Practical_Machine_Learning_modeling

Mohamed Azar El Mourabit
3 de febrero de 2018

This document is the Final Project for the MOOC “Practical Machine Learning”" from Johns Hopkins University. Assignement Instructions

(Background, Data and What you should submit sections are directly copied from course’s assignement page)

Background

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).

Data

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

The test data are available here: [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]. If you use the document you create for this class for any purpose please cite them as they have been very generous in allowing their data to be used for this kind of assignment.

What you should submit

The goal of your project is to predict the manner in which they did the exercise. This is the “classe” variable in the training set. You may use any of the other variables to predict with. You should create a report describing how you built your model, how you used cross validation, what you think the expected out of sample error is, and why you made the choices you did. You will also use your prediction model to predict 20 different test cases.

Your submission should consist of a link to a Github repo with your R markdown and compiled HTML file describing your analysis. Please constrain the text of the writeup to < 2000 words and the number of figures to be less than 5. It will make it easier for the graders if you submit a repo with a gh-pages branch so the HTML page can be viewed online (and you always want to make it easy on graders :-). You should also apply your machine learning algorithm to the 20 test cases available in the test data above. Please submit your predictions in appropriate format to the programming assignment for automated grading. See the programming assignment for additional details.

Let’s import the libraries wich we need for run our code .

```
library(e1071)
library(lattice)
library(ggplot2)
library(caret) # Caret package
library(randomForest) #Random forest for classification and regression

## randomForest 4.6-12

## Type rfNews() to see new features/changes/bug fixes.

##
## Attaching package: 'randomForest'

## The following object is masked from 'package:ggplot2':
##
##     margin

library(rpart) # Regressive Partitioning and Regression trees
library(rpart.plot) # Decision Tree plot

# setting the overall seed for reproduceability
set.seed(1234)
```

Getting Data

Let’s read the data and replace the missing values with NA

```
# Loading the training data set  replacing all missing with "NA"
trainingset <- read.csv("pml-training.csv", na.strings=c("NA","#DIV/0!", ""))

# Loading the testing data set
testingset <- read.csv('pml-testing.csv', na.strings=c("NA","#DIV/0!", ""))
```

Checking the dimensions of our training and testing sets.

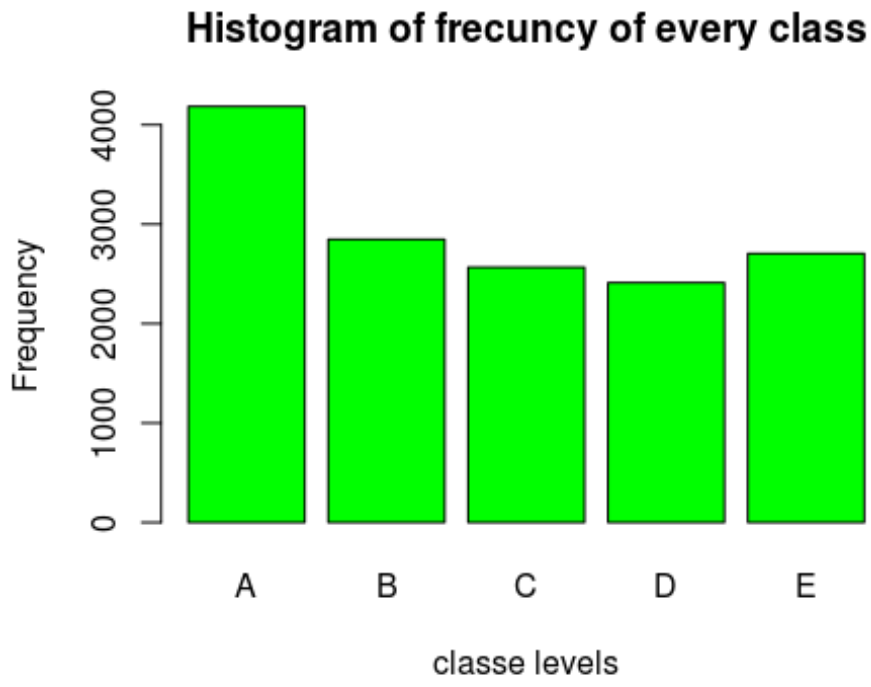
```
# Check dimensions for number of variables and number of observations
dim(trainingset)

## [1] 19622   160

dim(testingset)

## [1]  20 160

# Delete columns with all missing values
trainingset<-trainingset[,colSums(is.na(trainingset)) == 0]
testingset <-testingset[,colSums(is.na(testingset)) == 0]

# Some variables are irrelevant to our current project: user_name,
raw_timestamp_part_1, raw_timestamp_part_,2 cvtd_timestamp,
new_window, and  num_window (columns 1 to 7). We can delete these
variables.
trainingset   <-trainingset[,-c(1:7)]
testingset <-testingset[,-c(1:7)]
```

Data Spliting

```
subsamples <- createDataPartition(y=trainingset$classe, p=0.75,
list=FALSE)
subTraining <- trainingset[subsamples, ]
subTesting <- trainingset[-subsamples, ]
```

A look at the Data

The variable “classe” contains 5 levels: A, B, C, D and E. A plot of the outcome variable will allow us to see the frequency of each levels in the subTraining data set and compare one another.




```
## Confusion Matrix and Statistics
##
##          Reference
## Prediction    A    B    C    D    E
##         A 1395    1    0    0    0
##         B    0  946   11    0    0
##         C    0    2  843    8    0
##         D    0    0    1  796    0
##         E    0    0    0    0  901
##
## Overall Statistics
##
##               Accuracy : 0.9953
##                 95% CI : (0.993, 0.997)
##    No Information Rate : 0.2845
##    P-Value [Acc > NIR] : < 2.2e-16
##
##                  Kappa : 0.9941
##  Mcnemar's Test P-Value : NA
##
## Statistics by Class:
##
##                      Class: A Class: B Class: C Class: D Class: E
## Sensitivity          1.0000   0.9968   0.9860   0.9900   1.0000
## Specificity          0.9997   0.9972   0.9975   0.9998   1.0000
## Pos Pred Value       0.9993   0.9885   0.9883   0.9987   1.0000
## Neg Pred Value       1.0000   0.9992   0.9970   0.9981   1.0000
```

```
## Prevalence           0.2845   0.1935   0.1743   0.1639   0.1837
## Detection Rate       0.2845   0.1929   0.1719   0.1623   0.1837
## Detection Prevalence   0.2847   0.1951   0.1739   0.1625   0.1837
## Balanced Accuracy     0.9999   0.9970   0.9917   0.9949   1.0000
```

Accuracy for Random Forest model was 0.995 (95% CI: (0.993, 0.997)) compared to 0.739 (95% CI: (0.727, 0.752)) for Decision Tree model. The random Forest model is choosen. The accuracy of the model is 0.995. The expected out-of-sample error is estimated at 0.005, or 0.5%. The expected out-of-sample error is calculated as 1 - accuracy for predictions made against the cross-validation set. Our Test data set comprises 20 cases. With an accuracy above 99% on our cross-validation data, we can expect that very few, or none, of the test samples will be missclassified.

Submission

```
##  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20
##  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
```

References

[1] Velloso, E.; Bulling, A.; Gellersen, H.; Ugulino, W.; Fuks, H. Qualitative Activity Recognition of Weight Lifting Exercises. Proceedings of 4th International Conference in Cooperation with SIGCHI (Augmented Human ’13) . Stuttgart, Germany: ACM SIGCHI, 2013.

[2] Krzysztof Gra??bczewski and Norbert Jankowski. Feature Selection with Decision Tree Criterion.