abb\_retention\_starter—Amol.R

Amol Pathak

2023-05-05

# The final project  
# objective: design a targeted customer retaining strategy  
  
rm(list = ls())  
setwd("H:/Data Driven Marketing/Final Project")  
# install.packages("tree")  
# install.packages("ranger")  
require(gamlr)

## Loading required package: gamlr

## Warning: package 'gamlr' was built under R version 4.1.3

## Loading required package: Matrix

require(tree)

## Loading required package: tree

## Warning: package 'tree' was built under R version 4.1.3

require(ranger)

## Loading required package: ranger

## Warning: package 'ranger' was built under R version 4.1.3

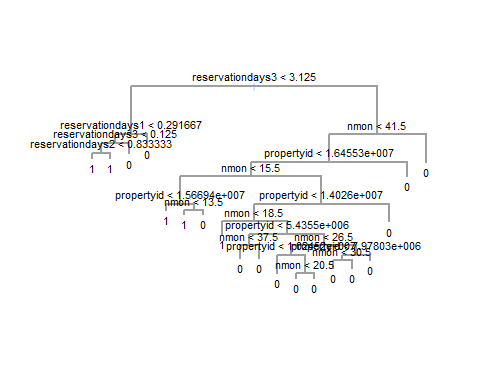
attr = read.csv("abb\_attrition.csv")  
attr$attrition = factor(attr$attrition)  
  
# training and test sample  
n = dim(attr)[1]  
set.seed(1)  
tsid = sample.int(n,floor(n/5))  
attr0 = attr[-tsid,] # training sample  
attr1 = attr[tsid,] # testing sample  
  
y0 = as.numeric(as.character(attr0$attrition))  
y1 = as.numeric(as.character(attr1$attrition))  
  
# define functions of error rate and deviance  
er = function(y,yhat){  
 y = as.numeric(y);yhat = as.numeric(yhat>0.5)  
 z = sum(abs(y-yhat))/length(y)  
 return(z)  
}  
  
devf = function(y,p){  
 lh = y\*p + (1-y)\*(1-p)  
 dev = -2\*sum(log(pmin(pmax(lh,1e-10),1-1e-10)))  
 return(dev)  
}  
  
  
  
# 0. lasso with interaction effects  
x0 = model.matrix(attrition ~ .\*latitude\*longitude,data = attr0)  
x1 = model.matrix(attrition ~ .\*latitude\*longitude,data = attr1)  
lr = cv.gamlr(x0,y0,lmr=1e-4,family = "binomial")  
yhatlr0 = predict(lr,x0,type = "response",select = "min")  
yhatlr1 = predict(lr,x1,type = "response",select = "min")  
  
# report the prediction error rate and deviance for the testing sample  
er1 = er(y1,yhatlr1)  
dev1 = devf(y1,yhatlr1)  
print('OOS error rates and deviance')

## [1] "OOS error rates and deviance"

c(er1,dev1)

## [1] 0.1975591 1079.5932420

# 1. classification tree (please replace this section with your best model for predicting attrition)  
attree = tree(attrition ~ ., data = attr0,mindev = 0.005)  
plot(attree,col=8,lwd = 2,cex = 0.5);text(attree,cex=0.68)



cvat = cv.tree(attree)  
  
yhattree0 = predict(attree,attr0,type = "vector",eps = 1e-10)[,2]  
yhattree1 = predict(attree,attr1,type = "vector",eps = 1e-10)[,2]  
  
# report the prediction error rate and deviance  
print('classification tree')

## [1] "classification tree"

er1 = er(y1,yhattree1)  
dev1 = devf(y1,yhattree1)  
print('OOS error rates and deviance')

## [1] "OOS error rates and deviance"

c(er1,dev1)

## [1] 0.1769641 1011.4585684

## My Model  
rf <- ranger(attrition ~ ., data=attr0, mtry = 20,  
 num.tree=500, classification = TRUE, probability = TRUE)  
  
yhatrf1 = predict(rf,attr1)$predictions[,2]  
  
# report the prediction error rate and deviance  
print('Random Forest Model')

## [1] "Random Forest Model"

er1 = er(y1,yhatrf1)  
dev1 = devf(y1,yhatrf1)  
print('OOS error rates and deviance')

## [1] "OOS error rates and deviance"

c(er1,dev1)

## [1] 0.1586575 891.6945870

## Notes:  
# With the Random Forest, I found the lowest error rate and deviance.  
# "OOS error rates and deviance"  
# 0.1495042 891.3377399  
  
# 2. calculation of total profits  
airbnb.perc = 0.15  
  
reservation.cols = grep("reservationdays", names(attr), ignore.case = TRUE) # Getting the reservation col ids  
  
# Helper function to calculate Revenue  
revenue = function(rate,res.cols){  
 total.res = rowSums(res.cols)  
 revenue = rate\*total.res  
 return(revenue)  
}  
  
# Helper function to calculate Profit  
profit = function(rate,res.cols){  
 revenue = revenue(rate,res.cols)  
 profit.amount = revenue\*airbnb.perc  
 return(profit.amount)  
}  
  
prop.2015 = attr1  
prop.2016 = prop.2015[which(prop.2015$attrition == 0),] # Properties retained  
  
# Total Profit in 2015  
profit.2015 = sum(profit(prop.2015$averagedailyrateusd, prop.2015[,reservation.cols]))   
print(paste("The net profit earned in 2015 is", round(profit.2015,2)))

## [1] "The net profit earned in 2015 is 6274837.87"

# Total Profit in 2016  
profit.2016 = sum(profit(prop.2016$averagedailyrateusd, prop.2016[,reservation.cols]))   
print(paste("The net profit earned in 2016 is", round(profit.2016,2)))

## [1] "The net profit earned in 2016 is 5232327.62"

# 3. what is the cutoff value for the attrition probability if a property's revenue is $25000?  
cost = 1000  
  
# Defining p.star as a function of revenue  
p.star = function(rev){  
 prof = airbnb.perc\*rev  
 return(cost/prof)  
}  
  
print(paste("The cutoff probability if for a property generating 250000 revenue is", p.star(25000)))

## [1] "The cutoff probability if for a property generating 250000 revenue is 0.266666666666667"

# 4. report the total net profit for properties in the testing sample under your retaining strategy  
  
# 4.1  
# Using the RF model we have the predicted attrition probabilities for Test Sample  
attr1$prob = yhatrf1  
  
# 4.2  
# Using the revenue function to obtain the revenue information for each property in Test Sample  
attr1$revenue = revenue(attr1$averagedailyrateusd, attr1[,reservation.cols])  
  
# Defining p.star for each property based on revenue  
attr1$pstar = p.star(attr1$revenue)  
  
attr1$targeted = as.numeric(attr1$prob > attr1$pstar)  
attr1$retained = as.numeric(attr1$attrition == 0 | attr1$targeted == 1)  
  
# Total Profit Calculation  
gross.profit = sum((attr1$revenue \* airbnb.perc)[attr1$retained == 1])  
campaign.cost = sum(cost \* attr1$targeted)  
print(paste("The gross profit earned from remaining properties is", round(gross.profit,2)))

## [1] "The gross profit earned from remaining properties is 6044265.81"

print(paste("The total cost of targeted campaign is", round(campaign.cost,2)))

## [1] "The total cost of targeted campaign is 363000"

net.profit = gross.profit - campaign.cost  
print(paste("The net profit earned from targeted campaign is", round(net.profit,2)))

## [1] "The net profit earned from targeted campaign is 5681265.81"

# Improvement?  
print(paste("The total increase in profit from targeted campaign is", round(net.profit-profit.2016,2)))

## [1] "The total increase in profit from targeted campaign is 448938.2"