Machine

TT

28 September 2015

Management Summary

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).

Steps are taken:

1., Load the data

2., Clean the data, removing NAs, etc.. The number of columns are reduced to 60 from 160

3., Create training and test datasets

4., Create Rpart Model

5., Create Random Forest Model

6., Select Model based on accuracy. Random forest model has more than 99.9% accuracy while Rpart has 66% accuracy

7., The Out-of-Sample Error Rate for Random forest near 0

Analysis

```
library(caret)
library(rpart)
library(randomForest)

## Load Data
setwd("C:/Users/HP/Documents/Tibi/R/Machine_Learning")
trainRaw <- read.csv("./data/trainData.csv", header = TRUE, sep = ",", na.strings = c("NA", "#DIV/0!",
testRaw <- read.csv("./data/testData.csv", header = TRUE, sep = ",", na.strings = c("NA", "#DIV/0!", ""

##Clean Data Set
trainClean <- trainRaw[,colSums(is.na(trainRaw))<(nrow(trainRaw)*0.9)]
testClean <- testRaw[,colSums(is.na(trainRaw))<(nrow(trainRaw)*0.9)]

##Create Subsets
set.seed(33833)
inTrain = createDataPartition(trainClean$classe, p = .60)[[1]]
trainingsubset = trainClean[ inTrain,]
testingsubset = trainClean[-inTrain,]
```

```
##Create Rpart Model
modFit<-train(classe~.,method="rpart", data=trainingsubset)

varImp(modFit)
```

```
## rpart variable importance
##
##   only 20 most important variables shown (out of 81)
##
##                                  Overall
## X                                100.000
## roll_belt                         34.030
## accel_belt_z                      20.090
## magnet_belt_y                     18.316
## pitch_forearm                     12.904
## total_accel_belt                   8.615
## user_nameeurico                    0.000
## pitch_dumbbell                     0.000
## total_accel_forearm                0.000
## magnet_arm_z                       0.000
## gyros_forearm_z                    0.000
## pitch_belt                         0.000
## num_window                         0.000
## `cvtd_timestamp02/12/2011 14:58`   0.000
## `cvtd_timestamp30/11/2011 17:11`   0.000
## gyros_arm_y                        0.000
## raw_timestamp_part_1               0.000
## total_accel_dumbbell               0.000
## gyros_dumbbell_x                   0.000
## magnet_dumbbell_z                  0.000
```

```
classepredict=predict(modFit,testingsubset)
confusionMatrix(testingsubset$classe,classepredict)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction    A    B    C    D    E
##          A 2232    0    0    0    0
##          B    0 1517    0    0    1
##          C    0    0    0    0 1368
##          D    0    0    0    0 1286
##          E    0    0    0    0 1442
##
## Overall Statistics
##
##                Accuracy : 0.6616
##                  95% CI : (0.651, 0.6721)
##     No Information Rate : 0.5222
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.5693
```

```
##  Mcnemar's Test P-Value : NA
##
## Statistics by Class:
##
##                      Class: A Class: B Class: C Class: D Class: E
## Sensitivity            1.0000   1.0000       NA       NA   0.3520
## Specificity            1.0000   0.9998   0.8256   0.8361   1.0000
## Pos Pred Value         1.0000   0.9993       NA       NA   1.0000
## Neg Pred Value         1.0000   1.0000       NA       NA   0.5854
## Prevalence             0.2845   0.1933   0.0000   0.0000   0.5222
## Detection Rate         0.2845   0.1933   0.0000   0.0000   0.1838
## Detection Prevalence   0.2845   0.1935   0.1744   0.1639   0.1838
## Balanced Accuracy      1.0000   0.9999       NA       NA   0.6760
```

```
##Create Random Forest Model
modFit2 <- train(classe ~ ., method="rf",trControl=trainControl(method = "cv", number = 4), data=trainin

varImp(modFit2)
```

```
## rf variable importance
##
##   only 20 most important variables shown (out of 81)
##
##                                 Overall
## X                              100.0000
## roll_belt                        7.3142
## raw_timestamp_part_1             2.1492
## pitch_forearm                    1.9148
## accel_belt_z                     1.3809
## roll_dumbbell                    1.0986
## num_window                       0.9413
## magnet_belt_y                    0.7910
## accel_forearm_x                  0.6874
## cvtd_timestamp02/12/2011 14:57   0.5944
## magnet_dumbbell_y                0.5236
## total_accel_belt                 0.5119
## cvtd_timestamp30/11/2011 17:12   0.4321
## yaw_belt                         0.3446
## pitch_belt                       0.3280
## cvtd_timestamp30/11/2011 17:11   0.3119
## accel_dumbbell_y                 0.2588
## pitch_dumbbell                   0.2353
## roll_forearm                     0.2338
## magnet_dumbbell_z                0.2276
```

```
classepredict2=predict(modFit2,testingsubset)
confusionMatrix(testingsubset$classe,classepredict2)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction    A    B    C    D    E
##          A 2232    0    0    0    0
```

```
##          B    0 1518    0    0    0
##          C    0    0 1368    0    0
##          D    0    0    0 1286    0
##          E    0    0    0    0 1442
##
## Overall Statistics
##
##                Accuracy : 1
##                  95% CI : (0.9995, 1)
##     No Information Rate : 0.2845
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 1
##  Mcnemar's Test P-Value : NA
##
## Statistics by Class:
##
##                      Class: A Class: B Class: C Class: D Class: E
## Sensitivity            1.0000   1.0000   1.0000   1.0000   1.0000
## Specificity            1.0000   1.0000   1.0000   1.0000   1.0000
## Pos Pred Value         1.0000   1.0000   1.0000   1.0000   1.0000
## Neg Pred Value         1.0000   1.0000   1.0000   1.0000   1.0000
## Prevalence             0.2845   0.1935   0.1744   0.1639   0.1838
## Detection Rate         0.2845   0.1935   0.1744   0.1639   0.1838
## Detection Prevalence   0.2845   0.1935   0.1744   0.1639   0.1838
## Balanced Accuracy      1.0000   1.0000   1.0000   1.0000   1.0000
```

```
##Model Selection

##Count the Model Accuracy
modelaccuracy <- postResample(testingsubset$classe, predict(modFit, testingsubset[,-60]))[[1]]
modelaccuracy
```

```
## [1] 0.661611
```

```
modelaccuracy2 <- postResample(testingsubset$classe, predict(modFit2, testingsubset[,-60]))[[1]]
modelaccuracy2
```

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

```
##Count the Out-of-Sample Error Rate
outOfSampleError <- 1 - modelaccuracy2
outOfSampleError
```

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

Reference

The data for this project come from this source: http://groupware.les.inf.puc-rio.br/har