Pratical Machine Learning Course Project

THIBAUT SAAH

18 dÃ©cembre 2018

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, our 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://web.archive.org/web/20161224072740/http:/groupware.les.inf.puc-rio.br/har (see the section on the Weight Lifting Exercise Dataset).

Import datasets

Training dataset

```
urlTrain <- "https://d396qusza40orc.cloudfront.net/predmachlearn/pml-training.csv"
if(!file.exists("MachineLearning/pml-training.csv")){
  dir.create("MachineLearning")
  download.file(url = urlTrain, destfile = "./MachineLearning/pml-training.csv")
}
```

```
library(dplyr)
```

```
## Warning: package 'dplyr' was built under R version 3.4.4
```

```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
pmlTrain <- read.csv("./MachineLearning/pml-training.csv")
```

```
dim(pmlTrain)
```

```
## [1] 19622   160
```

Dataset to make prediction

```
urlTest <- "https://d396qusza40orc.cloudfront.net/predmachlearn/pml-testing.csv"
if(!file.exists("MachineLearning/pml-testing.csv")){
  download.file(url = urlTest, destfile = "./MachineLearning/pml-testing.csv")
}
```

```
pmlTest <- read.csv("./MachineLearning/pml-testing.csv")
```

```
dim(pmlTest)
```

```
## [1]  20 160
```

Prepare Data

```
isAnyMissing <- sapply(pmlTest, function (x) any(is.na(x) | x == ""))
isPredictor <- !isAnyMissing & grepl("belt|[^(fore)]arm|dumbbell|forearm", names(isAnyMissing))
predCandidates <- names(isAnyMissing)[isPredictor]
predCandidates
```

```
##  [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"
```

Subset the primary dataset to include only the predictor candidates and the

outcome variable, ‘classe

```
varToInclude <- c(predCandidates, "classe")
pmlTrain <- pmlTrain[, varToInclude]
dim(pmlTrain)
```

```
## [1] 19622    53
```

Split the dataset into a 70% training and 30% probing dataset.

```
library(caret)
```

```
## Warning: package 'caret' was built under R version 3.4.4
```

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

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

```
## Warning: package 'ggplot2' was built under R version 3.4.4
```

```
set.seed(41983)
inTrain <- createDataPartition(pmlTrain$classe, p=0.7, list = FALSE)
```

```
trainSet <- pmlTrain[inTrain,]
testSet <- pmlTrain[-inTrain,]
```

Performing Machine Learning

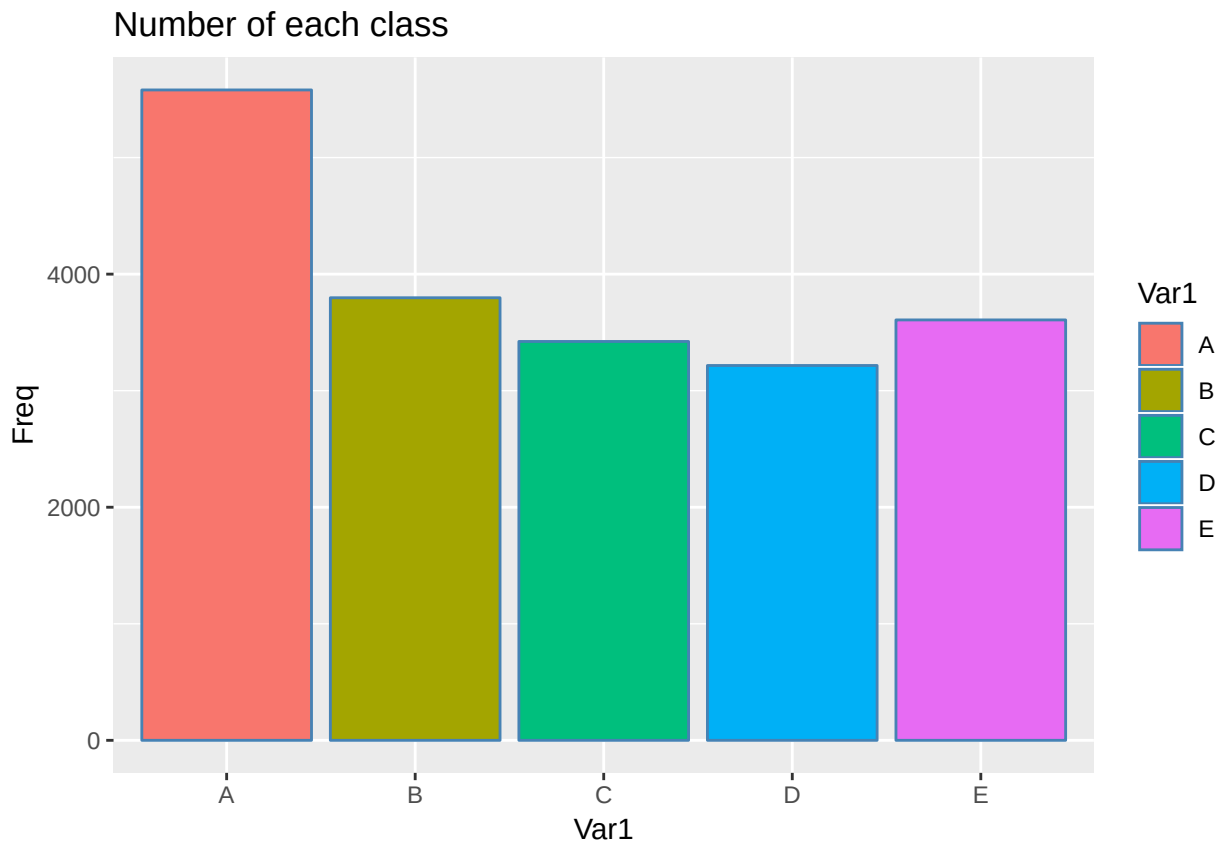
Number of classification 's variables of each type

```
as.data.frame(table(pmlTrain$classe))
```

```
##   Var1 Freq
## 1    A 5580
## 2    B 3797
## 3    C 3422
## 4    D 3216
## 5    E 3607
```

Bar plot

```
p <- ggplot(data = as.data.frame(table(pmlTrain$classe)), aes(Var1, Freq, fill= Var1))+ggtitle("Number
p+geom_bar(stat = "identity", color = "steelblue")
```



Train a prediction model

Using random forest, the out of sample error should be small. The error will be estimated using the 30% pmltrain sample. We would be quite happy with an error estimate of 5% or less.

```
x <- trainSet[,-53]
y <- trainSet[,53]
```

Model on the training set

```
# model fit
library(parallel)
library(doParallel)
```

```
## Warning: package 'doParallel' was built under R version 3.4.4
```

```
## Loading required package: foreach
```

```
## Warning: package 'foreach' was built under R version 3.4.4
```

```
## Loading required package: iterators
```

```
## Warning: package 'iterators' was built under R version 3.4.4
```

```
cluster <- makeCluster(detectCores() - 1) # convention to leave 1 core for OS
registerDoParallel(cluster)
fitControl <- trainControl(method = "cv",
                           number = 2,
                           allowParallel = TRUE)
modFitRandForest <- train(x,y, method="rf",data=trainSet,trControl = fitControl)
stopCluster(cluster)
registerDoSEQ()
```

Confusion Matrix and accuracy

```
# prediction on Test dataset
predictRandForest <- predict(modFitRandForest, newdata=testSet)
confMatRandForest <- confusionMatrix(predictRandForest, testSet$classe)
confMatRandForest
```

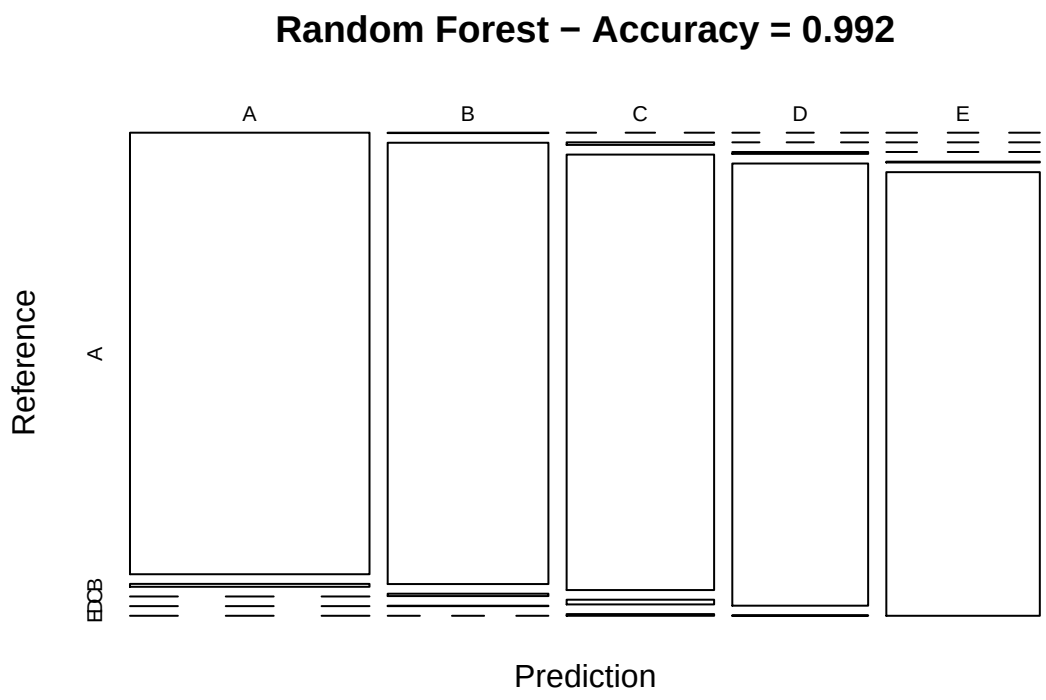
```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction    A    B    C    D    E
##          A 1673   11    0    0    0
##          B    1 1122    6    1    0
##          C    0    6 1016   11    4
##          D    0    0    4  950    1
##          E    0    0    0    2 1077
##
## Overall Statistics
##
##                Accuracy : 0.992
##                  95% CI : (0.9894, 0.9941)
##     No Information Rate : 0.2845
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.9899
##  Mcnemar's Test P-Value : NA
##
## Statistics by Class:
```

```
##
##                      Class: A Class: B Class: C Class: D Class: E
## Sensitivity            0.9994   0.9851   0.9903   0.9855   0.9954
## Specificity            0.9974   0.9983   0.9957   0.9990   0.9996
## Pos Pred Value         0.9935   0.9929   0.9797   0.9948   0.9981
## Neg Pred Value         0.9998   0.9964   0.9979   0.9972   0.9990
## Prevalence             0.2845   0.1935   0.1743   0.1638   0.1839
## Detection Rate         0.2843   0.1907   0.1726   0.1614   0.1830
## Detection Prevalence   0.2862   0.1920   0.1762   0.1623   0.1833
## Balanced Accuracy      0.9984   0.9917   0.9930   0.9922   0.9975
```

Result is good more than 99% of accuracy

plot matrix results

```
# plot matrix results
plot(confMatRandForest$table, col = confMatRandForest$byClass,
     main = paste("Random Forest - Accuracy =",
                  round(confMatRandForest$overall['Accuracy'], 3)))
```



Applying the rf Model to the Test Data

```
predictTEST <- predict(modFitRandForest, newdata=pmlTest[,c(predCandidates)])
predictTEST
```

```
##  [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
```