technical document

Alan Chen, Ssu Hsien Lee, Kasturi Pal, Deepanshu Kataria

2023-07-22

# Load library

library(dplyr)  
library(ggplot2)  
library(caret)  
library(class)  
library(rpart)  
library(pROC)  
library(rpart.plot)   
library(e1071)

# Load the file

XYZ = read.csv('/Users/chenshaokai/Desktop/UMN/Semester1/R/assignment2/grouppart/XYZData.csv')

# Observe data and do the oversampling

*# remove the userID column that is not helpful for our prediction*  
XYZ\_for\_test1 = XYZ[,2:27]  
  
*# split our sample data into 70% for cross-validation 30% for the final-testing*  
train\_rows = createDataPartition(y = XYZ\_for\_test1$adopter, p = 0.70, list = FALSE)  
XYZData\_train = XYZ\_for\_test1[train\_rows,]  
XYZData\_test = XYZ\_for\_test1[-train\_rows,]  
  
*#observing our data, discover it is imbalance and need oversampling for help*  
table(XYZ\_for\_test1$adopter)

##   
## 0 1   
## 40000 1540

library(ROSE)  
oversample\_traindata = ovun.sample(adopter ~ .,   
 data = XYZData\_train,   
 method = 'over',  
 N = nrow(XYZData\_train)\*1.5)$data

# Do the cross-validation for Decision tree model. Then pre-prune plus post-prune to find one best model.

*# Seperate our data into five folds for cross-validation test*  
cv = createFolds(y = oversample\_traindata$adopter, k = 5)  
auc\_all = c()  
**for** (test\_rows **in** cv) {  
 XYZcross\_train = oversample\_traindata[-test\_rows,]  
 XYZcross\_test = oversample\_traindata[test\_rows,]  
 tree\_preprun = rpart(adopter ~ ., data = XYZcross\_train,   
 method = "class",  
 parms = list(split = "information"),  
 control = rpart.control(cp = 0,   
 maxdepth = 4,  
 ))  
 pred\_tree = predict(tree\_preprun, XYZcross\_test, type = "prob")  
 tree.roc = roc(response = XYZcross\_test$adopter,   
 predictor = pred\_tree[,2])  
 auc\_all = c(auc\_all,auc(tree.roc))  
}

# Print out the auc for decision tree models of each folds and the mean auc

auc\_all

## [1] 0.7760578 0.7572555 0.7421945 0.7483321 0.7906612

mean(auc\_all)

## [1] 0.7629002

# Also test on the Naive Bayes model with the same way

auc\_all = c()  
**for** (test\_rows **in** cv) {  
 XYZcross\_train = oversample\_traindata[-test\_rows,]  
 nb\_XYZcross\_train = naiveBayes(adopter ~ ., data = XYZcross\_train)  
 XYZcross\_test = oversample\_traindata[test\_rows,]  
 pred\_nb = predict(nb\_XYZcross\_train, XYZcross\_test)  
 prob\_pred\_nb = predict(nb\_XYZcross\_train, XYZcross\_test, type = "raw")  
 nb.roc = roc(response = XYZcross\_test$adopter,   
 predictor = prob\_pred\_nb[,2])  
 auc\_all = c(auc\_all,auc(nb.roc))  
  
}

# Print out the auc for Naive Bayes models of each folds and the mean auc

auc\_all

## [1] 0.7351892 0.7471412 0.7373770 0.7495580 0.7455136

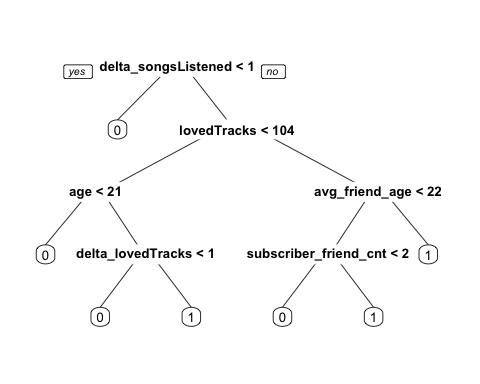
mean(auc\_all)

## [1] 0.7429558

## After comparing the AUC between these two models, we decided to choose the Decision Tree model with a higher AUC

# Do the final testing with the 30% of our test data

tree\_final = rpart(adopter ~ ., data = oversample\_traindata,   
 method = "class",  
 parms = list(split = "information"),  
 control = rpart.control(cp = 0,   
 maxdepth = 4,  
 ))  
pred\_tree\_final = predict(tree\_final, XYZData\_test, type = "prob")  
prp(tree\_final, varlen = 0)



tree.roc\_final = roc(response = XYZData\_test$adopter,   
 predictor = pred\_tree\_final[,2])  
  
auc\_final = tree.roc\_final  
auc\_final

##   
## Call:  
## roc.default(response = XYZData\_test$adopter, predictor = pred\_tree\_final[, 2])  
##   
## Data: pred\_tree\_final[, 2] in 11963 controls (XYZData\_test$adopter 0) < 499 cases (XYZData\_test$adopter 1).  
## Area under the curve: 0.7128