FMPH227 project

Daniel Zoleikhaeian, Keren Hu

2023-12-02

```
library(MASS)
library(class)
library(glmnet)
```

```
##      Matrix
```

```
## Loaded glmnet 4.1-8
```

```
library(compareGroups)
library(methods)
library(tree)
library(randomForest)
```

```
## randomForest 4.7-1.1
```

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

```
library(leaps)
library(Rfast)
```

```
##      Rcpp
```

```
##      RcppZiggurat
```

```
##      RcppParallel
```

```
##
##    'RcppParallel'
```

```
## The following object is masked from 'package:Rcpp':
##
##     LdFlags
```

```
##
## Rfast: 2.1.0
```

```
##  ___ __ __ __ __    __ __ __ __ __ _            _             __ __ __ __ __    __ __ __ __ __ __
## |  __ __ __ __  |  |  __ __ __ __ _/           / \           |  __ __ __ __ /   /__ __ _   _ __ _|
## | |           | |  | |                        / _ \          | |                       / /
## | |           | |  | |                       / / \ \         | |                      / /
## | |           | |  | |                      / /   \ \        | |                     / /
## | |__ __ __ __| |  | |__ __ __ __          / /     \ \       | |__ __ __ __ _      / /__/\
## |    __ __ __ __|  |  __ __ __ __|        / /__ _ __\ \      |_ __ __ __ _   |     / ___  /
## |   \              | |                    / _ _ _ _ _ _ \                  | |     \/  / /
## | |\ \             | |                   / /           \ \                 | |        / /
## | | \ \            | |                  / /             \ \                | |       / /
## | |  \ \           | |                 / /               \ \               | |      / /
## | |   \ \__ __ _   | |                / /                 \ \    _ __ __ __ _| |     / /
## |_|    \__ __ __\  |_|               /_/                   \_\  /_ __ __ __ ___|     \/
```

```
##
##     'Rfast'
```

```
## The following objects are masked from 'package:class':
##
##     knn, knn.cv
```

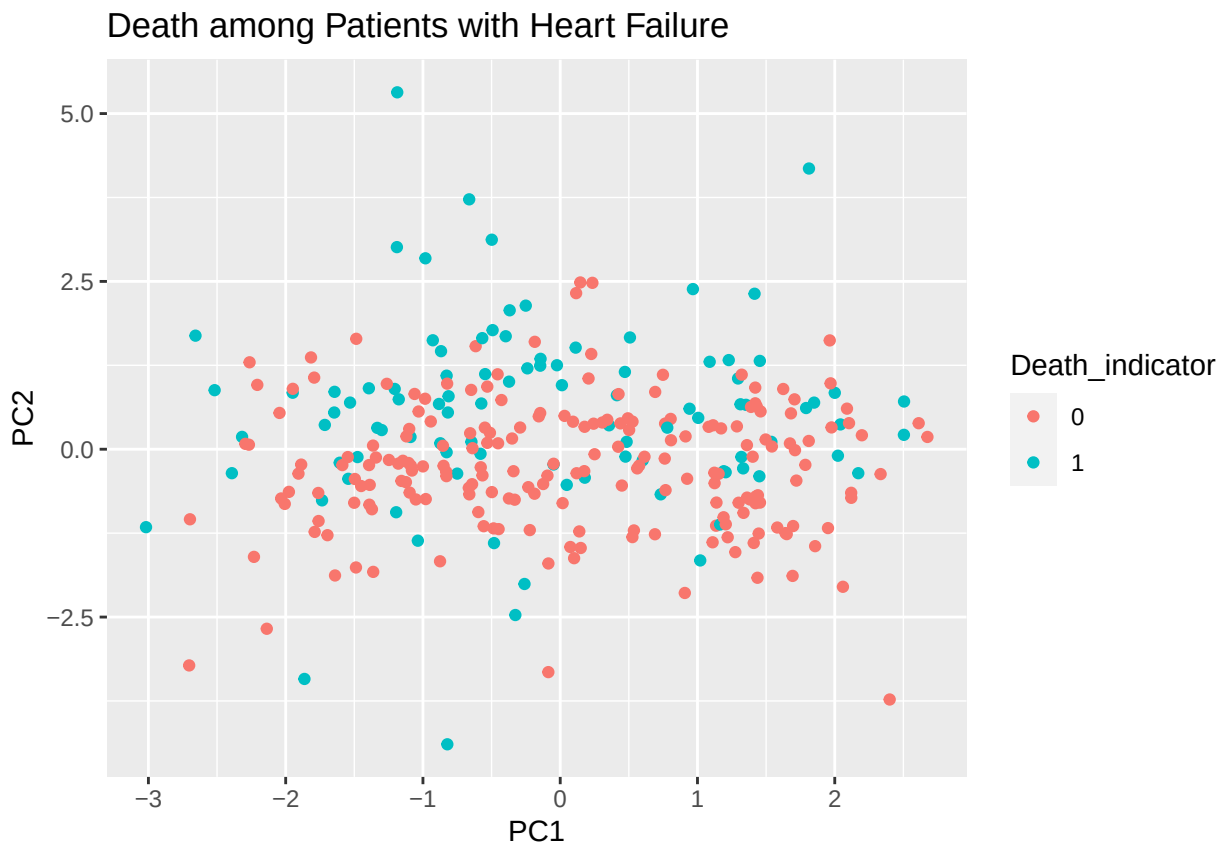
```
library(ggplot2)
```

```
##
##     'ggplot2'
```

```
## The following object is masked from 'package:randomForest':
##
##     margin
```

```
setwd("D:/Users/Karen/Desktop/FMPH227/Project")
heart <- read.csv("heart_failure_clinical_records_dataset.csv")
heart$DEATH_EVENT <- as.factor(heart$DEATH_EVENT)
xnams <- c("age", "anaemia", "creatinine_phosphokinase", "diabetes",
           "ejection_fraction", "high_blood_pressure", "platelets",
           "serum_creatinine", "serum_sodium", "sex", "smoking")
fmla <- as.formula(paste("DEATH_EVENT ~ ", paste(xnams, collapse= "+")))
heart <- heart[, -12]
```

```
# PCA to collapse the predictor variables into 2 dimensions
labels <- heart$DEATH_EVENT
pr.out=prcomp(heart[,xnams], scale=TRUE)

# Plotting the Results
df_pc12 <- data.frame(PC1 = pr.out$x[,1],
                      PC2 = pr.out$x[,2],
                      Death_indicator = heart$DEATH_EVENT)

p <- ggplot(df_pc12,aes(x=PC1,y=PC2,col=Death_indicator)) + geom_point() + ggtitle('Death among Patients
p
```




```
names(pr.out)
```

```
## [1] "sdev"     "rotation" "center"   "scale"    "x"
```

```
pr.var <- pr.out$sdev^2
pve <-pr.var/sum(pr.var)

sum(pve[1:2])
```

```
## [1] 0.2780344
```

```
# Demographic description
tab1 <- compareGroups( DEATH_EVENT~ ., data = heart)
restab <- createTable(tab1)
export2md(restab, caption = "Demographic characteristics of study participants")
```

```
## Warning in max(positions): max       -Inf
```

Table 1: Demographic characteristics of study participants

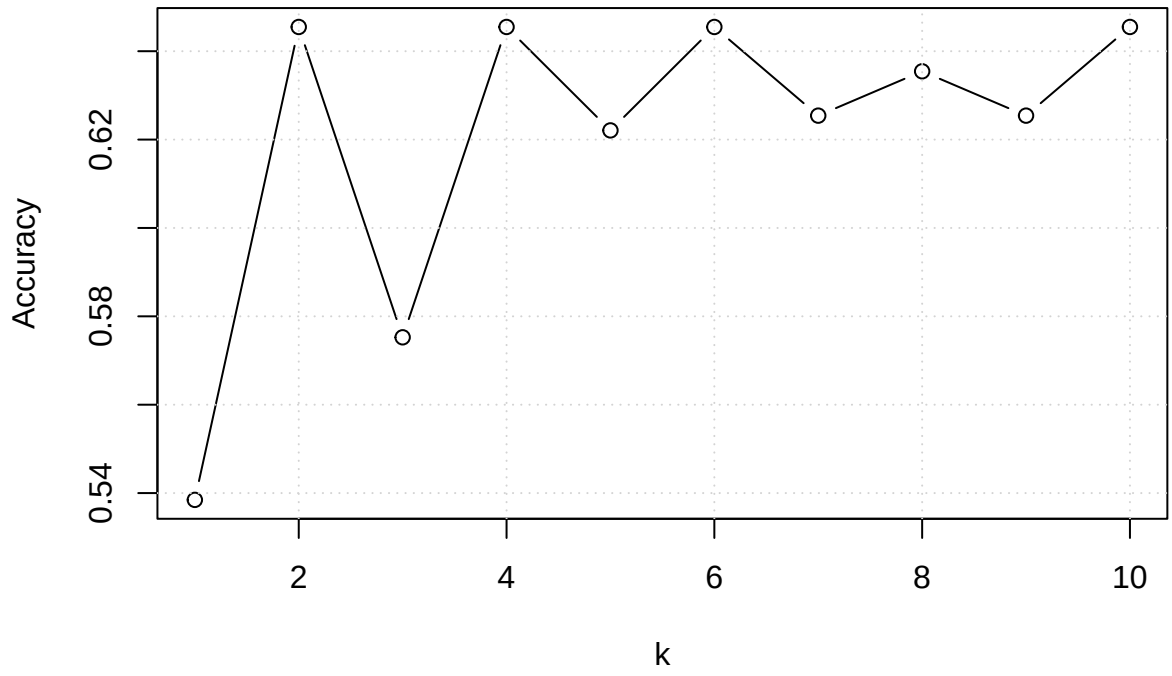
	0 N=203	1 N=96	p.overall
age	58.8 (10.6)	65.2 (13.2)	<0.001
anaemia	0.41 (0.49)	0.48 (0.50)	0.257
creatinine_phosphokinase	540 (754)	670 (1317)	0.369
diabetes	0.42 (0.49)	0.42 (0.50)	0.973
ejection_fraction	40.3 (10.9)	33.5 (12.5)	<0.001
high_blood_pressure	0.33 (0.47)	0.41 (0.49)	0.180
platelets	266657 (97531)	256381 (98526)	0.399
serum_creatinine	1.18 (0.65)	1.84 (1.47)	<0.001
serum_sodium	137 (3.98)	135 (5.00)	0.002
sex	0.65 (0.48)	0.65 (0.48)	0.941
smoking	0.33 (0.47)	0.31 (0.47)	0.828

```
# logistic
logis.fit <- glm(fmla,data = heart, family=binomial)
summary(logis.fit)
```

```
##
## Call:
## glm(formula = fmla, family = binomial, data = heart)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.3184  -0.7692  -0.4436   0.8293   2.4880
##
## Coefficients:
##                            Estimate Std. Error z value Pr(>|z|)
## (Intercept)               4.964e+00  4.601e+00   1.079 0.280625
## age                       5.569e-02  1.313e-02   4.241 2.23e-05 ***
## anaemia                   4.179e-01  3.009e-01   1.389 0.164904
## creatinine_phosphokinase  2.905e-04  1.428e-04   2.034 0.041907 *
## diabetes                  1.514e-01  2.974e-01   0.509 0.610644
## ejection_fraction        -7.032e-02  1.486e-02  -4.731 2.23e-06 ***
## high_blood_pressure       4.189e-01  3.061e-01   1.369 0.171092
## platelets                -7.094e-07  1.617e-06  -0.439 0.660857
## serum_creatinine          6.619e-01  1.734e-01   3.817 0.000135 ***
## serum_sodium             -5.667e-02  3.338e-02  -1.698 0.089558 .
## sex                      -3.990e-01  3.508e-01  -1.137 0.255394
## smoking                   1.356e-01  3.486e-01   0.389 0.697300
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 375.35  on 298  degrees of freedom
## Residual deviance: 294.28  on 287  degrees of freedom
## AIC: 318.28
##
## Number of Fisher Scoring iterations: 5
```

```r
logis.fit.step <- step(logis.fit, direction = "backward", trace = 0)
logis.pred <- predict(logis.fit.step, type = "response")


# logis.fit.sub <- regsubsets(fmla, data = heart, nvmax = 14)
# summary(logis.fit.sub)
#  cbind( Cp  = summary(logis.fit.sub)$cp,
#         r2 = summary(logis.fit.sub)$rsq,
#     Adj_r2 = summary(logis.fit.sub)$adjr2,
#     BIC    =summary(logis.fit.sub)$bic
# )

tbl <- table(heart$DEATH_EVENT,logis.pred>0.5)
crrcls <- (tbl[1,1] + tbl[2,2])/sum(tbl)
sens <- tbl[2,2]/(tbl[2,1] + tbl[2,2])
spec <- tbl[1,1]/(tbl[1,1] + tbl[1,2])
ppv <- tbl[2,2]/(tbl[1,2] + tbl[2,2])
npv <- tbl[1,1]/(tbl[1,1] + tbl[2,1])

Accuracy.logis <- c(crrcls,sens,spec,ppv,npv)
dd <- data.frame(Accuracy.logis,
                 row.names = c("Correct Classification",
                               "Sensitivity", "Specificity",
                               "Positive Predictive Value",
                               "Negative Predictive value"))

# # LDA
# lda.fit <- lda(fmla,data = heart)
# lda.pred <- predict(lda.fit,heart)
# tbl <- table(heart$DEATH_EVENT, lda.pred$posterior[,2]>0.5)
#
# crrcls <- (tbl[1,1] + tbl[2,2])/sum(tbl)
# sens <- tbl[2,2]/(tbl[2,1] + tbl[2,2])
# spec <- tbl[1,1]/(tbl[1,1] + tbl[1,2])
# ppv <- tbl[2,2]/(tbl[1,2] + tbl[2,2])
# npv <- tbl[1,1]/(tbl[1,1] + tbl[2,1])
# dd$Accuracy.lda <- c(crrcls,sens,spec,ppv,npv)
#
#
# # QDA
# qda.fit <- qda(fmla,data = heart)
# qda.pred <- predict(qda.fit,heart)
# tbl <- table(heart$DEATH_EVENT, qda.pred$posterior[,2]>0.5)
#
# crrcls <- (tbl[1,1] + tbl[2,2])/sum(tbl)
# sens <- tbl[2,2]/(tbl[2,1] + tbl[2,2])
# spec <- tbl[1,1]/(tbl[1,1] + tbl[1,2])
# ppv <- tbl[2,2]/(tbl[1,2] + tbl[2,2])
# npv <- tbl[1,1]/(tbl[1,1] + tbl[2,1])
#
# dd$Accuracy.qda <- c(crrcls,sens,spec,ppv,npv)

##k-NN
```

```r
##k=6

xxx <- as.matrix(heart[,xnams])
yyy <- heart$DEATH_EVENT

knn.res <- knn.cv(x=xxx, y=yyy, nfolds = 10, stratified = FALSE, k=1:10, type = "C")
plot(1:10, knn.res$crit, type = "b", xlab = "k", ylab = "Accuracy" )
grid()
```

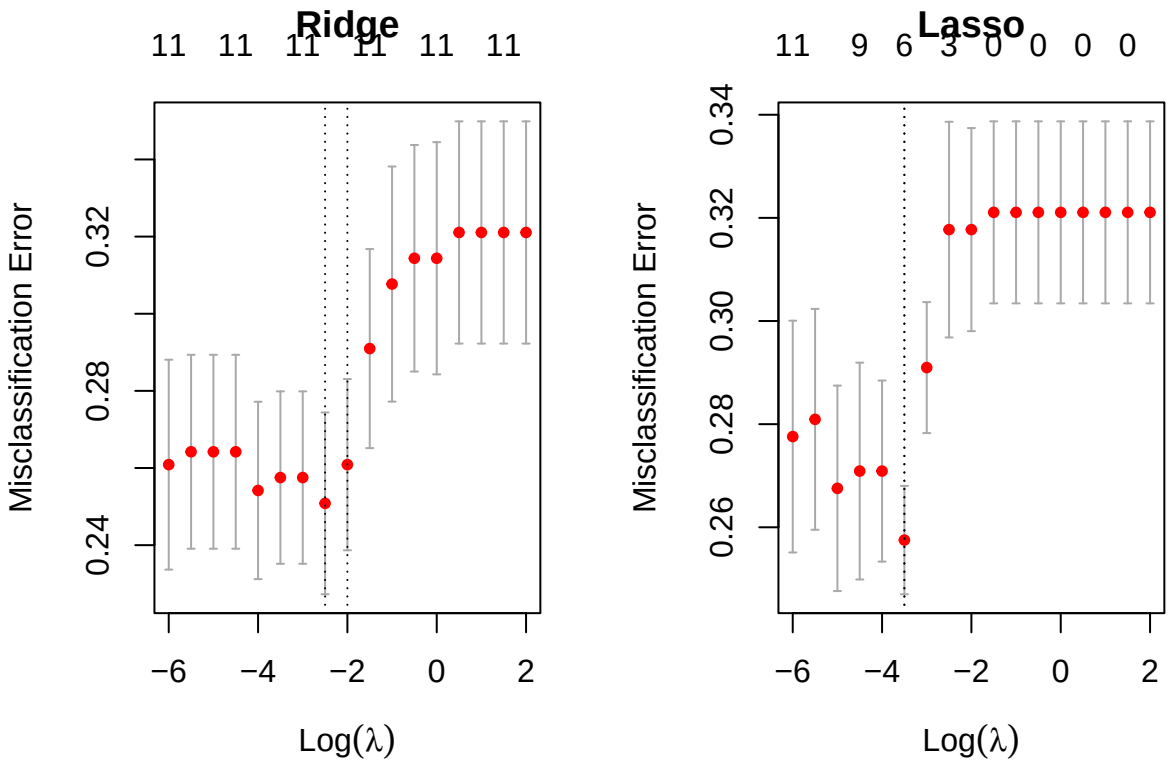



```r
# k=6 is best
knn.pred <- class::knn(heart[,xnams], heart[,xnams],heart$DEATH_EVENT, k=which.max(knn.res$crit))

tbl <- table(knn.pred, heart$DEATH_EVENT)
crrcls <- (tbl[1,1] + tbl[2,2])/sum(tbl)
sens <- tbl[2,2]/(tbl[2,1] + tbl[2,2])
spec <- tbl[1,1]/(tbl[1,1] + tbl[1,2])
ppv <- tbl[2,2]/(tbl[1,2] + tbl[2,2])
npv <- tbl[1,1]/(tbl[1,1] + tbl[2,1])
dd$Accuracy.knn <- c(crrcls,sens,spec,ppv,npv)

round(dd,3)
```

```
##                          Accuracy.logis Accuracy.knn
## Correct Classification            0.759        0.799
## Sensitivity                       0.469        0.670
```

```
## Specificity                          0.897        0.870
## Positive Predictive Value            0.682        0.740
## Negative Predictive value            0.781        0.828
```

```r
#RIDGE and LASSO fit
# some variable are not significant, and we want to
xxx <- as.matrix(heart[,xnams])
yyy <- heart$DEATH_EVENT

gridd <- exp(seq(2,-6,-0.5))  ##lambda values

##ridge fit
rdg.fit <- glmnet(xxx,yyy,family="binomial",alpha=0,lambda=gridd)

##lasso fit
lso.fit <- glmnet(xxx,yyy,family="binomial",alpha=1,lambda=gridd)

##cross-validation to select lambda
set.seed(2446)
cv.rdgeg <- cv.glmnet(xxx,yyy,family="binomial",alpha=0,
                      lambda=gridd,nfolds=10,
                      type.measure="class")

cv.lsoeg <- cv.glmnet(xxx,yyy,family="binomial",alpha=1,
                      lambda=gridd,nfolds=10,
                      type.measure="class")

par(mfrow=c(1,2))
plot(cv.rdgeg, main= "Ridge")
plot(cv.lsoeg, main="Lasso")
```




```
#################### OPTIMAL LAMBDA
####Variables (and coefficients) at optimal $\lambda$s for RIDGE
c(cv.rdgeg$lambda.min,cv.rdgeg$lambda.1se)
```

```
## [1] 0.0820850 0.1353353
```

```
log(c(cv.rdgeg$lambda.min,cv.rdgeg$lambda.1se))
```

```
## [1] -2.5 -2.0
```

```
##Coeff at "best" lambda
coef(cv.rdgeg,s="lambda.min")
```

```
## 12 x 1 sparse Matrix of class "dgCMatrix"
##                                      s1
## (Intercept)                4.273624e+00
## age                        3.347040e-02
## anaemia                    2.257373e-01
## creatinine_phosphokinase   1.577768e-04
## diabetes                   5.848502e-02
## ejection_fraction         -3.963952e-02
## high_blood_pressure        2.651457e-01
## platelets                 -4.515352e-07
## serum_creatinine           3.999138e-01
```

```
## serum_sodium             -4.608717e-02
## sex                      -1.532914e-01
## smoking                   9.474076e-03
```

```
##Coeff at  lambda + 1SE
coef(cv.rdgeg,s="lambda.1se")
```

```
## 12 x 1 sparse Matrix of class "dgCMatrix"
##                                      s1
## (Intercept)               3.834191e+00
## age                       2.757704e-02
## anaemia                   1.804542e-01
## creatinine_phosphokinase  1.246216e-04
## diabetes                  3.940739e-02
## ejection_fraction        -3.203018e-02
## high_blood_pressure       2.196873e-01
## platelets                -3.993189e-07
## serum_creatinine          3.325017e-01
## serum_sodium             -4.130068e-02
## sex                      -1.066390e-01
## smoking                  -6.106674e-03
```

```
###Variables (and coefficients) at optimal $\lambda$s for LASSO:
##Coeff at "best" lambda
c(cv.lsoeg$lambda.min,cv.lsoeg$lambda.1se)
```

```
## [1] 0.03019738 0.03019738
```

```
log(c(cv.lsoeg$lambda.min,cv.lsoeg$lambda.1se))
```

```
## [1] -3.5 -3.5
```

```
coef(cv.lsoeg,s="lambda.min")
```

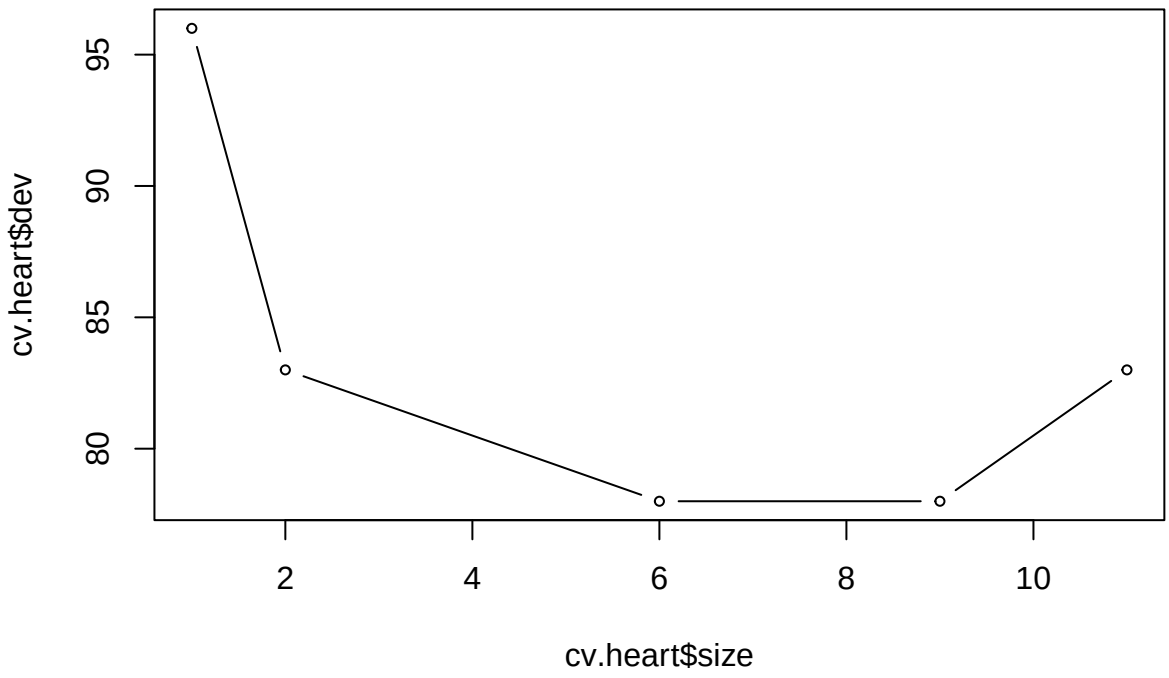
```
## 12 x 1 sparse Matrix of class "dgCMatrix"
##                                      s1
## (Intercept)               1.312048e+00
## age                       3.433664e-02
## anaemia                   .
## creatinine_phosphokinase  2.849715e-05
## diabetes                  .
## ejection_fraction        -4.460490e-02
## high_blood_pressure       5.478457e-02
## platelets                 .
## serum_creatinine          4.363465e-01
## serum_sodium             -2.336767e-02
## sex                       .
## smoking                   .
```

```r
##Coeff at  lambda + 1SE
coef(cv.lsoeg,s="lambda.1se")
```

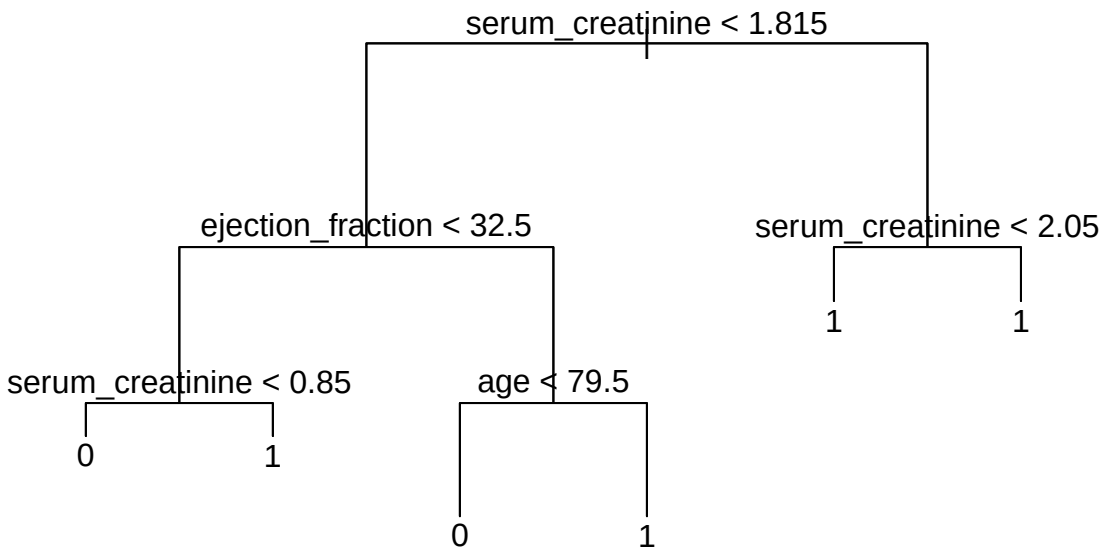
```
## 12 x 1 sparse Matrix of class "dgCMatrix"
##                                       s1
## (Intercept)                1.312048e+00
## age                        3.433664e-02
## anaemia                    .
## creatinine_phosphokinase   2.849715e-05
## diabetes                   .
## ejection_fraction         -4.460490e-02
## high_blood_pressure        5.478457e-02
## platelets                  .
## serum_creatinine           4.363465e-01
## serum_sodium              -2.336767e-02
## sex                        .
## smoking                    .
```

```r
# predict from ridge and lasso

ridge.pred <- predict(cv.rdgeg, s=cv.rdgeg$lambda.min, newx = xxx,
                      type = "response")
tbl <- table(heart$DEATH_EVENT,ridge.pred>0.5)
crrcls <- (tbl[1,1] + tbl[2,2])/sum(tbl)
sens <- tbl[2,2]/(tbl[2,1] + tbl[2,2])
spec <- tbl[1,1]/(tbl[1,1] + tbl[1,2])
ppv <- tbl[2,2]/(tbl[1,2] + tbl[2,2])
npv <- tbl[1,1]/(tbl[1,1] + tbl[2,1])
dd$Accuracy.ridge <- c(crrcls,sens,spec,ppv,npv)

lasso.pred <- predict(cv.lsoeg, s=cv.lsoeg$lambda.min, newx = xxx,
                      type = "response")
tbl <- table(heart$DEATH_EVENT,lasso.pred>0.5)
crrcls <- (tbl[1,1] + tbl[2,2])/sum(tbl)
sens <- tbl[2,2]/(tbl[2,1] + tbl[2,2])
spec <- tbl[1,1]/(tbl[1,1] + tbl[1,2])
ppv <- tbl[2,2]/(tbl[1,2] + tbl[2,2])
npv <- tbl[1,1]/(tbl[1,1] + tbl[2,1])
dd$Accuracy.lasso <- c(crrcls,sens,spec,ppv,npv)
```

```r
library(tree)

set.seed(2023)
t1 <- tree(fmla, heart)
cv.heart <- cv.tree(t1,FUN = prune.misclass, K=10)
plot(cv.heart$size,cv.heart$dev,type="b",cex=0.63) # best tree has 6 terminal nodes
```




```
t1.pruned <- prune.tree(t1,best=6)

plot(t1.pruned)
text(t1.pruned,pretty=0)
```




```
# tree primarily uses serum_creatinine, ejection fraction, and age
# serum_creatinine and ejection fraction are used the most
summary(t1.pruned)
```

```
##
## Classification tree:
## snip.tree(tree = t1, nodes = c(8L, 10L, 9L, 7L))
## Variables actually used in tree construction:
## [1] "serum_creatinine"  "ejection_fraction" "age"
## Number of terminal nodes:  6
## Residual mean deviance:  0.8974 = 262.9 / 293
## Misclassification error rate: 0.2074 = 62 / 299
```

```
# getting the in-sample testing characteristics
probs <- predict(t1.pruned, heart)
yhat <- ifelse(probs[,1] > 0.5, 0, 1)

tbl <- table(heart$DEATH_EVENT, yhat)

crrcls <- (tbl[1,1] + tbl[2,2])/sum(tbl)
sens <- tbl[2,2]/(tbl[2,1] + tbl[2,2])
spec <- tbl[1,1]/(tbl[1,1] + tbl[1,2])
ppv <- tbl[2,2]/(tbl[1,2] + tbl[2,2])
npv <- tbl[1,1]/(tbl[1,1] + tbl[2,1])
dd$Accuracy.tree <- c(crrcls,sens,spec,ppv,npv)
```

```
round(dd,3)
```

```
##                            Accuracy.logis Accuracy.knn Accuracy.ridge
## Correct Classification              0.759        0.799          0.759
## Sensitivity                         0.469        0.670          0.354
## Specificity                         0.897        0.870          0.951
## Positive Predictive Value           0.682        0.740          0.773
## Negative Predictive value           0.781        0.828          0.757
##                            Accuracy.lasso Accuracy.tree
## Correct Classification              0.763         0.793
## Sensitivity                         0.375         0.802
## Specificity                         0.946         0.788
## Positive Predictive Value           0.766         0.642
## Negative Predictive value           0.762         0.894
```

```
dd <- round(dd,3)
knitr::kable(dd,align = "c", caption = "Accuracy for 5 models", format = "simple")
```

Table 2: Accuracy for 5 models

	Accuracy.logis	Accuracy.knn	Accuracy.ridge	Accuracy.lasso	Accuracy.tree
Correct Classification	0.759	0.799	0.759	0.763	0.793
Sensitivity	0.469	0.670	0.354	0.375	0.802
Specificity	0.897	0.870	0.951	0.946	0.788
Positive Predictive Value	0.682	0.740	0.773	0.766	0.642
Negative Predictive value	0.781	0.828	0.757	0.762	0.894

```
best.fit<-glm(data=heart,
              DEATH_EVENT~age+creatinine_phosphokinase+ejection_fraction+high_blood_pressure+serum_crea
              family = binomial)
summary(best.fit)
```

```
##
## Call:
## glm(formula = DEATH_EVENT ~ age + creatinine_phosphokinase +
##     ejection_fraction + high_blood_pressure + serum_creatinine +
##     serum_sodium, family = binomial, data = heart)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.2936  -0.7625  -0.4830   0.8238   2.5145
##
## Coefficients:
##                            Estimate Std. Error z value Pr(>|z|)
## (Intercept)               4.2167350  4.4885461   0.939    0.348
## age                       0.0531381  0.0127147   4.179 2.92e-05 ***
## creatinine_phosphokinase  0.0002227  0.0001361   1.636    0.102
## ejection_fraction        -0.0673740  0.0145776  -4.622 3.81e-06 ***
## high_blood_pressure       0.4751499  0.2976117   1.597    0.110
## serum_creatinine          0.6535260  0.1662640   3.931 8.47e-05 ***
```

```
## serum_sodium             -0.0516233  0.0327201  -1.578    0.115
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 375.35  on 298  degrees of freedom
## Residual deviance: 298.16  on 292  degrees of freedom
## AIC: 312.16
##
## Number of Fisher Scoring iterations: 5
```

```
Model <- c("Logistic", "Ridge", "LASSO", "K-NN", "Class Tree")
Test_err <- c(0.2720, 0.2718, 0.2703, 0.4416, 0.3004)
res <- data.frame(Model, Test_err)
knitr::kable(res, align = "c", caption = "LOO Bootstrap Error for 5 models", format = "simple")
```

Table 3: LOO Bootstrap Error for 5 models

Model	Test_err
Logistic	0.2720
Ridge	0.2718
LASSO	0.2703
K-NN	0.4416
Class Tree	0.3004