```
#install.packages("data.table")
library(data.table)
#install.packages("gamlss")
library(gamlss)
#install.packages("woe")
library(woe)
#install.packages("xtable")
library(xtable)
library(tidyverse)
library(pROC)
library(caret)

# Leitura de Dados ----
dt <- fread("./data/heart.csv")

# Retirada de valores com zero em RestingBP e Cholesterol
dt <- dt[RestingBP != 0 & Cholesterol !=0 & Oldpeak > 0] # https://d-nb.info/1242792767/34

# Numero de variveis independentes (p) e numero de linhas (n)
p <- ncol(dt)-1
n <- nrow(dt)

# Categoricas: Sex, ChestPainType, FastingBS, RestingECG, ExerciseAngina, ST_Slope, HeartDisease
# Continuas: Age, RestingBP, Cholesterol, MaxHR, Oldpeak
cat(colnames(dt), sep = ', ')

# Valore unicos por variavel
print(lapply(lapply(dt, unique),sort))

# Tranformacao para variavel categorica ----
# dt[,c(2,3,6,7,9,11,12)] <- lapply(dt[,c(2,3,6,7,9,11,12)], as.factor)
dt$HeartDisease <- as.factor(dt$HeartDisease)


#Simplify ChestPainType, only check for Asymptomatic pain
dt$ChestPainASY <- 0
dt$ChestPainASY[dt$ChestPainType == "ASY"] <- 1
dt$ChestPainASY <- as.factor(dt$ChestPainASY)

#Simplify RestingECG, only check for ST
dt$RestingECGST <- 0
dt$RestingECGST[dt$RestingECG == "ST"] <- 1
dt$RestingECGST <- as.factor(dt$RestingECGST)

#Simplify ST_Slope, only check for Down/Flat
dt$ST_SlopeDownFlat <- 1
dt$ST_SlopeDownFlat[dt$ST_Slope == "Up"] <- 0
dt$ST_SlopeDownFlat <- as.factor(dt$ST_SlopeDownFlat)

#Turn Sex into factor
dt$Sex_fct <- 1
dt$Sex_fct[dt$Sex == "F"] <- 0
dt$Sex_fct <- as.factor(dt$Sex_fct)

dt$ExerciseAngina_fct <- 1
dt$ExerciseAngina_fct[dt$ExerciseAngina == "N"] <- 0
dt$ExerciseAngina_fct <- as.factor(dt$ExerciseAngina_fct)

#removecols <-
c("ChestPainType","Cholesterol_imp","RestingBP_imp","Oldpeak_imp","RestingECG","ST_Slope","ExerciseAngina","Sex")
dt <- dt[,c("Age","RestingBP", "Cholesterol", "FastingBS", "MaxHR", "Oldpeak",
            "HeartDisease", "ChestPainASY", "RestingECGST",
            "ST_SlopeDownFlat", "Sex_fct", "ExerciseAngina_fct")]

# Selecao de variaveis (IV e GAIC)
# Categoricas: Sex, ChestPainType, FastingBS, RestingECG, ExerciseAngina, ST_Slope, HeartDisease
# Continuas: Age, RestingBP, Cholesterol, MaxHR, Oldpeak

iv = {}

iv$age <- woe(Data=dt, Independent="Age", Continuous=TRUE, Dependent="HeartDisease", C_Bin=10, Bad=0, Good=1)
[,c("MIN", "MAX", "IV")]
iv$restingbp <- woe(Data=dt, Independent="RestingBP", Continuous=TRUE, Dependent="HeartDisease", C_Bin=10,
Bad=0, Good=1)[,c("MIN", "MAX", "IV")]
iv$choles <- woe(Data=dt, Independent="Cholesterol", Continuous=TRUE, Dependent="HeartDisease", C_Bin=10, Bad=0,
Good=1)[,c("MIN", "MAX", "IV")]
iv$maxhr <- woe(Data=dt, Independent="MaxHR", Continuous=TRUE, Dependent="HeartDisease", C_Bin=10, Bad=0,
Good=1)[,c("MIN", "MAX", "IV")]
iv$oldpeak <- woe(Data=dt, Independent="Oldpeak", Continuous=TRUE, Dependent="HeartDisease", C_Bin=10, Bad=0,
Good=1)[,c("MIN", "MAX", "IV")]

iv$sex <- woe(Data=dt, Independent="Sex_fct", Continuous=FALSE, Dependent="HeartDisease", C_Bin=10, Bad=0,
Good=1)[,c("BIN", "IV")]
iv$chest_pain <- woe(Data=dt, Independent="ChestPainASY", Continuous=FALSE, Dependent="HeartDisease", C_Bin=10,
Bad=0, Good=1)[,c("BIN", "IV")]
iv$fasting <- woe(Data=dt, Independent="FastingBS", Continuous=FALSE, Dependent="HeartDisease", C_Bin=10, Bad=0,
```

```r
Good=1)[,c("BIN", "IV")]
iv$restECG <- woe(Data=dt, Independent="RestingECGST", Continuous=FALSE, Dependent="HeartDisease", C_Bin=10,
Bad=0, Good=1)[,c("BIN", "IV")]
iv$exc_angi <- woe(Data=dt, Independent="ExerciseAngina_fct", Continuous=FALSE, Dependent="HeartDisease",
C_Bin=10, Bad=0, Good=1)[,c("BIN", "IV")]
iv$ST_slope <- woe(Data=dt, Independent="ST_SlopeDownFlat", Continuous=FALSE, Dependent="HeartDisease",
C_Bin=10, Bad=0, Good=1)[,c("BIN", "IV")]


# faz arquivo .tex da tabela
print(xtable(iv$age, type = "latex"), file = "iv_age.tex")
print(xtable(iv$restingbp, type = "latex"), file = "iv_restingbp.tex")
print(xtable(iv$choles, type = "latex"), file = "iv_choles.tex")
print(xtable(iv$maxhr, type = "latex"), file = "iv_maxhr.tex")
print(xtable(iv$oldpeak, type = "latex"), file = "iv_oldpeak.tex")
print(xtable(iv$sex, type = "latex"), file = "iv_sex.tex")
print(xtable(iv$chest_pain, type = "latex"), file = "iv_chest_pain.tex")
print(xtable(iv$fasting, type = "latex"), file = "iv_fasting.tex")
print(xtable(iv$restECG, type = "latex"), file = "iv_restECG.tex")
print(xtable(iv$exc_angi, type = "latex"), file = "iv_exc_angi.tex")
print(xtable(iv$ST_slope, type = "latex"), file = "iv_ST_slope.tex")

# Soma dos valoes de IV
iv_sum <- data.frame(
  age = sum(iv$age["IV"]),
  restingbp = sum(iv$restingbp["IV"]),
  choles = sum(iv$choles["IV"]),
  maxhr = sum(iv$maxhr["IV"]),
  oldpeak = sum(iv$oldpeak["IV"]),
  sex = sum(iv$sex["IV"]),
  chest_pain = sum(iv$chest_pain["IV"]),
  fasting = sum(iv$fasting["IV"]),
  restECG = sum(iv$restECG["IV"]),
  exc_angi = sum(iv$exc_angi["IV"]),
  ST_slope = sum(iv$ST_slope["IV"])
)

# Checa quais variaveis nao sao significativas. Criterio: iv < 0.1
# Variaveis com iv < 0.1: Colesterol, FastingBS e RestingECGST
iv_sum[1, iv_sum < 0.1]

# Retira-se as colunas acima mencionadas
iv_heart <- dt[, c("Age","RestingBP", "MaxHR", "Oldpeak",
                   "HeartDisease", "ChestPainASY",
                   "ST_SlopeDownFlat", "Sex_fct", "ExerciseAngina_fct")]
print(xtable(iv_sum, type = "latex"), file = "iv_sum.tex")

# Aplicacao do modelo Binomial com funcao de ligacao LOGIT
mod_logit <- gamlss(HeartDisease ~ ., data = iv_heart, family = BI(mu.link=logit), type = "response")
mod_probit <- gamlss(HeartDisease ~ ., data = iv_heart, family = BI(mu.link=probit), type = "response")
mod_clog <- gamlss(HeartDisease ~ ., data = iv_heart, family = BI(mu.link=cloglog), type = "response")

# Utilizacao do coeficiente GAIC passo a passo, tanto foward como backward
gamlss::stepGAIC(mod_logit,
                 scope = c(lower = ~ 1,
                           upper = ~ .),
                 direction = "both")
gamlss::stepGAIC(mod_probit,
                 scope = c(lower = ~ 1,
                           upper = ~ .),
                 direction = "both")
gamlss::stepGAIC(mod_clog,
                 scope = c(lower = ~ 1,
                           upper = ~ .),
                 direction = "both")


# Variaveis selecionadas em cada modelo
# logit e probit: retira-se MaxHR
# Complemento log log: retira-se Age
heart_step_logit <- iv_heart[,c("Sex_fct", "ChestPainASY", "Oldpeak", "ST_SlopeDownFlat",
                                "ExerciseAngina_fct", "Age", "RestingBP", "HeartDisease")]
heart_step_clog <- iv_heart[,c("Sex_fct", "ChestPainASY", "Oldpeak", "ST_SlopeDownFlat",
                               "ExerciseAngina_fct", "MaxHR", "RestingBP", "HeartDisease")]

step_model_logit <- gamlss(formula = HeartDisease ~ .,
                           family = BI(mu.link = logit), data = heart_step_logit, type = "response")
step_model_probit <- gamlss(formula = HeartDisease ~ .,
                            family = BI(mu.link = probit), data = heart_step_logit, type = "response")
step_model_clog <- gamlss(formula = HeartDisease ~ .,
                          family = BI(mu.link = cloglog), data = heart_step_clog, type = "response")

# Resultado do modelo
summary(step_model_logit)
summary(step_model_probit)
summary(step_model_clog)
```

```r
options(warn=-1)
# 10 fold cross validation of model
n <- dim(heart_step)[1]
k = 10
set.seed(2, sample.kind = "Rounding")
groups <- c(rep(1:k,floor(n/k)),1:(n-floor(n/k)*k))
set.seed(3, sample.kind = "Rounding")
cvgroups <- sample(groups,n)
predictvalsGLM <- rep(-1,n)
for (i in 1:k) {
  groupi <- (cvgroups == i)
  fit = gamlss(formula = HeartDisease ~ ., family = BI(mu.link = logit), data = heart_step[!groupi,])
  predictvalsGLM[groupi] = predict(object = fit, newdata = heart_step[groupi,], type = "response")
}

PROC_obj <- roc(predictor = predictvalsGLM, response=dt$HeartDisease,
                curve=TRUE)

auc_value <- auc(dt$HeartDisease, predictvalsGLM)
auc_value

plot(PROC_obj, main = "Curva ROC para o modelo ajustado")
mtext(paste('AUC = ',round(auc_value,3)),side = 1, line = -1)

hist(predictvalsGLM, main = "Histograma dos valores preditos")



#Find the best threshold value
probRng <- 20:80
errorRateMtx <- matrix(nrow = length(probRng), ncol = 4)
colnames(errorRateMtx) <- c("Threshold","ErrorRate","FalsePositive","FalseNegative")

for (i in 1:length(probRng)) {
  threshold <- probRng[i]/100
  PredictedHDGLM <- rep(0, n)
  PredictedHDGLM[predictvalsGLM >= threshold] <- 1

  tblGLM <- table(PredictedHDGLM, dt$HeartDisease)
  errorRate <- (tblGLM[1,2]+tblGLM[2,1])/n
  falsePos <-  tblGLM[2,1]/(tblGLM[2,2]+tblGLM[2,1])
  falseNeg <- tblGLM[1,2]/(tblGLM[1,1]+tblGLM[1,2])

  errorRateMtx[i,1] <- threshold
  errorRateMtx[i,2] <- errorRate
  errorRateMtx[i,3] <- falsePos
  errorRateMtx[i,4] <- falseNeg
}

plot(y = errorRateMtx[,2], x = errorRateMtx[,1], col = "red", pch = 20, ylim = c(0.1,0.2))
lines(errorRateMtx[,3], x = errorRateMtx[,1], col = "green", lty = 2)
lines(errorRateMtx[,4], x = errorRateMtx[,1], col = "blue", lty = 2)
legend("topleft", legend = c("Total error", "False positive","False Negative"), col=c("red", "green", "blue"),
lty=c(20,2,2), cex=0.8)

minerror <- min(errorRateMtx[,2])
#Possíveis treshold
possible_thresh <- errorRateMtx[errorRateMtx[,2]==minerror,1]
#Erro mínimo
errorRateMtx[errorRateMtx[,2]==minerror,2]

#menor falso negativo com menor erro total
minfn <- min(errorRateMtx[errorRateMtx[,1] %in% possible_thresh,4])
threshold <- errorRateMtx[errorRateMtx[,4]==minfn,1]

Predicted <- rep(0, n)
Predicted[predictvalsGLM >= threshold] <- 1


confusionMatrix(data=as.factor(Predicted), reference = dt$HeartDisease)
```