

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 es pot observar quins models

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

Un dels models que s'han utilitzat ha sigut la *Logistic Regresion* (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'han utilitzat 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 un dels *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 s'aplicarà un 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.

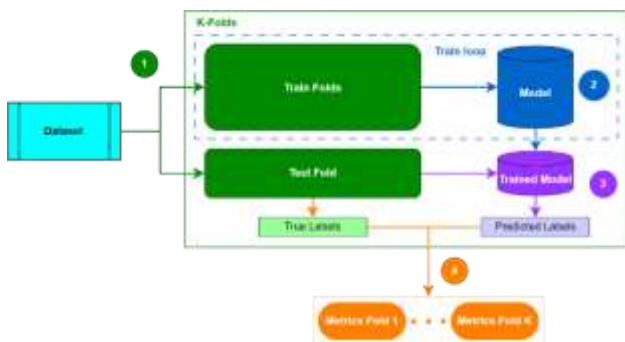


Fig. 1. Diagrama de la metodologia general *K-Fold* per l'entrenament dels models.

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 procediment per escollir models i entrar-los serà espleiat 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.

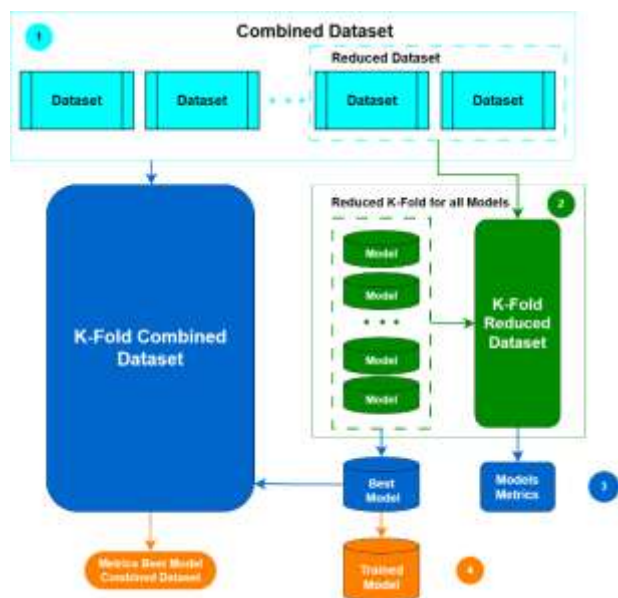


Fig. 2. Diagrama de la metodologia a seguir per decidir el millor model per l'aplicatiu.

Per fer això s'implementarà la metodologia de la Fig. 2. Al pas 1 es combinaran diversos conjunts de dades en un sol. Al pas 2 s'agafarà 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 s'utilitzarà. Al pas 3 s'obtingueran les mètriques de tots els *K-Folds* per a tots els models i s'escollirà el model amb millor rendiment en les prediccions. Al pas 4, amb aquest millor model, s'entrenarà fent servir la metodologia *K-Folds* amb totes les dades del conjunt complet, obtenint les mètriques finals i el model entrenat que s'utilitzarà a l'aplicatiu.

Els diferents models que es planifiquen treballar són LR, SVM, DT, RF i ANN.

3.3 Metodologia recomanador de millores

Per ajudar als estudiants, es desenvoluparà un recomanador de quins aspectes a millorar per augmentar el rendiment acadèmic.

Aquest recomanador, alterarà els valors actuals del estudiant de forma progressiva (augmentant-los o disminuint-los), fins que hi trobi l'alteració d'un d'ells que, segons el model classificador, millori el rendiment i suggerirà al estudiant a provar a fer el canvi i veure si el millora.

A partir d'aquí es podria implementar un sistema de validació que a futur, preguntí al estudiant si ha aplicat el canvi suggerint, com s'han vist modificat el rendiment i actualitzar el model amb aquesta informació.

3.4 Metodologia desenvolupament aplicatiu

Per desenvolupar l'aplicatiu web interactiu es planifica utilitzar 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 al annex. 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 es necessita dades sobre diferents estudiants. Es troba la 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 s'ajunten totes les bases de dades en una sola.

Al unificar les bases de dades, s'acaba amb variables que son:

1. Innates e invariants del estudiant, com: edat, educació dels pares, nivell actual de l'educació, etc.
2. Difícils de modificar pel estudiant, com: lloc de residència, accés a internet, etc.
3. Alterables pel estudiant, com: nombre d'hores estudiades, nombre d'absències, prendre notes a classe, etc.

Aquesta distinció serà rellevant pel recomandor.

Per poder unificar les base de dades primer s'ha de transformar les dades comuns de totes les bases de dades en un mateix format i rang de dades. Com que la base de dades té dades categòriques i contínues, s'ha de fer dos passos.

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

Segon, s'unifiquen els valors continus, 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 no les volen dir totes.

En el cas de les notes, que son l'objectiu, hi ha alguns sistemes i bases de dades que les categoritzen (Excelent, Notable, A+, B, etc.) i d'altres en que son numèriques a diferents escales. A la mateixa hora, alguns dels models son classificadors, d'altres regressius i la xarxa neuronal se'l pot 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. La raó de 5 "classes" es degut a que la majoria de les notes de la base de dades son pròximes a la màxima, al mínim o l'aprobat del sistema. Això no ens permet fer un sistema amb més exactitud utilitzant més possibles "classes" de notes. També hi ha pocs paràmetres relacionats directament

amb les notes, casi tots son relacionats a l'ambient del estudiant o alguns pocs als estudis d'aquest.

Totes aquestes operacions i transformacions de la base de dades es fan utilitzant el mòdul *Pandas* [27] de *Python*.

6 ENTRENAMENT MODELS

Per entrenar els models i decidir quin és el millor s'utilitza un subset aleatori de la base de dades que s'ha creat a l'apartat 5 i s'aplica un K-Fold amb 5 grups. On a cada Fold s'entrena tots els tipus de models amb les particions d'entrenament. A continuació, es calculen les mètriques de resultats amb la partició de validació i comparem els resultats. S'utilitza els mètodes de K-Fold de *Sklearn* [28] de *Python*.

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

Com s'ha 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. Per comparar els resultats s'utilitzaran mètriques de classificació i mètriques de regressió.

Degut a problemes i complicacions amb el model de ANN, degut a l'intent de fer un programa que integrés *Torch* i *Sklearn* de forma dinàmica que abstraguís els models per utilitzar després a l'aplicatiu i que calcules tots el tipus de mètriques per tots els models de la mateixa manera i degut a altres efectes externs, els models ANN i SVC han sigut descartats per donar temps ha desenvolupa l'aplicatiu. Al final es compararan els resultat dels models DT, RF i LR.

Model	Accuracy	F1 Score
DT Classifier	79.4% ± 0.6%	0.381 ± 0.063
DT Regressor	80.1% ± 0.8%	0.332 ± 0.073
RF Classifier	82.4% ± 1.0%	0.474 ± 0.039
RF Regressor	71.6% ± 0.6%	0.234 ± 0.013
LR	70.4% ± 1.5%	0.216 ± 0.025

Taula 1. Mitjana i desviació estàndard de les mètriques de classificacions sobre folds.

Model	MAE	MSE	RMSE	R2
DT Classifier	1.408 ± 0.047	4.409 ± 0.184	2.099 ± 0.044	-0.492 ± 0.063
DT Regressor	1.32 ± 0.033	4.047 ± 0.105	2.012 ± 0.026	-0.37 ± 0.033
RF Classifier	1.221 ± 0.103	3.879 ± 0.408	1.967 ± 0.104	-0.313 ± 0.136
RF Regressor	1.199 ± 0.037	2.46 ± 0.166	1.568 ± 0.053	0.167 ± 0.056
LR	1.223 ± 0.068	2.381 ± 0.166	1.542 ± 0.054	0.194 ± 0.055

Taula 2. Mitjana i desviació estàndard de les mètriques de regressions sobre folds.

Com es pot veure a la Taula 1 i 2, RF Classifier té una *accuracy* major especialment comparat amb LR i RF Regressor, en canvi LR y RF Regressor té una millor R2 i MSE (això significa que quan s'equivoquen ho fan a menys distancia), però RF Classifier està a prop, encara que té una MSE bastant més gran. En canvi els DT

s'aproximen a RF Classifier en *accuracy*, té el millor *F1 Score* però té errors molt grans i llunyans quan s'equivoca.

També s'observa que tots els models tenen desviacions estàndards similars, relativament petites, encara que hi ha diferència. Son bastant variades a les diferents mètriques, però podem observar que òbviament, el LR te molta més variació a les mètriques de classificacions que a les de regressions i RF Classifier l'oposat.

Encara així, la prioritat es la major *accuracy* possible, es millor no equivocar-se en les notes previstes. La major *accuracy* es donada pel RF Classifier, amb una desviació no significativament gran.

Model	Accuracy	F1 Score
RF Classifier	83.5% \pm 0.6%	0.475 \pm 0.013

Taula 3. Mitjana i desviació estàndard de les mètriques de classificacions de RF Classifier sobre folds.

Model	MAE	MSE	RMSE	R2
RF Classifier	1.121 \pm 0.055	3.535 \pm 0.221	1.879 \pm 0.06	-0.197 \pm 0.075

Taula 4. Mitjana i desviació estàndard de les mètriques de regressions de RF Classifier sobre folds.

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

Com es pot veure als resultats de la mitjana de les mètriques del model RF sobre 5 *K-Folds* de la base de dades-completa, té resultats molt similars als anteriors, inclús millorant-los tots, disminuint el marge d'error. Encara així el resultats no son molt positius, ja que falla bastant.

7. RECOMANADOR

Per poder fer un recomanador de com millorar el rendiment acadèmic es necessitar utilitzar aquells paràmetres que son possibles de canviar pel estudiant, no es de esperar que l'estudiant es pugui moure d'on viu a un altre ni canviar quin son els seus pares. Això redueix bastant el espai possible de la base de dades, de 51 paràmetres a 15.

El recomanador agafa per cada paràmetre, tots els possibles valors i mira com varia la predicció del classificador amb cada possibilitat. Al final guardar el valor de quin paràmetre dona el millor rendiment.

Hi ha situacions especials, hi ha estudiants a la base de dades dels quals no tenim valors del paràmetres modificables, i una altre gran part que ja posseeixen el màxim rendiment possible, i dels restants, amb els pocs paràmetres modificables, i, lo poc específic que són les opcions ("poc", "molt", etc.), el recomanador no sembla capaç de donar molt benefici tal com es veu a la

Es va provar a fer un recomanador que estudiants similars en clústers de canvis. I llavors preveia a quin clúster de canvis el nou estudiant pertany i els recomanava aquests canvis. S'ha descartat aquest model ja que feia prediccions que empitjoraven els resultats segons el predictor.

Com podem observar, dels estudiants que es poden

Resultat Recomanador	Percentage
Tenen ja el millor rendiment	45.9%
No tenen parametres que es poden modificar	44.6%
Millora	1.0%
Es manté	8.5%
Empitjora	0.0%

Taula 5. Resultats del recomanador del model sobre la base de dades.

millor els seu rendiment (9.5% del total dels estudiants), només s'obté una millorar en el 10.5% d'aquest estudiants (1% del total dels estudiants) comparat amb el 89.5% que es manté (8.5% del total dels estudiants), això es degut a que el classificador, tal i com es veu a la Fig. 3, aquells paràmetres de les dades utilitzades, que s'han considerat alterables per un estudiant (en vermell a la Fig. 3), i per tant, que el recomanador intenta modificar, tenen molt poc pes, casi nul, en la predicció que fa el model utilitzar.



Fig. 3. Gràfic importància dels diferents paràmetres d'un estudiant en la predicció del model, comparant els classificats com fàcilment modificables per un estudiant.

8. APLICATIU INTERACTIU

Al final, donades les complicacions i endarreriments s'ha canviat la metodologia de desenvolupament de *Test-driven Development* a *Rapid application development*, basat en iterativament desenvolupar l'aplicació afegint noves funcionalitats i polint usabilitat de manera que sempre sigui funcional l'aplicatiu i poc a poc vagi tendint al disseny final.

Inicialment es va desenvolupar utilitzant implementació *HTTP* nativa de *Python*, que es a bastant baix nivell i requeria molt de codi i treball que no donaria temps així que ràpidament es va canviar a *Python Flask*, *HTML*, *Bootstrap* i *Alpine.js* una pàgina web simple formulari per emplenar els paràmetres que reb el model i cridar a l'API que amb aquest feia una predicció i retornava el resultat. Utilitzant un fitxer *JSON* com a base de dades per saber quins paràmetres, amb quines opcions, rangs i tipologia reb el model.

Es va iterar sobre aquesta arquitectura afegint a la part del servidor més pàgines web que mostrin el resultat i permetin calcular les recomanacions i la part de l'API per calcular les recomanacions, verificar els paràmetres d'entrada que dona l'usuari son vàlids.

Després es va afegir la capacitat al servidor d'entrenar el propi model amb la base de dades *csv* que es va fer a

l'apartat 5, sense tenir que donar el model ja entrenat, a més d'una pagina d'inici HTML que explica les coses.

Es va afegir un *Docker* [30] per poder desplegar el servidor a *Digital Ocean* [31] per veure que funciona i es pot accedir.

Mentre es va fer aquest treball i va ser presentat el servidor estava disponible per us a la direcció IP "http://129.212.141.54/".

Com es de recerca acadèmica no s'ha afegit cap Proxy, mesures de seguretat complexes, *load balancers*, dominis, o similar, ja que es un simple servidor amb una IP per poder accedir i veure que funciona a petita escala.



Fig. 4. Pàgina principal de l'Aplicatiu.

Fig. 5. Pàgina del formulari de l'Aplicatiu.



Fig. 6. Pàgina dels resultats i recomanacions de l'Aplicatiu.

9. 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 ANN amb múltiple *losses* que maximitzin ambdues mètriques es possible que superi al RF.

Es necessitaria un base de dades molt més 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). Hi ha poques aparicions de valors d'alguns paràmetres i poques observacions en general. 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.

També seria molt millor tenir mes paràmetres amb dades més concretes, hi havia moltes de les dades classificades categòricament en "poc", "mitja" i "molt" o similar que podrien aportar més informació amb valors concrets o per exemple paràmetres que podrien estar mes extensos, como dades d'estudi setmanal per àrea no solament general, o notes per àrees i no tan generals, o més concret quines activitats extra curriculars es duen a terme, esports, instruments, escacs, voluntariat, etc.

La versió del aplicatiu es funcional, necessites canvis d'aspecte visual i presentació per no sobre carregar tant als usuaris, però amb el disseny modular que té, seria relativament senzill afegir nou camps per introduir dades, afegir una base de dades SQL per guardar noves entrades i variables i es podria utilitzar per crear la base de dades que s'ha comentat per poder continuar el estudi i desenvolupar models millors, més precisos i, inclús, models personalitzats a cada estudiant que aprenguin de notes anterior partint d'un model generalitzat.

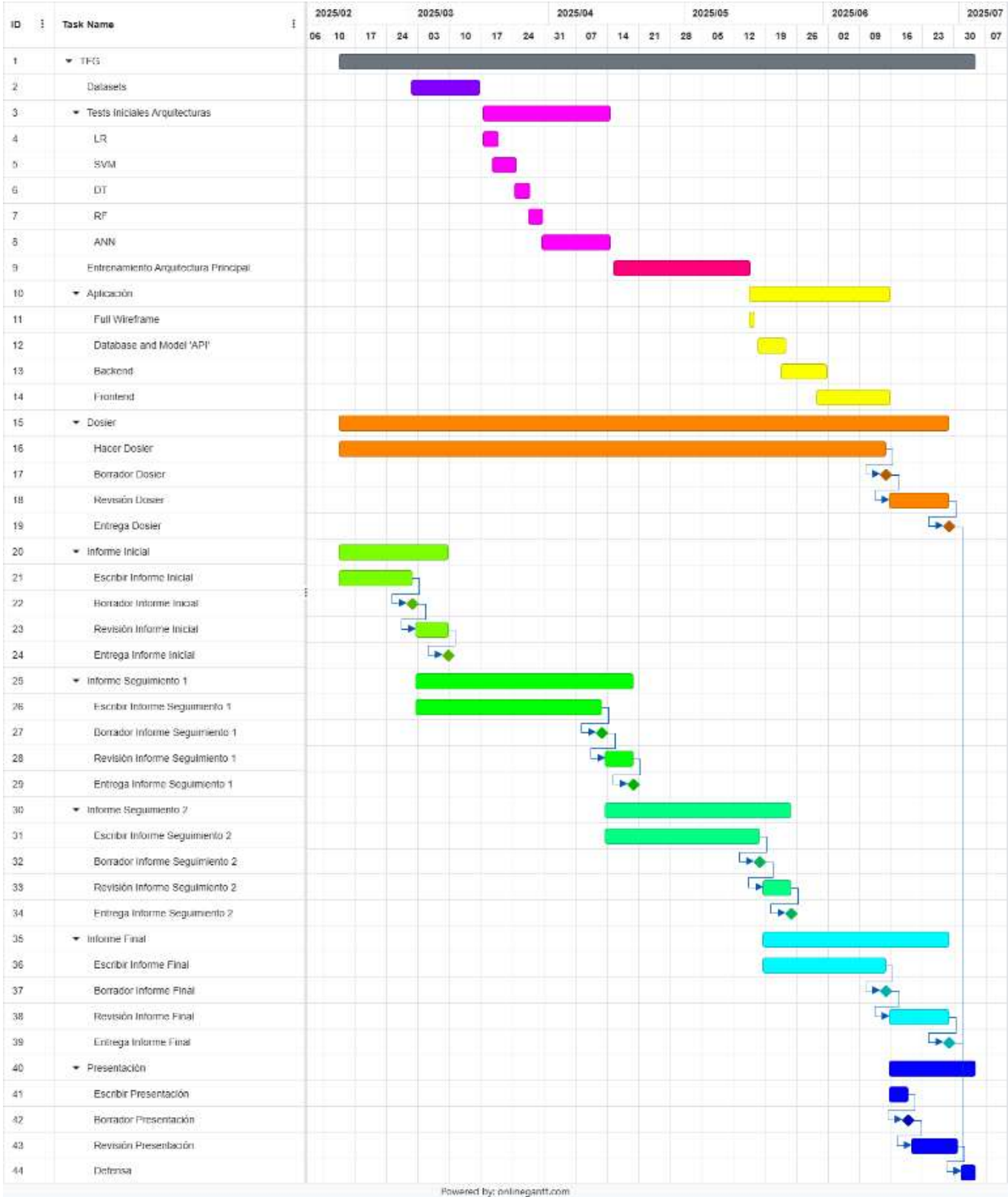
En resum una base de dades molt més extensiva, completa i amb més paràmetres relacionats amb els estudis i no solament ambient familiar, situació econòmica i estudis generals milloraria molt els resultats, models i recomanador.

BIBLIOGRAFIA

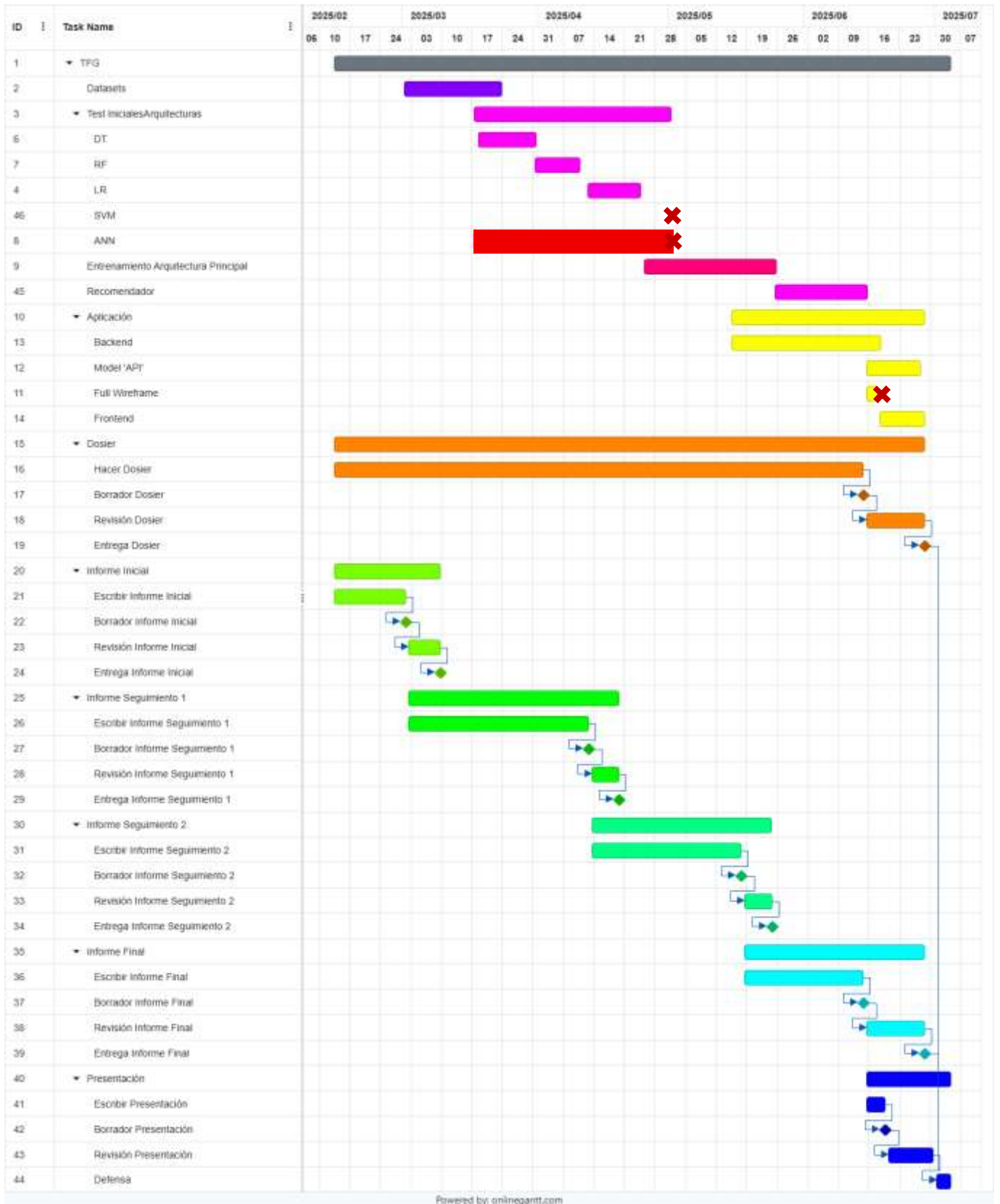
- [1] M. V. Martins, D. Toledo, J. Machado, L. M. T. Baptista, and V. Realinho, "Early Prediction of student's Performance in Higher Education: A Case Study", in Trends and Applications in Information Systems and Technologies, 2021, pp. 166–175. doi: https://doi.org/10.1007/978-3-030-72657-7_16.
- [2] 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. Thammassiri, 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/C5TG7T>.
 - [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>
 - [30] Docker Inc. Docker. (2025). Docker. Available: <https://www.docker.com>
 - [31] DigitalOcean, LLC. DigitalOcean. (2025). DigitalOcean. Available: <https://www.digitalocean.com>

APÈNDIX A: PLANIFICACIÓ



Planificació que es va fer a l'inici del projecte de la repartició de treball.



Powered by: onlinegantt.com

Progres que realment a seguit el projecte dur a complicacions i similar.