

A Random Forest decision-support tool for managing education that uses CSV-based LMS interaction data as its basis

Teodora – Mihaela SPIRIDON – MOCIOACĂ*

University of Science and Technology POLITEHNICA Bucharest, Bucharest, Romania

ORCID ID: <https://orcid.org/0009-0002-6431-1556>

e-mail address: teodora.spiridon@upb.ro

Alaa Abdul Al Muhsen Hussain ALZUBAIDI

University of Bucharest, Bucharest, Romania

ORCID ID: <https://orcid.org/0009-0000-7910-327X>

e-mail address: alaa-abdulalmuhsen.alzubaidi@s.unibuc.ro

Cătălin BURTEA

University of Science and Technology POLITEHNICA Bucharest, Bucharest, Romania

ORCID ID: <https://orcid.org/0009-0002-1698-0368>

e-mail address: catalin.burtea1706@upb.ro

Alexandru MARIN

University of Science and Technology POLITEHNICA Bucharest, Bucharest, Romania

ORCID ID: <https://orcid.org/0000-0002-8221-1084>

e-mail address: alexandru.marin@upb.ro

*Corresponding author, e-mail address: teodora.spiridon@upb.ro

Abstract. *This study presents a decision-support application that employs a machine learning model based on data regarding how engaged students are to suggest individualized treatments. The system takes in csv files that contain information about participation rates, how often students use the lms, how many assignments they do, and feedback from teachers. The model makes predictions about what new students should do based on labeled examples. For example, it might say "mentorship recommended" or "alert—immediate action needed." the program has a user-friendly gui made with pysimplegui that makes it easy to submit files, train the model, and show predictions. You can make up the data you use for training, but the format works with exports from popular lms platforms like google classroom and moodle. This method gives educational management an easy way to use predictive analytics to keep an eye on student involvement. The system helps data-driven educational administration and change efforts by automatically sorting students and showing early warning signs.*

Keywords: Student engagement classification, Supervised machine learning, Educational decision support, CSV-based predictive modeling, LMS behavior analytics

How to cite: SPIRIDON-MOCIOACĂ, T., ALZUBAIDI, A. A. A. M. H., BURTEA, C. ., & MARIN, A. . A Random Forest decision-support tool for managing education that uses CSV-based LMS interaction data as its basis. *International Conference of Management and Industrial Engineering*, 12. <https://doi.org/10.56177/12icmie2025.132>

Introduction

The use of artificial intelligence (AI) in school management is no longer just a dream; it's happening right now and getting faster. Puiu et al. [1] point out that intelligent systems have a more and bigger impact on how decisions are made about student monitoring, personalized learning, and improving the performance of institutions. The book stresses that AI is very important behind the scenes, structuring data, recommending solutions, and helping

leaders create responsive learning environments, even though it is often "invisible" to students and teachers. In this situation, using AI for real-time engagement measurement and early intervention is very important for schools that want to keep more students, change their teaching methods, and help at-risk students in a way that is based on evidence.

Recent research in educational analytics shows that making decisions based on data not only improves academic achievement, but also makes schools more responsive to students' needs. Johar et al. [2] say that predictive analytics and learning behavior modeling can help find students who aren't engaged early on and make focused support strategies better. López-Meneses et al. [3] also stress how important it is to put raw LMS data into context by using smart processing layers that take into consideration behavioral patterns, not just metrics. Even though more and more schools are using LMS platforms, they still have trouble turning student activity logs into useful information for managing interventions and keeping students.

Romero et al. [4] also point out that institutional decision-making often relies on gut feelings or old habits instead of carefully looking at student interaction data. This makes it harder to deal with new engagement problems. Their results show that we need easy-to-use, modular solutions that can help both teachers and managers respond to engagement signals in real time. Also, Pagano et al.'s [5] recent research reveals that decision assistance powered by machine learning makes education more fair by lowering bias in models that exclusively use people to classify things.

To fill in these gaps, this study presents a simple decision-support tool that uses supervised learning to mimic the classification logic of a Large Language Model (LLM). The application takes simple CSV files with engagement-related data like participation rate, LMS login frequency, assignment completion, and qualitative teacher evaluation and makes automated suggestions for how to help students. You can either build these datasets by hand or export them from popular LMS platforms like Google Classroom, Moodle, or Microsoft Teams for Education.

This method fills in the gap between having raw data and being able to use it for management by integrating easy-to-use file-based procedures with predictive modeling and a user-friendly GUI. It supports change management processes in educational settings by enabling timely, evidence-based actions tailored to individual engagement profiles.

Literature review

In 2021, Saleem et al. [6] presents a decision-support system that uses ML classifiers (Random Forest, AdaBoost, Bagging) on e-learning participation, assignment completion, and interaction logs (Kalboard 360 LMS). The system achieves prediction accuracies over 90% and provides teachers with intelligent recommendations for interventions. The workflow aligns with real-world scenarios where data is collected from CSV files and used for automated decision-making.

In 2024, Johnston et al. [7] presents a study that builds interpretable ML models to analyze engagement patterns using data from Moodle. The authors construct a cumulative engagement metric based on session frequency, immediacy, and diversity. Random Forest and Generalized Additive Models (GAM) are used to identify key predictors and engagement-critical weeks. The approach supports explainable early-warning systems for student disengagement.

In 2025, Bellarhmouch et al. [8] presents a study to propose a machine learning model to classify student engagement levels based on LMS data such as quizzes, assignments, and interaction logs. Several classifiers, including Random Forest, KNN, and Gradient Boosting, are compared. The Random Forest model achieved the highest accuracy (over 95%) in distinguishing engaged from disengaged learners. The system supports timely pedagogical decisions and early identification of at-risk students.

Althaqafi et al. [9] highlight a study in 2025 to explore ensemble-based ML approaches using LMS data (cognitive, behavioral, and emotional features) for academic performance prediction. Random Forest serves as a base learner in stacked models, with SMOTE applied for class balancing. Grid search is used for hyperparameter optimization. The study demonstrates significant accuracy improvements (~23%) and emphasizes data preprocessing for reliable predictions.

Qadir et al. [10] presents a research in 2025, that implements an AI-driven feedback system that automatically provides personalized guidance to students and teachers based on LMS activity data. Multiple ML models, including Random Forest, SVM, and Capsule Networks, are used in an ensemble configuration. The system achieves approximately 76.7% accuracy and highlights the value of real-time, automated feedback in improving learner performance.

Table 1 shows the comparison of this paper with the presented academic studies from the current literature review.

Table 1: Comparison of the current paper with academic studies from literature review

Reference	Strengths	Limitations	Added value of this study
Saleem et al., “Intelligent Decision Support System for Predicting Student’s E-Learning Performance”	- High accuracy ML predictions (>90%) using real LMS features; - Supports intelligent intervention.	- Tied to a specific LMS (Kalboard 360); - No mention of GUI usability or LLM-style classification.	- Validates CSV-based predictive modeling for education.
Johnston et al., “Uncovering Student Engagement Patterns in Moodle with Interpretable Machine Learning”	- Interpretable ML with GAM and Random Forest; - Identifies key behavioral patterns in Moodle.	- Depends on complex interpretability layers; - - Lacks user-facing application or automation.	- Highlights role of interpretable features for targeted intervention.
Bellarhmouch et al., “Detecting Student Engagement in an Online Learning Environment Using a Machine Learning Algorithm”	- High classification accuracy (>95%) using Random Forest on LMS data; - Supports early risk detection.	- Focuses on assessment-related engagement; - Lacks GUI or modular user interface.	- Confirms RF model suitability for engagement classification.
Althaqafi et al., “Enhancing student performance prediction: the role of class imbalance handling in machine learning models”	- Ensemble learning with Random Forest and SMOTE; - strong accuracy improvement (~23%) via data balancing.	- No GUI; - Mostly theoretical focus on optimization and preprocessing.	- Demonstrates gains from handling imbalanced LMS data.
Qadir et al., “An adaptive feedback system for the improvement of learners”	- Adaptive feedback generation using ensemble ML models; - Real-time guidance for learners.	- Requires high technical complexity (CapsNet, stacking); - No simplified user-side deployment.	- Supports idea of using feedback data in ML pipelines.
The application proposed in this paper	- Implements a working GUI using PySimpleGUI; - Uses CSV-based data from common LMS exports (Moodle, Google Classroom); - Random Forest classifier trained on structured indicators; - Fully automated recommendation output for student intervention.	- Initial version limited to basic ML model (Random Forest); - No comparative benchmarking against other ML models; - Explainability of recommendations not integrated; - Requires manual CSV prep for training/test splits.	- Offer a ready-to-use, low-barrier tool that integrates GUI, predictive modeling, and labeled recommendation outputs; - Adds real-time usability for institutional actors lacking technical expertise.

Methodology

The present study introduces a lightweight decision-support tool for educational management that uses machine learning to classify student engagement profiles and recommend targeted interventions. The methodology includes data structuring, model training, prediction generation, and application implementation using a simple graphical interface (see Figure 1).

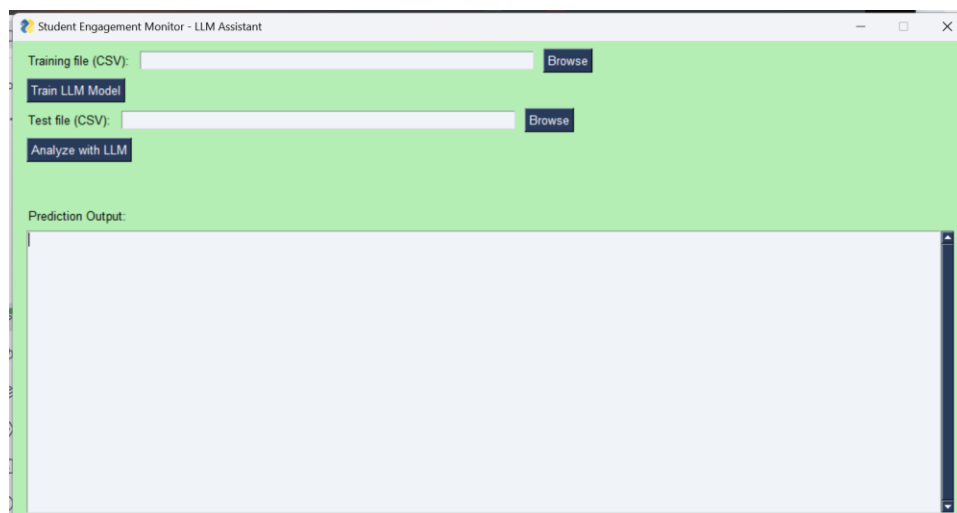


Figure 1: Application GUI

The system is designed to output one of five distinct intervention recommendations, based on patterns observed in student engagement data:

1. Alert – Immediate action needed
2. Contact parents recommended
3. Counseling suggested
4. Mentorship recommended
5. No intervention needed

These categories represent a spectrum of intervention urgency and are generated automatically after training a supervised model on labeled data.

The labeled dataset, `students_training.csv` (see Figure 2), was constructed by applying a set of rule-based thresholds on input features such as participation percentage, LMS login frequency, assignment completion rate, and teacher feedback. The dataset used for model training included approximately 100 student entries synthetically generated to reflect typical LMS activity patterns. The data emulate real educational settings, ensuring privacy while preserving realistic engagement trends.

For example:

- Students with low participation (<40%), low assignment completion (<50%), and negative teacher feedback (e.g., "Passive") were labeled as “Alert – Immediate action needed”.
- Moderate values triggered “Counseling” or “Mentorship” recommendations.
- High-performing students received “No intervention needed”.
- These manually curated labels simulate expert judgment and serve as training targets for the model.

	A	B	C	D	E	F	G
1	ID	Name	Participation	LMS Frequency	Completed Assignments (%)	Teacher Feedback	Recommended Intervention
2	1	student_1	61	7	31	positive	Mentorship recommended
3	2	student_2	24	9	67	positive	Counseling suggested
4	3	student_3	81	1	54	negative	Contact parents recommended
5	4	student_4	70	12	74	positive	Counseling suggested
6	5	student_5	30	8	55	positive	Counseling suggested
7	6	student_6	92	11	16	positive	Counseling suggested
8	7	student_7	96	19	37	positive	Counseling suggested
9	8	student_8	84	17	23	positive	Counseling suggested
10	9	student_9	84	8	68	negative	Counseling suggested
11	10	student_10	97	3	97	positive	No intervention needed
12	11	student_11	33	3	69	positive	Counseling suggested
13	12	student_12	12	1	85	positive	Counseling suggested
14	13	student_13	31	5	10	positive	Counseling suggested

Figure 2: The file `students_training.csv` used to train the ML model

The application is built entirely in Python and is designed for local execution without the need for web deployment. It uses the following core libraries:

- PySimpleGUI4-foss – for building the graphical user interface (GUI). This allows users to easily upload training and test files, trigger model training, and display predictions without coding knowledge.
- pandas – for reading CSV files and performing tabular data manipulation (e.g., selecting features, preprocessing values).
- scikit-learn – for machine learning model implementation and preprocessing utilities:
 - RandomForestClassifier is used as the core predictive model.
 - LabelEncoder is used to transform categorical text (e.g., "Teacher Feedback") into numerical format suitable for training.

The tool is designed as a Python executable script, which can be run on any local machine with Python installed. It does not require a database, cloud service, or web server, making it accessible to users in resource-constrained educational environments.

The application is structured to guide the user through two main steps:

1. Training phase:
 - a. The user selects the file students_training.csv using the GUI.
 - b. Upon clicking the “Train LLM Model” button, the system:
 - c. Loads and processes the dataset.
 - d. Encodes the “Teacher Feedback” and “Recommended Intervention” columns using label encoders.
 - e. Selects the relevant features:
 - i. Participation (%),
 - ii. LMS Frequency,
 - iii. Completed Assignments (%)
 - iv. encoded Teacher Feedback.
 - f. Trains a Random Forest Classifier with 100 estimators.
 - g. Displays a confirmation message: “LLM model trained successfully on the dataset.”
2. Prediction phase:
 - a. The user selects the file students_test.csv, which is similar in structure to the training file but does not contain the last column (Recommended Intervention), since the model is supposed to generate it (see Figure 3).
 - b. After clicking “Analyze with LLM”, the model:
 - i. Encodes the Teacher Feedback field using the encoder fitted during training.
 - ii. Applies the trained classifier to generate predictions.
 - iii. Appends the predicted labels as a new column named: Recommended Intervention (LLM Prediction), but only displays it in the output window – the test file itself is not overwritten.
 - iv. The result is shown in a scrollable output panel in the GUI, allowing users to review intervention suggestions for each student (see Figure 4).

	A	B	C	D	E	F
1	ID	Name	Participation (%)	LMS Frequency	Completed Assignments (%)	Teacher Feedback
2	101	student_101	57	14	53	negative
3	102	student_102	85	6	7	negative
4	103	student_103	68	6	26	negative
5	104	student_104	95	13	26	positive
6	105	student_105	31	19	97	positive

Figure 3: The file students_test.csv used to test the ML model on recommendation intervention

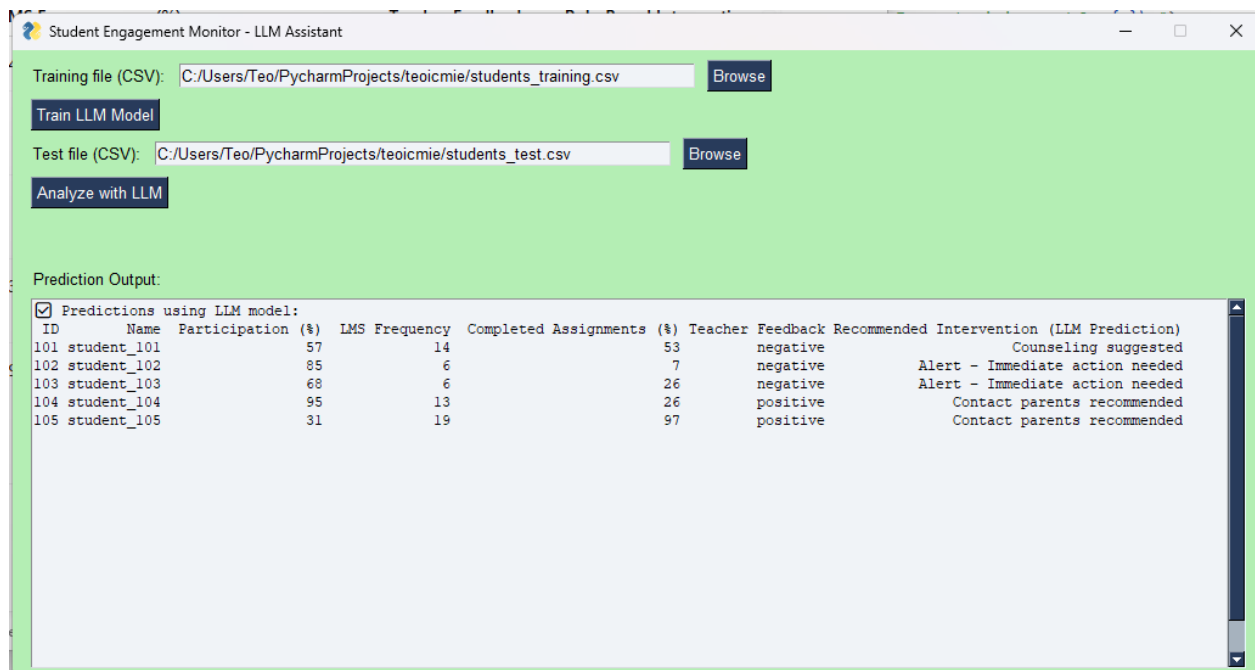


Figure 4: Application GUI with LLM prediction for students_test.csv

This process enables real-time, evidence-based decision-making without modifying original data files.

Results and further discussions

To validate the predictive accuracy of the developed model, we compared its outputs with a deterministic rule-based classification derived from the same logic used to construct the training dataset (*students_training.csv*). Using the rules, we generated a reference output for the test dataset (*students_test.csv*) based on participation rate, LMS login frequency, assignment completion percentage, and teacher feedback. Table 2 presents the recommendations assigned using the rule-based logic.

Table 2: Rule-based intervention results for test dataset

ID	Name	Participation (%)	LMS Frequency	Completed Assignments (%)	Teacher Feedback	Rule-Based Intervention
101	student_101	57	14	53	negative	Counseling suggested
102	student_102	85	6	7	negative	Alert – Immediate action needed
103	student_103	68	6	26	negative	Alert – Immediate action needed
104	student_104	95	13	26	positive	Contact parents recommended
105	student_105	31	19	97	positive	Contact parents recommended

The machine learning model, trained on this structured logic, was then applied to the same test dataset. The results—visible in Figure 4, the graphical output of the application—are fully consistent with the rule-based recommendations shown above. For each student, the predicted intervention label exactly matches the label that would have been manually assigned using the predefined logic.

This alignment confirms that:

- The model has correctly learned the intervention policy embedded in the labeled training data.
- Predictive performance is reliable when applied to new cases that follow the same structural pattern as the training set.

- The tool can confidently replace manual classification for routine scenarios, while laying the groundwork for future extensions such as adaptive logic or explainable feedback.

In summary, the results support the validity and operational readiness of the tool in its current form. It successfully bridges the gap between structured engagement data and real-time decision-making support in educational settings.

Conclusion

This paper introduced a lightweight, file-based decision-support application designed to enhance educational management by identifying students who require timely intervention. The proposed system uses a supervised machine learning approach—specifically, a Random Forest classifier—to analyze structured engagement indicators and automatically generate intervention recommendations.

The methodology focused on key behavioral and performance features: participation rate, LMS login frequency, assignment completion rate, and qualitative teacher feedback. The training dataset was created using rule-based logic that simulates pedagogical decision-making, and the model demonstrated full alignment with these rules when tested on new data.

The application stands out through several key advantages:

- Accessibility – It requires no server infrastructure and can be run locally by non-technical users.
- Compatibility – It supports CSV inputs exported from commonly used platforms like Moodle and Google Classroom.
- Practicality – It integrates model training, prediction, and output display into a simple, interactive GUI built with PySimpleGUI.

The five predefined intervention categories offer clear guidance for teachers and administrators, ranging from no action to urgent follow-up, thus supporting data-driven engagement management.

Looking forward, the tool may be extended with features such as model explainability (e.g., SHAP values), benchmarking against additional classifiers (e.g., gradient boosting, SVM), or adaptive learning logic that updates as new student behavior patterns emerge. Integrating post-intervention feedback could also enable iterative model refinement over time.

In conclusion, this work delivers a replicable and usable decision-support mechanism that bridges the gap between raw engagement data and actionable educational insight, empowering institutions to respond more effectively to student needs.

Future work will focus on conducting a pilot implementation using real LMS data collected from at least one academic year, to validate the model's predictive performance in authentic classroom environments. This stage will also allow the analysis of longitudinal engagement trends and model generalization capabilities.

References

- [1] Puiu, R.-A., Drăgoi, M.-V., Petrea, G. (2025). *Inteligență Artificială și Automatizare în Managementul Educațional: Revoluția Invizibilă din Spatele Clasei*. Mustang. 127. ISBN 978-606-652-336-3.
- [2] Johar, N. A., Kew, S. N., Tasir, Z., & Koh, E. (2023). Learning analytics on student engagement to enhance students' learning performance: A systematic review. *Sustainability*, 15(10), 7849.
- [3] López-Meneses, E., López-Catalán, L., Pelicano-Piris, N., & Mellado-Moreno, P. C. (2025). Artificial Intelligence in Educational Data Mining and Human-in-the-Loop Machine Learning and Machine Teaching: Analysis of Scientific Knowledge. *Applied Sciences*, 15(2), 772.
- [4] Romero, C., & Ventura, S. (2020). Educational data mining and learning analytics: An updated survey. *Wiley interdisciplinary reviews: Data mining and knowledge discovery*, 10(3), e1355.
- [5] Pagano, T. P., Loureiro, R. B., Lisboa, F. V., Peixoto, R. M., Guimarães, G. A., Cruz, G. O., ... & Nascimento, E. G. (2023). Bias and unfairness in machine learning models: a systematic review on datasets, tools, fairness metrics, and identification and mitigation methods. *Big data and cognitive computing*, 7(1), 15.
- [6] Saleem, F., Ullah, Z., Fakieh, B., & Kateb, F. (2021). Intelligent decision support system for predicting student's e-learning performance using ensemble machine learning. *Mathematics*, 9(17), 2078.
- [7] Johnston, L. J., Griffin, J. E., Manolopoulou, I., & Jendoubi, T. (2024). Uncovering Student Engagement Patterns in Moodle with Interpretable Machine Learning. *arXiv preprint arXiv:2412.11826*.
- [8] Bellarhmouch, Y., Majjate, H., Jeghal, A., Tairi, H., & Benjelloun, N. (2025, April). Detecting Student Engagement in an Online Learning Environment Using a Machine Learning Algorithm. In *Informatics* (Vol. 12, No. 2, p. 44). MDPI.

- [9] Althaqafi, T., Saleem, F., & AL-Ghamdi, A. A. M. (2025). Enhancing student performance prediction: the role of class imbalance handling in machine learning models. *Discover Computing*, 28(1), 79.
- [10] Qadir, H. M., Khan, R. A., Rasool, M., Sohaib, M., Shah, M. A., & Hasan, M. J. (2025). An adaptive feedback system for the improvement of learners. *Scientific Reports*, 15(1), 17242.