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

Març de 2025, Escola d’Enginyeria (UAB)

————————————————

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

—————————— ◆ ——————————

# 1 Introducció

A

QUEST 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 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 utilitzats ha sigut la L*ogistic 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’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%.

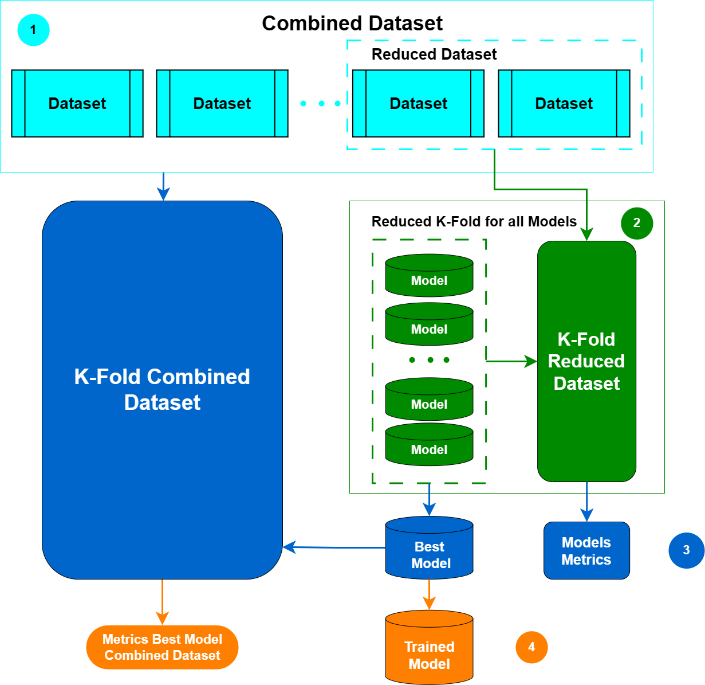


Fig. 2. Diagrama de la metodologia a seguir per decidir el millor model per l’aplicatiu.

## 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.

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’obtindran 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 esplaiat 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.

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-Fols* 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

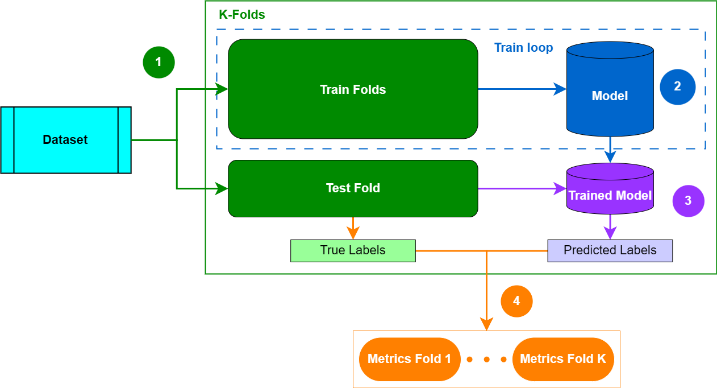


Fig. 1. Diagrama de la metodologia general *K-Fold­* per l’entrenament dels models.

Per dur a terme el projecte de forma correcta i a tems 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 prepara 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.



Fig. 1. Magnetization as a function of applied field. Note that “Fig.” is abbreviated. There is a period after the figure number, followed by one space. It is good practice to briefly explain the significance of the figure in the caption.

# 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 d0una 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.

Totes aquestes operacions i transformacions de la base de dades les durem 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 10 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*.

# 5 Equations

If you are using Word, use the MathType add-on (<http://www.mathtype.com>) for equations in your paper (Insert | Object | Create New | Microsoft Equation or MathType Equation). “Float over text” should not be selected.

For display equations as seen below, number equations consecutively with equation numbers in parentheses flush with the right margin, as in (1). First, use the equation editor to create the equation. Then, select the “Equation” markup style. Press the tab key and write the equation number in parentheses. To make your equations more compact, you may use the solidus ( / ), the exp function, or appropriate exponents. Use parentheses to avoid ambiguities in denominators. Punctuate equations when they are part of a sentence, as in

 (1)

Be sure that the symbols in your equation have been defined before the equation appears or immediately following. Italicize symbols (*T* might refer to temperature, but T is the unit tesla). Per IEEE Computer Society, please refer to “(1),” not “Eq. (1)” or “equation (1),” except at the beginning of a sentence: “Equation (1) shows ... .” Also see *The Handbook of Writing for the Mathematical Sciences*, 1993. Published by the Society for Industrial and Applied Mathematics, this handbook provides some helpful information about math typography and other stylistic matters.

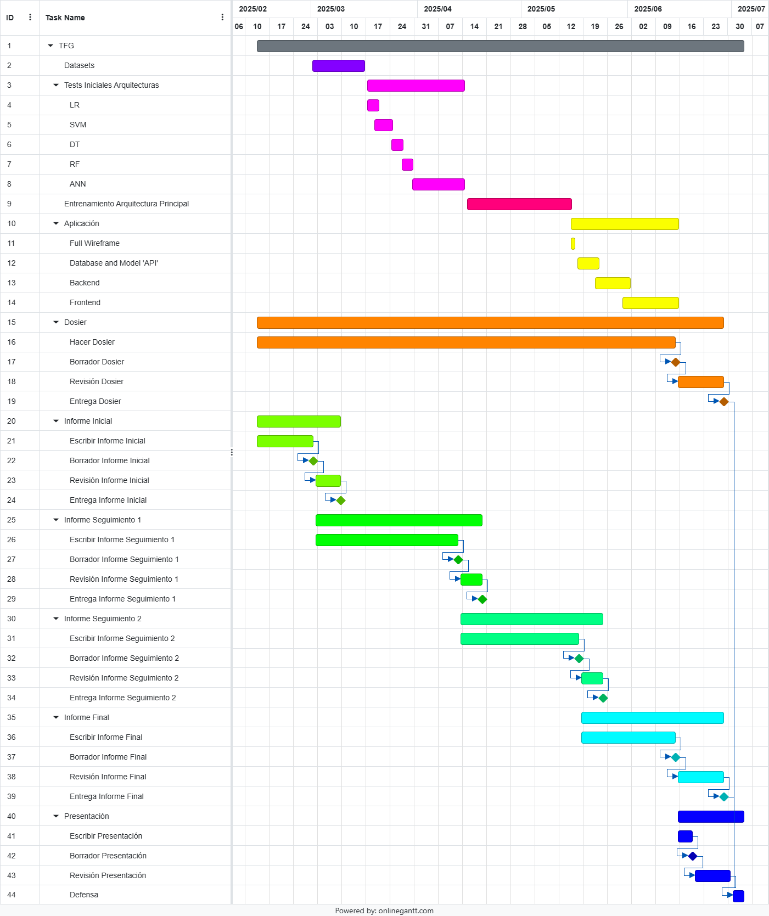


Fig. 3. Diagrama de Gant amb la planificació del projecte fins la presentació final.

Please note that math equations might need to be reformatted from the original submission for page layout reasons. This includes the possibility that some in-line equations will be made display equations to create better flow in a paragraph. If display equations do not fit in the two-column format, they will also be reformatted. Authors are strongly encouraged to ensure that equations fit in the given column width.

# 6 Helpful Hints

## 6.1 Figures and Tables

Because IEEE Computer Society staff will do the final formatting of your paper, some figures may have to be moved from where they appeared in the original submission. Figures and tables should be sized as they are to appear in print. Figures or tables not correctly sized will be returned to the author for reformatting.

We strongly encourage authors to carefully review the material posted here to avoid problems with incorrect files or poorly formatted graphics.

Place figure captions below the figures; place table titles above the tables. Figure captions appear as left justified. Table captions are restricted to one sentence and are formatted as title case. Any additional sentence in a table caption will be formatted as a footnote below the table (see Table 1 in this document). If your figure has two parts, include the labels “(a)” and “(b)” as part of the artwork. Please verify that the figures and tables you mention in the text actually exist. Figures and tables should be called out in sequential order, as this is how they will be placed in your paper. For example, avoid referring to figure “8” in the first paragraph of the article unless figure 8 will again be referred to after the reference to figure 7. **Please do not include figure captions as part of the figure. Do not put captions in “text boxes” linked to the figures. Do not put borders around the outside of your figures.** Per IEEE Computer Society, please use the abbreviation “Fig.” even at the beginning of a sentence. Do not abbreviate “Table.” Tables are numbered numerically.

For journals that use print for publication, please verify with IEEE Computer Society that the journal you are submitting to does indeed accept color before submitting final materials. **Do not use color unless it is necessary for the proper interpretation of your figures.**

Figures (graphs, charts, drawing or tables) should be named fig1.eps, fig2.ps, etc. If your figure has multiple parts, please submit as a single figure. Please do not give them descriptive names. Author photograph files should be named after the author’s LAST name. Please avoid naming files with the author’s first name or an abbreviated version of either name to avoid confusion. If a graphic is to appear in print as black and white, it should be saved and submitted as a black and white file (grayscale or bitmap.) If a graphic is to appear in color, it should be submitted as an RGB color file.

Figure axis labels are often a source of confusion. Use words rather than symbols. As an example, write the quantity “Magnetization,” or “Magnetization *M*,” not just “*M*.” Put units in parentheses. Do not label axes only with units. As in Fig. 1, for example, write “Magnetization (A/m)” or “Magnetization (Am−1),” not just “A/m.” Do not label axes with a ratio of quantities and units. For example, write “Temperature (K),” not “Temperature/K.” Table 1 shows some examples of units of measure.

Multipliers can be especially confusing. Write “Magnetization (kA/m)” or “Magnetization (103 A/m).” Do not write “Magnetization (A/m) × 1,000” because the reader would not know whether the top axis label in Fig. 1 meant 16,000 A/m or 0.016 A/m. Figure labels should be legible, approximately 8 to 12 point type. When creating your graphics, especially in complex graphs and charts, please ensure that line weights are thick enough that when reproduced at print size, they will still be legible. We suggest at least 1 point.

## 6.3 Footnotes

Number footnotes separately in superscripts (Insert | Footnote)[[1]](#footnote-1). Place the actual footnote at the bottom of the column in which it is cited; do not put footnotes in the reference list (endnotes). Use letters for table footnotes (see Table 1). Please do not include footnotes in the abstract and avoid using a footnote in the first column of the article. This will cause it to appear above the affiliation box, making the layout look confusing.

TABLE 1  
Units for Magnetic Properties



Statements that serve as captions for the entire table do not need footnote letters.

aGaussian units are the same as cgs emu for magnetostatics; Mx = maxwell, G = gauss, Oe = oersted; Wb = weber, V = volt, s = second, T = tesla, m = meter, A = ampere, J = joule, kg = kilogram, H = henry.

## 6.4 Lists

The IEEE Computer Society style is to create displayed lists if the number of items in the list is longer than three. For example, within the text lists would appear 1) using a number, 2) followed by a close parenthesis. However, longer lists will be formatted so that:

1. Items will be set outside of the paragraphs.
2. Items will be punctuated as sentences where it is appropriate.
3. Items will be numbered, followed by a period.

## 6.5 Theorems and Proofs

Theorems and related structures, such as axioms corollaries, and lemmas, are formatted using a hanging indent paragraph. They begin with a title and are followed by the text, in italics.

**Theorem 1.** *Theorems, corollaries, lemmas, and related structures follow this format. They do not need to be numbered, but are generally numbered sequentially.*

Proofs are formatted using the same hanging indent format. However, they are not italicized.

**Proof.** The same format should be used for structures such as remarks, examples, and solutions (though these would not have a Q.E.D. box at the end as a proof does).

# 7 End Sections

## Appendices

Appendices, if present, appear online as supplemental material. In the event multiple appendices are required, they will be labeled “Appendix A,” “Appendix B, “ etc.

IEEE Computer Society Transactions accepts supplemental materials for review with regular paper submissions. These materials may be published on our Digital Library with the electronic version of the paper and are available for free to Digital Library visitors. Please see our guidelines below for file specifications and information. Any submitted materials that do not follow these specifications will not be accepted. All materials must follow US copyright guidelines and may not include material previously copyrighted by another author, organization or company.

## 7.2 Acknowledgments

The preferred spelling of the word “acknowledgment” in American English is without an “e” after the “g.” Use the singular heading even if you have many acknowledgments. Avoid expressions such as “One of us (S.B.A.) would like to thank ... .” Instead, write “F. A. Author thanks ... .” Sponsor and financial support acknowledgments are included in the acknowledgment section. For example: This work was supported in part by the US Department of Commerce under Grant BS123456 (sponsor and financial support acknowledgment goes here). Researchers that contributed information or assistance to the article should also be acknowledged in this section. Also, if corresponding authorship is noted in your paper it will be placed in the acknowledgment section. Note that the acknowledgment section is placed at the end of the paper before the reference section.

## 7.3 References

Unfortunately, the Computer Society document translator cannot handle automatic endnotes in Word; therefore, type the reference list at the end of the paper using the “References” style.

The order in which the references are submitted in the manuscript is the order they will appear in the final paper, i.e., references submitted nonalphabetized will remain that way.

Please note that the references at the end of this document are in the preferred referencing style. Within the text, use “et al.” when referencing a source with more than three authors. In the reference section, give all authors’ names; do not use “et al.” Do not place a space between an authors' initials. Papers that have not been published should be cited as “unpublished” [4]. Papers that have been submitted or accepted for publication should be cited as “submitted for publication” [5]. Please give affiliations and addresses for personal communications [6].

Capitalize all the words in a paper title. For papers published in journals not published in English, please give the English citation first, followed by the original foreign-language citation [7].

# Conclusió

Although a conclusion may review the main points of the paper, do not replicate the abstract as the conclusion. A conclusion might elaborate on the importance of the work or suggest applications and extensions. Authors are strongly encouraged not to reference multiple figures or tables in the conclusion—these should be referenced in the body of the paper.

**Agraïments**

The authors wish to thank A, B, C. This work was supported in part by a grant from XYZ.

**Bibliografia**

1. M. V. Martins, D. Tolledo, 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. Yılmaz 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. *PostgresSQL*. (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. Skcit-learn. Skcit-learn. (2025). Skcit-learn. Available: https://scikit-learn.org/stable
29. Pytoech. The Linux Fundation. (2025). PyTorch. Available: https://pytorch.org

1. It is recommended that footnotes be avoided (except for the unnumbered footnote with the receipt date on the first page). Instead, try to integrate the footnote information into the text. [↑](#footnote-ref-1)