# Q1a.

dc<-read.csv("bank-market1.csv", na.strings="")

#Change the last column in dc to numeric: deposit=1 or 0 for deposit="yes" or "no" respectively and save them in d.

dc$deposit[dc$deposit=="yes"] <- as.numeric(1)

dc$deposit[dc$deposit=="no"] <- as.numeric(0)

d <- transform(dc, deposit = as.numeric(deposit))

# Q1b.

set.seed(82475)

# save random 80% data of d to d0 as training set

out<-ransub(d, 0.8)

d0<-out$train

d1<-out$test

y0<-factor(d0[,12])

# Q1c.

z0<-scale.con(d0[,c(1,4,7:10)]) # transform continuous or ordinal var.

z1<-scale.con(d1[,c(1,4,7:10)])

# Q1d.

bank.knn<-k\_nn(z0,z1,y0,d1[,12],v=10)

(tab<-table(bank.knn,d1[,12])) # classification table

print(tab)

print(erate(tab))

## output:

Text

Description automatically generated

# Q1e.

f1sc<-function(tab) { # assume the input tab is 2x2 with 1st row an column as negative

tp<-tab[2,2]

fp<-tab[1,2]

fn<-tab[2,1]

prec<-tp/(fp+tp)

recall<-tp/(fn+tp)

f1=2\*prec\*recall/(prec+recall)

er<-erate(tab)

cat('erate =',er, 'precision =', prec, ' recall =', recall, 'F1 score =', f1, '\n')

}

f1sc(tab)

## output:



# Q2a.

if(!require(e1071)) {install.packages('e1071')}

library(e1071)

cl<-factor(d0[,12])

bank.nb<-naiveBayes(d0[,1:11],cl)

# q2b.

prob<-predict(bank.nb, d1[,1:11], type="raw")

# Q2c.

pr<-(prob[,2]>0.5)

(tab<-table(pr,d1[,12])) # classification table

print(tab)

print(erate(tab))

## output



# Q2d.

## f1sc(tab)

## output



# Q2e.

## for (x in 1:9) {

## print(x/10)

## pr\_temp<-(prob[,2]>(x/10))

## f1sc(table(pr\_temp,d1[,12]))

## }

## Ans

0.4 has the highest score

## output

Text

Description automatically generated

# Q3a.

# lreg<-glm(deposit~age+balance+duration+campaign+pdays+previous,data=d0,binomial)

# bank.lreg<-step(lreg)

## output

# 

# Q3b.

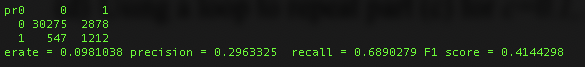
pr0<-(bank.lreg$fit>0.5)+0

(tab<-table(pr0,d0$deposit))

print(tab)

f1sc(tab)

## output



# Q3c.

## prob<-predict(bank.lreg,d1,type='response') # predition on d1 using type=response

## pr<-(prob>0.5)+0 # out-sample prediction

## (tab<-table(pr,d1$deposit))

## print(tab)

## f1sc(tab)

## output



# Q3d.

## for (x in 1:9) {

## print(x/10)

## pr\_temp<-(prob>(x/10))+0

## f1sc(table(pr\_temp,d1[,12]))

## }

## Ans

0.2 has the highest score

## output

Text

Description automatically generated

# Q4.

Naive Bayes in q2 has the lowest error rate and highest f1 score