Practical Machine Learning

Final Project

Tatiana Zolhof

September, 2018

About

This is the Final Project for the Practical Machine Learning Coursera's Course.

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.

The data for this project come from this source: http://groupware.les.inf.puc-rio.br/har. Six participants were asked to perform barbell lifts correctly and incorrectly in 5 different ways. The goal of this Final Project is to predict the manner in which they did the exercise, using data from accelerometers on their belt, forearm, arm, and dumbell.

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

Exploring Data

The original files were read as follows:

```
original_training <- read.csv("pml-training.csv")
original_testing <- read.csv("pml-testing.csv")
```

The dimensions of both data sets were obtained with the dim command:
training – 19,622 rows x 160 columns
testing – 20 rows x 160 columns

Then, using the summary command, it was possible to see that many columns had blank or NA values, as shown in the extract below:

```
kurtosis_roll_dumbbell kurtosis_picth_dumbbell kurtosis_yaw_dumbbell
            :19216                 :19216                 :19216
   #DIV/0!:    5          -0.5464:    2          #DIV/0!:  406
   -0.2583:    2          -0.9334:    2
   -0.3705:    2          -2.0833:    2
   -0.5855:    2          -2.0851:    2
   -2.0851:    2          -2.0889:    2
```

Because of that, the training data were read again, but now ignoring all columns with more than 90% of missing values:

```
training_to_reduce <- read.csv("pml-training.csv", na.strings = c("", NA))
cols_is_na <- ((colSums(is.na(training_to_reduce)) >
0.1*nrow(training_to_reduce)) == TRUE)
cols_original <- names(training_to_reduce)
cols_to_delete <- cols_original[cols_is_na]
training_final <- training_to_reduce[, -which(names(training_to_reduce) %in%
cols_to_delete)]
```

With that, the number of columns was reduced from 160 to 60. This allowed a closer exploration of the remaining columns, and the realization that the first 7 columns were also not relevant to the analysis since they contained information about indexes, names of the participants, time stamps of the measurements and a window variable that also had more than 90% of repeated values. Hence, those columns were deleted from the data set:

```
new_cols_to_delete <- names(training_to_reduce[1:7])
training_final <- training_final[, -which(names(training_final) %in%
new_cols_to_delete)]
```

With that, the number of columns in the training data set was reduced to 53.

The same steps needed to be applied to the original testing set. The last column was also excluded since it only had indexes related to the Coursera’s quiz:

testing <- original_testing[, -which(names(original_testing) %in% cols_to_delete)]
testing <- testing[, -which(names(testing) %in% new_cols_to_delete)]
testing <- testing[-53]

Creating Training and Validation Sets

With the data ready, the training and validation sets were created with the use of the caret package, and the proportion of 75% - 25%:

library(caret)
inTrain <- createDataPartition(y=training_final$classe, p=0.75, list = FALSE)
training <- training_final[inTrain,]
validation <- training_final[-inTrain,]

The final dimensions were:
training – 14,718 rows x 53 columns
validation – 4,904 rows x 53 columns
testing - 20 rows x 52 columns

Trying Different Models

1) rpart - Recursive Partitioning And Regression Trees

set.seed(334)
model1 <- train(classe ~ ., data=training, method="rpart")
pr_model1 <- predict(model1, validation)
confusionMatrix(pr_model1, validation$classe)

```
Confusion Matrix and Statistics

          Reference
Prediction    A    B    C    D    E
         A 1268  409  401  372  127
         B   18  311   27  133  119
         C  106  229  427  299  242
         D    0    0    0    0    0
         E    3    0    0    0  413

Overall Statistics

               Accuracy : 0.4933
                 95% CI : (0.4792, 0.5074)
    No Information Rate : 0.2845
```

```
    P-Value [Acc > NIR] : < 2.2e-16
```

By predicting the model on the validation data, and then plotting the corresponding confusion matrix, it is possible to see that the accuracy was approximately 0.50, which shows that this model was not a good choice to fit the data.

2) lda - Linear Discriminant Analysis:

```
model2 <- train(classe ~ ., data=training, method="lda")
pr_model2 <- predict(model2, validation)
confusionMatrix(pr_model2, validation$classe)
```

```
Confusion Matrix and Statistics

                Reference
Prediction    A    B    C    D    E
         A 1139  169   87   41   32
         B   32  592   84   45  141
         C  106  108  549   95   98
         D  113   28  113  598   84
         E    5   52   22   25  546

Overall Statistics

               Accuracy : 0.6982
                 95% CI : (0.6851, 0.711)
    No Information Rate : 0.2845
    P-Value [Acc > NIR] : < 2.2e-16
```

Here, according to the confusion matrix, the accuracy was approximately 0.70.

3) rf - Random Forest:

```
trControl <- trainControl(number=5, method="cv")
model3 <- train(classe ~ ., data=training, method="rf", trControl=trControl)
pr_model3 <- predict(model3, validation)
confusionMatrix(pr_model3, validation$classe)
```

```
Confusion Matrix and Statistics

          Reference
Prediction    A    B    C    D    E
         A 1393    9    0    0    0
         B    2  936    7    0    0
         C    0    4  848   12    0
         D    0    0    0  792    1
         E    0    0    0    0  900

Overall Statistics

               Accuracy : 0.9929
                 95% CI : (0.9901, 0.995)
    No Information Rate : 0.2845
```

```
    P-Value [Acc > NIR] : < 2.2e-16
```

In this third model, the accuracy on the validation data was approximately 0.99, the very best observed so far.

For this reason, this was the model selected to predict the Classe variable on the testing set. Using the varImp command, it was possible to see the variable importance rank in the random forest model:

varImp(model3)

```
rf variable importance

only 20 most important variables shown (out of 52)

                     Overall
roll_belt             100.00
yaw_belt               86.41
magnet_dumbbell_z      75.97
magnet_dumbbell_y      67.50
pitch_forearm          66.92
pitch_belt             65.62
magnet_dumbbell_x      56.05
roll_forearm           50.84
roll_dumbbell          48.72
accel_dumbbell_y       46.28
accel_belt_z           45.98
magnet_belt_y          44.90
magnet_belt_z          44.39
accel_dumbbell_z       41.42
roll_arm               37.98
accel_forearm_x        36.22
gyros_belt_z           32.34
accel_dumbbell_x       30.92
total_accel_dumbbell   29.83
yaw_dumbbell           29.66
```

Predictions for the testing set

Those were the predictions obtained to the testing set, using the chosen model:

predict(model3, testing)

```
 [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
```