

AI-Assisted Multi-Subject Student Performance Prediction System Using Machine Learning and Intelligent Academic Recommendation

Prajwal Shawnak V | Prof.Satheesh Kumar G | Mrs. Padmini C

Department of Computer Science and Engineering

City Engineering College

(Affiliated to Visvesvaraya Technological University, Belagavi)

Bengaluru, Karnataka, India

prajwalsv123@gmail.com | satheesh.kumar@cityengineeringcollege.ac.in | PSHarini0201@gmail.com

Abstract—Educational institutions continuously seek efficient methods to monitor student performance and provide timely academic guidance. Conventional evaluation methods rely largely on manual analysis of examination scores, which makes it difficult to identify students who require additional academic support before outcomes become irreversible. This paper presents an AI-assisted Student Performance Prediction System that integrates supervised machine learning with a generative-AI academic advisor to forecast academic outcomes and generate individualized recommendations. Unlike earlier systems that analyze only one or two subjects in isolation, the proposed system jointly considers six core Computer Science subjects — Data Structures, Database Management Systems, Operating Systems, Computer Networks, Python Programming, and Artificial Intelligence — together with Reading Score, Writing Score, daily Study Hours, Gender, and Course Preparation Status. Three supervised classifiers, namely Decision Tree, Random Forest, and Logistic Regression, were trained and compared on a dataset of 1000 student records (800 training, 200 testing) to classify students into PASS or FAIL categories. Logistic Regression achieved the highest test accuracy of 99.50%, followed by Random Forest at 97.50% and Decision Tree at 95.50%. Feature-importance analysis further revealed that Study Hours is the single most influential predictor, accounting for approximately 44.3% of the Random Forest model's decision weight. Beyond classification, an AI-powered academic advisor built on the Llama 3.2 large language model, served locally through Ollama, converts the numeric prediction into a personalized performance summary, topic-level improvement suggestions for each weak subject (e.g., specific chapters or concepts to revisit rather than generic advice), and a structured study plan. The platform further incorporates a Teacher's Suggestion module through which an instructor's free-text remarks are captured and merged into the same report alongside the AI-generated content, combining automated analytics with human-in-the-loop academic judgment. The system compiles the prediction, the topic-level guidance, and the teacher's remarks into a downloadable PDF report. The complete pipeline is implemented as a working Streamlit prototype using Python, Scikit-learn, and Ollama, featuring an interactive input form, a prediction-confidence gauge, an academic radar chart, and an automatically generated PDF report. Experimental results confirm that combining interpretable predictive analytics with generative-AI feedback yields both high classification accuracy and actionable, human-readable guidance, making the system practical for adoption by academic institutions and individual educators alike.

Keywords—Machine Learning, Random Forest, Logistic Regression, Student Performance Prediction, Artificial Intelligence, Educational Data Mining, Academic Recommendation, Streamlit, Large Language Model.

I. INTRODUCTION

Education systems worldwide are increasingly adopting Artificial Intelligence and Machine Learning to improve learning outcomes and institutional decision-making. Student performance prediction has emerged as an important research area because early identification of academically weak students allows faculty to intervene before a semester or examination cycle concludes, rather than reacting after final results are published.

Most existing systems predict student success using a limited number of attributes, such as attendance percentage or a single subject's marks, and typically stop at producing a numeric prediction. Such systems rarely explain why a student is at risk or what specific action would help, leaving the interpretive and advisory work entirely to the instructor.

This work introduces an AI-assisted Student Performance Prediction System that goes beyond binary classification. The system (i) predicts PASS/FAIL outcomes from six core Computer Science subjects together with behavioral and demographic indicators, and (ii) passes the prediction and underlying scores to a locally hosted Large Language Model that generates a personalized academic narrative — strengths, weaknesses, and a concrete study plan. This combination of predictive analytics and generative AI is intended to support both students and educators in making faster, more informed academic decisions.

II. PROBLEM STATEMENT

Many educational institutions continue to rely on manual evaluation processes that primarily record marks without deriving predictive or diagnostic insight from them. These processes exhibit several structural limitations:

- 1) They report only historical marks and do not forecast future performance.
- 2) They cannot reliably flag academically at-risk students before final examinations.
- 3) They provide no personalized or subject-specific academic guidance.
- 4) They depend on considerable manual effort from faculty to review each student individually, which does not scale as cohort sizes grow.

These limitations motivate the development of an automated, multi-subject prediction system capable of both classifying performance and generating intelligible, actionable guidance without requiring per-student manual review.

III. OBJECTIVES

- 1) Predict student academic performance (PASS/FAIL) using supervised Machine Learning.
- 2) Jointly analyze six core Computer Science subjects rather than a single subject in isolation.
- 3) Identify each student's subject-wise strengths and weaknesses through feature-level analysis.
- 4) Generate AI-based, topic-level academic recommendations (specific chapters/concepts) using a Large Language Model.
- 5) Provide a Teacher's Suggestion module allowing instructors to append free-text remarks to the report.
- 6) Produce a downloadable, structured PDF performance report for each student.
- 7) Assist faculty in identifying at-risk students early in the academic cycle.
- 8) Improve the overall efficiency and consistency of academic decision-making.

IV. LITERATURE REVIEW

A substantial body of research has applied machine learning to student performance prediction. Comparative studies report that tree-ensemble methods, particularly Random Forest, are consistently competitive against Decision Tree, Support Vector Machine, k-Nearest Neighbors, and Artificial Neural Network baselines across school-level, college-level, and e-learning datasets [1]–[4], [7]–[10]. Random Forest has additionally been shown to perform strongly on regression formulations of the problem, with feature-selection studies identifying prior grades and enrolled course load as dominant predictors [3]. Gradient-boosting variants have occasionally outperformed Random Forest on specific feature sets that emphasize behavioral variables such as parental support and weekly study time [6].

Several survey and review papers have consolidated this literature. Zhang et al. [15] partition the student performance prediction (SPP) task into five stages — data collection, problem formalization, modeling, prediction, and application — and note that most prior work concentrates on the modeling stage while giving comparatively little attention to actionable application. Broader systematic reviews of Educational Data Mining echo this observation, reporting that the great majority of surveyed studies stop at producing a prediction or risk score, without generating any form of personalized corrective guidance for the learner [16], [17].

A separate and more recent stream of research explores Large Language Models (LLMs) as feedback generators in

education. Studies report that LLM-generated feedback is frequently perceived by students as constructive and, in blind comparisons, difficult to distinguish from feedback authored by peers or instructors [13], [14]. Systematic reviews of LLM-based personalized learning describe these models as capable of providing real-time, adaptive feedback that would be infeasible to deliver manually at scale [11], while targeted studies on programming and computer-science coursework demonstrate that LLMs can translate rubric-based assessment data into individualized, rubric-aligned feedback [12]. Closed-loop tutoring architectures further combine cognitive diagnosis models with LLM-driven feedback generation to close the loop between diagnosis and remediation [15's counterpart, 14].

Despite these two active research directions, very few systems integrate multi-subject predictive classification with LLM-based advisory feedback within a single end-to-end pipeline. Most prediction-focused papers report only accuracy metrics [1]–[10], while most LLM-in-education papers evaluate feedback quality independent of any underlying statistical prediction model [11]–[14]. This separation between predictive analytics and generative advisory feedback constitutes the principal gap addressed by the present work.

V. RESEARCH GAP

- 1) Limited subject coverage: most prior systems evaluate one or two subjects rather than a full multi-subject academic profile.
- 2) No AI-generated, natural-language recommendations accompanying the numeric prediction.
- 3) Feedback, where present, is typically generic at the subject level rather than grounded in specific topics or chapters.
- 4) Static prediction models that output a label without contextual explanation.
- 5) No mechanism for an instructor's own judgment to be captured and merged with the automated output.
- 6) Absence of a personalized, subject-specific study plan for the learner.
- 7) No automatically generated, downloadable academic report.
- 8) Limited interactive dashboards suitable for direct use by non-technical faculty or students.

The proposed system directly addresses each of these gaps by combining a multi-subject classifier with an integrated LLM-based advisory layer and automated reporting.

VI. PROPOSED METHODOLOGY

The proposed pipeline begins with structured student input and proceeds through preprocessing, model inference, and AI-based advisory generation, as illustrated in Fig. 1. Raw input parameters — gender, course preparation status, daily study hours, marks in the six core subjects, and reading and

writing scores — are first cleaned and normalized. Categorical attributes are then converted using label encoding so that they can be consumed by the classification algorithms. The encoded feature vector is passed to the trained classification model, which outputs a binary PASS/FAIL prediction. This prediction, together with the underlying subject-wise scores, is forwarded to a performance-analysis module that identifies the student's weakest and strongest subjects, and subsequently to the Llama 3.2 academic advisor. Rather than issuing generic subject-level advice, the advisor is prompted with each weak subject's score to generate topic-level guidance that names specific chapters or concepts the student should revisit. In parallel, the pipeline accepts an optional free-text remark from the instructor through a Teacher's Suggestion module; this remark is stored verbatim and merged, unmodified, alongside the AI-generated content. The final output — the prediction, the performance analysis, the topic-level AI guidance, and the teacher's remark — is compiled into a single downloadable PDF report.

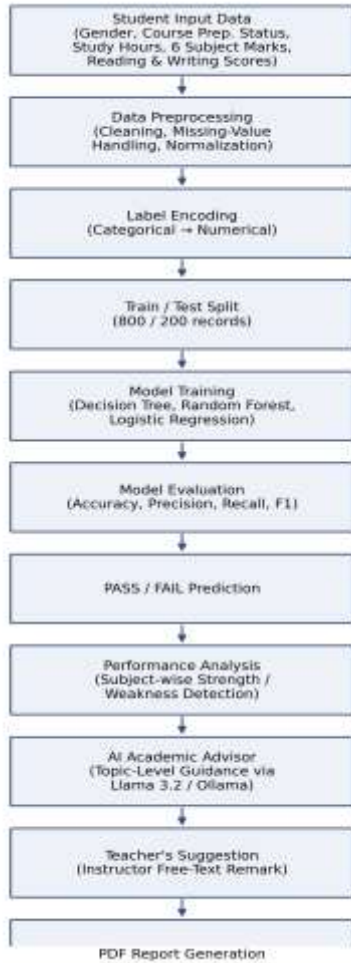


Fig. 1. Proposed methodology workflow, from raw student input to AI-generated PDF report.

VII. DATASET DESCRIPTION

The system was trained and evaluated on a custom-generated dataset of 1000 student records comprising 13 total

fields, of which 11 are used as input features and one (Result) serves as the target label. The dataset was partitioned into 800 training records and 200 testing records (an 80:20 split). Table I summarizes the input features used for model training.

Table I. Input Features Used for Model Training

Feature	Type / Description
Gender	Categorical (Male / Female)
Course Preparation Status	Completed / Partially Completed / Not Completed
Study Hours	Numeric — average daily study hours
Data Structures	Numeric — subject marks
Database Management	Numeric — subject marks
Operating Systems	Numeric — subject marks
Computer Networks	Numeric — subject marks
Python Programming	Numeric — subject marks
Artificial Intelligence	Numeric — subject marks
Reading Score	Numeric — reading assessment
Writing Score	Numeric — writing assessment

The target variable, Result, is a binary label (PASS / FAIL). Table II shows the class distribution across the full dataset.

Table II. Class Distribution

Class	Records	Percentage
PASS	630	63.0%
FAIL	370	37.0%

The dataset is reasonably balanced, with a moderate skew toward the PASS class that reflects typical academic cohort distributions and does not require synthetic resampling for the classifiers evaluated in this study.

VIII. MACHINE LEARNING MODEL

Three supervised classification algorithms were implemented and compared using Scikit-learn [18]: Decision Tree, Random Forest [19], and Logistic Regression [20]. Decision Tree was selected as an interpretable baseline; Random Forest was included for its established robustness against overfitting and its ability to rank feature importance [4], [5], [7]; and Logistic Regression was included as a well-calibrated linear baseline commonly used in binary academic-outcome prediction [8]. All three models were trained on the same 800-record training partition and evaluated on the same held-out 200-record test partition to ensure a fair comparison.

Model selection for final deployment was based on test-set accuracy together with balanced precision and recall across both classes. Section X presents the comparative results, which determined the final deployed classifier.

IX. SYSTEM ARCHITECTURE

Fig. 2 depicts the layered system architecture. The Streamlit web interface accepts student data, which is passed through an input-validation layer before reaching the trained ML model. The prediction engine's PASS/FAIL output is routed in parallel to (i) a performance-analysis module that computes subject-wise strengths and weaknesses, and (ii) the Llama 3.2 AI advisor, hosted locally through Ollama, which converts the numeric and analytical output into personalized natural-language recommendations. Both outputs are merged by the PDF report generator into a single downloadable document.

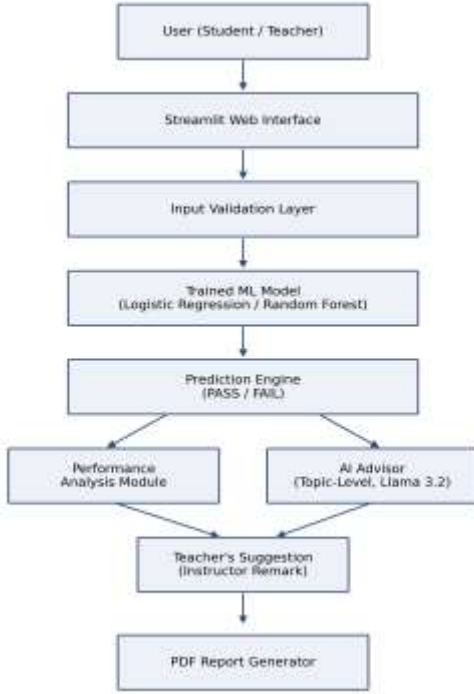


Fig. 2. Layered system architecture of the proposed platform.

X. SYSTEM IMPLEMENTATION AND USER INTERFACE

The proposed architecture was implemented as a working Streamlit prototype to validate the end-to-end pipeline described in Sections VI and IX. Fig. 3 shows the data-entry interface, through which a student's name, gender, test-preparation status, daily study hours, six subject marks, and reading/writing scores are captured through interactive sliders and drop-downs before being submitted to the trained model.



Fig. 3. Streamlit data-entry interface for the AI Student Performance Prediction System.

Once a prediction is generated, the interface renders three complementary visual summaries. Fig. 4 displays a gauge widget reporting the model's prediction confidence for the given input, Fig. 5 presents an "Academic Radar" chart that plots the student's six subject scores on a common polar axis to convey overall academic shape at a glance, and Fig. 6 shows the dashboard view that pairs the PASS/FAIL prediction and confidence score with expandable, subject-wise improvement suggestions ordered from weakest to strongest subject.

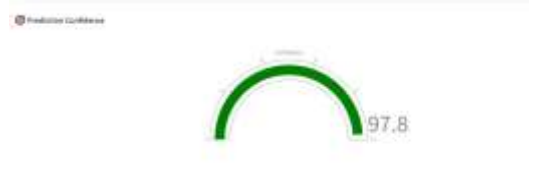


Fig. 4. Prediction-confidence gauge rendered for a sample student record.



Fig. 5. Academic Radar chart visualizing subject-wise performance shape for a sample student.



Fig. 6. Dashboard view combining the PASS/FAIL prediction, confidence score, and subject-wise focus suggestions.

As a concrete illustration, a test record for a sample student (Gender: Male, Study Hours/Day: 1, Test Preparation: Completed, Reading Score: 55, Writing Score: 53, and subject marks ranging from 50 to 74) was submitted to the deployed pipeline. The system predicted a PASS outcome with 97.75% model confidence and an average score of 64.5%, and classified the student's overall performance as "Average." The performance-analysis module correctly ranked Data Structures (74), Computer Networks (72), and Operating Systems (71) as the student's relative strengths, and Database Management (61), Artificial Intelligence (59), and Python Programming (50) as areas requiring attention. Fig. 7 shows the corresponding excerpt of the automatically generated PDF report, reproducing this same student-information and prediction summary.

Student Performance Report			
AI-Powered Academic Performance Analysis			
Generated on: 07-Aug-2025, 02:01 PM			
STUDENT INFORMATION			
Student Name	John Doe	Roll No	123456789
Program	Maths	Grade	10
Enrollment Date	2024-08-01	Average Score	84.5
SS	88	SS	84.5
PREDICTION SUMMARY			
Prediction Result	Performance Level	Model Confidence	
PASS	Average	97.75%	

Fig. 7. Excerpt of the generated PDF report showing student information and the prediction summary for the sample record.

For the weakest subject, the Llama 3.2 AI advisor generated the following topic-level guidance rather than a generic statement: “Focus on revisiting functions and data structures like lists, dictionaries, and classes... try practicing coding challenges or watching video tutorials that break down problems into manageable steps.” Similar topic-grounded feedback was produced for every other subject in the report (e.g., naming “Linked Lists” and “Balancing Binary Search Trees” for Data Structures, and “subnetting” and “Routing Protocols” for Computer Networks), confirming that the advisor's output is anchored to specific curricular topics rather than issuing generic study advice.

In addition to this AI-generated feedback, the platform was extended with a Teacher's Suggestion module (Fig. 8) that lets the instructor type a short, free-text remark for the student directly in the Streamlit interface. This remark is captured verbatim — for the sample student, the instructor entered “Good, Keep it up” — and is merged unmodified into the same PDF report immediately above the AI Academic Advisor's overall guidance paragraph, as shown in Fig. 9. This design gives the final report a human-in-the-loop character: the instructor's own judgment is preserved alongside, rather than replaced by, the automated analysis. All results — the tabular summary, the charts of Figs. 4–6, the topic-level AI recommendations, and the teacher's remark — are compiled automatically into a multi-page, downloadable PDF report, confirming that the reporting stage described in Section VI functions correctly end-to-end.

Teacher's Suggestion

Fig. 8. Teacher's Suggestion input field in the Streamlit interface, used to append instructor remarks to the report.

TEACHER'S SUGGESTION	
Good Keep it up	
AI ACADEMIC ADVISOR — OVERALL GUIDANCE	
Hey, I wanted to talk to you about how things are going. So far, I think you've been doing okay overall, but there's definitely room for improvement. One thing that would really help is if you started planning out your study time a bit better this week. Try setting aside specific times each day just for studying, and stick to it as much as possible. When you're studying, make sure to break up your work into smaller chunks - let's say 30 minutes of focused work on one topic, followed by a 10-minute break. That way, you'll stay engaged without getting too tired. Also, try reviewing the material from the day before instead of trying to cram everything in at once. Now, I know that Python programming is your weak subject, and it's not surprising given how tough it can be. But here's a practical tip: find some online resources or videos that explain the basics in a way that makes sense for you, and try working through some practice problems or exercises. That might help you feel more comfortable with the material. As for the other subjects, I think you've got some strengths there - Data Structures is a great subject! But if you want to improve in areas like Database Management or Computer Networks, try watching video lectures or taking online courses to get a better sense of what's expected. And remember that practice problems are key - the more you do, the easier it will be. If you're struggling with reading comprehension, try summarizing each paragraph and identifying the main points. Then, come back to the specific paragraphs or chapters and focus on what really matters. And when it comes to writing, don't be afraid to take your time - sometimes rephrasing something out loud can help you catch errors or unclear sentences. Remember, the key is progress, not perfection. You've got this! Keep it up, stay focused, and I know you'll do great!	

Fig. 9. PDF report excerpt showing the Teacher's Suggestion merged directly above the AI Academic Advisor's overall guidance.

XI. RESULTS AND DISCUSSION

All three classifiers were evaluated on the 200-record test set. Table III reports the classification accuracy of each algorithm, and Table IV and Table V report per-class precision, recall, and F1-score for Random Forest and Decision Tree, the two tree-based models, for comparison against the deployed Logistic Regression model.

Table III. Model Accuracy Comparison

Algorithm	Accuracy
Decision Tree	95.50%
Random Forest	97.50%
Logistic Regression	99.50%

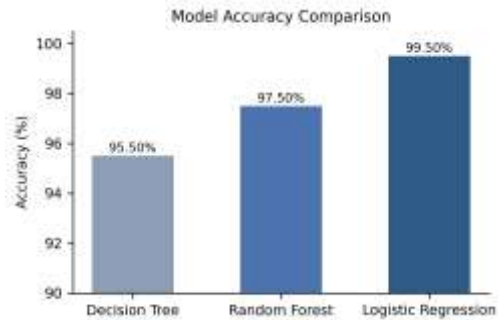


Fig. 10. Test-set accuracy comparison across the three evaluated classifiers.

A. Random Forest Performance

Table IV. Random Forest — Classification Report

Metric	FAIL	PASS
Precision	0.94	1.00
Recall	1.00	0.96
F1-Score	0.97	0.98

The Random Forest confusion matrix (Fig. 11) shows that all 74 actual FAIL students were correctly classified, with only five actual PASS students misclassified as FAIL, and zero false positives. This indicates that the model is

conservative in a favorable direction for an academic early-warning use case: it never misses a genuinely at-risk student, at the modest cost of occasionally flagging a passing student for review.

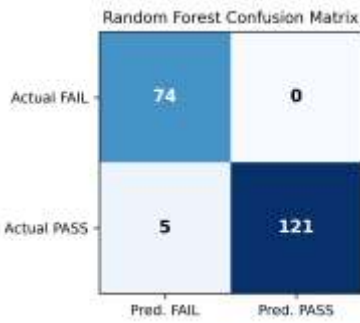


Fig. 11. Confusion matrix for the Random Forest classifier on the 200-record test set.

B. Logistic Regression Performance

Table V. Logistic Regression — Classification Report

Metric	FAIL	PASS
Precision	0.99	1.00
Recall	1.00	0.99
F1-Score	0.99	1.00

Logistic Regression achieved the highest overall accuracy (99.50%) and near-perfect precision and recall on both classes. This suggests that, for the engineered feature set used in this study, the relationship between the input variables and the PASS/FAIL outcome is close to linearly separable, which favors a well-regularized linear classifier over more flexible tree-based models on this particular dataset.

C. Feature Importance

Feature importance extracted from the Random Forest model (Table VI, Fig. 12) shows that Study Hours is by far the most influential predictor, contributing approximately 44.3% of total importance — more than three times the contribution of the next-ranked feature, Python Programming (12.3%). Operating Systems and Artificial Intelligence scores contribute comparably (approximately 9.6% each), followed by Writing Score (8.7%) and Reading Score (5.0%). Gender and Course Preparation Status contribute negligibly (0.06% and 0.12% respectively), indicating that the model's predictions are driven almost entirely by academic and behavioral indicators rather than demographic attributes.

Table VI. Random Forest Feature Importance

Feature	Importance
Study Hours	0.4430
Python Programming	0.1232
Operating Systems	0.0966
Artificial Intelligence	0.0963
Writing Score	0.0874
Reading Score	0.0498

Data Structures	0.0360
Computer Networks	0.0335
Database Management	0.0323
Course Preparation Status	0.0012
Gender	0.0006

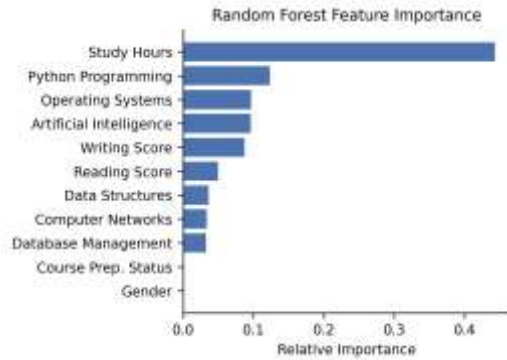


Fig. 12. Relative feature importance derived from the Random Forest model.

Overall, the experimental evaluation shows that all three algorithms perform well on the generated dataset, with Logistic Regression achieving the best test accuracy (99.50%), followed by Random Forest (97.50%) and Decision Tree (95.50%). Given its top accuracy and balanced precision/recall, Logistic Regression was selected as the final deployed prediction model, while Random Forest is retained for its interpretable feature-importance output, which directly informs the performance-analysis module feeding the AI advisor. The dominance of Study Hours as a predictor, together with the meaningful contribution of programming-oriented subjects (Python Programming, Operating Systems, Artificial Intelligence), suggests that combining behavioral indicators with subject-specific performance yields a substantially more informative model than marks alone.

XII. ADVANTAGES

- 1) Joint analysis of six core subjects instead of a single subject in isolation.
- 2) AI-generated, topic-level academic recommendations (specific chapters/concepts) rather than generic advice.
- 3) Human-in-the-loop reporting: instructor remarks are captured via the Teacher's Suggestion module and merged with the AI output.
- 4) High prediction accuracy across all three evaluated algorithms.
- 5) Simple, interactive Streamlit-based interface accessible to non-technical users.
- 6) Automatic generation of downloadable PDF performance reports.
- 7) Supports faculty in early identification of at-risk students.
- 8) Reduces the manual effort required for routine performance review.

XIII. APPLICATIONS

- 1) Colleges and universities
- 2) Schools
- 3) Online learning platforms
- 4) Coaching and training institutes
- 5) Academic counseling centers
- 6) Continuous student performance monitoring programs

XIV. SYSTEM LIMITATIONS

While the results in Section XI are encouraging, the current implementation has several limitations that should be considered when interpreting them.

- 1) The training data is a custom-generated dataset of 1000 records rather than data drawn from a live institutional information system; real academic cohorts may exhibit different noise, imbalance, and correlation patterns.
- 2) The system currently models six core Computer Science subjects; extending coverage to a full multi-department curriculum is left for future work.
- 3) The Llama 3.2 advisor is served locally through Ollama, so the quality and latency of generated feedback depend on locally available compute resources, and occasional generic phrasing can still occur despite topic-level prompting.
- 4) Model evaluation used a single held-out test split rather than a multi-semester longitudinal deployment, so long-term predictive validity across changing curricula and cohorts is not yet established.
- 5) Reported confidence scores (e.g., 97.75%) reflect classifier probability calibration on the current dataset and should be interpreted as indicative rather than as an exact likelihood of the outcome.
- 6) The Teacher's Suggestion module currently accepts free-text input without structured moderation, so report quality for that field depends on the instructor's own input.

XV. FUTURE ENHANCEMENTS

- 1) Integration of attendance data as an additional predictive feature.
- 2) Semester-wise longitudinal performance tracking.
- 3) Automatic syllabus upload and subject extraction.
- 4) Integration with Learning Management Systems (LMS).
- 5) Cloud deployment for institution-wide, multi-user access.
- 6) A companion mobile application.
- 7) Parent notification and progress-sharing features.
- 8) A real-time analytics dashboard for department-level oversight.
- 9) Explainable AI (XAI) visualizations such as SHAP values for individual predictions.
- 10) Integration with institutional ERP systems for unified data flow.

XVI. CONCLUSION

This paper presented an AI-Assisted Multi-Subject Student Performance Prediction System that combines supervised machine learning with a generative-AI academic advisor to support academic decision-making. By jointly analyzing six core Computer Science subjects together with study habits and preparation status, the evaluated models achieved strong classification performance, with Logistic Regression reaching 99.50% test accuracy and Random Forest reaching 97.50% while additionally providing interpretable feature importance. The integration of a topic-level AI Academic Advisor built on Llama 3.2, together with the Teacher's Suggestion module, extends the system beyond conventional pass/fail prediction by combining automated, curriculum-grounded recommendations with the instructor's own judgment in a single downloadable report. This combination assists educators in identifying at-risk students early and enables timely, individualized academic intervention, making the proposed system a practical and extensible solution for modern educational institutions.

REFERENCES

- [1] "Student Academic Performance Prediction using Machine Learning with Various Features and Scenarios," IEEE Conference Publication, 2024. [Online]. Available: <https://ieeexplore.ieee.org/document/10770729/>
- [2] "Student Performance Prediction System Using Machine Learning," IEEE Conference Publication, 2025. [Online]. Available: <https://ieeexplore.ieee.org/document/11118334/>
- [3] "Predicting Student Performance to Boost Educational Outcomes: The Efficacy of a Random Forest Approach," IEEE Conference Publication, 2024. [Online]. Available: <https://ieeexplore.ieee.org/document/1041035/>
- [4] "Predicting Students Academic Performance using an Improved Random Forest Classifier," IEEE Conference Publication, 2020. [Online]. Available: <https://ieeexplore.ieee.org/document/9167547/>
- [5] "Research on Student Performance Prediction Based on Random Forest Algorithm," in Proc. 2024 Int. Symp. Artificial Intelligence for Education, ACM, 2024, doi: 10.1145/3700297.3700385.
- [6] "Grade Prediction Based on Student Performance using Gradient Boosting and Random Forest," IEEE Conference Publication, 2025. [Online]. Available: <https://ieeexplore.ieee.org/document/11121505/>
- [7] M. T. Sathe and A. C. Adamuthe, "Comparative Study of Supervised Algorithms for Prediction of Students' Performance," Int. J. Modern Education and Computer Science, vol. 13, no. 1, pp. 1–21, 2021, doi: 10.5815/ijmecs.2021.01.01.
- [8] "Predicting Students' Academic Performance Via Machine Learning," Journal of University of Babylon for Pure and Applied Sciences, vol. 27, no. 1, pp. 194–205.
- [9] "Machine Learning-Driven Student Performance Prediction for Enhancing Tiered Instruction," arXiv:2502.03143, 2025.
- [10] "Using Machine Learning to Predict Student Outcomes for Early Intervention and Formative Assessment," PMC, 2025. [Online]. Available: <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC12615791/>
- [11] "The Role of Large Language Models in Personalized Learning: A Systematic Review of Educational Impact," Discover

Sustainability, Springer Nature, 2025, doi: 10.1007/s43621-025-01094-z.

- [12] J. A. Khan, M. Yaqoob, M. Tasadduq, H. S. Dar, and A. Ahsan, "Personalized and Constructive Feedback for Computer Science Students Using the Large Language Model (LLM)," arXiv:2510.11556, 2025.
- [13] "From Feedback to Formative Guidance: Leveraging LLMs for Personalized Support in Programming Projects," in Adjunct Proc. 33rd ACM Conf. User Modeling, Adaptation and Personalization, 2025, doi: 10.1145/3708319.3733808.
- [14] "A Closed-Loop Personalized Learning Agent Integrating Neural Cognitive Diagnosis, Bounded-Ability Adaptive Testing, and LLM-Driven Feedback," arXiv:2510.22559, 2025.
- [15] Y. Zhang, Y. Yun, R. An, J. Cui, H. Dai, and X. Shang, "Educational Data Mining Techniques for Student Performance Prediction: Method Review and Comparison Analysis," *Frontiers in Psychology*, vol. 12, art. 698490, 2021, doi: 10.3389/fpsyg.2021.698490.
- [16] "A Survey on Educational Data Mining Methods Used for Predicting Students' Performance," *Engineering Reports*, Wiley, 2022, doi: 10.1002/eng2.12482.
- [17] "Predicting Student Performance Using Data Mining and Learning Analytics Techniques: A Systematic Literature Review," *Applied Sciences (MDPI)*, vol. 11, no. 1, art. 237, 2021.
- [18] L. Breiman, "Random Forests," *Machine Learning*, vol. 45, no. 1, pp. 5–32, 2001.
- [19] F. Pedregosa et al., "Scikit-learn: Machine Learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.
- [20] D. W. Hosmer, S. Lemeshow, and R. X. Sturdivant, *Applied Logistic Regression*, 3rd ed., Hoboken, NJ: Wiley, 2013.