Machine Learning Project

Javier Eslava-Schmalbach

8/11/2018

Background from the assignment

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

Objective

The goal of this project is to predict the manner in which tech geeks did the exercise using the “classe” variable as the outcome to be predicted

Methodology

Databases will be downloaded from url addresses provided in the assignment: one for training and getting the best model and the second for predicting the behaviour of 20 specific cases. After that, they will be cleaned in the same way, deleting variables with NA values, Near Zero Variance, and not relevant with the objective. The training dataset will be splited in two subsets to facilitate cross-validation: training and testing data sets. All models will be tested in this testing dataset, and the best of all, will be used to predict the behaviour of cases of the test dataset provided for the exercise. A graphic comparison will be done with clusters got from the data in both datasets. In all models accuracy will be measured comparing their results with variable “class” of the subset(training). To evaluate accuracy in the test dataset, 20 randomly cases will be selected from the training dataset, and their respective “classe” values will be used to compare the predictive results in the Test dataset (as a second process of cross-validation), given that, this data did not have included the “class”" of the subjects.

Development

Getting the databases

```
       training <- read.csv(url("https://d396qusza40orc.cloudfront.net/predmachlearn/
pml-training.csv"), header = TRUE, sep=",", na.strings=c("NA","#DIV/0!",""))
        test <- read.table(url("https://d396qusza40orc.cloudfront.net/predmachlearn/p
ml-testing.csv"), sep = ",", header = TRUE, na.strings=c("NA","#DIV/0!",""))
```

Cleaning and deleting Near Zero Variance, NA and non relevant variables

```
        ## library(lubridate)
       dimension1 <- data.frame(cbind(dim(training), dim(test)))
        colnames(dimension1) <- c("initial dim training", "initial dim testing")
        rownames(dimension1) <- c("rows", "columns")
        dimension1
```

```
        initial dim training initial dim testing
rows                   19622                  20
columns                  160                 160
```

```
          ## deleting columns with NA values
        training <- training[,colSums(is.na(training)) == 0]
        test <- test[,colSums(is.na(test)) == 0]

         ## removing near zero covariance variables
        nsv <- nearZeroVar(training,saveMetrics=TRUE)

        ## removing first 5 variables, that include new window that it is a noon zero
variance variable
        training <- training[,-c(1:6)]
        test <- test[,-c(1:6)]

        ## dimensions after cleaning
         dimension2 <- data.frame(cbind(dim(training), dim(test)))
        colnames(dimension2) <- c("after dim training", "after dim testing")
        rownames(dimension2) <- c("rows", "columns")
        dimension2
```

```
        after dim training after dim testing
rows                 19622                20
columns                 54                54
```

Subsetting a database from the training dataset to test and cross-validate initial findings. Test database will be used to final validation

```
        set.seed(1234)
        inTrain <- createDataPartition(y=training$classe, p=0.7, list=FALSE)
        training1 = training[inTrain,]
        testing1 = training[-inTrain,]  ## from the training database

        dimension3 <- data.frame(cbind(dim(training1), dim(testing1)))
        colnames(dimension3) <- c("subset(training)", "subset(testing)")
        rownames(dimension3) <- c("rows", "columns")
        dimension3
```

```
        subset(training) subset(testing)
rows               13737            5885
columns               54              54
```

Looking for the best model

```
        modFit.rpart <- train(classe ~., method = "rpart",data=training1)
        modFit.rf <-  train(classe ~., method = "rf",data=training1)
        modFit.gbm <-  train(classe ~., method = "gbm",data=training1, verbose = FALS
E)
        modFit.lda <- train(classe ~., method = "lda",data=training1)

        predict.rpart <- predict(modFit.rpart, testing1)
        predict.rf <- predict(modFit.rf, testing1, type = "raw")
        predict.gbm <- predict(modFit.gbm, testing1)
        predict.lda <-  predict(modFit.lda, testing1)

        y <- testing1$classe
        accuracies <- data.frame(cbind(postResample(predict.rpart, y),
        postResample(predict.rf, y),
        postResample(predict.gbm, y),
        postResample(predict.lda, y)))
        colnames(accuracies) <- c("RPART model", "RF model", "GBM model", "LDA model"
)
        accuracies
```

```
         RPART model  RF model GBM model LDA model
Accuracy   0.4890399 0.9981308 0.9887850 0.7184367
Kappa      0.3311096 0.9976356 0.9858103 0.6435904
```

The bests models are Random Forest and Gradient Boosting Nachine (GBM). However, Random Forest is less time-consuming, and this model is seleted then, to be cross-validated in the Test dataset.

Characteristics of the selected model

```
        modFit.rf$finalModel
```

```
Call:
 randomForest(x = x, y = y, mtry = param$mtry)
               Type of random forest: classification
                     Number of trees: 500
No. of variables tried at each split: 27

        OOB estimate of  error rate: 0.23%
Confusion matrix:
     A    B    C    D    E  class.error
A 3904    1    0    0    1 0.0005120328
B    4 2650    3    1    0 0.0030097818
C    0    5 2390    1    0 0.0025041736
D    0    0    8 2243    1 0.0039964476
E    0    1    0    6 2518 0.0027722772
```

```
        importance <- data.frame(round(importance(modFit.rf$finalModel, 2), 2))
        importance
```

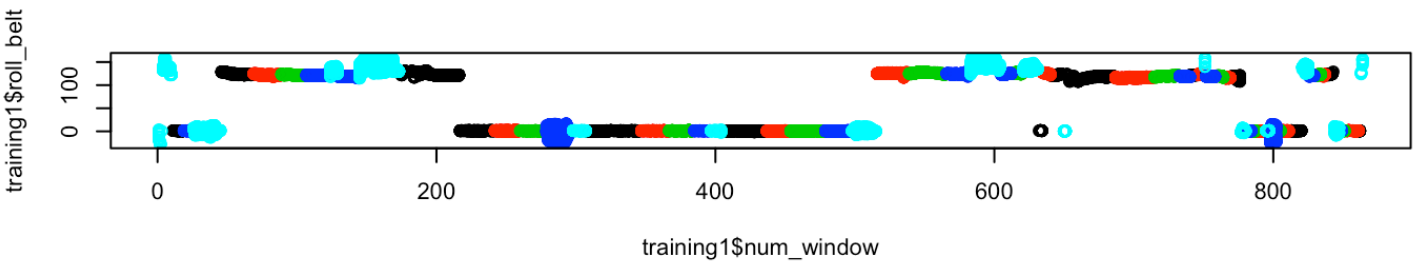
```
                     MeanDecreaseGini
num_window                     2001.93
roll_belt                      1282.42
pitch_belt                      565.28
yaw_belt                        677.28
total_accel_belt                 57.80
gyros_belt_x                     29.41
gyros_belt_y                     40.45
gyros_belt_z                    101.71
accel_belt_x                     30.51
accel_belt_y                     46.82
accel_belt_z                    189.66
magnet_belt_x                   124.44
magnet_belt_y                   159.01
magnet_belt_z                   173.87
roll_arm                        104.51
pitch_arm                        64.86
yaw_arm                          80.14
total_accel_arm                  31.27
gyros_arm_x                      34.90
gyros_arm_y                      47.20
gyros_arm_z                      14.35
accel_arm_x                      87.16
accel_arm_y                      41.80
accel_arm_z                      37.51
magnet_arm_x                     64.42
magnet_arm_y                     67.38
magnet_arm_z                     47.69
roll_dumbbell                   216.22
```

```
pitch_dumbbell                61.81
yaw_dumbbell                 108.61
total_accel_dumbbell         202.19
gyros_dumbbell_x              36.11
gyros_dumbbell_y              85.91
gyros_dumbbell_z              23.01
accel_dumbbell_x              94.15
accel_dumbbell_y             255.95
accel_dumbbell_z             158.50
magnet_dumbbell_x            234.08
magnet_dumbbell_y            587.69
magnet_dumbbell_z            575.36
roll_forearm                 427.22
pitch_forearm                774.84
yaw_forearm                   75.90
total_accel_forearm           31.26
gyros_forearm_x               16.96
gyros_forearm_y               32.86
gyros_forearm_z               25.65
accel_forearm_x              209.42
accel_forearm_y               38.04
accel_forearm_z               97.68
magnet_forearm_x              73.43
magnet_forearm_y              73.51
magnet_forearm_z             139.85
```

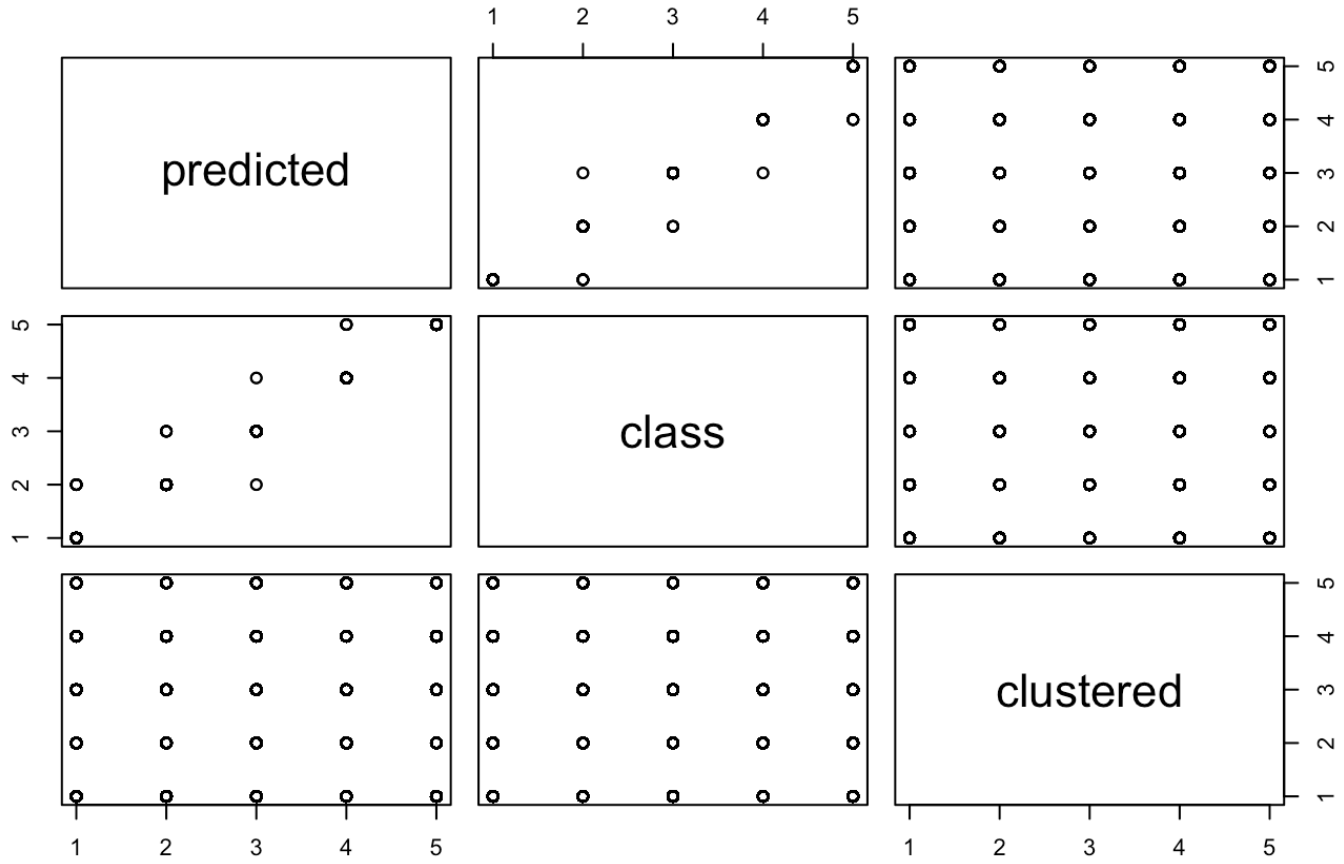
The most important variables in the model are num_window, roll belt and pitch forearm

Plotting prediction and clustering results against original class variable

```
        ## library(kernlab); library(ade4)

        training1s <- subset(training1, select=-c(54))

        kmeans1 <- kmeans(na.omit(training1s),centers=5)
        kmeansRes<-factor(kmeans1$cluster)


        combined <- data.frame(cbind(predicted=predict.rf, class=y, clustered=kmeans1
$cluster))

        testing1s <- subset(testing1, select=-c(54))
        kmeans2 <- kmeans(na.omit(testing1s),centers=5)
        kmeansRes2<-factor(kmeans2$cluster)


        combined <- data.frame(cbind(predicted=predict.rf, class=y, clustered=kmeans1
$cluster))
        plot.new()
        par(mfrow = c(3,1));

        plot(training1$num_window, training1$roll_belt, col=training1$classe)
        plot(combined, main="Random Forest model, original and clustered classes, tra
ining subset")
```

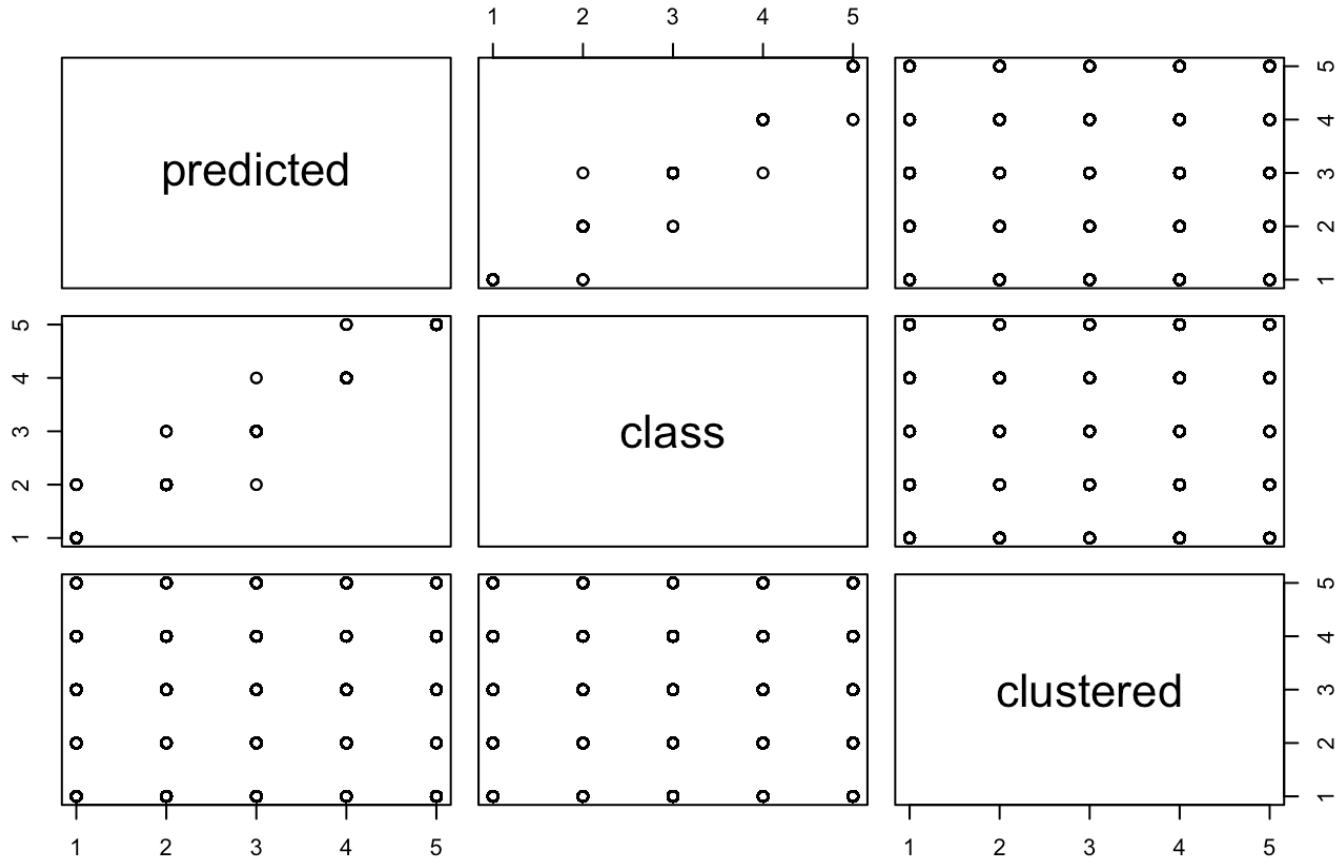

training1$roll_belt
0
100
0
200
400
600
800
training1$num_window

Random Forest model, original and clustered classes, training subset




```
        plot(combined, main="Random Forest model, original and clustered classes, tes
ting subset")
```

Random Forest model, original and clustered classes, testing subset



In the Figure is evident that Cluster analysis could not identify as good as Random Forest does, the classes of subjects.

Prediction of Cases in the data, and accuracy of the Random Forest model, using 20 classes selected randomly from the training dataset.

```
        set.seed(123) ## selecting a sample of 20 classes from the training dataset

        training20 <- dplyr::sample_n(training, 20)
        y <- training20$classe

        predict.rparttest <- predict(modFit.rpart, test)
        predict.rftest <- predict(modFit.rf, test, type = "raw")
        predict.gbmtest <- predict(modFit.gbm, test)
        predcit.ldatest <-  predict(modFit.lda, test)


        accuracies1 <- data.frame(cbind(postResample(predict.rparttest, y),
        postResample(predict.rftest, y),
        postResample(predict.gbmtest, y),
        postResample(predcit.ldatest, y)))
        colnames(accuracies1) <- c("RPART model", "RF model", "GBM model", "LDA model
")
        accuracies1
```

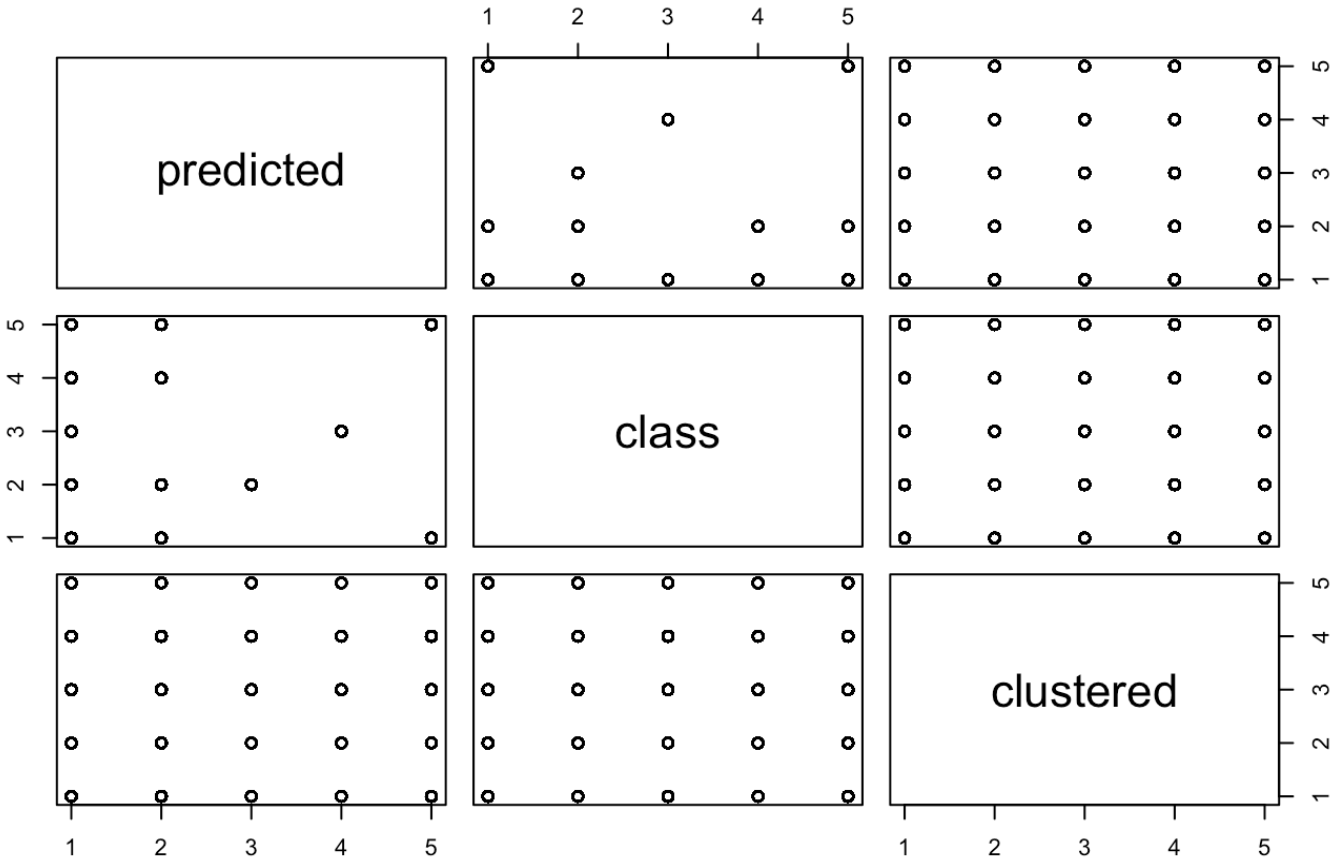
```
          RPART model   RF model  GBM model    LDA model
Accuracy  0.15000000 0.25000000 0.25000000  0.200000000
Kappa    -0.03343465 0.02912621 0.02912621 -0.009463722
```

Accuracies are extremely low in this final dataset. It could be explained, because the “classe” variable of these data, was build from a randaom sample of 20 data, from training dataset. However, Random Forest keep being the best of all models.

Cross-vallidation with the Test dataset, and prediction of the 20 cases

```
        kmeans2 <- kmeans(testing1s,centers=5)
        kmeansRes2<-factor(kmeans2$cluster)
        plot.new()
         par(mfrow = c(1,1))


        combined <- data.frame(cbind(predicted=predict.rftest, class=y, clustered=kme
ans2$cluster))
        plot(combined, main="Random Forest model, original and clustered classes, Tes
t dataset")
```

Random Forest model, original and clustered classes, Test dataset



Prediction is not as good as it was in the training dataset, using the Test Dataset

With the results of this predictive random forest analysis, the quiz of 20 cases was answered.

Acknowledgments.

To the developers of the original research available (here)[http://groupware.les.inf.puc-rio.br/har (http://groupware.les.inf.puc-rio.br/har)], in the section on the “Weight Lifting Exercise Dataset”