TREBALL DE FI DE GRAU

Grau en Estadística

Títol: REGRESSIÓ LOGÍSTICA PENALITZADA

Autor: Montserrat Muñoz Aragón

Director: Francesc Carmona Pontaque

Departament: Genètica, Microbiologia i Estadística

Convocatòria: Juny 2020

UNIVERSITAT DE BARCELONA

UNIVERSITAT POLITÈCNICA DE CATALUNYA
BARCELONATECH
Facultat de Matemàtiques i Estadística

RESUM

Actualment, un dels temes d'interès en el món de l'estadística és el *Big Data*. A l'hora d'estimar un model estadístic amb moltes variables poden sorgir alguns problemes com la multicol·linealitat i la manca d'eficiència, entre d'altres. Tot això, comporta que hi hagi dubtes sobre les estimacions dels paràmetres pels mètodes tradicionals i per aquest motiu s'utilitzen els mètodes de penalització, en concret, en aquest estudi se'n parlarà de tres: *Ridge Regression, Lasso Regression* i *Elastic Net Regression*.

Aquests mètodes van ser creats a partir de l'estimació per mínims quadrats dels Models Lineals, però el tipus de model que s'analitzarà en aquest treball és la regressió logística, la qual forma part dels Models Lineals Generalitzats (MLG). Com els MLG s'estimen mitjançant l'estimador de màxima versemblança, aquests mètodes esmentats s'aplicaran d'una manera generalitzada.

PARAULES CLAU:

Regressió logística, màxima versemblança, estimació, coeficients, variables explicatives, paràmetres, *Stepwise*, *Ridge regression, Lasso regression, Elastic Net regression*.

ABSTRACT

Nowadays, the Big Data is one of a kind issues of interests in the statistics world. Problems as multicollinearity and lack of efficiency, among others, can appear when we pretend to estimate a statistical model with a huge number of variables. By the traditional methods this kind of problems cause doubts about the parameter's estimation, this is why we use the penalized methods. In this analysis we are going to focus on three of them: Ridge Regression, Lasso Regression and Elastic Net Regression.

These methods were created based on the least square estimation of the Lineal Models, but the type of model that we are going to analyze in this project is the logistic regression, which is part of the Generalized Linear Models (GLM). These GLM are estimated by the maximum likelihood estimation, so we will applicate these methods in a general way.

KEY WORDS:

Logistic regression, maximum likelihood, estimation, coefficients, predictor variables, parameters, Stepwise, Ridge regression, Lasso regression, Elastic Net regression.

CLASSIFICACIÓ AMS

- 62J05 *Linear regression*
- 62J12 *Generalized linear models*
- 62J07 *Ridge regression; shrinkage estimator*

ÍNDEX

1. INTRODUCCIÓ

Avui en dia, les empreses, els governs i les organitzacions tenen la necessitat d'obtenir informació de tot el que sigui possible i guardar-la en grans bases de dades. A l'hora d'estimar un model estadístic, una de les preguntes a plantejar és: Quines variables han de formar part del model?

Si la base de dades que s'està estudiant conté diversos predictors segurament hi haurà un augment en l'ajust de les dades, però pot sorgir el problema de la multicol·linealitat i, per tant, hi haurà un increment en la variància de les estimacions. Per a resoldre aquests problemes estan el que s'anomenen mètodes de penalització. S'analitzaran tres en concret: *Ridge Regression*, *Lasso Regression* i *Elastic Net Regression*.

En aquest treball, s'estudiarà la regressió logística, a la qual se li aplicaran els mètodes de selecció de variables tradicionals i els tres mètodes mencionats anteriorment, però aplicant una generalització d'aquests. Això és degut a que la regressió logística s'estima mitjançant l'estimador de màxima versemblança en comptes de l'estimació per mínims quadrats.

1.1. OBJECTIUS DEL TREBALL

Els objectius d'aquest estudi consisteixen en presentar la regressió logística i aplicar a aquesta els mètodes de penalització, en concret, la regressió *Ridge, Lasso* i *Elastic Net*. Per tant, el que s'analitzarà és la regressió logística penalitzada.

Una vegada explicats aquests conceptes, s'aplicaran els mètodes esmentats a dues bases de dades per veure si funcionen millor que els mètodes tradicionals.

1.2. METODOLOGIA I ESTRUCTURA

Aquest treball està estructurat en dues parts: la primera és purament teòrica on es presentaran els mètodes, i la segona és la part pràctica on s'aplicaran els mètodes mencionats anteriorment.

Pel que fa a l'índex, la part teòrica comprèn del punt 2 al 4. En el segon punt s'explica que són els Models Lineals Generalitzats, com s'estimen els seus coeficients, alguns mètodes de mesura de bondat d'ajust, l'estimació del paràmetre de dispersió, els diversos tipus de residus a poder analitzar i per últim el test *Anova* per a aquests tipus de models.

En el punt 3 s'explica en que consisteix la regressió logística, la funció *link*, com s'interpreten els coeficients sota aquesta funció, quin mètode s'utilitza per a l'estimació dels seus paràmetres, diverses mesures per a avaluar la capacitat predictiva d'un model i els residus que es poden estudiar.

Per últim, en el quart punt s'explica els mètodes tradicionals de selecció de variables i la regressió logística penalitzada, en la qual s'introdueixen els tres mètodes a analitzar, les característiques de cadascun d'ells i com s'estima el paràmetre λ.

La segona part del treball és el punt 5 de l'índex on es farà un estudi de dos *datasets* amb el software *R*. En cada una d'elles es realitzarà un estudi descriptiu de les variables, s'aplicaran els mètodes esmentats i es farà una petita conclusió dels resultats obtinguts.

En el darrer punt s'exposen les conclusions a les que s'ha arribat a partir de l'anàlisi de les dues bases de dades.

Pel que fa als *datasets,* s'han obtingut a la pàgina web *Kaggle* i es poden trobar en els següents links:

- https://www.kaggle.com/mirlei/hcc-survival-data-set

- https://www.kaggle.com/sulianova/cardiovascular-disease-dataset

La primera base de dades tracta sobre el carcinoma hepatocel·lular (càncer de fetge), conté 50 variables i 165 observacions. La segona, informa sobre persones que presenten o no alguna malaltia cardiovascular, la qual consta de 12 atributs i 70.000 individus.

1.3. AGRAÏMENTS

Gràcies al meu tutor, Francesc Carmona Pontaque, per tota la ajuda proporcionada i resoldre tots els dubtes que m'han anat sorgint durant el treball de la manera més ràpida i eficient possible tot i la situació que s'està vivint.

2. MODELS LINEALS GENERALITZATS

2.1. INTRODUCCIÓ ALS MODELS LINEALS

Els models lineals estan definits per la següent estructura:

$$y_i = \beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \cdots + \beta_k x_{ik}\, \epsilon_i \quad i = 1, \dots, n \quad k > 1\ variables\ independents$$

On *Y* (variable resposta) és igual a un valor fix (η) més una variable aleatòria (ϵ):

$$Y = \eta + \epsilon$$

S'està assumint que les dades segueixen un patró lineal, però com que l'ajust no és exacte i presenta errors, s'afegeix el terme ϵ_i (errors aleatoris). Aquests errors són la part no controlable del model degut a causes aleatòries que presenten les dades.

L'objectiu és estimar els coeficients β_j, els quals s'estimen mitjançant el mètode dels mínims quadrats. Aquest mètode consisteix en calcular tots aquells valors β_j que minimitzen els errors (Carmona, 2001).

$$\min_{\beta} \sum_{i=1}^{n} \epsilon_i^2 = \sum_{i=1}^{n} (y_i - (\beta_0 + \beta_1 x_{i1} + \dots + \beta_k x_{ik}))^2$$

Si es deriva aquesta expressió i s'iguala a 0 s'obtenen els estimadors MQ (mínims quadrats).

Aquest mètode proporciona els millors estimadors lineals i unes bones prediccions si es compleixen les condicions de Gauss-Markov. Aquestes condicions són les següents:

1) Els errors aleatoris han de tenir una esperança de 0: $E(\epsilon_i) = 0 \quad i = 1, \dots, n$
2) La variància dels errors aleatoris ha de ser constant (homoscedasticitat): $Var(\epsilon_i) = \sigma^2 \quad i = 1, \dots, n$
3) Els errors aleatoris no han d'estar correlacionats: $E(\epsilon_i \cdot \epsilon_j) = 0 \quad \forall\, i \neq j$

En algunes ocasions es troba que les dades no compleixen alguna/es d'aquestes condicions. Aquests problemes es poden solucionar aplicant transformacions a la variable resposta (com per exemple el logaritme), però no sempre s'aconsegueix que es corregeixi la falta de normalitat, l'homoscedasticitat o la linealitat entre les dades. A més, segons la transformació que s'apliqui a les dades pot dificultar la interpretació d'aquestes.

També, pot ser que es doni el cas de que la variable resposta *Y* sigui:

- Una variable de recompte, per exemple el número de morts per malalties cardiovasculars.

- Una variable qualitativa (binària), per exemple si una persona té una malaltia cardiovascular o no.
- Una variable per expressar proporcions, per exemple la proporció de persones amb malalties cardiovasculars.

En el cas de que la variable resposta sigui un recompte que s'expressa amb nombres enters, la variància podria anar augmentant linealment amb la mitjana. Si la variable explicada són proporcions, la variància tindria forma de U invertida amb relació amb la mitjana.

Per tant, una alternativa als Models Lineals seria utilitzar els Models Lineals Generalitzats (MLG).

2.2. EXPLICACIÓ MODELS LINEALS GENERALITZATS

Els models lineals generalitzats són una extensió dels models lineals que permet que la variable resposta *Y* tingui models de distribució diferents d'una normal.

Els MLG tenen tres components (McCullahg, Nelder, 1989):

- La component aleatòria que correspon a la distribució de probabilitat de la variable *Y*, la qual segueix una distribució de la família exponencial.
- La component sistemàtica, també anomenada predictor lineal (η), que especifica les variables explicatives ($X_1, X_2, \dots, X_k$) del model i la combinació lineal de paràmetres desconeguts (β_j): $\eta = X'\beta$.
- La funció *link,* $g(\mu)$, especifica la relació entre la component aleatòria i la sistemàtica. Té la funció de linealitzar la relació entre el valor esperat de la variable resposta *Y*, *E(Y)*, i el predictor lineal de les variables explicatives: $g(\mu) = \eta$.
  Per exemple, si la distribució és Binomial (1,p), les funcions *link* més habituals són el model lògit o pròbit.

Els supòsits que es fan sobre els MLG són els següents:

- Les dades $y_1, y_2, \dots, y_n$ estan distribuïdes independentment.
- La variable Y no necessita estar distribuïda normalment.
- No assumeixen una relació lineal entre la variable depenent i les independents, però sí entre la funció *link* i les variables explicatives. Per exemple, la regressió logística binària: $logit(\pi) = \beta_0 + \beta X$.
- La homogeneïtat de la variància no és necessari que es satisfaci. De fet, moltes vegades no és ni possible degut a l'estructura del model, a més, pot ser que hi hagi sobre dispersió (la variància observada és major de lo que suposa el model).
- Els errors han de ser independents però no distribuïts normalment.

- S'utilitza l'estimació de màxima versemblança en comptes dels mínims quadrats per a estimar els paràmetres β_j.

S'assumeix que cada component *Y* segueix una distribució de la família exponencial i la funció de densitat és (McCullahg, Nelder, 1989):

$$f_Y(y;\theta,\phi) = e^{\frac{y\theta - b(\theta)}{a(\phi)} + c(y,\phi)}$$

On θ és el paràmetre canònic, $\Phi = a(\phi) = \frac{\phi}{w}$ (w és un pes que varia a cada observació) és el paràmetre de dispersió i $\sqrt{\Phi}$ és el paràmetre d'escala. Per tant, la funció de log-versemblança és:

$$l(\theta,\phi;y) = log\, f_Y(y;\theta,\phi) = \frac{y\theta - b(\theta)}{\Phi} + c(y,\phi)$$

Per a obtenir l'esperança es deriva respecte el paràmetre θ (McCullahg, Nelder, 1989):

$$\frac{\partial l}{\partial \theta} = \frac{y - b'(\theta)}{\Phi}$$

Per tant,

$$E\left(\frac{\partial l}{\partial \theta}\right) = \frac{E(y) - b'(\theta)}{\Phi} = 0 \Longrightarrow E(Y) = b'(\theta) \Longrightarrow \mu = b'(\theta)$$

Si es calcula la segona derivada de la funció de versemblança:

$$\frac{\partial^2 l}{\partial \theta^2} = \frac{-b"(\theta)}{\Phi}$$

Es té que,

$$E\left(\frac{\partial^2 l}{\partial \theta^2}\right) + E\left[\left(\frac{\partial l}{\partial \theta}\right)^2\right] = 0$$

D'on s'obté:

$$\frac{-b"(\theta)}{\Phi} = -\frac{E((y - b'(\theta))^2)}{\Phi^2} \Longrightarrow \frac{-b"(\theta)}{\Phi} = -\frac{Var(Y)}{\Phi^2} \Longrightarrow Var(Y) = b"(\theta)\Phi$$

La funció $b"(\theta)$ depèn del paràmetre θ i, per tant, de la mitjana. S'anomenarà funció de variància:

$$V(\mu) = b"(\theta)$$

En canvi, $a(\phi)$ només depèn del paràmetre de dispersió que normalment és constant.

Per totes les famílies, la variància del model té la següent expressió:

$$V(Y) = \phi V(\mu)$$

Cada distribució té el seu *link* canònic que és el que fa que es compleixi la següent condició:

$$\eta = \theta = X'\beta$$

2.2.1. ESTIMACIÓ DELS COEFICIENTS

Com ja s'ha mencionat anteriorment, l'estimació dels paràmetres β_j s'obtenen amb l'estimació de màxima versemblança. Els elements que es necessitaran són:

- El paràmetre $\mu = b'^{-1}(\theta)$.
- El predictor lineal $\eta = X'\beta$.
- La funció *link* $g(\mu) = \eta$.

Es parteix de l'expressió:

$$l(\theta, \phi; y) = log\, f_Y(y; \theta, \phi) = \frac{y\theta - b(\theta)}{\Phi} + c(y, \phi)$$

Aplicant la regla de la cadena (McCullahg, Nelder, 1989):

$$\frac{\partial l}{\partial \beta_j} = \frac{\partial l}{\partial \theta}\frac{\partial \theta}{\partial \mu}\frac{\partial \mu}{\partial \eta}\frac{\partial \eta}{\partial \beta_j}$$

A partir d'aquí s'obté que:

$$\frac{\partial l}{\partial \theta} = \frac{y - b'(\theta)}{\Phi}, on\ b'(\theta) = \mu \Longrightarrow \frac{\partial l}{\partial \theta} = \frac{y - \mu}{\Phi}$$

$$\frac{\partial \mu}{\partial \theta} = b"(\theta) = Var(\mu) \Longrightarrow \frac{\partial \theta}{\partial \mu} = \frac{1}{Var(\mu)}$$

$$\frac{\partial \eta}{\partial \mu} = g'(\mu) \Longrightarrow \frac{\partial \mu}{\partial \eta} = \frac{1}{g'(\mu)}$$

$$\frac{\partial \eta}{\partial \beta_j} = x_j$$

I substituint:

$$\frac{\partial l}{\partial \beta_j} = \frac{y - \mu}{\Phi Var(\mu)}\frac{x_j}{g'(\mu)}$$

Les estimacions de màxima versemblança s'obtenen resolvent les equacions per als paràmetres β_j.

2.2.2. MESURES DE BONDAT D'AJUST

La desviància és una mesura que es pot interpretar de manera anàloga a la suma residual dels mínims quadrats en els models lineals.

L'expressió de la desviància escalada és la següent (McCullahg, Nelder, 1989):

$$D^2(y,\hat{\mu},\phi) = 2\big(l(y,\phi;y) - l(\hat{\mu},\phi;y)\big) = 2\sum_{i=1}^{N}\left(\frac{\tilde{\theta}_i y_i - b(\tilde{\theta}_i)}{\phi} - \frac{\hat{\theta}_i y_i - b(\hat{\theta}_i)}{\phi}\right) \sim \chi^2_{N-K}$$

És l'estadístic de contrast del test de raó de versemblança per a comparar un model simple amb un altre de més complex.

I la no escalada és la desviància escalada multiplicada per el paràmetre ϕ (McCullahg, Nelder, 1989):

$$D(y,\hat{\mu}) = \phi D^2(y,\hat{\mu},\phi) = 2\sum_{i=1}^{N}\Big(\big(\tilde{\theta}_i y_i - b(\tilde{\theta}_i)\big) - \big(\hat{\theta}_i y_i - b(\hat{\theta}_i)\big)\Big)$$

La desviància del model s'anomena *Residual deviance* que entre més petita millor. La del model nul s'anomena *Null deviance* que és el valor màxim que pot obtenir la desviància.

La següent expressió es pot considerar un equivalent del coeficient de determinació R^2:

$$\frac{Null\ deviance - Residual\ deviance}{Null\ deviance} = 1 - \frac{Residual\ deviance}{Null\ deviance} \in [0,1]$$

L'estadístic de Pearson generalitzat χ^2 és un altre mesura de la bondat d'ajust del model, que es defineix com (McCullahg, Nelder, 1989):

$$\chi^2 = \sum_{i=1}^{N}\frac{(y_i - \hat{\mu}_i)^2}{V(\hat{\mu}_i)} \Longrightarrow \frac{\chi^2}{\phi} \sim \chi^2_{N-K}$$

2.2.3. ESTIMACIÓ DEL PARÀMETRE DE DISPERSIÓ

Estimar el paràmetre de dispersió ϕ per màxima versemblança té dificultats, però una vegada ja s'han calculat les estimacions de $\widehat{\beta_j}$, hi ha dues opcions:

- A partir de l'estadístic de Pearson generalitzat χ^2:

$$\frac{\chi^2}{\phi} \sim \chi^2_{N-K} \Longrightarrow E\left[\frac{\chi^2}{\phi}\right] = N - K \Longrightarrow \hat{\phi}_{pearson} = \frac{\chi^2}{N-K}$$

- A partir de la *deviance* $\mathrm{D}(\mathrm{y},\hat{\mu})$:

$$\frac{D(y,\hat{\mu})}{\phi} \sim \chi^2_{N-K} \Longrightarrow E\left[\frac{D(y,\hat{\mu})}{\phi}\right] = N - K \Longrightarrow \hat{\phi}_{deviance} = \frac{D(y,\hat{\mu})}{N-K}$$

2.2.4. RESIDUS

A partir d'aquests tres mètodes, es poden calcular els residus d'un model lineal generalitzat (McCullahg, Nelder, 1989):

- Residus resposta, els quals són intuïtius però poc útils per als MLG:

$$r_i = y_i - \hat{\mu}_i$$

- Residus de Pearson, que també són intuïtius. És com el de resposta però homogeneïtzant les variàncies:

$$r_i^P = \frac{y_i - \hat{\mu}_i}{\sqrt{Var(\hat{\mu}_i)\phi}}$$

Es compleix que:

$$\sum_{i=1}^{N} (r_i^P)^2 = \chi^2$$

- Residus de la *deviance*, els quals són poc intuïtius perquè s'obtenen de la desviància, la qual ve de la màxima versemblança i té les variàncies homogeneïtzades.

$$r_i^P = sign(y_i - \hat{\mu}_i) d_i, on\ d_i = \sqrt{2\left(\left(\tilde{\theta}_i y_i - b(\tilde{\theta}_i)\right) - \left(\hat{\theta}_i y_i - b(\hat{\theta}_i)\right)\right)}$$

Es compleix que:

$$\sum_{i=1}^{N} (r_i^D)^2 = \sum_{i=1}^{N} {d_i}^2 = D(y, \hat{\mu}_i)$$

2.2.5. TEST ANOVA

El test òmnibus es basa en la comparació de dos models encaixats mitjançant el test de la raó de versemblança. Es tenen dos models: m_1 amb K_1 paràmetres i m_2 amb K_2. El model 2 té més paràmetres que el 1 i els dos contenen *N* dades. El test de raó de versemblança en general es contrasta amb (Cortés, Valero, 2019):

$$\wedge = 2(l_2 - l_1) \sim \chi^2_{K_2 - K_1}$$

En el models lineals generalitzats:

$$\wedge = \frac{deviance_1 - deviance_2}{\phi} = \frac{diferència\ de\ deviances}{\phi}$$

Hi ha dues possibilitats:

1) El paràmetre de dispersió ϕ és conegut com en el cas de les distribucions Binomial, Poisson... Aleshores,

$$\Lambda = \frac{diferència\ de\ deviances}{\phi} \sim \chi^2_{K_2-K_1}$$

2) El paràmetre de dispersió ϕ és desconegut com en el cas de la distribució Normal, Gamma... Aleshores,

$$\Lambda = \frac{diferència\ de\ deviances}{\phi}$$

On la diferència de desviàncies $\sim\chi^2_{K_2-K_1}$ i $\hat{\phi} \sim \frac{\chi^2_{N-K_2}}{N-K_2}$ ($\hat{\phi}$ tant pot ser la de Pearson com la de la *deviance*).

$$F = \frac{\Lambda}{K_2 - K_1} \sim \frac{\frac{\chi^2_{K_2-K_1}}{K_2-K_1}}{\frac{\chi^2_{N-K_2}}{N-K_2}} \sim F(K_2 - K_1, N - K_2)$$

En aquest estudi com que s'analitzarà la regressió logística binària es té el primer cas: el paràmetre de dispersió ϕ és conegut (pren un valor de 1).

3. REGRESSIÓ LOGÍSTICA

3.1. DISTRIBUCIÓ DE BERNOULLI I BINOMIAL

La variable dependent Y és binària, per tant, només pren dos valors (1 o 0):

$$Y = \begin{cases} 1, & amb\ probabilitat\ \pi \\ 0, & amb\ probabilitat\ 1-\pi \end{cases}$$

Hi ha dues possibilitats de representació de la base de dades:

1) Dades desagregades: la unitat és cada observació (individu) i cada una té la seva pròpia sortida (1 o 0). Llavors, Y segueix una distribució de Bernoulli $Y \sim Ber(\pi)$ amb la següent funció de probabilitat (McCullahg, Nelder, 1989):

   $$Pr(Y = y) = \pi^y (1-\pi)^{1-y}$$

   El valor esperat és:

   $$E(Y) = \mu = 1\,Pr(Y = 1) + 0 Pr(Y = 0) = \pi$$

   I la variància:

   $$Var(Y) = \sigma^2 = E(Y^2) - E(Y)^2 = \pi - \pi^2 = \pi(1-\pi)$$

   Per tant, la variància no és constant i depèn de la probabilitat del succés.

2) Dades agregades: la unitat és cada classe de covariable i cada classe està definida per el nombre d'individus (m_k) i el nombre d'èxits (y_k). Aleshores, Y segueix una distribució Binomial $Y \sim Bin(m, \pi)$ amb la següent funció de probabilitat (McCullahg, Nelder, 1989):

   $$P(Y = y) = \binom{m}{y} \pi^y (1-\pi)^{m-y}$$

   L'esperança és:

   $$E[Y] = \mu = m\pi$$

   I la variància:

   $$Var[Y] = m\pi(1-\pi)$$

   De fet, la distribució Bernoulli es pot veure com un cas degenerat de la Binomial ja que es considera que la *m* pren un valor de 1.

3.2. FUNCIÓ LINK

Un cop donada la distribució de probabilitat, cal especificar una funció per modelar les probabilitats π en terme de variables explicatives. El model de probabilitat lineal és (McCullahg, Nelder, 1989):

$$\eta = \sum_{j=1}^{p} x_j \beta_j$$

Però això no garanteix que la resposta sigui un valor que es situï entre el rang [0,1] com la probabilitat d'èxit. Per tant, es necessita una funció *link* que relacioni el vector π amb el predictor lineal η (McCullahg, Nelder, 1989):

$$\eta_i = g(\pi_i) = \sum_{j=1}^{p} x_{ij}\beta_j \quad i = 1, \dots, n$$

Hi ha diverses possibilitats de funció *link* per a variables binàries. Les quatre més comunes són (McCullahg, Nelder, 1989):

- La funció lògit:

$$\eta_i = g(\pi_i) = logit(\pi_i) = log\left(\frac{\pi_i}{1-\pi_i}\right)$$

- La funció pròbit:

$$\eta_i = g(\pi_i) = \phi^{-1}(\pi_i)$$

  On ϕ^{-1} representa la funció inversa de la distribució Normal.

- La funció complementari log-log:

$$\eta_i = g(\pi_i) = log\left(log\left(\frac{1}{1-\pi_i}\right)\right)$$

- Funció log-log:

$$\eta_i = g(\pi_i) = -log\left(log\left(\frac{1}{\pi_i}\right)\right)$$

La més utilitzada és la lògit i és en la que es centrarà aquest estudi, per tant:

$$\eta_i = g(\pi_i) = logit(\pi_i) = log\left(\frac{\pi_i}{1-\pi_i}\right) = \sum_{j=1}^{p} x_{ij}\beta_j \quad i = 1, \dots, n$$

3.3. INTERPRETACIÓ SOTA LA FUNCIÓ LÒGIT

L'odd d'un esdeveniment representa el rati de les probabilitats de que l'esdeveniment succeeixi i de que no succeeixi (McCullahg, Nelder, 1989):

$$odd_i = \frac{\pi_i}{1-\pi_i} \Longrightarrow logodd = log\left(\frac{\pi_i}{1-\pi_i}\right) \quad i = 1, \dots, n$$

Imaginem que el predictor lineal conté dues covariables (x_1, x_2). L'odd de la resposta positiva seria:

$$\frac{\pi_i}{1-\pi_i} = exp(\eta_i) = exp\,(\beta_0 + \beta_1 x_1 + \beta_2 x_2)$$

La probabilitat de la resposta positiva és:

$$\pi_i = g_1^{-1}(\eta_i) = \frac{exp\ (\eta_i)}{1 + exp\ (\eta_i)} = \frac{exp\ (\beta_0 + \beta_1 x_1 + \beta_2 x_2)}{1 + exp\ (\beta_0 + \beta_1 x_1 + \beta_2 x_2)}$$

La probabilitat de la resposta negativa és el complementari de π_i:

$$1 - \pi_i = \frac{1}{1 + exp\ (\eta_i)} = \frac{1}{1 + exp\ (\beta_0 + \beta_1 x_1 + \beta_2 x_2)}$$

Els odds Rati (OR) és el rati dels odds. El OR de la resposta positiva d'una sola variable és l'exponencial del coeficient d'aquesta:

$$OR_i = exp\ (\beta_i)$$

3.4. ESTIMACIÓ DELS PARÀMETRES

La funció de *log-likelihood* per a la distribució binomial presenta la següent forma (McCullahg, Nelder, 1989):

$$l(\pi, y) = \sum_{i=1}^{n} \left[y_i log \left(\frac{\pi_i}{1 - \pi_i} \right) + m_i log\ (1 - \pi_i) \right]$$

Es vol obtenir els valors estimats dels paràmetres β_j. Per tant, aplicant la regla de la cadena a la funció $l(\pi, y)$, s'obté (McCullahg, Nelder, 1989):

$$\frac{\partial l}{\partial \beta_j} = \frac{\partial l}{\partial \pi_i} \frac{\partial \pi_i}{\partial \eta} \frac{\partial \eta}{\partial \beta_j}$$

Calculant les derivades:

$$\frac{\partial l}{\partial \pi_i} = \frac{y_i}{\pi_i(1 - \pi_i)} - \frac{m_i}{1 - \pi_i}$$

$$\frac{\partial \eta}{\partial \beta_j} = x_{ij}$$

En el cas dels models logístics:

$$\pi_i = \frac{exp\ (\eta)}{1 + exp\ (\eta)} \Longrightarrow \frac{\partial \pi_i}{\partial \eta} = \frac{exp(\eta)}{[1 + exp(\eta)]^2} = \pi_i(1 - \pi_i)$$

Substituint els resultats obtinguts:

$$\frac{\partial l}{\partial \beta_j} = \sum_{i=1}^{n} \frac{y_i - m_i \pi_i}{\pi_i(1 - \pi_i)} \pi_i(1 - \pi_i) x_{ij} = \sum_{i=1}^{n} (y_i - m_i \pi_i) x_{ij} = \sum_{i=1}^{n} (y_i - \mu_i) x_{ij}$$

En forma matricial s'escriu:

$$\frac{\partial l}{\partial \beta} = X^T(Y - \mu)$$

Aquesta expressió s'iguala a 0 i mitjançant mètodes iteratius es troba la solució de cada β_j.

3.5. MESURES DE BONDAT D'AJUST

En el cas de la distribució binomial, la desviància escalada es defineix com (Cortés, Valero, 2019):

$$D'(y, \hat{\mu}) = 2l(y, y) - 2l(\hat{\mu}, y)$$

I la desviància és la $D'(y, \hat{\mu})$ multiplicada per el paràmetre de dispersió ϕ, però com ja s'ha mencionat anteriorment, al ser una distribució binomial pren un valor de 1. Per tant:

$$D(y, \hat{\mu}) = D'(y, \hat{\mu})\ \phi = D'(y, \hat{\mu})$$

Amb dades agregades, en el model saturat $l(y, y)$, les probabilitats ajustades són iguals a les observades:

$$\widehat{\pi_i} = \frac{y_i}{m_i} \quad i = 1, \dots, n$$

L'expressió específica de la desviància en el model binomial és la següent (Cortés, Valero, 2019):

$$D(y, \hat{\mu}) = D(y, \hat{\pi}) = 2\sum_{i=1}^{n}\left\{y_i log\left(\frac{y_i}{m_i\widehat{\pi_i}}\right) + (m_i - y_i) log\left(\frac{m_i - y_i}{m_i - m_i\widehat{\pi_i}}\right)\right\}$$

On:

- y_i: els valors observats de la resposta positiva per l'observació "i".
- $m_i - y_i$: els valors observats de la resposta negativa per l'observació "i".
- $m_i\widehat{\pi_i}$: les respostes positives esperades per l'observació "i".
- $m_i - m_i\widehat{\pi_i}$: les respostes negatives esperades per l'observació "i".

En el test de la bondat d'ajust del model *(M)* es contrasten les següents hipòtesis:

$$\begin{cases} H_0: El\ model\ s'ajusta\ bé\ a\ les\ dades \\ H_1: El\ model\ no\ s'ajusta\ bé\ a\ les\ dades \end{cases}$$

La distribució de l'estadístic de *deviance* per el model amb dades agregades sota la H_0 amb *p* paràmetres és:

$$D_M = D(Y, \hat{\pi}) \sim \chi^2_{n-p}$$

Un cop calculat el p-valor $P(\chi^2_{n-p} > D_M)$:

- Si el p-valor és inferior al 0'05 (nivell de significació) hi ha evidència de rebutjar H_0 i, per tant, el model no s'ajusta bé a les dades.
- Si el p-valor és superior al 0'05 (nivell de significació) no hi ha evidència per rebutjar H_0, és a dir, no hi ha evidència de que el model no s'ajusta bé a les dades.

L'estadístic generalitzat de Pearson (X^2) es distribueix asimptòticament com (Cortés, Valero, 2019):

$$X^2 = \sum_{i=1}^{n} \frac{(y_i - m_i\hat{\pi}_i)^2}{m_i\hat{\pi}_i(1-\hat{\pi}_i)} = \sum_{i=1}^{n} \frac{m_i(y_i - \hat{\pi}_i)^2}{\hat{\mu}_i(m_i - \hat{\mu}_i)}$$

Dos mètodes més per a comprovar quin model presenta un millor ajust és el criteri AIC, proposat per *Akaike*, i el criteri BIC, proposat per *Schwartz*, els quals es defineixen com (Cortés, Valero, 2019):

$$AIC_M = 2(p - l(\hat{\pi}_M, y))$$

$$BIC_M = plog(n) - 2l(\hat{\pi}_M, y)$$

Els models que presenten un AIC i BIC inferior són els que presenten un millor ajust.

3.6. AVALUACIÓ DEL MODEL

Per avaluar la capacitat predictiva del model es dividirà el *data set* en *train* i *test.* Es crearà un model utilitzant les observacions que conté les dades *train* i després s'avaluarà el model amb les observacions de les dades *test.*

Es calcula la matriu de confusió, la qual ajuda a mesurar com es comporta el model quan se li aplica una base de dades nova (Zelada, 2017).

Taula 3.1. Matriu de confusió.

		PREDICCIÓ	
		Positius	Negatius
OBSERVACIÓ	Positius	Veritables positius (VP)	Falsos negatius (FN)
	Negatius	Falsos positius (FP)	Veritables negatius (VN)

On:

- VP és la quantitat de positius que s'han classificat correctament com positius.
- VN és la quantitat de negatius que s'han classificat correctament com negatius.
- FN és la quantitat de positius que s'han classificat incorrectament com negatius.
- FP és la quantitat de negatius que s'han classificat incorrectament com positius.

A partir d'aquesta matriu de confusió es poden calcular diverses mesures per a saber com es comporta el model (Zelada, 2017):

- L'exactitud indica el percentatge de les dades classificades correctament.

$$Exactitud = \frac{VP + VN}{Total}$$

- La taxa d'error indica el percentatge de les dades que s'han classificat incorrectament.

$$Taxa\ d'error = \frac{FP + FN}{Total}$$

- La sensibilitat o taxa de veritables positius indica quin percentatge aconsegueix classificar correctament quan la classe és positiva.

$$Sensibilitat = \frac{VP}{Total\ de\ positius}$$

- L'especificitat o taxa de vertaders negatius indica el percentatge que aconsegueix classificar correctament quan la classe és negativa.

$$Especificitat = \frac{VN}{Total\ de\ negatius}$$

- La precisió indica el percentatge classificat correctament quan es prediuen valors positius.

$$Precisió = \frac{VP}{Total\ classificats\ positius}$$

- El valor de predicció negatiu indica quin percentatge classifica correctament quan es prediuen valors negatius.

$$VPN = \frac{VN}{Total\ classificats\ negatius}$$

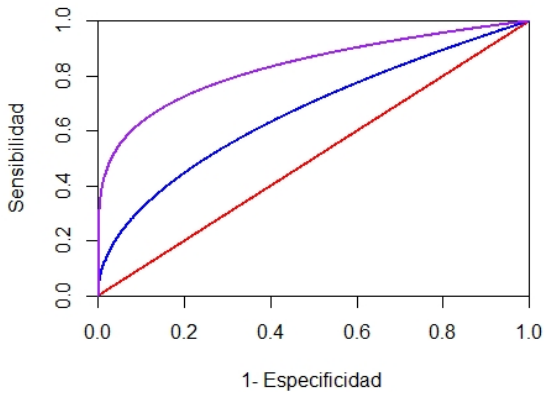
A partir de la matriu de confusió es pot realitzar el gràfic anomenat corba *ROC* on es representa la sensibilitat en funció dels falsos positius (complementari de la especificitat), és a dir, la proporció de casos negatius que es classifiquen com positius, per cada possible punt de tall.

Aquesta corba ens informa del següent:

- Si la proba és perfecte, l'especificitat i la sensibilitat prendrien valor de 1 i, per tant, la corba contindria només el punt (0,1).
- Si la proba és ineficaç, la sensibilitat és igual a la proporció de falsos positius i conseqüentment, la corba seria la diagonal del punt (0,0) a (1,1).

A continuació, es mostra un exemple de corba *ROC*. El color lila ens informa de que s'ha obtingut una proba bona, el color blau significa que s'han obtingut uns resultats regulars i el color vermell seria el cas en que la proba és ineficaç:

Gràfic 3.1. Corba ROC.



Un paràmetre per avaluar la bondat de la proba és l'àrea que es situa per sota de la corba, aquesta s'anomena *AUC,* la qual pren valors entre 1 (si la proba és perfecte) i 0'5 (si la proba és ineficaç). *L'AUC* reflexa que tan bo és el test per a distingir els individus que pertanyen a un grup o l'altre durant el rang de talls de punts possibles.

3.7. RESIDUS

En l'apartat 2.2.4 ja es va explicar dues formes d'obtenir els residus: Pearson i la deviància. Tot seguit, es mostra una extensió de mètodes d'anàlisi de residus per a la regressió logística (Cortés, Valero, 2019):

- Els residus de la resposta que no s'utilitzen en els MLG a causa de que ignoren la variància:

$$e_i = y_i - \hat{\mu}_i = y_i - m_i\hat{\pi}_i$$

- Els residus de Pearson:

$$e_i^P = \frac{y_i - \hat{\mu}_i}{\sqrt{Var(\hat{\mu}_i)\phi}}$$

- Els residus de Pearson estandarditzats:

$$e_i^{PS} = \frac{y_i - \hat{\mu}_i}{\sqrt{Var(\hat{\mu}_i)(1-h_{ii})}}$$

- Els residus de la *deviance*, els més utilitzats en el models lineals generalitzats:

$$e_i^D = sign(y_i - \hat{\mu}_i)\sqrt{d_i}$$

- Els residus de la *deviance* estandarditzats:

$$e_i^{DS} = \frac{r_i^D}{\sqrt{\widehat{\Phi}\,(1-h_{ii})}}$$

- Els residus de la *deviance* estudentitzats:

$$e_i^{Stu} = sign(y_i - \hat{\mu}_i)\sqrt{(1-h_{ii})\left(r_i^{DS}\right)^2 + h_{ii}(r_i^P)^2}$$

On h_{ii} fa referència als *hat-values*.

4. REGRESSIÓ LOGÍSTICA PENALITZADA

Hi ha moltes situacions en que la base de dades consta de moltes variables explicatives i si es construeix un model amb totes elles, en general s'obtindrà un millor ajust però una major variància per a cadascun dels paràmetres a estimar i, per tant, es disminueix la precisió de les estimacions dels coeficients. En el cas contrari, si es construeix un model amb menys variables de les que es necessiten, la variància disminueix però augmentarà el biaix i s'obtindrà un mal ajust de les dades. També pot ser que es doni el cas que al haver tants atributs, alguns estiguin correlacionats entre ells.

Per a resoldre aquests problemes i saber quines variables s'han d'introduir al model estan els mètodes de selecció de variables, els quals ajusten models amb un subconjunt de predictors i tenen la funció de trobar un model que s'ajusti bé a les dades i que a la vegada, hi hagi un equilibri entre la bondat d'ajust i la variància, però aquests mètodes poden ser inestables quan presenta un nombre elevat de variables ja que, hi ha el risc de que es produeixi el problema de la multicol·linealitat i, a més, això pot causar un augment en la variabilitat de les estimacions dels predictors, per tant, una altra alternativa són els mètodes de penalització on s'ajusta un model (afegint una penalització que es modula amb el paràmetre λ) amb totes les variables predictores disminuint el valor dels coeficients cap a zero. Aquests mètodes obtenen menys variància en les estimacions comparat amb els mètodes de selecció de variables, és a dir, corregeixen la variabilitat. Hi ha diversos mètodes, però en aquest estudi se'n parlarà de tres en concret:

- *Ridge Regression*
- *Lasso Regression*
- *Elastic Net Regression*

Aquests tres mètodes es solen aplicar als models lineals, és a dir, a l'estimació per mínims quadrats, però es farà una generalització aplicada a la regressió logística (MLG).

4.1. MÈTODES DE SELECCIÓ DE VARIABLES

En aquest apartat s'explicarà en que consisteixen els mètodes de selecció de variables tradicionals. S'explicaran tres en concret:

- Mètode *Backward*
- Mètode *Forward*
- Mètode *Stepwise*

4.1.1. MÈTODE BACKWARD

Aquest mètode consisteix en eliminar les variables que tenen menys impacte en l'ajust del model, partint d'un model inicial que conté totes les variables predictores (model complet). El procés és el següent (James; Witten; Hastie; Tibshirani, 2013):

1) Es parteix del model complet M_p, el qual conté totes les variables explicatives.
2) Es té que $k = p, p-1, \dots, 1$. S'ajusten tots els possibles *k* models que continguin *k-1* predictors i entre aquests *k* models es selecciona el que té un major ajust. I aquest serà el model M_{k-1}.
3) Un cop es tenen $M_0, \dots, M_p$ es selecciona el que té un AIC o BIC inferior.

En aquest mètode, una vegada s'ha exclòs una variable del model ja no es pot tornar a afegir.

4.1.2. MÈTODE FORWARD

En canvi, el mètode *Forward* comença amb el model nul que conté com a única variable predictora l'intercept i va afegint una a una les variables que milloren el model. El procés que es segueix és el següent (James; Witten; Hastie; Tibshirani, 2013):

1) Es parteix del model nul M_0.
2) Es té que $k = 0, 1, \dots, p-1$. S'ajusten tots els possibles $p-k$ models que augmenten M_k amb un predictor addicional i entre aquests es selecciona el que presenta un millor ajust.
3) Un cop es tenen $M_0, \dots, M_p$, es selecciona el que té un AIC o BIC inferior.

En aquest cas, una vegada s'ha incorporat una variable al model ja no es pot excloure.

4.1.3. MÈTODE STEPWISE

Per últim, el mètode *Stepwise* és una combinació dels dos mètodes explicats anteriorment, el *Forward* i el *Backward.* Es comença amb el model nul que conté l'intercept i es van afegint els predictors que més contribueixen al model (com el mètode *Forward*). Després d'afegir cada nova variable, s'eliminen les que no aporten un millor ajust en el model (com el mètode *Backward*).

4.2. MÈTODES DE PENALITZACIÓ

Un cop explicats els mètodes de selecció de variables es procedirà als mètodes de penalització.

4.2.1. RIDGE REGRESSION

Aquesta tècnica, també anomenada L2, va ser proposada a la dècada dels 70 per a lidiar amb el problema de la multicol·linealitat d'un model lineal estimat per mínims quadrats, on el nombre de variables és inferior al nombre d'observacions ($p < n$).

L'objectiu d'aquest mètode és aproximar a zero els coeficients de les variables explicatives però sense excloure cap.

Per a aplicar-ho als models lineals generalitzats, en concret a la regressió logística, s'ha de maximitzar la funció de versemblança amb un paràmetre de penalització que s'aplica a tots els coeficients excepte a l'intercept (β_0). Partint de l'expressió de l'estimador de màxima versemblança:

$$l(\pi, y) = \sum_{i=1}^{n} \left[y_i log\left(\frac{\pi_i}{1-\pi_i}\right) + m_i log\,(1-\pi_i) \right]$$

També es pot escriure aquesta funció depenent dels paràmetres β_j:

$$l(\beta, y) = \sum_i \sum_j y_i x_{ij} \beta_j - \sum_i m_i log\left(1 + exp \sum_j x_{ij}\beta_j\right)$$

S'afegeix una penalització (Pereira; Basto; Ferreira, 2015):

$$l_{ridge}(\beta) = l(\beta, y) - \lambda \sum_{j=1}^{p} \| \beta_j \|^2$$

Que en forma matricial, queda l'expressió següent:

$$\frac{\partial l_{ridge}(\beta)}{\partial \beta} = X^T(Y-\mu) - \lambda\beta, \;\; on\; \mu = m\pi = E(Y)$$

També es podria escriure com:

$$\hat{\beta}^{ridge} = \max_{\beta} \sum_i \sum_j y_i x_{ij} \beta_j - \sum_i m_i log\left(1 + exp \sum_j x_{ij}\beta_j\right)$$

$$subjecte\; a \sum_{j=1}^{p} {\beta_j}^2 \leq t$$

On *t* és un paràmetre d'ajust. Existeix una correspondència directa entre *t* i λ de les equacions.

A mesura que augmenta la penalització λ, les estimacions dels coeficients tendeixen cap a zero, però cap d'elles prendrà el valor exacte de zero. Aquest és l'inconvenient de la *Ridge regression*, ja que encara que aconsegueix minimitzar la influència sobre el model de les variables predictores menys relacionades amb la variable resposta, inclou totes les variables explicatives al model final i quan el nombre de variables és elevat resulta més problemàtic per a la interpretació.

4.2.2. LASSO REGRESSION

El mètode *Lasso*, també anomenat L1, va ser introduït per *Tibshirani*, aquest mètode combina la contracció d'alguns paràmetres cap a zero, és a dir, l'estimació i la selecció de variables imposant una penalització sobre els coeficients de regressió.

A diferència de la regressió *Ridge*, els coeficients de les variables que contribueixen menys en el model són forçats a prendre un valor de zero. Només les variables més importants es queden en el model.

Partint de l'expressió de màxima versemblança, se li afegeix la penalització $\lambda \sum |\beta_j|$*:*

$$l_{lasso}(\beta) = l(\beta, y) - \lambda \sum_{j=1}^{p} |\beta_j|$$

Es pot expressar de la següent forma (Pereira; Basto; Ferreira, 2015):

$$\hat{\beta}^{lasso} = \max_{\beta} \sum_i \sum_j y_i x_{ij} \beta_j - \sum_i m_i log\left(1 + exp \sum_j x_{ij} \beta_j\right)$$

$$subjecte\ a \sum_{j=1}^{p} |\beta_j| \leq t$$

On t és un paràmetre de regularització que s'aplica als estimadors. Quan el paràmetre λ és elevat, força a que algunes de les estimacions dels coeficients preguin un valor de zero.

Lasso té un avantatge sobre la regressió *Ridge* ja que el model final només conté un subconjunt de les variables explicatives i això fa que millori la interpretabilitat del model, però també presenta algunes limitacions:

- Quan el nombre de variables és major que el nombre d'observacions *(p > N)*, selecciona N variables.
- Si el grup de variables presenta correlacions elevades dos a dos, *Lasso* tendeix a escollir només una variable d'aquest grup sense importar quina selecciona.

4.2.3. ELASTIC NET REGRESSION

La regressió *Elastic Net* és una tècnica de regularització i selecció de variables que conté els avantatges de L1 i L2 i supera certes limitacions d'aquests. En aquest cas hi haurà coeficients que tendeixin a zero (com en el cas de la regressió *Ridge*) i altres que prendran un valor exacte de zero (com en la regressió *Lasso*).

Aquest mètode és particularment útil quan el nombre de variables *(p)* és major que el nombre d'observacions *(N).*

La penalització aplicada en aquest cas a la funció de màxima versemblança és (Pereira; Basto; Ferreira, 2015):

$$l_{EN}(\beta) = l(\beta, y) - \lambda \sum_{j=1}^{p} \left(\alpha \left|\beta_j\right| + (1-\alpha) \parallel \beta_j \parallel^2\right)$$

És equivalent a escriure l'expressió de la següent manera:

$$\hat{\beta}^{EN} = \max_{\beta} \sum_i \sum_j y_i x_{ij} \beta_j - \sum_i m_i log\left(1 + exp \sum_j x_{ij} \beta_j\right)$$

$$subjecte\ a \sum_{j=1}^{p} (\alpha \left|\beta_j\right| + (1-\alpha) \parallel \beta_j \parallel^2) \leq t$$

On α pren un valor entre 0 i 1: $\alpha \in (0,1)$. Aquest paràmetre és una barreja de la penalització dels dos mètodes explicats anteriorment:

- Si α pren un valor de 1 es té el cas de la regressió *Lasso*.
- Si α pren un valor de 0 s'obté la regressió *Ridge*.
- Si α és situa entre 0 i 1, el terme de penalització s'interpola entre la norma L1 de β_j i la norma L2 al quadrat de β_j.

4.2.4. ESTIMACIÓ DEL PARÀMETRE λ

EL paràmetre λ controla la importància de la penalització, és a dir, determinarà quins predictors han d'estar al model (en el cas de la regressió *Lasso* i *Elastic Net*) i quins han de tendir més cap a zero (en el cas de la regressió *Ridge*).

Un dels mètodes més utilitzats per a determinar l'estimació de λ és mitjançant la validació creuada (*cross-validation*). Aquesta tècnica consisteix en dividir la base de dades en dos: una serà el *data set* d'entrenament (*training*) per a ajustar el model i l'altre el set de prova (*test*) per avaluar la capacitat predictiva mitjançant l'error de predicció. Es dividirà la base de dades *training* en *K* parts iguals. Un dels subconjunts s'utilitza com dades de *test* i la resta *(K-1)* com dades de *training*. Aquest procés de cross-validació es repetirà *k* vegades (normalment k=10), amb cada un dels possible subconjunts de dades *test* i a cada iteració es calcularà l'error. Per últim, es realitzarà la mitjana aritmètica dels resultats de cada iteració per a obtenir un únic valor. El valor del paràmetre λ serà el que doni un mínim error global de la validació creuada.

L'avantatge que té aquest mètode és que és molt precís ja que s'avalua a partir de *K* combinacions de dades *train* i de *test*, però l'inconvenient és que presenta un cost computacional elevat.

Un altre mètode a mencionar és la cross-validació generalitzada, la qual és similar a la *cross-validation* però té un cost computacional inferior.

Per últim, una altra tècnica que es pot utilitzar per a l'estimació de λ són les traces *Ridge*, que consisteix en provar diversos valors de λ representant les diferents estimacions dels paràmetres β_j i es reté aquell valor de *k* a partir del qual s'estabilitzen les estimacions.

En aquest estudi el mètode que s'utilitzarà per a l'estimació del paràmetre de penalització serà les traces *Ridge* i la *cross-validation* dividida en *k* parts.

5. ANÀLISI PRÀCTIC AMB R

En aquest apartat es durà a terme l'anàlisi dels mètodes de selecció de variables i de penalització per a dues bases de dades. La primera d'elles contindrà un nombre elevat de variables mentre que l'altre en tindrà menys.

Es vol comprovar si els mètodes de regularització funcionen millor que els tradicionals, com el mètode *Stepwise*.

5.1. BASE DE DADES: CARCINOMA HEPATOCEL·LULAR

Com ja s'ha comentat a l'inici del treball, la primera base de dades a estudiar tracta sobre individus que tenen càncer de fetge, en concret, de carcinoma hepatocel·lular. Aquesta consta de 50 variables i 165 observacions.

Respecte les variables, 27 són categòriques (24 amb dos nivells i 3 amb més de dos nivells), les quals proporcionen la següent informació de cada individu:

- *Gender*: el sexe.
- *Sym*: si presenta símptomes.
- *Alc*: si veu alcohol.
- *HepBSur*: si presenta l'antigen de superfície d'hepatitis B.
- *HepBant*: si presenta l'anticòs de superfície d'hepatitis B.
- *HepBCore*: si presenta l'anticòs del nucli d'hepatitis B.
- *HepC*: si presenta l'anticòs d'hepatitis C.
- *Cir*: si pateix cirrosis.
- *End*: si viu en un país endèmic .
- *Smo*: si fuma.
- *Dia*: si presenta diabetis.
- *Obe*: si pateix d'obesitat.
- *Hem*: si presenta hemocromatosis.
- *Art*: si pateix d'hipertensió arterial.
- *CRen*: si pateix d'insuficiència crònica renal.
- *HIV*: si presenta el virus d'immunodeficiència humana.
- *Non*: si presenta esteatohepatitis no alcohòlica.
- *EVar*: si pateix de varius esofàgiques.
- *Spl*: si presenta la melsa dilatada.
- *PHyp*: si pateix d'hipertensió portal.

- *Thr*: si pateix de trombosis a la vena portal.
- *LMet*: si pateix metàstasis.
- *Rad*: si presenta carcinoma hepatocel·lular distintiu radiològic.
- *Sta*: grau d'estat funcional.
- *Encdeg*: grau d'encefalopatia.
- *Ascdeg*: grau d'ascites.

Aquestes variables categòriques estan categoritzades de la següent manera:

- La variable *Gender* conté dos nivells: *woman* si és dona i *man* si és home.
- Les variables de *Sym* a *Rad* són variables binàries. Prenen valor 1 les classes que tenen més risc i 0 les que en tenen menys. Per exemple, la variable *Sym* pren valor 1 si l'individu presenta símptomes i 0 en cas contrari.
- La variable *Sta* conté 5 nivells:

Taula 5.1. Descripció nivells de la variable Sta.

0	Completament actiu
1	No activitat física extrema
2	Pot caminar i auto cuidar-se
3	Auto cuidat limitat
4	Incapacitat

- Les variables *Encdeg* i *Ascdeg* contenen 3 nivells:

Taula 5.2. Descripció nivells de les variables Encdeg i Ascdeg.

1	Lleu
2	Moderat
3	Greu

- La variable *Class* té dos nivells: si el pacient ha mort *(died)* o si ha sobreviscut *(survived).*

Respecte a les variables numèriques, proporcionen les següents característiques de cada individu:

- *Agedia*: l'any del diagnòstic.
- *Alcpd*: els grams d'alcohol que consumeixen per dia.
- *Cigpy*: els paquets de tabac que consumeixen per any.
- *IntNorRat*: el rati normalitzat internacional.
- *Alp*: quantitat d'alfa-fetoproteïna (ng/mL).
- *Hae*: quantitat d'hemoglobina (g/dL).
- *MCorVol*: volum corpuscular mitjà.
- *Leu*: quantitat de leucòcits (g/L).
- *Plat*: quantitat de plaquetes (g/L).
- *Alb*: quantitat d'albúmina (mg/dL).

- *Bil1*: quantitat de bilirubina (mg/dL).
- *Ala*: quantitat d'alanina (U/L).
- *Aspa*: quantitat d'aspartat transaminasa (U/L).
- *Gam*: quantitat de gamma-glutamil (U/L).
- *Alk*: quantitat de fosfatasa alcalina (U/L).
- *Prot*: quantitat de proteïnes (g/dL).
- *Crea*: quantitat de creatinina (mg/mL).
- *NNod*: nombre de nòduls.
- *Dnod*: la major dimensió del nòdul (cm).
- *Bil2*: quantitat de bilirubina directa (mg/dL).
- *Iron*: quantitat de ferro (mcg/dL).
- *Oxy*: saturació d'oxigen (%) .
- *Fer*: quantitat de ferritina (ng/mL).

La variable resposta d'aquest estudi és *Class* que indica si l'individu ha mort o sobreviscut a causa del càncer de carcinoma hepatocel·lular.

5.1.1. ANÀLISI DESCRIPTIU

Per a realitzar l'anàlisi descriptiu de les dades s'han creat dues bases de dades: una que conté les variables categòriques i l'altre les numèriques.

5.1.1.1. VARIABLES CATEGÒRIQUES

Per a les variables categòriques s'ha observat quants individus pertanyen a cada classe:

- Variables binàries:

Taula 5.3. Descriptiva de les variables binàries I.

	0	1
Gender	32	133
Sym	53	94
Alc	43	122
HepBSur	132	16
HepBAnt	125	1
HepBCore	103	38
HepC	122	34
Cir	16	149
End	116	10
Smo	61	63
Dia	106	56
Obe	135	20

	0	1
Hem	135	7
Art	103	59
Cren	143	20
HIV	148	3
Non	135	8
Evar	44	69
Spl	66	84
PHyp	44	110
Thr	126	36
LMet	125	36
Rad	52	111

Es pot observar que la majoria de les variables no contenen, aproximadament, el mateix nombre d'individus a cadascuna de les classes, però l'atribut que més descompensat està és

HepBAnt que un 75'76% pertany a la classe 0, un 0'61% a la classe 1 i les dades restants són *missings* que s'analitzaran més endavant.

- Variables ordinals:

Taula 5.4. Descriptiva de la variable ordinal Sta.

	0	1	2	3	4
Sta	80	30	32	18	5

Es pot veure que la variable *Sta* no està molt compensada, la classe 0 és la que conté més individus (48'48%) i la 4 la que menys (3'03%). El nivell 1, 2 i 3 es podria dir que estan força compensats.

Taula 5.5. Descriptiva de les variables ordinals Encdeg i Ascdeg.

	1	2	3
Encdeg	142	18	4
Ascdeg	109	36	18

Els atributs *Encdeg* i *Ascdeg* tampoc contenen el mateix nombre d'observacions, la classe 1 és la que presenta més individus (més d'un 60% de les dades) i els altres es situen entre el nivell 2 i 3. Les dades que falten són *missings*.

- Variable resposta:

Taula 5.6. Descriptiva de la variable resposta I.

	died	survived
Class	63	102

Per últim, pel que fa a les variables categòriques, la variable resposta *(Class)* conté un 38'18% de les dades a la classe *died* i un 61'81% a *survived.*

5.1.1.2. VARIABLES NUMÈRIQUES

Respecte la descriptiva de les variables numèriques, s'han obtingut els següents resultats:

Taula 5.7. Descriptiva de les variables numèriques I.

	Mitjana	Variància	Desv. Típica	Covariància	Min	Q1	Mediana	Q3	Max
Agedia	64,69	177,41	13,32	0,21	20,00	57,00	66,00	74,00	93,00
Alcpd	71,01	5.818,28	76,28	1,07	0,00	0,00	75,00	100,00	500,00
cigpy	20,46	2.658,96	51,57	2,52	0,00	0,00	0,00	30,50	510,00
IntNorRat	1,42	0,29	0,48	0,34	0,84	1,17	1,30	1,53	4,82
Alp	19.299,95	22.230.313.672,94	149.098,34	7,73	1,20	5,20	33,00	615,00	1.810.346,00
Hae	12,88	4,60	2,15	0,17	5,00	11,43	13,05	14,60	18,70
MCorVol	95,12	70,66	8,41	0,09	69,50	89,78	94,95	100,68	119,60
Leu	1.473,96	8.462.897,75	2.909,11	1,97	2,20	5,10	7,20	19,53	13.000,00
Plat	113.206,44	11.474.401.424,50	107.118,63	0,95	1,71	255,75	93.000,00	171.500,00	459.000,00
Alb	3,45	0,47	0,69	0,20	1,90	3,00	3,40	4,05	4,90
Bil1	3,09	30,24	5,50	1,78	0,30	0,80	1,40	2,93	40,50
Ala	67,09	3.310,82	57,54	0,86	11,00	31,00	50,00	78,00	420,00
Aspa	96,38	7.653,49	87,49	0,91	17,00	46,25	71,00	110,25	553,00
Gam	268,03	66.951,73	258,75	0,97	23,00	91,25	179,50	345,25	1.575,00
Alk	212,21	28.205,04	167,94	0,79	1,28	108,25	162,00	261,50	980,00

Prot	8,96	137,56	11,73	1,31	3,90	6,30	7,05	7,58	102,00
Crea	1,13	0,91	0,96	0,85	0,20	0,70	0,85	1,10	7,60
NNod	2,74	3,23	1,80	0,66	0,00	1,00	2,00	5,00	5,00
dnod	6,85	25,96	5,10	0,74	1,50	3,00	5,00	9,00	22,00
Bil2	1,93	17,73	4,21	2,18	0,10	0,37	0,70	1,40	29,30
Iro	85,60	3.102,38	55,70	0,65	0,00	40,50	83,00	118,00	224,00
Oxy	37,03	840,63	28,99	0,78	0,00	16,00	27,00	56,00	126,00
Fer	439,00	208.953,62	457,11	457,11	0,00	84,00	295,00	706,00	2.230,00

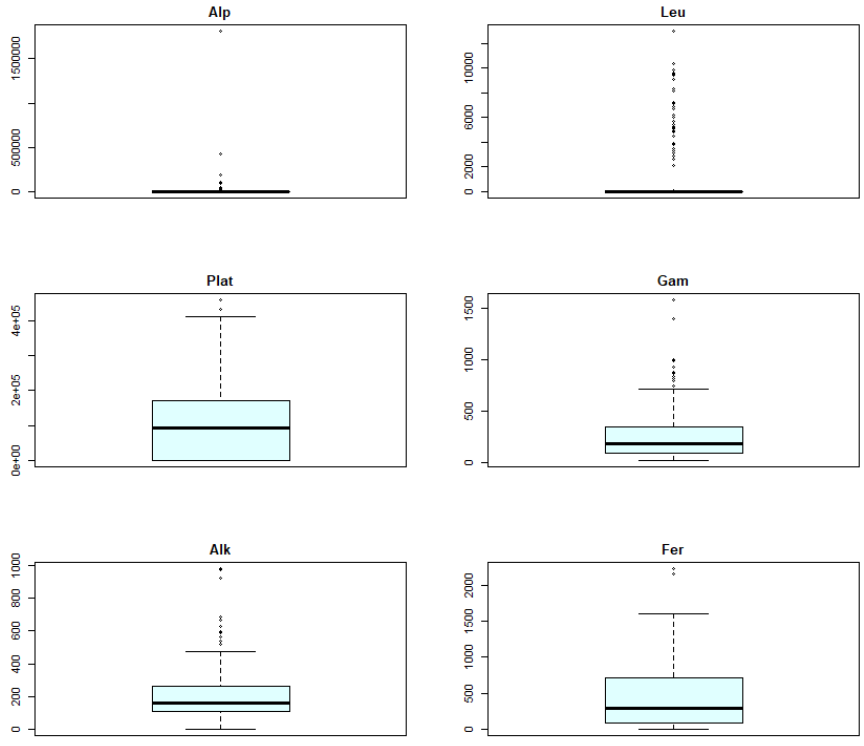
Les variàncies que més destaquen són la de les variables *Alp, Leu, Plat, Gam, Alk* i *Fer* ja que obtenen valors molt elevats, això és degut a que contenen valors extrems, és a dir, hi ha molta diferència entre el mínim i el màxim. Com a conseqüència, també s'obtenen valors elevats de la desviació típica per a aquests atributs.

5.1.2. PROCESSAMENT DE LA BASE DE DADES

5.1.2.1. OUTLIERS

Totes les variables numèriques contenen valors extrems, algunes observacions tenen valors més allunyats de la mediana que les altres. A continuació, es mostren les gràfiques dels atributs on més destaquen els *outliers*. Aquests coincideixen amb les variables comentades anteriorment en la descriptiva de les variables numèriques (les que obtenien variàncies elevades).

Gràfic 5.1. Outliers I.



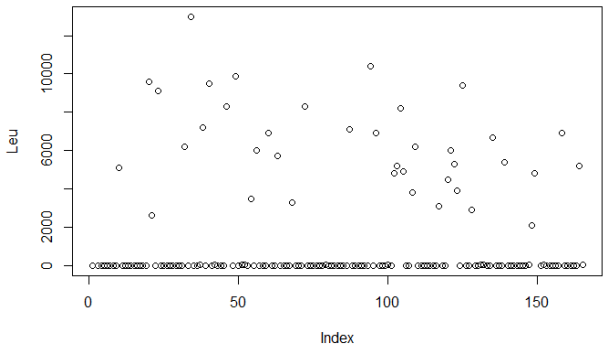
5.1.2.2. TRANSFORMACIONS

Per a veure si es podia millorar la base de dades es va realitzar un gràfic per a cadascuna de les variables numèriques i es va observar que hi havia variables que s'havien d'aplicar transformacions.

Els atributs *Bil1, Ala, Aspa, Gam, Alk, Prot, Crea, dnod* i *Bil2* necessitaven alguna transformació per a millorar les dades, per tant, se'ls hi va aplicar el logaritme.

Pel que fa a la variable *Leu,* les dades presenten la següent forma:

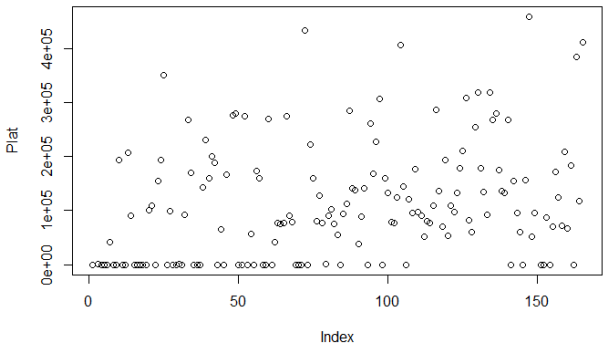
Gràfic 5.2. Distribució dades de la variable Leu.



Es va aplicar la transformació logarítmica a les dades però no milloraven, per tant, es va decidir convertir aquesta variable a factor amb dos nivells on valen 1 els valors inferiors a 4 G/L i superiors a 11 G/L i tots els altres prenen valor de 0. Es va decidir categoritzar així ja que la quantitat normal de leucòcits en un individu és de 4'00-11'00 G/L.

Respecte la variable *Plat* s'obté el següent gràfic:

Gràfic 5.3. Distribució dades de la variable Plat.

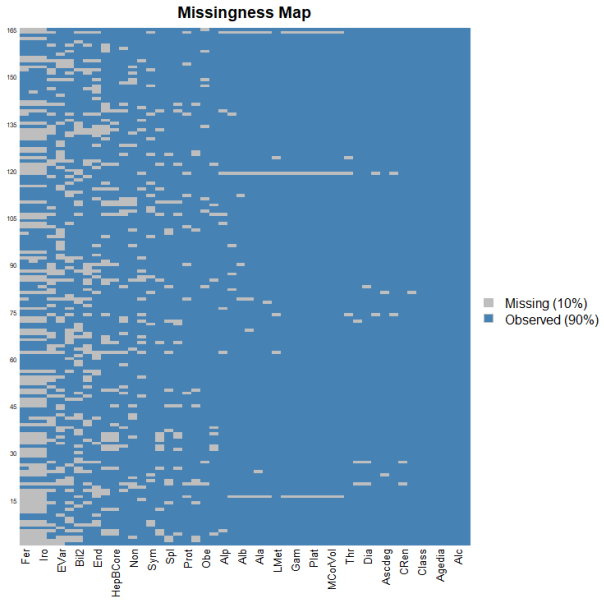


Succeeix el mateix que s'ha explicat en el cas anterior, es va aplicar la transformació logarítmica a les dades però no milloraven i, per tant, es va passar a factor amb dos nivells on els valors inferiors a 150 G/L i superiors a 450 G/L prenen valor 1 i la resta de 0. Es va decidir categoritzar-la d'aquesta manera perquè els valors normals de plaquetes en un individu es situen entre 150-450 G/L.

5.1.2.3. DESCRIPTIVA DELS NA

Primer de tot es va estudiar quants *missings* contenia en total el *dataset*, aquests s'han representat en un *missmap*:

Gràfic 5.4. Missmap.



Com es pot observar, un 10% de les dades són *missings*. També, es va analitzar quantes dades mancants contenia cadascuna de les variables i el percentatge que representaven. Els atributs que no contenen cap *missing* són *Gender, Alc, Cir, Agedia, Sta* i *Class:*

Taula 5.8. Nombre i percentatge de missings.

	Nº missings	% missings
Fer	81	49,1%
Oxy	80	48,5%
Iro	79	47,9%
cigpy	53	32,1%
EVar	52	31,5%
Alcpd	48	29,1%
Bil2	44	26,7%
Smo	41	24,8%
HepBAnt	39	23,6%
End	39	23,6%
HepBCore	24	14,5%
Hem	23	13,9%
Non	22	13,3%
dnod	20	12,1%
Sym	18	10,9%
HepBSur	17	10,3%
Spl	15	9,1%
HIV	14	8,5%
PHyp	11	6,7%
Prot	11	6,7%
Obe	10	6,1%
HepC	9	5,5%

	Nº missings	% missings
Alp	8	4,8%
Crea	7	4,2%
Alb	6	3,6%
Bil1	5	3,0%
LMet	4	2,4%
IntNorRat	4	2,4%
Ala	4	2,4%
Dia	3	1,8%
Art	3	1,8%
Thr	3	1,8%
Hae	3	1,8%
MCorVol	3	1,8%
Leu	3	1,8%
Plat	3	1,8%
Aspa	3	1,8%
Gam	3	1,8%
Alk	3	1,8%
CRen	2	1,2%
Rad	2	1,2%
Ascdeg	2	1,2%
NNod	2	1,2%
Encdeg	1	0,6%

Els atributs *Fer, Oxy* i *Iro* són els que més dades mancants presenten, quasi un 50%. Per aquest motiu, per a obtenir un millor anàlisi, es va decidir excloure aquestes tres variables, ja que al no tenir informació d'aproximadament la meitat de les observacions, no aportaven gran informació per a

l'estudi. Per últim, els individus que contenien més d'un 40% de *missings* també es van eliminar de la base de dades.

Després de realitzar aquests canvis, s'obté un *dataset* de 47 atributs i 163 observacions.

5.1.2.4. IMPUTACIÓ DE DADES MANCANTS

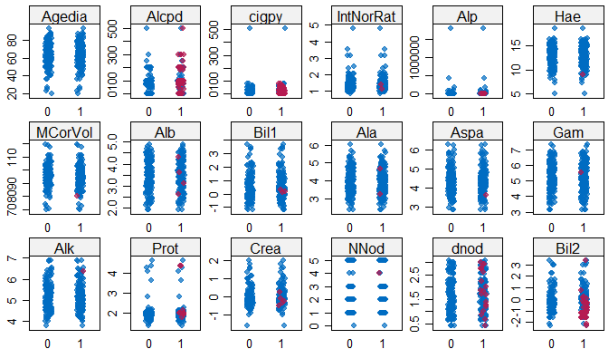
La imputació de dades mancants s'ha realitzat mitjançant la funció *mice.* Es va decidir utilitzar aquesta ja que permet combinar la imputació de variables numèriques, binàries, ordinals, entre d'altres.

La funció *mice* conté l'opció de realitzar la imputació mitjançant diversos mètodes, per aquest estudi es va provar entre els que aplica per defecte i el *CART*, que consisteix en la imputació mitjançant arbres de classificació i regressió. Els mètodes per defecte aplicats en aquest *dataset* són:

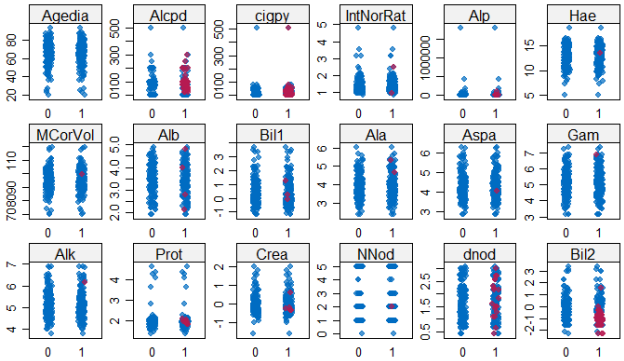
- *logreg (logistic regression)* per a les variables binàries.
- *pmm (predictive mean matching)* per a les variables numèriques.
- *polr (proportional odds model)* per a les variables ordinals.

Un cop aplicats aquests mètodes es va comprovar que no hi hagués cap *missing* i seguidament, es va realitzar la diagnosi per a escollir quin dels dos mètodes funcionava millor.

Gràfic 5.5. Stripplot dels mètodes per defecte.



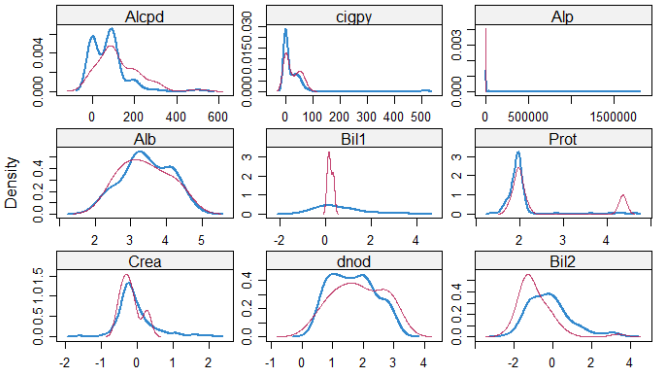
Gràfic 5.6. Stripplot del mètode CART.



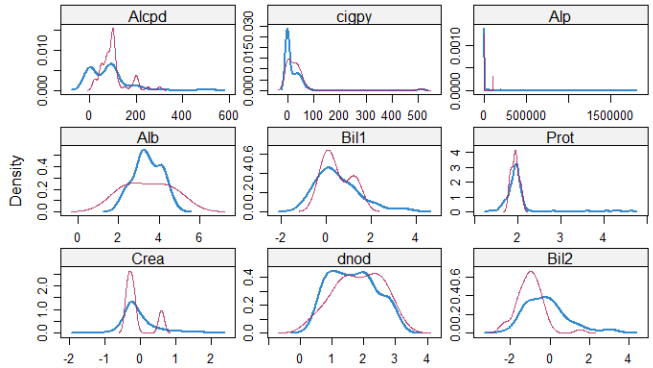
Els *stripplots* dels dos mètodes són bastant similars exceptuant alguna observació. Aquest és un mètode gràfic on es representa en color blau els valors dels individus observats i en vermell els valors

que la funció *mice* ha atribuït a les observacions que presentaven *missings,* per a cada una de les variables. A continuació, es mostren els gràfics de densitat:

Gràfic 5.7. Densityplot dels mètodes per defecte.



Gràfic 5.8. Densityplot del mètode CART.



Amb els gràfics de *densityplot* es pot apreciar millor quin mètode seria el més adequat. Es pot observar que la densitat de les variables amb els mètodes aplicats per defecte a les dades mancants són més similars a les dades observades que pel mètode *CART.* Per tant, els pròxims anàlisi que es realitzaran es duran a terme mitjançant la imputació per als mètodes que aplica la funció *mice* per defecte.

5.1.3. APLICACIÓ DELS MÈTODES DE SELECCIÓ DE VARIABLES

En aquest apartat es durà a terme l'aplicació del mètode de selecció de variables amb el *Stepwise*. S'ha decidit aplicar aquest ja que és una combinació del B*ackward* i *Forward* i, com s'ha comentat a l'inici del treball, l'objectiu és mirar si els mètodes de penalització funcionen millor que els tradicionals.

5.1.3.1. CORRELACIÓ

Abans d'aplicar el mètode *Stepwise,* s'ha estudiat si existeix multicol·linealitat entre les variables. Per a dur a terme aquest anàlisi, s'ha creat un model additiu amb 46 variables com a predictores i la variable *Class* com a resposta.

Una vegada obtingut el model, s'han calculat els VIF (factor d'inflació de la variància) de totes les variables mitjançant la funció *vif* de l'R:

- Si VIF = 1, hi ha manca de col·linealitat.
- Si 1 < VIF < 5, hi ha lleu col·linealitat de la variable analitzada amb les altres.
- Si 5 < VIF <10, hi ha certa col·linealitat de la variable estudiada amb les altres.
- Si VIF > 10, hi ha una alta col·linealitat.

Els resultats obtinguts són:

Taula 5.9. VIF de les variables I.

Gender	2,22	Non	1,79	Hae	2,55
Sym	1,98	Evar	3,84	MCorVol	2,19
Alc	4,20	Spl	3,10	Leu	1,89
HepBSur	2,72	PHyp	3,87	Plat	1,89
HepBAnt	1,76	Thr	1,77	Alb	2,67
HepBCore	2,79	LMet	2,75	Bil1	7,94
HepC	2,05	Rad	1,67	Ala	4,61
Cir	2,49	Agedia	2,13	Aspa	5,51
End	2,50	Alcpd	4,22	Gam	2,65
Smo	2,08	cigpy	2,79	Alk	3,35
Dia	1,85	Sta	12,59	Prot	1,76
Obe	1,90	Encdeg	4,01	Crea	2,54
Hem	1,52	Ascdeg	3,55	NNod	2,28
Art	1,83	IntNorRat	2,03	dnod	2,71
Cren	1,96	Alp	1,81	Bil2	10,45
HIV	1,97				

Hi ha dues variables que tenen un valor superior a 10, per tant, hi ha presència de multicol·linealitat, dues prenen un valor superior a 5, és a dir, hi ha certa col·linealitat i la resta de variables prenen valors entre 1 i 5, el que significa que presenten una correlació lleu.

5.1.3.2. MÈTODE STEPWISE

Primer de tot, s'ha dividit la base de dades en *train,* que conté el 70% de les dades i s'utilitzarà per a crear el model lineal generalitzat, i *test*, que conté el 30% restant i servirà per a avaluar el model. Seguidament s'ha creat el model complet, el qual consta de 46 variables explicatives i la variable resposta *Class* amb les observacions de les dades *train*. També, s'ha creat un model nul que només conté l'intercept. Un cop aplicat el mètode *Stepwise*, s'ha obtingut que el model ha de contenir 14 variables explicatives de les quals, 11 d'elles són significatives amb un nivell de significació del 5%. El model és el següent:

$$Class \sim Alk + IntNorRat + Alp + Sym + CRen + cigpy + Art + Smo + Agedia + HepBSur + Aspa + Ala + HepC + Thr$$

Tots els models esmentats han estat creats a partir de la funció *glm* on s'ha seleccionat la família binomial al paràmetre *family.*

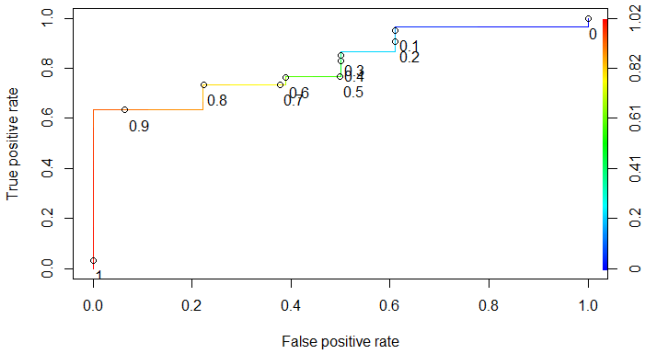
Per avaluar el model s'ha creat la matriu de confusió amb la qual s'han calculat les mesures de predicció:

Taula 5.10. Stepwise: Mesures de predicció I.

Exactitud	Taxa d'error	Sensibilitat	Especificitat	Precisió	VPN
68,75%	31,25%	76,67%	55,56%	74,19%	58,82%

I la corba *ROC* amb un *AUC* de 0'82:

Gràfic 5.9. Stepwise: Corba ROC I.



5.1.4. APLICACIÓ DELS MÈTODES DE PENALITZACIÓ

L'aplicació dels mètodes de penalització s'ha realitzat mitjançant la funció *glmnet* de l'R on s'han utilitzat els següents paràmetres:

- a*lpha:* va variant segons el mètode que s'aplica. Com ja s'ha comentat al punt 4.2.3, si *alpha* pren un valor de 0 s'aplica la *Ridge regression,* si pren un valor de 1 s'està aplicant *Lasso* i si pren un valor entre 0 i 1 s'aplica *Elastic Net.*
- *family*: s'ha seleccionat la família binomial ja que la variable resposta és binària.
- *lambda:* valor numèric que defineix el grau de penalització. Per a trobar el valor adequat de λ s'utilitzarà la *cross-validation* dividint les dades *train* en 10 parts.

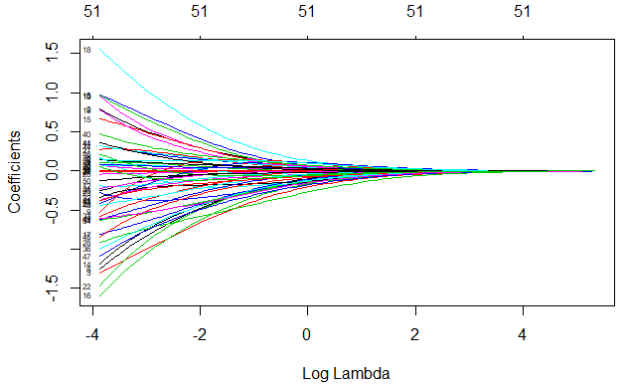
La funció *glmnet* requereix crear una matriu amb la funció *model.matrix*, la qual conté tots els predictors.

5.1.4.1. RIDGE REGRESSION

El primer mètode de penalització que s'ha aplicat és la *Ridge regression* amb les observacions del *dataset train* i el paràmetre *alpha* pren un valor de 0.

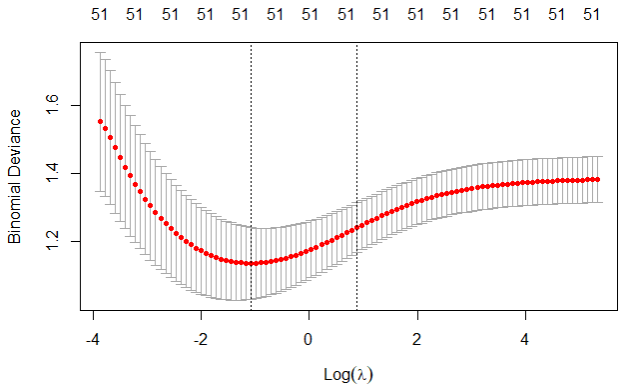
Un cop s'ha aplicat la funció *glmnet*, sense el paràmetre *lambda,* s'ha realitzat un gràfic de l'evolució dels coeficients a mesura que el valor de λ va augmentant.

Gràfic 5.10. Ridge regression: Evolució dels coeficients I.



Es pot observar que a mesura que s'incrementa el valor λ els coeficients es van fent més petits. Per a obtenir el valor de λ adequat, s'ha utilitzat el mètode de *cross-validation* i s'ha obtingut el següent gràfic:

Gràfic 5.11. Ridge regression: Valors de λ I.



Es mostren els diversos errors de *cross-validation* per diferents valors del logaritme de λ. Es pot observar que el valor amb el qual s'obté l'error mínim es situa entorn a -1. La xifra exacte d'aquest paràmetre és de 0'3397 que al aplicar el logaritme s'obté un resultat de -1'0796.

Una altre λ que es pot estudiar és l'anomenada valor òptim, la qual és major que el valor mínim però no s'allunya més de 1 de l'error estàndard. Aquesta val 2'3969.

Un cop obtingut els dos valors de λ s'ha creat un model per a cada un d'ells i s'ha comprovat que cap coeficient pren un valor nul, ja que el mètode de regressió *Ridge* els aproxima cap a zero però no els hi dona un valor exacte de zero.

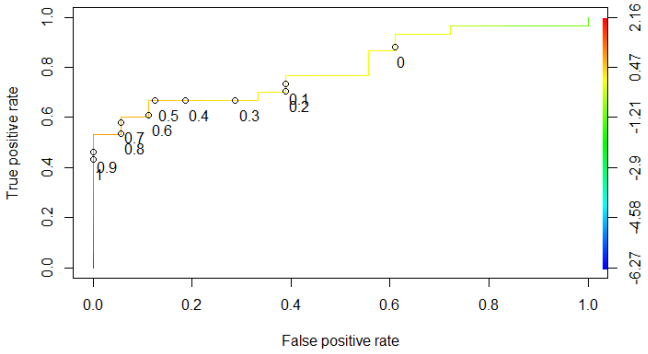
Per últim, s'ha realitzat la matriu de confusió per a calcular les mesures de predicció dels dos models:

Taula 5.11. Ridge regression: Mesures de predicció I.

	Exactitud	Taxa d'error	Sensibilitat	Especificitat	Precisió	VPN
λ mínima	75,00%	25,00%	66,67%	88,89%	90,91%	61,54%
λ òptima	70,83%	29,17%	60,00%	88,89%	90,00%	57,14%

Es pot observar que el model que conté la λ mínima té una menor taxa d'error i, per tant, una major exactitud. El model amb aquesta λ, amb un *AUC* de 0'80, presenta la següent corba *ROC:*

Gràfic 5.12. Ridge regression: Corba ROC I.

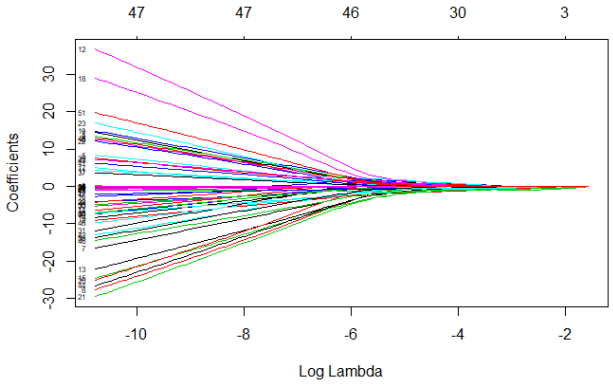


5.1.4.2. LASSO REGRESSION

S'ha seguit el mateix procediment explicat anteriorment, però amb la única diferència que, en aquest cas, α pren un valor de 1.

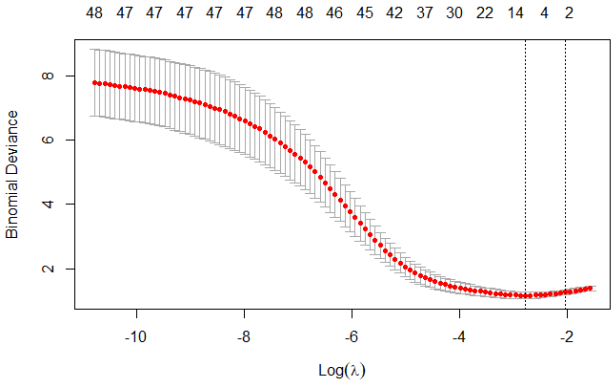
En el següent gràfic es pot observar com evolucionen els coeficients segons els diversos valors de λ i que aquests s'estabilitzen quan prenen un valor del logaritme de λ de -5, aproximadament:

Gràfic 5.13. Lasso regression: Evolució dels coeficients I.



Respecte el gràfic per a saber on es situa el valor adequat de λ per a aconseguir l'error mínim, s'ha obtingut:

Gràfic 5.14. Lasso regression: Valors de λ I.



Es pot veure que el valor mínim i òptim estan al voltant de -3 i -2, respectivament. Exactament s'obté una λ mínima de 0'0622 i òptima de 0'1309.

Una vegada realitzats els dos models, un per cada λ, s'ha obtingut que per la λ mínima 11 coeficients prenen valors diferents de zero, mentre que per l'òptima s'han trobat només 4 casos.

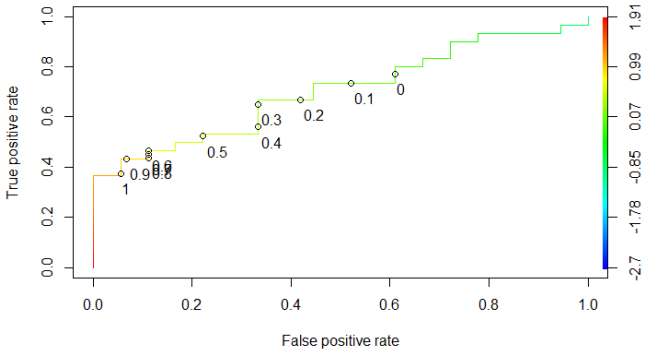
Al avaluar al model s'han obtingut els següents resultats:

Taula 5.12. Lasso regression: Mesures de predicció I.

	Exactitud	Taxa d'error	Sensibilitat	Especificitat	Precisió	VPN
λ mínima	60,42%	39,58%	50,00%	77,78%	78,95%	48,28%
λ òptima	58,33%	41,67%	46,67%	77,78%	77,78%	46,67%

El model amb λ mínima té una major exactitud, és a dir, té una taxa d'error inferior que el model amb λ òptima. La corba *ROC* del model amb millor predicció, que té una àrea per sota de la corba de 0'70, és la següent:

Gràfic 5.15. Lasso regression: Corba ROC I.



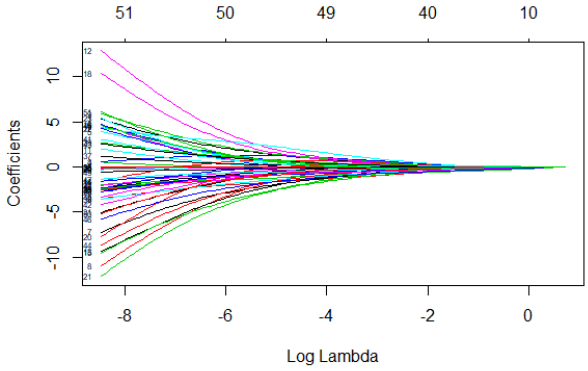
5.1.4.3. ELASTIC NET REGRESSION

Per a aplicar aquest mètode, es prendran diversos valors de α per veure amb quin d'aquests s'obté un millor model:

- $\alpha = 0'10$

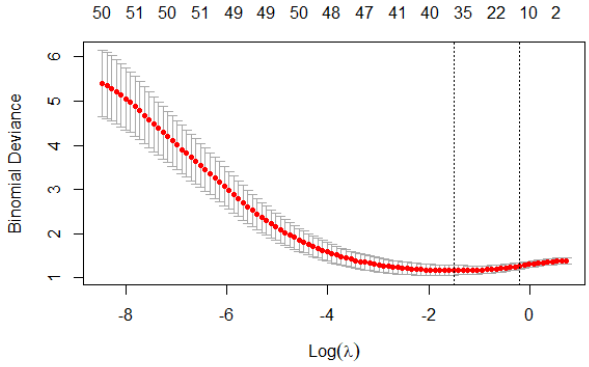
  Com que α està més propera a zero, se li dona més pes a la penalització *Ridge regression*. L'evolució dels coeficients per a diversos valors del $\log(\lambda)$ pren la següent forma:

Gràfic 5.16. Elastic Net regression ($\alpha = 0'10$): Evolució dels coeficients I.



S'observa que els coeficients es comencen a estabilitzar, aproximadament, en el -3. En el gràfic següent es pot trobar el valor mínim i òptim del $\log(\lambda)$, els quals estan situats entorn el -1'55 i -0'20, respectivament:

Gràfic 5.17. Elastic Net regression $(\alpha = 0'10)$*: Valors de* λ *I.*



El valor mínim de λ és de 0'2235 i l'òptim de 0'8222. Una vegada creats els dos models amb els respectius valors, s'obté que per el mínim 14 coeficients de les variables prenen valor de zero, mentre que per l'òptim són 38.

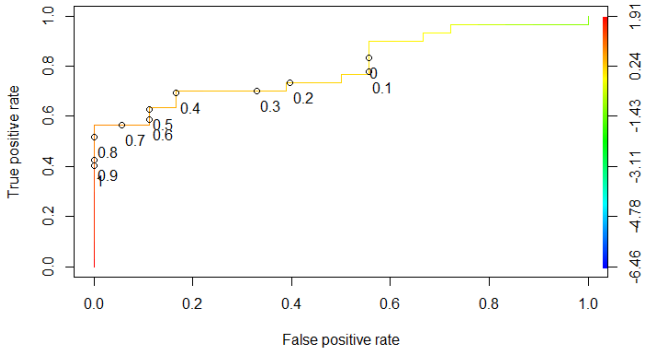
Respecte l'avaluació del model:

Taula 5.13. Elastic Net regression $(\alpha = 0'10)$*: Mesures de predicció I.*

	Exactitud	Taxa d'error	Sensibilitat	Especificitat	Precisió	VPN
λ mínima	70,83%	29,17%	60,00%	88,89%	90,00%	57,14%
λ òptima	68,75%	31,25%	53,33%	94,44%	94,12%	54,84%

El model que conté la λ mínima és el que obté una menor taxa d'error. La corba *ROC* per a aquest model, amb un *AUC* de 0'80, presenta la següent forma:

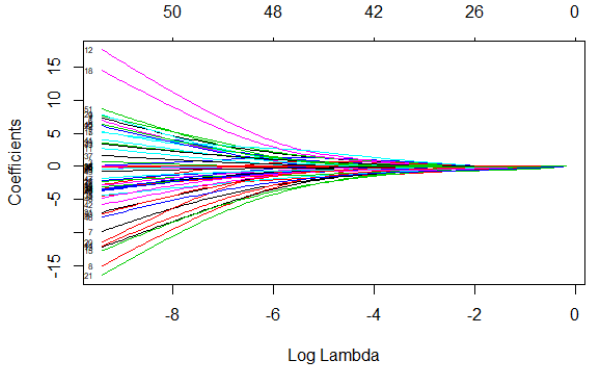
Gràfic 5.18. Elastic Net regression $(\alpha = 0'10)$*: Corba ROC I.*



- $\alpha = 0'25$

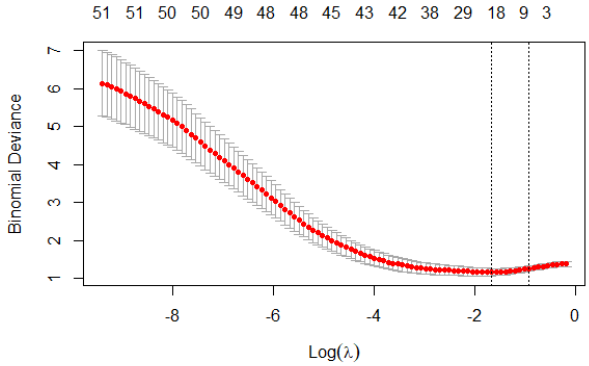
Per aquesta situació, α està més allunyat de zero però encara se li està donant més pes a la penalització *Ridge regression*. L'evolució dels coeficients de les variables amb diversos valors del $\log(\lambda)$, que es comencen a estabilitzar aproximadament en el -4, presenten la següent forma:

Gràfic 5.19. Elastic Net regression ($\alpha = 0'25$): Evolució dels coeficients I.



Respecte la gràfica per trobar el valor mínim i òptim de λ, s'aprecia que aquests es situen al voltant de -1'80 i -1, respectivament:

Gràfic 5.20. Elastic Net regression ($\alpha = 0'25$): Valors de λ I.



El valor mínim de λ és de 0'1882 i l'òptim de 0'3962. Una vegada creats els models per cada λ, el que conté el mínim obté 33 coeficients que prenen valor zero, mentre que el model amb el valor òptim presenta 44 coeficients amb una xifra de zero.

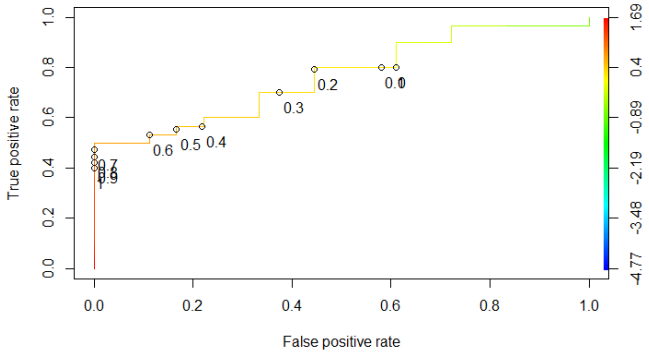
Pel que fa a les mesures de predicció per avaluar els models, s'han obtingut els següents resultats:

Taula 5.14. Elastic Net regression ($\alpha = 0'25$): Mesures de predicció I.

	Exactitud	Taxa d'error	Sensibilitat	Especificitat	Precisió	VPN
λ mínima	64,58%	35,42%	53,33%	83,33%	84,21%	51,72%
λ òptima	62,50%	37,50%	53,33%	77,78%	80,00%	50,00%

Per $\alpha = 0'25$, el model amb el valor mínim de λ té una menor taxa d'error i, per tant, una major exactitud. Presenta un *AUC* de 0'76 i la corba *ROC* té la següent forma:

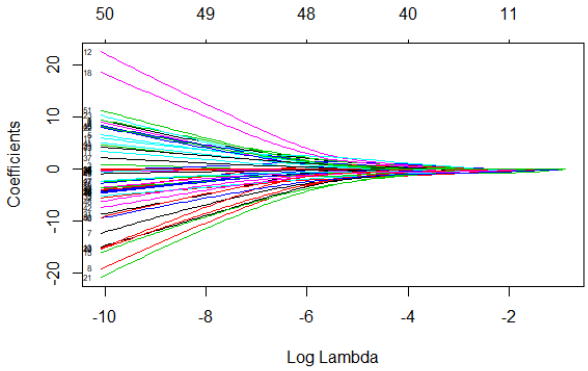
Gràfic 5.21. Elastic Net regression ($\alpha = 0'25$)*: Corba ROC I.*



- $\alpha = 0'50$

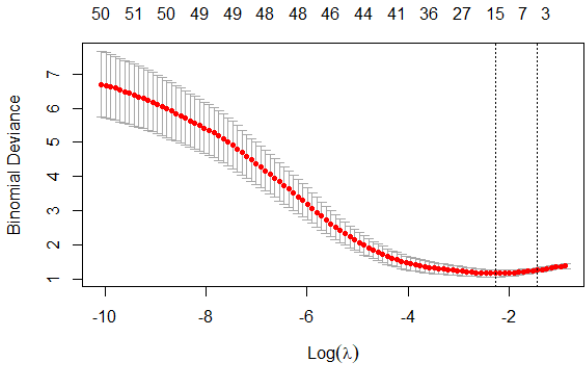
Com que α pren un valor de 0'50, se li dona el mateix pes a la penalització *Ridge* i a la *Lasso*. Observant el gràfic de l'evolució dels coeficients de les variables per diversos valors del logaritme de λ, sembla que aquests es comencen a estabilitzar aproximadament en el -5:

Gràfic 5.22. Elastic Net regression ($\alpha = 0'50$)*: Evolució dels coeficients I.*



En el següent gràfic, el valor mínim i òptim de λ es situa entorn el $\log(\lambda)$ de -2'25 i -1'65, respectivament:

Gràfic 5.23. Elastic Net regression ($\alpha = 0'50$)*: Valors de* λ *I.*



El valor mínim exacte de λ és de 0'1033 mentre que l'òptim pren un valor de 0'2386. S'ha creat un model per cada λ, el que conté la mínima presenta 36 coeficients iguals a zero, en canvi, el model amb λ òptima en presenta 47.

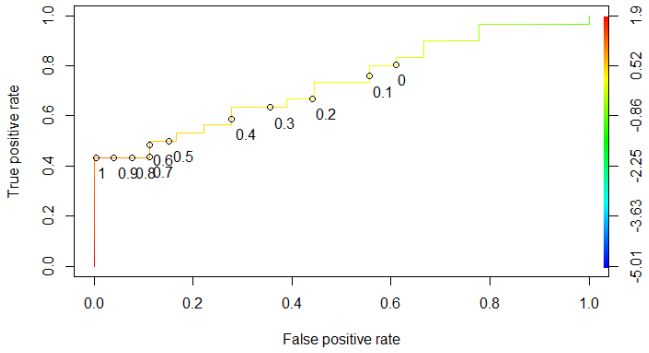
A continuació, es mostren els resultats de les mesures de predicció dels models per avaluar-los:

Taula 5.15. Elastic Net regression ($\alpha = 0'50$)*: Mesures de predicció I.*

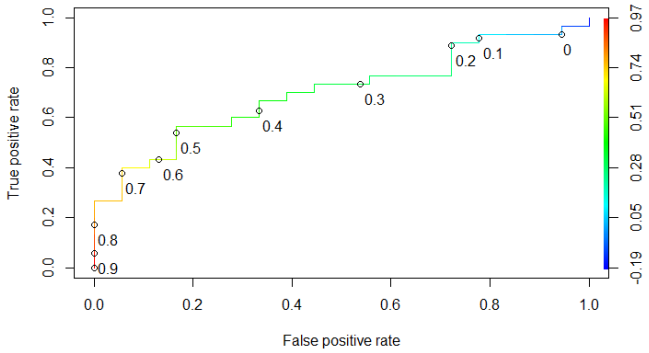
	Exactitud	Taxa d'error	Sensibilitat	Especificitat	Precisió	VPN
λ mínima	64,58%	35,42%	50,00%	88,89%	88,24%	51,61%
λ òptima	64,58%	35,42%	53,33%	83,33%	84,21%	51,72%

En aquest cas, s'ha obtingut que els dos models tenen la mateixa taxa d'error, però el model que conté el valor mínim de λ té un major percentatge tant d'especificitat com de precisió. Per aquest motiu es realitzarà la corba *ROC* per als dos models:

Gràfic 5.24. Elastic Net regression ($\alpha = 0'50$)*: Corba ROC amb* λ *mínima I.*



Gràfic 5.25. Elastic Net regression ($\alpha = 0'50$)*: Corba ROC amb* λ *òptima I.*

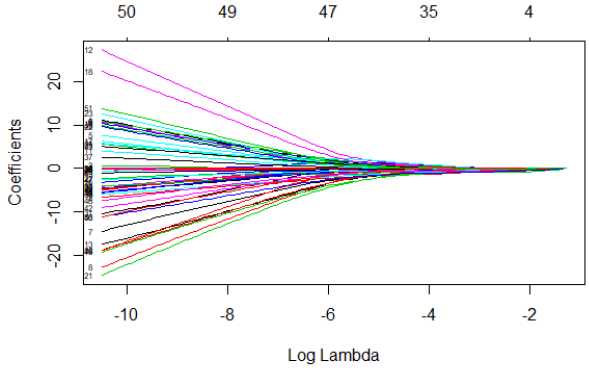


El valor de *l'AUC* per el model amb λ mínima és de 0'73 i el de λ òptima de 0'70. Per tant, s'ha corroborat que el primer model és millor.

- $\alpha = 0'75$

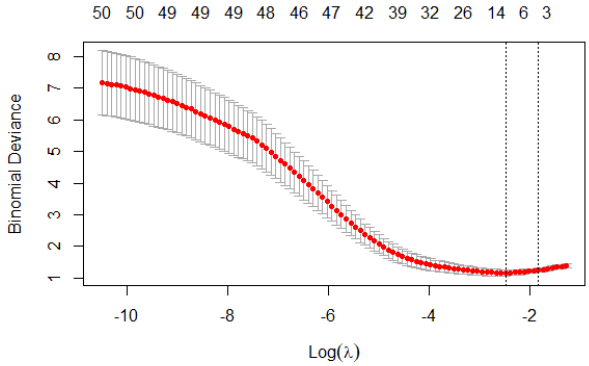
El paràmetre α pren un valor més proper a 1, per tant, se li dona més pes a la penalització *Lasso*. Observant el gràfic de l'evolució dels coeficients de les variables per diversos valors del $\log(\lambda)$, sembla que aquests es comencen a estabilitzar aproximadament en el -5:

Gràfic 5.26. Elastic Net regression $(\alpha = 0'75)$*: Evolució dels coeficients I.*



Respecte la gràfica per trobar els valors adequats de λ, es pot observar que el mínim i l'òptim es situen al voltant del $\log(\lambda)$ de -2'35 i -1'90, respectivament:

Gràfic 5.27. Elastic Net regression $(\alpha = 0'75)$*: Valors de* λ *I.*



El valor mínim de λ és de 0'0829 mentre que l'òptim val 0'1591. S'ha creat els dos models i el que conté la mínima presenta 40 coeficients iguals a zero, en canvi, amb λ òptima s'han donat 48 casos.

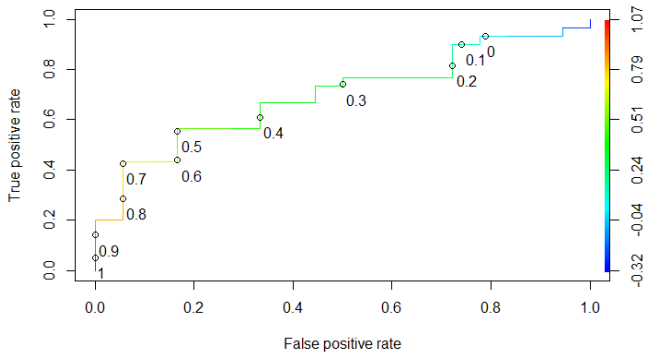
Els resultats de l'avaluació dels models són els següents:

Taula 5.16. Elastic Net regression $(\alpha = 0'75)$*: Mesures de predicció I.*

	Exactitud	Taxa d'error	Sensibilitat	Especificitat	Precisió	VPN
λ mínima	60,42%	39,58%	50,00%	77,78%	78,95%	48,28%
λ òptima	64,58%	35,42%	53,33%	83,33%	84,21%	51,72%

El model amb λ òptima conté una major exactitud i, per tant, una menor taxa d'error. En aquesta ocasió, s'ha realitzat la corba *ROC* del model que conté el valor òptim de λ on s'ha obtingut un *AUC* de 0'70:

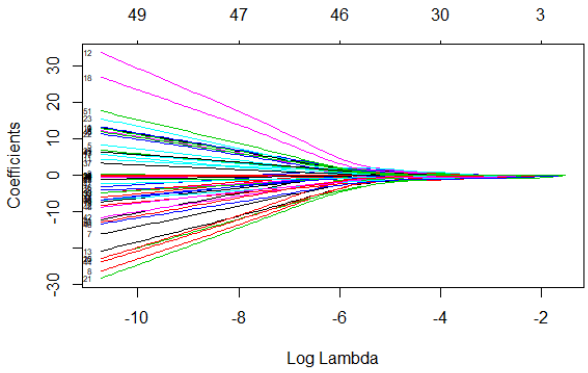
Gràfic 5.28. Elastic Net regression ($\alpha = 0'75$): Corba ROC I.



- $\alpha = 0'95$

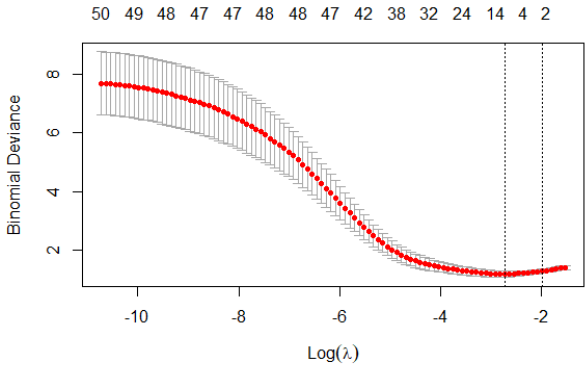
L'últim valor de α que s'analitzarà és de 0'95, el qual és molt proper a 1 i, per tant, també se li dona més pes a la penalització *Lasso*. El gràfic de l'evolució dels coeficients de les variables per diversos valors del $\log(\lambda)$ presenta la següent forma:

Gràfic 5.29. Elastic Net regression ($\alpha = 0'95$): Evolució dels coeficients I.



Sembla que aquests es comencen a estabilitzar en el -4'5, aproximadament. Respecte la gràfica per trobar el valor adequat de λ:

Gràfic 5.30. Elastic Net regression ($\alpha = 0'95$): Valors de λ I.



La λ mínima i òptima es situen entorn un valor del $\log(\lambda)$ de -2'90 i -2, respectivament. El valor mínim exacte de λ és de 0'0655 mentre que l'òptim és de 0'1378. S'ha creat un model per cada λ, el que conté la mínima presenta 41 coeficients iguals a zero, mentre que el model amb λ òptima 48 coeficients valen zero.

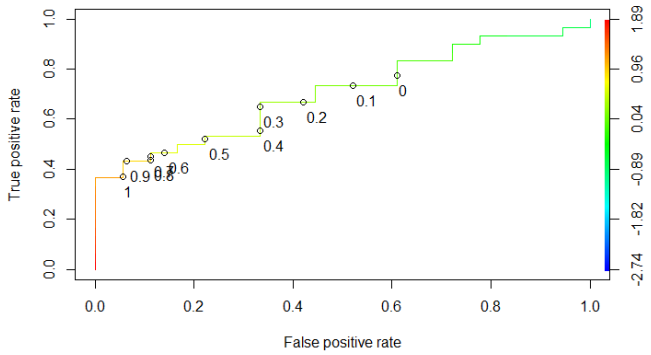
Els resultats de les mesures de predicció per a cadascun dels models són els següents:

Taula 5.17. Elastic Net regression ($\alpha = 0'95$)*: Mesures de predicció I.*

	Exactitud	Taxa d'error	Sensibilitat	Especificitat	Precisió	VPN
λ mínima	60,42%	39,58%	50,00%	77,78%	78,95%	48,28%
λ òptima	58,33%	41,67%	46,67%	77,78%	77,78%	46,67%

S'ha obtingut que el model amb λ mínima conté una menor taxa d'error i en conseqüència, una major exactitud. Respecte la corba *ROC*, que presenta un *AUC* de 0'71, és la següent:

Gràfic 5.31. Elastic Net regression ($\alpha = 0'95$)*: Corba ROC I.*



Entre tots els models realitzats amb els diversos valors del paràmetre α, es considera que el millor model és el que conté un valor α de 0'10 ja que obté una exactitud del 70%, tant per el valor de λ mínim com òptim, i un *AUC* de 0'80. En canvi, quan es du a terme l'avaluació del model per una α propera 1, s'obté una taxa d'error major i un pitjor valor de *l'AUC*.

Aquest fet podria interpretar-se com que el mètode de la regressió *Ridge* és el que ens proporciona una millor predicció per a aquesta base de dades d'entre els tres mètodes de penalització aplicats.

5.1.5. CONCLUSIÓ

Tal i com s'ha comentat a l'apartat anterior, es podria intuir que el model que obté unes millors prediccions és el de *Ridge regression* amb λ mínima. S'han comparat els diversos resultats i efectivament, aquest mètode de penalització presenta una menor taxa d'error en les prediccions, aquesta és del 25%, i un *AUC* del 0'80, el qual indica que la proba és bastant eficient. El segon mètode que obté un menor percentatge d'error és *l'Elastic Net* amb una α de 0'10 i un *AUC* de 0'80. La diferència en la taxa d'error d'aquests dos models és del 4%.

L'inconvenient de la *Ridge regression* és que en cap cas les estimacions dels coeficients de les variables prenen un valor exacte de zero, cosa que dificulta la interpretació perquè s'hauria de tenir en compte els 52 atributs, per tant, podria ser millor aplicar el mètode *Elastic Net* en aquest *dataset,* ja que 14 coeficients prenen valor de zero i no hi hauria tants atributs a ser interpretats.

A continuació, es comparen els dos millors models de penalització mencionats amb el model obtingut amb el mètode *Stepwise*:

Taula 5.18. Comparació mètode Stepwise, Ridge regression i Elastic Net regression.

	Exactitud	Taxa d'error	Sensibilitat	Especificitat	Precisió	VPN	AUC
Stepwise	68,75%	31,25%	76,67%	55,56%	74,19%	58,82%	0,82
Ridge	75,00%	25,00%	66,67%	88,89%	90,91%	61,54%	0,77
Elastic Net	70,83%	29,17%	60,00%	88,89%	90,00%	57,14%	0,80

Tot i que el mètode *Stepwise* ha obtingut un *AUC* més elevat, no hi ha quasi diferència entre els altres mètodes. També, és el que obté una major taxa d'error, per tant, la regressió *Ridge* i *Elastic Net* serien una bona opció.

Tal i com s'ha comentat al punt 4, els mètodes de penalització redueixen la variabilitat de les estimacions. Com que la funció *glmnet* no proporciona els intervals de confiança ni els errors estàndard d'aquests, s'han hagut de calcular mitjançant *bootstrap.*

- Intervals de confiança de les diverses variables amb el mètode *Stepwise:*

Taula 5.19. Stepwise: Intervals de confiança (95%) I.

	2,5%	97,5%
Intercept	11,9908	38,3936
Alk	-3,0718	-0,5372
IntNorRat	-4,6827	-1,0913
Alp	-0.0002	0,0000
Sym1	-6,5154	-1,7765
CRen1	-5,8131	-1,0573
cigpy	-0,0976	-0,0107
Art1	0,3837	3,4958
Smo1	0,7342	5,3197
Agedia	-0,1331	-0,0191
HepBSur1	-0,1086	3,5718
Aspa	-4,1803	-0,5073
Ala	0,1511	3,5430
HepC1	-3,6791	-0,2316
Thr1	-3,3129	0,1820

- Intervals de confiança dels coeficients dels atributs amb el mètode *Ridge regression:*

Taula 5.20. Ridge regression: Intervals de confiança (95%).

	2,5%	97,5%
Intercept	1,0912	3,8791
Gender1	0,0196	0,2978
Sym1	-0,5153	-0,2923
Alc1	-0,0839	0,1289
HepBSur1	0,0440	0,3661
HepBAnt1	-0,3560	-0,0249
HepBCore1	-0,0748	0,1477
HepC1	-0,3822	-0,1143
Cir1	-0,0701	0,3599
End1	0,0023	0,3330
Smo1	-0,0042	0,2045
Dia1	-0,1758	0,0806
Obe1	0,0109	0,2927

	2,5%	97,5%
Hem1	-0,4542	-0,0902
Art1	0,0330	0,2648
CRen1	-0,5648	-0,1419
HIV1	-0,5232	0,0349
Non1	0,1401	0,4798
EVar1	0,0245	0,2218
Spl1	-0,1171	0,0846
PHyp1	-0,0141	0,1909
Thr1	-0,4109	-0,1249
LMet1	-0,4139	-0,1024
Rad1	-0,0249	0,2207
Agedia	-0,0133	-0,0039
Alcpd	-0,0012	0,0002

- Intervals de confiança dels coeficients de les variables amb el mètode *Elastic Net regression:*

Taula 5.21. Elastic Net regression: Intervals de confiança (95%).

	2,5%	97,5%
Intercept	1,6052	4,4674
Gender1	-0,0465	0,2227
Sym1	-0,5942	-0,2686
Alc1	-0,0683	0,0683
HepBSur1	-0,0273	0,3014
HepBAnt1	-0,3155	0,0317
HepBCore1	-0,0804	0,0804
HepC1	-0,3610	-0,0427
Cir1	-0,1471	0,2484
End1	-0,1023	0,2081
Smo1	-0,0326	0,1606
Dia1	-0,0996	0,0996
Obe1	-0,0832	0,1661

	2,5%	97,5%
Hem1	-0,3507	0,0377
Art1	-0,0412	0,1859
CRen1	-0,5275	-0,0281
HIV1	-0,2844	0,1638
Non1	0,0009	0,3688
EVar1	-0,0630	0,1244
Spl1	-0,0631	0,0631
PHyp1	-0,0789	0,0839
Thr1	-0,3716	-0,0242
LMet1	-0,4580	-0,0732
Rad1	-0,1084	0,1084
Agedia	-0,0122	-0,0009
Alcpd	-0,0006	0,0006

S'observa que les estimacions dels coeficients de les variables per el mètode *Stepwise* tenen una major variabilitat ens el intervals de confiança que no pas la regressió *Ridge* i *Elastic Net.* Per tant, aquests últims mètodes corregeixen el problema de la variabilitat i conseqüentment, la multico·linealitat.

5.2. BASE DE DADES: MALALTIA CARDIOVASCULAR

La segona base de dades a estudiar aporta informació de diversos individus sobre si pateixen alguna malaltia cardiovascular. Aquesta consta de 12 variables i 70.000 observacions.

Respecte els atributs, 7 són categòrics (5 amb dos nivells i 2 amb més de 3 nivells) els quals proporcionen la següent informació de cada individu:

- *Gender*: el sexe.
- *Cho*: grau de colesterol.
- *Glu*: grau de glucosa.
- *Smoke*: si fuma.
- *Alcohol*: si consumeix alcohol.
- *PhysAct*: si realitza activitats físiques.
- *Cardio*: si pateix d'alguna malaltia cardiovascular.

Les variables categòriques esmentades estan categoritzades de la següent manera:

- La variable *gender* conté dos nivells: *woman* si és dona i *man* si és home.
- Les variables *cho* i *glu* contenen 3 nivells:

Taula 5.22. Descripció nivells de les variables cho i glu.

1	*Normal*
2	*Above normal*
3	*Well above normal*

- La resta de variables estan categoritzades com a *yes* aquells grups que tenen més risc, és a dir, si l'individu fuma està categoritzat com a *yes* i *no* en cas contrari.

Respecte les variable numèriques, la base de dades en conté 5 i proporcionen la següent informació de cada observació:

- *Age*: l'edat en dies.
- *Height*: l'altura en centímetres.
- *Weight*: el pes en kilograms.
- *Systolic*: la pressió sistòlica a la sang en mm Hg.
- *Diastolic*: la pressió diastòlica a la sang en mm Hg.

La variable resposta d'aquest anàlisi és *cardio* que informa si l'individu presenta alguna malaltia cardiovascular o no.

5.2.1. ANÀLISI DESCRIPTIU

Per a realitzar l'anàlisi descriptiu de les variables, s'ha seguit el mateix procediment que en la base de dades anterior, s'ha dividit el *dataset* en variables categòriques i numèriques.

5.2.1.1. VARIABLES CATEGÒRIQUES

Per a les variable categòriques s'ha observat quants individus pertanyen a cada nivell:

- Variables binàries:

Taula 5.23. Descriptiva de les variables binàries II.

	no	yes
gender	45.530	24.470
smoke	63.831	6.169
alcohol	66.236	3.764
physAct	13.739	56.261

Els nivells estan bastant descompensats, és a dir, no contenen aproximadament el mateix nombre d'individus. Per exemple, la variable s*moke* conté el 91'19% al nivell *no* (no fumadors).

- Variables ordinals:

Taula 5.24. Descriptiva de les variables ordinals.

	1	2	3
cho	52.385	9.549	8.066
glu	59.479	5.190	5.331

Amb les variables ordinals succeeix el mateix que amb les binàries, els nivells estan descompensats. La categoria 1 (nivells normals) conté més del 70% de les observacions, mentre que el percentatge restant el conten el nivell 2 i 3.

- Variable resposta:

Taula 5.25. Descriptiva de la variable resposta II.

	no	yes
cardio	35.021	34.979

Pel que fa a la variable resposta, és l'única amb els nivells compensats, els individus que no presenten cap malaltia cardiovascular representen el 50'03% de les dades i els que sí pateixen una malaltia són el 49'97%.

5.2.1.2. VARIABLES NUMÈRIQUES

Respecte la descriptiva de les variables numèriques s'han obtingut els següents resultats:

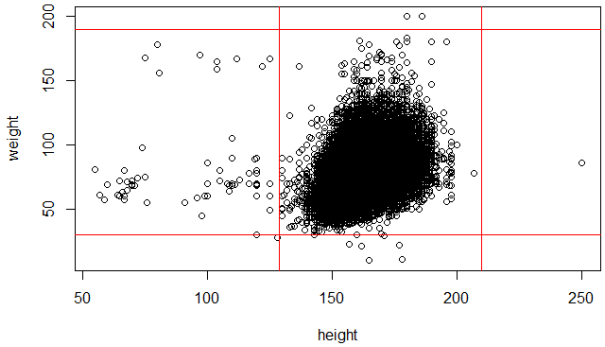
Taula 5.26. Descriptiva de les variables numèriques II.

	Mitjana	Variància	Desv. Típica	Covariància	Min	Q1	Mediana	Q3	Max
age	53,58	47,07	6,86	0,13	29	49	54	59	65
height	164,36	67,41	8,21	0,05	55	159	165	170	250
weight	74,21	207,24	14,4	0,19	10	65	72	82	200
systbl	128,82	23.719,52	154,01	1,2	-150	120	120	140	16.020
diastbl	96,63	35.521,89	188,47	1,95	-70	80	80	90	11.000

Les variables *height* i *weight* obtenen valors estranys, ja que l'altura i el pes mínim són de 55 cm i 10 kg, respectivament, mentre que l'edat mínima és de 29 anys. Les variables *systolic* i *diastolic* també presenten dades impròpies, donat que un individu no pot contenir valors negatius a la pressió sistòlica o diastòlica de la sang i valors extremadament baixos que duen a pensar que hi ha errors en les dades. Per tant, s'ha fet un anàlisi d'aquests atributs dos a dos:

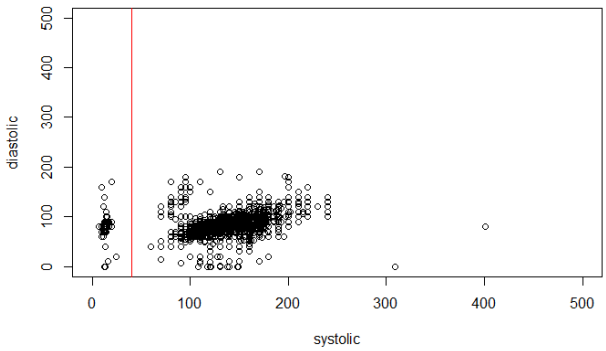
- *Height vs weight:*

Gràfic 5.32. Anàlisi height vs weight.



- *Systolic vs diastolic:*

Gràfic 5.33. Anàlisi systolic vs diastolic.



Per a obtenir una millor base de dades i un anàlisi més acurat, s'han eliminat totes aquelles observacions que s'allunyaven del núvol de punts central. La base de dades final consta de 68.700 observacions en comptes de 70.000. Aquestes dades eliminades representen un 0'019% del *dataset*.

Una vegada dut a terme aquest procés, s'ha tornat a realitzar la descriptiva de les variables numèriques i s'han obtingut els següents resultats:

Taula 5.27. Descriptiva de les variables numèriques III.

	Mitjana	Variància	Desv. Típica	Covariància	Min	Q1	Mediana	Q3	Max
age	52,79	45,75	6,76	0,13	29	48	53	58	64
height	164,44	61,60	7,85	0,05	130	159	165	170	207
weight	74,11	203,70	14,27	0,19	30	65	72	82	183
systbl	126,62	281,06	16,77	0,13	60	120	120	140	240
diastbl	81,37	94,67	9,73	0,12	15	80	80	90	190

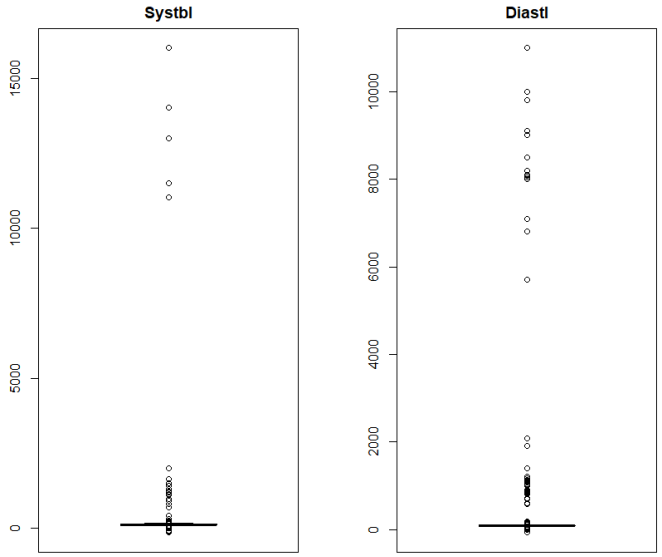
Els atributs presenten uns valors més normals que no pas abans. On més es veu reflectit aquest canvi és a la variància de les variables *weight* i *systolic,* ja que al no haver tanta diferència entre el mínim i el màxim aquesta ha disminuït considerablement.

5.2.2. PROCESSAMENT DE LA BASE DE DADES

5.2.2.1. OUTLIERS

Totes les variables numèriques presenten *outliers,* però les que més destaquen són *weight* i *systolic*. Aquestes dues són les que obtenen un valor més elevat de la variància comparat amb la resta de variables degut a aquests valors extrems. A continuació, es mostren gràficament aquests *outliers:*

Gràfic 5.34 . Outliers II.



5.2.2.2. TRANSFORMACIONS

Es va realitzar un gràfic de cadascuna de les variables numèriques per a veure si aquestes necessitaven alguna transformació per a millorar la base de dades tal i com es va dur a terme amb l'exemple del carcinoma hepatocel·lular, però en aquest cas, les dades estan bastant ben distribuïdes, de manera que no s'ha realitzat cap canvi a ningun atribut.

5.2.2.3. DESCRIPTIVA DELS NA

Aquesta base de dades no conté cap registre mancant i, per aquest motiu, no s'ha hagut de dur a terme un mètode d'imputació de *missings*.

5.2.3. APLICACIÓ DELS MÈTODES DE SELECCIÓ DE VARIABLES

El mètode aplicat de selecció de variables és el *Stepwise* que, com ja s'ha comentat a la part teòrica del treball, és una combinació del *Forward* i *Backward.*

5.2.3.1. CORRELACIÓ

Abans d'aplicar aquest mètode, es procedirà a analitzar la correlació entre les diverses variables a través dels VIF. Per a dur a terme aquest càlcul, s'ha creat un model que conté la variable *cardio* com a resposta i la resta d'atributs com a predictors i s'ha aplicat la funció *vif* de l'R. Els resultats obtinguts són els següents:

Taula 5.28. VIF de les variables II.

age	1,02
gender	1,48
height	1,49
weight	1,18
systolic	1,62
diastolic	1,60
cho	1,50
glu	1,48
smoke	1,25
alcohol	1,14
physAct	1,00

Seguint els criteris explicats en la base de dades de carcinoma hepatocel·lular, les variables *age* i *physAct* prenen un valor exacte de 1, la qual cosa implica que no hi ha correlació. La resta d'atributs presenten valors superiors a 1 però no s'allunyen molt d'aquest, tenen una lleu correlació entre ells. Es podria considerar que no hi ha multicol·linealitat.

5.2.3.2. MÈTODE STEPWISE

Tal i com es va realitzar en el primer *dataset*, s'ha dividit aquest en dos on les dades *train* contenen el 70% de les observacions i s'utilitzen per a crear els diversos models lineals generalitzats i, les dades *test* contenen l'altre 30% dels individus amb les quals s'avaluaran els diferents models creats.

S'ha creat el model complet amb el conjunt de dades *train* que conté totes les variables explicatives i també el model nul on només hi ha l'intercept. Un cop aplicat el mètode *Stepwise*, s'ha obtingut el següent model:

$$cardio \sim age + gender + height + weight + systolic + diastolic + cho + glu$$
$$+ smoke + alcohol + physAct$$

Conté 11 variables explicatives i totes han resultat significatives amb un nivell de significació del 5%. Els models creats s'han realitzat mitjançant la funció *glm* de *l'R* i aplicant la família binomial.

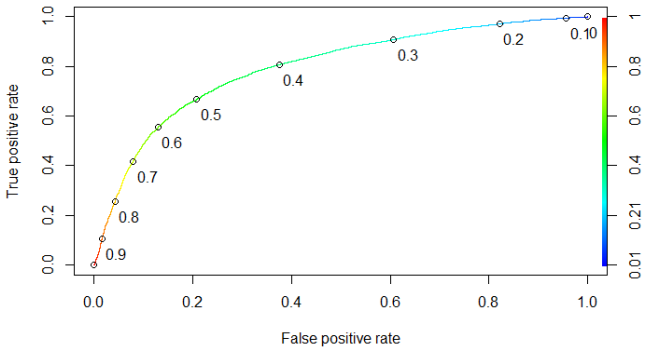
Per tal d'observar com es comporta el model s'ha creat la matriu de confusió amb les dades *test* i a partir d'aquests resultats, s'han calculat les següents mesures de predicció del model:

Taula 5.29. Stepwise: Mesures de predicció II.

Exactitud	Taxa d'error	Sensibilitat	Especificitat	Precisió	VPN
72,99%	27,01%	66,68%	79,18%	75,84%	70,80%

La corba *ROC,* amb un AUC de 0'79*,* presenta la següent forma:

Gràfic 5.35. Stepwise: Corba ROC II.



5.2.4. APLICACIÓ DELS MÈTODES DE PENALITZACIÓ

S'ha seguit el mateix procediment explicat en el punt 5.1.4, on s'ha utilitzat la funció *glmnet* per a aplicar els diversos mètodes de penalització.

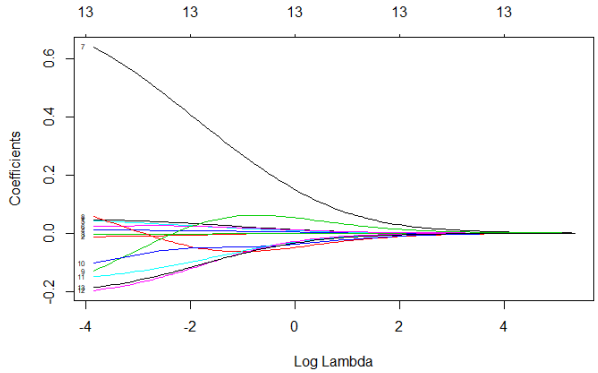
Per a trobar el valor adequat de λ, s'ha dut a terme mitjançant el mètode *cross-validation* dividint el *dataset train* en 10 parts.

5.2.4.1. RIDGE REGRESSION

El primer mètode de penalització aplicat al model lineal generalitzat és la *Ridge regression* on el paràmetre α pren valor de 0. Aquest model *glmnet* s'ha creat amb les dades *train.*

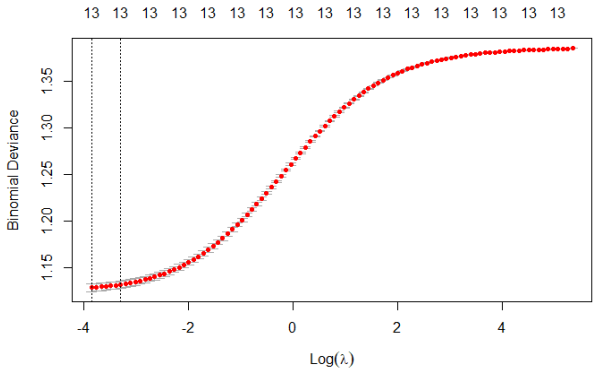
A continuació, es mostra el gràfic on es poden visualitzar els diversos valors dels coeficients a mesura que el valor del logaritme de λ va augmentant, aquests s'estabilitzen en un valor aproximat de 2:

Gràfic 5.36. Ridge regression: Evolució dels coeficients II.



En el següent gràfic es pot observar que el valor de λ mínim i òptim, per a obtenir l'error més petit, està situat entorn el $\log(\lambda)$ de -3'9 i -3'5, respectivament:

Gràfic 5.37. Ridge regression: Valors de λ II.



El valor mínim de λ pren un valor de 0'0211 i l'òptim de 0'0369. S'ha creat un model per a cadascun d'aquests valors on s'ha obtingut que cap coeficient pren valor exacte de zero, ja que el mètode de regressió *Ridge* el que fa és aproximar els coeficients de les variables a zero però sense donar cap valor nul.

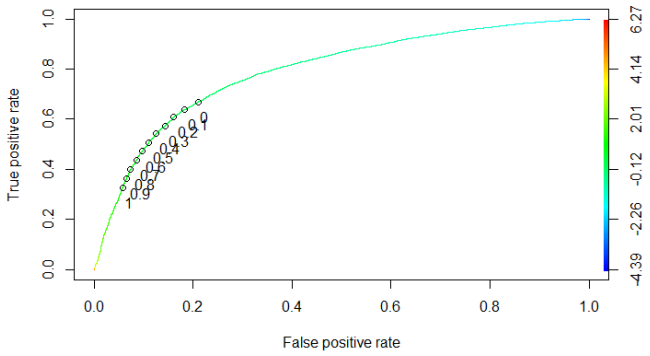
Per a saber la capacitat predictiva del model s'han calculat les diverses mesures d'avaluació mitjançant la matriu de confusió. S'han obtingut resultats molt similars per ambdós models:

Taula 5.30. Ridge regression: Mesures de predicció II.

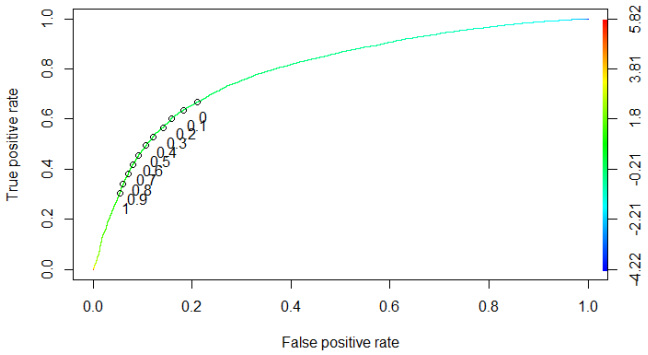
	Exactitud	Taxa d'error	Sensibilitat	Especificitat	Precisió	VPN
λ mínima	69,97%	30,03%	50,59%	88,97%	81,80%	64,76%
λ òptima	69,56%	30,44%	49,37%	89,34%	81,94%	64,30%

Com no hi ha quasi diferències, s'ha realitzat la corba *ROC* per cada model on s'ha obtingut una àrea per sota de la corba de 0'79 per als dos:

Gràfic 5.38. Ridge regression: Corba ROC amb λ mínima II.



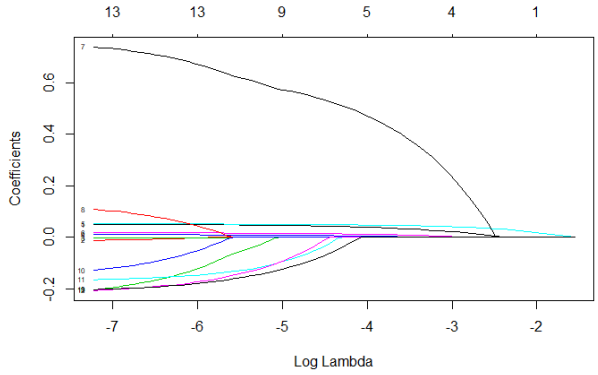
Gràfic 5.39. Ridge regression: Corba ROC amb λ òptima.



5.2.4.2. LASSO REGRESSION

Per a aplicar el mètode de regressió *Lasso* al model *glmnet* amb les dades *train,* s'ha donat un valor de 1 al paràmetre α. A continuació, es mostra l'evolució dels coeficients per a diversos valors de λ:

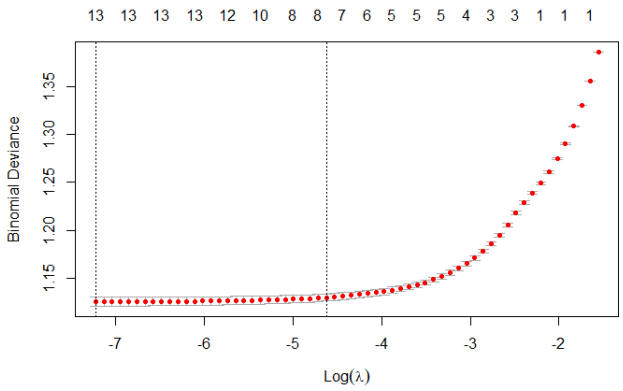
Gràfic 5.40. Lasso regression: Evolució dels coeficients II.



S'aprecia que, comparat amb la *Ridge regression*, els coeficients del mètode *Lasso* no s'estabilitzen tots a l'hora. Hi ha una gran diferència amb una de les estimacions, la qual tendeix a zero per un valor del log (λ) més elevat.

En el següent gràfic es mostren els valors mínim i òptim del log (λ) que estan situats al voltant de -7'3 i -4'6, respectivament. La λ mínima és de 0'0007 i l'òptima de 0'0098:

Gràfic 5.41. Lasso regression: Valors de λ II.



Per un valor de λ mínim cap coeficient val zero, en canvi, per el valor òptim 5 coeficients presenten un valor exacte de zero.

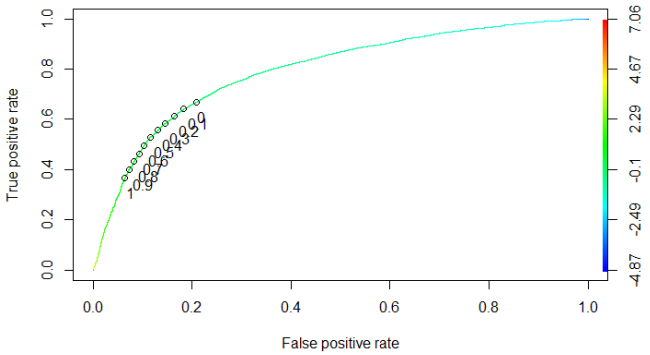
Les mesures de predicció dels models obtinguts són les següents:

Taula 5.31. Lasso regression: Mesures de predicció II.

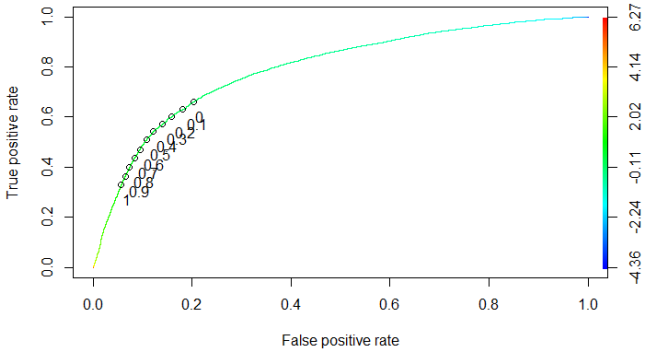
	Exactitud	Taxa d'error	Sensibilitat	Especificitat	Precisió	VPN
λ mínima	70,75%	29,25%	52,68%	88,46%	81,73%	65,61%
λ òptima	70,26%	29,74%	50,89%	89,23%	82,24%	94,96%

Succeeix el mateix que en la regressió *Ridge,* no hi ha quasi diferència en els resultats. Els dos models tenen una diferència en la taxa d'error de 0'49. Com que els resultats són molt similars entre els dos casos, s'ha dut a terme el gràfic de la corba *ROC* per cada un on s'ha obtingut un valor de *l'AUC* de 0'79 per ambdós:

Gràfic 5.42. Lasso regression: Corba ROC amb λ mínima II.



Gràfic 5.43. Lasso regression: Corba ROC amb λ òptima.



5.2.4.3. ELASTIC NET REGRESSION

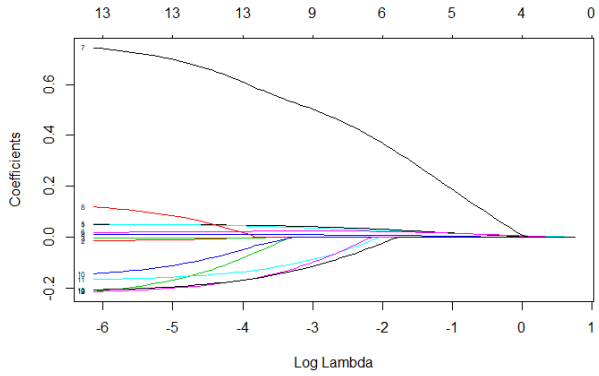
S'han realitzat diversos models per diferents valors de α per veure amb quin d'ells s'obté una millor predicció per al mètode *Elastic Net regression:*

- $\alpha = 0'10$

  El paràmetre α pren un valor molt proper a 0, per tant, se li dona més pes a la penalització

Ridge regression. L'evolució de les estimacions de les variables a mesura que augmenta el valor de λ presenta la següent forma:

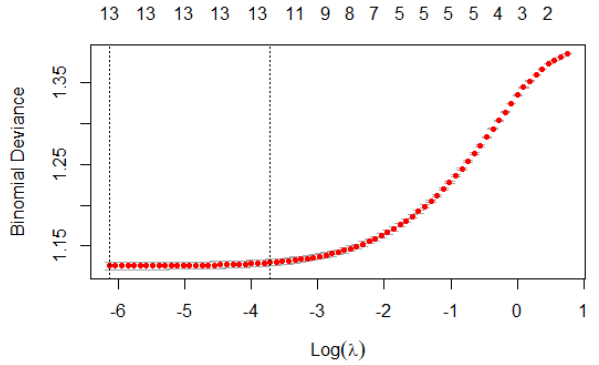
Gràfic 5.44. Elastic Net regression $(\alpha = 0'10)$*: Evolució dels coeficients II.*



Ocorre el mateix que amb el mètode *Lasso*, hi ha coeficients que es comencen a estabilitzar més tard que d'altres. Sobretot hi ha una gran diferència amb un dels coeficients comparat amb les altres estimacions.

El valor mínim i òptim del logaritme de λ, amb el que s'obté un error inferior, es situa al voltant de -6 i -3'7, respectivament:

Gràfic 5.45. Elastic Net regression $(\alpha = 0'10)$*: Valors de* λ *II.*



El valor de λ mínim és de 0'0022 i l'òptim de 0'0243. S'ha creat un model per a cada λ i s'ha observat que amb el mínim cap coeficient pren valor zero, mentre que en l'altre cas només 1 coeficient pren valor zero.

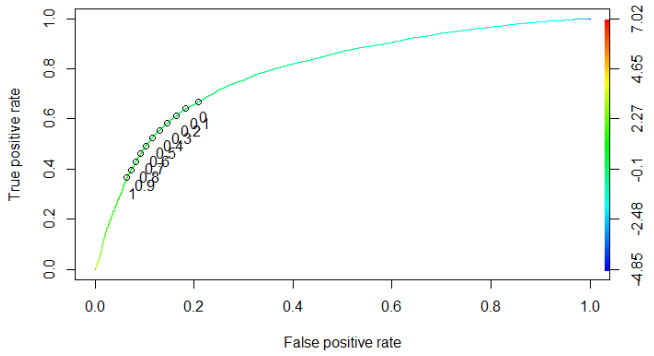
Les mesures d'avaluació dels dos models *glmnet* obtenen els següents resultats:

Taula 5.32. Elastic Net regression $(\alpha = 0'10)$*: Mesures de predicció II.*

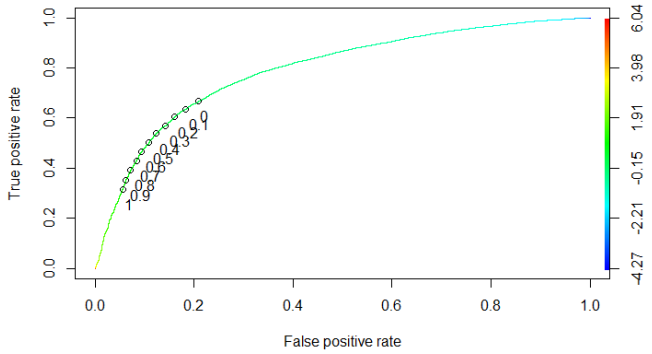
	Exactitud	Taxa d'error	Sensibilitat	Especificitat	Precisió	VPN
λ mínima	70,71%	29,29%	52,56%	88,49%	81,74%	65,56%
λ òptima	69,80%	30,20%	50,10%	89,12%	81,85%	64,57%

Tot i que la λ òptima considera que el coeficient d'una variable és de 0, el model amb la λ mínima obté una menor taxa d'error, encara que no hi ha molta diferència en els resultats. Per aquest motiu, s'ha realitzat la corba *ROC* per als dos models on han obtingut un *AUC* de 0'79:

Gràfic 5.46. Elastic Net regression ($\alpha = 0'10$): Corba ROC amb λ mínima II.



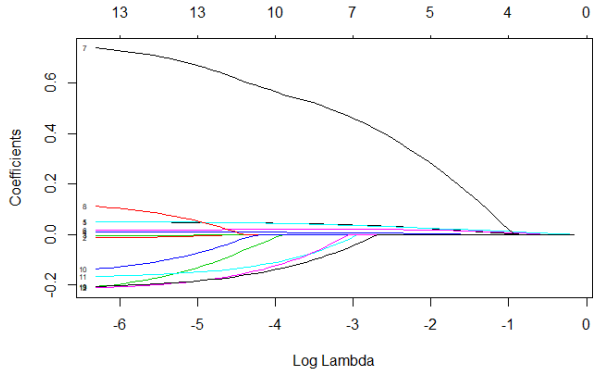
Gràfic 5.47. Elastic Net regression ($\alpha = 0'10$): Corba ROC amb λ òptima.



- $\alpha = 0'25$

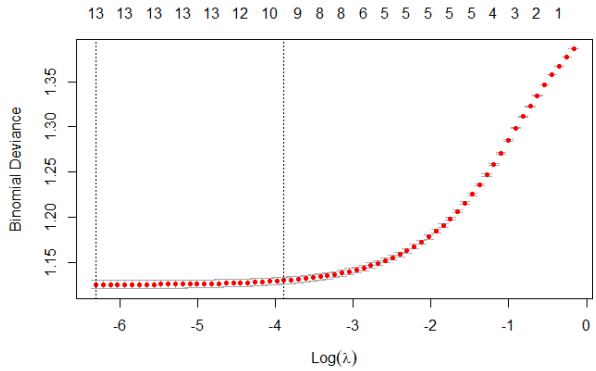
  En aquesta situació, el paràmetre α pren valor de 0'25 que és més proper a 0 que no pas a 1, per tant, es dona més pes a la regressió *Ridge*. L'evolució dels coeficients per a diversos valors de λ és molt similar que amb $\alpha = 0'10$:

Gràfic 5.48. Elastic Net regression ($\alpha = 0'25$): Evolució dels coeficients II.



El valor mínim i òptim del logaritme de λ, per a obtenir un error mínim, està situat al voltant de -6'5 i -3'9, respectivament:

Gràfic 5.49. Elastic Net regression $(\alpha = 0'25)$*: Valors de* λ *II.*



La λ mínima pren un valor de 0'0018, mentre que l'òptima val 0'0205. S'ha creat un model per cada valor de λ i s'ha obtingut que cap estimació dels coeficients de les variables prenen valor nul per la λ mínima, en canvi, hi ha 4 coeficients que valen zero per la λ òptima.

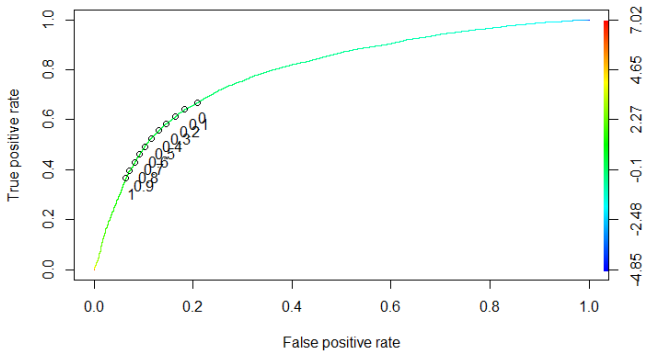
S'han calculat les diverses mesures de predicció, obtingudes a partir de la matriu de confusió, per avaluar els models:

Taula 5.33. Elastic Net regression $(\alpha = 0'25)$*: Mesures de predicció II.*

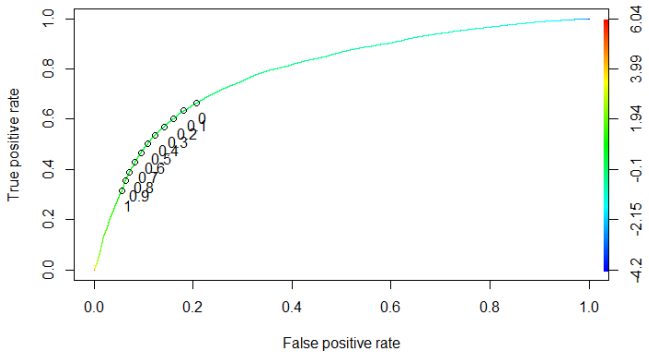
	Exactitud	Taxa d'error	Sensibilitat	Especificitat	Precisió	VPN
λ mínima	70,74%	29,26%	52,63%	88,48%	81,74%	65,59%
λ òptima	69,90%	30,10%	50,18%	89,22%	82,02%	64,63%

Com en els anteriors casos, no hi ha gran diferència en els resultats segons la λ i per aquesta raó, s'ha dut a terme els gràfic de la corba *ROC* per als dos models on s'ha obtingut un *AUC* de 0'79 per ambdós:

Gràfic 5.50. Elastic Net regression $(\alpha = 0'25)$*: Corba ROC amb* λ *mínima II.*



Gràfic 5.51. Elastic Net regression $(\alpha = 0'25)$*: Corba ROC amb* λ *òptima.*

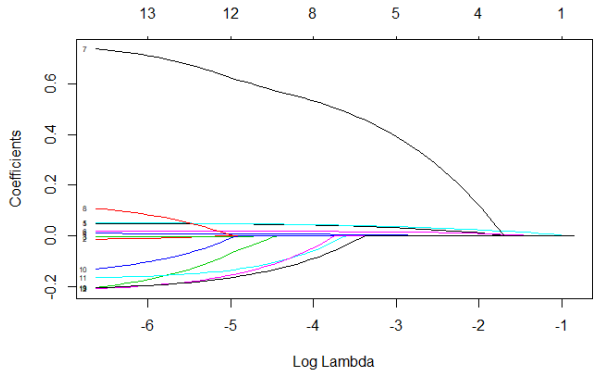


- $\alpha = 0'50$

El paràmetre α del model prendrà valor de 0'50, el que significa que se li dona el mateix pes a la penalització *Ridge* i a la *Lasso.*

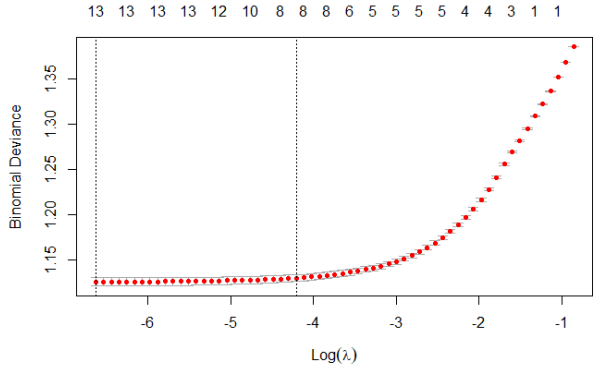
En l'evolució dels coeficients succeeix el mateix que en el cas anterior, un dels coeficients tendeix cap a 0 més tard que la resta:

Gràfic 5.52. Elastic Net regression $(\alpha = 0'50)$*: Evolució dels coeficients II.*



El logaritme de la λ mínima i òptima, per a minimitzar l'error, està situat entorn el valor -7 i -4'2, respectivament:

Gràfic 5.53. Elastic Net regression $(\alpha = 0'50)$*: Valors de* λ *II.*



El valor de λ mínima és de 0'0013 i l'òptima de 0'0148. Amb aquests valors s'han creat dos models on s'ha obtingut que cap coeficient pren valor de zero amb el valor mínim i que 5 són nuls amb l'òptim.

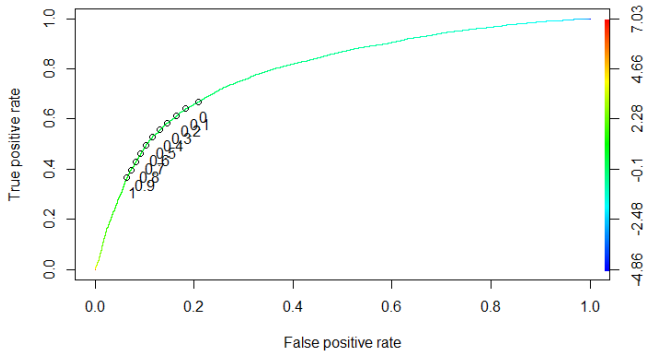
S'ha dut a terme els càlculs de les mesures d'avaluació del model mitjançant la matriu de confusió i s'han obtingut els següents resultats:

Taula 5.34. Elastic Net regression $(\alpha = 0'50)$*: Mesures de predicció II.*

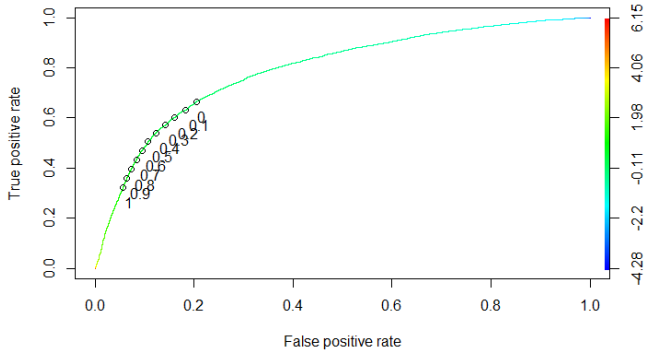
	Exactitud	Taxa d'error	Sensibilitat	Especificitat	Precisió	VPN
λ mínima	70,76%	29,24%	52,68%	88,48%	81,76%	65,61%
λ òptima	70,11%	29,89%	50,53%	89,29%	82,21%	64,81%

Es pot veure que quasi no hi ha diferència entre la taxa d'error dels dos models, aquesta és de 0'65. Les corbes *ROC*, amb un AUC de 0'79, presenten les següents formes:

Gràfic 5.54. Elastic Net regression ($\alpha = 0'50$): Corba ROC amb λ mínima II.



Gràfic 5.55. Elastic Net regression ($\alpha = 0'50$): Corba ROC amb λ òptima II.

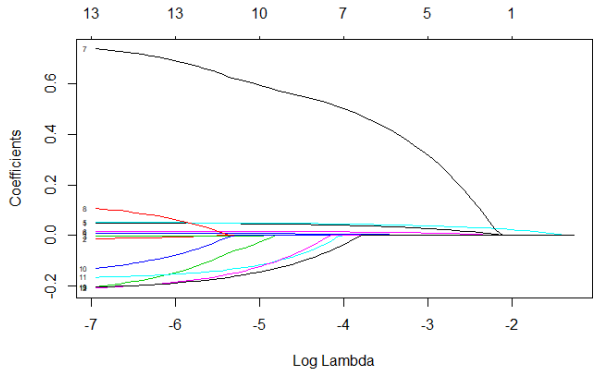


- $\alpha = 0'75$

  Com que el paràmetre α pren un valor que es situa més a prop de 1, se li dona més pes a la penalització *Lasso.*

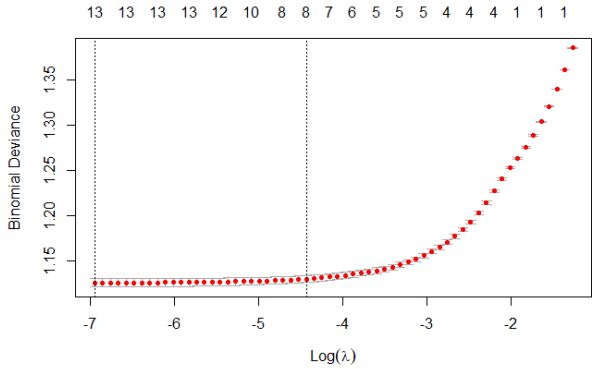
  L'evolució de les estimacions dels coeficients a mesura que va augmentant el valor de λ presenta la següent forma:

Gràfic 5.56. Elastic Net regression ($\alpha = 0'75$): Evolució dels coeficients II.



Respecte el valor logarítmic de λ mínima i òptima, de forma que s'aconsegueixi el menor error possible, s'ha obtingut que estan situats al voltant de -7 i -6, respectivament:

Gràfic 5.57. Elastic Net regression ($\alpha = 0'75$): Valors de λ II.



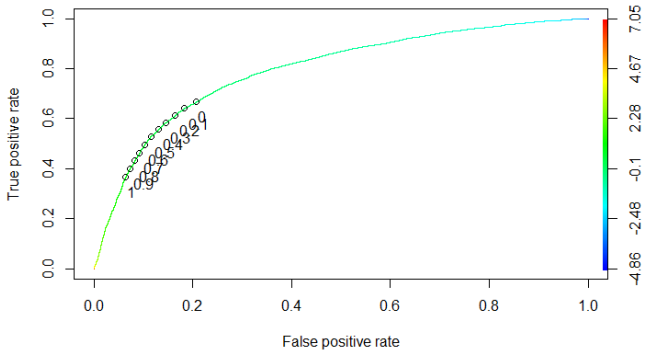
La λ mínima obté un valor de 0'0001 i l'òptima de 0'0119. S'han creat els dos models on cap estimació dels coeficients prenen un valor exacte de 0 per la mínima i 5 valen zero per l'òptima. Al realitzar els càlculs de les mesures de predicció per avaluar els models, s'han obtingut resultats molt similars:

Taula 5.35. Elastic Net regression ($\alpha = 0'75$): Mesures de predicció II.

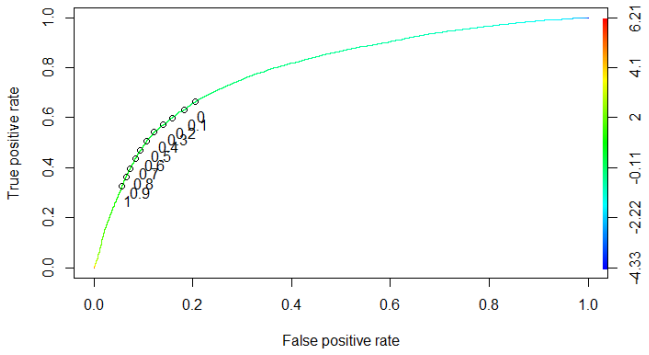
	Exactitud	Taxa d'error	Sensibilitat	Especificitat	Precisió	VPN
λ mínima	70,74%	29,26%	52,67%	88,45%	81,72%	65,60%
λ òptima	70,19%	29,81%	50,68%	89,31%	82,28%	64,88%

Les corbes *ROC,* amb un AUC de 0'79, presenten les següent formes:

Gràfic 5.58. Elastic Net regression ($\alpha = 0'75$): Corba ROC amb λ mínima.



Gràfic 5.59. Elastic Net regression ($\alpha = 0'75$): Corba ROC amb λ òptima II.

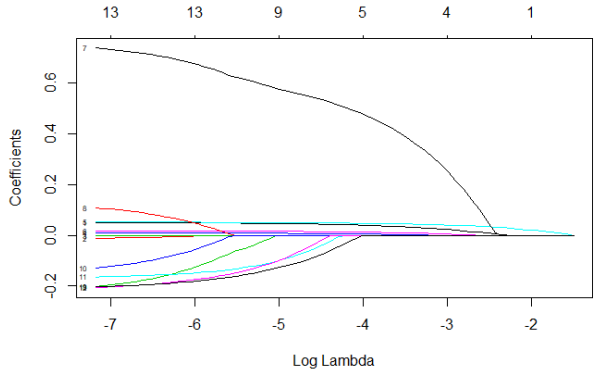


- $\alpha = 0'95$

L'últim valor del paràmetre α aplicat a la funció *glmnet* és de 0'95, el qual li dona molt pes a la penalització *Lasso.*

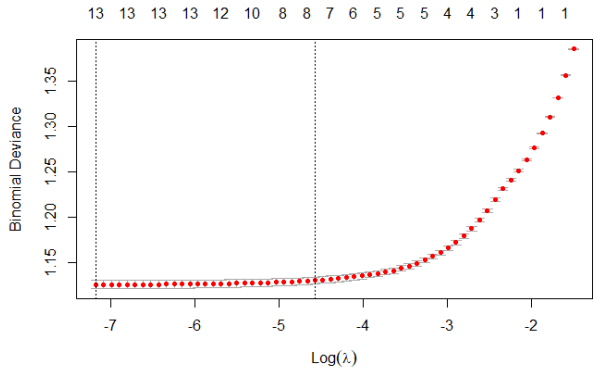
Observant el gràfic de l'evolució dels coeficients, es pot veure que ocorre el mateix que en tots els casos anteriors, un d'ells triga més en estabilitzar-se que no pas els altres:

Gràfic 5.60. Elastic Net regression $(\alpha = 0'95)$*: Evolució dels coeficients II.*



El logaritme de λ mínima està situat entorn el -7'2 i l'òptima al voltant de -4'5 tal i com es mostra a continuació:

Gràfic 5.61. Elastic Net regression $(\alpha = 0'95)$*: Valors de* λ *II.*



Aquests dos paràmetres prenen un valor de 0'0008 i 0'0103, respectivament. S'ha creat un model per cadascun d'ells i s'ha obtingut que amb λ mínima cap coeficient pren un valor exacte de zero, en canvi, amb λ òptima 5 presenten un valor nul.

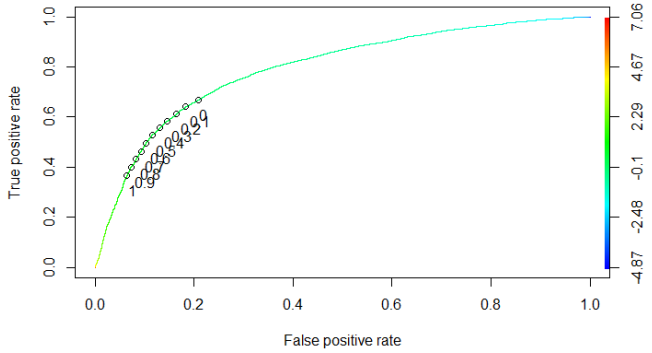
Respecte les mesures de predicció per avaluar ambdós models, els resultats són molts similars:

Taula 5.36. Elastic Net regression $(\alpha = 0'95)$*: Mesures de predicció II.*

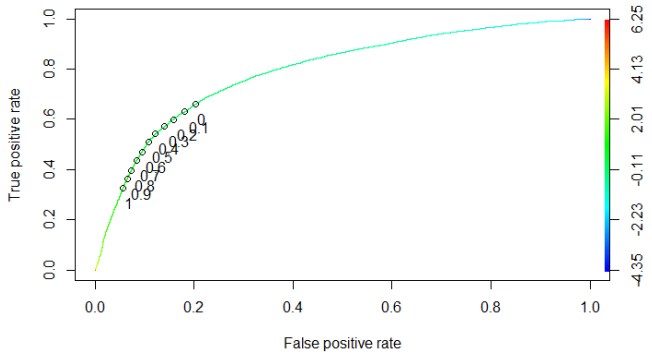
	Exactitud	Taxa d'error	Sensibilitat	Especificitat	Precisió	VPN
λ mínima	70,75%	29,25%	52,68%	88,46%	81,73%	65,61%
λ òptima	70,24%	29,76%	50,85%	89,24%	82,24%	64,95%

Les corbes *ROC,* amb un *AUC* de 0'79, han quedat representades de la següent manera:

Gràfic 5.62. Elastic Net regression ($\alpha = 0'95$)*: Corba ROC amb* λ *mínima II.*



Gràfic 5.63. Elastic Net regression ($\alpha = 0'95$)*: Corba ROC amb* λ *òptima.*



D'entre tots els models realitzats amb la funció *glmnet* per als diversos valors del paràmetre α, no hi ha quasi diferència entre ells ja que tots obtenen una taxa d'error i un valor de *l'AUC* similars. Respecte els valors de λ, es podria dir que és millor la mínima ja que obté una major exactitud però per dècimes, per tant, tampoc hi ha molta diferència entre escollir un valor o l'altre.

5.2.5. CONCLUSIÓ

Un cop aplicat el mètode de selecció de variables *Stepwise* i els mètodes de penalització *(Ridge, Lasso i Elastic Net),* s'ha pogut observar que no caldria aplicar un paràmetre de regularització, ja que el *Stepwise* ha obtingut una millor exactitud (amb poca diferència respecte la resta de mètodes). Pel que fa als valors de *l'AUC,* tots els mètodes han obtingut un valor aproximat de 0'79.

Quan es va estudiar la correlació mitjançant els VIF, es va obtenir que no hi havia multicol·linealitat entre els atributs i, per tant, es podia intuir que no hi hauria una gran variabilitat en les estimacions dels paràmetres, la qual cosa feia pensar que no seria necessari aplicar un mètode de penalització.

A continuació, es mostren els intervals de confiança obtinguts amb el mètode tradicional i es pot veure que no hi ha una gran variabilitat en les estimacions dels coeficients de les variables, al contrari que passava en el primer *dataset* explicat:

Taula 5.37. Stepwise: Intervals de confiança (95%) II.

	2,5%	97,5%
Intercept	-10,9268	-9,8567
systolic	0,0502	0,0542
age	0,0470	0,0533
cho.L	0,7111	0,8294
cho.Q	0,0782	0,1949
weight	0,0090	0,0122
diastolic	0,0144	0,0206
physActyes	-0,2651	-0,1626
glu.L	-0,3038	-0,1735
glu.Q	-0,2387	-0,0865
smokeyes	-0,2576	-0,0998
alcoholyes	-0,3214	-0,1247
height	-0,0067	-0,0010

6. CONCLUSIONS

Un cop realitzat l'anàlisi pràctic en *R* de les dues bases de dades s'ha estudiat l'objectiu establert en aquest treball: *Funcionen millor els mètodes de penalització que els mètodes tradicionals de selecció de variables?* També s'ha pogut corroborar i contrastar certs punts que s'han comentat a la part teòrica d'aquest estudi.

Respecte la primera base de dades, la qual tracta sobre el càncer de carcinoma hepatocel·lular i conté un nombre elevat d'atributs (50 variables) i poques observacions (165 individus), s'ha realitzat la descriptiva corresponent, les transformacions necessàries a les diverses variables, la imputació de dades mancants que s'ha cregut convenient i l'aplicació dels diversos mètodes tant de selecció de variables com de penalització. S'ha arribat a la conclusió de que funcionen millor els mètodes de penalització, és a dir, aplicar un paràmetre de regularització a les dades. Això és degut a que hi ha multicol·linealitat entre les variables i, per tant, quan es presenta aquest problema, la variància de les estimacions augmenta considerablement i, com a conseqüència, l'amplada dels intervals de confiança és més gran.

Tal i com s'ha explicat al punt 4, tant el mètode de regressió *Ridge, Lasso* i *Elastic Net* corregeixen aquest problema d'augment de la variabilitat, fent que els coeficients de les variables que aporten menys informació al model convergeixin cap a 0 i a la vegada no es produeixi tanta variabilitat en les estimacions. En canvi, els mètodes de selecció de variables, com el *Stepwise,* no aconsegueixen obtenir uns millors resultats pel que respecte a la variabilitat dels intervals de confiança.

Per a corroborar que els mètodes de penalització provoquen la disminució en la variància de les estimacions, s'han calculat els intervals de confiança del mètode *Stepwise, Ridge* i *Elastic Net* amb un valor de $\alpha = 0'10$ (ja que són els que han obtingut una major exactitud) i efectivament, s'obtenen uns amples més petits en ells amb els mètodes de regularització que no pas amb el *Stepwise.*

La segona base de dades informa de certes característiques dels individus a estudiar per predir si presenten algun tipus de malaltia cardiovascular, la qual comparada amb l'exemple anterior, consta de pocs atributs (12 variables) i un nombre molt elevat d'observacions (70.000 individus). S'ha realitzat la descriptiva dels diversos atributs per a saber com es comportaven les dades, s'han eliminat totes aquelles observacions que duien a confusions a causa de la presència de valors molt estranys i, en aquest cas, no s'ha dut a terme cap imputació de dades mancants, ja que la base de dades no contenia *missings.* Un cop aplicat el mètode *Stepwise* i la regressió *Ridge, Lasso* i *Elastic Net,* s'ha arribat a la conclusió de que per aquest *dataset* la millor opció és utilitzar un mètode tradicional degut a que obté una exactitud més elevada que la resta de mètodes. A més, quan s'ha estudiat si les variables presentaven correlació entre elles mitjançant el càlcul dels VIF s'ha obtingut que no hi havia presència

de multicol·linealitat, per tant, es podria donar el cas de que no hi hagués una gran variabilitat en les estimacions dels coeficients i que no calgués aplicar un paràmetre de regularització. S'han calculat els intervals de confiança de les variables per el mètode escollit i s'obtenen unes variàncies petites.

Es podria concloure que avui en dia la selecció de variables té un paper molt rellevant en el món de l'estadística, ja que quan una base de dades presenta una nombre elevat de variables corre el risc de que es produeixi el problema de la multicol·linealitat entre elles i, per tant, hi hagi un augment de la variància en les estimacions dels coeficients dels atributs, fet que es pot intuir com que seria millor aplicar un mètode de penalització en comptes d'aplicar un dels mètodes tradicionals de selecció de variables. A més, si no es fa una selecció correcta dels atributs s'obtindrien males interpretacions en la recerca perquè podria ocorre que si s'afegeixen al model més variables de les necessàries hi hauria un millor ajust de les dades però augmentaria la variabilitat i en cas contrari, si es posessin menys variables, augmentaria el biaix.

7. BIBLIOGRAFIA

Faraway, Julian (2004). <<Linear Models with R. Second Edition>>.

Carmona, Francesc (2001). << Modelos Lineales>>. Disponible a:

http://www.ub.edu/stat/docencia/Diplomatura/ModelsLineals/regre.pdf

McCullagh P; Nelder J.A (1989). <<Generalized Linear Models. Second Edition>>. Dsiponible a:

http://www.utstat.toronto.edu/~brunner/oldclass/2201s11/readings/glmbook.pdf

Dobson, Annette (1945). <<An introduction to Generalized Linear Models. Second Edition.>> Disponible a:

https://reneues.files.wordpress.com/2010/01/an-introduction-to-generalized-linear-models-second-edition-dobson.pdf

Cortés, Jordi; Valero, Jordi (2019). <<Models Lineals Generalitzats>>. Disponible al Campus Virtual UB.

Rodríguez, Germán (2007). <<Lecture Notes on Generalized Linear Models>>. Disponible a: http://data.princeton.edu/wws509/notes/

James, Gareth; Witten, Daniela; Hastie, Trevor; Tibshirani, Robert (2013). << An Introduction to Statistical Learning>>. Disponible a:

https://faculty.marshall.usc.edu/gareth-james/ISL/ISLR%20Seventh%20Printing.pdf

Pereira, Jose Manuel; Basto, Mario; Ferreira, Amelia (2015). << The logistic Lasso and ridge regression in predicting corporate failure>>. Disponible a:

https://www.sciencedirect.com/science/article/pii/S2212567116303100

Tusell, F. (2011). <<Análisis de Regresión. Introducción Teórica y Práctica basada en R>>. Disponible a: http://www.et.bs.ehu.es/~etptupaf/nuevo/ficheros/estad3/nreg1.pdf

Fu, Wenjiang (1997). <<Penalized Regressions: The Bridge Versus the Lasso>>. Disponible a: http://www.hpc.unm.edu/~andriese/doc/ref1_fu.pdf

Tibshirani, Robert (1996). <<Regression shrinkage and selection via the Lasso>>. Disponible a: http://www.math.yorku.ca/~hkj/Teaching/6621Winter2013/Coverage/Lasso.pdf

Zou, Hui; Hastie, Trevor (2003). <<Regularization and variable selection via the elastic net>>. Disponible a:

https://web.stanford.edu/~hastie/Papers/B67.2%20(2005)%20301-320%20Zou%20&%20Hastie.pdf

Castro, Sebastián. Análisis de datos en grandes dimensiones. Estimación y selección de variables en regresión. Disponible a:

http://www.iesta.edu.uy/wp-content/uploads/2014/05/TJA_2011_Castro.pdf

Kassambara, Alboukadel (2018). <<Penalized Logistic Regression Essentials in R: Ridge, LASSO and Elastic Net>>. Disponible a:

http://www.sthda.com/english/articles/36-classification-methods-essentials/149-penalized-logistic-regression-essentials-in-r-ridge-Lasso-and-elastic-net/#computing-penalized-logistic-regression

Feuerriegel, Stefan (2015). << Regularization Methods>>. Disponible a:

https://www.is.uni-freiburg.de/resources/business-analytics/XX_Regularization.pdf

Gil, Cristina (2018). << Técnicas de regularización y selección del mejor modelo>>. Disponible a:

https://rpubs.com/Cristina_Gil/Regularizacion_Seleccion

Gil, Cristina (2018). << Métodos de remuestreo y validación de modelos: validación cruzada y bootstrap>>. Disponible a:

https://rpubs.com/Cristina_Gil/CV_Bootstrap

Zelada, Carlos (2018). << Evaluación de modelos de clasificación>>. Disponible a:

https://rpubs.com/chzelada/275494

Amat, Joaquín (2017). << Selección de predictores y mejor modelo lineal múltiple: subset selection, ridge regression, Lasso regression y dimension reduction>>. Disponible a:

https://rpubs.com/Joaquin_AR/242707

Friedman, Jerome; Hastie, Trevor; Tibshirani, Rob (2010). <<Regularization Paths for Generalized Linear Models via Coordinate Descent>>. Disponible a:

https://www.ncbi.nlm.nih.gov/pmc/articles/PMC2929880/

Burren, Stef (2011). << MICE: Multivariate Imputation by Chained Equations in R>>. Disponible a: http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.169.5745&rep=rep1&type=pdf

Efron, Bradley; Narasimhan, Balasubramanian (2018). << Automatic Construction of Bootstrap Confidence Intervals>>. Disponible a:

https://cran.r-project.org/web/packages/bcaboot/vignettes/bcaboot.html

Akpor, Isaac; Karim, Rezaul (2016). << An Application of Bootstrapping in Logistic Regression Model>>. Disponible a: https://www.scirp.org/html/70962_70962.htm

8. ANNEXOS

8.1. CARCINOMA HEPATOCEL·LULAR

```
#Llibreries

libraries <- function(){
  libr <- c("readxl","dplyr","ggplot2","GGally","Hmisc","corrplot","PerformanceAnalytics","knitr","kableExtra","FactoMineR","car","glmnet","pspearman","mice","ltm","tidyverse","caret","glmnet","lattice")
  invisible(lapply(libr, library, character.only=TRUE))
}
libraries()
```

LECTURA DE LES DADES

```
var.names <- c("Gender", "Sym", "Alc", "HepBSur", "HepBAnt", "HepBCore", "HepC", "Cir", "End", "Smo", "Dia", "Obe", "Hem", "Art", "CRen", "HIV", "Non", "EVar", "Spl", "PHyp", "Thr", "LMet", "Rad", "Agedia", "Alcpd", "cigpy", "Sta", "Encdeg", "Ascdeg", "IntNorRat", "Alp", "Hae", "MCorVol", "Leu", "Plat", "Alb", "Bil1", "Ala", "Aspa", "Gam", "Alk", "Prot", "Crea", "NNod", "dnod", "Bil2", "Iro", "Oxy", "Fer", "Class")

var.type <- c(rep("factor",23),rep("numeric",26),"factor")

ddcarc <- read.csv("carcinoma.csv", sep=";", header=TRUE, na.strings = "?", stringsAsFactors = FALSE, col.names = var.names, colClasses = var.type)

ddcarc$Gender <- factor(ddcarc$Gender, labels = c("woman","man"))
ddcarc$Class <- factor(ddcarc$Class, labels = c("died","survived"))
dim(ddcarc)
```

ANÀLISI DESCRIPTIVA DE LES DADES

```
summary(ddcarc)

ddcat <- ddcarc[,c(1:23,27:29,50)]
ddnum <- ddcarc[,-c(1:23,27:29,50)]
```

VARIABLES CATEGÒRIQUES

```
sapply(ddcat,table)
```

VARIABLES NUMÈRIQUES

```
library(dplyr)
columnas <- c("Mitjana","Variància","Desv. Típica","Covariància","Min","Q1","Mediana","Q3","Max")
M <- as.data.frame(matrix(ncol=length(columnas),nrow=ncol(ddnum),NA))
colnames(M) <- columnas

for(i in 1:ncol(ddnum)){

  nom <- names(ddnum[i])
  rownames(M)[i] <- nom
  res = summarise(
    ddnum,
```

```
   'mean' = round(mean(ddnum[,i], na.rm = TRUE),4),
   'var' = round(var(ddnum[,i], na.rm = TRUE), 4),
   'sd' = round(sd(ddnum[,i], na.rm = TRUE), 4),
   'cv' = round(sd/mean,4),
   'min' = round(min(ddnum[,i], na.rm = TRUE), 4),
   'Q1' = round(quantile(ddnum[,i], 0.25, names = F, na.rm = TRUE),4),
   'median' = round(median(ddnum[,i], na.rm = TRUE), 4),
   'Q3' = round(quantile(ddnum[,i], 0.75, names = F, na.rm = TRUE),4),
   'max' = round(max(ddnum[,i], na.rm = TRUE),4) )
 res
 M[i,] <- res

}
kable(M, caption = "Descrpitiva de les variables numèriques") %>%
  kable_styling(bootstrap_options = c("striped", "hover", "condensed", "responsive"), full_width = F,
position = "center") %>%
  column_spec(1:(ncol(M)+1), bold = T, border_left = T, border_right = T, color = "black")
```

OUTLIERS

```
for(i in 1:ncol(ddnum)){
nom <- names(ddnum)[i]
  boxplot(ddnum[,i],main=nom, col="lightcyan")
}
par(mfrow=c(3,2))
boxplot(ddnum$Alp,main="Alp", col="lightcyan")
boxplot(ddnum$Leu,main="Leu", col="lightcyan")
boxplot(ddnum$Plat,main="Plat", col="lightcyan")
boxplot(ddnum$Gam,main="Gam", col="lightcyan")
boxplot(ddnum$Alk,main="Alk", col="lightcyan")
boxplot(ddnum$Fer,main="Fer", col="lightcyan")
```

GRÀFICS

```
for( i in c(24:26,30:49)){

  plot(ddcarc[,i], ylab= var.names[i], main= var.names[i])

}
```

- Alcpd

```
plot(log(ddcarc$Alcpd))

table(ddcarc$Alcpd)

ddcarc$Alcpdf <- cut(ddcarc$Alcpd, breaks=c(-1,20,Inf), labels=c(0,1))
table(ddcarc$Alcpdf)
ddcarc <- ddcarc[,-51]
```

- cigpy

```
plot(log(ddcarc$cigpy))

table(ddcarc$cigpy)
```

```
ddcarc$cigpyf <- cut(ddcarc$cigpy, breaks=c(-1,0,Inf), labels=c(0,1))
table(ddcarc$cigpyf)
ddcarc <- ddcarc[,-51]
```

- Leu

```
plot(log(ddcarc$Leu))

table(ddcarc$Leu)

ddcarc$Leu <- cut(ddcarc$Leu, breaks=c(1,3.8,11,Inf), labels=c(1,0,1))
table(ddcarc$Leu)
```

- Plat

```
plot(log(ddcarc$Plat))

table(ddcarc$Plat)

ddcarc$Plat <- cut(ddcarc$Plat, breaks=c(1,142,450, Inf), labels=c(1,0,1))
table(ddcarc$Plat)
```

- Bil1

```
plot(log(ddcarc$Bil1))

ddcarc$Bil1 <- log(ddcarc$Bil1)
```

- Ala

```
plot(log(ddcarc$Ala))

ddcarc$Ala <- log(ddcarc$Ala)
```

- Aspa

```
plot(log(ddcarc$Aspa))

ddcarc$Aspa <- log(ddcarc$Aspa)
```

- Gam

```
plot(log(ddcarc$Gam))

ddcarc$Gam <- log(ddcarc$Gam)
```

- Alk

```
plot(ddcarc$Alk)
summary(ddcarc$Alk)
which.min(ddcarc$Alk)
#Error
ddcarc$Alk[73]
ddcarc$Alk[73] <- 128

plot(log(ddcarc$Alk))
ddcarc$Alk <- log(ddcarc$Alk)
summary(ddcarc$Alk)
```

- Prot

```
plot(log(ddcarc$Prot))
```

```
ddcarc$Prot <- log(ddcarc$Prot)
```

- Crea

```
plot(log(ddcarc$Crea))

ddcarc$Crea <- log(ddcarc$Crea)
```

- dnod

```
plot(log(ddcarc$dnod))

ddcarc$dnod <- log(ddcarc$dnod)
```

- Bil2

```
plot(log(ddcarc$Bil2))

ddcarc$Bil2 <- log(ddcarc$Bil2)
```

- Fer

```
summary(ddcarc$Fer)
which(ddcarc$Fer==0)
ddcarc$Fer[65] <- NA
```

Descriptiva dels NA

```
which(names(ddcarc)=="Class")
missmap(ddcarc, col=c('grey', 'steelblue'), y.cex=0.5, x.cex=0.8)

sort(sapply(ddcarc[,-50], function(x){sum(is.na(x))}), decreasing=TRUE)

round(100*sort(sapply(ddcarc[,-39], function(x){sum(is.na(x))}), decreasing=TRUE)/165, 1)

exclude <- c("Fer","Oxy","Iro")
include <- setdiff(names(ddcarc),exclude)
ddcarc.new <- ddcarc[include]

which(names(ddcarc.new)=="Class")
num.na <- apply(ddcarc.new[,-47], 1, function(x){sum(is.na(x))})
sort(num.na, decreasing=TRUE)
which(num.na > 0.4*47) # más del 40% de NAs
ddcarc.new <- ddcarc.new[-which(num.na > 0.4*47),]
```

IMPUTACIÓ DADES MANCANTS

```
#Mètodes per defecte
ddcarc.imp <- ddcarc.new
miced <- mice(ddcarc.imp, m=1, seed=123)
ddcarc.imp<- complete(miced)
print(miced)
#CART
ddcarc.cart <- ddcarc.new
micec <- mice(ddcarc.cart, m=1, seed=123, method = "cart", printFlag=FALSE)
ddcarc.cart<- complete(micec)
print(micec)

sum(is.na(ddcarc.imp))
sum(is.na(ddcarc.cart))
```

DIAGNOSIS

```
stripplot(miced, pch=20, cex=1.2)
densityplot(miced, layout = c(3, 3))

stripplot(micec, pch=20, cex=1.2)
densityplot(micec, layout = c(3, 3))

data.frame(miced$imp$Sym,micec$imp$Sym)
table(ddcarc$Sym)
table(miced$imp$Sym)
table(micec$imp$Sym)

data.frame(miced$imp$Alcpd,micec$imp$Alcpd)
table(ddcarc$Alcpd)
table(miced$imp$Alcpd)
table(micec$imp$Alcpd)

data.frame(miced$imp$Smo,micec$imp$Smo)
table(ddcarc$Smo)
table(miced$imp$Smo)
table(micec$imp$Smo)

data.frame(miced$imp$EVar,micec$imp$EVar)
data.frame(miced$imp$Leu,micec$imp$Leu)
data.frame(miced$imp$Plat,micec$imp$Plat)
```

MODEL COMPLET

```
full_model <- glm(Class ~ Gender + Sym + Alc + HepBSur + HepBAnt + HepBCore + HepC + Cir + End +
Smo + Dia + Obe + Hem + Art + CRen + HIV + Non + EVar + Spl + PHyp + Thr + LMet + Rad + Agedia + A
lcpd + cigpy +  Sta + Encdeg + Ascdeg + IntNorRat + Alp + Hae + MCorVol + Leu + Plat + Alb + Bil1 + Ala
+ Aspa + Gam + Alk + Prot + Crea + NNod + dnod + Bil2, family="binomial", data=ddcarc.imp)

summary(full_model)
```

- CORRELACIÓ

```
round(vif(full_model),2)
```

MÈTODES TRADICIONALS

```
set.seed(10)
train <- ddcarc.imp$Class %>% createDataPartition(p = 0.7, list = FALSE)
train.data  <- ddcarc.imp[train, ]
test.data <- ddcarc.imp[-train, ]

full_model2 <- glm(Class ~ Gender + Sym + Alc + HepBSur + HepBAnt + HepBCore + HepC + Cir + End
+ Smo + Dia + Obe + Hem + Art + CRen + HIV + Non + EVar + Spl + PHyp + Thr + LMet + Rad + Agedia +
Alcpd + cigpy +  Sta + Encdeg + Ascdeg + IntNorRat + Alp + Hae + MCorVol + Leu + Plat + Alb + Bil1 + Al
a + Aspa + Gam + Alk + Prot + Crea + NNod + dnod + Bil2, family="binomial", data=train.data)

null_model <- glm(Class ~ 1, family="binomial", data=train.data)

step(null_model, scope = list(lower = null_model, upper = full_model2), direction=c("both"), trace=0)
```

```
modboth1 <- glm(Class ~ Alk + IntNorRat + Alp + Sym + CRen + cigpy + Art + Smo + Agedia + HepBSur
+ Aspa + Ala + HepC + Thr, family = "binomial", data = train.data)
summary(modboth1)
AIC(modboth1)
```

Modboth1

```
prediccio <- predict(modboth1,test.data,type = "response")
predresp <- ifelse(prediccio>0.5, "survived","died")
perfdat<-data.frame(obs=test.data$Class,pred= predresp)

pos <- sum(perfdat$obs=="survived")
neg <- sum(perfdat$obs=="died")
predpos <- sum(perfdat$pred=="survived")
predneg <- sum(perfdat$pred=="died")
total <- nrow(perfdat)
data.frame(pos, neg,predpos,predneg)

tp<-sum(perfdat$obs=="survived" & perfdat$pred=="survived")
tn<-sum(perfdat$obs=="died" & perfdat$pred=="died")
fp<-sum(perfdat$obs=="died" & perfdat$pred=="survived")
fn<-sum(perfdat$obs=="survived" & perfdat$pred=="died")
data.frame(tp,tn,fp,fn)

exac <- (tp+tn)/total
error <- (fp+fn)/total
sens <- tp/pos
espec <- tn/neg
prec <- tp/predpos
npv <- tn / predneg
data.frame(exac,error,sens,espec,prec,npv)
```

- Corba ROC

```
predROC = prediction(prediccio, test.data$Class)
ROC = performance(predROC, "tpr", "fpr")
plot(ROC, colorize = TRUE, print.cutoffs.at=seq(0,1,0.1),text.adj=c(-0.2,1.7))

auc = as.numeric(performance(predROC,"auc")@y.values)
auc
```

MÈTODES DE PENALITZACIÓ RIDGE REGRESSION

```
x <- model.matrix(Class~., data = train.data)[,-1]
y <- train.data$Class

set.seed(10)
ridge <- glmnet(x, y, family = "binomial", alpha = 0)

plot(ridge, xvar = "lambda", label = TRUE)
```

DETERMINAR EL VALOR DE λ

```
set.seed(10)
cv_error_ridge <- cv.glmnet(x = x, y = y, alpha = 0, nfolds = 10, family="binomial")
plot(cv_error_ridge)
```

```
cv_error_ridge$lambda.min

cv_error_ridge$lambda.1se
```

MODEL FINAL RIDGE REGRESSION

```
model_final_ridge_lmin <- glmnet(x = x, y = y, alpha = 0, lambda = cv_error_ridge$lambda.min, family="binomial")

model_final_ridge_lse <- glmnet(x = x, y = y, alpha = 0, lambda = cv_error_ridge$lambda.1se, family="binomial")
```

COEFICIENTS

```
ridge.coef <- predict(model_final_ridge_lmin,type="coefficients",s=cv_error_ridge$lambda.min)
sum(ridge.coef!=0)
sum(ridge.coef==0)

ridge.coef2=predict(model_final_ridge_lse,type="coefficients",s=cv_error_ridge$lambda.1se)
sum(ridge.coef2!=0)
sum(ridge.coef2==0)
```

PREDICCIONS

- Lambda mínima:

```
x.test <- model.matrix(Class ~., test.data)[,-1]
prob <- model_final_ridge_lmin %>% predict(newx = x.test)
predresp <- ifelse(prob > 0.5,"survived", "died")
perfdat<-data.frame(obs=test.data$Class, pred= predresp)
colnames(perfdat) <- c("obs","pred")

pos <- sum(perfdat$obs=="survived")
neg <- sum(perfdat$obs=="died")
predpos <- sum(perfdat$pred=="survived")
predneg <- sum(perfdat$pred=="died")
total <- nrow(perfdat)
data.frame(pos, neg,predpos,predneg)

tp<-sum(perfdat$obs=="survived" & perfdat$pred=="survived")
tn<-sum(perfdat$obs=="died" & perfdat$pred=="died")
fp<-sum(perfdat$obs=="died" & perfdat$pred=="survived")
fn<-sum(perfdat$obs=="survived" & perfdat$pred=="died")
data.frame(tp,tn,fp,fn)

exac <- (tp+tn)/total
error <- (fp+fn)/total
sens <- tp/pos
espec <- tn/neg
prec <- tp/predpos
npv <- tn / predneg
data.frame(exac,error,sens,espec,prec,npv)
```

- Corba ROC

```
predROC = prediction(prob, test.data$Class)
ROC = performance(predROC, "tpr", "fpr")
plot(ROC, colorize = TRUE, print.cutoffs.at=seq(0,1,0.1),text.adj=c(-0.2,1.7))

auc = as.numeric(performance(predROC,"auc")@y.values)
auc
```

- Lambda òptima:

```
prob <- model_final_ridge_lse %>% predict(newx = x.test)
predresp <- ifelse(prob > 0.5,"survived", "died")
perfdat<-data.frame(obs=test.data$Class,pred= predresp)
colnames(perfdat) <- c("obs","pred")

pos <- sum(perfdat$obs=="survived")
neg <- sum(perfdat$obs=="died")
predpos <- sum(perfdat$pred=="survived")
predneg <- sum(perfdat$pred=="died")
total <- nrow(perfdat)
data.frame(pos, neg,predpos,predneg)

tp<-sum(perfdat$obs=="survived" & perfdat$pred=="survived")
tn<-sum(perfdat$obs=="died" & perfdat$pred=="died")
fp<-sum(perfdat$obs=="died" & perfdat$pred=="survived")
fn<-sum(perfdat$obs=="survived" & perfdat$pred=="died")
data.frame(tp,tn,fp,fn)

exac <- (tp+tn)/total
error <- (fp+fn)/total
sens <- tp/pos
espec <- tn/neg
prec <- tp/predpos
npv <- tn / predneg
data.frame(exac,error,sens,espec,prec,npv)
```

- Corba ROC

```
predROC = prediction(prob, test.data$Class)
ROC = performance(predROC, "tpr", "fpr")
plot(ROC, colorize = TRUE, print.cutoffs.at=seq(0,1,0.1),text.adj=c(-0.2,1.7))

auc = as.numeric(performance(predROC,"auc")@y.values)
auc
```

LASSO REGRESSION

```
set.seed(10)
lasso <- glmnet(x, y, family = "binomial", alpha = 1)

plot(lasso, xvar = "lambda", label = TRUE)
```

DETERMINAR EL VALOR DE λ

```
set.seed(10)
cv_error_lasso <- cv.glmnet(x = x, y = y, alpha = 1, nfolds = 10,family="binomial")
plot(cv_error_lasso)
```

```
cv_error_lasso$lambda.min

cv_error_lasso$lambda.1se
```

MODEL FINAL LASSO REGRESSION

```
model_final_lasso_lmin <- glmnet(x = x, y = y, alpha = 1, lambda = cv_error_lasso$lambda.min, family ="binomial")

model_final_lasso_lse <- glmnet(x = x, y = y, alpha = 1, lambda = cv_error_lasso$lambda.1se, family= "binomial")
```

COEFICIENTS

```
lasso.coef=predict(model_final_lasso_lmin,type="coefficients",s=cv_error_lasso$lambda.min)
sum(lasso.coef!=0)
sum(lasso.coef==0)

lasso.coef2=predict(model_final_lasso_lse,type="coefficients",s=cv_error_lasso$lambda.1se)
sum(lasso.coef2!=0)
sum(lasso.coef2==0)
```

PREDICCIONS

- Lambda mínima:

```
prob <- model_final_lasso_lmin %>% predict(newx = x.test)
predresp <- ifelse(prob > 0.5,"survived", "died")
perfdat<-data.frame(obs=test.data$Class, pred= predresp)
colnames(perfdat) <- c("obs","pred")

pos <- sum(perfdat$obs=="survived")
neg <- sum(perfdat$obs=="died")
predpos <- sum(perfdat$pred=="survived")
predneg <- sum(perfdat$pred=="died")
total <- nrow(perfdat)
data.frame(pos, neg,predpos,predneg)

tp<-sum(perfdat$obs=="survived" & perfdat$pred=="survived")
tn<-sum(perfdat$obs=="died" & perfdat$pred=="died")
fp<-sum(perfdat$obs=="died" & perfdat$pred=="survived")
fn<-sum(perfdat$obs=="survived" & perfdat$pred=="died")
data.frame(tp,tn,fp,fn)

exac <- (tp+tn)/total
error <- (fp+fn)/total
sens <- tp/pos
espec <- tn/neg
prec <- tp/predpos
npv <- tn / predneg
data.frame(exac,error,sens,espec,prec,npv)
```

- Corba ROC

```
predROC = prediction(prob, test.data$Class)
ROC = performance(predROC, "tpr", "fpr")
plot(ROC, colorize = TRUE, print.cutoffs.at=seq(0,1,0.1),text.adj=c(-0.2,1.7))
```

```
auc = as.numeric(performance(predROC,"auc")@y.values)
auc
```

- Lambda òptima:

```
prob <- model_final_lasso_lse %>% predict(newx = x.test)
predresp <- ifelse(prob > 0.5,"survived", "died")
perfdat<-data.frame(obs=test.data$Class, pred= predresp)
colnames(perfdat) <- c("obs","pred")

pos <- sum(perfdat$obs=="survived")
neg <- sum(perfdat$obs=="died")
predpos <- sum(perfdat$pred=="survived")
predneg <- sum(perfdat$pred=="died")
total <- nrow(perfdat)
data.frame(pos, neg,predpos,predneg)

tp<-sum(perfdat$obs=="survived" & perfdat$pred=="survived")
tn<-sum(perfdat$obs=="died" & perfdat$pred=="died")
fp<-sum(perfdat$obs=="died" & perfdat$pred=="survived")
fn<-sum(perfdat$obs=="survived" & perfdat$pred=="died")
data.frame(tp,tn,fp,fn)

exac <- (tp+tn)/total
error <- (fp+fn)/total
sens <- tp/pos
espec <- tn/neg
prec <- tp/predpos
npv <- tn / predneg
data.frame(exac,error,sens,espec,prec,npv)
```

- Corba ROC

```
predROC = prediction(prob, test.data$Class)
ROC = performance(predROC, "tpr", "fpr")
plot(ROC, colorize = TRUE, print.cutoffs.at=seq(0,1,0.1),text.adj=c(-0.2,1.7))

auc = as.numeric(performance(predROC,"auc")@y.values)
auc
```

ELASTIC NET REGRESSION

AMB ALPHA=0.1

```
set.seed(10)
elasticnet1 <- glmnet(x, y, family = "binomial", alpha = 0.1)

plot(elasticnet1, xvar = "lambda", label = TRUE)
```

DETERMINAR EL VALOR DE λ

```
set.seed(10)
cv_error_elastic1 <- cv.glmnet(x = x, y = y, alpha = 0.1, nfolds = 10, family="binomial")
plot(cv_error_elastic1)

cv_error_elastic1$lambda.min
```

```
cv_error_elastic1$lambda.1se
```

MODEL FINAL ELASTIC NET REGRESSION

```
model_final_elastic_lmin1 <- glmnet(x = x, y = y, alpha = 0.1, lambda = cv_error_elastic1$lambda.min
, family="binomial")

model_final_elastic_lse1 <- glmnet(x = x, y = y, alpha = 0.1, lambda = cv_error_elastic1$lambda.1se, f
amily="binomial")
```

COEFICIENTS

```
elastic.coef1=predict(model_final_elastic_lmin1,type="coefficients",s=cv_error_elastic1$lambda.min
)
sum(elastic.coef1!=0)
sum(elastic.coef1==0)

elastic.coef11=predict(model_final_elastic_lse1,type="coefficients",s=cv_error_elastic1$lambda.1se)
sum(elastic.coef11!=0)
sum(elastic.coef11==0)
```

PREDICCIONS

- Lambda mínima:

```
prob <- model_final_elastic_lmin1 %>% predict(newx = x.test)
predresp <- ifelse(prob > 0.5,"survived", "died")
perfdat<-data.frame(obs=test.data$Class, pred= predresp)
colnames(perfdat) <- c("obs","pred")

pos <- sum(perfdat$obs=="survived")
neg <- sum(perfdat$obs=="died")
predpos <- sum(perfdat$pred=="survived")
predneg <- sum(perfdat$pred=="died")
total <- nrow(perfdat)
data.frame(pos, neg,predpos,predneg)

tp<-sum(perfdat$obs=="survived" & perfdat$pred=="survived")
tn<-sum(perfdat$obs=="died" & perfdat$pred=="died")
fp<-sum(perfdat$obs=="died" & perfdat$pred=="survived")
fn<-sum(perfdat$obs=="survived" & perfdat$pred=="died")
data.frame(tp,tn,fp,fn)

exac <- (tp+tn)/total
error <- (fp+fn)/total
sens <- tp/pos
espec <- tn/neg
prec <- tp/predpos
npv <- tn / predneg
data.frame(exac,error,sens,espec,prec,npv)
```

- Corba ROC

```
predROC = prediction(prob, test.data$Class)
ROC = performance(predROC, "tpr", "fpr")
plot(ROC, colorize = TRUE, print.cutoffs.at=seq(0,1,0.1),text.adj=c(-0.2,1.7))
```

```
auc = as.numeric(performance(predROC,"auc")@y.values)
auc
```

- Lambda òptima:

```
prob <- model_final_elastic_lse1 %>% predict(newx = x.test)
predresp <- ifelse(prob > 0.5,"survived", "died")
perfdat<-data.frame(obs=test.data$Class, pred= predresp)
colnames(perfdat) <- c("obs","pred")

pos <- sum(perfdat$obs=="survived")
neg <- sum(perfdat$obs=="died")
predpos <- sum(perfdat$pred=="survived")
predneg <- sum(perfdat$pred=="died")
total <- nrow(perfdat)
data.frame(pos, neg,predpos,predneg)

tp<-sum(perfdat$obs=="survived" & perfdat$pred=="survived")
tn<-sum(perfdat$obs=="died" & perfdat$pred=="died")
fp<-sum(perfdat$obs=="died" & perfdat$pred=="survived")
fn<-sum(perfdat$obs=="survived" & perfdat$pred=="died")
data.frame(tp,tn,fp,fn)

exac <- (tp+tn)/total
error <- (fp+fn)/total
sens <- tp/pos
espec <- tn/neg
prec <- tp/predpos
npv <- tn / predneg
data.frame(exac,error,sens,espec,prec,npv)
```

- Corba ROC

```
predROC = prediction(prob, test.data$Class)
ROC = performance(predROC, "tpr", "fpr")
plot(ROC, colorize = TRUE, print.cutoffs.at=seq(0,1,0.1),text.adj=c(-0.2,1.7))

auc = as.numeric(performance(predROC,"auc")@y.values)
auc
```

AMB ALPHA=0.25

```
set.seed(10)
elasticnet2 <- glmnet(x, y, family = "binomial", alpha = 0.25)

plot(elasticnet2, xvar = "lambda", label = TRUE)
```

DETERMINAR EL VALOR DE λ

```
set.seed(10)
cv_error_elastic2 <- cv.glmnet(x = x, y = y, alpha = 0.25, nfolds = 10, family="binomial")
plot(cv_error_elastic2)

cv_error_elastic2$lambda.min

cv_error_elastic2$lambda.1se
```

MODEL FINAL ELASTIC NET REGRESSION

```
model_final_elastic_lmin2 <- glmnet(x = x, y = y, alpha = 0.25, lambda = cv_error_elastic2$lambda.min, family="binomial")

model_final_elastic_lse2 <- glmnet(x = x, y = y, alpha = 0.25, lambda = cv_error_elastic2$lambda.1se, family="binomial")
```

COEFICIENTS

```
elastic.coef2=predict(model_final_elastic_lmin2,type="coefficients",s=cv_error_elastic2$lambda.min)
sum(elastic.coef2!=0)
sum(elastic.coef2==0)

elastic.coef22=predict(model_final_elastic_lse2,type="coefficients",s=cv_error_elastic2$lambda.1se)
sum(elastic.coef22!=0)
sum(elastic.coef22==0)
```

PREDICCIONS

- Lambda mínima:

```
prob <- model_final_elastic_lmin2 %>% predict(newx = x.test)
predresp <- ifelse(prob > 0.5,"survived", "died")
perfdat<-data.frame(obs=test.data$Class, pred= predresp)
colnames(perfdat) <- c("obs","pred")

pos <- sum(perfdat$obs=="survived")
neg <- sum(perfdat$obs=="died")
predpos <- sum(perfdat$pred=="survived")
predneg <- sum(perfdat$pred=="died")
total <- nrow(perfdat)
data.frame(pos, neg,predpos,predneg)

tp<-sum(perfdat$obs=="survived" & perfdat$pred=="survived")
tn<-sum(perfdat$obs=="died" & perfdat$pred=="died")
fp<-sum(perfdat$obs=="died" & perfdat$pred=="survived")
fn<-sum(perfdat$obs=="survived" & perfdat$pred=="died")
data.frame(tp,tn,fp,fn)

exac <- (tp+tn)/total
error <- (fp+fn)/total
sens <- tp/pos
espec <- tn/neg
prec <- tp/predpos
npv <- tn / predneg
data.frame(exac,error,sens,espec,prec,npv)
```

- Corba ROC

```
predROC = prediction(prob, test.data$Class)
ROC = performance(predROC, "tpr", "fpr")
plot(ROC, colorize = TRUE, print.cutoffs.at=seq(0,1,0.1),text.adj=c(-0.2,1.7))
```

```
auc = as.numeric(performance(predROC,"auc")@y.values)
auc
```

- Lambda òptima:

```
prob <- model_final_elastic_lse2 %>% predict(newx = x.test)
predresp <- ifelse(prob > 0.5,"survived", "died")
perfdat<-data.frame(obs=test.data$Class, pred= predresp)
colnames(perfdat) <- c("obs","pred")

pos <- sum(perfdat$obs=="survived")
neg <- sum(perfdat$obs=="died")
predpos <- sum(perfdat$pred=="survived")
predneg <- sum(perfdat$pred=="died")
total <- nrow(perfdat)
data.frame(pos, neg,predpos,predneg)

tp<-sum(perfdat$obs=="survived" & perfdat$pred=="survived")
tn<-sum(perfdat$obs=="died" & perfdat$pred=="died")
fp<-sum(perfdat$obs=="died" & perfdat$pred=="survived")
fn<-sum(perfdat$obs=="survived" & perfdat$pred=="died")
data.frame(tp,tn,fp,fn)

exac <- (tp+tn)/total
error <- (fp+fn)/total
sens <- tp/pos
espec <- tn/neg
prec <- tp/predpos
npv <- tn / predneg
data.frame(exac,error,sens,espec,prec,npv)
```

- Corba ROC

```
predROC = prediction(prob, test.data$Class)
ROC = performance(predROC, "tpr", "fpr")
plot(ROC, colorize = TRUE, print.cutoffs.at=seq(0,1,0.1),text.adj=c(-0.2,1.7))

auc = as.numeric(performance(predROC,"auc")@y.values)
auc
```

AMB ALPHA= 0.5

```
set.seed(10)
elasticnet3 <- glmnet(x, y, family = "binomial", alpha = 0.5)

plot(elasticnet3, xvar = "lambda", label = TRUE)
```

DETERMINAR EL VALOR DE λ

```
set.seed(10)
cv_error_elastic3 <- cv.glmnet(x = x, y = y, alpha = 0.5, nfolds = 10, family="binomial")
plot(cv_error_elastic3)

cv_error_elastic3$lambda.min

cv_error_elastic3$lambda.1se
```

MODEL FINAL ELASTIC NET REGRESSION

```
model_final_elastic_lmin3 <- glmnet(x = x, y = y, alpha = 0.5, lambda = cv_error_elastic3$lambda.min
, family="binomial")

model_final_elastic_lse3 <- glmnet(x = x, y = y, alpha = 0.5, lambda = cv_error_elastic3$lambda.1se, f
amily="binomial")
```

COEFICIENTS

```
elastic.coef3=predict(model_final_elastic_lmin3,type="coefficients",s=cv_error_elastic3$lambda.min
)
sum(elastic.coef3!=0)
sum(elastic.coef3==0)

elastic.coef33=predict(model_final_elastic_lse3,type="coefficients",s=cv_error_elastic3$lambda.1se)
sum(elastic.coef33!=0)
sum(elastic.coef33==0)
```

- Lambda mínima:

```
prob <- model_final_elastic_lmin3 %>% predict(newx = x.test)
predresp <- ifelse(prob > 0.5,"survived", "died")
perfdat<-data.frame(obs=test.data$Class, pred= predresp)
colnames(perfdat) <- c("obs","pred")

pos <- sum(perfdat$obs=="survived")
neg <- sum(perfdat$obs=="died")
predpos <- sum(perfdat$pred=="survived")
predneg <- sum(perfdat$pred=="died")
total <- nrow(perfdat)
data.frame(pos, neg,predpos,predneg)

tp<-sum(perfdat$obs=="survived" & perfdat$pred=="survived")
tn<-sum(perfdat$obs=="died" & perfdat$pred=="died")
fp<-sum(perfdat$obs=="died" & perfdat$pred=="survived")
fn<-sum(perfdat$obs=="survived" & perfdat$pred=="died")
data.frame(tp,tn,fp,fn)

exac <- (tp+tn)/total
error <- (fp+fn)/total
sens <- tp/pos
espec <- tn/neg
prec <- tp/predpos
npv <- tn / predneg
data.frame(exac,error,sens,espec,prec,npv)
```

- Corba ROC

```
predROC = prediction(prob, test.data$Class)
ROC = performance(predROC, "tpr", "fpr")
plot(ROC, colorize = TRUE, print.cutoffs.at=seq(0,1,0.1),text.adj=c(-0.2,1.7))

auc = as.numeric(performance(predROC,"auc")@y.values)
auc
```

- Lambda òptima:

```
prob1 <- model_final_elastic_lse3 %>% predict(newx = x.test)
predresp <- ifelse(prob1 > 0.5,"survived", "died")
perfdat<-data.frame(obs=test.data$Class, pred= predresp)
colnames(perfdat) <- c("obs","pred")

pos <- sum(perfdat$obs=="survived")
neg <- sum(perfdat$obs=="died")
predpos <- sum(perfdat$pred=="survived")
predneg <- sum(perfdat$pred=="died")
total <- nrow(perfdat)
data.frame(pos, neg,predpos,predneg)

tp<-sum(perfdat$obs=="survived" & perfdat$pred=="survived")
tn<-sum(perfdat$obs=="died" & perfdat$pred=="died")
fp<-sum(perfdat$obs=="died" & perfdat$pred=="survived")
fn<-sum(perfdat$obs=="survived" & perfdat$pred=="died")
data.frame(tp,tn,fp,fn)

exac <- (tp+tn)/total
error <- (fp+fn)/total
sens <- tp/pos
espec <- tn/neg
prec <- tp/predpos
npv <- tn / predneg
data.frame(exac,error,sens,espec,prec,npv)
```

- Corba ROC

```
predROC1 = prediction(prob1, test.data$Class)
ROC1 = performance(predROC1, "tpr", "fpr")
plot(ROC1, colorize = TRUE, print.cutoffs.at=seq(0,1,0.1),text.adj=c(-0.2,1.7))

auc = as.numeric(performance(predROC1,"auc")@y.values)
auc
```

AMB ALPHA= 0.75

```
set.seed(10)
elasticnet4 <- glmnet(x, y, family = "binomial", alpha = 0.75)

plot(elasticnet4, xvar = "lambda", label = TRUE)
```

DETERMINAR EL VALOR DE λ

```
set.seed(10)
cv_error_elastic4 <- cv.glmnet(x = x, y = y, alpha = 0.75, nfolds = 10, family="binomial")
plot(cv_error_elastic4)

cv_error_elastic4$lambda.min

cv_error_elastic4$lambda.1se
```

MODEL FINAL ELASTIC NET REGRESSION

```
model_final_elastic_lmin4 <- glmnet(x = x, y = y, alpha = 0.75, lambda = cv_error_elastic4$lambda.min, family="binomial")

model_final_elastic_lse4 <- glmnet(x = x, y = y, alpha = 0.75, lambda = cv_error_elastic4$lambda.1se, family="binomial")
```

COEFICIENTS

```
elastic.coef4=predict(model_final_elastic_lmin4,type="coefficients",s=cv_error_elastic4$lambda.min)
sum(elastic.coef4!=0)
sum(elastic.coef4==0)

elastic.coef44=predict(model_final_elastic_lse4,type="coefficients",s=cv_error_elastic4$lambda.1se)
sum(elastic.coef44!=0)
sum(elastic.coef44==0)
```

PREDICCIONS

- Lambda mínima:

```
prob <- model_final_elastic_lmin4 %>% predict(newx = x.test)
predresp <- ifelse(prob > 0.5,"survived", "died")
perfdat<-data.frame(obs=test.data$Class, pred= predresp)
colnames(perfdat) <- c("obs","pred")

pos <- sum(perfdat$obs=="survived")
neg <- sum(perfdat$obs=="died")
predpos <- sum(perfdat$pred=="survived")
predneg <- sum(perfdat$pred=="died")
total <- nrow(perfdat)
data.frame(pos, neg,predpos,predneg)

tp<-sum(perfdat$obs=="survived" & perfdat$pred=="survived")
tn<-sum(perfdat$obs=="died" & perfdat$pred=="died")
fp<-sum(perfdat$obs=="died" & perfdat$pred=="survived")
fn<-sum(perfdat$obs=="survived" & perfdat$pred=="died")
data.frame(tp,tn,fp,fn)

exac <- (tp+tn)/total
error <- (fp+fn)/total
sens <- tp/pos
espec <- tn/neg
prec <- tp/predpos
npv <- tn / predneg
data.frame(exac,error,sens,espec,prec,npv)
```

- Corba ROC

```
predROC = prediction(prob, test.data$Class)
ROC = performance(predROC, "tpr", "fpr")
plot(ROC, colorize = TRUE, print.cutoffs.at=seq(0,1,0.1),text.adj=c(-0.2,1.7))

auc = as.numeric(performance(predROC,"auc")@y.values)
auc
```

- Lambda òptima:

```
prob <- model_final_elastic_lse4 %>% predict(newx = x.test)
predresp <- ifelse(prob > 0.5,"survived", "died")
perfdat<-data.frame(obs=test.data$Class, pred= predresp)
colnames(perfdat) <- c("obs","pred")

pos <- sum(perfdat$obs=="survived")
neg <- sum(perfdat$obs=="died")
predpos <- sum(perfdat$pred=="survived")
predneg <- sum(perfdat$pred=="died")
total <- nrow(perfdat)
data.frame(pos, neg,predpos,predneg)

tp<-sum(perfdat$obs=="survived" & perfdat$pred=="survived")
tn<-sum(perfdat$obs=="died" & perfdat$pred=="died")
fp<-sum(perfdat$obs=="died" & perfdat$pred=="survived")
fn<-sum(perfdat$obs=="survived" & perfdat$pred=="died")
data.frame(tp,tn,fp,fn)

exac <- (tp+tn)/total
error <- (fp+fn)/total
sens <- tp/pos
espec <- tn/neg
prec <- tp/predpos
npv <- tn / predneg
data.frame(exac,error,sens,espec,prec,npv)
```

- Corba ROC

```
predROC = prediction(prob, test.data$Class)
ROC = performance(predROC, "tpr", "fpr")
plot(ROC, colorize = TRUE, print.cutoffs.at=seq(0,1,0.1),text.adj=c(-0.2,1.7))

auc = as.numeric(performance(predROC,"auc")@y.values)
auc
```

AMB ALPHA= 0.95

```
set.seed(10)
elasticnet5 <- glmnet(x, y, family = "binomial", alpha = 0.95)

plot(elasticnet5, xvar = "lambda", label = TRUE)
```

DETERMINAR EL VALOR DE λ

```
set.seed(10)
cv_error_elastic5 <- cv.glmnet(x = x, y = y, alpha = 0.95, nfolds = 10, family="binomial")
plot(cv_error_elastic5)

cv_error_elastic5$lambda.min

cv_error_elastic5$lambda.1se
```

MODEL FINAL ELASTIC NET REGRESSION

```
model_final_elastic_lmin5 <- glmnet(x = x, y = y, alpha = 0.95, lambda = cv_error_elastic5$lambda.min, family="binomial")

model_final_elastic_lse5 <- glmnet(x = x, y = y, alpha = 0.95, lambda = cv_error_elastic5$lambda.1se, family="binomial")
```

COEFICIENTS

```
elastic.coef5=predict(model_final_elastic_lmin5,type="coefficients",s=cv_error_elastic5$lambda.min)
sum(elastic.coef5!=0)
sum(elastic.coef5==0)

elastic.coef55=predict(model_final_elastic_lse5,type="coefficients",s=cv_error_elastic5$lambda.1se)
sum(elastic.coef55!=0)
sum(elastic.coef55==0)
```

PREDICCIONS

- Lambda mínima:

```
prob <- model_final_elastic_lmin5 %>% predict(newx = x.test)
predresp <- ifelse(prob > 0.5,"survived", "died")
perfdat<-data.frame(obs=test.data$Class, pred= predresp)
colnames(perfdat) <- c("obs","pred")

pos <- sum(perfdat$obs=="survived")
neg <- sum(perfdat$obs=="died")
predpos <- sum(perfdat$pred=="survived")
predneg <- sum(perfdat$pred=="died")
total <- nrow(perfdat)
data.frame(pos, neg,predpos,predneg)

tp<-sum(perfdat$obs=="survived" & perfdat$pred=="survived")
tn<-sum(perfdat$obs=="died" & perfdat$pred=="died")
fp<-sum(perfdat$obs=="died" & perfdat$pred=="survived")
fn<-sum(perfdat$obs=="survived" & perfdat$pred=="died")
data.frame(tp,tn,fp,fn)

exac <- (tp+tn)/total
error <- (fp+fn)/total
sens <- tp/pos
espec <- tn/neg
prec <- tp/predpos
npv <- tn / predneg
data.frame(exac,error,sens,espec,prec,npv)
```

- Corba ROC

```
predROC = prediction(prob, test.data$Class)
ROC = performance(predROC, "tpr", "fpr")
plot(ROC, colorize = TRUE, print.cutoffs.at=seq(0,1,0.1),text.adj=c(-0.2,1.7))

auc = as.numeric(performance(predROC,"auc")@y.values)
auc
```

- Lambda òptima:

```
prob <- model_final_elastic_lse5 %>% predict(newx = x.test)
predresp <- ifelse(prob > 0.5,"survived", "died")
perfdat<-data.frame(obs=test.data$Class, pred= predresp)
colnames(perfdat) <- c("obs","pred")

pos <- sum(perfdat$obs=="survived")
neg <- sum(perfdat$obs=="died")
predpos <- sum(perfdat$pred=="survived")
predneg <- sum(perfdat$pred=="died")
total <- nrow(perfdat)
data.frame(pos, neg,predpos,predneg)

tp<-sum(perfdat$obs=="survived" & perfdat$pred=="survived")
tn<-sum(perfdat$obs=="died" & perfdat$pred=="died")
fp<-sum(perfdat$obs=="died" & perfdat$pred=="survived")
fn<-sum(perfdat$obs=="survived" & perfdat$pred=="died")
data.frame(tp,tn,fp,fn)

exac <- (tp+tn)/total
error <- (fp+fn)/total
sens <- tp/pos
espec <- tn/neg
prec <- tp/predpos
npv <- tn / predneg
data.frame(exac,error,sens,espec,prec,npv)
```

- Corba ROC

```
predROC = prediction(prob, test.data$Class)
ROC = performance(predROC, "tpr", "fpr")
plot(ROC, colorize = TRUE, print.cutoffs.at=seq(0,1,0.1),text.adj=c(-0.2,1.7))

auc = as.numeric(performance(predROC,"auc")@y.values)
auc

Xy <- cbind(model.matrix(Class ~ ., data = train.data)[,-1], train.data$Class)
dim(Xy)
```

RIDGE

```
columnas <- c("2.5%","97.5%")
M <- as.data.frame(matrix(ncol=length(columnas),nrow=nrow(ridge.coef),NA))
colnames(M) <- columnas
rownames(M) <- rownames(ridge.coef)

for( i in 1:52){

 rfun <- function(Xy) {

  y <- Xy[, 52]
  X <- Xy[, 1:51]

  model_final_ridge_lmin <- glmnet(x = X, y = y, alpha = 0, lambda = cv_error_ridge$lambda.min, fa
```

```
mily="binomial")

  ridge.coef <- predict(model_final_ridge_lmin,type="coefficients",s=cv_error_ridge$lambda.min)

  ridge.coef[i,1] # Gender

 }

 set.seed(10)
 result <- bcajack(x = Xy, B = 2000, func = rfun)
 #result$lims
 res <- result$stats
 #bcaplot(result)
 IC <- res[1,1]+ c(-1,1)*res[1,2]

 M[i,] <- IC
}
```

ELASTIC NET

```
columnas <- c("2.5%","97.5%")
Melastic <- as.data.frame(matrix(ncol=length(columnas),nrow=nrow(elastic.coef1),NA))
colnames(Melastic) <- columnas
rownames(Melastic) <- rownames(elastic.coef1)

for( i in 1:52){

 rfun <- function(Xy) {

  y <- Xy[, 52]
  X <- Xy[, 1:51]

  model_final_elastic_lmin <- glmnet(x = X, y = y, alpha = 0.1, lambda = cv_error_elastic1$lambda.mi
n, family="binomial")

  elastic.coef <- predict(model_final_elastic_lmin,type="coefficients",s=cv_error_elastic1$lambda.m
in)

  elastic.coef[i,1] # Gender

 }

 set.seed(10)
 result <- bcajack(x = Xy, B = 2000, func = rfun)
 #result$lims
 res <- result$stats
 #bcaplot(result)
 IC <- res[1,1]+ c(-1,1)*res[1,2]

 Melastic[i,] <- IC
}
```

8.2. MALALTIA CARDIOVASCULAR

LECTURA DE LES DADES

```
var.names <- c("ID","age", "gender", "height", "weight", "systolic", "diastolic", "cho", "glu","smoke","
alcohol","physAct","cardio")

var.type <- c("integer","integer", "factor", rep("numeric",4),rep("factor",6))

cardio <- read.csv("cardiovascular.csv", sep=";", header=TRUE, na.strings = "?", stringsAsFactors = FA
LSE, col.names = var.names, colClasses = var.type)

cardio <- cardio[,-1]
cardio$age <- trunc(cardio$age/365.25)  # years
cardio$gender <- factor(cardio$gender, labels = c("woman","man"))
cardio$cho <- factor(cardio$cho, order=TRUE,
                     levels = 1:3,
                     labels = c("n","an","wan"))
cardio$glu <- factor(cardio$glu, order=TRUE,
                   levels = 1:3,
                   labels = c("n","an","wan"))
cardio$smoke <- factor(cardio$smoke, labels = c("no","yes"))
cardio$alcohol <- factor(cardio$alcohol, labels = c("no","yes"))
cardio$physAct <- factor(cardio$physAct, labels = c("no","yes"))
cardio$cardio <- factor(cardio$cardio, labels = c("no","yes"))
cardio$systolic <- cardio$systolic
cardio$diastolic <- cardio$diastolic
```

ANÀLISI DESCRIPTIVA DE LES DADES

```
summary(cardio)

with(cardio, plot(height,weight))
abline(v=129, col="red")
abline(v=210, col="red")
abline(h=30, col="red")
abline(h=190, col="red")

cardio <- cardio[cardio$height >= 129 & cardio$height < 210, ]
cardio <- cardio[cardio$weight >=  30 & cardio$weight < 190, ]

with(cardio, plot(systolic,diastolic, xlim=c(0,500), ylim=c(0,500)))
abline(v=40, col="red")

cardio <- cardio[cardio$systolic > 40 & cardio$systolic < 300, ]
cardio <- cardio[cardio$diastolic > 10 & cardio$diastolic < 300, ]

with(cardio, plot(systolic,diastolic, xlim=c(0,300), ylim=c(0,300)))

summary(cardio)

dim(cardio)
```

```
ddnum <- cardio[,c(1,3:6)]
ddcat <- cardio[,-c(1,3:6)]
```

VARIABLES CATEGÒRIQUES

```
sapply(ddcat,table)
```

VARIABLES NUMÈRIQUES

```
library(dplyr)
columnas <- c("Mitjana","Variància","Desv. Típica","Covariància","Min","Q1","Mediana","Q3","Max")
M <- as.data.frame(matrix(ncol=length(columnas),nrow=ncol(ddnum),NA))
colnames(M) <- columnas

for(i in 1:ncol(ddnum)){

 nom <- names(ddnum[i])
 rownames(M)[i] <- nom
 res = summarise(
  ddnum,
  'mean' = round(mean(ddnum[,i], na.rm = TRUE),4),
  'var' = round(var(ddnum[,i], na.rm = TRUE), 4),
  'sd' = round(sd(ddnum[,i], na.rm = TRUE), 4),
  'cv' = round(sd/mean,4),
  'min' = round(min(ddnum[,i], na.rm = TRUE), 4),
  'Q1' = round(quantile(ddnum[,i], 0.25, names = F, na.rm = TRUE),4),
  'median' = round(median(ddnum[,i], na.rm = TRUE), 4),
  'Q3' = round(quantile(ddnum[,i], 0.75, names = F, na.rm = TRUE),4),
  'max' = round(max(ddnum[,i], na.rm = TRUE),4) )
 res
 M[i,] <- res

}

kable(M, caption = "Descrpitiva de les variables numèriques") %>%
 kable_styling(bootstrap_options = c("striped", "hover", "condensed", "responsive"), full_width = F,
position = "center") %>%
 column_spec(1:(ncol(M)+1), bold = T, border_left = T, border_right = T, color = "black")
```

OUTLIERS

```
for(i in 1:ncol(ddnum)){
nom <- names(ddnum)[i]
 boxplot(ddnum[,i],main=nom, col="lightcyan")
}

par(mfrow=c(1,2))
boxplot(ddnum[,3],main="Weight", col="lightcyan")
boxplot(ddnum[,4],main="Systolic", col="lightcyan")
```

GRÀFICS

```
for( i in c(3:6)){
```

```
  plot(cardio[,i], ylab= var.names[i], main= var.names[i])

}
```

MODEL COMPLET

```
full_model <- glm(cardio ~ age + gender + height + weight + systolic + diastolic + cho + glu + smoke +
alcohol + physAct, family="binomial", data=cardio)

summary(full_model)
```

- CORRELACIÓ

```
round(vif(full_model),2)
```

MÈTODES TRADICIONALS

```
set.seed(10)
train <- cardio$cardio %>% createDataPartition(p = 0.7, list = FALSE)
train.data  <- cardio[train, ]
test.data <- cardio[-train, ]

full_model2 <- glm(cardio ~ age + gender + height + weight + systolic + diastolic + cho + glu + smoke +
alcohol + physAct, family="binomial", data=train.data)

null_model <- glm(cardio ~ 1, family="binomial", data=train.data)

step(null_model, scope = list(lower = null_model, upper = full_model2), direction=c("both"), trace=0)

modboth1 <- glm(cardio ~ systolic + age + cho + weight + diastolic + physAct + glu + smoke + alcohol
+ height, family = "binomial", data = train.data)
summary(modboth1)
round(confint(modboth1),4)
AIC(modboth1)
```

Modboth1

```
prediccio <- predict(modboth1,test.data,type = "response")
predresp <- ifelse(prediccio>0.5, "yes","no")
perfdades<-data.frame(obs=test.data$cardio,pred= predresp)

pos <- sum(perfdades$obs=="yes") #yes malaltia
neg <- sum(perfdades$obs=="no") #no malaltia
pred_pos <- sum(perfdades$pred=="yes")
pred_neg <- sum(perfdades$pred=="no")
total <- nrow(perfdades)
data.frame(pos, neg,pred_pos,pred_neg)

tp<-sum(perfdades$obs=="yes" & perfdades$pred=="yes")
tn<-sum(perfdades$obs=="no" & perfdades$pred=="no")
fp<-sum(perfdades$obs=="no" & perfdades$pred=="yes")
fn<-sum(perfdades$obs=="yes" & perfdades$pred=="no")
data.frame(tp,tn,fp,fn)

exac <- (tp+tn)/total
error <- (fp+fn)/total
```

```
sens <- tp/pos
espec <- tn/neg
prec <- tp/pred_pos
npv <- tn / pred_neg
data.frame(exac,error,sens,espec,prec,npv)
```

- Corba ROC

```
predROC = prediction(prediccio, test.data$cardio)
ROC = performance(predROC, "tpr", "fpr")
plot(ROC, colorize = TRUE, print.cutoffs.at=seq(0,1,0.1),text.adj=c(-0.2,1.7))

auc = as.numeric(performance(predROC,"auc")@y.values)
auc
```

MÈTODES DE PENALITZACIÓ

RIDGE REGRESSION

```
x <- model.matrix(cardio ~ age + gender + height + weight + systolic + diastolic + cho + glu + smoke +
alcohol + physAct, data = train.data)[,-1]
y <- train.data$cardio

set.seed(10)
ridge <- glmnet(x, y, family = "binomial", alpha = 0)

plot(ridge, xvar = "lambda", label = TRUE)
```

DETERMINAR EL VALOR DE λ

```
set.seed(10)
cv_error_ridge <- cv.glmnet(x = x, y = y, alpha = 0, nfolds = 10, family="binomial")
plot(cv_error_ridge)

cv_error_ridge$lambda.min

cv_error_ridge$lambda.1se
```

MODEL FINAL RIDGE REGRESSION

```
model_final_ridge_lmin <- glmnet(x = x, y = y, alpha = 0, lambda = cv_error_ridge$lambda.min, famil
y="binomial")

model_final_ridge_lse <- glmnet(x = x, y = y, alpha = 0, lambda = cv_error_ridge$lambda.1se, family=
"binomial")
```

COEFICIENTS

```
ridge.coef <- predict(model_final_ridge_lmin,type="coefficients",s=cv_error_ridge$lambda.min)
sum(ridge.coef!=0)
sum(ridge.coef==0)

ridge.coef2=predict(model_final_ridge_lse,type="coefficients",s=cv_error_ridge$lambda.1se)
sum(ridge.coef2!=0)
sum(ridge.coef2==0)
```

PREDICCIONS

- Lambda mínima:

```
x.test <- model.matrix(cardio ~., test.data)[,-1]
prob <- model_final_ridge_lmin %>% predict(newx = x.test)
predresp <- ifelse(prob > 0.5,"yes", "no")
perfdades<-data.frame(obs=test.data$cardio, pred= predresp)
colnames(perfdades) <- c("obs","pred")

pos <- sum(perfdades$obs=="yes")
neg <- sum(perfdades$obs=="no")
pred_pos <- sum(perfdades$pred=="yes")
pred_neg <- sum(perfdades$pred=="no")
total <- nrow(perfdades)
data.frame(pos, neg,pred_pos,pred_neg)

tp<-sum(perfdades$obs=="yes" & perfdades$pred=="yes")
tn<-sum(perfdades$obs=="no" & perfdades$pred=="no")
fp<-sum(perfdades$obs=="no" & perfdades$pred=="yes")
fn<-sum(perfdades$obs=="yes" & perfdades$pred=="no")
data.frame(tp,tn,fp,fn)

exac <- (tp+tn)/total
error <- (fp+fn)/total
sens <- tp/pos
espec <- tn/neg
prec <- tp/pred_pos
npv <- tn / pred_neg
data.frame(exac,error,sens,espec,prec,npv)
```

- Corba ROC

```
predROC = prediction(prob, test.data$cardio)
ROC = performance(predROC, "tpr", "fpr")
plot(ROC, colorize = TRUE, print.cutoffs.at=seq(0,1,0.1),text.adj=c(-0.2,1.7))

auc = as.numeric(performance(predROC,"auc")@y.values)
auc
```

- Lambda òptima:

```
prob <- model_final_ridge_lse %>% predict(newx = x.test)
predresp <- ifelse(prob > 0.5,"yes", "no")
perfdades<-data.frame(obs=test.data$cardio,pred= predresp)
colnames(perfdades) <- c("obs","pred")

pos <- sum(perfdades$obs=="yes")
neg <- sum(perfdades$obs=="no")
pred_pos <- sum(perfdades$pred=="yes")
pred_neg <- sum(perfdades$pred=="no")
total <- nrow(perfdades)
data.frame(pos, neg,pred_pos,pred_neg)

tp<-sum(perfdades$obs=="yes" & perfdades$pred=="yes")
tn<-sum(perfdades$obs=="no" & perfdades$pred=="no")
fp<-sum(perfdades$obs=="no" & perfdades$pred=="yes")
```

```
fn<-sum(perfdades$obs=="yes" & perfdades$pred=="no")
data.frame(tp,tn,fp,fn)

exac <- (tp+tn)/total
error <- (fp+fn)/total
sens <- tp/pos
espec <- tn/neg
prec <- tp/pred_pos
npv <- tn / pred_neg
data.frame(exac,error,sens,espec,prec,npv)

predROC = prediction(prob, test.data$cardio)
ROC = performance(predROC, "tpr", "fpr")
plot(ROC, colorize = TRUE, print.cutoffs.at=seq(0,1,0.1),text.adj=c(-0.2,1.7))

auc = as.numeric(performance(predROC,"auc")@y.values)
auc
```

LASSO REGRESSION

```
set.seed(10)
lasso <- glmnet(x, y, family = "binomial", alpha = 1)

plot(lasso, xvar = "lambda", label = TRUE)
```

DETERMINAR EL VALOR DE λ

```
set.seed(10)
cv_error_lasso <- cv.glmnet(x = x, y = y, alpha = 1, nfolds = 10,family="binomial")
plot(cv_error_lasso)

cv_error_lasso$lambda.min

cv_error_lasso$lambda.1se
```

MODEL FINAL LASSO REGRESSION

```
model_final_lasso_lmin <- glmnet(x = x, y = y, alpha = 1, lambda = cv_error_lasso$lambda.min, family
="binomial")

model_final_lasso_lse <- glmnet(x = x, y = y, alpha = 1, lambda = cv_error_lasso$lambda.1se, family=
"binomial")
```

COEFICIENTS

```
lasso.coef=predict(model_final_lasso_lmin,type="coefficients",s=cv_error_lasso$lambda.min)
sum(lasso.coef!=0)
sum(lasso.coef==0)

lasso.coef2=predict(model_final_lasso_lse,type="coefficients",s=cv_error_lasso$lambda.1se)
sum(lasso.coef2!=0)
sum(lasso.coef2==0)
```

PREDICCIONS

- Lambda mínima:

```
prob <- model_final_lasso_lmin %>% predict(newx = x.test)
predresp <- ifelse(prob > 0.5,"yes", "no")
perfdades<-data.frame(obs=test.data$cardio, pred= predresp)
colnames(perfdades) <- c("obs","pred")

pos <- sum(perfdades$obs=="yes")
neg <- sum(perfdades$obs=="no")
pred_pos <- sum(perfdades$pred=="yes")
pred_neg <- sum(perfdades$pred=="no")
total <- nrow(perfdades)
data.frame(pos, neg,pred_pos,pred_neg)

tp<-sum(perfdades$obs=="yes" & perfdades$pred=="yes")
tn<-sum(perfdades$obs=="no" & perfdades$pred=="no")
fp<-sum(perfdades$obs=="no" & perfdades$pred=="yes")
fn<-sum(perfdades$obs=="yes" & perfdades$pred=="no")
data.frame(tp,tn,fp,fn)

exac <- (tp+tn)/total
error <- (fp+fn)/total
sens <- tp/pos
espec <- tn/neg
prec <- tp/pred_pos
npv <- tn / pred_neg
data.frame(exac,error,sens,espec,prec,npv)

predROC = prediction(prob, test.data$cardio)
ROC = performance(predROC, "tpr", "fpr")
plot(ROC, colorize = TRUE, print.cutoffs.at=seq(0,1,0.1),text.adj=c(-0.2,1.7))

auc = as.numeric(performance(predROC,"auc")@y.values)
auc
```

- Lambda òptima:

```
prob <- model_final_lasso_lse %>% predict(newx = x.test)
predresp <- ifelse(prob > 0.5,"yes", "no")
perfdades<-data.frame(obs=test.data$cardio, pred= predresp)
colnames(perfdades) <- c("obs","pred")

pos <- sum(perfdades$obs=="yes")
neg <- sum(perfdades$obs=="no")
pred_pos <- sum(perfdades$pred=="yes")
pred_neg <- sum(perfdades$pred=="no")
total <- nrow(perfdades)
data.frame(pos, neg,pred_pos,pred_neg)

tp<-sum(perfdades$obs=="yes" & perfdades$pred=="yes")
tn<-sum(perfdades$obs=="no" & perfdades$pred=="no")
fp<-sum(perfdades$obs=="no" & perfdades$pred=="yes")
fn<-sum(perfdades$obs=="yes" & perfdades$pred=="no")
data.frame(tp,tn,fp,fn)
```

```
exac <- (tp+tn)/total
error <- (fp+fn)/total
sens <- tp/pos
espec <- tn/neg
prec <- tp/pred_pos
npv <- tn / pred_neg
data.frame(exac,error,sens,espec,prec,npv)

predROC = prediction(prob, test.data$cardio)
ROC = performance(predROC, "tpr", "fpr")
plot(ROC, colorize = TRUE, print.cutoffs.at=seq(0,1,0.1),text.adj=c(-0.2,1.7))

auc = as.numeric(performance(predROC,"auc")@y.values)
auc
```

ELASTIC NET REGRESSION

AMB ALPHA=0.1

```
set.seed(10)
elasticnet1 <- glmnet(x, y, family = "binomial", alpha = 0.1)

plot(elasticnet1, xvar = "lambda", label = TRUE)
```

DETERMINAR EL VALOR DE λ

```
set.seed(10)
cv_error_elastic1 <- cv.glmnet(x = x, y = y, alpha = 0.1, nfolds = 10, family="binomial")
plot(cv_error_elastic1)

cv_error_elastic1$lambda.min

cv_error_elastic1$lambda.1se
```

MODEL FINAL ELASTIC NET REGRESSION

```
model_final_elastic_lmin1 <- glmnet(x = x, y = y, alpha = 0.1, lambda = cv_error_elastic1$lambda.min
, family="binomial")

model_final_elastic_lse1 <- glmnet(x = x, y = y, alpha = 0.1, lambda = cv_error_elastic1$lambda.1se, f
amily="binomial")
```

COEFICIENTS

```
elastic.coef1=predict(model_final_elastic_lmin1,type="coefficients",s=cv_error_elastic1$lambda.min
)
sum(elastic.coef1!=0)
sum(elastic.coef1==0)

elastic.coef11=predict(model_final_elastic_lse1,type="coefficients",s=cv_error_elastic1$lambda.1se)
sum(elastic.coef11!=0)
sum(elastic.coef11==0)
```

PREDICCIONS

- Lambda mínima:

```
prob <- model_final_elastic_lmin1 %>% predict(newx = x.test)
predresp <- ifelse(prob > 0.5,"yes", "no")
perfdades<-data.frame(obs=test.data$cardio, pred= predresp)
colnames(perfdades) <- c("obs","pred")

pos <- sum(perfdades$obs=="yes")
neg <- sum(perfdades$obs=="no")
pred_pos <- sum(perfdades$pred=="yes")
pred_neg <- sum(perfdades$pred=="no")
total <- nrow(perfdades)
data.frame(pos, neg,pred_pos,pred_neg)

tp<-sum(perfdades$obs=="yes" & perfdades$pred=="yes")
tn<-sum(perfdades$obs=="no" & perfdades$pred=="no")
fp<-sum(perfdades$obs=="no" & perfdades$pred=="yes")
fn<-sum(perfdades$obs=="yes" & perfdades$pred=="no")
data.frame(tp,tn,fp,fn)

exac <- (tp+tn)/total
error <- (fp+fn)/total
sens <- tp/pos
espec <- tn/neg
prec <- tp/pred_pos
npv <- tn / pred_neg
data.frame(exac,error,sens,espec,prec,npv)
```

- Corba ROC

```
predROC = prediction(prob, test.data$cardio)
ROC = performance(predROC, "tpr", "fpr")
plot(ROC, colorize = TRUE, print.cutoffs.at=seq(0,1,0.1),text.adj=c(-0.2,1.7))

auc = as.numeric(performance(predROC,"auc")@y.values)
auc
```

- Lambda òptima:

```
prob <- model_final_elastic_lse1 %>% predict(newx = x.test)
predresp <- ifelse(prob > 0.5,"yes", "no")
perfdades<-data.frame(obs=test.data$cardio, pred= predresp)
colnames(perfdades) <- c("obs","pred")

pos <- sum(perfdades$obs=="yes")
neg <- sum(perfdades$obs=="no")
pred_pos <- sum(perfdades$pred=="yes")
pred_neg <- sum(perfdades$pred=="no")
total <- nrow(perfdades)
data.frame(pos, neg,pred_pos,pred_neg)

tp<-sum(perfdades$obs=="yes" & perfdades$pred=="yes")
tn<-sum(perfdades$obs=="no" & perfdades$pred=="no")
fp<-sum(perfdades$obs=="no" & perfdades$pred=="yes")
fn<-sum(perfdades$obs=="yes" & perfdades$pred=="no")
data.frame(tp,tn,fp,fn)
```

```
exac <- (tp+tn)/total
error <- (fp+fn)/total
sens <- tp/pos
espec <- tn/neg
prec <- tp/pred_pos
npv <- tn / pred_neg
data.frame(exac,error,sens,espec,prec,npv)
```

- Corba ROC

```
predROC = prediction(prob, test.data$cardio)
ROC = performance(predROC, "tpr", "fpr")
plot(ROC, colorize = TRUE, print.cutoffs.at=seq(0,1,0.1),text.adj=c(-0.2,1.7))

auc = as.numeric(performance(predROC,"auc")@y.values)
auc
```

AMB ALPHA=0.25

```
set.seed(10)
elasticnet2 <- glmnet(x, y, family = "binomial", alpha = 0.25)

plot(elasticnet2, xvar = "lambda", label = TRUE)
```

DETERMINAR EL VALOR DE λ

```
set.seed(10)
cv_error_elastic2 <- cv.glmnet(x = x, y = y, alpha = 0.25, nfolds = 10, family="binomial")
plot(cv_error_elastic2)

cv_error_elastic2$lambda.min

cv_error_elastic2$lambda.1se
```

MODEL FINAL ELASTIC NET REGRESSION

```
model_final_elastic_lmin2 <- glmnet(x = x, y = y, alpha = 0.25, lambda = cv_error_elastic2$lambda.mi
n, family="binomial")

model_final_elastic_lse2 <- glmnet(x = x, y = y, alpha = 0.25, lambda = cv_error_elastic2$lambda.1se,
family="binomial")
```

COEFICIENTS

```
elastic.coef2=predict(model_final_elastic_lmin2,type="coefficients",s=cv_error_elastic2$lambda.min
)
sum(elastic.coef2!=0)
sum(elastic.coef2==0)

elastic.coef22=predict(model_final_elastic_lse2,type="coefficients",s=cv_error_elastic2$lambda.1se)
sum(elastic.coef22!=0)
sum(elastic.coef22==0)
```

PREDICCIONS

- Lambda mínima:

```
prob <- model_final_elastic_lmin2 %>% predict(newx = x.test)
predresp <- ifelse(prob > 0.5,"yes", "no")
perfdades<-data.frame(obs=test.data$cardio, pred= predresp)
colnames(perfdades) <- c("obs","pred")

pos <- sum(perfdades$obs=="yes")
neg <- sum(perfdades$obs=="no")
pred_pos <- sum(perfdades$pred=="yes")
pred_neg <- sum(perfdades$pred=="no")
total <- nrow(perfdades)
data.frame(pos, neg,pred_pos,pred_neg)

tp<-sum(perfdades$obs=="yes" & perfdades$pred=="yes")
tn<-sum(perfdades$obs=="no" & perfdades$pred=="no")
fp<-sum(perfdades$obs=="no" & perfdades$pred=="yes")
fn<-sum(perfdades$obs=="yes" & perfdades$pred=="no")
data.frame(tp,tn,fp,fn)

exac <- (tp+tn)/total
error <- (fp+fn)/total
sens <- tp/pos
espec <- tn/neg
prec <- tp/pred_pos
npv <- tn / pred_neg
data.frame(exac,error,sens,espec,prec,npv)
```

- Corba ROC

```
predROC = prediction(prob, test.data$cardio)
ROC = performance(predROC, "tpr", "fpr")
plot(ROC, colorize = TRUE, print.cutoffs.at=seq(0,1,0.1),text.adj=c(-0.2,1.7))

auc = as.numeric(performance(predROC,"auc")@y.values)
auc
```

- Lambda òptima:

```
prob <- model_final_elastic_lse2 %>% predict(newx = x.test)
predresp <- ifelse(prob > 0.5,"yes", "no")
perfdades<-data.frame(obs=test.data$cardio, pred= predresp)
colnames(perfdades) <- c("obs","pred")

pos <- sum(perfdades$obs=="yes")
neg <- sum(perfdades$obs=="no")
pred_pos <- sum(perfdades$pred=="yes")
pred_neg <- sum(perfdades$pred=="no")
total <- nrow(perfdades)
data.frame(pos, neg,pred_pos,pred_neg)

tp<-sum(perfdades$obs=="yes" & perfdades$pred=="yes")
tn<-sum(perfdades$obs=="no" & perfdades$pred=="no")
fp<-sum(perfdades$obs=="no" & perfdades$pred=="yes")
fn<-sum(perfdades$obs=="yes" & perfdades$pred=="no")
data.frame(tp,tn,fp,fn)
```

```
exac <- (tp+tn)/total
error <- (fp+fn)/total
sens <- tp/pos
espec <- tn/neg
prec <- tp/pred_pos
npv <- tn / pred_neg
data.frame(exac,error,sens,espec,prec,npv)
```

- Corba ROC

```
predROC = prediction(prob, test.data$cardio)
ROC = performance(predROC, "tpr", "fpr")
plot(ROC, colorize = TRUE, print.cutoffs.at=seq(0,1,0.1),text.adj=c(-0.2,1.7))

auc = as.numeric(performance(predROC,"auc")@y.values)
auc
```

AMB ALPHA= 0.5

```
set.seed(10)
elasticnet3 <- glmnet(x, y, family = "binomial", alpha = 0.5)

plot(elasticnet3, xvar = "lambda", label = TRUE)
```

DETERMINAR EL VALOR DE λ

```
set.seed(10)
cv_error_elastic3 <- cv.glmnet(x = x, y = y, alpha = 0.5, nfolds = 10, family="binomial")
plot(cv_error_elastic3)

cv_error_elastic3$lambda.min

cv_error_elastic3$lambda.1se
```

MODEL FINAL ELASTIC NET REGRESSION

```
model_final_elastic_lmin3 <- glmnet(x = x, y = y, alpha = 0.5, lambda = cv_error_elastic3$lambda.min
, family="binomial")

model_final_elastic_lse3 <- glmnet(x = x, y = y, alpha = 0.5, lambda = cv_error_elastic3$lambda.1se, f
amily="binomial")
```

COEFICIENTS

```
elastic.coef3=predict(model_final_elastic_lmin3,type="coefficients",s=cv_error_elastic3$lambda.min
)
sum(elastic.coef3!=0)
sum(elastic.coef3==0)

elastic.coef33=predict(model_final_elastic_lse3,type="coefficients",s=cv_error_elastic3$lambda.1se)
sum(elastic.coef33!=0)
sum(elastic.coef33==0)
```

- Lambda mínima:

```
prob <- model_final_elastic_lmin3 %>% predict(newx = x.test)
predresp <- ifelse(prob > 0.5,"yes", "no")
perfdades<-data.frame(obs=test.data$cardio, pred= predresp)
```

```
colnames(perfdades) <- c("obs","pred")

pos <- sum(perfdades$obs=="yes")
neg <- sum(perfdades$obs=="no")
pred_pos <- sum(perfdades$pred=="yes")
pred_neg <- sum(perfdades$pred=="no")
total <- nrow(perfdades)
data.frame(pos, neg,pred_pos,pred_neg)

tp<-sum(perfdades$obs=="yes" & perfdades$pred=="yes")
tn<-sum(perfdades$obs=="no" & perfdades$pred=="no")
fp<-sum(perfdades$obs=="no" & perfdades$pred=="yes")
fn<-sum(perfdades$obs=="yes" & perfdades$pred=="no")
data.frame(tp,tn,fp,fn)

exac <- (tp+tn)/total
error <- (fp+fn)/total
sens <- tp/pos
espec <- tn/neg
prec <- tp/pred_pos
npv <- tn / pred_neg
data.frame(exac,error,sens,espec,prec,npv)
```

- Corba ROC

```
predROC = prediction(prob, test.data$cardio)
ROC = performance(predROC, "tpr", "fpr")
plot(ROC, colorize = TRUE, print.cutoffs.at=seq(0,1,0.1),text.adj=c(-0.2,1.7))

auc = as.numeric(performance(predROC,"auc")@y.values)
auc
```

- Lambda òptima:

```
prob <- model_final_elastic_lse3 %>% predict(newx = x.test)
predresp <- ifelse(prob > 0.5,"yes", "no")
perfdades<-data.frame(obs=test.data$cardio, pred= predresp)
colnames(perfdades) <- c("obs","pred")

pos <- sum(perfdades$obs=="yes")
neg <- sum(perfdades$obs=="no")
pred_pos <- sum(perfdades$pred=="yes")
pred_neg <- sum(perfdades$pred=="no")
total <- nrow(perfdades)
data.frame(pos, neg,pred_pos,pred_neg)

tp<-sum(perfdades$obs=="yes" & perfdades$pred=="yes")
tn<-sum(perfdades$obs=="no" & perfdades$pred=="no")
fp<-sum(perfdades$obs=="no" & perfdades$pred=="yes")
fn<-sum(perfdades$obs=="yes" & perfdades$pred=="no")
data.frame(tp,tn,fp,fn)

exac <- (tp+tn)/total
error <- (fp+fn)/total
sens <- tp/pos
```

```
espec <- tn/neg
prec <- tp/pred_pos
npv <- tn / pred_neg
data.frame(exac,error,sens,espec,prec,npv)
```

- Corba ROC

```
predROC = prediction(prob, test.data$cardio)
ROC = performance(predROC, "tpr", "fpr")
plot(ROC, colorize = TRUE, print.cutoffs.at=seq(0,1,0.1),text.adj=c(-0.2,1.7))

auc = as.numeric(performance(predROC,"auc")@y.values)
auc
```

AMB ALPHA= 0.75

```
set.seed(10)
elasticnet4 <- glmnet(x, y, family = "binomial", alpha = 0.75)

plot(elasticnet4, xvar = "lambda", label = TRUE)
```

DETERMINAR EL VALOR DE λ

```
set.seed(10)
cv_error_elastic4 <- cv.glmnet(x = x, y = y, alpha = 0.75, nfolds = 10, family="binomial")
plot(cv_error_elastic4)

cv_error_elastic4$lambda.min

cv_error_elastic4$lambda.1se
```

MODEL FINAL ELASTIC NET REGRESSION

```
model_final_elastic_lmin4 <- glmnet(x = x, y = y, alpha = 0.75, lambda = cv_error_elastic4$lambda.mi
n, family="binomial")

model_final_elastic_lse4 <- glmnet(x = x, y = y, alpha = 0.75, lambda = cv_error_elastic4$lambda.1se,
family="binomial")
```

COEFICIENTS

```
elastic.coef4=predict(model_final_elastic_lmin4,type="coefficients",s=cv_error_elastic4$lambda.min
)
sum(elastic.coef4!=0)
sum(elastic.coef4==0)

elastic.coef44=predict(model_final_elastic_lse4,type="coefficients",s=cv_error_elastic4$lambda.1se)
sum(elastic.coef44!=0)
sum(elastic.coef44==0)
```

PREDICCIONS

- Lambda mínima:

```
prob <- model_final_elastic_lmin4 %>% predict(newx = x.test)
predresp <- ifelse(prob > 0.5,"yes", "no")
perfdades<-data.frame(obs=test.data$cardio, pred= predresp)
colnames(perfdades) <- c("obs","pred")
```

```
pos <- sum(perfdades$obs=="yes")
neg <- sum(perfdades$obs=="no")
pred_pos <- sum(perfdades$pred=="yes")
pred_neg <- sum(perfdades$pred=="no")
total <- nrow(perfdades)
data.frame(pos, neg,pred_pos,pred_neg)

tp<-sum(perfdades$obs=="yes" & perfdades$pred=="yes")
tn<-sum(perfdades$obs=="no" & perfdades$pred=="no")
fp<-sum(perfdades$obs=="no" & perfdades$pred=="yes")
fn<-sum(perfdades$obs=="yes" & perfdades$pred=="no")
data.frame(tp,tn,fp,fn)

exac <- (tp+tn)/total
error <- (fp+fn)/total
sens <- tp/pos
espec <- tn/neg
prec <- tp/pred_pos
npv <- tn / pred_neg
data.frame(exac,error,sens,espec,prec,npv)
```

- Corba ROC

```
predROC = prediction(prob, test.data$cardio)
ROC = performance(predROC, "tpr", "fpr")
plot(ROC, colorize = TRUE, print.cutoffs.at=seq(0,1,0.1),text.adj=c(-0.2,1.7))

auc = as.numeric(performance(predROC,"auc")@y.values)
auc
```

- Lambda òptima:

```
prob <- model_final_elastic_lse4 %>% predict(newx = x.test)
predresp <- ifelse(prob > 0.5,"yes", "no")
perfdades<-data.frame(obs=test.data$cardio, pred= predresp)
colnames(perfdades) <- c("obs","pred")

pos <- sum(perfdades$obs=="yes")
neg <- sum(perfdades$obs=="no")
pred_pos <- sum(perfdades$pred=="yes")
pred_neg <- sum(perfdades$pred=="no")
total <- nrow(perfdades)
data.frame(pos, neg,pred_pos,pred_neg)

tp<-sum(perfdades$obs=="yes" & perfdades$pred=="yes")
tn<-sum(perfdades$obs=="no" & perfdades$pred=="no")
fp<-sum(perfdades$obs=="no" & perfdades$pred=="yes")
fn<-sum(perfdades$obs=="yes" & perfdades$pred=="no")
data.frame(tp,tn,fp,fn)

exac <- (tp+tn)/total
error <- (fp+fn)/total
sens <- tp/pos
espec <- tn/neg
```

```
prec <- tp/pred_pos
npv <- tn / pred_neg
data.frame(exac,error,sens,espec,prec,npv)
```

- Corba ROC

```
predROC = prediction(prob, test.data$cardio)
ROC = performance(predROC, "tpr", "fpr")
plot(ROC, colorize = TRUE, print.cutoffs.at=seq(0,1,0.1),text.adj=c(-0.2,1.7))

auc = as.numeric(performance(predROC,"auc")@y.values)
auc
```

AMB ALPHA= 0.95

```
set.seed(10)
elasticnet5 <- glmnet(x, y, family = "binomial", alpha = 0.95)

plot(elasticnet5, xvar = "lambda", label = TRUE)
```

DETERMINAR EL VALOR DE λ

```
set.seed(10)
cv_error_elastic5 <- cv.glmnet(x = x, y = y, alpha = 0.95, nfolds = 10, family="binomial")
plot(cv_error_elastic5)

cv_error_elastic5$lambda.min

cv_error_elastic5$lambda.1se
```

MODEL FINAL ELASTIC NET REGRESSION

```
model_final_elastic_lmin5 <- glmnet(x = x, y = y, alpha = 0.95, lambda = cv_error_elastic5$lambda.mi
n, family="binomial")

model_final_elastic_lse5 <- glmnet(x = x, y = y, alpha = 0.95, lambda = cv_error_elastic5$lambda.1se,
family="binomial")
```

COEFICIENTS

```
elastic.coef5=predict(model_final_elastic_lmin5,type="coefficients",s=cv_error_elastic5$lambda.min
)
sum(elastic.coef5!=0)
sum(elastic.coef5==0)

elastic.coef55=predict(model_final_elastic_lse5,type="coefficients",s=cv_error_elastic5$lambda.1se)
sum(elastic.coef55!=0)
sum(elastic.coef55==0)
```

PREDICCIONS

- Lambda mínima:

```
prob <- model_final_elastic_lmin5 %>% predict(newx = x.test)
predresp <- ifelse(prob > 0.5,"yes", "no")
perfdades<-data.frame(obs=test.data$cardio, pred= predresp)
colnames(perfdades) <- c("obs","pred")
```

```
pos <- sum(perfdades$obs=="yes")
neg <- sum(perfdades$obs=="no")
pred_pos <- sum(perfdades$pred=="yes")
pred_neg <- sum(perfdades$pred=="no")
total <- nrow(perfdades)
data.frame(pos, neg,pred_pos,pred_neg)

tp<-sum(perfdades$obs=="yes" & perfdades$pred=="yes")
tn<-sum(perfdades$obs=="no" & perfdades$pred=="no")
fp<-sum(perfdades$obs=="no" & perfdades$pred=="yes")
fn<-sum(perfdades$obs=="yes" & perfdades$pred=="no")
data.frame(tp,tn,fp,fn)

exac <- (tp+tn)/total
error <- (fp+fn)/total
sens <- tp/pos
espec <- tn/neg
prec <- tp/pred_pos
npv <- tn / pred_neg
data.frame(exac,error,sens,espec,prec,npv)
```

- Corba ROC

```
predROC = prediction(prob, test.data$cardio)
ROC = performance(predROC, "tpr", "fpr")
plot(ROC, colorize = TRUE, print.cutoffs.at=seq(0,1,0.1),text.adj=c(-0.2,1.7))

auc = as.numeric(performance(predROC,"auc")@y.values)
auc
```

- Lambda òptima:

```
prob <- model_final_elastic_lse5 %>% predict(newx = x.test)
predresp <- ifelse(prob > 0.5,"yes", "no")
perfdades<-data.frame(obs=test.data$cardio, pred= predresp)
colnames(perfdades) <- c("obs","pred")

pos <- sum(perfdades$obs=="yes")
neg <- sum(perfdades$obs=="no")
pred_pos <- sum(perfdades$pred=="yes")
pred_neg <- sum(perfdades$pred=="no")
total <- nrow(perfdades)
data.frame(pos, neg,pred_pos,pred_neg)

tp<-sum(perfdades$obs=="yes" & perfdades$pred=="yes")
tn<-sum(perfdades$obs=="no" & perfdades$pred=="no")
fp<-sum(perfdades$obs=="no" & perfdades$pred=="yes")
fn<-sum(perfdades$obs=="yes" & perfdades$pred=="no")
data.frame(tp,tn,fp,fn)

exac <- (tp+tn)/total
error <- (fp+fn)/total
sens <- tp/pos
espec <- tn/neg
prec <- tp/pred_pos
```

```
npv <- tn / pred_neg
data.frame(exac,error,sens,espec,prec,npv)
```

- Corba ROC

```
predROC = prediction(prob, test.data$cardio)
ROC = performance(predROC, "tpr", "fpr")
plot(ROC, colorize = TRUE, print.cutoffs.at=seq(0,1,0.1),text.adj=c(-0.2,1.7))

auc = as.numeric(performance(predROC,"auc")@y.values)
auc
```