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
```

```
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

```
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:"
```

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

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

```
dim(train2)
```

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

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

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

```
dim(test2)
```

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

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

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 1514    8    0    0
##          C    0    2 1360   14    0
##          D    0    0    0 1272    1
##          E    0    0    0    0 1441
##
## Overall Statistics
##
##                Accuracy : 0.9966
##                  95% CI : (0.995, 0.9977)
##     No Information Rate : 0.2845
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.9956
##
##  Mcnemar's Test P-Value : NA
##
## Statistics by Class:
##
##                      Class: A Class: B Class: C Class: D Class: E
## Sensitivity            1.0000   0.9974   0.9942   0.9891   0.9993
## Specificity            0.9996   0.9987   0.9975   0.9998   1.0000
## Pos Pred Value         0.9991   0.9947   0.9884   0.9992   1.0000
## Neg Pred Value         1.0000   0.9994   0.9988   0.9979   0.9998
## Prevalence             0.2845   0.1935   0.1744   0.1639   0.1838
## Detection Rate         0.2845   0.1930   0.1733   0.1621   0.1837
## Detection Prevalence   0.2847   0.1940   0.1754   0.1622   0.1837
## Balanced Accuracy      0.9998   0.9981   0.9958   0.9945   0.9997
```

The out of sample error is 0.0037. The accuracy is 0.9964 and lies within the 95% confidence interval.

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