practical machine learning course project

Cuiming Zheng

Oct 16,2017

introduction

In this project,the goal is 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. The data for this project come from this source: http://groupware.les.inf.puc-rio.br/har. the goal is to predict the manner in which they did the exercise . This is the "classe" variable in the training set.

summary

I download the data in my computer. then clean the data to remove NA values and near 0 varialbes as well as non relevant variables. then I create data partation for training dataset. Since this is predict for classe, I comaprared models build with rpart method and random forest method. the result showed rnadom forest is more accuarate with 99.9% accuacy. I used 10 fold cross validation to increase accuacy for prediction. I use to model build from training data to predict test data and calculate the out sample error rate is low as 0.49%.

getting the data

following is code for Loading and preprocessing the data

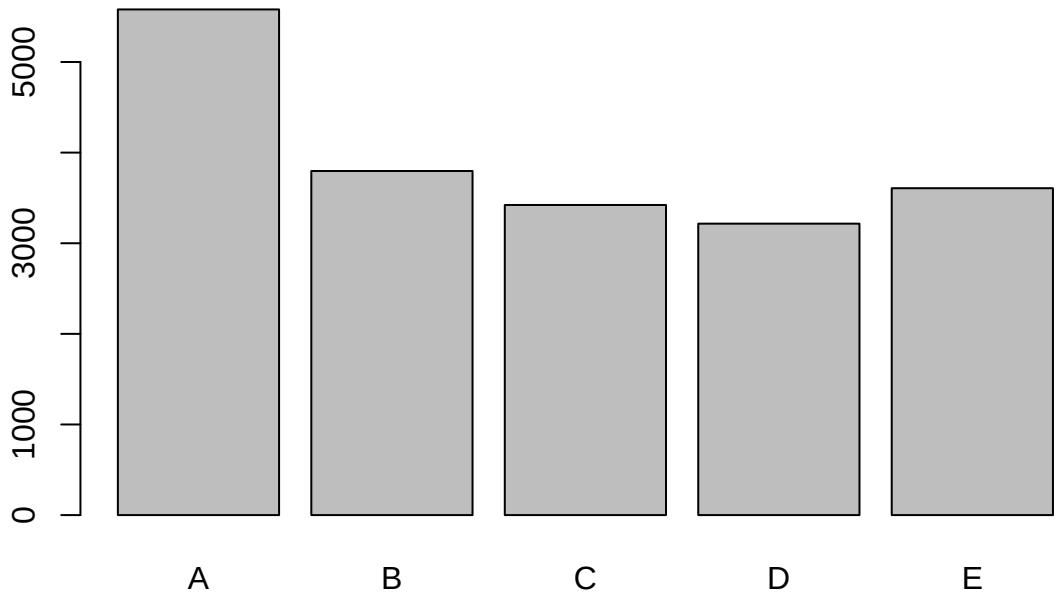
```
setwd("C:/Users/u292859/Desktop/data science/PML")
url="https://d396qusza40orc.cloudfront.net/predmachlearn/pml-training.csv"

download.file(url,destfile="C:/Users/u292859/Desktop/data science/PML/pml-training.csv")
data<-read.csv("pml-training.csv" )
URL="https://d396qusza40orc.cloudfront.net/predmachlearn/pml-testing.csv"
download.file(url,destfile="C:/Users/u292859/Desktop/data science/PML/pml-test.csv")
testdata<-read.csv("pml-test.csv" )
dim(data)
```

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

```
dim(testdata)
```

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

```
plot(data$classe,pch=19)
```



#cleanning the data ## following is code for data processing * follwoing is code for remove NA colomns and near 0 varaibles

```
library(caret)
```

```
## Warning: package 'caret' was built under R version 3.4.2
```

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

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

```
## Warning: package 'ggplot2' was built under R version 3.4.2
```

```
nsv<-nearZeroVar(data)
new<-data[,-nsv]
dim(new)
```

```
## [1] 19622   100
```

```
nacol<-which(colSums(is.na(new))>19000)
newtrain<-new[,-c(nacol)]
```

- following is code that remove first 6 variables

```
protrain<-newtrain[,-c(1:6)]
```

- following is code for data partition for training dataset

```
inTrain = createDataPartition(protrain$classe, p = 3/4)[[1]]
training = protrain[ inTrain,]
testing = protrain[-inTrain,]
dim(training)
```

```
## [1] 14718    53
```

```
dim(testing)
```

```
## [1] 4904   53
```

build prediction model

following are r code to build random forest model using traning data. choosed 10 fold of cross validation

```
model1<-train(classe~.,method="rf",data=training,trControl=trainControl(method="cv"),number=10)
```

```
## Warning: package 'randomForest' was built under R version 3.4.1
```

```
## randomForest 4.6-12
```

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

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

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

```
model1$finalModel
```

```
##
## Call:
##  randomForest(x = x, y = y, mtry = param$mtry, number = 10)
##                Type of random forest: classification
##                      Number of trees: 500
## No. of variables tried at each split: 2
##
##         OOB estimate of  error rate: 0.69%
## Confusion matrix:
##      A    B    C    D    E  class.error
## A 4181    3    1    0    0 0.0009557945
## B   12 2829    7    0    0 0.0066713483
## C    0   18 2544    5    0 0.0089598753
## D    1    0   45 2364    2 0.0199004975
## E    0    0    0    7 2699 0.0025868441
```

following use the model build from training data to predict testing data and evaluate prediction accuracy

```
pred<-predict(model1,testing)
confusionMatrix(testing$classe,pred)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction    A    B    C    D    E
##          A 1395    0    0    0    0
##          B    5  941    3    0    0
##          C    0    6  849    0    0
##          D    0    0   11  792    1
##          E    0    0    0    2  899
```

```
##
## Overall Statistics
##
##                Accuracy : 0.9943
##                  95% CI : (0.9918, 0.9962)
##     No Information Rate : 0.2855
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.9928
##  Mcnemar's Test P-Value : NA
##
## Statistics by Class:
##
##                      Class: A Class: B Class: C Class: D Class: E
## Sensitivity            0.9964   0.9937   0.9838   0.9975   0.9989
## Specificity            1.0000   0.9980   0.9985   0.9971   0.9995
## Pos Pred Value         1.0000   0.9916   0.9930   0.9851   0.9978
## Neg Pred Value         0.9986   0.9985   0.9965   0.9995   0.9998
## Prevalence             0.2855   0.1931   0.1760   0.1619   0.1835
## Detection Rate         0.2845   0.1919   0.1731   0.1615   0.1833
## Detection Prevalence   0.2845   0.1935   0.1743   0.1639   0.1837
## Balanced Accuracy      0.9982   0.9958   0.9911   0.9973   0.9992
```

following are R code to predict testdata using random forest model and eveluate prediction accuracy. Accuracy : 0.9988. 95% CI : (0.9982, 0.9992). The out sample error rate is calcuated in follwoing R code.the result showed is 0.49%.

```
testing$predright<-pred==testing$classe
(sum(testing$predright==FALSE))/(dim(testing)[1])
```

```
## [1] 0.005709625
```

follow ing are r code to build model using rpart method and evalaute prediction accuracy on testing data

```
model2<-train(classe~.,method="rpart",data=training)
pred2<-predict(model2,testing)
confusionMatrix(testing$classe,pred2)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction    A    B    C    D    E
##          A 1265   25  101    0    4
##          B  410  297  242    0    0
##          C  399   24  432    0    0
##          D  367  135  302    0    0
##          E  134  131  249    0  387
##
## Overall Statistics
##
##                Accuracy : 0.4855
##                  95% CI : (0.4714, 0.4996)
##     No Information Rate : 0.5251
##     P-Value [Acc > NIR] : 1
##
##                   Kappa : 0.3272
```

```
##  Mcnemar's Test P-Value : NA
##
## Statistics by Class:
##
##                      Class: A Class: B Class: C Class: D Class: E
## Sensitivity            0.4913  0.48529  0.32579       NA  0.98977
## Specificity            0.9442  0.84809  0.88178   0.8361  0.88611
## Pos Pred Value         0.9068  0.31296  0.50526       NA  0.42952
## Neg Pred Value         0.6267  0.92035  0.77920       NA  0.99900
## Prevalence             0.5251  0.12480  0.27039   0.0000  0.07973
## Detection Rate         0.2580  0.06056  0.08809   0.0000  0.07892
## Detection Prevalence   0.2845  0.19352  0.17435   0.1639  0.18373
## Balanced Accuracy      0.7177  0.66669  0.60378       NA  0.93794
```

```
library(rattle)
```

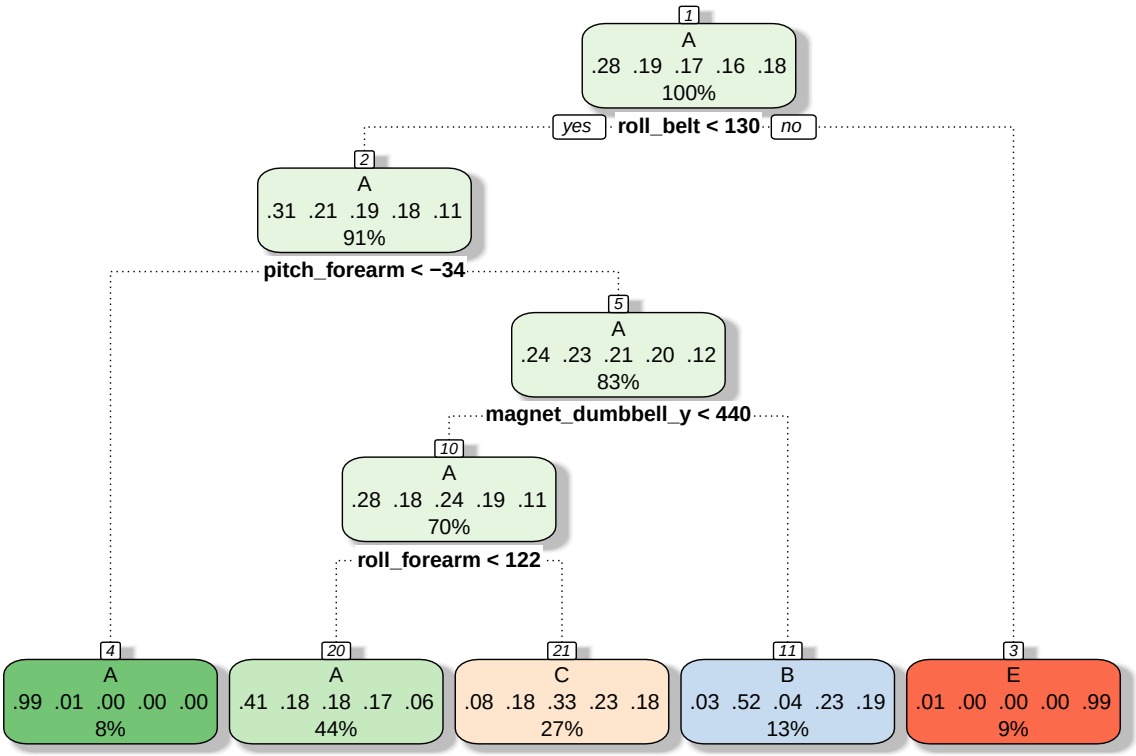
```
## Warning: package 'rattle' was built under R version 3.4.2
```

```
## Rattle: A free graphical interface for data science with R.
## Version 5.1.0 Copyright (c) 2006-2017 Togaware Pty Ltd.
## Type 'rattle()' to shake, rattle, and roll your data.
```

```
##
## Attaching package: 'rattle'
```

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

```
fancyRpartPlot(model2$finalModel)
```



Rattle 2017−Oct−17 22:30:13 u292859

conclusion using random forest model has good predition on testing data compared with rpart method. therefore I choose random forest as final model to predict 20 testcases.