ECON3105-Tree.R

kayliang

2021-12-03

#Tree model  
#1 data preparation  
setwd("/Users/kayliang/Desktop/R 语言学习")  
churn1 = read.csv("hotel\_bookings.csv")  
colnames(churn1)

## [1] "hotel" "is\_canceled"   
## [3] "lead\_time" "arrival\_date\_year"   
## [5] "arrival\_date\_month" "arrival\_date\_week\_number"   
## [7] "arrival\_date\_day\_of\_month" "stays\_in\_weekend\_nights"   
## [9] "stays\_in\_week\_nights" "adults"   
## [11] "children" "babies"   
## [13] "meal" "country"   
## [15] "market\_segment" "distribution\_channel"   
## [17] "is\_repeated\_guest" "previous\_cancellations"   
## [19] "previous\_bookings\_not\_canceled" "reserved\_room\_type"   
## [21] "assigned\_room\_type" "booking\_changes"   
## [23] "deposit\_type" "agent"   
## [25] "company" "days\_in\_waiting\_list"   
## [27] "customer\_type" "adr"   
## [29] "required\_car\_parking\_spaces" "total\_of\_special\_requests"   
## [31] "reservation\_status" "reservation\_status\_date"

nrow(churn1)

## [1] 119390

str(churn1)

## 'data.frame': 119390 obs. of 32 variables:  
## $ hotel : chr "Resort Hotel" "Resort Hotel" "Resort Hotel" "Resort Hotel" ...  
## $ is\_canceled : int 0 0 0 0 0 0 0 0 1 1 ...  
## $ lead\_time : int 342 737 7 13 14 14 0 9 85 75 ...  
## $ arrival\_date\_year : int 2015 2015 2015 2015 2015 2015 2015 2015 2015 2015 ...  
## $ arrival\_date\_month : chr "July" "July" "July" "July" ...  
## $ arrival\_date\_week\_number : int 27 27 27 27 27 27 27 27 27 27 ...  
## $ arrival\_date\_day\_of\_month : int 1 1 1 1 1 1 1 1 1 1 ...  
## $ stays\_in\_weekend\_nights : int 0 0 0 0 0 0 0 0 0 0 ...  
## $ stays\_in\_week\_nights : int 0 0 1 1 2 2 2 2 3 3 ...  
## $ adults : int 2 2 1 1 2 2 2 2 2 2 ...  
## $ children : int 0 0 0 0 0 0 0 0 0 0 ...  
## $ babies : int 0 0 0 0 0 0 0 0 0 0 ...  
## $ meal : chr "BB" "BB" "BB" "BB" ...  
## $ country : chr "PRT" "PRT" "GBR" "GBR" ...  
## $ market\_segment : chr "Direct" "Direct" "Direct" "Corporate" ...  
## $ distribution\_channel : chr "Direct" "Direct" "Direct" "Corporate" ...  
## $ is\_repeated\_guest : int 0 0 0 0 0 0 0 0 0 0 ...  
## $ previous\_cancellations : int 0 0 0 0 0 0 0 0 0 0 ...  
## $ previous\_bookings\_not\_canceled: int 0 0 0 0 0 0 0 0 0 0 ...  
## $ reserved\_room\_type : chr "C" "C" "A" "A" ...  
## $ assigned\_room\_type : chr "C" "C" "C" "A" ...  
## $ booking\_changes : int 3 4 0 0 0 0 0 0 0 0 ...  
## $ deposit\_type : chr "No Deposit" "No Deposit" "No Deposit" "No Deposit" ...  
## $ agent : chr "NULL" "NULL" "NULL" "304" ...  
## $ company : chr "NULL" "NULL" "NULL" "NULL" ...  
## $ days\_in\_waiting\_list : int 0 0 0 0 0 0 0 0 0 0 ...  
## $ customer\_type : chr "Transient" "Transient" "Transient" "Transient" ...  
## $ adr : num 0 0 75 75 98 ...  
## $ required\_car\_parking\_spaces : int 0 0 0 0 0 0 0 0 0 0 ...  
## $ total\_of\_special\_requests : int 0 0 0 0 1 1 0 1 1 0 ...  
## $ reservation\_status : chr "Check-Out" "Check-Out" "Check-Out" "Check-Out" ...  
## $ reservation\_status\_date : chr "2015-07-01" "2015-07-01" "2015-07-02" "2015-07-02" ...

library(ISLR)   
library(tidyverse)

## ── Attaching packages ─────────────────────────────────────── tidyverse 1.3.1 ──

## ✓ ggplot2 3.3.5 ✓ purrr 0.3.4  
## ✓ tibble 3.1.4 ✓ dplyr 1.0.7  
## ✓ tidyr 1.1.3 ✓ stringr 1.4.0  
## ✓ readr 2.0.1 ✓ forcats 0.5.1

## ── Conflicts ────────────────────────────────────────── tidyverse\_conflicts() ──  
## x dplyr::filter() masks stats::filter()  
## x dplyr::lag() masks stats::lag()

library(ggplot2)  
library(e1071)   
library(rpart)   
library(rpart.plot)   
library(randomForest)

## randomForest 4.6-14

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

##   
## Attaching package: 'randomForest'

## The following object is masked from 'package:dplyr':  
##   
## combine

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

library(pROC)

## Type 'citation("pROC")' for a citation.

##   
## Attaching package: 'pROC'

## The following objects are masked from 'package:stats':  
##   
## cov, smooth, var

library(caret)

## Loading required package: lattice

##   
## Attaching package: 'caret'

## The following object is masked from 'package:purrr':  
##   
## lift

churn1<-churn1%>%  
 mutate(  
 hotel=as.factor(hotel),   
 is\_canceled=as.factor(is\_canceled),  
 meal=as.factor(meal),  
 country=as.factor(country),  
 market\_segment=as.factor(market\_segment),  
 distribution\_channel=as.factor(distribution\_channel),  
 is\_repeated\_guest=as.factor(is\_repeated\_guest),  
 reserved\_room\_type=as.factor(reserved\_room\_type),  
 assigned\_room\_type=as.factor(assigned\_room\_type),  
 deposit\_type=as.factor(deposit\_type),  
 customer\_type=as.factor(customer\_type),  
 reservation\_status=as.factor(reservation\_status),  
 agent=as.factor(agent),  
 company=as.factor(company),  
 arrival\_date\_day\_of\_month=as.factor(arrival\_date\_day\_of\_month),  
 arrival\_date\_month=as.factor(arrival\_date\_month),  
 arrival\_date\_year=as.factor(arrival\_date\_year)  
   
 )  
# Split the dataset into 70% training and 30% test data  
set.seed(1)  
a = seq(1,nrow(churn1),by=1)   
i = sample(a,nrow(churn1)\*0.7, replace=FALSE)   
training = churn1[i,]   
testing = churn1[-i,]   
nrow(training)

## [1] 83573

nrow(testing)

## [1] 35817

dim(training)

## [1] 83573 32

training\_1 <- training[c('hotel','is\_canceled','lead\_time','adults','children','babies','meal',  
 'market\_segment','distribution\_channel','is\_repeated\_guest',  
 'previous\_cancellations','previous\_bookings\_not\_canceled','reserved\_room\_type',  
 'deposit\_type','days\_in\_waiting\_list','customer\_type','adr',  
 'required\_car\_parking\_spaces')]  
training\_2 <- testing[c('hotel','is\_canceled','lead\_time','adults','children','babies','meal',  
 'market\_segment','distribution\_channel','is\_repeated\_guest',  
 'previous\_cancellations','previous\_bookings\_not\_canceled','reserved\_room\_type',  
 'deposit\_type','days\_in\_waiting\_list','customer\_type','adr',  
 'required\_car\_parking\_spaces')]  
  
library(C50)   
att1 <- c("hotel","lead\_time",   
 "adults", "children", "babies",   
 "meal", "market\_segment", "distribution\_channel",   
 "is\_repeated\_guest", "previous\_cancellations", "previous\_bookings\_not\_canceled",  
 "reserved\_room\_type", "deposit\_type", "days\_in\_waiting\_list",   
 "customer\_type", "adr", "required\_car\_parking\_spaces")   
colnames(training\_1)

## [1] "hotel" "is\_canceled"   
## [3] "lead\_time" "adults"   
## [5] "children" "babies"   
## [7] "meal" "market\_segment"   
## [9] "distribution\_channel" "is\_repeated\_guest"   
## [11] "previous\_cancellations" "previous\_bookings\_not\_canceled"  
## [13] "reserved\_room\_type" "deposit\_type"   
## [15] "days\_in\_waiting\_list" "customer\_type"   
## [17] "adr" "required\_car\_parking\_spaces"

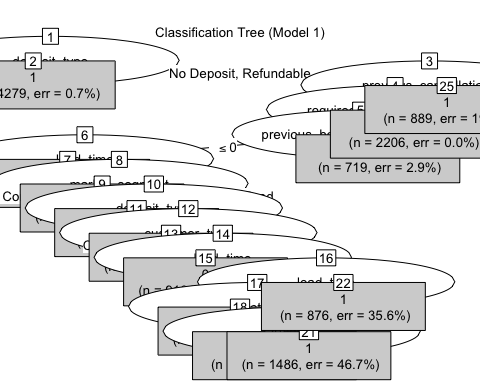
# 2 full-tree model(500)   
mod1 = C5.0(x=training\_2[,att1], y=training\_2$is\_canceled,   
 control=C5.0Control(minCases=500))   
# Plot  
library(partykit)

## Loading required package: grid

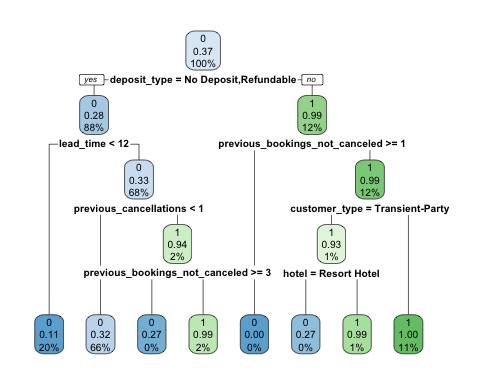
## Loading required package: libcoin

## Loading required package: mvtnorm

plot(mod1,   
 main="Classification Tree (Model 1)",   
 type="simple",  
 gp = gpar(fontsize=10))



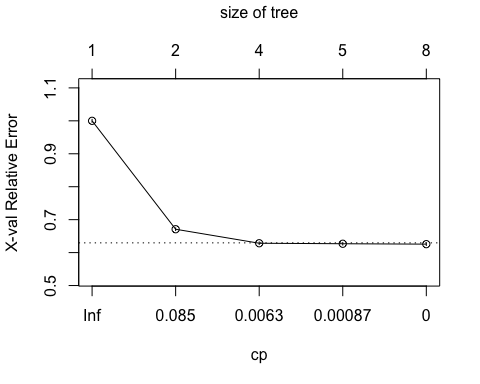
testing$Yhat\_tree = predict(mod1, newdata = training\_2[,att1], type="prob")[,2]  
attach(testing)  
class\_tree <- function(x){ifelse(Yhat\_tree > x, 1, 0)}  
predicted\_tree = class\_tree(0.5)  
  
#3 Prune tree use "rpart"  
set.seed(1)   
full\_tree<-rpart(is\_canceled~.,  
 data=training\_1,   
 method="class",  
 control=rpart.control(cp=0, maxdepth = 4))  
  
rpart.plot(full\_tree)



printcp(full\_tree)

##   
## Classification tree:  
## rpart(formula = is\_canceled ~ ., data = training\_1, method = "class",   
## control = rpart.control(cp = 0, maxdepth = 4))  
##   
## Variables actually used in tree construction:  
## [1] customer\_type deposit\_type   
## [3] hotel lead\_time   
## [5] previous\_bookings\_not\_canceled previous\_cancellations   
##   
## Root node error: 30946/83573 = 0.37029  
##   
## n= 83573   
##   
## CP nsplit rel error xerror xstd  
## 1 0.32908938 0 1.00000 1.00000 0.0045110  
## 2 0.02187682 1 0.67091 0.67091 0.0040366  
## 3 0.00180960 3 0.62716 0.62838 0.0039473  
## 4 0.00042009 4 0.62535 0.62687 0.0039440  
## 5 0.00000000 7 0.62409 0.62561 0.0039412

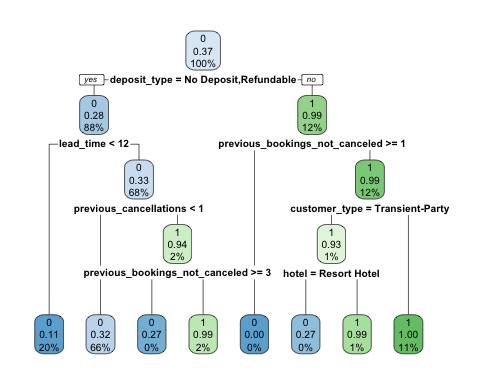
plotcp(full\_tree)



min\_xerror<-full\_tree$cptable[which.min(full\_tree$cptable[,"xerror"]),]  
min\_xerror

## CP nsplit rel error xerror xstd   
## 0.000000000 7.000000000 0.624087119 0.625605894 0.003941183

#4 prune tree with minimum cp value  
min\_xerror\_tree<-prune(full\_tree, cp=min\_xerror[1])  
rpart.plot(min\_xerror\_tree)



bp\_tree<-min\_xerror\_tree  
testing$ct\_bp\_pred\_prob<-predict(bp\_tree,testing)[,2]  
testing$ct\_bp\_pred\_class=ifelse(testing$ct\_bp\_pred\_prob>0.5,"Yes","No")  
  
table(testing$ct\_bp\_pred\_class,testing$is\_canceled, dnn=c("predicted","actual"))

## actual  
## predicted 0 1  
## No 22527 8394  
## Yes 12 4884

(22486+4964)/nrow(testing)

## [1] 0.7663958

#Loading Logistic part（The same as above)  
# 1: Load the data  
library(foreign)  
library(dplyr)  
library(ROCR) # For evaluation metrics  
library(caret) # For confusion matrix  
library(ggplot2)  
library(lattice)  
hotel\_bookings = read.csv("hotel\_bookings.csv")  
colnames(hotel\_bookings)

## [1] "hotel" "is\_canceled"   
## [3] "lead\_time" "arrival\_date\_year"   
## [5] "arrival\_date\_month" "arrival\_date\_week\_number"   
## [7] "arrival\_date\_day\_of\_month" "stays\_in\_weekend\_nights"   
## [9] "stays\_in\_week\_nights" "adults"   
## [11] "children" "babies"   
## [13] "meal" "country"   
## [15] "market\_segment" "distribution\_channel"   
## [17] "is\_repeated\_guest" "previous\_cancellations"   
## [19] "previous\_bookings\_not\_canceled" "reserved\_room\_type"   
## [21] "assigned\_room\_type" "booking\_changes"   
## [23] "deposit\_type" "agent"   
## [25] "company" "days\_in\_waiting\_list"   
## [27] "customer\_type" "adr"   
## [29] "required\_car\_parking\_spaces" "total\_of\_special\_requests"   
## [31] "reservation\_status" "reservation\_status\_date"

str(hotel\_bookings)

## 'data.frame': 119390 obs. of 32 variables:  
## $ hotel : chr "Resort Hotel" "Resort Hotel" "Resort Hotel" "Resort Hotel" ...  
## $ is\_canceled : int 0 0 0 0 0 0 0 0 1 1 ...  
## $ lead\_time : int 342 737 7 13 14 14 0 9 85 75 ...  
## $ arrival\_date\_year : int 2015 2015 2015 2015 2015 2015 2015 2015 2015 2015 ...  
## $ arrival\_date\_month : chr "July" "July" "July" "July" ...  
## $ arrival\_date\_week\_number : int 27 27 27 27 27 27 27 27 27 27 ...  
## $ arrival\_date\_day\_of\_month : int 1 1 1 1 1 1 1 1 1 1 ...  
## $ stays\_in\_weekend\_nights : int 0 0 0 0 0 0 0 0 0 0 ...  
## $ stays\_in\_week\_nights : int 0 0 1 1 2 2 2 2 3 3 ...  
## $ adults : int 2 2 1 1 2 2 2 2 2 2 ...  
## $ children : int 0 0 0 0 0 0 0 0 0 0 ...  
## $ babies : int 0 0 0 0 0 0 0 0 0 0 ...  
## $ meal : chr "BB" "BB" "BB" "BB" ...  
## $ country : chr "PRT" "PRT" "GBR" "GBR" ...  
## $ market\_segment : chr "Direct" "Direct" "Direct" "Corporate" ...  
## $ distribution\_channel : chr "Direct" "Direct" "Direct" "Corporate" ...  
## $ is\_repeated\_guest : int 0 0 0 0 0 0 0 0 0 0 ...  
## $ previous\_cancellations : int 0 0 0 0 0 0 0 0 0 0 ...  
## $ previous\_bookings\_not\_canceled: int 0 0 0 0 0 0 0 0 0 0 ...  
## $ reserved\_room\_type : chr "C" "C" "A" "A" ...  
## $ assigned\_room\_type : chr "C" "C" "C" "A" ...  
## $ booking\_changes : int 3 4 0 0 0 0 0 0 0 0 ...  
## $ deposit\_type : chr "No Deposit" "No Deposit" "No Deposit" "No Deposit" ...  
## $ agent : chr "NULL" "NULL" "NULL" "304" ...  
## $ company : chr "NULL" "NULL" "NULL" "NULL" ...  
## $ days\_in\_waiting\_list : int 0 0 0 0 0 0 0 0 0 0 ...  
## $ customer\_type : chr "Transient" "Transient" "Transient" "Transient" ...  
## $ adr : num 0 0 75 75 98 ...  
## $ required\_car\_parking\_spaces : int 0 0 0 0 0 0 0 0 0 0 ...  
## $ total\_of\_special\_requests : int 0 0 0 0 1 1 0 1 1 0 ...  
## $ reservation\_status : chr "Check-Out" "Check-Out" "Check-Out" "Check-Out" ...  
## $ reservation\_status\_date : chr "2015-07-01" "2015-07-01" "2015-07-02" "2015-07-02" ...

unique(hotel\_bookings$is\_canceled)

## [1] 0 1

# 2: Data pre-processing  
hotel\_bookings<-hotel\_bookings%>%  
 mutate(  
 hotel=as.factor(hotel),   
 is\_canceled=as.factor(is\_canceled),  
 meal=as.factor(meal),  
 country=as.factor(country),  
 market\_segment=as.factor(market\_segment),  
 distribution\_channel=as.factor(distribution\_channel),  
 is\_repeated\_guest=as.factor(is\_repeated\_guest),  
 reserved\_room\_type=as.factor(reserved\_room\_type),  
 assigned\_room\_type=as.factor(assigned\_room\_type),  
 deposit\_type=as.factor(deposit\_type),  
 customer\_type=as.factor(customer\_type),  
 reservation\_status=as.factor(reservation\_status),  
 agent=as.factor(agent),  
 company=as.factor(company),  
 arrival\_date\_day\_of\_month=as.factor(arrival\_date\_day\_of\_month),  
 arrival\_date\_month=as.factor(arrival\_date\_month),  
 arrival\_date\_year=as.factor(arrival\_date\_year)  
   
 )  
data = transform(hotel\_bookings, is\_canceled = ifelse(is\_canceled == '1', 1, 0))  
data = sapply(data, as.numeric)

## Warning in lapply(X = X, FUN = FUN, ...): NAs introduced by coercion

data = as.data.frame(data)   
data<-subset(data,data$adr<4000)  
# 3: Create the training and test data   
set.seed(1)  
a = seq(1,nrow(data),1)   
ind = sample(a, floor(nrow(data)\*0.7), replace=FALSE)   
train = data[ind,]  
test = data[-ind,]  
# 4: Train the logistic regression model using glm() function  
## (4a) a Logistic regression Estimation  
m1 = glm(is\_canceled~hotel+lead\_time+adults+children+babies+meal+market\_segment+distribution\_channel+  
 is\_repeated\_guest+previous\_cancellations+previous\_bookings\_not\_canceled+reserved\_room\_type+  
 deposit\_type+days\_in\_waiting\_list+customer\_type+adr+required\_car\_parking\_spaces,   
 data=train,  
 family=binomial(link ='logit')  
)

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

summary(m1)#There might be some outliers in the dataset but I don't know how to get rid of them...

##   
## Call:  
## glm(formula = is\_canceled ~ hotel + lead\_time + adults + children +   
## babies + meal + market\_segment + distribution\_channel + is\_repeated\_guest +   
## previous\_cancellations + previous\_bookings\_not\_canceled +   
## reserved\_room\_type + deposit\_type + days\_in\_waiting\_list +   
## customer\_type + adr + required\_car\_parking\_spaces, family = binomial(link = "logit"),   
## data = train)  
##   
## Deviance Residuals:   
## Min 1Q Median 3Q Max   
## -7.5509 -0.8404 -0.5526 0.3215 5.9163   
##   
## Coefficients:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) -7.534e+00 1.621e-01 -46.471 < 2e-16 \*\*\*  
## hotel -5.578e-02 1.990e-02 -2.803 0.005057 \*\*   
## lead\_time 2.717e-03 9.481e-05 28.652 < 2e-16 \*\*\*  
## adults 8.642e-02 1.662e-02 5.200 1.99e-07 \*\*\*  
## children 1.229e-01 2.166e-02 5.676 1.38e-08 \*\*\*  
## babies -3.854e-01 1.066e-01 -3.616 0.000299 \*\*\*  
## meal 2.137e-02 7.857e-03 2.719 0.006543 \*\*   
## market\_segment 4.175e-01 1.405e-02 29.714 < 2e-16 \*\*\*  
## distribution\_channel -2.224e-01 1.923e-02 -11.562 < 2e-16 \*\*\*  
## is\_repeated\_guest -5.587e-01 9.198e-02 -6.074 1.25e-09 \*\*\*  
## previous\_cancellations 2.732e+00 6.898e-02 39.605 < 2e-16 \*\*\*  
## previous\_bookings\_not\_canceled -5.386e-01 3.041e-02 -17.712 < 2e-16 \*\*\*  
## reserved\_room\_type 2.020e-02 6.162e-03 3.278 0.001046 \*\*   
## deposit\_type 4.733e+00 7.299e-02 64.845 < 2e-16 \*\*\*  
## days\_in\_waiting\_list -5.853e-04 5.516e-04 -1.061 0.288685   
## customer\_type -3.260e-02 1.603e-02 -2.033 0.042051 \*   
## adr 3.645e-03 2.166e-04 16.827 < 2e-16 \*\*\*  
## required\_car\_parking\_spaces -4.364e+01 8.250e+02 -0.053 0.957814   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## (Dispersion parameter for binomial family taken to be 1)  
##   
## Null deviance: 110174 on 83567 degrees of freedom  
## Residual deviance: 79129 on 83550 degrees of freedom  
## (4 observations deleted due to missingness)  
## AIC: 79165  
##   
## Number of Fisher Scoring iterations: 11

help(glm)  
## (4b) Check the predicted probability on train   
fit = m1$fitted.values  
m1.fit = predict(m1, data=train, type="link")   
head(train$is\_canceled)

## [1] 0 1 0 1 1 0

## (4c) Check predicted log-odds on training data  
m1.logodd = predict(m1, data=train, type="link")   
## accuracy on traning data  
m1.class = as.numeric(m1.logodd>0)   
m1.acc = mean(as.numeric(train$is\_canceled == m1.class))#warning message might be due to 4 outliers as the difference between two datasets is 4

## Warning in train$is\_canceled == m1.class: longer object length is not a multiple  
## of shorter object length

# 5: Prediction on test data   
# (5a) predict the log-odds ratio,calculate the accuracy rate  
m1.logodd\_1 = predict(m1, newdata=test, type="link")   
m1.class\_1 = as.numeric(m1.logodd\_1>0)   
m1.correct\_1 = sum(as.numeric(test$is\_canceled == m1.class\_1))   
m1.acc\_1 = mean(as.numeric(test$is\_canceled == m1.class\_1))   
  
# plot ROC graphs between Tree and Logit  
class\_log = function(x){ifelse(m1.logodd\_1 > x, 1, 0)}  
predicted\_log = class\_log(0.5)  
  
predict\_tree <- prediction(Yhat\_tree, is\_canceled)  
predict\_log <- prediction(m1.logodd\_1, is\_canceled)   
ROC\_tree <- performance(predict\_tree, "tpr", "fpr")  
ROC\_log <- performance(predict\_log, "tpr", "fpr")   
plot(ROC\_log, col= "deeppink",lwd=2)  
plot(ROC\_tree, add = TRUE, col= "blueviolet",lwd=2)  
abline(0,1, col = "black")  
title("ROC curves")  
legend(0.7, 0.5 ,c("Logistic", "TREE"),   
 lty = c(1,1,1),   
 lwd = c(2,2,2),  
 col = c("deeppink", "blueviolet"),  
 ncol=1, cex=0.9, y.intersp=1.2)

