

Aplicació Interactiva per a la Predicció i Seguiment del Rendiment Acadèmic

Joel Tapia Salvador

Resum— Projecte amb l'objectiu de crear una aplicació web interactiva que permeti a les famílies visualitzar una predicció o anàlisis del rendiment acadèmic dels seus fills. S'assoleix l'objectiu a partir de dades introduïdes per la família mateixa sobre les activitats, temps d'estudi, rendiment anterior dels seus fills i altra informació, a partir de la qual un model, o diversos, predictiu crea un perfil. A més a més, oferirà funcionalitats interactives i recomanacions per millorar el rendiment, centrat a facilitar el suport per part de les famílies als seus fills.

Paraules clau— Aplicacions, Avaluació del rendiment, Educació, Intel·ligència Artificial, Modelatge, Perfils d'usuari i serveis d'alerta, Personalització, Serveis basats en web, Suport a la presa de decisions, Visualització

Abstract—Project with the objective of creating an interactive web application to allow families to visualize a prediction or analysis of their kid's academic performance. This is achieved using data introduced by the own family about the activities, time spent studying, past performance of their kid and other information, which one, or several, predictive models will use to create a profile. In addition, it will offer interactive recommendations to improve the performance of the kids, focused on improving the assistance given by the families to their own kids.

Index Terms— Applications, Artificial Intelligence, Decision support, Education, Modelling, Performance evaluation, Personalization, User profiles and alert services, Visualization, Web-based services

1 INTRODUCCIÓ

AQUEST treball té com a objectiu desenvolupar, en un àmbit purament acadèmic, una aplicació web interactiva amb models d'intel·ligència artificial que puguin predir el rendiment acadèmic dels fills de famílies a través de la informació introduïda a l'aplicatiu.

Un dels principals punts a investigar és quin or quins models performen de forma més remarcable a l'hora de predir els resultats acadèmics i com millorar-los.

Per poder fer l'entrenament necessitem un conjunt de dades extens, unificat i ampli amb possibles dades acadèmiques, d'estudi i oci que poden afectar el rendiment.

L'aplicatiu serà senzill, dut que l'objectiu es recerca acadèmica, no es posarà molta feina en l'aspecte visual de l'aplicatiu ni que pugui ser massivament desplegat com a servei web, sinó, que sigui funcional.

2 ESTAT DE L'ART

2.1 Model predictiu

L'avaluació i predicció del rendiment acadèmic utilitzant intel·ligència artificial ha sigut objectiu d'estudi en altres recerques. A partir d'aquest podem observar quins mo-

dels han sigut fet servits mes sovint i quins tenen millor resultat per reduir la quantitat de models que entrenar i testejar.

Un dels models utilitzats ha sigut la *Logistic Regression* (LR) tal com es presenta a [1], [2], [3], [4], [6], [7], [8] i [10] amb una *accuracy* oscil·lant d'un 57.9% a un 92,1%.

Un altre model és *Support Vector Machine* (SVM) vist a [1], [3], [5], [6], [9] i [10] amb l'interval de resultats en *accuracy* de 60.0% a 93,9%.

També s'utilitza de forma habitual els *Decision Trees* (DT) amb l'*accuracy* variant de 42.0% fins 95.7% tal com mostren [1], [2], [3], [4], [5], [6], [9] i [10]. Algunes d'aquestes implementacions són DT especials tal com *Bagging Trees* [9] i *Boosting Trees* [9].

A [1], [3], [6], [7], [8] i [9] també es fa servir *Random Forest* amb un *accuracy* que varia del 52.9% al 96.1%.

Un model menys utilitzat és *Naïve Bayesian Algorithm* en [3], [4], [5], [6] i [9] amb una *accuracy* de 50.0% a 76.0%.

També s'implementa molt *Artificial Neural Networks* (ANN) com vist a [2], [3], [4], [5], [6], [8] i [10]. Aquests inclouen models com *Backpropagation Neural Networks* (BPNN) [2], *Radial-Basis Function Neural Networks* (RBFNN) [2], *Recurrent Neural Networks* (RNN) [4], *Long Short Term Memory* (LSTM) [4] i *Multi Layer Perceptrons* (MLP) [4]. Aquests models tenen una *accuracy* de 65.0% a 91.5%.

- E-mail de contacte: Joel.TapiaS@uab.cat
- Treball tutoritzat per: Oriol Cortés Comellas (Àrea de Ciències de la Computació i Intel·ligència Artificial)
- Curs 2024/25

2.2 Aplicatiu interactiu

Per a desenvolupar aplicatius webs en Python [11] s'utilitza Django [12], Flask [13] i FastAPI [14].

HTMX [15] en una opció popular per afegir interactivitat sense necessitat d'un *framework* JavaScript [16] complet. Alpine.js [17] és una llibreria JavaScript lleugera sovint feta servir per a petits elements interactius. Tot i que *frameworks* més complets com React [18] es poden integrar amb *backends* Python, normalment es reserven per a aplicacions més complexes.

Pel que fa a l'estil, *Bootstrap* [19] continua sent el *framework* més utilitzat per la seva facilitat d'ús i capacitat de disseny responsiu.

Els desenvolupadors Python sovint fan servir motors de plantilles com Jinja [20] per generar HTML [21] dinàmicament i integrar-se fàcilment amb aquestes eines de *frontend*.

En el *backend* amb Python normalment es fan servir base de dades com PostgreSQL [22] o SQLite [23].

3 METODOLOGIA

3.1 Metodologia entrenament model

Per entrenar els models aplicarem entrenament amb *K-Folds* com es veu a Fig. 1. En el pas 1 dividirem el conjunt de dades en *K folds*. S'aplicaran *K* iteracions on a cada iteració s'agafarà un subconjunt dels *folds* com a conjunt d'entrenament i els restants com a conjunt de test.

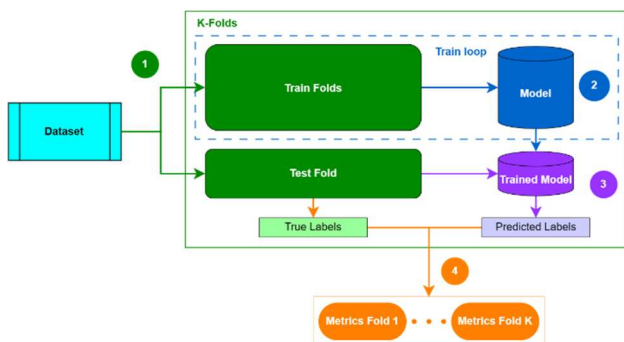


Fig. 1. Diagrama de la metodologia general *K-Fold* per l'entrenament dels models.

Al pas 2, en cada iteració, s'agafarà el conjunt d'entrenament i s'entrenarà el model. En el pas 3, una vegada el model s'hagi entrenat, es faran prediccions sobre el conjunt de test. En el pas 4 es compararan les prediccions amb els *groundtruth*, s'obtingueran mètriques i s'acumularan totes les mètriques obtingudes per cada iteració del *K-Fold*.

En concret el nostre procediment per escollir models i entrar-los serà esplotat a la següent secció.

3.2. Metodologia de decisió de models

Com es veu a l'estat de l'art hi ha moltes possibles arquitectures per obtenir el nostre objectiu, totes aquestes amb variats resultats. Per poder decidir quina utilitzar per a l'aplicatiu és necessari provar-les amb les dades que es faran servir i veure el seu rendiment.

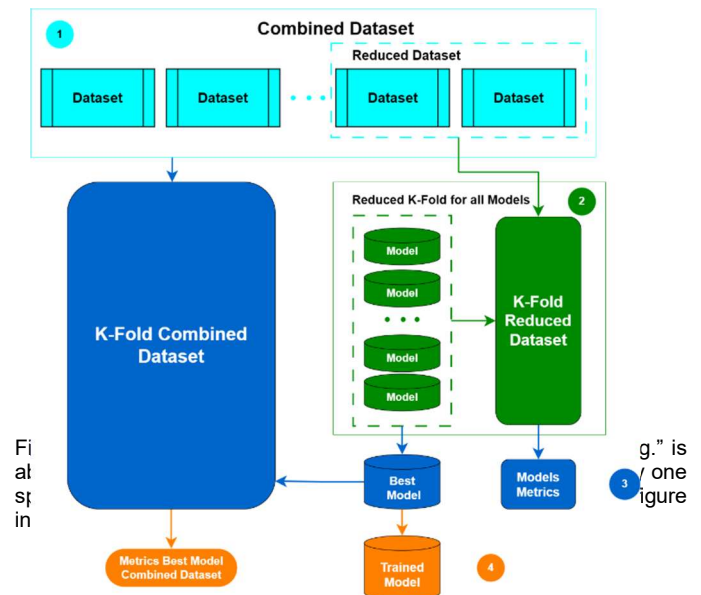


Fig. 2. Diagrama de la metodologia a seguir per decidir el millor model per l'aplicatiu.

Per fer això implementarem la metodologia de la Fig. 2. Al pas 1 combinarem diversos conjunts de dades en un sol. Al pas 2 agafarem un petit subconjunt del nou conjunt de dades i aplicarem la metodologia d'entrenament de model de la Fig. 1 sobre aquest subconjunt per cada model que utilitzarem. Al pas 3 tindrem les mètriques de tots els *K-Folds* per a tots els models i escollirem el model amb millor rendiment en les prediccions. Al pas 4, amb aquest millor model, l'entrenarem fent servir la metodologia *K-Folds* amb totes les dades del conjunt complet, obtenint les mètriques finals i el model entrenat que utilitzarem a l'aplicatiu.

Els diferents models que es treballarà seran LR, SVM, DT, RF i ANN.

3.3 Metodologia desenvolupament aplicatiu

Per desenvolupar l'aplicatiu web interactiu s'utilitzarà la metodologia *Test-driven Development*. Aquesta metodologia comença dissenyant una sèrie de tests que el programari ha de superar pel seu correcte funcionament i a continuació es crea el programari satisfent els tests.

Primer es treballarà en les connexions del *backend* amb el model i base de dades necessàries. A continuació la estructura del *backend* web. I, finalment, el *frontend*.

4 PLANIFICACIÓ

Per dur a terme el projecte de forma correcta i a temps s'ha designat una planificació a seguir fins al final del període. S'ha fet un gràfic de Gant amb aquesta, es pot veure a la Fig. 3. S'ha planificat pel temps que es trigarà en preparar tots els conjunts de dades, fer les proves inicials de decisió de model. Fer l'entrenament del model final i desenvolupar cada fase de l'aplicatiu. També s'ha planificat les entregues durant el termini, incloent mantenint el dossier i la preparació de la presentació.

5 DATA

Per poder entrenar els models necessitem dades sobre diferents estudiants. Trobem base de dades ([24], [25] i [26]) no molt grans però que a més tenen alguns paràmetres en comú, altres no i tots expressats de forma diferent. Per això ajuntarem tots les bases de dades en una sola.

Per poder unificar les base de dades primer em de transformar les dades comuns de tots les bases de dades en un mateix format i rang de dades. Com que la base de dades te dades categòriques i continues, em de fer dos passos.

Primer unifiquem totes les variables categòriques amb la mateix codificació, fent un mapeig per cada base de dades de forma que al final quedi una codificació consistent.

Segon unifiquem es valors continues, escalant-los tots de forma que tinguin el mateix rang de 0 a 1.

El següent pas es afegir a les bases de dades les variables de les altres que no tenen per fer això escollim un valor, "-1", que indicarà que no hi ha informació d'una variable en una observació donada. Això també ens permetrà entrenar models que no necessitin totes les dades ja que a l'aplicatiu real es d'esperar que no sempre es sàpiga totes les dades o que els usuaris les volen dir totes.

En el cas de les notes, quines son el nostre objectiu, hi ha alguns sistemes i bases de dades que les categoritzen (Excelent, Notable, A+, B, etc.) i d'altres que son numèriques a diferents escales. A la mateixa hora, alguns dels nostres models son classificadors, d'altres regressius i la xarxa neuronal se'l por fer entrenar ambdues a l'hora amb múltiple *losses* i sortides. Per això, com que inherent hi ha un ordre encara que siguin classes, s'han codificat les classes en 5 numèrics enters depenent del país d'origen.

Totes aquestes operacions i transformacions de la base de dades les drem a terme utilitzant el mòdul *Pandas* [27] de *Python*.

6 ENTRENAMENT MODELS

Per entrenar els models i decidir quin és el millor utilitzem un subset aleatori de la base de dades que hem creat y apliquem un K-Fold amb 5 grups on a cada Fold entrenem tots els tipus de models, calculem les mètriques i comparem els resultats. Utilitzem els mètodes de K-Fold de *Sklearn* [28] de *Python*.

Els DT, RF i LR que utilitzarem també son les implementacions del mòdul *Sklearn* i per a les NN utilitzarem el mòdul *Torch* [29] de *Python*.

Com comentat abans, les notes son numèriques dividides en classes i per tant a partir d'ara tots les prediccions de les regressions seran aproximades als enters y els valors *clipped* dins del rang 0-4. Utilitzarem per comparar els resultats mètriques de classificació i mètriques de regressió.

Degut a problemes i complicacions amb el model de NN, calcular tots el tipus de mètrics i efectes externs, el NN i SVC han estar enrederits, de moment compararem el resultat dels models DT, RF i LR.

Com es pot veure RF té una *accuracy* major especialment comparat amb LR, en canvi LR té una millor R2 i MSE, quan s'equivoca ho fa a més distancia ja que no son classes, però RF està a prop, encara que té una MSE bastant més gran. En canvi DT s'aproxima a RF en *accuracy* però té errors molt grans i llunyans quan s'equivoca.

Per la segona fase s'ha decidit utilitzar RF amb els *K-Folds* de totes les dades.

TABLE 1
MITJANA MÈTRQUES CLASSIFICACIONS SOBRE FOLDS

Model	Accuracy	F1 Score
DT	79.778%	0.38333
RF	82.720%	0.47010
LR	70.353%	0.21594

TABLE 2
MITJANA MÈTRQUES REGRESSIONS SOBRE FOLDS

Model	MAE	MSE	RMSE	R2
DT	1.3625	4.214	2.0525	-0.4262
RF	1.1852	3.749	1.9341	-0.2687
LR	1.2226	2.381	1.5422	0.1943

Com podem veure als resultats de la mitjana de les mètriques del model RF sobre 5 *K-Folds* del data set complet. Té resultats molt similars als anteriors, inclús millorant-los tots. Encara això el resultats no son molt positius, ja que falla bastant.

TABLE 3
MITJANA MÈTRQUES DT SOBRE FOLDS

Model	Accuracy	F1 Score	MAE	MSE	RMSE	R2
RF	83.425%	0.47518	1.1271	3.562	1.8866	-0.2066

8. CONCLUSIÓ

Degut a les complicacions no s'han pogut finalitzar de comparar tots els models, però per les previsions amb els altres models, la falta de moltes dades en alguns camps i considerant com de bo solen ser els RF amb dades tabulars es possible que SVC no doni bons resultats.

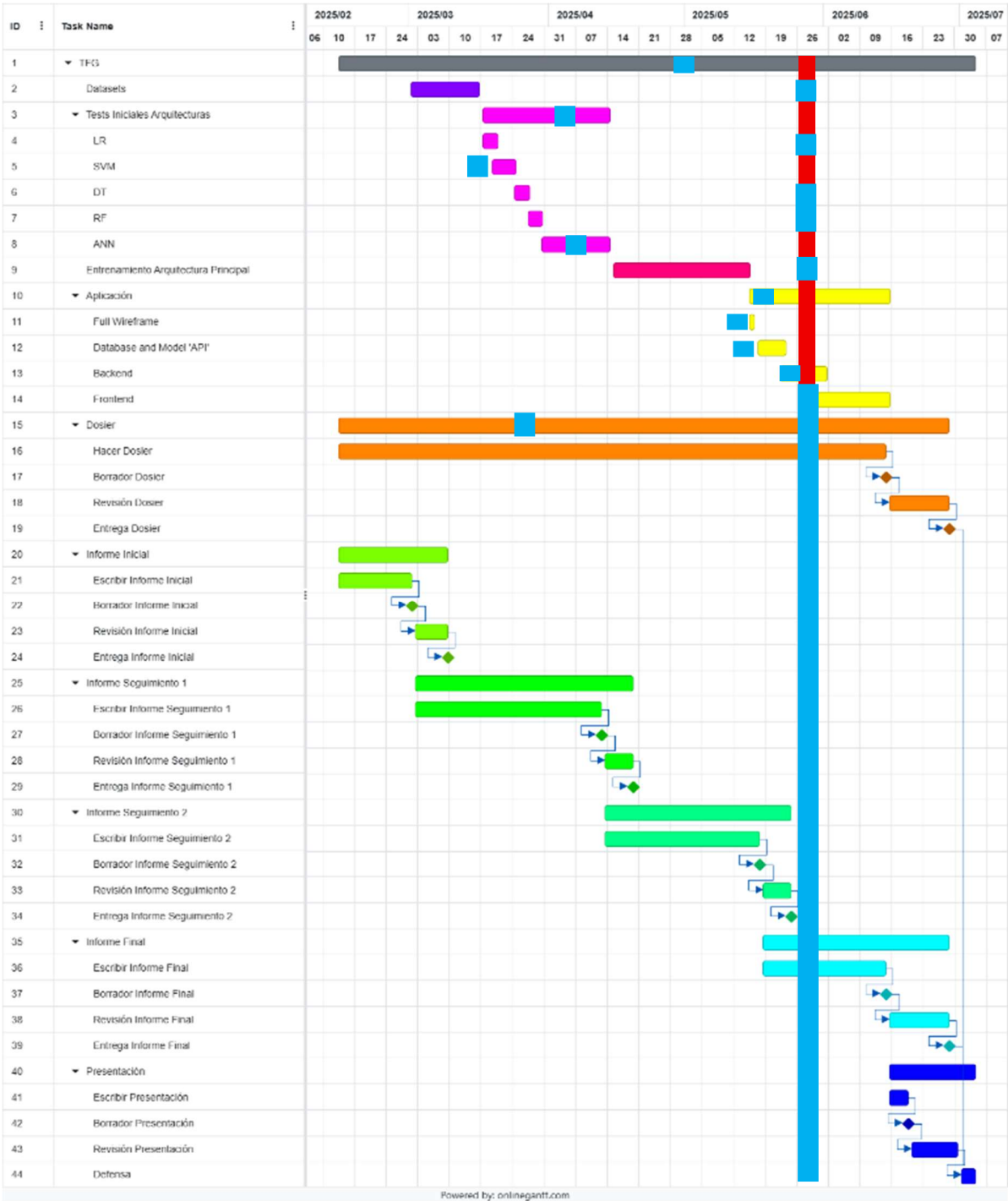
En canvi NN amb múltiple *losses* que maximitzin ambdues mètriques es possible que superi al RF.

Es necessitaria un data set molt mes gran, amb més dades variabilitat de dades (donat el fet que son múltiples base de dades fusionats hi ha bastants atributs que no son comuns i només hi ha dades d'una base de dades i no la resta). També més variabilitat de les notes, en les base de dades obtinguts les notes estaven molt juntes en uns valors com la meitat alts i baixos però les zones intermitges no hi tenien cap casi.

BIBLIOGRAFIA

- [1] M. V. Martins, D. Toledo, J. Machado, L. M. T. Baptista, and V. Realinho, "Early Prediction of student's Performance in Higher Education: A Case Study", in Trends and Applications in Information Systems and Technologies, 2021, pp. 166–175. doi: https://doi.org/10.1007/978-3-030-72657-7_16.
- [2] N. Yilmaz and B. Sekeroglu, "Student Performance Classification Using Artificial Intelligence Techniques", in 10th International Conference on Theory and Application of Soft Computing, Computing with Words and Perceptions - ICSCCW-2019, 2020, pp. 596–603. doi: https://doi.org/10.1007/978-3-030-35249-3_76.
- [3] C. Romero and S. Ventura, "Educational Data Mining: A Review of the State of the Art", IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews), vol. 40, no. 6, pp. 601–618, 2010. doi: <https://doi.org/10.1109/TSMCC.2010.2053532>.
- [4] N. Mduma, K. Kalegele, and D. Machuve, "A Survey of Machine Learning Approaches and Techniques for Student Dropout Prediction", Data Science Journal, Apr. 2019. doi: <https://doi.org/10.5334/dsj-2019-014>.
- [5] A. M. Shahiri, W. Husain, and N. A. Rashid, "A Review on Predicting Student's Performance Using Data Mining Techniques", Procedia Computer Science, vol. 72, pp. 414–422, 2015. doi: <https://doi.org/10.1016/j.procs.2015.12.157>.
- [6] J. L. Rastrollo-Guerrero, J. A. Gómez-Pulido, and A. Durán-Domínguez, "Analyzing and Predicting Students Performance by Means of Machine Learning: A Review", Applied Sciences, vol. 10, no. 3, 2020. doi: <https://doi.org/10.3390/app10031042>.
- [7] C. Beaulac and J. S. Rosenthal, "Predicting University Students Academic Success and Major Using Random Forests", Research in Higher Education, vol. 60, no. 7, pp. 1048–1064, Nov. 2019. doi: <https://doi.org/10.1007/s11162-019-09546-y>.
- [8] A.-S. Hoffait and M. Schyns, "Early detection of university students with potential difficulties", Decision Support Systems, vol. 101, pp. 1–11, Sep. 2017, doi: <https://doi.org/10.1016/j.dss.2017.05.003>.
- [9] V. Miguéis, A. Freitas, P. J. Garcia, and A. Silva, "Early segmentation of students according to their academic performance: A predictive modelling approach", Decision Support Systems, vol. 115, pp. 36–51, Nov. 2018, doi: <https://doi.org/10.1016/j.dss.2018.09.001>.
- [10] D. Thammasiri, D. Delen, P. Meesad, and N. Kasap, "A critical assessment of imbalanced class distribution problem: The case of predicting freshmen student attrition", Expert Systems with Applications, vol. 41, no. 2, pp. 321–330, Feb. 2014, doi: <https://doi.org/10.1016/j.eswa.2013.07.046>.
- [11] Python Software Foundation. *Python*. (2025). Python Software Foundation. [Online]. Available: <https://www.python.org>
- [12] Django Software Foundation. *Django*. (2025). Django Software Foundation. [Online]. Available: <https://www.djangoproject.com>
- [13] Pallets. *Flask*. (2025). Pallets Projects. [Online]. Available: <https://flask.palletsprojects.com>
- [14] Tiangolo. *FastAPI*. (2025). Tiangolo. [Online]. Available: <https://fastapi.tiangolo.com>
- [15] Big Sky Software. *HTMX*. (2025). HTMX. [Online]. Available: <https://htmx.org>
- [16] Mozilla. *JavaScript*. (2025). Mozilla Developer Network. [Online]. Available: <https://developer.mozilla.org/en-US/docs/Web/JavaScript>
- [17] Caleb Porzio. *AlpineJS*. (2025). AlpineJS. [Online]. Available: <https://alpinejs.dev>
- [18] Meta Open Source. *React*. (2025). React. [Online]. Available: <https://react.dev>
- [19] Bootstrap. *Bootstrap*. (2025). Bootstrap. [Online]. Available: <https://getbootstrap.com>
- [20] Pallets. *Jinja*. (2025). Pallets Projects. [Online]. Available: <https://jinja.palletsprojects.com>
- [21] World Wide Web Consortium, Web Hypertext Application Technology Working Group. *HTML Standard*. (2025). WHATWG. Available: <https://html.spec.whatwg.org>
- [22] The PostgreSQL Global Development Group. *PostgreSQL*. (2025). The PostgreSQL Global Development Group. Available: <https://www.postgresql.org>
- [23] D. Richard Hipp. *SQLite*. (2025). The SQLite Consortium. Available: <https://sqlite.org>
- [24] V. Realinho, M. Vieira Martins, J. Machado, and L. Baptista, "Predict Students Dropout and Academic Success", UCI Machine Learning Repository, 2021. doi: <https://doi.org/10.24432/C5MC89>.
- [25] N. Yilmaz and B. Şekeroğlu, "Higher Education Students Performance Evaluation", UCI Machine Learning Repository, 2019. Available: <https://doi.org/10.24432/C51G82>.
- [26] P. Cortez. "Student Performance", UCI Machine Learning Repository, 2008. [Online]. Available: <https://doi.org/10.24432/C51G7T>.
- [27] NumFOCUS, Inc. *Pandas*. (2025). pandas. Available: <https://pandas.pydata.org>
- [28] Scikit-learn. *Scikit-learn*. (2025). Scikit-learn. Available: <https://scikit-learn.org/stable>
- [29] Pytorch. *The Linux Foundation*. (2025). PyTorch. Available: <https://pytorch.org>

APÈNDIX A
 PLANIFICACIÓ



Les barres blaves indiquen el progrés actual del treball i la barra vermella el progrés esperat tal i com s’havia espe-
 rat.