Practical Machine Learning: Course Project

Prediction Assignment Writeup

Author: Marion Grould

Date: 18 January, 2018

Synopsis

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, the 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: exactly according to the specification (Class A), throwing the elbows to the front (Class B), lifting the dumbbell only halfway (Class C), lowering the dumbbell only halfway (Class D) and throwing the hips to the front (Class E).

Concretely, the aim of this project is to predict the manner in which they did the exercise. This is the *classe* variable in the training set. We may use any of the other variables to predict with.

Sources

The whole data used for this project are available here: http://web.archive.org/web/20161224072740/http:/groupware.les.inf.puc-rio.br/har

The data used to train and test the model are given here: https://d396qusza40orc.cloudfront.net/predmachlearn/pml-training.csv

The testing set used to answer the quiz are given here: https://d396qusza40orc.cloudfront.net/predmachlearn/pml-testing.csv

This work is inspired by the study perfomed by http://web.archive.org/web/20161125212224/http://groupware.les.inf.puc-rio.br:80/work.jsf?p1=10335

Data Loading & Exploratory

We first download and load the data from internet,

```
fileUrl <- "https://d396qusza40orc.cloudfront.net/predmachlearn/pml-training.csv"
download.file(fileUrl, destfile = "pml-training.csv")
data <- read.csv("pml-training.csv", na.strings=c("NA","#DIV/0!",""))
```

and explore a little bit the data:

```
dim(data)
```

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

```
names(data[,1:10])
```

```
##   [1] "X"                      "user_name"            "raw_timestamp_part_1"
##   [4] "raw_timestamp_part_2" "cvtd_timestamp"       "new_window"
##   [7] "num_window"             "roll_belt"            "pitch_belt"
##  [10] "yaw_belt"
```

```
sum(is.na(data))
```

```
## [1] 1925102
```

There are 19622 rows and 160 columns, the first seven columns are not useful for the prediction since they correspond to ID, time-stamp and window data. We also note that there are a huge number of missing values. We thus define a cleaning data set by removing the unnecessary columns and by removing the columns containing more than 95% of missing values:

```
newdata <- data[,-1:-7]
ColRate <- round(apply(newdata, 2, function(x) sum(is.na(x))) / dim(newdata)[1] * 100, 0)
table(ColRate)
```

```
## ColRate
##   0  98 100
##  53  94   6
```

```
newdata <- newdata[,ColRate<95]
dim(newdata)
```

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

From the above table, we note that there are 53 columns without missing values and 100 columns wich contain more than 97% of missing values. Thus, there are no NA in the new data set.

Let us now split it in a training and a testing set, necessary to build the prediction model:

```
library(caret)
```

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

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

```
set.seed(1234)
inTrain <- createDataPartition(y = newdata$classe, p = 0.7, list = FALSE)
training <- newdata[inTrain, ]
testing <- newdata[-inTrain, ]
```

Model Building

Since the data in the training set are already labelled, we use a supervised machine learning method to build our model and predict the classes from the testing test. To build a robust model by using the training set we use cross-validation: the original training set is splitted in a second training and testing set and the model selection is done by using the testing set. More precisely, we use the K-fold method for the cross-validation by using the trainControl() function, and we set K to 3:

```
trainC <- trainControl(method = "cv", number = 3)
```

To build the model we choose the Random Forest algorithm since it is one of the most accurate method:

```
RF <- train(classe ~ ., data = training, method = "rf", trControl = trainC, ntree = 200)
```

```
RF
```

```
## Random Forest
##
```

```
## 13737 samples
##    52 predictor
##     5 classes: 'A', 'B', 'C', 'D', 'E'
##
## No pre-processing
## Resampling: Cross-Validated (3 fold)
## Summary of sample sizes: 9157, 9157, 9160
## Resampling results across tuning parameters:
##
##   mtry  Accuracy   Kappa
##    2    0.9886437  0.9856322
##   27    0.9892266  0.9863707
##   52    0.9836952  0.9793672
##
## Accuracy was used to select the optimal model using  the largest value.
## The final value used for the model was mtry = 27.
```

Model Testing

We now test the accuracy of the model obtained by using both the cross-validation and the Random Forest methods, on the testing set:

```
predRF <- predict(RF, testing)
confusionMatrix(testing$classe, predRF)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction    A    B    C    D    E
##          A 1674    0    0    0    0
##          B   11 1127    1    0    0
##          C    0    3 1019    4    0
##          D    0    1    5  957    1
##          E    0    1    2    4 1075
##
## Overall Statistics
##
##                Accuracy : 0.9944
##                  95% CI : (0.9921, 0.9961)
##     No Information Rate : 0.2863
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.9929
##  Mcnemar's Test P-Value : NA
##
## Statistics by Class:
##
##                      Class: A Class: B Class: C Class: D Class: E
## Sensitivity            0.9935   0.9956   0.9922   0.9917   0.9991
## Specificity            1.0000   0.9975   0.9986   0.9986   0.9985
## Pos Pred Value         1.0000   0.9895   0.9932   0.9927   0.9935
## Neg Pred Value         0.9974   0.9989   0.9984   0.9984   0.9998
## Prevalence             0.2863   0.1924   0.1745   0.1640   0.1828
## Detection Rate         0.2845   0.1915   0.1732   0.1626   0.1827
## Detection Prevalence   0.2845   0.1935   0.1743   0.1638   0.1839
## Balanced Accuracy      0.9967   0.9965   0.9954   0.9951   0.9988
```

As showed, the accuracy of the model is very satisfactory since it reaches 99% and the out-of-sample error reaches 0.56% (1 minus the accuracy).

Predictions (Quiz 4)

Since the model is very accurate, we can perform predictions on the testing test provided from internet. Let us load it and do some cleaning:

```
fileUrl <- "https://d396qusza40orc.cloudfront.net/predmachlearn/pml-testing.csv"
download.file(fileUrl, destfile = "pml-testing.csv")
data2 <- read.csv("pml-testing.csv", na.strings=c("NA","#DIV/0!",""))
newtesting <- data2[,-1:-7]
ColRate <- round(apply(newtesting, 2, function(x) sum(is.na(x))) / dim(newtesting)[1] * 100, 0)
newtesting <- newtesting[,ColRate<95]
```

Let us now perform the predictions of each *problem_id* (1 to 20) of the new testing set:

```
n <- length(newtesting$problem_id)
for (i in 1:n){
    IndRow <- newtesting$problem_id == i
    predRF <- predict(RF, newtesting[IndRow,-53])
    print(paste0("The predicted classe for the case ", i, " is: ", as.character(predRF)))
}
```

```
## [1] "The predicted classe for the case 1 is: B"
## [1] "The predicted classe for the case 2 is: A"
## [1] "The predicted classe for the case 3 is: B"
## [1] "The predicted classe for the case 4 is: A"
## [1] "The predicted classe for the case 5 is: A"
## [1] "The predicted classe for the case 6 is: E"
## [1] "The predicted classe for the case 7 is: D"
## [1] "The predicted classe for the case 8 is: B"
## [1] "The predicted classe for the case 9 is: A"
## [1] "The predicted classe for the case 10 is: A"
## [1] "The predicted classe for the case 11 is: B"
## [1] "The predicted classe for the case 12 is: C"
## [1] "The predicted classe for the case 13 is: B"
## [1] "The predicted classe for the case 14 is: A"
## [1] "The predicted classe for the case 15 is: E"
## [1] "The predicted classe for the case 16 is: E"
## [1] "The predicted classe for the case 17 is: A"
## [1] "The predicted classe for the case 18 is: B"
## [1] "The predicted classe for the case 19 is: B"
## [1] "The predicted classe for the case 20 is: B"
```