Practical Machine Learning - Human Activity Recognition

Mark Stevenson

September 26, 2015

Assignment Information (from Coursera for Reference):

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://groupware.les.inf.puc-rio.br/har (see the section on the Weight Lifting Exercise Dataset).*

Data:

1. *The training data for this project are available here: [https://d396qusza40orc.cloudfront.net/predmachlearn/pml-training.csv]*
2. *The test data are available here: [https://d396qusza40orc.cloudfront.net/predmachlearn/pml-testing.csv]*
3. *The data for this project come from this source: [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.*

What you should submit: *The goal of your project is to predict the manner in which they did the exercise. This is the "classe" variable in the training set. You may use any of the other variables to predict with. You should create a report describing how you built your model, how you used cross validation, what you think the expected out of sample error is, and why you made the choices you did. You will also use your prediction model to predict 20 different test cases.*

1. *Your submission should consist of a link to a Github repo with your R markdown and compiled HTML file describing your analysis. Please constrain the text of the writeup to < 2000 words and the number of figures to be less than 5. It will make it easier for the graders if you submit a repo with a gh-pages branch so the HTML page can be viewed online (and you always want to make it easy on graders :-).*
2. *You should also apply your machine learning algorithm to the 20 test cases available in the test data above. Please submit your predictions in appropriate format to the programming assignment for automated grading. See the programming assignment for additional details.*

Student Solution/Analysis and Response:

Initial Approach:

1. Read reference material, acquire, and explore data.

2. Perform preparation for round of cross validation by subsetting 60% of training data to a training subset and the remaining 40% to a testing subset. Perform data cleansing/preparation on both sets to ensure equivalence.
3. Develop models and estimate out of sample error rate. A reasonable expectation for out of sample error is expected to be 100% minus the accuracy of the trained model. This would further be substantiated by reviewing the model accuracy on application to the cross-validation test set.

4. Acquire and prepare Coursera test data, and apply model to upload results

```
library(caret)
```

Loading Required Libraries:

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

```
library(ggplot2)
library(randomForest)
```

```
## randomForest 4.6-10
## Type rfNews() to see new features/changes/bug fixes.
```

```
set.seed(333) #to ensure this can be reproduced
```

```
URL <- "https://d396qusza40orc.cloudfront.net/predmachlearn/pml-training.csv";
temp <- tempfile(); download.file(URL,temp)
train<- read.csv(temp, na.strings = c('','NA','#DIV/0!'))
```

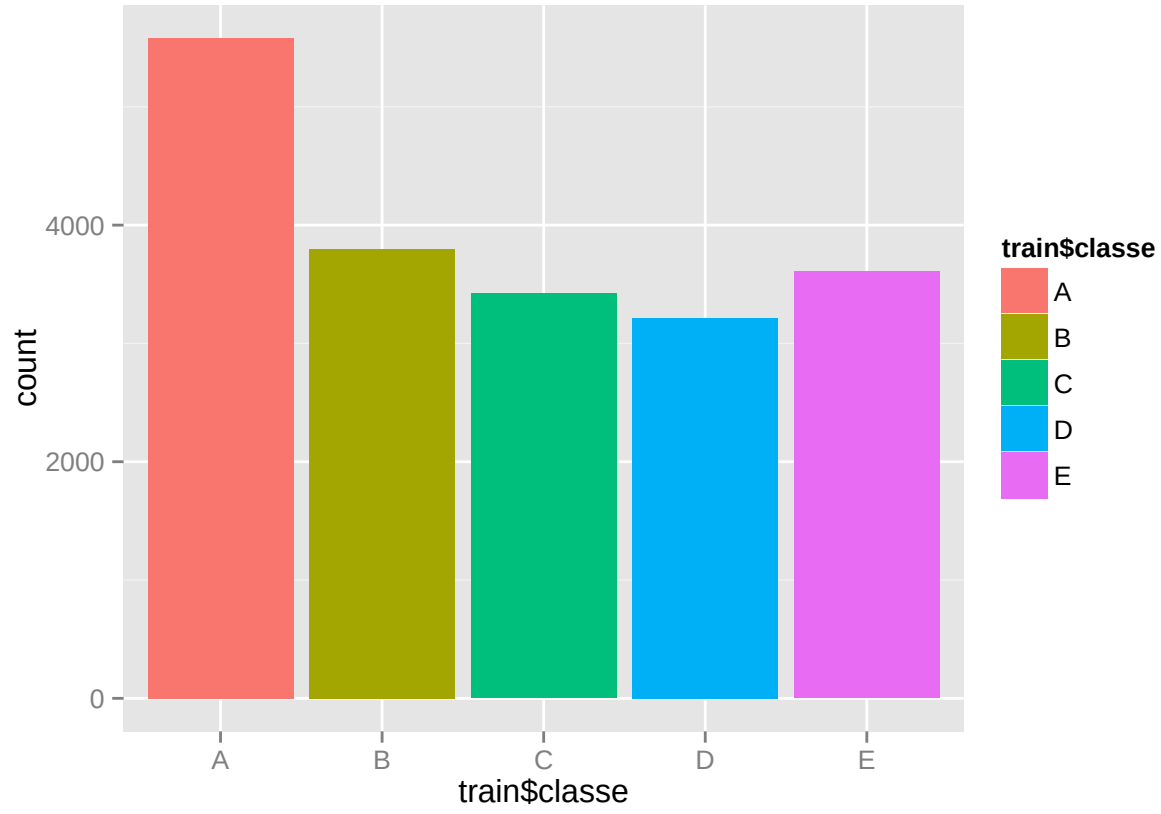
Data Acquisition:

Exploratory Data Analysis: The author has reviewed the structure of the training data set by reading reference information in the Background section above. We are to predict the value of the 'classe' variable which represents five different exercise fashions of the Unilateral Dumbbell Biceps Curl. Per the reference information provided we understand the 'classe' values are:

Value	Description
A	Exactly according to the specification
B	Throwing the elbows to the front
C	Lifting the dumbbell only halfway
D	Lowering the dumbbell only halfway
E	Throwing the hips to the front

Let's develop an understanding of the frequency of each of the fashions in the training set by performing a histogram:

```
qplot(train$classe, geom="histogram", fill=train$classe)
```



Let's now review the data via the following commands (not printed to save space):

```
utils::View(train)
summary(train)
```

Data Preparation: Author's Observations and Rationale for Data Preparation Choices from Exploratory Analysis:

1. Let's initially remove the 'ID' column as this variable would lead to linearity and throw off our algorithm.
2. Let's remove the user name and other administrative fields in the remaining six columns
3. There are lots of columns with NA's and they are sparse so we will remove them

```
na_count <- data.frame(lapply(train, function(y) sum(length(which(is.na(y))))))
NA_cols <-names(which(apply(na_count>0, 2, any)))
new_train <- train[,!names(train) %in% NA_cols]
new_train<-new_train[complete.cases(new_train),]
new_train<-new_train[,-1:-7]
```

As we haven't performed any numerical calculations (only column wise operations) we can break out our cross validation train and test sets:

```
train_flag <- createDataPartition(y=new_train$classe, p=0.6, list=FALSE)
final_train <- new_train[train_flag, ]
cross_validation_test <- new_train[-train_flag, ]
```

Prediction: Authors rationale for building initial model: As we're somewhat limited by space in this analysis we'll use all available variables to build the best 'general' model we can. First we'll try Linear Discriminant Analysis:

```
model_LDA <- train(classe ~. , data=final_train, method='lda')
```

```
## Loading required package: MASS
```

```
model_LDA
```

```
## Linear Discriminant Analysis
##
## 11776 samples
##    52 predictor
##     5 classes: 'A', 'B', 'C', 'D', 'E'
##
## No pre-processing
## Resampling: Bootstrapped (25 reps)
## Summary of sample sizes: 11776, 11776, 11776, 11776, 11776, 11776, ...
## Resampling results
##
##   Accuracy   Kappa      Accuracy SD  Kappa SD
##   0.6991449  0.6191006  0.007192963  0.008905924
##
##
```

```
LDA_predictions <- predict(model_LDA, cross_validation_test, type="raw")
confusionMatrix(data=LDA_predictions, cross_validation_test$classe)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction    A    B    C    D    E
##          A 1841  230  142   64   56
##          B   50  953  149   48  244
##          C  167  187  881  168  131
##          D  168   72  156  951  159
##          E    6   76   40   55  852
##
## Overall Statistics
##
##                Accuracy : 0.6982
##                  95% CI : (0.6879, 0.7083)
##     No Information Rate : 0.2845
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.618
```

```
##  Mcnemar's Test P-Value : < 2.2e-16
##
## Statistics by Class:
##
##                      Class: A Class: B Class: C Class: D Class: E
## Sensitivity            0.8248   0.6278   0.6440   0.7395   0.5908
## Specificity            0.9124   0.9224   0.8992   0.9154   0.9724
## Pos Pred Value         0.7891   0.6600   0.5743   0.6315   0.8280
## Neg Pred Value         0.9291   0.9117   0.9228   0.9472   0.9135
## Prevalence             0.2845   0.1935   0.1744   0.1639   0.1838
## Detection Rate         0.2346   0.1215   0.1123   0.1212   0.1086
## Detection Prevalence   0.2973   0.1840   0.1955   0.1919   0.1311
## Balanced Accuracy      0.8686   0.7751   0.7716   0.8274   0.7816
```

Immediately reviewing the accuracy of the model we see it is low at 69% and the test/out of sample set is nearly equal (~69%). The Author's hypothesis is that the low performance is attributable to the assumptions for the distributions of variables for each class.

Author's rationale for building secondary model: Let's attempt to make a better model. For our second model we will use Random Forests. Random forests are selected as they exhibit the following characteristics:

1. They are an all-purpose model performing well on most problems
2. The approach inherently performs feature selection. This is advantageous as we are providing all available variables.
3. Handles large amounts of data well and does not have a lengthy processing time in the Author's experience

```
model_rf <- randomForest(classe ~. , data=final_train)
model_rf
```

```
##
## Call:
##  randomForest(formula = classe ~ ., data = final_train)
##                Type of random forest: classification
##                      Number of trees: 500
## No. of variables tried at each split: 7
##
##         OOB estimate of  error rate: 0.77%
## Confusion matrix:
##      A    B    C    D    E class.error
## A 3343    2    1    1    1 0.001493429
## B   18 2253    8    0    0 0.011408513
## C    0   15 2038    1    0 0.007789679
## D    0    0   32 1895    3 0.018134715
## E    0    0    2    7 2156 0.004157044
```

Author's Expectation of out of sample error: Reviewing the model above we see the OOB (out-of-bag) error estimate is .7% (that is less than 1%). From our lectures we know the OOB estimate for the generalization error is the error rate for the out-of-bag passes on the training set *(see 'Out-of-Bag Estimation', Leo Breiman, 1996)*. Thus we may reasonably conclude that our out of sample error will be near 1% (equivalently our accuracy will be 99%).

```
rf_predictions <- predict(model_rf, cross_validation_test, type = "class")
confusionMatrix(rf_predictions, cross_validation_test$classe)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction    A    B    C    D    E
##          A 2232    7    0    0    0
##          B    0 1506   18    0    0
##          C    0    5 1350   15    0
##          D    0    0    0 1271    6
##          E    0    0    0    0 1436
##
## Overall Statistics
##
##                Accuracy : 0.9935
##                  95% CI : (0.9915, 0.9952)
##     No Information Rate : 0.2845
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.9918
##  Mcnemar's Test P-Value : NA
##
## Statistics by Class:
##
##                      Class: A Class: B Class: C Class: D Class: E
## Sensitivity            1.0000   0.9921   0.9868   0.9883   0.9958
## Specificity            0.9988   0.9972   0.9969   0.9991   1.0000
## Pos Pred Value         0.9969   0.9882   0.9854   0.9953   1.0000
## Neg Pred Value         1.0000   0.9981   0.9972   0.9977   0.9991
## Prevalence             0.2845   0.1935   0.1744   0.1639   0.1838
## Detection Rate         0.2845   0.1919   0.1721   0.1620   0.1830
## Detection Prevalence   0.2854   0.1942   0.1746   0.1628   0.1830
## Balanced Accuracy      0.9994   0.9946   0.9919   0.9937   0.9979
```

Applying to the CV test set we see results are better and at 99% accuracy. This further supports the author's view that the model will generalize well on more out of sample data.

Applying Prediction to Testing Data: From our discussion above we see the better model is the Random Forest.

Let's acquire our Coursera 20 test set data and prepare according to the data preparation steps performed above.

Then we'll apply our random forest model to the Coursera testing data set and output the results via the function provided in the Coursera site. These will be uploaded for the second assignment.

```
URL <- "https://d396qusza40orc.cloudfront.net/predmachlearn/pml-testing.csv"; temp <- tempfile(); downl
test<- read.csv(temp,na.strings = c('','NA','#DIV/0!'))

coursera_test <- test[,!names(test) %in% NA_cols]
coursera_test <- coursera_test[complete.cases(coursera_test),]
coursera_test <- coursera_test[,-1:-7]
```

```
coursera_test_predictions <- predict(model_rf, coursera_test, type="class")

pml_write_files = function(x){
  n = length(x)
  for(i in 1:n){
    filename = paste0("problem_id_",i,".txt")
    write.table(x[i],file=filename,quote=FALSE,row.names=FALSE,col.names=FALSE)
  }
}

pml_write_files(coursera_test_predictions)
```