Practical Machine Learning Project

Marlon Santiago Viñan Ludeña

26/08/2020

Context

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 data for this project come from this source: http://groupware.les.inf.puc-rio.br/har. The training data are available here and the test data here: http://groupware.les.inf.puc-rio.br/har

Six young healthy participants were asked to perform one set of 10 repetitions of the Unilateral Dumbbell Biceps Curl in five different fashions: * Class A - exactly according to the specification * Class B - throwing the elbows to the front * Class C - lifting the dumbbell only halfway * Class D - lowering the dumbbell only halfway * Class E - throwing the hips to the front

Reading files

```
library(caret)
```

```
## Warning: package 'caret' was built under R version 3.5.2
```

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

```
## Warning: package 'lattice' was built under R version 3.5.2
```

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

```
## Warning: package 'ggplot2' was built under R version 3.5.2
```

```
library(rpart)
```

```
## Warning: package 'rpart' was built under R version 3.5.2
```

```
library(rpart.plot)
```

```
## Warning: package 'rpart.plot' was built under R version 3.5.2
```

```
library(RColorBrewer)
library(rattle)
```

```
## Loading required package: tibble
```

```
## Warning: package 'tibble' was built under R version 3.5.2
```

```
## Loading required package: bitops
```

```
## Rattle: A free graphical interface for data science with R.
## Version 5.4.0 Copyright (c) 2006-2020 Togaware Pty Ltd.
## Type 'rattle()' to shake, rattle, and roll your data.
```

```
library(e1071)
```

```
## Warning: package 'e1071' was built under R version 3.5.2
```

```
library(randomForest)
```

```
## randomForest 4.6-14
```

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

```
##
## Attaching package: 'randomForest'
```

```
## The following object is masked from 'package:rattle':
##
##     importance
```

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

```
set.seed(1)

downloadcsv <- function(url, nastrings) {
    temp <- tempfile()
    download.file(url, temp, method = "curl")
    data <- read.csv(temp, na.strings = nastrings)
    unlink(temp)
    return(data)
}

trainurl <- "https://d396qusza40orc.cloudfront.net/predmachlearn/pml-training.csv"
train <- downloadcsv(trainurl, c("", "NA", "#DIV/0!"))

testurl <- "https://d396qusza40orc.cloudfront.net/predmachlearn/pml-testing.csv"
test <- downloadcsv(testurl, c("", "NA", "#DIV/0!"))
```

```
dim(train)
```

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

```
table(train$classe)
```

```
##
##    A    B    C    D    E
## 5580 3797 3422 3216 3607
```

Preprocessing

We separate the training data into training set and validation set

```
set.seed(12)
trainset <- createDataPartition(train$classe, p = 0.8, list = FALSE)
```

```
Training <- train[trainset, ]
Validation <- train[-trainset, ]
```

Feature selection

Its necesary to exclude near zero variance features and exclude missing values.

```
# exclude near zero variance features
nzvcol <- nearZeroVar(Training)
Training <- Training[, -nzvcol]

# exclude columns with m40% ore more missing values exclude descriptive
# columns like name etc
cntlength <- sapply(Training, function(x) {
    sum(!(is.na(x) | x == ""))
})
nullcol <- names(cntlength[cntlength < 0.6 * length(Training$classe)])
descriptcol <- c("X", "user_name", "raw_timestamp_part_1", "raw_timestamp_part_2",
    "cvtd_timestamp", "new_window", "num_window")
excludecols <- c(descriptcol, nullcol)
Training <- Training[, !names(Training) %in% excludecols]
```

Model Train

We will use random forest as our model.

```
library(randomForest)
rfModel <- randomForest(classe ~ ., data = Training, importance = TRUE, ntrees = 10)
```

Model Validation

```
ptraining <- predict(rfModel, Training)
print(confusionMatrix(ptraining, Training$classe))
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction    A    B    C    D    E
##          A 4464    0    0    0    0
##          B    0 3038    0    0    0
##          C    0    0 2738    0    0
##          D    0    0    0 2573    0
##          E    0    0    0    0 2886
##
## Overall Statistics
##
##                Accuracy : 1
##                  95% CI : (0.9998, 1)
##     No Information Rate : 0.2843
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 1
##
##  Mcnemar's Test P-Value : NA
```

```
##
## Statistics by Class:
##
##                      Class: A Class: B Class: C Class: D Class: E
## Sensitivity            1.0000   1.0000   1.0000   1.0000   1.0000
## Specificity            1.0000   1.0000   1.0000   1.0000   1.0000
## Pos Pred Value         1.0000   1.0000   1.0000   1.0000   1.0000
## Neg Pred Value         1.0000   1.0000   1.0000   1.0000   1.0000
## Prevalence             0.2843   0.1935   0.1744   0.1639   0.1838
## Detection Rate         0.2843   0.1935   0.1744   0.1639   0.1838
## Detection Prevalence   0.2843   0.1935   0.1744   0.1639   0.1838
## Balanced Accuracy      1.0000   1.0000   1.0000   1.0000   1.0000
```

Now it is necessary to perform cross-validation against the held out set and see if we avoid the overfitting

Validation set accuracy

We use our Validation set to perform this process

```
pvalidation <- predict(rfModel, Validation)
print(confusionMatrix(pvalidation, Validation$classe))
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction    A    B    C    D    E
##          A 1115    1    0    0    0
##          B    1  758    3    0    0
##          C    0    0  681    5    0
##          D    0    0    0  638    0
##          E    0    0    0    0  721
##
## Overall Statistics
##
##                Accuracy : 0.9975
##                  95% CI : (0.9953, 0.9988)
##     No Information Rate : 0.2845
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.9968
##
##  Mcnemar's Test P-Value : NA
##
## Statistics by Class:
##
##                      Class: A Class: B Class: C Class: D Class: E
## Sensitivity            0.9991   0.9987   0.9956   0.9922   1.0000
## Specificity            0.9996   0.9987   0.9985   1.0000   1.0000
## Pos Pred Value         0.9991   0.9948   0.9927   1.0000   1.0000
## Neg Pred Value         0.9996   0.9997   0.9991   0.9985   1.0000
## Prevalence             0.2845   0.1935   0.1744   0.1639   0.1838
## Detection Rate         0.2842   0.1932   0.1736   0.1626   0.1838
## Detection Prevalence   0.2845   0.1942   0.1749   0.1626   0.1838
## Balanced Accuracy      0.9994   0.9987   0.9970   0.9961   1.0000
```

The cross validation accuracy is 99.75% our model performs very good.

Test set prediction

The prediction of our algorithm for the test set is:

```
ptest <- predict(rfModel, test)
ptest
```

```
##  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
```

Saving files

```
answers <- as.vector(ptest)

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(answers)
```