PLM_project

Pedro Rebelo

8 Dec 2014

Introduction

This work is a course project of a coursera *Practical Machine Learning by Jeff Leek, PhD, Roger D. Peng, PhD, Brian Caffo, PhD.* The aim of this work is to demonstrate that is easy to classify the physical activity through the use of data colected by embedded devices on clothing (Wearable data). This Human Activity Recognition - *HAR* - is possible because collecting data increased greatly in recent years, thanks to proliferation of electronic devices that colect data. Furthermore the incrise in computational power and sophisticated algorithms allow the use of these data in creative ways such as algorithms that can learn from data (*Machine learning*). I use the colected data (* http://groupware.les.inf.puc-rio.br/har ; Ugulino, W.; Cardador, D.; Vega, K.; Velloso, E.; Milidiu, R.; Fuks, H. Wearable Computing: Accelerometers' Data Classification of Body Postures and Movements*) to create a statistical model that classifies physical activity.

Get and prepare the data

The folowing code is clear by itself:

```
library(caret)
```

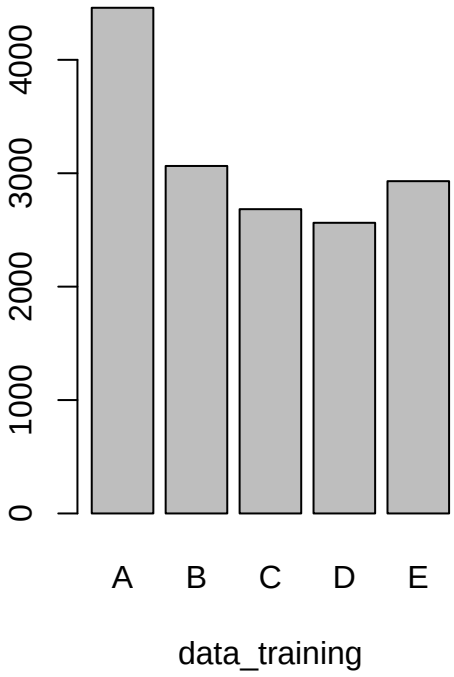
```
## Loading required package: lattice
## Loading required package: ggplot2
```

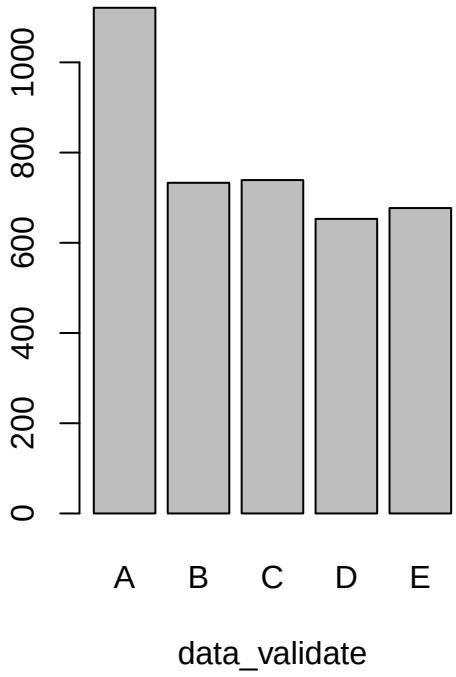
```
library(ggplot2)
########## Get the data, clean it and prepare to use #################
dat_training <- read.csv("/Users/pedrorebelo/Desktop/pml-training.csv",
                         na.strings = c('NA','#DIV/0!',''))
dat_testing <- read.csv("/Users/pedrorebelo/Desktop/pml-testing.csv",
                        na.strings = c('NA','#DIV/0!',''))

pre_util_features <- colnames(dat_training[colSums(is.na(dat_training)) == 0])
#removes coluns with lot of NA
#ncol(pre_util_features)=60, last col. is class (or problem_id in dat_testing)
util_features<-pre_util_features[8:60]#the first 7 are not Wearable data

set.seed(999)#for reproducibility
index_training <- createDataPartition(y=dat_training$user_name, p=0.80)
data_training <- dat_training[index_training$Resample1, util_features]
data_validate <- dat_training[-index_training$Resample1, util_features]
data_testing <- dat_testing[1:20,util_features[-53]]#20 problems, col 53 is problem_id
```

After a quick *EDA*(names of variables, summary of variables -with lots of NAs-) i eliminated the columns containing NA elements and the first 7 columns (because they are not part of werable data) and could make a bias in the model. Then i split the original data in training data (80%) and validate data (20%). The testing data is for quiz answer. We can see in the folowing graph, that the distribution of data_training is similar to the subset data_validation for the variable class.





The model

I chose random forest as method, because its one of the best Machine Learning Algoritms. I want to see the iterations count so i put verboseIter = TRUE, but for generate this document i put it =FALSE. After train the model, i validade the model using model_rf to predict the class of validate data subset.

```
############# Create and train a model ####################
model_rf <- train(classe ~ ., data = data_training, method = 'rf', trControl = trainControl(method = "cv
```

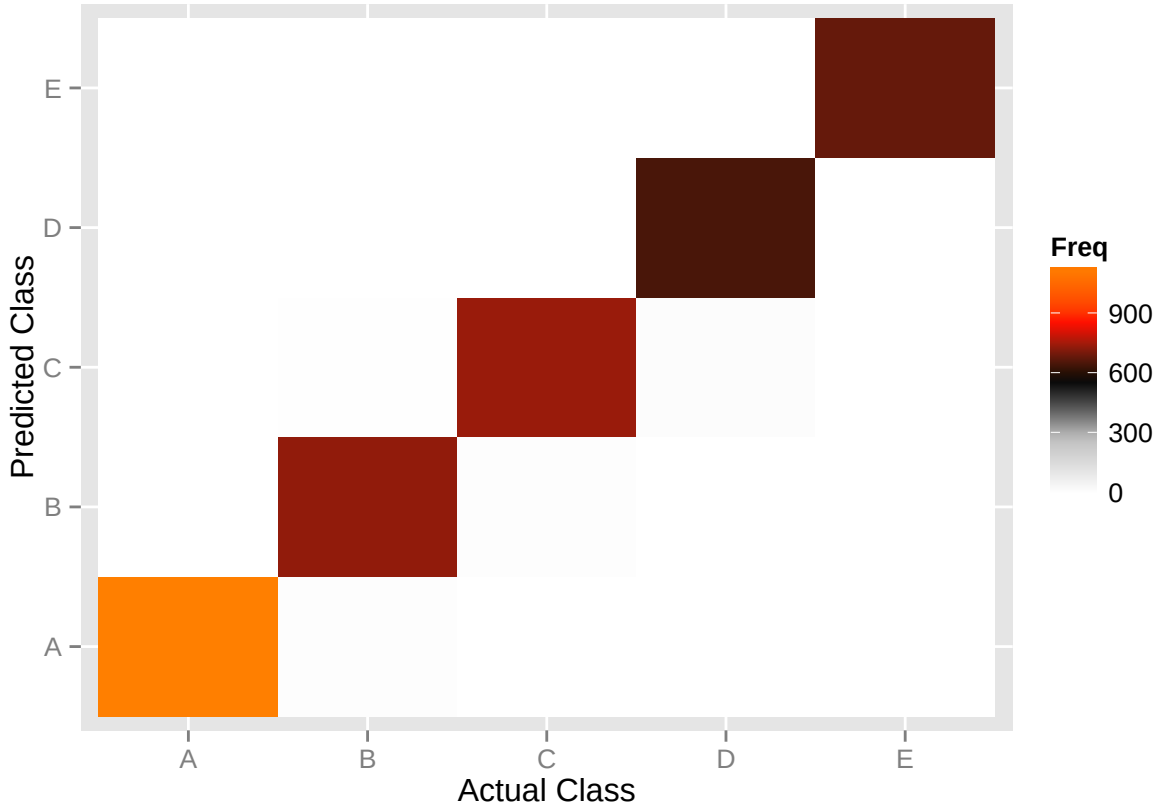
```
## Loading required package: randomForest
## randomForest 4.6-10
## Type rfNews() to see new features/changes/bug fixes.
## Loading required namespace: e1071
```

```
############# Validate the model ####################
pred_rf_validate <- predict(model_rf,data_validate)
cm_rf_validate <- confusionMatrix(pred_rf_validate,data_validate$classe)
cm_rf_validate
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction    A    B    C    D    E
##          A 1121    4    0    0    0
##          B    0  727    4    0    0
##          C    0    2  735    9    0
##          D    0    0    0  643    1
##          E    0    0    0    1  676
##
```

```
## Overall Statistics
##
##                Accuracy : 0.995
##                  95% CI : (0.992, 0.997)
##     No Information Rate : 0.286
##     P-Value [Acc > NIR] : <2e-16
##
##                   Kappa : 0.993
##  Mcnemar's Test P-Value : NA
##
## Statistics by Class:
##
##                      Class: A Class: B Class: C Class: D Class: E
## Sensitivity             1.000    0.992    0.995    0.985    0.999
## Specificity             0.999    0.999    0.997    1.000    1.000
## Pos Pred Value          0.996    0.995    0.985    0.998    0.999
## Neg Pred Value          1.000    0.998    0.999    0.997    1.000
## Prevalence              0.286    0.187    0.188    0.166    0.173
## Detection Rate          0.286    0.185    0.187    0.164    0.172
## Detection Prevalence    0.287    0.186    0.190    0.164    0.173
## Balanced Accuracy       0.999    0.995    0.996    0.992    0.999
```

Looking at the confusion matrix we find that there are few cases of misclassification. As we can see the model have an overall accuracy of 99.5% over the validate data. The following graph show the confusion matrix in a visual way.



We can now test the model for the quiz, with the testing data.

```
############# Answer the quiz  ####################
pred_rf_quiz <- predict(model_rf,data_testing)
pred_rf_quiz
```

```
##  [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
```