UNIVERSITAT POLITÈCNICA
DE CATALUNYA
BARCELONATECH

Escola Politècnica Superior d'Enginyeria de Manresa

Predicció de Matrícula Basada en Aprenentatge Automàtic

Manresa, 2 de juliol de 2023

treball de fi de grau que presenta

Anass Anhari Talib

en compliment dels requisits per assolir el

Grau d'Enginyeria en Sistemes TIC

Direcció: Sebastià Vila Marta i Aleix Llusà Serra

Resum

Fins ara, el procediment per a les matriculacions universitàries ha sigut de forma manual. A més, sempre ha requerit un gran esforç i una gran experiència per part de l'equip qui les gestiona.

Ens trobem davant d'un problema complex, és a dir, si volguéssim automatitzar aquest procediment, no podríem aplicar un enfocament tradicional, ni cap mena de regla genèrica o algorisme per determinar si un estudiant es matricularà o no, ja que el comportament d'un estudiant és en certa manera imprevisible.

Per aquest motiu, es proposa aplicar *Machine Learning* (aprenentatge automàtic) amb l'objectiu de generar i analitzar diversos models predictius basant-se en l'històric previ de tots els estudiants.

Abstract

So far, the procedure for university enrollments it has been done manually. In addition, it has always required a great effort and experience on the part of the team that manages them.

We are facing a complex problem, that is, if we wanted to automate this procedure, we could not apply a traditional approach nor any generic rule or algorithm to determine whether or not a student will enroll because of the behavior of each student is quite unpredictable.

For this reason, it is proposed to apply Machine Learning with the purpose of generating and analyzing various predictive models based on the previous history of all students.

Índex

1 Introducció

1.1 Context

A la Universitat Politècnica de Catalunya (UPC) una tasca administrativa important és la planificació del següent curs acadèmic. En aquesta planificació hi juga un paper destacat la previsió d'estudiants que es matricularan a cada assignatura que s'oferirà. Aquesta previsió determina el nombre de grups que caldrà preveure per a cada assignatura en funció dels estudiants que previsiblement s'hi matricularan. El nombre de grups, a la vegada, determina la necessitat de professorat per a cada assignatura i la necessitat d'espais. La previsió de matrícula, doncs, impacta en dimensions fonamentals com la plantilla i els espais que, a la vegada, tenen un impacte pressupostari important.

La previsió de matrícula és difícil. Els estudiants es matriculen d'acord amb les seus interessos i circumstàncies acadèmiques. Aspectes com la seva preferència horària, el professorat concret que imparteix tal o qual assignatura, els resultats acadèmics, el marc normatiu i, fins i tot, el seu estat d'ànim sembla que poden incidir en la decisió de quines assignatures es matriculen. La manera en com tots aquests factors incideixen en la matrícula és incerta. A més, amb tota seguretat, els pes dels diferents factors en la decisió de matrícula és canviant amb el temps com ho són els mateixos estudiants i el seu comportament general.

Actualment, aquest procés es fa en base a l'experiència personal dels gestors. Això fa que la qualitat de les prediccions sigui problemàtica i que el procediment, que recordem és crític, estigui en mans de poques persones. A més, aquest procés demana un esforç important de l'equip gestor davant de la dificultat per automatitzar-lo.

Dels problemes que presenta el procediment actual en destaquen els següents:

1) És un procediment sensible a la disponibilitat de la persona o persones que tenen el coneixement. Transferir l'experiència i el «bon saber» a tercers es fa difícil atesa la naturalesa intuïtiva del coneixement.

2) El procediment s'ha d'executar en uns terminis estrets de temps.

3) Són freqüents les prediccions poc acurades. Quan les prediccions subestimen la matrícula, els estudiants s'enfronten amb la impossibilitat de cursar certa assignatura per que la capacitat està saturada. Quan se sobreestima la matrícula, es malbaraten recursos cars i el cost d'explotació augmenta.

L'escenari que s'ha presentat condueix a pensar que seria interessant automatitzar la previsió de matrícula. Això, però, presenta reptes tècnics importants sent el més destacable el fet que es basa en un coneixement difús i gens estructurat. Els algoritmes d'aprenentatge automàtic (*machine learning*), que s'enquadren en la intel·ligència artificial, estan ben condicionats per a problemes d'aquest tipus.

Un algoritme de aprenentatge automàtic calcula un model capaç de classificar o predir. El model es construeix automàticament a partir de dades històriques en un procés que es coneix

com «aprenentatge». El treball amb aquests algoritmes requereix d'un treball de depuració i transformació de les dades històriques per tal que en pugui extreure els patrons que s'incorporaran al model. Addicionalment, els models acostumen a ser parametritzables. L'obtenció dels paràmetres òptims ajuda a que el comportament del model sigui el millor possible. L'obtenció d'un model precís basat en aprenentatge automàtic és una tasca de naturalesa empírica que requereix experimentar i analitzar diverses configuracions de les dades i dels paràmetres.

En aquest treball s'ha optat per aplicar models basats en arbres de decisió, un del algoritmes habituals en aprenentatge automàtic. Els arbres de decisió presenten una avantatge substancial quan és necessari poder explicar les raons de les decisions que pren el model.

Les dades històriques s'han obtingut de l'expedient acadèmic de l'estudiantat adequadament anonimitzades. Això deixa fora factors que possiblement són rellevants en la decisió de matricular-se, com per exemple els horaris en que s'imparteix una determinada assignatura. Així doncs, la naturalesa de les dades històriques ja limita la qualitat del model que s'obtindrà.

1.2 Objectius

L'objectiu d'aquest treball és aplicar l'ús de tècniques de aprenentatge automàtic per automatizar la predicció de la matrícula universitària i determinar-ne la viabilitat. Predir la matrícula universitària consisteix en determinar el nombre d'estudiants que es matricularan en una assignatura concreta durant el següent període de matrícula. Per assolir aquest objectiu es plantegen els següents objectius específics:

- Entendre en detall els algoritmes d'aprenentatge automàtic que s'aplicaran en aquest treball.
- Obtindre la màxima precisió possible en la predicció de matriculacions per a cada assignatura.
- Construir una sistema d'entrenament i processat de dades escalable, eficient, i fàcil de mantenir.
- Avaluar i comparar diferents models i tècniques d'aprenentatge automàtic per identificar el més adequat per aquesta aplicació.

1.3 Estructura del treball

Aquesta memòria comença amb un capítol dedicat als antecedents en què s'abordarà el context en el qual es situa el treball, es presentaran coneixements fonamentals de *Machine Learning* i s'introduiran les notacions clau que es faran servir al llarg del treball.

A continuació, es descriurà detalladament l'estructura de les dades amb les quals es treballarà. S'analitzaran aspectes rellevants com el format inicial de les dades, les seves característiques i peculiaritats. A més, s'abordaran els possibles problemes o reptes que poden sorgir en l'anàlisi i el processament d'aquestes dades, ja que aquest serà el focus principal d'aquest treball.

Posteriorment, es presentaran els models de classificació que s'aplicaran en aquest treball. Donada la gran varietat de models de *Machine Learning* disponibles, s'explicarà detalladament l'algorisme de cada model seleccionat.

Seguidament, es mostraran els resultats obtinguts a partir de l'aplicació dels models de classificació. Es realitzarà una avaluació dels models obtinguts, la qualitat de les prediccions,

les similituds i peculiaritats de cada model. Es valoraran aspectes com per exemple l'impacte de la profunditat d'un arbre de decisió i l'impacte de la tranformació de les dades.

Finalment, en les conclusions, es realitzarà un anàlisi dels resultats obtinguts per seleccionar el model de matriculació més viable i òptim.

2 Antecedents

2.1 Context

En aquest treball, és necessari posar en context la problemàtica de les matriculacions universitàries i la importància de l'obtenció d'una previsió amb la màxima precisió possible dels grups de cada assignatura. En aquest cas, les dades disponibles per a aquest treball són els expedients dels estudiants de TIC, anonimitzats per garantir la confidencialitat. A la Taula 2.1, es mostra una estructura de dades d'exemple que representa les dades dels expedients, incloent informació com l'identificador (id) de l'estudiant, el codi i l'acrònim de l'assignatura, el curs acadèmic, el quadrimestre i la nota numèrica obtinguda.

Aquest conjunt de dades proporciona un històric acadèmic dels estudiants en les diferents assignatures del pla d'estudis. Llavors, amb aquesta informació s'aplicaran tècniques d'aprenentatge automàtic per generar i analitzar models predictius. L'objectiu és utilitzar aquests models per predir amb la màxima precisió possible el nombre de grups necessaris per a cada assignatura. Això permetrà a l'equip de secretaria prendre decisions necessàries, com la gestió de les aules disponibles i la contractació de professorat, entre altres.

id	estudianth	assignatura	acronim	curs	quad	nota_num
0	193b01116d	330212	MBE	2010	1	7.4
1	193b01116d	330213	F	2010	1	6.2
2	193b01116d	330214	I	2010	1	10.0
3	193b01116d	330215	ISD	2010	1	9.0
	...					
11122	283732537e	330212	MBE	2020	1	9.2
11123	283732537e	330213	F	2020	1	7.9
11124	283732537e	330214	I	2020	1	9
11125	283732537e	330215	ISD	2020	1	7

Taula 2.1: Estructura dels expedients

Finalment, en l'àmbit universitari, hi ha conceptes que poden tenir significats diferents, però que a primera vista poden semblar evidents. Per aquest motiu, s'estableix la següent nomenclatura per evitar confusions per a la comprensió del treball:

- **Notació**
  - *Expedient* (e_i). Informació acadèmica d'un estudiant fins a l'any i.
  - *Any relatiu* (a_r). Any corresponent dins de la vida de l'expedient, és a dir, la primera matrícula correspon l'any 1.
  - *Quadrimestre relatiu* (Q_r). Definim $Q_r \in 1, 2$ on Q_r divideix en un espai de quatre mesos un curs o un any relatiu a_r.

  - *Quadrimestre parcial* (Q_p). Definim $1 \leq Q_p \leq 8$, on Q_p representa els quadrimestres parcials amb las distribució corresponent de les assignatures.

- **Operacions**
  - $mat(x, e_i)$. Donada una assignatura x i un expedient e_i, es retorna un booleà (0/1) si en l'expedient e_i l'estudiant es va matricular a l'assignatura x.
  - $nota(x, e_i, n_\nexists)$. Retorna la nota de l'assignatura x en l'expedient e_i, en cas de no haver-la matriculada, es retorna la nota $n_\nexists$.

2.2 *Machine Learning*

«*Machine Learning* (aprenentatge automàtic), és la ciència de la programació que presenta i defineix una col·lecció d'algorismes perquè un computador pugui aprendre de les dades.» [Gér19]

Prèviament, davant un problema, l'humà obtenia la solució definint certes regles (com podria ser un programa). Però, per a problemes de gran complexitat, determinar aquestes regles seria complicat i la solució no seria escalable. Per aquest motiu, algorismes de ML (*Machine Learning*) s'encarreguen de determinar aquestes regles basant-se en les dades.

La fase d'entrenament, és a dir, la d'aprenentatge, es basa en un *dataset* (conjunt de dades) amb n mostres. Convencionalment, el *dataset* presenta la forma de la Taula 2.2 i es defineixen els següents conceptes:

- *Label.* L'atribut o el conjunt d'atributs que representa cada mostra.
- *Feature.* La solució o el conjunt de solucions per a cada mostra.

Sample	*Feature 1*	...	*Feature n*	*Label 1*	...	*Label n*
Sample 1		...			...	
⋮						
Sample n		...			...	

Taula 2.2: Estructura habitual d'un *dataset*

2.2.1 Aprenentatge supervisat

Existeixen diversos tipus de sistemes de ML. En aquest cas, l'aprenentatge supervisat, com el nom indica, fa servir un conjunt de dades que inclou les solucions desitjades. En particular, la forma de les dades s'ajustaria a la Taula 2.2. A més, els problemes d'aprenentatge supervisat es poden diferenciar segons:

- *Problemes de classificació.* En aquests casos, les mostres es poden classificar en dues o més classes. L'algorisme del model aprèn de les dades i dels atributs de cada mostra per determinar la solució. Quan es presenten noves mostres o mostres que el model no «ha vist» prèviament, el model entrenat és capaç de classificar-les amb precisió.

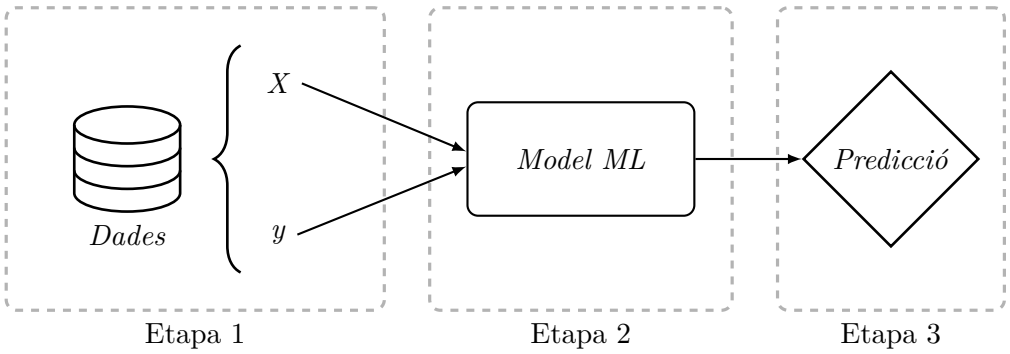

Figura 2.1: Generació i avaluació del model. Font: Pròpia.

- *Problemes de regressió.* En aquest tipus de problemes, l'objectiu és predir sortides contínues, és a dir, un valor o un conjunt de valors numèrics. El model utilitza les dades i els atributs associats per estimar o predir els valors de sortida desitjats.

2.2.2 Overfitting

L'overfitting és un problema comú en la construcció de models de machine learning. Es produeix quan el model s'ajusta massa als detalls i al soroll de les dades d'entrenament, perdent així la seva capacitat de generalització i predir amb precisió dades noves o de test. [Wik23]

Per entendre millor l'overfitting, podem prendre l'exemple d'un problema de classificació, considerem un model d'arbre de decisió per a la classificació de flors. Partim d'un conjunt de dades que conté les mesures de longitud i amplada dels pètals de diverses flors, juntament amb les seves etiquetes de classe (*Setosa*, *Versicolor* i *Virginica*).

Si entrenem un arbre de decisió amb una profunditat massa petita, el model pot ser massa simple i no aprendre els detalls en les dades d'entrenament. Aquest és un cas d'underfitting, ja que el model no ajusta prou bé les dades i no pot classificar amb precisió les flors.

Però, si incrementem la complexitat de l'arbre, com augmentar la seva profunditat, el model pot ajustar-se massa bé al les mostres d'entrenament, classificant-los correctament. A primera vista, sembla que aquest és un model perfecte. Ara bé, quan utilitzem el model amb noves flors (mostres prèviament mai vistes) per a la classificació, potser no serà tan precís i cometi errors importants. Això és el que coneixem com overfitting, com a exemple, a la Figura 2.2 el classificador «verd» pateix d'overfitting, en contrast, el classificador «negre» seria una millor opció, ja que generalitza millor i és més capaç de classificar correctament les noves mostres.

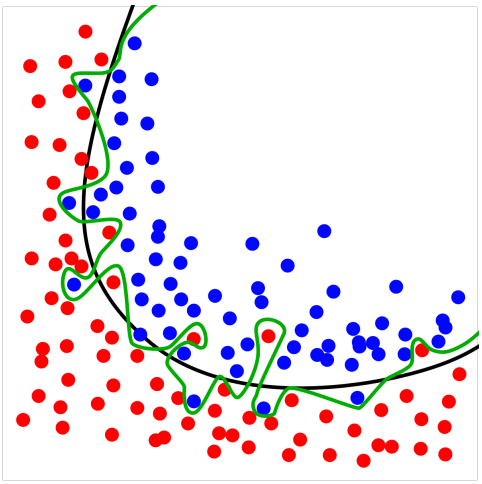

Figura 2.2: Exemple d'overfitting. Font: Wikipedia

3 Estructura de les dades

En l'àmbit del Machine Learning, les dades són un factor crucial. Són la base sobre la qual es construeixen i s'entrenen els models predictius. No obstant, en la majoria de casos, les dades no sempre són consistents. Sovint, les dades pateixen d'anomalies, errors o inconsistències que afecten la qualitat dels resultats. Anomalies com per exemple, com dades perdudes, valors anormals, inconsistències en els formats, entre altres.

Llavors, en l'anàlisi dels expedients acadèmics, és important detectar i tractar les anomalies i problemes que es poden presentar en el conjunt de dades. Per aquest motiu, és important depurar i realitzar una profunda neteja sobre les dades amb l'objectiu d'assegurar la consistència d'aquestes i obtenir resultats fiables.

Un dels problemes comuns en les dades dels expedients acadèmics és la falta de consistència en la nomenclatura utilitzada per identificar les assignatures (acrònims), notes buides o altres elements. Això pot dificultar l'anàlisi i la comprensió dels expedients. Llavors, és necessari establir una nomenclatura estàndard per evitar confusions i garantir la consistència en les dades. Algunes possibles anomalies a tenir en compte són:

- *Acrònims repetits o faltants.* Els acrònims repetits provoquen confusió a l'hora d'identificar assignatures. Analitzar les dades i els resultats utilitzant acrònims és millor per a la comprensió dels resultats en comparació amb els codis d'assignatura, ja que els codis d'assignatura no tenen una representació intuitiva de les assignatures.

- *Convalidacions.* Les convalidacions entre assignatures o cursos requereixen d'un tractament especial per assegurar la correcta representació d'una convalidació.

- *Estudiants que deixen la carrera.* És important identificar i gestionar els expedients dels estudiants que deixen la carrera per evitar anomalies en l'anàlisi de les dades.

- *Notes buides.* La presència de notes buides en les dades pot afectar el procés d'anàlisi. Doncs, és important gestionar adequadament l'absència de qualificacions i distingir entre diferents casos. Una nota buida pot ser interpretada com a valor nul (0), indicar que l'assignatura no ha estat presentada, que ha sigut convalidada, o que l'estudiant ha abandonat la carrera. És essencial establir una metodologia clara per a tractar aquest tipus de situacions.

3.1 Descripció de les dades

Les dades dels expedients acadèmics representen l'historial i l'evolució acadèmica de cada estudiant. Aquestes dades inclouen diversos registres que descriuen detalladament tota mena d'informació relacionada en l'àmbit universitari i el recorregut de cada estudiant al llarg del temps.

Dins de les dades dels expedients acadèmics, podem distingir dues categories principals:

- *Les dades de «l'històric acadèmic»*. Aquestes dades es centren principalment en la matriculació de l'estudiant en les diferents assignatures i les notes obtingudes. Amb aquesta informació es pot obtenir una visió generalitzada del comportament acadèmic de l'estudiant al llarg del temps.
- *Les dades de la «motxilla de l'estudiant»*. Principalment inclouen informació addicional sobre l'estudiant. Aquestes dades poden incloure com l'assignació de beques, la nota d'accés a la universitat, la via d'accés, entre altres.

Cada universitat disposa d'un sistema de gestió de dades, i la forma en què s'organitzen , s'emmagatzemen, i manipulen les dades pot variar significativament. La gestió de les dades dels expedients acadèmics no és una tasca senzilla i el sistema de gestió pot ser complex. En particular, en aquest treball, les dades dels expedients acadèmics són proporcionades en un fitxer CSV (valors separats per comes). Aquest fitxer conté els següents registres:

- *codi_expedient*. Codi únic que identifica l'expedient acadèmic de cada estudiant, en aquest cas, anonimitzat per assegurar la confidencialitat.
- *curs*. L'any acadèmic el qual es va realitzar una matriculació.
- *quad*. Quadrimestre en què es va realitzar la matrícula (tardor o primavera).
- *codi_upc_ud*. El codi identificador de cada assignatura.
- *nota_num_def*. Nota númerica obtinguda per l'estudiant en l'assignatura en particular.
- *nota_des_def*. La descripció de la nota obtinguda per l'estudiant en l'assignatura en concret, com per exemple "NP" (no presentada), "C" (assignatura convalidada), "MH" (matrícula d'honor), "N" (notable), entre altres.
- *beca*. Indica si l'estudiant disposa de beca, s'ha de tenir en compte que aquest atribut no és fixe, pot variar segons el rendiment acadèmic de l'estudiant en cada curs, segons notes obtingudes i altres factors.
- *any_naix*. L'any de naixement de l'estudiant.
- *via_acces*. Via d'accés a la universitat de l'estudiant, per exeple, selectivitat, CFGS (grau superior), etc.
- *ordre_assignacio*. Representa l'ordre d'assignació durant la fase de matriculació de l'estudiant.
- *nota_acces*. Nota d'accés a la universitat obtinguda per l'estudiant.

3.2 Transformació de les dades

Com s'ha discutit a la Secció 5.2, en el context de l'aprenentatge supervisat en el l'àmbit del *Machine Learning*, és necessari disposar una estructura de dades ben definida. En particular, les dades acadèmiques obtingudes de la universitat requereixen un procés de neteja i transformació per ajustar-se a l'estructura típica d'un conjunt de dades (*dataset*). Així doncs, en aquesta secció es presenten dues possibles transformacions de les dades.

3.2.1 Primera transformació

És important mantenir una seqüencialitat i linealitat temporal en les matriculacions de cada estudiant, de manera que es pugui obtenir la progressió dels estudiants a mesura que avancen en els seus estudis. Per aconseguir-ho, s'ha decidit presentar i transformar la evolució de cada estudiant com es mostra a la Figura 3.1. Això permetrà tenir una visió clara de com cada estudiant ha anat matriculant-se al llarg del temps.

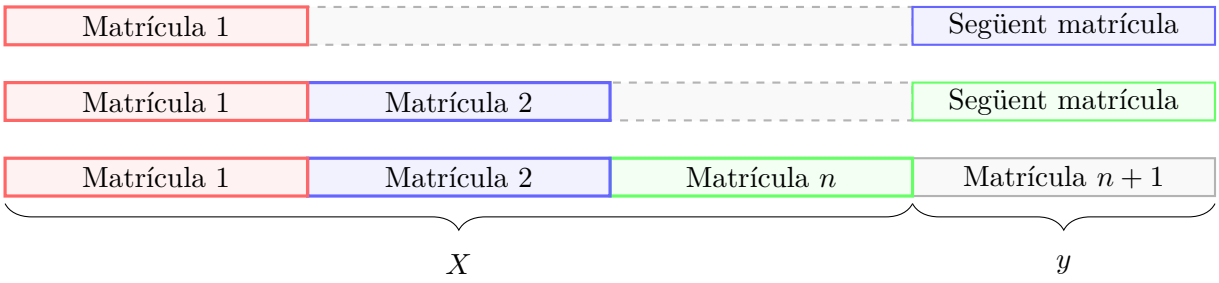

Figura 3.1: Evolució del expedient d'un estudiant (primera representació). Font: Pròpia

La Figura 3.1 representa l'evolució de les matriculacions de cada estudiant. Per aquest motiu, es transformen les dades crues (Taula 2.1) per extreure els *features* i *labels* com s'ha vist a la Secció 2.2. Concretament, construïnt els vectors d'entrada $\underline{X}$ que contenen les característiques (informació) de la progressió de les matriculacions (històric actual de cada estudiant), i els vectors de sortida $\underline{y}$ que definiran d'acord amb l'històric actual de cada estudiant, quines assignatures matricularà a la següent matrícula.

$$e_i = \Big\|_{i=1}^{N_{mat}} \left(\Big\|_{x=1}^{N_{assig}} \Big(mat(x, e_i) \,\Big\|\, nota(x, e_i, 0) \Big) \,\Big\|\, \left(\Big\|_{x=1}^{N_{assig}} mat(x, e_i + 1) \right) \right) \tag{3.1}$$

Aplicant la transformació 3.1 a les dades crues s'obtè el *dataset* de la Taula 3.1. Per distingir entre una assignatura matriculada i una no, s'ha optat per afegir un booleà indicatiu, d'aquesta manera, es podrà diferenciar una assignatura matriculada amb nota 0 i una assignatura no matriculada $n_{\nexists} = 0$. A més, com s'ha mencionat prèviament, l'estructura d'un *dataset* ha de ser uniforme, llavors, l'expedient de cada estudiant serà la concatenació de n matrícules fins un límit que fixarem, per exemple, 15 matrícules.

estudianth	mat_id	curs	quad	MBE:1	M?MBE:1	F:1	M?F:1	I:1	M?I:1	...
193b01116d	1	2010	1	7.4	True	6.2	True	10.0	True	
193b01116d	2	2010	2	7.4	True	6.2	True	10.0	True	...
193b01116d	3	2011	1	7.4	True	6.2	True	10.0	True	...
	⋮									
283732537e	1	2020	2	9.2	True	7.9	True	9	True	...
283732537e	2	2021	1	9.2	True	7.9	True	9	True	...

Taula 3.1: Estructura d'un expedient

3.2.2 Segona transformació

Un cop definit el primer *dataset*, es procedeix a definir-ne un segon amb la finalitat de comparar models entrenats amb *datasets* diferents i extreure'n una conclusió en funció dels resultats de predicció. Doncs, s'estableix:

- est. Representa un estudiant.
- n. Matriculacions totals d'un estudiant on $e_n - 1$ representa el darrer expedient d'aquest.

Llavors al *dataset* ha de constar:

$$\begin{array}{llll} est_1 & \rightarrow & e_1^x & \cdots \quad mat(x, e_2) \mid nota(x, e_2, 0) < 5 \\ est_1 & \rightarrow & e_2^x & \cdots \quad mat(x, e_3) \mid nota(x, e_3, 0) < 5 \\ \vdots & & & \\ est_1 & \rightarrow & e_k^x & \cdots \quad mat(x, e_{k+1}) \mid nota(x, e_{k+1}, 0) \geq 5 \end{array}$$

Observem que k ha de complir:

$$\begin{array}{ll} (1) & k \leq n - 1 \\ (2) & \forall i : 2 \leq i < k + 1 \rightarrow (\ nota(x, e_i, 0) < 5\ ) \ \wedge \ (\ nota(x, e_{k+1}, 0) \geq 5\ ) \end{array}$$

Doncs, es defineix la transformació següent:

$$e_i = \Big\Vert_{i=1}^{n-1} \left(\Big\Vert_{x=1}^{N_{assig}} \left(a'_r \Big\Vert mat(x, e_i) \Big\Vert nota(x, e_i, 0) \right) \Big\Vert \left(\Big\Vert_{x=1}^{N_{assig}} mat(x, e_i + 1) \right) \right) \tag{3.2}$$

on a'_r representa el darrer any relatiu en que es va cursar l'assignatura x, com a resultat, obtenim la Taula 3.2.

Finalment, cal recordar que les transformacions 3.1 i 3.2 són completament diferents, no es poden comparar. En funció dels resultats obtinguts (taxa de predicció, etc.) es valorarà quina transormació aparenta ser millor.

estudianth	darrer_any	MBE	M?MBE	darrer_any	F	M?F	...
est_1	1	4	True	1	True	10.0	
est_1	1	4	False	1	False	10.0	
est_1	2	7.5	True	1	False	10.0	
⋮							

Taula 3.2: Estructura d'un expedient

4 Arbres de decisió

Els arbres de decisió són un tipus d'aprenentatge supervisat utilitzats per classificació i regressió. A diferència d'altres tipus d'aprenentatge, els arbres són simples d'entendre i d'interpretar. L'algorisme d'aprenentatge es basa en recursivament particionar l'espai en subarbres en funció del *feauture* que proporcioni el màxim guany d'informació a cada etapa (particionat).

A més, els arbres de decisió són particularment atractius en el l'àmbit de la Intel·ligència Artificial Explicable (*Explainable AI*). A diferència de mètodes més complexos com les xarxes neuronals, coneguts com a models de «caixa negra» on la interpretació dels resultats és pràcticament nul·la, els arbres de decisió ofereixen una interpretació més clara. Les regles de decisió en els arbres de decisió són fàcilment interpretables, cosa que permet entendre i explicar les prediccions realitzades per l'arbre de decisió. En aquest cas, és ideal i de gran importància poder interpretar certs patrons en el comportament dels estudiants durant la fase de la matrícula.

A la Figura 4.1 observem un arbre de decisió, en aquest cas, usant el dataset `iris` d'ús habitual en l'aprenentatge automàtic, especialment per a tasques de classificació i algorismes d'arbre de decisió. És un conjunt de dades conté mesures de la longitud del sèpal, l'amplada del sèpal, la longitud del pètal i l'amplada del pètal de tres espècies diferents de flors d'iris (*Setosa*, *Versicolor* i *Virginica*). Observem que els arbres de decició poden ser visualitzats i fàcilment interpretats, en concret, a cada node del arbre es parteix d'una condició en funció d'un *feature* i un llindar t_m, doncs, a l'hora de predir si la condició es certa recorrem cap al node de l'equerra, sinó cap al node de la dreta fins convergir al node final obtenint la classe resultant.

4.1 Algorisme CART

4.1.1 Definició

L'algorisme *CART* (*Classification and Regression Trees*) és un mètode de *Machine Learning* utilitzat per a la construcció de models predictius basats en arbres de decisió. L'objectiu principal de l'algorisme és dividir l'espai dimensional format per les dades, en cocnret, la dimensió equival a la quantitat de *features* o característiques que es disposa en les dades. Per exemple, es pot considerar un conjunt de dades que consisteix en n assignatures, juntament amb les notes obtingudes. En aquest cas, l'espai dimensional estaria format per la suma de les n assignatures i les corresponents notes, de manera que la dimensió de l'espai seria $D = 2n$.

Doncs, l'algorisme *CART* intenta dividir i particionar l'espai dimensional de les dades en subespais més petits amb l'objectiu de representar les diferents classes o grups de la manera més acurada possible, i obtenir una predicció o classificació més precisa. D'aquesta manera, es crea una estructura jeràrquica en forma d'arbre, on cada node representa una condició o decisió basada en una característica (*feature*) i un valor de tall (*threashold*).

A la Figura 4.2 es presenta un simple exemple amb dos *features* contínues (x_1 i x_2) i 5 possibles classes. Intuïtivament, sense aplicar cap mena d'algorisme, es definiria idealment el

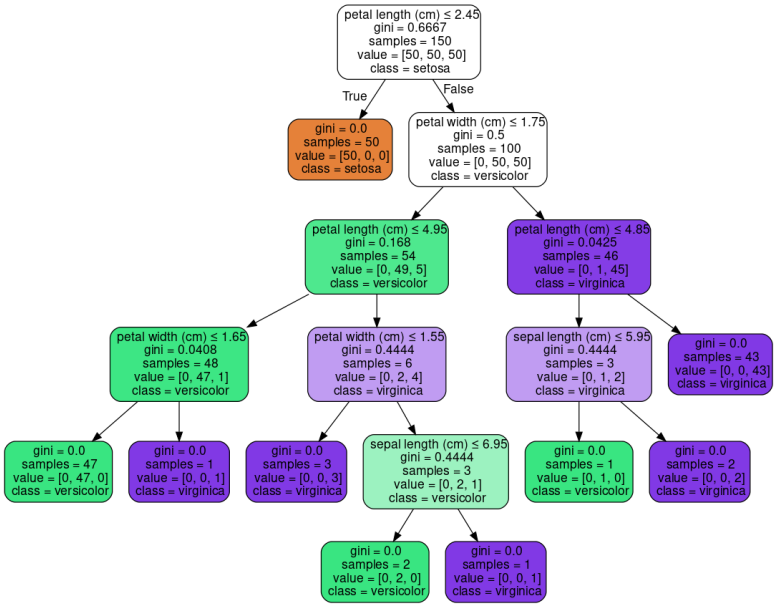

Figura 4.1: Arbre de decisió (Iris). Font: Scikit-learn

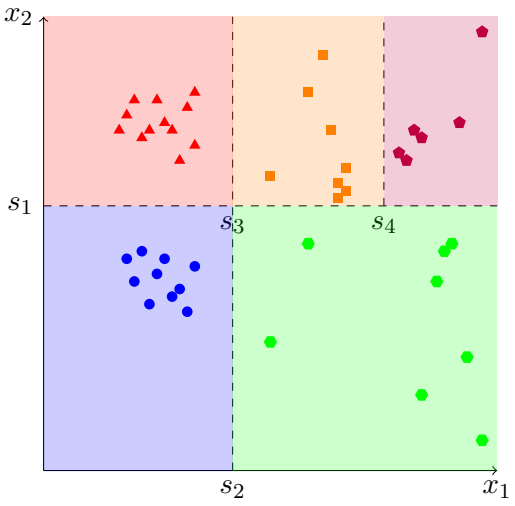

Figura 4.2: Separació de l'espai dimensional mitjançant *CART*. Font: Pròpia.

següent pseudocodi per a la presa de decisions:

```
if x1 < s2:
  if x2 < s1: Classe: Blau
  else:       Classe: Vermell
else
  if x2 < s1: Classe: Verd
  else:
      if x1 < s4: Classe: Taronja
      else:       Classe: Lila
```

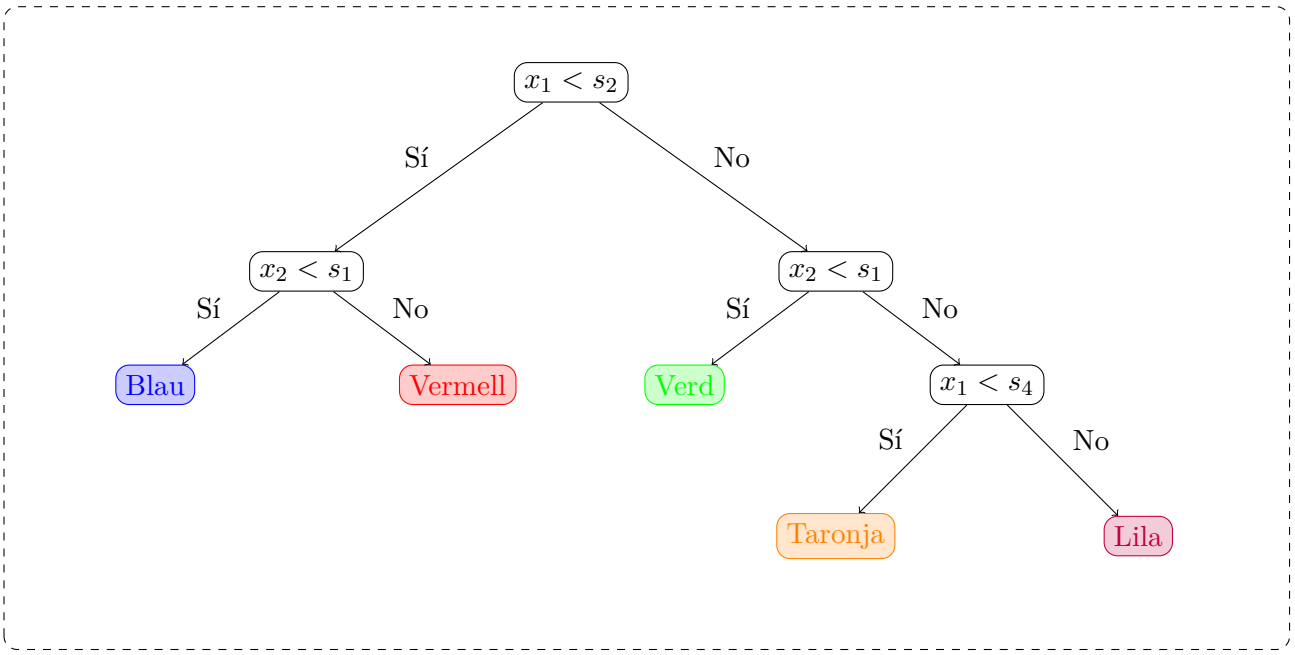

Figura 4.3: Arbre resultant aplicant *CART*. Font: Pròpia.

No obstant, aplicant l'algorisme *CART*, s'obtindrà l'arbre de decisió de la Figura 4.3, que intenta representar de la manera més precisa possible totes les classes, segurament obtenint valors de partició semblants (o inclús diferents) a cada node.

4.1.2 Procediment de l'algorisme

Disposem de vectors d'entrenament $x_i \in R^n$ (*features*) i un vector de classificació $y \in R^l$ (*labels*). Definim Q_m el conjunt dades al node m amb n_m mostres. Per cada candidat particionem $\theta = (f, t_m)$ amb un *feature* f i un llindar (*threshold*) t_m. Doncs, particionem l'arbre binari en $Q_m^{esquerra}(\theta)$ i $Q_m^{dreta}(\theta)$ subsets. Doncs:

$$Q_m^{esquerra}(\theta) = (x, y)|x_f \leq t_m \tag{4.1}$$

$$Q_m^{dreta}(\theta) = Q_m \setminus Q_m^{esquerra}(\theta) \tag{4.2}$$

Llavors, es seleccionen els paràmetres que minimitzen l'impuritat, en concret, un *feature* f i un llindar t_m. El candidat de la partició del node m serà seleccionat en base a una funció de pèrdua $H()$ (o impuritat):

$$G(Q_m, \theta) = \frac{n_m^{esquerra}}{n_m} H(Q_m^{esquerra}(\theta)) + \frac{n_m^{dreta}}{n_m} H(Q_m^{dreta}(\theta)) \tag{4.3}$$

Existeixen vàries funcions d'impuritat $H()$ en problemes de classificació, en el nostre cas, aplicarem *Gini Index*:

$$H(Q_m) = \sum_{k=1} p_{mk}(1 - p_{mk}) \tag{4.4}$$

Doncs, p_{mk} defineix la proporció de classes k que sobserven a la partició al node m. El valor de l'índex de Gini pot variar entre 0 i 1, mesura la variància total entre les classes, doncs, si pren un valor proper al 0, significa que la partició al node m conté principalment observacions d'una sola classe (menor impuritat), si pren un valor proper a l'1, les mostres de les dades es distribueixen uniformement entre les classes (major impuritat).

4.2 Poda d'arbres (*Pruning*)

Com s'observa a la Secció 2.2.2, l'overfitting pot ser un problema significatiu en el procés d'entrenament. Per evitar-lo, en l'algorisme de decisió d'arbres, és important limitar la profunditat de l'arbre i utilitzar altres tècniques de regularització, com podria ser la reducció de la dimensionalitat (reducció del nombre de *features* del dataset) o aplicant *Pruning*.

En aquest cas, usarem principalment *Minimal Cost-Complexity Pruning (MCCP)*, és un algorisme parametritzat per $\alpha \geq 0$, aquest, definirà la mesura de complexitat $R_\alpha(T)$ del arbre (T):

$$R_\alpha(T) = R(T) + \alpha|\tilde{T}| \tag{4.5}$$

On $\tilde{T}$ és el nombre de nodes terminals i $R(T)$ definit com la taxa total de classificació errònia, però, Scikit-learn utilitza la mitjana poderada total de la impuritat dels nodes terminals (nodes que no tenen fills/branques que continuïn) per a $R(T)$. La impueresa vidrà definida pel *Gini Index* dels nodes, doncs, *MCCP* trobarà un subarbre de T que minimitzi $R_\alpha(T)$.

La mesura de complexitat d'un node és $R_\alpha(t) = R(t) + \alpha$, a més, la branca T_t defineix l'arbre on el nod t és l'arrel principal. Llavors, normalment la impuritat d'un node és major a la suma de la impuritat dels seus nodes terminals $R(T_t) < R(t)$. Però, la mesura de complexitat al node t i la branca T_t poden ser iguals en funció del paràmetre de complexitat α. Per aquest motiu, es defineix un α efectiu d'un node on la mesura de complexitat siguin iguals on:

$$R_\alpha(T_t) = R_\alpha(t) \tag{4.6}$$

$$\alpha_{eff}(t) = \frac{R(t) - R(T_t)}{|\tilde{T}| - 1} \tag{4.7}$$

Finalment, un node (no terminal) amb el valor més petit de α_{eff} serà el node amb l'enllaç més feble, per aquest motiu, serà retallat del arbre. El procès finalitzarà quan α_{eff} sigui major al paràmetre de complexitat α.

4.3 Boscos d'arbres aleatoris

A la Secció 4.1, s'observa la gran versatilitat dels arbres de decisió i el potent algorisme *CART* per a la construcció i generació d'aquests arbres. No obstant, existeix una altra metodologia d'aprenentatge coneguda com a *Ensemble Methods* (mètodes d'aprenentatge en conjunt). Aquesta metodologia es basa en el principi de la «sabiduria de la multitud» (*wisdom of the crowd*), que fa referència a la idea que, un conjunt d'individus pot prendre decisions més precises i fiables que un sol individu. Els mètodes d'aprenentatge en conjunt combinen les prediccions de diversos models i classificadors per arribar a una decisió més precisa i robusta. [Gér19]

Els boscos d'arbres aleatoris, com el seu nom indica, representen un conjunt d'arbres de decisió. La seva principal característica és la introducció d'aleatorietat durant el procés d'entrenament de cada arbre que forma el bosc. En lloc de seleccionar la millor característica per a fer les divisions en cada pas de l'arbre, es pren de manera aleatòria un subconjunt de característiques. Aquesta introducció d'aleatorietat permet obtenir arbres diferents que aprenen patrons diversos en les dades.

Ara bé, aquesta aleatorietat també té un impacte en la reducció del impacte de l'*overfitting* com s'ha vist en la Secció 2.2.2. Amb els boscos d'arbres aleatoris, la incorporació d'aleatorietat en la selecció de *features* (característiques) en el procés d'entrenament ajuda a diversificar els arbres. Doncs, els boscos d'arbres aleatoris tenen una millor capacitat per a generalitzar i obtenir resultats més robustos en noves dades, fent que l'overfitting ja no sigui un problema tan greu com en altres models de machine learning.

Finalment, una vegada finalitzat l'entrenament dels boscos d'arbres, com s'observa a la Figura 4.4, la predicció es basa en una simple votació. Cada arbre del bosc emet una predicció, i la predicció final s'obté a partir d'una votació de les prediccions individuals. Aquest procés, pot reduir l'impacte dels errors individuals i obtenir una predicció final més precisa i fiable en comparació amb la predicció d'un únic model.

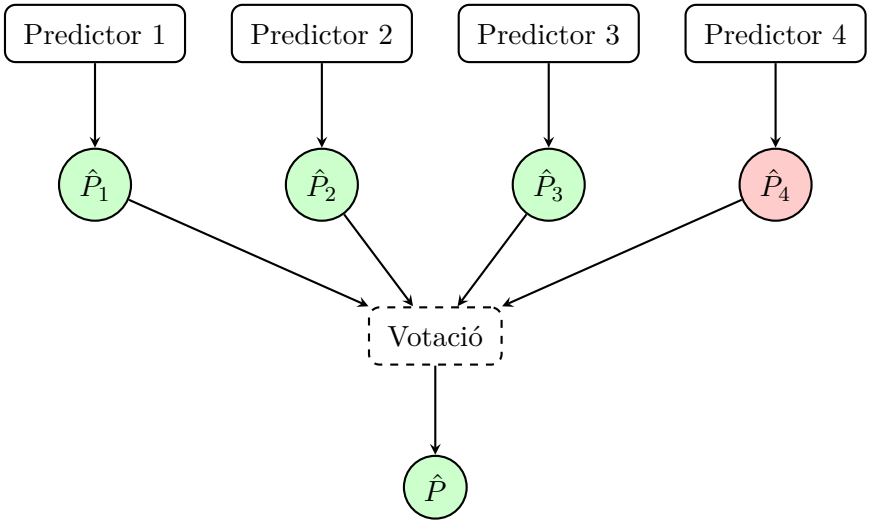

Figura 4.4: Aprenentatge en conjunt. Font: Pròpia.

4.3.1 *Bootstraping*

Com s'ha mencionat prèviament, l'aplicació de *Ensemble Methods* com ara permet obtenir diferents arbres utilitzant dades diferents. En concret, una tècnica comuna és l'aplicació del *Bootstraping* també anomenat *Bagging*. Aquest mètode obté una selecció aleatòria de les mostres (*samples*) de les dades originals. Això significa que per a cada model s'obtindrà un subconjunt de dades diferent, com es mostra a la Figura 4.5.

Com s'observa a la Figura 4.5, amb l'aplicació del *Bootstraping*, algunes mostres poden ser seleccionades vàries vegades en diferents subconjunts, però, altres mostres poden no ser seleccionades en cap dels subconjunts com per exemple $\{x_6, x_7, x_9\}$. Per a cada subconjunt de dades, les mostres no seleccionades es coneixen com a *out-of-bag samples (OOB)*.

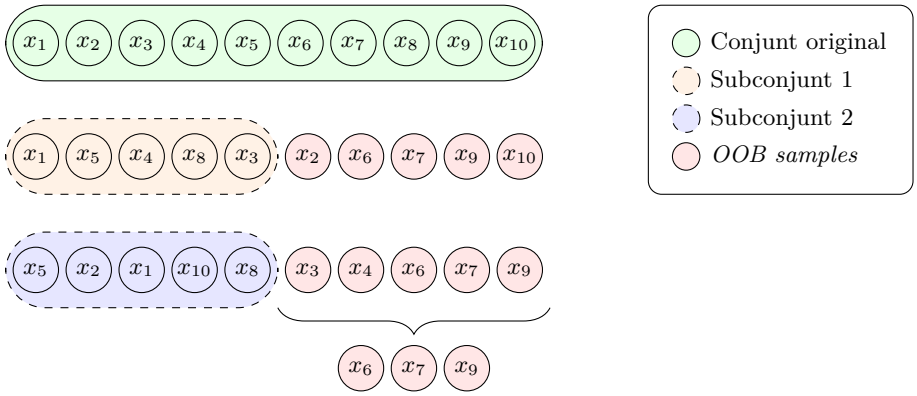

Figura 4.5: *Bootstrap Aggregating*. Font: Pròpia

La creació de subconjunts diferents permet treballar amb diverses característiques de les dades, introduint variabilitat i possiblement millorant la generalització del model final. A més, les *OOB samples* poden utilitzar-se per avaluar el rendiment del model **sense necessitat de disposar d'un conjunt de validació separat**. Si s'en disposen de varis models, es pot avaluar el rendiment general en funció de les *OOB samples*. Per exemple, es defineix un bosc aleaotri amb 3 arbres amb el següent conjunt de dades:

Feature 1	*Feature 2*	*Label*
0.5	0	A
1.5	1	B
2.0	2	A
1.0	3	B
2.5	4	A

Taula 4.1: Conjunt de dades original

Aplicant *Bootstraping*, cada arbre genera un subconjunt de dades a partir d'una selecció aleatoria de les mostres del conjunt de dades original, tal com es pot apreciar a la Taula 4.3.1.

Model	*Feature 1*	*Feature 2*	*Label*
Arbre 1	0.5	0	A
	2.0	2	A
	1.5	1	B
Arbre 2	1.5	1	B
	1.0	3	B
	2.0	2	A
Arbre 3	0.5	0	A
	2.5	4	A
	1.0	3	B

Taula 4.2: Bootstrap de cada arbre

Llavors, una vegada entrenats tots els arbres, es pot obtenir una avaluació d'aquests sense disposar d'un conjunt de validació a part. Per exemple, en aquests cas, l'arbre 1 disposa de les següents mostres *OOB samples*, és a dir, mostres que no ha seleccionat del conjunt original.

Feature 1	*Feature 2*	*Label*
1.0	3	B
2.5	4	A

Taula 4.3: *OOB samples* de l'arbre 1

A continuació, partint de les *OOB samples* de l'arbre 1, es pot obtenir la taxa d'error d'aquest. Doncs, aquest procés es repeteix per a tots els arbres i s'obté la taxa d'error mitjana del bosc d'arbres aleatoris. D'aquesta manera, la taxa d'error a partir de les *OOB samples* permet obtenir una estimació de l'evaluació del bosc.

Feature 1	*Feature 2*	*Label*	*Predicció*
1.0	3	B	A
2.5	4	A	A

Taula 4.4: Predicció basada en les *OOB samples*

$$\text{OOB Error}_{\text{Arbre 1}} = \frac{\text{Classificacions errònies}}{\text{Mostres totals}} = \frac{1}{2} = 0.5$$

$$\text{OOB Error (Bosc)} = \frac{\sum_1^n \text{Error}_{\text{Arbre } n}}{\text{Quantitat d'arbres}} x = 2$$

4.3.2 *Feature Importance*

Un altre concepte rellevant en els boscos d'arbres aleatoris és el *feature importance*, que permet determinar quines *features* són més importants i tenen un major impacte en la predicció del model. De la mateixa manera que aplicant *Bootsraping* en les mostres de les dades (4.5), es selecciona un subconjunt aleatori de *features* per a cada arbre. A la Figura 4.6 es mostra un exemple per determinar la importància dels píxels en la classificació d'una imatge.

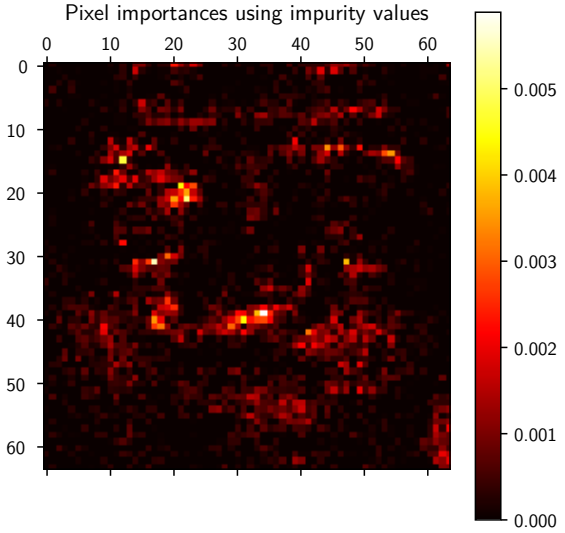

Figura 4.6: Exemple de la importància dels pixels en una imatge. Font: Scikit-learn

«*Gini Importance or Mean Decrease in Impurity (MDI) calculates each feature importance as the sum over the number of splits (across all tress) that include the feature, proportionally to the number of samples it splits*» [Lee17]. En altres paraules, la importància es basa en la reducció de la impuritat *Gini Index* com s'ha vist a la Secció 4.1.2. Llavors, a cada partició de l'arbre es mesura la reducció de la impuritat, es diu que una *feature* és important quan el valor de la impuritat es veu reduïda significativament. Resumidament, es pot obtenir informació addicional sobre les dades, com per exemple, determinar quines assignatures son importants per a la matriculació d'una altra assignatura.

4.4 Mètriques per a models de classificació

Després de finalitzar la fase d'entrenament d'un model, serà necessari realitzar un anàlisi i avaluació de qualitat d'aquest. Les mètriques de classificació són eines que permeten mesurar la qualitat d'un model de classificació, aquestes, s'utilitzen per avaluar el rendiment d'un model predictiu i determinar la seva precisió en la classificació de dades desconegudes que el model mai haurà vist ni memoritzat. A la Taula 4.4 representa una matriu de confusió que compara les prediccions d'un model amb les dades reals (actuals), en concret, és una eina important per avaluar la qualitat d'un model de classificació.

	Positiu (predicció)	Negatiu (predicció)
Positiu (actual)	Positiu Real (TP)	Fals Negatiu (FN)
Negatiu (actual)	Fals Positiu (FP)	Negatiu Real (TN)

Taula 4.5: Matriu de confusió. Comparació de la predicció d'un model amb les dades reals (actuals)

Les mètriques més comunes son les següents:

- *Exactitud (accuracy).* És la proporció de prediccions correctes obtingudes pel model sobre tot el conjunt de prediccions. Mesura la capacitat del model per classificar correctament els casos positius i negatius. Un valor d'exactitud elevat significa que el model és bo en la identificació tant de casos positius com negatius. Però, l'exactitud pot ser bastant **enganyosa** en alguns casos. Una de les principals limitacions és que no té en compte la distribució de les classes en el conjunt de dades. En altres paraules, si un conjunt de dades no disposa d'una distribució uniforme (és a dir, una classe té una representació molt més gran que d'altres), un classificador que simplement prediu la classe "majoritària" o amb més casos per a cada classe pot aconseguir una alta exactitud. Per exemple, en un conjunt de dades amb un 95% d'exemples positius i un 5% d'exemples negatius, un classificador que sempre prediu "posiu" aconseguiria una alta exactitud del 95%. No obstant això, aquest classificador no seria realment útil, ja que no seria capaç de predir correctament cap dels exemples negatius.

$$exactitud = \frac{TP + TN}{TP + TN + FP + FN} \tag{4.8}$$

- *Recall.* És la proporció de casos positius reals que són correctament identificats pel model com a positius. En altres paraules, mesura la capacitat del model per identificar tots els casos positius. Un valor de recall elevat significa que el model té bona detecció de tots els casos rellevants.

$$recall = \frac{TP}{TP + FN} \tag{4.9}$$

- *Precisió.* És la proporció de casos positius que són correctament identificats pel model com a positius. En altres paraules, mesura la capacitat del model per identificar correctament els casos positius sense classificar incorrectament casos negatius com a positius. Un valor de precisió elevat significa que el model té bona detecció dels casos positius, però pot perdre alguns casos positius reals.

$$precisió = \frac{TP}{TP + FP} \tag{4.10}$$

- *F1-score.* És la mitjana harmònica de la precisió i el recall. Una puntuació F1 elevada significa que el model té una bona precisió i recall, i pot identificar correctament els casos positius sense classificar incorrectament casos negatius com a positius.

$$f1_{score} = 2\ \frac{precisió \cdot recall}{precisió + recall} \tag{4.11}$$

Resumidament, segons l'aplicació interessarà maximitzar una mètrica o una altra. Per exemple, en l'àmbit de la salut és extremadament important evitar falsos negatius (casos positius que són classificats incorrectament com a negatius) on la detecció precoç d'una malaltia pot ser crítica per a la supervivència d'un pacient. Doncs, el valor de recall és una mètrica important a considerar. En altres aplicacions es volen evitar falsos positius (casos negatius que són classificats incorrectament com a positius), on la precisió pot ser més important. Per exemple, en la detecció d'alarmes, és millor que es doni una falsa alarma (fals positiu) que no pas una alarma real no s'arribi a detectar (fals negatiu).

5 Arquitectura software

En aquest capítol, es presentarà l'arquitectura software utilitzada en el desenvolupament del treball. Aquesta arquitectura és essencial per a la preparació i anàlisi dels experiments realitzats. Es descriuran els components principals d'aquesta arquitectura i les seves interconnexions.

5.1 Eines utilitzades

Per utilitzar estratègies basades en *Machine Learning*, existeix una gran quantitat de llibreries populars com TensorFlow, Keras, PyTorch, entre d'altres. En aquest treball, s'ha decidit treballar amb *Scikit-learn*, una llibreria popular de Python que ofereix una gran varietat d'algorismes, models, mètriques, etc. La Figura 5.1 mostra alguns dels algorismes disponibles en Scikit-learn.

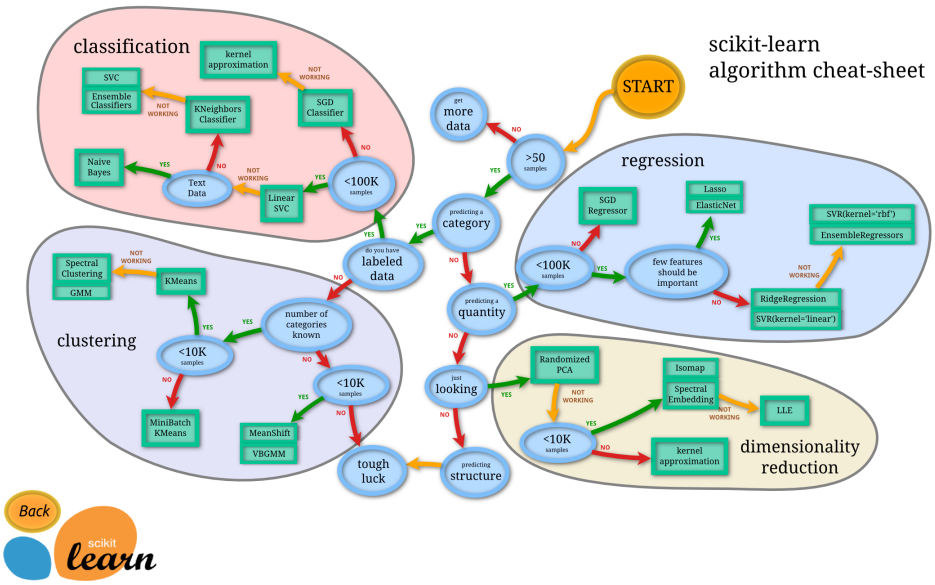

Figura 5.1: Algorismes (models) en Scikit-learn. Font: Wikipedia

En primer lloc, Python és un llenguatge de programació molt popular en l'àmbit de la ciència de dades i el *Machine Learning*. La seva sintaxi simple facilita el desenvolupament de codi, a més, permet una gran flexibilitat i rapidesa en l'àmbit de la recerca i la experimentació. També, Python disposa d'una gran quantitat de paquets i llibreries, com per exemple:

- *NumPy*. Una extensió de Python que ofereix operacions matemàtiques avançades i manipulació de dades (vectors i matrius) de manera eficient.

- *Pandas.* És eina extremadament potent, flexible i fàcil d'utilitzar per a l'anàlisi i la manipulació de dades. Proporciona estructures de dades com per exemple *DataFrames*, que faciliten la gestió i la transformació de conjunts de dades.
- *Matplotlib.* Una biblioteca especialitzada en la generació de gràfics estàtics o animats a partir de les dades. Amb Matplotlib, és possible crear una gran varietat de visualitzacions, com ara gràfics de línies, barres, dispersió, histograma, entre d'altres.

En segon lloc, encara que altres llenguatges com C++ són més eficients (velocitat d'execució, etc.), en aquest treball es prioritza la facilitat de desenvolupament i la facilitat d'iterar i dur a terme varis experiments. De fet, en l'àmbit profesional, un cop s'ha finalitzat la recerca amb un model eficient i provat, es pot considerar portar aquest model a producció i optimitzar-lo utilitzant altres llenguatges més eficients com C++.

En resum, l'elecció de Python i l'ús de *Scikit-learn* es basa en la flexibilitat de la sintaxi, la facilitat de desenvolupament i la gran quantitat de paquets i llibreries disponibles, amb l'objectiu de realitzar una recerca eficient.

5.2 Sistema de tractament i depuració de dades

Com s'ha explicat anteriorment, és important tractar les possibles anomalies i problemes per aconseguir un conjunt de dades consistent i coherent que permeti una anàlisi adequada i resultats fiables.

Per aquest motiu, es defineix un sistema de tractament i manipulació de dades que consta de diversos mòduls. Aquests mòduls realitzen operacions sobre el conjunt de dades per depurar, filtrar i assegurar que el conjunt de dades finals no presentin cap tipus d'anomalies.

A partir d'un històric amb els expedients de tots els estudiants, amb el format de la Taula 2.1, es volen definir possibles transformacions per crear dos conjunts de dades que seran utilitzats per entrenar els models. Per tant, s'estableix una conjunt de mòduls que faciliten el tractament i la manipulació de les dades tal i com es mostra a la Figura 5.2.

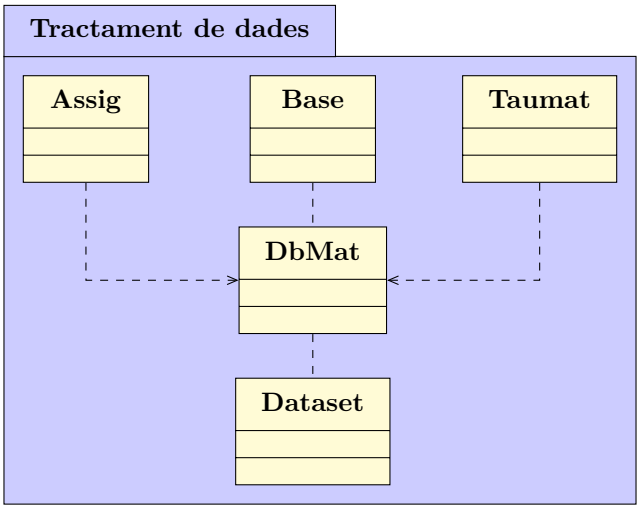

Figura 5.2: Manipulació i creació del dataset. Font: Pròpia.

A continuació, es mostra una breu introducció sobre mòduls principals del sistema (representats en la Figura 5.2):

- Assig. Aquest mòdul conté informació sobre les assignatures del pla d'estudis, en concret, el codi, acrònim i el nom de la assignatura.
- Base. Defineix la representació d'un quadrimestre, amb l'ajuda d'operacions i redefinicions natives de Python com per exemple la definició d'igualtat, d'ordre, entre altres.
- Taumat. Conté la taula d'acrònims i proporciona informació sobre possibles anomalies, com ara la falta d'acrònims, acrònims iguals o repetits. repetits.
- DbMat. Estableix vàries representacions (classes objecte) sobra la estructura del historials acadèmics. En concret, defineix les següents representacions:
  - La classe `Mat` representa la matricula d'una assignatura i el resultat obtingut.
    - `assig (Assig)`: L'objecte assignatura
    - `nota (float)`: Nota obtinguda

    - `notad (str)`: Descripció de la nota, per exemple 'N' (notable), 'NP' (no presentada), 'MH' (matricula d'honor), etc.
    - `tipusn (str)`: Tipus de nota
  - La classe `BlkMat` representa el bloc de matriculas que es matriculen simulaniament.
    - `becat (bool)`: Si l'estudiant disposa de beca
    - `lstmat (list[Mat])`: El llistat de matricules realitzades.
  - La classe `Exped` representa l'expedient d'un estudiant per complet.
    - `idexp (int)`: L'identificador de l'expedient (anonimitzat)
    - `notae (float)`: Nota d'accés
    - `viae (str)`: Via d'accés
    - `ordre (int)`: Ordre d'assignació
    - `anyn (int)`: Any de naixement
    - `mats (dict[Quad, BlkMat])`: Totes les matriculacions que ha realitzat l'estudiant
  - La classe `BDExped` representa l'expedient d'un estudiant per complet.
    - `tm (int)`: La taula de matriculacions de forma crua
    - `ta (float)`: La taula d'acrònims
    - `pe (str)`: El pla d'estudis corresponent
    - `bd (dict[idexp, Exped])`: Conté els expedients de tots els estudiants

- DATASET.PY. És el mòdul encarregat de definir transformacions sobre el preprocessat previ de les dades crues que realitza DBMAT.PY. A continuació la superclasse `Dataset` simplement defineix les operacions principals *transform* i *load_data*, on subclasses de `Dataset` definiran la seva pròpia transformació, d'aquesta manera s'obté una classe abstracta que representa una transformació qualsevol on només s'haurà de redefinir aquests mètodes.

```python
class Dataset(object):
    def __init__(self, file: str = None) -> None:
        self.raw_df =  pd.read_csv(file, low_memory=False) if file else None
        self.ta      = None
        self.dataset = None

    def add_ta(self, ta) -> None:
        """Afegeix la taula `ta` d'acrònims
        """
        self.ta = ta

    def transform(self) -> pd.DataFrame:
        """Aplica la transformació de dades per crear i retornar un dataset
        """
        raise NotImplemented

    def load_data(self) -> Tuple[float, str]:
        """Retorna les les dades d'entrenament de la forma (X, y)
        """
        raise NotImplemented
```

Listing 1: Representació d'un *dataset*

Llavors, una vegada definida tota la arquitectura necessària per a la gestió, depuració i manipulació de les dades, s'obtè un programa principal (Codi 2) fàcil de tractar i executar (Figures 5.3 i 5.4).

```python
if __name__ == '__main__':
    # Taula acrònims
    (tm, ta) = CarregaTaules(
        nom_mat=RAW_MAT_FILE_PATH,
        nom_acr=RAW_ACRO_FILE_PATH,
        reporta=False
    )
    # Càrrega de les dades crues per a la 1a transformació
    pt = PrimeraTransformacio(file=INTERIM_PATH / 'dataset_base.csv')
    pt.add_ta(ta)
    # Transformació
    ds = pt.transform()
    # Emmagatzemament
    ds.to_csv(PROCESSED_PATH / 'primerDataset.csv', index=False)
```

Listing 2: Programa principal (mòdul *dataset*).

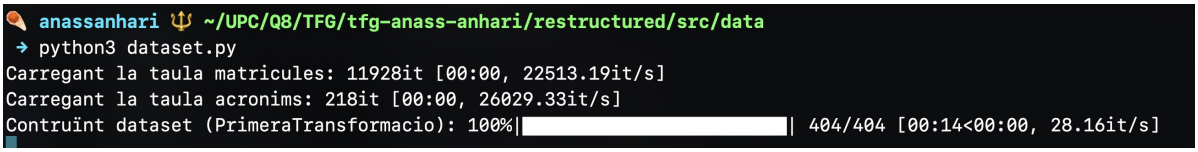

Figura 5.3: Execució de la primera tranformació. Font: Pròpia

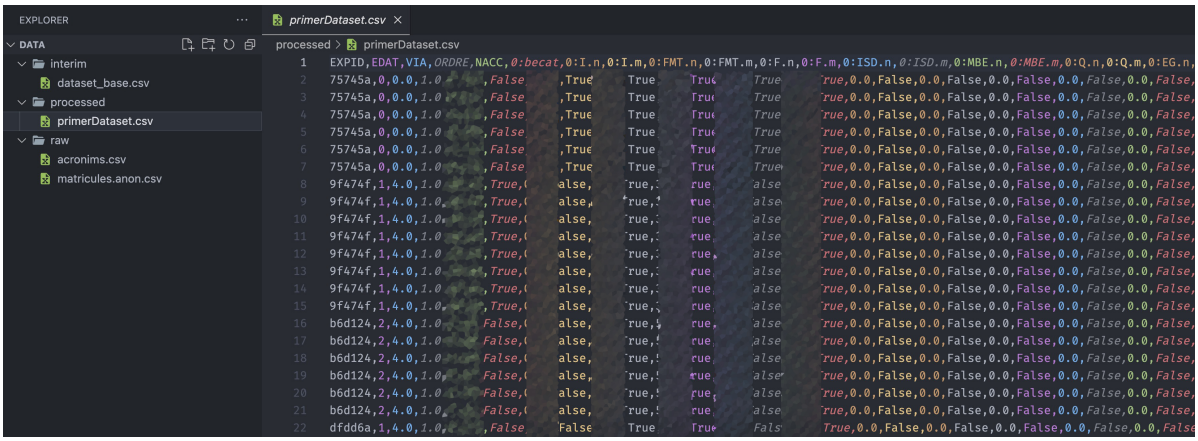

Figura 5.4: Primera tranformació resultant. Font: Pròpia

5.3 Sistema d'entrenament i predicció

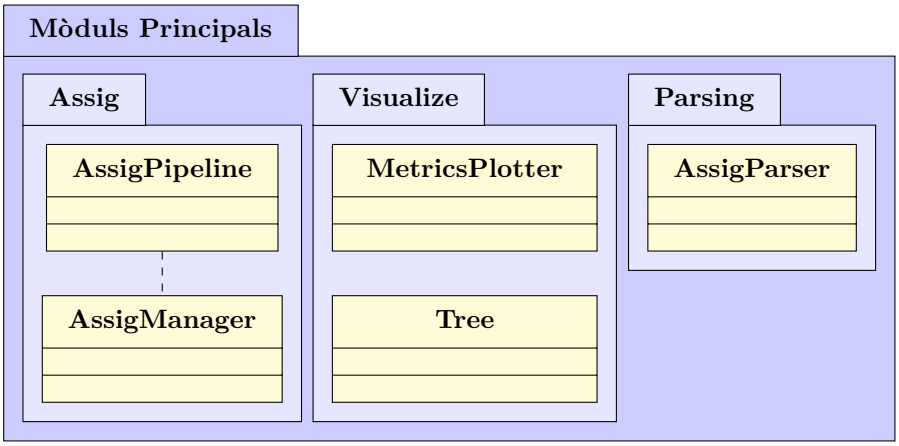

Figura 5.5: Mòduls del sistema d'entrenament

A continuació, es mostra una breu introducció sobre mòduls principals del sistema (representats en la Figura 5.5):

- Assig
  - *AssigPipeline*. Aquest mòdul conté una *pipeline* (Figura 5.8) d'entrenament amb el model que es vulgui utilitzar. S'encarrega d'aplicar les últimes transformacions necessàries sobre les dades ja processades i netes, com ara les transformacions de les *features* categòriques.
  - *AssigManager*. Disposa de les *pipeline*s de totes les assignatures. S'encarrega d'afegir el model de cada una de les assignatures seleccionades i entrenar-les.
- Visualize
  - *MetricsPlotter*. S'encarrega de visualitzar les dades i totes les possibles mètriques necessàries, mantenint sempre la mateixa estructura de diagrames i visualitzacions.
  - *Tree*. Defineix la visualització i la representació dels arbres de decisió.
- Parsing
  - *AssigParser*. Parseja les instruccions del programa principal. Defineix les possibles opcions necessàries per obtenir un aplicatiu senzill amb menús d'ajuda i altres funcionalitats.

```
anassanhari 🔱 ~/UPC/Q8/TFG/tfg-anass-anhari/restructured/src
➜ python3 main.py --help
usage: main.py [-h] -ds {v1,v2} (--all | --normal | --opt | --q2 | --q3 | --q4 | --q5 | --q6 | --q7)

Fase 1: Arbre de decisió

optional arguments:
  -h, --help            show this help message and exit
  -ds {v1,v2}, --dataset {v1,v2}
                        Dataset (v1) sense la motxilla de l'estudiant i (v2) amb la motxilla
  --all                 Tots els acrònims
  --normal              Acrònims habituals per testeig del model
  --opt                 Acrònims de test per optatives
  --q2                  Acrònims de test (Q2)
  --q3                  Acrònims de test (Q3)
  --q4                  Acrònims de test (Q4)
  --q5                  Acrònims de test (Q5)
  --q6                  Acrònims de test (Q6)
  --q7                  Acrònims de test (Q7)
```

Figura 5.6: Menú d'ajuda del mòdul d'entrenament i predicció. Font: Pròpia

```
anassanhari 🔱 ~/UPC/Q8/TFG/tfg-anass-anhari/restructured/src
➜ python3 main.py -ds v1 --all
Carregant la taula matricules: 11928it [00:00, 22324.62it/s]
Carregant la taula acronims: 218it [00:00, 24854.12it/s]
[INFO] Carregant dataset (Versió 1)
[INFO] X_test shape: (342, 900)
[INFO] y_test shape: (342, 45)
[INFO] X_train shape: (1368, 900)
[INFO] y_train shape: (1368, 45)
Entrenant els models: 100%|████████████████████████████████████| 40/40 [00:08<00:00,  4.93it/s]
8.121639966964722
```

Figura 5.7: Exemple d'entrenament

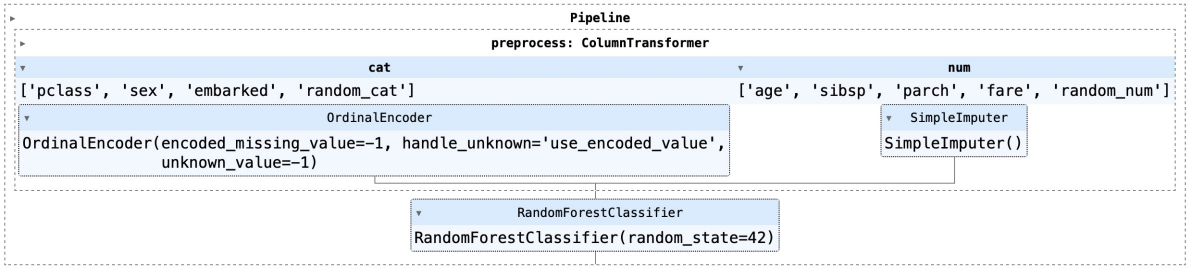

Figura 5.8: Exemple d'una *Pipeline*. Font: Scikit-learn

6 Resultats obtinguts

En aquest treball, s'ha decidit aplicar una metodologia clara i concisa per a la experimentació i comparació de resultats amb l'objectiu d'obtenir el màxim de detalls possibles i poder realitzar comparacions entre els diferents models, el gran ventall de paràmetres, entre d'altres. Degut a la gran quantitat de diferents possibilitats i experiments, la metodologia principal és la descripció dels experiments en forma de taula, cadascun identificat amb un identificador únic. Aquestes taules representen una descripció exhaustiva dels paràmetres utilitzats en cada experiment incloent-hi l'ús de tècniques com el *Bootstraping*, com s'ha discutit anteriorment a la secció 4.3.1.

Mitjançant aquest enfocament, es pot explorar de manera exhaustiva un ampli ventall de combinacions de paràmetres, amb l'objectiu d'identificar les configuracions més efectives. A més, aquesta metodologia permet una comparació estructurada i reproductible dels resultats obtinguts, proporcionant un anàlisi més fiable.

6.1 Primeres impressions, arbres de decisió

L'objectiu principal d'aquesta secció és proporcionar una primera impressió inicial mitjançant l'ús d'arbres de decisió. Doncs, es defineix la següent taula d'experimentació per obtenir una primera avaluació de les prediccions de cadascun dels models.

Identificador (Id)	*Model*	*Profunditat*	*Transformació*	*Motxilla*
DT3T1	ARBRE	3	1	No
DT4T1	ARBRE	4	1	No
DT5T1	ARBRE	5	1	No

Taula 6.1: Experiments (0)

A partir dels model DT3T1 definit a la Taula: 6.1 amb la primera transformació de dades sense cap informació addicional de l'estudiantat, s'han obtingut els resultats representats a la Figura 6.1 i Taula 6.2. A primera vista, es pot observar que la majoria dels models aconsegueixen una capacitat de predicció superior al 80%. No obstant això, cal destacar un aspecte important en les assignatures del primer quadrimestre, ja que la seva capacitat de predicció és pràcticament nul·la. Aquest fenòmen, es produeix principalment per la falta d'historial prèvi dels estudiants en la carrera. Com es mostra a la Figura 6.2, això implica que les úniques matriculacions possibles són les dels estudiants repetidors, ja que no existeixen dades prèvies per als estudiants que es matriculen per primera vegada. En resum, es pot concloure que no té sentit crear models per a les cinc primeres assignatures del primer quadrimestre. De fet, les regles de matrícula durant la fase inicial de la carrera ja es coneixen, ja que són implícites.

Adicionalment, es pot observar el mateix efecte amb les assignatures optatives i el Treball de Fi de Grau (TFG). En concret, hi ha poques matriculacions en les optatives, especialment

a causa de les convalidacions. Aquesta situació dificulta que el model pugui generalitzar el comportament dels estudiants en funció de les seves matriculacions anteriors.

Assignatura	Recall	Precisió	F1-score
MBE	0.000	0.000	0.000
F	0.000	0.000	0.000
I	0.000	0.000	0.000
ISD	0.000	0.000	0.000
FMT	0.400	0.667	0.500
ES	0.899	0.984	0.939
TCO1	0.955	0.985	0.970
TP	0.886	0.954	0.919
SD	0.867	1.000	0.929
TCI	0.871	0.984	0.924
MAE	0.684	0.839	0.754
TCO2	0.725	0.740	0.733
DP	0.648	0.920	0.760
EM	0.490	0.800	0.608
CSL	0.719	0.885	0.793
SA	0.845	0.778	0.810
PBN	0.726	0.833	0.776
ACO	0.807	0.821	0.814
CSR	0.650	0.907	0.757
SS	0.574	0.946	0.714
PCTR	0.627	0.889	0.736
GOP	0.717	0.905	0.800
SO	0.667	0.865	0.753
XC	0.660	0.892	0.759
PDS	0.704	0.760	0.731
SEN	0.684	0.867	0.765
ESI	0.750	0.909	0.822
ASSI	0.825	0.825	0.825
SEC	0.857	0.833	0.845
IS	0.794	1.000	0.885
SAR	0.967	0.879	0.921
TFG	0.667	0.667	0.667
MIC	0.214	0.273	0.240
SC	0.167	0.250	0.200
SSCI	0.000	0.000	0.000
AE	0.000	0.000	0.000
GQSIQSMA	0.000	0.000	0.000
BD	0.333	0.562	0.419
IU	0.111	0.333	0.167
RE	0.000	0.000	0.000

Taula 6.2: Avaluació dels models de les assignatures (Id: Dt3t1)

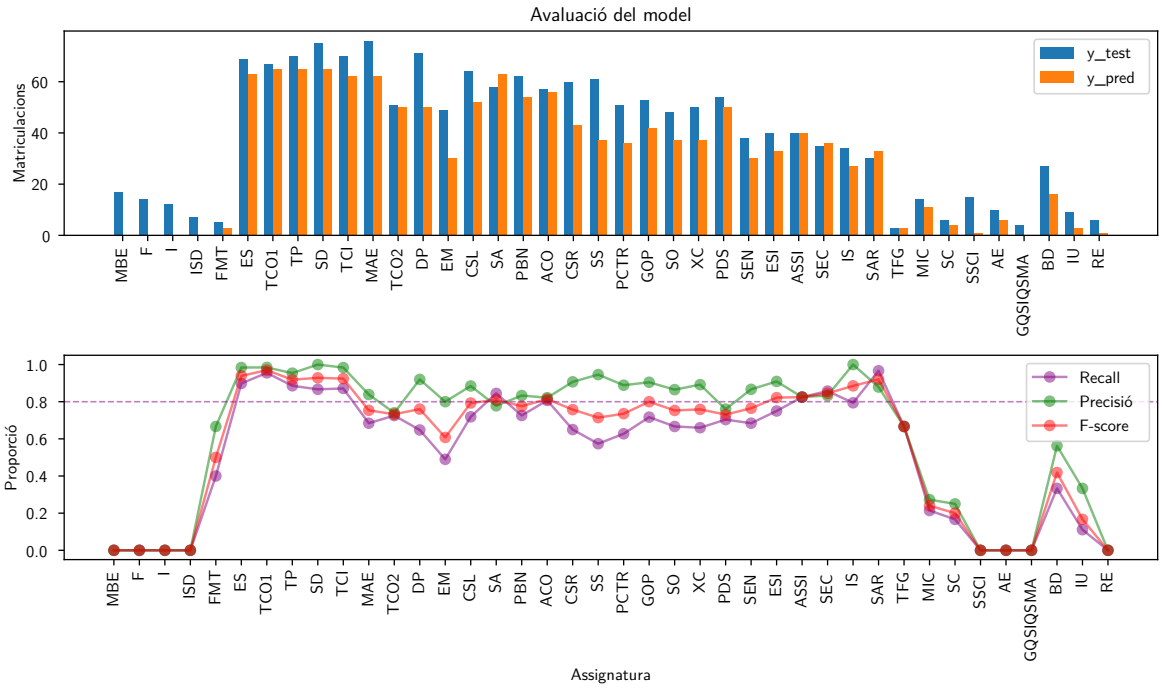

Figura 6.1: Avaluació dels models de les assignatures (Id: Dt3t1)

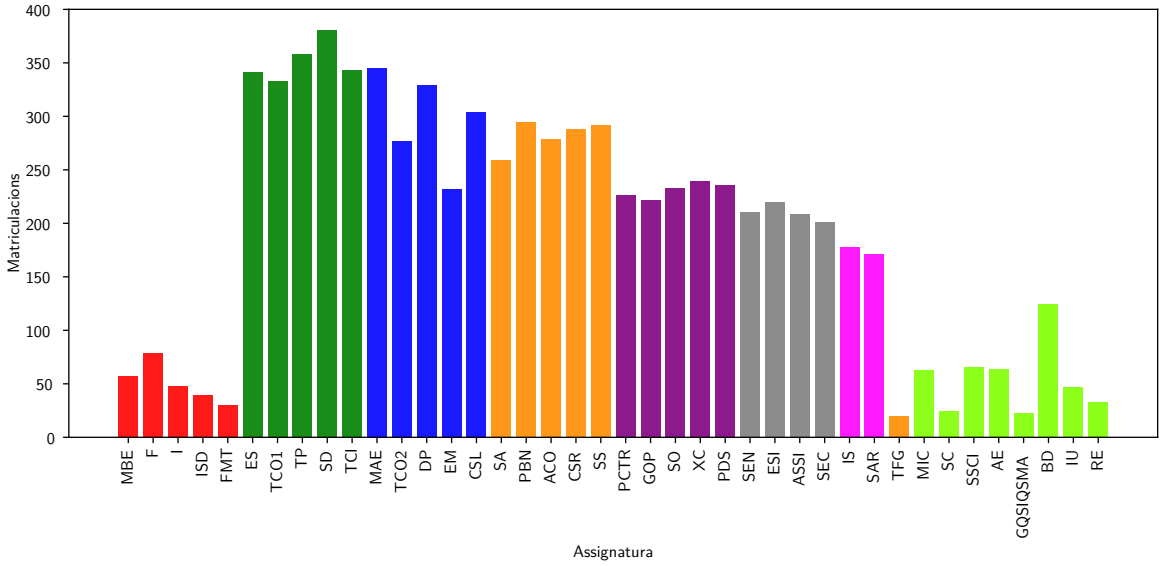

Figura 6.2: Matriculacions totals de cada assignatura[1]

[1]No es té en compte les matriculacions realitzades en la primera inscripció de la carrera

6.1.1 Anàlisi dels arbres

Dels arbres s'obtenen certes regles que ajuden a analitzar, visualitzar i entendre la matriculació d'una assignatura. A continuació, es mostren les regles obtingudes per a algunes assignatures específiques a partir dels arbres generats amb el model DT3T1:

- De la Figura 6.3, obtenim les següents regles per a *Programació a Baix Nivell* (PBN):
  - Es matricula si l'estudiant va matricular *Dispositius Programables* (DP).
  - No la matricula si ja té aprobada PBN.
- De la Figura 6.4, es pot observar que per matricular-se a *Circuits i Sistemes Lineals* (CLS), hi ha una gran dependència de l'assignatura de *Teoria de Circuits* (TCI). En general, tant els estudiants com els professors perceben que TCI té un pes significatiu per adquirir els coneixements necessaris per a la matriculació de CSL. Així doncs, l'arbre de decisió concorda amb la percepció comuna en la carrera de Sistemes TIC.
- De la Figura 6.5, obtenim les següents regles per a *Aplicacions i Serveis Sobre Internet* (ASSI):
  - Es matricula si va matricular a *Xarxes de Comunicació* (XC) i no a *Enginyeria de Sistemes* (ESI).
  - Es matricula si ha suspès ASSI.
  - No obté la regla bàsica, si aprova ASSI no la hauria de matricular.

Resumidament, a través de l'anàlisi de les regles obtingudes dels arbres de decisió, s'obté una visió més profunda del procés de matriculació i permet identificar els factors clau que influeixen en les decisions dels estudiants. A més aquestes regles poden ajudar a validar i comprovar si les percepcions que tenen tant els professors com els alumnes sobre la importància de determinades assignatures concorden amb els resultats obtinguts. Això permet disposar de varis punts de vista en les decisions relacionades amb la planificació acadèmica.

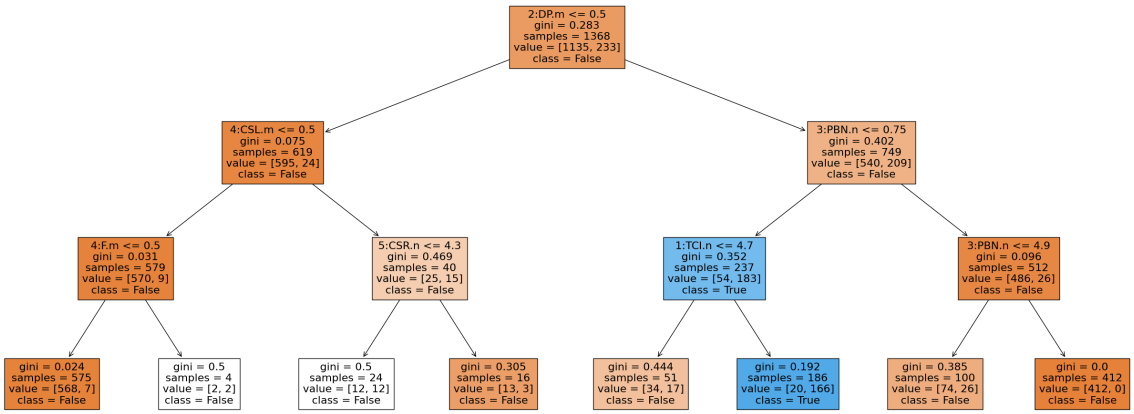

Figura 6.3: Arbre de PBN (ID: DT3T1)

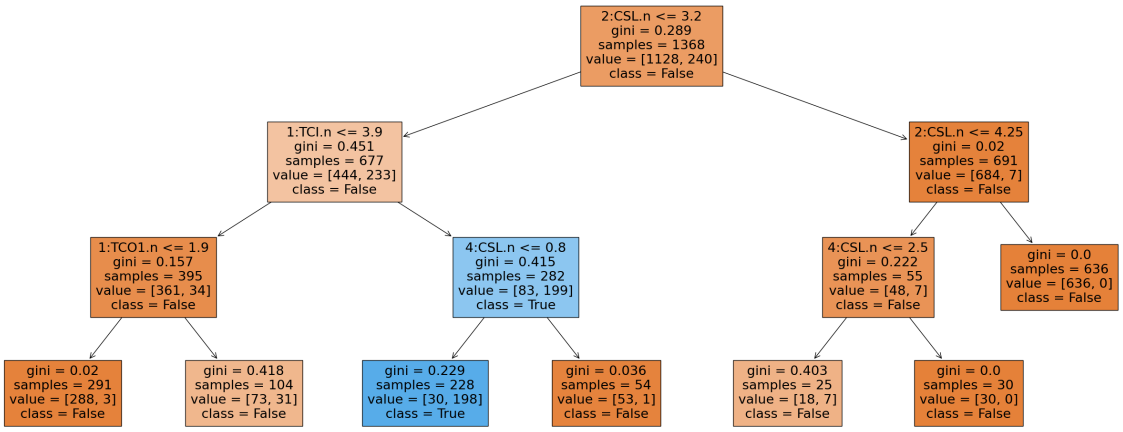

Figura 6.4: Arbre de CSL (Id: Dt3t1)

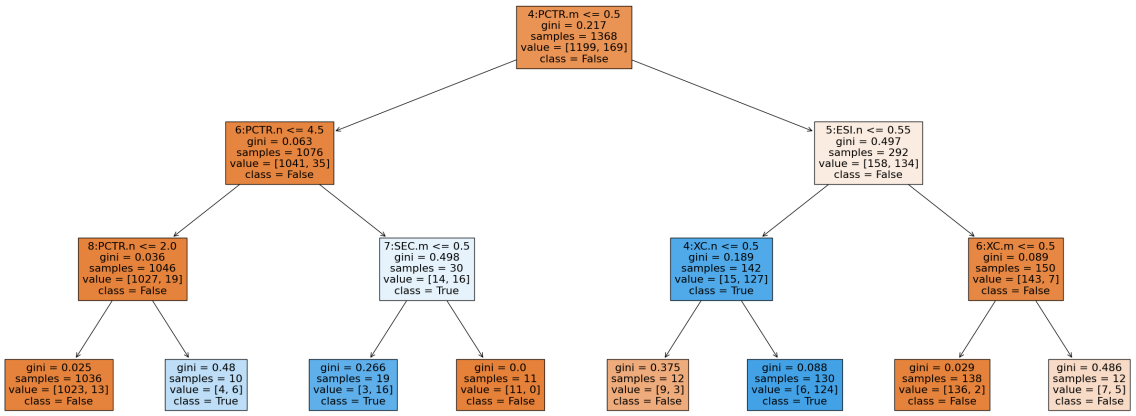

Figura 6.5: Arbre de ASSI (Id: Dt3t1)

6.1.2 Profunditat de l'arbre

S'ha observat prèviament que el model DT3T1 amb una profunditat de 3 nodes proporciona bons resultats. No obstant això, es planteja experimentar amb diferents profunditats i analitzar el comportament resultant. Tot i que s'ha mencionat anteriorment que no té sentit disposar d'un model per a les assignatures del primer quadrimestre, ja que només els estudiants repetidors s'hi matriculen en les assignatures de la fase comuna del primer quadrimestre, és interessant determinar si augmentant la profunditat de l'arbre s'obté aquesta regla implícita. Per aquest motiu, s'han definit els models DT4T1 i DT5T1 de la Taula 6.1, amb profunditats de 4 i 5 nodes, respectivament.

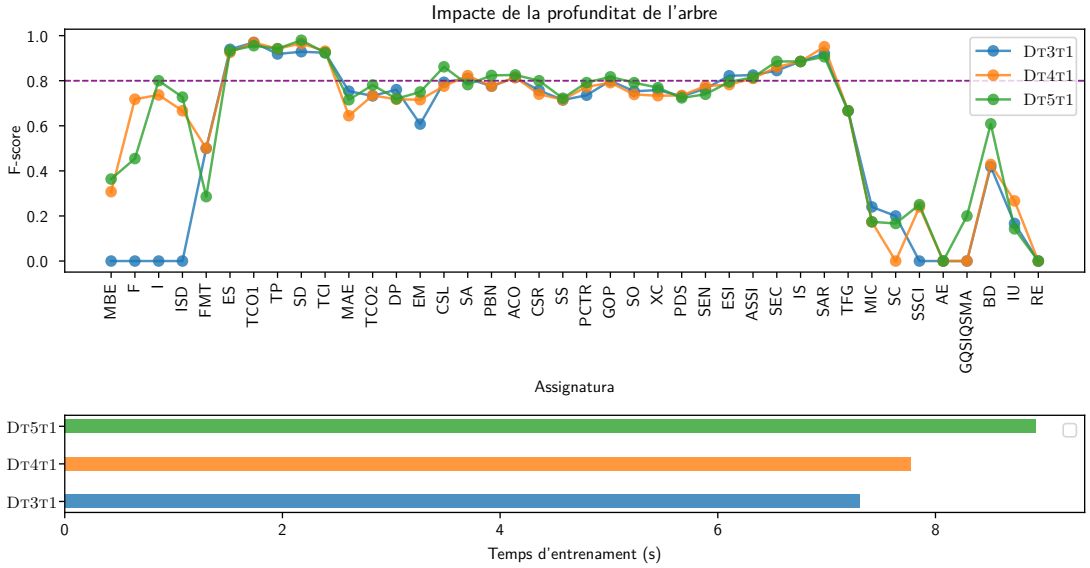

Figura 6.6: Observació de l'impacte de la profunditat dels arbres

De la Figura 6.6 s'observa clarament l'impacte del temps d'entrenament en funció de la profunditat de l'arbre, a més, dels resultats s'obtenen les següents observacions:

- Per a les assignatures obligatòries, la profunditat de l'arbre sembla tenir un impacte mínim en els resultats, excepte per a algunes assignatures. Això indica que el model DT3T1 és capaç de generalitzar adequadament la matriculació i el comportament dels estudiants en aquestes assignatures amb un arbre de profunditat 3.

- En el cas de les assignatures optatives, s'observa que els resultats varien en funció de la profunditat de l'arbre. Algunes optatives milloren en els resultats amb una major profunditat, mentre que en altres optatives els resultats empitjoren. Aquest comportament pot ser degut a la poca quantitat de matriculacions existents (Figura 6.2) i la gran variabilitat i dispersió en les matriculacions a causa de factors com la convalidació de crèdits.

- En les assignatures del primer quadrimestre, s'observa una millora significativa en els resultats:

– A la Figura 6.7, amb el model DT3T1, l'arbre no és capaç de capturar una regla per a la matriculació de l'assignatura *Física* (F), però sí aconsegueix generalitzar la regla bàsica que indica que si l'estudiant aprova l'assignatura, no la torna a matricular.

– A la Figura 6.8, amb el model DT4T1, l'arbre fins a profunditat 3 és identic a l'arbre de profunditat 3 (Figura 6.7). Però, a conseqüència d'afegir un nivell més, l'arbre generalitza que si l'estudiant no aprova l'assignatura (nota superior a 4.9), acaba tornant a matriculant-la.

– Finalment, a la Figura 6.9, amb el model DT5T1, es pot observar un nivell significatiu de soroll en l'arbre, enfontsant-se en detalls com, per exemple, tenir en compte la nota en assignatures de 4 quadrimestres superiors. Aquest comportament pot indicar un fenomen de sobreajustament (*overfitting*), on l'arbre està capturant detalls del conjunt de dades d'entrenament que no generalitzen bé.

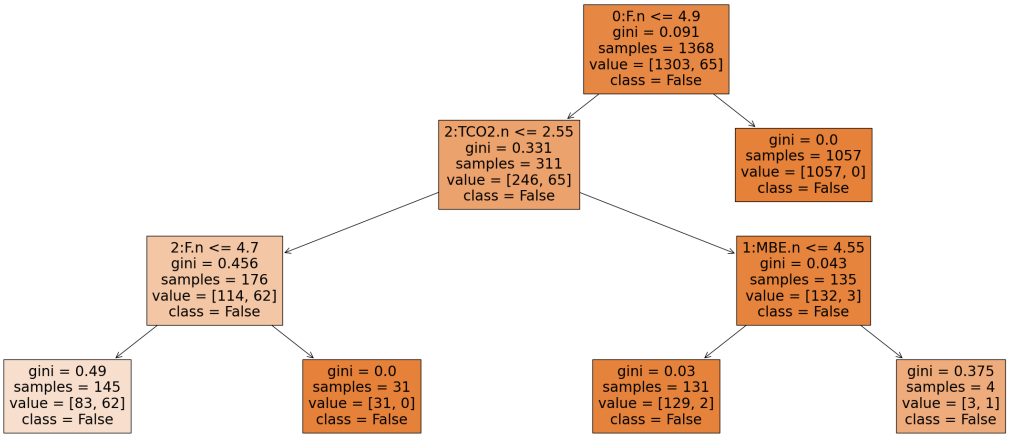

Figura 6.7: Arbre de F (ID: DT5T1)

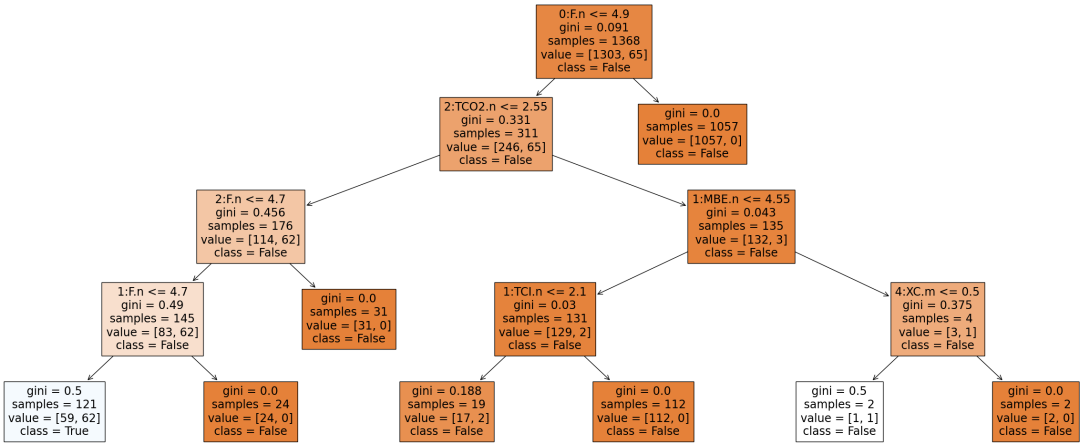

Figura 6.8: Arbre F (Id: Dt5t1)

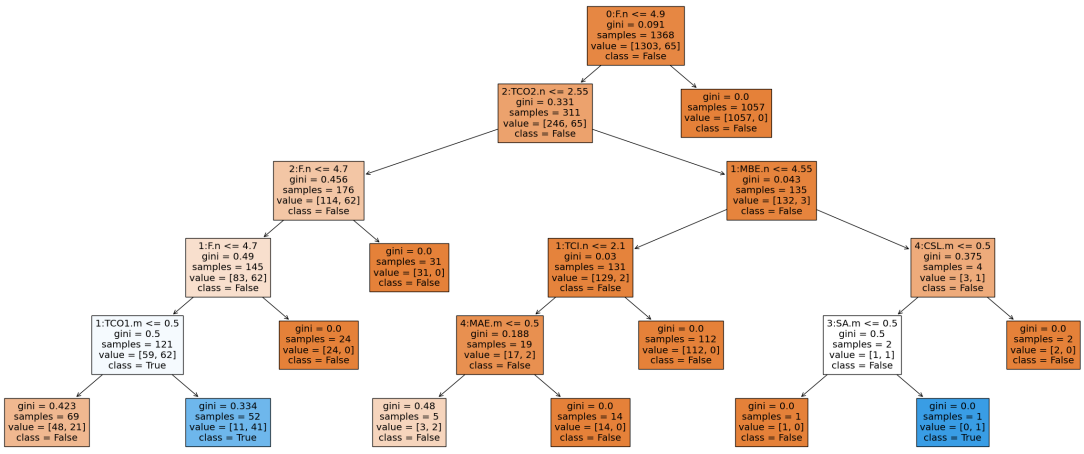

Figura 6.9: Arbre F (Id: Dt5t1)

6.1.3 Pruning dels arbres

Aplicant la tècnica *Minimal Cost-Complexity Pruning (MCCP)*, com s'ha explicat a la Secció 2.2.2 amb el model DT3T1 sobre l'assignatura de PBN, s'obtenen els resultats que es mostren a la Figura 6.10. L'objectiu és evitar l'*overfitting*, és a dir, que el model memoritzi les dades d'entrenament. Per aconseguir això, s'ha utilitzat un valor de α superior a 0.075, el que ha permès reduir l'efecte de l'*overfitting* mantenint al mateix temps un nombre raonable de nodes en el model. A la Figura 6.13 es mostren dos possibles arbres obtinguts per a dos valors diferents de α.

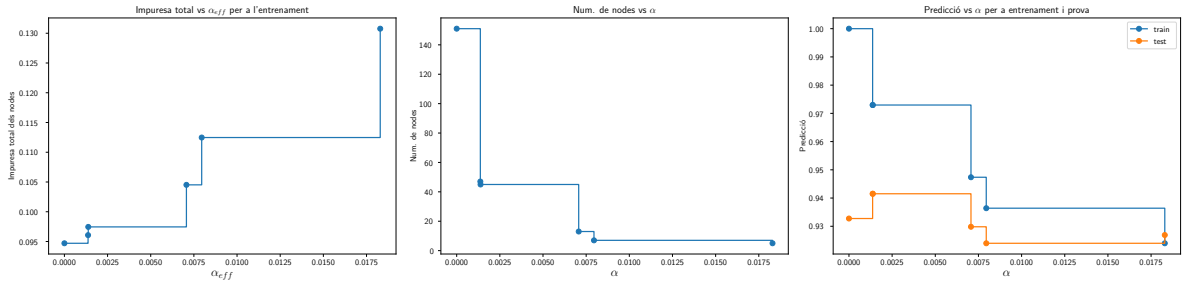

Figura 6.10: *Post-prunning* aplicat al model de PBN (ID: DT3T1)

A la Taula 6.3 s'observen els diferents arbres generats. La primera fila correspon a l'arbre inicial sense aplicar pruning ($\alpha = 0$). Observem que l'arbre amb $\alpha = 0.2$ millora els resultats en les dades de test respecte a l'arbre inicial. A la Figura 6.13, es veu que l'arbre amb $\alpha = 0.01$ té una partició addicional en el node amb la regla `1:TCI.n <= 4.7` amb una impuritat (*Gini Index*) del 0.352. No obstant això, als nodes terminals de la branca esquerra, la impuritat incrementa fins al 0.444, generant una representació poc precisa de les mostres (i a més, escasses). Així, s'opta per mantenir l'arbre amb $\alpha = 0.02$, on es retalla el node previ, evitant la bifurcació prèvia i optimitzant la impuritat. Cal destacar que amb aquesta opció s'aconsegueix una representació més ajustada a les dades de test.

`Arbre`	`Recall`	`Precisió`	`F1-score`
$\alpha = 0$ (Inicial)	0.7258	0.8333	0.7759
$\alpha = 0.01$	0.7258 =	0.8333 =	0.7759 =
$\alpha = 0.02$	0.8548 ↑	0.7681 ↓	0.8092 ↑

Taula 6.3: Avaluació dels arbres generats de PBN (ID: DT3T1)

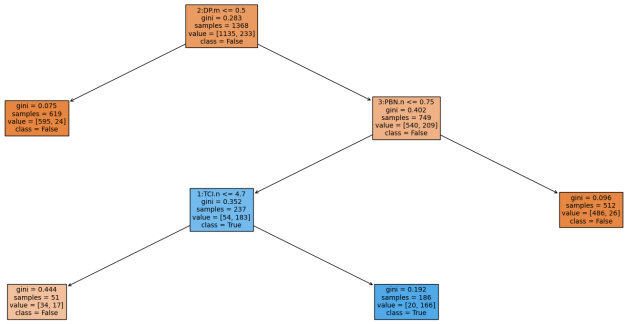

Figura 6.11: Pruning amb $\alpha = 0.01$ (Id: Dt3t1)

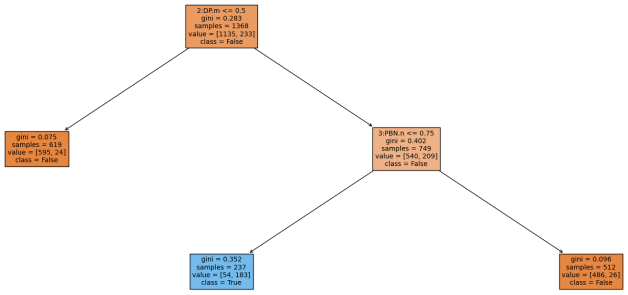

Figura 6.12: Prunning amb $\alpha = 0.02$

Figura 6.13: Arbres resultants (PBN) aplicant *Post-prunning* (Id: Dt3t1)

6.1.4 Transformació de les dades

En aquesta secció es valorarà l'impacte dels resultats en funció de la transformació de les dades definides a la Secció 3.2. En particular, les dades de entrenament de la primera transformació tenen una dimensió de 1368 files i 900 columnes i les de la segona transformació de 1368 files i 135 columnes.

Identificador (Id)	*Model*	*Profunditat*	*Transformació*	*Motxilla*
DT3T1	ARBRE	3	1	No
DT3T2	ARBRE	3	2	No
DT4T1	ARBRE	4	1	No
DT4T2	ARBRE	4	2	No

Taula 6.4: Experiments (1)

A partir dels models definits a la Taula 6.4 s'obtenen els resultats de la Figura 6.14. Concretament, s'observa que amb mateixa profunditat variant únicament la transformació de les dades que el temps d'entrenament del models DT3T2 i DT4T2 és significativament menor que el dels models DT3T1 i DT4T1. En concret, la primera tranformació disposa de bastantes més columnes, en altres paraules, més *features* que la segona transformació, obligant a l'arbre a analitzar-ne més per trobar el millor punt de tall en cada partició de l'arbre.

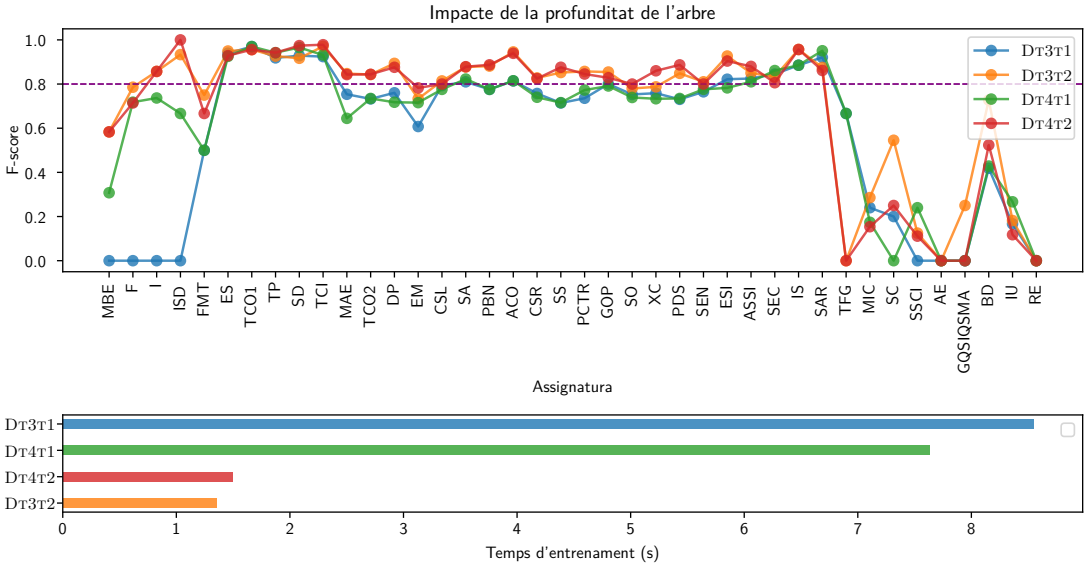

Figura 6.14: Avaluació dels experiments de la Taula 6.4

Amb el model DT3T2, s'obtenen els resultats de la Figura 6.15, on s'observa una millora respecte al model DT3T1 de la Figura 6.1. Per exemple, en l'assignatura de *Senyals i Sistemes* (SS), anteriorment no s'obtenia una avaluació del tot precisa utilitzant la primera transformació de les dades. No obstant això, mitjançant la segona transformació s'obtenen prediccions significativament més precises.

A la Figura 6.18, s'observa que els arbres són diferents, però presenten algunes similituds en les decisions basades en els resultats de certes assignatures. A la Figura 6.19, es mostra com l'arbre amb la segona transformació presenta valors d'impuresa (*Gini Index*) més baixos en les particions dels nodes. Això indica que l'arbre és capaç de representar millor el conjunt de mostres.

Adicionalment, es pot observar que amb la segona transformació, per matricular-se a SS, es parteix del node principal i es considera si l'estudiant s'ha matriculat a CSL. D'altra banda, amb la primera transformació, es fixa en *Tecnologies Complementàries 2* (TCO2). Això sembla concordar amb la percepció tant dels estudiants com dels professors, ja que els coneixements adquirits a CSL solen ser rellevants per a l'assignatura de SS.

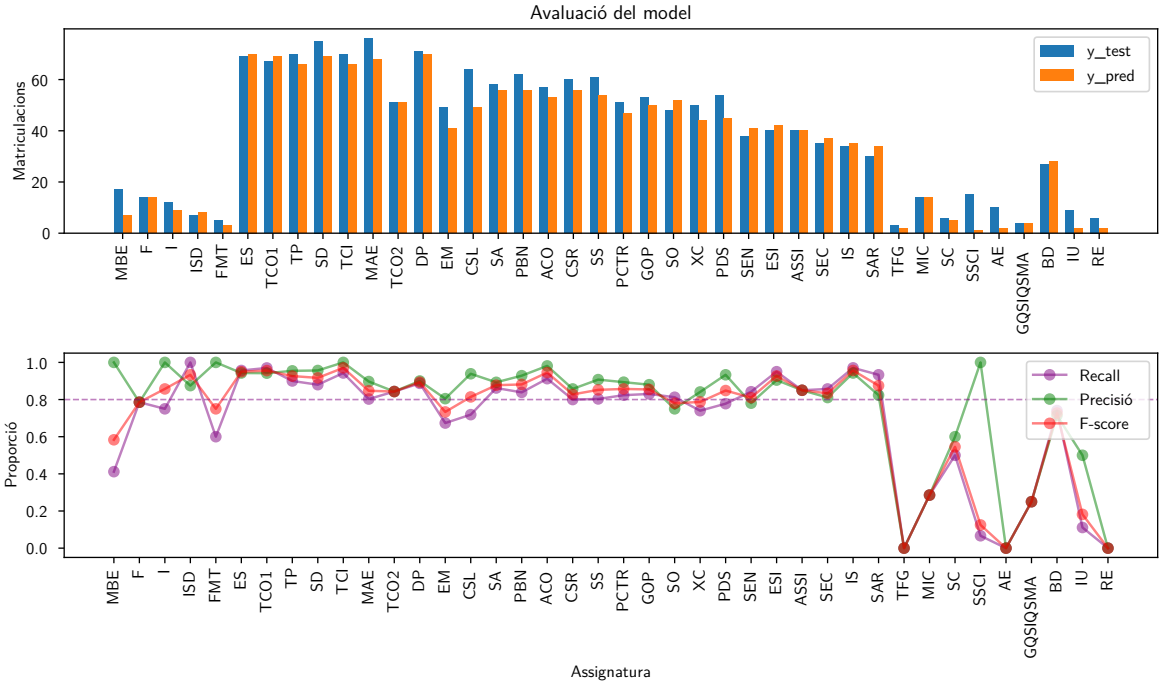

Figura 6.15: Avaluació dels models de les assignatures (ID: DT3T2)

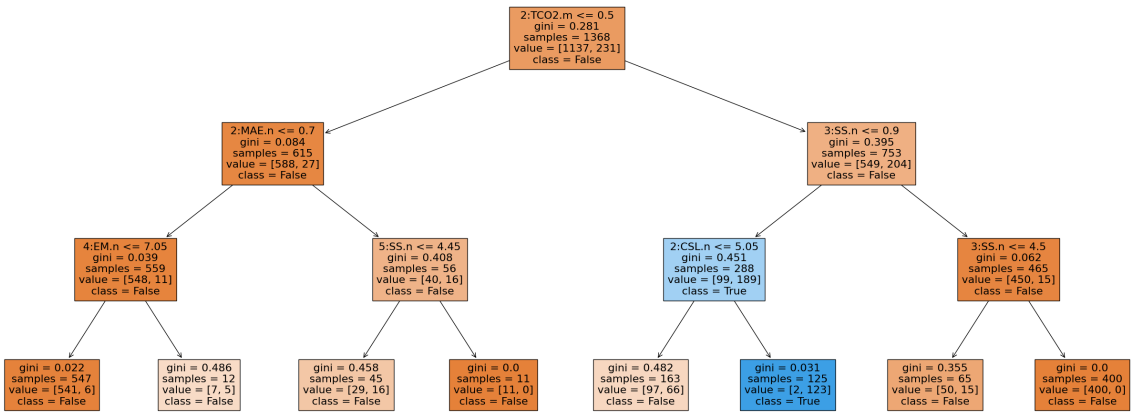

Figura 6.16: Arbre SS (Id: Dt3t1)

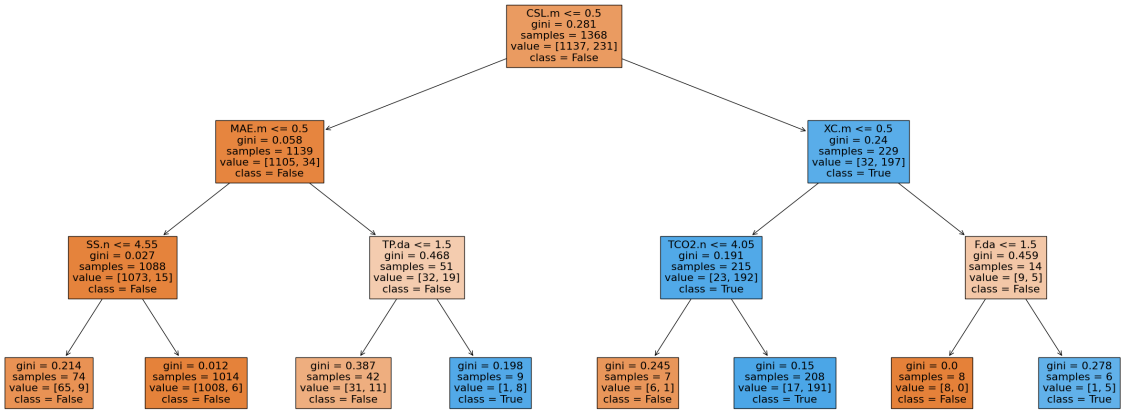

Figura 6.17: Arbre SS (Id: Dt3t2)

Figura 6.18: Comparació dels models de l'assignatura de *Senyals i Sistemes* (SS)

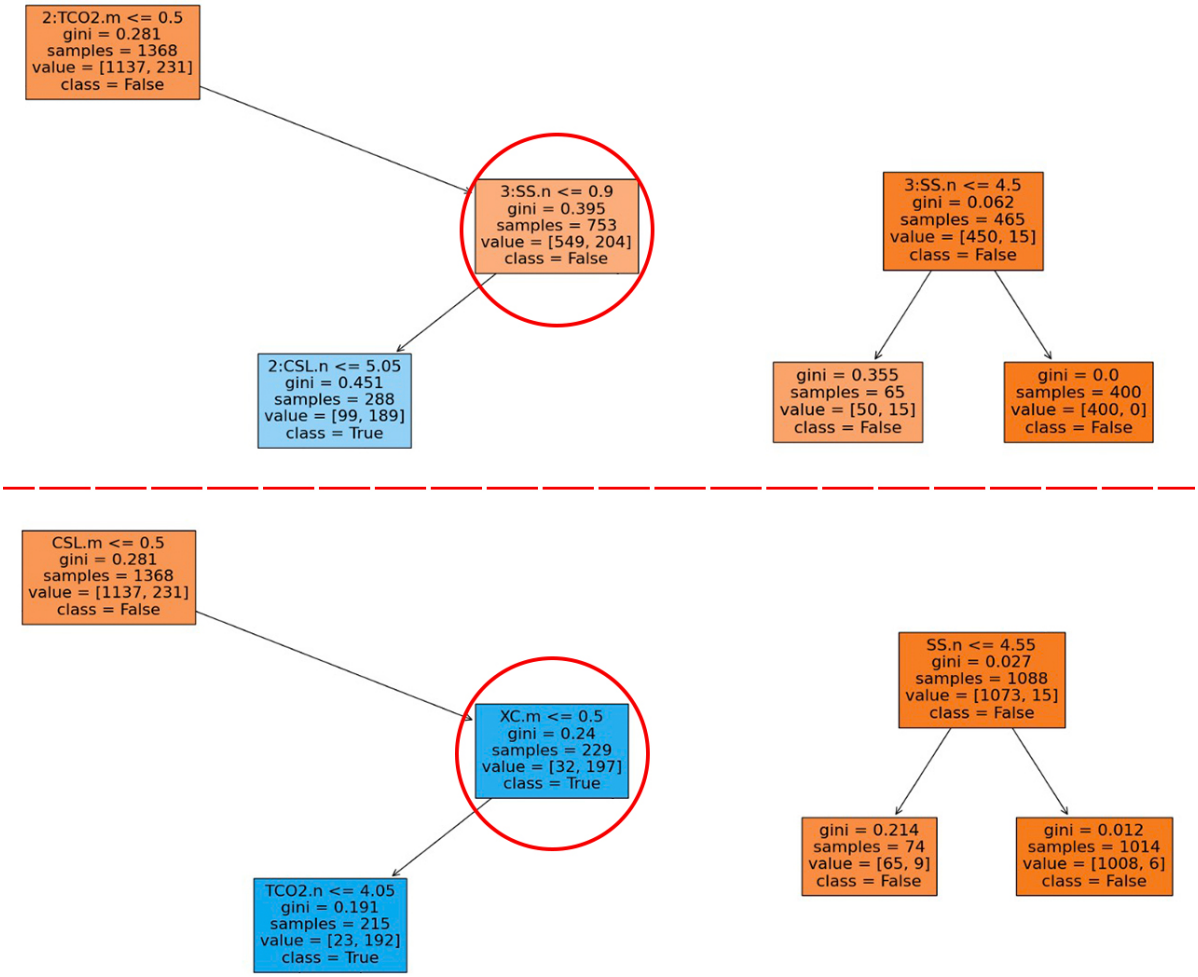

Figura 6.19: Diferències entre DT3T1 i DT3T2 de l'assignatura de SS[2]

També s'observa que per les assignatures del primer quadrimestre, el model DT3T2 amb una profunditat de 3 nodes és capaç d'obtenir les regles implícites, com s'ha comentat prèviament a la Secció 6.1.2. Això significa que el model és capaç de capturar els comportaments dels estudiants repetidors en aquestes assignatures.

Per exemple, a la Figura 6.22, el model DT3T1 de l'assignatura de *Física* (F) només és capaç d'obtenir la regla que diu que si un estudiant aprova la matèria, no la torna a matricular. No obstant això, amb el model DT3T2, s'obté aquesta mateixa regla, però també s'obté la regla implícita dels estudiants repetidors. En concret, si un estudiant suspèn F en el primer quadrimestre i en algun moment ha matriculat TCO1 (Q2), tornarà a matricular F. Aquest mateix comportament es veu replicat en l'assignatura d'*Informàtica* (I), com es mostra a la Figura 6.25.

En resum, es pot observar que amb la segona transformació de les dades s'obtenen millors patrons i decisions de matriculació, ja que el model DT3T2 és capaç de capturar les regles

[2]La part superior representa el model DT3T1 i la inferior el model DT3T2

implícites dels estudiants repetidors en les assignatures del primer quadrimestre.

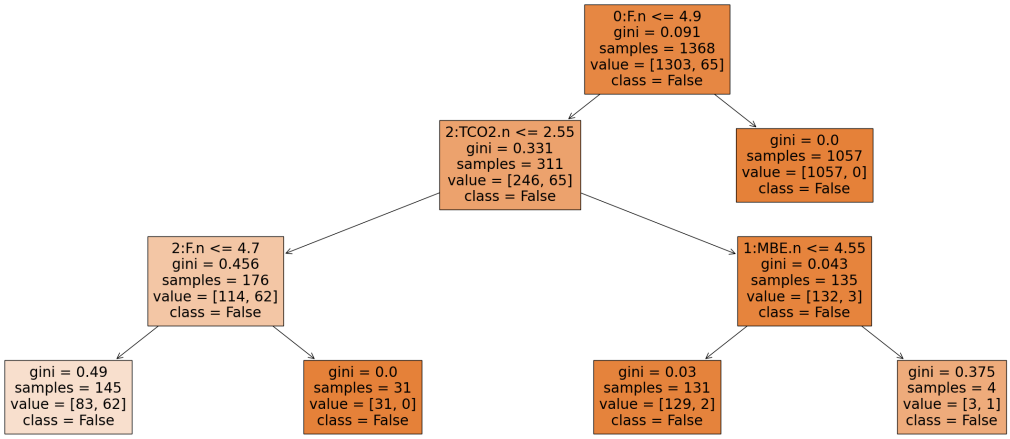

Figura 6.20: Arbre F (Id: Dt3t1)

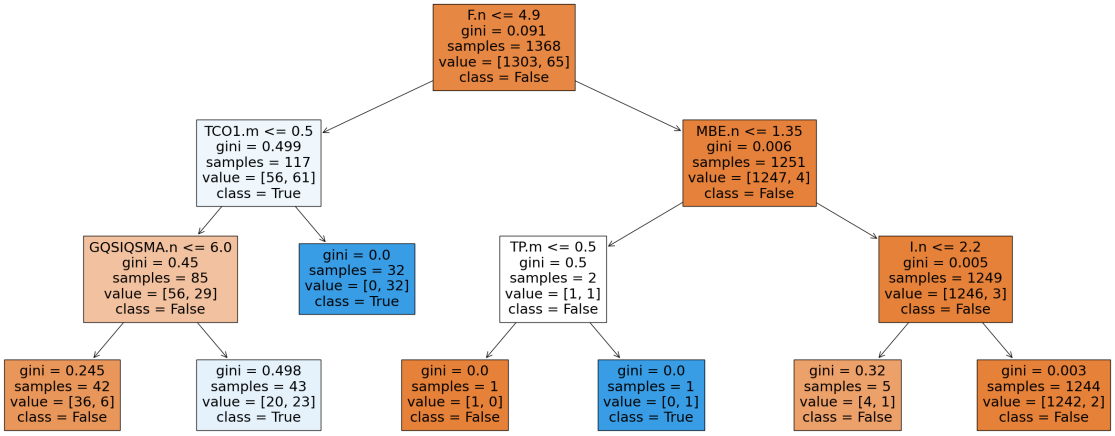

Figura 6.21: Arbre F (Id: Dt3t2)

Figura 6.22: Comparació dels models de l'assignatura de *Física* (F)

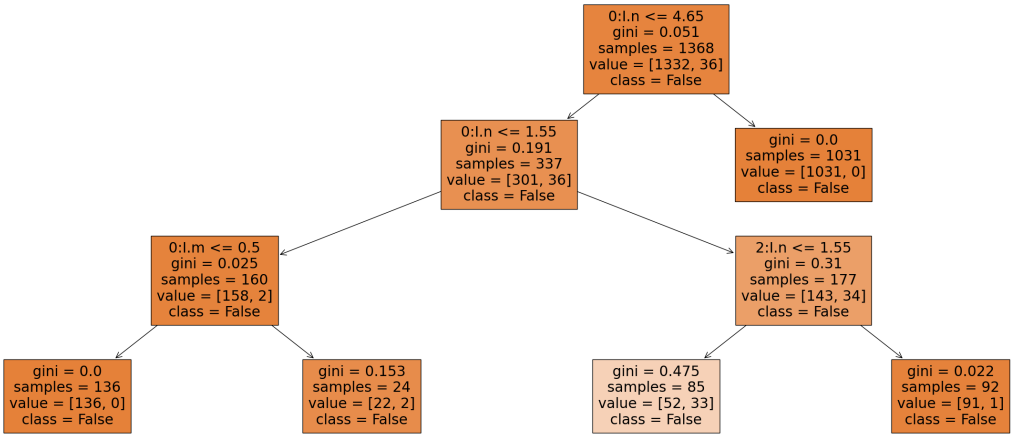

Figura 6.23: Arbre I (Id: Dt3t1)

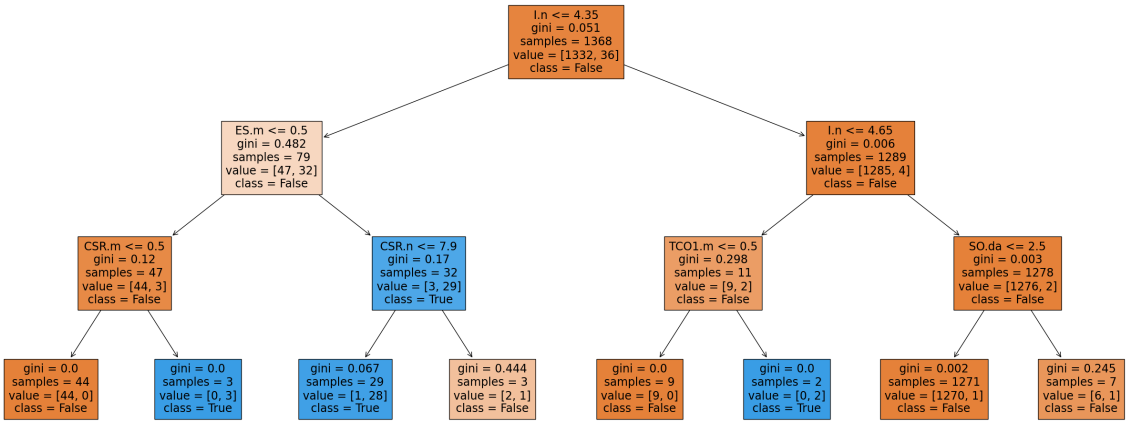

Figura 6.24: Arbre I (Id: Dt3t2)

Figura 6.25: Comparació dels models de l'assignatura de *Informàtica* (I)

6.1.5 Motxilla de l'estudiant

En aquest experiment, es consideraran les dades de la «motxilla de l'estudiant» com s'ha descrit a la Secció 3.1. L'objectiu és analitzar el possible impacte d'aquestes dades en la presa de decisions durant la fase de matriculació d'assignatures, i observar possibles diferències relacionades amb la nota d'accés, via d'accés, disposició de beca i altres dades descrites prèviament. Per aquest motiu, es defineix la següent taula d'experimentació:

Identificador (Id)	*Model*	*Profunditat*	*Transformació*	*Motxilla*
DT3T1	ARBRE	4	1	No
DT3T2	ARBRE	4	2	No
DT3T1M	ARBRE	4	1	Sí
DT3T2M	ARBRE	4	2	Sí

Taula 6.5: Experiments (2)

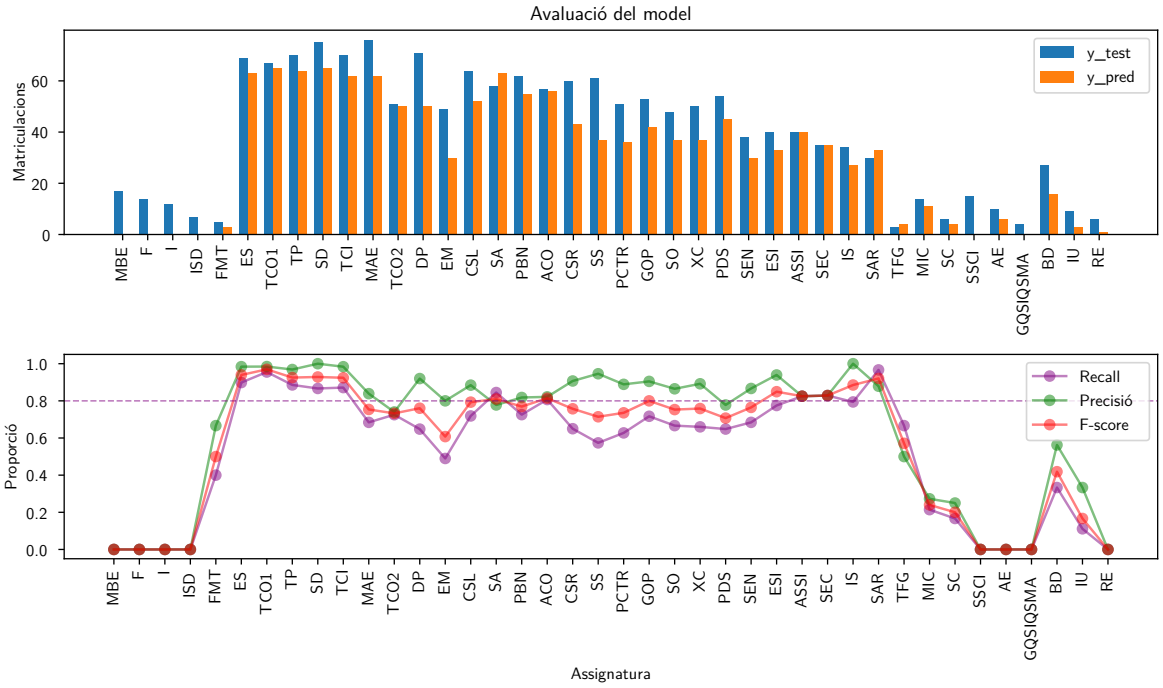

Figura 6.26: Avaluació dels models de les assignatures afegint la *«motxilla dels estudiants»*)

Doncs, a la Figura 6.26 es mostra l'avaluació del model DT3T1M incorporant la «*motxilla dels estudiants*». En primer lloc, s'observa que l'avaluació dels models no millora significativament, és a dir, l'avaluació de cada model és pràcticament igual en la majoria de les assignatures. En concret, comparant cada model segons la transformació de les dades, s'observa que la majoria dels models de les assignatures són iguals, excepte les assignatures representades en les Taules 6.6 i 6.7. Llavors, investigant els arbres d'aquestes assignatures, no apareixen atributs sobre la motxilla de l'estudiant excepte els models de PBN (Figura 6.27) i PDS (Figura

6.28) incorporen regles en funció de la beca. Malgrat això, l'avaluació general dels models és pràcticament la mateixa que la dels arbres previs, sense incloure cap mena d'informació adicional sobre els estudiants.

Assignatura	Recall[3]		Precisió[3]		F1-score[3]	
TP	0.0000	0.0000	0.9541	0.9691	0.9190	0.9250
PBN	0.0000	0.0000	0.8330	0.818	0.7759	0.7690
PDS	0.7041	0.648	0.7600	0.7780	0.7310	0.7071
ESI	0.7500	0.7750	0.9090	0.9391	0.8221	0.8491
SEC	0.8570	0.8291	0.8330	0.8291	0.8450	0.8291
TFG	0.0000	0.0000	0.6670	0.5000	0.6670	0.5711

Taula 6.6: Diferències entre els models Dt3t1 i Dt3t1m

Assignatura	Recall[3]		Precisió[3]		F1-score[3]	
I	0.0000	0.0000	1.0000	0.8180	0.8570	0.7829
ISD	0.0000	0.0000	0.8750	1.0000	0.9329	1.0000
TCO1	0.0000	0.0000	0.9420	0.9560	0.9560	0.9630
SD	0.8800	0.8930	0.0000	0.0000	0.9170	0.9240
EM	0.6729	0.8570	0.8050	0.8240	0.7330	0.8400
ASSI	0.8500	0.8000	0.8500	0.8420	0.8500	0.8210
SEC	0.0000	0.0000	0.8109	0.7690	0.8330	0.8109
SC	0.5000	0.332996	0.600	0.400	0.545	0.3640

Taula 6.7: Diferències entre els models Dt3t2 i Dt3t2m

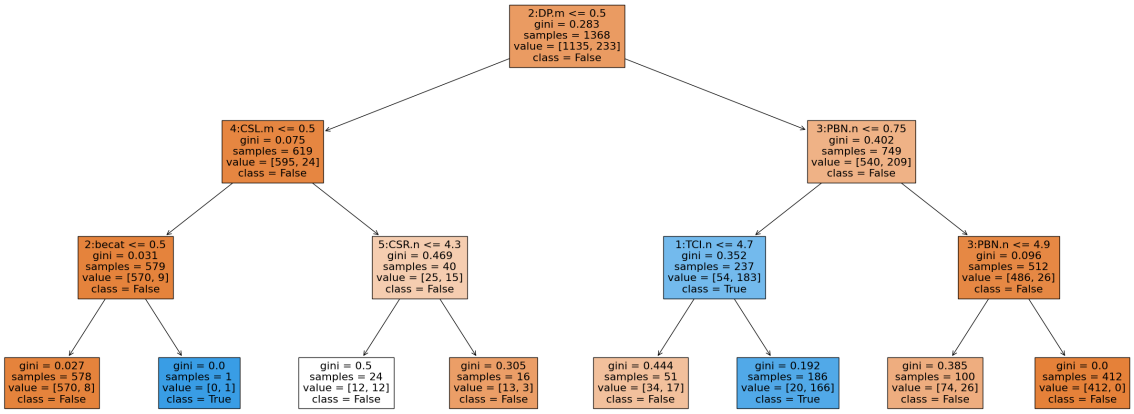

Figura 6.27: Arbre PBN (Id: Dt3t1m)

[3]La columna dreta representa l'avaluació del model afegint la «*motxilla de l'estudiant*».

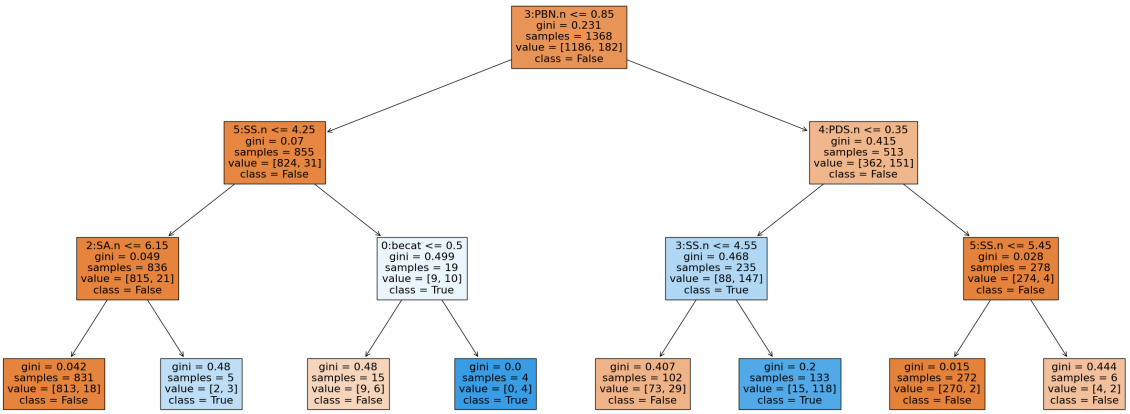

Figura 6.28: Arbre PBN (ID: DT3T1M)

6.1.6 Conclusions

Una cop realitzada aquesta avaluació inicial, es poden extreure algunes conclusions importants. S'ha observat que l'ús d'arbres de decisió amb una profunditat de 5 nodes pot resultar en problemes de *overfitting*, ja que l'arbre es torna massa complex i es sobreajusta al les dades d'entrenament fixant-se en resultats d'assignatures de quadrimestres superiors.

Adicionalment, s'ha demostrat que l'aplicació de post-pruning pot ser una opció viable per reduir l'efecte del *overfitting* aquest efecte, tot i que requereix una supervisió detallada per part d'una persona experta per parametritzar adequadament l'arbre per a cada assignatura específica, la qual cosa implica un treball important i en certa manera «excessiu». Una altra opció seria automatitzar en certa manera la fase de post-pruning, de manera iterativa provar diferents valors de α_{eff} per avaluar el model i trobar el valor òptim que millori els resultats. D'aquesta manera, també es pot reduir l'efecte del *overfitting*, però, es perdria tota la fase de interpretació de cada arbre en funció de la parametrització en particular.

Finalment, també s'ha observat que la incorporació de dades addicionals de l'estudiant, com ara la nota d'accés, la via d'accés, l'assignació de beca, entre altres, no altera el patró de matriculació.

6.2 Boscos d'arbres aleatoris

6.2.1 Parametrització dels boscos

Al treballar amb boscos d'arbres aleatoris, es disposa d'una gran quantitat d'experiments possibles ja que aquests són altament parametritzables. Concretament, es poden ajustar diversos paràmetres que afecten l'estratègia de *Bootstraping* de les mostres i les característiques (*features*) del conjunt de dades original. A continuació es mostren alguns dels paràmetres rellevants que permet *Scikit-learn*:

- `n_estimators`: Especifica el nombre d'arbres totals que es construeixen en el bosc. Cal recordar que el temps d'entrenament del bosc augmenta proporcionalment al nombre d'arbres.
- `max_depth`: Limita la profunditat màxima de cada arbre en el bosc.
- `random_state`: És un paràmetre extremadament útil si es volen realitzar experiments i obtenir experiments consistents. En altres paraules, durant la recerca a vegades interessa reproduïr un experiment i obtenir els mateixos resultats.
- `max_features`: Determina el màxim nombre de *features* que es seleccionen de manera aleatòria en cada partició de cada arbre.
  - Si és un enter, es consideren `max_features` característiques en cada partició de l'arbre.
  - Si és un valor decimal en l'interval $(0.0, 1.0]$, `max_features` representa una fracció del nombre total de *features* i es seleccionen `max(1, int(max_features * n_features))` *features* en cada partició de l'arbre.
  - Si és "auto", es seleccionen `max_features=sqrt(n_features)`.
  - Si és "sqrt", es seleccionen `max_features=sqrt(n_features)`.
  - Si és "log2", es seleccionen `max_features=log2(n_features)`.
  - Si és None, es seleccionen `max_features=n_features`.
- `max_samples`: Determina el màxim nombre de mostres utilitzades per a entrenar cada arbre individual.
  - Si és un enter, es consideren `max_samples` mostres per a entrenar cada arbre.
  - Si és un valor decimal en l'interval $(0.0, 1.0]$, s'utilitzen `max(1, int(max_samples * total_samples))` mostres.
  - Si és None, llavors s'utilitzen totes les mostres del conjunt de dades.
- `bootstrap`: Indica si s'ha d'aplicar *Bootstraping*. Si no s'estableix, totes les mostres i *features* del conjunt de dades s'utilitzen per a entrenar cada arbre.
- `oob_score`: Estableix si s'ha de calcular la taxa d'error *Out-of-Bag (OOB)* estimada durant l'entrenament del model. Només es calcula si s'aplica *Bootstraping*.

6.2.2 Primeres probes

Com s'ha mencionat prèviament, els boscos d'arbres són extremadament parametritzables donant la possibilitat de definir varis experiments possibles. Doncs, es defineixen els experiments descrits en la següent taula:

Identificador	*Model*	*Profunditat*	*Transformació*	*Estimadors*	*Features*	*Mostres*
Dt3	Arbre	3	1	1	`n_features`	`n_samples`
Dt4	Arbre	4	1	1	`n_features`	`n_samples`
RftLog2	Bosc	4	1	20	`log_2`	`n_samples`
RftSqrt	Bosc	4	1	20	`sqrt`	`n_samples`
Rft0.5	Bosc	4	1	20	`0.5`	`n_samples`
RftAll	Bosc	4	1	20	`n_features`	`n_samples`

Taula 6.8: Experiments (1)

A la Figura 6.29 es mostren els resultats obtinguts. En primer lloc, es pot observar una situació bastant curiosa, l'experiment Dt4 obté resultats millors que els experiments RftLog2 i RftSqrt, on RftLog2 obté resultats pràcticament nuls. Podria semblar que un bosc d'arbres amb 20 estimadors hauria d'obtenir resultats similars o fins i tot millors. No obstant això, en aquests dos últims experiments, en cada divisió de l'arbre, s'estan seleccionant un nombre reduït de *features*, específicament $log_2(n_{features})$ per a RftLog2 i $\sqrt{n_{features}}$ per a RftSqrt.

Es pot apreciar en la Figura 6.30 que la selecció de *features*, en concret, amb `sqrt` i `log2` no són bones opcions per a un conjunt de dades gran amb un nombre considerable de *features*, ja que aquestes opcions estan pensades per a conjunts de dades amb una quantitat significativament menor.

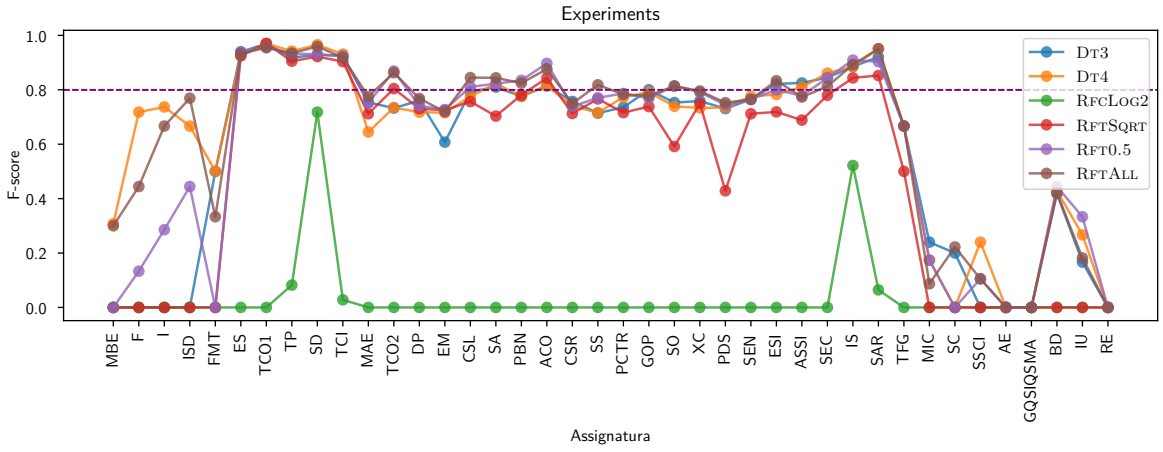

Figura 6.29: Avaluació dels experiments de la Taula 6.8

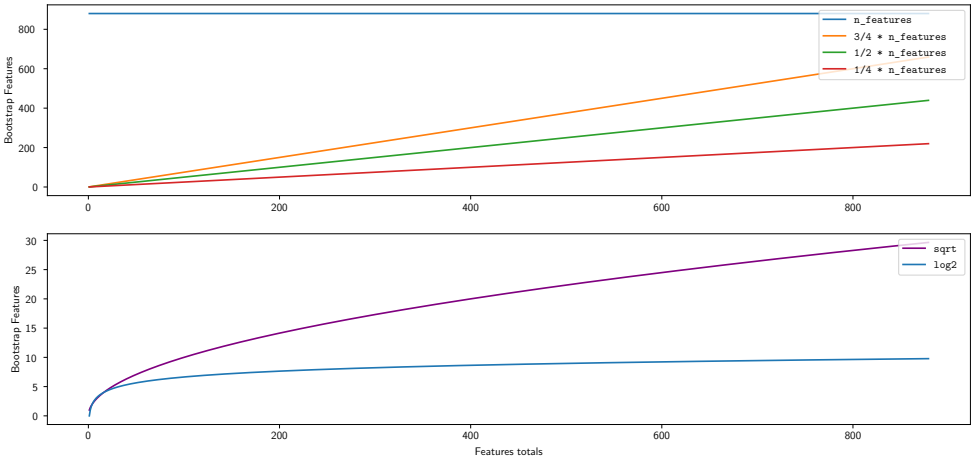

Figura 6.30: Quantitat de *features* seleccionades

Arribats a aquest punt, es pot concloure que la selecció de `log2` no és un paràmetre ideal en aquest cas. No obstant, val la pena explorar com afecta la selecció de `sqrt` en funció del nombre d'estimadors que s'estableix. Amb aquest objectiu, es plantegen els següents experiments per comparar l'impacte:

Identificador	*Model*	*Profunditat*	*Transformació*	*Estimadors*	*Features*	*Mostres*
RftSqrt	Bosc	4	1	20	`sqrt`	`n_samples`
Rft100Sqrt	Bosc	4	1	100	`sqrt`	`n_samples`
Rft200Sqrt	Bosc	4	1	200	`sqrt`	`n_samples`
Rft500Sqrt	Bosc	4	1	500	`sqrt`	`n_samples`

Taula 6.9: Experiments (2)

A la Figura 6.31 es mostra l'avaluació d'aquests experiments. Es pot observar que, en la majoria dels casos, l'experiment RftSqrt amb 20 estimadors obté els millors resultats. Això es pot explicar pel fet que, la resta d'experiments s'utilitzen més de 100 estimadors i s'aplica una selecció reduïda de *features*. Aquest fet pot introduir una major variabilitat en el bosc d'arbres, sense permetre que els arbres puguin generalitzar el comportament de les dades de manera òptima. Com a conseqüència, quan els boscos emeten les prediccions durant la votació, els resultats poden ser més dispars i menys coherents, afectant negativament la precisió i la fiabilitat de les prediccions finals del bosc d'arbres aleatoris.

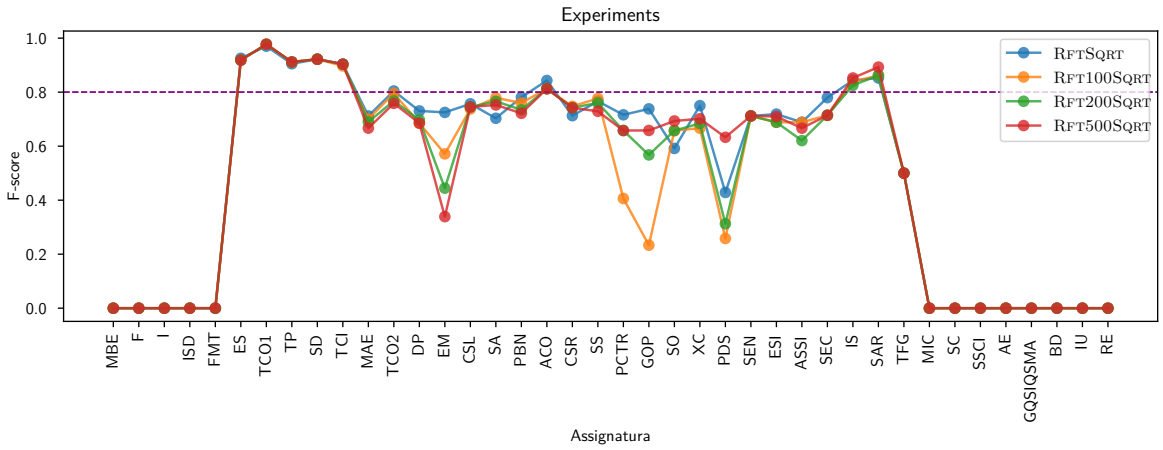

Figura 6.31: Avaluació dels experiments de la Taula 6.9

6.2.3 Resultats definitius

Com s'ha vist anteriorment, és crucial considerar la selecció adequada dels paràmetres per aconseguir una configuració òptima per als boscos d'arbres. Per a això, és important trobar un equilibri entre la profunditat dels arbres del bosc, el nombre d'estimadors i una selecció coherent de l'estratègia de *Bagging*. En aquest cas, es proposen les següents configuracions:

Identificador	*Model*	*Profunditat*	*Transformació*	*Estimadors*	*Features*	*Mostres*
RftConf1	Bosc	4	1	15	0.5	0.8
RftConf2	Bosc	4	2	15	0.5	0.8

Taula 6.10: Experiments (3)

Doncs, a la Figura 6.32 s'observa l'avaluació dels experiments de la Taula 6.10. En resum, el model RftConf2 amb la segona transformació de dades obté millors resultats (Figura 6.33), com s'ha observat també amb els arbres de decisió a la Secció 6.1.4. Així doncs, en aquest cas, utilitzant boscos d'arbres s'obtenen arbres diferents. De fet, com es mostra a la Figura 6.36, es pot veure clarament que els arbres formats en el bosc de l'assignatura de *Programació a Baix Nivell* presenten diferències en el recorregut de la presa de decisions, però també comparteixen similituds en certes característiques, com ara la importància de la matrícula de *Dispositius Programables* (DP) i *Tecnologies Complementàries 2* (TCO2), entre altres.

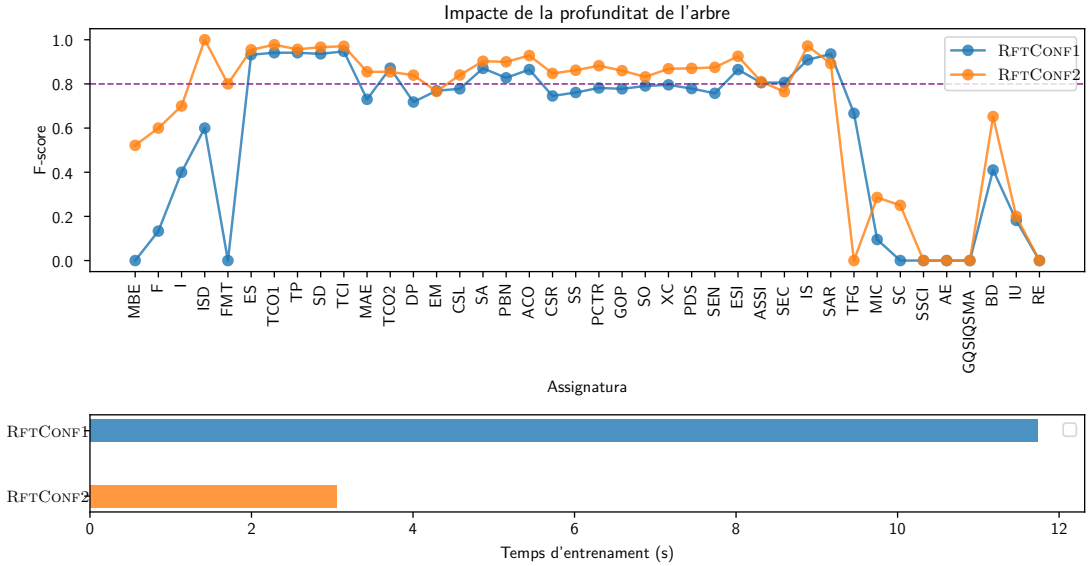

Figura 6.32: Avaluació dels experiments de la Taula 6.10

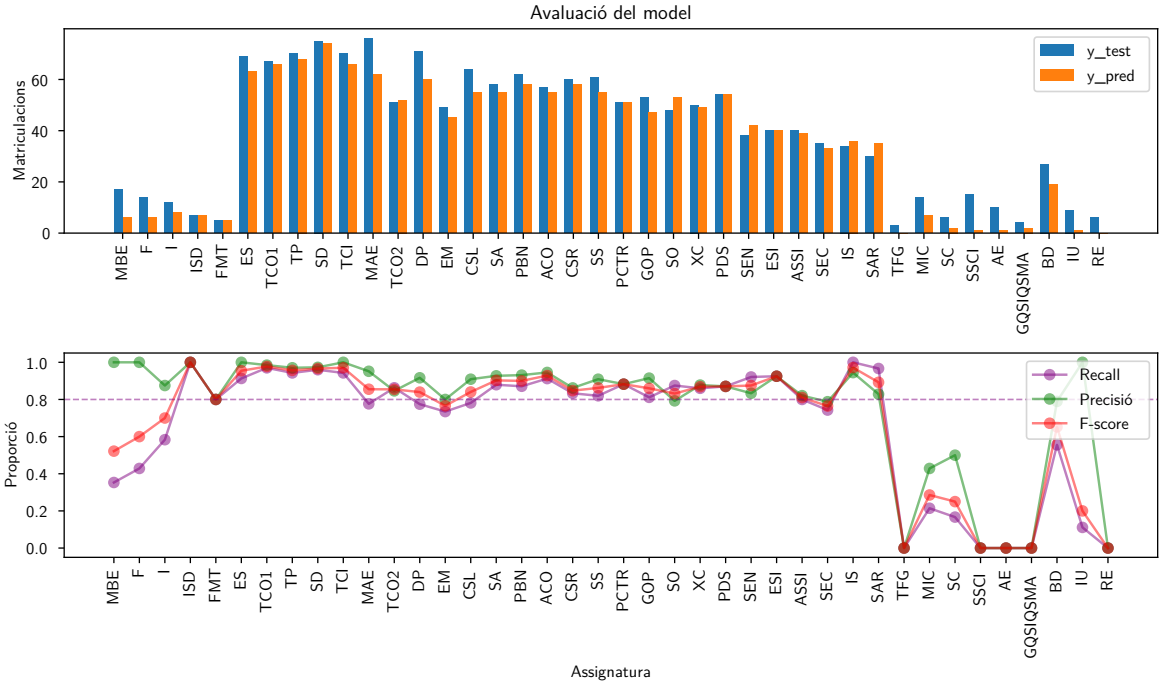

Figura 6.33: Avaluació dels models de les assignatures (Id: RftConf2)

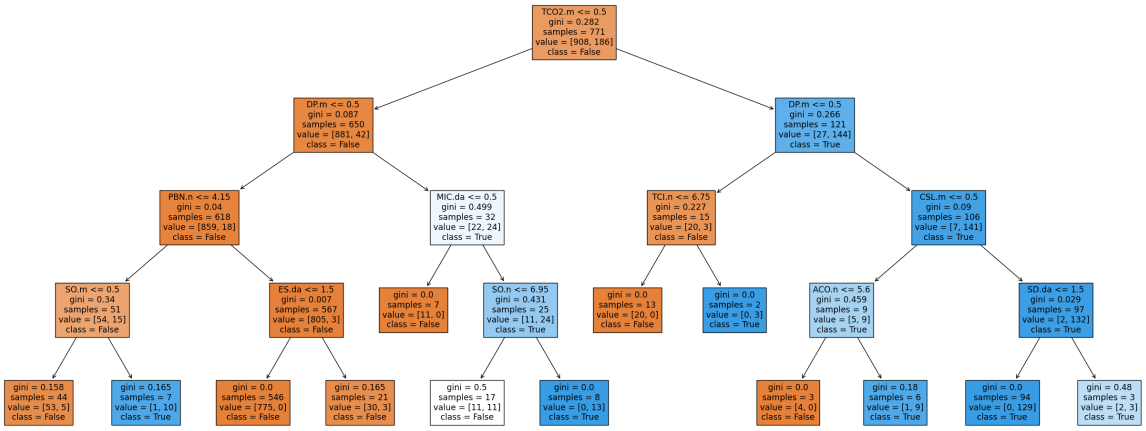

Figura 6.34: Primer arbre de PBN (Id: RftConf2)

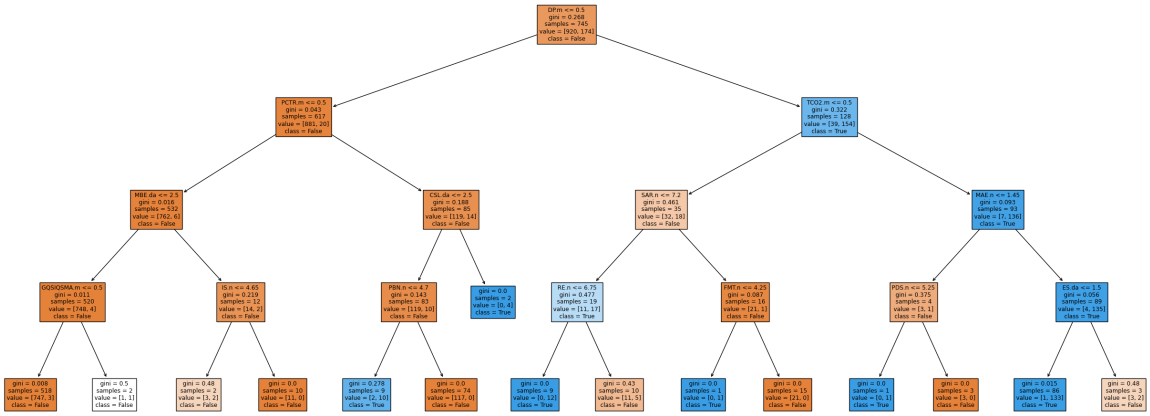

Figura 6.35: Quart arbre de PBN (Id: RftConf2)

Figura 6.36: Arbres formats en el bosc de l'assignatura de *Programació a Baix Nivell* (PBN)

6.3 Obtenció del millor bosc

S'ha observat que els boscos d'arbres són altament parametritzables. A la Secció 4.3.1, s'ha mencionat la tècnica de *Bootstraping*, on el bosc d'arbres pot proporcionar una primera avaluació general sense necessitat de dades de prova. Això es pot aconseguir utilitzant les mostres *OOB* que no són seleccionades per entrenar durant la selecció aleatòria de les característiques. A través d'aquesta tècnica, es pot obtenir una avaluació prou precisa, com es mostra a la Figura 6.37, utilitzant les assignatures del quart quadrimestre i el model RftConf2, parametritzant únicament la quantitat d'estimadors (arbres) que formaran el bosc.

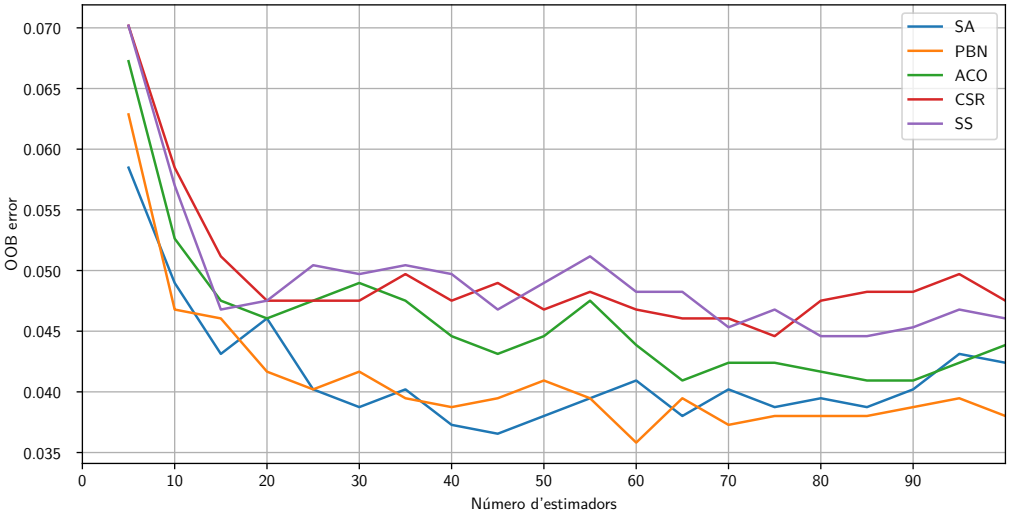

Figura 6.37: Obtenció del millor bosc d'arbres de les assignatures de Q4 (Id: RftConf2)

Llavors, a la Figura 6.38 es pot observar la quantitat d'arbres que conformen el bosc per a cada assignatura, els quals obtenen la menor taxa d'error utilitzant les mostres *OOB* (*Out-of-Bag samples*). Aquesta automatització és interessant i molt útil, ja que permet millorar la qualitat de les prediccions per a cada assignatura de manera eficient.

Finalment, a la Figura 6.39 s'observa l'avaluació del bosc d'arbres de cada assignatura amb els paràmetres ideals que minimitzen la taxa d'errors, conformant l'arbre resultant més òptim possible.

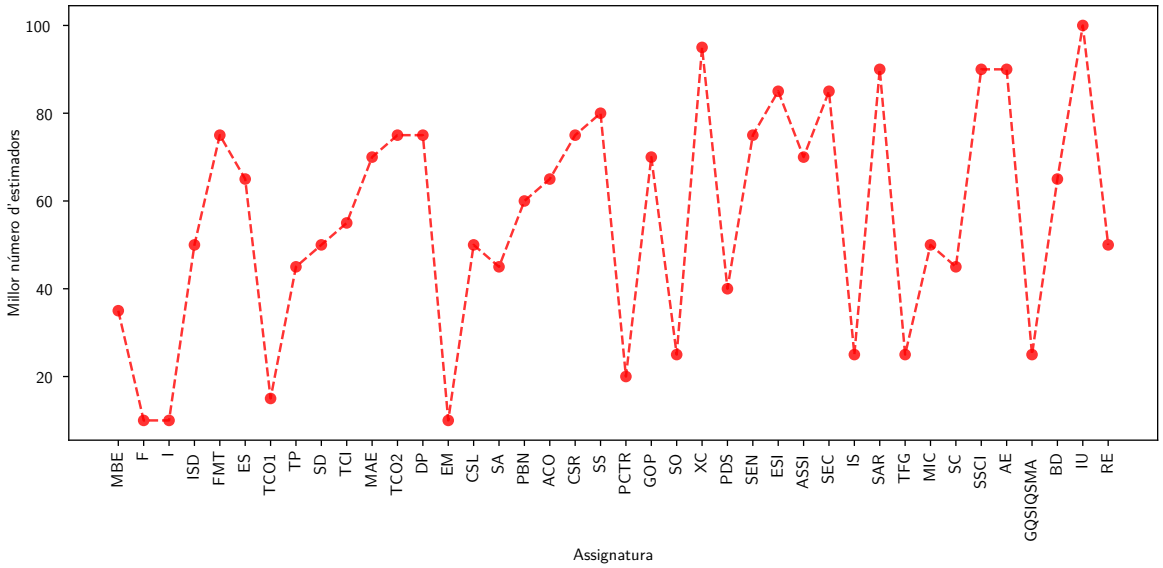

Figura 6.38: Obtenció del millor bosc d'arbres de cada assignatura (Id: RftConf2)

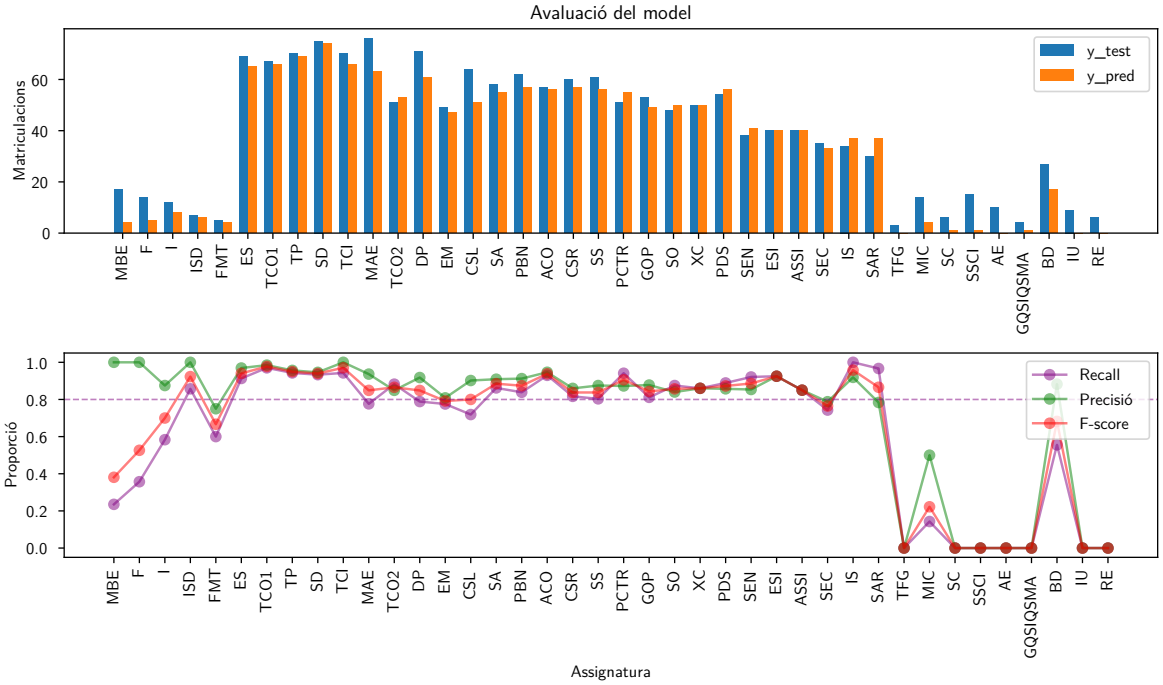

Figura 6.39: Avaluació del millor bosc d'arbres de cada assignatura (Id: RftConf2)

6.4 Importància de les assignatures

Arribats a aquest punt, s'ha demostrat que els arbres de decisió són capaços d'obtenir prediccions precises, i un avantatge d'aquests és la seva simplicitat i facilitat d'anàlisi en comparació amb els boscos d'arbres amb múltiples estimadors. No obstant això, els boscos d'arbres també ofereixen certs avantatges, com ara la capacitat de calcular la importància de les variables (*Feature Importance*) mitjançant el valor del decreixement mitjà de la impuresa (*Mean Decrease in Impurity*) obtingut dels arbres que conformen el bosc.

A partir del valor del MDI, és possible analitzar la importància de cada assignatura. En aquest sentit, en un pla d'estudis, cada assignatura hauria de tenir una certa importància per a l'avaluació contínua en base als coneixements adquirits en cadascuna d'elles. En la Figura 6.40, es mostra un mapa de calor que representa la importància de cada assignatura. S'observa un efecte molt interessant: les assignatures dels primers quadrimestres, especialment les del primer i segon quadrimestre, tenen una gran importància per a la continuïtat dels estudiants en la carrera. A més, també s'observen vàries assignatures que destaquen sobre les alres, assignatures com per exemple SS, DP, PBN, XC entre altres.

Finalment, aquesta observació sembla suggerir que els resultats obtinguts en les assignatures inicials poden tenir un impacte significatiu en la motivació i continuïtat dels estudiants en el pla d'estudis.

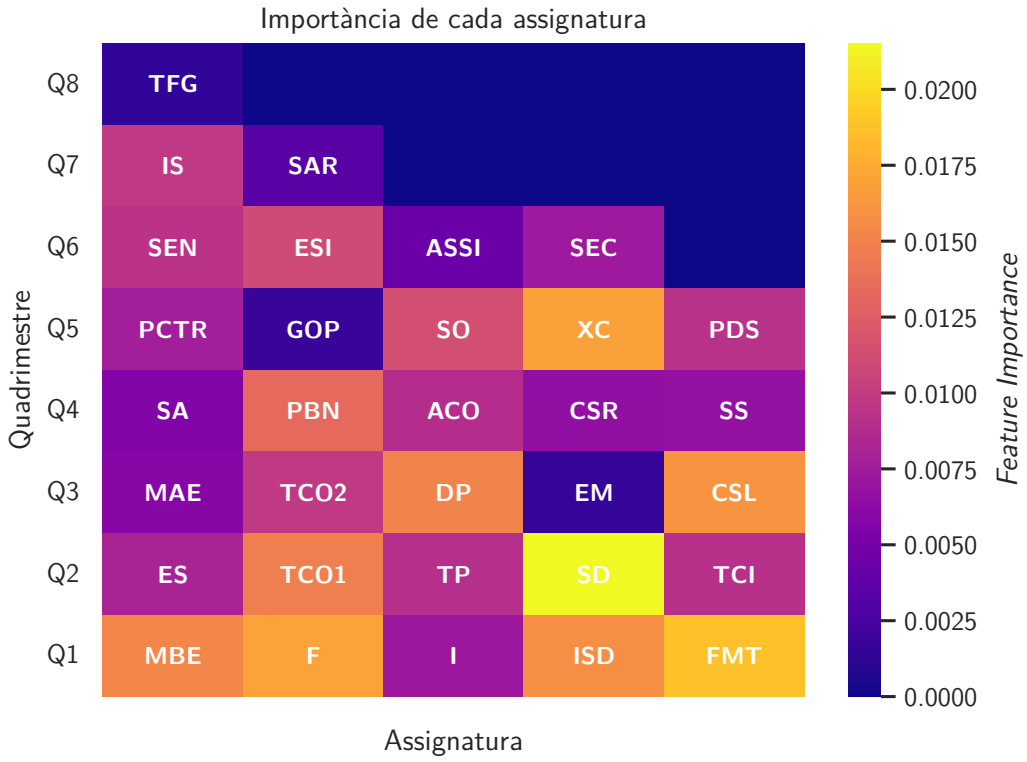

Figura 6.40: Importància de cada assignatura

7 Conclusions

En conclusió, s'ha vist que els arbres de decisió són un bon primer model inicial per a predir les matriculacions universitàries, amb resultats precisos en la majoria dels casos. No obstant això, s'ha observat que utilitzar arbres de decisió amb una profunditat de 5 nodes pot conduir a problemes de *overfitting*, ja que aquests arbres es tornen massa complexos i se centren en resultats d'assignatures de quadrimestres superiors. Això pot limitar la capacitat del model per a generalitzar i predir amb precisió casos nous.

Adicionalment, s'ha demostrat que l'aplicació de post-pruning pot ser una opció viable per reduir l'efecte del *overfitting*. No obstant això, aquesta tasca requereix una supervisió detallada per a parametritzar adequadament l'arbre per a cada assignatura específica. Això implica un treball important i, en certa manera, exhaustiu. Una altra opció seria automatitzar en certa manera la fase de post-pruning, provant iterativament diferents valors de α_{eff} per avaluar el model i trobar el valor òptim que millori els resultats.

També, s'ha observat que la incorporació de dades addicionals de l'estudiant, com ara la nota d'accés, la via d'accés, l'assignació de beca, entre altres, no altera el patró de matriculació. Això suggereix que aquesta informació té un impacte significatiu en la decisió dels estudiants a l'hora de matricular-se en les assignatures.

Pel que fa a l'obtenció dels models, a diferència dels arbres de decisió, que destaquen sobretot per la simplicitat i facilitat d'anàlis, s'ha observat que amb els boscos d'arbres es poden realitzar tots els experiments possibles. Així mateix, els boscos d'arbres ofereixen avantatges com ara la capacitat de calcular la importància de les característiques de les dades.

En resum, les conclusions suggereixen que les assignatures dels primers quadrimestres tenen una importància significativa per a la continuïtat dels estudiants en la carrera. A més, s'han identificat diverses assignatures que destaquen per sobre de les altres, indicant que poden tenir un impacte important en les decisions dels estudiants. Com a resultat final, els boscos d'arbres semblen ser la millor solució per ara, optimitzant cada bosc de forma automàtica trobant la quantitat d'estimadors ideals en cada assignatura.

Amb aquest treball, s'ha demostrat la utilitat dels arbres de decisió i els boscos d'arbres, així com la gran quantitat d'experiments i proves que permeten realitzar. Aquest fet obre la porta a futures investigacions, experiments i millores en els models de predicció.

Bibliografia

[Gér19] Aurélien Géron. «Hands-On Machine Learning with Scikit-Learn and TensorFlow». A: 2nd. 2019. Cap. 1, pàg. 2. ISBN: 978-1-492-03264-9.

[Lee17] Ceshine Lee. *Feature Importance Measures for Tree Models – Part I*. Medium. Consultat el 10 de maig de 2023. 2017. URL: https://medium.com/the-artificial-impostor/feature-importance-measures-for-tree-models-part-i-47f187c1a2c3.

[Wik23] Wikipedia. *Overfitting*. Wikipedia. Consultat el 3 de febrer de 2023, des de https://ca.wikipedia.org/wiki/Sobreajustament. Abr. de 2023.