capston\_project.R

ashlyabraham

2020-12-04

#------------------------------------------------------- Bank Marketing dataset Analysis ---------------------------------------------  
  
library(tidyverse) # manipulation

## ── Attaching packages ─────────────────────────────────────── tidyverse 1.3.0 ──

## ✓ ggplot2 3.3.2 ✓ purrr 0.3.4  
## ✓ tibble 3.0.4 ✓ dplyr 1.0.2  
## ✓ tidyr 1.1.2 ✓ stringr 1.4.0  
## ✓ readr 1.4.0 ✓ forcats 0.5.0

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

library(ggplot2) # Visualzation   
library("corrplot") # correlation

## corrplot 0.84 loaded

library(gridExtra) # grid view of the plot

##   
## Attaching package: 'gridExtra'

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

library('DMwR') # SMOTE the data - balancing

## Loading required package: lattice

## Loading required package: grid

## Registered S3 method overwritten by 'quantmod':  
## method from  
## as.zoo.data.frame zoo

library(caret) # confusionMatrix

##   
## Attaching package: 'caret'

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

library(rpart) #decision tree   
library(rpart.plot) # decision tree plot   
library(Boruta) # features selection   
library(e1071) #naive bayes  
library(ROCR) #Prediction   
  
#importing the data  
bank\_additional <- read\_delim("/Users/ashlyabraham/Downloads/bank-additional/bank-additional.csv",   
 ";", escape\_double = FALSE, trim\_ws = TRUE)

##   
## ── Column specification ────────────────────────────────────────────────────────  
## cols(  
## .default = col\_character(),  
## age = col\_double(),  
## duration = col\_double(),  
## campaign = col\_double(),  
## pdays = col\_double(),  
## previous = col\_double(),  
## emp.var.rate = col\_double(),  
## cons.price.idx = col\_double(),  
## cons.conf.idx = col\_double(),  
## euribor3m = col\_double(),  
## nr.employed = col\_double()  
## )  
## ℹ Use `spec()` for the full column specifications.

data<- bank\_additional  
dim(data) # dimension of the data set

## [1] 4119 21

str(data) #structure of data

## tibble [4,119 × 21] (S3: spec\_tbl\_df/tbl\_df/tbl/data.frame)  
## $ age : num [1:4119] 30 39 25 38 47 32 32 41 31 35 ...  
## $ job : chr [1:4119] "blue-collar" "services" "services" "services" ...  
## $ marital : chr [1:4119] "married" "single" "married" "married" ...  
## $ education : chr [1:4119] "basic.9y" "high.school" "high.school" "basic.9y" ...  
## $ default : chr [1:4119] "no" "no" "no" "no" ...  
## $ housing : chr [1:4119] "yes" "no" "yes" "unknown" ...  
## $ loan : chr [1:4119] "no" "no" "no" "unknown" ...  
## $ contact : chr [1:4119] "cellular" "telephone" "telephone" "telephone" ...  
## $ month : chr [1:4119] "may" "may" "jun" "jun" ...  
## $ day\_of\_week : chr [1:4119] "fri" "fri" "wed" "fri" ...  
## $ duration : num [1:4119] 487 346 227 17 58 128 290 44 68 170 ...  
## $ campaign : num [1:4119] 2 4 1 3 1 3 4 2 1 1 ...  
## $ pdays : num [1:4119] 999 999 999 999 999 999 999 999 999 999 ...  
## $ previous : num [1:4119] 0 0 0 0 0 2 0 0 1 0 ...  
## $ poutcome : chr [1:4119] "nonexistent" "nonexistent" "nonexistent" "nonexistent" ...  
## $ emp.var.rate : num [1:4119] -1.8 1.1 1.4 1.4 -0.1 -1.1 -1.1 -0.1 -0.1 1.1 ...  
## $ cons.price.idx: num [1:4119] 92.9 94 94.5 94.5 93.2 ...  
## $ cons.conf.idx : num [1:4119] -46.2 -36.4 -41.8 -41.8 -42 -37.5 -37.5 -42 -42 -36.4 ...  
## $ euribor3m : num [1:4119] 1.31 4.86 4.96 4.96 4.19 ...  
## $ nr.employed : num [1:4119] 5099 5191 5228 5228 5196 ...  
## $ y : chr [1:4119] "no" "no" "no" "no" ...  
## - attr(\*, "spec")=  
## .. cols(  
## .. age = col\_double(),  
## .. job = col\_character(),  
## .. marital = col\_character(),  
## .. education = col\_character(),  
## .. default = col\_character(),  
## .. housing = col\_character(),  
## .. loan = col\_character(),  
## .. contact = col\_character(),  
## .. month = col\_character(),  
## .. day\_of\_week = col\_character(),  
## .. duration = col\_double(),  
## .. campaign = col\_double(),  
## .. pdays = col\_double(),  
## .. previous = col\_double(),  
## .. poutcome = col\_character(),  
## .. emp.var.rate = col\_double(),  
## .. cons.price.idx = col\_double(),  
## .. cons.conf.idx = col\_double(),  
## .. euribor3m = col\_double(),  
## .. nr.employed = col\_double(),  
## .. y = col\_character()  
## .. )

summary(data)

## age job marital education   
## Min. :18.00 Length:4119 Length:4119 Length:4119   
## 1st Qu.:32.00 Class :character Class :character Class :character   
## Median :38.00 Mode :character Mode :character Mode :character   
## Mean :40.11   
## 3rd Qu.:47.00   
## Max. :88.00   
## default housing loan contact   
## Length:4119 Length:4119 Length:4119 Length:4119   
## Class :character Class :character Class :character Class :character   
## Mode :character Mode :character Mode :character Mode :character   
##   
##   
##   
## month day\_of\_week duration campaign   
## Length:4119 Length:4119 Min. : 0.0 Min. : 1.000   
## Class :character Class :character 1st Qu.: 103.0 1st Qu.: 1.000   
## Mode :character Mode :character Median : 181.0 Median : 2.000   
## Mean : 256.8 Mean : 2.537   
## 3rd Qu.: 317.0 3rd Qu.: 3.000   
## Max. :3643.0 Max. :35.000   
## pdays previous poutcome emp.var.rate   
## Min. : 0.0 Min. :0.0000 Length:4119 Min. :-3.40000   
## 1st Qu.:999.0 1st Qu.:0.0000 Class :character 1st Qu.:-1.80000   
## Median :999.0 Median :0.0000 Mode :character Median : 1.10000   
## Mean :960.4 Mean :0.1903 Mean : 0.08497   
## 3rd Qu.:999.0 3rd Qu.:0.0000 3rd Qu.: 1.40000   
## Max. :999.0 Max. :6.0000 Max. : 1.40000   
## cons.price.idx cons.conf.idx euribor3m nr.employed   
## Min. :92.20 Min. :-50.8 Min. :0.635 Min. :4964   
## 1st Qu.:93.08 1st Qu.:-42.7 1st Qu.:1.334 1st Qu.:5099   
## Median :93.75 Median :-41.8 Median :4.857 Median :5191   
## Mean :93.58 Mean :-40.5 Mean :3.621 Mean :5166   
## 3rd Qu.:93.99 3rd Qu.:-36.4 3rd Qu.:4.961 3rd Qu.:5228   
## Max. :94.77 Max. :-26.9 Max. :5.045 Max. :5228   
## y   
## Length:4119   
## Class :character   
## Mode :character   
##   
##   
##

#Feature engineering age ,duration variable for visualization  
#age  
data<-data %>%  
 mutate(age\_group = case\_when(  
 age >= 13 & age <= 19 ~'Teen',  
 age >= 20 & age <=35 ~'younger adult',  
 age >= 36 & age <= 45 ~ "middle aged",  
 age >= 46 & age <= 60 ~"older adult",  
 age >= 61 ~ "retired "))  
  
data$age\_group<-as.factor(data$age\_group)  
  
  
  
#duration from seconds to minutes  
data<- data %>%  
 mutate(duration\_min = round(duration/60,2))  
  
  
  
#fixing pdays attribute : are marked as 999   
  
data$pdays[data$pdays == '999']<- -1  
table(data$pdays)

##   
## -1 0 1 2 3 4 5 6 7 9 10 11 12 13 14 15   
## 3959 2 3 4 52 14 4 42 10 3 8 1 5 2 1 2   
## 16 17 18 19 21   
## 2 1 2 1 1

which(data$job =='unknown')

## [1] 149 230 240 345 355 468 481 500 502 529 616 937 950 1059 1083  
## [16] 1109 1283 1330 1384 1465 1619 1643 1761 1832 1984 2076 2230 2288 2544 2710  
## [31] 2766 2852 3148 3277 3425 3551 3580 3934 4010

sum(is.na(data)) # searching for missing values

## [1] 0

sum(apply(data, 1, function(x) any(x %in% c("unknown")))) #checking unknown category variable

## [1] 1029

length(which(data$loan == 'unknown'))

## [1] 105

table(data$loan)

##   
## no unknown yes   
## 3349 105 665

data$loan[data$loan =='unknown']<- 'no' # imputation of default 'unknown ' to 'No'  
table(data$loan)

##   
## no yes   
## 3454 665

length(which(data$marital == 'unknown'))

## [1] 11

table(data$marital)

##   
## divorced married single unknown   
## 446 2509 1153 11

data$marital[data$marital == 'unknown'] <- 'married' #imputation of default 'unknown' to 'married'  
table(data$marital)

##   
## divorced married single   
## 446 2520 1153

length(which(data$job == 'unknown'))

## [1] 39

table(data$job)

##   
## admin. blue-collar entrepreneur housemaid management   
## 1012 884 148 110 324   
## retired self-employed services student technician   
## 166 159 393 82 691   
## unemployed unknown   
## 111 39

unknown\_2<-which(data$job == 'unknown' & data$default == 'unknown' ) # cross checking 2 attributes - Job - default  
unknown\_ed.job<-which(data$job == 'unknown' & data$education == 'unknown' ) # cross checking 2 attribute - job - education   
  
data<-data[-unknown\_2,] # deleting the rows with unknown values in job and default  
data<-data[-unknown\_ed.job,] # deleting the rows with unknown values in job and education  
  
  
data$job[data$job == 'unknown'] <- 'admin.' #imputation of default 'unknown' to 'married  
  
data$job <- as.factor(data$job)  
  
# combining   
#self-employed and entrepreneur  
# services and house maid   
#unemployed and retired   
  
levels(data$job) <- c("admin", "blue-collar", "self-employed",'services','management',  
 'unemployed','self-employed','services','student','technician','unemployed')  
  
table(data$job)

##   
## admin blue-collar self-employed services management   
## 1027 883 306 503 322   
## unemployed student technician   
## 276 81 690

table(data$default)

##   
## no unknown yes   
## 3304 783 1

data$default[data$default =='unknown']<- 'no' # imputation of default 'unknown ' to 'No'  
data$default<- as.factor(data$default) # as factor   
table(data$default)

##   
## no yes   
## 4087 1

table(data$housing)

##   
## no unknown yes   
## 1826 105 2157

unk\_house<-which(data$housing == 'unknown') # housing loan status  
table(data$housing)

##   
## no unknown yes   
## 1826 105 2157

mid\_var<- round(length(unk\_house)/2) # splitting the unknown values among Y/N  
for (v in unk\_house){  
 if (v <=unk\_house[mid\_var]){  
 data$housing[v] ='no'  
   
 }  
 else{   
 data$housing[v] ='yes'  
 }  
}  
   
table(data$housing)

##   
## no yes   
## 1878 2210

table(data$education)

##   
## basic.4y basic.6y basic.9y high.school   
## 425 227 573 914   
## illiterate professional.course university.degree unknown   
## 1 533 1258 157

data$education[data$education == 'basic.4y'] <- 'basic' # combining 3 basic 4y,6y,9y education as 'one'basic'   
data$education[data$education == 'basic.6y'] <- 'basic'  
data$education[data$education == 'basic.9y'] <- 'basic'  
  
unk\_edu<- which(data$education =='unknown')   
mid\_val<-round(length(unk\_edu)/2) # splitting unknown values among 'basic and 'university degree'  
  
for (v in unk\_edu){  
 if (v <=unk\_edu[mid\_val]){  
 data$education[v] ='basic'  
   
 }  
 else{   
 data$education[v] ='university.degree'  
 }  
}  
  
table(data$education) # table to check for levels

##   
## basic high.school illiterate professional.course   
## 1303 914 1 533   
## university.degree   
## 1337

#converting character variables to factor  
data<-data %>%  
 mutate(y = as.factor(y),education = as.factor(education),  
 marital= as.factor(marital),default = as.factor(default),  
 housing = as.factor(housing),loan = as.factor(loan),  
 contact = as.factor(contact),poutcome = as.factor(poutcome),  
 day\_of\_week = as.factor(day\_of\_week),month = as.factor(month),  
 previous = as.factor(previous))  
  
#Final check for any other unknown   
which(data == 'unknown')

## integer(0)

data<-data[,c(2:10,12:23)]  
  
#final summary of the data   
summary(data)

## job marital education default   
## admin :1027 divorced: 442 basic :1303 no :4087   
## blue-collar : 883 married :2501 high.school : 914 yes: 1   
## technician : 690 single :1145 illiterate : 1   
## services : 503 professional.course: 533   
## management : 322 university.degree :1337   
## self-employed: 306   
## (Other) : 357   
## housing loan contact month day\_of\_week  
## no :1878 no :3425 cellular :2642 may :1366 fri:764   
## yes:2210 yes: 663 telephone:1446 jul : 705 mon:846   
## aug : 631 thu:854   
## jun : 524 tue:836   
## nov : 446 wed:788   
## apr : 215   
## (Other): 201   
## campaign pdays previous poutcome   
## Min. : 1.000 Min. :-1.0000 0:3493 failure : 453   
## 1st Qu.: 1.000 1st Qu.:-1.0000 1: 475 nonexistent:3493   
## Median : 2.000 Median :-1.0000 2: 78 success : 142   
## Mean : 2.533 Mean :-0.7361 3: 24   
## 3rd Qu.: 3.000 3rd Qu.:-1.0000 4: 14   
## Max. :35.000 Max. :21.0000 5: 2   
## 6: 2   
## emp.var.rate cons.price.idx cons.conf.idx euribor3m   
## Min. :-3.40000 Min. :92.20 Min. :-50.80 Min. :0.635   
## 1st Qu.:-1.80000 1st Qu.:93.08 1st Qu.:-42.70 1st Qu.:1.334   
## Median : 1.10000 Median :93.60 Median :-41.80 Median :4.857   
## Mean : 0.08009 Mean :93.58 Mean :-40.51 Mean :3.616   
## 3rd Qu.: 1.40000 3rd Qu.:93.99 3rd Qu.:-36.40 3rd Qu.:4.961   
## Max. : 1.40000 Max. :94.77 Max. :-26.90 Max. :5.045   
##   
## nr.employed y age\_group duration\_min   
## Min. :4964 no :3643 middle aged :1251 Min. : 0.000   
## 1st Qu.:5099 yes: 445 older adult :1119 1st Qu.: 1.720   
## Median :5191 retired : 79 Median : 3.020   
## Mean :5166 Teen : 4 Mean : 4.272   
## 3rd Qu.:5228 younger adult:1635 3rd Qu.: 5.280   
## Max. :5228 Max. :60.720   
##

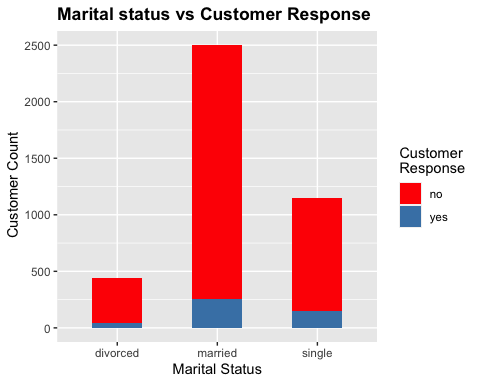
table(data$y) # checking the class variable (target)

##   
## no yes   
## 3643 445

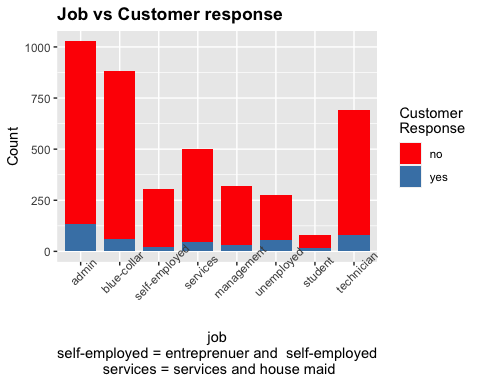
# plot   
ggplot(data,aes(y,fill = y))+geom\_bar(stat="count", width=0.8)+  
 ggtitle("Class variable distribution")+xlab("Customer response ")+  
 ylab(" Customer Count")+  
 theme(plot.title=element\_text(face ="bold") )+labs(fill = "Customer\nResponse")+  
 scale\_fill\_manual(values = c("Red","Steelblue"))



#marital  
mar<-ggplot(data,aes(marital,fill = y))+geom\_bar(stat="count", width=0.5)+  
 ggtitle("Marital status vs Customer Response")+xlab("Marital Status")+  
 ylab(" Customer Count")+  
 theme(plot.title=element\_text(face ="bold")) +labs(fill = "Customer\nResponse")+  
 scale\_fill\_manual(values = c("Red","Steelblue"))  
   
mar

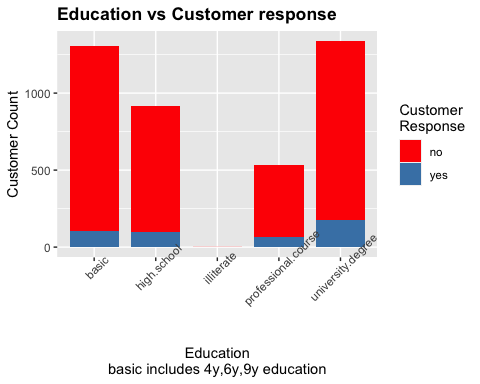


#job  
j<-ggplot(data,aes(job,fill = y))+geom\_bar(stat="count", width=0.8)+  
 ggtitle("Job vs Customer response ")+xlab(" job \nself-employed = entreprenuer and self-employed\n services = services and house maid" )+  
 ylab("Count")+labs(fill = "Customer\nResponse")+  
 theme(plot.title=element\_text(face ="bold"),axis.text.x = element\_text(angle = 45)) +  
 scale\_fill\_manual(values = c("Red","Steelblue"))  
j

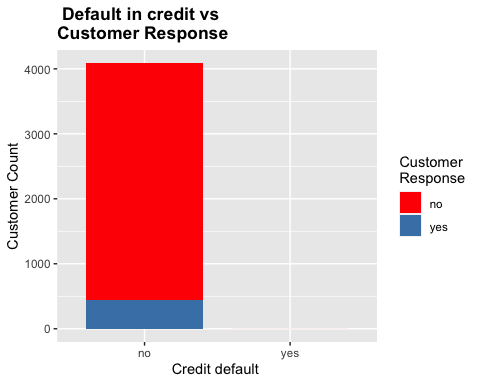


#education  
  
e<-ggplot(data,aes(data$education,fill = y))+geom\_bar(stat="count", width=0.8,)+  
 ggtitle("Education vs Customer response ")+xlab("Education\n basic includes 4y,6y,9y education ")+  
 ylab("Customer Count")+labs(fill = "Customer\nResponse")+  
 theme(plot.title=element\_text(face ="bold"),axis.text.x = element\_text(angle = 45)) +  
 scale\_fill\_manual(values = c("Red","Steelblue"))  
e

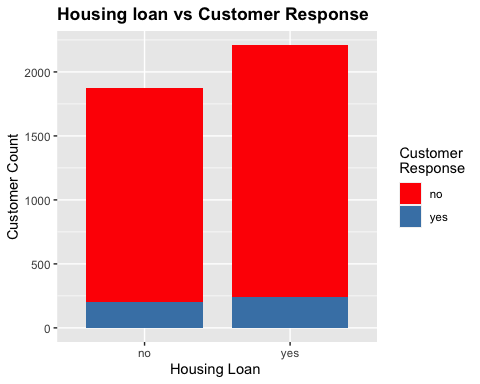
## Warning: Use of `data$education` is discouraged. Use `education` instead.



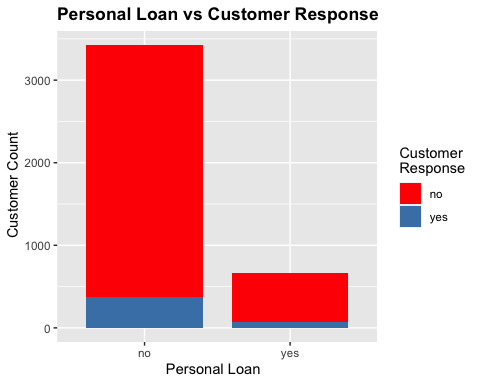
#default  
d<-ggplot(data,aes(default,fill = y))+geom\_bar(stat="count", width=0.8)+  
 ggtitle(" Default in credit vs \nCustomer Response ")+xlab("Credit default ")+  
 ylab("Customer Count")+labs(fill = "Customer\nResponse")+  
 theme(plot.title=element\_text(face ="bold")) +  
 scale\_fill\_manual(values = c("Red","Steelblue"))  
d



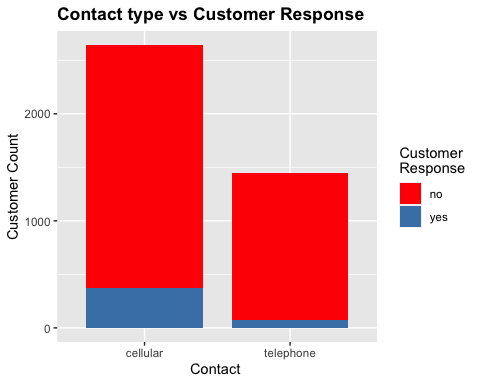
#housing loan  
h<-ggplot(data,aes(housing,fill = y))+geom\_bar(stat="count", width=0.8)+  
 ggtitle("Housing loan vs Customer Response ")+xlab("Housing Loan ")+  
 ylab("Customer Count")+labs(fill = "Customer\nResponse")+  
 theme(plot.title=element\_text(face ="bold")) +  
 scale\_fill\_manual(values = c("Red","Steelblue"))  
h



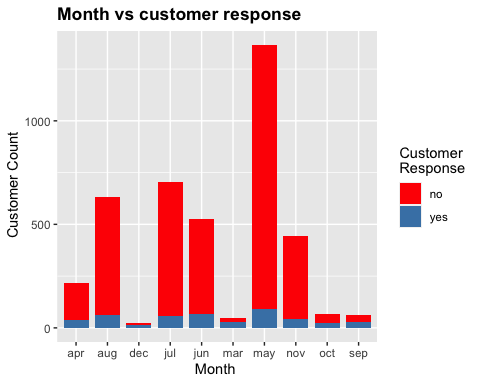
#loan  
l<-ggplot(data,aes(loan,fill = y))+geom\_bar(stat="count", width=0.8)+  
ggtitle("Personal Loan vs Customer Response ")+xlab("Personal Loan ")+  
 ylab(" Customer Count")+labs(fill = "Customer\nResponse")+  
 theme(plot.title=element\_text(face ="bold")) +  
 scale\_fill\_manual(values = c("Red","Steelblue"))  
l



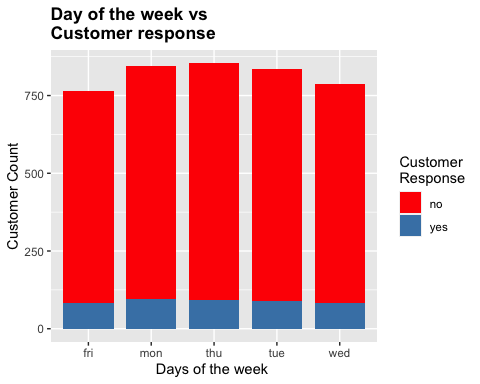
#contact  
c<-ggplot(data,aes(contact,fill = y))+geom\_bar(stat="count", width=0.8)+  
 ggtitle("Contact type vs Customer Response ")+xlab("Contact ")+  
 ylab("Customer Count")+labs(fill = "Customer\nResponse")+  
 theme(plot.title=element\_text(face ="bold")) +  
 scale\_fill\_manual(values = c("Red","Steelblue"))  
c



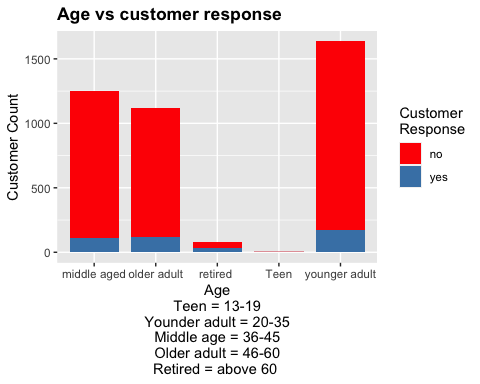
#month  
mnth<-ggplot(data,aes(month,fill = y))+geom\_bar(stat="count", width=0.8)+  
 ggtitle("Month vs customer response ")+xlab("Month ")+  
 ylab(" Customer Count")+labs(fill = "Customer\nResponse")+  
 theme(plot.title=element\_text(face ="bold")) +  
 scale\_fill\_manual(values = c("Red","Steelblue"))#  
mnth



#day\_of\_week  
d<-ggplot(data,aes(day\_of\_week,fill = y))+geom\_bar(stat="count", width=0.8)+  
 ggtitle("Day of the week vs \nCustomer response ")+xlab("Days of the week ")+  
 ylab("Customer Count")+labs(fill = "Customer\nResponse")+  
 theme(plot.title=element\_text(face ="bold")) +  
 scale\_fill\_manual(values = c("Red","Steelblue"))  
d



#age  
ag<-ggplot(data,aes(age\_group,fill = y))+geom\_bar(stat="count", width=0.8)+  
ggtitle("Age vs customer response ")+  
 xlab("Age\nTeen = 13-19\nYounder adult = 20-35\nMiddle age = 36-45\nOlder adult = 46-60\nRetired = above 60 ")+  
 ylab("Customer Count")+labs(fill = "Customer\nResponse")+  
 theme(plot.title=element\_text(face ="bold")) +  
 scale\_fill\_manual(values = c("Red","Steelblue"))  
ag

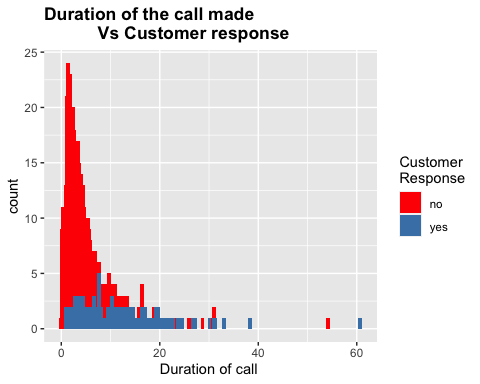


#duration  
dur<-ggplot(data,aes(duration\_min,fill = y))+geom\_histogram(stat="count", width=0.8,bins = 3)+  
 ggtitle("Duration of the call made   
 Vs Customer response ")+xlab("Duration of call ")+  
 ylab("count")+labs(fill = "Customer\nResponse")+  
 theme(plot.title=element\_text(face ="bold")) +  
 scale\_fill\_manual(values = c("Red","Steelblue"))

## Warning: Ignoring unknown parameters: binwidth, bins, pad

dur

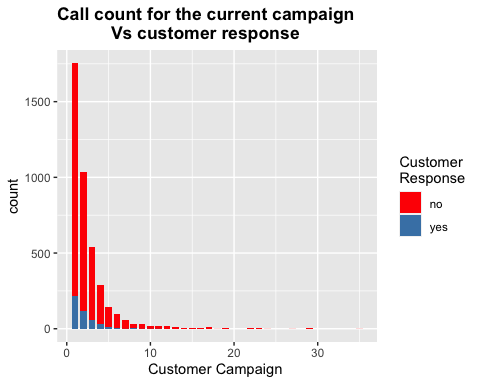
## Warning: position\_stack requires non-overlapping x intervals



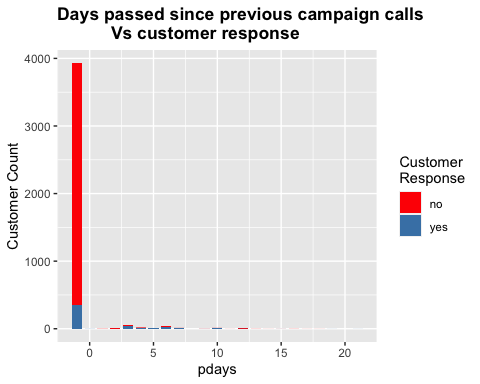
#campaign  
camp<-ggplot(data,aes(campaign,fill = y))+geom\_histogram(stat="count", width=0.8)+  
 ggtitle("Call count for the current campaign  
 Vs customer response ")+  
 xlab("Customer Campaign ")+ylab("count")+labs(fill = "Customer\nResponse")+  
 theme(plot.title=element\_text(face ="bold")) +  
 scale\_fill\_manual(values = c("Red","Steelblue"))

## Warning: Ignoring unknown parameters: binwidth, bins, pad

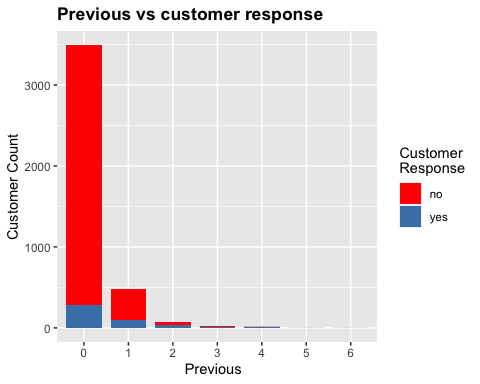
camp



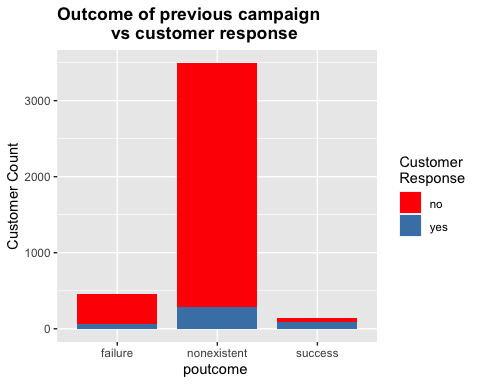
#pday  
pday<-ggplot(data,aes(pdays,fill = y))+geom\_bar(stat="count", width=0.8)+  
 ggtitle("Days passed since previous campaign calls   
 Vs customer response ")+  
 xlab("pdays")+ylab("Customer Count")+labs(fill = "Customer\nResponse")+  
 theme(plot.title=element\_text(face ="bold")) +  
 scale\_fill\_manual(values = c("Red","Steelblue"))  
pday



# previous  
prev<-ggplot(data,aes(previous,fill = y))+geom\_bar(stat="count", width=0.8)+  
 ggtitle("Previous vs customer response ")+xlab("Previous ")+  
 ylab("Customer Count")+labs(fill = "Customer\nResponse")+  
 theme(plot.title=element\_text(face ="bold")) +  
 scale\_fill\_manual(values = c("Red","Steelblue"))#  
  
prev



#poutcome  
pout<-ggplot(data,aes(poutcome,fill = y))+geom\_bar(stat="count", width=0.8)+  
 ggtitle("Outcome of previous campaign   
 vs customer response ")+xlab("poutcome ")+  
 ylab("Customer Count")+labs(fill = "Customer\nResponse")+  
 theme(plot.title=element\_text(face ="bold")) +  
 scale\_fill\_manual(values = c("Red","Steelblue"))  
pout

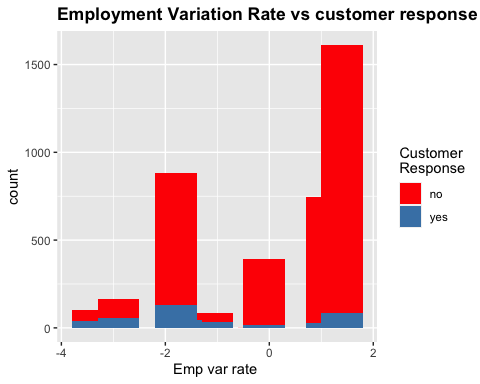


#emp\_var\_rate  
evar<-ggplot(data,aes(emp.var.rate,fill = y))+geom\_histogram(stat="count", width=0.8)+  
 ggtitle("Employment Variation Rate vs customer response ")+xlab("Emp var rate ")+  
 ylab("count")+labs(fill = "Customer\nResponse")+  
 theme(plot.title=element\_text(face ="bold")) +  
 scale\_fill\_manual(values = c("Red","Steelblue"))#

## Warning: Ignoring unknown parameters: binwidth, bins, pad

evar

## Warning: position\_stack requires non-overlapping x intervals

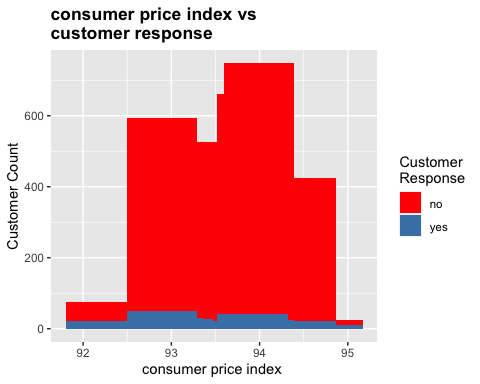


#cons.price.idx  
price.idx<-ggplot(data,aes(cons.price.idx,fill = y))+geom\_histogram(stat="count", width=0.8,binwidth = 500)+  
 ggtitle("consumer price index vs \ncustomer response ")+  
 xlab("consumer price index ")+ylab("Customer Count")+labs(fill = "Customer\nResponse")+  
 theme(plot.title=element\_text(face ="bold")) +  
 scale\_fill\_manual(values = c("Red","Steelblue"))

## Warning: Ignoring unknown parameters: binwidth, bins, pad

price.idx

## Warning: position\_stack requires non-overlapping x intervals

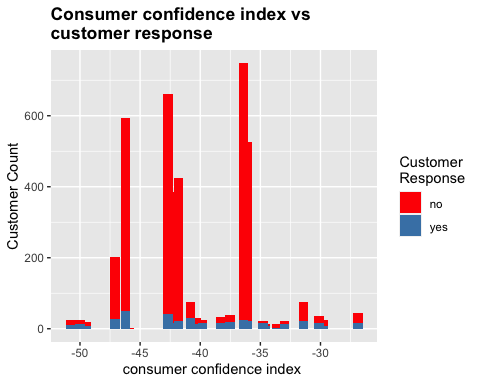


#cons.conf.indx  
conf.indx<-ggplot(data,aes(cons.conf.idx,fill = y))+geom\_histogram(stat="count", width=0.8)+  
 ggtitle("Consumer confidence index vs\ncustomer response ")+  
 xlab("consumer confidence index ")+ylab("Customer Count")+labs(fill = "Customer\nResponse")+  
 theme(plot.title=element\_text(face ="bold")) +  
 scale\_fill\_manual(values = c("Red","Steelblue"))

## Warning: Ignoring unknown parameters: binwidth, bins, pad

conf.indx

## Warning: position\_stack requires non-overlapping x intervals

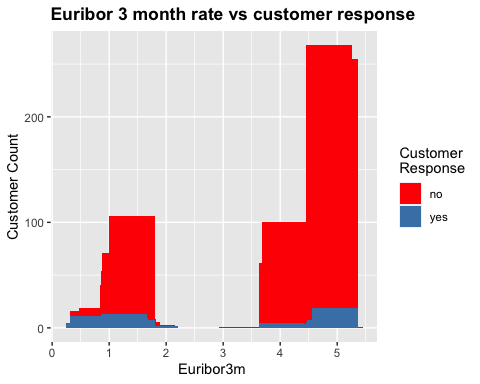


#euribor3m  
eur<-ggplot(data,aes(euribor3m,fill = y))+geom\_histogram(stat="count", width=0.8)+  
 ggtitle("Euribor 3 month rate vs customer response ")+xlab("Euribor3m ")+  
 ylab("Customer Count")+labs(fill = "Customer\nResponse")+  
 theme(plot.title=element\_text(face ="bold")) +  
 scale\_fill\_manual(values = c("Red","Steelblue"))

## Warning: Ignoring unknown parameters: binwidth, bins, pad

eur

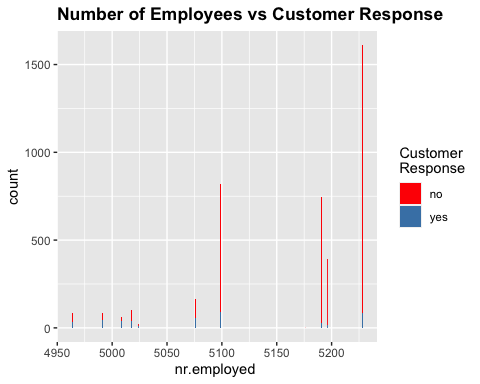
## Warning: position\_stack requires non-overlapping x intervals



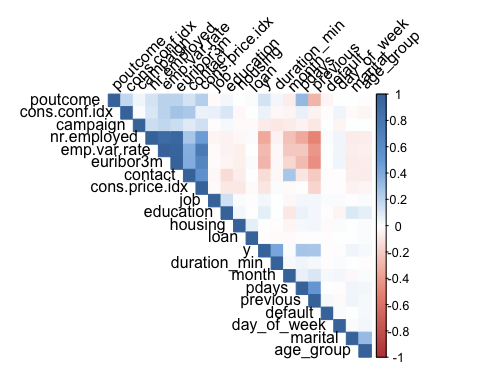
#nr.emp  
nr.emp<-ggplot(data,aes(nr.employed,fill = y))+geom\_histogram(stat="count", width=0.8,bins = 3,binswidth = 0.05)+  
 ggtitle("Number of Employees vs Customer Response ")+xlab("nr.employed ")+  
 ylab("count")+labs(fill = "Customer\nResponse")+  
 theme(plot.title=element\_text(face ="bold")) +  
 scale\_fill\_manual(values = c("Red","Steelblue"))

## Warning: Ignoring unknown parameters: binwidth, bins, pad, binswidth

nr.emp



cor\_data<-data %>%  
 mutate(job = as.numeric(job),  
 marital = as.numeric(marital),  
 education = as.numeric(education),  
 contact = as.numeric(contact),  
 month = as.numeric(month),  
 previous = as.numeric(previous),  
 poutcome = as.numeric(poutcome),  
 y = as.numeric(y),  
 pdays = as.numeric(pdays),  
 default = as.numeric(default),  
 housing = as.numeric(housing),  
 loan = as.numeric(loan),  
 month = as.numeric(month),  
 day\_of\_week = as.numeric(day\_of\_week),  
 campaign = as.numeric(campaign),  
 age\_group = as.numeric(age\_group),  
 y = as.numeric(y)  
   
)  
 cor\_test<- round(cor(cor\_data),2)  
   
 col <- colorRampPalette(c("#BB4444", "#EE9988", "#FFFFFF", "#77AADD", "#4477AA"))  
 #plot   
 corrplot(cor\_test, method="color", col=col(200),   
 type="upper", order="hclust",   
 tl.col="black", tl.srt=45, #Text label color and rotation  
 diag=TRUE   
 )



(findCorrelation(cor\_test,cutoff = 0.7)) # finding attributes with more than 0.7 correlation coefficient

## [1] 17 14

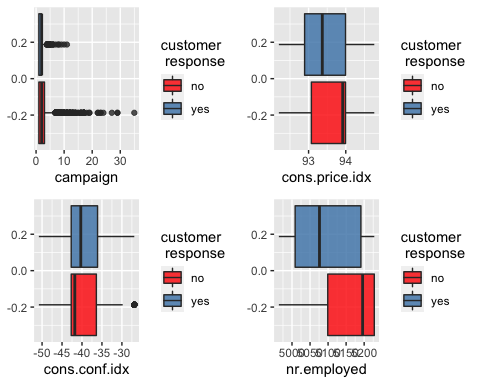
# (emp.var.rate - euribor3m) , (emp.var.rate-nr.employed),(euribor3m - nr.employed) are highly correlated   
  
  
data<-data[,-c(14,17,21)] # emp.var.rate and euribor3m are removed along with duration\_min(based on UCI repository description)  
  
str(data)

## tibble [4,088 × 18] (S3: tbl\_df/tbl/data.frame)  
## $ job : Factor w/ 8 levels "admin","blue-collar",..: 2 4 4 4 1 4 1 3 4 2 ...  
## $ marital : Factor w/ 3 levels "divorced","married",..: 2 3 2 2 2 3 3 2 1 2 ...  
## $ education : Factor w/ 5 levels "basic","high.school",..: 1 2 2 1 5 5 5 5 4 1 ...  
## $ default : Factor w/ 2 levels "no","yes": 1 1 1 1 1 1 1 1 1 1 ...  
## $ housing : Factor w/ 2 levels "no","yes": 2 1 2 1 2 1 2 2 1 1 ...  
## $ loan : Factor w/ 2 levels "no","yes": 1 1 1 1 1 1 1 1 1 1 ...  
## $ contact : Factor w/ 2 levels "cellular","telephone": 1 2 2 2 1 1 1 1 1 2 ...  
## $ month : Factor w/ 10 levels "apr","aug","dec",..: 7 7 5 5 8 10 10 8 8 7 ...  
## $ day\_of\_week : Factor w/ 5 levels "fri","mon","thu",..: 1 1 5 1 2 3 2 2 4 3 ...  
## $ campaign : num [1:4088] 2 4 1 3 1 3 4 2 1 1 ...  
## $ pdays : num [1:4088] -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 ...  
## $ previous : Factor w/ 7 levels "0","1","2","3",..: 1 1 1 1 1 3 1 1 2 1 ...  
## $ poutcome : Factor w/ 3 levels "failure","nonexistent",..: 2 2 2 2 2 1 2 2 1 2 ...  
## $ cons.price.idx: num [1:4088] 92.9 94 94.5 94.5 93.2 ...  
## $ cons.conf.idx : num [1:4088] -46.2 -36.4 -41.8 -41.8 -42 -37.5 -37.5 -42 -42 -36.4 ...  
## $ nr.employed : num [1:4088] 5099 5191 5228 5228 5196 ...  
## $ y : Factor w/ 2 levels "no","yes": 1 1 1 1 1 1 1 1 1 1 ...  
## $ age\_group : Factor w/ 5 levels "middle aged",..: 5 1 5 1 2 5 5 1 5 5 ...

dim(data)

## [1] 4088 18

num\_variable = list("campaign","cons.price.idx","cons.conf.idx","nr.employed") # list of numerical variables   
b= list()  
for (value in num\_variable){  
 b[[value]]<-ggplot(data,aes\_string(x = value,fill = data$y)) +  
 geom\_boxplot(alpha = 0.8) +   
 scale\_fill\_manual(values = c("Red","Steelblue"),name = 'customer\n response')  
   
}  
  
do.call(grid.arrange, c(b, ncol=2))



table(data$campaign)

##   
## 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16   
## 1754 1035 540 288 141 99 59 34 32 20 18 16 11 6 2 7   
## 17 18 19 22 23 24 27 29 35   
## 14 1 2 2 2 1 1 2 1

(length(boxplot.stats(data$campaign)$out)) # total outliers in campaign attribute

## [1] 231

quant<-quantile(data$campaign,0.98) # checking for outliers above 98th percentile   
quant

## 98%   
## 11

out\_val<-(which(data$campaign >=quant))  
data<- data[-out\_val,] # removing the outliers beyond 98 %  
  
dim(data)

## [1] 4002 18

bor<-Boruta(y~.,data = data,doTrace = 2)

## 1. run of importance source...

## 2. run of importance source...

## 3. run of importance source...

## 4. run of importance source...

## 5. run of importance source...

## 6. run of importance source...

## 7. run of importance source...

## 8. run of importance source...

## 9. run of importance source...

## 10. run of importance source...

## 11. run of importance source...

## After 11 iterations, +24 secs:

## confirmed 9 attributes: cons.conf.idx, cons.price.idx, contact, marital, month and 4 more;

## rejected 4 attributes: campaign, default, housing, loan;

## still have 4 attributes left.

## 12. run of importance source...

## 13. run of importance source...

## 14. run of importance source...

## 15. run of importance source...

## After 15 iterations, +33 secs:

## confirmed 1 attribute: job;

## still have 3 attributes left.

## 16. run of importance source...

## 17. run of importance source...

## 18. run of importance source...

## 19. run of importance source...

## 20. run of importance source...

## 21. run of importance source...

## 22. run of importance source...

## 23. run of importance source...

## 24. run of importance source...

## 25. run of importance source...

## 26. run of importance source...

## 27. run of importance source...

## 28. run of importance source...

## After 28 iterations, +1.1 mins:

## rejected 1 attribute: age\_group;

## still have 2 attributes left.

## 29. run of importance source...

## 30. run of importance source...

## 31. run of importance source...

## 32. run of importance source...

## 33. run of importance source...

## 34. run of importance source...

## 35. run of importance source...

## 36. run of importance source...

## 37. run of importance source...

## 38. run of importance source...

## 39. run of importance source...

## After 39 iterations, +1.6 mins:

## confirmed 1 attribute: education;

## still have 1 attribute left.

## 40. run of importance source...

## 41. run of importance source...

## 42. run of importance source...

## 43. run of importance source...

## 44. run of importance source...

## 45. run of importance source...

## 46. run of importance source...

## 47. run of importance source...

## 48. run of importance source...

## 49. run of importance source...

## 50. run of importance source...

## 51. run of importance source...

## 52. run of importance source...

## 53. run of importance source...

## 54. run of importance source...

## 55. run of importance source...

## 56. run of importance source...

## 57. run of importance source...

## 58. run of importance source...

## 59. run of importance source...

## 60. run of importance source...

## 61. run of importance source...

## 62. run of importance source...

## 63. run of importance source...

## 64. run of importance source...

## 65. run of importance source...

## 66. run of importance source...

## 67. run of importance source...

## 68. run of importance source...

## 69. run of importance source...

## 70. run of importance source...

## 71. run of importance source...

## 72. run of importance source...

## 73. run of importance source...

## 74. run of importance source...

## 75. run of importance source...

## 76. run of importance source...

## 77. run of importance source...

## 78. run of importance source...

## 79. run of importance source...

## 80. run of importance source...

## 81. run of importance source...

## 82. run of importance source...

## 83. run of importance source...

## 84. run of importance source...

## 85. run of importance source...

## 86. run of importance source...

## 87. run of importance source...

## 88. run of importance source...

## 89. run of importance source...

## 90. run of importance source...

## 91. run of importance source...

## 92. run of importance source...

## 93. run of importance source...

## 94. run of importance source...

## 95. run of importance source...

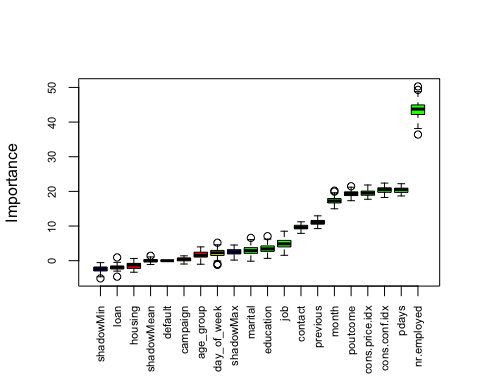
## 96. run of importance source...

## 97. run of importance source...

## 98. run of importance source...

## 99. run of importance source...

plot(bor,cex = 1,las = 3,xlab = "",cex.axis = 0.7) # plot



(getSelectedAttributes(bor, withTentative = F)) # not rejected attributes

## [1] "job" "marital" "education" "contact"   
## [5] "month" "pdays" "previous" "poutcome"   
## [9] "cons.price.idx" "cons.conf.idx" "nr.employed"

data\_bor<-data[,c(1:3,7:8,11:17)] # eliminating rejected attributes   
  
summary(data\_bor)

## job marital education   
## admin :1003 divorced: 436 basic :1283   
## blue-collar : 868 married :2450 high.school : 890   
## technician : 675 single :1116 illiterate : 1   
## services : 489 professional.course: 522   
## management : 315 university.degree :1306   
## self-employed: 301   
## (Other) : 351   
## contact month pdays previous poutcome   
## cellular :2606 may :1343 Min. :-1.0000 0:3407 failure : 453   
## telephone:1396 jul : 682 1st Qu.:-1.0000 1: 475 nonexistent:3407   
## aug : 613 Median :-1.0000 2: 78 success : 142   
## jun : 502 Mean :-0.7304 3: 24   
## nov : 446 3rd Qu.:-1.0000 4: 14   
## apr : 215 Max. :21.0000 5: 2   
## (Other): 201 6: 2   
## cons.price.idx cons.conf.idx nr.employed y   
## Min. :92.20 Min. :-50.80 Min. :4964 no :3558   
## 1st Qu.:93.08 1st Qu.:-42.70 1st Qu.:5099 yes: 444   
## Median :93.44 Median :-41.80 Median :5191   
## Mean :93.57 Mean :-40.53 Mean :5165   
## 3rd Qu.:93.99 3rd Qu.:-36.40 3rd Qu.:5228   
## Max. :94.77 Max. :-26.90 Max. :5228   
##

data\_model<- as.data.frame(data\_bor)  
  
# balancing the data set  
  
data\_model$y<- as.factor(data\_model$y) # before balancing in SMOTE class variable should be factor  
  
balanced\_data\_base<- SMOTE(y~.,data\_model, perc.over = 920, perc.under = 111,k = 5) # balancing the class variable   
as.data.frame(table(balanced\_data\_base$y))

## Var1 Freq  
## 1 no 4435  
## 2 yes 4440

dim(balanced\_data\_base)

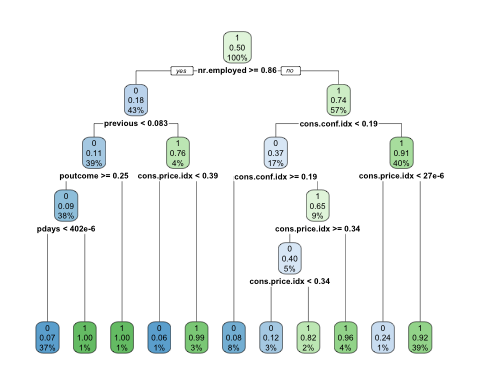
## [1] 8875 12

balanced\_data\_base<- balanced\_data\_base %>%  
 mutate(job = as.numeric(job),  
 marital = as.numeric(marital),  
 education = as.numeric(education),  
 contact = as.numeric(contact),  
 month = as.numeric(month),  
 previous = as.numeric(previous),  
 poutcome = as.numeric(poutcome),  
 y = as.numeric(y),  
 pdays = as.numeric(pdays)  
 )  
  
normalize <- function(x) {  
 return ((x - min(x)) / (max(x) - min(x))) # Normalizing function   
}  
data\_normalize <- as.data.frame(lapply(balanced\_data\_base, normalize)) # normalizing   
  
  
as.data.frame(table(data\_normalize$y))

## Var1 Freq  
## 1 0 4435  
## 2 1 4440

set.seed(111)  
  
data\_normalize$y<- as.factor(data\_normalize$y)  
trainIndex <- createDataPartition(data\_normalize$y,   
p = .75,   
list = FALSE,   
times = 1)  
  
train\_norm <- data\_normalize[ trainIndex,] # train set   
test\_norm <- data\_normalize[-trainIndex,] # test set

model\_norm<- rpart(train\_norm$y~., data = train\_norm, method = 'class') # model   
  
rpart.plot(model\_norm) # plot



# performance evaluation   
predict\_norm <-predict(model\_norm, test\_norm, type = 'class') # on test set   
   
table\_norm <- confusionMatrix(predict\_norm,test\_norm$y,positive = '1') # confusion matrix for class '1'  
 (table\_norm)

## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction 0 1  
## 0 1024 69  
## 1 84 1041  
##   
## Accuracy : 0.931   
## 95% CI : (0.9197, 0.9412)  
## No Information Rate : 0.5005   
## P-Value [Acc > NIR] : <2e-16   
##   
## Kappa : 0.862   
##   
## Mcnemar's Test P-Value : 0.2577   
##   
## Sensitivity : 0.9378   
## Specificity : 0.9242   
## Pos Pred Value : 0.9253   
## Neg Pred Value : 0.9369   
## Prevalence : 0.5005   
## Detection Rate : 0.4693   
## Detection Prevalence : 0.5072   
## Balanced Accuracy : 0.9310   
##   
## 'Positive' Class : 1   
##

predict\_norm1 <-predict(model\_norm, test\_norm, type = 'prob')  
 pred\_norm<- prediction(predict\_norm1[,2],test\_norm$y)  
   
# precision,recall,f1 score and accuracy  
 table\_tree<- as.matrix(table(predict\_norm,test\_norm$y))  
 dia\_t = diag(table\_tree) # correctly classified instances per class   
 rows\_t = apply(table\_tree, 1, sum) # instances per class  
 cols\_t= apply(table\_tree, 2, sum) # prediction per class   
   
 recall\_t = dia\_t /cols\_t   
 precision\_t = dia\_t / rows\_t   
 f1\_t = 2 \* precision\_t \* recall\_t / (precision\_t + recall\_t)   
 accuracy\_t = sum(dia\_t)/sum(table\_tree)  
   
 data.frame(precision\_t,recall\_t,accuracy\_t,f1\_t) # precision,recall,accuracy,F1 score of Decision tree

## precision\_t recall\_t accuracy\_t f1\_t  
## 0 0.9368710 0.9241877 0.9310189 0.9304861  
## 1 0.9253333 0.9378378 0.9310189 0.9315436

# AUC  
 Pred\_auc\_norm <- performance(pred\_norm, "auc")  
 auc\_t<-round(as.numeric(Pred\_auc\_norm@y.values), 2)  
 auc\_t # decision tree AUC

## [1] 0.94

naive\_Bayes\_norm<-naiveBayes(y~.,data = train\_norm) #model  
NB\_Predict\_norm=predict(naive\_Bayes\_norm,test\_norm) # on test set  
nb\_table\_norm <- confusionMatrix(NB\_Predict\_norm,test\_norm$y,positive = '1') #confusion matrix  
nb\_table\_norm

## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction 0 1  
## 0 936 318  
## 1 172 792  
##   
## Accuracy : 0.7791   
## 95% CI : (0.7612, 0.7962)  
## No Information Rate : 0.5005   
## P-Value [Acc > NIR] : < 2.2e-16   
##   
## Kappa : 0.5582   
##   
## Mcnemar's Test P-Value : 5.737e-11   
##   
## Sensitivity : 0.7135   
## Specificity : 0.8448   
## Pos Pred Value : 0.8216   
## Neg Pred Value : 0.7464   
## Prevalence : 0.5005   
## Detection Rate : 0.3571   
## Detection Prevalence : 0.4346   
## Balanced Accuracy : 0.7791   
##   
## 'Positive' Class : 1   
##

# performance evaluation   
predict\_norm\_nb <-predict(naive\_Bayes\_norm, test\_norm, type = 'raw')  
pred\_nb<- prediction(predict\_norm\_nb[,2],test\_norm$y)  
  
# precision, recall,accuracy and F1 score   
table\_nb<- as.matrix(table(NB\_Predict\_norm,test\_norm$y))  
dia = diag(table\_nb) # correctly classified instances per class   
rows = apply(table\_nb, 1, sum) # instances per class  
cols = apply(table\_nb, 2, sum) # prediction per class   
  
precision = dia /rows  
recall = dia / cols   
f1 = 2 \* precision \* recall / (precision + recall)   
accuracy = sum(dia)/sum(table\_nb)  
  
data.frame(precision,recall,accuracy,f1) # precision,recall,accuracy,F1 score for Naive bayes

## precision recall accuracy f1  
## 0 0.7464115 0.8447653 0.7790803 0.7925487  
## 1 0.8215768 0.7135135 0.7790803 0.7637416

#AUC   
Pred\_auc\_nbnor <- performance(pred\_nb, "auc")  
auc\_nb<-round(as.numeric(Pred\_auc\_nbnor@y.values), 2)  
auc\_nb # auc for Naive bayes

## [1] 0.87

# ROC curve of both models   
  
#Decision tree  
ROC\_tree<-performance(pred\_norm,'tpr','fpr')  
plot(ROC\_tree,  
 col = 'red',xlab = "sensitivity",ylab = '1-specificity',main = "ROC of both classifers ",print.auc = T)  
abline(a = 0,b = 1)  
  
#naive bayes  
ROC<-performance(pred\_nb,'tpr','fpr')  
plot(ROC,add = T,  
 col = 'green',xlab = "sensitivity",ylab = '1-specificity',print.auc = T)  
legend("right", c("Decision tree", "Naive bayes"), lty=1,   
 col = c("red", "green"), bty="o", inset=c(0,-0.15))

