Coursera-Pratical-Machine_learning-Assginment

Peng Lu

30/07/2019

Introduction

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 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

The data for this project come from this source: http://groupware.les.inf.puc-rio.br/har. If you use the document you create for this class for any purpose please cite them as they have been very generous in allowing their data to be used for this kind of assignment.

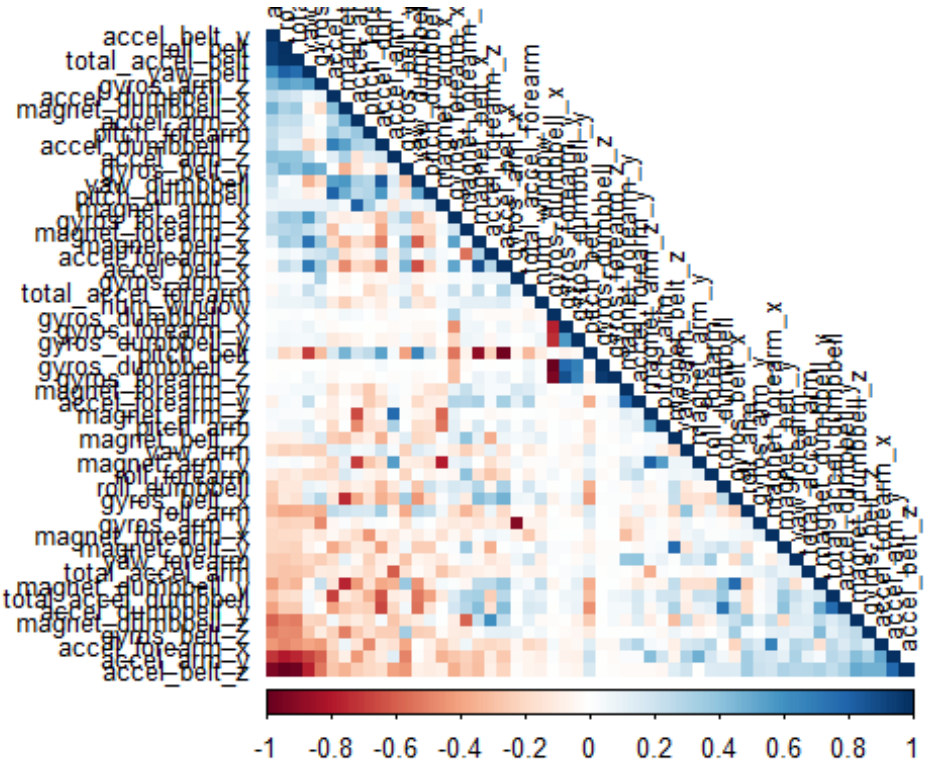
```
##Read training and test datasets into R environment
pml_raw <- read.csv("https://d396qusza40orc.cloudfront.net/predmachlearn/pml-
training.csv", header = TRUE)
pml_validation <-
read.csv("https://d396qusza40orc.cloudfront.net/predmachlearn/pml-
testing.csv", header = TRUE)

##Check both Training and testing datasets
dim(pml_raw);dim(pml_validation)

## [1] 19622   160

## [1]  20 160
```

Preprocessing data

```
#Load package for this process
if (!require(caret)) install.packages("caret")

## Loading required package: caret

## Warning: package 'caret' was built under R version 3.5.3

## Loading required package: lattice

## Loading required package: ggplot2

## Warning: package 'ggplot2' was built under R version 3.5.3

library(caret)
if (!require(e1071)) install.packages("e1071")

## Loading required package: e1071

## Warning: package 'e1071' was built under R version 3.5.3

library(e1071)
##split raw data into training and testing datasets
set.seed(123)
split_label <- createDataPartition(y=pml_raw$classe,p=0.7,list = FALSE)

train<-pml_raw[split_label,]
test<-pml_raw[-split_label,]

dim(train); dim(test)

## [1] 13737   160

## [1] 5885  160

## remove unique data columns
NZV <- nearZeroVar(train, freqCut = 19)
train <- train[,-NZV]
test<- test[,-NZV]

##remove columns with mostly null values
nul_label <- sapply(train, function(x) mean(is.na(x))) >0.95
train<- train[,nul_label==FALSE]
test<- test[,nul_label==FALSE]

## remove identification variables
train<-train[,-c(1:5)]
test<-test[,-c(1:5)]
```

Data Exploratory Analysis

```
#Load package for this process
if (!require(corrplot)) install.packages("corrplot")

## Loading required package: corrplot

## Warning: package 'corrplot' was built under R version 3.5.3

## corrplot 0.84 loaded

library(corrplot)
##Check some of columns in the dataset
summary(pml_raw$classe)

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

##Show correlation between all variables
corrplot(cor(train[,-54]), method = "color", type = "lower", order = "FPC",
tl.cex=0.8, tl.col = rgb(0,0,0))
```



##Model Selection Process

###Decision Tree

```
# Load package for this process
if (!require(rpart)) install.packages("rpart")
```

```
## Loading required package: rpart
```

```
if (!require(rpart.plot)) install.packages("rpart.plot")
```

```
## Loading required package: rpart.plot
```

```
## Warning: package 'rpart.plot' was built under R version 3.5.3
```

```
library(rpart.plot);library(rpart)
if (!require(rattle)) install.packages("rattle")
```

```
## Loading required package: rattle
```

```
## Warning: package 'rattle' was built under R version 3.5.3
```

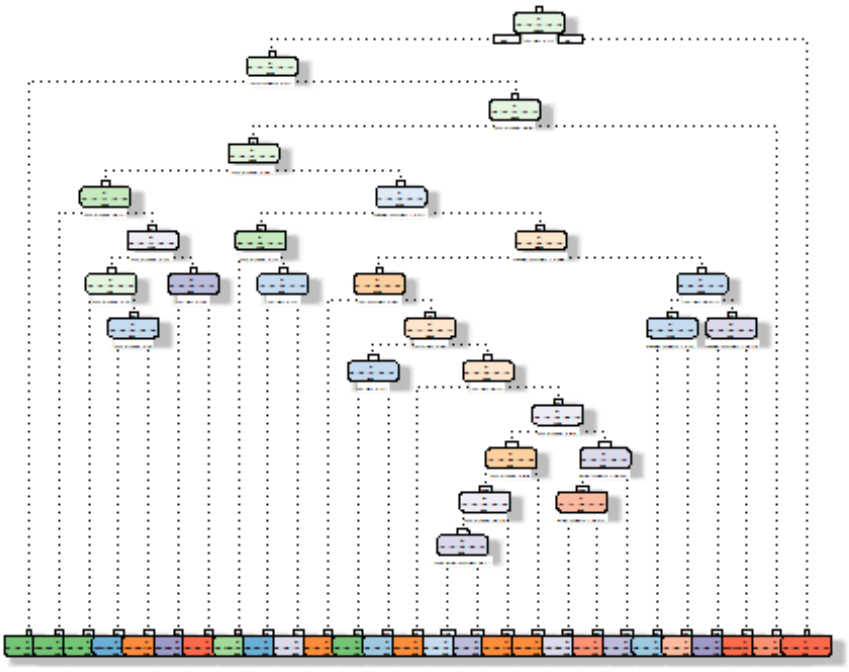
```
## Rattle: A free graphical interface for data science with R.
## Version 5.2.0 Copyright (c) 2006-2018 Togaware Pty Ltd.
## Type 'rattle()' to shake, rattle, and roll your data.
```

```
library(rattle)
```

```
## Modeling process
dt_model <- rpart(classe ~ ., data=train, method = 'class')
```

```
fancyRpartPlot(dt_model)
```

```
## Warning: labs do not fit even at cex 0.15, there may be some overplotting
```



Rattle 2019-Aug-02 20:05:36 Will

```
dt_test_predict <- predict(dt_model, test, type = "class")
conf_matrix_dt_test <- confusionMatrix(dt_test_predict, test$classe)
conf_matrix_dt_test
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction    A    B    C    D    E
##          A 1496  114    3   18    7
##          B   74  843   55   71   44
##          C    0   57  830   37    3
##          D   84   51  124  778   68
##          E   20   74   14   60  960
##
## Overall Statistics
##
##                Accuracy : 0.8338
##                  95% CI : (0.8241, 0.8432)
##     No Information Rate : 0.2845
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.7901
##
##  Mcnemar's Test P-Value : < 2.2e-16
##
## Statistics by Class:
##
##                      Class: A Class: B Class: C Class: D Class: E
## Sensitivity            0.8937   0.7401   0.8090   0.8071   0.8872
## Specificity            0.9663   0.9486   0.9800   0.9336   0.9650
## Pos Pred Value         0.9133   0.7755   0.8954   0.7041   0.8511
## Neg Pred Value         0.9581   0.9383   0.9605   0.9611   0.9744
## Prevalence             0.2845   0.1935   0.1743   0.1638   0.1839
## Detection Rate         0.2542   0.1432   0.1410   0.1322   0.1631
## Detection Prevalence   0.2783   0.1847   0.1575   0.1878   0.1917
## Balanced Accuracy      0.9300   0.8444   0.8945   0.8703   0.9261
```

###Random Forest

```
if (!require(randomForest)) install.packages("randomForest")
```

```
## Loading required package: randomForest
```

```
## Warning: package 'randomForest' was built under R version 3.5.3
```

```
## 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
```

```
library(randomForest)

set.seed(123)
rf_model <- randomForest(classe ~ ., data=train, importance=TRUE)

rf_test_predict <- predict(rf_model, test, type="class")
conf_matrix_rf_test <- confusionMatrix(rf_test_predict, test$classe)
conf_matrix_rf_test
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction    A    B    C    D    E
##          A 1674    0    0    0    0
##          B    0 1139    7    0    0
##          C    0    0 1019    6    0
##          D    0    0    0  957    0
##          E    0    0    0    1 1082
##
## Overall Statistics
##
##                Accuracy : 0.9976
##                  95% CI : (0.996, 0.9987)
##     No Information Rate : 0.2845
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.997
##
##  Mcnemar's Test P-Value : NA
##
## Statistics by Class:
##
##                      Class: A Class: B Class: C Class: D Class: E
## Sensitivity            1.0000   1.0000   0.9932   0.9927   1.0000
## Specificity            1.0000   0.9985   0.9988   1.0000   0.9998
## Pos Pred Value         1.0000   0.9939   0.9941   1.0000   0.9991
## Neg Pred Value         1.0000   1.0000   0.9986   0.9986   1.0000
## Prevalence             0.2845   0.1935   0.1743   0.1638   0.1839
## Detection Rate         0.2845   0.1935   0.1732   0.1626   0.1839
## Detection Prevalence   0.2845   0.1947   0.1742   0.1626   0.1840
## Balanced Accuracy      1.0000   0.9993   0.9960   0.9964   0.9999
```

###Data Validation The random forest model has been decided using to predict the test data due to the over 99% accuracy and Kappa rate.

```
pml_validation <- pml_validation[,-NZV]
pml_validation<- pml_validation[,nul_label==FALSE]
pml_validation<-pml_validation[,-c(1:5)]

predict(rf_model, pml_validation, type="class")
```

```
##  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
```