code0505

zihan

2018年5月5日

#论文实验 期望利润法及EMP法比较  
library(plyr)  
library(ggthemes)  
library(sampling)  
library(DMwR)  
library(EMP)  
#library(car)  
library(gbm)  
library(caret)  
  
setwd("C:/自建/川大商学院/导师/R数据集")  
#公用函数  
#期望利润法  
#基于流失预测结果流失概率计算客户的期望利润  
ep<-function(x){ #参数为流失预测中预测集计算的流失概率  
 n<-length(x)  
 m<-vector(mode="numeric",length=0)   
 for(i in 1:n)  
{ #pi=rbeta(1,6,14)   
 #ri=rbeta(1,9,1)   
 m[i]<-x[i]\*190\*0.3-10\*(1-x[i])\*0.9-1 } #Vip,δ,φ提前计算出结果  
 return (m)  
}  
  
#计算总利润TotalProfit，需自定义函数  
TP<-function(x){  
 n<-length(x)  
 percents<-vector(mode<-"numeric",length=0)  
 tp<-vector(mode<-"numeric",length=0)  
 tp<-cumsum(x)  
 nn<-c(1:n)  
 percents<-nn/n\*100  
 out<-list(tp,percents)  
}  
  
#期望利润法对应的实际利润  
actual\_profits<-function(x){ #参数为数据集实际状态churner/nonchurner  
 n<-length(x)  
 m<-vector(mode="numeric",length=0)  
 for(i in 1:n)  
 { pi<-rbeta(1,6,14)   
 ri<-rbeta(1,9,1)   
 if(x[i]==0)  
 m[i]<--ri\*10-1 #非流失者  
 else  
 m[i]<-pi\*190-1 #流失者  
 }  
 return (m)  
}  
  
#数据集Churn.data\_s  
load("KDD.RData")  
#Churn.data\_s$churn <- ifelse(Churn.data\_s$churn == "1",1,0)  
#str(Churn.data\_s)  
#SMOTE抽样  
lst\_act<-list() #训练集实际累积利润  
lst\_act\_per<-list() #训练集实际累积利润对应百分比  
lst\_exp<-list() #训练集期望累积利润  
lst\_exp\_perplot<-list() #训练集期望累积利润对应百分比  
lst\_exp\_per<-list() #期望利润法最大利润百分比  
lst\_exp\_test<-list() #测试集期望利润(>0)  
lst\_stim\_test<-list() #测试集模拟利润  
lst\_exp\_per<-list() #测试集最大利润下的客户比例  
lst\_emp\_act<-list() #训练集emp法实际利润 （累积）  
#lst\_emp\_actper<-list() #训练集emp法实际累积利润对应百分比  
lst\_emp\_emp<-list() #训练集emp法emp利润（累积）  
#lst\_emp\_empper<-list() #训练集emp法emp累积利润对应百分比  
lst\_emp\_per<-list() #训练集emp法百分比记录（用于求均值）  
lst\_emp\_stim<-list() #emp法测试集模拟利润  
lst\_emp\_EMP<-list() #emp法测试集emp利润  
  
  
for(i in 1:10)  
{  
sub<-createDataPartition(Churn.data\_s$churn,p=0.75,list=FALSE)  
trainset<-Churn.data\_s[sub,]  
testset<-Churn.data\_s[-sub,]  
trainset<-SMOTE(churn~.,trainset,perc.over=100,perc.under=200)  
   
#训练模型  
trainset$churn <- ifelse(trainset$churn== "1",1,0)  
churn.gbm <-gbm(formula = churn ~ .,distribution = "bernoulli",  
 data = trainset,n.trees = 1000,interaction.depth = 7,  
 shrinkage = 0.01,cv.folds = 3)  
  
 chst<-trainset$churn  
 churnP<- predict(churn.gbm,trainset,type="response")   
 newsubtset<-cbind(chst,churnP)  
 table\_1<-as.data.frame(newsubtset)  
 table\_2<-arrange(table\_1,desc(churnP))  
  
 #50次计算实际利润,期望利润 -期望利润法  
 actm<-matrix(sapply(1:50,function(x)(actual\_profits(chst))),50,  
 length(table\_2$chst),byrow=TRUE) #按行存储   
 actualprofits<-colMeans(actm)  
 actual\_total\_tr<-TP(actualprofits) #训练集每次划分后计算累积利润及对应百分比  
 lst\_act<-list(lst\_act,actual\_total\_tr[[1]]) #每次划分下的累积值存入列表  
 lst\_act\_per<-list(lst\_act\_per,actual\_total\_tr[[2]]) #百分比   
  
 #期望利润求均值  
 expect<-ep(churnP)  
 total\_exp\_tr<-TP(expect) #每次划分后训练集累积期望利润及对应百分比  
 lst\_exp<-list(lst\_exp,total\_exp\_tr[[1]])  
 lst\_exp\_perplot<-list(lst\_exp\_perplot,total\_exp\_tr[[2]]) #百分比  
  
 #训练集期望最大利润的目标客户比例  
 optper=length(expect[expect>0])/length(expect)  
 lst\_exp\_per<-list(lst\_exp\_per,optper)  
  
  
  
#测试集  
#预测  
churn.predict <- predict(churn.gbm,testset,type="response")  
chrate<-churn.predict   
chst\_te<- ifelse(testset$churn== "1",1,0) #测试集客户实际状态  
table\_te\_1<-as.data.frame(cbind(chst\_te,chrate))  
table\_te\_2<-arrange(table\_te\_1,desc(chrate))  
#profit of trainningdata   
newlength\_ex=floor(optper\*length(chrate)) #训练集目标客户比例在测试集中的目标客户数  
expect\_test<-ep(table\_te\_2$chrate[1:newlength\_ex])  
expect\_test\_max<-sum(expect\_test) #目标客户的最大总利润  
  
  
chst\_te\_1<-table\_te\_2$chst\_te[1:newlength\_ex] #选取期望利润大于0的客户对应的流失概率  
stim\_test<-matrix(sapply(1:50,function(x)(actual\_profits(chst\_te\_1))),50,  
 newlength\_ex,byrow=TRUE) #目标客户的模拟利润  
stim\_test\_max<-sum(colMeans(stim\_test)) #目标客户的模拟总利润  
  
#50次计算结果累积和按行存储-一行一次计算结果  
lst\_exp\_test<-list(lst\_exp\_test,expect\_test\_max) #期望利润法总利润  
lst\_stim\_test<-list(lst\_stim\_test,stim\_test\_max) #模拟利润总利润  
  
#EMP法  
EMP\_tr<-empChurn(churnP, trainset$churn)  
newlength<-floor(length(churnP)\*(EMP\_tr$EMPfrac)) #EMP下目标客户规模  
#训练集实际利润  
act\_emp\_tr\_m<-matrix(sapply(1:50,function(x)(actual\_profits(table\_2$chst[1:newlength]))),50,  
 newlength,byrow=TRUE)  
act\_emp\_tr<-TP(colMeans(act\_emp\_tr\_m))  
lst\_emp\_act<-list(lst\_emp\_act,act\_emp\_tr[[1]]) #训练集累积实际利润  
lst\_emp\_actper<-list(lst\_emp\_actper,act\_emp\_tr[[2]]) #训练集累积实际利润对应百分比forplot  
  
emp\_emp\_tr<-TP(rep(EMP\_tr$EMP,time=newlength))  
lst\_emp\_emp<-list(lst\_emp\_emp,emp\_emp\_tr[[1]]) #训练集emp累积利润  
lst\_emp\_empper<-list(lst\_emp\_empper,emp\_emp\_tr[[2]]) #训练集emp累积利润对应百分比forplot  
lst\_emp\_per<-list(lst\_emp\_per,EMP\_tr$EMPfrac) #记录每次目标比例  
  
#EMP下测试集模拟利润  
emp\_stim\_te<-matrix(sapply(1:50,function(x)(actual\_profits(table\_te\_2$chst\_te[1:newlength]))),50,  
 newlength,byrow=TRUE)  
emp\_stim<-sum(colMeans(emp\_stim\_te)) #每次划分下经50次计算的模拟利润和  
lst\_emp\_stim<-list(lst\_emp\_stim,emp\_stim)  
  
EMP\_emp<-EMP\_tr$EMP\*(floor(length(churn.predict)\*(EMP\_tr$EMPfrac))) #EMP法每次划分下EMP总利润  
lst\_emp\_EMP<-list(lst\_emp\_EMP,EMP\_emp)   
  
  
 } #endfor  
  
#训练集--期望利润法-10次划分期望利润  
exp\_tr<-matrix(unlist(lst\_exp),10,length(expect),byrow=TRUE)  
exp\_trM<-colMeans(exp\_tr) #累积期望利润  
exp\_per\_tr1<-matrix(unlist(lst\_exp\_perplot),10,length(total\_exp\_tr[[2]]),byrow=TRUE)  
exp\_per\_tr<-colMeans(exp\_per\_tr1)  
  
#训练集--期望利润法-10次划分对应实际利润  
act\_tr<-matrix(unlist(lst\_act),10,length(actualprofits),byrow=TRUE)  
act\_trM<-colMeans(act\_tr) #累计实际利润  
act\_per\_tr1<-matrix(unlist(lst\_act\_per),10,length(actual\_total\_tr[[2]]),byrow=TRUE) #对应累积百分比  
act\_per\_tr<-colMeans(act\_per\_tr1) #累积百分比  
  
#模拟图（训练集期望利润法下的利润与实际利润图）  
#x11(width = 6, height = 4)  
plot(act\_per\_tr,act\_trM,type="l",main="Churn.data\_s of expected profits",  
xlab="percentile",ylab="profit")  
lines(exp\_per\_tr,exp\_trM,type="l",lty=2) #实际利润  
abline(0,0)  
legend("bottomright", inset=0.05, lty=c(1,2),c("actual profit","predict profit"))  
  
#训练集--emp-10次划分实际利润  
#for(i in 1:2)  
#{  
# if(length(lst\_emp\_act[[i]])<max(lengths(lst\_emp\_act)))  
# { lst\_emp\_act[[i]]<-c(lst\_emp\_act[[i]],rep(0,max(lengths(lst\_emp\_act))-length(lst\_emp\_act[[i]])))}  
#}  
#emp\_act<-matrix(unlist(lst\_emp\_act),10,max(lengths(lst\_emp\_act)),byrow=TRUE) #累积实际利润  
#emp\_actM<-colMeans(emp\_act) #累积实际利润  
#emp\_per\_act<-colMeans(matrix(unlist(lst\_emp\_actper),10,newlength,byrow=TRUE)) #累积利润对应百分比   
  
#训练集——EMP-10次划分emp利润  
#for(i in 1:2)  
#{  
# if(length(lst\_emp\_emp[[i]])<max(lengths(lst\_emp\_emp)))  
# { lst\_emp\_emp[[i]]<-c(lst\_emp\_emp[[i]],rep(0,max(lengths(lst\_emp\_emp))-length(lst\_emp\_emp[[i]])))}  
#}  
#emp\_emp<-colMeans(matrix(unlist(lst\_emp\_emp),10,max(lengths(lst\_emp\_emp)),byrow=TRUE))  
#emp\_empper<-colMeans(matrix(unlist(lst\_emp\_empper),10,newlength,byrow=TRUE))  
  
  
#emp法训练集模拟图  
plot(emp\_per\_act,emp\_actM,type="l",main="Churn.data\_s of EMP",  
xlab="percentile",ylab="profit") #实际利润  
lines(emp\_empper,emp\_emp,type="l",lty=2) #emp利润  
legend("bottomright", inset=0.05, lty=c(1,2),c("actual profit","emp profit"))  
  
  
#测试集--期望利润法利润及模拟利润，百分比  
exp\_te<-matrix(unlist(lst\_exp\_test),10,1,byrow=TRUE)  
exp\_teMax<-mean(exp\_te) #最大期望利润之和  
exp\_stim<-mean(matrix(unlist(lst\_stim\_test),10,1,byrow=TRUE))  
exp\_percent<-mean(matrix(unlist(lst\_exp\_per),10,1,byrow=TRUE)) #10次划分后期望利润法最大利润下的平均客户比例  
  
  
#测试集--emp法emp利润及模拟利润，百分比  
 emp\_stim\_sum<-mean(matrix(unlist(lst\_emp\_stim),10,1,byrow=TRUE))  
 #emp\_stim\_sum<-mean(emp\_stim\_1) #emp目标比例下模拟总利润  
 EMP\_emp\_last<-mean(matrix(unlist(lst\_emp\_EMP),10,length(EMP\_emp),byrow=TRUE)) #emp法10次划分的平均EMP利润  
 emp\_percent<-mean(matrix(unlist(lst\_emp\_per),10,1,byrow=TRUE))   
  
  
#最终结果  
  
#期望利润法：  
  
exp\_percent #最大利润下的客户比例，  
exp\_stim #测试集目标客户的模拟总利润  
exp\_teMax # 测试集最大期望利润之和  
  
#emp法  
emp\_stim\_sum #emp法下目标规模对应的模拟总利润  
EMP\_emp\_last #emp法EMP利润  
emp\_percent #EMP法目标客户比例

```