Practical Machine Learning Assignment

Paul Yap

19 March, 2015

Background & Objective

6 participants were asked to perform barbell lifts correctly and incorrectly in 5 different ways.Participants were asked to perform one set of 10 repetitions of the Unilateral Dumbbell Biceps Curl in five different fashions: exactly according to the specification (Class A), throwing the elbows to the front (Class B), lifting the dumbbell only halfway (Class C), lowering the dumbbell only halfway (Class D) and throwing the hips to the front (Class E). Class A corresponds to the specified execution of the exercise, while the other 4 classes correspond to common mistakes.

Source: 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)

Therefore, our objective is to predict the manner in which the participants exercise (Class A - E, as the classe" variable), based on the various data sources gathered.

Exploratory Data

```
setwd("/Users/paulyap/R_Stat/Coursera/8 Machine Learning/Week 3/Project")
library(Hmisc)

URLTrain<-"https://d396qusza40orc.cloudfront.net/predmachlearn/pml-training.csv"
URLTest<-"https://d396qusza40orc.cloudfront.net/predmachlearn/pml-testing.csv"
download.file(URLTrain,destfile="pml-training.csv",method="curl")
download.file(URLTest,destfile="pml-testing.csv",method="curl")
tblTrain<-read.csv("pml-training.csv",header=TRUE,na.strings=c("NA","#DIV/0!",""))
tblTest<-read.csv("pml-testing.csv",header=TRUE,na.strings=c("NA","#DIV/0!",""))
```

The data has many potentially redundant columns and incomplete entries, as identified below. Fortunately, for these potentially redundant columns, approximately 98% of the data are missing in each variable field, which would render them not meaningful.

```
dim(tblTrain)
```

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

```
sumNAtblTrain<-apply(tblTrain,2,function(x){sum(is.na(x))})
```

We will then exclude these columns from the training data sets. Additional variables that are not key to our analysis are also excluded.

```
#removes columns with NAs
validTrain <- subset(tblTrain[, which(sumNAtblTrain == 0)])
validTrain <-validTrain[,colnames(validTrain)%nin%c('X', 'user_name', 'raw_timestamp_par
t_1', 'raw_timestamp_part_2', 'cvtd_timestamp', 'new_window', 'num_window')]
```

Proposed Approach: Random Forest Model

As this is a non-linear prediction model requiring high level of accuracy, a Random Forest prediction modelling is adopted.

```
library(caret)
library(randomForest)
set.seed(100)
inTrain <- createDataPartition(y=validTrain$classe, p=0.7, list=F)
training <- validTrain[inTrain,]
testing <- validTrain[-inTrain,]
model <- randomForest(classe ~., data=training, type="classification")
```

Evaluating the model using training and testing sets

The results from the training sets suggest a very highly accurate model (in-sample accuracy of 100%, that is the rate at which its prediction is correct); we then put our prediction model to the test using the testing set, which had not been used to build our prediction model.

We should reasonably expect that the testing set (out-of-sample) will achieve a lower level of accuracy. Let's now dwell deeper into the out of sample error, with an estimation of the error using the appropriate cross-validation test technique (confusionMatrix).

```
predmodeltrain <- predict(model,newdata=training)
confusionMatrix(predmodeltrain,training$classe)$table
```

```
##           Reference
## Prediction    A    B    C    D    E
##          A 3906    0    0    0    0
##          B    0 2658    0    0    0
##          C    0    0 2396    0    0
##          D    0    0    0 2252    0
##          E    0    0    0    0 2525
```

Out of Sample Cross Validation Test

The results from the out-of-sample set suggests that our prediction model is still highly accurate (out-of-sample accuracy of 99.6%), despite some incorrect predictions (26 out of 5,887). It also has high specificity (predicting “No” when it is actually “No”) and sensitivity (predicting “Yes” when it is actually “Yes”).

```
predmodeltest <- predict(model,newdata=testing)
confusionMatrix(predmodeltest,testing$classe)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction    A    B    C    D    E
##          A 1673    6    0    0    0
##          B    1 1133    3    0    0
##          C    0    0 1021   11    1
##          D    0    0    2  952    1
##          E    0    0    0    1 1080
##
## Overall Statistics
##
##                Accuracy : 0.9956
##                  95% CI : (0.9935, 0.9971)
##     No Information Rate : 0.2845
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.9944
##  Mcnemar's Test P-Value : NA
##
## Statistics by Class:
##
##                      Class: A Class: B Class: C Class: D Class: E
## Sensitivity            0.9994   0.9947   0.9951   0.9876   0.9982
## Specificity            0.9986   0.9992   0.9975   0.9994   0.9998
## Pos Pred Value         0.9964   0.9965   0.9884   0.9969   0.9991
## Neg Pred Value         0.9998   0.9987   0.9990   0.9976   0.9996
## Prevalence             0.2845   0.1935   0.1743   0.1638   0.1839
## Detection Rate         0.2843   0.1925   0.1735   0.1618   0.1835
## Detection Prevalence   0.2853   0.1932   0.1755   0.1623   0.1837
## Balanced Accuracy      0.9990   0.9969   0.9963   0.9935   0.9990
```

Conclusion

It is perhaps unusual to be getting such a high level of accuracy that is close to 100% for both the in-sample and out-of-sample sets. Given that only 6 participants were involved, the results collected could be biased, with more predictable behaviours and less variances in the variables. With more diverse participants, the predictability/ accuracy of the model is likely to be reduced.

Appendix

Running the Course Project Submission

```
tblTest<-read.csv("pml-testing.csv",header=TRUE,na.strings=c("NA","#DIV/0!",""))
sumNAtblTest<-apply(tblTest,2,function(x){sum(is.na(x))})
validtblTest <- subset(tblTest[, which(sumNAtblTest == 0)])
validtblTest <-validtblTest[,colnames(validtblTest)%nin%c('X', 'user_name', 'raw_timesta
mp_part_1', 'raw_timestamp_part_2', 'cvtd_timestamp', 'new_window', 'num_window')]
predvalidtblTest <- predict(model,newdata=validtblTest)
```