coursera 8

Muskaan

21/11/2020

TITLE: Prediction Assignment by Muskaan Parmar

SUMMARY: The goal of this assignment is to use the data of 6 participants obtained from accelerometers on the belt, forearm, arm, and dumbell and predict the manner in which they exercise. They were asked to perform barbell lifts correctly and incorrectly in 5 different ways. Since, people regularly quantify how much of a particular activity they do, but they rarely quantify how well they do it which would be dealt in this assignment. We will build a model, use cross validation and thus make choices. Also, we will use the model on 20 different test cases.

1. Setting the directory and loading the data

```
setwd("~/R/Coursera 8")
library(randomForest)
```

```
## Warning: package 'randomForest' was built under R version 4.0.3
```

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

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

```
library(lattice)
library(ggplot2)
```

```
##
## Attaching package: 'ggplot2'
```

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

```
library(doParallel)
```

```
## Warning: package 'doParallel' was built under R version 4.0.3
```

```
## Loading required package: foreach
```

```
## Warning: package 'foreach' was built under R version 4.0.3
```

```
## Loading required package: iterators
```

```
## Warning: package 'iterators' was built under R version 4.0.3
```

```
## Loading required package: parallel
```

```
library(rpart)
```

```
## Warning: package 'rpart' was built under R version 4.0.3
```

```
library(caret)
```

```
## Warning: package 'caret' was built under R version 4.0.3
```

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

```
## Warning: package 'rpart.plot' was built under R version 4.0.3
```

```
set.seed(127)
train <- read.csv("pml-training.csv", na.strings=c("NA","","#DIV/0!"))
test <- read.csv("pml-testing.csv", na.strings=c("NA", "", "#DIV/0!"))
#loading data and removing NA,#DIV/0! and blank values from data
#head(train)
print("After removing NA,#DIV/0! and blank values from the two datasets:")
```

```
## [1] "After removing NA,#DIV/0! and blank values from the two datasets:"
```

```
print("Dimension of training data=")
```

```
## [1] "Dimension of training data="
```

```
dim(train)
```

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

```
print("Dimension of testing data=")
```

```
## [1] "Dimension of testing data="
```

```
dim(test)
```

```
## [1]  20 160
```

2.Data pre-processing

```r
train2<-train[,-c(1:7)]
test2 <-test[,-c(1:7)]
#removing index, timestamp, new window, num window and subject name i.e. first 7 columns
print("After removing non predictors from the two datasets:")
```

```
## [1] "After removing non predictors from the two datasets:"
```

```r
print("Dimension of training data=")
```

```
## [1] "Dimension of training data="
```

```r
dim(train2)
```

```
## [1] 19622   153
```

```r
print("Dimension of testing data=")
```

```
## [1] "Dimension of testing data="
```

```r
dim(test2)
```

```
## [1]  20 153
```

```r
#checking for non zero values in training dataset
train3<-nzv(train2[,-ncol(train2)],saveMetrics=TRUE)
row(train3)
```

```
##        [,1] [,2] [,3] [,4]
##   [1,]    1    1    1    1
##   [2,]    2    2    2    2
##   [3,]    3    3    3    3
##   [4,]    4    4    4    4
##   [5,]    5    5    5    5
##   [6,]    6    6    6    6
##   [7,]    7    7    7    7
##   [8,]    8    8    8    8
##   [9,]    9    9    9    9
##  [10,]   10   10   10   10
##  [11,]   11   11   11   11
##  [12,]   12   12   12   12
##  [13,]   13   13   13   13
##  [14,]   14   14   14   14
##  [15,]   15   15   15   15
##  [16,]   16   16   16   16
##  [17,]   17   17   17   17
##  [18,]   18   18   18   18
##  [19,]   19   19   19   19
##  [20,]   20   20   20   20
##  [21,]   21   21   21   21
```

```
##  [22,]   22   22   22   22
##  [23,]   23   23   23   23
##  [24,]   24   24   24   24
##  [25,]   25   25   25   25
##  [26,]   26   26   26   26
##  [27,]   27   27   27   27
##  [28,]   28   28   28   28
##  [29,]   29   29   29   29
##  [30,]   30   30   30   30
##  [31,]   31   31   31   31
##  [32,]   32   32   32   32
##  [33,]   33   33   33   33
##  [34,]   34   34   34   34
##  [35,]   35   35   35   35
##  [36,]   36   36   36   36
##  [37,]   37   37   37   37
##  [38,]   38   38   38   38
##  [39,]   39   39   39   39
##  [40,]   40   40   40   40
##  [41,]   41   41   41   41
##  [42,]   42   42   42   42
##  [43,]   43   43   43   43
##  [44,]   44   44   44   44
##  [45,]   45   45   45   45
##  [46,]   46   46   46   46
##  [47,]   47   47   47   47
##  [48,]   48   48   48   48
##  [49,]   49   49   49   49
##  [50,]   50   50   50   50
##  [51,]   51   51   51   51
##  [52,]   52   52   52   52
##  [53,]   53   53   53   53
##  [54,]   54   54   54   54
##  [55,]   55   55   55   55
##  [56,]   56   56   56   56
##  [57,]   57   57   57   57
##  [58,]   58   58   58   58
##  [59,]   59   59   59   59
##  [60,]   60   60   60   60
##  [61,]   61   61   61   61
##  [62,]   62   62   62   62
##  [63,]   63   63   63   63
##  [64,]   64   64   64   64
##  [65,]   65   65   65   65
##  [66,]   66   66   66   66
##  [67,]   67   67   67   67
##  [68,]   68   68   68   68
##  [69,]   69   69   69   69
##  [70,]   70   70   70   70
##  [71,]   71   71   71   71
##  [72,]   72   72   72   72
##  [73,]   73   73   73   73
##  [74,]   74   74   74   74
##  [75,]   75   75   75   75
```

```
##  [76,]   76   76   76   76
##  [77,]   77   77   77   77
##  [78,]   78   78   78   78
##  [79,]   79   79   79   79
##  [80,]   80   80   80   80
##  [81,]   81   81   81   81
##  [82,]   82   82   82   82
##  [83,]   83   83   83   83
##  [84,]   84   84   84   84
##  [85,]   85   85   85   85
##  [86,]   86   86   86   86
##  [87,]   87   87   87   87
##  [88,]   88   88   88   88
##  [89,]   89   89   89   89
##  [90,]   90   90   90   90
##  [91,]   91   91   91   91
##  [92,]   92   92   92   92
##  [93,]   93   93   93   93
##  [94,]   94   94   94   94
##  [95,]   95   95   95   95
##  [96,]   96   96   96   96
##  [97,]   97   97   97   97
##  [98,]   98   98   98   98
##  [99,]   99   99   99   99
## [100,]  100  100  100  100
## [101,]  101  101  101  101
## [102,]  102  102  102  102
## [103,]  103  103  103  103
## [104,]  104  104  104  104
## [105,]  105  105  105  105
## [106,]  106  106  106  106
## [107,]  107  107  107  107
## [108,]  108  108  108  108
## [109,]  109  109  109  109
## [110,]  110  110  110  110
## [111,]  111  111  111  111
## [112,]  112  112  112  112
## [113,]  113  113  113  113
## [114,]  114  114  114  114
## [115,]  115  115  115  115
## [116,]  116  116  116  116
## [117,]  117  117  117  117
## [118,]  118  118  118  118
## [119,]  119  119  119  119
## [120,]  120  120  120  120
## [121,]  121  121  121  121
## [122,]  122  122  122  122
## [123,]  123  123  123  123
## [124,]  124  124  124  124
## [125,]  125  125  125  125
## [126,]  126  126  126  126
## [127,]  127  127  127  127
## [128,]  128  128  128  128
## [129,]  129  129  129  129
```

```
## [130,]  130  130  130  130
## [131,]  131  131  131  131
## [132,]  132  132  132  132
## [133,]  133  133  133  133
## [134,]  134  134  134  134
## [135,]  135  135  135  135
## [136,]  136  136  136  136
## [137,]  137  137  137  137
## [138,]  138  138  138  138
## [139,]  139  139  139  139
## [140,]  140  140  140  140
## [141,]  141  141  141  141
## [142,]  142  142  142  142
## [143,]  143  143  143  143
## [144,]  144  144  144  144
## [145,]  145  145  145  145
## [146,]  146  146  146  146
## [147,]  147  147  147  147
## [148,]  148  148  148  148
## [149,]  149  149  149  149
## [150,]  150  150  150  150
## [151,]  151  151  151  151
## [152,]  152  152  152  152
```

3.Partitioning train data into validation(testing) set and training set

```
intr<- createDataPartition(train2$classe, p = 0.6, list = FALSE)
training<- train2[intr,]#training set(60%)
validation<- train2[-intr,]#validation set(40%)
print("After partitioning training data into validation set(40%) and training set(60%) :")
```

```
## [1] "After partitioning training data into validation set(40%) and training set(60%) :"
```

```
print("Dimension of training set=")
```

```
## [1] "Dimension of training set="
```

```
dim(training)
```

```
## [1] 11776   153
```

```
print("Dimension of validation set=")
```

```
## [1] "Dimension of validation set="
```

```
dim(validation)
```

```
## [1] 7846  153
```

4.Model building by cross validation using Random Forest algorithm

```
mfn <- "myModel.RData"
if (!file.exists(mfn))
{
  nc <- makeCluster(detectCores() - 1)
  registerDoParallel(cores=nc)
  getDoParWorkers() # 3

  myModel  <- train(classe ~ ., data = training, method = "rf", metric = "Accuracy",        preProcess=
  )
  save(myModel , file = "myModel.RData")
  stopCluster(nc)
}else
{
  load(file = mfn, verbose = TRUE)
}
```

```
## Loading objects:
##   myModel
```

```
print(myModel, digits=4)
```

```
## Random Forest
##
## 11776 samples
##    52 predictor
##     5 classes: 'A', 'B', 'C', 'D', 'E'
##
## Pre-processing: centered (52), scaled (52)
## Resampling: Cross-Validated (4 fold)
## Summary of sample sizes: 8833, 8831, 8832, 8832
## Resampling results across tuning parameters:
##
##   mtry  Accuracy  Kappa
##    2    0.9881    0.9850
##   27    0.9875    0.9842
##   52    0.9783    0.9726
##
## Accuracy was used to select the optimal model using the largest value.
## The final value used for the model was mtry = 2.
```

5.Prediction

```
predTest <- predict(myModel, newdata=validation)
```

6.Confusion Matrix

```
confusionMatrix(predTest, factor(validation$classe))
```

```
## Confusion Matrix and Statistics
##
##           Reference
```

```
## Prediction    A    B    C    D    E
##          A 2232    2    0    0    0
##          B    0 1512    9    0    0
##          C    0    4 1357   17    0
##          D    0    0    2 1269    1
##          E    0    0    0    0 1441
##
## Overall Statistics
##
##                Accuracy : 0.9955
##                  95% CI : (0.9938, 0.9969)
##     No Information Rate : 0.2845
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.9944
##
##  Mcnemar's Test P-Value : NA
##
## Statistics by Class:
##
##                      Class: A Class: B Class: C Class: D Class: E
## Sensitivity            1.0000   0.9960   0.9920   0.9868   0.9993
## Specificity            0.9996   0.9986   0.9968   0.9995   1.0000
## Pos Pred Value         0.9991   0.9941   0.9848   0.9976   1.0000
## Neg Pred Value         1.0000   0.9991   0.9983   0.9974   0.9998
## Prevalence             0.2845   0.1935   0.1744   0.1639   0.1838
## Detection Rate         0.2845   0.1927   0.1730   0.1617   0.1837
## Detection Prevalence   0.2847   0.1939   0.1756   0.1621   0.1837
## Balanced Accuracy      0.9998   0.9973   0.9944   0.9932   0.9997
```

7.Complete data about the model

```
myModel$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: 2
##
##         OOB estimate of  error rate: 0.78%
## Confusion matrix:
##      A    B    C    D    E  class.error
## A 3345    3    0    0    0 0.0008960573
## B   16 2255    8    0    0 0.0105309346
## C    0   16 2037    1    0 0.0082765336
## D    0    0   42 1886    2 0.0227979275
## E    0    0    0    4 2161 0.0018475751
```

```
varImp(myModel)
```

```
## rf variable importance
```

```
##
##   only 20 most important variables shown (out of 52)
##
##                    Overall
## roll_belt           100.00
## yaw_belt             77.09
## magnet_dumbbell_z    69.90
## pitch_forearm        64.13
## magnet_dumbbell_y    63.00
## pitch_belt           57.83
## magnet_dumbbell_x    53.86
## roll_forearm         46.29
## accel_dumbbell_y     44.05
## accel_belt_z         42.33
## magnet_belt_z        42.26
## roll_dumbbell        41.21
## magnet_belt_y        39.30
## accel_dumbbell_z     36.27
## roll_arm             32.35
## accel_forearm_x      32.26
## gyros_belt_z         31.35
## accel_dumbbell_x     28.59
## yaw_dumbbell         28.48
## accel_arm_x          27.76
```

8. Quiz Coursera The testing is now performed on the Quiz set.

```
print(predict(myModel, newdata=test2))
```

```
##  [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
```

The above sequence is the set of answers obtained for the Quiz.

CITATIONS: The data for this project comes from : http://web.archive.org/web/20161224072740/http://groupware.les.inf.puc-rio.br/har