Introducció a l'Aprenentatge Automàtic

CIRRHOSIS PATIENT SURVIVAL PREDICTION

Práctica

GIA UPC
Curso 2023/2024

Abril Risso

ÍNDEX

1. Introducció

Aquesta pràctica té com a objectiu crear un model per predir la supervivència de pacients amb cirrosi hepàtica, basant-se en 17 característiques clíniques.

Ens endinsem en l'entrenant i avaluació de tres models de machine learning amb l'objectiu de predir la supervivència de pacients amb cirrosi hepàtica. Utilitzant un dataset detallat, que conté 418 instàncies, que inclou diverses variables clíniques, ens centrem en analitzar, preprocessar i preparar les dades per a la modelització.

El treball cobreix la selecció de característiques, l'aplicació de diferents tècniques de classificació, i la determinació del millor model basant-nos en mètriques de rendiment com la precisió, la sensibilitat i l'especificitat.

En aquesta pràctica es farà un anàlisi detallat de totes les variables tant categòriques com numèriques. Es realitzarà un preprocessament de dades, tant tractament de valors atípics com de valors faltants. Es farà un procés de preparació i estudi de les relacions entre les variables i la importància que tenen aquestes en el nostre dataset respecte la variable objectiu que és la qual volem predir.

Finalment es realitzarà l'entrenament de tres models diferents (KNN, arbre de decisió, i SVM) i es farà la selecció del model més adequat basant-se en les seves limitacions i capacitats, a més de la documentació del model final.

2. Anàlisi i preprocessat de dades

Abans de començar a fer un anàlisi de cada variable, és necessari importar les llibreries necessàries i carregar el conjunt de dades amb el qual volem treballar. Un cop ja s'ha carregat el dataset, s'ha iniciat la seva exploració.

Per realitzar aquesta exploració prèvia a l'inici de l'anàlisi, s'ha examinat el nombre de variables contingudes al dataset, què representa cadascuna d'aquestes, i quin és el seu tipus. A més, també s'ha observat la mida del dataset, que consta de 418 files i 20 columnes.

2.1. Anàlisi estadístic de les variables

Per dur a terme l'anàlisi estadístic, s'han creat dues variables. La variable *var_num* conté totes les variables numèriques i *var_cat* conté totes les variables categòriques.

A l'hora de realitzar els gràfics per examinar les distribucions i estudiar cada variable, s'ha observat que algunes de les variables no es detectaven correctament. És a dir, algunes de les variables categòriques eren identificades com a numèriques i algunes numèriques com categòriques. Per tant, s'ha realitzat una modificació del tipus de variable per aquelles que estaven classificades de manera incorrecta.

També, per visualitzar d'una manera més clara els gràfics de les variables *Age* i *N_Days,* que inicialment estan representades en dies, s'han canviat a anys. Per tant, a més d'unicament canviar el seu valor, la variable *N_Days* s'ha reanomenat a *N_Years*.

A més, també s'ha eliminat la variable *ID* del dataset prèviament a l'anàlisi ja que aquesta variable és un identificador, per tant, no ens aporta cap informació útil pel model i a més podria arribar a distorsionar l'aprenentatge.

Ara, amb les variables correctament classificades i emmagatzemades a les variables mencionades anteriorment, podem realitzar l'anàlisi detallat de cada variable.

Realitzant els gràfics de cada variable s'ha observat que algunes variables tenen valors faltants que no són detectats correctament com a missing values, sinó que són representats com una classe diferent anomenada NaNN. Per tant, s'ha realitzat l'estudi de les variables un cop s'ha modificat el valor d'aquests missing values.

2.1.1. Anàlisi de variables numèriques

Com s'ha pogut examinar, el dataset conté 12 variables numèriques. Seguidament, s'ha realitzat una taula per cada variable per poder observar un resum estadístic de cada variable.

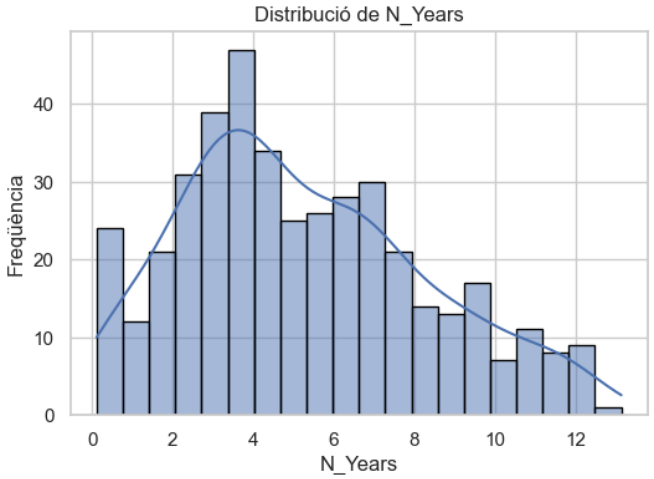
En aquestes taules es troba la quantitat d'observacions no nul·les, la mitjana, la desviació estàndard, el valor mínim, els diferents percentils (25, 50 i 75) i el valor màxim de cada variable.

A continuació, s'han realitzat tots els histogrames corresponents a cada variable numèrica.

N_YEARS

Aquesta variable representa el nombre d'anys que han transcorregut des de l'inici de l'estudi fins al final, el qual té una variació d'entre 0 i 12.5 anys.

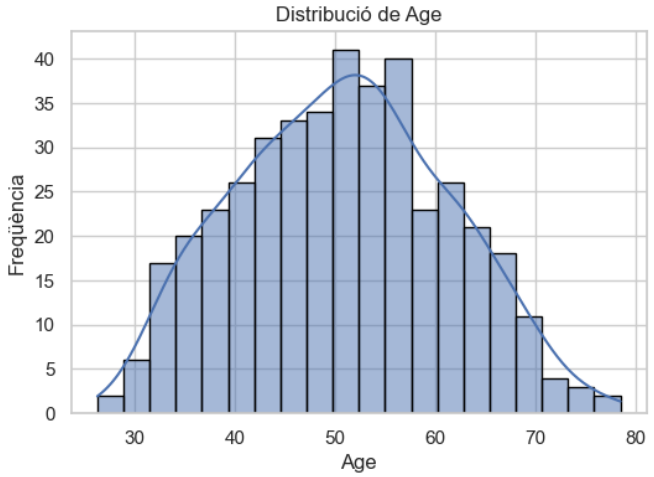
S'ha pogut examinar que la variable *N_Years*, sembla tenir una distribució simètrica al voltant de la mitjana, una característica que pertany a les distribucions normals. Però es pot veure que té un lleuger biaix cap a l'esquerra. Per comprovar si es tracta d'una distribució normal s'ha realitzat la prova de normalitat D'Agostino i Pearson, la cual ha indicat que aquesta distribució no és normal amb un p-value inferior a 0.05.



count	mean	std	min	25%	50%	75%	max
418.0000	5.250602	3.024430	0.112252	2.991786	4.736482	7.155373	13.12799

AGE

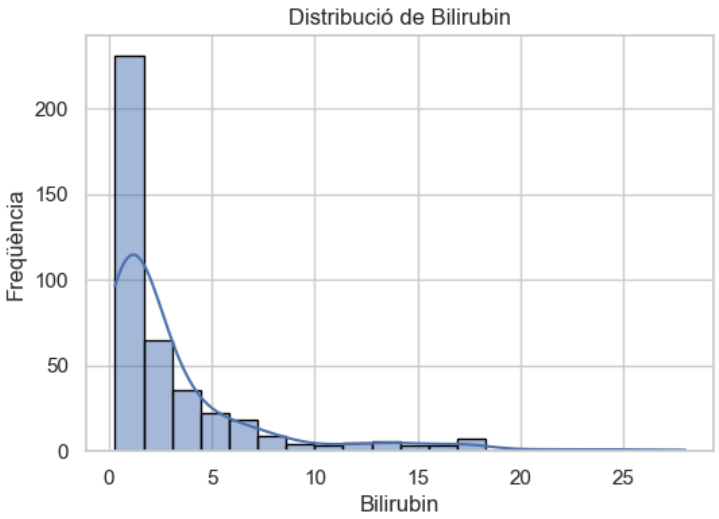
La variable *Age* representa el nombre d'anys que té el pacient enregistrat. En aquest estudi s'han pres mostres de pacients d'entre 26 a 78 anys. A més, com s'ha pogut observar a la taula anterior, la mitjana d'aquesta variable és 50.7 i aquesta variable sembla tenir una distribució simètrica al voltant de la mitjana. Tot i així, realitzant la prova de normalitat, aquesta no ha donat un p-value major a 0.05.



count	mean	std	min	25%	50%	75%	max
418.0000	50.74155	10.44721	26.27789	42.83230	51.00068	58.24093	78.43942

BILIRUBIN

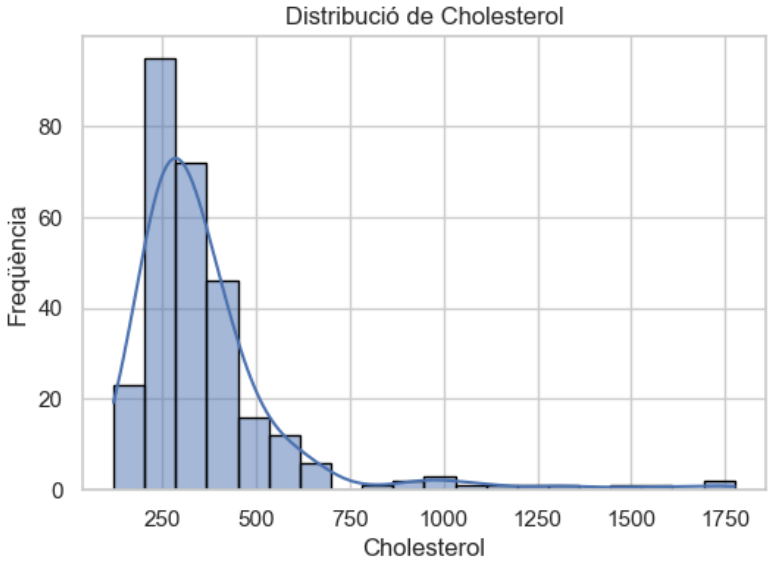
Aquesta variable representa la concentració de bilirrubina mesurada en miligrams per centilitre. Com es pot observar, no consta d'una distribució normal. Probablement es tracta d'una distribució exponencial, ja que com podem veure, la gran majoria de mostres es troben a prop del 0 i decreix ràpidament d'una manera significativa.



count	mean	std	min	25%	50%	75%	max
418.0000	3.220813	4.407506	0.300000	0.800000	1.400000	3.400000	28.00000

CHOLESTEROL

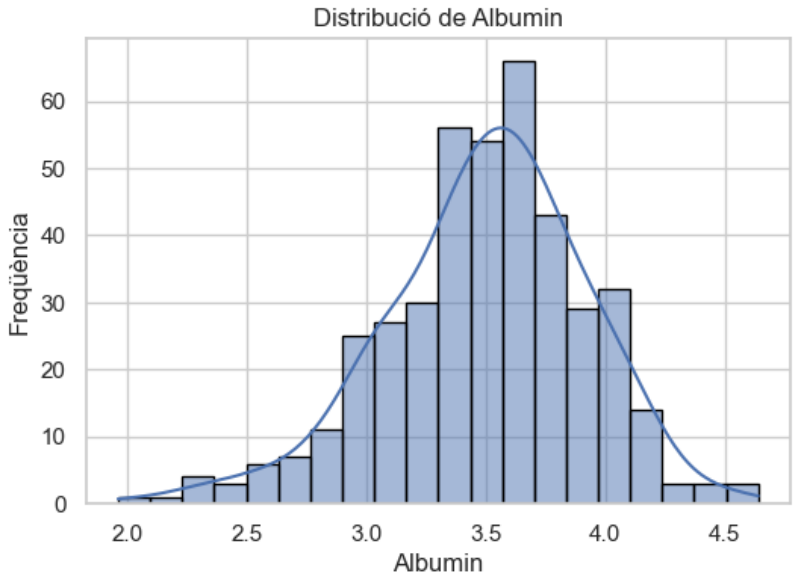
Aquesta variable representa la concentració de colesterol (tipus de grasa) en sang dels individus mesurat en miligrams per decilitre. Com es pot observar, no es tracta d'una variable amb una distribució normal. La majoria de pacients tenen un nivell de colesterol prop de 250. Tot i així es pot veure que hi ha bastants pacients que tenen nivells molt més alts.



count	mean	std	min	25%	50%	75%	max
284.0000	369.510	231.944	120.000	249.500	309.500	400.000	1775.00

ALBUMIN

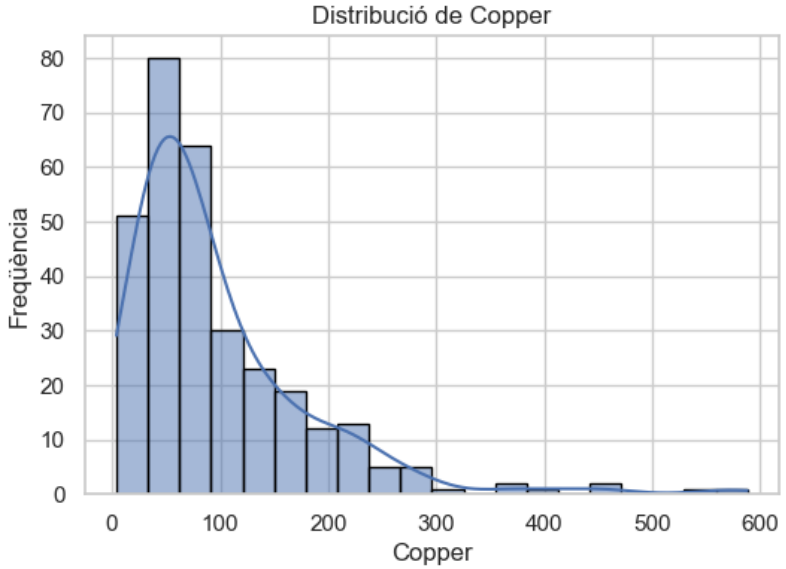
Aquesta variable representa els nivells d'albumina, proteïna produïda pel fetge, en sang mesurats en grams per decilitre. Nivells baixos d'aquesta proteïna estan associats a un major risc de mortalitat tant a llarg com a curt termini. Com es pot observar, aquesta variable sembla tenir una distribució normal, pero a l'hora de fer la prova de normalitat D'Agostino i Pearson, el p-value menor a 0.05 ha indicat que aquesta variable no té una distribució normal. Els seus valors es troben entre 2 i 4.5 g/dl aproximadament, encara que la majoria de pacients tenen un nivell d'albumina prop del 3.5.



count	mean	std	min	25%	50%	75%	max
418.0000	3.497440	0.424972	1.960000	3.242500	3.530000	3.770000	4.640000

COPPER

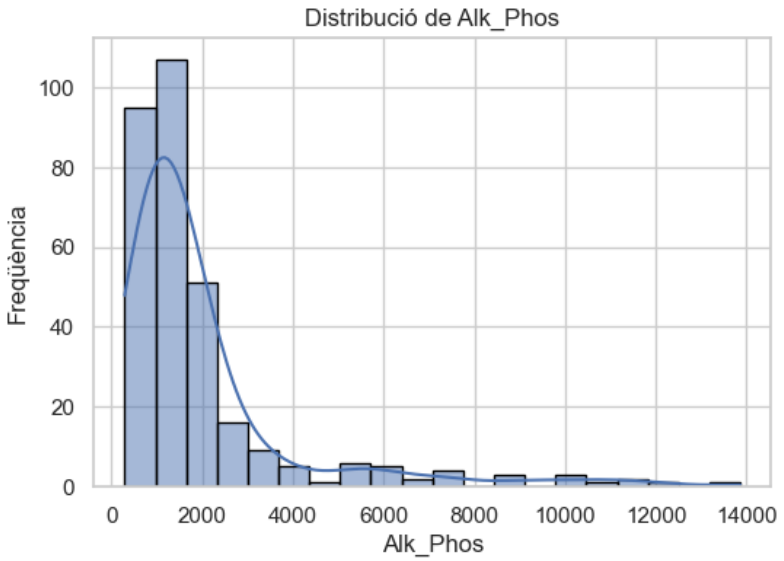
La variable Copper, representa la quantitat de coure present a l'orina d'un pacient mesurada en micrograms per dia (ug/day). Com es pot observar, a simple vista aquesta variable no té una distribució normal. La major part dels individus presenten 50 micrograms per dia de quantitat de coure a l'orina, encara que també podem trobar altres individus amb nivells molt més elevats.



count	mean	std	min	25%	50%	75%	max
310.0000	97.64838	85.61392	4.000000	41.25000	73.00000	123.0000	588.0000

ALK_PHOS

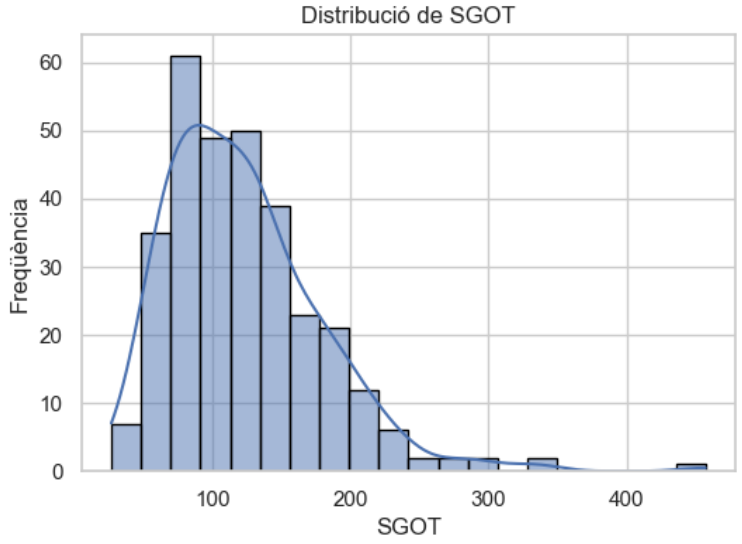
Aquesta variable representa el nivell de fosfatasa alcalina, un tipus d'enzim, mesurat en unitats per litre (U/liter). Com es pot observar a simple vista, la distribució d'aquesta variable no és una distribució normal. A més, es pot veure que la majoria de pacients tenen uns nivells de fosfatasa alcalina entre 50 i 2000 unitats per litre.



count	mean	std	min	25%	50%	75%	max
312.0000	1982.655	2140.388	289.0000	871.5000	1259.000	1980.000	13862.40

SGOT

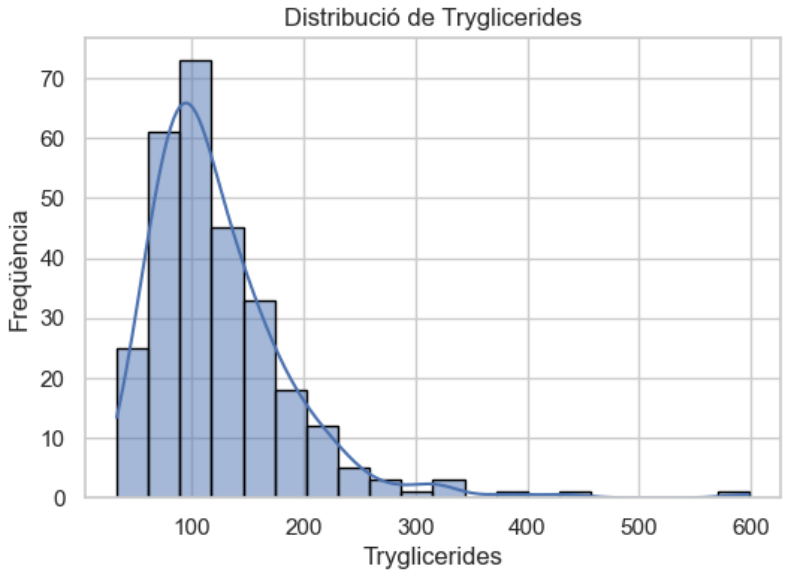
La variable *SGOT* (Serum Glutamic-Oxaloacetic Transaminase), representa la concentració de l'enzim SGOT en la sang. La unitat de mesura utilitzada per aquesta variable són unitats per mililitre (U/ml). Com podem observar no es tracta d'una distribució normal. La majoria de pacients presenten unes concentracions de SGOT d'entre 80 i 150 U/ml. Encara que també podem observar valors bastant més alts.



count	mean	std	min	25%	50%	75%	max
312.0000	122.5563	56.69952	26.35000	80.60000	114.7000	151.9000	457.2500

TRYCLICERIDES

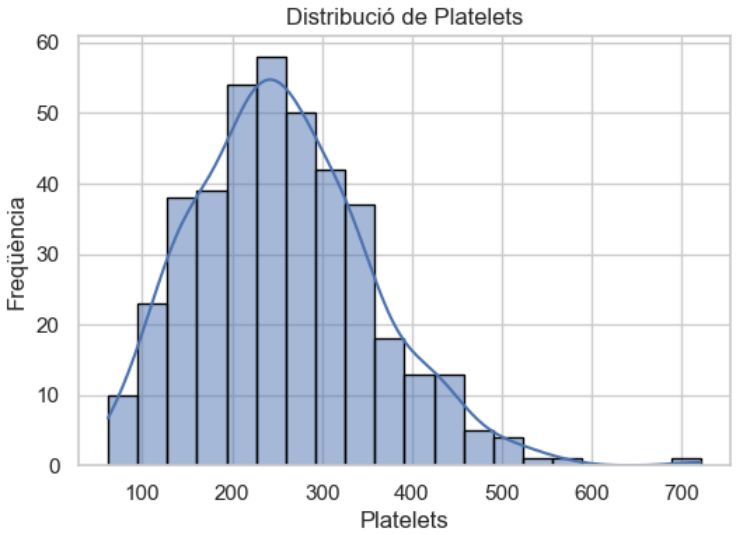
Aquesta variable representa la concentració de Triglicèrids mesurada en miligrams per decilitre (mg/dl). Com podem observar la majoria d'individus tenen una quantitat de triglicèrids entre 70 i 150 mg/dl, encara que també podem trobar pacients amb valors més extrems. No es tracta d'una distribució normal.



count	mean	std	min	25%	50%	75%	max
282.0000	124.7021	65.14863	33.00000	84.25000	108.0000	151.0000	598.0000

PLATELETS

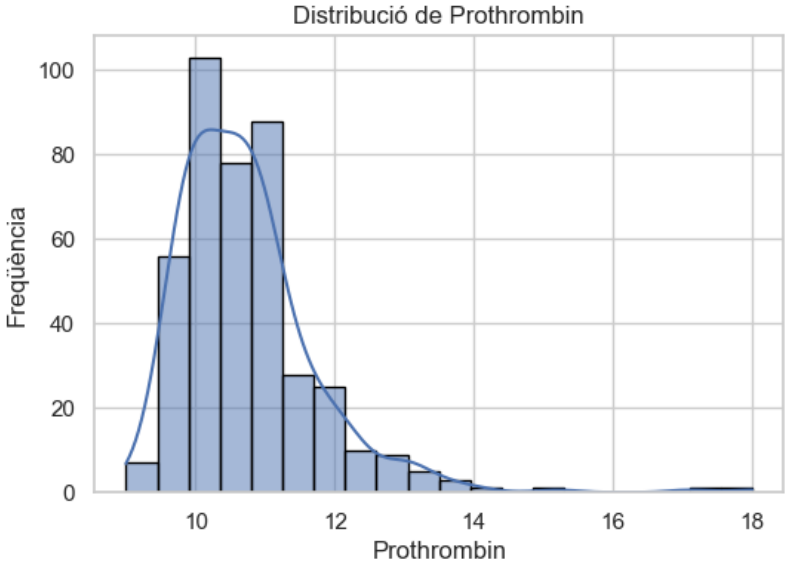
Aquesta variable representa la quantitat de plaquetes en sang mesurades en unitats per mil·límetre cúbic (ml/1000). La mitjana d'aquesta variable és 257.02 unitats de plaquetes ml/1000. Encara que trobem un valor de plaquetes bastant més elevat que la resta. Tot i semblar que es tracta d'una distribució normal, realitzant la prova de normalitat, no ha donat que aquesta sigui normal indicant un p-value menor a 0.05.



count	mean	std	min	25%	50%	75%	max
407.0000	257.0245	98.32558	62.00000	188.5000	251.0000	318.0000	721.0000

PROTHROMBIN

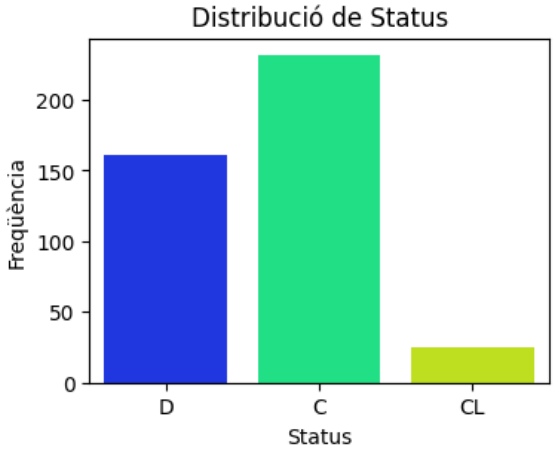
La variable *Prothrombin* representa el temps que triga la sang en coagular-se (temps de protrombina) mesurada en segons. Com podem observar, la majoria de pacients presenten un temps d'entre 6 i 11 segons, encara que també observem alguns valors més extrems majors a 14 segons. La mitjana de temps dels pacients d'aquest estudi és de 10.73 segons. A simple vista no es tracta d'una distribució normal i realitzant la prova de normalitat, confirmem que és així amb un p-value inferior a 0.05.

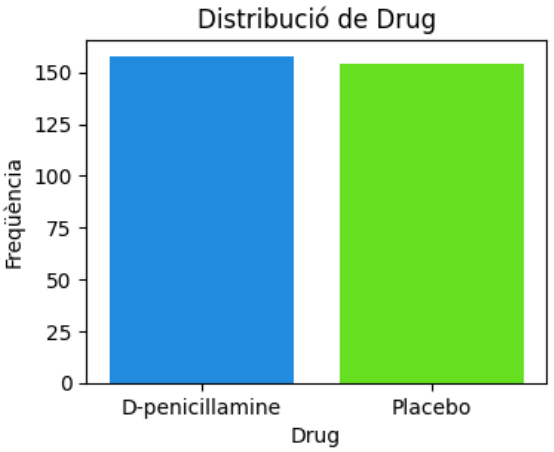


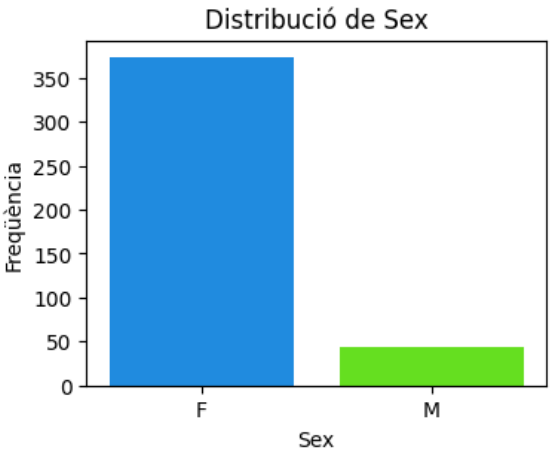
count	mean	std	min	25%	50%	75%	max
416.0000	10.73173	1.022000	9.000000	10.00000	10.60000	11.10000	18.00000

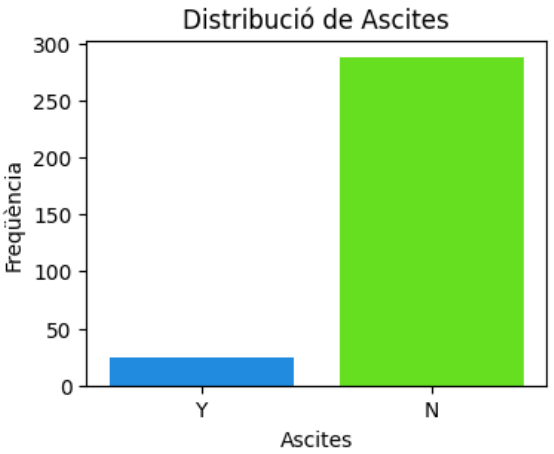
2.1.2. Anàlisi de variables categòriques

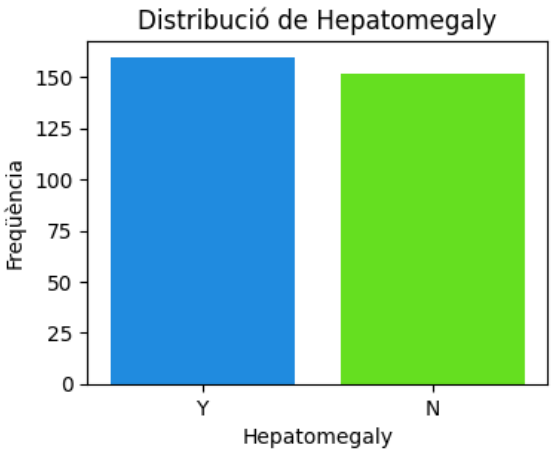
Seguidament, es van realitzar els plots de les variables categòriques per observar la freqüència de cada classe.

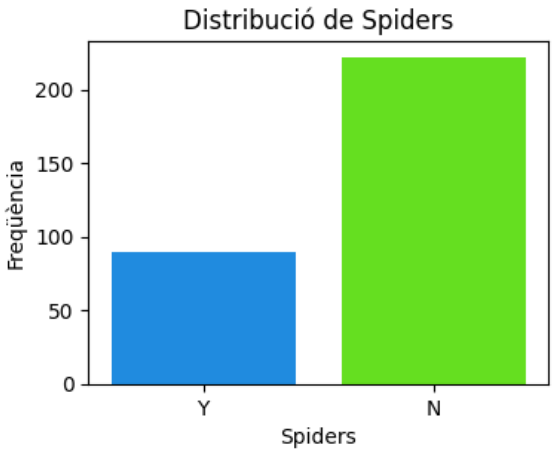


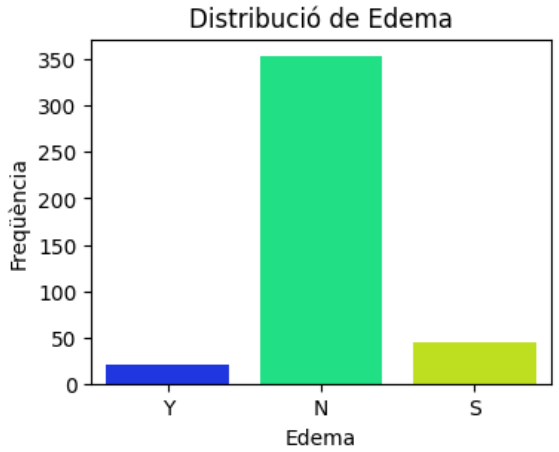


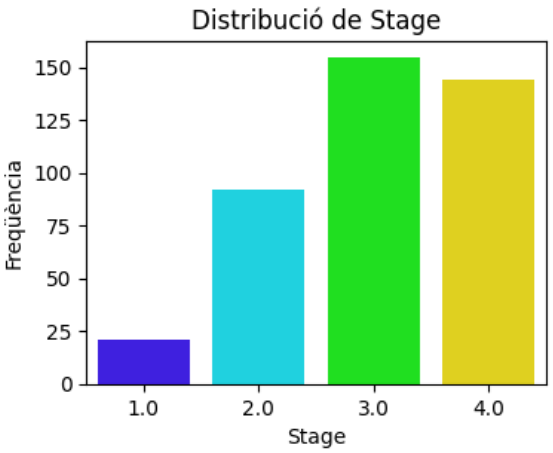












Com es pot observar, en la variable *Status* (la nostra variable objectiu), la categoria "*C*" (indica que el pacient ha sobreviscut) té la freqüència més alta, seguida per "*D*" (indica que el pacient ha mort) i finalment "*CL*" (indica que al pacient se li ha realitzat un trasplantament de fetge) amb moltes menys instàncies que les dues classes anteriors. Aquest desbalanceig de classes en la variable objectiu ens suggereix realitzar un balanceig ja que sinó el model podria no aprendre correctament els patrons en la classe minoritària, la qual cosa resultaria un menor rendiment.

En la variable *Drug* es troba una quantitat molt similar d'individus que reben el medicament D-penicilamina i els que reben placebo.

En la variable *Sex* es pot observar que hi ha una quantitat molt més elevada de persones del sexe femení (*F*) a comparació amb el sexe masculí (*M*). Això ens indica que la majoria de persones que han participat en aquest estudi són dones.

En la variable *Ascites*, podem observar que una gran majoria de pacients no presenten ascitis (*N*), acumulació de líquid en l'abdomen.

En la variable *Hepatomegaly*, el número d'individus amb i sense hepatomegalia és pràcticament el mateix.

En la variable *Spiders*, la freqüència d'individus sense aranyes vasculars (*N*) és bastant major a aquells individus amb (*S*).

En la variable *Edema*, la majoria d'individus no tenen edema "*N*". Amb una freqüència bastant més baixa trobem la classe "*S*" (presència d'edema sense diürètics) i seguidament amb una freqüència bastant similar a "*S*" trobem la classe "*Y*", els individus que sí que presenten edema.

Finalment, en la variable *Stage* s'observa una distribució variada entre les diferents etapes de la cirrosis, sent la tercera etapa la més comú, seguida per l'etapa 3, la 2 i la 1. Això ens indica que la majoria dels pacients es trobàven en les etapes 3 o 4 de la cirrosis quan van ser enquestats.

2.2. Observació i tractament d'outliers

Per observar els outliers de les nostres variables numèriques del nostre dataset, hem realitzat boxplots. D'aquesta manera visualitzarem la distribució de cada variable destacant la seva mediana, quartils i els valors atípics (outliers).

La caixa central del boxplot representa la distància entre el el quartil inferior (Q1) i el quartil superior (Q3). Les dues línies que sobresurten de la caixa central, ens indiquen els valors màxims i mínims que es troben dins d'un límit establert.

Qualsevol punt que es trobi fora d'aquestes línies es considera un valor atípic. Aquests punts són visualment detectables ja que es presenten com punts individuals separats.

Alguns valors atípics poden indicar una variació significativa, errors de mesura o entrada de dades. Tot i així, cal destacar que és important no descartar automàticament els outliers com errors ja que en alguns casos es poden tractar de dades vàlides i significatives que aporten informació important al nostre model.

Per tant, després d'observar els outliers es realitzarà una recerca informativa sobre les variables que presenten aquests valors atípics i ens assegurarem de quines dades són outliers i quines podrien ser vàlides.

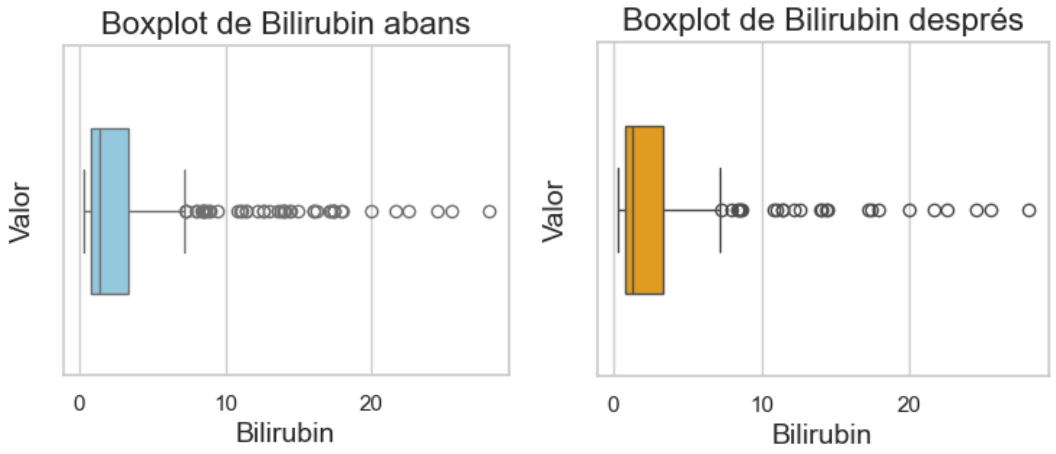
En aquest tractament d'outliers, s'ha pres la decisió de no eliminar les files que continguin aquests valors atípics. Aquesta elecció es basa en el fet que aquest dataset no disposa de moltes files, i per tant, eliminar-les podria afectar negativament la integritat i la representativitat de les dades. En lloc d'eliminar-les, s'ha optat per marcar com a valors faltants aquells outliers identificats. Això permet mantenir la màxima quantitat de dades possible. És important destacar que aquesta estratègia requereix una anàlisi cuidadosa per identificar correctament els outliers i decidir quins d'ells han de ser considerats com a valors faltants.

Les variables categòriques no tenen outliers com les variables numèriques. Les variables categòriques al ser variables qualitatives i no quantitatives, no presenten valors numèricament distants a la majoria dels altres valors. Per tant, únicament tractarem outliers de les variables numèriques.

S’han realitzat boxplots i s’han detectat outliers en les variables següents:

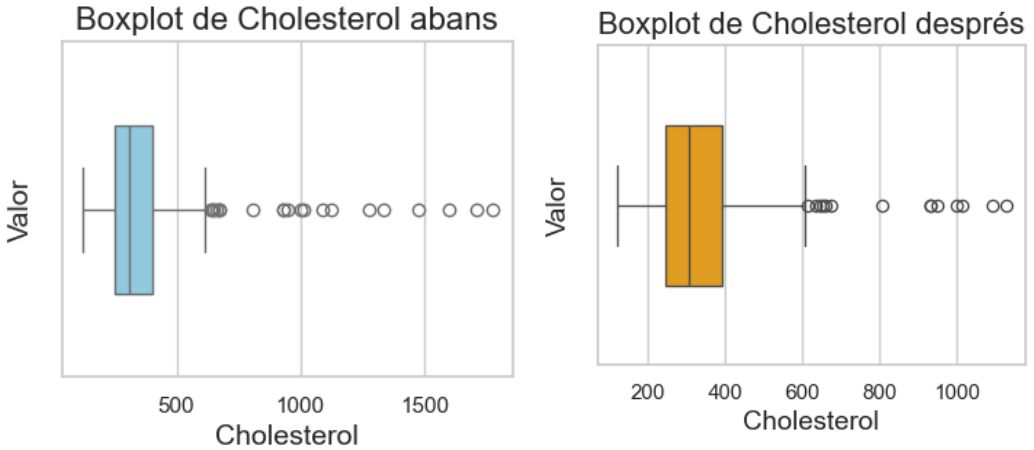
- **Bilirubin**

Realitzant una recerca sobre els valors de bilirrubina que podria tenir una persona, s’ha trobat que si que és posible que una persona pugui tenir un nivell de bilirrubina de 28 mg/dl, encara que aquests siguin extremadament alts, podrien estar associats a condicions hepàtiques greus, com la cirrosis avançada. Per tant, no canviarem com a missing value cap outlier d’aquesta variable.



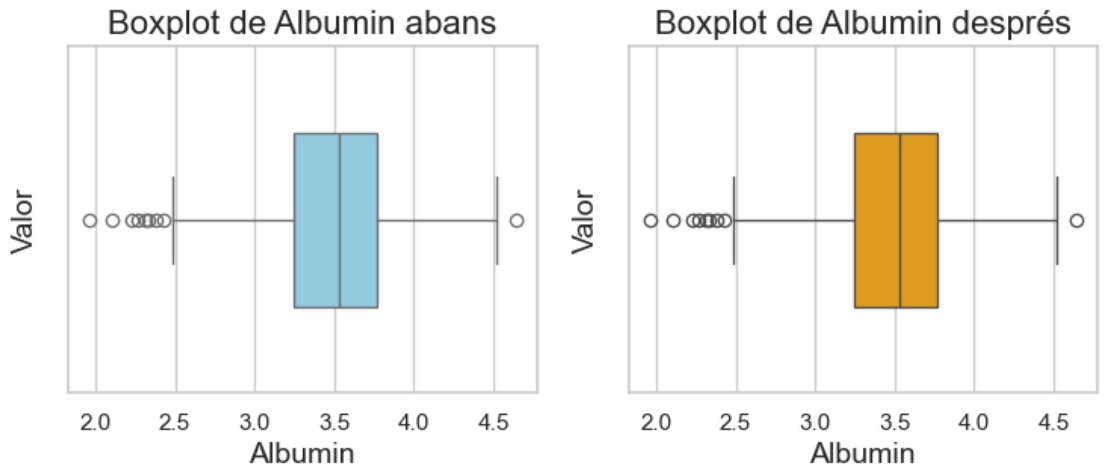
- **Cholesterol**

Alguns dels valors de la variable Cholesterol es troben molt més elevats que la mediana d’aquesta variable. Realitzant una recerca per internet no s’ha trobat una resposta completament vàlida per decidir treure o mantenir els outliers. Per tant, s’ha decidit ficar com a missing values aquelles dades que tenen nivells de colesterol majors a 1250 mg/dl, aquells 4 punts que es troben a l’extrem del boxplot.



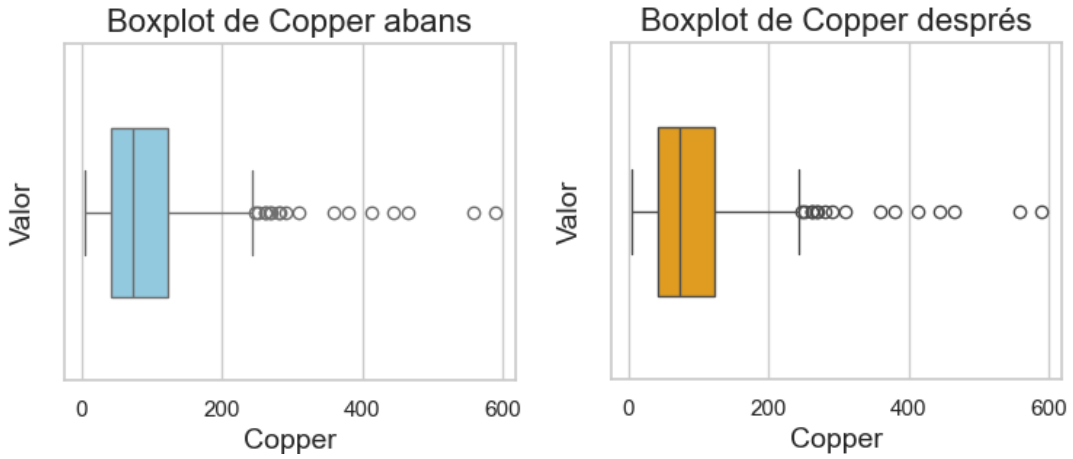
- **Albumin**

En aquesta variable, s'ha descobert que els nivells d'albumina menors a 2 es consideren nivells molt baixos però poden senyalar malalties hepàtiques com la cirrosis. Per tant, com el mínim valor d'albumina en aquest dataset és de 1.96 s'ha decidit no canviar cap outlier d'aquesta variable. També veiem un outlier en la part dreta del boxplot, però fent una recerca s'ha comprovat que els valors òptims d'albumina es troben entre 3.4 i 5.4 g/dl, per tant, com el valor màxim d'aquest dataset és 4.64, tampoc es canviarà aquest outlier.



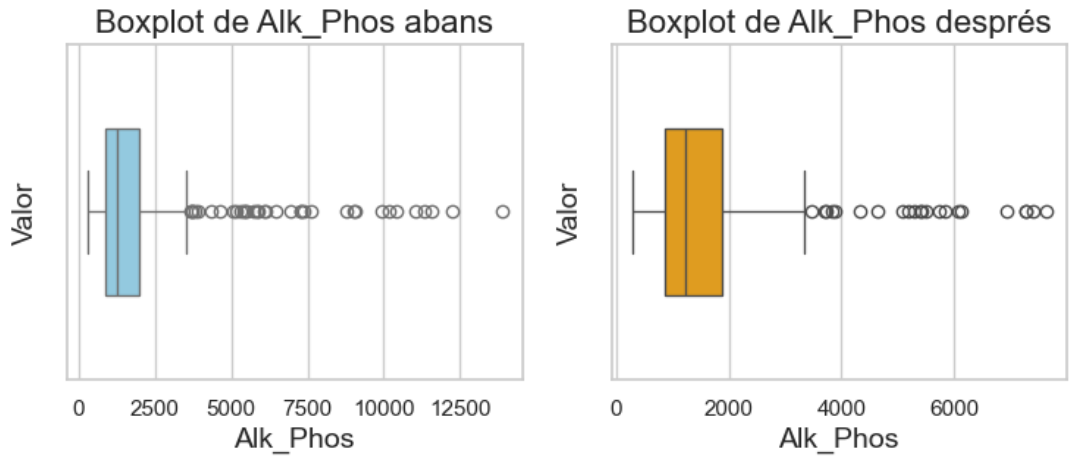
- **Copper**

S'ha investigat sobre els valors que pot tenir una persona de copper i s'ha trobat que els límits màxims establerts per la Junta d'Aliments i Nutrició són de 10.000 micrograms per dia. Per tant, en aquesta variable tampoc s'ha marcat com NA cap outlier, ja que no hi ha cap pacient que superi aquests valors.



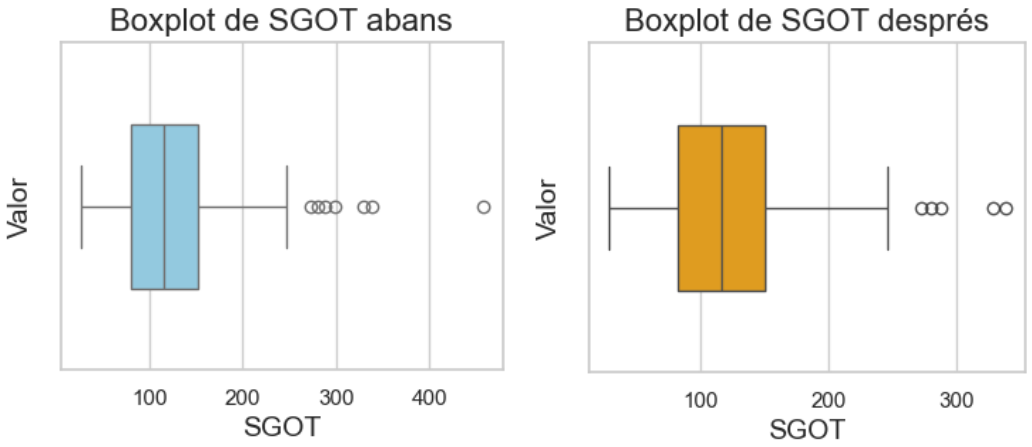
- **Alk_Phos**

En aquesta variable trobem bastants valors atípics. Trobem molts pacients que tenen nivells de fosfatasa alcalina molt alts i superiors a la mediana. Un rang normal de fosfatasa alcalina seria de 44 a 147 unitats per litre. Per tant, finalment s'han acabat marcant com valors faltants aquells outliers que estan més allunyats de la mediana, aquells valors més grans de 8000.



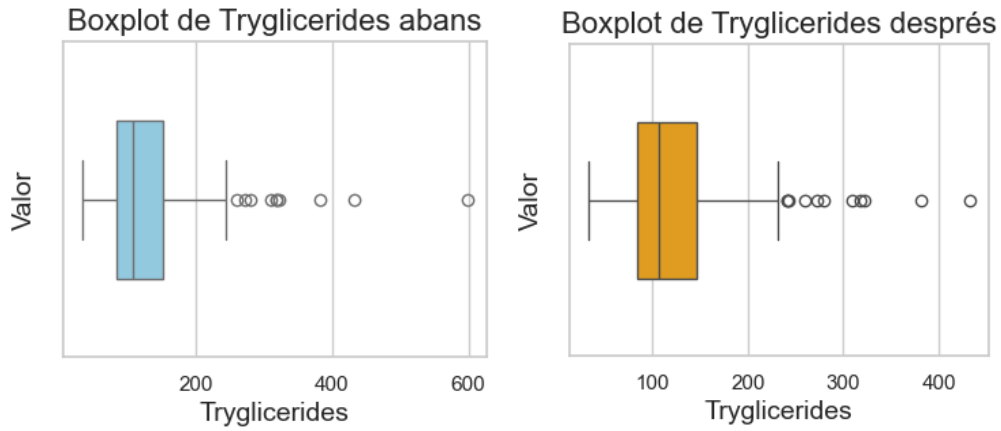
- **SGOT**

En aquesta variable trobem pocs valors atípics. Encara que trobem bastants pacients que tenen nivells per sobre del rang normal de 8 a 33 unitats per litre. Valors per sobre de 150 U/L indiquen complicacions greus amb la cirrosis. Per tant, finalment s'ha decidit marcar com missing value aquell valor que es troba molt més allunyat de la resta (457.25 U/L).



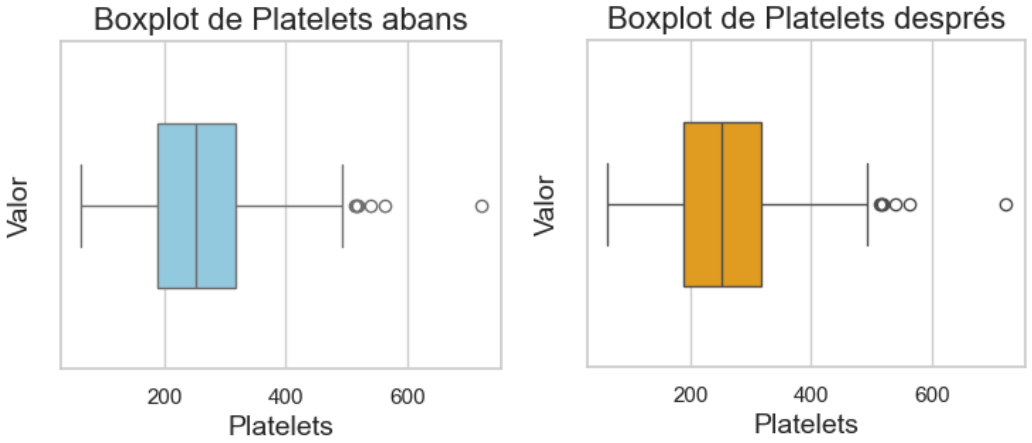
- **Triglicerides**

En aquesta variable també trobem valors atípics. Fent recerca, s'ha trobat que un nivell alt de triglicèrids seria entre 200 i 499 mg/dl, per tant, valors més alts de 500 mg/dl són considerats extremadament alts. Finalment en aquesta variable s'ha marcat com NA la dada d'aquell pacient que presenta un nivell de triglicèrids de 598 mg/dl ja que es considera un outlier amb un valor massa elevat.



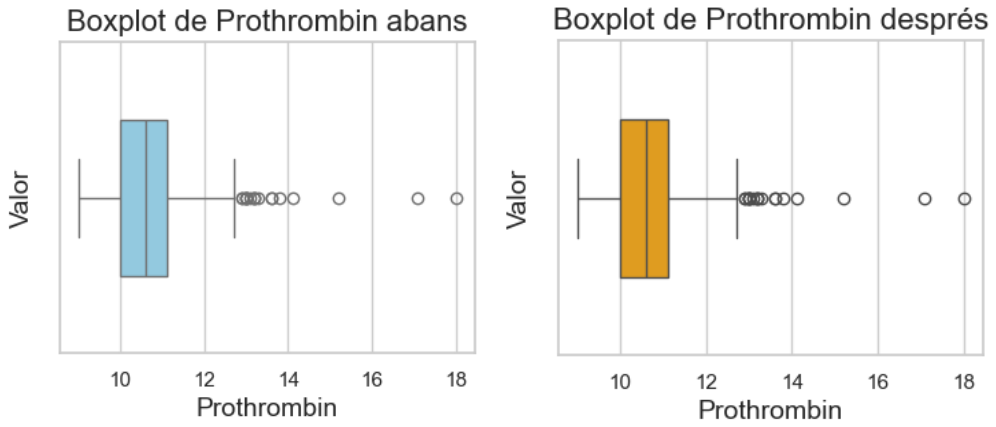
- **Platelets**

En aquesta variable trobem alguns outliers. Fent una recerca s'ha comprovat que un número normal de plaquetes oscil·la entre 150 i 450 plaquetes per microlitre de sang. Encara que valors més alts propers a 700 també podrien ser possibles. Per tant, no ficarem cap cap outlier com NA d'aquesta variable.



- **Prothrombin**

En aquesta variable també trobem alguns valors atípics. Fent recerca s'ha trobat que els temps de protrombina normalment oscil·la entre 10 i 13 segons. Estudiant el valor màxim de 18 segons també podria ser possible per tant no es marcarà com missing value.



Finalment, hem acabat marcant com missing values 19 outliers.
Cal destacar que a l'hora de provar el rendiment dels models també s'ha provat tractar els models sense cap modificació als outliers per veure si aquests realment aportaven informació important per la predicció de 'Status'. Tot i així, finalment s'han decidit marcar com missing values per després imputar-los ja que era la opció que millors rendiments donava.

2.3. Observació i tractament de missing values

Per a la realització dels models, s'ha fet una **partició de les dades** en tres parts: entrenament (train), prova (test) i validació (validation). Aquesta estratègia de partició és crucial per assegurar que el model no només aprengui de les dades, sinó que també pugui generalitzar bé dades noves i desconegudes.

El conjunt d'entrenament és el conjunt de dades més gran, en aquest cas s'ha triat que sigui un 70%, i es farà servir per construir i entrenar el model. Aquí, el model aprendrà a identificar els patrons i relacions de les dades. L'objectiu és que el model aprengui de manera profunda i eficaç, però sense memoritzar les dades de manera que perdi la capacitat de generalitzar a situacions noves.

El conjunt de prova s'utilitzarà un cop el model hagi estat entrenat. S'ha triat que sigui un 15% de les dades. Aquest conjunt no s'utilitzarà durant l'entrenament, per tant servirà per evaluar com de bé el model pot aplicar el que ha après a informació nova i desconeguda. És crucial per detectar l'overfitting.

El conjunt de validació s'utilitzarà per ajustar els hiperparàmetres (15% de les dades). Permetrà provar diferents combinacions d'hiperparàmetres per veure quines ofereixen el millor rendiment del model. Un cop trobats els hiperparàmetres ideals, es realitzarà una avaluació final amb el conjunt de prova.

La partició en aquests tres conjunts és fonamental per a **evitar l'overfitting** i per assegurar que els hiperparàmetres escollits són els que millor funcionen, no només per les dades amb les quals s'ha entrenat el model, sinó també per noves dades que el model pugui trobar en el futur. Això farà que el model sigui més robust i fiable.

Un cop s'ha fet la partició, ja es pot realitzar el tractament dels valors faltants. Abans del tractament de missing values s'ha observat i tractat els valors atípics ja que si tractem els valors faltants abans de tractar els outliers, pot ser que les tècniques d'imputació es vegin afectades. Els outliers podrien distorsionar els resultats portant una imputació incorrecta.

Per a les variables numèriques, s'ha utilitzat l'algoritme **KNN Imputer**. Aquest mètode d'imputació es basa en l'algoritme K-Nearest Neighbors. Tracta de trobar els k veïns més propers a un missing value i imputa aquesta dada basant-se en les característiques dels seus veïns. A més, s'ha triat que la k sigui 5 ja que és un número adequat per la mida del dataset.

Per a les variables categòriques, també s'ha optat per utilitzar el mètode **KNN Imputer** com a estratègia d'imputació. Aquest mètode, com ja s'ha dit abans, considera els veïns més propers de cada punt de dades amb valors mancants i utilitza aquests per a fer una imputació més informada.

Com es pot observar en el codi, hi ha una funció anomenada *imputacio_knn* on s'aplica el mètode d'imputació anomenat anteriorment. Cal destacar que s'ha triat el mètode d'imputació KNN ja que a diferència de la imputació per la moda o mitjana (segons si imputem a variables categòriques o numèriques respectivament), el KNN realitza una imputació més precisa ja que utilitza informació similar de l'entorn de la instància amb dades faltants. A més, la imputació per moda o mitjana pot introduir un biaix si el valor més freqüent o la mitjana no representa adequadament la distribució de les dades, especialment si hi ha molts outliers com és en aquest cas, en canvi el KNN Imputer és menys propens a aquest tipus de biaix.

Abans de procedir amb la imputació de les variables categòriques, ha estat necessària una codificació d'aquestes variables per poder-les passar a format numèric i poder aplicar el mètode d'imputació KNN ja que aquest requereix d'entrades numèriques. Per a aquesta conversió, s'han utilitzat les tècniques de codificació ***One Hot Encoding*** o ***Ordinal Encoder***, ja que permeten la traducció de dades categòriques a una representació numèrica sense pèrdua d'informació significativa.

La variable booleana anomenada *one_hot* permet indicar quin tipus de codificació es vol realitzar. Si es tria el mètode *One Hot Encoding*, aquesta variable s'indicarà com a *true*, en canvi si es vol l'altre mètode, la variable s'indicarà com a *false*. D'aquesta manera, es triarà quin tipus de codificació es vol aplicar i es cridarà a la funció *encode* per realitzar la codificació desitjada. Finalment s'observarà si té una influència en el rendiment del model, i si és així, es triarà la que millor rendiment doni.

La funció *encode* s'aplicarà a les variables categòriques de les tres particions del dataset per separat. Un cop ja s'hagi fet la codificació es concatenaran les variables numèriques de cada partició amb les variables que han sigut codificades i després s'aplicarà el mètode d'imputació de KNN a cadascuna de les particions concatenades.

Tot aquest procés es realitza en la funció *entrenament* que serà cridada un cop es vulgui executar el model.

2.4. Balanceig de dades

Després d'imputar els valors faltants, s'ha realitzat un balanceig de les dades en la variable objectiu "Status" ja que s'ha observat un desbalanceig de classes en aquesta variable. En aquest cas, la classe CL (al pacient se li ha realitzat un trasplantament de fetge) té moltes menors instàncies que les classes D (el pacient ha mort) i C (el pacient ha sobreviscut). Fent un recompte, s'ha observat que CL té 25 instàncies, D 161 i C 232 instàncies.

Aquest desbalanceig podria comportar un biaix en el model cap a les classes que presenten més instàncies i podria comportar a que el model no identifiqui correctament la classe minoritària.

Per tant, balancejar les dades permet al model aprendre de manera més efectiva les característiques de la classe CL. D'aquesta manera el model també generalitzarà millor.

Per fer aquest balanceig, es poden utilitzar diferents tècniques com el sobremostreig (oversampling) o el submostreig (undersampling). En aquest cas, com el dataset consta únicament de 418 files, s'ha decidit utilitzar oversampling per tractar el desequilibri de les classes.

S'ha prés aquesta decisió ja que el dataset és relativament petit i l'undersampling implicaria eliminar mostres i reduiria encara més la mida del dataset. Això podria portar a la pèrdua d'informació valuosa i reduir la capacitat del model per aprendre i generalitzar.

Per aquestes raons, s'ha considerat que l'oversampling és una opció més adequada. Aquest mètode augmenta la mida de la classe minoritària, afegint mostres artificials o replicant les existents, sense perdre informació de les altres classes.

S'ha optat per provar tant el random oversampling com *SMOTE* (Synthetic Minority Over-sampling Technique) per a realitzar l'oversampling.

El random oversampling tracta de duplicar les mostres aleatòries de la classe minoritària, mentre que l'SMOTE crea mostres sintètiques basant-se en els veïns més propers de la classe minoritària.

Utilitzant aquests dos mètodes d'oversampling, s'ha pogut comparar els resultats i determinar quina estratègia funciona millor per al nostre dataset.

Cal destacar que l'oversampling únicament es realitza en la partició del train (un cop ja s'ha fet la imputació).

3. Preparació de variables

En aquest apartat es veurà el procés de transformació i anàlisi de dades per optimitzar-les per l'ús de models predictius.

3.1. Normalització de variables

Per normalitzar les variables s'ha utilitzat ***StandardScaler*** de la biblioteca *scikit-learn*. S'ha aplicat aquest mètode de normalització ja que, a diferència d'altres mètodes, *StandardScaler* no canvia la forma de la distribució de la variable ja que no elimina els valors extrems, simplement reescala aquesta distribució perquè la mitjana sigui 0 i la desviació estàndard sigui igual a 1.

S'ha decidit utilitzar *StandardScaler* en comptes de *MinMaxScaler* ja que la transformació amb StandardScaler és menys sensible als valors extrems que *MinMaxScaler*, ja que aquest escala les dades en funció del valor mínim i màxim, per tant, un valor extrem pot distorsionar tota l'escala.

A més, alguns algoritmes com el *KNN* o *l'SVM*, es beneficien quan les característiques estan en la mateixa escala. En aquest cas *StandardScaler* pot ser més adequat ja que normalitza la variança a més de centrar les dades.

Així mateix, la utilització de *MinMaxScaler* podria acabar afectant el rendiment del model. Aquest mètode transforma les dades en un rang de [0,1], la qual cosa pot afectar als valors extrems ja que poden acabar sent comprimits en un rang molt estret.

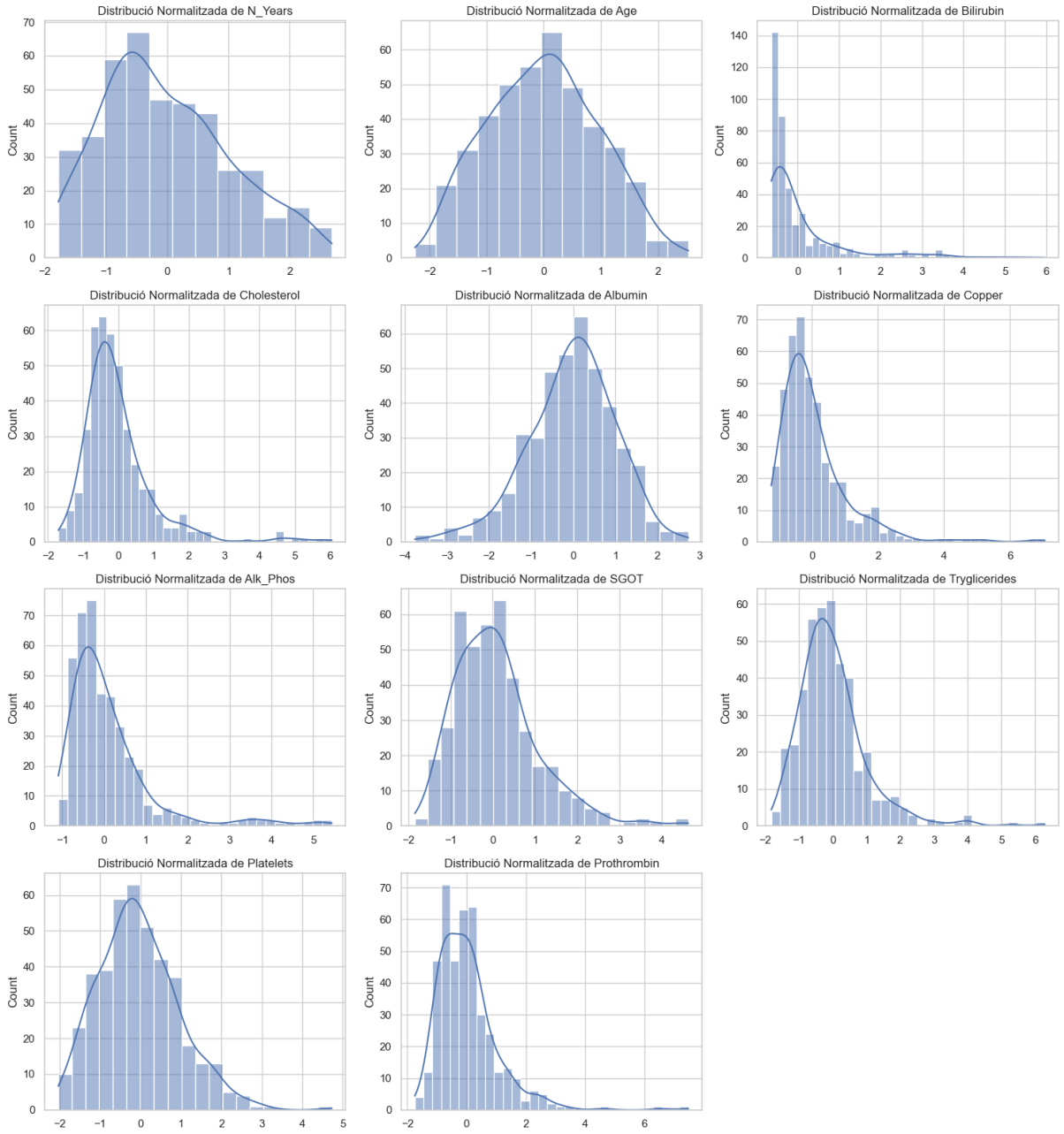
Per la realització dels models, s'ha realitzat l'opció de normalitzar les dades o no i així comprovar el seu rendiment.

Com ja s'ha mencionat anteriorment, en alguns algoritmes sí que és important la normalització de variables ja que poden ser sensibles a l'escala de les característiques. En aquest cas, el *KNN* i *l'SVM* són dos models que veurem més endavant, i que necessitaran normalització de les dades. En canvi, el *Decision Tree* no és sensible a l'escala de les característiques però més endavant veurem si normalitzar les dades o no afecta al rendiment del model.

Després de realitzar la normalització de variables, s'ha realitzat uns plots per veure les distribucions de les variables numèriques.

La normalització de variables s'ha realitzat en la funció *entrenament* un cop ja s'han codificat les variables categòriques com a numèriques (com s'ha indicat anteriorment) i abans de la imputació mitjançant *KNN Imputer*.

S'ha decidit realitzar la normalització de les dades abans d'imputar ja que el mètode utilitzat (*KNN Imputer*) fa servir la distància entre els punts per imputar valors faltants. Per tant, si les variables estan a diferents escales, les variables amb majors rangs numèrics podrien influir de manera desproporcionada en els resultats portant a una imputació poc precisa. Per això, la normalització assegura que totes les variables contribueixin d'una manera equitativa.

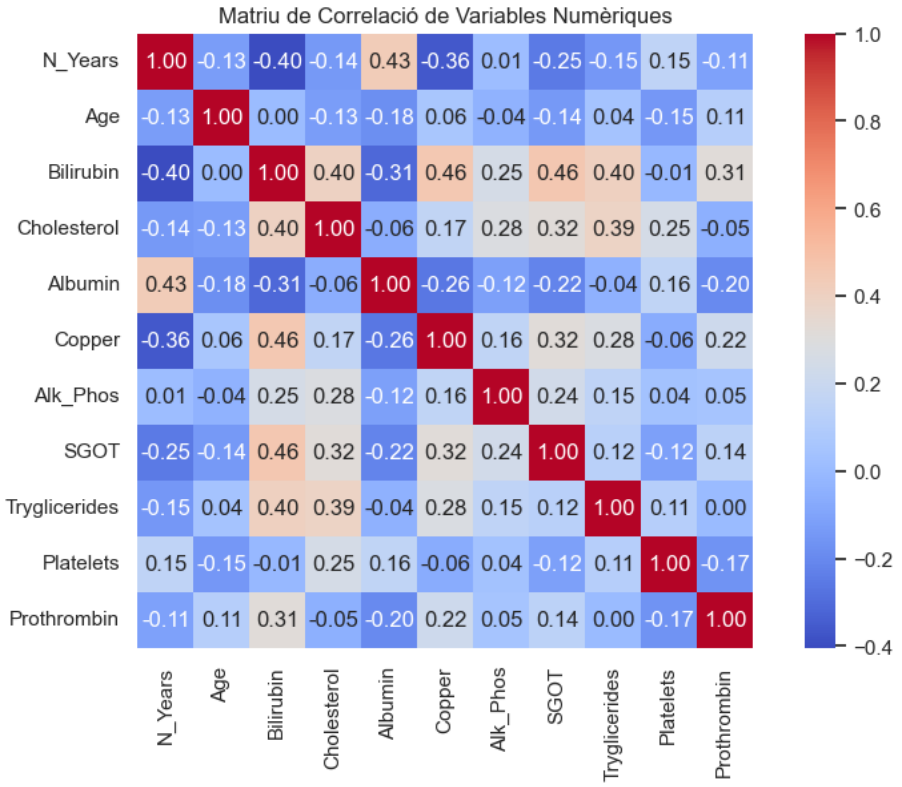


Com es pot observar, la imatge anterior ens mostra les distribucions de les variables numèriques un cop normalitzades amb StandardScaler.

Es pot comprovar que totes les distribucions estan centrades al zero, les variables mostren una variància normalitzada, amb la majoria de les dades caient dins del rang de -2 a 2. La qual cosa suggereix que la desviació estàndard de cada variable ha estat escalada a 1.

3.2. Anàlisi de correlacions

A continuació, s'ha realitzat una matriu de correlació per poder observar d'una manera general les relacions entre les variables numèriques del dataset.



S'aniran analitzant les correlacions de cada variable. Sabem que aquells valors propers al +1 o -1 indiquen correlacions fortes entre variables. Tot i que com es pot veure en aquesta matriu de correlacions, cap d'aquestes és major que 0.5 o menor a -0.5. Aquelles variables amb correlacions properes al 0 indiquen que hi ha una correlació dèbil. Cal destacar que en una matriu de correlacions s'observen les relacions lineals entre variables, és a dir, tot i que entre variables no hi hagi relacions lineals altes, podrien existir relacions no lineals entre elles.

La variable *N_Years* té una alta correlació amb la variable *Albumin* (0.43). Això pot ser degut a la importancia de l'albumina en el progrés de la cirrosis. En aquesta malaltia, els nivells d'albumina solen disminuir degut a la funció hepàtica deteriorada i la síntesi reduïda d'albumina. Tenir l'albumina en nivells baixos està associat a major mortalitat i complicacions en la cirrosis. Per tant, una major duració de la malaltia (*N_Years*) podria estar relacionat a tenir nivells baixos d'albumina.

La variable *N_Years* també té correlacions negatives bastant fortes amb *Bilirubin* i *Copper*. Això pot ser degut a que a mesura que la cirrosis avança, la funció hepàtica empitjora, el que pot comportar a un augment de la bilirrubina i del coure en sang.

La variable *Age* sembla tenir una correlació molt baixa amb la majoria de les altres variables, amb valors que entre -0.13 a 0.15. Això indica que l'edat no està altament relacionada amb les altres variables clíniques del dataset.

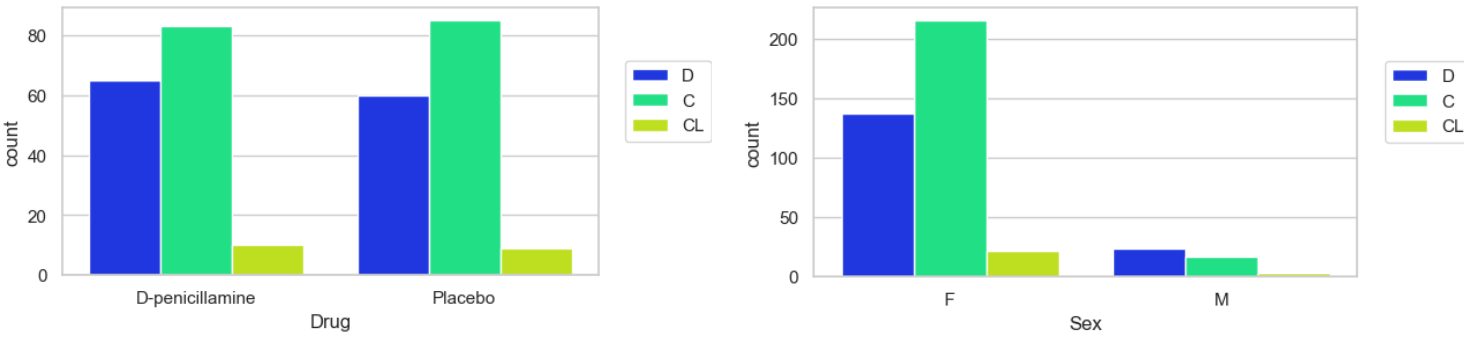
Es pot observar que la variable Bilirubin té una forta correlació amb quatre variables: *Cholesterol* (0.4), *Trygliceride*s (0.4), *SGOT* (0.46) i *Copper* (0.46). La correlació entre aquesta variable i *Cholesterol* i *Trygliceride*s es degut a que aquests marcadors bioquímics tenen grans interaccions metabòliques amb la funció hepàtica. Aquells pacients amb alts nivells de bilirrubina, solen tenir nivells alts de colesterol i triglicèrids. La correlació entre la variable *Bilirubin* amb *Copper* i *SGOT*, pot ser deguda a la funció del fetge en el metabolisme. En la cirrosis, la funció hepàtica deteriorada pot comportar un augment de bilirrubina i nivells de coure i d'enzims hepàtics com l'SGOT.

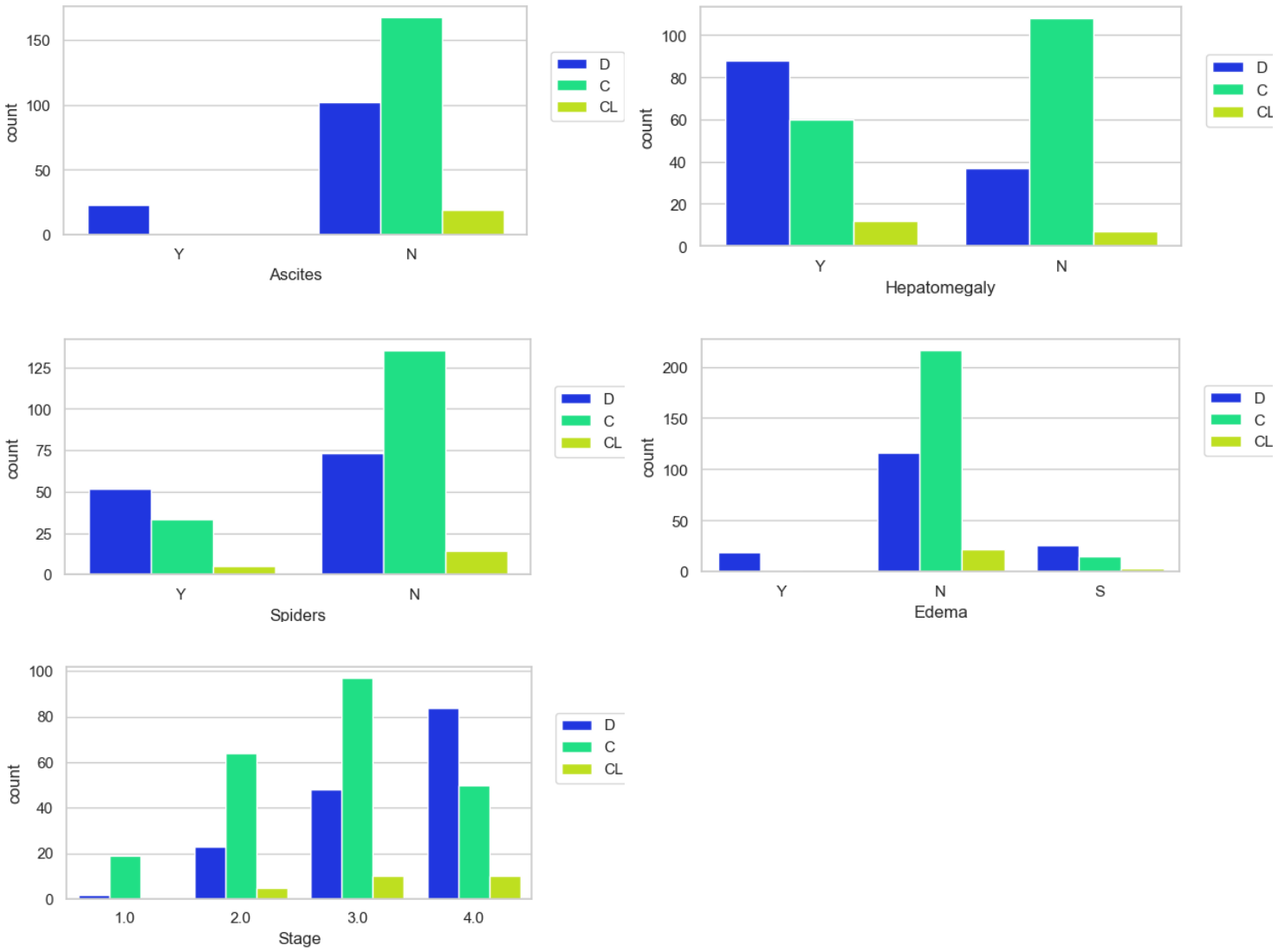
Altres variables com *Platelets* o *Prothrombin* no tenen correlacions gaire fortes amb les altres variables. Les plaquetes i el temps de protrombina estan relacionats amb la coagulació sanguínia i la funció hepàtica, però poden no estar directament influenciades per factors que afecten altres marcadors bioquímics. Això indica que poden representar processos independents o menys directament relacionats amb altres aspectes de la cirrosis.

L'anàlisi d'aquesta matriu de correlacions és un procés important per identificar relacions entre variables i possibles variables predictives. Cal destacar que la presència de correlacions febles pot indicar que aquestes variables ofereixen una visió única de l'estat del pacient, la qual no es pot capturar simplement analitzant altres marcadors, i per tant, també han de ser considerades en els models predictius.

3.3. Anàlisi de variables categòriques i variable objectiu

També s'ha realitzat un anàlisi bivariat sobre les variables categòriques i la variable objectiu.





La primera gràfica mostra la relació entre el tipus de tractament (D-penicillamine o Placebo) i l'estat dels pacients (viu, mort o viu amb trasplantament de fetge). A primera vista, sembla que hi ha una distribució similar entre els pacients que estan vius i els que han mort, independentment del tipus de tractament.

La segona gràfica relaciona el gènere dels pacients amb l'estat. Aparentment, hi ha una major proporció de pacients de gènere femení que viuen en comparació amb els pacients masculins. Però això és degut a que hi ha moltes més instàncies de pacients del gènere femení en comparació amb el masculí. Per tant, no es podria afirmar que el sexe afecta a viure o morir en situacions de cirrosis.

La tercera gràfica mostra la relació entre la presència d'ascites i l'estat dels pacients. Aquí s'observa que una majoria de pacients sense ascites estan vius, mentre que entre els pacients amb ascites, únicament hi ha pacients morts. Encara que, com en el cas anterior, hi ha moltes menys mostres de pacients amb ascites, per tant, no es podria afirmar que si el pacient té ascites té una probabilitat de morir del 100%. Caldrien més mostres per afirmar-ho.

La quarta gràfica mostra la relació entre la hepatomegalia (augment de la mida del fetge) i l'estat dels pacients. En aquest cas no trobem una tendència clara pero sí que podem veure que la majoria dels pacients sense hepatomegalia són vius. En canvi, la proporció de morts augmenta quan els pacients sí que presenten aquest augment de la mida del fetge.

La cinquena gràfica mostra la relació entre la presència d'Spiders (dilatacions vasculars en la pell) i l'estat dels pacients. Com es pot observar, hi ha una major quantitat de pacients vius en aquells pacients que no presenten Spiders. Mentre que la proporció de morts augmenta i és més gran que la de vius quan els pacients si que presenten Spiders.

La sisena gràfica mostra la relació entre edema (inflamació causada per acumulació de líquid) i l'estat del pacient. Com es pot observar, la majoria de pacients que no presenten edema estan vius. A més, sembla haver pocs casos d'edema en pacients que se'ls ha realitzat un transplantament, la qual cosa podria indicar la qual cosa podria indicar que l'edema podria ser un factor que afecta a la decisió o realitzar un trasplantament.

La última gràfica, mostra la relació entre les etapes de la cirrosis desde la 1 fins la 4 i l'estat dels pacients. Aquí es pot observar una clara tendència. A mesura que augmenta la gravetat de la cirrosis, augmenta el número de morts i transplantaments de fetge. En les tres primeres etapes de la malaltia veiem que el número de pacients vius és bastant major a la resta de classes. En canvi, en l'última etapa de la malaltia, s'observa que el número de vius disminueix significativament.

3.4. Eliminació de variables redundants

En aquesta secció estudiarem les variables categòriques i la informació significativa que aquestes ens aporten pel nostre model de predicció. No analitzarem les variables numèriques ja que ja s'ha estudiat la relació entre aquestes anteriorment. A més, també s'estudiarà la significància d'aquestes pròximament amb l'Anàlisi de Components Principals.

Per poder veure la significància de les variables categòriques respecte la variable objectiu, realitzarem la prova del chi-quadrat. Aquesta tècnica estadística ens permet determinar si hi ha una associació significativa entre les diferents variables categòriques i la variable objectiu 'Status'. A diferència de les variables numèriques, les variables categòriques no poden ser analitzades amb mètodes com la correlació. Per tant, la prova de chi-quadrat és un mètode ideal per a aquesta finalitat.

Ara s'analitzaran els resultats de les proves realitzades:

Com hem pogut observar en l'apartat anterior de l'anàlisi de variables categòriques, la variable *Drug*, ens indica que independentment del tipus de tractament aplicat, la tendencia de morir o viure és la mateixa. Amb la prova del chi-quadrat hem pogut comprovar que no hi ha una associació estadísticament significativa entre el tipus de medicació i l'estat del pacient. El valor del chi2 és de 0.225 i el valor del p-value és de 0.894.

La variable *Sex* ens mostra un valor de chi2 de 5.858 i un p-value de 0.053. Aquest resultat està a prop del llindar comú de significació (0.05), suggerint que pot haver una associació lleugerament

significativa entre el sexe del pacient i el seu estat, però no és estadísticament significativa al nivell del 5%.

Tota la resta de variables categòriques (*Ascites*, *Hepatomegaly*, *Spiders*, *Edema*, i *Stage*) presenten un p-valor menor a 0.05 i pràcticament 0. Això ens indica que aquestes variables sí tenen associacions estadísticament significatives amb l'estat del pacient i són considerades importants pel nostre model predictiu.

Finalment, després de realitzar les proves de chi-quadrat, s'ha decidit que s'eliminarà la variable *Drug,* ja que, com hem vist en l'apartat anterior i aquest, aquesta variable no ens aporta cap informació important per a la predicció que volem realitzar. El p-valor obtingut a la prova de chi-quadrat per a la variable ha sigut de 0.894, molt per sobre del llindar comunament acceptat de 0.05 per a la significació estadística.

Incloure variables que no mostren una relació significativa amb la variable objectiu podria conduir a models sobreajustats, menys eficients i que poden donar una importància incorrecta a factors que en realitat no influeixen en el resultat. Per tant, eliminant la variable "Drug", podem simplificar el model, millorar la seva interpretabilitat i incrementar la seva precisió.

3.5. Estudi de dimensionalitat amb PCA

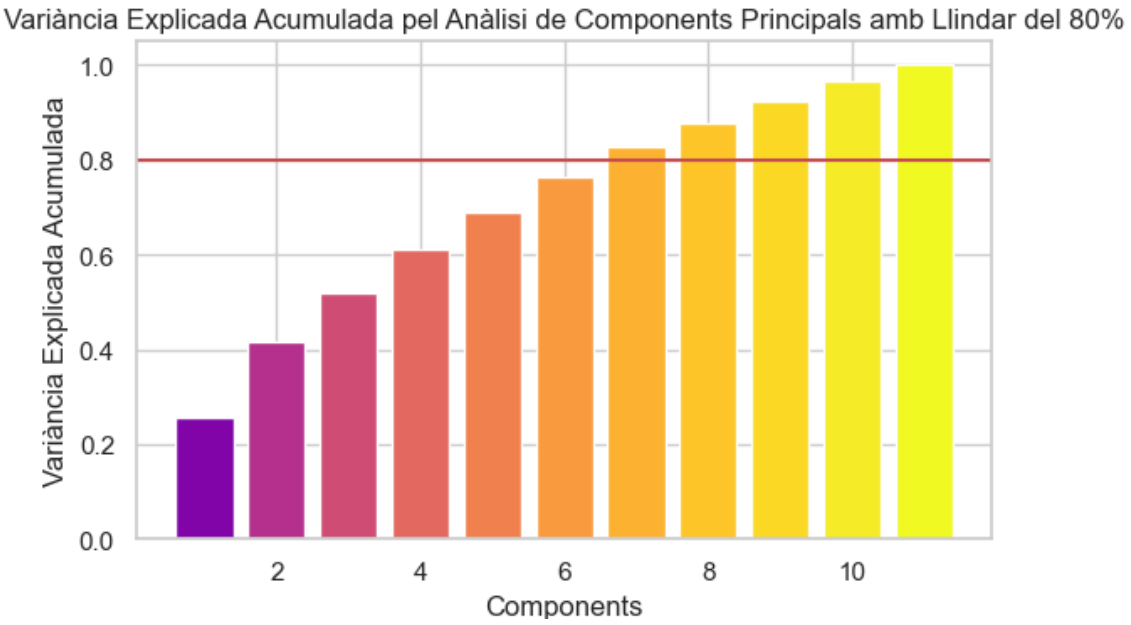
S'ha realitzat una funció anomenada *pca* per realitzar els gràfics necessaris per l'avaluació de la tècnica de reducció de dimensionalitat PCA (Anàlisi de Components Principals). Aquesta tècnica es realitza únicament amb variables numèriques, per tant, s'ha decidit eliminar les variables categòriques per no tenir-les en compte.

L'Anàlisi de Components Principals és una tècnica estadística utilitzada per reduir la dimensionalitat de les dades mentre es conserva la major part de la informació. El PCA busca transformar el conjunt original de variables en un nou conjunt de variables no correlacionades, anomenades components principals. Aquestes components són obtingudes de manera que el primer component tingui la major variància possible (és a dir, explica la major part de la variabilitat en les dades), i cada component següent tingui la màxima variància possible sota la restricció que sigui ortogonal als components anteriors. Aquesta tècnica és particularment útil per visualitzar i analitzar dades en dimensions més manejables.

El primer gràfic realitzat, ens permet visualitzar la variància acumulada explicada pels components principals obtinguts mitjançant el PCA. Cada barra d'aquest gràfic representa la variància acumulada per cada component. La línia horitzontal indica el llindar del 80%, que s'ha realitzat per observar quants components calen per aconseguir el 80% de la variància acumulada, i en aquest cas podem veure que són necessaris 7 components.

Com es pot observar, per capturar la major part de la informació no són necessaris tots els components del PCA. A més, a mesura que augmenta el número de components, la variància augmenta més lentament. Aquest fet, reflecteix que cada component successiu captura menys informació que el component anterior.

Dins la variable *var_exp*, s'ha afegit la variància explicada per cada component principal, on cada valor representa la proporció de variància total del conjunt de dades. El primer component principal explica aproximadament el 25.54% de la variança total de les dades. El segon component principal explica al voltant del 15.97% de la variança total. El tercer component explica aproximadament el 10.26%, i així successivament. Si sumem tots aquests valors, obtindrem la variància total explicada pels 11 components principals. Aquesta suma hauria de ser igual a 1.

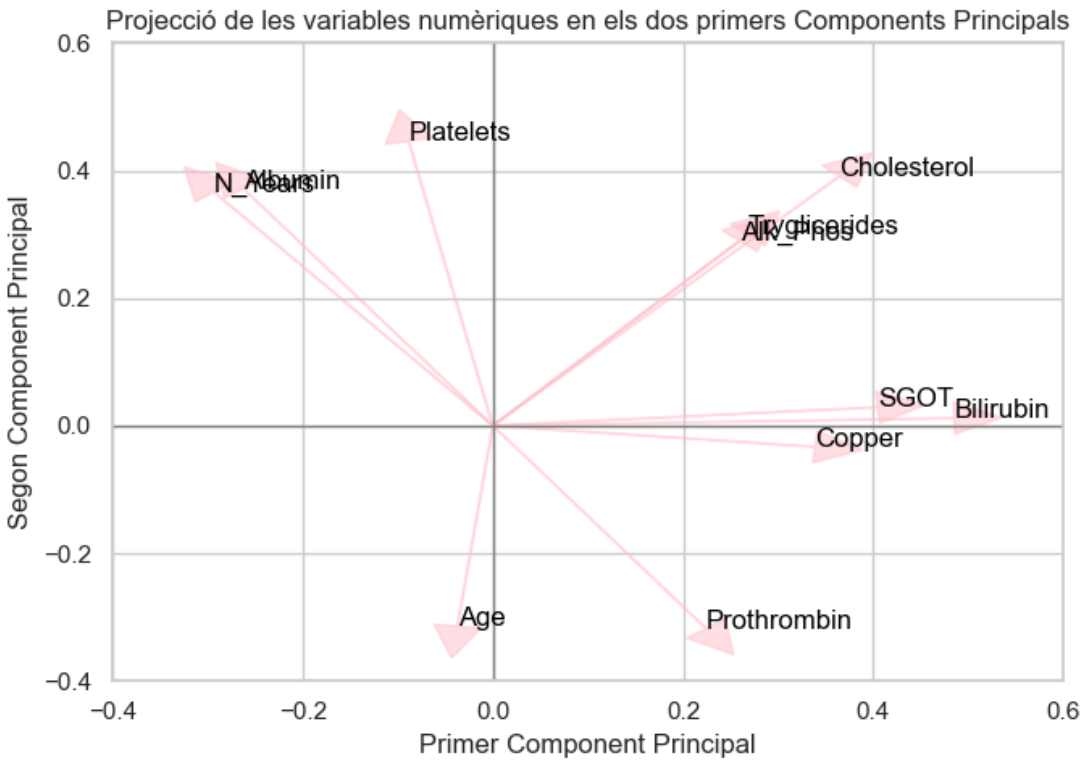


Tot i així, s'ha decidit mantenir tots els components per la realització de models. Cal destacar que el dataset únicament està compost per 418 files. Per tant, encara que set components puguin capturar el 80% de la variància, el 20% restant pot contenir informació important per la predicció de la variable objectiu. A més, la reducció de dimensions podria comportar a un overfitting.

En el següent imatge es pot veure una gràfica de vectors que projecta les variables numèriques en l'espai definit pels primers dos components principals obtinguts a partir del PCA.

Les variables que apunten en la mateixa direcció vol dir que estan significativament correlacionades, tendeixen a augmentar o disminuir juntes. Com es pot veure a la imatge, les variables N_Years i Albumin estan fortament correlacionades, la qual cosa es va poder observar a la matriu de correlacions. Una altra relació que també es va veure a la matriu de correlacions és la relació significativa entre les variables *SGOT*, *Bilirubin* i *Copper*. Per últim, les variables *Cholesterol*, *Tryglicerides* i *Alk_Phos* també semblen tenir una correlació molt forta en els dos primers components principals, les quals en la matriu de correlacions únicament s'havia detectat una correlació alta entre *Cholesterol* i *Tryglicerides*.

Les variables que apunten en direccions oposades estan negativament correlacionades. Com es pot observar en la imatge, les variables que van en direccions oposades són: *Platelets* i *Age* i Albumin amb N_Years i Prothrombin. Com es pot observar a la matriu de correlacions totes aquestes variables tenen correlacions negatives, indicant el que podem veure a la imatge.


Projecció de les variables numèriques en els dos primers Components Principals
Segon Component Principal
Primer Component Principal
0.6
0.4
0.2
0.0
−0.2
−0.4
−0.4
−0.2
0.0
0.2
0.4
0.6
Platelets
Cholesterol
Albumin
Triglycerides
Alk_Phos
SGOT
Bilirubin
Copper
Age
Prothrombin

4. Definició de models

Un cop ja s'ha fet tot el procés de preprocessament i anàlisi de les dades, es procedirà a la definició i entrenament de 3 models, K Nearest Neighbors (KNN), un arbre de decisió, i un Support Vector Machine (SVM). L'objectiu principal és seleccionar el model més adequat que ens permeti predir amb precisió l'estat dels pacients a partir de les dades proposades.

Per fer això, es realitzarà un estudi per seleccionar les mètriques més adequades per avaluar els models, s'entrenaran cadascun amb el mateix conjunt de dades, es farà un ajust dels hiperparàmetres necessaris per cada model per poder millorar els rendiments i finalment es seleccionarà el model més prometedor basant-nos en les mètriques de valoració definides.

Abans de començar amb la definició dels models, com ja s'ha mencionat abans, cal destacar que s'han fet tres particions del dataset, entrenament, validació i test.

4.1. Definició de mètriques

A l'hora d'avaluar models en el context de la predicció de l'estat dels pacients amb cirrosi, la selecció de mètriques apropiades és fonamental per assegurar una valoració precisa i fiable del rendiment del model. Trobem les següents mètriques que es consideren generalment com les més adequades:

- **Precisió (Accuracy)**

Aquesta mètrica mesura la proporció de prediccions correctes (tant positives com negatives) en comparació amb el total de casos. En situacions on hi ha un gran desequilibri entre les classes, la precisió pot ser enganyosa. En aquests casos, el model podria simplement predir sempre la classe majoritària i encara així obtenir una alta precisió, tot i no ser capaç de captar adequadament les característiques de la classe minoritària.

- **Sensibilitat (Recall)**

Aquesta mètrica mesura la proporció de casos positius reals que han estat correctament identificats pel model. La sensibilitat es calcula com el nombre de veritables positius dividit pel nombre total de casos que realment són positius (veritables positius més falsos negatius). Això ens dóna la proporció de positius reals que el model ha detectat correctament. Tot i que una alta sensibilitat és desitjable, no s'ha de considerar en aïllament. Una alta sensibilitat amb una baixa especificitat pot conduir a molts falsos positius.

- **F1-Score**

Aquesta mètrica proporciona una única puntuació que resumeix la capacitat del model de classificar correctament les instàncies positives, considerant tant la seva precisió com la seva sensibilitat. El F1-Score es calcula com la mitjana harmònica de la precisió i la sensibilitat. En datasets on hi ha un desbalanceig de classes, la precisió o la sensibilitat per si soles poden donar una idea enganyosa del rendiment del model. El F1-Score, al considerar ambdues mètriques, esdevé particularment valuós en aquestes situacions, proporcionant una visió més equilibrada del rendiment. Malgrat que aquesta

mètrica ajuda a equilibrar la precisió i la sensibilitat, no proporciona informació específica sobre el tipus d'error (fals positius vs falsos negatius) que el model està cometent.

- **Àrea Sota la Corba ROC (AUC-ROC)**

La corba ROC (Receiver Operating Characteristic) és una representació gràfica que mostra la relació entre la sensibilitat (taxa de veritables positius) i l'especificitat (1 - taxa de falsos positius) del model a diferents llindars de classificació. L'AUC proporciona una única puntuació que resumeix el rendiment del model sobre tots els possibles llindars. Un AUC de 1 representa un model perfecte que classifica correctament totes les instàncies positives i negatives. Un AUC de 0.5 indica un rendiment no millor que l'atzar. A diferència de la precisió, la corba ROC és menys sensible al desbalanceig de les classes.

- **Matriu de confusió**

Aquesta eina analítica no és una mètrica numèrica a diferència de les altres, però proporciona una visió detallada del rendiment del model en termes de la seva capacitat per classificar correctament o incorrectament les instàncies. Aquesta matriu té diferents components: **True Positive** (el nombre de casos que el model ha predit correctament com a positius, i realment són positius), **False Positive** (els casos que el model ha predit com a positius, però en realitat són negatius), **True Negative** (el nombre de casos que el model ha predit correctament com a negatius, i realment són negatius) i **False Negative** (els casos que el model ha predit com a negatius, però en realitat són positius).

Per l'avaluació dels nostres models tindrem en compte totes les mètriques mencionades anteriorment.

4.2. KNN

4.2.1. Motivació del model triat

El primer model a explorar s'ha triat que sigui K Nearest Neighbors. El KNN és un dels models més simples i intuïtius en machine learning. Es basa en el principi que les instàncies similars amb característiques semblants tendeixen a pertànyer a la mateixa categoria. KNN no fa suposicions prèvies sobre la distribució de les dades, és a dir, no assumeix normalitat.

El principal hiperparàmetre en KNN és el nombre de veïns (k). Altres hiperparàmetres inclouen la mètrica de distància (com euclidiana o manhattan) i el pesatge dels veïns.

4.2.2. Hiperparàmetres

És important analitzar els diferents hiperparàmetres possibles que pot tenir el model KNN. Aquest té diversos hiperparàmetres que poden afectar significativament el seu rendiment. Els paràmetres que s'han seleccionat són: *n_neighbors*, *weights* i *metric*.

Taula d'hiperparàmetres i valors provats

Hiperparàmetre	Descripció	Valors Provats
n_neighbors	Nombre de veïns utilitzats per la predicció.	5, 7, 9, 11
weights	Pes assignat als veïns.	'uniform', 'distance'
metric	Mètrica utilitzada per calcular la distància.	'euclidean', 'manhattan'

- **n_neighbors**

Aquest hiperparàmetre especifica el nombre de veïns a considerar quan es fa una predicció. Un nombre petit de veïns pot fer que el model sigui més sensible al soroll de les dades, mentre que un nombre massa gran pot portar a un model massa generalitzat. Per tant, considerant que el nostre dataset no té gaires files, s'han triat de provar els valors 5,7,9, i 11. Tots valors senars per no generar empats en la classificació.

- **weights**

Aquest hiperparàmetre determina com es pondera cada veí a la predicció final. Amb 'uniform', tots els veïns contribueixen de manera igual. Mentre que amb 'distance' s'assigna una major influència als veïns més propers.

- **metric**

Aquest hiperparàmetre defineix la métrica de distancia utilitzada per trobar els veïns. Amb 'euclidean', es calcula la distància euclidiana, que es calcula com la longitud de la línia recta entre dos punts a l'espai n-dimensional. Mentre que 'Manhattan', es calcula com la suma de les diferències absolutes entre les coordenades dels punts.

Aquests hiperparàmetres estan en una variable anomenada *parametres_knn* la qual és un diccionari que emmagatzema totes les possibles opcions d'hiperparàmetres.

Perquè els millors hiperparàmetres siguin triats, a la funció *entrenament* s'ha realitzat un *for* per anar recorrent els paràmetres del diccionari. Entrenarem el model amb els primers hiperparàmetres i realitzarem les prediccions amb el conjunt de validació. Es guarden els resultats del F1 Score en la variable *best_score* i els paràmetres corresponents a *best_params.* S'ha decidit triar la mètrica d'F1 Score per avaluar els hiperparàmetres ja que considera tant la precisió com la sensibilitat. A mesura que es va recorrent el *for*, el millor resultat d'F1 Score es guardarà juntament amb els seus paràmetres fins que finalment al acabar el bucle ja es tindran els millors paràmetres guardats.

4.2.3. Entrenament amb train

Cal destacar que a part de provar els diferents hiperparàmetres, per veure de quines maneres el rendiment del model és millor, també s'ha provat de donar la opció de triar si es vol normalitzar les dades, aplicar oversampling i realitzar one hot encoding o ordinal encoder per codificar les variables categòriques.

S'ha creat una funció anomenada *entrenament* com ja hem mencionat anteriorment. Aquesta tindrà com a paràmetres la variable *parametres_knn* que conté tots els hipèrparàmetres possibles, la variable *stand* que indica si volem normalitzar (true) o no (false), la variable *apply_oversampling* que indica si volem realitzar oversampling mitjançant SMOTE (true) o no (false) i per últim la variable *one_hot* que indica si volem realitzar la codificació de variables categòriques mitjançant One Hot Encoding (true) o Ordinal Encoding (false).

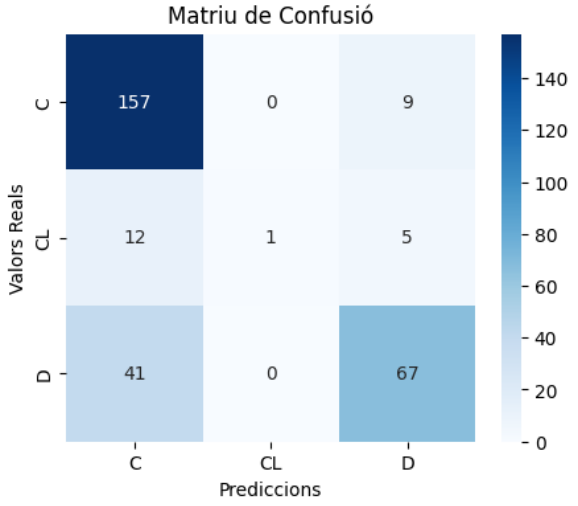
En aquest apartat ens fixarem en les mètriques obtingudes amb la partició train.

Els **millors paràmetres triats** en el model KNN han sigut els següents: {'metric': 'euclidean', 'n_neighbors': 7, 'weights': 'uniform'}

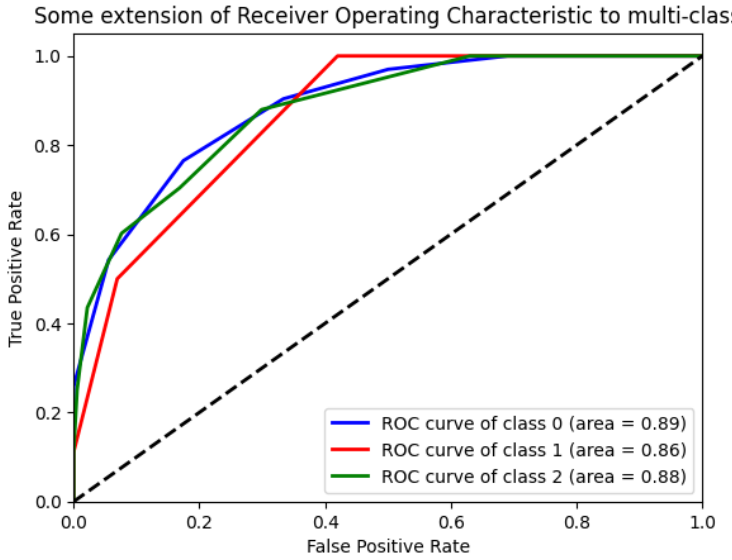
Un cop s'han triat aquests paràmetres, s'ha entrenat el model amb model.fit (model ha estat definit com KNeighborsClassifier()). També s'ha decidit triar que les dades **si seran normalitzades** ja que pel KNN és necessari, no s'aplicarà oversampling i es farà codificació mitjançant ordinal encoder, ja que després d'anar provant les combinacions, s'ha arribat a la conclusió que aquesta és la que millor rendiment dona.

Seguidament s'han mirat les mètriques per avaluar el rendiment del conjunt d'entrenament:

- **Accuracy en entrenament:** 0.7705
- **F1-Score en entrenament:** 0.7434
- **Recall en entrenament**: 0.7705
- **Matriu de confusió en entrenament:**



- **Corba ROC en entrenament:**



Com es pot observar en els resultats presentats, la matriu de confusió ens mostra que la classe "C" ha estat majoritàriament ben classificada amb 157 casos correctament identificats. La classe "D" també ha estat relativament ben classificada amb 67 casos correctament identificats. Però ha tingut els següents errors de classificació: La classe "C" té 9 casos que han estat erròniament classificats com a "D", per tant podem dir que la classe C la classifica pràcticament perfecta. La classe "D" té 41 casos que han estat erròniament classificats com a "C", la qual cosa ens indica que té un marge bastant gran d'error a l'hora de classificar la classe "D". Finalment, la classe "CL" presenta dificultats amb una majoria d'errors en la seva classificació (12 casos com a "C" i 5 com a "D"). Podem afirmar que la classe "CL" sembla ser la més difícil de classificar per al model, això és degut a que al ser la classe amb menys instàncies, el model té molt problemes per classificar-la.

En l'avaluació de la corba ROC es pot veure que totes les classes tenen una AUC bastant alta (0.89, 0.86 i 0.88), la qual cosa indica una bona capacitat del model per distingir entre les classes. No obstant això, la classe amb la corba ROC vermella (classe CL en aquest cas) té una AUC lleugerament més baixa, el que podria correlacionar-se amb la major dificultat observada en la matriu de confusió.

També es veu que l'accuracy indica que voltant del 77% de les prediccions eren correctes. Això és relativament alt, però cal considerar el desequilibri de classes, ja que això podria estar inflant aquesta mètrica si una classe és predominant. El F1-Score, que harmonitza la precisió i el recall, és una mica més baix (0.74), suggerint que hi ha un desequilibri entre la precisió i el recall del model. El recall també és de 0.77, el mateix que l'accuracy, indicant una capacitat consistent del model per identificar les instàncies positives.

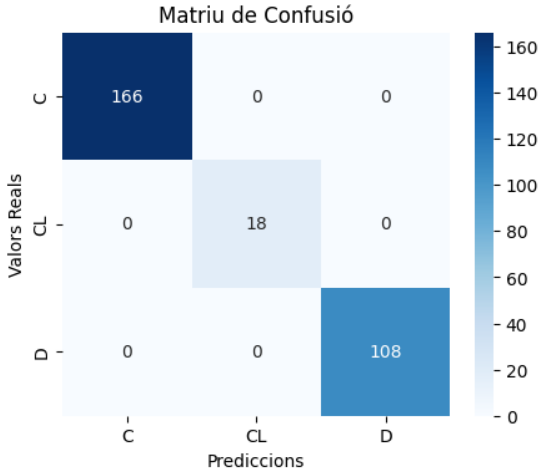
Com es pot observar, el model té problemes en classificar la classe "CL" (si el pacient ha rebut trasplantament de fetge). Per tant, en aquest model observarem com queden les mètriques si apliquem oversampling:

Els millors hiperparàmetres obtinguts són els següents: {'metric': 'manhattan', 'n_neighbors': 11, 'weights': 'distance'}

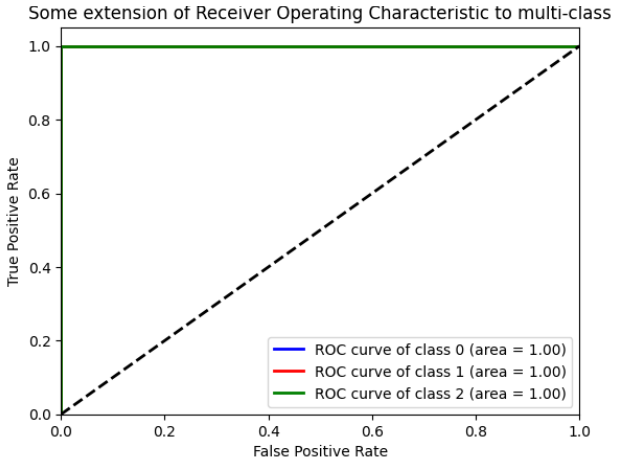
I les mètriques obtingudes són:

- Accuracy en entrenament: 1.0
- F1-Score en entrenament: 1.0
- Recall en entrenament: 1.0

- Matriu de confusió



- Corba ROC



Com es pot observar totes aquestes mètriques indiquen que el model classifica les classes perfectament sense cap tipus de marge d'error. Això indica que el model està fent un overfitting extrem de les dades. Pot ser degut a que amb l'oversampling, especialment amb mètodes com SMOTE, es poden crear moltes instàncies artificials que poden causar que el model aprengui massa bé les característiques d'aquestes instàncies artificials, portant a un overfitting.

4.2.4. Anàlisi de resultats al validation i test

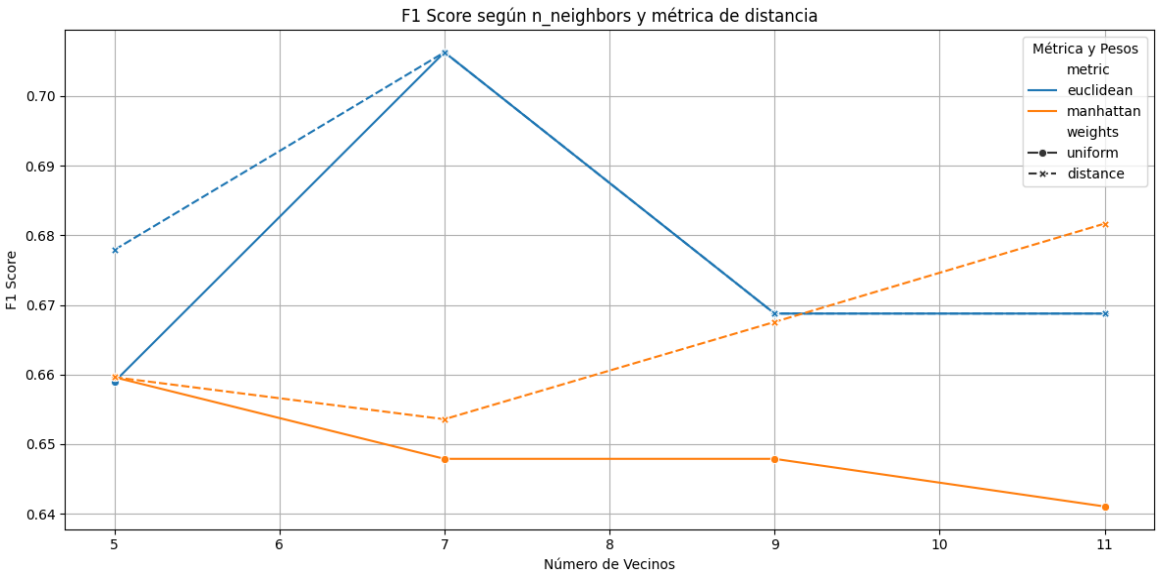
En aquest apartat s'observaran els resultats obtinguts en els conjunts de validació i test.

S'ha realitzat un gràfic per observar com actuen els diferents hiperparàmetres en el conjunt de validació. Finalment s'ha triat l'execució de model sense aplicar oversampling ja que no volem que hi hagi un overfitting de les dades.

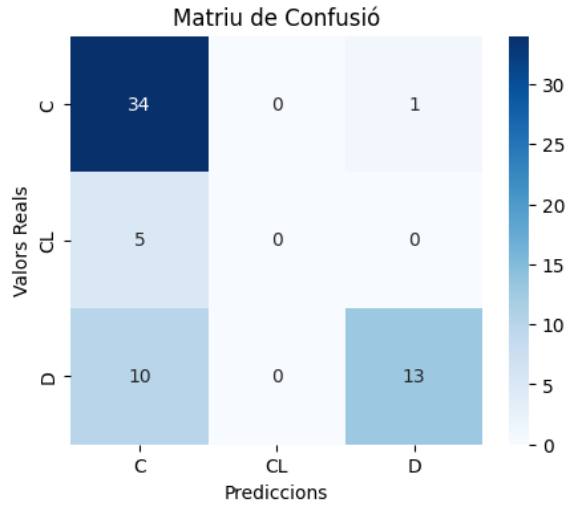
Com es pot observar a la imatge, els millors hiperparàmetres són:

- metric: euclidiana
- neighbors: 7

El tipus de pesos amb 7 veïns no afecta al rendiment del model, per tant, és indiferent quin tipus de weight utilitzar en l'avaluació del model en aquest cas.
Aquests hiperparàmetres donen un F1-Score de 0.7062, una mica més baix del que donava al train.



La matriu de confusió en el conjunt de validació és la següent:



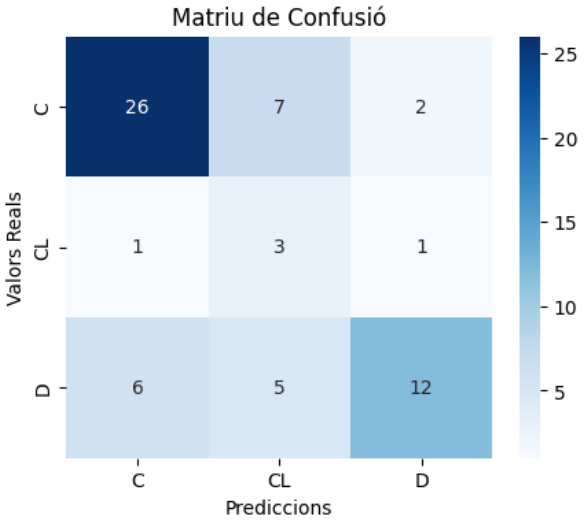
Es pot veure que El model ha predit correctament la classe C per a 34 instàncies, però hi ha 1 ha estat incorrectament classificada com a classe D. A més, hi ha 10 instàncies que pertanyen a la classe D però que han estat incorrectament classificades com a classe C.

Respecte la classe CL, no hi ha cap predicció correcta. Totes les instàncies de la classe CL han estat incorrectament classificades com a classe C.

Respecte la classe D, el model ha sapigut predir correctament 13 instàncies, tot i que hi ha 5 instàncies que pertanyen a la classe C que han estat incorrectament classificades com a classe D.

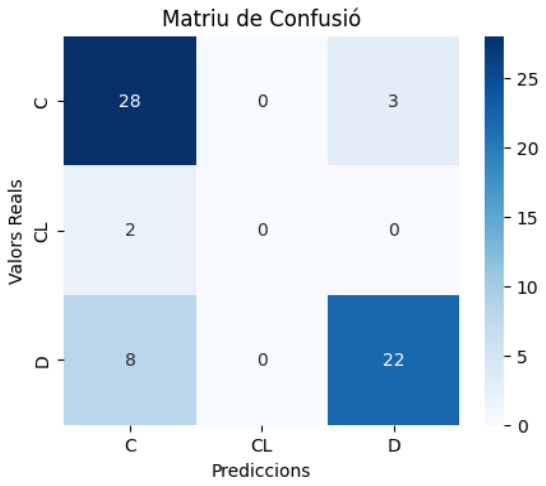
Com era d'esperar, en el conjunt de validació el model no sap classificar cap instància de la classe CL. Encara que la classe C la prediu pràcticament a la perfecció, hi ha algunes instàncies de la classe D que no les classifica correctament.

Si fem una comparació entre aquesta matriu de confusió amb la matriu de confusió del conjunt de validació realitzant oversampling, es pot observar que en el segon model, sap col·locar correctament alguna de les instàncies de la classe CL, però en canvi, disminueix molt la capacitat de detectar les altres dues classes.

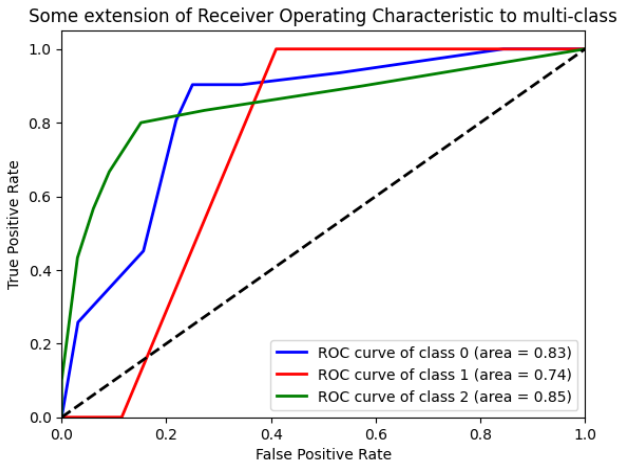


En el test únicament avaluarem el primer model ja que és el que millor resultat ens ha donat.
Els resultats del rendiment del conjunt de test han sigut els següents:

- **Accuracy en entrenament:** 0.7936
- **F1-Score en entrenament:** 0.7803
- **Recall en entrenament**: 0.7936
- **Matriu de confusió en entrenament:**



- **Corba ROC:**



L'anàlisi del rendiment del model KNN en el conjunt de prova revela una sèrie d'aspectes clau en la seva capacitat de classificació. La matriu de confusió indica que la classe C s'ha identificat amb una exactitud considerable, amb 28 instàncies correctament classificades i només 3 equivocacions on s'ha confós amb la classe D. No obstant això, es pot observar una dificultat significativa en la detecció de la classe CL, ja que totes les instàncies d'aquesta classe han estat erròniament assignades a la classe C. Això pot suggerir una manca de característiques distintives o una representació insuficient dins del conjunt d'entrenament que impedeix al model aprendre a diferenciar aquesta classe específica. Per altra banda, la classe D mostra una forta capacitat de classificació amb 22 instàncies correctament identificades, tot i que encara presenta confusions amb la classe C en 8 ocasions.

La corba ROC proporciona una visió addicional, amb una Àrea Sota la Corba (AUC) de 0.83 per a la classe C i de 0.85 per a la classe D, mostrant una habilitat relativament alta per part del model per a la distinció d'aquestes classes. Tanmateix, la classe CL es queda enrere amb una AUC de només 0.74, reafirmant els desafiaments observats en la matriu de confusió.

Les mètriques de rendiment globals són prometedores, amb una precisió (accuracy) del 79.36%, un F1 Score de 0.7803 i un recall igualment de 0.7936. Malgrat això, la congruència d'aquestes mètriques amb la informació proporcionada per la matriu de confusió i la corba ROC implica que, tot i que el model funciona bé en general, pot beneficiar-se de millores específiques per a les classes menys representades o més difícils de classificar.

4.3. Decision Tree

4.3.1. Motivació del model triat

El segon model a estudiar tracta de l'arbre de decisió. Els arbres de decisió són models altament interpretatius que ofereixen una representació visual clara de les decisions preses, cosa que facilita l'explicació dels resultats als stakeholders no tècnics. A diferència de models més complexos, els arbres de decisió divideixen l'espai de característiques en un conjunt de decisions binàries, fent-los més fàcils d'entendre i d'analitzar.

Pel que fa als hiperparàmetres, els arbres de decisió presenten diverses opcions ajustables que poden influir significativament en el rendiment del model. Els principals hiperparàmetres inclouen:

- **Profunditat de l'arbre (max_depth):** Aquest paràmetre controla la màxima profunditat de l'arbre i és crucial per prevenir l'overfitting. Arbres més profunds poden capturar més detalls de les dades però també poden aprendre el soroll present en l'entrenament.

- **Nombre mínim de mostres per dividir (min_samples_split):** Aquest valor determina quantes mostres són necessàries per considerar una divisió en un node. Valors més alts prevenen que el model sigui massa específic amb les seves decisions.

- **Nombre mínim de mostres en un full (min_samples_leaf):** Ajustar aquest paràmetre ajuda a suavitzar el model, imposant un límit en la mida de les fulles de l'arbre.

- **Criteri de divisió (criterion):** Com s'avalua la qualitat d'una divisió pot ser basada en la impuresa Gini o l'entropia, entre d'altres.

4.3.2. Hiperparàmetres

Quan parlem dels arbres de decisió, la selecció dels hiperparàmetres correctes és vital per construir un model que no només aprengui bé de les dades d'entrenament, sinó que també generalitzi bé a dades noves. Aquests són els hiperparàmetres que s'han considerat per al model d'arbre de decisió i els valors que s'han explorat per a cada un:

Hiperparàmetre	Descripció	Valors Provats
criterion	Criteri per a mesurar la qualitat d'una divisió	'gini', 'entropy'
splitter	Estratègia utilitzada per dividir els nodes	'best', 'random'
max_depth	Màxima profunditat de l'arbre	None, 10, 20, 30, 40, 50
min_samples_split	Nombre mínim de mostres per dividir un node	2, 5, 10

min_samples_leaf	Nombre mínim de mostres que ha de tenir un full	1, 2, 4
max_features	Nombre de característiques a considerar	None, 'log2', 'sqrt'
class_weight	Pesos associats a les classes	None, 'balanced'
random_state	Semilla per a la reproductibilitat	42

- **criterion:** Aquest paràmetre determina es mesurarà la qualitat d'una divisió de l'arbre. 'Gini' pot ser eficient per a dades desbalancejades, mentre que 'entropy' podria capturar millor la informació mútua entre les característiques i les classes.

- **splitter:** 'Best' busca la millor divisió possible, mentre que 'random' afegeix un element d'aleatorietat que pot ser útil per a models que sobreajusten.

- **max_depth:** Un arbre profund pot capturar més detalls, però també pot aprendre soroll i sobreajustar. Un arbre sense una profunditat màxima (None) pot créixer fins que totes les fulles siguin pures o continguin menys de min_samples_split mostres.

- **min_samples_split:** Aquest hiperparàmetre controla el nombre mínim de mostres necessàries per considerar una divisió, on valors més alts previndran l'aprenentatge de soroll en les dades.

- **min_samples_leaf:** Similar a min_samples_split, però per a les fulles. Ajuda a suavitzar el model i proporciona una capa addicional contra l'overfitting.

- **max_features:** Limitar el nombre de característiques pot millorar els temps d'entrenament i pot també ajudar a reduir l'overfitting, forçant l'arbre a considerar diferents subconjunts de les dades.

- **class_weight:** Ajustar els pesos de les classes pot ser crític en dades desbalancejades, ajudant a assegurar que les classes minoritàries no siguin ignorades.

Aquests hiperparàmetres estan en una variable anomenada *parametres_dt* la qual és un diccionari que emmagatzema totes les possibles opcions d'hiperparàmetres.

La manera per triar els hiperparàmetres, serà la mateixa que en el model anterior.

4.3.3. Entrenament amb train

Cal destacar que a part de provar els diferents hiperparàmetres, per veure de quines maneres el rendiment del model és millor, també s'ha provat de donar la opció de triar si es vol normalitzar les dades, aplicar oversampling i realitzar one hot encoding o ordinal encoder per codificar les variables categòriques.

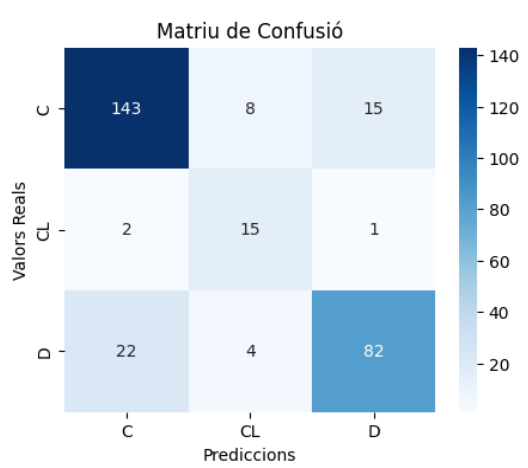
A la funció entrenament ara li passarem com a paràmetres els mateixos d'abans però ara amb la variable *parametres_dt* en comptes de *parametres_knn*. I el model sera DecisionTreeClassifier().

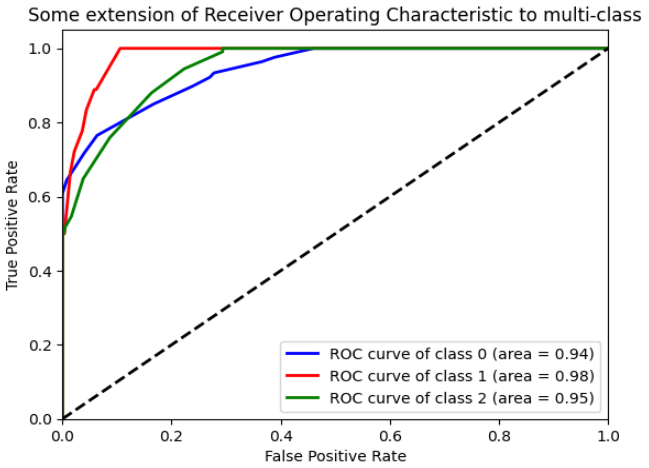
Els **millors paràmetres triats** en el model DecisionTree han sigut els següents: {'class_weight': None, 'criterion': 'entropy', 'max_depth': 10, 'max_features': None, 'min_samples_leaf': 4, 'min_samples_split': 2, 'random_state': 42, 'splitter': 'random'}

Un cop s'han triat aquests paràmetres, s'ha entrenat el model amb model.fit. També s'ha decidit triar que les dades **no seran normalitzades**, si s'aplicarà oversampling i es farà codificació mitjançant one hot encoding, ja que després d'anar provant les combinacions, s'ha arribat a la conclusió que aquesta és la que millor rendiment dona.

Seguidament s'han mirat les mètriques per avaluar el rendiment del conjunt d'entrenament:

- **Accuracy en entrenament:** 0.8219
- **F1-Score en entrenament:** 0.8238
- **Recall en entrenament**: 0.8219
- **Matriu de confusió en entrenament:**
- **Corba ROC en entrenament:**





L'entrenament del model d'arbre de decisió ha revelat un rendiment força satisfactori, reflectit tant en les mètriques de classificació com en les visualitzacions gràfiques de la matriu de confusió i la corba ROC.

Amb els paràmetres triats anteriorment, l'arbre de decisió ha assolit una precisió en l'entrenament de 0.8219, un F1-Score de 0.8238 i un recall també de 0.8219, indicant una alta consistència en la capacitat del model per fer prediccions exactes i equilibrades. La matriu de confusió ens mostra una classificació bastant bona, amb una majoria d'instàncies correctament classificades per a les classes majoritàries, però també dona com a resultat algunes àrees de confusió, particularment entre les classes C i D, així com un rendiment menys òptim en la identificació de la classe CL.

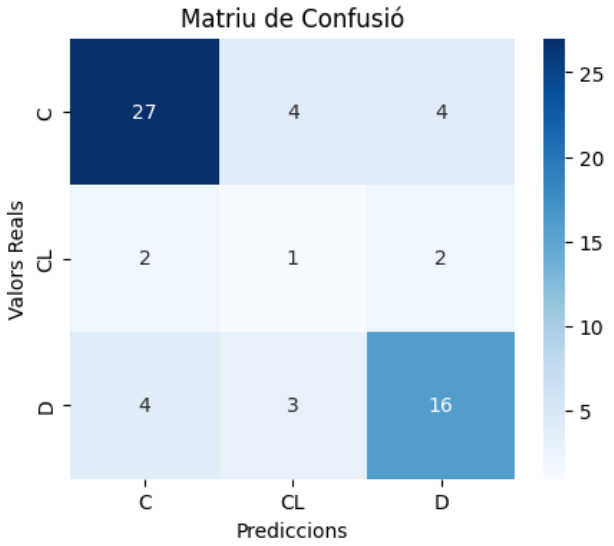
La corba ROC ofereix una visió encoratjadora amb un AUC de 0.94 per a la classe C, 0.98 per a la classe CL, i 0.95 per a la classe D, demostrant una excel·lent capacitat discriminativa del model. Aquestes àrees sota la corba són indicatives de la fortalesa del model a l'hora d'identificar correctament les classes positives i negatives a través de diferents llindars de decisió.

4.3.4. Anàlisi de resultats al validation i test

La matriu de confusió obtinguda del conjunt de validació del model d'arbre de decisió reflecteix un rendiment diferencial a l'hora de classificar les tres classes en estudi: C, CL i D. S'observa que el model té una tendència a predir millor la classe C, amb 27 instàncies correctament identificades. No obstant això, hi ha una certa confusió amb les altres classes, ja que 4 instàncies de la classe C són incorrectament classificades com CL i 4 com D.

La classe CL, que representa una categoria més desafiant per al model, mostra una tendència a ser sobreclassificada com a C, amb 2 casos, i com a D, també amb 2 casos, amb només 1 cas correctament classificat com CL. Això indica que les característiques utilitzades pel model podrien no ser prou distintives per aquesta classe o que no disposa de prou exemples per aprendre adequadament.

Pel que fa a la classe D, es mostra una capacitat relativament millor de classificació amb 16 instàncies correctament etiquetades. No obstant això, el model encara confon aquesta classe amb la classe C en 4 ocasions i amb la classe CL en 3 ocasions, suggerint que encara hi ha espai per millorar la distinció entre aquestes categories.

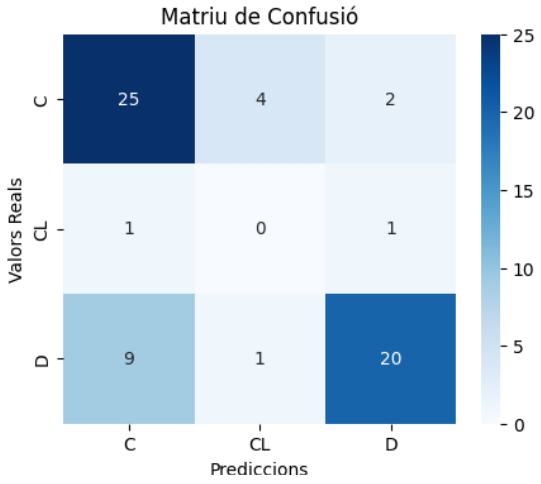


A més de la matriu de confusió, el F1-Score obtingut en el conjunt de validació és de 0.7129.Aquesta puntuació indica un rendiment acceptable encara que no perfecte.

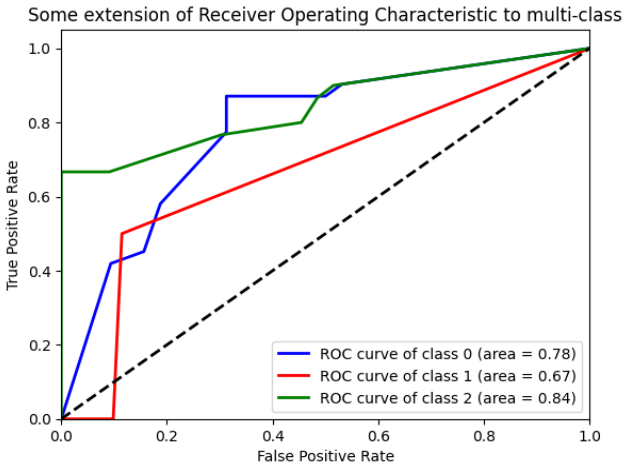
Finalment, en el conjunt de prova, s'han obtingut els següents resultats de rendiment:

- **Accuracy:** 0.7142
- **F1-Score:** 0.7321
- **Recall**: 0.7142

- **Matriu de confusió:**



- **Corba ROC:**



En el conjunt de prova, l'anàlisi del rendiment del model d'arbre de decisió mostra una precisió (accuracy) del 71.42%, un F1-Score del 73.21% i un recall també del 71.42%. Aquests resultats són indicatius d'un model amb una capacitat moderada de classificació correcta de les instàncies.

La matriu de confusió revela que el model ha tingut un èxit relatiu en la identificació de la classe C, amb 25 instàncies classificades correctament, però també mostra que s'han produït confusions, amb 4 instàncies classificades com a classe CL i 2 com a classe D. Això pot indicar que mentre el model té una tendència a reconèixer correctament la classe C, pot beneficiar-se d'una millora en la seva capacitat per distingir entre les classes similars.

La classe CL continua sent un punt dèbil significatiu per al model, ja que no ha classificat correctament cap instància d'aquesta classe, amb 1 instància incorrectament atribuïda a la classe C i una altra a la classe D. Això ressalta la necessitat d'una atenció específica per millorar la classificació d'aquesta classe, que podria incloure l'exploració de noves característiques o tècniques de reequilibri de classe.

Per la classe D, el model mostra una millora amb 20 instàncies correctament classificades, però encara amb un nombre considerable d'errors, ja que 9 instàncies han estat incorrectament classificades com a classe C i 1 com a classe CL.

La corba ROC proporciona una visió més granular de la capacitat del model per distingir entre cada classe a diferents llindars. El model mostra una Àrea Sota la Corba (AUC) de 0.78 per a la classe C, 0.67 per a la classe CL i 0.84 per a la classe D. Aquests valors AUC reafirmen la tendència ja observada en la matriu de confusió: una capacitat decent de discriminació per a les classes C i D, però una actuació insuficient per a la classe CL.

En conclusió, mentre que l'arbre de decisió ha demostrat ser relativament competent en la identificació de les classes C i D, la seva actuació en la classe CL és clarament insuficient. Potser caldria revisar els hiperparàmetres per evitar l'overfitting ja que com s'ha observat, les mètriques del conjunt d'entrenament són bastant millors que les de prova.

4.4. SVM

4.4.1. Motivació del model triat

El tercer model que s'ha decidit explorar en aquest estudi és la Màquina de Vectors de Suport (SVM). L'SVM té la capacitat de crear fronteres de decisió complexes i altament adaptatives entre les classes. A diferència del KNN, que es basa en la proximitat local i la votació dels veïns més propers, l'SVM intenta trobar el pla o hiperplà que millor separa les classes en l'espai de característiques, optimitzant així la marge entre les categories de dades.

Un dels principals avantatges de l'SVM és la seva eficàcia en espais de dimensions altes i quan les classes no són linealment separables. A través de l'ús de funcions kernel, SVM és capaç de projectar les dades a un espai de característiques més gran, on es pot trobar una separació lineal.

4.4.2. Hiperparàmetres

L'SVM és un model potent que pot ser ajustat a través d'un conjunt diversos hiperparàmetres, els quals poden influir en el rendiment del model. A continuació, es detallen els hiperparàmetres que s'han seleccionat per explorar en el model SVM.

Hiperparàmetre	Descripció	Valors Provats
C	Paràmetre de regularització.	1, 10
kernel	Tipus de kernel utilitzat en la transformació.	'rbf', 'linear'
gamma	Coeficient per a kernels no lineals com 'rbf'.	'scale', 'auto'
shrinking	Utilitzar la heurística de shrinking per l'optimització.	True
probability	Habilitar l'estimació de probabilitats.	True
tol	Tolerància per al criteri de parada de l'entrenament.	1e-3, 1e-5
class_weight	Ponderacions de les classes per a dades desbalancejades.	None, 'balanced'

- **C:** Aquest hiperparàmetre controla la força de la regularització, és a dir, com de molt vol evitar el model l'ajust excessiu (overfitting) als dades d'entrenament. Valors més baixos impliquen més regularització, mentre que valors més alts permeten al model ajustar-se més lliurement als dades.

- **kernel:** Aquesta elecció determina la manera en què les dades són projectades en un nou espai de característiques. El 'rbf' és útil per trobar fronteres no lineals, mentre que 'linear' és preferible per dades que ja són linealment separables.

- **gamma:** Només rellevant per al kernel 'rbf', aquest paràmetre afecta la influència d'una única instància d'entrenament. Un valor 'scale' adapta 'gamma' a la variància de les característiques, mentre que 'auto' l'estableix basant-se en la mida de les característiques.

- **shrinking:** Aquesta és una tècnica que pot millorar el temps d'entrenament en alguns problemes. Mantenim això activat per defecte per a la nostra exploració.

- **probability:** Encara que calcular les probabilitats pot ser costós en termes de càlcul, pot ser útil quan es necessiten decisions més informades més enllà de la simple classificació.

- **tol:** Aquest hiperparàmetre permet al model determinar quan aturar l'entrenament si no es guanya en millora, amb valors més baixos que resulten en un entrenament més llarg i potencialment més ajustat.

- **class_weight:** Aquesta opció és crítica quan es treballa amb dades que tenen un desequilibri de classes. 'Balanced' ajustarà els pesos inversament proporcional a les freqüències de classe en les dades d'entrenament.

Aquests hiperparàmetres estan en una variable anomenada *parametres_dt* la qual és un diccionari que emmagatzema totes les possibles opcions d'hiperparàmetres.

La manera per triar els hiperparàmetres, serà la mateixa que en els models anteriors.

4.4.3. Entrenament amb train

Cal destacar que a part de provar els diferents hiperparàmetres, per veure de quines maneres el rendiment del model és millor, també s'ha provat de donar la opció de triar si es vol aplicar oversampling i realitzar one hot encoding o ordinal encoder per codificar les variables categòriques.

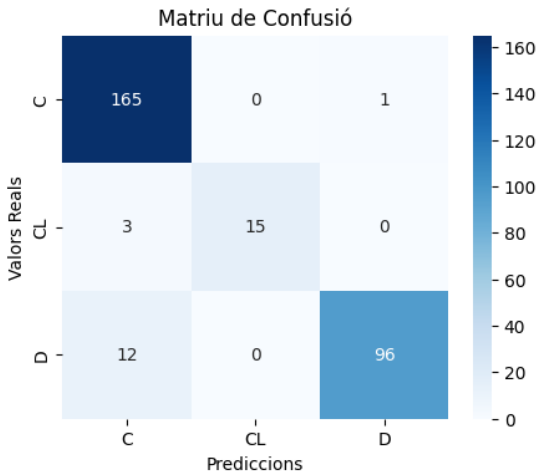
A la funció entrenament ara li passarem com a paràmetres els mateixos d'abans però ara amb la variable *parametres_svm*. I el model serà SVM().
Els **millors paràmetres triats** en el model SVM han sigut els següents: {'C': 10, 'class_weight': None, 'gamma': 'auto', 'kernel': 'rbf', 'probability': True, 'shrinking': True, 'tol': 0.001}

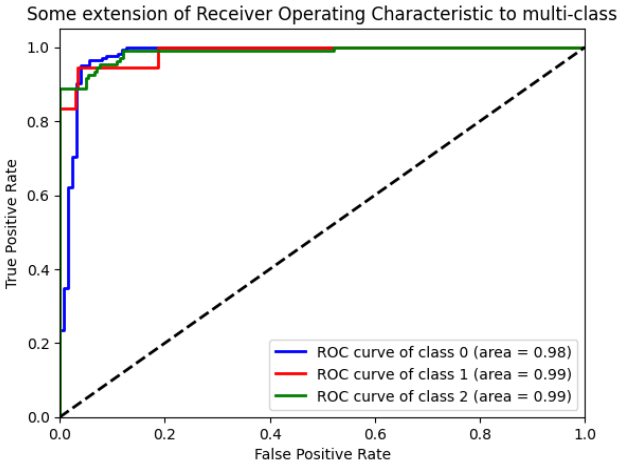
Un cop s'han triat aquests paràmetres, s'ha entrenat el model amb model.fit. També s'ha decidit triar que les dades **sí seran normalitzades** ja que el model SVM ho requereix, no s'aplicarà oversampling

i es farà codificació mitjançant one hot encoding, ja que després d'anar provant les combinacions, s'ha arribat a la conclusió que aquesta és la que millor rendiment dona.

Seguidament s'han mirat les mètriques per avaluar el rendiment del conjunt d'entrenament:

- **Accuracy en entrenament:** 0.9452
- **F1-Score en entrenament:** 0.9446
- **Recall en entrenament**: 0.9452
- **Matriu de confusió en entrenament:**
- **Corba ROC en entrenament:**





Els resultats obtinguts en el conjunt d'entrenament amb el model SVM reflecteixen un alt grau de precisió, com es demostra amb una accuracy de 0.9452, un F1-Score de 0.9446 i un recall igualment alt de 0.9452. Aquests valors indiquen que el model és capaç de classificar les dades d'entrenament amb un grau considerable d'exactitud.

Analitzant la matriu de confusió, podem veure que la majoria de les instàncies han estat classificades correctament. La classe C té una alta taxa d'encert amb 165 prediccions correctes i només una incorrecta. La classe CL, tot i ser la més difícil de classificar, com s'ha observat en altres models, mostra un rendiment millorat amb 15 instàncies correctament identificades. No obstant això, encara hi ha confusió, amb 3 instàncies classificades com a classe C i 12 com a classe D, el que indica que encara hi ha espai per a la millora. La classe D té 96 prediccions correctes amb només 12 casos confosos amb la classe C, mostrant que el model té una tendència a ser més precís en aquesta categoria.

La corba ROC per a l'entrenament mostra uns valors d'Àrea Sota la Corba (AUC) excepcionals, amb 0.98 per a la classe C, 0.99 per a la classe CL i 0.99 per a la classe D. Aquests valors altíssims suggereixen que el model és capaç de distingir molt bé entre les classes a diferents llindars de decisió.

Tot i que els resultats són imprescindiblement positius, és important destacar que un rendiment elevat en el conjunt d'entrenament no sempre es tradueix en una bona generalització. La proximitat entre les métriques de F1-Score i recall, juntament amb la precisió, podrien indicar que el model està propens a

l'overfitting, especialment si aquest rendiment no es manté quan s'aplica al conjunt de prova. Per això, a continuació s'observaran els resultats obtinguts en els conjunts de prova i validació i d'aquesta manera es comprovarà si s'ha fet overfitting.

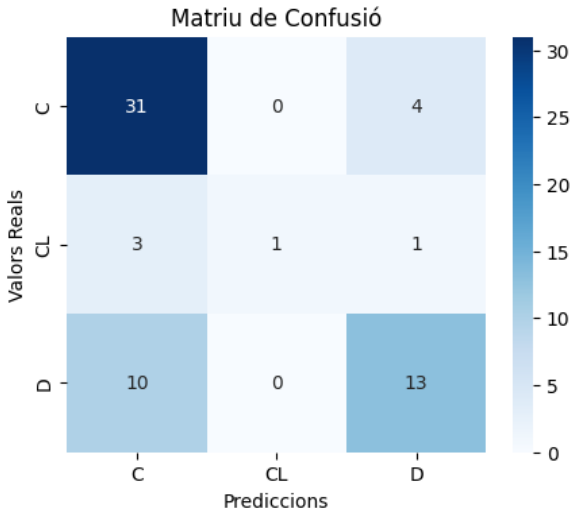
4.4.4. Anàlisi de resultats al validation i test

En el conjunt de validació, el model SVM ha demostrat un rendiment lleugerament inferior al que havia mostrat en el conjunt d'entrenament. La matriu de confusió mostra que, per a la classe C, 31 instàncies han estat correctament identificades, mentre que 4 han estat classificades com a D. Això indica una bona capacitat del model per reconèixer la classe C, però amb una certa inclinació cap a confondre-la amb la classe D.

Per a la classe CL, només 1 instància ha estat correctament classificada, amb 1 instància erròniament atribuïda a la classe C i una altra a la classe D. Això suggereix que la classe CL continua sent un repte per al model, possiblement a causa de la manca de característiques distintives o la presència d'un nombre insuficient d'instàncies d'entrenament per aquesta classe específica.

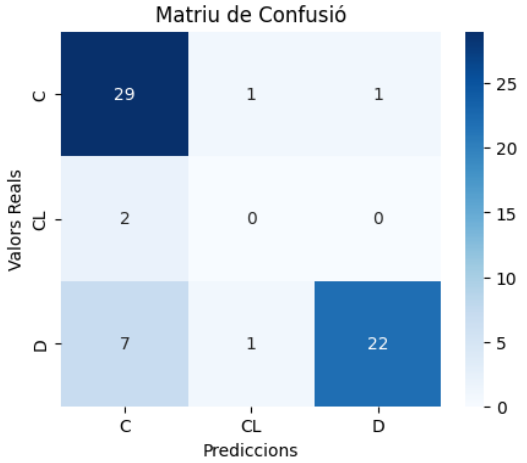
Respecte la classe D, s'ha identificat correctament 13 instàncies, però 10 han estat equivocadament classificades com a classe C. Aquesta confusió podria indicar que les fronteres de decisió entre la classe C i la classe D no són tan clares com s'hauria desitjat.

L'F1 score obtingut en el conjunt de validació és de 0.6939, una caiguda respecte a l'F1 score obtingut en el conjunt d'entrenament. Aquesta baixada en el rendiment podria ser un indicatiu d'overfitting, ja que el model no ha estat capaç de mantenir el nivell d'exactitud quan s'enfronta a noves dades.
Combinant aquests resultats amb la pròxima observació del rendiment en el conjunt de prova, es podrà determinar si aquesta tendència cap a l'overfitting és consistent i si cal fer ajustaments en el model.

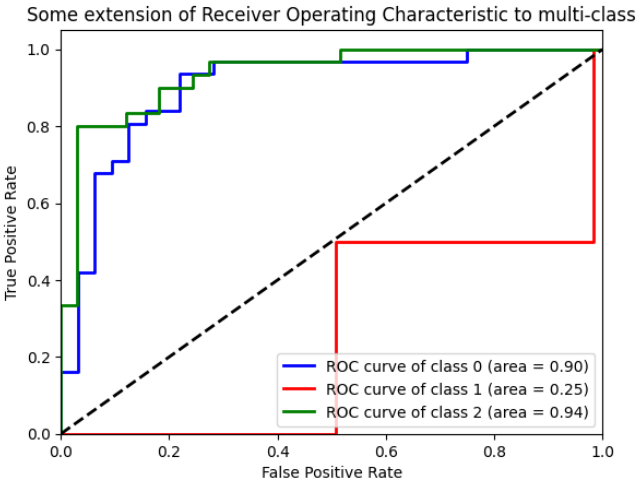


Finalment, en el conjunt de prova, s'han obtingut els següents resultats de rendiment:

- **Accuracy:** 0.8095
- **F1-Score:** 0.8089
- **Recall**: 0.8095
- **Matriu de confusió:**



- **Corba ROC:**



Els resultats del conjunt de prova per al model SVM mostren un rendiment robust amb una precisió (accuracy) de 0.8095, un F1-Score de 0.8089 i un recall igualment de 0.8095. Aquestes mètriques indiquen que el model té una bona capacitat general per fer prediccions correctes en dades no vistes durant l'entrenament.

Observant la matriu de confusió, veiem que la classe C té 29 instàncies correctament classificades amb només dos errors menors, indicant una forta capacitat de discriminació per aquesta classe. La classe CL, malgrat la seva petita grandària, no ha estat classificada correctament en cap ocasió, la qual cosa posa de manifest la dificultat del model per a identificar aquesta classe. Per a la classe D, el model ha classificat 22 instàncies correctament, però ha confós 7 instàncies com a classe C, mostrant una precisió respectable però amb marges de millora en la distinció entre aquestes dues classes.

La corba ROC destaca amb una Àrea Sota la Corba (AUC) de 0.90 per a la classe C i 0.94 per a la classe D, la qual cosa demostra que el model té una excel·lent capacitat per distingir aquestes dues classes en diferents llindars de decisió. No obstant això, la classe CL mostra una AUC de només 0.25, reafirmant que la capacitat del model per distingir aquesta classe és notablement baixa.

Les conclusions clau d'aquest anàlisi són les següents:

El model SVM ha mostrat una bona capacitat per classificar les classes més representades, C i D, com es reflecteix en les altes AUCs, tot i així com la resta de models, no és capaç de classificar correctament la classe amb menys instàncies tot i aplicant l'oversempling. Comparant aquests resultats amb el rendiment en el conjunt d'entrenament, on la precisió va ser de 0.9452, el F1-Score de 0.9446 i el recall de 0.9452, es pot observar una disminució en el rendiment però no tant marcada com per suggerir un greu problema d'overfitting. Això indica que el model generalitza raonablement bé a noves dades.

5. Selecció de model

Després d'un anàlisi de diferents models i estratègies de classificació, s'ha seleccionat la Màquina de Vectors de Suport (SVM) com el model final a implementar. La decisió s'ha basat en la seva superioritat en termes de rendiment dins del conjunt de prova, superant altres models (KNN i l'arbre de decisió) en les mètriques clau de precisió, F1-Score i recall.

5.1. Descripció del model triat

SVM és un model poderós que destaca en la capacitat de crear fronteres de decisió complexes, utilitzant el truco del kernel per transformar les dades a un espai de dimensions més altes on la separació lineal és possible. Aquesta característica el fa ideal per a conjunts de dades on la relació entre les classes no és immediatament aparent o és altament no lineal.

5.2. Anàlisi de les limitacions i capacitats del model

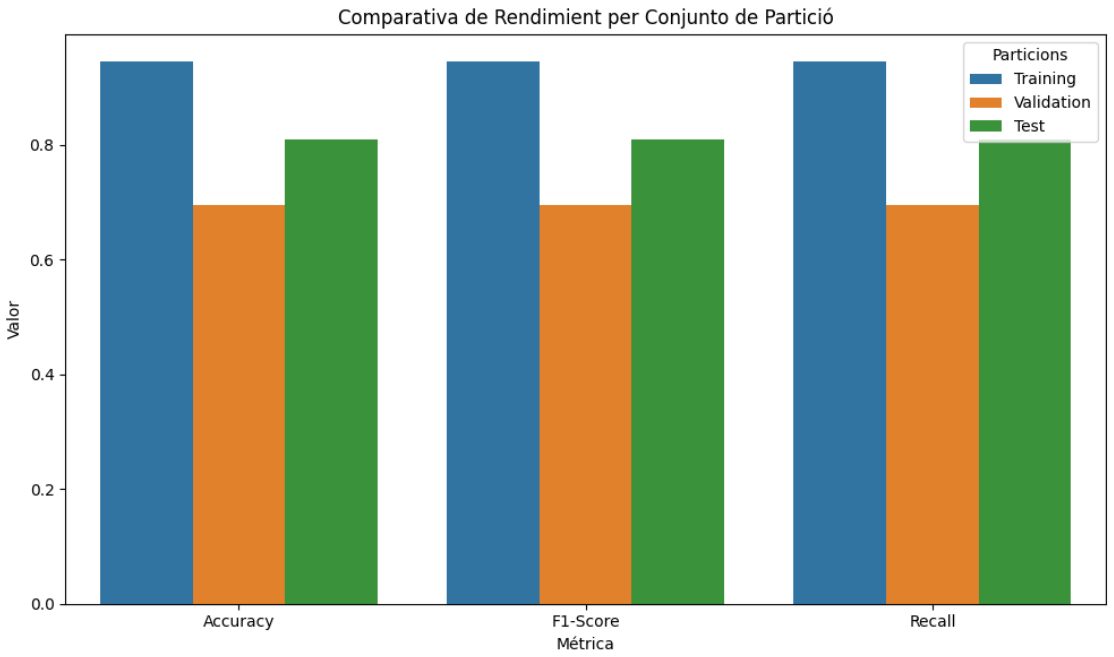
Tot i que SVM ha demostrat un rendiment sòlid en el conjunt de prova, és important reconèixer les seves limitacions. Per exemple, el model pot ser sensible a la selecció de hiperparàmetres, especialment en la elecció del kernel i els seus paràmetres associats. A més, mentre SVM maneja bé les característiques d'alta dimensionalitat, pot ser menys eficient en termes computacionals quan s'enfronta a conjunts de dades de gran escala.

D'altra banda, les capacitats de SVM són notables. És capaç de modelar la complexitat inherent a les dades de manera eficient, proporcionant una bona generalització i mantenint alhora una alta precisió. Això es deu en gran part a la seva capacitat d'optimitzar el marge entre les classes, reduint així el risc d'overfitting.

5.3. Resultats en la partició Test en comparació amb Train i Val

Com ja s'ha vist anteriorment, els resultats obtinguts pels tres conjunts de dades són:

Mètrica	Entrenament	Validació	Test
Accuracy	0.9452	0.7142	0.8095
F1-Score	0.9446	0.6939	0.8089
Recall	0.9452	0.7142	0.8095



Podem observar que el model exhibeix la major precisió en el conjunt d'entrenament, el que era d'esperar ja que els models tendeixen a tenir millor rendiment en les dades sobre les quals s'han entrenat. Encara que als resultats de validació sembla haver-hi un gran overfitting degut a la caiguda dels resultats, la recuperació del rendiment en el conjunt de prova és notable i molt positiva, ja que s'apropa als nivells d'entrenament, suggerint que el model té una bona capacitat de generalització quan s'enfronta a dades noves.

Tot i així, amb les matrius de confusió obtingudes als apartats anteriors, es pot comprovar que els models no classifiquen correctament la classe CL, que és la classe amb moltes menys instàncies de la variable objectiu. Tot i aplicant oversampling, els models segueixen sense classificar correctament aquesta classe classificant-la majoritàriament en la classe C.

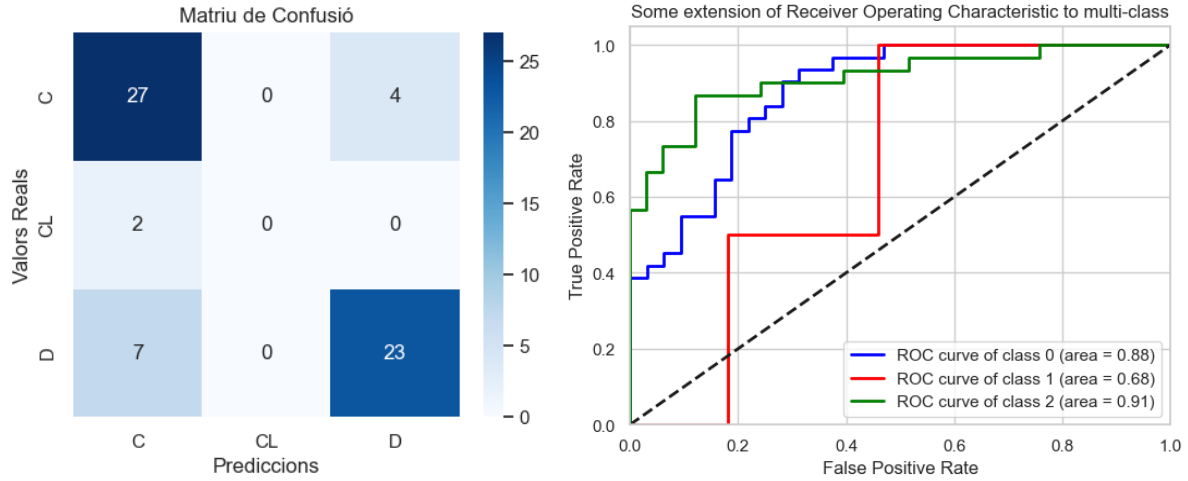
6. Bonus 1

El model Explainable Boosting Machine (EBM) és un tipus de model de machine learning interpretable. Combina les fortaleses dels models lineals amb la flexibilitat dels models no lineals com els arbres de decisió. Funciona entrenant una sèrie de models més simples, com ara arbres de decisió petits, sobre diferents trams de les dades, i després combinant-los per a millorar la precisió i mantenir la interpretabilitat. Aquesta tècnica permet entendre com cada característica influeix en la predicció del model, facilitant així la transparència i la confiança en les seves prediccions.

RESULTATS DEL MODEL:

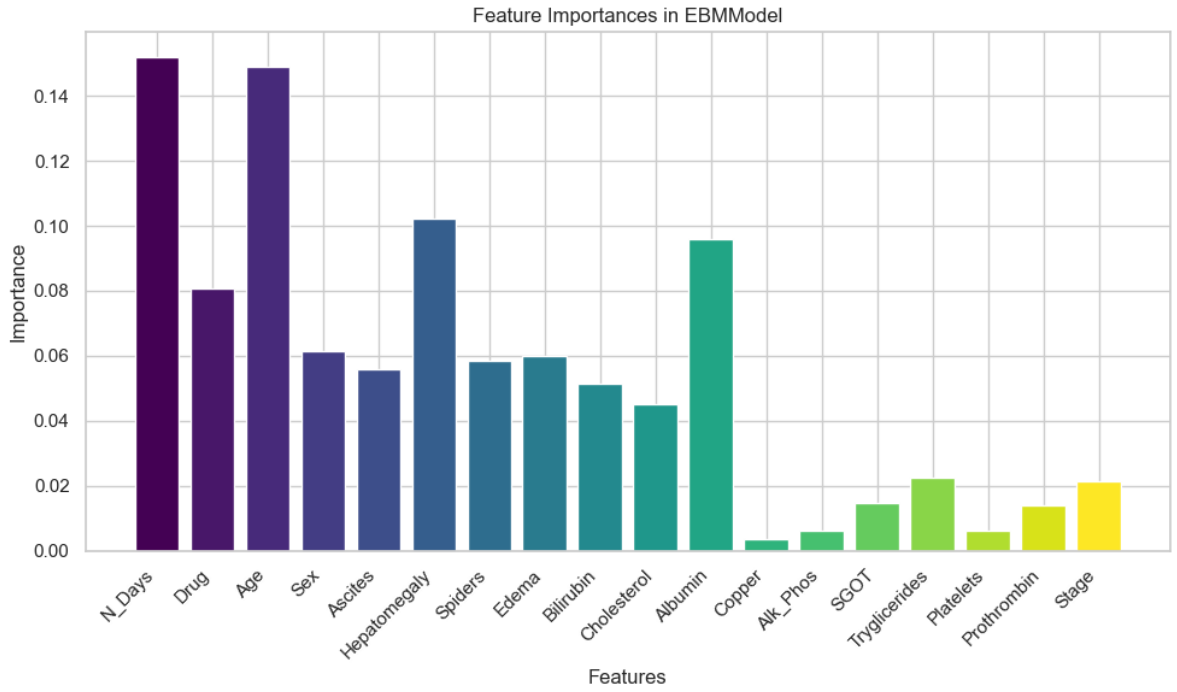
Mètrica	Entrenament	Test
Accuracy	0.8253	0.7936
F1-Score	0.7966	0.7808
Recall	0.8253	0.7936

Confusion Matrix i Corba ROC del conjunt de dades test:



Analitzant el model Explainable Boosting Machine (EBM) a través de la matriu de confusió i la corba ROC, juntament amb les mètriques de rendiment, podem concloure que el model presenta una bona generalització, amb una alta precisió reflectida en l'accuracy tant en el conjunt d'entrenament com en el de test. No obstant això, la classificació de la classe minoritària CL és problemàtica, com s'indica per la seva baixa representació en la matriu de confusió i una AUC inferior en la corba ROC. Això suggereix que el model podria beneficiar-se de més dades o de tècniques especialitzades per a classes desequilibrades.

VARIABLES MÉS IMPORTANTS:



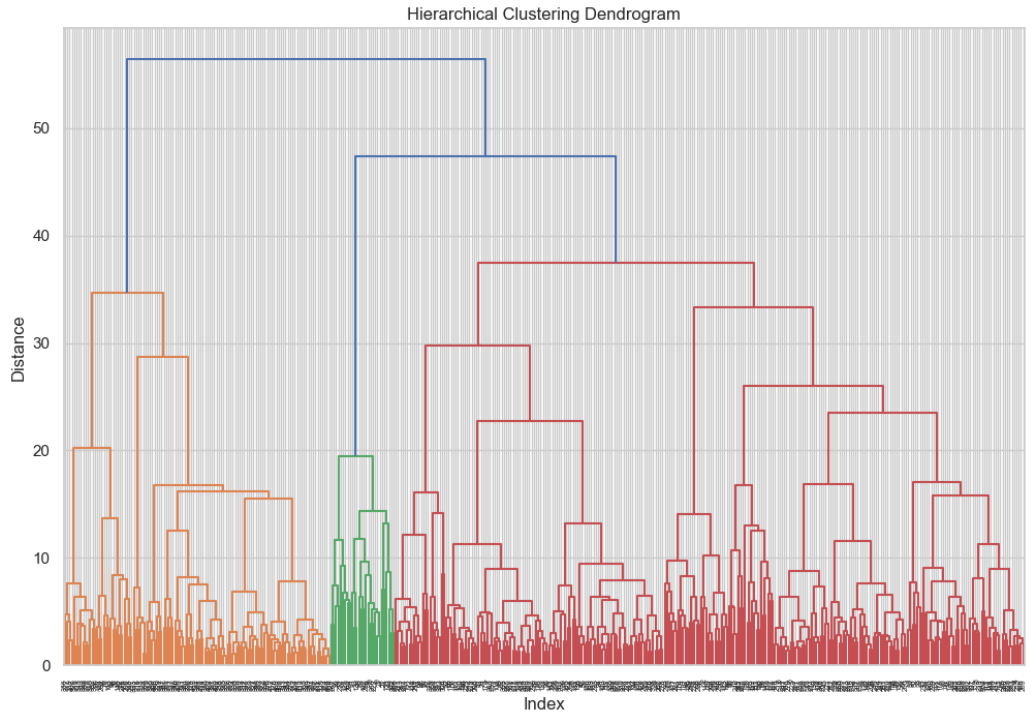
Com es pot observar en aquesta imatge, les variables amb les barres més altes indiquen que tenen una major importància dins del nostre dataset per predir la variable objectiu.

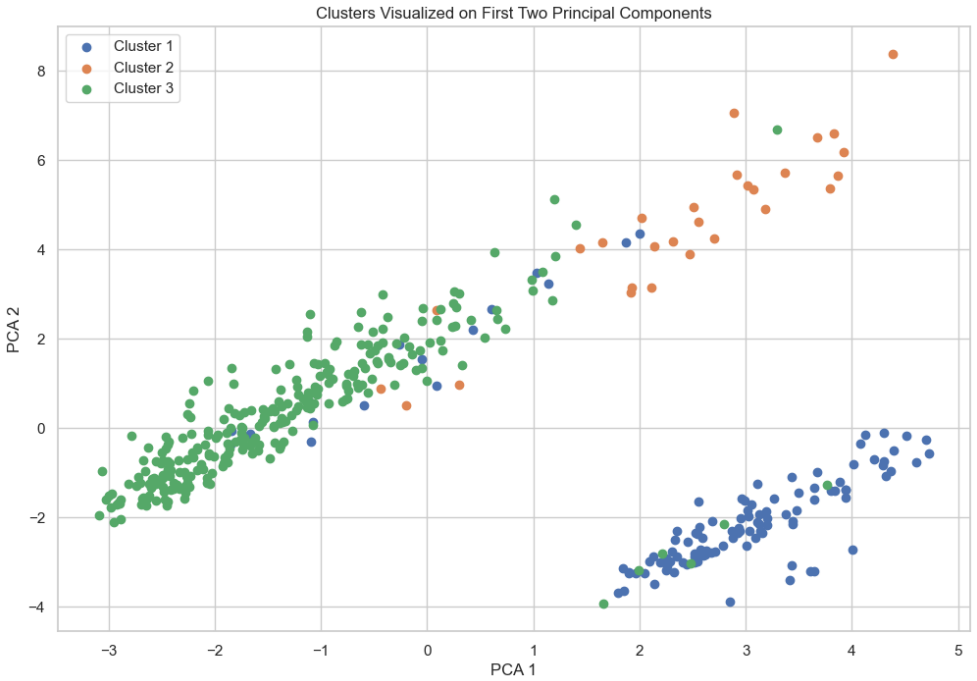
El gràfic mostra la importància de diferents característiques segons el model EBM. Les variables *'N_Years'* i *'Drug'* semblen ser les més influents en el model, indicant una gran importància en la predicció dels resultats. Això podria reflectir l'impacte del temps i del tractament farmacològic en la supervivència dels pacients amb cirrosi.

Altres variables com *'Age'*, *'Ascites'* i *'Bilirubin'* també tenen una importància considerable, suggerint que factors com l'edat del pacient, la presència de líquid en la cavitat abdominal i els nivells de bilirrubina són rellevants per a la progressió de la malaltia. En canvi, 'Stage', 'Prothrombin' i 'Platelets' tenen la menor importància segons aquest model, el que pot indicar que aquestes variables tenen menys pes en les decisions del model respecte a la supervivència.

7. Bonus 2

Per tal de realitzar el clustering sense la variable *Status* per poder veure si podíem agrupar les dades en uns grups que ens ajudessin a entendre millor la base dades es va crear un dendrograma de clustering jeràrquic. En aquest podem observar que el nombre de clústers ideals estaria entre 3 i 4 però degut a que el nostre objectiu principal era veure si es formarien clústers depenent de si el pacient a sobreviscut, a mort o li han fet un trasplantament de fetge, escollirem tres per a veure si realment es formen així els grups.





Com es pot observar, trobem la distinció de 3 clusters en les dues components principals. Això indica que les característiques de cada clúster son diferents. Els clústers 1 i 2 mostren una certa superposició, encara que es poden distingir entre ells. Això suggereix que les característiques que defineixen aquests dos grups són similars però no idèntics.

Les dimensions dels clústers són molt variades ja que trobem 116 observacions en el primer grup, 28 en el segon i finalment 274 en l'últim.

Si ens fiquem a analitzar el segon clúster està format per les persones que porten menys temps en l'experiment. Pel que fa a la bilirrubina els pacients que en tenen més es troben en el segon clúster també.

En aquest estudi no podem treure les conclusions necessàries degut a que faria falta acabar d'estudiar del tot els diferents clústers mitjançant millors tècniques de profiling, però ens pot donar una idea de que els valors de la bilirrubina alta probablement serà el grup de la gent que ha mort.

8. Model Card

Model Card for Cirrhosis Patient Survival Prediction Dataset

Model Details

Overview

This SVM model is designed to predict outcomes for cirrhosis patients, determining whether they will live, die, or require a liver transplant.

Type: Support Vector Machine (SVM)

Training Algorithms: SVM with kernel rbf

Hyperparameters: C = 10, gamma = auto, tol = 0.001

Version

- **Date:** 2023 - 12 - 28
- **Developer:** Abril Risso Matas, abril.maria.risso@estudiantat.upc.edu

References

- https://www.mayoclinic.org/es/diseases-conditions/high-blood-cholesterol/in-depth/triglycerides/art-20048186
- https://www.ncbi.nlm.nih.gov/pmc/articles/PMC7268936/
- https://www.hopkinsguides.com/hopkins/view/Johns_Hopkins_Diabetes_Guide/547086/all/Liver_function
- https://lipidworld.biomedcentral.com/articles/10.1186/s12944-023-01979-w

Considerations

Use Cases

- Clinical prognosis for cirrhosis patients
- Support in medical decision-making for treatments and transplants

Factors

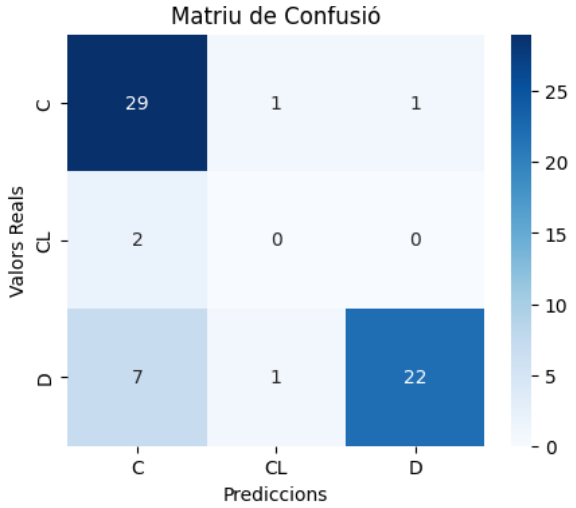
- Patient demographic data, clinical biomarkers, biomedical history.
- Model accuracy, sensitivity, specificity, and F1 score.

Metrics

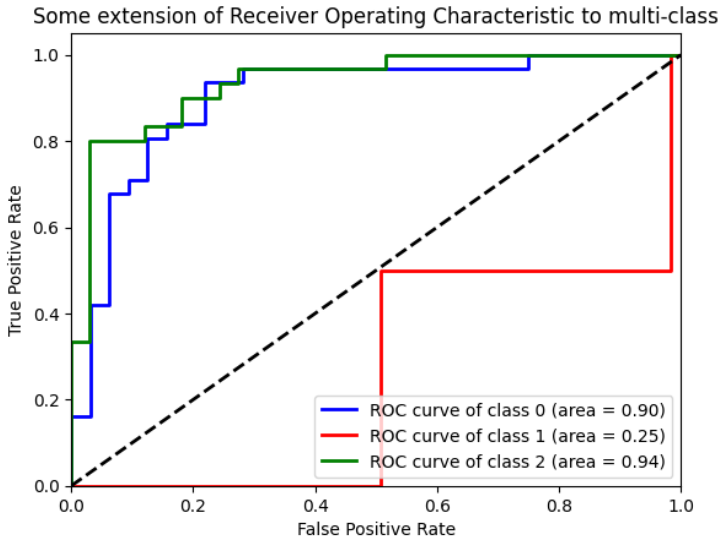
- Model Performance Measures: See the Performance section below.
- Decision Thresholds: Not applicable, as the SVM model uses margins.

Performance

- Confusion Matrix



- ROC Curve



9. Conclusions

Amb aquesta pràctica he pogut consolidar i expandir les competències en el preprocessament i anàlisi de dades complexos. La manipulació acurada de les dades, des de la neteja fins a la normalització, ha sigut fonamental per assegurar que els models de machine learning poguessin interpretar correctament els patrons de les dades.

El tractament de les dades desequilibrades, especialment amb la classe CL de la variable objectiu, ha requerit tècniques específiques de balanceig que han posat a prova la comprensió de com l'escassetat de dades pot afectar la generalització del model.

L'avaluació dels models KNN, arbre de decisió i SVM ha sigut exercici rigorós que ha implicat no només la comparació de mètriques bàsiques com l'accuracy, sinó també una anàlisi més profunda mitjançant l'F1-score i el Recall, que proporcionen una visió més matizada del rendiment del model en les diferents classes. Aquest procés va ser crucial per entendre les limitacions de cada model. La interpretació de les matrius de confusió i les corbes ROC ha permès obtenir més informació sobre el rendiment i estudi del model.

Després d'un examen exhaustiu, el model SVM ha sigut el model final triat. La seva elecció com a model final es basa en la seva superioritat en el rendiment global, la seva robustesa davant dades variades, i la seva eficiència computacional. Tot i així com els altres models, el SVM també ha demostrat complicacions a l'hora de classificar la classe amb menys instàncies.

Aquest projecte ha sigut una demostració valiosa del poder del machine learning aplicat a problemes reals de salut que ha permès extendre el meu coneixement sobre diferents models de machine learning i anàlisi de dades.