FIB

Universitat Politècnica de Catalunya

Facultat d'Informàtica de Barcelona

Anàlisi d'una base de dades

INTRODUCCIÓ A L'APRENENTATGE AUTOMÀTIC (IAA)

Autora:

Zhiqian Zhou

28 de desembre de 2023

Índex

1 Introducció i objectius

El present treball té com a finalitat crear un model per tal de predir l'estat de supervivència dels pacients amb cirrosi hepàtica. Amb el conjunt de dades proporcionat, s'ha volgut utilitzar diferents mètodes i algoritmes apresos durant el curs per resoldre el problema.

Els objectius de la pràctica no es focalitzen tant en trobar el model més adient, sinó saber raonar a cada pas quines tècniques són les més adients per aplicar, quins resultats s'obtenen, amb quines conseqüències, com fer un preprocessament segons les dades i algoritmes que es volen emprar, entre d'altres.

L'estudi comença amb una anàlisi estadístic de les variables del dataset de manera conjunta i individual. Analitzar les característiques és imprescindible per la bona interpretació de les dades. Amb una sèrie d'estudi dels paràmetres amb la variable resposta, es comença la modelització. El treball utilitza models KNN, arbre de decisió i SVM. Presenta les motivacions d'aquesta elecció i les iteracions per trobar el model adequat. Un cop seleccionat, s'estudia les seves mètriques en les diferents particions de dades per arribar a concloure el rendiment del model que, finalment, s'especifica en un Model Card.

2 SECCIÓ 1: Anàlisis i preprocessat de dades

2.1 Recodificació de variables

Abans d'entrar a l'anàlisi i l'estudi de la base de dades, cal comprovar si la llibreria ha pogut identificar bé la tipologia de les variables. Aquesta part és molt important pel fet que els tractaments de les dades numèriques i categòriques són diferents.

Gràcies a la metadada proporcionada per la llibreria "fetch_ucirepo", podem comprovar que Çholesterol", Çopper", "Trygliceridesï "Platelets"estan malament considerants com categòriques, segurament pel fet que tenen missing data codificats amb strings. A més, s'ha pogut detectar que "Stage"ha sigut codificat com numèrica erròniament, possiblement perquè utilitza valors enters per diferenciar les modalitats. Després de fer els canvis necessaris, s'ha comprovat que la codificació estigui correcte i es poden veure els resultats a l'apartat de notebook que porta el mateix subtítol. Aquesta exploració i reidentificació és fonamental per fer l'estudi posterior i el tractament adequat de cada variable.

A més de la tipologia de les variables, cal recodificar els valors missing. En la base de dades proporcionada, els valors faltats es codifiquen com a "NaN"o "NaNN", però en el segon cas la llibreria no ho pot detectar com a valor missing. És per això que, a les dades expressades com a "NaNNçal fer la recodificació. Això permetrà l'estudi correcte de missing i la imputació.

2.2 Anàlisi estadístic de les variables

L'anàlisi de les variables conjunta i per separats és important i essencial per tenir una idea de quins són els comportaments de les variables de la base de dades per poder-les estudiar correctament.

2.2.1 Taula de dades

Per obtenir informació numèrica de les variables, cal fer la descripció de les variables quantitatives i qualitatives:

	count	mean	std	min	25%	50%	75%	max
ID	418.0	209.500000	120.810458	1.00	105.2500	209.50	313.75	418.00
N_Days	418.0	1917.782297	1104.672992	41.00	1092.7500	1730.00	2613.50	4795.00
Age	418.0	18533.351675	3815.845055	9598.00	15644.5000	18628.00	21272.50	28650.00
Bilirubin	418.0	3.220813	4.407506	0.30	0.8000	1.40	3.40	28.00
Cholesterol	284.0	369.510563	231.944545	120.00	249.5000	309.50	400.00	1775.00
Albumin	418.0	3.497440	0.424972	1.96	3.2425	3.53	3.77	4.64
Copper	310.0	97.648387	85.613920	4.00	41.2500	73.00	123.00	588.00
Alk_Phos	312.0	1982.655769	2140.388824	289.00	871.5000	1259.00	1980.00	13862.40
SGOT	312.0	122.556346	56.699525	26.35	80.6000	114.70	151.90	457.25
Tryglicerides	282.0	124.702128	65.148639	33.00	84.2500	108.00	151.00	598.00
Platelets	407.0	257.024570	98.325585	62.00	188.5000	251.00	318.00	721.00
Prothrombin	416.0	10.731731	1.022000	9.00	10.0000	10.60	11.10	18.00

Figura 1: Taula d'anàlisi de les variables numèriques

Es pot observar que a la figura 1, les variables, totes numèriques, tenen rangs molt diferents. Per això, en els futurs estudis es caldrà considerar la necessitat d'escalar-les o estandaritzar-les si l'algoritme que es pretén utilitzar es veu afectat.

Per altra banda, només observant la taula no es detecten valors anormals però sí valors missing en les variables "Cholesterol", "Copper", "Alk_Phos", "SGOT", "Tryglicerides", "Platelets" i "Prothrombin". En els casos de "Cholesterol" i "Tryglicerides" hi ha una gran quantitat de valors que falten, mentre que a "Prothrombin" i "Platelets" en falten menys.

	count	unique	top	freq
Status	418	3	C	232
Drug	312	2	D-penicillamine	158
Sex	418	2	F	374
Ascites	312	2	N	288
Hepatomegaly	312	2	Y	160
Spiders	312	2	N	222
Edema	418	3	N	354
Stage	412.0	4.0	3.0	155.0

Figura 2: Taula d'anàlisi de les variables categòriques

A la figura 2 podem observar que les variables categòriques tenen entre 2 a 4 modalitats. És possible que hi hagi un desbalanceig en les dades. Com en el cas anterior, també tenen valors missing i hi són a les variables "Drug", "Ascites", "Hepatomegaly", "Spiders" i "Stage". És possible que hi hagi una causa externa en la recol·lecció de dades que fa que la quantitat de valors que falten "Drug", "Ascites", "Hepatomegaly" i "Spiders" siguin idèntics.

2.2.2 Figura per variable

Després d'una exploració global, entrem a l'anàlisi individual de les distribucions que surten per cada variable començant per les numèriques.

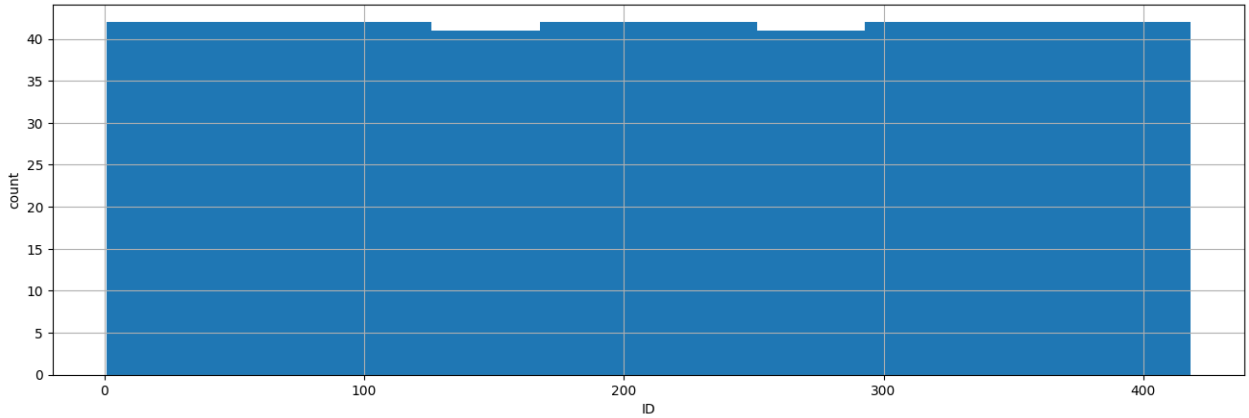

Figura 3: Histograma de ID

"ID" és una identificació única de les mostres. Segueix una distribució uniforme discreta. No aporta informació rellevant per a l'estudi de la variable resposta.

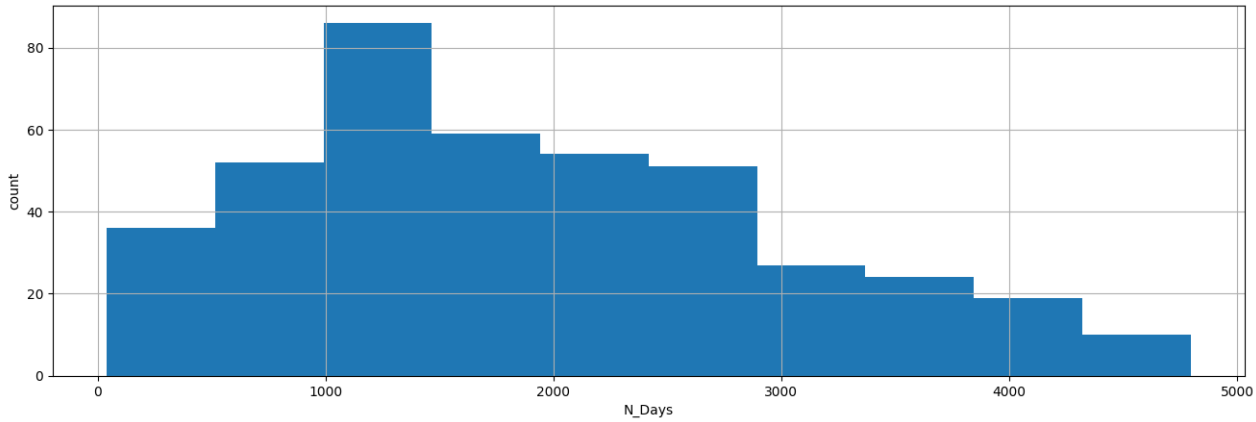

Figura 4: Histograma de N_Days

"N_Days" mostra el nombre de dies entre el registre i el primer temps de mort, trasplantament o anàlisi de l'estudi el juliol de 1986. Sembla seguir una distribució Beta on majoritàriament les mostres tenen valors al voltant de 1000 i 2000.

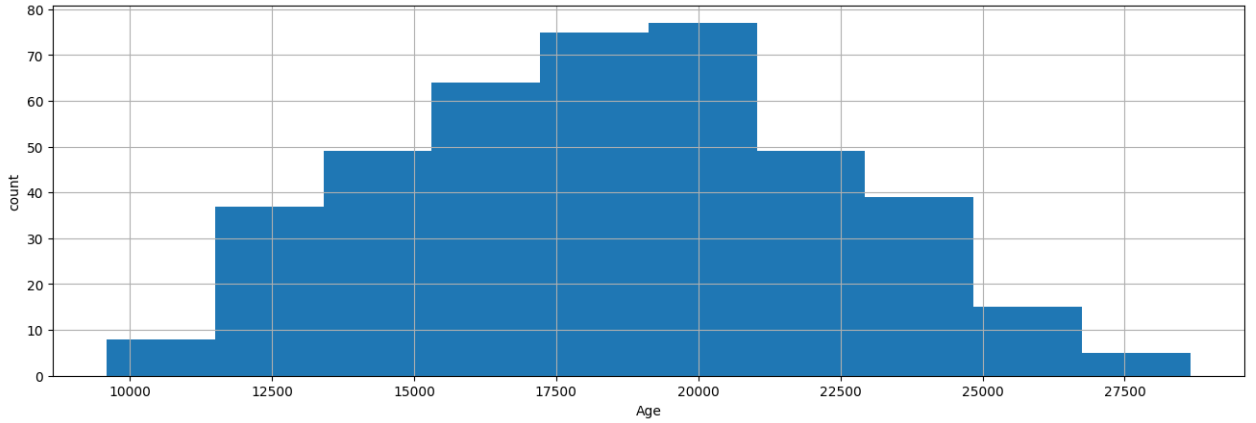

Figura 5: Histograma de Age

"Age" mostra l'edat de l'individu registrat en dies. La distribució és normal on la majoria se centra al voltant de 17500 i 200000 dies. No sembla que tingui outliers.

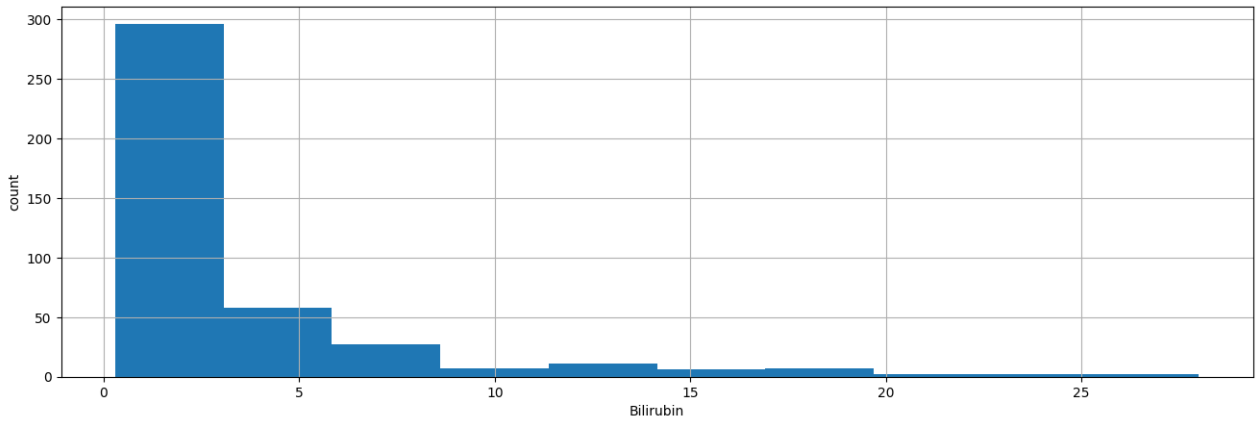

Figura 6: Histograma de Bilirubin

"Bilirubin" mostra la quantitat de bilirubina en mg/dl. Sembla ser una distribució log-normal o exponencial. Es veu que la gran part de mostres tenen valors entre 0 i 5.

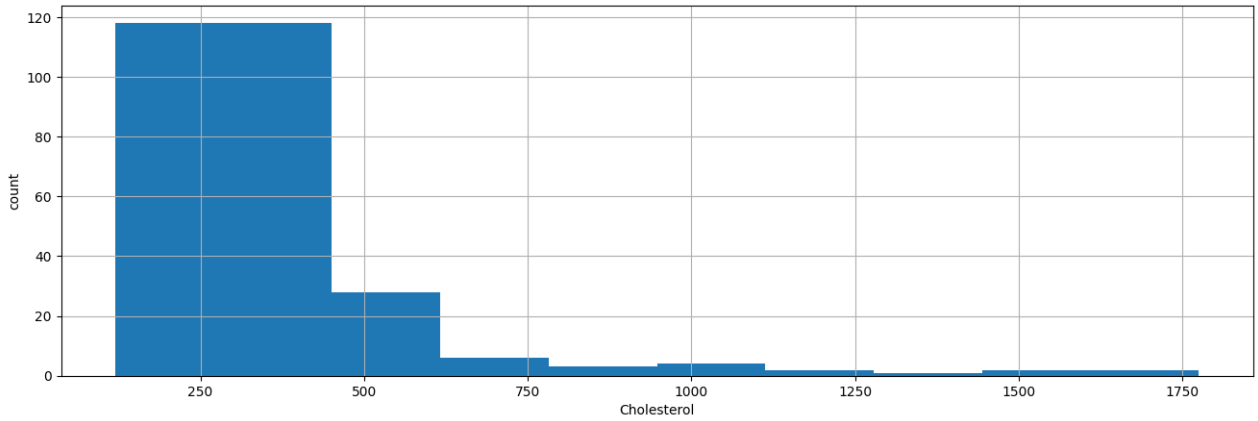

Figura 7: Histograma de Cholesterol

"Cholesterol" mostra la quantitat de Cholesterol en mg/dl. La distribució és semblant a l'anterior on els casos se centren al voltant de 200 i 500. Es pot veure que té una cua llarga possiblement d'ouliers.

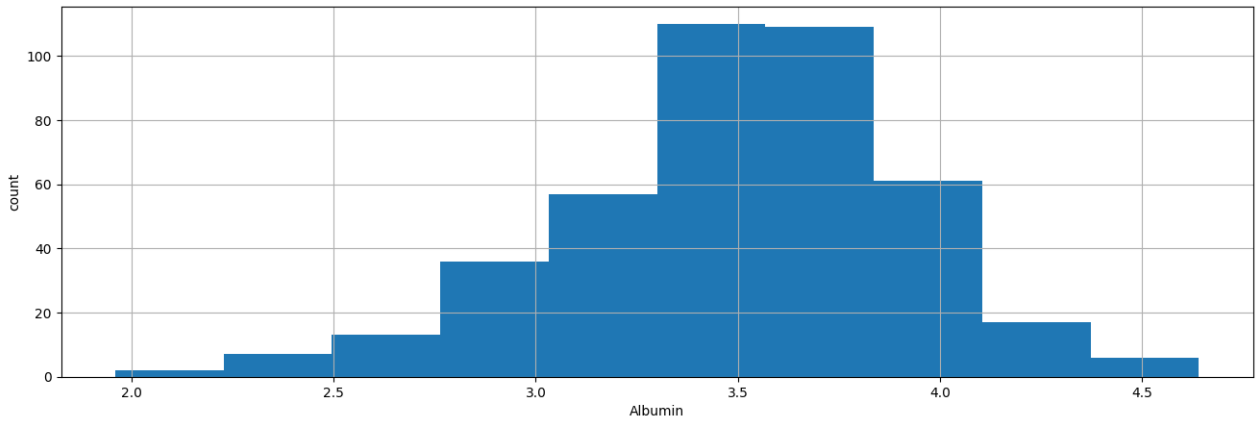

Figura 8: Histograma de Albumin

"Albumin" mostra la quantitat d'Albúmina en g/dl. La distribució que segueix és una normal centrada al voltant de 3,5 i amb presència d'ouliers a l'inici.

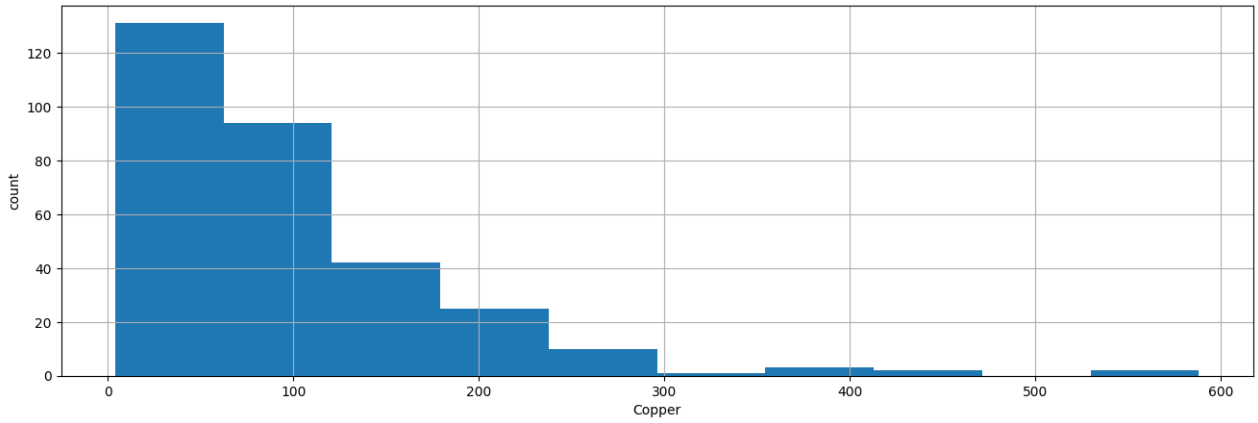

Figura 9: Histograma de Copper

"Copper" mostra el coure d'orina en ug/dia. La distribució sembla ser una exponencial i té presència d'una cua d'ouliers.

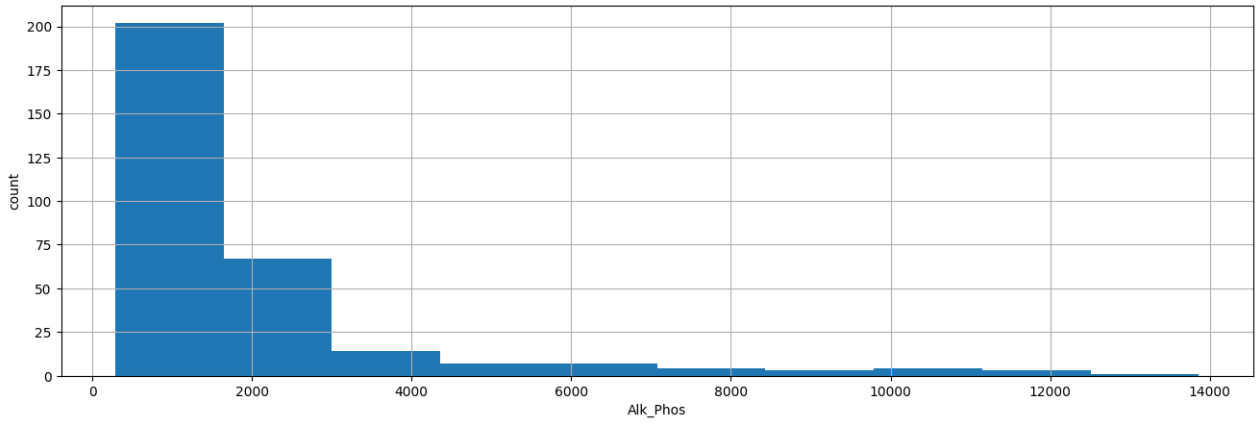

Figura 10: Histograma de Alk_Phos

"Alk_Phos" mostra la quantitat de fosfatasa alcalina en U/litre. La distribució que segueix és una log-normal amb gran quantitat de valors entre 0 i 2000. Hi ha presència d'una cua llarga d'outliers.

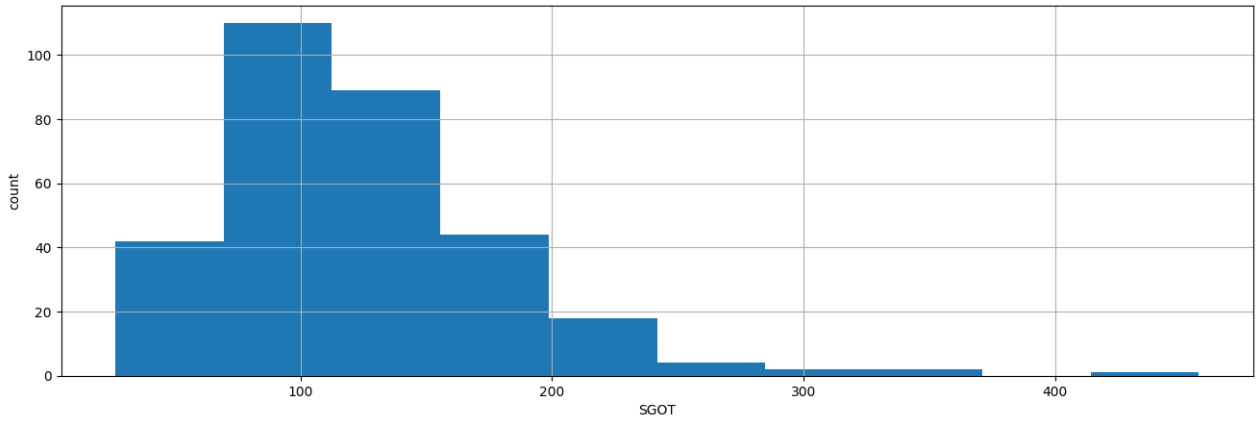

Figura 11: Histograma de SGOT

"SGOT" mostra la quantitat de SGOT en U/ml. La distribució sembla una normal centrada al valor 100 amb presència d'outliers entre 300 i 400.

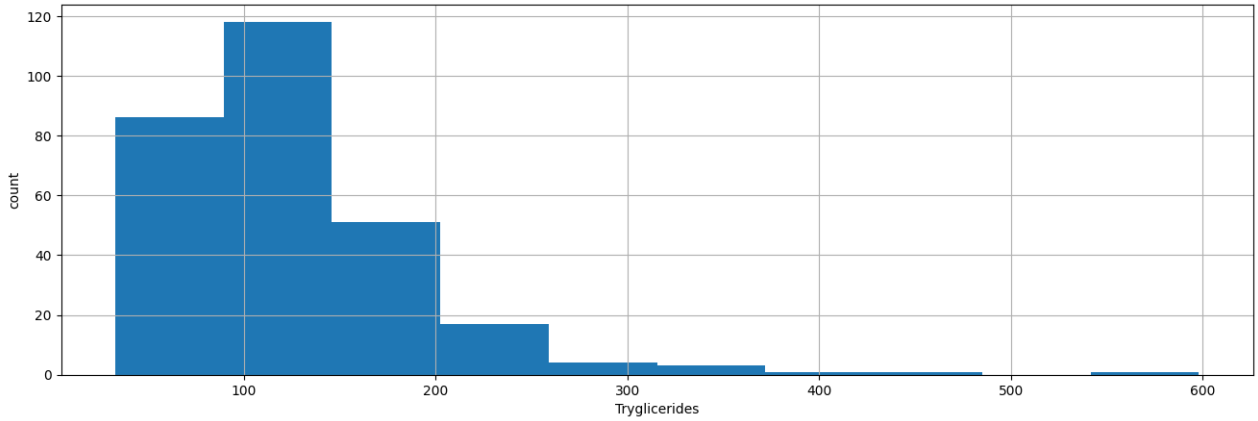

Figura 12: Histograma de Tryglicerides

"Tryglicerides" mostra la quantitat de triglicèrids. Sembla seguir una distribució beta on la majoria de les mostres en troben al voltant de 100 i hi ha presència de cua d'outliers entre 400 i 600.

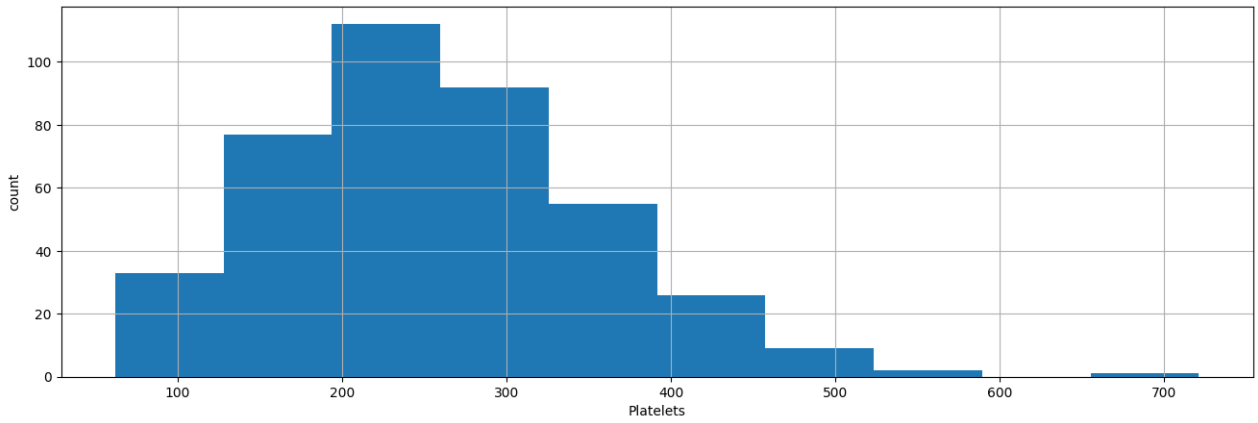

Figura 13: Histograma de Platelets

"Platelets" mostra la quantitat de plaquetes per cúbic en unitats ml/1000. Segueix la distribució normal centrat al voltant de 200 i 300 amb presència d'outliers entre 600 i 700.

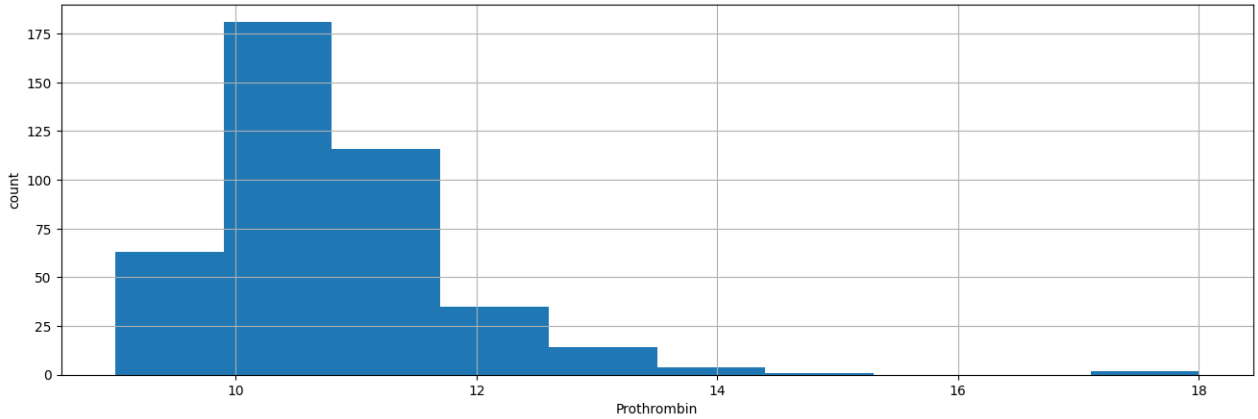

Figura 14: Histograma de Prothrombin

"Prothrombin" mostra el temps de protrombina en segons. Sembla seguir una distribució normal centrada al voltant de 10 segons i amb presència d'outliers entre 14 i 18 segons.

2.2.3 Estudi dels outliers

La base de dades és molt reduïda i petita per fer les anàlisis, per tant, en el tractament dels outliers s'ha d'intentar mantenir la majoria de les dades possibles, però sense treure gaire l'efectivitat dels models.

Sabem que es poden considerar outliers els que es troben inferiorment a la cota mínima i els que es troben superiorment a la cota màxima. La cota mínima és Q1 - threshold * IQR i la cota màxima Q3 + threshold * IQR. La tècnica per treure els outliers serà l'eliminació d'aquestes mostres. El threshold normalment és 1.5, però si es vol mantenir més dades es pot regular entre 1.5 i 3.

Els histogrames i boxplots següents mostren la diferència entre les variables originals (en la primera columna), les variables on s'han tret els outliers amb threshold 1.5 (en la segona columna) i les

variables on s'han tret els outliers amb threshold 3 (en la tercera columna). Les variables "ID", "N_Days" i "Age" no s'han vist afectades i segueixen la distribució inicial, per tant, no s'analitzen.

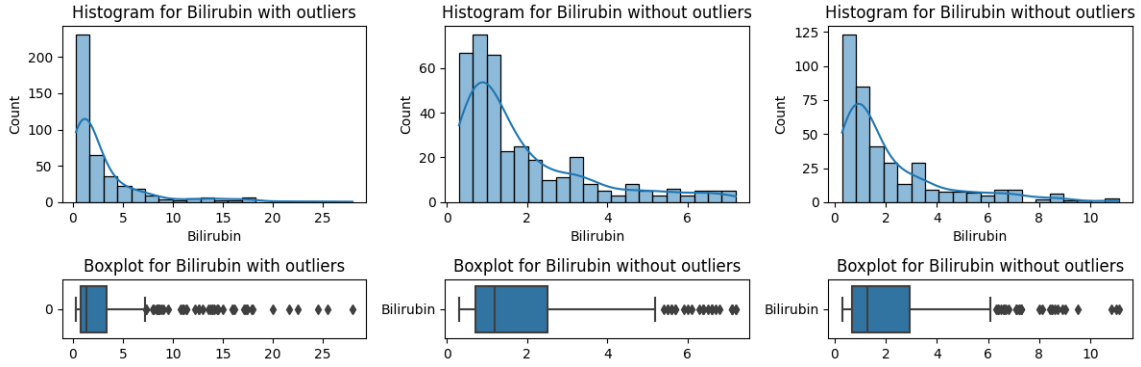

Figura 15: Comparació amb la variable Bilirubin

En la variable "Bilirubin" es pot observar que el fet de tenir threshold 1.5 o 3 no canvia la distribució que segueix. Però és en el cas de 3 que es mantenen més mostres.

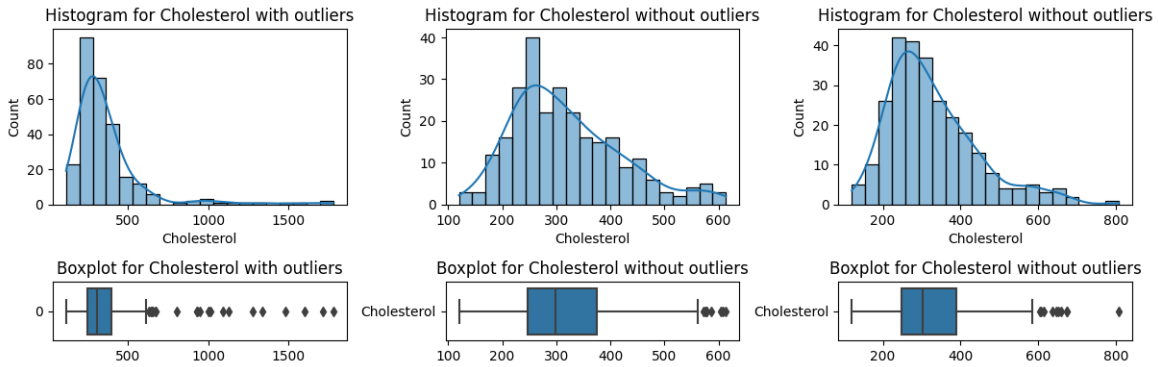

Figura 16: Comparació amb la variable Cholesterol

En "Cholesterol" es pot veure que, el fet d'eliminar la cua d'outliers fa que la distribució estigui més centrada.

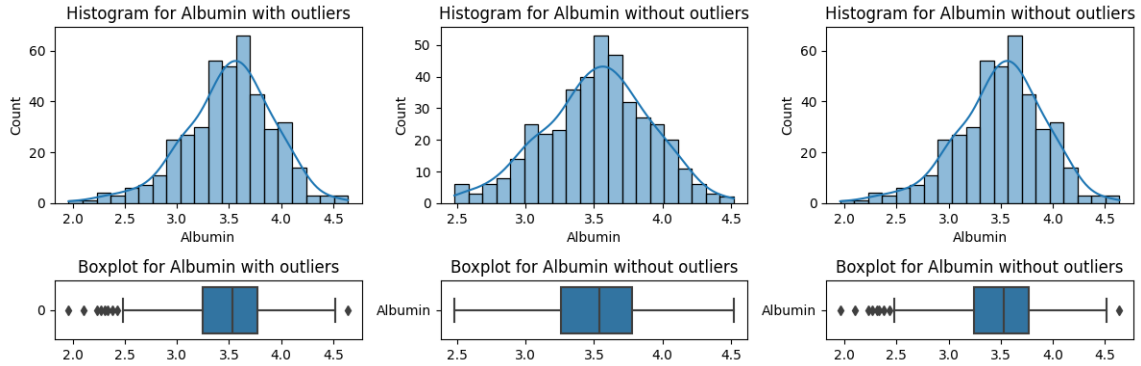

Figura 17: Comparació amb la variable Albumin

Es pot veure que a "Albumin" l'efecte de treure els outliers no és tan evident. En cas de tenir

threshold igual a 3 no elimina cap mostra. Això és perquè els outliers no estan massa lluny de la distribució.

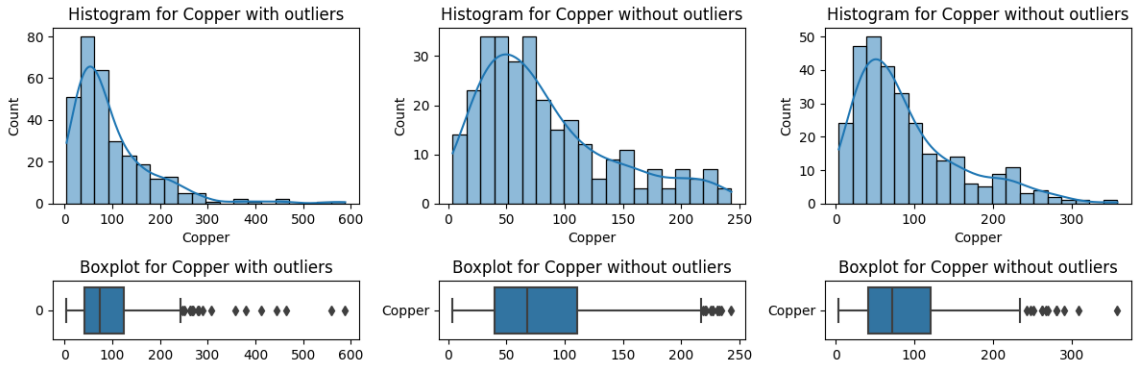

Figura 18: Comparació amb la variable Copper

En "Copper" es pot veure que el fet d'eliminar els outliers escurça el rang més o menys a la meitat. Això és degut al fet que tenia una cua llarga d'ouliers.

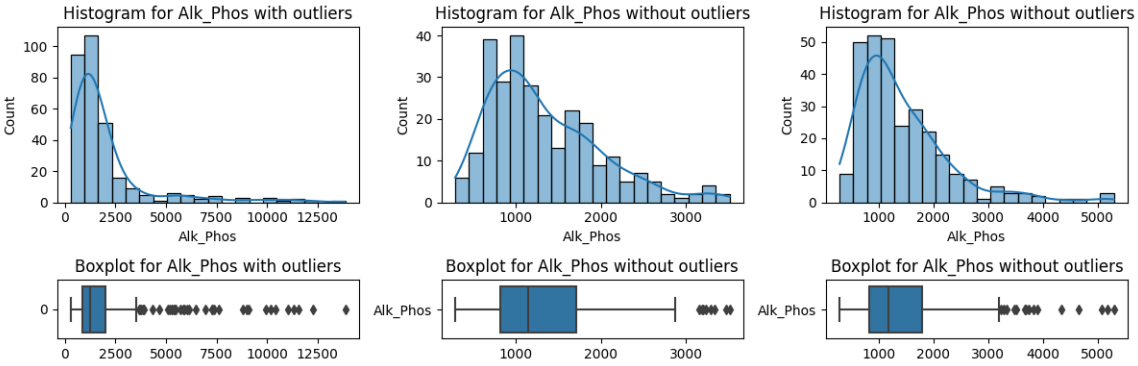

Figura 19: Comparació amb la variable Alk_Phos

"Alk_Phos" és una variable amb una cua molt llarga d'ouliers. La distribució inicial sembla ser una log-normal. En el cas de tenir threshold igual a 1.5, la distribució té una tendència a convertir-se a una normal. En aquest cas és molt important tractar els outliers per no amagar la distribució de les dades majoritàries.

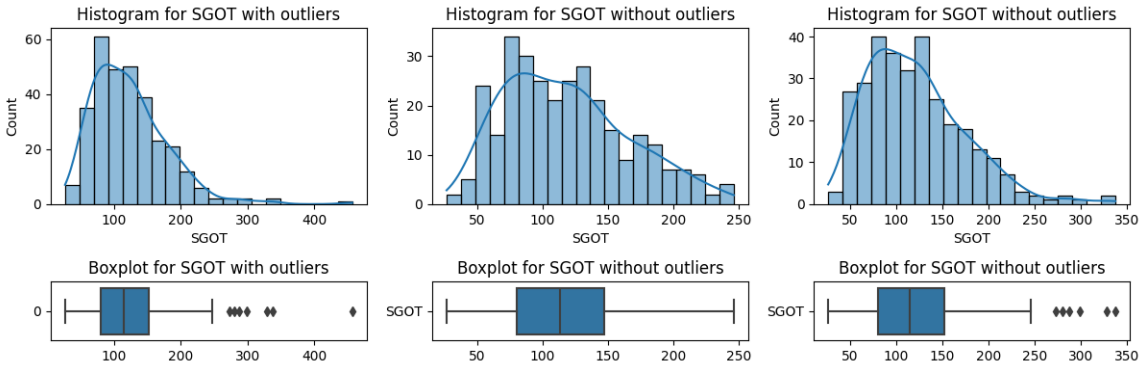

Figura 20: Comparació amb la variable SGOT

"SGOT" sembla ser també beneficiat. La distribució es manté i s'aconsegueix veure més detalladament els valors de les mostres majoritàries.

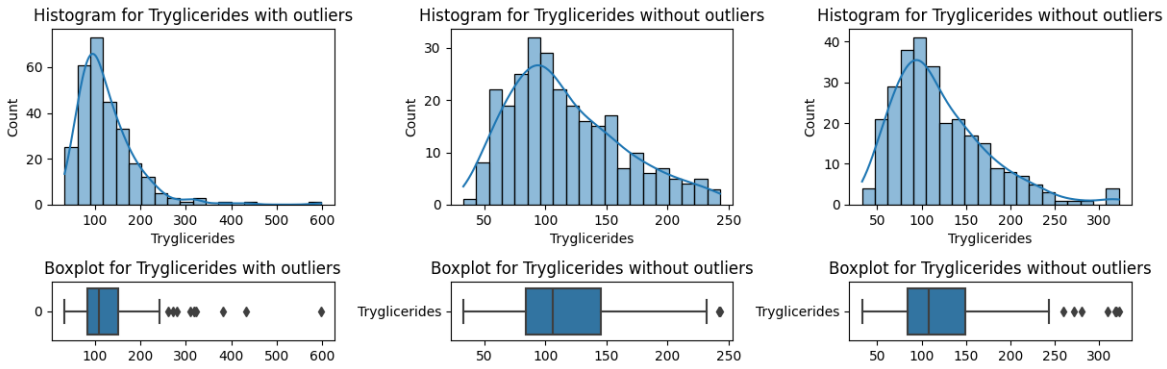

Figura 21: Comparació amb la variable Triglicerides

La variable "Triglicerides" té una cua llarga d'outliers. El fet d'eliminar-los acurça el rang de 600 fins a 250 en cas de threshold igual a 1.5 i fins a 300 en cas de threshold igual a 3.

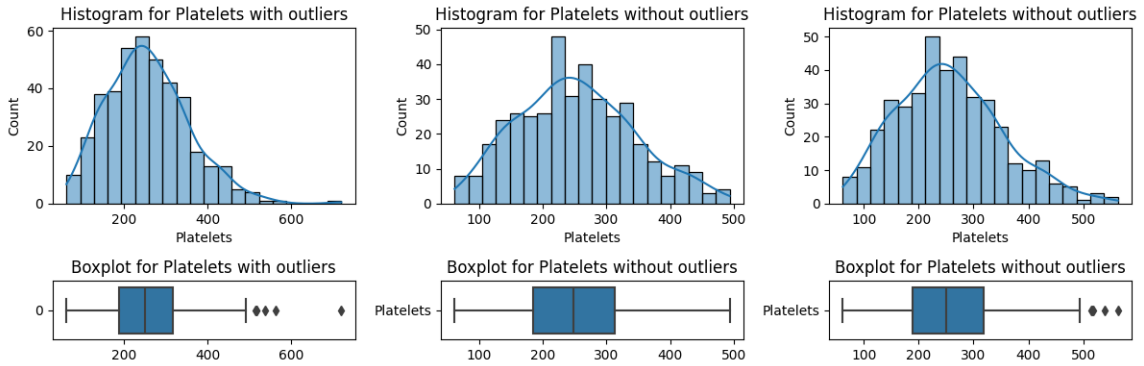

Figura 22: Comparació amb la variable Platelets

"Platelets" és una variable amb pocs outliers. L'eliminació no modifica gaire la distribució.

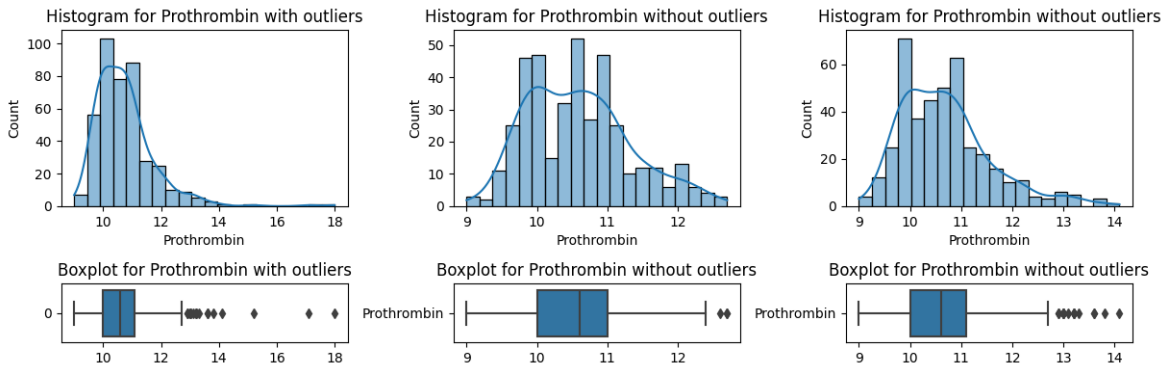

Figura 23: Comparació amb la variable Prothrombin

El fet d'eliminar els outliers de la variable "Prothrombin", es pot veure que a la distribució sembla que hi ha dos pics que abans estaven amagats.

2.2.4 Particionat del dataset

La dimensió del dataset és molt reduïda. Per fer un estudi adequat s'ha d'utilitzar el màxim de dades possibles. És per aquest fet que s'ha decidit particionar les dades en només train i test. No hi haurà una partició de validació per avaluar el model final, però es farà un cross validation per saber l'efectivitat del model.

Tal com ja s'ha dit, el que es vol és intentar maximitzar la quantitat de dades per l'entrenament del model, però sense treure la informació que ens aporta el fet de testejar-lo amb la part de test. Per tant, s'ha arribat a la conclusió de partir-lo de manera que el 80% de les dades corresponguin a train i el 20% de dades al test.

2.2.5 Estudi dels valors missing

El tractament dels valors faltants s'ha d'ajustar a cada cas depenent de la quantitat i percentatge de missing que hi hagi. És per això que cal un estudi previ abans de prendre cap decisió.

En l'apartat de recodificació ja s'ha recodificat perquè tots els valors missing estiguin ben identificats. Aleshores, es pot crear la taula on contingui informació de la quantitat exacta i el percentatge de missing que conté cada variable.

	Quantitat de missing	Percentatge de missing
ID	0	0.0
N_Days	0	0.0
Status	0	0.0
Drug	106	25.358852
Age	0	0.0
Sex	0	0.0
Ascites	106	25.358852
Hepatomegaly	106	25.358852
Spiders	106	25.358852
Edema	0	0.0
Bilirubin	0	0.0
Cholesterol	134	32.057416
Albumin	0	0.0
Copper	108	25.837321
Alk_Phos	106	25.358852
SGOT	106	25.358852
Tryglicerides	136	32.535885
Platelets	11	2.631579
Prothrombin	2	0.478469
Stage	6	1.435407

Figura 24: Taula d'estudi dels valors missing

A la taula anterior es pot observar la quantitat i percentatge de missing de cada variable. Amb aquesta informació s'obté el següent:

- Les variables "Cholesterol" i "Tryglicerides" són els que tenen major quantitat de missing, amb un percentatge superior al 30%.
- Les variables "Drug", "Ascites", "Hepatomegaly", "Spiders", "Copper", "Alk_Phos" i "SGOT" tenen un 25% de valors missing.
- Les variables "Platelets", "Prothrombin" i "Stage" tenen poca quantitat de missing.
- Les altres variables no tenen valors missing.

Vist l'estudi dels missings per cada variable, es pot arribar a una proposta de resolució:

- Eliminar les variables que tenen més de 30% de valors missing. En aquests casos, la imputació tindria un efecte escàs i és probable que afegeixi molt de soroll a les dades.

- Imputar les variables numèriques restants amb el mètode de KNN. Altres mètodes com imputar per "mean" o "median" no serien efectius perquè imputar uns 25% de dades causaria una acumulació de dades en un valor i els estudis posteriors es veurien afectats per aquesta acumulació. En canvi, KNN no té aquest problema perquè imputa tenint en compte els K veïns i la distribució de les variables no es veuria gaire diferent.
- Imputar les variables categòriques restants amb una nova modalitat "missing".

Per comprovar l'efectivitat de la imputació amb KNN, s'ha imputat totes les variables numèriques amb aquest algoritme i s'ha dibuixat els histogrames corresponents, comparant les variables originals i les imputades. A la secció de "estudi dels valors missing" del notebook es pot veure que la distribució de les variables no estan gaire afectades, cosa que vol dir que KNN és una bona opció. Però cal tenir en compte que les variables "Cholesterol" i "Trygliceride", els que tenen més de 30% de missing. A la figura 25 i 26 es poden observar la distribució d'aquestes variables originals i imputats amb KNN. Es veu que la distribució després d'imputar-los és semblant, però la freqüència o count màxim és significativament més alt. Amb tants missing, la imputació probablement afegeix soroll i no sigui fiable. Es pot considerar treure'ls.

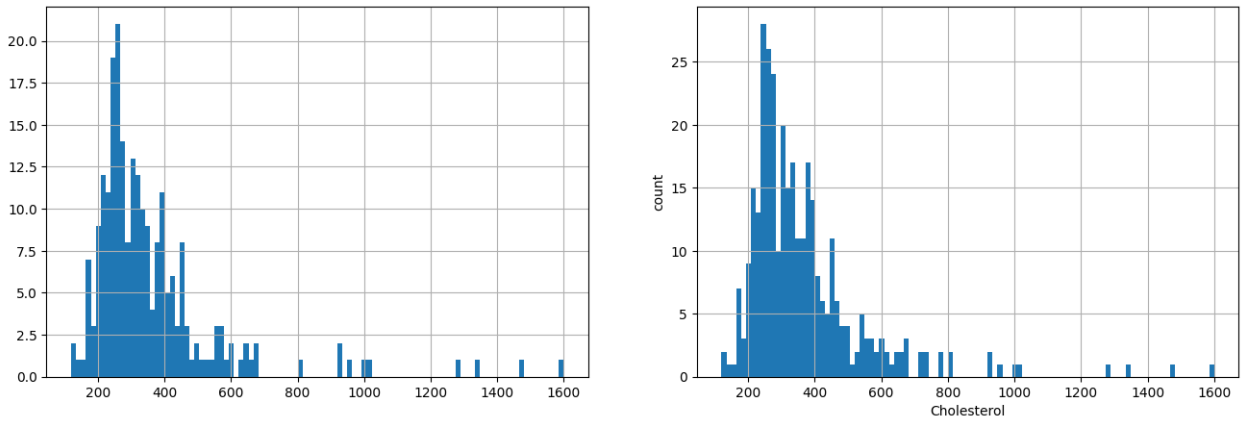

Figura 25: Comparació de la distribució original i imputat amb KNN de la variable Cholesterol

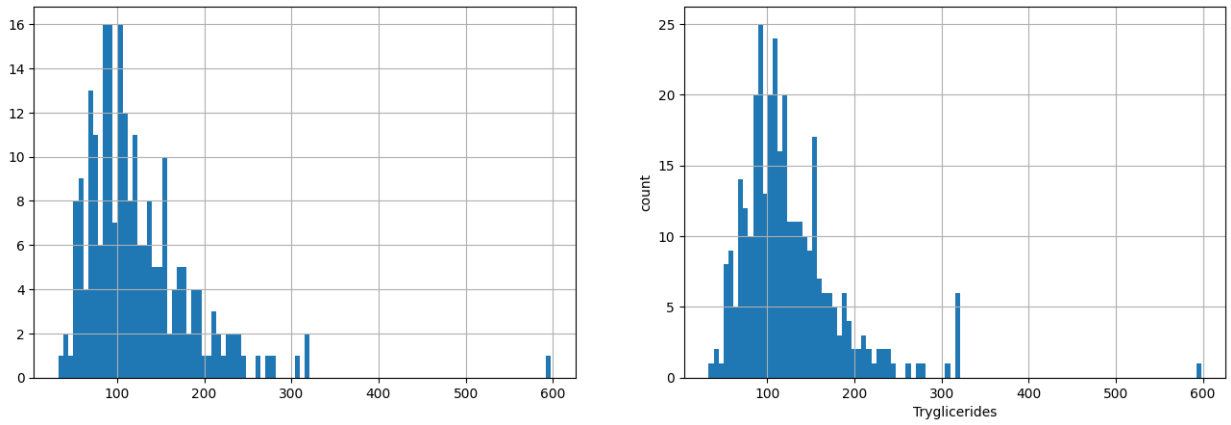

Figura 26: Comparació de la distribució original i imputat amb KNN de la variable Trygliceride

En el mateix apartat del notebook es pot veure els efectes d'imputar les variables categòriques afegint una modalitat "missing". S'observa que a les variables "Drug", "Ascites", "Hepatomegaly", "Spiders" i "Stage" hi ha dades amb valor "missing".

2.2.6 Estudi de balanceig de classes

Tenir una base de dades balancejada o no és important per decidir si cal realitzar-ne tractaments als models perquè tingui en compte la classe minoritària. Al següent histograma es pot veure la freqüència de dades per cada classe de la variable resposta.

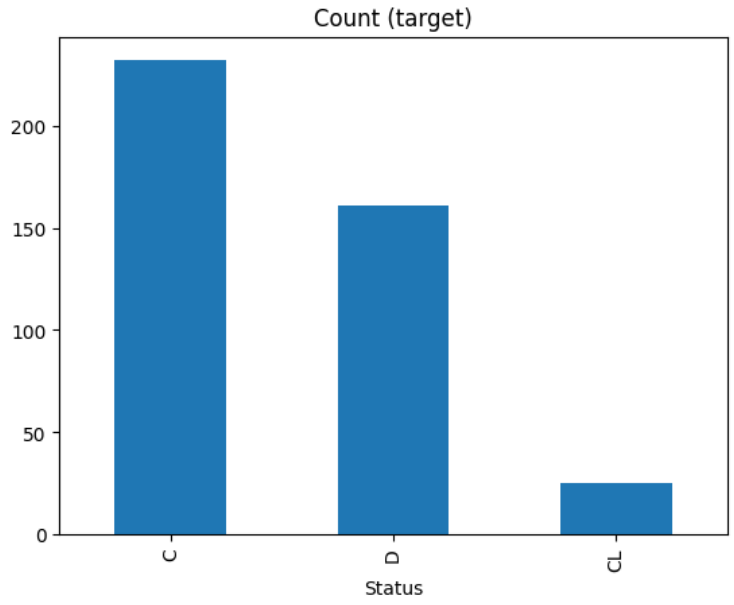

Figura 27: Histograma amb freqüència per classe

A l'histograma s'observa que, per a la variable resposta, hi ha gran part de mostres que són de la classe censurat (C) i moltes són de la classe mort (D). La classe censurat per trasplantament hepàtic (CL) només té una quantitat molt reduïda de mostres. Es pot veure que la base de dades està desbalancejada.

La proposta davant d'aquest fet és utilitzar class_weight en els models, les raons de la tria són les següents:

- Millora la capacitat dels models per generalitzar les classes minoritàries.
- Ajusta automàticament els pesos durant la modelització per donar més pes a les classes minoritàries.
- No cal introduir-ne dades sintètiques
- El fet d'utilitzar SMOTETomek requereix un OneHotEncoding primer de les variables categòriques. Aleshores, SMOTETomek pot crear dades intermèdies que no s'ajusten a les modalitats de les variables i serien soroll pel model.

3 SECCIÓ 2: Preparació de variables

A la secció 2 ja es té en compte els canvis duts a terme en la secció 1. Les anàlisis es faran al particionat de dades corresponent. No serà fins a la creació dels models que es considerarà l'eliminació dels outliers, ja que la quantitat eliminada dependrà del rendiment del model.

3.0.1 Normalització de variables

La normalització s'ha dut a terme sobre les variables numèriques ja imputades en l'apartat anterior per poder analitzar i extreure conclusions. A l'hora de crear els models també s'haurà de considerar els canvis en els outliers.

Per fer la normalització de les variables, s'analitza la tècnica StandardScaler. Aquesta tècnica és útil en cas que les dades es distribueixin de manera aproximada a una distribució normal, és per això que s'utilitza pel dataframe que es disposa. Després d'aplicar StandardScaler, cada variable numèrica tindrà una mitjana de 0 i una desviació estàndard d'1.

	ID	N_Days	Age	Bilirubin	Cholesterol	Albumin	Copper	Alk_Phos	SGOT	Tryglicerides	Platelets	Prothrombin
0	0.938726	-1.650298	-0.490454	0.426822	0.654555	-1.342742	0.851944	-0.122870	1.154108	0.040343	0.038036	-0.322439
1	-0.231767	0.576744	1.857645	-0.599282	-0.370536	1.379023	-0.590659	-0.635768	-0.899462	0.000189	0.701831	-0.525381
2	-1.310945	2.196810	-0.183784	-0.296115	-0.584023	0.479657	2.269674	4.357317	1.216762	0.900781	-0.431727	0.286388
3	-0.597028	1.075833	-0.849084	-0.575962	1.104466	0.408654	-0.789639	-0.282777	0.079343	-0.355459	1.743479	-0.018026
4	1.021740	-0.403891	1.172526	-0.575962	-0.119997	-0.230369	-0.487024	-0.408137	-0.307288	-0.103064	1.171594	-0.626852
...	...	...	...	...	...	...	...	...	...	...	...	...
329	0.091986	-0.993325	-1.461084	1.919337	1.263258	-0.443376	-0.416552	-0.058566	0.539956	1.451461	0.426101	0.083445
330	0.847411	-0.674925	0.994684	-0.645923	-0.652833	0.195647	-0.926437	-0.568043	-0.591148	-0.217789	0.323979	-0.322439
331	-0.555521	0.160983	-0.415996	-0.482679	-0.218801	-0.443376	0.752454	-0.330492	-0.525213	0.143595	-1.167008	0.895214
332	-1.169822	1.815257	-1.057726	-0.459359	-0.599902	0.242982	-0.304626	-0.605497	0.338438	-0.303833	1.774116	-0.728324
333	0.905520	0.808308	0.292156	-0.575962	-0.178221	1.260685	-0.652840	-0.306378	0.060150	0.091969	-0.309180	-0.931266

Figura 28: Dataframe de partició train després de la normalització

Es pot observar que la normalització de les variables numèriques s'ha realitzat correctament. També es pot observar la comparació dels histogrames de cada variable abans i després de ser normalitzada:

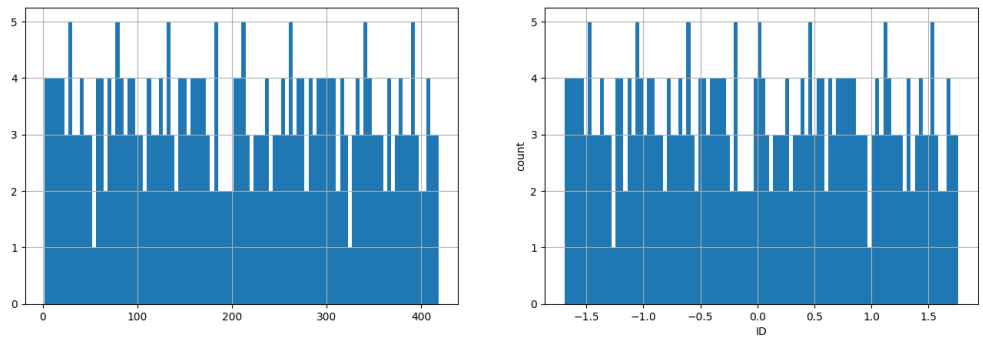

Figura 29: Comparació de la distribució de ID abans i després de la normalització

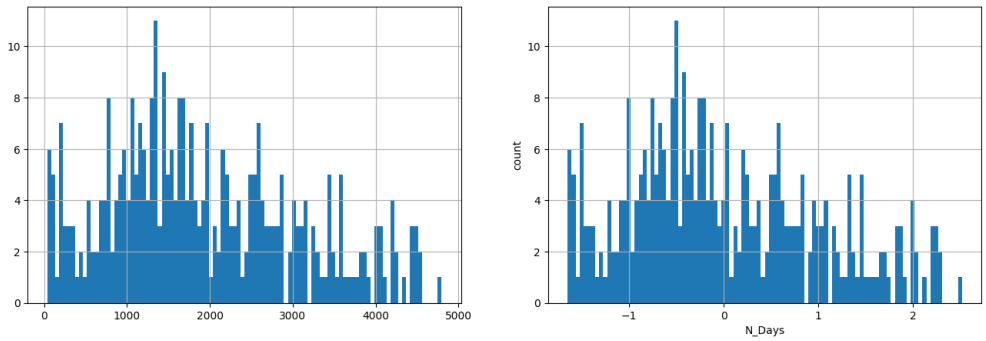

Figura 30: Comparació de la distribució de N_Days abans i després de la normalització

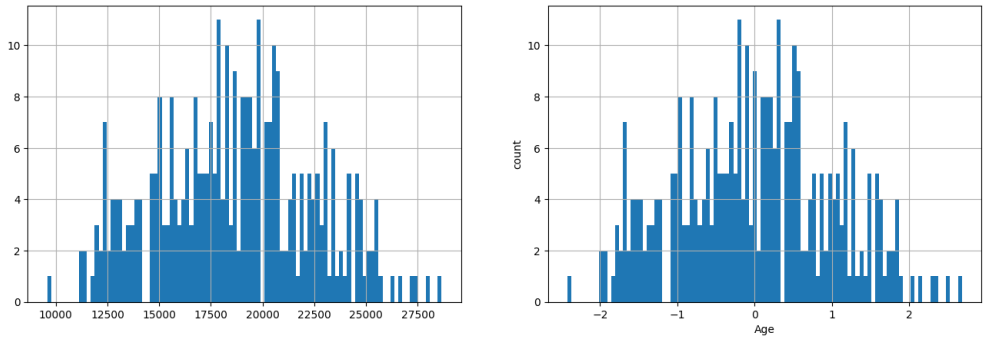

Figura 31: Comparació de la distribució de Age abans i després de la normalització

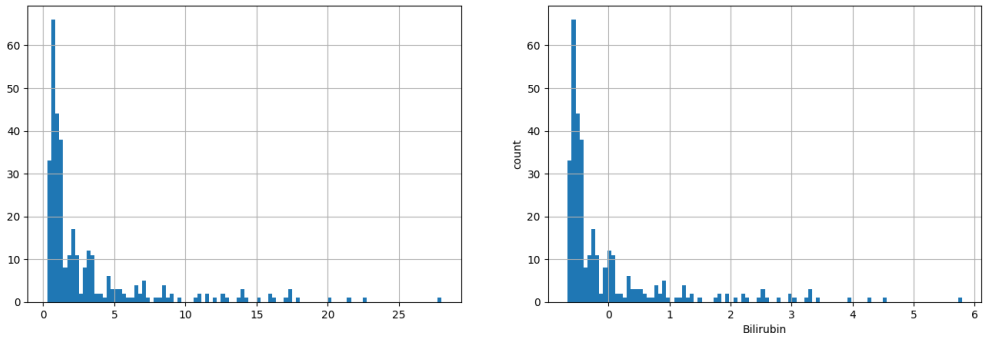

Figura 32: Comparació de la distribució de Bilirubin abans i després de la normalització

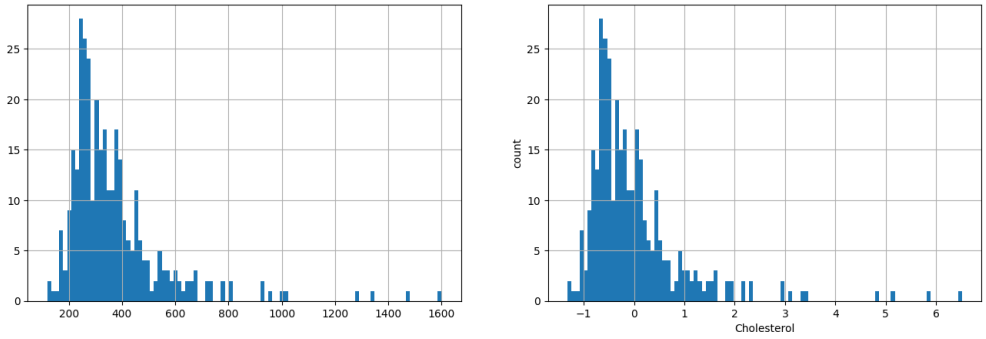

Figura 33: Comparació de la distribució de Cholesterol abans i després de la normalització

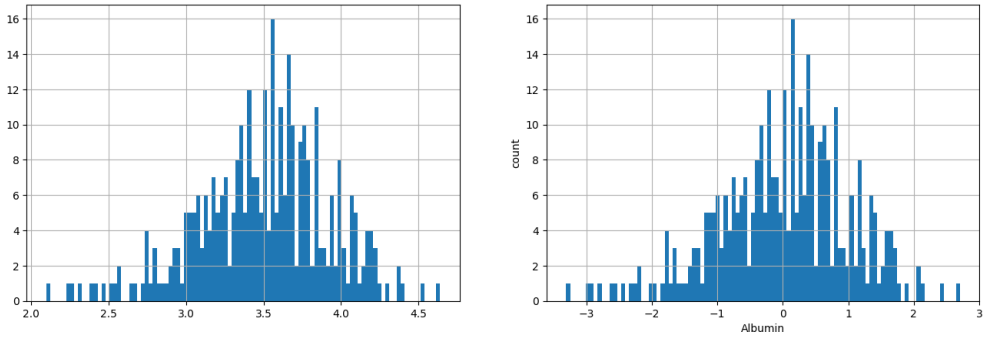

Figura 34: Comparació de la distribució de Albumin abans i després de la normalització

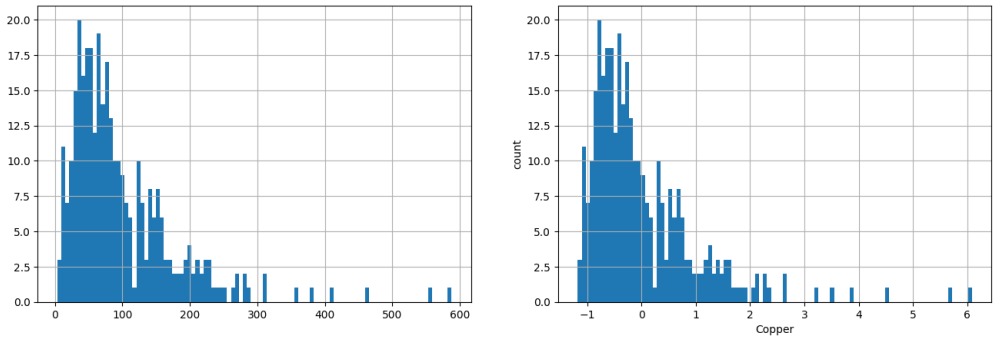

Figura 35: Comparació de la distribució de Copper abans i després de la normalització

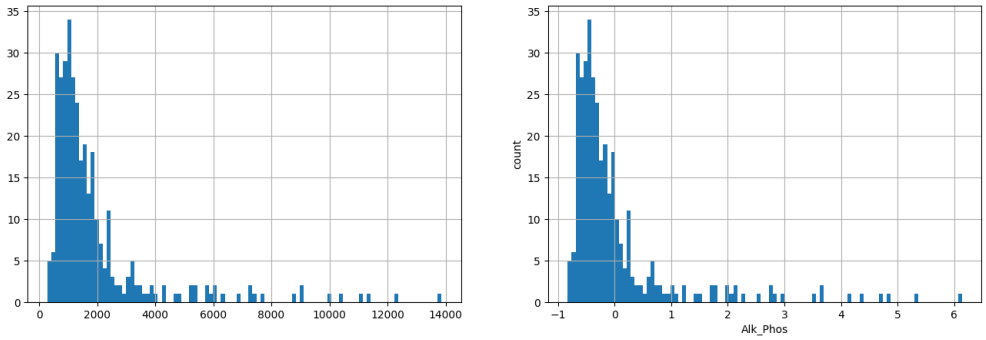

Figura 36: Comparació de la distribució de Alk_Phos abans i després de la normalització

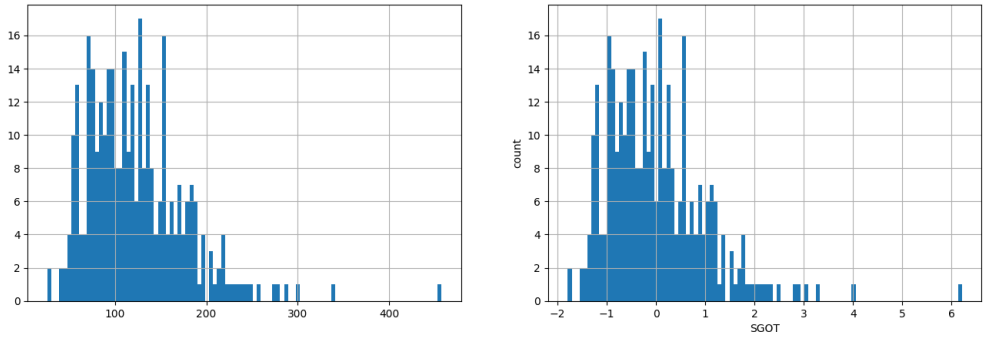

Figura 37: Comparació de la distribució de SGOT abans i després de la normalització

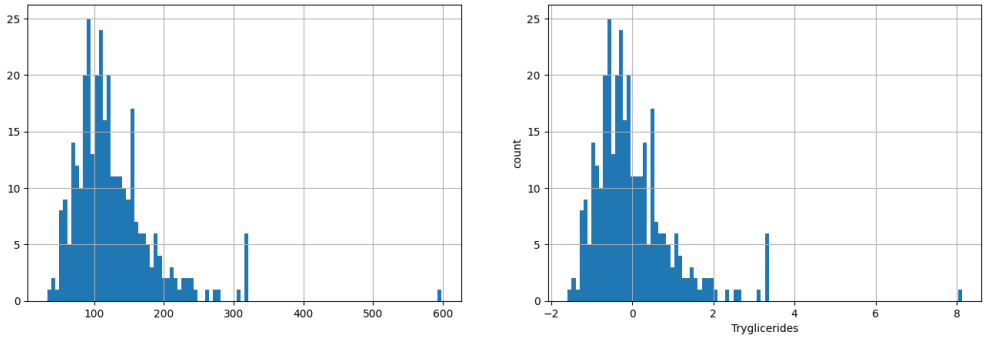

Figura 38: Comparació de la distribució de Tryglicerides abans i després de la normalització

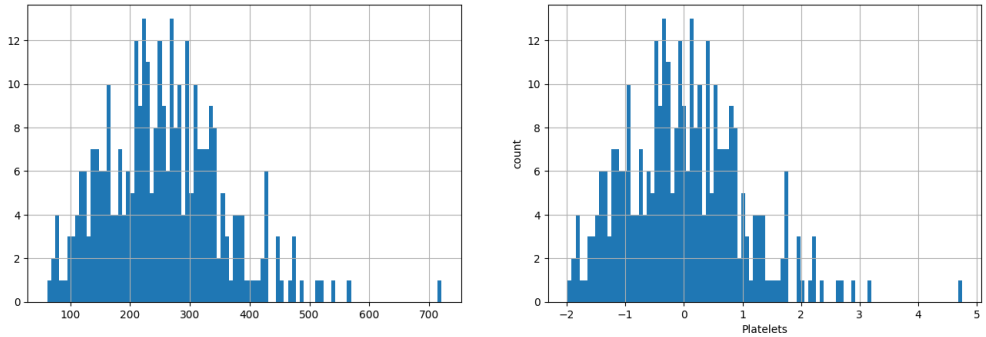

Figura 39: Comparació de la distribució de Platelets abans i després de la normalització

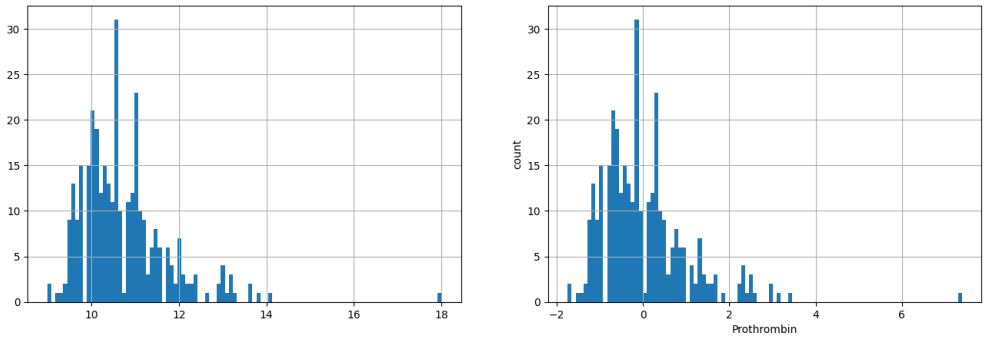

Figura 40: Comparació de la distribució de Prothrombin abans i després de la normalització

Amb la comparació dels histogrames de cada variable es pot observar que la normalització s'ha fet correctament. Hi ha alguns casos com "Copper" i "Alk_Phos" que no es veu el pic en la mitjana 0 pel fet que la distribució inicial no seguia una distribució normal.

3.0.2 Anàlisi de correlacions entre variables numèriques

La correlació expressa fins a quin punt dues variables numèriques canvien conjuntament a una taxa constant. A la següent matriu es pot observar les correlacions entre les variables numèriques, calculada de dos en dos.

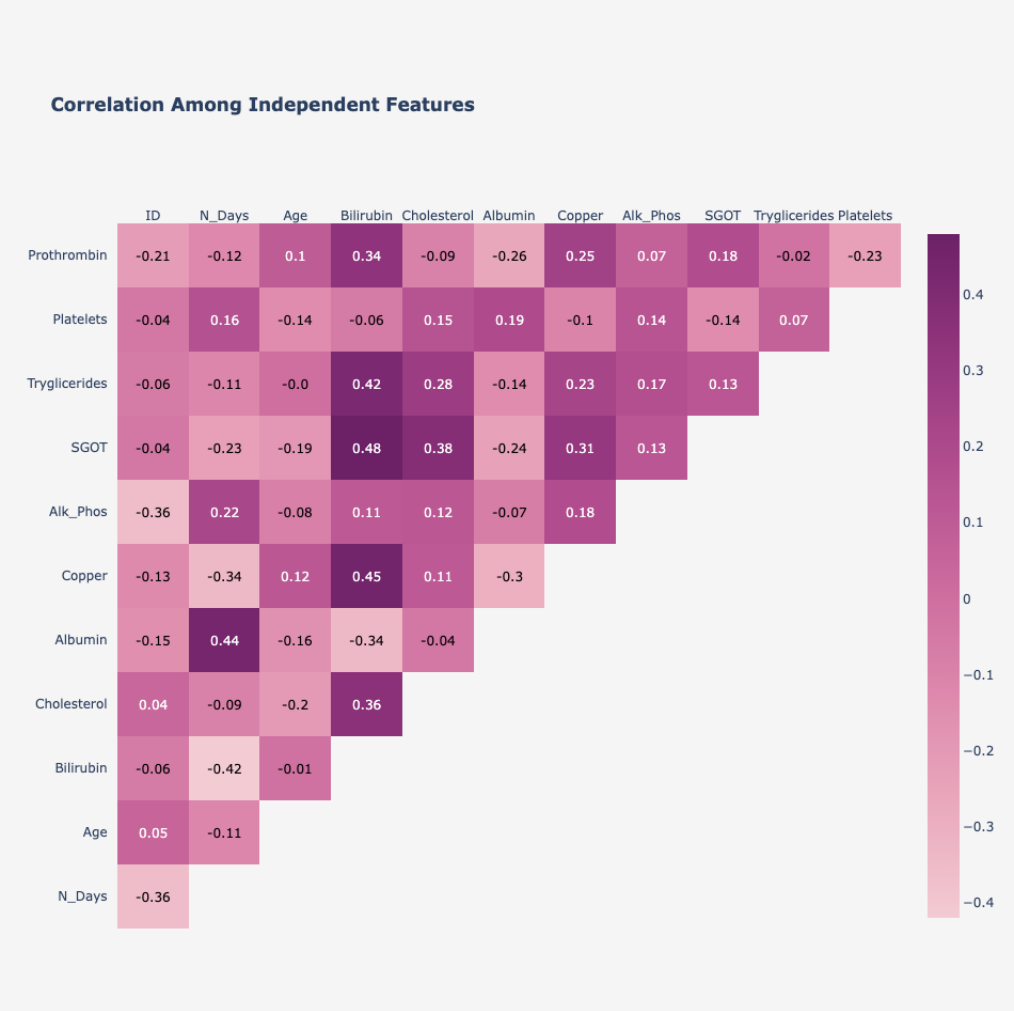

Figura 41: Matriu de correlació

Observant la matriu es pot extreure que:

- La variable "ID" té una correlació quasi 0 amb totes les altres variables. Es pot considerar eliminar-la, ja que no aporta informació rellevant al model.
- No hi ha cap parell de variables numèriques que tinguin una correlació molt alta, per tant, no cal eliminar cap variable per reduïr dimensionalitat pel fet que aporti la mateixa informació que una altra variable aporta.
- La correlació entre "Bilirubin" i "SGOT" és la més alta, amb un valor de 0.48.
- "Bilirubin" és la variable que té més casos que la correlació amb altres variables sigui entre 0.4 i 0.5.

3.0.3 Anàlisi de variables categòriques i variable objectiu

Per tal d'analitzar les relacions entre les variables categòriques i la variable resposta "Status", es creen els següents plots:

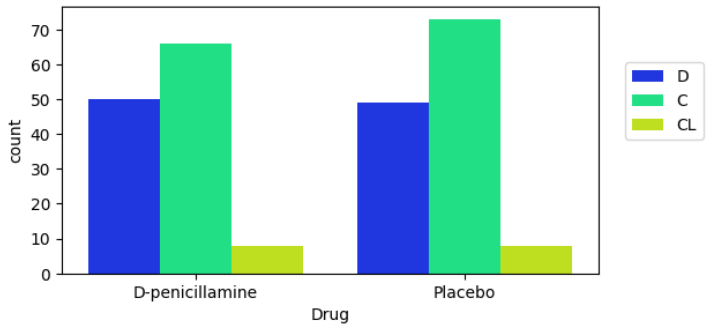

Figura 42: Anàlisi de la variable Drug i la variable objectiu

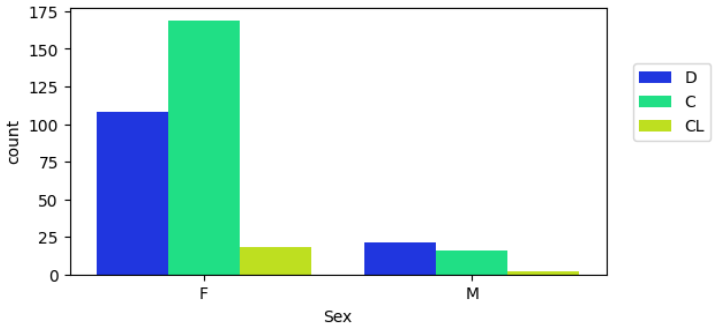

Figura 43: Anàlisi de la variable Sex i la variable objectiu

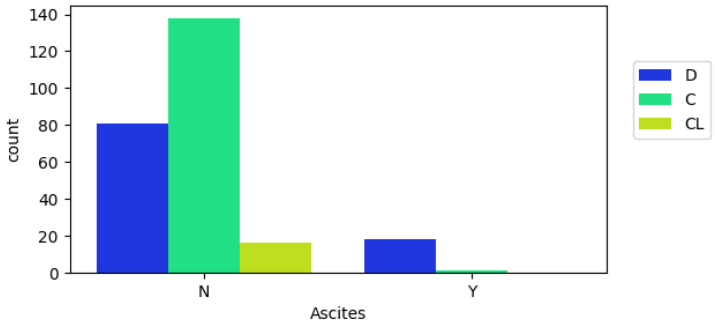

Figura 44: Anàlisi de la variable Ascites i la variable objectiu

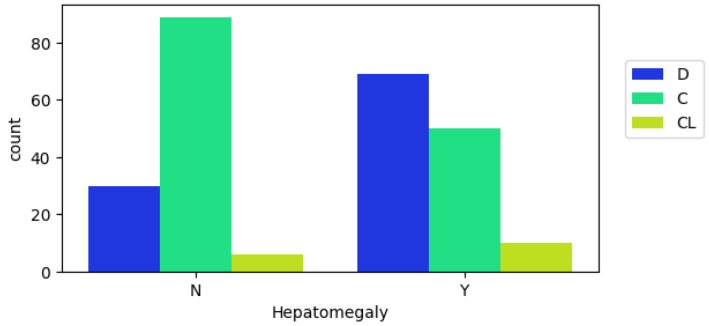

Figura 45: Anàlisi de la variable Hepatomegaly i la variable objectiu

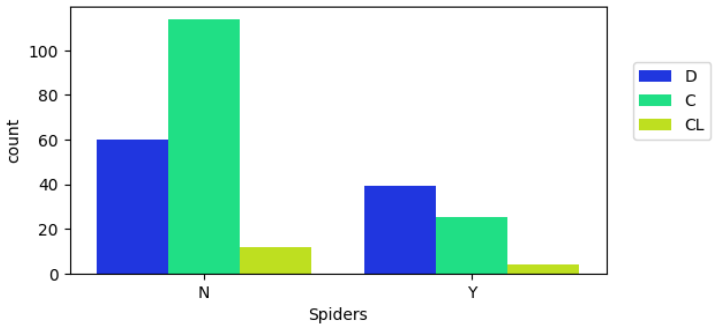

Figura 46: Anàlisi de la variable Spiders i la variable objectiu

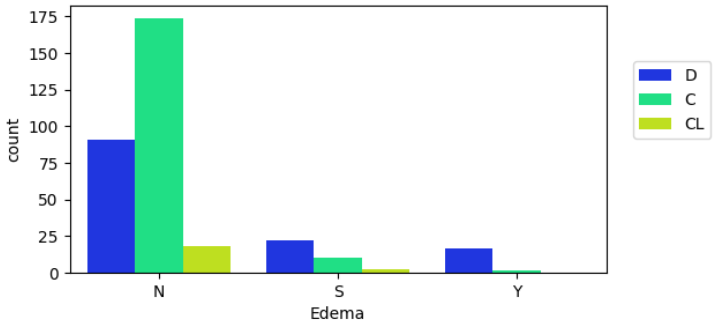

Figura 47: Anàlisi de la variable Edema i la variable objectiu

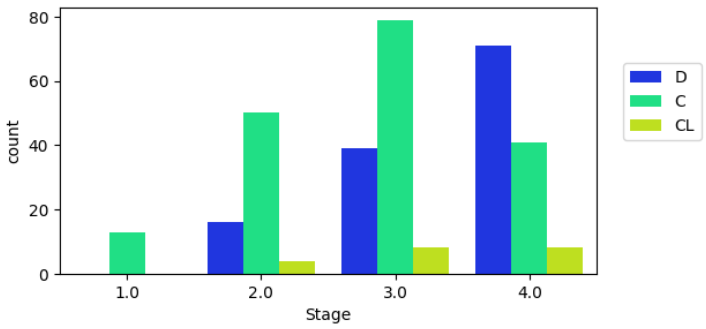

Figura 48: Anàlisi de la variable Stage i la variable objectiu

Als plots anteriors es poden observar que:

- La distribució de la variable resposta 'Status' no es veu gaire afectat per la variable "Drug".
- Hi ha molta més informació sobre les dones que dels homes.
- El fet de ser dona o no tenir presència d'Ascites augmenta els casos de censurat i de censurat per trasplantament hepàtic
- El fet de tenir presència d'Hepatomegàlia augmenta els casos de morts.
- El fet de no tenir presència d'Aranyes o sense Edema i sense tractament diürètic per a l'Edema augmenta els casos de censurat i de mort.

- La quantitat de morts augmenten en funció de Stage
- Globalment, es veu un desequilibri de casos per cada variable.

3.0.4 Eliminació de variables

La baixa dimensionalitat és beneficiosa a l'hora de crear els models, específicament per aquells que calculen distàncies. Per tant, segons les analitzacions i estudis dels apartats anteriors, s'eliminarà les següents variables del train i test:

- "Cholesterol": té una gran quantitat de missing (superior al 30%) i la imputació podria afegir molt de soroll.
- "Tryglicerides": té una gran quantitat de missing (superior al 30%) i la imputació podria afegir molt de soroll.
- "ID": no aporta informació al model. És només una identificació única per a les mostres.
- "Drug": no aporta informació rellevant tenint en compte que la variable resposta que cal analitzar és "Status".

3.0.5 Estudi de dimensionalitat amb PCA

PCA és una tècnica que permet reduir la dimensionalitat de les dades mantenint la major inèrcia possible. Pot ser una eina útil si els algoritmes que s'utilitzen per crear els models pateixen de la maledicció de la dimensionalitat, com seria el cas de l'algoritme KNN.

Per poder realitzar la PCA amb tota la base de dades, cal primer:

- Transformar les variables categòriques en tantes binàries com modalitats tingui. El PCA es realitza normalment amb variables numèriques i per tal de poder utilitzar les categòriques cal transformar-les.
- Normalitzar les variables. PCA es veu molt afectat pels rangs i la variància de cada variable.

Després d'utilitzar PCA a la base de dades, cal estudiar la variància explicada pel nombre de dimensions:

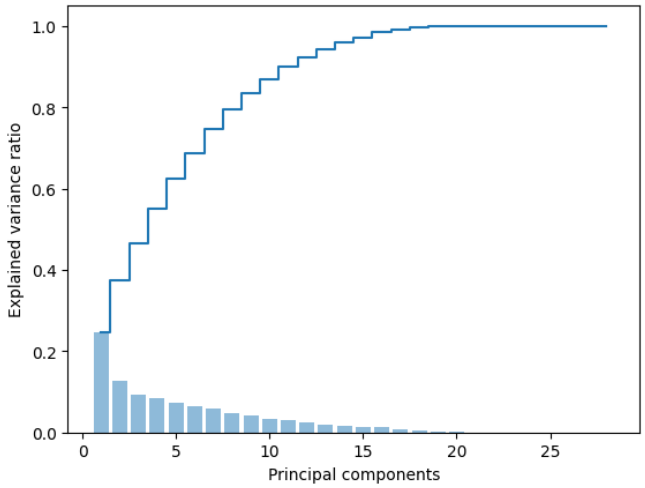

Figura 49: Variància explicada per número de dimensions

En el plot anterior es pot observar que, per obtenir una variància de 0.8 cal considerar 8 o 9 components principals.

4 SECCIÓ 3: Definició de models

4.1 k-Nearest Neighbor

4.1.1 Motivació i característiques desitjables

KNN és un algoritme supervisat que simplement cerca en les observacions més properes a la qual predir i el classifica basat en la majoria de dades que l'envolten.

La gran avantatge de l'algoritme és la seva senzillesa d'aprendre i implementar. Com a punt dèbil seria l'ús de tota la base de dades per entrenar cada punt, cosa que requereix grans quantitats de memòria. Però el dataset disposat és ideal per aquest cas, ja que és molt petit i amb poques variables.

A més, l'hiperparàmetre K és flexible. Això vol dir que es pot adaptar segons el tamany de les dades i a la qualitat del model. Cal considerar que amb un dataset desbalancejat, la K no pot ser superior a la classe minoritària perquè sinó serà incapaç de predir-la.

Encara que s'hagi fet la normalització de les variables, cal dir que KNN és un algoritme no paramètric. Altrament, és robust davant d'outliers perquè la decisió és basa en els k veïns més propers.

Finalment, la interpretació dels resultats no és tan directa, però tampoc la perd. És intuïtiva, ja que les prediccions es basen en els veïns més propers.

4.1.2 Definició de mètriques

En el cas de l'algoritme KNN, KNeighborsClassifier no proporciona l'opció d'utilitzar el class_weight. Per tant, si no es fa cap tècnica de resampling no es podran considerar les següents mètriques:

- Accuracy: té en compte les prediccions correctes de la població total.
- Precision: té en compte les prediccions positives respecte totes les prediccions fetes.
- Recall: té en compte les prediccions positives respecte la població positiva.
- F1-score: és la combinació de precision i recall

En totes les mètriques anteriors el desbalanceig de dades podria causar problemes en el rendiment del model, menyspreant la classe minoritària. Tenint en compte això, les mètriques que s'utilitzaran seran "balanced_accuracy" i "f1_weighted" que tenen en compte el desbalanceig de les dades.

4.1.3 Entrenament del model

Els hiperparàmetres utilitzats seran en funció de les tècniques de preparació de variables emprats.

Anteriorment ja s'ha dit que la imputació de les variables numèriques es farà amb KNN i de les variables categòriques amb una nova modalitat "missing". A més, per les variables numèriques es fara la normalització per afavorir els algoritmes que utilitzen distàncies, i per les categòriques es farà

un OneHotEncoder. Per tant, a cada iteració de creació del model el que es provarà seran els efectes de:

- Tractament d'outliers: no treure'n cap, treure amb threshold igual a 3 o treure amb threshold igual a 1.5.
- PCA: utilitzar o no la tècnica de PCA per reduir la dimensionalitat i mantenir una variància de 0.8.

Segons les tècniques de preprocessament emprades, la funció GridSearchCV (que prova tots els paràmetres proporcionats i troba la millor) retornarà uns paràmetres o uns altres.

- **Versió 1** Com a primera versió, el que s'ha provat és mantenir tots els outliers, utilitzar la tècnica PCA.

  El número de veïns com a paràmetre perquè GridSearchCV trobi la millor és entre 1 i 5. Això és perquè el dataset està desbalancejat i un número gran de K faria impossible la classificació com a classe minoritària.

  Els paràmetres que retorna la funció són: 'n_neighbors': 5, 'p': 1, 'weights': 'distance'. Els resultats de la cross_validation amb les mètriques F1 weighted i Balanced accuracy són:

	Hiperparàmetres	F1 weighted	Balanced accuracy
KNN + PCA	{'n_neighbors': 5, 'p': 1, 'weights': 'distance'}	0.69471	0.519731

Figura 50: Resultats de la cross validation de la versió 1

  Es pot observar que dona una millor resultat la primera mètrica. Ara, cal provar el model amb la partició de train per observar si hi ha overfitting. Després de l'execució, s'obté que balanced accuracy score és de 1. S'observa la següent gràfica:

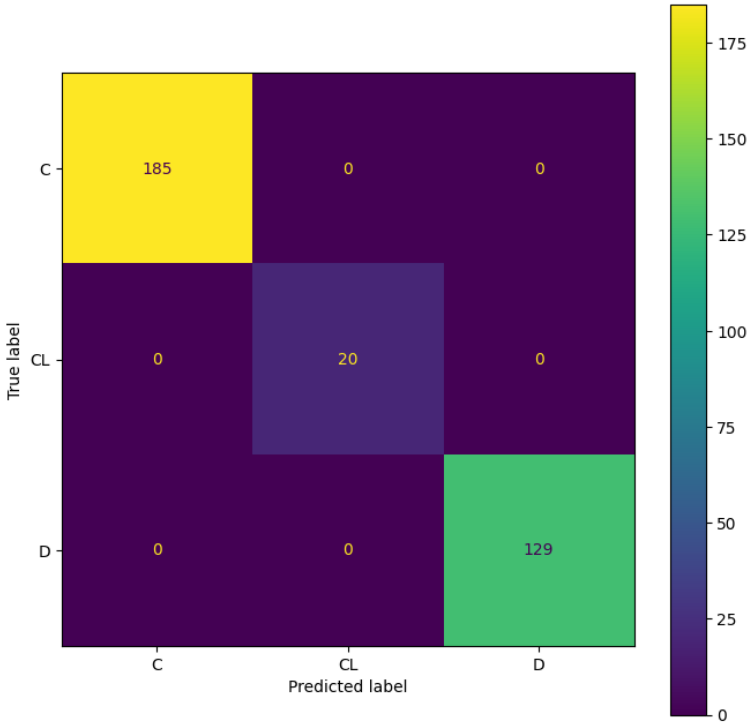

Figura 51: Matriu de confusió de train

S'observa que prediu bé tots els casos de train i té un balanced accuracy major que la validació, per tant, hi ha un overfitting.

- **Versió 2** Com a segona versió, el que s'ha provat és treure outliers amb un threshold de 3 i utilitzar la tècnica PCA.

  Els paràmetres que retorna la funció ara són: 'n_neighbors': 5, 'p': 2, 'weights': 'distance'. Els resultats de la cross_validation amb les mètriques F1 weighted i Balanced accuracy són:

	Hiperparàmetres	F1 weighted	Balanced accuracy
KNN + PCA	{'n_neighbors': 5, 'p': 1, 'weights': 'distance'}	0.694710	0.519731
KNN + PCA + threshold 3	{'n_neighbors': 5, 'p': 2, 'weights': 'distance'}	0.676797	0.469017

Figura 52: Resultats de la cross validation de la versió 2

Es pot observar que dona una millor resultat la primera mètrica. En comparació a la versió 1, els valors han baixat. Ara, cal provar el model amb la partició de train per observar si hi ha overfitting. Després de l'execució, s'obté altra vegada que balanced accuracy score és de 1. S'observa la següent gràfica:

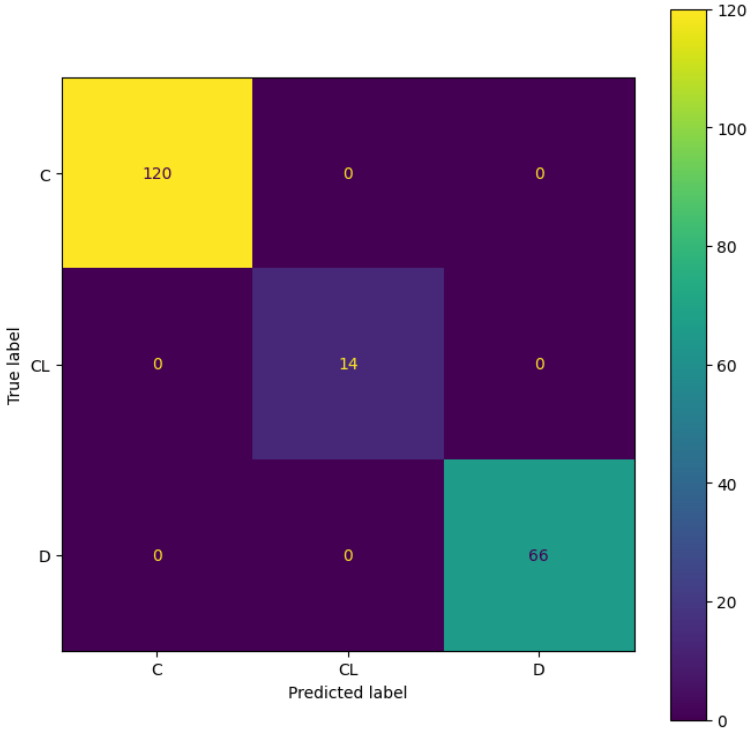

Figura 53: Matriu de confusió de train

S'observa la mateixa situació. Hi ha overfitting.

- **Versió 3** Ara, el que s'ha provat és treure outliers amb un threshold de 1.5 i utilitzar la tècnica PCA.

  Els paràmetres que retorna la funció són: 'n_neighbors': 1, 'p': 2, 'weights': 'uniform'. Els resultats de la cross_validation amb les mètriques F1 weighted i Balanced accuracy són:

	Hiperparàmetres	F1 weighted	Balanced accuracy
KNN + PCA	{'n_neighbors': 5, 'p': 1, 'weights': 'distance'}	0.694710	0.519731
KNN + PCA + threshold 3	{'n_neighbors': 5, 'p': 2, 'weights': 'distance'}	0.676797	0.469017
KNN + PCA + threshold 1.5	{'n_neighbors': 1, 'p': 2, 'weights': 'uniform'}	0.653454	0.471351

Figura 54: Resultats de la cross validation de la versió 3

Es pot observar que dona una millor resultat la primera mètrica. En comparació a la versió 2, el valor de la primera mètrica ha pujat i el de la segona ha baixat. Ara, cal provar el model amb la partició de train per observar si hi ha overfitting. Després de l'execució, s'obté altra vegada que balanced accuracy score és de 1. S'observa la següent gràfica:

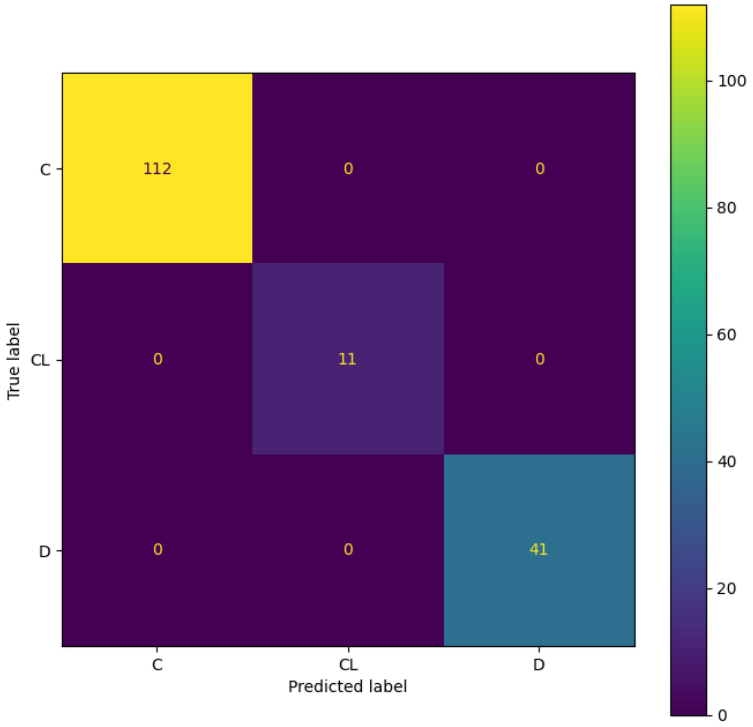

Figura 55: Matriu de confusió de train

S'observa la mateixa situació. Hi ha overfitting.

- **Versió 4** S'ha vist que en tots tres versions anteriors hi ha hagut overfitting. Per tant, cal pensar si el problema apareix en la PCA. Per aquesta versió es prova de mantenir els outliers, no utilitzar PCA.

  Els paràmetres que retorna la funció són: 'n_neighbors': 4, 'p': 1, 'weights': 'distance'. Els resultats de la cross_validation amb les mètriques F1 weighted i Balanced accuracy són:

	Hiperparàmetres	F1 weighted	Balanced accuracy
KNN + PCA	{'n_neighbors': 5, 'p': 1, 'weights': 'distance'}	0.694710	0.519731
KNN + PCA + threshold 3	{'n_neighbors': 5, 'p': 2, 'weights': 'distance'}	0.676797	0.469017
KNN + PCA + threshold 1.5	{'n_neighbors': 1, 'p': 2, 'weights': 'uniform'}	0.653454	0.471351
KNN	{'n_neighbors': 4, 'p': 1, 'weights': 'distance'}	0.684090	0.525792

Figura 56: Resultats de la cross validation de la versió 3

En comparació amb versions anteriors, els valors de les mètriques han pujat. Ara, cal provar el model amb la partició de train per observar si hi ha overfitting. Després de l'execució, s'obté altra vegada que balanced accuracy score és de 1. S'observa la següent gràfica:

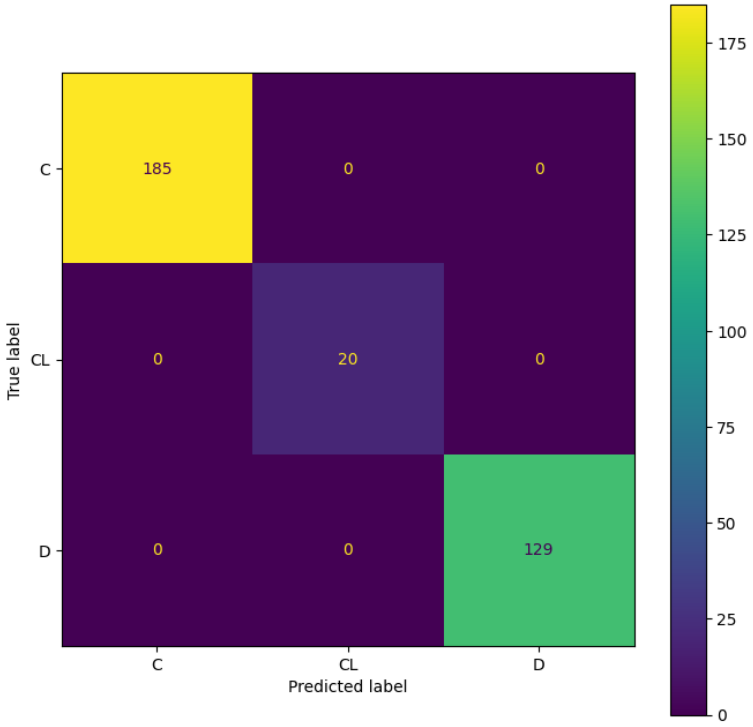

Figura 57: Matriu de confusió de train

S'observa la mateixa situació. Hi ha overfitting en tots els casos.

4.2 Arbre de decisió

4.2.1 Motivació i característiques desitjables

Un arbre de decisió és un algoritme supervisat i no paramètric. Té una estructura d´arbre jeràrquica, que consta d´un node arrel, branques, nodes interns i nodes fulla.

L'avantatge principal dels arbres de decisió és la seva interpretabilitat. Amb una naturalesa jeràrquica i representacions visuals, es poden observar els atributs importants i els mètodes de decisió a cada pas. També es beneficien de tenir un conjunt de dades reduït.

Una altra característica important és que no li fa falta gaire preparació de les dades. Són flexibles a nivell de que poden rebre tant variables categòriques com numèriques sense transformacions especials.

Són capaços de modelar relacions no lineals i identificar les variables més importants per prendre les decisions.

4.2.2 Definició de mètriques

Per l'arbre de decisió, DecisionTreeClassifier proporciona l'opció d'utilitzar class_weight. Com en el cas anterior, no es farà servir tècniques de resampling, per tant, no s'empraran les mètriques que no tenen en consideració el desbalanceig de clases de la base de dades.

Les mètriques que s'utilitzaran tornen a ser "balanced_accuracy" i "f1_weighted" que tenen en compte el desbalanceig de les dades.

4.2.3 Entrenament del model

Els hiperparàmetres utilitzats seran en funció de les tècniques de preparació de variables emprats. Cal dir que els arbres de decisió tenen molts hiperparàmetres per considerar. En aquest cas, es provaran:

- "criterion" que especifica la funció per mesurar la qualitat d'una divisi. Pot tenir les següents expressions: "gini", "entropy" i "log_loss".
- "max_depth" que especifica la profunditat màxima de l'arbre.
- "ccp_alpha" és el paràmetre per controlar la poda de l'arbre per evitar l'overfitting (el problema del model KNN). Un valor elevat de "ccp_alpha" podarà més intensament l'arbre. Els valors que s'ha volgut provar es troben entre 0 i 1.

Els hiperparàmetres que s'ha tingut en compte són les anterior. No estan totes perquè s'ha volgut provar una quantitat alta de valors per "ccp_alpha". Altres hiperparàmetres que els arbre de decisió disposen són: "splitter" que controla l'estratègia de divisió de nodes, "min_samples_split" que controla el nombre mínim de mostres requerides per dividir un node intern, entre d'altres.

En l'entrenament d'aquest model, es mantindrà la imputació amb KNN de les variables numèriques i amb una nove modalitat les categòriques. Pels arbres de decisió no cal fer un OneHotEncoder que augmenta el nombre de dimensions, per tant, es realitzarà un OrdinalEncoder. També es treurà la PCA, ja que no utilitza distàncies en la modelització.

A cada iteració de creació del model el que es provarà seran els efectes de:

- Tractament d'outliers: no treure'n cap, treure amb threshold igual a 3 o treure amb threshold igual a 1.5.
- class_weight: es provarà si utilitzant aquest paràmetre millora el rendiment del model.

Segons les tècniques de preprocessament emprades, la funció GridSearchCV (que prova tots els paràmetres proporcionats i troba la millor) retornarà uns paràmetres o uns altres.

- **Versió 1** La primera versió del model s'entrenarà amb el preprocessament indicat anteriorment i sense treure outliers ni utilitzar class_weight.

  Tenint en compte els paràmetres que ha tingut com a entrada, el GridSearchCV retorna com a millors els següents: 'ccp_alpha': 0.005, 'criterion': 'gini', 'max_depth': 3.

  Amb aquests paràmetres, el cross validation retorna els següents valors per les mètriques:

	Hiperparàmetres	F1 weighted	Balanced accuracy
DT	{'ccp_alpha': 0.005, 'criterion': 'gini', 'max...	0.693561	0.496613

Figura 58: Resultats de la cross validation de la versió 1

F1 weighted torna a ser major que Balanced accuracy. Ara, cal provar el model amb la partició de train per observar si hi ha overfitting com en el cas de KNN. Després de l'execució, s'obté que balanced accuracy score és de 0.61. S'observa la següent gràfica:

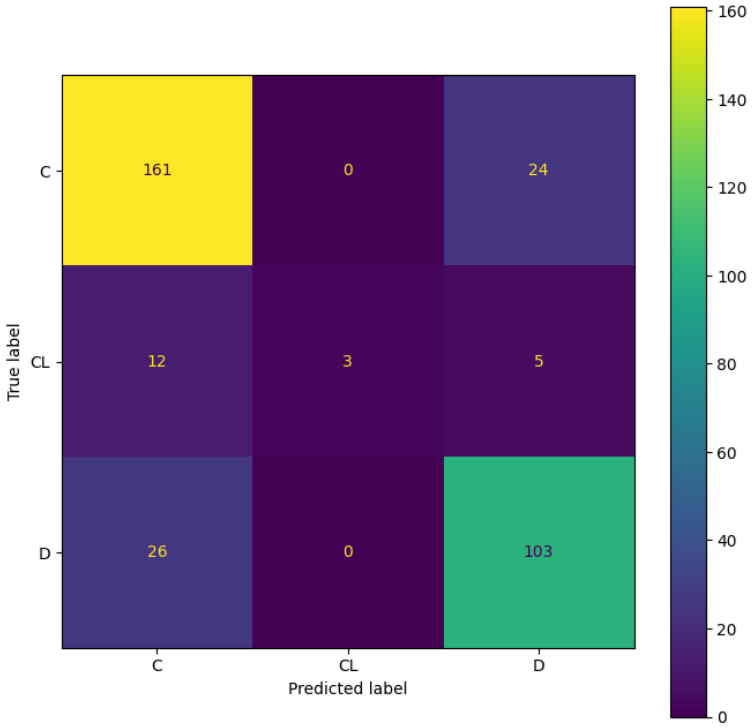

Figura 59: Matriu de confusió de train

S'observa que aquesta vegada ja no hi ha overfitting en la partició de train. Tendeix a predir més bé la classe majoritària C, i es veu que la classe minoritària només hi ha 3 prediccions correctes.

- **Versió 2** La segona versió del model s'entrenarà amb el preprocessament indicat anteriorment itreient els outliers amb un threshold de 3. No s'utilitza class_weight.

  Tenint en compte els paràmetres que ha tingut com a entrada, el GridSearchCV retorna com a millors els següents: 'ccp_alpha': 0.025, 'criterion': 'entropy', 'max_depth': 6.

  Amb aquests paràmetres, el cross validation retorna els següents valors per les mètriques:

	Hiperparàmetres	F1 weighted	Balanced accuracy
DT + Threshold 3	{'ccp_alpha': 0.025, 'criterion': 'entropy', '...	0.642756	0.464591

Figura 60: Resultats de la cross validation de la versió 2

F1 weighted, com sempre, torna a ser major que Balanced accuracy amb un valor de 0.64. Comparat amb la versió anterior, es veu que aquest té pitjors valors en les mètriques. Ara, cal provar el model amb la partició de train per observar si hi ha overfitting. Després de l'execució, s'obté que balanced accuracy score és de 0.8 que comença a haver una mica d'ouverfitting. S'observa la següent gràfica:

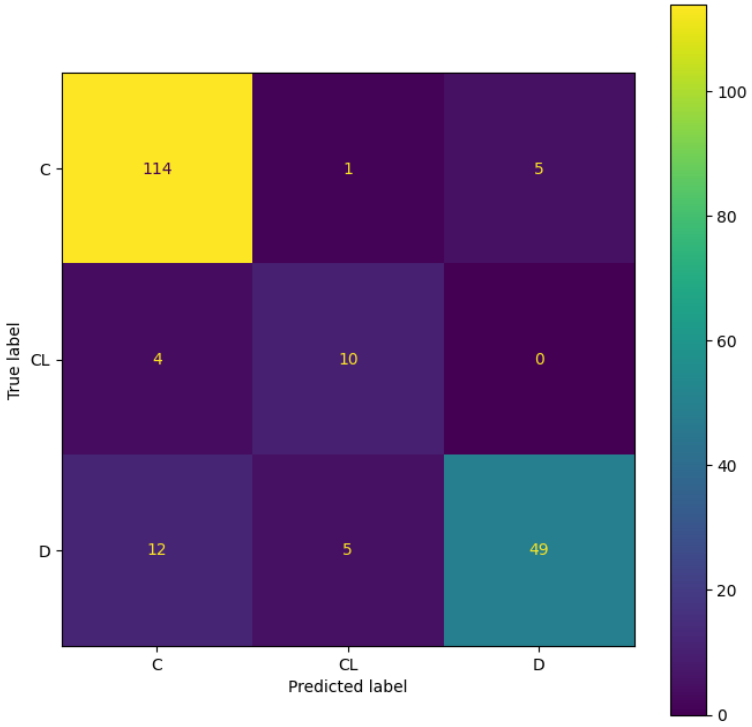

Figura 61: Matriu de confusió de train

Es veu que fa millors prediccions que la versió anterior. Ja no té tanta tendència a predir la classe majoritària però es veu que possiblement és causa de l'overfitting.

- **Versió 3** Es pot observar que minimitzar el valor de threshold es manté poques dades i hi ha perill d'overfitting. Per tant, a la versió 3 es comprovara l'eficiència del model sense treure outliers i utilitzant class_weight.

  Tenint en compte els paràmetres que ha tingut com a entrada, el GridSearchCV retorna com a millors els següents: 'ccp_alpha': 0.005, 'criterion': 'gini', 'max_depth': 5.

  Amb aquests paràmetres, el cross validation retorna els següents valors per les mètriques:

	Hiperparàmetres	F1 weighted	Balanced accuracy
DT + class_weight	{'ccp_alpha': 0.025, 'criterion': 'entropy', '...	0.667962	0.511099

Figura 62: Resultats de la cross validation de la versió 3

F1 weighted, com sempre, torna a ser major que Balanced accuracy amb un valor de 0.67. Comparat amb la versió anterior, es veu que aquest té millors valors en les mètriques. Ara, cal provar el model amb la partició de train per observar si hi ha overfitting. Després de l'execució, s'obté que balanced accuracy score és de 0.84, amb més tendència a ser overfitting. S'observa la següent gràfica:

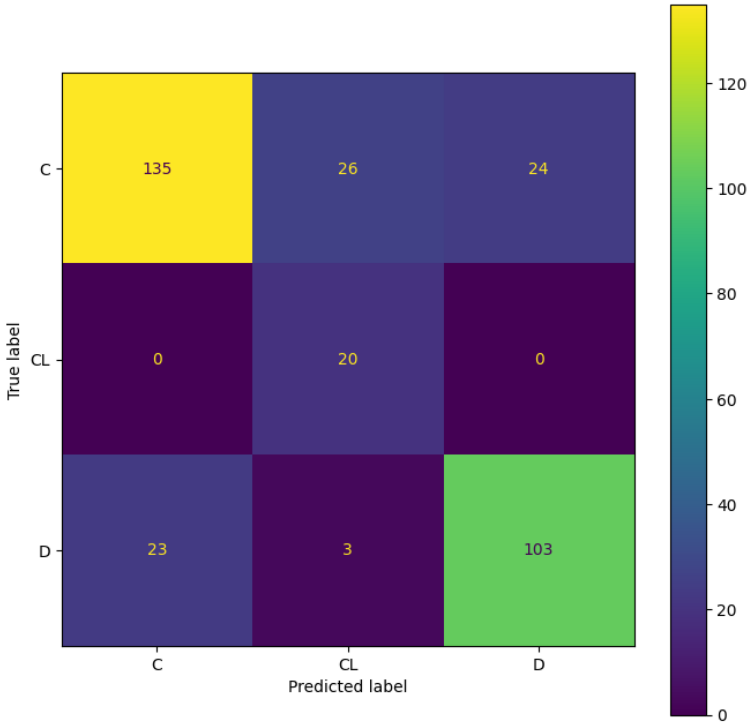

Figura 63: Matriu de confusió de train

En aquesta gràfica s'observa la importància d'utilitzar class_weight. Es veu que prediu molt bé la classe minoritària.

- **Versió 4** En aquesta versió es fa el mateix que l'anterior però treient outliers amb threshold igual a 3.

  Tenint en compte els paràmetres que ha tingut com a entrada, el GridSearchCV retorna com a millors els següents: 'ccp_alpha': 0.025, 'criterion': 'entropy', 'max_depth': 6.

  Amb aquests paràmetres, el cross validation retorna els següents valors per les mètriques:

	Hiperparàmetres	F1 weighted	Balanced accuracy
DT + class_weight + Threshold 3	{'ccp_alpha': 0.025, 'criterion': 'entropy', '...	0.690502	0.586294

Figura 64: Resultats de la cross validation de la versió 4

Es pot observar que s'ha obtingut les millors mètriques en aquest model. Ara, cal provar el model amb la partició de train per observar si hi ha overfitting. Després de l'execució, s'obté que balanced accuracy score és de 0.85, amb més tendència a ser overfitting. S'observa la següent gràfica:

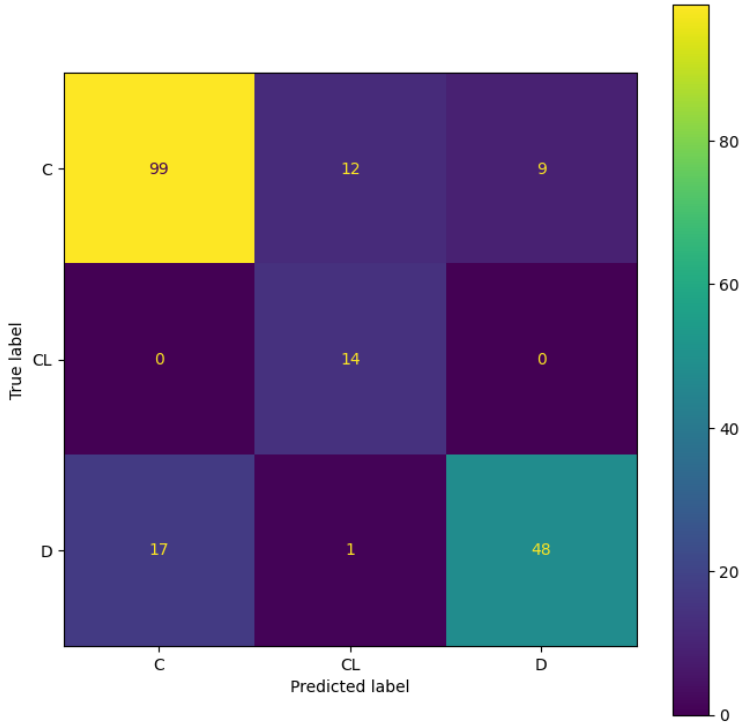

Figura 65: Matriu de confusió de train

Aquesta gràfica és molt semblant a la anterior, però amb menys mostres. També pateix el perill d'overfitting.

4.3 Support vector machine

4.3.1 Motivació i característiques desitjables

Les SVM poden generalitzar bé les dades de baixes o grans dimensions. Amb el control dels hiperparàmetres i del kernel, l'algoritme és flexible podent ajustar la complexitat del model. Per tant, a partir de la regularització de paràmetres, es pot evitar l'overfitting.

A més, gràcies a l'ús de les funcions de kernel, SVM pot controlar dades no lineals. Davant de dades no balancejades, també pot ajustar el cost per penalitzar de formes diferents les classes majoritàries i minoritàries.

4.3.2 Definició de metriques

Com en l'arbre de decisió, SVM proporciona l'opció d'utilitzar class_weight. Per tant, per tractar bé les dades desequilibrades es tornaran a utilitzar les mètriques "balanced_accuracy" i "f1_weighted" que tenen en compte el desbalanceig de les dades.

4.3.3 Entrenament del model

Els hiperparàmetres que es tindran en compte per SVM són el cost i la gamma. Aquests prendran valors segons les tècniques de preparació de variables emprats.

Segons les tècniques de preprocessament emprades, la funció GridSearchCV (que prova tots els paràmetres proporcionats i troba la millor) retornarà uns paràmetres o uns altres.

La imputació, com en tots els casos anteriors, es farà amb KNN per les variables numèriques i crear una nova modalitat per les categòriques. A més, per un SVM cal tenir en compte que :

- Es requereix que totes les variables d'entrada siguin numèriques. Per tant, es transformaran les categòriques amb un OneHotEncoder.
- Cal que les variables numèriques tinguin la mateixa escala, i això s'aconsegueix com anteriorment normalitzant les variables
- No és necessari utilitzar un PCA, ja que perd la interpretabilitat del model i, a més, SVM ja treballa bé amb altes dimensions.

Tenint el preprocessament inicial, els canvis que s'observaran seran els següents:

- Tractament d'outliers: no treure'n cap i treure amb threshold igual a 3. Com ja s'ha observat anteriorment, treure outliers amb threshold igual a 1.5 no aporta beneficis però disminueix el nombres de mostres que hi ha.
- class_weight: es provarà si utilitzant aquest paràmetre millora el rendiment del model.

Les versions dutes a terme es troben a continuació:

- **Versió 1** Com a primera versió, el que s'ha provat és mantenir tots els outliers i no utilitzar class_weight.

  Els paràmetres donats a GridSearchCV són:

  - C: 0.01,0.1,1,2,3,4,5,10
  - kernel: 'linear', 'rbf', 'sigmoid'
  - gamma: 'scale', 'auto'

  Els paràmetres que retorna la funció són: 'C': 10, 'gamma': 'scale', 'kernel': 'rbf'. Els resultats de la cross_validation amb les mètriques F1 weighted i Balanced accuracy són:

	Hiperparàmetres	F1 weighted	Balanced accuracy
SVC	{'C': 10, 'gamma': 'scale', 'kernel': 'rbf'}	0.729456	0.574473

Figura 66: Resultats de la cross validation de la versió 1

  Es pot observar que els valors de les dues mètriques són bastant altes. Arriba a uns números superiors a tots els models analitzats anteriorment. Ara, cal provar el model amb la partició

de train per observar si hi ha overfitting. Després de l'execució, s'obté que balanced accuracy score és de 0.93. S'observa la següent gràfica:

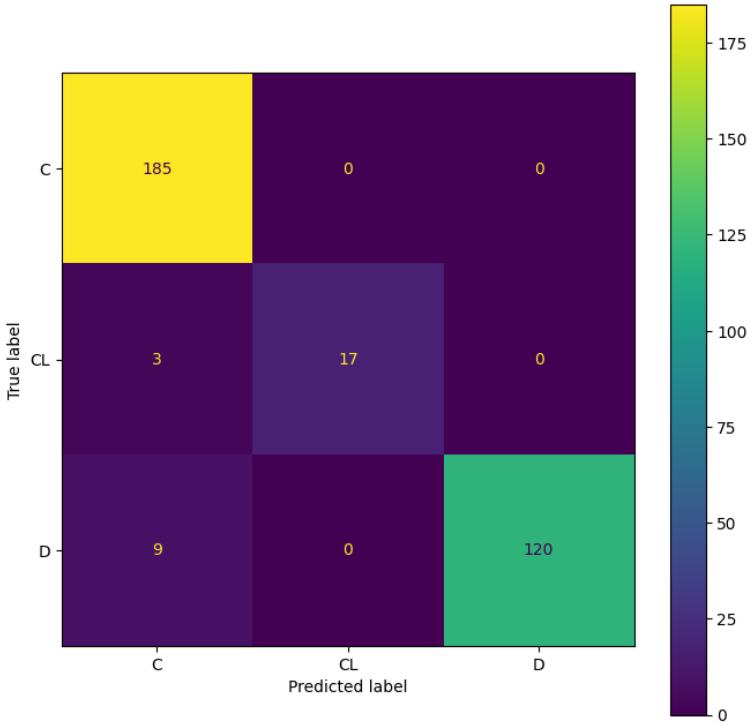

Figura 67: Matriu de confusió de train

S'observa que prediu bé tant la classe majoritària com la minoritària. Però té un balanced accuracy major que la validació, per tant, hi ha un overfitting.

- **Versió 2** Com a segona versió, el que s'ha provat és treure els outliers amb un threshold igual a 3 i no utilitzar class_weight.

  Amb els mateixos paràmetres que es dona a GridSearchCV, retorna: 'C': 3, 'gamma': 'scale', 'kernel': 'linear'. Els resultats de la cross_validation amb les mètriques F1 weighted i Balanced accuracy són:

	Hiperparàmetres	F1 weighted	Balanced accuracy
SVC + Threshold 3	{'C': 3, 'gamma': 'scale', 'kernel': 'linear'}	0.705483	0.556258

Figura 68: Resultats de la cross validation de la versió 2

Els valors de les mètriques tornen a ser superiors a tots els models analitzats anteriorment, però és menor que la versió 1. Ara, cal provar el model amb la partició de train per observar si hi ha overfitting. Després de l'execució, s'obté que balanced accuracy score és de 0.64. A diferència de l'anterior, aquesta versió no té overfitting al train. S'observa la següent gràfica:

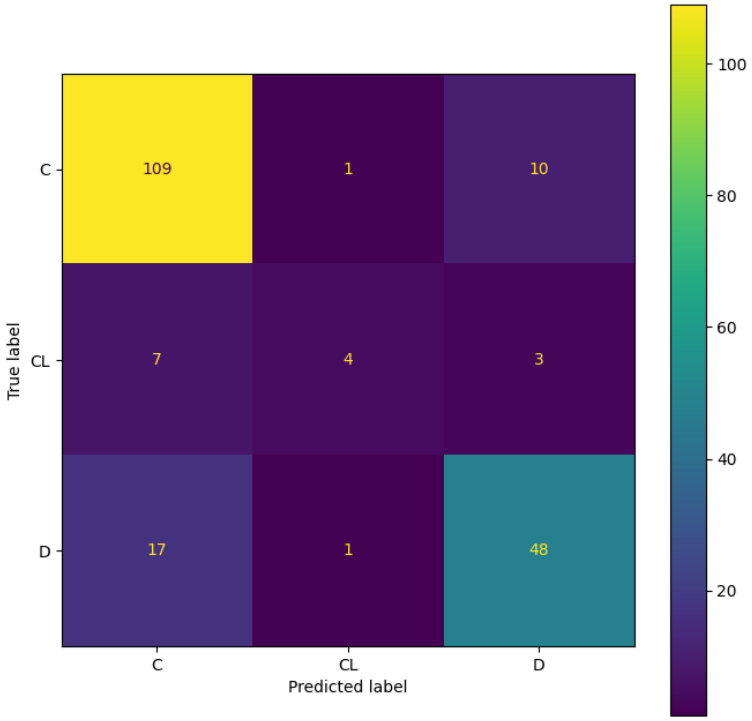

Figura 69: Matriu de confusió de train

S'observa que prediu bé tant la classe majoritària com la minoritària. Encara que no utilitza class_weight, pot predir bé molts valors de la classe CL.

- **Versió 3** Com a tercera versió, el que s'ha provat és mantenir els outliers i utilitzar class_weight.

  Amb els mateixos paràmetres que es dona a GridSearchCV, retorna: 'C': 10, 'gamma': 'scale', 'kernel': 'rbf'. Els resultats de la cross_validation amb les mètriques F1 weighted i Balanced accuracy són:

	Hiperparàmetres	F1 weighted	Balanced accuracy
SVC + class weight	{'C': 10, 'gamma': 'scale', 'kernel': 'rbf'}	0.702556	0.554653

Figura 70: Resultats de la cross validation de la versió 3

Els valors de les mètriques són semblants a la versió anterior. Ara, cal provar el model amb la partició de train per observar si hi ha overfitting. Després de l'execució, s'obté que balanced accuracy score és de 0.97, amb lo qual vol dir que hi ha overfitting en les dades de train. S'observa la següent gràfica:

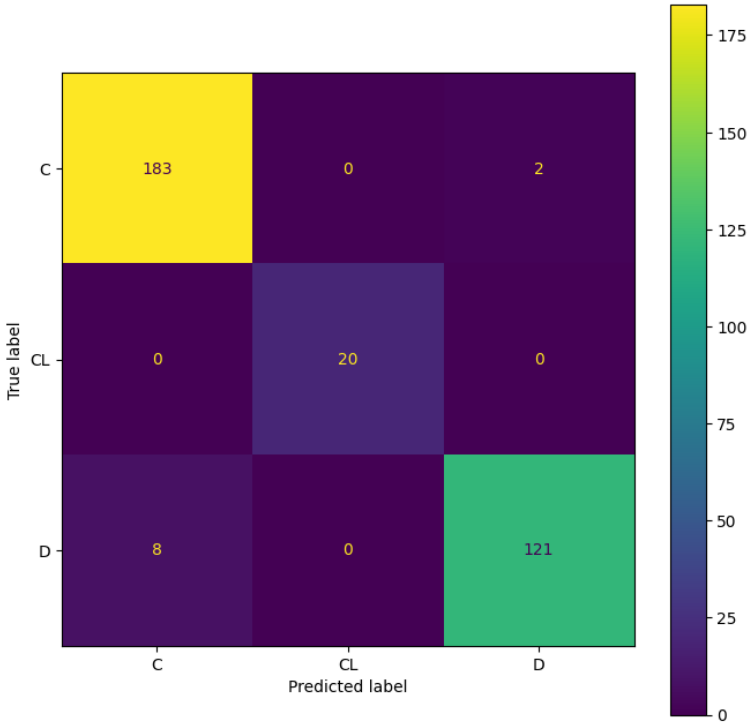

Figura 71: Matriu de confusió de train

S'observa que prediu molt bé totes les classes i és exactament un overfitting.

- **Versió 4** Com a quarta versió, el que s'ha provat és treure els outliers amb threshold igual a 3 i utilitzar class_weight.

  Amb els mateixos paràmetres que es dona a GridSearchCV, retorna: 'C': 3, 'gamma': 'scale', 'kernel': 'linear'. Els resultats de la cross_validation amb les mètriques F1 weighted i Balanced accuracy són:

	Hiperparàmetres	F1 weighted	Balanced accuracy
SVC + class weight + Threshold 3	{'C': 3, 'gamma': 'scale', 'kernel': 'linear'}	0.638251	0.503816

Figura 72: Resultats de la cross validation de la versió 4

Els valors de les mètriques són menors que les versions anteriors. A més, després de l'execució, s'obté que balanced accuracy score és de 0.83, amb lo qual vol dir que és la versió menys desitjada de SVC perquè els valors de mètriques són baixos i hi ha overfitting en les dades de train. S'observa la següent gràfica:

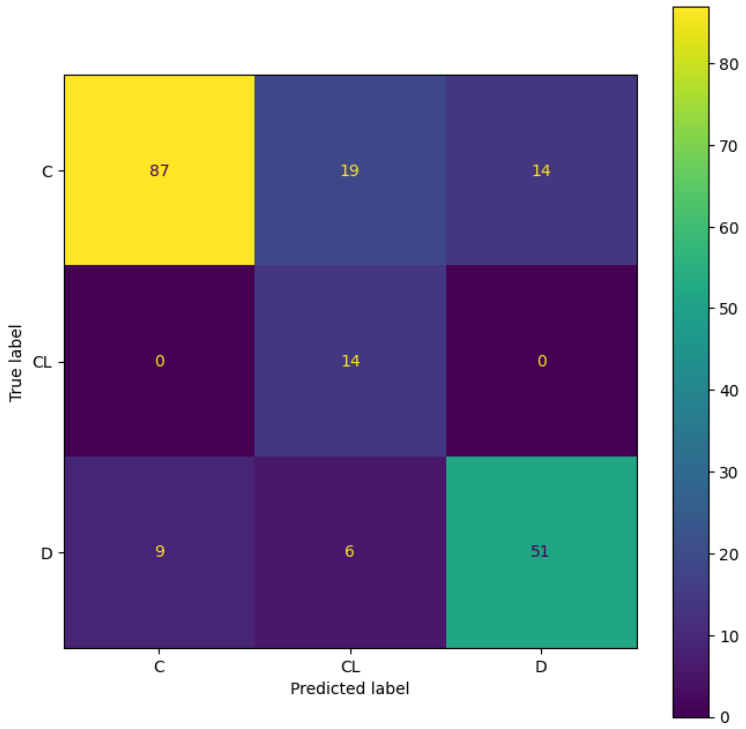

Figura 73: Matriu de confusió de train

S'observa que prediu més bé les classes minoritàries que les altres a causa de l'ús de class_weight.

5 SECCIÓ 4: Selecció de model

El model que cal triar hauria de tenir uns valors de les mètriques escollides (F1 weighted i Balanced accuracy) altes per a què pugui encertar en moltes prediccions, ja que s'espera que es tingui un ús en el camp mèdic. A més, cal que el balanced accuracy obtingut en en la prova amb la partició de train no sigui massa superior a la obtingut en el cross validation. No hi ha interés en que el model aprengui massa de les dades de train, perquè hi hauria perill d'overfitting i predir-ne malament les dades futures dels pacients.

Tenint en compte els punts anteriors, el model escollit és la versió 2 de l'algoritme SVM. Aquest treu els outliers de la base de dades amb un threshold de 3, intentant mantenir la màxima quantitat de dades. Curiosament, aquest model no fa servir class_weight. Segurament, degut a això, el model que no presenta un overfitting en la partició de train com en els altres casos.

El model escollit té un F1 weighted de 0.7 i un Balanced accuracy de 0.56. Això vol dir que té una bona capacitat per classificar les mostres. A més, la mètrica F1 weighted és la unió de la precision i la recall. Aleshores, el rendiment del model és banstant convenient en termes de que gran part de les prediccions són certes i gran part de les prediccions prediuen la classe correcta. D'altra banda, encara que el Balanced accuracy no és tan alt, indica també un rendiment acceptable. Les dues mètriques són indicadors importants per les classes debalancejades i encara més si es tracta de problemes mèdics. Addicionalment, cal esmentar que el kernel que utilitza el model és la lineal, que dona més interpretabilitat comparat amb altres kernels de SVM.

Com a limitacions cal especificar que el model no és excepcionalment bo com a rendiment global. A més, l'entrenament s'ha fet en un conjunt de dades molt reduït i sense class_weight, això pot portar al perill que en conjunt de dades més grans, el model no el pugui generalitzar bé.

Per observar el rendiment del model, s'ha volgut comparar els valors de les mètriques F1 weighted i Balanced accuracy que s'obtenen en el cross validation, en la partició de train i en la partició de test:

	Hiperparàmetres	F1 weighted VAL	Balanced accuracy VAL	F1 weighted TRAIN	Balanced accuracy TRAIN	F1 weighted TEST	Balanced accuracy TEST
SVC + Threshold 3	{'C': 3, 'gamma': 'scale', 'kernel': 'linear'}	0.705483	0.556258	0.794445	0.64044	0.842829	0.749673

Figura 74: Comparació de les mètriques a train, cross validatio i test

La partició de test té un F1 weighted i un Balanced accuracy millors que les altres. Això és possiblement degut a un overfitting reduït a les dades d'entrenament que fa que hagi après altres aspectes que potser es troben en el conjunt de proves. El tamany del dataset també pot haver influït a aquests resultats. En tot cas, això pot conduir a la necessitat d'estudis posteriors amb altres mètriques per assegurar de la causa d'aquest efecte.

Els següents plots mostren les matrius de confusió de la partició de train i de test:

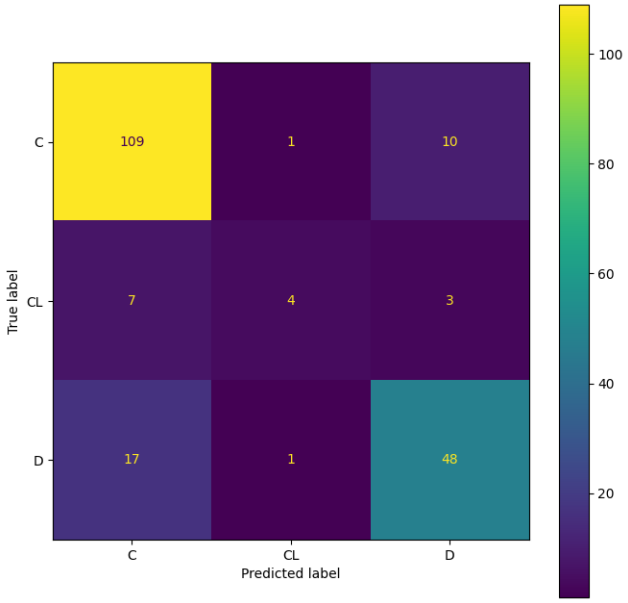

Figura 75: Matriu de confusió de train

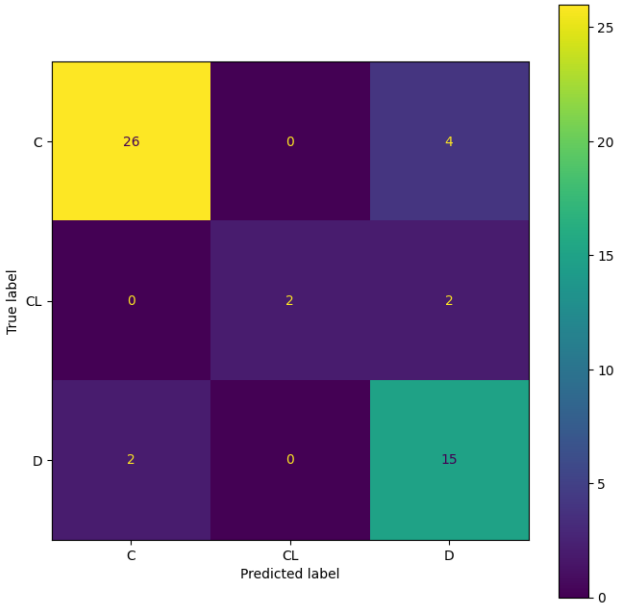

Figura 76: Matriu de confusió de test

S'observa que en la partició de test, el model prediu una mica millor tal com s'esperava pels valors obtinguts a les mètriques.

6 SECCIÓ 5: Model Card

Detalls del model

Autora

Zhiqian Zhou, zhiqian.zhou@estudiantat.upc.edu

Data i versió

data: 28 - 12 - 2023
version_name: svm_zqz_10086

Tipus de model

El model és un Support Vector Machine entrenat amb finalitat de predir la variable resposta demanada.

Informació

La base de dades original és proporcionada per la UCI (la Universitat de Califòrnia a Irvine) anomenada "Cirrhosis Patient Survival Prediction" amb la qual s'ha entrenat el model consta de 20 variables sobre característiques clíniques del pacient. El preprocessament està composta per l'eliminació de variables menys informatius, eliminació d'outliers, imputació amb knn i normalització per les variables numèriques i imputació simple afegint una modalitat i transformació a numèrica per les variables categòriques. Els hiperparàmetres del model són: 'C': 3, 'gamma': 'scale', 'kernel': 'linear'.

Citacions

Més informació a: Fleming, Thomas R. i David P. Harrington. Processos de recompte i anàlisi de supervivència. Vol. 625. John Wiley & Sons, 2013.

Ús previst

Ús principal

Amb una finalitat educativa, el present model està creat per a la pràctica de les tècniques i algoritmes apreses a les classes de IAA del departament de FIB de UPC.

Usuaris destinats

- Estudiants de IAA de la FIB de UPC
- Professors de IAA de la FIB de UPC

Ús fora de l'abast

El model creat no està pensat per ser utilitzat amb la finalitat que es proposa en cas real, és a dir, per realitzar prediccions de l'estat de supervivència dels pacients amb cirrosi hepàtica. Un ús no recomanat podria portar al risc vital del pacient.

Factors

Els factors que poden alterar la funcionalitat o el rendiment del model, serien possibles grups de pacients amb les característiques recol·lectades a les variables alterades per altres malalties. Cal una especificació per part de professional mèdics.

Mètriques

Les mètriques que s'han tingut en compte per modelar i avaluar el model són el F1 weighted i el Balanced accuracy. La raó d'aquesta tria és per resoldre els problemes que porta el desequilibri de les dades.
El threshold que s'ha considerat per eliminar les dades outliers és de 3, per tal de millorar el rendiment del model sense perdre grans quantitats de dades, ja que el dataset original en té poques.

Figura 77: Model card

Dades d'avaluació

Les dades emprats per l'avaluació del model han estat escollits de forma aleatòria dintre del data set i conté el 20% de les mostres. Dintre d'aquesta partició de test s'ha eliminat les variables que durant el preprocessament s'han considerat com no informatius. Ha sigut recodificat, ja que el dataset original contenia expressions per identificals els missing values que no es detectaven. També ha sigut normalitzat (les variables numèriques) i transformat a numèrica (les variables categòriques) abans de testejar-lo al model.

Dades d'entrenament

Les dades emprats per l'entrenament del model han estat escollits de forma aleatòria dintre del data set i conté el 80% de les mostres. Dintre hi conté les variables que especifiquen les característiques clíniques del pacient per poder predir la variable 'Status', una variable categòrica amb les modalitats D igual a mort, C igual a censurat i CL igual a censurat per trasplantament hepàtic. Les variables que conté són: ID, N_Dies, Estat, Droga, Edat, Sexe, Ascites, Ascites, Aranyes, Edema, Bilirubina, Colesterol, Albúmina, Coure, Alk_Phos, SGOT, Triglicèrids, Les plaquetes, Protrombina i Etapa.

Rendiment del model

A continuació són els resultats obtinguts sobre la partició de test:

- Balanced accuracy: 0.75
- F1 weighted: 0.84

Consideracions ètiques

Les dades que es conté en la base són informacions de persones reals, per tant, són dades sensibles i no s'ha d'utilitzar per males intensions.

Recomanacions

No utilitzar el model en cap cas sense que el professorat responsable l'hagi validat.

Figura 78: Model card

7 Conclusions i futures millores

El treball s'ha dut a terme amb èxit, complint els objectius inicial que es tractaven de crear un model per predir l'estat de supervivència dels pacients amb cirrosi hepàtica i posar en pràctica les tècniques i algoritmes que s'han après durant el curs.

El model final no té un rendiment extraordinari, però ha permés l'estudi a fons de tot un proces de manipulació de les dades fins a assolir unes propostes.

Com a futures millores, caldria un anàlisi més a fons de les influències d'aplicació de diferents tècniques en el processament. Seria interessant saber com són els models després de realitzar altres tipus de tractament de dades, a part de les que s'han suposat que serien adients durant l'estudi estadistic de les variables.

A part d'això, també seria recomanable un estudi profund posterior de la causa o causes per les quals s'obtenen els valors de les mètriques. S'hauria de fer una anàlisi més completa de com és el rendiment del model vist amb diferents eines.