PRACTICAL MACHINE LEARNING-ASSIGNMENT

CARMEN LUQUE

Saturday, October 18, 2014

ABSTRACT

The goal of this assignment is to predict the manner in which six participants did the exercise. The data for this project come from the source http://groupware.les.inf.puc-rio.br/har (http://groupware.les.inf.puc-rio.br/har) (Weight Lifting Exercises Dataset). The activity-quality classes are categorized in 5 different ways: A, B, C, D y E. we will have to build a model using the random forest method and cross validation. Then apply the final model to predict the 20 test cases that not classified.

LIBRARIES

First we load the necessary libraries to run our task.

```
library(corrplot)
library(caret)
```

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

```
library(randomForest)
```

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

```
library(kernlab)
library(e1071)
```

DOWNLOAD FILES

We download the files in the selected directory. For it will indicate the url addresses of the files and the directory on the local computer where you want to save. Once the download of the files we will proceed to

read the csv file for training

```
download.file("http://d396qusza40orc.cloudfront.net/predmachlearn/pml-testing.csv","/users/xxx/downloads/pml-testing.csv")
download.file("http://d396qusza40orc.cloudfront.net/predmachlearn/pml-training.csv","/users/xxx/downloads/pml-training.csv")
data_training <- read.csv ("/Users/xxx/Downloads/pml-training.csv",header=TRUE,na.strings=c("NA","","#DIV/0!"))
```

DATA: CLEANING AND PROCESSING

1.- CLEANING DATA

Before processing the information is necessary to clean the data. To do this we will remove the columns that may generate noise in the output, such as columns with data columns blank or with unnecessary information such as name, timestamp, etc.

```
data_training_NA <- apply(data_training, 2, function(x) {sum(is.na(x))})
data_training_LIMPIO <- subset(data_training[,which(data_training_NA == 0)],select=-c(X,user_name, raw_timestamp_part_1, raw_timestamp_part_2, cvtd_timestamp , new_window, num_window))
```

2.- SPLIT DATA

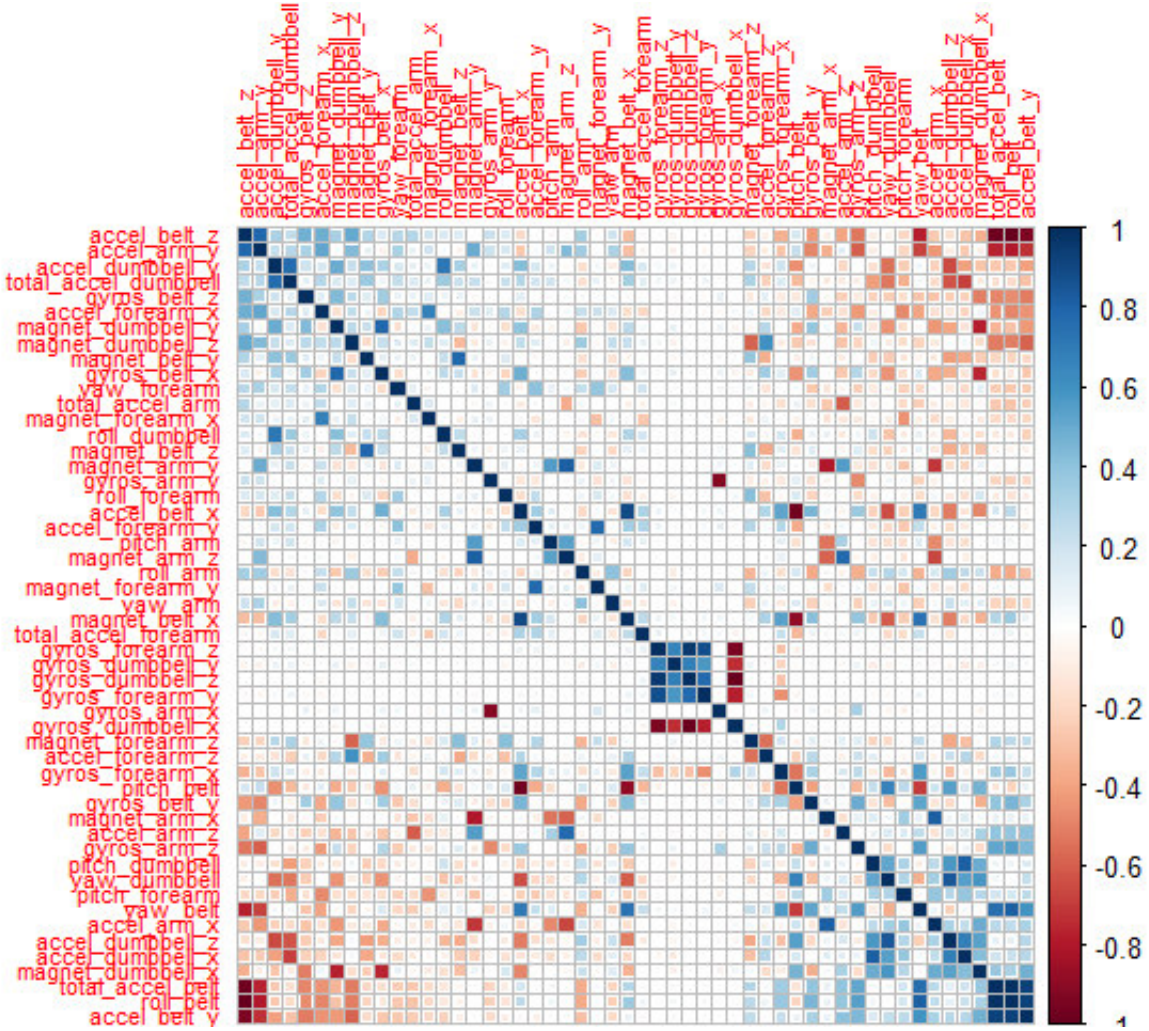
Now we split the clean data into a training dataset (“training”) and a validation dataset (“testing”). The particion is 70/30.

```
inTrain <- createDataPartition(y = data_training_LIMPIO$classe, p = 0.7, list = FALSE)
training <- data_training_LIMPIO[inTrain, ]
testing <- data_training_LIMPIO[-inTrain, ]
```

3.- CORRELATION MATRIX

To study the correlation between variables we perform the representation of the correlation matrix

```
correlMatrix <- cor(training[, -length(training)])
corrplot(correlMatrix, order = "FPC", method = "square", type = "full", tl.cex = 0.7)
```



4.- RANDOM FOREST and CROSS-VALIDATION. CONFUSION MATRIX

Now we will build the training model using the random forest method and applying a cross-validation (n = 4). We then generate the confusion matrix to determine the effectiveness of the generated model.Each column of the matrix represents the instances in a predicted class, while each row represents the instances in an actual class.

```
CONTROL <-trainControl(method="cv",number=4)
model <- train(classe ~.,data=training, model="rf",trControl=CONTROL)
predictCVal <- predict(model,newdata=testing)
mconfusion <- confusionMatrix(testing$classe,predictCVal)
mconfusion$table
```

```
##            Reference
## Prediction    A    B    C    D    E
##          A 1672    1    1    0    0
##          B    4 1132    3    0    0
##          C    0    4 1021    1    0
##          D    0    0    8  955    1
##          E    0    0    0    4 1078
```

```
salida_accu <-print(mconfusion$overall)
```

```
##       Accuracy          Kappa  AccuracyLower  AccuracyUpper   AccuracyNull
##      0.9954121      0.9941966      0.9933318      0.9969744      0.2847918
## AccuracyPValue  McnemarPValue
##      0.0000000            NaN
```

5.- ACCURACY

The accuracy is the proportion of true results (both true positives and true negatives). The estimated accuracy of this model is 99.54%.

```
accu <- salida_accu[[1]]
accu
```

```
## [1] 0.9954121
```

6.- OUT-OF-SAMPLE ERROR

For out-of-sample errors subtract 1 minus the value of the accuracy. The estimated out-of-sample error obtained is 0.46%.

```
out_of_sample <- 1-accu
out_of_sample
```

```
## [1] 0.004587935
```

PREDICTED RESULTS

Having overcome the previous phases, will now apply the same initial steps of data cleaning to the DataTest. After cleaning unnecessary data, we applied our model execute against testing dataset and obtain results of prediction for the 20 proposed cases.

```
data_test <- read.csv("/users/xxx/downloads/pml-testing.csv",header=TRUE,na.strings=c("NA","","
 "))
data_test_NA <- apply(data_test, 2, function(x) {sum(is.na(x))})
data_test_LIMPIO <- data_test[,which(data_test_NA == 0)]
data_test_LIMPIO <- subset(data_test[,which(data_test_NA == 0)],select=-c(X,user_name, raw_time
stamp_part_1, raw_timestamp_part_2, cvtd_timestamp , new_window, num_window))
prediccionTest <- predict(model, data_test_LIMPIO)
prediccionTest
```

```
##  [1] B A B A A E D B A A B C B A E E A B B B
## Levels: A B C D E
```

Through this function we will obtain 20 text file with the contents of the prediction for each case.

```
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(prediccionTest)
```