Aprenentatge Automàtic I

Grau en Ciència i Enginyeria de Dades

Estudi comparatiu de models d'aprenentatge automàtic per a la detecció de cardiopaties

---

Autors:
Laia Mogas Pladevall
Roger Bargalló Roselló

Professors:
Marta Arias Vicente
Alexis Molina Martínez de los Reyes

Quadrimestre de primavera 2023-2024

Abstract

Preventive medicine is key in reducing deaths from heart disease. This project aims to evaluate different machine learning models for its detection. The dataset used has been obtained from the UCI Machine Learning Heart disease dataset. It contains 76 variables of 617 patients related to heart problems. After properly preprocessing the data, the following supervised models have been fitted with the best parameters according to the F2-score: QDA, LDA, Naive Bayes, Logistic Regression, KNN, Random Forest, Gradient Boosting, and SVC. Considering the metrics and properties of the models, Gradient Boosting has proven to be the most advantageous model. Subsequently, unimportant variables have been removed. In the test, an F2-score = 0.8923 and an accuracy = 0.8381 were obtained.

Resum

La medicina preventiva és clau en la reducció de morts per cardiopaties. Aquest projecte té com a objectiu avaluar diferents models d'aprenentatge automàtic per a la seva detecció. El dataset utilitzat ha estat obtingut a partir de Heart disease de UCI Machine Learning. Conté 76 variables de 617 pacients relacionades amb problemes cardíacs. Després de preprocessar degudament les dades, els següents models supervisats han estat ajustats amb amb els millors paràmetres segons la F2-score: QDA, LDA, Naive Bayes, Regressió Logística, KNN, Random Forest, Gradient Boosting i SVC. Tenint en compte les mètriques i propietats dels models, Gradient Boosting és el model més avantatjós. Posteriorment, s'han eliminat variables de poca importància. Al test s'ha obtingut un F2-score = 0.8923 i una accuracy = 0.8381.

Índex

1 Introducció

L'Organització Mundial de la Salut estima que 17.9 milions de morts són resultat de malalties cardiovasculars, constituint la primera causa de defunció a escala global[1]. En el context actual, cada cop s'aposta més per la medicina preventiva, de manera que disposar d'un sistema eficient per detectar prematurament les cardiopaties és clau per a evitar conseqüències fatals a llarg termini.

En aquest projecte, avaluarem la capacitat predictiva de diferents models d'aprenentatge supervisat en la diagnosi de malalties cardíaques. Primerament, es durà a terme un extens preprocessessat de les dades per tal de poder ser explotades òptimament. Posteriorment, s'ajustaran diferents models supervisats i s'escollirà el de millor performance en funció de diverses mètriques, prioritzant la F2-score. Seguidament, intentarem millorar el model escollit per obtenir el màxim rendiment possible i estimarem l'error de generalització.

El dataset utilitzat és Heart disease - UCI Machine Learning Repository, publicat en data 30/6/1988. Esta format per dades de quatre centres mèdics d'Europa i Estats Units:

1. Cleveland Clinic Foundation (cleveland.data)
2. Hungarian Institute of Cardiology, Budapest (hungarian.data)
3. V.A. Medical Center, Long Beach, CA (long-beach-va.data)
4. University Hospital, Zurich, Switzerland (switzerland.data)

En primer lloc, notem que l'arxiu cleveland.data està corromput, ja que conté caràcters il·legibles. De fet, això s'indica en una anotació del dataset. Per tant, ens haurem de limitar a la resta de centres mèdics. Com que tots els datasets contenen les mateixes 76 variables A podem combinar els tres datasets restants en un de sol. Malauradament, els arxius no estan estructurats com desitjaríem, per tant, farem un script que ens estructuri bé els arxius. Una vegada els tenim ben estructurats els combinarem.

Cal destacar que havíem considerat utilitzar dos hospitals com a entrenament i l'hospital restant com a test, però com provenen de diferents països, amb diferents nivells socioeconòmics i sistemes sanitaris, probablement les distribucions no siguin les mateixes. En conseqüència, qualsevol model que proposéssim tindria baixa performance al test. En canvi, si tots els hospitals fossin del mateix país, segurament sí que haguéssim aplicat aquesta metodologia.

En el dataset combinat (d'ara endavant anomenat simplement dataset) tenim 617 files i 76 variables.

	id	ccf	age	sex	painloc	painexer	relrest	pncaden	cp	trestbps	...	rcaprox	rcadist	lvx1	lvx2	lvx3	lvx4	lvf	cathef	junk	name
0	0	0	40	1	1	0	0	-9	2	140	...	-9	-9	1	1	1	1	1	-9.0	-9.0	name
1	1	0	49	0	1	0	0	-9	3	160	...	-9	-9	1	1	1	1	1	-9.0	-9.0	name
2	2	0	37	1	1	0	0	-9	2	130	...	-9	-9	1	1	1	1	1	-9.0	-9.0	name
3	3	0	48	0	1	1	1	-9	4	138	...	2	-9	1	1	1	1	1	-9.0	-9.0	name
4	4	0	54	1	1	0	1	-9	3	150	...	1	-9	1	1	1	1	1	-9.0	-9.0	name

Figura 1: Head del dataset

[1] World Health Organization on Cardiovascular Diseases

2 Exploració i Preprocessing de les dades

2.1 Feature engineering

La variable target originalment prenia quatre valors en funció de la severitat de la cardiopatia, sent el 0 l'únic valor descrit al repositori (absència de cardiopatia). En comprovar la documentació, aquesta només feia referència a la següent classificació binària: Value 0: $< 50\%$ diameter narrowing i Value 1: $> 50\%$ diameter narrowing.

A causa de l'absència d'informació relativa als valors 1, 2, 3, 4, hem optat per convertir el problema en un problema de classificació binària, on els valors 2, 3 i 4 els mapegem a 1 i considerem que 0 és absència de malaltia i 1 és presència.

D'altra banda, també teníem dues variables molt relacionades: *rldv5* i *rldv5e*, on una mesura l'alçada (intuïm que del pic d'algun mesurament) normal i l'altra l'alçada després de realitzar exercici. En aquest cas, el que ens aporta una informació valuosa per les prediccions és la diferència entre elles. Així doncs, prescindirem de la variable *rldv5e* i crearem la variable *diff_rldv5*, la qual serà la diferència entre elles.

2.2 Eliminació de variables

A la mateixa documentació apareixien algunes variables indicades com a "dummy", "not used"o ïrrelevant"sense cap explicació addicional. Altres no incloïen cap mena d'informació i no hem aconseguit esbrinar què representen. Per precaució, hem decidit prescindir de totes aquestes variables. També hem suprimit variables que indicaven dates de la realització de proves o la columna de noms dels pacients. També tenim algunes variables que no prenen valors d'acord amb la seva descripció, com és el cas de *proto*, que segons la seva descripció hauria de ser categòrica, però veiem que pren valors diferents dels indicats. En conseqüència, i a causa del fet que teníem més de 70 variables, hem decidit eliminar-les.

2.3 Missing values

A partir de l'exploració de les dades, hem pogut constatar que els missing values no estan indicats de consistentment. A la documentació consta que alguns missing values estan indicats amb un -9. Així i tot, ens hem adonat que alguns estaven indicats amb 0. Substituirem tots aquests valors per NaNs.

Com que més endavant farem imputació de valors, hem eliminat totes les variables numèriques amb més d'un 30% de NaNs. Les categòriques no les imputarem, sinó que eliminarem les files amb NaNs. Per això, hem sigut més estrictes amb el nombre de NaNs a les categories, permetent un màxim de 10% de NaNs. Cal destacar que moltes d'aquestes variables haurien sigut de gran utilitat per a la tasca, com el colesterol o si la persona és fumadora. Malauradament, el dataset té menys qualitat del que hauria.

2.4 Valors atípics

Atès que es tracta d'un dataset de cardiopaties i es poden donar situacions fora de l'usual, hem sigut relativament permissius quant als outliers. Tot i així, com més endavant es veurà hem obtingut resulats similars en models sensibles a outliers i en models més robustos.

Tanmateix, en el cas de la variable met és massa dràstica la diferència. Consultant referències, hem descobert que més de 25 mets és infactible pels éssers humans. De fet, fins i tot trobem valors al dataset que sobrepassen enormement aquesta fita. Així doncs, hem considerat que eren errors de mesurament o unitats incorrectes. Hem decidit indicar-los com a NaNs per a imputar-los més endavant en lloc d'eliminar les instàncies, ja que tenim una quantitat força limitada de dades.

La variable tpeakbpd també té un valor inversemblant perquè és massa baix, de manera que també cel ataloguem com NaN.

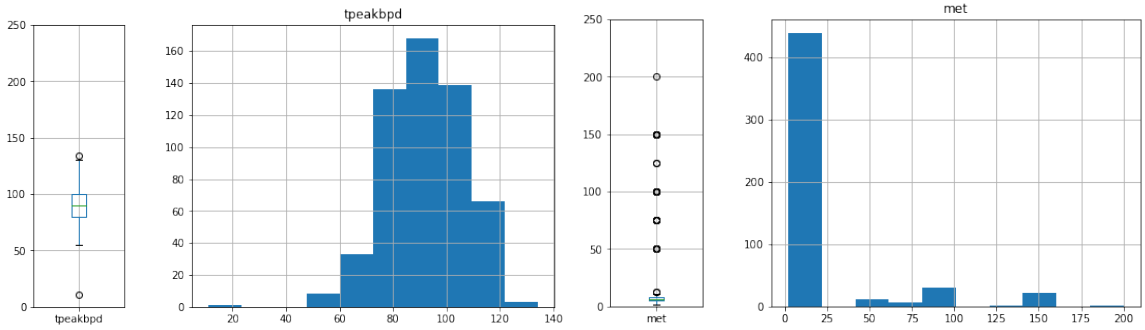

Figura 2: Histogrames de tpeakbpd i met

2.5 Distribució de les variables

És convenient mirar les distribucions de les variables per fer-nos una idea de l'impacte que puguin tenir en els models.

Clarament, observem com la distribució d'algunes variables numèriques és diferent per malalts que per no malalts. Les diferències de distribució més evidents entre malalts i no malalats les trobem a *age, thaldur, thalach, thalrest.*

Respecte les variables categòriques, *painexer* i *relrest*, a priori semblen ser bons indicadors de malaltia, ja que depenent de si valen 0 o 1 la proporció de malalts és molt diferent.
D'altra banda, veiem que algunes variables categòriques estan desequilibrades, com *cp*, *painloc* i *sex*. En el cas de *sex*, veiem que les dones estan infrarrepresentades. A més, hi ha més dones sanes que dones malaltes, mentre que amb els homes és a l'inrevés. Haurem d'anar amb compte amb si els nostres models infradiagnostiquen les dones.

Quant a la variable target, *num*, veiem que està força equilibrada amb 304 positius i 220 negatius.

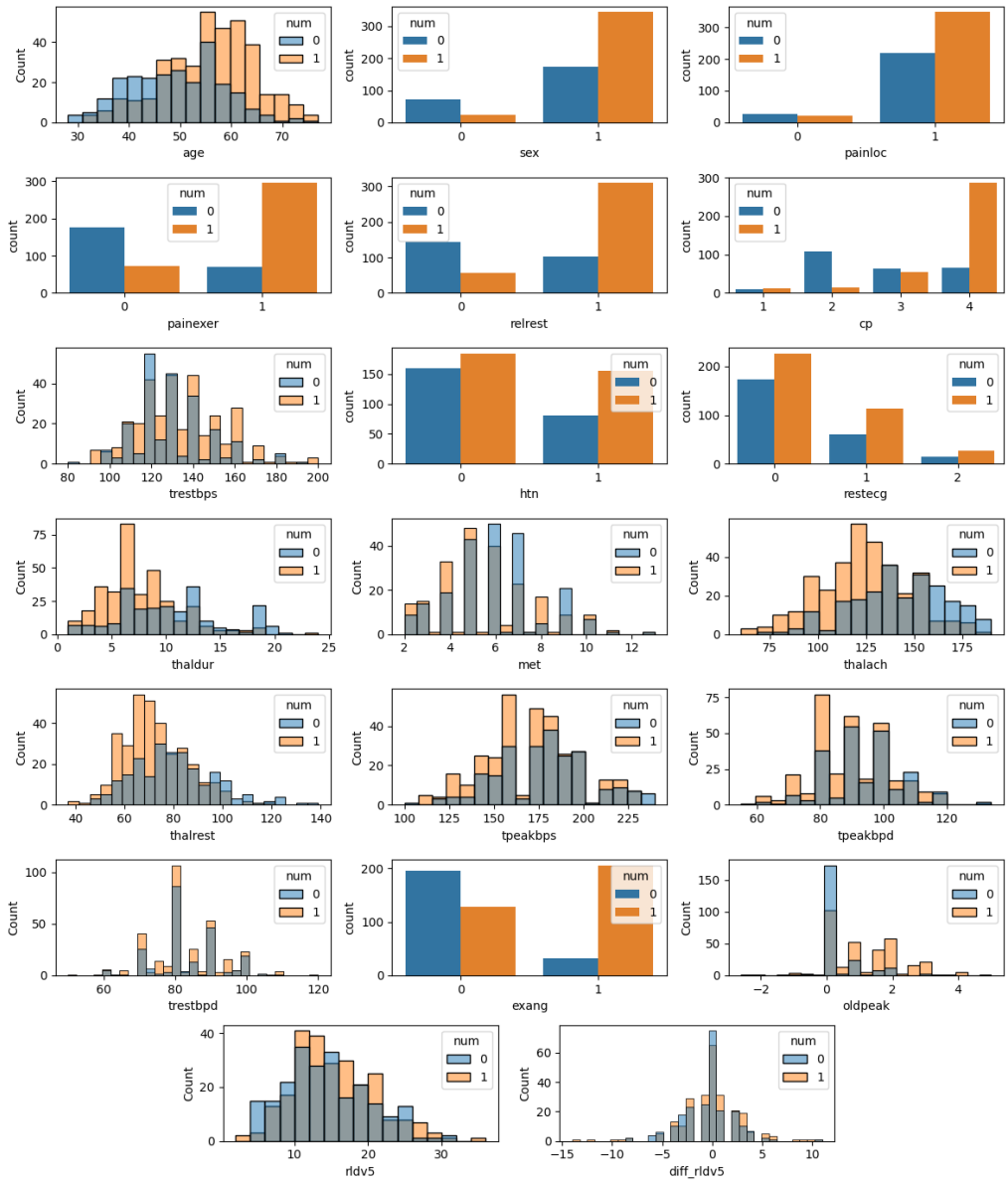

Figura 3: Distribució de les variables respecte al valor del target

2.6 Correlació entre variables

Ara mirarem la correlació entre les variables numèriques[2]. Podem observar que no tenim cap correlació major a 0.8, que seria el llindar que utilizaríem per decidir si eliminar una variable o no a causa de multicolinearitat. No obstant això, tenim diverses correlacions força destacables, com met i thaldur, trestbpd i trestbps, thalach i thaldur, i thalrest i thalach.

[2]Per a més detall consultar A.1

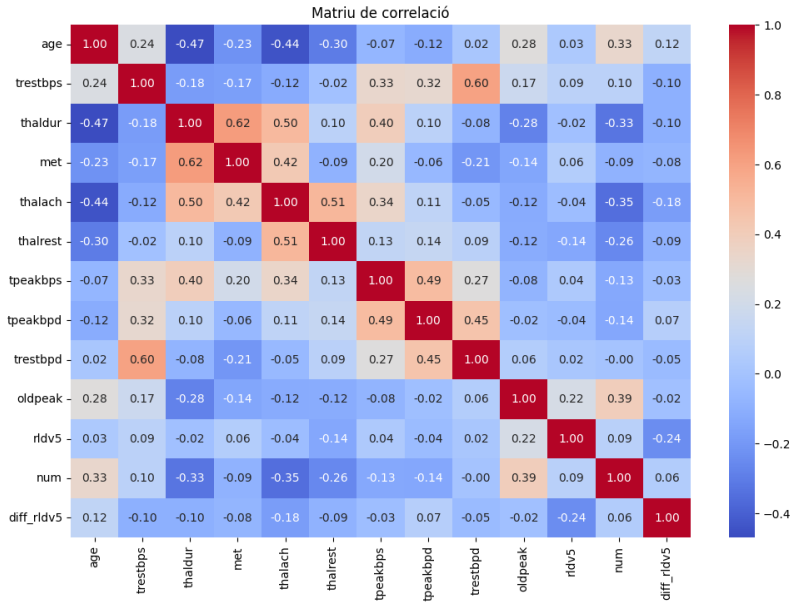

Figura 4: Matriu de correlació de les variables numèriques

2.7 Partició train-test

Abans de fer els últims passos de preprocessing, hem dedicat un 80% de les dades al train i el 20% restant al test a fi d'evitar data leakage. Per garantir la reproducibilitat hem fixat *random_state* = 1234.

2.8 Escalat de variables numèriques

Les magnituds de les dades numèriques no són les mateixes, de manera que hi ha el risc que els models es centrin només en les variables de major magnitudd. Així doncs, escalem les variables numèriques amb MinMax. Per tal d'evitar data leakage, hem ajustat el scaler al train i hem escalat el train i el test utilitzant aquest ajust.

2.9 Encoding de variables categòriques

Per tal que els models puguin interpretar les variables categòriques, aplicarem one-hot encoding amb la funció de pandas get_dummies.

2.10 Tractament de missing values

Donat que la majoria de models d'aprenentatge supervisat de classificació no accepten valors NaN, i perquè és considerat bona pràctica no tenir-ne, imputarem els missing values. Només imputarem variables numèriques, mentre que les files amb NaNs a les variables categòriques simplement les eliminarem. Hem decidit seguir aquest procediment perquè un error en les variables numèriques no resulta gaire perniciós sempre que estigui en un rang proper al valor que s'hauria d'haver imputat.

En el cas de les categòriques, en canvi, els errors comesos tindrien un major impacte en els resultats. Així doncs, eliminem les files que tenen algun valor NaN en alguna columna categòrica, i passem de 616 a 524 files. Com que aconseguim mantenir més del 80% de les files originals, considerarem que aquesta reducció és acceptable.

D'altra banda, per imputar les dades numèriques utilitzarem el predictor K-Nearest Neighbor (KNN), ja que proporciona estimacions més robustes de les features que altres mètodes com la mitjana. El predictor KNN utilitza un veïnat local per fer les prediccions, ja que si entenem cada fila com un punt en l'espai de variables, i fixada una funció distància i una funció d'imputació, agafa els k punts més propers i utilitza la funció d'imputació en els valors de la columna que correspon al valor que volem imputar per fer la predicció. Tenint en compte això, triem com a funció de distància la mètrica euclidiana i com a funció d'imputació la mitjana aritmètica.

	age	trestbps	thaldur	met	thalach	thalrest	tpeakbps	tpeakbpd	trestbpd	oldpeak	diff_rldv5
0	0.244898	0.444444	0.739130	0.454545	0.861538	0.480392	0.714286	0.675676	0.514286	0.285714	0.222222
1	0.428571	0.629630	0.391304	0.454545	0.738462	0.617647	0.857143	0.621622	0.571429	0.428571	0.407407
2	0.183673	0.351852	0.391304	0.272727	0.292308	0.205882	0.571429	0.540541	0.428571	0.285714	0.333333
3	0.408163	0.425926	0.173913	0.181818	0.369231	0.166667	0.785714	0.621622	0.514286	0.500000	0.555556
4	0.530612	0.537037	0.043478	0.090909	0.476923	0.362745	0.214286	0.540541	0.571429	0.285714	0.296296
...	...	...	...	...	...	...	...	...	...	...	...
511	0.693878	0.611111	0.304348	0.545455	0.600000	0.480392	0.728571	0.513514	0.571429	0.285714	0.518519
512	0.367347	0.388889	0.195652	0.454545	0.507692	0.500000	0.528571	0.729730	0.571429	0.285714	0.370370
513	0.530612	0.324074	0.282609	0.545455	0.723077	0.450980	0.414286	0.324324	0.400000	0.285714	0.444444
514	0.551020	0.277778	0.186957	0.272727	0.307692	0.362745	0.785714	0.540541	0.285714	0.285714	0.370370
515	0.693878	0.259259	0.247826	0.454545	0.253846	0.294118	0.457143	0.675676	0.428571	0.285714	0.296296

Figura 5: Variables numèriques escalades

Com amb el scaler, hem ajustat el KNN només amb el train amb l'objectiu d'evitar data leakage del test. Quant a l'hiperparàmetre k, per simplicitat hem mantingut el valor per defecte, $k = 5$. A més, és un valor prou gran per a assumir certa robustesa.

3 Mètriques

Com que el nostre dataset és sobre dades mèdiques i, per tant, és molt més greu diagnosticar com no malalt (0) a una persona malalta que no pas al revés. En cas de fals positiu, més endavant es poden fer més proves mèdiques que descartin la malaltia. Per tant, volem una mètrica que penalitzi especialment els falsos negatius. Aquesta mètrica serà el recall, que es defineix com:

$$recall = \frac{TP}{TP + FN}$$

On TP són els positius que s'han predit com a tals, i FN els falsos negatius.

Tanmateix, si basem les nostres decisions sobre els models únicament en el recall, podríem estar predint tots els casos com a positius. Per això, necessitem una mètrica que tingui en compte la

precisió en certa mesura, però que doni més pes al recall. Aquesta mètrica serà el f2-score:

$$F_2 = \frac{5TP}{5TP + 4FN + FP} = 5 \cdot \frac{precision \cdot recall}{4 \cdot precision + recall}$$

Finalment, a més d'aquestes mètriques, també utilitzarem l'accuracy, l'f1, la precisió i el recall per fer-nos una idea més detallada de la performance dels models.

4 Models proposats

Per entrenar i avaluar els nostres models, utilitzarem les particions train (80%) i test (20%) que havíem fet a la secció 2.7. Recordem que aquestes dades estan escalades.

Hem considerat tant models lineals com no lineals.

- Lineals: LDA, Regressió Logística, SVM
- No lineals: QDA, Naive Bayes, KNN, Random Forest i Gradient Boosting.

A fi de determinar els millors paràmetres pels models proposats, hem utilitzat la funció de Sklearn gridSearchCV. Donats diferents valors per cada paràmetre i les mètriques, aquesta funció prova totes les combinacions de paràmetres i, amb l'argument *refit* = 'f2', obtenim els paràmetres que proporcionen millor F2-score.

Com teníem poques dades, hem realitzat validació creuada amb 5 folds i hem promitjat els resultats per a obtenir estimacions més robustes. D'altra banda, per tal de garantir que els nostres resultats són reproduïbles, en tots els models on hi havia el paràmetre hem fixat $random\_state = 1234$.

4.1 Regressió logística

Hem utilitzat la penalty L2 ja que de moment no ens interessa produir sparsity. D'altra banda, hem fixat $max\_iter = 10000$ per tal de facilitar que l'algorisme del *solver* convergeixi.

Per la resta d'hiperparàmetres del model hem fet una cerca exhaustiva. Hem escollit valors del paràmetre C, que representa l'invers del coeficient de regularització, entre diferents ordres de magnitud, per així provar valors significativament diferents. D'altra banda, provem dos solvers: liblinear, que en principi està dissenyat per datasets grans i empra coordinate descent, i lbfgs, que funciona millor amb datasets densos i utilitza gradient descent.

Paràmetre	**Valors provats**
Solver	'lbfgs', 'liblinear'
C	0.0001, 0.001, 0.01, 0.1, 1, 10, 100

Taula 1: Paràmetres i valors per la Regressió Logística

La millor configuració és $solver = liblinear$ i $C = 0.001$, que implica una forta regularització.

4.2 Anàlisi discriminant

Els models d'anàlisi discirminant assumeixen normalitat a les dades, de manera que hem hagut de transformar les variables numèriques per tal que seguissin una distribució gaussiana. Com que algunes prenien valors negatius, hem utilitzat una transformació yeo-johnson. Tanmateix, tot i aplicar-la les dades no acaben d'ajustar-se a una normal, la qual cosa es traduirà en una performance força inferior a la resta de models, tal i com es veurà més endavant.

A continuació, amb la transformació de les dades ja feta avaluarem el model d'Anàlisi Discriminant Quadràtic (QDA) i d'Anàlisi Discriminant Lineal (LDA). També hem provat Naive Bayes per la seva simplicitat. Com tenim variables numèriques i variables categòriques convertides en dummy, hem utilitzat un GaussianNB i un BernoulliNB, respectivament. D'aquesta manera, com que considerem que les features són independents, hem fet les prediccions per cada model per separat i les hem ajuntades fent el seu producte, que posteriorment hem escalat per tal que sigui una distribució de probabilitat.

4.3 K-Nearest Neighbors

Amb KNN hem explorat diferents nombres de neighbors, paràmetre que té una implicació directa en la performance. Com menys neighbors considerem, més risc d'overfitting tindrem. També hem modificat el paràmetre *weights*. Si escollim l'opció 'uniform', tots els veïns tindran la mateixa importància, mentre que amb 'distance' tindrem més en compte els veïns més propers. Finalment, per simplificar l'espai de cerca, només hem considerat les distàncies euclidianes i manhattan (p = 1 i p = 2).

Paràmetre	**Valors provats**
n_neighbors	2, 3, 4, 5, 6, 7, 8, 9, 10
weights	'uniform', 'distance'
p	1, 2

Taula 2: Paràmetres i valors pel K Nearest Neighbors

La millor configuració obtinguda és *n_neighbors* = 10, *p* = 2, *weights* = 'distance'. És lògic que *n_neighbors* = 10 i *weights* = 'distance' estiguin a la configuració òptima, ja que estem ponderant els valors del màxim nombre de veïns avaluats, donant més pes a aquells que estan més a prop.

4.4 Random Forest

Respecte els hiperparàmetres de Random Forest, hem considerat el nombre d'arbres que intervenen en la decisió final, així com el màxim nombre de variables que hem de tenir en compte per fer el millor split cada vegada. El paràmetre *sqrt* intenta reduir la correlació entre els arbre triant un nombre baix de variables, però sense perdre massa informació. *log*2 és encara més agressiu que *sqrt* i *None* utilitza tota la informació disponible emprant totes les variables.

Quant a l'evaluació del model, tot i que l'error OOB (Out-Of-the-Bag) podria substituir l'error de cross-validation i ens estalviaríem cost computacional, hem optat per utilitzar aquest últim per ser

consistents amb la resta de models i tenir una estimació lleugerament més robusta. A més, al no tenir un gran nombre de dades el temps de computació addicional no resulta excessiu.

Paràmetre	**Valors provats**
n_estimators	1, 5, 25, 50, 75, 100, 200
max_features	'sqrt', 'log2' i None

Taula 3: Paràmetres i valors pel Random Forest

Els millors paràmetres obtinguts són *max_features*: 'log2' i *n_estimators*: 50.

4.5 Gradient Boosting

Per Gradient Boosting modificarem *n_estimators*, que representa el nombre d'arbres que utilitzarem. Degut a la naturalesa seqüencial d'aquests models, on cada arbre corregeix els errors de l'anterior, aquest paràmetre és extremadament important.

D'altra banda, també hem provat diferents valors de *max_depth*, la profunditat dels arbres. Hem observat que quan fixàvem *max_depth*= None, obteníem un recall molt proper o igual al 100%, mentre que la precisió era inferior al 60%.

També hem modificat *learning_rate*, paràmetre que disminueix la influència de cada arbre addiciona, i *loss*, la funció que de pèrdua que busquem optimitzar. El cas de *loss* = 'exponential' és equivalent a usar AdaBoost.

Paràmetre	**Valors provats**
n_estimators	1, 5, 25, 50, 75, 100, 150, 200
max_depth	3, 4, 5
learning_rate	0.05, 0.1, 0.15, 1, 2
loss	'log_loss', 'exponential'

Taula 4: Paràmetres i valors pel Gradient Boosting

Els paràmetres òptims que hem obtingut son: *learning_rate*: 0.1, *loss*: exponential, *max_depth*: 3 i *n_estimators*: 100. A priori això té sentit, atès que un *learning_rate* baix ofereix un aprenentatge suau i estable i el *n_estimators* obtingut assegura resultsts força robusts.

4.6 Support Vector Classifier

En aquest cas hem modificat el paràmetre C, paràmetre de regularització que és inversament proporcional a la regularització del model. També hem provat diferents tipus de *kernel* (lineal, rbf i polinòmic), que tindrà un efecte clau a l'hora de classificar dades no linealment separables. Finalment, *gamma*, coeficient del kernel per 'rbf' i 'poly', i 'degree', el qual només és necessari per kernels polinòmics.

Paràmetre	Valors provats
C	0.001, 0.005, 0.01, 0.05, 0.1, 1, 5, 10, 20
kernel	'lineal', 'rbf', 'poly'
gamma	'scale', 'auto', 0.1, 0.5, 1, 5
degree	2, 3, 4, 5

Taula 5: Paràmetres i valors per SVM

La millor configuració trobada és: *C*: 0.05, *gamma*: 0.5, *kernel*: rbf. És a dir, estem regularitzant força i a més utilitzem un kernel que permet tractar dades linealment no separables.

5 Anàlisi els resultats i elecció del model

Per analitzar la performance dels models, no podem basar-nos en la partició del test, ja que són dades que hem de reservar per estimar l'error de generalització un cop escollit un model. Hem fet servir els resultats amb els millors paràmetres obtinguts durant la validació creuada amb gridSearch. Aquestes són les mètriques obtingudes:

	F2	Accuracy	F1	Precision	Recall
QDA	0.801474	0.761474	0.794262	0.794113	0.809238
Naive Bayes	0.843370	0.799684	0.832822	0.816933	0.850884
LDA	0.847546	0.816322	0.843403	0.839165	0.851039
KNN	0.849989	0.806713	0.838605	0.821299	0.858041
Random forest	0.870325	0.830436	0.859251	0.841999	0.878041
Gradient boosting	0.885766	0.847189	0.873323	0.853721	0.894367
Logistic regression	0.886107	0.718359	0.799312	0.687502	0.955510
SVC	0.897520	0.732731	0.810022	0.697344	0.967510

Figura 6: Mètriques de cross validation

Observem la pitjor performance amb els models d'anàlisi discriminant (QDA, Naive Bayes i LDA). Això és coherent amb el fet que les variables, tot i haver estat transformades, no s'ajustaven a una distribució normal. D'altra banda, al disposar de variables categòriques, que a priori no són compatibles amb aquests models, fent one-hot encoding els forçàvem a interpretar-les d'una forma que no és natural.

D'altra banda, la performance de KNN no és gaire bona tampoc. Això podria ser perquè, com és un model basat en el còmput de distàncies, dona la mateixa importància a totes les variables numèriques. A més, al tenir les variables escalades les categòriques codificades amb one-hot encoding també tindran el mateix pes o més que les numèriques.

Quant a la performance dels dos models ensemble avaluats, Random forest i Gradient boosting, veiem que és molt similar, sent la del Gradient Boosting lleugerament superior. Observem que, a més de tenir una F2-score força bona, tenen un recall i una precision bastant similars. Això té sentit, utilitzen un mètode més robust per classificar que no veu les dades com a punts individuals, sinó que exploten la informació que aporta cada variable en relació amb el target per tal de millorar les prediccions. A més, al ser la combinació de diferents models són més robustos i equilibrats quant a mètriques.

Finalment, veiem que els millors valors de F2-score han estat obtinguts pels dos models lineals diferents a LDA: Regressió Logística i SVC. Tots dos s'han enfocat a maximitzar el recall a costa de la precisió. És a dir, s'han centrat a ajustar un hiperplà que contingui la majoria de casos positius, sacrificant casos negatius continguts erròniament dins el semiplà de prediccions positives (predits com a malalts). Analitzant l'impacte dels paràmetres, veiem que això succeeix amb els valors de C petits, mentre que amb valors de C més grans obtenim un recall i una precisió similars, tot i que el F2-score no sigui tan bo. Entre SVC i regressió logística ens decantem per SVC, ja que totes les mètriques són millors.

L'elecció final, per tant, estaria entre Gradient boosting i SVC. Amb Gradient boosting, malgrat tenir inferior F2-score, veiem que la resta de mètriques estan força més equilibrades, mentre que SVC clarament es decanta per maximitzar el recall. En consonància amb la idea de no predir malament els malalts, podríem quedar-nos amb SVC.

Tanmateix, cal recordar el cas d'ús dels models, i és que seran professionals de la medicina els que faran servir aquests models. El fet que Gradient boosting sigui un model molt explicable és una característica d'alt valor pels metges. D'altra banda, si ens basem en la situació d'Espanya, donada la saturació de les llistes d'espera i els recursos que normalment manquen en la sanitiat pública, un nombre excessivament elevat de falsos positius pot empitjorar aquesta situació.

Així doncs, finalment ens decantarem pel model **Gradient Boosting**, que tot i tenir un recall aproximadament un 5% inferior a SVC, és més explicable i té una precisió d'un 15% més. Aquesta major precisió disminuiria molt la càrrega dels metges, que no haurien de perdre temps en falsos positius, i potencialment permetria atendre adequadament els pacients que realment estan malalts.

6 Importància de les variables en el model

Quan entrenem un model, hi ha variables que contribueixen en major mesura a la performance. El model de Gradient Boosting permet estimar la importància d'aquestes variables. Per a cada arbre de decisió utilitzat, i per a cada node, es calcula el decrement de la mètrica gini ponderada pel nombre d'observacions a les que el node afecta. La importància general de cada variable es determina fent la mitjana d'aquestes reduccions de pèrdua per a tots els arbres.

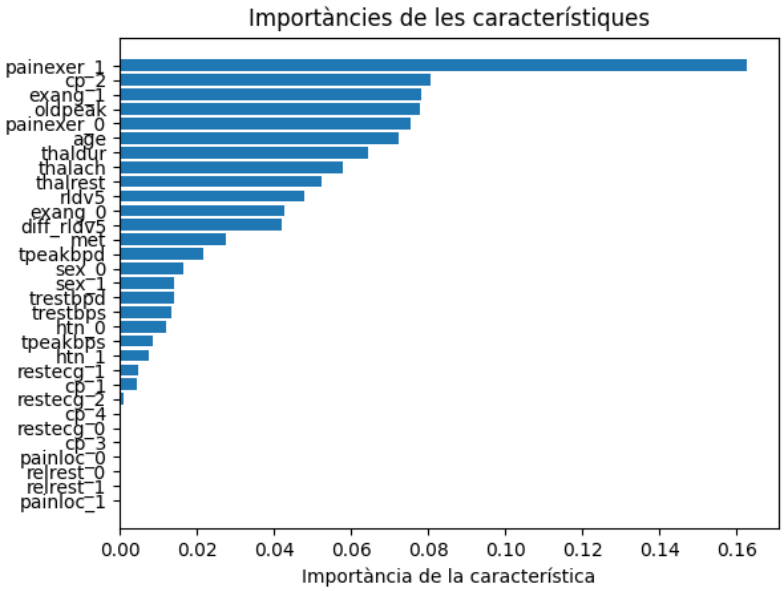

Figura 7: Importància de les variables de Gradient Boosting

Per entendre aquests resultats, cal tenir present la secció 2.5. Podem observar que el model no dona importància a variables com *relrest* o alguns valors de *painloc*. Pel que fa a aquests valors de *painloc*, probablement sigui perquè la gran majoria dels casos es concentren a la classe 1 de la variable i per tant la resta de valors proporcionalment són negligibles. Respecte a *relrest*, resulta sorprenent que no sigui gaire rellevant pel model, tenint em compte que segons si val 0 o 1 les proporcions de malalts canvien significativament. Una explicació plausible és que l'efecte d'aquesta variable queda diluït per altres variables amb més impacte a l'hora de predir el target i que transmeten informació prou similar a la que transmet *relrest* sobre l'estat del pacient.

D'altra banda, és natural que el model consideri importants per a la predicció de cardiopaties variables relacionades amb l'angina de pit com *painexer_1*, *exang_1* o *cp_2*. A més, veiem que *age* és força rellevant, fet que concorda amb que l'edat sigui un factor de risc per multitud de malalties.

7 Reducció de la dimensionalitat

Si seleccionem un subconjunt de les variables utilizades per ajustar el model, donat que no totes tenen la mateixa importància, i realitzem els entrenaments amb elles aconseguirem models amb menys tendència al sobreajustament, amb menys paràmetres i, per tant, que requereixin menys cost computacional. A més, seran més fàcilment interpretables pels metges i potencialment podríem millorar les mètriques, ja que eliminaríem el soroll dels paràmetres poc importants.

Malgrat existir funcions com RFECV, la qual elimina variables en funció de la importància que un model donat els hi dona, per propòsits de recerca hem anat eliminant nosaltres les features menys importants i a cada iteració hem ajustat de nou el Gradient Boosting amb gridSearch. La raó és que els millors paràmetres per totes les variables no seran necessàriament els millors per un subconjunt de variables. Cal destacar que aquesta metodologia és computacionalment molt intensiva.

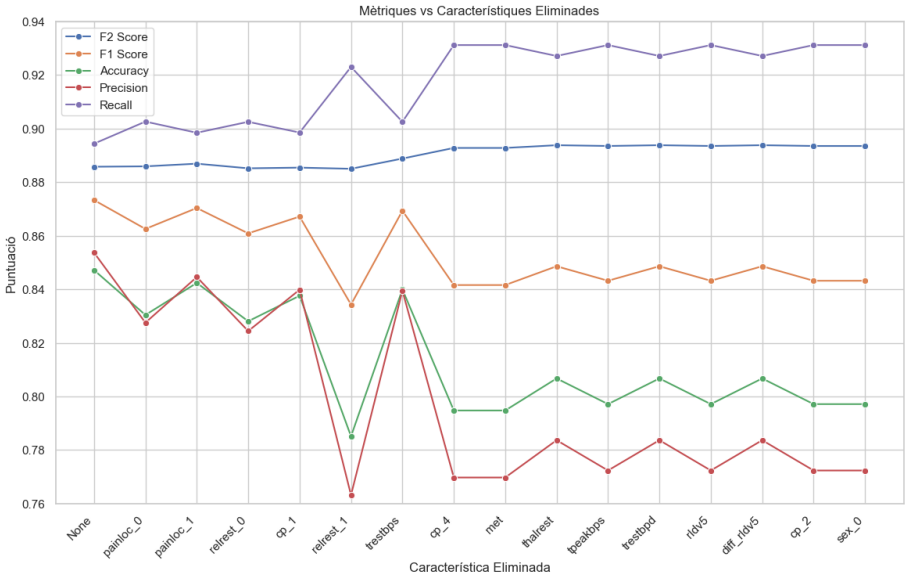

Figura 8: Evolució de les mètriques en funció de la reducció de variables[3]

En termes de precisió i accuracy, és evident que no ens interessa eliminar variables més enllà de *trestbps*. Si a més ens fixem en la F2-score, escolliríem suprimir totes les variables fins a aquest punt. El recall també millora eliminant fins a *trestbps* respecte al model amb totes les variables. Moltes d'aquestes variables ja les hem comentades a la secció anterior. L'única que no apareixia abans com a poc important és *trestbps*, la qual té una distribució similar per malalts i no malalts (secció 2.5)

Per tant, considerem que és una bona idea eliminar les variables *painloc_0, painloc_1, relrest_0, relrest_1, cp_1* i *trestbps*.

8 Estimació de l'error de generalització

Per tal de veure com es comportaria el nostre model amb dades noves, l'avaluarem amb el conjunt de test. El Gradient Boosting amb reducció de features aconsegueix mantenir una performance al test molt similar a la d'entrenament, senyal que el model no s'ha sobreajustat.

Mètrica	**Valor al train**	**Valor al test**
F2 Score	0.8888	0.8923
Accuracy	0.8399	0.8381
F1 Score	0.8693	0.8618
Precision	0.8395	0.8154
Recall	0.9025	0.9138

Taula 6: Mètriques de Gradient Boosting amb dimensionalitat reduïda

[3]Aquest gràfic s'ha d'interpretar seqüencialment d'esquerra a dreta. Cada punt de l'eix horitzontal implica haver eliminat aquella variable i totes les variables de la seva esquerra.

Per tenir una idea més clara de com es desenvolupa el model, mostrarem la matriu de confusió. Notem que la performance és l'esperada: tenim més falsos positius que falsos negatius degut a un recall superior a la precisió. Un aspecte desitjable a destacar és que el nombre de falsos positius és menor al de vertaders positius.

A la secció 2.5 hem ressaltat la manca de paritat del dataset, així com la diferència de proporció de malalts segons el sexe. Per assegurar-nos que el nostre model no estigui esbiaixat, farem els plots de la matriu de confusió segons el gènere. En el cas dels homes, la matriu de confusió és molt similar a la general. És amb la matriu de les dones que notem la diferència més notable. Veiem que la majoria de dones han estat predites com a sanes correctament. Tanmateix, la quantitat de falsos positius, falsos negatius i veritables negatius és la mateixa. Així i tot, no podem fer inferència de cap biaix del model degut a que la quantitat de dades no és significativa.

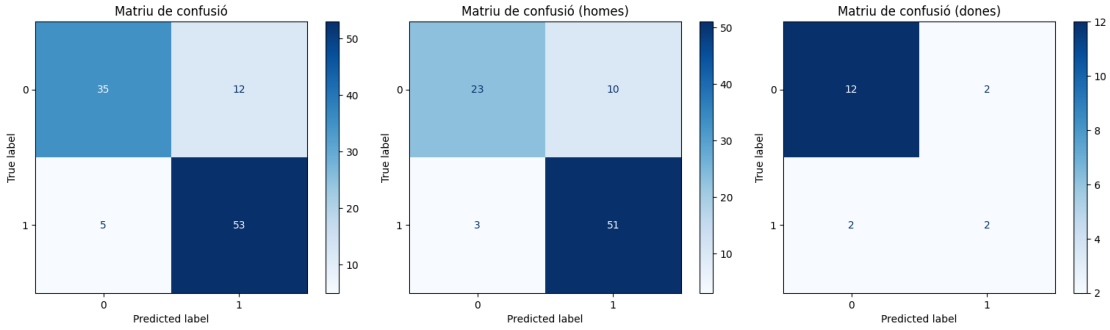

Figura 9: Matrius de confusió, d'esquerra a dreta: general, homes i dones

A continuació analitzarem la corba ROC. La corba ROC representa la ràtio de vertaders positius respecte a la ràtio de falsos positius. Per tant, com millor rendiment tingui el model, la corba ROC serà més propera a la cantonada superior esquerra del gràfic, ja que indicarà una ràtio de vertaders positius major i una ràtio de falsos positius menor.

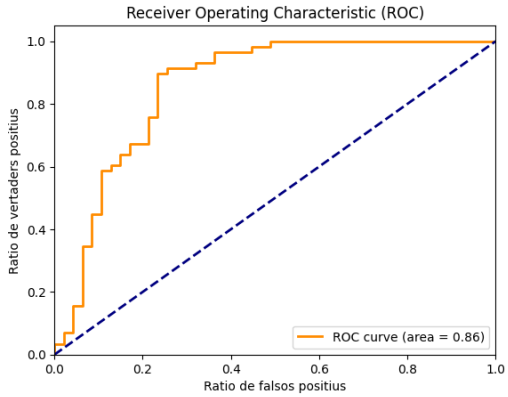

Figura 10: Corva ROC

Podem observar que la cantonada superior esquerra de la corba s'apropa bastant a la del marge.

Finalment, quant a l'area de la corba (AUC), un valor d'1 indica que s'ha aconseguit una ràtio de vertaders positius d'1 i un ratio de falsos positius de 0. De la mateixa forma, un valor de 0.5 seria el que obtindria un classificador aleatori. Per tant, quan més alt sigui aquest valor, millor distingeix el model entre casos positius i negatius.

En el nostre cas obtenim un AUC de 0.86, que és considerat molt correcte i per tant el nostre model distingeix notablement la classe positiva i la negativa.

9 Conclusions

La principal problemàtica d'aquest dataset és la poca quantitat i qualitat de les dades. D'una banda, hem hagut de prescindir de moltes variables i instàncies a causa de la presència de molts NaNs. D'altra banda, no estaven equilibrades, sobretot en el cas del sexe, on les dones estaven infrarrepresentades, de manera que no podem extreure conclusions prou fortes, per exemple, quant al biaix de gènere del model.

Així i tot, fent un correcte preprocessing hem aconseguit bons resultats. En tot problema és crucial definir les mètriques que més s'hi adeqüin. En el cas d'aquest projecte sobre l'àmbit clínic, era crucial minimitzar el nombre de malalts no detectats (falsos negatius), de manera que la mètrica F2-score era la més adequada.

Després d'una cerca exhaustiva dels paràmetres de diferents models, ens hem decantat pel Gradient Boosting. Posteriorment, hem eliminat les variables menys importants i hem obtingut un model amb F2-score=0.8923 i accuracy=0.8381 al test. El Gradient boosting final, per tant, és capaç de mantenir un compromís entre pocs falsos negatius i pocs falsos positius, adaptant-se així al context clínic i a la tasca dels professionals sanitaris. A més, té l'avantatge que és un model molt explicable.

Com a ampliació del treball, es podria considerar avaluar si els resultats diferirien molt si s'haguessin entrenat els models amb dades de dos hospitals i s'hagués reservat l'hospital restant com a test. També es podria mirar de potenciar la capacitat predictiva fent ús d'un ensemble a partir dels millors models ajustats.

Appendices

A Variables del dataset (original)

0. **id:** patient identification number
1. **ccf:** social security number (replaced with a dummy value of 0)
2. **age:** age in years
3. **sex:** sex (1 = male; 0 = female)
4. **painloc:** chest pain location (1 = substernal; 0 = otherwise)
5. **painexer:** provoked by exertion (1 = yes; 0 = no)
6. **relrest:** relieved after rest (1 = yes; 0 = no)
7. **pncaden:** sum of painloc, painexer, and relrest
8. **cp:** chest pain type
    - Value 1: typical angina
    - Value 2: atypical angina
    - Value 3: non-anginal pain
    - Value 4: asymptomatic
9. **trestbps:** resting blood pressure (in mm Hg on admission to the hospital)
10. **htn**: **Suposem que vol dir 0 no hipertens, 1 hipertens**
11. **chol:** serum cholesterol in mg/dl
12. **smoke:** smoking status (1 = yes; 0 = no)
13. **cigs:** cigarettes per day
14. **years:** number of years as a smoker
15. **fbs:** fasting blood sugar > 120 mg/dl (1 = true; 0 = false)
16. **dm:** history of diabetes (1 = yes; 0 = no)
17. **famhist:** family history of coronary artery disease (1 = yes; 0 = no)
18. **restecg:** resting electrocardiographic results
    - Value 0: normal
    - Value 1: having ST-T wave abnormality
    - Value 2: probable or definite left ventricular hypertrophy by Estes' criteria
19. **ekgmo:** month of exercise ECG reading

20. **ekgday:** day of exercise ECG reading
21. **ekgyr:** year of exercise ECG reading
22. **dig:** digitalis used during exercise ECG (1 = yes; 0 = no)
23. **prop:** beta blocker used during exercise ECG (1 = yes; 0 = no)
24. **nitr:** nitrates used during exercise ECG (1 = yes; 0 = no)
25. **pro:** calcium channel blocker used during exercise ECG (1 = yes; 0 = no)
26. **diuretic:** diuretic used during exercise ECG (1 = yes; 0 = no)
27. **proto:** exercise protocol
    - Value 1: Bruce
    - Value 2: Kottus
    - Value 3: McHenry
    - Value 4: fast Balke
    - Value 5: Balke
    - Value 6: Noughton
    - Value 7: bike 150 kpa min/min
    - Value 8: bike 125 kpa min/min
    - Value 9: bike 100 kpa min/min
    - Value 10: bike 75 kpa min/min
    - Value 11: bike 50 kpa min/min
    - Value 12: arm ergometer
28. **thaldur:** duration of exercise test in minutes
29. **thaltime:** time when ST measure depression was noted
30. **met:** mets achieved
31. **thalach:** maximum heart rate achieved
32. **thalrest:** resting heart rate
33. **tpeakbps:** peak exercise blood pressure (first of 2 parts)
34. **tpeakbpd:** peak exercise blood pressure (second of 2 parts)
35. **dummy**
36. **trestbpd:** resting diastolic blood pressure
37. **exang:** exercise induced angina (1 = yes; 0 = no)
38. **xhypo:** (1 = yes; 0 = no)

39. **oldpeak:** ST depression induced by exercise relative to rest
40. **slope:** the slope of the peak exercise ST segment
    - Value 1: upsloping
    - Value 2: flat
    - Value 3: downsloping
41. **rldv5:** height at rest
42. **rldv5e:** height at peak exercise
43. **ca:** number of major vessels (0-3) colored by fluoroscopy
44. **restckm:** irrelevant
45. **exerckm:** irrelevant
46. **restef:** rest radionuclide ejection fraction
47. **restwm:** rest wall motion abnormality
    - 0 = none
    - 1 = mild or moderate
    - 2 = moderate or severe
    - 3 = akinesis or dyskinesis
48. **exeref:** exercise radionuclide ejection fraction
49. **exerwm:** exercise wall motion
50. **thal:**
    - 3 = normal
    - 6 = fixed defect
    - 7 = reversible defect
51. **thalpul:** not used
52. **thalsev:** not used
53. **earlobe:** not used
54. **cmo:** Month of cardiac catheterization (presumed)
55. **cday:** Day of cardiac catheterization (presumed)
56. **cyr:** Year of cardiac catheterization (presumed)
57. **num:** Diagnosis of heart disease (angiographic disease status)
    - Value 0: Less than 50% diameter narrowing
    - Value 1: More than 50% diameter narrowing

58. **lmt:** Left main trunk
59. **ladprox:** Proximal left anterior descending artery
60. **laddist:** Distal left anterior descending artery
61. **diag:** Diagonal branch
62. **cxmain:** Main circumflex artery
63. **ramus:** Ramus intermedius
64. **om1:** First obtuse marginal
65. **om2:** Second obtuse marginal
66. **rcaprox:** Proximal right coronary artery
67. **rcadist:** Distal right coronary artery
68. **lvx1:** not used
69. **lvx2:** not used
70. **lvx3:** not used
71. **lvx4:** not used
72. **lvf:** not used
73. **cathef:** not used
74. **junk:** not used
75. **name:** last name of patient (replaced with a dummy string "name")

A.1 Scatterplot de les variables

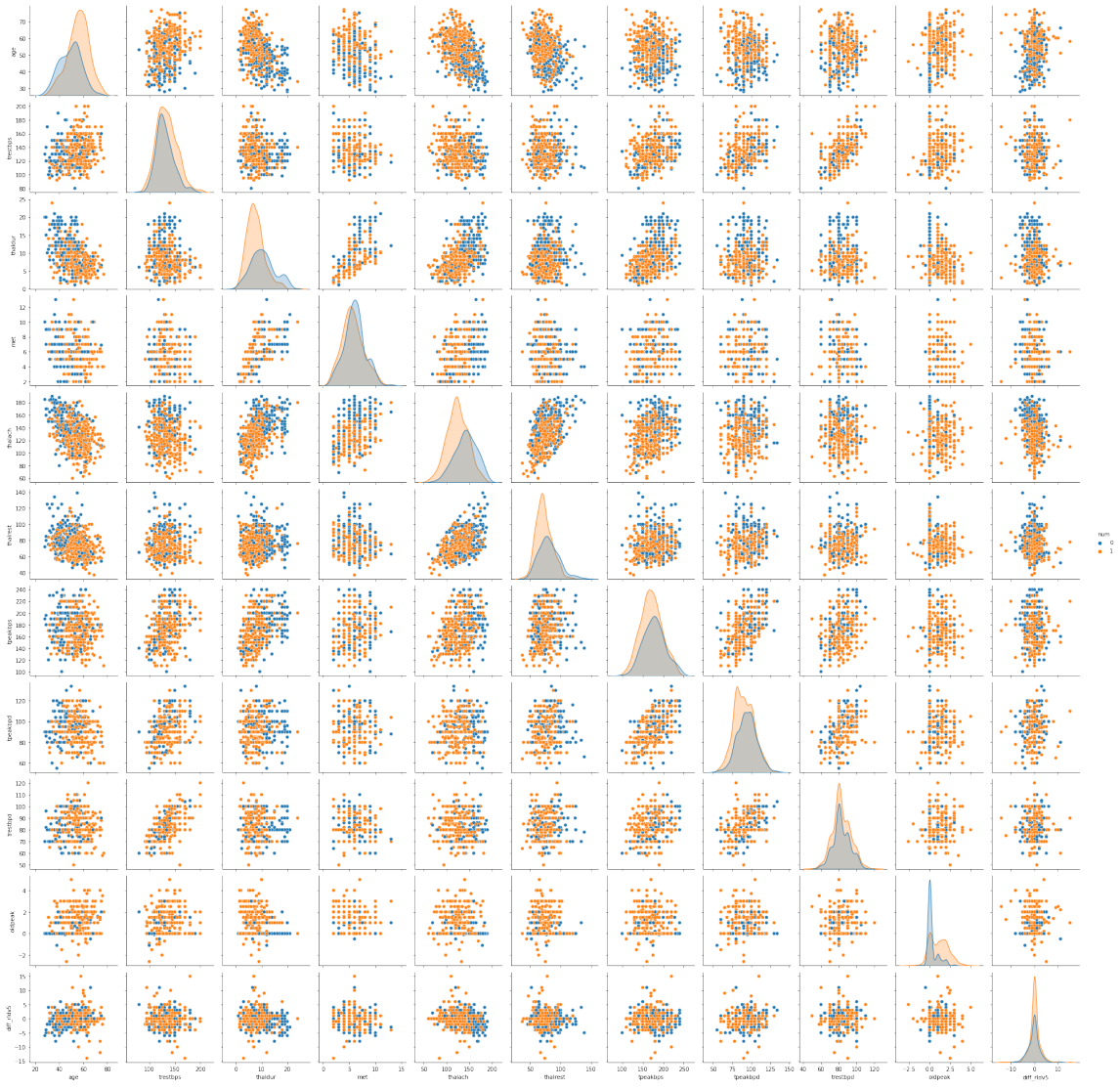

Figura 11: Scatterplot de les variables numèriques

Explicació de correlacions:

- met-thaldur: correlació entre met, que és la quantitat d'oxigen consumida per fer l'exercici i thaldur, la duració de l'exercici té sentit que estiguin positivament correlades.
- trestbpd (pressió sanguínia en repòs) i trestbps (pressió sanguínia en repòs mesurada en entrar al centre mèdic) haurien d'estar força positivament correlades, ja que en principi la diferència

que hi hauria d'haver hauria de ser poca, probablement fruit de l'alteració que ha dut el pacient a acudir a l'hospital.

- Quant a correlacions una mica menys significatives tindríem, entre d'altres, thalach i age, que és lògic que estiguin negativament correlades, ja que normalment els problemes que podrien ocasionar un major heart rate en repòs acostumen a accentuar-se amb l'edat. Exemples d'això són l'enduriment de les artèries, menor activitat física o el mateix envelliment del cor, que al perdre capacitat de bombeig de sang necessita compensar-ho augmentant els batecs per minut
- age-trestbps: En aquest gràfic podem veure que les persones, tan les malaltes com no, estàn equidistribuides en la pressió en estat de repòs de quan van entrar al centre mèdic. Això ens mostra la poca correlació entre les variables.
- age-thalach: Aquest scatter-plot mostra un correlació negativa entre les variables, i podem observar que la concentració de gent amb cardiopaties té lloc a una edat més elevada i un hearth rate màxim inferior.
- thaldur-thalach: Aquí veiem la correlació positiva que havíem comentat abans entre les dues variables. A més a més, veiem que la gent amb cardiopaties es concentra en els nivells més baixos de la variable thaldur. Thalach i thaldur també té sentit que estiguin correlades, ja que representen el màxim heart rate assolit i la duració de l'exercici.
- thalach-thalrest: thalrest i thalach també estan positivament correlats, fet que indica que a priori un major heart rate en repòs implica un major heart rate en l'exercici.