Practical Machine Learning - Course Project

Larisa Terenteva

16 04 2021

Created with knitr

Overview & Background

The goal of course project is to predict the manner in which people performed the exercise using data from accelerometers on the belt, forearm, arm, and dumbell of 6 participants. This is the “classe” variable in the training set. The machine learning algorithm is applied to the 20 test cases available in the test data and the predictions are submitted to the Course Project Prediction Quiz for automated grading.

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 (http://groupware.les.inf.puc-rio.br/har) (see the section on the Weight Lifting Exercise Dataset).

Read more: http://groupware.les.inf.puc-rio.br/har#ixzz3xsbS5bVX (http://groupware.les.inf.puc-rio.br /har#ixzz3xsbS5bVX)

Data Loading and Exploratory Analysis

The training data for this project are available here:

https://d396qusza40orc.cloudfront.net/predmachlearn/pml-training.csv (https://d396qusza40orc.cloudfront.net/predmachlearn/pml-training.csv)

The test data are available here:

https://d396qusza40orc.cloudfront.net/predmachlearn/pml-testing.csv (https://d396qusza40orc.cloudfront.net/predmachlearn/pml-testing.csv)

The data for this project come from http://groupware.les.inf.puc-rio.br/har (http://groupware.les.inf.puc-rio.br /har).

Velloso, E.; Bulling, A.; Gellersen, H.; Ugulino, W.; Fuks, H. “Qualitative Activity Recognition of Weight Lifting Exercises. Velloso, E.; Bulling, A.; Gellersen, H.; Ugulino, W.; Fuks, H. “Qualitative Activity Recognition of Weight Lifting Exercises. Proceedings of 4th International Conference in Cooperation with SIGCHI (Augmented Human ’13)”. Stuttgart, Germany: ACM SIGCHI, 2013.

We first upload the R libraries that are necessary for the complete analysis.

```
library(caret)
```

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

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

```
library(randomForest)
```

```
## Warning: package 'randomForest' was built under R version 4.0.5
```

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

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

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

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

```
library(corrplot)
```

```
## Warning: package 'corrplot' was built under R version 4.0.5
```

```
## corrplot 0.84 loaded
```

Data Loading and Cleaning

```
testdata <- read.csv("pml-testing.csv")
traindata <- read.csv("pml-training.csv")
# create a partition with the training dataset
inTrain  <- createDataPartition(traindata$classe, p=0.7, list=FALSE)
trainset <- traindata[inTrain, ]
testset  <- traindata[-inTrain, ]
dim(trainset)
```

```
## [1] 13737   160
```

```
dim(testset)
```

```
## [1] 5885  160
```

```
# remove variables that are mostly NA
AllNA  <- sapply(trainset, function(x) mean(is.na(x))) > 0.95
trainset <- trainset[, AllNA==FALSE]
testset  <- testset[, AllNA==FALSE]
dim(trainset)
```

```
## [1] 13737    93
```

```
dim(testset)
```

```
## [1] 5885   93
```

```
# remove variables with Nearly Zero Variance
NZV <- nearZeroVar(trainset)
trainset <- trainset[, -NZV]
testset  <- testset[, -NZV]
dim(trainset)
```

```
## [1] 13737    59
```

```
dim(testset)
```

```
## [1] 5885   59
```

```
# remove identification only variables (columns 1 to 5)
trainset <- trainset[, -(1:5)]
testset  <- testset[, -(1:5)]
dim(trainset)
```

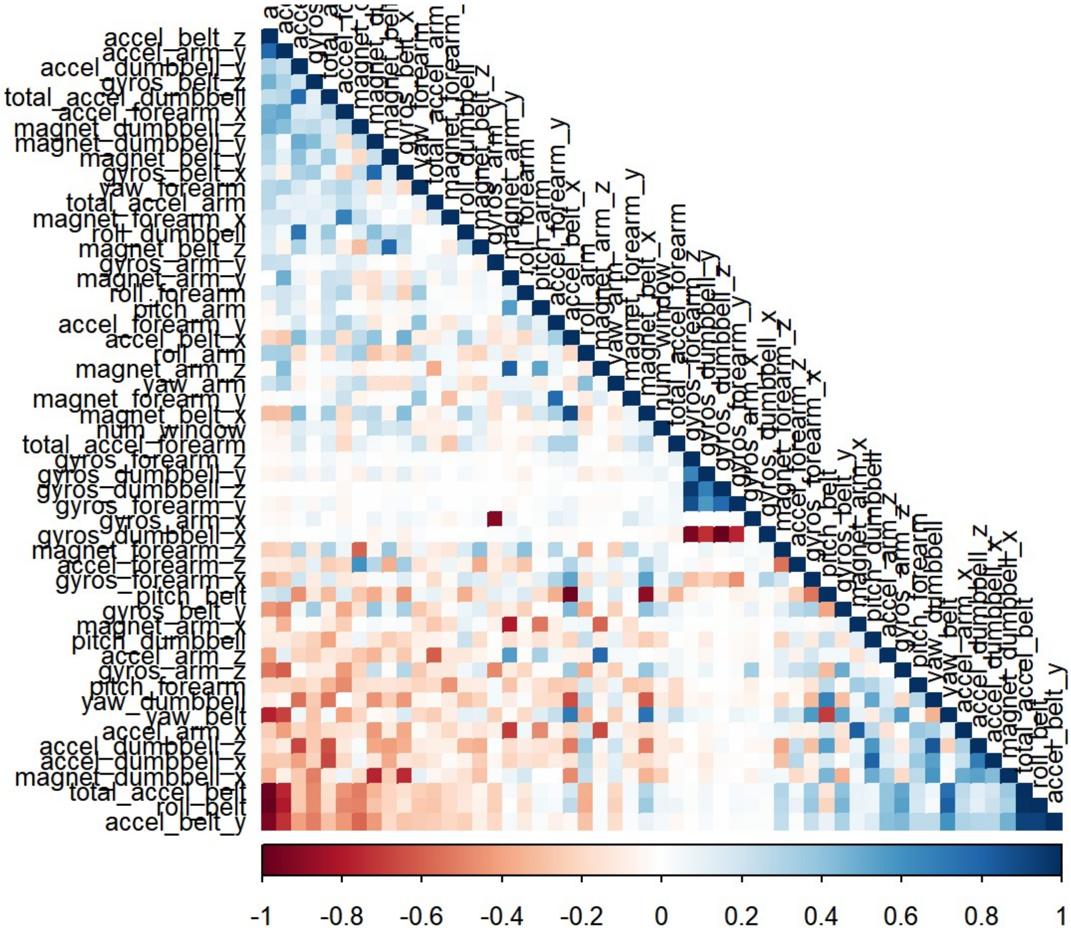
```
## [1] 13737    54
```

```
dim(testset)
```

```
## [1] 5885   54
```

Correlation Analysis

```
corMatrix <- cor(trainset[, -54])
corrplot(corMatrix, order = "FPC", method = "color", type = "lower", tl.cex = 0.8,
tl.col = rgb(0, 0, 0))
```




```
#Get the column indices of the features that are strongly correlated with one anoth
er
highlyCorrelated <- findCorrelation(corMatrix, cutoff=0.75)

#New data set without variables that had the largest mean absolute correlation
trainset <- trainset[,-c(highlyCorrelated)]
testset<- testset[,-c(highlyCorrelated)]
dim(trainset)
```

```
## [1] 13737    33
```

```
dim(testset)
```

```
## [1] 5885   33
```

Prediction Model Building. Method: Random Forest

```r
# model fit
set.seed(12345)

#Use 3 fold cross validation
control <- trainControl(method = "cv", number = 3)

#Create predictive model by random forests method
fitmod <- train(classe~., data = trainset, method = "rf", trControl = control)

fitmod$finalModel
```

```
##
## Call:
##  randomForest(x = x, y = y, mtry = param$mtry)
##                Type of random forest: classification
##                      Number of trees: 500
## No. of variables tried at each split: 17
##
##         OOB estimate of  error rate: 0.2%
## Confusion matrix:
##      A    B    C    D    E  class.error
## A 3903    2    0    0    1 0.0007680492
## B    7 2647    4    0    0 0.0041384500
## C    0    3 2392    1    0 0.0016694491
## D    0    1    4 2246    1 0.0026642984
## E    0    1    0    3 2521 0.0015841584
```

```r
#Use model to predict "classe" variable on the testing partition
pred <- predict(fitmod, newdata=testset)

#Print confusion matrix to see results of predictions
confusionMatrix(pred, factor(testset$classe))
```

```
## Confusion Matrix and Statistics
## 
##           Reference
## Prediction    A    B    C    D    E
##          A 1674    2    0    0    0
##          B    0 1136    2    1    0
##          C    0    0 1021    0    0
##          D    0    0    3  962    0
##          E    0    1    0    1 1082
## 
## Overall Statistics
##                                           
##                Accuracy : 0.9983          
##                  95% CI : (0.9969, 0.9992)
##     No Information Rate : 0.2845          
##     P-Value [Acc > NIR] : < 2.2e-16       
##                                           
##                   Kappa : 0.9979          
##                                           
##  Mcnemar's Test P-Value : NA              
## 
## Statistics by Class:
## 
##                      Class: A Class: B Class: C Class: D Class: E
## Sensitivity            1.0000   0.9974   0.9951   0.9979   1.0000
## Specificity            0.9995   0.9994   1.0000   0.9994   0.9996
## Pos Pred Value         0.9988   0.9974   1.0000   0.9969   0.9982
## Neg Pred Value         1.0000   0.9994   0.9990   0.9996   1.0000
## Prevalence             0.2845   0.1935   0.1743   0.1638   0.1839
## Detection Rate         0.2845   0.1930   0.1735   0.1635   0.1839
## Detection Prevalence   0.2848   0.1935   0.1735   0.1640   0.1842
## Balanced Accuracy      0.9998   0.9984   0.9976   0.9987   0.9998
```

Prediction on Test dataset

The accuracy of the Random Forest model is 0.9983. This model was applied to predict the 20 quiz results.

```
predictTEST <- predict(fitmod, newdata=testdata)
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
```