CART\_IIN\_R.R

shiwam

2022-01-31

list.files()

## [1] "~$RT\_IIN\_R.docx" "~$runing.docx"   
## [3] "Banking\_prediction.Rproj" "BEST\_PREDICTED\_Y.csv"   
## [5] "CART\_IIN\_R.R" "CART\_IIN\_R.spin.R"   
## [7] "CART\_IIN\_R.spin.Rmd" "CART\_IIN\_R.tex"   
## [9] "new\_test.csv" "new\_train.csv"   
## [11] "pruning.docx" "pruning.R"

df=read.csv("new\_train.csv");head(df)

## age job marital education default housing loan contact  
## 1 49 blue-collar married basic.9y unknown no no cellular  
## 2 37 entrepreneur married university.degree no no no telephone  
## 3 78 retired married basic.4y no no no cellular  
## 4 36 admin. married university.degree no yes no telephone  
## 5 59 retired divorced university.degree no no no cellular  
## 6 29 admin. single university.degree no no no cellular  
## month day\_of\_week duration campaign pdays previous poutcome y  
## 1 nov wed 227 4 999 0 nonexistent no  
## 2 nov wed 202 2 999 1 failure no  
## 3 jul mon 1148 1 999 0 nonexistent yes  
## 4 may mon 120 2 999 0 nonexistent no  
## 5 jun tue 368 2 999 0 nonexistent no  
## 6 aug wed 256 2 999 0 nonexistent no

df[,c(2,3,4,5,6,7,8,9,10,15,16)]=lapply(df[,c(2,3,4,5,6,7,8,9,10,15,16)], FUN = as.factor)  
str(df)

## 'data.frame': 32950 obs. of 16 variables:  
## $ age : int 49 37 78 36 59 29 26 30 50 33 ...  
## $ job : Factor w/ 12 levels "admin.","blue-collar",..: 2 3 6 1 6 1 9 2 2 1 ...  
## $ marital : Factor w/ 4 levels "divorced","married",..: 2 2 2 2 1 3 3 2 2 3 ...  
## $ education : Factor w/ 8 levels "basic.4y","basic.6y",..: 3 7 1 7 7 7 3 1 1 4 ...  
## $ default : Factor w/ 3 levels "no","unknown",..: 2 1 1 1 1 1 1 1 2 1 ...  
## $ housing : Factor w/ 3 levels "no","unknown",..: 1 1 1 3 1 1 1 3 1 3 ...  
## $ loan : Factor w/ 3 levels "no","unknown",..: 1 1 1 1 1 1 1 1 1 1 ...  
## $ contact : Factor w/ 2 levels "cellular","telephone": 1 2 1 2 1 1 2 1 2 1 ...  
## $ month : Factor w/ 10 levels "apr","aug","dec",..: 8 8 4 7 5 2 2 8 7 4 ...  
## $ day\_of\_week: Factor w/ 5 levels "fri","mon","thu",..: 5 5 2 2 4 5 5 5 1 4 ...  
## $ duration : int 227 202 1148 120 368 256 449 126 574 498 ...  
## $ campaign : int 4 2 1 2 2 2 1 2 1 5 ...  
## $ pdays : int 999 999 999 999 999 999 999 999 999 999 ...  
## $ previous : int 0 1 0 0 0 0 0 0 0 0 ...  
## $ poutcome : Factor w/ 3 levels "failure","nonexistent",..: 2 1 2 2 2 2 2 2 2 2 ...  
## $ y : Factor w/ 2 levels "no","yes": 1 1 2 1 1 1 2 1 1 1 ...

levels(df$job)=0:(length(levels(df$job))-1)  
levels(df$marital)=0:(length(levels(df$marital))-1)  
levels(df$contact)=0:(length(levels(df$contact))-1)  
levels(df$education)=0:(length(levels(df$education))-1)  
levels(df$default)=0:(length(levels(df$default))-1)  
levels(df$housing)=0:(length(levels(df$housing))-1)  
  
levels(df$loan)=0:(length(levels(df$loan))-1)  
levels(df$month)=0:(length(levels(df$month))-1)  
levels(df$day\_of\_week)=0:(length(levels(df$day\_of\_week))-1)  
levels(df$poutcome)=0:(length(levels(df$poutcome))-1)  
levels(df$y)=0:(length(levels(df$y))-1)  
df[,c(2,3,4,5,6,7,8,9,10,15,16)]=lapply(df[,c(2,3,4,5,6,7,8,9,10,15,16)], FUN = as.character)  
df[,c(2,3,4,5,6,7,8,9,10,15,16)]=lapply(df[,c(2,3,4,5,6,7,8,9,10,15,16)], FUN = as.numeric)  
head(df)#NOW ALL ARE IN NUMERIC VARIABLE

## age job marital education default housing loan contact month day\_of\_week  
## 1 49 1 1 2 1 0 0 0 7 4  
## 2 37 2 1 6 0 0 0 1 7 4  
## 3 78 5 1 0 0 0 0 0 3 1  
## 4 36 0 1 6 0 2 0 1 6 1  
## 5 59 5 0 6 0 0 0 0 4 3  
## 6 29 0 2 6 0 0 0 0 1 4  
## duration campaign pdays previous poutcome y  
## 1 227 4 999 0 1 0  
## 2 202 2 999 1 0 0  
## 3 1148 1 999 0 1 1  
## 4 120 2 999 0 1 0  
## 5 368 2 999 0 1 0  
## 6 256 2 999 0 1 0

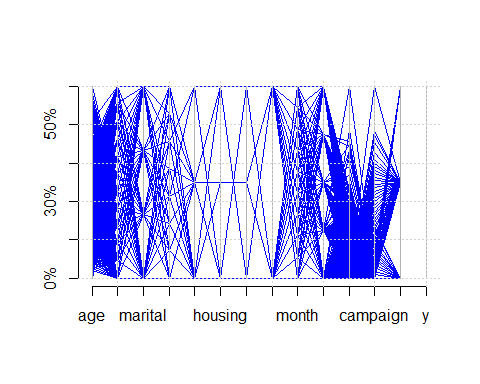
df1=df[,-c(13,14)]  
dftest=read.csv("new\_test.csv");head(dftest)

## age job marital education default housing loan contact month day\_of\_week  
## 1 32 4 0 6 0 0 0 0 3 3  
## 2 37 10 3 6 0 0 0 0 4 3  
## 3 55 5 0 5 1 2 0 0 3 2  
## 4 44 2 1 0 1 0 0 1 4 3  
## 5 28 0 2 3 0 0 0 0 5 0  
## 6 45 10 1 2 0 0 0 0 1 0  
## duration campaign poutcome  
## 1 131 5 1  
## 2 100 1 1  
## 3 131 2 1  
## 4 48 2 1  
## 5 144 2 1  
## 6 126 3 1

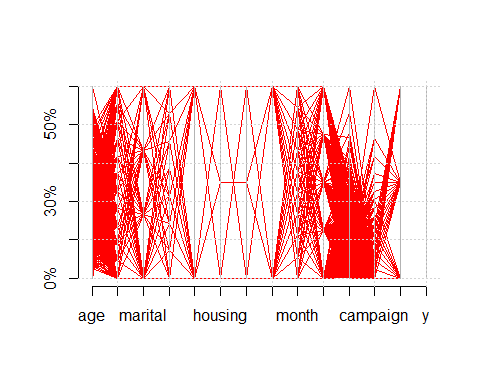
# VISUALIZING THIS DATASET USING MULTIVARIATE PLOTS  
#######PARALLEL COORDINATE PLOTS#############  
palette("default")  
library(MASS)

## Warning: package 'MASS' was built under R version 4.0.5

parcoord(df1[which(df1$y=="0"),],col = "blue")  
axis(2,at=axTicks(2),labels = c("0%","20%","30%","40%","50%","60%"))  
grid()



parcoord(df1[which(df1$y=="1"),],col = "red")  
axis(2,at=axTicks(2),labels = c("0%","20%","30%","40%","50%","60%"))  
grid()



##APPLYING CART MODEL##  
##USING LIBRARY(RPART)  
library(rpart)

## Warning: package 'rpart' was built under R version 4.0.5

library(rpart.plot)

## Warning: package 'rpart.plot' was built under R version 4.0.5

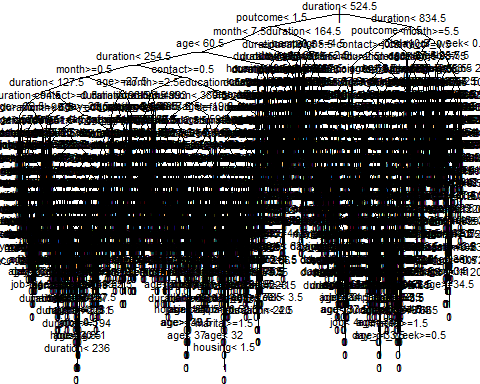
mod=rpart(df1$y~.,method = "class",data = df1,  
 control = rpart.control(cp=0,minsplit = 2,minbucket = 1,  
 maxcompete = 0,maxsurrogate = 0,xval = 0,parms=list(split="gini")))  
  
##node number is also include in it if we add nn and nn.cex in prp command  
#prp(mod,type = 1,extra = 1,under = T,varlen = 0,cex=0.7,compress = T,Margin = 0,digits = 0,split.cex = 0.8,under.cex = 0.8,nn=T,nn.cex = 0.6)  
  
  
  
####NOW PARTITIONG THE TRAINING DATASET INTO VALIDATION PART BECAUSE THE GIVEN DATASET IS MUCH LARGER  
partidx=sample(1:nrow(df1),26000,replace = F)#20% of train dataset convert to validation dataset  
df1train=df1[partidx,];head(df1train)#training

## age job marital education default housing loan contact month day\_of\_week  
## 20510 53 4 1 6 1 0 0 0 3 4  
## 28483 38 0 2 2 0 0 0 1 4 2  
## 11246 45 0 1 7 0 2 0 1 4 1  
## 21763 38 0 1 6 0 0 0 0 8 4  
## 31179 45 1 1 2 0 0 0 1 4 4  
## 14927 47 6 1 0 1 1 1 1 4 3  
## duration campaign poutcome y  
## 20510 184 2 1 0  
## 28483 406 1 1 0  
## 11246 1068 1 1 1  
## 21763 180 2 0 0  
## 31179 119 1 1 0  
## 14927 72 8 1 0

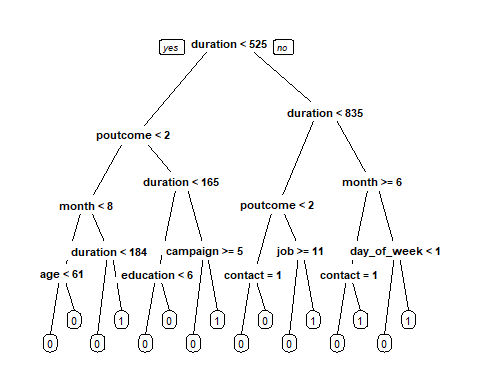
partidx1=sample(1:nrow(df1[-partidx]),6950,replace=F)  
#intersect(partidx,partidx1)  
df1valid=df1[partidx1,]  
head(df1valid)#VALIDATION

## age job marital education default housing loan contact month day\_of\_week  
## 24890 56 9 1 5 0 0 0 0 3 0  
## 3910 24 8 2 7 0 2 0 0 8 0  
## 19542 55 7 2 2 0 1 1 0 6 2  
## 12774 67 0 1 0 1 0 0 0 8 2  
## 7697 31 1 1 2 0 2 0 1 4 1  
## 24628 31 0 1 6 0 0 0 0 1 3  
## duration campaign poutcome y  
## 24890 196 2 1 0  
## 3910 460 1 1 1  
## 19542 717 1 1 1  
## 12774 299 2 2 1  
## 7697 198 2 1 0  
## 24628 164 1 1 1

#BUILD MODEL ON TRAINING PARTITION  
mod1=rpart(y~.,method = "class",data=df1train,control = rpart.control(  
 cp=0,minsplit = 2,minbucket = 1,maxcompete = 0,maxsurrogate = 0,xval = 0,  
 parms=list(split="gini")))  
par(mar=c(0,0,0,0),oma=c(0,0,0,0),xpd=NA)  
plot(mod1,uniform = T,branch = 0.1,compress = T,margin = 0,nspace = 1)  
text(mod1,splits = T,use.n = F,all=F,minlength = 0,cex=0.7)#here all the text shon in full grown tree



#####NICER VERSION OF THIS FULL GROWN TREE###########  
#prp(mod1,varlen = 0,cex=0.7,extra = 0,compress = T,Margin = 0,digits = 0,nn=T,nn.cex=0.6)  
#FIRST FOUR LEVEL OF FULL GROWN TREE  
toss1=as.integer(row.names(mod1$frame))  
toss2=sort(toss1)  
toss3=toss2[which(toss2==16):length(toss2)]  
mod1sub=snip.rpart(mod1,toss=toss3)  
prp(mod1sub,varlen = 0,cex=0.7,extra=0,compress = T,Margin = 0,digits = 0)#ALL THE NODES SHOWS CLEARLY EXCEPT 16 LAST NODE



#DESCRIPTION OF EACH SPLITTING STEP OF THE FULL GROWN TREE  
  
#TOTAL NUMBER OF NODES  
nrow(mod1$splits)

## [1] 2402

#NUMBER OF TOTAL NODES  
nrow(mod1$frame)

## [1] 4805

#NUMBER OF TERMINAL NODES  
nrow(mod1$frame)-nrow(mod1$splits)# WHICH IS ONE MORE THAN DECISION NODES

## [1] 2403

#THIS IS THE PROPERTY OF BINARY TREES, THE NUMBER OF TERMINAL NODES OR NUMBER OF   
#LEAFS ARE ONE MORE THAN THE DECISION NODES  
################################################################################  
  
#IF WE ARE INTERESTED IN HAVING A TABLE WHERE WE HAVE THE INFORMATION ABOUT THE VARIABLES  
#WHICH HAVE BEEN USED FOR SPLITTING AND THE SPLIT VALUE, THE PREDICTOR VALUE COMBINATION THEN  
#THIS IS DONE AS FOLLOWS:-  
  
  
#j is the counter of split variables  
#i is the counter of split values  
splitvalue=NULL  
j=1  
i=1  
for (x in mod1$frame$var) {  
 if(as.character(x)!="<leaf>"){  
 if (!is.factor(df1[,as.character(x)])) {  
 splitvalue[i]=mod1$splits[j,"index"]  
 }  
 else{  
 cl=NULL  
 #split variable is factor  
 #k={1,largest number of levels in the factor}  
   
 for (k in 1:ncol(mod1$csplit)) {  
 temp=mod1$csplit[mod1$splits[j,"index"],k]  
 #if level(temp)goes to the left child  
 if (temp==1) {  
 cl=paste(cl,levels(df1[,as.character(x)])[k],sep = ",")  
 }  
 }  
 splitvalue[i]=substr(cl,start = 2,stop = nchar(cl))  
 }  
 j=j+1  
 }  
 else{  
 splitvalue[i]=NA  
 }  
 i=i+1  
}  
dat=data.frame("NODE\_NUMBER"=row.names(mod1$frame),  
 "SPLIT\_VAR"=mod1$frame$var,  
 "SPLIT\_VALUE"=splitvalue,  
 "CASES"=mod1$frame$n,  
 "CLASS"=mod1$frame$yval-1,check.names = F)  
head(dat,20)

## NODE\_NUMBER SPLIT\_VAR SPLIT\_VALUE CASES CLASS  
## 1 1 duration 524.5 26000 0  
## 2 2 poutcome 1.5 23152 0  
## 3 4 month 7.5 22404 0  
## 4 8 age 60.5 21846 0  
## 5 16 duration 254.5 21519 0  
## 6 32 month 0.5 16143 0  
## 7 64 duration 127.5 15260 0  
## 8 128 duration 94.5 7982 0  
## 9 256 age 32.5 5269 0  
## 10 512 age 57.5 3875 0  
## 11 1024 loan 1.5 3693 0  
## 12 2048 <leaf> NA 3100 0  
## 13 2049 job 8.5 593 0  
## 14 4098 <leaf> NA 468 0  
## 15 4099 age 38.5 125 0  
## 16 8198 <leaf> NA 80 0  
## 17 8199 age 37.5 45 0  
## 18 16398 <leaf> NA 37 0  
## 19 16399 campaign 2.5 8 0  
## 20 32798 <leaf> NA 6 0

#Prediction For Training Dataset  
mod1train=predict(mod1,df1train,type = "class")  
table("ACTUAL\_VALUE"=df1train$y,"PREDICTED\_VALUE"=mod1train)

## PREDICTED\_VALUE  
## ACTUAL\_VALUE 0 1  
## 0 23081 0  
## 1 0 2919

#CLASSIFICATION ACCURACY  
mean(mod1train==df1train$y)

## [1] 1

#misclassification error  
mean(mod1train!=df1train$y)

## [1] 0

#PREDICTION FOR VALIDATION DATASET  
mod1valid=predict(mod1,df1valid,type = "class")  
table("ACTUAL\_VALUE"=df1valid$y,"PREDICTED\_VALUE"=mod1valid)#some error are seen here

## PREDICTED\_VALUE  
## ACTUAL\_VALUE 0 1  
## 0 6062 101  
## 1 92 695

#CLASSIFICATION ACCURACY  
mean(mod1valid==df1valid$y)

## [1] 0.9722302

#misclassification error  
mean(mod1valid!=df1valid$y)

## [1] 0.02776978

######################################3  
#PREDICTION TO TESTING DATASET  
predicted\_y=predict(mod1,dftest,type = "class")  
df1test=cbind(dftest,predicted\_y);head(df1test)

## age job marital education default housing loan contact month day\_of\_week  
## 1 32 4 0 6 0 0 0 0 3 3  
## 2 37 10 3 6 0 0 0 0 4 3  
## 3 55 5 0 5 1 2 0 0 3 2  
## 4 44 2 1 0 1 0 0 1 4 3  
## 5 28 0 2 3 0 0 0 0 5 0  
## 6 45 10 1 2 0 0 0 0 1 0  
## duration campaign poutcome predicted\_y  
## 1 131 5 1 0  
## 2 100 1 1 0  
## 3 131 2 1 0  
## 4 48 2 1 0  
## 5 144 2 1 0  
## 6 126 3 1 0

#######################################  
  
#ABOVE MODEL IS OVERFITTING CASE BECAUSE ACCURACY OF TRAING DATASET IS 1 AND ERROR IS 0.  
#SO IT WILL FIT THE NOISE OF THE DATA