Prediction with Machine Learning

Date: 22 Mar 2015

1. Objective

The objective of this project is to predict the manner in which participant did the exercise as measured by the “classe” variable. Six participants were asked to perform barbell lifts correctly and incorrectly in 5 different ways.The data from accelerometers on the belt, forearm, arm and dumbell of the participants were recorded to see how well they do it.

2. Data Source

The training data for this project are available here:
https://d396qusza40orc.cloudfront.net/predmachlearn/pml-training.csv
(https://d396qusza40orc.cloudfront.net/predmachlearn/pml-training.csv)

The test data are available here: https://d396qusza40orc.cloudfront.net/predmachlearn/pml-testing.csv
(https://d396qusza40orc.cloudfront.net/predmachlearn/pml-testing.csv)

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

Save the datasets into your working directory.

3. Data Transformation

To use non-zero and NA values of belt, arm, dumbbell, and forearm variables as predictors of classe.

```
missingvar <- sapply(dataset, function (x) any(is.na(x)|x==""))
predictvar <- !missingvar & grepl("belt|[^(fore)]arm|dumbbell|forearm", names(missingvar
))
predictors <- names(missingvar)[predictvar]
predictors
```

```
##  [1] "roll_belt"            "pitch_belt"           "yaw_belt"
##  [4] "total_accel_belt"     "gyros_belt_x"         "gyros_belt_y"
##  [7] "gyros_belt_z"         "accel_belt_x"         "accel_belt_y"
## [10] "accel_belt_z"         "magnet_belt_x"        "magnet_belt_y"
## [13] "magnet_belt_z"        "roll_arm"             "pitch_arm"
## [16] "yaw_arm"              "total_accel_arm"      "gyros_arm_x"
## [19] "gyros_arm_y"          "gyros_arm_z"          "accel_arm_x"
## [22] "accel_arm_y"          "accel_arm_z"          "magnet_arm_x"
## [25] "magnet_arm_y"         "magnet_arm_z"         "roll_dumbbell"
## [28] "pitch_dumbbell"       "yaw_dumbbell"         "total_accel_dumbbell"
## [31] "gyros_dumbbell_x"     "gyros_dumbbell_y"     "gyros_dumbbell_z"
## [34] "accel_dumbbell_x"     "accel_dumbbell_y"     "accel_dumbbell_z"
## [37] "magnet_dumbbell_x"    "magnet_dumbbell_y"    "magnet_dumbbell_z"
## [40] "roll_forearm"         "pitch_forearm"        "yaw_forearm"
## [43] "total_accel_forearm"  "gyros_forearm_x"      "gyros_forearm_y"
## [46] "gyros_forearm_z"      "accel_forearm_x"      "accel_forearm_y"
## [49] "accel_forearm_z"      "magnet_forearm_x"     "magnet_forearm_y"
## [52] "magnet_forearm_z"
```

Reduce the columns in the original dataset to include only predictors of classe

```
dataR <- dataset[,c("classe",predictors)]
```

Number of observations in each classe

```
summary(dataR$classe)
```

```
##    A    B    C    D    E
## 5580 3797 3422 3216 3607
```

To carry out cross validation, we split the reduced dataset into 60% training and 40% probing.

```
set.seed(1234)
inTrain <- createDataPartition(dataR$classe, p=0.6, list=FALSE)
Train <- dataR[inTrain,]
Probe <- dataR[-inTrain,]
```

4. Develop Prediction Model

Apply classification tree to narrow down the number of predictors before running the Random Forest Algorithm.

```
set.seed(1234)
model1 <-train(classe~., data=Train, method="rpart")
```

```
## Loading required package: rpart
```

```
## Warning: package 'rpart' was built under R version 3.1.2
```

```
print(model1$finalModel)
```

```
## n= 11776
##
## node), split, n, loss, yval, (yprob)
##       * denotes terminal node
##
##  1) root 11776 8428 A (0.28 0.19 0.17 0.16 0.18)
##    2) roll_belt< 130.5 10774 7436 A (0.31 0.21 0.19 0.18 0.11)
##      4) pitch_forearm< -34.55 919    2 A (1 0.0022 0 0 0) *
##      5) pitch_forearm>=-34.55 9855 7434 A (0.25 0.23 0.21 0.2 0.12)
##       10) magnet_dumbbell_y< 436.5 8314 5944 A (0.29 0.18 0.24 0.19 0.11)
##         20) roll_forearm< 122.5 5137 3022 A (0.41 0.18 0.18 0.17 0.061) *
##         21) roll_forearm>=122.5 3177 2124 C (0.08 0.18 0.33 0.23 0.18) *
##       11) magnet_dumbbell_y>=436.5 1541  743 B (0.033 0.52 0.039 0.23 0.18) *
##    3) roll_belt>=130.5 1002   10 E (0.01 0 0 0 0.99) *
```

```
varImp(model1)
```

```
## rpart variable importance
##
##   only 20 most important variables shown (out of 52)
##
##                   Overall
## pitch_forearm      100.00
## roll_forearm        74.53
## roll_belt           72.71
## magnet_dumbbell_y   53.09
## yaw_belt            44.68
## accel_belt_z        44.44
## magnet_belt_y       41.56
## total_accel_belt    37.68
## magnet_arm_x        25.95
## accel_arm_x         25.07
## roll_dumbbell       19.31
## magnet_dumbbell_x   19.04
## magnet_dumbbell_z   18.41
## accel_dumbbell_y    16.39
## roll_arm            15.98
## magnet_forearm_x     0.00
## gyros_arm_y          0.00
## magnet_forearm_z     0.00
## magnet_belt_z        0.00
## gyros_dumbbell_z     0.00
```

By studying the varImp, apply the smaller predictor-set to run the Random Forest Algorithm.

```
TrainR <- Train[,c("classe","magnet_dumbbell_y","pitch_forearm",
                   "roll_belt","roll_dumbbell","roll_forearm",
                   "accel_belt_z","magnet_dumbbell_x","magnet_belt_y",
                   "magnet_dumbbell_z","pitch_belt","total_accel_belt",
                   "magnet_arm_x","accel_arm_x","yaw_belt","accel_forearm_x",
                   "accel_dumbbell_y","gyros_belt_z","yaw_forearm")]
```

Refer to the Rmd document for the full documentation.