FIB

Universitat Politècnica de Catalunya

Facultat d'Informàtica de Barcelona

PREDICCIÓ DE SUPERVIVÈNCIA DELS PACIENTS AMB CIRROSI

Introducció a l'Aprenentatge Automàtic

Autor:

Zhihao Chen

28 de desembre de 2023

Índex

1 Introducció

"Cirrhosis Patient Survival Prediction" és una base de dades que proporciona la Universitat de Califòrnia a Irvine (UCI). Conté les informacions necessàries, en concret, variables, per investigar la supervivència dels pacients amb cirrosi, una malaltia de fetge que finalment impedeix que funcioni correctament.

La base de dades conté 418 instàncies i 20 variables, la variable resposta és la que s'anomena 'Status', que representa l'estat de supervivència del pacient. Aquesta variable té 3 modalitats:

- **0:** Mort ('Death')
- **1:** Censurat ('Censored')
- **2:** Censurat per trasplantament hepàtic ('Censored due to liver transplantation')

La metadada que proporciona la base de dades és la següent:

	name	role	type	demographic	description	units	missing_values
0	ID	ID	Integer	None	unique identifier	None	no
1	N_Days	Other	Integer	None	number of days between registration and the ea...	None	no
2	Status	Target	Categorical	None	status of the patient C (censored), CL (censor...	None	no
3	Drug	Feature	Categorical	None	type of drug D-penicillamine or placebo	None	yes
4	Age	Feature	Integer	Age	age	days	no
5	Sex	Feature	Categorical	Sex	M (male) or F (female)	None	no
6	Ascites	Feature	Categorical	None	presence of ascites N (No) or Y (Yes)	None	yes
7	Hepatomegaly	Feature	Categorical	None	presence of hepatomegaly N (No) or Y (Yes)	None	yes
8	Spiders	Feature	Categorical	None	presence of spiders N (No) or Y (Yes)	None	yes
9	Edema	Feature	Categorical	None	presence of edema N (no edema and no diuretic ...	None	no
10	Bilirubin	Feature	Continuous	None	serum bilirubin	mg/dl	no
11	Cholesterol	Feature	Integer	None	serum cholesterol	mg/dl	yes
12	Albumin	Feature	Continuous	None	albumin	gm/dl	no
13	Copper	Feature	Integer	None	urine copper	ug/day	yes
14	Alk_Phos	Feature	Continuous	None	alkaline phosphatase	U/liter	yes
15	SGOT	Feature	Continuous	None	SGOT	U/ml	yes
16	Tryglicerides	Feature	Integer	None	tryglicerides	None	yes
17	Platelets	Feature	Integer	None	platelets per cubic	ml/1000	yes
18	Prothrombin	Feature	Continuous	None	prothrombin time	s	yes
19	Stage	Feature	Categorical	None	histologic stage of disease (1, 2, 3, or 4)	None	yes

Figura 1: Les variables de la base de dades

L'objectiu principal d'aquesta pràctica és construir models estadístics a partir de les altres 17 característiques clíniques, en el cas que siguin necessaris, per predir la variable objectiva, és a dir, predir l'estat de supervivència dels pacients. Concretament, els models que es construiran són tres:

- **KNN (K-Nearest Neighboors):** Veí més proper
- **Decision Tree:** Arbre de decisió
- **SVM (Support Vector Machine):** Màquina de vector de suport

2 Anàlisis i preprocessat de dades

Abans de començar qualsevol construcció de models, és important conèixer l'estructura de la base de dades que en la qual es vol manipular, és a dir, aprendre les distribucions de les variables numèriques i els balancejos de les classes de les variables categòriques.

Un cop acabada l'anàlisi estadística de les variables, el següent procés serà el preprocessament de les dades, per tal de facilitar la construcció de models.

2.1 Recodificació de les variables: Interpretació correcta per Software

Probablement en importar la base de dades al "Notebook" el tipus de les variables canvien, aleshores caldrà recodificar-los per evitar problemes en els procediments posteriors.

Abans de començar res, s'ha detectat a la base de dades una variable que no aporta cap informació útil per investigar la supervicència dels pacients: "ID". El "ID" és l'identificador del pacient, no es necessita perquè es pot identificar-los per l'índex de la fila, aleshores s'ha decidit eliminar directament aquesta variable de la base.

Posteriorment, s'ha creat una funció "data_explore", tal com el seu nom indica, serveix per explorar les dades, veure els tipus de les variables, les modalitats que tenen, etc. D'aquí es pot veure diversos problemes que dificulten la construcció dels models d'inferència.

Primer, comparant els tipus de les variables interprats amb els de la metadada s'ha vist que hi ha unes quantes variables numèriques detectades com a categòriques ("Cholesterol", "Copper", "Tryglicerides", "Platelets"), i una variable qualitativa ordinal detectada com a numèrica ("Stage"). És imprescindible convertir-los als seus formats correctes.

Per altra banda, a algunes variables booleanes ("Drug", "Ascites", "Hepatomegaly", "Spiders") han descobert 3 modalitats, a més de "Y" ("Yes") i "N" ("No"). Això és per causa d'una traducció incorrecta d'alguns valors buits (NA), que han sigut considerats com una cadena de caràcters ("strings"). Per solucionar-lo, s'ha transformat els valors d'aquesta tercera modalitat en NA com hauríen de ser.

2.2 Anàlisis de dades abans de la imputació

Una anàlisi de les variables es realitza abans del preprocessament de dades per tenir una vista prèvia de l'estructura de dades per tal de saber quines variables necessitaran ser tractats durant el procés de preprocessament.

2.2.1 Anàlisi estadístic de les variables numèriques de manera independent

Per les variables numèriques s'ha observat primer el rang de valors i les mesures estadístiques (mitjana, mediana, desviació estàndard) de cada variable:

```
STATISTICAL ANALYSIS OF NUMERICAL DATA
**********************************************************************
```

	count	mean	std	min	25%	\
N_Days	418.0	1917.782297	1104.672992	41.00	1092.7500	
Age	418.0	18533.351675	3815.845055	9598.00	15644.5000	
Bilirubin	418.0	3.220813	4.407506	0.30	0.8000	
Cholesterol	284.0	369.510563	231.944545	120.00	249.5000	
Albumin	418.0	3.497440	0.424972	1.96	3.2425	
Copper	310.0	97.648387	85.613920	4.00	41.2500	
Alk_Phos	312.0	1982.655769	2140.388824	289.00	871.5000	
SGOT	312.0	122.556346	56.699525	26.35	80.6000	
Tryglicerides	282.0	124.702128	65.148639	33.00	84.2500	
Platelets	407.0	257.024570	98.325585	62.00	188.5000	
Prothrombin	416.0	10.731731	1.022000	9.00	10.0000	

	50%	75%	max
N_Days	1730.00	2613.50	4795.00
Age	18628.00	21272.50	28650.00
Bilirubin	1.40	3.40	28.00
Cholesterol	309.50	400.00	1775.00
Albumin	3.53	3.77	4.64
Copper	73.00	123.00	588.00
Alk_Phos	1259.00	1980.00	13862.40
SGOT	114.70	151.90	457.25
Tryglicerides	108.00	151.00	598.00
Platelets	251.00	318.00	721.00
Prothrombin	10.60	11.10	18.00

Per visualitzar els valors anteriors de forma directa, s'ha fet servir els histogrames, on l'eix x és el rang de valor de la variable i l'eix y el nombre d'individus que té un valor dins del rang corresponent. Aquesta eina també mostra la distribució estadística de cada variable.

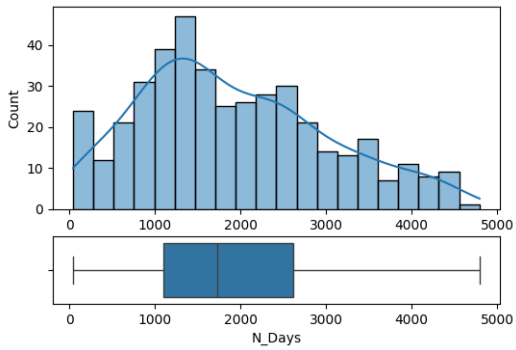


Figura 2: Histograma i boxplot de "N_Days"

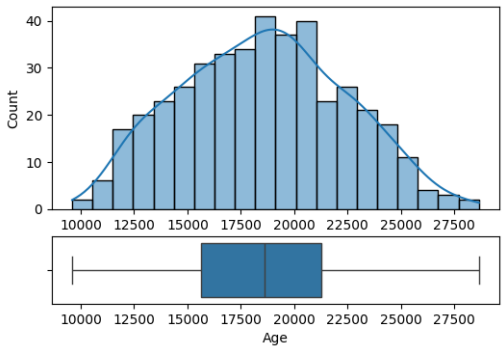


Figura 3: Histograma i boxplot de "Age"

La variable "N_Days" vol dir el nombre de dies que el pacient porta ingressat a l'hospital i té una distribució Dirichlet. La variable "Age" és l'edat del pacient, té una distribució binomial. Cal destacar que la unitat de la variable "Age" és dia en lloc d'any del sentit comú, la qual cosa dificulta la interpretació de les variables, potser requerirà una transformació de dades, però finalment no s'ha decidit fer-lo perquè sinó es perdrien informacions. Dels boxplots anteriors es pot observar que aquestes dues variables no tenen bàsicament valors extrems.

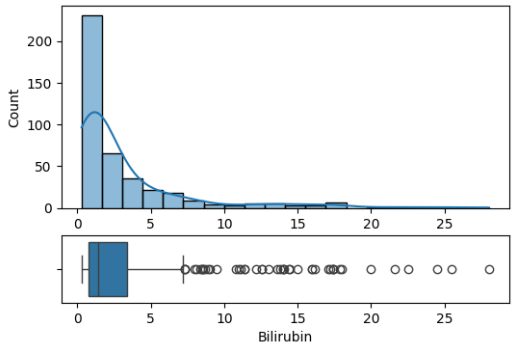


Figura 4: Histograma i boxplot de “Bilirubin”

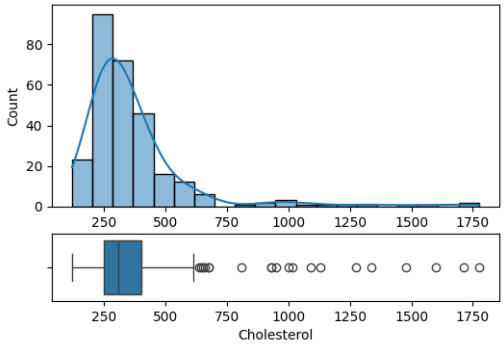


Figura 5: Histograma i boxplot de “Cholesterol”

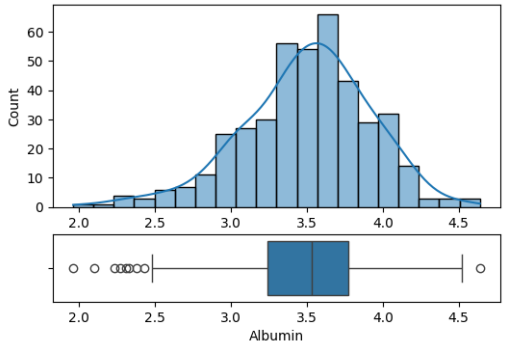


Figura 6: Histograma i boxplot de “Albumin”

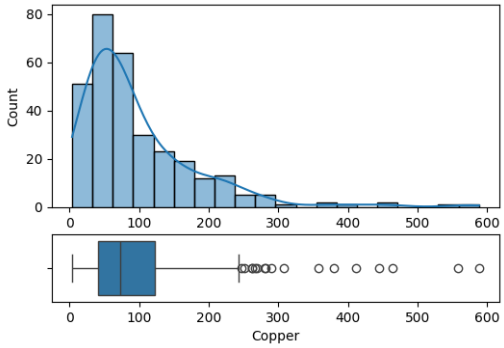


Figura 7: Histograma i boxplot de “Copper”

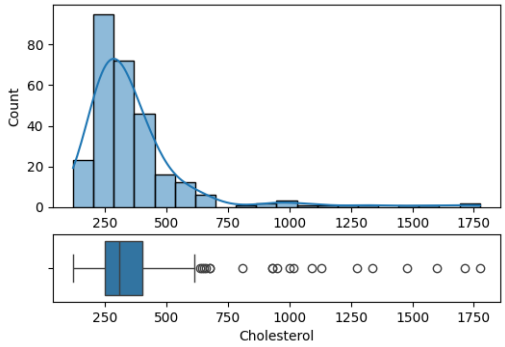


Figura 8: Histograma i boxplot de “Bilirubin”

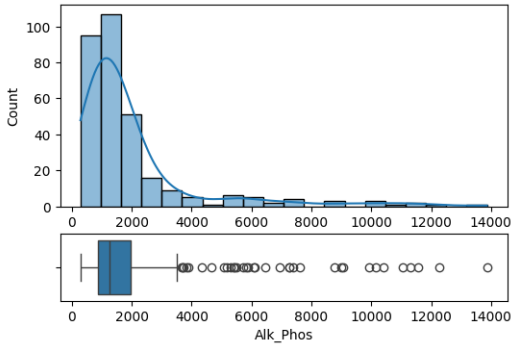


Figura 9: Histograma i boxplot de “Alk_Phos”

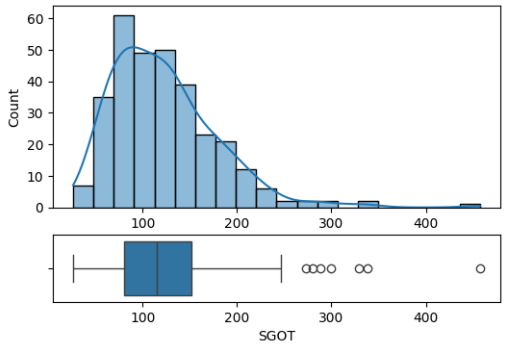


Figura 10: Histograma i boxplot de “SGOT”

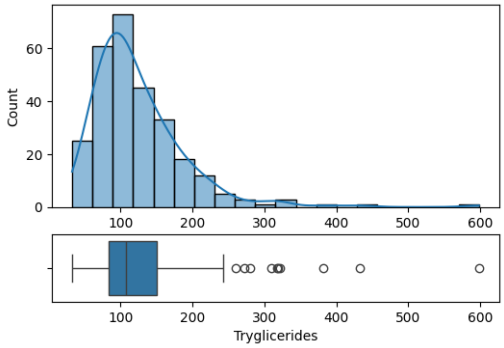


Figura 11: Histograma i boxplot de “Triglicerides”

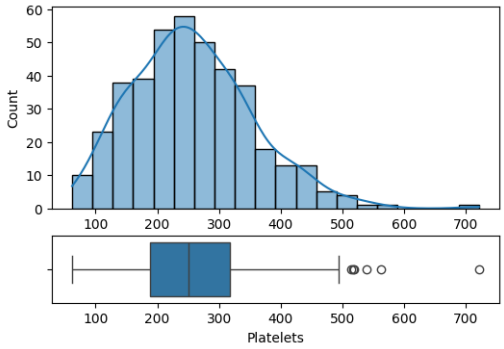


Figura 12: Histograma i boxplot de "Platelets"

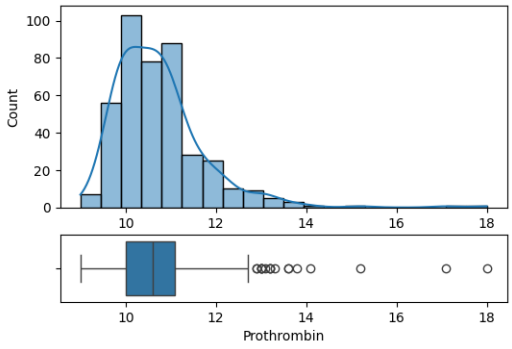


Figura 13: Histograma i boxplot de "Prothrombin"

Les variables anteriors tenen majoritàriament una distribució lognormal, són molt semblants a la distribució gaussiana, apareix campana de Gauss en quasi totes elles, tanmateix, no està en el mig de la distribució, és a dir, la distribució no és asimètrica.

Un possible motiu que va portar a cap aquest efecte és la presència dels "outliers". Totes les variables anteriors són les que tenen una certa quantitat de valors atípics, que han de ser tractats acuradament. Malauradament, el nombre de "outliers" en algunes variables pot arribar fins a aproximadament 50, la qual cosa és més d'una dècima part de la base de dades.

La manera que s'ha utilitzat per identificar els "outliers" és mitjançant el boxplot: totes les mostres que estan fora de l'interval (Q1 - 1.5 * IQR) i (Q3 + 1.5 * IQR) són considerats com valors atípics, sent Q1 i Q3 el primer i el tercer quartil, respectivament, i IQR l'amplitud interquartílica. S'ha decidit utilitzar aquest mètode d'identificació estàndard per simplificació. A causa de no haver profunditzat gaire l'estudi del rang de les variables (potser s'hauria de canviar el llindar perquè els que es vol estudiar són justament els "outliers" que s'ha detectat aquí, són observacions normals en lloc d'anormals produïts pels errors), que requeriran un cert coneixement mèdic.

S'ha provat de veure les distribucions després de l'eliminació dels "outliers" en la base de dades per saber si canvien d'una forma exagerada.

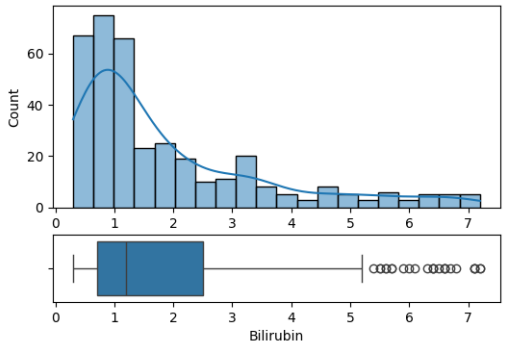


Figura 14: Histograma i boxplot de "Bilirubin"

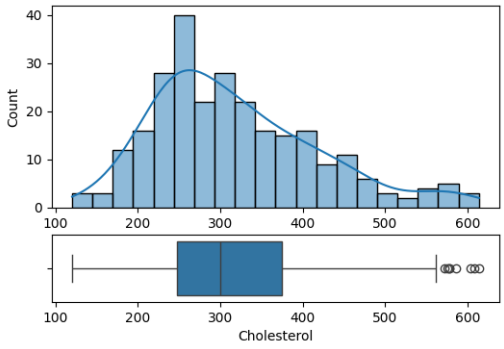


Figura 15: Histograma i boxplot de "Colesterol"

Tal com s'ha observat, l'impacte que porta els valors atípics a les variables numèriques és signifi-

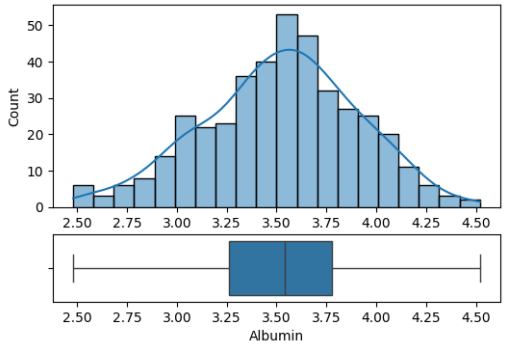


Figura 16: Histograma i boxplot de "Albumin"

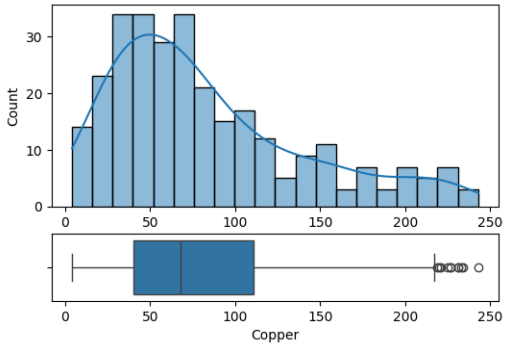


Figura 17: Histograma i boxplot de "Copper"

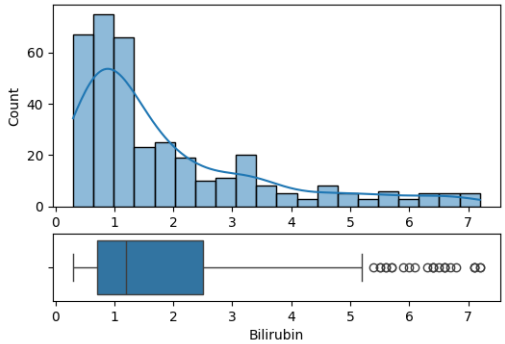


Figura 18: Histograma i boxplot de "Bilirubin"

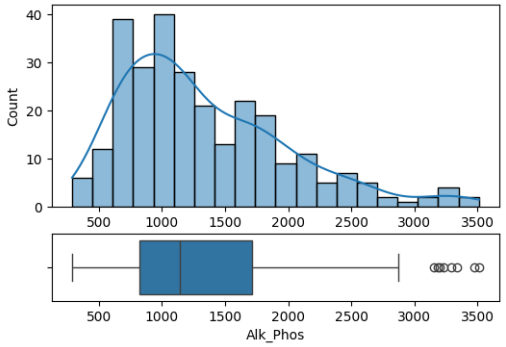


Figura 19: Histograma i boxplot de "Alk_Phos"

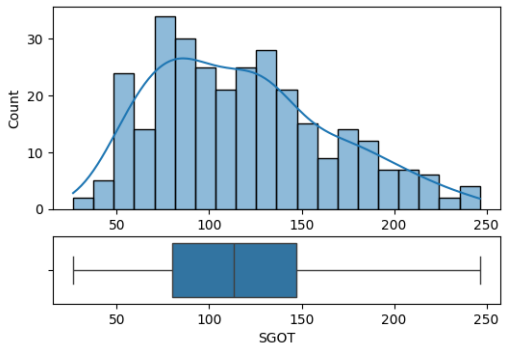


Figura 20: Histograma i boxplot de "SGOT"

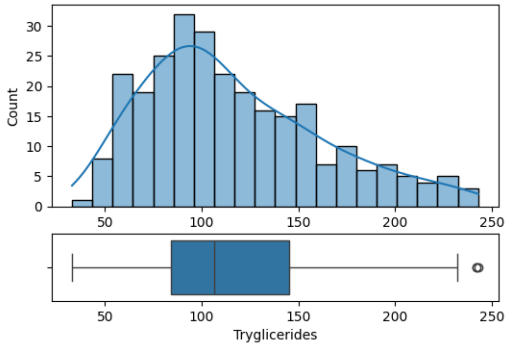


Figura 21: Histograma i boxplot de "Triglicerides"

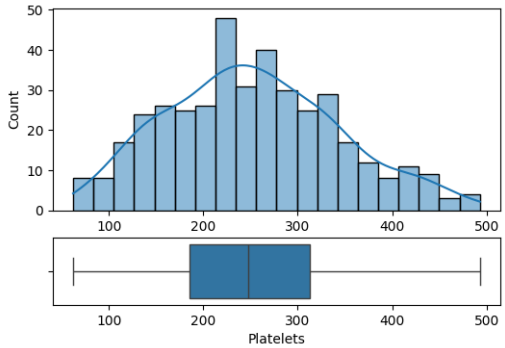


Figura 22: Histograma i boxplot de "Platelets"

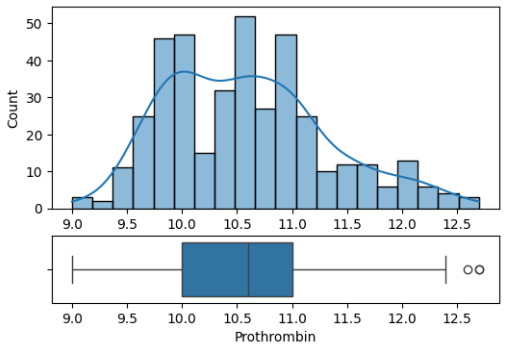


Figura 23: Histograma i boxplot de "Prothrombin"

cant. Després d'eliminar-los, les variables que tenien una distribució lognormal s'han transformat a assemblar-se a una distribució gaussiana perquè són els valors atípics els que va portar la campana de Gauss cap a un dels costats, tal com s'ha esperat.

Tot i que les distribucions de les variables sense "outliers" són majoritàriament gaussianes, que són molt convenients a l'hora de construir alguns models, però els models que es construirà més tard no són de regressió, sinó que de classificació, que no requereixen que les variables numèriques tinguin distribució normal. Per tant, s'ha proposat dos mètodes per tractar els valors atípics:

- No eliminar aquelles files que contenen "outliers" d'alguna variable per conservar les informacions precioses i necessitades.
- Canviar els valors atípics per valors buits i imputar-los, de manera que tampoc reduïrà el nombre de mostres que tenia la base de dades.

Es pensa que la primera proposta serà millor perquè en estudiar una malaltia, tenir característiques anormals sembla que seria més habitual, en altres paraules, els valors atípics aportarien informacions més útils que els normals. Per consegüent, la segona proposta seria més vàlida si es podria identificar realment quins valors atípics són realment causats pels erros de mesura o d'experiment.

2.2.2 Estudi de balanceig de classes de les variables categòriques

Per les variables categòriques s'ha usat diagrames de barres per estudiar el balanceig de cada classe, sent l'eix y el nombre d'individus que pertanyen a alguna classe de la variable i l'eix x la classe corresponent.

Tal com es pot observar en les gràfiques següents, hi ha un desbalanceig obvi de classes en quasi totes les variables. A més el desbalanceig en algunes variables és exagerat, amb una proporció d'1:10 o més.

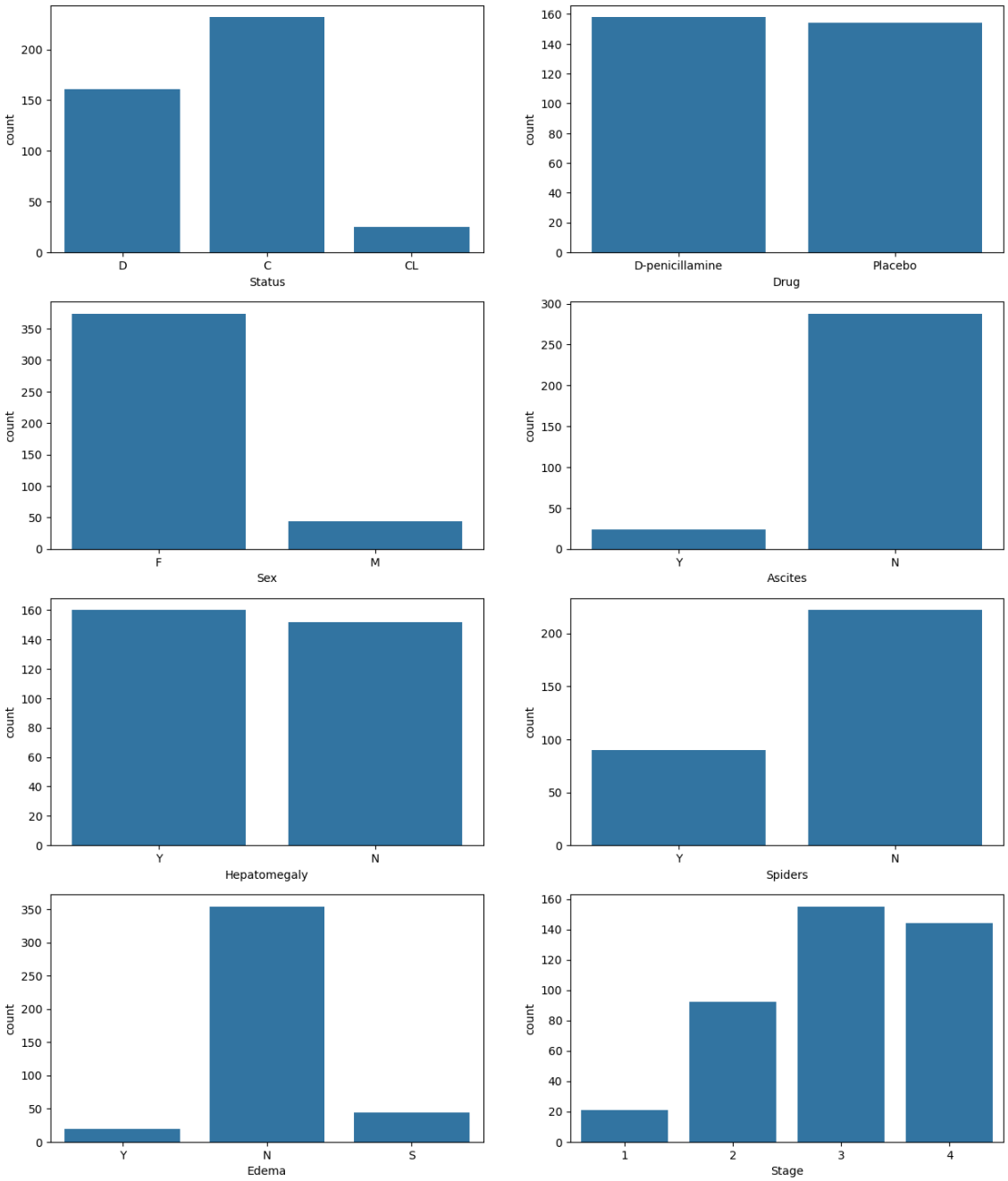


Figura 24: Estudi de balanceig de totes les variables qualitatives

Per solucionar-lo hi ha vàries estrategies, les més comunes són el "Resampling" i l'assignació de pesos a les classes a l'hora de construir models.

La tècnica del "Resampling" pot portar a cap alguns riscos a la construcció de models: mitajançant el "Undersampling" s'abandonaran parts de les informacions de la base de dades, la qual cosa és preciosa; el "Oversampling" duplica els individus de la classe minoritària, que podria augmentar els errors que aporten els sorolls. La tècnica de "SMOTE" ("Synthetic Minority Oversampling TEchnique") és bàsicament un "Oversampling" de versió "Soft", no simplement duplica individus, sinó que crea individus semblants amb una tècnica similar a "KNN" ("K-Nearest Neighbors"), no té el

perill que s'ha esmentat anteriorment, però pot causar "overfitting" perquè construirà models massa complexos. A més, tal com li passa al "KNN", "SMOTE" no és idealment aplicable en una base de dades amb variables categòriques perquè pot generar valors de "one-hot-enconding" intermedis que no corresponen a una classe categòrica concreta. Probablement seria millor fer servir "SMOTENC" ("Synthetic Minority Over-sampling Technique for Nominal and Continuous"), però com que aquest mètode no ha sigut estudiat gaire, s'ha decidit no utilitzar-lo.

Addicionalment, per raó de tenir tantes variables desbalancejades, s'hauria d'aplicar els mètodes de balanceig a cadascuna d'elles i, a més, el nivell de desbalanceig és considerablement alt, augmentarien encara més els riscos comentats (encara que potser els individus que pertanyen a classes minoritàries de diferents variables són els mateixos, és a dir, el fet de "Undersampling" o "Oversampling" no eliminaria o duplicaria tantes mostres...).

Per consegüent, no s'aplicarà cap mètode de balanceig a la base de dades, per conservar totalment les informacions originals, i per enfrontar els problemes que aporten el desbalanceig de dades, s'utilitzarà el mètode d'assignació de pesos a les classes. Per altra banda, encara no s'ha considerat els valors buits de la base de dades, potser que en acabar la imputació es milloraria la condició de desbalanceig.

En el cas que els models construits no obtenen resultats tan desitjats, es tornarà a pensar si valdria la pena aplicar el mètode de "Resampling".

2.3 Particionat del dataset

La partició de dades és fonamental en aquest treball, ja que l'objectiu és la creació de distints models per dur a terme classificacions, per tant, es necessita una part de la base de dades per construir i entrenar models ("train"), i una altra part per predir la variable objectiva mitjançant els models ("test").

Com que en la base de dades hi ha molt poques mostres, s'ha considerat important conservar la major part de les informacions per la creació de models i abandonar la part per a la validació dels models. La proporció entre "trainï "test"ha sigut 8:2, una proporció bastant habitual. Tanmateix, a l'hora d'entrenament es necessitarà mètriques d'avaluació dels models, per realitzar-lo es farà servir el mètode "Cross-Validation", que serà comentat posteriorment.

A partir d'ara, la partició de "test" no es farà servir fins al moment d'avaluació dels models.

	Base de dades original	Partició "train"	Partició "test"
Mida (#files, #columnes)	(418, 19)	(334, 19)	(84, 19)
Percentatge de dades	100%	80%	20%

Taula 1: Mida de les particions

2.4 Imputació de Missings

Per detectar el nombre d'elements buits que tenen les variables es torna a utilitzar la funció "data_explore" que s'havia definit a l'inici. Com que els NAs que són malinterpretats com a una nova modalitat ja són tractats anteriorment, es pot cridar a la funció "data_explore" directament. Sobre la base de dades inicial els nombres i percentatges de "missing values" que es troba són els següents, en ordre descendent:

Variable	**Nombre de "missing values"**	**Percentatge de "missing values"**
Tryglicerides	136	32.54%
Cholesterol	134	32.06%
Copper	108	25.36%
Drug	106	25.36%
Ascites	106	25.36%
Hepatomegaly	106	25.36%
Spiders	106	25.36%
SGOT	106	25.36%
Alk_Phos	106	25.36%
Platelets	11	2.63%
Stage	6	1.44%
Prothrombin	2	0.48%
N_Days	0	0.00%
Albumin	0	0.00%
Status	0	0.00%
Edema	0	0.00%
Sex	0	0.00%
Age	0	0.00%
Bilirubin	0	0.00%

Taula 2: Nombre i percentatge de "missings" per variable en la base de dades original

De la taula anterior s'ha vist que hi ha aproximadament la meitat de variables tenen un percentatge de "missing values" alt (més de 25%), una possible de gestió és eliminar directament aquestes variables, però no és gens racional perquè s'ha de conservar al màxim possible les informacions restants. Per tant, aquesta proposta no serà acceptada. Per la mateixa raó, l'eliminació de les mostres que tenen elements buits tampoc serà presa. Així doncs, només es podrà escollir alguns mètodes de la imputació.

Per a les variables numèriques la imputació amb mitjana podria no resultar tan adient, per l'existència dels valors atípics que s'ha citat anteriorment. Llavors seria millor imputar els valors amb mediana, que evita el problema anterior, ja que agafa el valor intermedi i és robust amb els "outliers". Tanmateix, tenen un incovenient tant la imputació amb mediana com la imputació amb mitjana que és que hi ha massa valors buits en la base de dades, a l'hora d'imputar-los amb mediana, massa mostres es quedaran amb un mateix valor, la qual cosa podria canviar massa la distribució original. Una altra possible manera és la imputació amb l'algorisme de "KNN", és més flexible que la imputació amb mediana, no totes les mostres tindran el mateix valor, sinó que depèn dels seus veïns. És computacionalment costós, però en tenir una base de dades tan petita sembla ser acceptable. No obstant això, "KNN" no és aconsellablement aplicable en una base de dades amb

variables categòriques, per la mateixa raó que l'aplicació de SMOTE, comentada anteriorment.

Per a les variables categòriques es podria imputar amb la moda, però com que hi ha un desbalanceig obvi de classes i un nombre substantiu d'elements buits, després de la imputació el desbalanceig es convertirà més exagerat encara. Per tant, s'ha considerat més aconsellable crear una nova modalitat anomenada "Unknown" pels valors buits. No obstant això, a la variable "Stage" si s'havia creat una altra classe per guardar els elements buits, aleshores es crearia una altra classe minoritària, ja que té poquíssimes NAs. Així doncs, s'ha decidit tractar la variable "Stage" d'una manera diferent: imputar-la amb la moda.

En conclusió, s'ha decidit tres maneres d'imputació diferents segons si la variable és numèrica, categòrica o "Stage". Per les numèriques s'utilitzarà la mediana per imputar i per les categòriques es crearà una nova modalitat anomenada "Unknown" per guardar els valors buits, a excepció de la variable "Stage" que s'imputarà amb la moda.

2.5 Anàlisis de dades després de la imputació

Un cop acabada la imputació de les variables, s'espera que hi hagi canvis significatius a les variables. Així doncs, cal una altra anàlisi de dades perquè entendre quins són els canvis de distribucions de les variables. Cal tenir en compte que les anàlisis es faran sobre les dades de la partició "train", per tant, hi hauria alguns petits canvis igualment encara que no s'hagués fet la imputació.

2.5.1 Anàlisi estadístic de les variables numèriques de manera independent

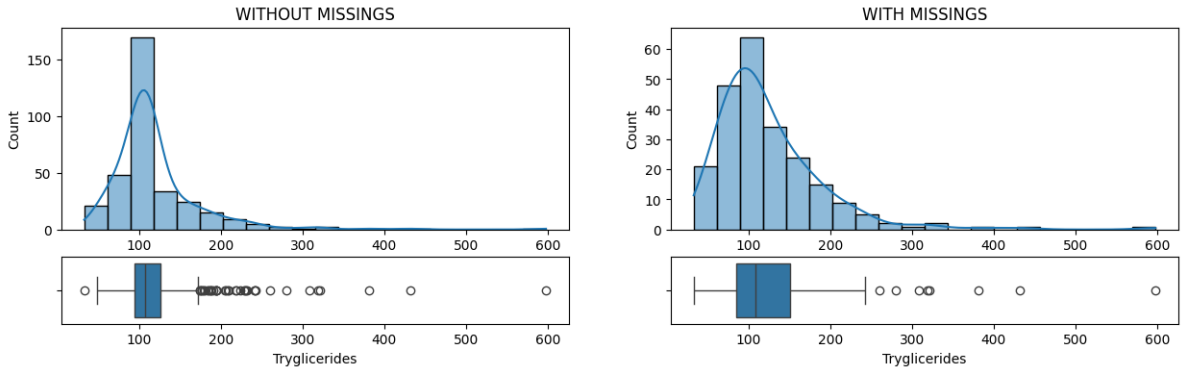


Figura 25: Després i Abans de la imputació de la "Triglicerides"

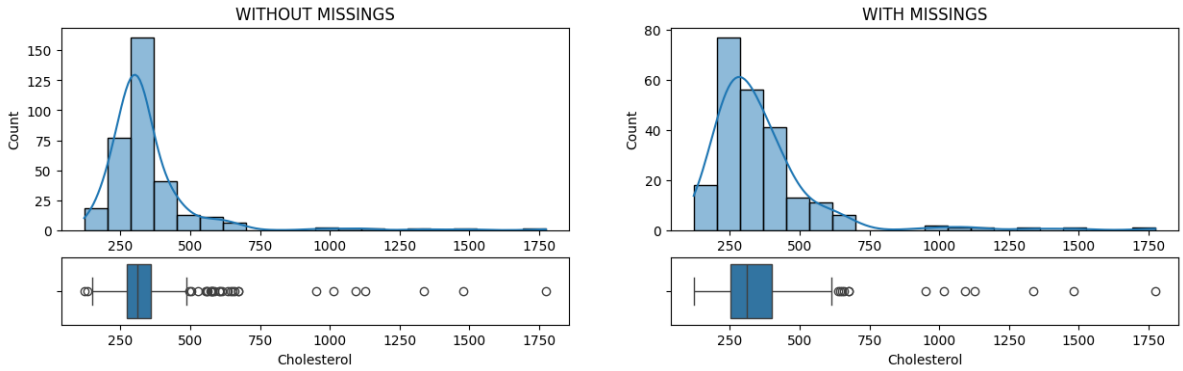


Figura 26: Després i Abans de la imputació de la “Cholesterol”

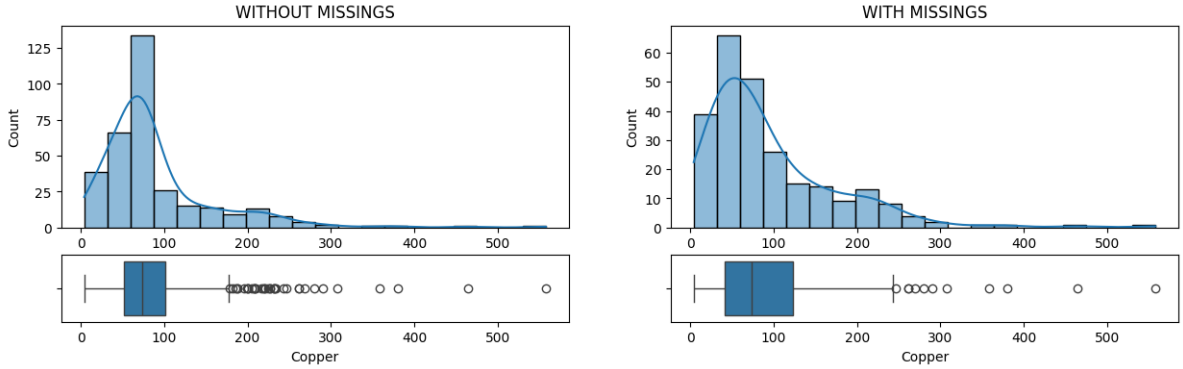


Figura 27: Després i Abans de la imputació de la “Copper”

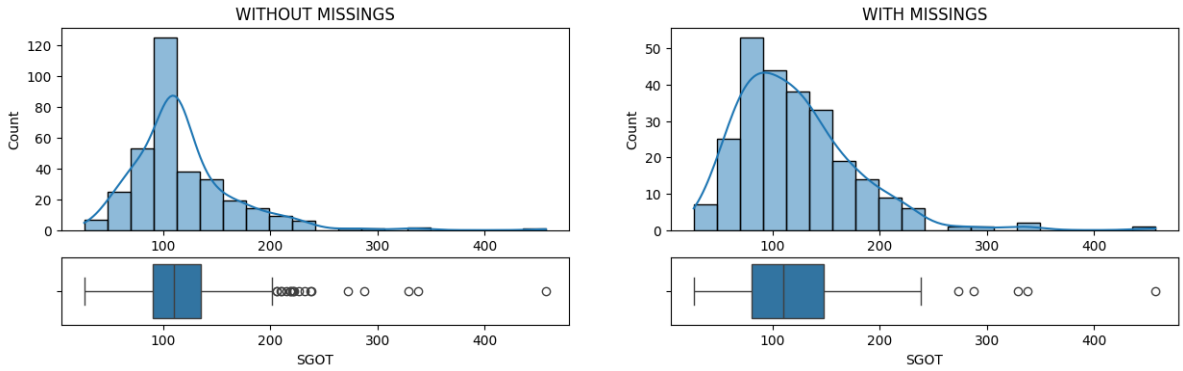


Figura 28: Després i Abans de la imputació de la “SGOT”

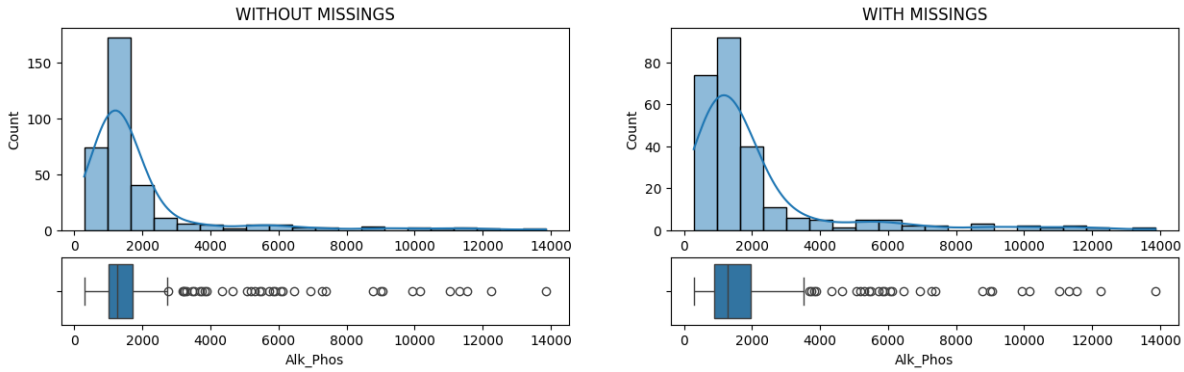


Figura 29: Després i Abans de la imputació de la "Alk_Phos"

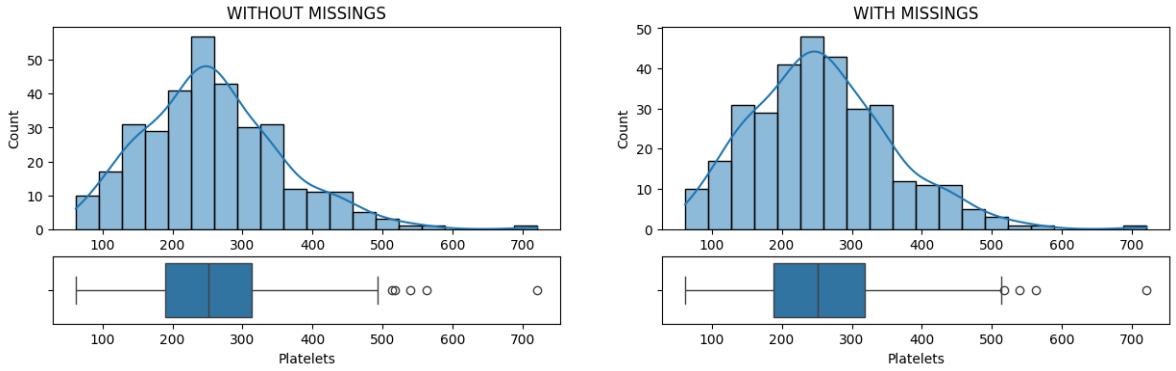


Figura 30: Després i Abans de la imputació de la "Platelets"

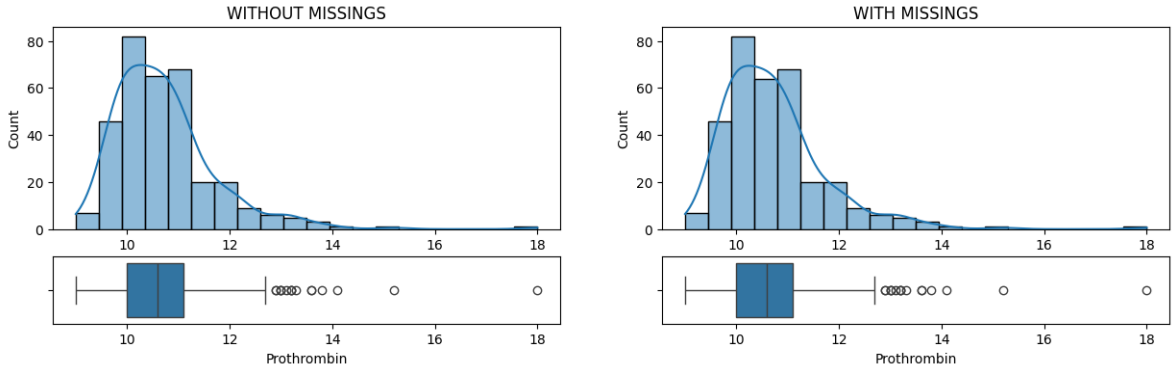


Figura 31: Després i Abans de la imputació de la "Prothrombin"

De les gràfiques anteriors es pot observar la diferència de distribucions entre abans de la imputació (dreta) i després de la imputació (esquerra). Tal com s'ha comentat en l'apartat de preprocessament, en imputar els elements buits per la mediana, moltes mostres es quedaran amb el seu valor i causa l'efecte que es veu en els histogrames: una barra que és obviament més alta que les altres. La incrementació de l'altura de la barra depèn del nombre de "outliers" que tenia la variable, per tant, a la variable "Prothrombin" no es nota gaires canvis.

Cal destacar que després de la imputació les distribucions tendeixen a assemblar-se més a les

gaussianes, tanmateix, com abans, a causa de la presència dels "outliers", la campana de Gauss no apareix en el mig de la distribució.

Un impacte inesperat ha sigut l'augmentació dels nombres dels valors atípics, la forma del diagrama de caixa també ha canviat després de la imputació, aleshores més individus han sigut considerats com valors atípics.

2.5.2 Estudi de balanceig de classes de les variables categòriques

Al procés del preprocessament s'ha creat una nova modalitat de classe anomenada "Unknown". La quantitat de mostres que pertanyen a aquesta nova classe depèn del nombre d'elements buits que tenien les variables. Afortunadament, a la majoria de les variables categòriques hi apareixen un nombre extensiu de valors buits, per això el nou desbalanceig de classes no ha sigut tan greu. Per a la variable "Stage" s'ha imputat els valors buits per la moda, que és la classe "3", per això no hi ha hagut una nova barra al seu diagrama.

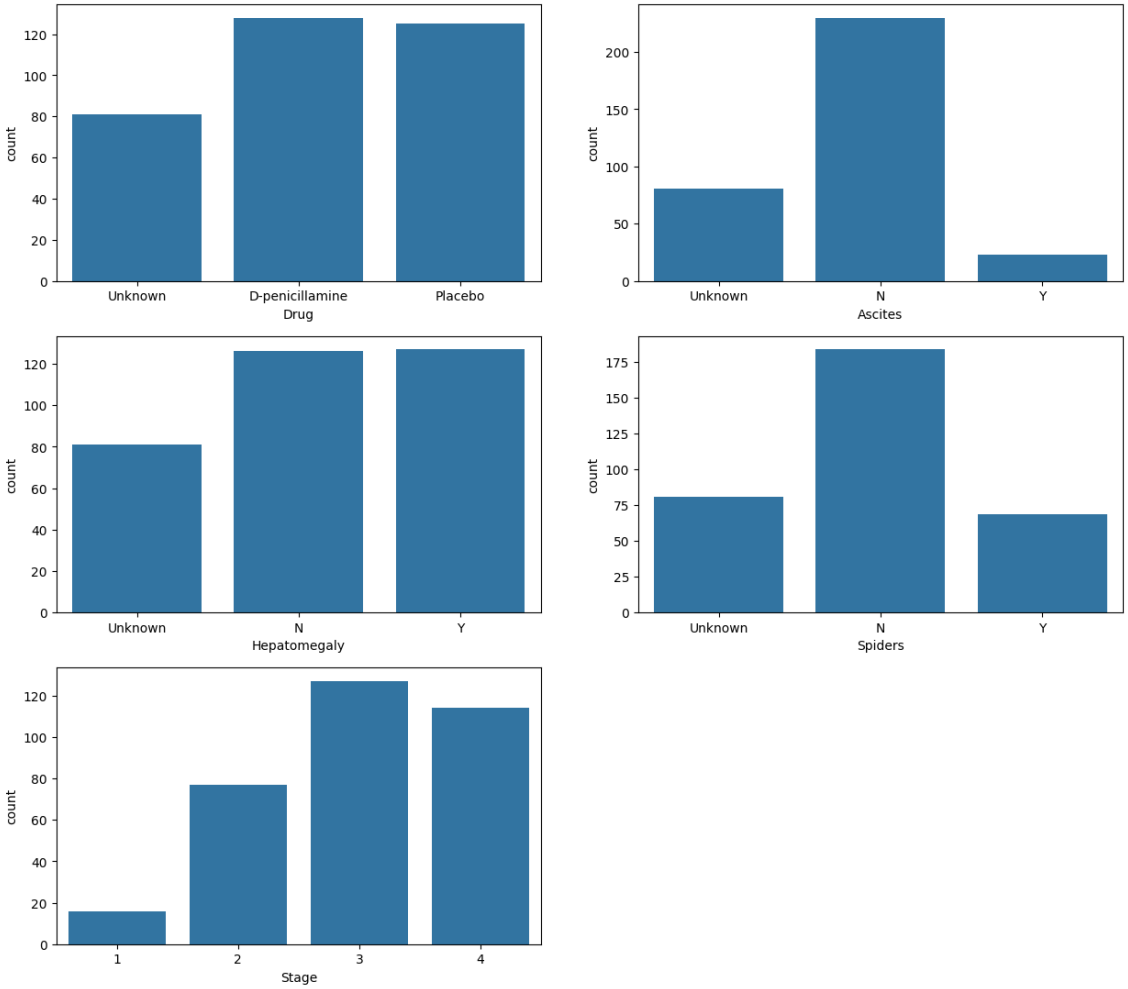


Figura 32: Estudi de balanceig de les variables qualitatives sense "missings"

2.6 Recodificació de les variables

La recodificació de variables fa referència al procés de modificar o transformar els valors d'una variable per satisfer certs requisits. En el cas de la construcció dels models com "KNN", "SVM", que només manipulen sobre variables numèriques, es necessari codificar les variables categòriques presentes a tenir forma numèrica.

Com que la majoria de les variables no són ordinals (només "Stage" es ordinal), aplicarà "one-hot-enconding" a la resta de variables categòriques, que creen noves columnes binàries per mantenir la informació original. I per la variable "Stage" utilitzar "ordinal-enconding". Això també beneficiarà a l'hora de utilitzar "pipeline" perquè la imputació de "Stage" també és diferent a les altres variables, llavors es podria crear un tractament especial només per a la variable "Stage" i no caldrà posar condicions addicionals dins del tractament per a les variables categòriques.

Els arbres de decisió són capaços de manejar directament variables categòriques en la forma original sense requerir codificació addicional, pel seu entrenament només es farà "ordinal-enconding" ja que només canvia les modalitats per números sense crear noves variables.

3 Preparació de variables

Fins a aquest punt ja es coneix l'estructura bàsica de la base de dades, el procés següent serà la preparació de les variables per a la definició dels models.

3.1 Normalització de les variables

Com que en la futura construcció de models inclou la definició de "SVM" i "KNN", que depenen de les mesures de distància, aleshores és imprescindible l'escalat de les característiques per normalitzar el rang de les variables. El mètode de l'escalat ha sigut l'estandarització, que centra les variables al voltant de zero i les escala dividint per la desviació estàndard. Això pot canviar el significat de les variables discretes com "Age" o "N_Days", tanmateix s'ha decidit aplicar-lo igualment per assegurar l'obtenció dels resultats desitjats en el "SVM" i "KNN". No obstant això, l'arbre de decisió no requereix la normalització de les variables, per això no s'inclourà aquest procediment per a la seva definició.

Les variables normalitzades tenen les formes que presenten les figures 33 - 44. Tal com s'observa, el rang de la variable queda reduït per tal que totes elles tinguin el mateix pes a l'hora d'entrenar els models de "SVM" i "KNN" mentre que la distribució continua sent la mateixa que abans.

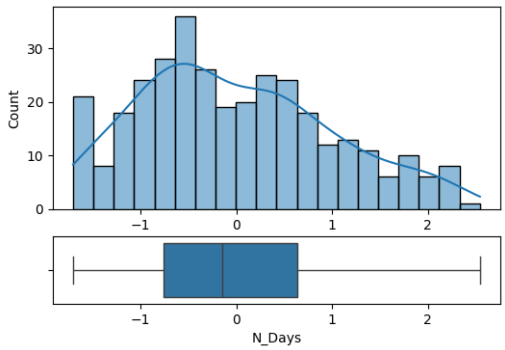


Figura 33: Histograma i boxplot de "N_Days" normalitzada

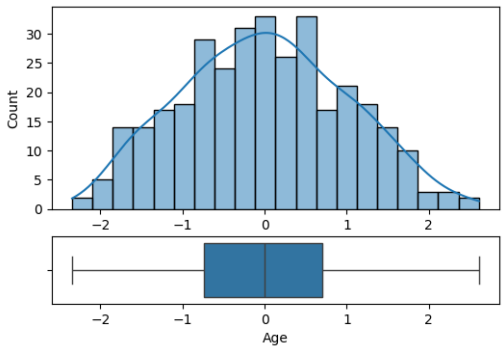


Figura 34: Histograma i boxplot de "Age" normalitzada

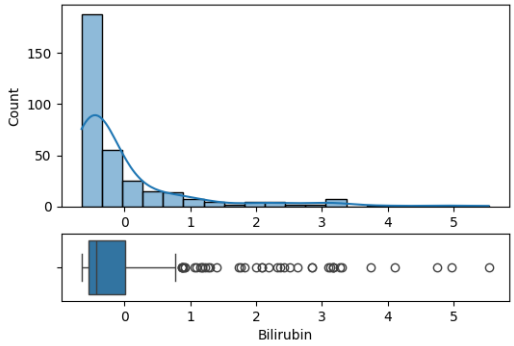


Figura 35: Histograma i boxplot de "Bilirubin" normalitzada

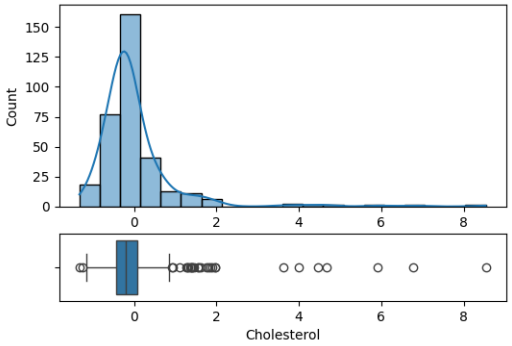


Figura 36: Histograma i boxplot de "Cholesterol" normalitzada

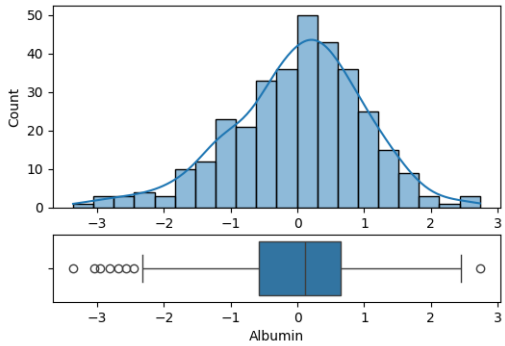


Figura 37: Histograma i boxplot de "Albumin" normalitzada

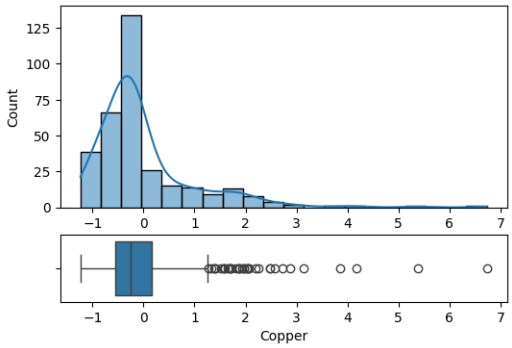


Figura 38: Histograma i boxplot de "Copper" normalitzada

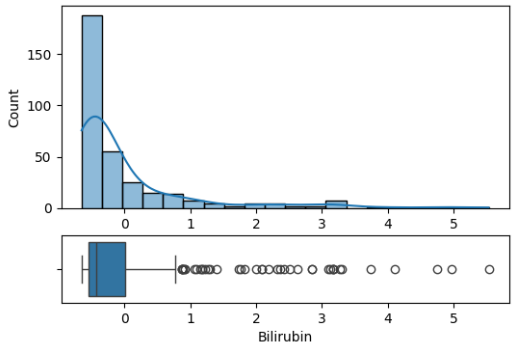


Figura 39: Histograma i boxplot de "Bilirubin" normalitzada

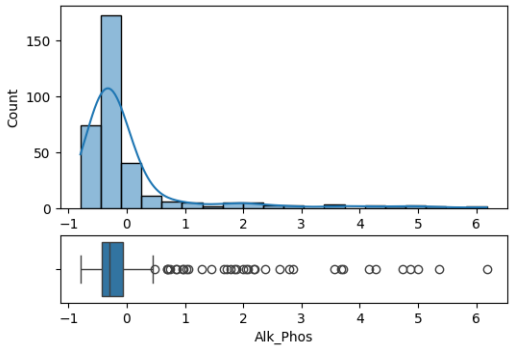


Figura 40: Histograma i boxplot de "Alk_Phos" normalitzada

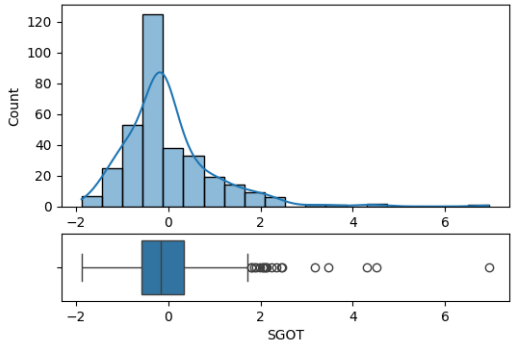


Figura 41: Histograma i boxplot de "SGOT" normalitzada

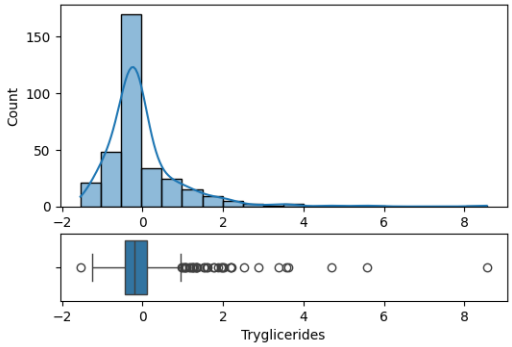


Figura 42: Histograma i boxplot de "Triglicerides" normalitzada

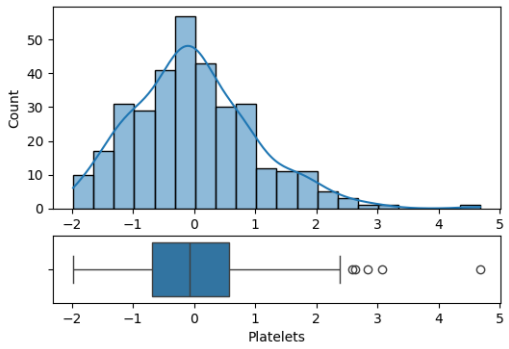


Figura 43: Histograma i boxplot de "Platelets"

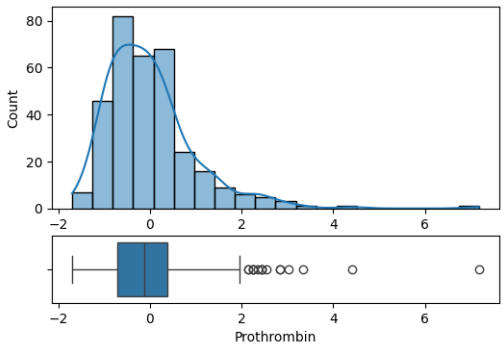


Figura 44: Histograma i boxplot de "Prothrombin"

3.2 Eliminació de variables redundants o sorolloses

En eliminar variables redundants, se simplifica el model. Els models més senzills solen ser més fàcils d'interpretar i generalitzar. A més, reduir la dimensionalitat pot millorar l'eficiència computacional i reduir el risc d'"overfitting" especialment en conjunts de base de dades petits com aquest.

3.2.1 Anàlisi de correlacions entre variables númeriques

La correlació estadística és una mesura estadística que indica la força i la direcció d'una relació lineal entre dues variables aleatòries. Quan la correlació entre dues variables apropa cap al valor absolut d'1, aleshores quan una d'elles augmenta, l'altra augmenta o disminueix de la mateixa manera, depenent del signe de la correlació.

La matriu següent indica les correlacions entre qualsevol parella de variables:

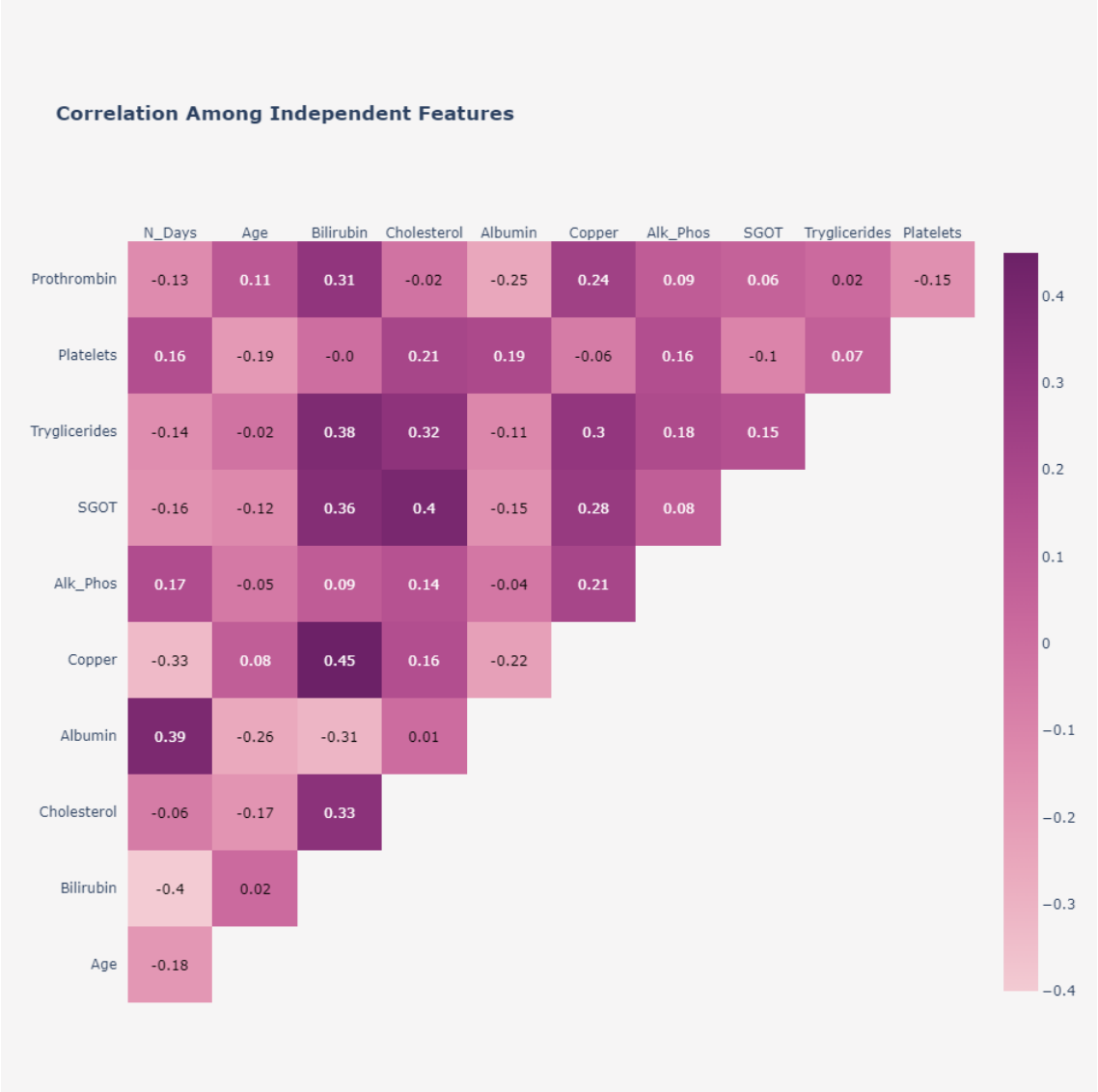


Figura 45: Matriu de correlació entre les variables numèriques

Un dels problemes que existeixen a la definició dels models és la multicolinealitat, que es refereix a l'alta correlació entre dues o més variables predictores en un model, la qual cosa vol dir la alta similitud de les informacions aportades. Pot causar la inestabilitat del model i la disminució de la precisió de les estimacions.

Per exemple, l'algorisme de "KNN" pateix multicolinealitat perquè suposa que cada punt es pot representar com una coordenada en un espai multidimensional. No mesura la quantitat d'informació útil i tracta aquestes variables de característiques com el mateix. Per tant, és concebible que els punts de dades entre dues característiques altament correlacionades s'agrupin al llarg d'una línia, interferint així amb la distància interdimensional, és a dir, les instàncies poden semblar més properes del que realment són en termes de similitud real.

En general totes les correlacions són baixes, totes les variables tenen una certa quantitat d'informacions útils, per tant, no s'eliminarà cap variable numèrica.

3.2.2 Anàlisis de variables categòriques i variable objectiu

En aquest punt s'ha decidit utilitzar les variables categòriques abans de la recodificació per no tenir una anàlisis massa complexa ja que sinó generaria una gràfica per cada nova variable creada després de dur a terme la recodificació. Tanmateix, els tractaments si que seran aplicats a la base de dades amb la recodificació realitzada, eliminant tantes columnes com calgui.

Per analitzar la relació entre les variables categòriques i la variable resposta s'ha fet servir els diagrames de barres, dividint una de les barres per tres, que són les tres modalitats de la variable resposta.

Tal com es veu, en la variable "Drug" sembla que és totalment independent a la variable resposta, sigui quin sigui el tipus de fàrmac que utilitza el pacient, la proporció dels resultats finals (l'estat de supervivència) és quasi la mateixa. En atenció a això, la variable "Drug" ha considerat com variable sorollosa sobre la variable objectiu. A més, té una quantitat significant de valors buits, ja que la classe "Unknown" ocupa quasi un terç de les mostres. El tractament d'aquesta variable ha sigut una eliminació directa.

Per les altres variables, sembla que cadascuna provenen diferents missatges sobre la resposta, encara que pateixen del desbalanceig de classes. A causa de la manca d'informacions que té la base de dades, es conservaran totes elles per a la construcció dels models.

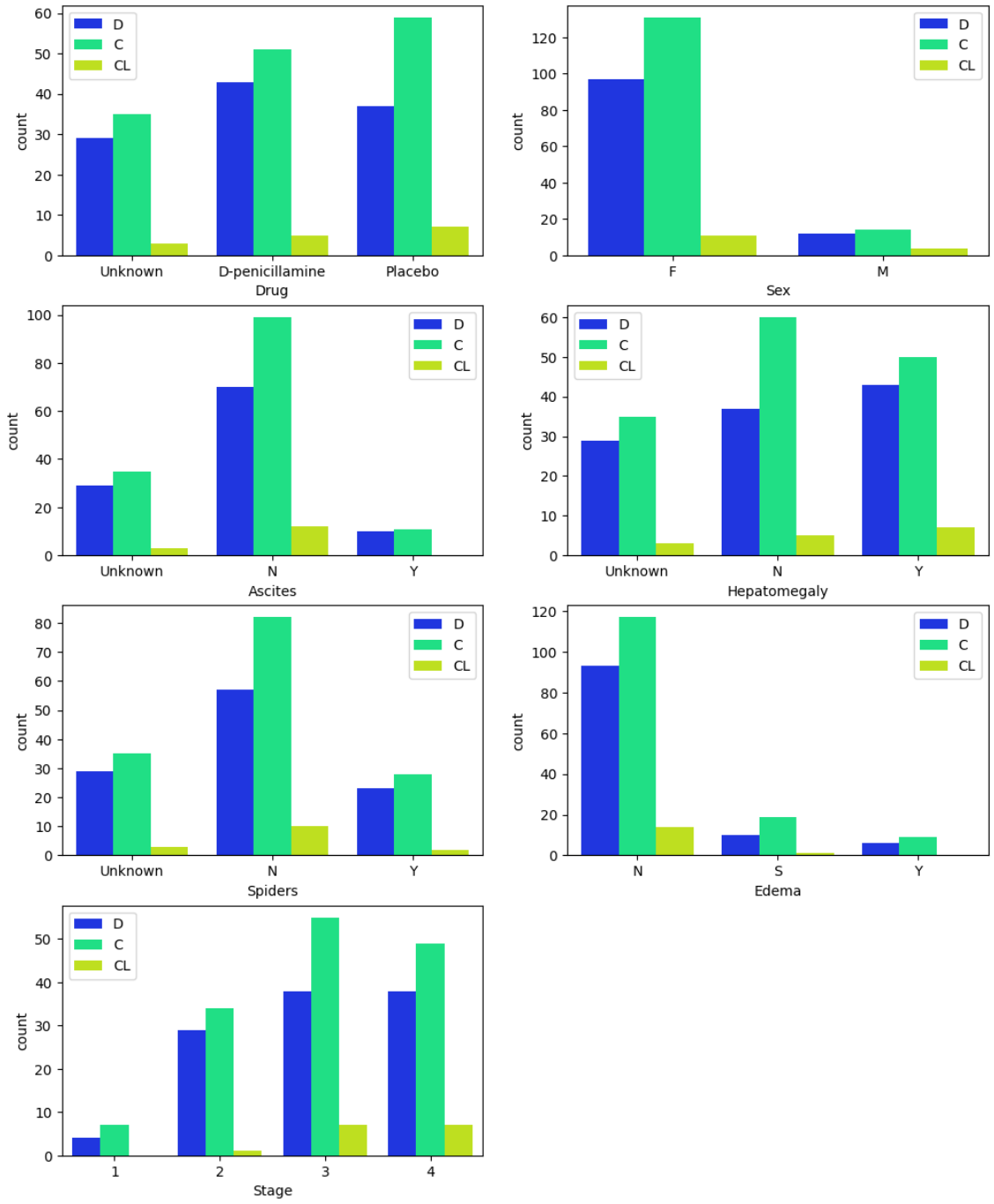


Figura 46: Barplots de les variables categòriques classificant les barres per les classes de "Statuts

3.3 Estudi de dimensionalitat amb PCA

La base de dades que s'està estudiant té més de 3 variables independents, per tant, és necessari aplicar algun algorisme de reducció de dimensionalitat per visualitzar l'estructura de les dades.

L'anàlisis de components principals permet la reducció de la dimensionalitat de la base de dades per tal de poder visualitzar d'una manera directa la projecció de les mostres. Consiteix principalment en modificar la base formada per les variables originals, per tal d'obtenir els components principals

que conserven la major quatitat possible de la variància.

Cal destacar que la "PCA" només pot manipular sobre les variables numèriques (com molts altres algorisme), era per aquesta raó s'havia fet una recodificació de variables categòriques.

La variància obtinguda per cada component principal és limitada, quan més components principals s'afegeix, menys incrementació de variància s'obté. Concretament, la variància explicada per cada components i l'augmentació de la inèrcia tenen la forma següent:

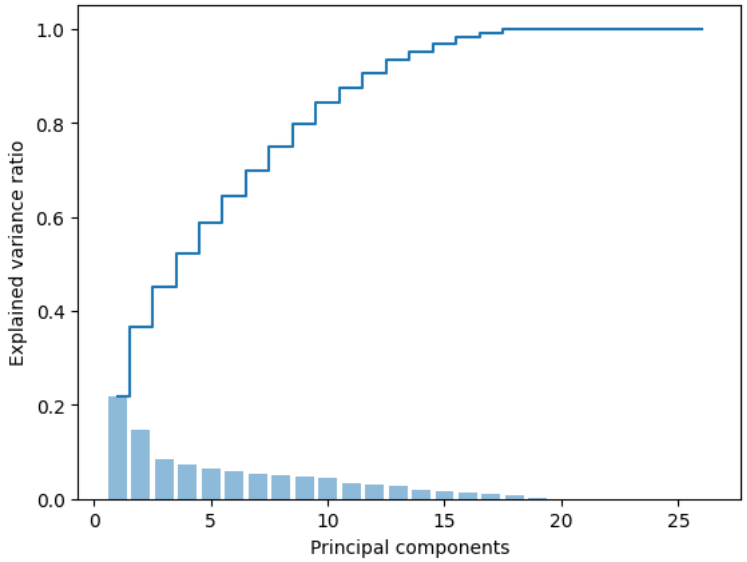


Figura 47: Variància explicada per cada components i l'augmentació de la inèrcia

Segons la imatge anterior, la variància explicada pels dos primers components principals (creen una base que pot ser visualitzada pels humans) és un 36.52%. Crea unes coordenades com la figura 48 i es pot projectar les mostres sobre aquestes coordenades:

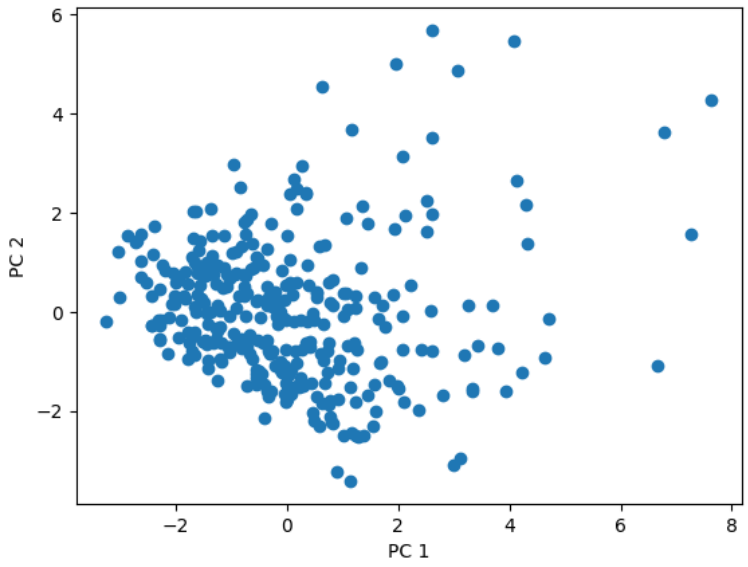


Figura 48: Projecció dels punts de "train" sobre dos components principals

Amb la projecció anterior es pot identificar patrons importants en la base de dades: la major part dels punts es concentren en un lloc i hi ha alguns altres que es distribueixen a tots els altres llocs. Si s'aplica la classificació sobre les dades, probablement es veurà que els punts dispersos pertanyeran en una o més classes i els punts agrupats a un altre.

Sobre la necessitat de reducció de variables, després de anàlisis de componenets principals es veu de la Figura 47 que els últims components principals ja no aporten gaire informacions, però s'ha de usar més o menys 15 components per poder explicar un 90% de variància. Per tant, la resposta és negativa, perquè augmentació dels components principals és deguda a la recodificació de les variables categòriques; abans de dur a terme només hi havia 17 variables i es podia explicar 100% de variància. Sembla que val més la pena no realitzar la recodificació one-hot-enconding que substituir les presents variables pels components principals que conserven major part de la inèrcia.

4 Definició de models

Després d'haver acabat l'anàlisis de dades i preparació de variables ja es pot començar la part més important del treball: la definició de models. S'ha d'entrenar tres models de classificació: Un "KNN", un arbre de decisió i un "SVM" .

4.1 Definició de mètriques

Hi ha diverses mètriques d'avaluació pels models de classificació, les més comunes són:

- **Accuracy:** la proporció de prediccions correctes sobre el total de prediccions
- **Precision:** la proporció de veritables positius sobre la suma de veritables positius i falsos positius
- **Recall:** la proporció de veritables positius sobre la suma de veritables positius i falsos negatius
- **F1-score:** la mitjana harmònica de precisió i recall

La mètrica principal que s'escollirà per avaluar els models serà l'exactitud ("accuracy"). La raó del qual és que s'interessa que el model pugui classificar correctament al màxim possible, ja sigui veritables positius o veritables negatius, perquè és el que avalua el rendiment general del model. No obstant això, l'exactitud no és tan adequat en tenir una base de dades desequilibrades, per tant, es farà servir l'exactitud balancejada, que no té el problema anterior.

Una altra mètrica interessada és la "F1-score" ja que tant la precisió com la sensibilitat són importants en problemes mèdics. La "F1-score" és útil quan es desitja avaluar el rendiment del model a totes les classes de manera equitativa, sense donar més pes a cap classe en particular. És especialment rellevant quan hi ha desequilibris en la distribució de classes, tal com presenta la base de dades que s'està estudiant. Com que les classes de la variable resposta són desequilibrades entre si, aleshores es utilitzarà "F1-score" ponderada.

Per a l'entrenament dels models s'ha decidit utilitzar l'algorisme de "k-Fold Cross Validation" per substituir la partició de "validation", per tant, en les definicions de models posteriors, la mètrica d'avaluació serà sobre la partició de train i la mitjana obtinguda per "Cross Validation", per tal de observar si s'ha fet un "overfitting" al model.

4.2 K-Nearest Neighboors

El primer model és "K-Nearest Neighboors", consisteix en trobar k veïns més semblants per poder predir la nova mostra incorporada. És un model basat en distància, per tant, l'escalat de les variables independents és significant.

Addicionalment, l'algorisme de "KNN" té una interpretabilitat regular, a diferència d'alguns models, com ara arbres de decisió, que generen regles explícites, KNN no genera un model explícit

que pugui ser fàcilment interpretat. El procés de presa de decisions es basa en la proximitat i no en regles específiques.

A més, per poder aprofitar també les variables categòriques per calcular les distàncies, s'ha de convertir-les en variables binàries, cosa que genera més dimensió i, conseqüentment, "KNN" tendeix a perdre eficàcia ja que la noció de "proximitat"pot tornar-se menys significativa a causa del fenomen conegut com la "maledicció de la dimensionalitat".

4.2.1 Discussió dels hiperparàmetres

Els hiperparàmetres de "KNN' hi ha molts, però els més bàsics són:

- el nombre de veïns (n_neighbors)
- els pesos dels veïns (weights)
- la mètrica distància (p & metric)

Per a la mètrica de distància s'ha usat la que vé per defecte: la distància euclidiana perquè és la més senzilla i coneguda. Pels nombre de veïns s'ha de provar amb valors petits, ja que la base de dades és petita i en configurar un nombre de veïns el model pot no funcionar correctament. Pels pesos de veïns només hi ha dos valors disponibles: "uniform" significa que tots els veïns tenen un pes uniforme, i "distance" que vol dir els veïns més propers tenen més influència.

En resum, els hiperparàmetres que es provarà seran els següents:

Hiperparàmetre	**Valors provats**
n_neighbors	{1, 2, 3, 4, 5}
weights	{"uniform", "distance" }
p	2
metric	"minkowski"

Taula 3: Hiperparàmetres i valors provats en KNN

Per trobar la combinació de hiperparàmtres més óptima s'ha aprofitat l'algorisme de GridSearchCV, configurant CV a 10, és a dir, dividir la partició de "train" en deu parts.

4.2.2 Primer entrenament amb "train"

En el primer entrenament s'ha provat amb la combinació de hiperparàmtres que vé per defecte, és a dir, n_neighbors = 5, weights = "uniform", p = 2, metric = "minkownski". Els mètodes de preprocessament són els que s'ha comentat al llarg del treball: imputació per mediana i estandarització a les variables numèriques, imputació per una nova modalitat i "one-hot-enconding" a les variables categòriques no ordinals i imputació per moda i "ordinal-enconding" a les variables ordinals.

No obstant això, una incidència inesperada va dificultar l'entrenament del model: el model KNN no té un hiperparàmetre per balancejar les classes desequilibrades, tal com fan els altres models. Així

doncs, després de provar el model sense cap tractament a les classes desequilibrades, que no va treure resultats gens bons, es va aplicar mètodes de "Resampling". Es va provar dos mètodes: "RandomOverSampling" i "SMOTE + Tomek links". S'ha observat que els resultats de "RandonOverSampling" amb els hiperparàmetres que venen per defecte han sortit millor, però continua sent desagradable. Així doncs, s'ha provat també de convertir els "outliers" per NAs i imputar-los i de trobar els millors hiperparàmetres amb GridSearchCV, però de totes les maneres l'exactitud balancejada és molt baixa en "val". Aleshores s'ha concluït que el model "KNN" no és el més apropiat per enfrontar aquesta base de dades.

	precision	recall	f1-score	support
C	0.89	0.71	0.79	227
CL	0.10	0.40	0.16	5
D	0.59	0.75	0.66	102
accuracy			0.72	334
macro avg	0.52	0.62	0.54	334
weighted avg	0.78	0.72	0.74	334

Figura 49: Resultats del primer model KNN

La imatge anterior reflexiona els entrenaments del model de "KNN". És el resultat del primer model entrenat. Tal com s'ha vist, fins a aquest moment encara no s'ha fet la tècnica de "Resampling", per tant, la majoria de les mostres no són classificades a la classe "CL", que és minòritaria. L'exactitud no és gens baixa, però no és fiable.

	precision	recall	f1-score	support
C	0.63	0.76	0.69	151
CL	0.35	0.09	0.15	76
D	0.56	0.69	0.62	107
accuracy			0.59	334
macro avg	0.51	0.52	0.49	334
weighted avg	0.54	0.59	0.54	334

Figura 50: Resultats del segon model KNN

Ara bé, després d'aplicar el mètode de "SMOTE" el model pot classificar mostres a qualsevol de les classes, tanmateix, totes les mètriques l'avaluació són molt baixes. Els models posteriors també tenen el mateix comportament...

4.2.3 Anàlisi de resultats

La taula següent mostra les mètricques que han obtingut tots els models de KNN definits durant l'entrenament:

	Balanced accuracy VAL	F1-score (W) VAL	Balanced accuracy TRAIN	F1-score (W) TRAIN
KNN	0.531929	0.712498	0.598464	0.780986
KNN + SMOTENC	0.472390	0.586957	0.868571	0.586460
KNN + Oversampling	0.511676	0.656618	0.843352	0.543019
KNN + Oversampling + RemovingOutliers	0.528630	0.621788	0.848816	0.431310
KNN + Oversampling + RemovingOutliers + GridSearchCV	0.488686	0.654316	1.000000	0.431310

Figura 51: Resultats de tots els models KNN

Tal com s'observa, en aplicar els mètodes de "Oversampling", el resultat tret per la partició "train" és generalment millor que el resultat tret per "validation", sobretot l'exactitud balancejada, que pot arribar a ser el doble en la "train" que en la "val".

Aquest risc d'"overfitting" ja ha sigut esmentat anteriorment i és esperat, era per aquesta raó s'havia decidit donar més pes a les classes minoritàries en lloc de portar a cap "Resampling". A més, el mètode de "Undersampling" podrien eliminar massa mostres que té la base de dades, coses que són precioses, per tant, és directament descartat de la prova.

A més a més, 'eliminació dels "outliers" i la configuració dels millors hiperparàmetres trets pel "GridSeach", on n_neighbors=3, weights='distance', tampoc milloren gaire el rendiment del model.

Les puntuacions ("Balanced Accuracy") tretes per cada configuració de hiperparàmetres són les següents:

	n_neighbors	weights	score VAL	score TRAIN
0	1	uniform	0.847466	1.000000
1	1	distance	0.847466	1.000000
2	2	uniform	0.800487	0.920463
3	2	distance	0.847466	1.000000
4	3	uniform	0.809649	0.904060
5	3	distance	0.855166	1.000000
6	4	uniform	0.796589	0.859949
7	4	distance	0.849513	1.000000
8	5	uniform	0.765595	0.839107
9	5	distance	0.827583	1.000000

Figura 52: Puntacions dels models KNN en funció dels hiperparàmetres

S'observa que els valors de les puntuacions són molt més altes que l'exactitud balancejada calculada a la figura 49 sobre la partició "val". Aquest fet és causat per tenir un "refit=True" en els paràmetres que rep l'algorisme de "GridSearchCV"

Per altra banda, s'ha vist que generalment sigui quin sigui la combinació d'hiperparàmetres, el model sempre fa un "overfitting" (la puntuació en "train" és major que en "test"). Se considera la complexitat és deguda a la tècnica de "Resampling", però sense "Resampling" el resultat no seria gaire confiable a causa de "Bias".

4.3 Support Vector Machine

El segon model de classificació és la SVM ("Support Vector Machine"). La idea fonamental darrere de les SVM és trobar un hiperplà òptim que pugui separar de la millor manera possible dos conjunts de dades en un espai de característiques d'alta dimensió.

La limitació més rellevant de la "SVM" és que s'ha de seleccionar una forma de hiperplà (nucli) que millor classifiqui les dades, que forma part dels hiperparàmetres del model. A més, les "SVM" poden ser més difícils d'interpretar i entendre en comparació amb models més simples. La funció de decisió a l'espai de característiques pot ser complexa, especialment amb nucli no lineals.

4.3.1 Discussió dels hiperparàmetres

Els hiperparàmetres més coneguts d'una "SVM" són:

- Paràmetre de regularització (C)
- Tipus de nucli (kernel)
- Coeficient del nucli (gamma)

El paràmetre de regularització controla l'equilibri entre dos objectius oposats a l'entrenament d'una SVM: maximitzar el marge i minimitzar la classificació errònia dels punts de dades. Un valor petit de C dona més importància al marge ("underfitting") i en el cas contrari més importància a una classificació errònia ("overfitting"). Per tant, es provarà diferents valors que tenen gran diferència entre si sense ser valors extrems per veure quin obtindrà millor resultat.

El tipus de kernel determina la forma del hiperplà que separa el conjunt de dades. Té pocs tipus, es provarà tots els tipus excepte la polinòmica perquè tal com s'ha vist amb "PCA" anteriorment, se sap que en la projecció no sembla que els punts distribueixen d'una forma que les funciones polinòmiques els puguin classificar bé. A més, si tingués tants factors, el cost computacional es convertiria elevat.

El coeficient del nucli pot tenir valors dècimals. Per simplicitat s'utilitzarà "auto" i "scale" que configuren un valor de gamma automàticament.

Així doncs, els hiperparàmetres possibles seran els següents:

Hiperparàmetre	**Valors provats**
C	{0.01, 0.5, 1, 5, 100}
kernel	{"linear", "rbf", "sigmoid" }
gamma	{"scale","auto"}

Taula 4: Hiperparàmetres i valors provats en SVM

4.3.2 Primer entrenament amb "train"

L'entrenament del model de "SVM" no ha sigut tan complicat com el de "KNN" perquè no fa falta aplicar cap mètode de "Resampling", ja que el classificador és capaç d'assignar pesos a les classes de

les variables categòriques. Consegüentment, totes les decisions del preprocessament del model són completament idèntiques a les que s'ha proposat durant el treball.

Tal com s'havia fet a "KNN", el primer entrenament del model de"SVM" també s'ha fet amb els hiperparàmetres que venen per defecte. Gràcies a l'existència d'un "pipeline", es pot dur a terme el preprocessament ràpidament, aleshores també s'ha construit un segon model imputant els valors atípics. Finalment, s'ha aprofitat també de l'algorisme de "GridSearchCV" per trobar millor combinació d'hiperparàmetre.

	precision	recall	f1-score	support
C	0.69	0.80	0.74	158
CL	0.35	0.14	0.20	51
D	0.73	0.76	0.74	125
accuracy			0.69	334
macro avg	0.59	0.57	0.56	334
weighted avg	0.65	0.69	0.66	334

Figura 53: Resultats del primer model SVM

En el primer model "SVM" entrenat, que obté resultats similars als posteriors , les puntuacions obtingudes sobre les mètriques d'avaluació de la classe "CL" no són gens altes, encara que és una mica millor que en "KNN". Fins a aquest punt ja es pot posar com a hipòtesis que la classificació a la classe "CL" no és gens precisa deguda a l'estructura de la dades.

4.3.3 Anàlisi de resultats

Les puntacions aconseguides per cada configuració d'hiperparàmetres tenen generalment menys "overfitting" que el model "KNN", encara que el nivell d'"overfitting" pot variar molt segons la combinació d'hiperparàmetres. Afortunadament, per al millor valor obtingut en "cross validation" la puntuació en la partició de "train" n'és relativament semblant. Per tant, es pot observar que la combinació d'hiperparàmetre més adequada és: C= 0.01, kernel="linear", gamma= "scale".

	C	gamma	kernel	score VAL	score TRAIN
0	0.01	scale	linear	0.604848	0.698246
1	0.01	scale	rbf	0.333333	0.333333
2	0.01	scale	sigmoid	0.333333	0.333333
3	0.01	auto	linear	0.604848	0.698246
4	0.01	auto	rbf	0.333333	0.333333
5	0.01	auto	sigmoid	0.333333	0.333333
6	0.50	scale	linear	0.561563	0.761290
7	0.50	scale	rbf	0.558702	0.806485
8	0.50	scale	sigmoid	0.598632	0.616918
9	0.50	auto	linear	0.561563	0.761290
10	0.50	auto	rbf	0.570008	0.778434
11	0.50	auto	sigmoid	0.578506	0.658617
12	1.00	scale	linear	0.556378	0.782438
13	1.00	scale	rbf	0.578353	0.856613
14	1.00	scale	sigmoid	0.574359	0.603019
15	1.00	auto	linear	0.556378	0.782438
16	1.00	auto	rbf	0.561694	0.829424
17	1.00	auto	sigmoid	0.579495	0.630847
18	5.00	scale	linear	0.575410	0.799390
19	5.00	scale	rbf	0.536721	0.943261
20	5.00	scale	sigmoid	0.574477	0.544297
21	5.00	auto	linear	0.575410	0.799390
22	5.00	auto	rbf	0.590323	0.915077
23	5.00	auto	sigmoid	0.599528	0.592573
24	100.00	scale	linear	0.568527	0.803238
25	100.00	scale	rbf	0.464877	0.998099
26	100.00	scale	sigmoid	0.583922	0.542477
27	100.00	auto	linear	0.568527	0.803238
28	100.00	auto	rbf	0.460644	0.995751
29	100.00	auto	sigmoid	0.556928	0.560015

Figura 54: Puntacions dels models SVM en funció dels hiperparàmetres

I la taula següent mostra les puntuacions de diferents mètriques obtingudes pels models que s'ha entrenat:

	Balanced accuracy VAL	F1-score (W) VAL	Balanced accuracy TRAIN	F1-score (W) TRAIN
SVM + class weight balanced	0.586539	0.703830	0.800007	0.777920
SVM + class weight balanced + RemovingOutliers	0.621424	0.728237	0.850531	0.824459
SVM + class weight balanced + RemovingOutliers + GridSearchCV	0.605875	0.660499	0.699390	0.703957

Figura 55: Resultats de tots els models SVM

D'aquí es veu que els resultats són generalment millor que els models de "KNN", hi ha "overfittings" però no són tan greus comparat amb el que apareix en "KNN" gràcies a la configuració d'hiperparàmetres definides amb "GridSearch".

4.4 Decision Tree

Per últim, és l'entrenament d'un arbre de decisió ("Decision Tree"). Un arbre de decisió és una estructura d'arbre que es fa servir per modelar decisions i assignar classificacions a objectes.

Té una interpretabilitat relativament alta perquè la presa de decisions es basa en regles específiques que es defineix en el model. Tendeix a tenir "overfitting" perquè la profunditat de l'arbre solen ser molt alta si no es controla adequadament. A més, els arbres de decisió se centren en la presa de decisions a nivell local en comptes de modelar relacions globals entre característiques. Això pot ser una limitació en problemes en què les interaccions entre variables són importants, tal com són les característoques clíniques de la present base de dades.

4.4.1 Discussió dels hiperparàmetres

Els hiperparàmetres d'un arbre de decisió són:

- El criteri de qualitat de la divisió de fulles ("criterion")
- El coeficient que controlar el procés de poda de l'arbre ("ccp_alpha")

El paràmetre "criterion" s'utilitza per especificar la funció que utilitza l'algorisme per mesurar la qualitat d'una divisió en un node de l'arbre. Pot prendre tres valors: "gini", "entropy", "log_loss". Es provarà tots ells per trobar és més óptim.

El procés de poda és una tècnica que busca reduir la complexitat de l'arbre eliminant-ne algunes de les branques o nodes. La idea és evitar el sobreajustament (overfitting) del model, permetent que l'arbre sigui més general i, per tant, millor en la generalització de noves dades. Com més gran sigui el valor de "ccp_alpha", més gran serà la penalització per afegir nodes a l'arbre. Per a aquest paràmetre es provarà un rang de valors per veure com varia la qualitat del model.

Hiperparàmetre	**Valors provats**
criterion	{'gini', 'entropy', 'log_loss'}
ccp_alpha	{0.0, 0.02, 0.04, 0.06, 0.08, 0.1, 1}

Taula 5: Hiperparàmetres i valors provats en Decision Tree

4.4.2 Primer entrenament amb "train"

L'entrenament del model d'arbre de decisió és molt similar al que es va fer amb el model "SVM", ja que els dos tenen un paràmetre per tractar classes desequilibrades. No obstant això, a diferència del model de "SVM", en els arbres de decisió no es farà "one-hot-enconding" per les variables categòriques, ja que té el risc de generar valors intermitjos que no pertanyen a les classes categòriques directament. A més, els arbres de decisió són capaços de manejar directament tant les variables numèriques com les categòriques, només li cal un "ordinal-enconding" per recodificar les variables categòriques a tenir valors numèrics.

El primer entrenament és amb els paràmetres de defecte, el segon és sense els valors "atípics" amb els mateixos paràmetres que el primer, i el tercer entrenament s'ha fet servir els hiperparàmetres que s'ha tret amb "GridSearchCV" en base del segon model.

	precision	recall	f1-score	support
C	0.70	0.74	0.72	175
CL	0.10	0.08	0.09	26
D	0.68	0.67	0.67	133
accuracy			0.66	334
macro avg	0.49	0.49	0.49	334
weighted avg	0.65	0.66	0.65	334

Figura 56: Resultats del primer model Decision Tree

En els models d'arbre de decisió entrenats, el comportament és molt similar al anteriors models. Encara que en el primer model entrenat s'ha observat que ha classificat menor quantitat a classe "CL", la qual cosa és bona (hi ha molt poques de la classe), tant la precisió com la sensibilitat és molt baixa. Es pot "acceptar" la hipòtesi proposada en l'entrenament del model SVM: és difícil arribar a obtenir una classificació extremedamament bona a causa de l'estructura de la base de dades estudiada.

4.4.3 Anàlisi de resultats

L'exactitud balancejada calculada per cada combinació d'hiperparàmetres són:

	criterion	ccp_alpha	score VAL	score TRAIN
0	0.00	gini	0.498152	1.000000
1	0.00	entropy	0.482340	1.000000
2	0.00	log_loss	0.482340	1.000000
3	0.02	gini	0.468638	0.782697
4	0.02	entropy	0.493704	0.880661
5	0.02	log_loss	0.493704	0.880661
6	0.04	gini	0.512227	0.631857
7	0.04	entropy	0.529909	0.798914
8	0.04	log_loss	0.529909	0.798914
9	0.06	gini	0.494031	0.560635
10	0.06	entropy	0.497136	0.735843
11	0.06	log_loss	0.497136	0.735843
12	0.08	gini	0.353846	0.415725
13	0.08	entropy	0.493139	0.645950
14	0.08	log_loss	0.493139	0.645950
15	0.10	gini	0.333333	0.333333
16	0.10	entropy	0.519237	0.612123
17	0.10	log_loss	0.519237	0.612123
18	1.00	gini	0.333333	0.333333
19	1.00	entropy	0.333333	0.333333
20	1.00	log_loss	0.333333	0.333333

Figura 57: Puntacions dels models Decision Tree en funció dels hiperparàmetres

Aquí s'ha tornat a haver una gran variabilitat de nivell d'"overfitting" depenent de la configuració dels hiperparàmetres. Tanmateix, generalment l'exactitud balancejada és baixa en "cross validation", per tant probablement no treurà resultats bons sigui quin sigui la combinació de paràmetres.

Per altra banda, els resultats finals (exactitud balancejada i "f1-score") dels models entrenats són:

	Balanced accuracy VAL	F1-score (W) VAL	Balanced accuracy TRAIN	F1-score (W) TRAIN
Decision Tree + class weight balanced	0.532978	0.682820	1.0000	1.00000
Decision Tree + class weight balanced + RemovingOutliers	0.494382	0.649651	1.0000	1.00000
Decision Tree + class weight balanced + GridSearchCV	0.518925	0.657508	0.7754	0.68409

Figura 58: Resultats de tots els models Decision Tree

Es veu que tots els primer dos models tenen un "overfitting" molt greu, tenint una exactitud balancejada igual 1 en la partició de "train", mentre que amb "cross validation" només arriba a la seva meitat. L'últim model si que es pot considerar més o menys bo, no hi ha tanta diferència entre els resultats obtinguts al "train" i a "val", però tant l'exactitud com la "f1-score" són molt baixes.

5 Selecció de model

En la secció anterior s'havia definit uns quants models de tots els tres tipus de classificadors, ara s'ha de seleccionar el millor d'ells.

	Balanced accuracy VAL	F1-score (W) VAL	Balanced accuracy TRAIN	F1-score (W) TRAIN
KNN	0.531929	0.712498	0.598464	0.780986
KNN + SMOTENC	0.472390	0.586957	0.868571	0.586460
KNN + Oversampling	0.511676	0.656618	0.843352	0.543019
KNN + Oversampling + RemovingOutliers	0.528630	0.621788	0.848816	0.431310
KNN + Oversampling + RemovingOutliers + GridSearchCV	0.488686	0.654316	1.000000	0.431310
SVM + class weight balanced	0.586539	0.703830	0.800007	0.777920
SVM + class weight balanced + RemovingOutliers	0.621424	0.728237	0.850531	0.824459
SVM + class weight balanced + RemovingOutliers + GridSearchCV	0.605875	0.660499	0.699390	0.703957
Decision Tree + class weight balanced	0.532978	0.682820	1.000000	1.000000
Decision Tree + class weight balanced + RemovingOutliers	0.494382	0.649651	1.000000	1.000000
Decision Tree + class weight balanced + GridSearchCV	0.518925	0.657508	0.775400	0.684090

Figura 59: Resultats de tots els models

Generalment els resultats obtinguts amb els models de "SVM" són millors que els d'altres models, ja que tenen els majors valors de les mètriques d'avaluació tant en la partició "train" com en la "cross validation", sense haver causat molt "underfitting" o bé "overfitting". Sembla que els hiperparàmetres escollits per les "SVM" són relativament correctes per crear un model que classifiqui bé les mostres, encara que amb "PCA" s'havia vist que no era una tasca gens senzilla.

Entre els models de "SVM" el que s'ha tret major valors de les mètriques és "SVM + class weight balanced + RemovingOutliers", un "SVM" que té en compte la donació de pesos a les classes i l'eliminació dels valors atípics. No obstant això, el que té millor "fit", és a dir, resultats més semblants entre els de "train" i els de "val", és el que té en compte també la millor combinació d'hiperparàmetres trobada amb "GridSearchCV". Per tant, s'ha decidit escollir aquest model que té un ajust relativament bo com a model final.

5.1 Anàlisi de les limitacions i capacitats del model

Com ja se sap a l'inici del treball, el principal problema de la base de dades investigada són el desbalanceig de les classes. Per aquest motiu, el model podria tenir una taxa veritable positiva baixa i una taxa veritable negativa alta. Per resoldre-ho s'ha aplicat el mètode de donació de pesos a les classes, però probablement no ho resoldrà del tot.

Per altra banda també hi són l'existència dels valors atípics, la gran quantitat de valors buits i la falta de mostres observades. Tot això ha limitat també la capacitat predictiva del model, ja que no existeix cap preprocessament que pugui solucionar-los completament.

5.2 Resultats en partició de test

Finalment es pot aplicar el model final per predir l'estat de supervivència dels pacients de la partició "test". S'ha tret les mateixes mètriques d'avaluació per a "test", tal com s'havia fet per a "train" i

"val":

	Balanced accuracy VAL	F1-score (W) VAL	Balanced accuracy TRAIN	F1-score (W) TRAIN	Balanced accuracy TEST	F1-score (W) TEST
Final Model	0.621424	0.728237	0.850531	0.824459	0.521995	0.669304

Figura 60: Resultats finals del model incloent train, val i test

Sorprenentment l'exactitud balancejada i la "f1-score" no són gens baixes en la partició de "test": pitjor que en la partició de "train", la qual cosa és normal i esperat i millor que en la partició de "val".

Es pot utilitzar una matriu de confusió per veure com s'ha classificat exactament el model de la partició de "test":

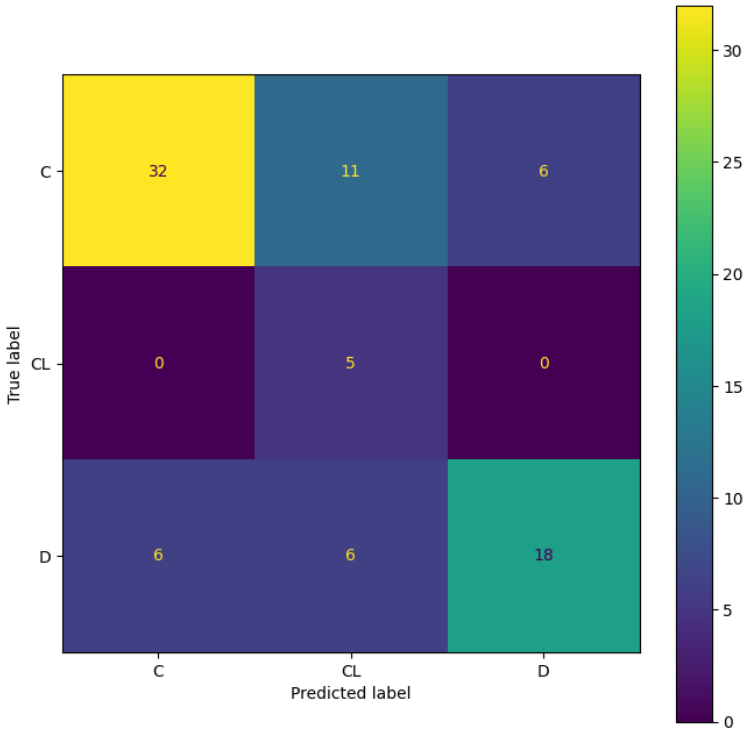


Figura 61: Matriu de Confusió de la predicció sobre test

S'observa que totes les mostres que pertanyen a la classe "CL", que és minoritària en relació de la "C", són classificades correctament, la qual cosa vol dir que la donació de pesos a les classes va tenir èxit. A més, per les altres dos classes la predicció tampoc és gens mala: la majoria s'ha classificat correctament.

6 Model Card

MODEL CARD

Visió general del model

El model present és bàsicament un Support Vector Machine per classificar l'estat de supervivència dels pacients donats unes característiques clíniques. Concretament, la base de dades és la que proporciona UCI (la Universitat de Califòrnia a Irvine) anomenada "Cirrhosis Patient Survival Prediction". En aquest "Model card" trobaràs informacions tant sobre la variable resposta com les variables predictores, la intenció principal del model, les seves limitacions i les seves puntuacions sobre les mètriques d'avaluació que hi ha.

Versió

name: hdayiuy8e1231dasshi91
data: 28 - 12 - 2023

Autor

- Zhihao Chen, zhihao.chen@estudiantat.upc.edu

Referència

https://archive.ics.uci.edu/ml/datasets/Breast+Cancer+Wisconsin+(Diagnostic)
https://minds.wisconsin.edu/bitstream/handle/1793/59692/TR1131.pdf
https://scikit-learn.org/stable/

Consideracions

Usuaris destinats

- Estudiants de IAA
- Professors de IAA

Cas d'ús

El conjunt de dades en què es va entrenar aquest model es va crear per donar suport a la comunitat d'aprenentatge de màquines en la realització d'anàlisis empíriques dels algorismes de ML. La base de dades es pot utilitzar en estudis relacionats amb la predicció de la supervivència dels pacients donats les seves mesures clíniques.

Limitacions

El principal problema de la base de dades investigada són el desbalanceig de les classes. Per aquest motiu, el model podria tenir una taxa veritable positiva baixa i una taxa veritable negativa alta. Per resoldre-ho s'ha aplicat el mètode de donació de pesos a les classes, però probablement no ho resoldrà del tot. Per altra banda també hi són l'existència dels valors atípics, la gran quantitat de valors buits i la falta de mostres observades. Tot això ha limitat també la capacitat predictiva del mode.

Consideracions ètiques

Risc: La predicció errònia pot causar decisions inadequades en el tractament del pacient.
Estratègia de mitigació: Només utilitzar el model per a estudis relacionats amb ML.

Detalls del model

Descripció del Model

La idea fonamental darrere de les SVM és trobar un hiperplà òptim que pugui separar de la millor manera possible dos conjunts de dades en un espai de característiques d'alta dimensió. Té dos hiperparàmetres més rellevants:

- El paràmetre de regularització C controla l'equilibri entre dos objectius oposats a l'entrenament d'una SVM: maximitzar el marge i minimitzar la classificació errònia dels punts de dades.
- El tipus de kernel determina la forma del hiperplà que separa el conjunt de dades.

Preprocessament

Per les variables numèriques s'ha fet la transformació dels valors atípics a valors buits, la imputació amb la mediana i l'estandardització. Per les variables categòriques no ordinals s'ha creat una nova modalitat per guardars els valors buits i s'ha fet la recodificació binària. Per les variables ordinals s'ha fet la imputació amb la moda i la recodificació ordinal.
De les particions de dades no s'ha decidit guardar una part per la validació, per tant només hi són train i test, amb una proporció habitual de 8:2. A l'hora de necessitar la partició de validació es farà servir l'algorisme de 10-fold Cross Validation sobre la partició de train.

Entrenament del model

Degut al desbalanceig de classes que hi ha en la base de dades, s'ha decidit aplicar la tècnica de donació de pesos a les classes. Per trobar la millor configuració dels hiperparàmetres s'ha fet servir l'algorisme de GridSearchCV, provant diverses combinacions d'hiperparàmetres disponibles. Finalment els hiperparàmetres són:

{C=0.01, class_weight='balanced', kernel='linear', random_state=1}, la resta de paràmetres són els de defecte.

Mètrica d'avaluació

A causa de tenir dades desequilibrades, les mètriques d'avaluació no poden ser les estàndards:

- L'exactitud balancejada (Balanced accuracy)
- La F1 score ponderada

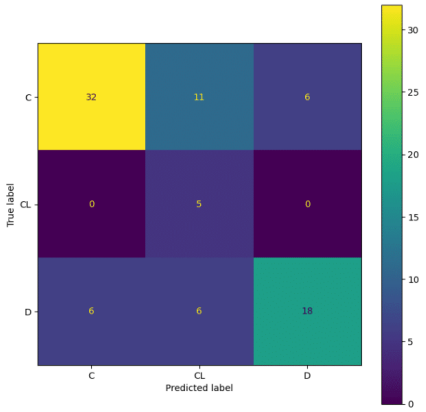
Aquestes dues no són influenciades pel desbalanceig de classes i proporcionen informacions suficients fiables sobre el rendiment del model: tant com la taxa de veritables positius com la de veritables negatius són importants en la present investigació.

Rendiment del model

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

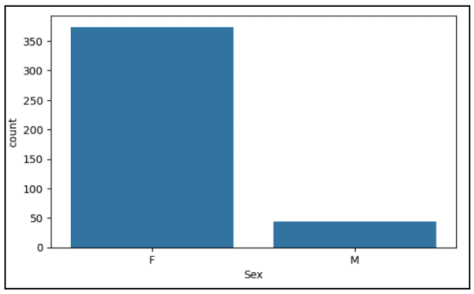
- Balanced accuracy: 0.75102
- F1 score average weighted: 0.68926

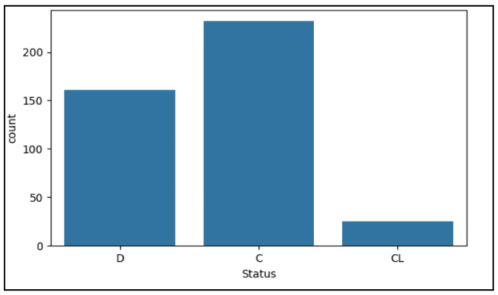
En la matriu de confusió de les prediccions sobre la partició de test s'observa que totes les mostres que pertanyen a la classe "D", que és minoritària, són classificades correctament, la qual cosa vol dir que la donació de pesos a les classes va tenir èxit. A més, per les altres dos classes la predicció tampoc és gens mala: la majoria s'ha classificat correctament.



Variables categòriques

En aquesta secció inclouen els diagrames de barres de la variable "Sex" i la variable objectiu "Status". S'ha seleccionat especialment aquestes dos variables perquè s'ha considerat rellevant que l'usuari vegi el desbalanceig de classes que existeix tant en les variables independents com en la variable resposta.





Variables numèriques

La intenció d'aquesta secció és mostrar a l'usuari la quantitat de valors atípics i valors buits que hi ha en les variables numèriques mitjançant els histogrames i els diagrames de caixes. Només s'ha mostrat una variable perquè ja que les altres tenen el mateix comportament majoritàriament.