Final\_Prediction\_model

Tom Kugener

3/28/2022

# R Markdown including relevant code

## Install and load required functions & import data

## We were not sure about the best way to include our code so we did an R markdown file. It does not include all the code we used but the most relevant steps should included.   
## LASSO and cubic splines were both performed but not included in the main analysis due to uncertainties or results that did not differ from the main analysis.  
  
# Install and load necessary packages   
  
source("Functions.R")  
list.of.packages <- c("haven", "dplyr", "reshape","reshape2", "httr","kableExtra", "ggplot2", "GGally", "epiDisplay", "tidyverse", "caret", "leaps", "arsenal", "skimr", "MASS", "rms", "rmda", "glmnet")   
installRequiredPackages(list.of.packages)  
  
  
# Import Data  
pathToData <- '.'   
pancreatitis <- read.csv( file.path(pathToData,'pancreatitis.csv') )

## Descriptive statistics

# To allow summary statistics to work, the level of measurement of the variables will be defined  
recoded\_pancreatitis <- dplyr::mutate(pancreatitis,  
 site = as.factor(site),  
 age = as.numeric(age),  
 risk = as.numeric(risk),  
 gender = as.factor(gender),  
 outcome = as.factor(outcome),  
 sod = as.factor(sod),  
 pep = as.factor(pep),  
 recpanc = as.factor(recpanc) ,  
 psphinc = as.factor(psphinc),  
 precut = as.factor(precut),  
 difcan = as.factor(difcan),   
 pneudil = as.factor(pneudil),  
 amp = as.factor(amp),  
 paninj = as.factor(paninj),  
 acinar = as.factor(acinar),  
 brush = as.factor(brush),  
 asa81 = as.factor(asa81),  
 asa325 = as.factor(asa325),  
 asa = as.factor(asa),  
 prophystent = as.factor(prophystent),   
 therastent = as.factor(therastent),   
 pdstent = as.factor(pdstent),   
 sodsom = as.factor(sodsom),  
 bsphinc = as.factor(bsphinc),  
 bstent = as.factor(bstent),   
 chole = as.factor(chole),  
 pbmal = as.factor(pbmal),  
 train = as.factor(train),   
 status = as.factor(status),  
 type = as.factor(type),  
 rx = as.factor(rx),  
 bleed = as.factor(bleed)   
 )  
  
  
t3 <- tableby(~ ., data = recoded\_pancreatitis)  
summary(t3, text = TRUE, title = "Descriptive Statistics on Pancreatitis")

## Variable Selection (p-value)

# Model 1: with all pre-selected variables  
model\_rms\_p1 <- lrm(data = pancreatitis, outcome ~ rx + age + acinar + amp +  
 pep + train+ gender + difcan + recpanc + sod+ therastent,  
 x = TRUE, y = TRUE)  
model\_rms\_p1  
  
# Model 2 recpanc = excluded  
model\_rms\_p2 <- lrm(data = pancreatitis, outcome ~ rx + age + acinar + amp +  
 pep + train+ gender + difcan + sod+ therastent,  
 x = TRUE, y = TRUE)  
model\_rms\_p2  
  
# Model 3 gender = excluded  
model\_rms\_p3 <- lrm(data = pancreatitis, outcome ~ rx + age + acinar + amp +  
 pep + train+ difcan + sod+ therastent,  
 x = TRUE, y = TRUE)  
model\_rms\_p3  
  
# Model 4 therastent = excluded  
model\_rms\_p4 <- lrm(data = pancreatitis, outcome ~ rx + age + acinar + amp +  
 pep + train+ difcan + sod,  
 x = TRUE, y = TRUE)  
model\_rms\_p4  
  
# Final model  
model\_rms\_pf <- lrm(data = pancreatitis, outcome ~ rx + age + acinar + amp +  
 pep + train+ difcan + sod,  
 x = TRUE, y = TRUE)  
model\_rms\_pf

### Variable selection (AIC)

model\_p1 <- glm(outcome ~ rx + age + acinar + amp +  
 pep + train+ gender + difcan + recpanc + sod+ therastent,  
 family = binomial, data = pancreatitis)  
model\_p1  
  
step.model <- stepAIC(model\_p1, direction = "backward",   
 trace = FALSE)  
summary(step.model)  
  
## Would keep therastent in th model

# LASSO

library(glmnet)  
x <- data.matrix(pancreatitis[,c('rx', 'age', 'gender' ,'amp',  
 'pep', 'train', 'chole' ,'difcan' ,'recpanc' , 'sod', 'pdstent')])  
y <- pancreatitis$outcome  
lasso\_model <- cv.glmnet(x,y, alpha = 1)  
best\_lambda <- lasso\_model$lambda.min  
best\_lambda  
plot(lasso\_model)

coef(lasso\_model)  
best\_model <- glmnet(x, y, alpha = 1, lambda = best\_lambda) ## the best lambda is really small and therefore does not lead to the exclusion of variables  
coef(best\_model)  
  
best\_model <- glmnet(x, y, alpha = 1, lambda = 0.005)  
coef(best\_model)  
  
best\_model <- glmnet(x, y, alpha = 1, lambda = 0.01)  
coef(best\_model)

## ROC Curve

library(pROC) # library for ROC curve  
p <- predict(model\_rms\_pf, type = "fitted") # prediction factor  
  
ROC <- roc(pancreatitis$outcome, p, ci = TRUE)  
ROC # for AUC value  
plot(ROC)

## Bootstrap validation (final model only)

validation\_rms <- validate(model\_rms\_pf, method= "boot", B=200)  
validation\_rms  
  
plot(validation\_rms, B=200)

## Bootstrap validation (including backward regression)

validation\_rms\_stepwise <- validate(model\_rms\_p1, method = "boot", B=200, bw = TRUE, rule = "p", sls = 0.05)  
validation\_rms\_stepwise  
plot(validation\_rms\_stepwise, B=200)

# Same conclusion as initial backward regression

## C statistic

# Bootstrap of final model  
cstatapp <- 0.5\*(validation\_rms[1,]+1)  
cstatapp  
  
# Bootstrap including backward regression  
cstatapp <- 0.5\*(validation\_rms\_stepwise[1,]+1)  
cstatapp

## Model calibration

# For final model  
cal <- calibrate(model\_rms\_pf, method = "boot", B = 200)  
plot(cal)

# Including backward regression   
cal <- calibrate(model\_rms\_p1, bw = TRUE, rule = "p", sls = 0.05, method = "boot", B = 200)  
plot(cal)

## Using restricted cubic splines for continuous variable age

library(plyr)  
library(dplyr)  
library(ggplot2)  
library(janitor)  
library(rms)  
library(rmda)  
library(arsenal)  
library(pROC)  
# changing outcome to integer to get an average later  
  
class(pancreatitis$outcome) = "integer"  
class(pancreatitis$age) = "integer"  
  
# creating age x outcome frequency table  
  
table.age.pancreatitis <- tabyl(pancreatitis, age, outcome, digits = 0)  
table.age.pancreatitis  
  
  
# adding columns for total and incidence rate  
age.pancreatitis.df <- as.data.frame(table.age.pancreatitis)  
df.total <- age.pancreatitis.df$`0`+age.pancreatitis.df$`1`  
df.percentage <- age.pancreatitis.df$`1` / df.total  
age.pancreatitis.df$total <- df.total  
age.pancreatitis.df$incidence <- df.percentage  
  
# plotting incidence x age, with n for frequency transforming that table into a dataframe  
  
age.pancreatitis.df <- as.data.frame(table.age.pancreatitis)  
View(age.pancreatitis.df)  
age.outcome.plot <- ggplot(age.pancreatitis.df,   
 aes(x = age,   
 y = df.percentage,  
 size = df.total)) + geom\_count()  
?ggplot  
age.outcome.plot # we see a nonlinear relationship with several 'knots'

# let's try three rcs models for age, with knots = 3, 4, 5  
  
kx <- 5 ## easier to calculate, fill knots here (min. 3, max 16)  
  
model.kx <- lrm(data = pancreatitis, outcome~rcs(age,kx) + rx + acinar + amp +  
 pep + train+ difcan + sod,  
 x = TRUE, y = TRUE)  
model.kx

### ROC curves for each model

# ROC curves and AUC values for each model  
  
p.kx <- predict(model.kx, type = "fitted")  
  
ROC.kx <- roc(pancreatitis$outcome, p.kx, ci = TRUE)  
  
ROC.kx   
## k=3 -> AUC = 0.7301 95% CI: 0.6944-0.7658 , equal to non-rcs model  
## k=4 -> AUC = 0.732595% CI: 0.6969-0.7681  
## k=5 -> 0.7331 95% CI: 0.6969-0.7693 use this.   
  
# AUC keeps getting better with more knots, but!  
# literature/'rule of thumb' is to use max 5 knots, depending on sample size  
# large sample sizes should use 5; more knots can lead to overfitting  
# k=10 AUC 0.7467  
# k=16 AUC 0.7526 (highest possible knot value)  
  
# altering the final model, adding 5 knots for age  
  
  
model\_rms.rcs\_pf <- lrm(data = pancreatitis, outcome ~ rx + rcs(age,5) + amp +  
 pep + train+ acinar + difcan + sod,  
 x = TRUE, y = TRUE)   
model\_rms.rcs\_pf

## ROC curve final model

library(pROC)   
px <- predict(model\_rms.rcs\_pf, type = "fitted")  
  
ROCx <- roc(pancreatitis$outcome, px, ci = TRUE)  
ROCx  
plot(ROCx)

## Bootstrap Validation

validation\_rms.rcs <- validate(model\_rms.rcs\_pf, method= "boot", B=200)  
validation\_rms.rcs  
  
# C-statistic  
0.5 \* (validation\_rms.rcs[1, ] + 1)  
  
  
plot(validation\_rms.rcs, B=200)

## Calibration of the model

calx <- calibrate(model\_rms.rcs\_pf, B = 200)  
plot(calx)

## Subgroup Analysis

#data divided by risk  
  
df1 <- pancreatitis[pancreatitis$risk <= 2, ]  
df2 <- pancreatitis[pancreatitis$ris > 2, ]  
  
  
modelrisk1 <- lrm(data = df1, outcome ~ rx + age + acinar + amp +  
 pep + train+ difcan + sod,  
 x = TRUE, y = TRUE)  
modelrisk1  
  
modelrisk2 <- lrm(data = df2, outcome ~ rx + age + acinar + amp +  
 pep + train+ difcan + sod,  
 x = TRUE, y = TRUE)  
modelrisk2  
## roc for risk 1 group  
library(pROC)  
prisk1 <- predict(modelrisk1, type = "fitted")  
  
ROCrisk1 <- roc(df1$outcome, prisk1, ci = TRUE)  
ROCrisk1  
plot(ROCrisk1)

# Validation with bootstrapping = 200  
  
validation\_risk1 <- validate(modelrisk1, method= "boot", B=200)  
validation\_risk1  
0.5 \* (validation\_risk1[1, ] + 1)  
  
## roc for risk 2 group  
  
prisk2 <- predict(modelrisk2, type = "fitted")  
  
ROCrisk2 <- roc(df2$outcome, prisk2, ci = TRUE)  
ROCrisk2  
plot(ROCrisk2)

# Validation with bootstrapping = 200  
  
validation\_risk2 <- validate(modelrisk2, method= "boot", B=200)  
validation\_risk2  
0.5 \* (validation\_risk2[1, ] + 1)  
  
nrow(df2)