Pràctica 1: Anàlisi i classificació del dataset del Titanic

Aprenentatge Computacional

Nerea de la Torre Veguillas, 1669013
Mara Montero Jurado, 1671506
Júlia Morán Fluvià, 1667730
Adrián Prego Gallart, 1672251
23/10/2024

1 Anàlisi exploratori de dades (EDA)

La nostra base de dades té 12 atributs descrits a la taula 1, 5 dels quals són categòrics (`Name, Sex, Ticket, Cabin, Embarked`), 6 són numèrics (`PassengerId, Pclass, Age, SibSp, Parch, Fare`) i el nostre target que és `Survived`, un atribut binari, el qual el valor 1 ens indica que el passatger va sobreviure i 0 que no.

Trobem Nans en els atributs `Age`, `Cabin` i `Embarked`, els quals més endavant solucionem. Les etiquetes no estan balancejades, ja que el percentatge de supervivents és de $38,36\%$ per al conjunt d'entrenament. Encara que no estiguin balancejades, no és un cas crític.

Analitzant els gràfics de la Figura 2 podem treure diferents conclusions. Per començar, que clarament hi ha una distinció per gèneres, ja que van sobreviure més dones que homes. A més, podem deduir que per la distribució del vaixell, els passatgers que van embarcar a la porta C estarien més ben situats quant a la facilitació de recursos d'emergència. També observem la classe té una alta relació amb la supervivència dels passatgers, ja que la primera i la segona classe tenen una major proporció de supervivents. Finalment, relacionat amb l'anterior, hi ha hagut una major proporció de salvació en els clients que van pagar més pel seu tiquet.

Amb el gràfic de la Figura 3 podem diferenciar les franges d'edat que existeixen entre les persones del vaixell, a primera vista veiem una major afluència de persones joves, sobretot entre els 20 i 30 anys. A més cal esmentar que un $61,8\%$ dels passatgers no van sobreviure, tal com es pot veure al gràfic de sectors de la Figura 4.

2 Preprocessament de dades

En aquest apartat, s'ha netejat i transformat el conjunt de dades per preparar-lo per a l'entrenament del model.
Sabem que alguns mètodes de classifiació no admeten NaNs, com el SVM. Com havíem vist abans, tenim tres atributs amb NaNs: age, cabin i embarked: `Age` (9.87% NaNs), `Embarked` (0.22% NaNs) i `Cabin` (77.1% NaNs).

Primer, s'han eliminat les columnes considerades irrellevants per a la predicció: `Name`, `PassengerId`, `Ticket` y `Cabin`. Les tres primeres pel fet que són identificadors únics i per tant no són rellevants, i `Cabin` pel gran percentatge de NaNs que contenia.

A continuació, s'han identificat els atributs categòrics i numèrics del conjunt de dades. Per a

la variable categòrica `Sex`, s'ha aplicat Label Encoding per convertir-la en valors numèrics (0 i 1).

Per omplir els NaNs de `Embarked` s'han omplert els valors nuls mitjançant KNN en funció dels atributs de `Class` i `Sex`, i posteriorment s'ha transformat utilitzant One-Hot Encoding, creant 3 columnes binàries per a cada possible valor de `Embarked`. Per omplir els NaNs de `Age`, utilitzarem el promig d'edat en funció dels atributs de `Class` i `Sex`.

Veiem també interessant analitzar si tenir família a bord en el titanic va influir en la supervivència (creiem que si). Per això afegim una nova variable `HasFamily` que pren valor 0 si la persona té familiars al vaixell i 1 en cas contrari.

Un cop ens hem quedat amb els atributs rellevants, tornem a observar la correlació entre les dades amb la Figura 5 i la Figura 6. Veiem que afegir l'atribut `HasFamily` influeix postivament en la supervivència.

En el procés de normalització que hem realitzat, volíem aplicar estandardització (Z-score) a les variables `Fare` i `Age` ja que presenten diferències importants a les seves escales. Un cop verificat que la variable `Age` té una distribució més o menys normal, mitjançant els gràfics Q-Q plot, hem aplicat l'estandardització utilitzant l'escala StandardScaler assegurant així que la variable tingui valors a la mateixa escala. Per altre banda, `Fare` no segueix exactament una distribució normal, però ho normalitzem perquè més endavant obtenim millors resultats tenint en compte aquest atribut que no pas eliminant-lo.

Per últim, el PCA s'utilitza per reduir dimensions en conjunts de dades amb moltes variables numèriques, resumint-les en components, també per multicol·linearitat, encara que pot dificultar la interpretació, i a més no es recomana amb poques variables. En aquest cas, no s'ha aplicat perquè només hi ha 4 variables numèriques.

3 Selecció de la mètrica

Quan treballem amb dades desbalancejades, l'`accuracy_score` pot ser enganyós, ja que pot donar la falsa impressió d'un alt rendiment a causa de la presència d'un nombre major d'instàncies d'una classe. Per exemple, si una classe constitueix el 90% de les dades, un model que sempre la predigués tindria una precisió del 90%, però no seria adequat per identificar la classe menys freqüent. En aquesta situació, el `f1_score` destaca per ser una opció més adequada, ja que integra tant la precisió com el *recall*, sent particularment beneficiós en casos de desigualtat entre les classes. Proporciona una visió més global de l'acompliment del model en considerar tant els

falsos positius com els falsos negatius. A més a més, com el nostre objectiu és identificar millor les persones que sobreviuen, és a dir, minimitzar els falsos negatius (aquells que el model no identifica com a supervivents quan realment ho són), llavors l' `F1 Score` seria una mètrica més adequada. El `average_precision_score` resumeix la corba de precisió-*recall* de manera menys intuïtiva que el `f1_score` en interpretació directa. Per tant, no la utilitzarem en aquest cas.

La corba ROC (Receiver Operating Characteristic) mostra la relació entre la taxa de falsos positius (FPR) i la taxa de veritables positius (TPR). És útil quan les classes estan més equilibrades o quan es vol analitzar la capacitat del model per distingir entre les classes.
La corba de precisió-recall mostra la relació entre precisió i recall per diferents llindars de classificació. És útil en conjunts de dades no balancejades, on la classe positiva és més important i és més difícil de detectar.
En aquest cas, la corba de precisió-recall és més rellevant, ja que proporciona una millor mesura de rendiment per a problemes amb dades no balancejades, i es concentra en la precisió de la classe positiva. A més, mostra com el model gestiona la classe minoritària (sobrevivents) i com equilibra la precisió amb el recall.

4 Selecció de model amb validació creuada

Una vegada ja hem vist les diferents mètriques, i havent triat l'adequada, procedim a la comparativa dels diferents models per a veure la seva actuació en el dataset, a més de l'elecció del model més adequat. Realitzem aquest procés:

Regressió Logística: aquest primer mètode és bona opció perquè és senzilla i funciona bé en problemes de classificació binària. També funciona bé amb conjunts de dades no massa grans, com el nostre cas. Com el nostre conjunt de dades no té una relació clarament lineal entre les variables i la supervivència, i tot i que un model lineal com la regressió logística pot oferir una primera aproximació útil, sovint és necessari considerar models més complexos.

Decision Tree: provem el model de Decision Tree (Arbre de decisió), ja que també són fàcils d'interpretar, es un model de classificació binària, gestionen bé les interaccions complexes entre les variables (en el nostre cas ens va bé pqerquè hi ha moltes diferències entre sexe y classes, per exemple).

Random Forest Classifier: ara fem Random Forest Classifier, ja que acumula tots els avantatges del decision tree, però ara combinem múltiples arbres per reduir el overfitting, per tant, és més robust.

KNN: també ens interessa mirar amb el KNN, ja que les probabilitats de sobreviure es possible que s'agrupin en característiques similars dels passatgers. Però només serà útil si la supervivència

depèn de la separació d'aquestes característiques.

SVM: el SVM (Màquines de vectors de suport) és una bona opció perquè també sap treballar amb dades numèriques i categòriques, a més de poder treballar amb dades no lineals gràcies als kernels.

Gradient Boosting:Per últim, el Gradient Boosting també el considerem una bona opció ja que combina diferents models més débils per millorar el rendiment, és precís i robust al overfitting.

Amb la validació creuada trobem el millor model amb els hiperparàmetres per defecte. En el nostre cas és millor utilitzar la StratifiedKFold, ja que garanteix que cada "fold"tingui una distribució similar de la classe objectiu (la proporció de supervivents i no supervivents es manté constant).
Podem veure en la Figura 7 que obtenim millors resultats amb el model de Random Forest, tenint en compte els hiperparàmetres per defecte. Per aconseguir-ho, hi havia diferents mètodes, com Random Search (per tenir una cerca eficient i ràpida) o Bayesian Search (si es vol optimitzar al màxim amb menys proves), nosaltres ens hem quedat amb el Grid Search, tot i que té un major cost computacional, ens dona la precisió més alta, ja que fa totes les combinacions d'hiperparàmetres. Un cop fet això, els resultats obtinguts amb aquests nous valors han tingut una millora la qual podem observar en la taula (Figura 8).

5 Anàlisi final

Un cop realitzat l'estudi del dataset, podem afirmar que la millor opció és el Random Forest seguida del Gradient Boosting, ja que es tracta de models robustos per datasets com el Titanic que inclouen diversitat de característiques, relacions no lineals, interaccions complexes entre variables i que eviten l'overfitting.
Per acabar de comprovar que els models escollits són els millors fem un gràfic amb les corves PR de tots els models, ja que aquesta corva no només avalua que el model predigui bé, sinó també, que en les vegades que s'equivoca, ho faci amb probabilitats menors a 0,5.
Com es pot veure a la Figura 9, els models que tenen l'AUC major són el Random Forest i el Gradient Boosting. Un valor alt d'AUC indica que el model té un millor rendiment en la detecció de classes positives (en el nostre cas la supervivència). Els models com SVM i Regressió Logística tot i que tenen bons resultats, no són tan competitius com els models basats en arbres en aquest cas.
Per últim, aquest model pot ser útil en el futur per a prediccions de supervivència, priorització en emergències, investigacions històriques i en sectors com la salut o la seguretat en el transport. Adaptant-lo a diferents contextos, podria oferir valor en la presa de decisions basada en dades, millorant la seguretat i l'eficiència en diverses aplicacions pràctiques.

6 Annex

Aquí es troben un recull de gràfics més rellevants per l'anàlisi a l'hora de trobar el millor model. En el notebook d'aquest treball es troben altres gràfics i més informació i respostes a preguntes que complementen la informació donada.

Nom	Tipus	Descripció
`passengerId`	int64	Valor d'identificació únic de cada passatger
`name`	object	Nom del passatger
`sex`	object	Gènere (masculí o femení)
`age`	float64	Edat de la persona (menors de 12 anys representats en fracció d'any)
`pclass`	int64	Classe del ticket (1 = 1a, 2 = 2a, 3 = 3a)
`embarked`	object	Porta d'embarcament dels passatgers
`ticket`	object	Número de ticket (NA per la tripulació)
`fare`	float64	Preu del ticket (NA per la tripulació, empleats i altres)
`sibsp`	int64	Número de germans/familiars
`cabin`	object	Tipus de cabina ocupada pel passatger
`parch`	int64	Número de pares i fills a bord
`survived`	int64	Sobreviu a l'enfonsament (0 = No, 1 = Sí)

Figura 1: Descripció de les variables del dataset Titanic

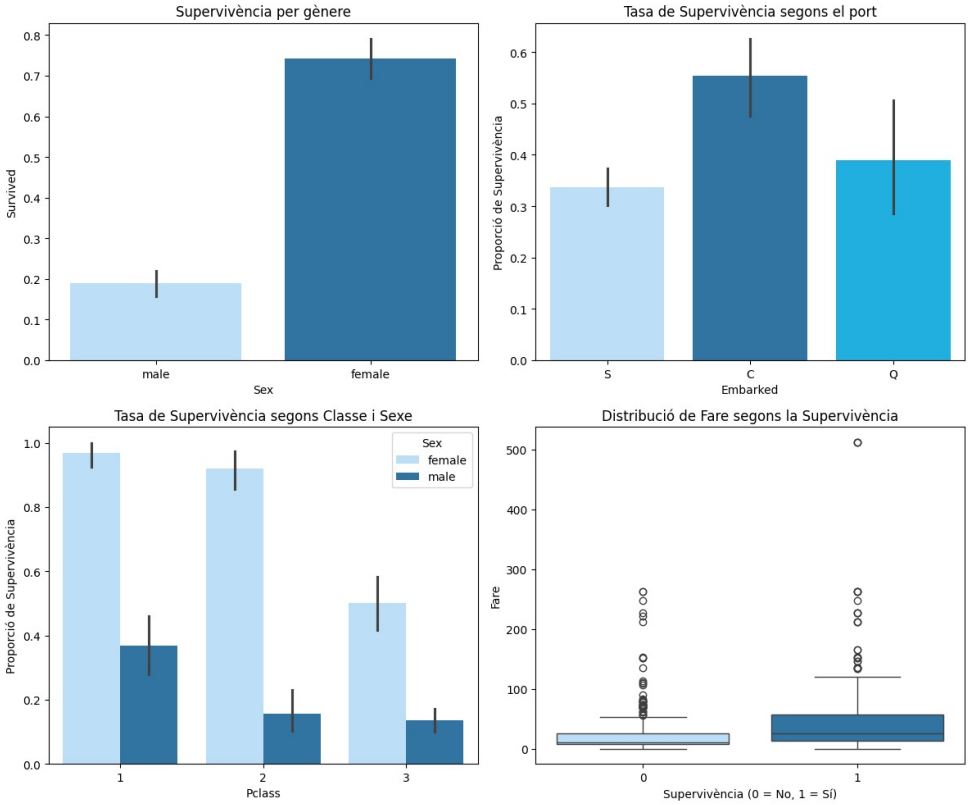

Figura 2: Gràfics de Correlacions

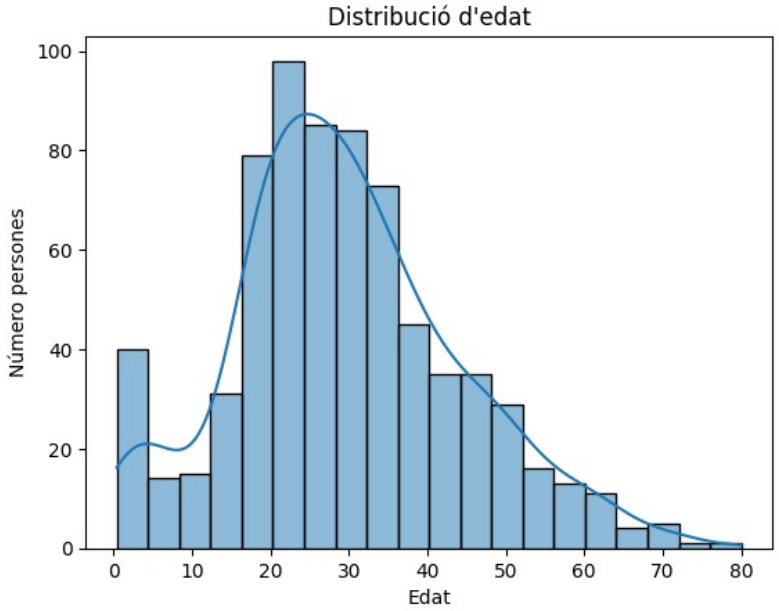

Figura 3: Gràfic de distribució d'edats

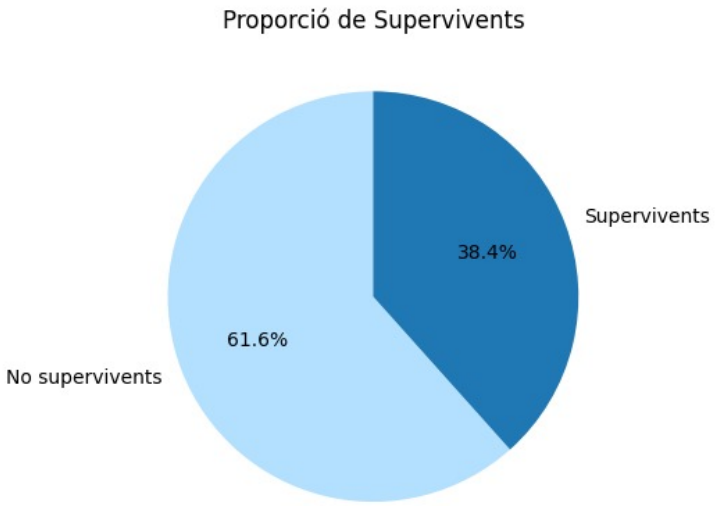

Figura 4: Gràfic de sectors de supervivència

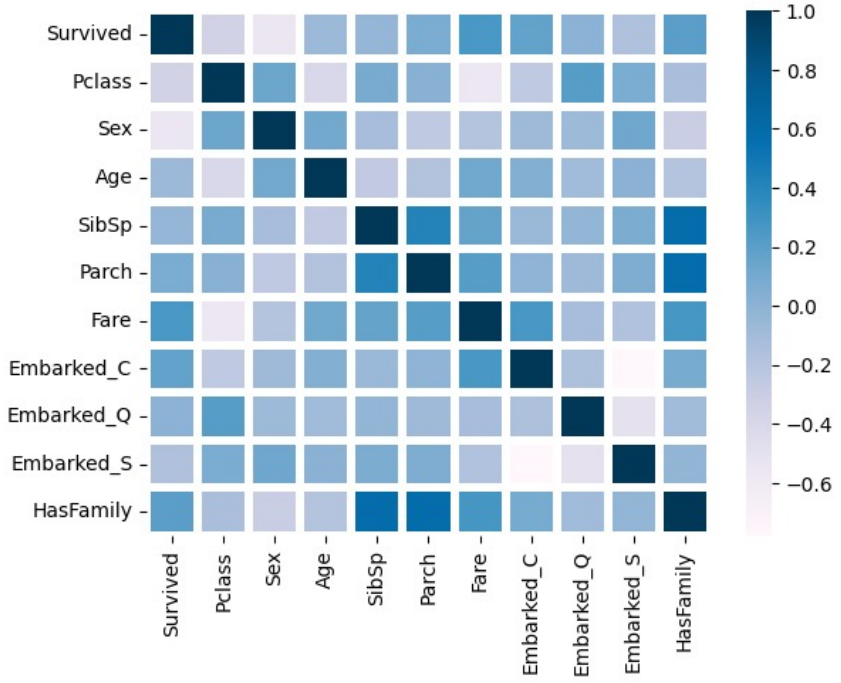

Figura 5: Matriu de Correlació

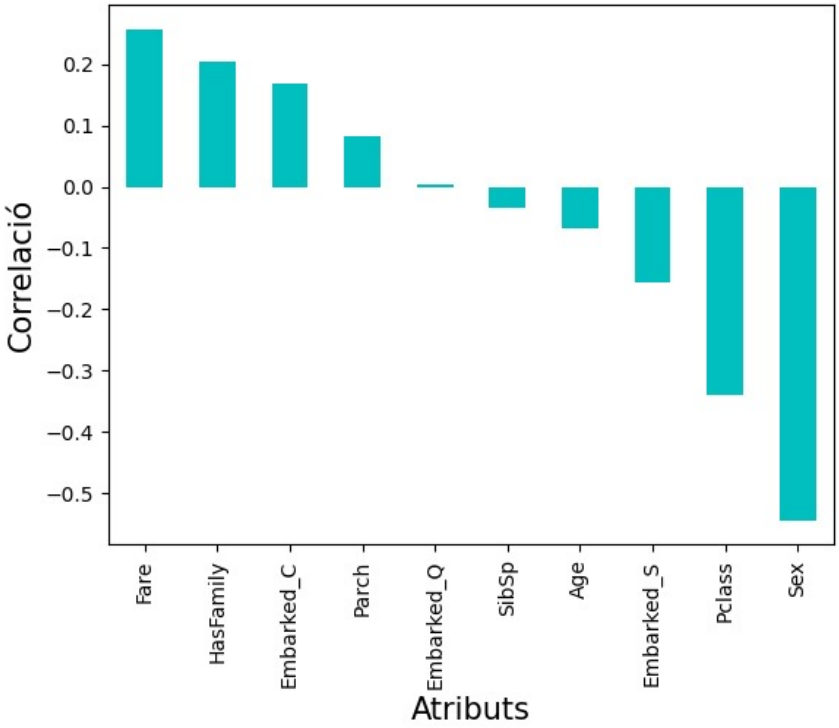

Figura 6: Gràfic de barres de la correlació

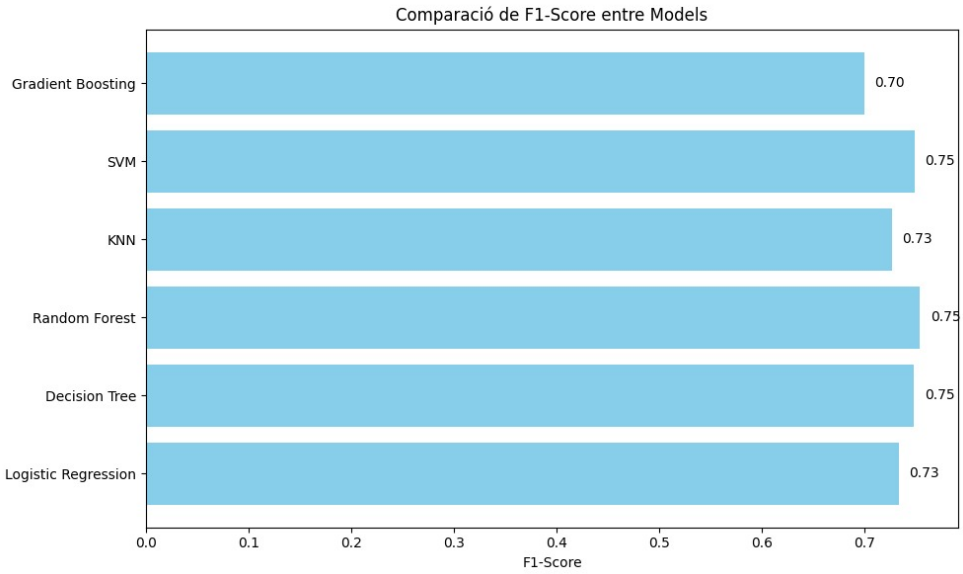

Figura 7: Comparació dels `f1_score` dels models

Model	Paràmetres	Mitjana F1 scores	Train F1 scores	Temps (s)
`Logistic Regression`	C: 1, penalty: l2, solver: lbfgs	0.7415	0.747	15.033
`Decision Tree`	criterion: entropy, max_depth: 10, min_samples_leaf: 4, min_samples_split: 5	0.7447	0.896	3.812
`Random Forest`	criterion: log_loss, max_depth: 30, min_samples_split: 10, n_estimators: 100	0.7779	0.847	64.417
`KNN`	metric: manhattan, n_neighbors: 11, weights: distance	0.7436	0.983	1.055
`SVM`	C: 100, gamma: 0.01, kernel: rbf	0.7668	0.786	9.913
`Gradient Boosting`	learning_rate: 0.1, max_depth: 4, min_samples_split: 2, n_estimators: 150, subsample: 1.0	0.7697	0.797	238.435

Figura 8: Taula d'informació dels models

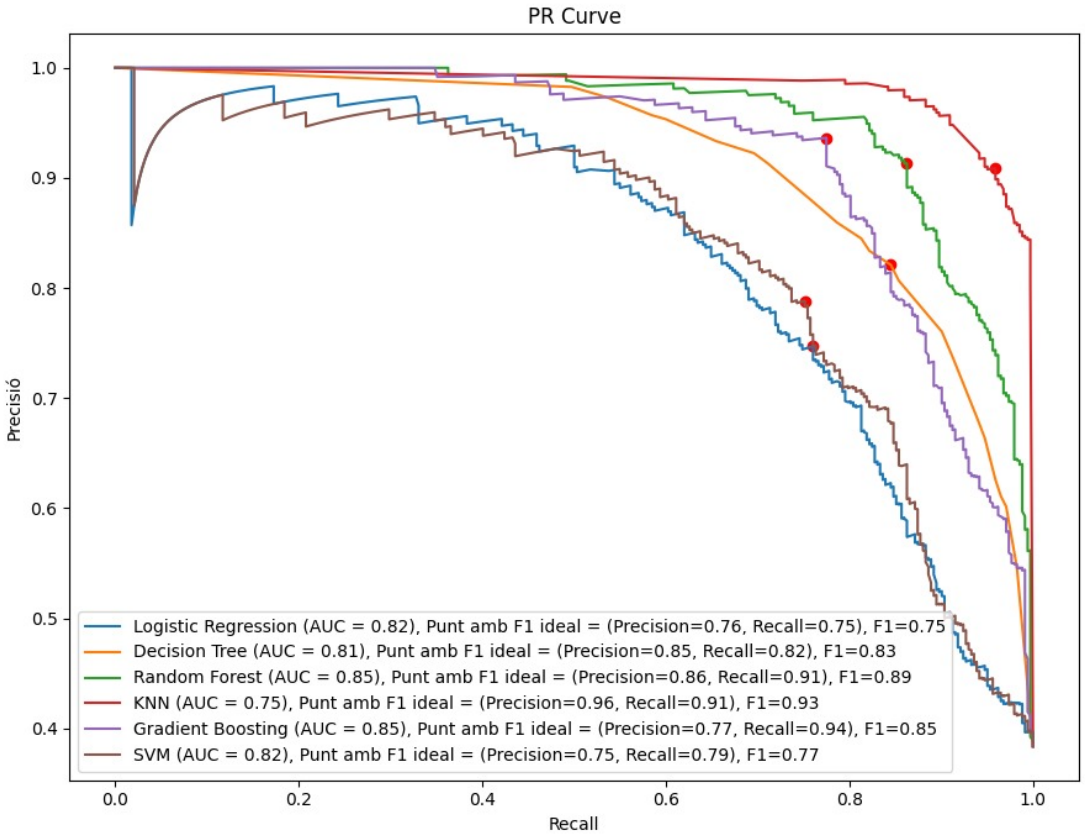

Figura 9: PR Curve amb tots els models