ParticalMachineLeaning_couse4

Y.Yokota

27/1/2020

Overview

This reports describes that

1.Preprocess Remove the colums which includs NA valume more than 80%. Also, check the zero convariates. 2.Create prediction model.Be aware of multi-corrleation. The methods of dicision tree,randam fores, and boosting will be used.

3.Cross validation with training set. In sample versus out of sample error, prevent from overfitting.

4.The resons of the prediction model choice

5.Test the data with the prediction model

Method

The data was obtained from following Websites. 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 this source: http://web.archive.org/web/20161224072740/http:/groupware.les.inf.puc-rio.br/har (http://web.archive.org/web/20161224072740/http:/groupware.les.inf.puc-rio.br/har).

Results

install library

```
library(caret)
```

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

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

```
library(ISLR)
library(rattle)
```

```
## Rattle: A free graphical interface for data science with R.
## バージョン 5.3.0 Copyright (c) 2006-2018 Togaware Pty Ltd.
## 'rattle()' と入力して、データを多角的に分析します。
```

```
library(doParallel)
```

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

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

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

```
library(tictoc)
library(e1071)
```

set the traing/test data.

```
  train<-read.csv("/Users/yuki/Documents/R/Cousera_PML/pml-training.
csv",header = T,na.strings = c("#DIV/0!","","NA"))
  test<-read.csv("/Users/yuki/Documents/R/Cousera_PML/pml-testing.cs
v",header = T,na.strings= c("#DIV/0!","","NA"))
```

remove the colums which includs NA valume more than 80%.

```
# Following colum names are improtant for later analyisis.
  y1<-names(train[c(1:7,160)])

na.ratio<-function(x){sum(is.na(x)=="TRUE")/length(x)}
  y<-apply(train[,-c(1:7,160)],2,na.ratio)
  y2<-names(subset(y,y<0.8))

train2<-data.frame(train[160],train[y2])
```

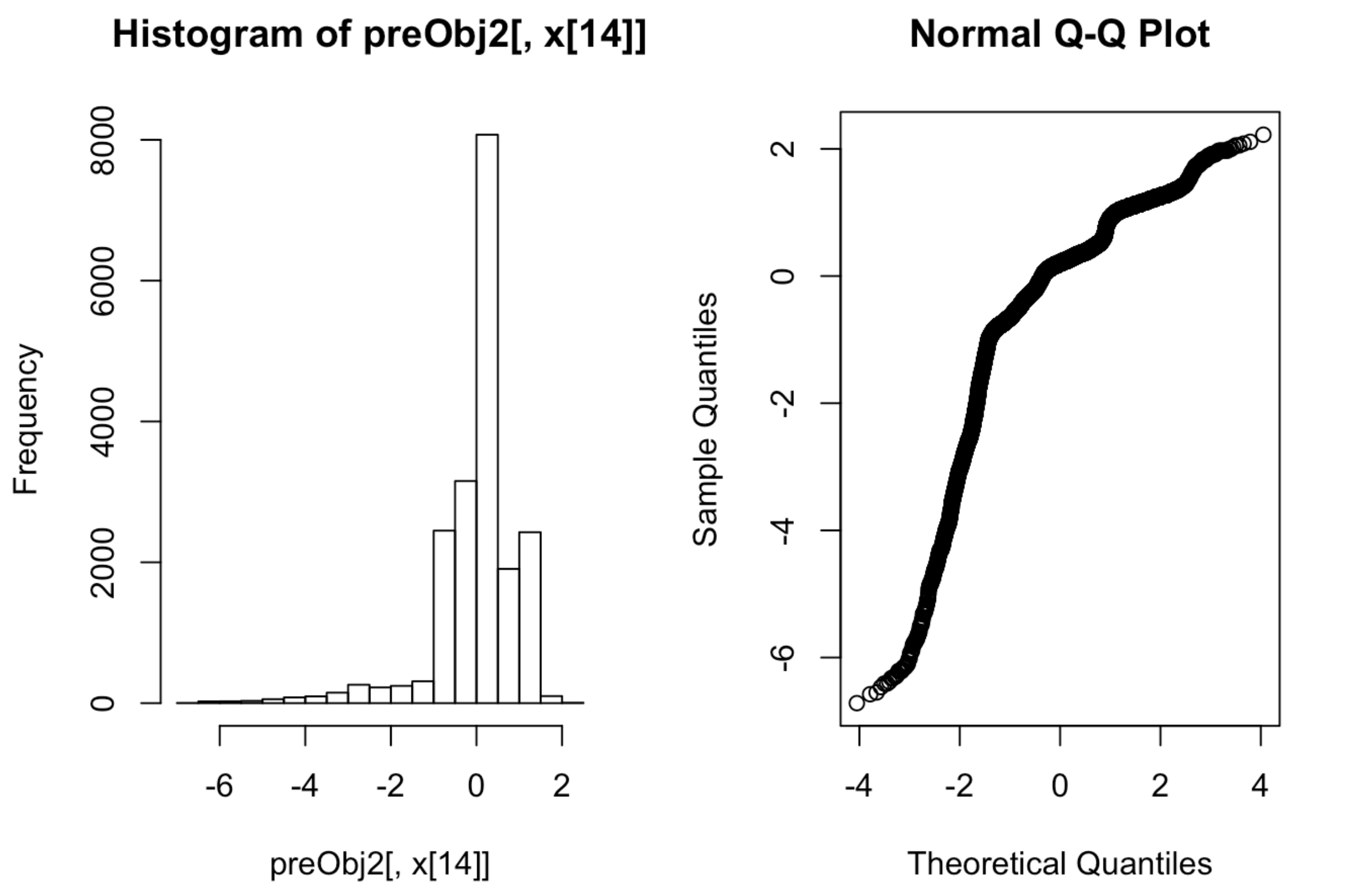
removing zero convariates

```
 preObj<-preProcess(train2,method=c("center","scale"))
 predict(preObj,train2)->preObj2

nsv<-nearZeroVar(train2,saveMetrics = T)
nsv2<-subset(nsv,nzv=="FALSE")
train3<-(train[,rownames(nsv2)])

x<-0
for (i in 1:nrow(nsv2)){
  x<-append(x,grep(rownames(nsv2)[i],colnames(train3)))
}

# check the histgram
par(mfrow=c(1,2))
hist(preObj2[,x[14]])
qqnorm(preObj2[,x[14]])
```



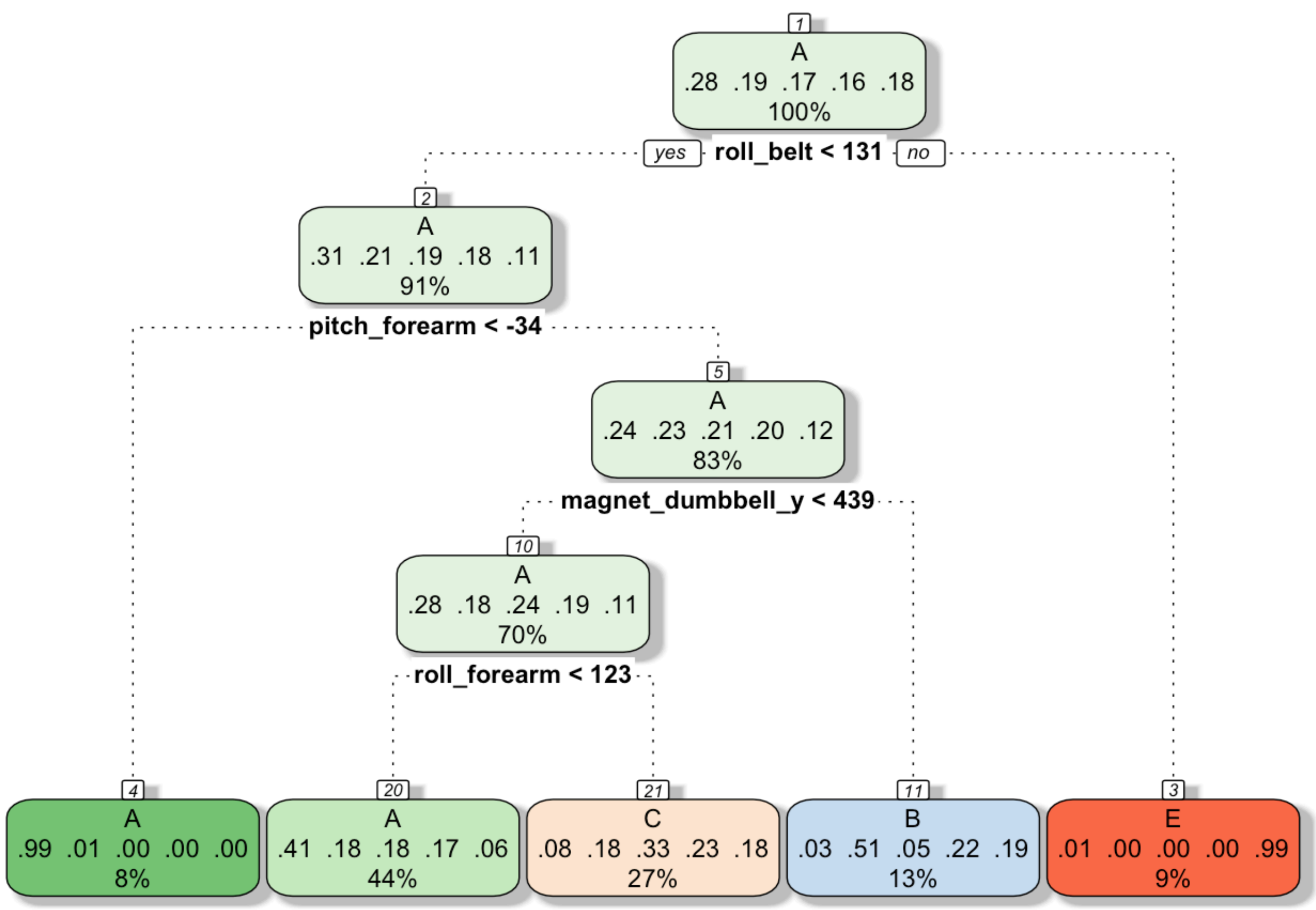

```
test2<-data.frame(test[y2])
```

Create the model “Predicting with trees”

```
cl <- makePSOCKcluster(4)
registerDoParallel(cl)
tic()

   #folds<-createFolds(y=train2$classe,k=10,list=T,returnTrain =F)
  trainPart <- createDataPartition(train3$classe, p=0.70, list=F)
    trainSubset <- train3[trainPart, ]
    validSubset <- train3[-trainPart, ]

  modFit<-train(classe~.,data=trainSubset,method="rpart")
  fancyRpartPlot(modFit$finalModel)
```



Rattle 2020- 1-27 08:06:20 yuki

```
    Pred<-predict(modFit,validSubset)
    table(Pred,validSubset$classe)
```

```
##
## Pred    A    B    C    D    E
##    A 1527  469  481  428  147
##    B   29  384   25  168  149
##    C  117  286  520  368  313
##    D    0    0    0    0    0
##    E    1    0    0    0  473
```

```
toc()
```

```
## 13.049 sec elapsed
```

```
stopCluster(cl)
```

Create the model “Randam Forest”

```
cl <- makePSOCKcluster(4)
registerDoParallel(cl)
tic()
    modFit2<-train(classe~.,data=trainSubset,method="rf",prox=T)
    modFit2
```

```
## Random Forest
##
## 13737 samples
##    52 predictor
##     5 classes: 'A', 'B', 'C', 'D', 'E'
##
## No pre-processing
## Resampling: Bootstrapped (25 reps)
## Summary of sample sizes: 13737, 13737, 13737, 13737, 13737, 13737
, ...
## Resampling results across tuning parameters:
##
##   mtry  Accuracy   Kappa
##    2    0.9885105  0.9854658
##   27    0.9886352  0.9856246
##   52    0.9796360  0.9742395
##
## Accuracy was used to select the optimal model using the largest v
alue.
## The final value used for the model was mtry = 27.
```

```
    Pred2<-predict(modFit2,validSubset)
    table(Pred2,validSubset$classe)
```

```
##
## Pred2    A    B    C    D    E
##     A 1673   12    0    0    0
##     B    0 1125    5    0    0
##     C    1    2 1018   11    3
##     D    0    0    3  953    8
##     E    0    0    0    0 1071
```

```
toc()
```

```
## 4197.716 sec elapsed
```

```
stopCluster(cl)
```

Create the model “Boosting”

```
cl <- makePSOCKcluster(4)
registerDoParallel(cl)
tic()
    modFit3<-train(classe~.,data=trainSubset,method="gbm",verbose=F)
    modFit3
```

```
## Stochastic Gradient Boosting
##
## 13737 samples
##    52 predictor
##     5 classes: 'A', 'B', 'C', 'D', 'E'
##
## No pre-processing
## Resampling: Bootstrapped (25 reps)
## Summary of sample sizes: 13737, 13737, 13737, 13737, 13737, 13737
, ...
## Resampling results across tuning parameters:
##
##   interaction.depth  n.trees  Accuracy   Kappa
##   1                   50      0.7492410  0.6820228
##   1                  100      0.8164125  0.7675400
##   1                  150      0.8498122  0.8098757
##   2                   50      0.8525329  0.8130941
##   2                  100      0.9039753  0.8784375
##   2                  150      0.9280041  0.9088712
##   3                   50      0.8944516  0.8663303
##   3                  100      0.9382539  0.9218437
##   3                  150      0.9571930  0.9458308
##
## Tuning parameter 'shrinkage' was held constant at a value of 0.1
##
## Tuning parameter 'n.minobsinnode' was held constant at a value of
10
## Accuracy was used to select the optimal model using the largest v
alue.
## The final values used for the model were n.trees = 150, interacti
on.depth =
##  3, shrinkage = 0.1 and n.minobsinnode = 10.
```

```
    Pred3<-predict(modFit3,validSubset)
    table(Pred3,validSubset$classe)
```

```
##
## Pred3    A    B    C    D    E
##     A 1646   45    1    1    3
##     B   18 1051   24    3    8
##     C    6   40  989   33    9
##     D    3    3   11  920   20
##     E    1    0    1    7 1042
```

```
toc()
```

```
## 314.567 sec elapsed
```

```
stopCluster(cl)
```

Conclusion

Comparied the dicision tree,randam fores, and boosting methods, the randam forest shows the best accuracy. The 52 predictors which has less than 50 % NA value in each colum was adapted for this models. Therefore, it is concluded that the randam forest as used this dataset, and the prediction with test set shows below.

```
 predict(modFit2,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
```

Supplymental data

No data.