prevalence   0.5670  0.23993  0.10875   0.0000  0.08428
## Balanced Accuracy      0.6715  0.61688  0.62302   0.5000  0.72807
```

```
fancyRpartPlot(mod.rpart$finalModel)
```



Rattle 2016-May-04 20:47:55 Fred

The recursive partitioning model did not perform well at all, with only 45% classification accuracy. In fact, it was unable to classify class D exercises at all.

Random Forest (rf)

The paper authors selected a Random Forest approach with bagging to build their classification model.

```
library(randomForest)
```

```
## randomForest 4.6-12
```

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

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

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

```
set.seed(867)
mod.rf <- train(classe~.,method="rf",data=training,trControl = trainControl(num
ber = 4))
pred.rf <- predict(mod.rf,newdata=testing)
confusionMatrix(pred.rf,testing$classe)
```

```
## Confusion Matrix and Statistics
## 
##           Reference
## Prediction    A    B    C    D    E
##          A 1608   35   22   16    4
##          B    9 1036   20    9   13
##          C   18   48  959   37    7
##          D   31   13   23  900    4
##          E    8    7    2    2 1054
## 
## Overall Statistics
##                                           
##                Accuracy : 0.9443          
##                  95% CI : (0.9381, 0.95)  
##     No Information Rate : 0.2845          
##     P-Value [Acc > NIR] : < 2.2e-16       
##                                           
##                   Kappa : 0.9295          
##  Mcnemar's Test P-Value : 5.738e-06       
## 
## Statistics by Class:
## 
##                      Class: A Class: B Class: C Class: D Class: E
## Sensitivity            0.9606   0.9096   0.9347   0.9336   0.9741
## Specificity            0.9817   0.9893   0.9774   0.9856   0.9960
## Pos Pred Value         0.9543   0.9531   0.8971   0.9269   0.9823
## Neg Pred Value         0.9843   0.9785   0.9861   0.9870   0.9942
## Prevalence             0.2845   0.1935   0.1743   0.1638   0.1839
## Detection Rate         0.2732   0.1760   0.1630   0.1529   0.1791
## Detection Prevalence   0.2863   0.1847   0.1816   0.1650   0.1823
## Balanced Accuracy      0.9711   0.9494   0.9560   0.9596   0.9851
```

This simple Random Forest model, with variables picked by subjective intuition produced overall accuracy of 94.4% on the test dataset. The authors also used a bagging procedure to cross-validate during training. But since this model is likely to get at least an 80% on the final classification quiz, I will stop here.

Final Quiz

The Random Forest model above is now run against the quiz dataset to produce the final project predictions.

```
setwd("C:/Users/Fred/Documents/Academic/DataScience")
quiz.data <- read.csv("./MachineLearning/Project/pml-testing.csv")
quiz.select.data <- quiz.data[,select.features[2:11]]
quiz.answers <- predict(mod.rf,newdata=quiz.select.data)
quiz.answers
```

```
##  [1] B A B A A E C D A A B C B A E E A B B B
## Levels: A B C D E
```

These classifications have been submitted to the Coursera project/final quiz and received a grade of 18/20 (90%).

Appendix A - Environment

```
sessionInfo()
```

```
## R version 3.2.5 (2016-04-14)
## Platform: x86_64-w64-mingw32/x64 (64-bit)
## Running under: Windows 8.1 x64 (build 9600)
## 
## locale:
## [1] LC_COLLATE=English_United States.1252 
## [2] LC_CTYPE=English_United States.1252   
## [3] LC_MONETARY=English_United States.1252
## [4] LC_NUMERIC=C                          
## [5] LC_TIME=English_United States.1252    
## 
## attached base packages:
## [1] stats     graphics  grDevices utils     datasets  methods   base     
## 
## other attached packages:
## [1] randomForest_4.6-12 rattle_4.1.0        rpart_4.1-10       
## [4] caret_6.0-68        ggplot2_2.1.0       lattice_0.20-33    
## 
## loaded via a namespace (and not attached):
##  [1] Rcpp_0.12.4        RColorBrewer_1.1-2 compiler_3.2.5    
##  [4] nloptr_1.0.4       plyr_1.8.3         class_7.3-14      
##  [7] iterators_1.0.8    tools_3.2.5        digest_0.6.9      
## [10] lme4_1.1-12        evaluate_0.8.3     nlme_3.1-127      
## [13] gtable_0.2.0       mgcv_1.8-12        Matrix_1.2-5      
## [16] foreach_1.4.3      yaml_2.1.13        parallel_3.2.5    
## [19] SparseM_1.7        e1071_1.6-7        RGtk2_2.20.31     
## [22] stringr_1.0.0      knitr_1.12.3       MatrixModels_0.4-1
## [25] stats4_3.2.5       grid_3.2.5         nnet_7.3-12       
## [28] rmarkdown_0.9.5    minqa_1.2.4        reshape2_1.4.1    
## [31] car_2.1-2          magrittr_1.5       scales_0.4.0      
## [34] codetools_0.2-14   htmltools_0.3.5    MASS_7.3-45       
## [37] splines_3.2.5      rpart.plot_1.5.3   pbkrtest_0.4-6    
## [40] colorspace_1.2-6   quantreg_5.21      stringi_1.0-1     
## [43] munsell_0.4.3
```