Pràctica 1: Regressió

Aprenentatge Computacional

Carlota Castro Sánchez 1568268
Gerard Vinyes Sánchez 1563545

1. Objectius

L'objectiu d'aquesta primera pràctica és analitzar el nostre dataset *Heart Attack Pediction* i realitzar prediccions mitjançant la creació de models que descriuen les nostres dades i permeten generar noves conclusions.

S'espera que arribem a aquestes conclusions mitjançant l'aplicació de regressions de diferents tipus: lineals, univariades o multivariades.

2. Anàlisi del dataset

Aquest directori conté 4 bases de dades relacionades amb el diagnòstic de malalties del cor amb 293 mostres de diferents pacients amb les seves caracerístiques respectives.
Tots els atributs tenen valors numèrics (majoritàriament de tipus int64) i les seves dades van ser recollides a les quatre ubicacions següents:

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)*

Mentre que les bases de dades consten de 76 atributs en total, només s'utilitzen 14 d'ells (seleccionats a partir de la seva rellevància dins del dataset). Aquest pas no ha estat una decisió nostra, sinó que el dataset ja venia retallat per part del creador.

Informació de cada atribut:

ÍNDEX	VARIABLE	
0	**age**	L'edat en anys
1	**sex**	1 = home, 0 = dona
2	**cp**	Tipus de dolor de pit -- Valor 1: angina típica

		-- Valor 2: angina atípica -- Valor 3: dolor no anginós -- Valor 4: asimptomàtic
3	**trestbps**	Pressió arterial en repòs (en mm Hg a l'ingrés a l'hospital)
4	**chol**	Colesterol sèric en mg/dl
5	**fbs**	Sucre en sang en dejú > 120 mg/dl (1 = cert; 0 = fals)
6	**restecg**	Resultats electrocardiogràfics en repòs -- *Valor 0*: normal -- *Valor 1*: amb anomalies de l'ona ST-T (inversions de l'ona T i/o elevació o depressió ST de> 0,05 mV) -- *Valor 2*: mostrant hipertròfia ventricular esquerra probable o definitiva segons els criteris d'Estes
7	**thalach**	Freqüència cardíaca màxima aconseguida
8	**exang**	Angina induïda per l'exercici (1 = sí; 0 = no)
9	**oldpeak**	Depressió ST induïda per l'exercici en relació amb el repòs
10	**slope**	El pendent del segment ST de l'exercici màxim) -- *Valor 1*: pendent amunt -- *Valor 2:* plana -- *Valor 3:* pendent avall
11	**ca**	Nombre de vasos principals (0-3) acolorits per fluoroscòpia
12	**thal**	3 = normal; 6 = defecte fixat; 7 = defecte reversible
13	**num**	Diagnòstic de malalties del cor (estat de malaltia angiogràfica)

		-- *Valor 0*: < 50% d'estrenyiment del diàmetre -- *Valor 1*: > 50% d'estrenyiment del diàmetre

	age	sex	cp	trestbps	chol	fbs	restecg	thalach	exang	oldpeak	slope	ca	thal	num
0	28	1	2	130	132	0	2	185	0	0.000	?	?	?	0
1	29	1	2	120	243	0	0	160	0	0.000	?	?	?	0
2	29	1	2	140	?	0	0	170	0	0.000	?	?	?	0
3	30	0	1	170	237	0	1	170	0	0.000	?	?	6	0
4	31	0	2	100	219	0	1	150	0	0.000	?	?	?	0

Només al començament decidim eliminar la variable *ca* ja que la gran majoria de valors són inexistents i això ens donarà problemes a l'hora de continuar amb l'anàlisi i efectuar els càlculs que calguin.

3. Disseny de la solució

El primer pas en el nostre procediment consisteix en fer una neteja de les dades.

Primer comencem per veure com estan representades les dades

:

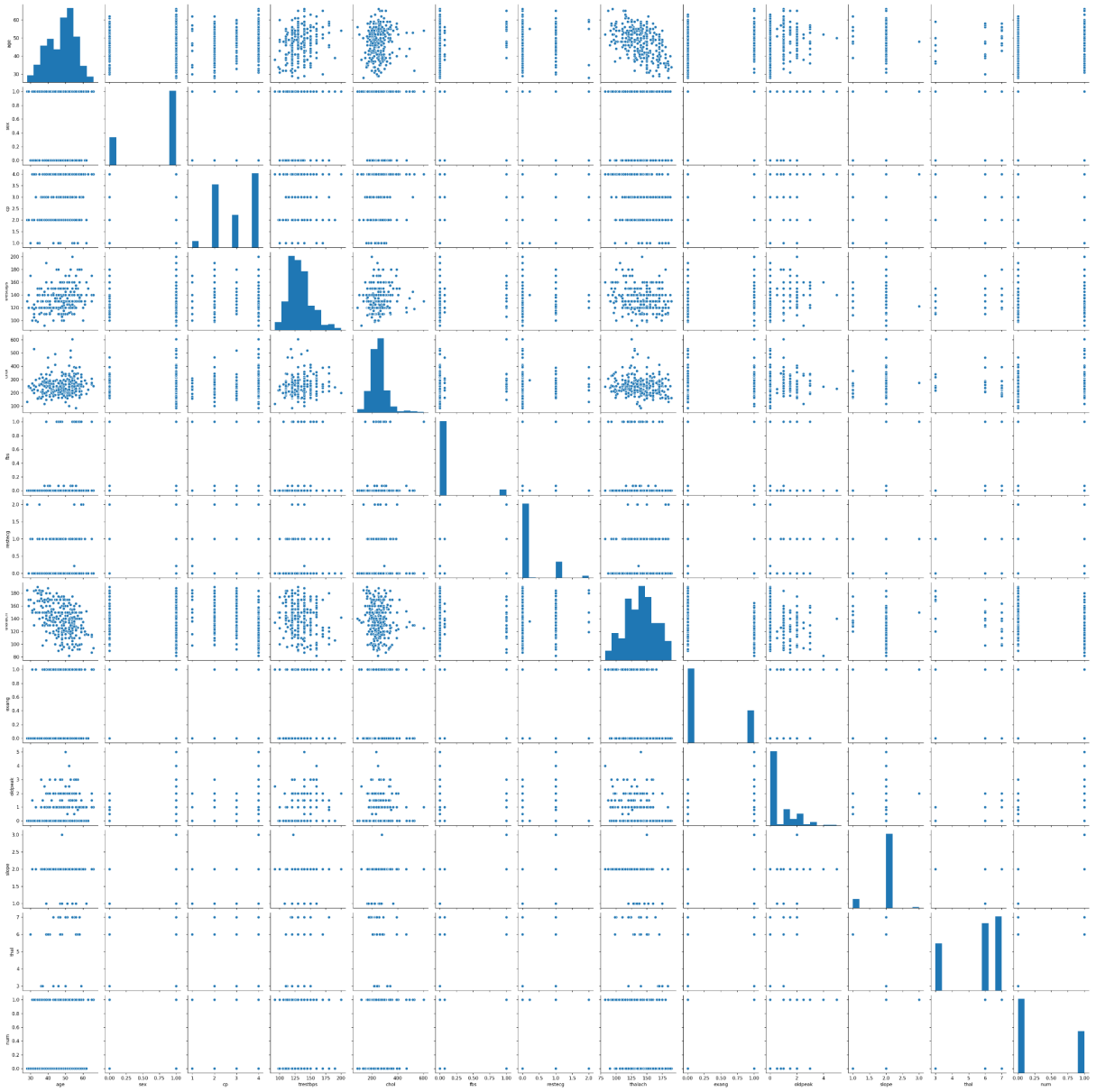
	age	sex	cp	trestbps	thalach	exang	oldpeak	num
count	293.000	293.000	293.000	293.000	293.000	293.000	293.000	293.000
mean	47.826	0.727	2.986	132.584	139.130	0.304	0.581	0.362
std	7.825	0.446	0.965	17.627	23.590	0.461	0.906	0.481
min	28.000	0.000	1.000	92.000	82.000	0.000	0.000	0.000
25%	42.000	0.000	2.000	120.000	122.000	0.000	0.000	0.000
50%	49.000	1.000	3.000	130.000	140.000	0.000	0.000	0.000
75%	54.000	1.000	4.000	140.000	155.000	1.000	1.000	1.000
max	66.000	1.000	4.000	200.000	190.000	1.000	5.000	1.000

Amb aquest primer pas podem veure que hi ha una sèrie d'atributs que ens donen problemes: *chol, fbs, restecg, slope i thal* ja que ni tan sols surten representats en la nostra taula. Això és degut a que dins d'aquests hi ha valors inexistents. Apliquem l'estrategia d'omplir aquells valors inexistents amb la mitjana de les dades en aquells atributs que tinguin més del 80% dels valors totals. En aquest cas, només *chol, fbs* i *restecg* compleixen aquesta condició.

	age	sex	cp	trestbps	chol	fbs	restecg	exang	oldpeak	num
count	293.000	293.000	293.000	293.000	293.000	293.000	293.000	293.000	293.000	293.000
mean	47.826	0.727	2.986	132.584	250.637	0.070	0.216	0.304	0.581	0.362
std	7.825	0.446	0.965	17.627	64.973	0.252	0.459	0.461	0.906	0.481
min	28.000	0.000	1.000	92.000	85.000	0.000	0.000	0.000	0.000	0.000
25%	42.000	0.000	2.000	120.000	211.000	0.000	0.000	0.000	0.000	0.000
50%	49.000	1.000	3.000	130.000	248.000	0.000	0.000	0.000	0.000	0.000
75%	54.000	1.000	4.000	140.000	277.000	0.000	0.000	1.000	1.000	1.000
max	66.000	1.000	4.000	200.000	603.000	1.000	2.000	1.000	5.000	1.000

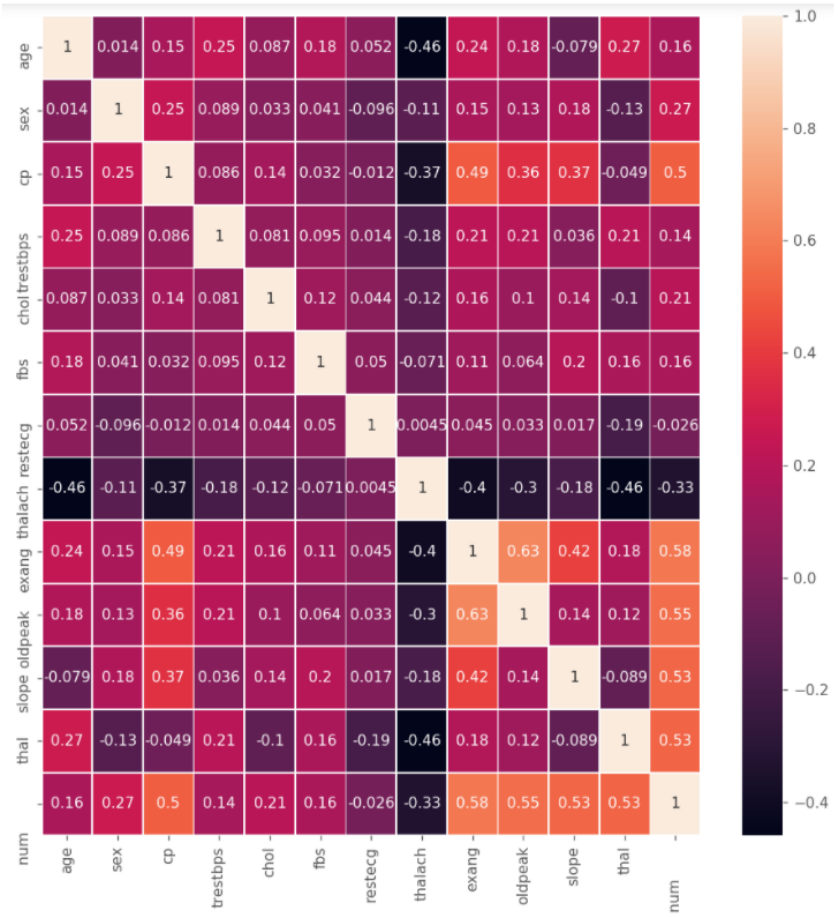
Efectivament, després d'omplir aquests valors, el nostre dataset sembla funcionar correctament.

Seguidament fem una visualització de tots els nostres atributs per veure les seves distribucions i les relacions entre ells.



Veiem que les distribucions dels atributs són binàries excepte de *age, trestbps, chol i thalach* que segueixen una distribució gaussiana (les variables *oldpeak, exang* i *cp* no podríem dir que segueixen una distribució binària, però només accepten com a màxim 4 valors).

A continuació visualitzem les correlacions dins del nostre dataset.



Eliminem *thal* i *slope* per falta d'informació ja que no ens podem basar en la seva correlació.

En general veiem que les correlacions no semblen ser gaire elevades. Si ens fixem en els atributs que estem valorant com a possibles a partir de la seva distribució (*age, trestbps, chol i thalach*), *thalach* sembla ser l'únic que manté unes correlacions significatives, ja que la resta es mantenten per sota de 0.2.

A partir d'aquest primer anàlisi decidim triar com a variable objectiu *thalach*, ja que no es tracta d'una variable binària que ens pot donar problemes a l'hora de fer

regressió lineal i manté una bona correlació amb un grup considerable d'atributs. A més a més, aquest atribut té un sentit intentar predir-ho ja que predir la màxima freqüència cardíaca que pot arribar a assolir una persona és determinant a l'hora de combatre diferents malalties i especialment a l'hora de detectar un atac al cor.

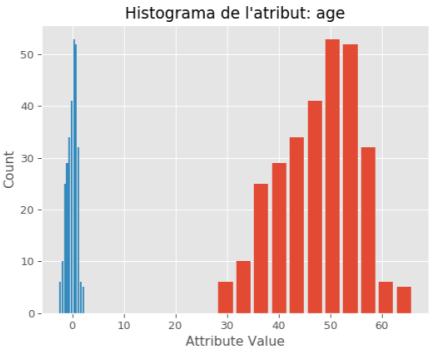
Un cop ja hem netejat el nostre dataset i hem fet un bon anàlisi comencem amb l'implementació de les regressions.

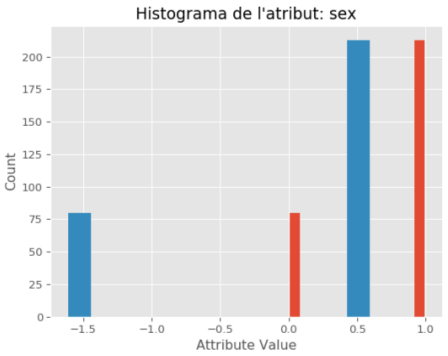
Primerament calcularem l'error quadràtic mitjà del regressor lineal que se'ns demana per a cadascun dels atributs de la nostra base de dades. A partir d'aquest càlcul obtenim que l'atribut amb l'error més baix és *age* (437.52420493542695 que tot i així és molt elevat degut a la manca de estandardització) això pot signficar rellevant a l'hora de triar-ho com a un dels atributs per a fer la nostra regressió.

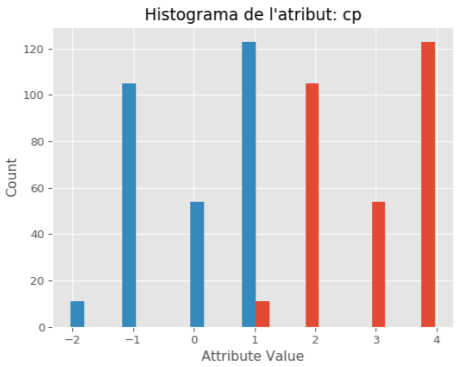
A continuació, es modificaran tots els atributs mitjançant procediments de normalització (normal, estàndard), i s'avaluarà el rendiment del regressor après. Per a això, caldrà analitzar la mitja i variança de cada variable per totes les mostres, per identificar aquells valors que tenen una distribució normal, els preferits per fer regressió, i descartar altres atributs que no són representatius per fer la regressió, i que afegeixen soroll al model.

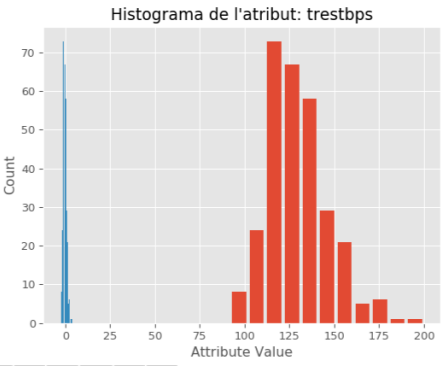
Els procediment de normalització és molt rellevant en el nostre cas ja que tractem amb dades del nostre atribut objectiu bastant grans que, per tant, ens donen errors molt elevats mitjançant el *mean squared error*. D'aquesta manera escalem les dades a un rang més petit i podem obtenir un error menor

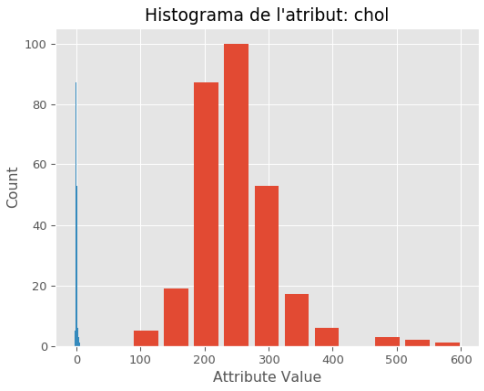
Carlota Castro Sánchez 1568268
Gerard Vinyes Sánchez 1563545

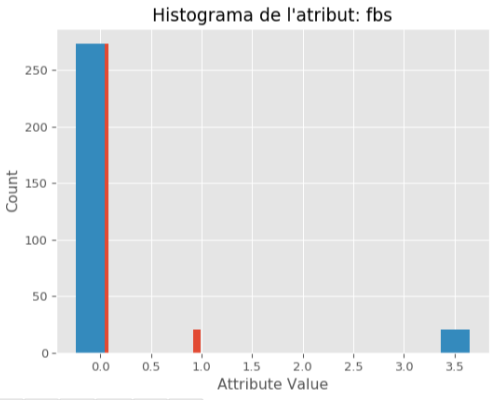


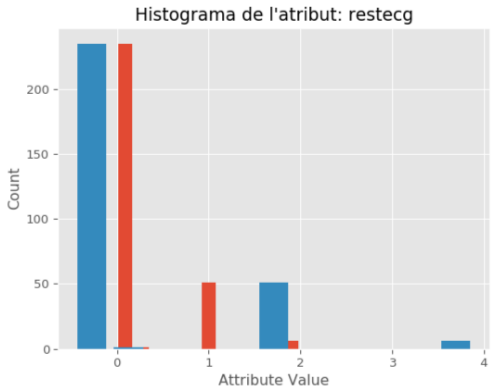


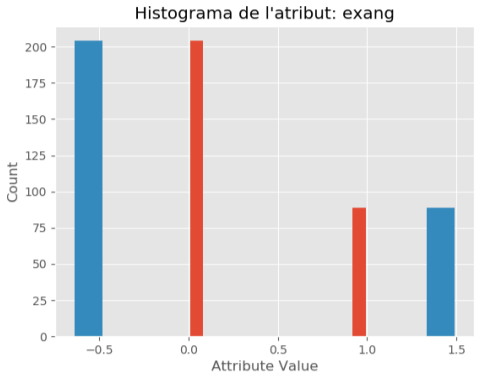


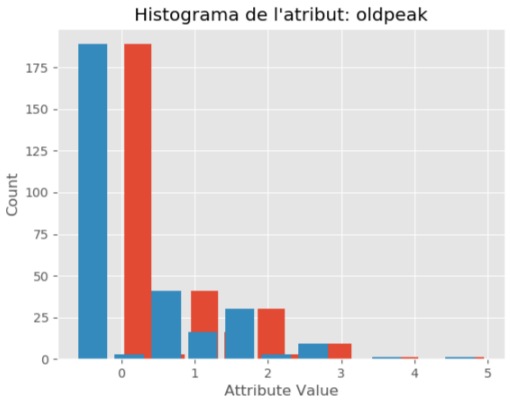


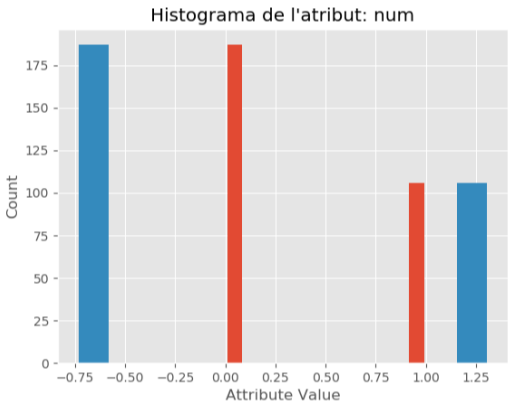






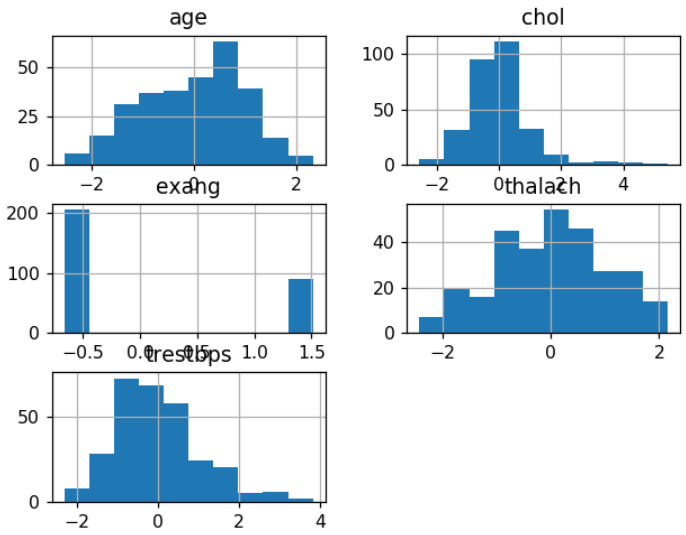






Efectivament, veiem com el nostre rang s'ha reduit de manera considerable (en blau les dades estandaritzades i en vermells les dades incials). A partir d'aquesta estandarització ja podem observar quins són els atributs que tenen una millor distribució dels valors (gaussiana). Observem que no segueixen una bona distribució els atributs *sex, cp, fbs, restecg, olpeak, exang* i *num*, per tant, els eliminem de la nostra base de dades (a excepció de *exang* que és la variable que manté una correlació més alta amb el nostre atribut objectiu i no ens ha semblar coherent eliminar-la).

No resoluta sorprenent que siguin *age, chol, exang* i *trestbps* els que semblen aportar informació més valuosa al nostre atribut *thalach* (freqüència cardíaca màxima aconseguida), ja que l'edat és un factor determinant d'aquest (quan més jove, major serà la freqüència cardíaca i a l'inversa), el mateix passa amb el colesterol on la nostra freqüència cardíaca ens determina el nostre nivell, *exang* (tipus d'angina de pit), per a freqüències cardíaques altes, és més probable que sigui de tipus zero, és a dir, que no hagi sofert una angina de pit, així com trestbps (pressió arterial en repòs)



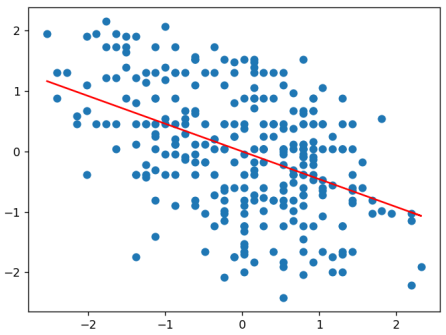
histograma dels atributs finals estandarditzats que hem seleccionat com a més òptims

Per a comprovar que la nostra estandarització ha estat correcta calculem l'error dels atributs restants.

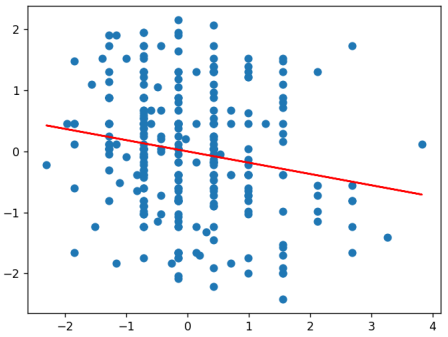
```
age [0.78624054]
trestbps [0.96249305]
chol [0.98175209]
exang [0.83555096]
```

Com podem observar el nostre error ha disminuit considerablement i *age* es manté com l'atribut que ens proporciona una error més baix.

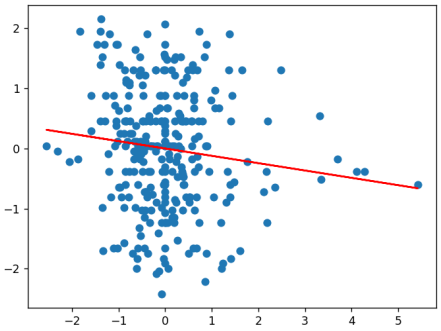
Un cop que hem carregat les dades podem entrenar un regressor lineal per a aproximar la funció que les genera i el visualitzem. També calculem el seu r2 score per determinar com de bona es la regressió que estem realitzant.



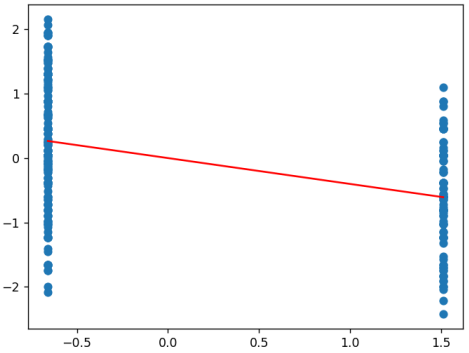
age



tresbps



chol



exang

```
R2 score:  0.21106685600197084
R2 score:  0.034210740324470024
R2 score:  0.01488574793399322
R2 score:  0.1615875591253918
```

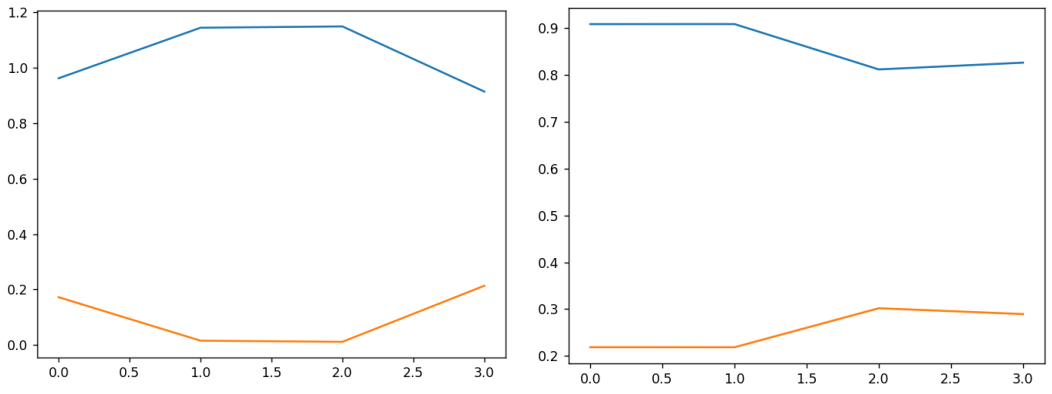
Com veiem el valor de l'r2 score no és gaire elevat, el que ens indica que la nostra regressió no és bona, però això s'explica donat que estem aplicant una regressió lineal a un conjunt de dades amb molt poca correlació entre els atributs i amb distribucions que varien significativament entre ells, tot i així *thalach* continua essent el millor atribut a predir fent servir una regressió lineal.

Finalment, per a assegurar-nos que el model s'ajusta be a dades noves, no vistes, cal avaluar-lo en un conjunt de validació (i un altre de test en situacions reals). Com que en aquest cas no en tenim, el generarem separant les dades en un 80% d'entrenament i un 20% de validació.

```
Error en atribut 0: 0.804507
R2 score en atribut 0: 0.248987

Error en atribut 1: 1.014608
R2 score en atribut 1: 0.052856

Error en atribut 2: 1.059762
R2 score en atribut 2: 0.010704

Error en atribut 3: 0.841713
R2 score en atribut 3: 0.214254
```

Observem que tant l'error com l'r2 score que ens proporciona el conjunt és molt similar al que hem calculat amb el nostre model, per tant, podem continuar amb els nostres càlculs havent fet aquesta comprovació.

Una manera de millorar lleugerament aquests resultats consisteix en utilitzar un PCA (*Principal Components Analysis*), per tal de passar a tenir el mínim nombre de noves variables i que representin totes les antigues variables de la forma més representativa possible. Per a fer-ho, hem utilitzat la funció PCA.fit_transform() de la llibreria sklearn.decomposition i l'hem avaluat amb el conjunt d'entrenament. El que busquem es que es redueixi el valor de l'mse i que augmenti el valor de l'r2. Al següent gràfic es veu en color blau el mse i en color groc el r2_score, i a l'esquerra tenim el gràfic de les dades sense haver aplicat el PCA i a la dreta s'hi ha aplicat. L'eix de les x actua només com a integers, on cada valor del 0 al 3 indica l'índex de cada una de les 4 variables utilitzades.



A partir d'aquest primer regressor podem determinar quins són els atributs més importants per a fer una bona predicció. En principi miraríem aquells atributs que tenen una correlació més alta amb el nostre atribut objectiu, essent aquests: age,

exang, cp, oldpeak i num, però en voler adaptar-nos al nostre regressor agafarem aquells que tenen una bona distribució (gaussiana): age, trestbps, chol i exang. Entenem que els millors serien aquells que formen part dels dos grups (una bona correlació i una bona distribució), és a dir, age i exang.

Un cop hem seleccionat l'atribut objectiu, els atributs del dataset amb els que volem treballar i hem estandaritzat les dades restants correctament, ara comença el procés de crear el regressor. La idea del regressor es basa en en minimitzar utilitzant el descens de gradient la funció de cost $J(w) = \frac{1}{2m}[\sum_{i=1}^{m}(f(x^i; w) - y^i)^2 + \lambda \sum_{j=1}^{n}(w_j^2)]$ respecte de la variable w, que serà un vector de mida n+1. Cada valor representarà el pes que se li dona a cada un dels atributs excepte el primer que actuarà com a intercept. En aquesta fòrmula entendrem *m* com el nombre de mostres, n com el nombre d'atributs, y el conjunt de resultats reals (i per tant cada y^i representarà el valor de l'atribut objectiu per a la mostra i-èssima) i f(x^i;w) serà una funció que calcularà el valor de la y predita per a la mostra i-èssima de x. Ja per acabar, la variable λ actuarà com a un regressor. A cada iteració del descens del gradient, es calcularan nous valors de la w utilitzant la fòrmula $w_j = w_j - \alpha[\frac{1}{m}\sum_{i=1}^{m}(f(x^i; w) - y^i)\cdot x_j^i - \frac{\lambda}{m}w_j]$. Aquest procés es repetirà fins que la diferència entre la funció de cost d'una iteració i la de la següent sigui més petita que una tolerància concreta, en el nostre cas establerta a $1x10^{-8}$ o fins que s'arribi al nombre màxim d'iteracions establert. En la fòrmula de la w_j les variables segueixen éssent el mateix que a la funció de cost, i només queda per identificar la α que és el learning rate (i que també es passarà com a paràmetre) i el x_j^i que és el valor de l'atribut j-èssim de la mostra i-èssima de x.

Per dur a terme aquest procés, hem creat una classe anomenada Regressor que té com a atributs el nombre d'atributs del dataset, el learning rate a fer servir (inicialitzat a 0.1 si no se li passa res) i el nombre màxim d'iteracions (inicialitzat a 1000 si no se li passa res). A més la classe consta de 5 mètodes:

- cost: els paràmetres que es passen són la x, la y i el lambda_value. S'encarrega de calcula la funció de cost $J(w)$. Retorna el cost, una llista amb

les prediccions per a cada mostra i una llista amb les diferències entre el valor predit i el valor real de cada mostra.

- predict: els paràmetre que es passa és la x. S'encarrega de fer una predicció del thalach per a cada mostra de x utilitzant els pesos w. Retorna el producte escalar entre els valors dels atributs d'una mostra i els valors de w més un w_0 que actuarà com a intercept.
- _update: els paràmetres que es passen són la hy, la x, la y i el lambda_value S'encarrega de calcula els nous valors de les w.
- train: els paràmetres que es passen són la x, la y, la epsilon i el lambda_value. S'encarrega de fer el bucle per anar calculant les w i els costos fins que o bé s'arribi al nombre màxim d'iteracions o bé la diferència entre la funció de cost d'una iteració i el de la següent sigui més petit que epsilon. A més guarda en la llista cost_list creada al inicialitzar la classe el cost de cada iteració.
- inference: el paràmetre que es passa és x. Retorna la llista obtinguda de la crida del mètode predict.

A continuació tenim una llegenda amb la descripció de les variables anomenades als mètodes:

- x : són el dataset excepte la columna de la variable objectiu
- y: la columna objectiu del dataset
- lambda_value: regularitzador
- hy: llista amb les diferències entre els valors predits i els reals per a cada mostra
- epsilon: diferència mínima a la que volem arribar al fer la resta entre el la funció de cost d'una iteració i la de la següent

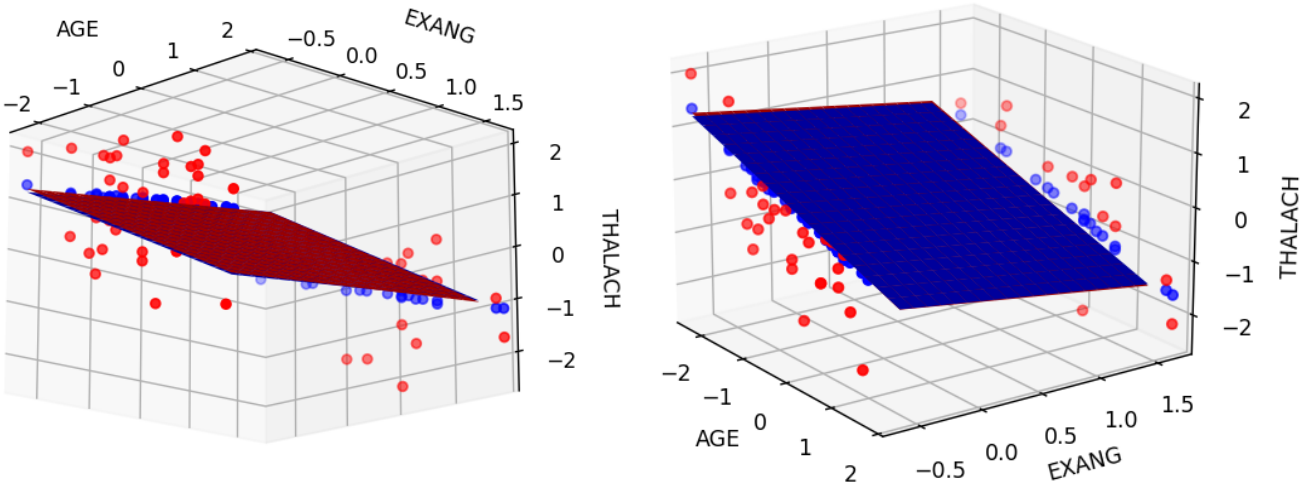
Un cop creat el regressor, ja només cal entrenar-lo amb el mètode train(x,y) i utilitzant els valors x i y del conjunt d'entrenament i a continuació fer una predicció pels valors de x del conjunt de testeig amb la funció inference(x). Un cop fet podem mirar com de precís és el regressor mirant quina proporció de prediccions coincideixen amb els valors reals.

Un cop entrenat el regressor, haurem de triar els dos atributs més importants per tal de poder fer una representació en 3D dels resultats obtinguts utilitzant aquests dos

atributs i l'atribut objectiu com a tercera coordenada. Per escollir els atributs utilitzem la llista de les w ja que aquestes actuen com els pesos dels atributs i per tant com més gran sigui el valor més important serà. És per això que en el nostre cas ens quedem amb les variables *age* i *exang* que representen els pesos més grans.

4. Conclusions

Per representar els resultats finals, ens hem basat en fer una malla de punts utilitzant els valors dels atributs *age* i *exang* per a cada una de les mostres. Per representar la coordenada de les z, hem fet un producte escalar de les w i els dos atributs i finalment se li ha sumat el valor de la w_0 que actua com a intercept. A més, en el codi hem fet dos plots superposats, un que representa els resultats predits amb el nostre regressor i un altre que ens ensenya els valor predits pel regressor si utilitzem la llibreria sklearn (representats amb color vermell i blau respectivament), a més de mostrar també els punts amb coordenades $(x, y, z) = (age, exang, thalach)$ utilitzant els valors reals i també els punts que utilitzen els valors que hem predit amb el nostre regressor per a la variable thalach (representats en vermell i blau respectivament).

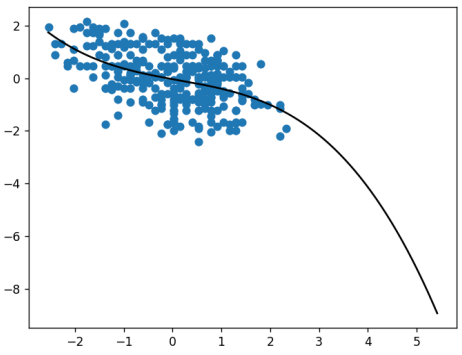


El primer que es pot observar en els gràfics és que els punts només prenen dos valors per a l'atribut *exang* i això és completament lògic ja que es tracta d'una variable binària a diferència de la variable *age* que veiem que pren valors més

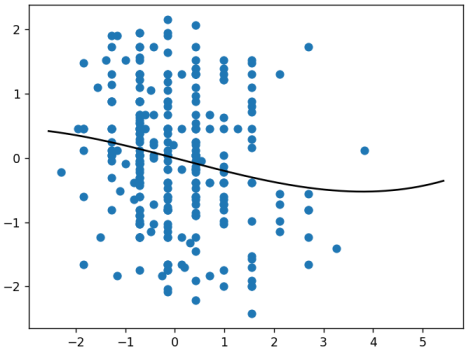
distribuïts. Com a conclusió diríem que no es tracta d'un bon regressor donat que el la funció de cost ronda el valor 30 i això és molt gran donat que les dades han estat normalitzades i es situen totes en el rang [-2, 2], a més que a simple vista ja es veu que els punts vemells estan, com a norma general, força allunyats dels plans.

Observem clarament que la nostra regressió no és gaire bona ja que si obervem el notre gràfic aquells punts que s'allunyen més de de la nostra malla seràn pitjors prediccions i trobem molts pocs punts que estiguin prop de la malla. Té sentit que aquest acabi resultant en la nostra predicció ja que comparem dues variables amb distribucions molt diferents i amb correlacions, encara que altes en comparació amb altes variables del dataset, baixes. El nostre model està més destinat a fer una classificació.

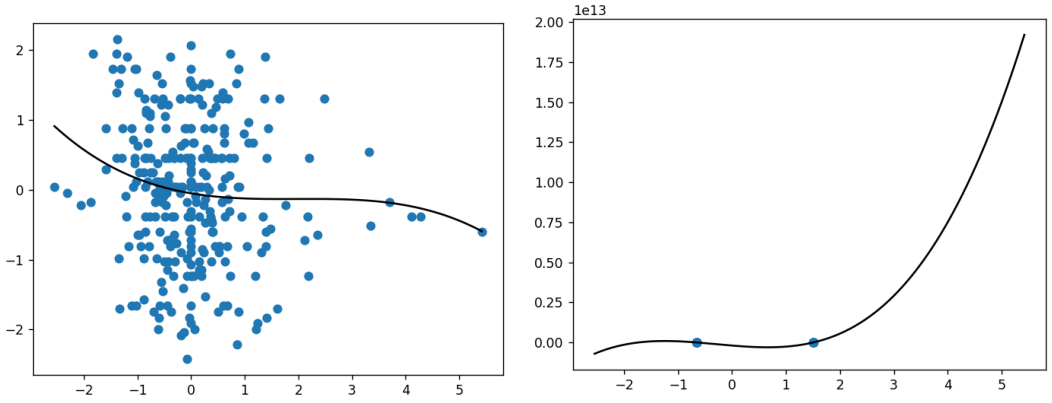
Una altra prova que hem fet ha sigut utilitzar pipeline per poder fer servir les funcions PolynomialFeatures() i LinearRegression() de la llibreria sklearn per tal de poder fer regressions polinomials i els resultats obtinguts (en aquest cas utilitzant un polinomi de grau 3) han sigut els següents:



age



trestbps



chol *exang*

Amb les gràfiques es veu que per a les variables age, trestbps i chol el regressor polinomial funciona lleugerament millor que el regressor lineal però pel cas de la variable exang es comporta exactament igual ja que es tracta d'una variable binària i per tant els resultats no milloraran.

5. Problemes trobats

Els primers problemes que vam trobar estaven relacionats directament amb el dataset amb el que vam treballar i principalment van ser tres. El primer va ser que el nostre dataset tenia moltes variables que tenien la majoria de valors nuls i que per tant els vam haver de descartar. En segon lloc la variable que estava pensada per a ser la variable objectiu (el num) era una variable binària, de manera que no ens servia com a una bona variable objectiu per a realitzar una regressió lineal (hauria sigut molt més útil per a una regressió lògica per exemple) i per això vam haver de buscar-ne i ens vam acabar quedant amb el thalach. Això però no ho vam acabar fent fins passades moltes hores dedicades a la pràctica ja que era la variable que tenia les correlacions més altes amb la resta però finalment vam haver de deixar-la de banda perquè els resultats que obteníem no eren gens bons. L'últim problema que vam tenir va ser relacionat precisament amb les correlacions, i és que eren molt baixes i això sumat a que pràcticament cap dels nostres atributs segui una distribució gaussiana va fer que haguéssim de descartar molts atributs i com a conseqüència els mean squared error, els r2 score i el regressor en general no ens donessin resultats massa bons.

A nivell de programació ens hem trobat dos problemes bàsicament. El primer va ser amb el PCA i és que ens va costar molt veure la manera d'aplicar-lo i ens van caler forces tutories per tal de poder entendre ben bé què és el que feia i com s'havia d'aplicar per tal de poder obtenir unes prediccions que fossin més properes als resultats reals i així reduïr els mse i augmentar els r2, que fins al moment d'utilitzar el PCA quedaven molt petits i molt grans respectivament. L'altre problema va ser a l'hora de fer el regressor. En un principi, vam voler fer-ho creant les diferents funcions de manera separada i sense agrupar-les en una classe. Els càlculs semblava que estaven bé però després de forces comprovacions vam acabar arribant a la conclusió de que els resultats que estaven obtinguts no acabaven de ser del tot correctes i pensem que això es podia donar perquè estàvem treballant amb llistes i numpy arrays, i potser en algunes ocasions per la pròpia manera com està definit python nosaltres ens penséssim que s'estaven modificant les nostres variables però que realment això no estigués passant i s'acabéssin fent comprovacions repetides a les diferents iteracions. Aquest problema però es va poder solucionar després de que en una turoria se'ns recomanés utilitzar la classe que se'ns proporcionava i el problema sembla que va desaparèixer.

Link al codi en Github

https://github.com/carlotacastro/APC-Pr-ctica1.git