How can machine learning personalize educational content generated by LLMs

Petr Tsekoyev[1,*,†], Talgat Sembayev[2,†] and Zhanat Nurbekova[3,†]

[1,2] *Astana IT University, Mangilik El avenue, 55/11, 010000 Astana, Kazakhstan*
[3] *Abai Kazakh National Pedagogical University, Dostyk avenue, 13, 050010 Almaty, Kazakhstan*

Abstract

Educational content is typically delivered uniformly despite wide differences in student backgrounds and goals. This study proposes a framework that combines predictive machine learning (ML) with large language models (LLMs) to tailor difficulty and style. Using a large public dataset of performance factors (attendance, study hours, resource access, motivation), we followed CRISP-DM procedures and trained multiple regressors to forecast final exam scores. Linear Regression was selected for its performance ($R^2 \approx 0.80$, RMSE < 1.5 on a 0–100 scale) and interpretability, providing a reasonably accurate grade forecast. A survey of 60 students (secondary to graduate levels) collected desired exam scores and self-rated difficulty (Hard/Medium/Easy). For each student we computed a score gap (predicted − desired) and mapped it to difficulty via a deterministic rule with thresholds tuned on the cohort and fixed for all analyses at $t_1 = -7$ and $t_2 = +9$. In this pilot, the mapping aligned with self-reports on observed cases; we do not treat this as a generalizable accuracy estimate. A larger multi-course evaluation with proper train/validation/test splits and cross-validation is planned. The predicted difficulty label controlled LLM prompting: learners forecasted to struggle received simplified, scaffolded explanations; on-track learners received balanced depth; advanced learners received technical treatments. Student feedback indicated that when predictions matched needs, 92% reported the generated content met their needs; satisfaction fell to 36% when predictions were incorrect. These findings suggest that ML-guided prompt routing can improve perceived relevance of instructional materials while remaining transparent and easy to calibrate. In conclusion, integrating interpretable ML predictions with prompt-engineered LLMs can automate individualized content generation, potentially reduce instructor workload and improve learning outcomes.

Keywords

adaptive learning, intelligent tutoring systems, machine learning, personalized education

1. Introduction

Educational content refers to all instructional materials used in teaching (textbooks, video lectures, exercises, assessments, etc.). In practice, this content is often delivered uniformly to all students, even though learners differ widely in background knowledge, skills, and interests. Adapting or personalizing content means selecting and organizing materials so that each student encounters topics at the right level of difficulty.

Personalization (also called adaptive learning) means tailoring the learning experience to each student's unique profile. In a personalized model, aspects such as difficulty, pace, and content format are adjusted for each learner, rather than using a uniform curriculum. This strategy allows students to progress at their own rate and engage more actively with the material. By optimizing content and teaching methods to individual needs, personalization makes learning more efficient and motivating.

There are different approaches to personalization. Traditional methods include one-on-one tutoring and teacher-driven differentiation, which can be effective but do not easily scale to many students. In contrast, technology-based approaches automate adaptation using data and algorithms. For example, many adaptive platforms begin with a diagnostic quiz or assessment to gauge a

[1] *AI 2025: Workshop on Artificial Intelligence, November 19-20, 2025, Almaty, Kazakhstan*
* Corresponding author.
† These authors contributed equally.
242704@astanait.edu.kz (P. Tsekoyev); talgat.sembayev@astanait.edu.kz (T. Sembayev); zh.nurbekova@abaiuniversity.edu.kz (Z. Nurbekova)
0009-0000-0220-2579 (P. Tsekoyev); 0000-0003-2360-8767 (T. Sembayev); 0000-0003-0249-7690 (Z. Nurbekova)

student's current understanding and then recommend a tailored sequence of lessons and exercises. More advanced systems apply machine learning to analyze each student's activity and predict areas of difficulty, automatically selecting the optimal next content. In this way, an intelligent tutoring system or learning platform can dynamically adjust difficulty and examples for every student without manual effort.

Artificial intelligence (AI) and large language models (LLMs) further enable scalable personalization. AI-driven systems can continuously monitor a student's performance (for example, quiz responses or interaction patterns) and adapt the learning path in real time. For instance, if a student struggles with a concept, the system can automatically provide additional explanations or simpler practice problems on the spot. LLMs (such as ChatGPT/DeepSeek) extend this capability by generating customized educational content on demand: they can produce explanations, examples, or practice questions tailored to a learner's level and context. Together, these AI technologies allow each student to receive personalized content and feedback at scale, effectively acting as a virtual tutor that adapts to individual needs.

Modern education in high schools and universities is increasingly facing the need for individualized approach with increasing volumes of educational material, diversity of learning trajectories and limited resources of teachers [1, 2, 3]. Digitalization of the learning process penetrates into all aspects: from the automation of administration to distance platforms focused on the analysis of student's actions. Its main goal is not only to reduce the workload of teachers and speed up data processing, but also to improve the quality of learning by taking into account the personal preferences, perception style and motivation of each student [4, 5].

In response to these challenges, AIED systems that combine algorithms for analyzing educational data, recommendation mechanisms, and adaptive tests are being actively developed. A key element of such platforms has become predictive models that can identify students at risk of poor performance at an early stage and select optimal materials for them [6, 7, 8]. The idea of machine learning support originated in the 1960s-70s with the emergence of "learning machines" and expert systems based on programmed learning theories. In the 1990s-2000s, the development of statistical methods and classical ML algorithms (SVM, decision trees) led to the first adaptive systems [9]. With the introduction of deep neural networks and the emergence of Generative Pre-trained Transformers (GPT-series) after 2022, AIED got a qualitative leap: systems learned not only to collect and analyze large amounts of training data, but also to generate explanations, tests and interactive tasks on-the-fly [10, 11].

Modern AIED platforms use a variety of algorithms, from recurrent networks to SVR and gradient boosting. Another type of models, Large Language Models (LLMs) are capable of generating quality explanations, test questions and examples, targeting the context and the student's level of proficiency. However, without careful prompt customization, they can generate inaccurate or redundant answers – "hallucinations" [12, 13, 14].

The goal of the study is to improve the quality of education by applying machine learning algorithms to personalize instructional materials generated by large language models.

Research questions:

1. How accurately can predictive regression models forecast students' final exam performance to guide the personalization of educational content?
2. To what extent does adaptive prompt engineering of LLMs—based on these predictions—affect student satisfaction and perceived relevance of the generated materials?

2. Materials and methods

The study tested the hypothesis that machine learning techniques can be used to personalize educational content for individual students by combining data-driven prediction with adaptive content generation.

Literature and platform review. A review of AI tools and platforms in education was conducted to identify current personalization mechanisms (e.g., task difficulty and pacing adaptation) and to inform the integration of ML predictions with an LLM-based content generator [15, 16, 17, 18]. Major online learning systems (e.g., Coursera, Khan Academy) and specialized LLM initiatives (e.g., Google Learn About pilot) were examined to understand existing workflows and constraints.

Dataset selection. To ensure relevance and scale, candidate datasets were restricted to ≥1,000 records and not older than 2021. After reviewing seven large datasets on academic performance, the Student-performance-factors dataset was selected [19], containing learning habits, background factors, and exam results in secondary or post-secondary education.

Preprocessing. The initial dataset underwent standard cleaning: (i) missingness assessment showed ~1.18% missing values across three variables, which were imputed using plausible values drawn from observed distributions; (ii) duplicate records were removed; (iii) three categorical features were encoded (integer codes or one-hot encodings); (iv) outliers were identified as observations far from the bulk of the data and 1,645 such records were removed to avoid distortion during model training. Exploratory data analysis included plotting histograms for key variables (e.g., Hours Studied, Previous Scores, Attendance, Exam Score) (Figure 1–4) and computing a Pearson correlation matrix across numerical features (Figure 5), without drawing inferential conclusions at this stage.

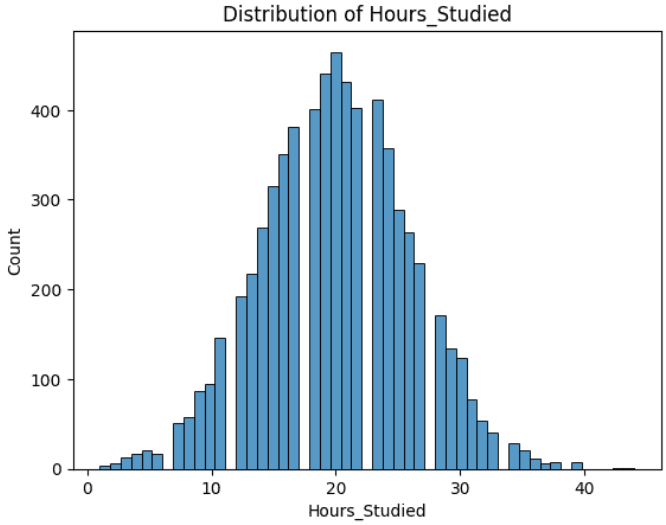

Figure 1: Hours Studied distribution.

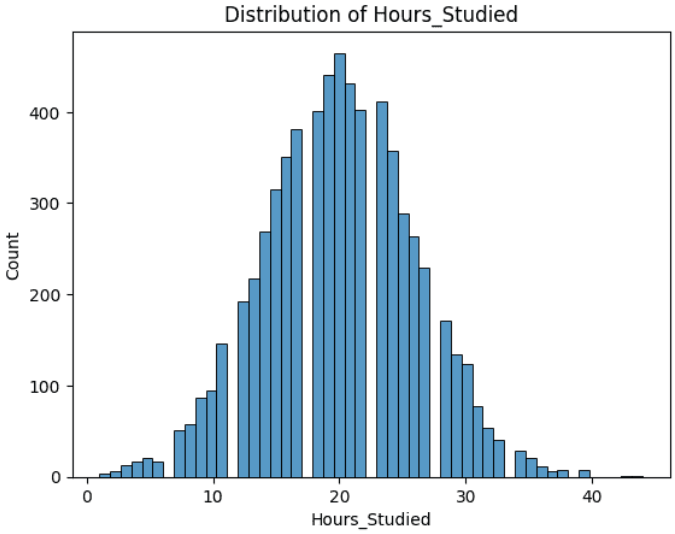

Figure 2: Previous Scores distribution.

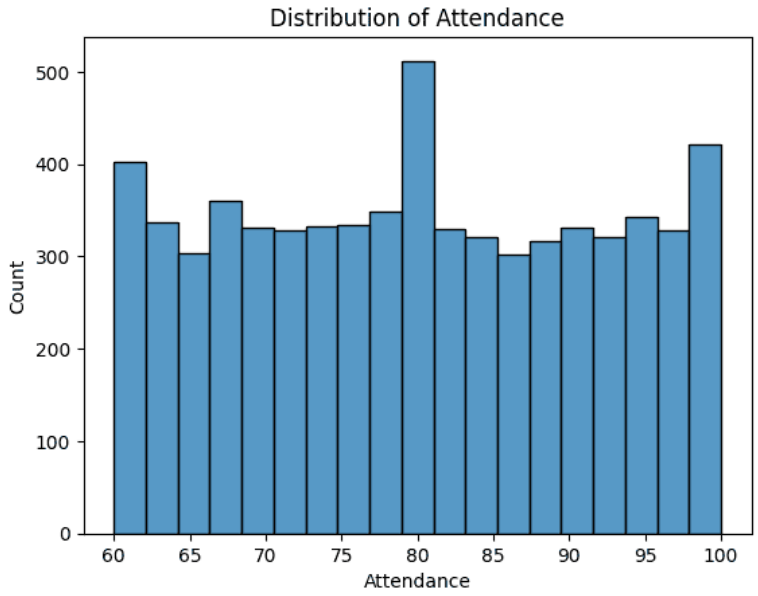

Figure 3: Attendance distribution.

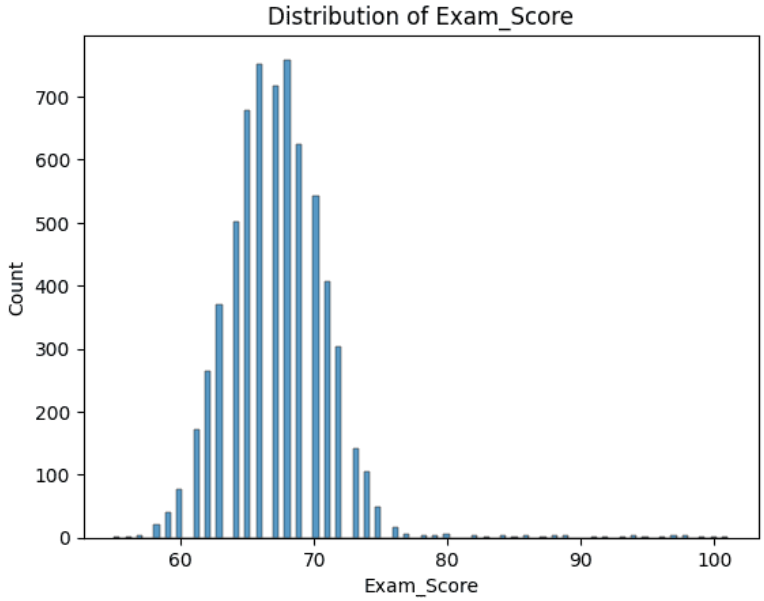

Figure 4: Exam Score distribution.

Feature engineering and scaling. Based on domain knowledge, numerical features were scaled/normalized as needed. A derived metric improved_percentage = (Exam_Score − Previous_Scores)/100 was defined to represent relative improvement; this variable was used descriptively within analysis workflows with care to avoid leakage in predictive scenarios.

Predictive modeling for exam score. A supervised regression approach modeled Exam Score as a function of student factors. Predictors included Hours_Studied, Attendance, Access_to_Resources_m, and Motivation_Level_m. The dataset was split 80:20 into train/test subsets. Ten regression algorithms were trained, including Linear Regression, Support Vector Regression (SVR), and tree-based methods. Evaluation employed MSE, MAE, RMSE, and R^2. Target criteria, set a priori for satisfactory performance, were $R^2 > 0.80$ and RMSE < 1.5 (Figure 6 summarizes evaluated metrics without implying selection).

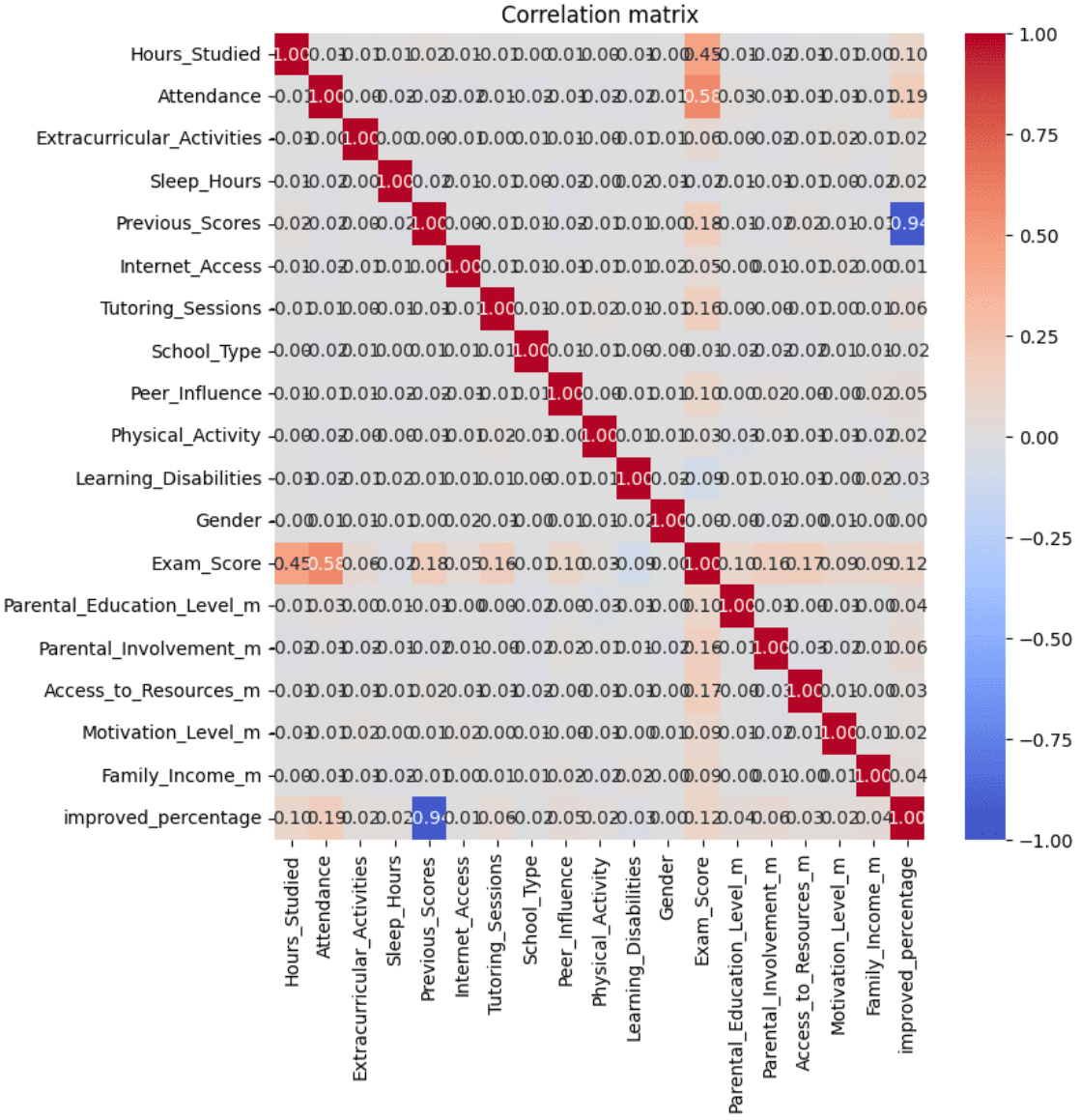

Figure 5: Correlation Matrix.

Model	MAE	RMSE	R^2
Linear Regression	1.14	1.44	0.8012
Ridge Regression	1.14	1.44	0.8011
Support Vector Regressor	1.16	1.46	0.7954
Gradient Boosting	1.16	1.46	0.7934
Neural Network	1.17	1.47	0.7917
K-Nearest Neighbors	1.30	1.64	0.7423
Random Forest	1.32	1.65	0.7366
ElasticNet Regression	1.33	1.68	0.7286
Lasso Regression	1.33	1.68	0.7268
Decision Tree	1.59	1.99	0.6179

Figure 6: Metrics evaluation.

Model selection protocol and interpretability criteria. We compared a suite of regression algorithms, such as Linear Regression, Ridge/Lasso/Elastic Net, Support Vector Regression (RBF),

Random Forest, and Gradient Boosting, under a consistent validation protocol. The primary selection criterion was out-of-sample error (RMSE) with R^2 as a secondary indicator; tie-breakers favored interpretability and stability on small/medium samples. To prevent leakage, all preprocessing steps were fitted on the training folds only. In the extended study, we will report 5-fold cross-validation on the training split, tune hyperparameters on validation data, and evaluate the final model on a held-out test set, additionally providing 95% confidence intervals via bootstrap. This protocol ensures that any observed gains over Linear Regression reflect true generalization rather than overfitting.

ML⟶LLM Adaptation Pipeline. To make the control link explicit, we route the regression prediction into prompt selection through a simple, deterministic mapping. Let pred denote the predicted exam score from the regression model, and desired the student's self-declared target score. We define the score gap gap = pred − desired. Two thresholds (t_1, t_2) partition gap into three difficulty tiers: gap < t_1 ⟶ Hard, $t_1 \le$ gap $\le t_2$ ⟶ Medium, gap > t_2 ⟶ Easy. The selected tier activates a corresponding prompt template that modifies the explanation depth and style delivered by the LLM. In other words, the ML model provides a quantitative control signal (gap), which deterministically selects an instruction set for the LLM to follow when generating the response. In this study, the thresholds were fixed at $t_1 = -7$ and $t_2 = +9$, obtained via validation tuning on the current cohort.

Survey design and difficulty mapping. A separate survey of 60 students (secondary to graduate level, studying mathematics or natural sciences) collected (a) desired_score and (b) self-rated exam difficulty (Hard/Medium/Easy). To mitigate threshold clustering at round numbers, desired_score values were perturbed by a small ±5% random deviation. For each student, we computed the score gap as gap = predicted_exam_score − desired_score. Rather than training a classifier on this small pilot, we applied a deterministic threshold rule with two thresholds (t_1, t_2) to assign difficulty tiers: gap < t_1 ⟶ Hard, $t_1 \le$ gap $\le t_2$ ⟶ Medium, gap > t_2 ⟶ Easy. On the pilot cohort, this rule coincided with the self-reported labels on the observed cases; however, we do not treat this as a generalizable accuracy estimate. Thresholds were validation-tuned on the current cohort and fixed at $t_1 = -7$ and $t_2 = +9$; when deploying to new cohorts, thresholds will be re-tuned with the same protocol and evaluated on a held-out test set.

Threshold tuning. With n = 60 participants, we partitioned the data into a validation subset (≈40) and a held-out test subset (≈20), stratified by the self-reported difficulty tiers (Hard/Medium/Easy). On the validation subset, we performed a coarse grid search over two thresholds to map gap = predicted_exam_score − desired_score into three difficulty tiers (gap < t_1 ⟶ Hard, $t_1 \le$ gap $\le t_2$ ⟶ Medium, gap > t_2 ⟶ Easy). The thresholds that best aligned with the self-reported tiers on validation were $t_1 = -7$ and $t_2 = +9$; these values were fixed prior to any held-out evaluation and used for all results reported in this paper.

Table 1
Variables and definitions (pilot)

Variable	Definition
predicted_exam_score	Regression model output (0–100)
desired_score	Student self-declared target score (0–100)
gap	predicted_exam_score − desired_score

Table 2
Variables and definitions (pilot)

Rule	Assigned difficulty
gap < t_1	Hard
$t_1 \le$ gap $\le t_2$	Medium
gap > t_2	Easy

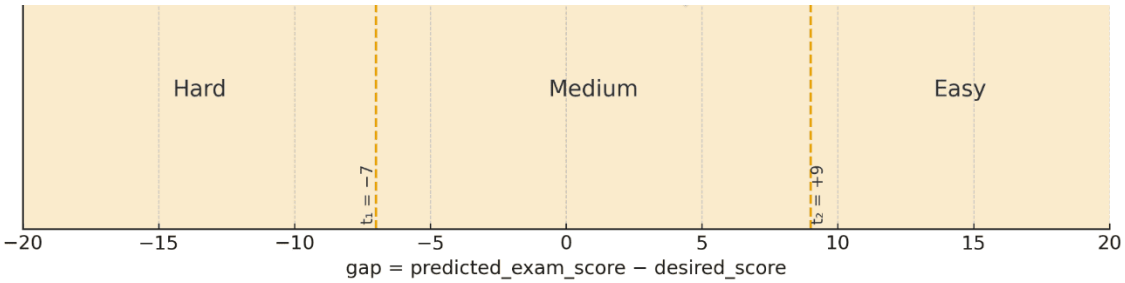

Figure 7: Deterministic threshold mapping from gap = predicted − desired to (Hard/Medium/Easy) with fixed thresholds $t_1 = -7$ and $t_2 = +9$.

Prompt engineering and content generation. An adaptation layer modifies student queries to a GPT-based model by attaching difficulty-specific hidden instructions selected by the ML⟶LLM pipeline. For each request, the system (i) computes the difficulty tier from gap = predicted_exam_score − desired_score using thresholds (t_1, t_2), (ii) selects a prompt template for that tier, and (iii) appends the user's query. This produces simpler, scaffolded explanations for Hard, balanced depth for Medium, and advanced, technical treatments for Easy. The templates below show the fixed, reusable instruction blocks; only the user's query changes at runtime.

Table 3
Difficulty-specific prompt templates (instruction blocks)

Tier	Hidden instruction template (excerpt)
Hard	"Explain step by step in plain, simple language. Start from definitions, avoid jargon, and include 1–2 concrete, everyday examples. Break the solution into short steps and check understanding after each step. Do not assume prior mastery; prefer intuition over formalism."
Medium	"Provide a balanced explanation: concise definitions, 1 worked example, and 1 short exercise suggestion. Use standard terminology where helpful; include brief intuition and the key formula. Keep the answer focused and 6–10 sentences long."
Easy	"Give an advanced, in-depth treatment. Include formal definitions, relevant theorems/lemmas, typical pitfalls, and a compact derivation or proof sketch where applicable. Add 1–2 challenging exercises and briefly relate the concept to adjacent topics."

In operation, a student provides profile inputs (for prediction) and then a content query (e.g., "Explain derivatives"); the system computes the tier, selects the template, and assembles the final prompt automatically. Demonstrative outputs for Hard, Medium, and Easy remain illustrative (Figure 8–10) and are not used as evaluation metrics.

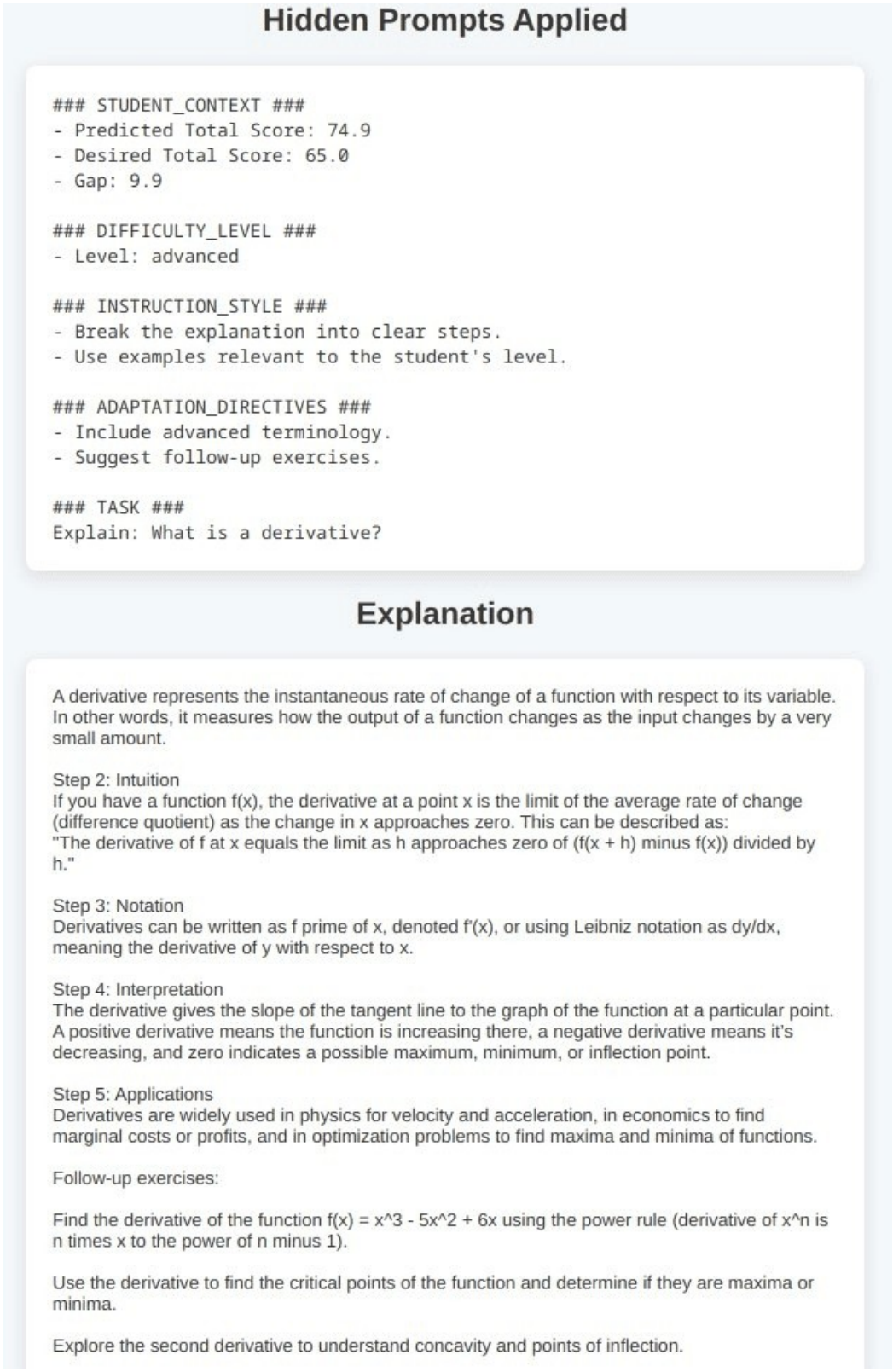

Figure 8: Output result of "easy" difficulty.

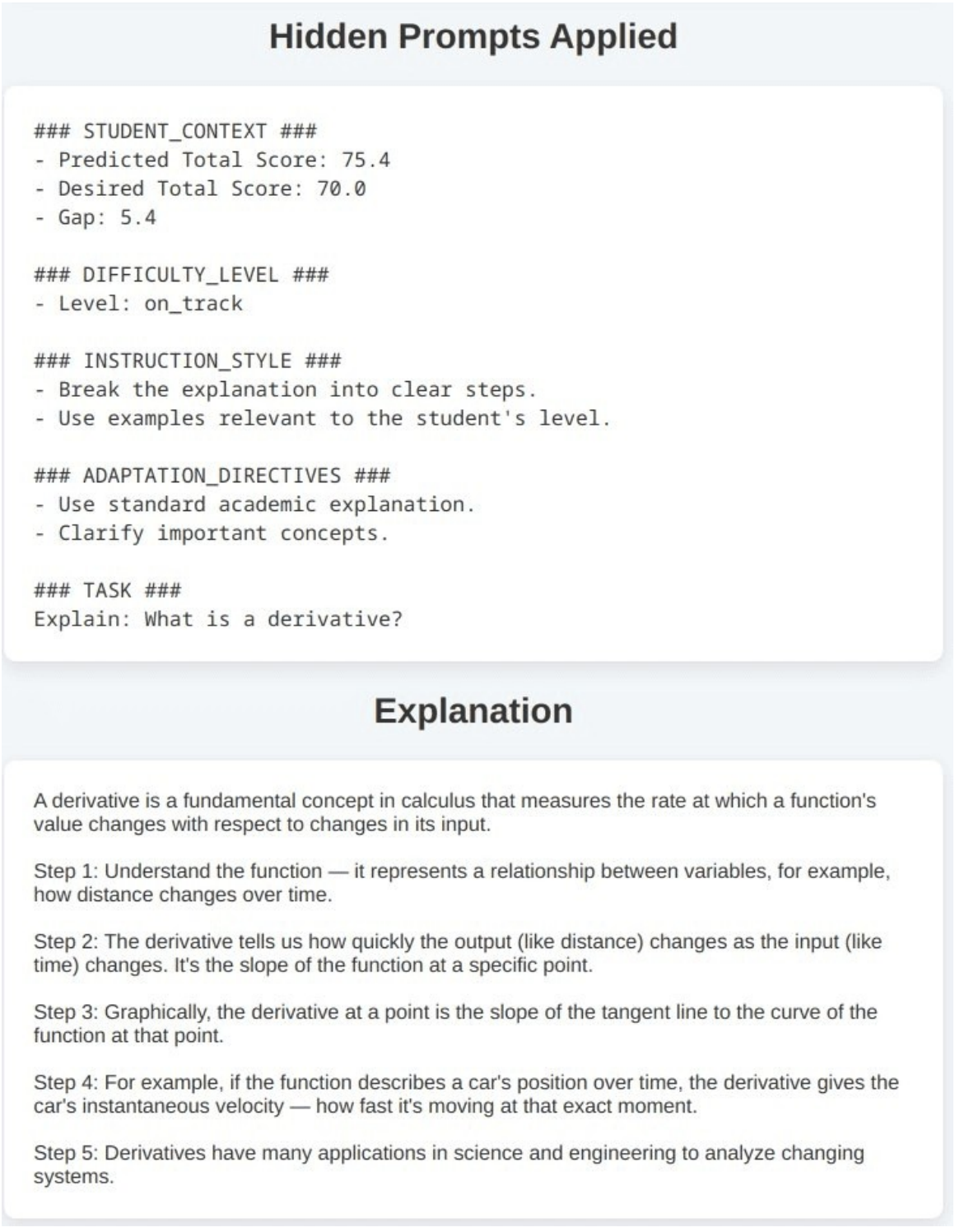

Figure 9: Output result of “medium” difficulty.

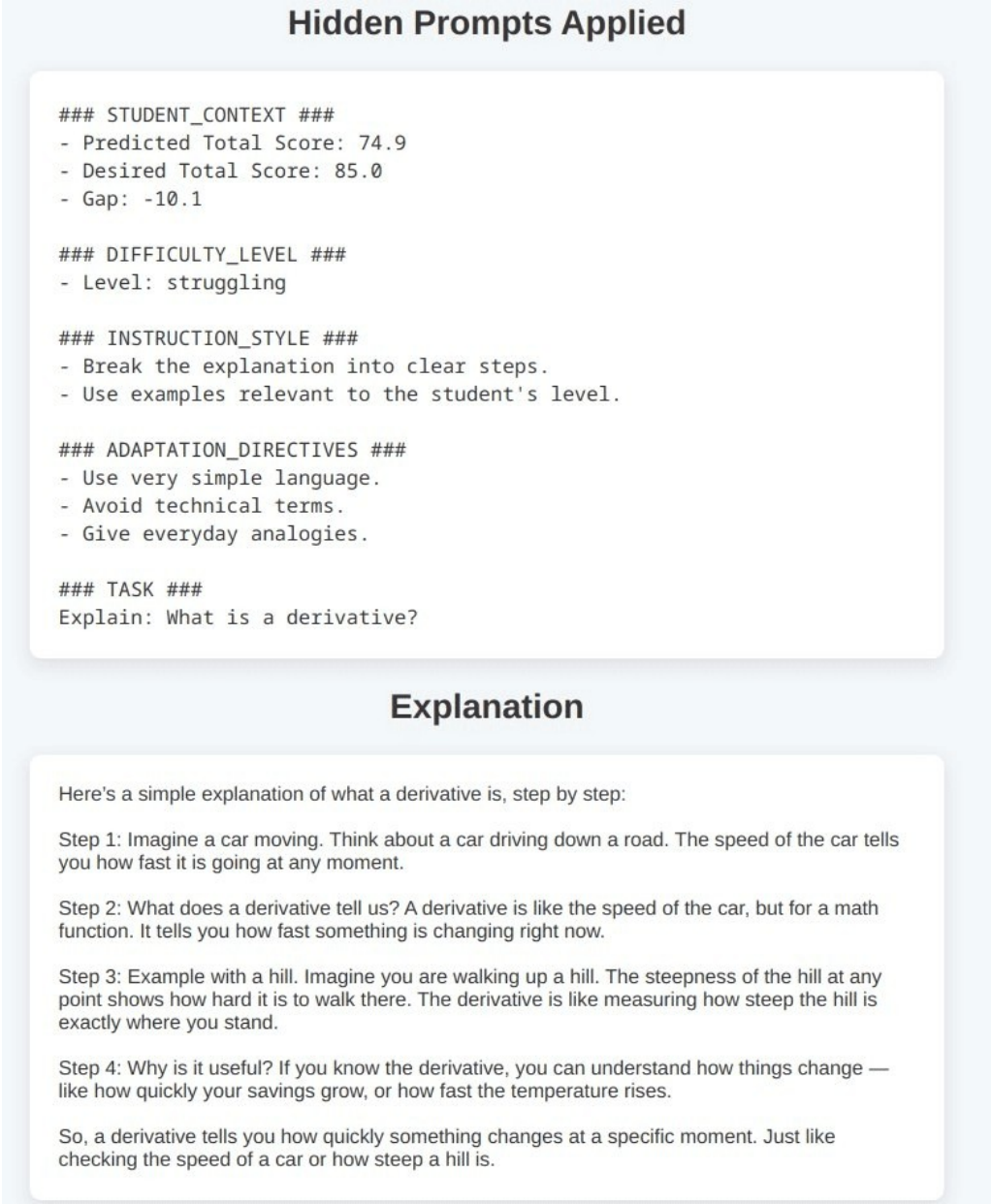

Figure 10: Output result of "hard" difficulty.

User feedback procedure. After interacting with the system, the same cohort provided satisfaction ratings on a 5-point scale regarding perceived fit of the generated materials, including a hypothetical misclassification scenario.

3. Results and discussion

Exploratory relationships. The Pearson correlation matrix (Figure 5) showed the strongest positive associations with Exam Score for Attendance ($r \approx 0.58$) and Hours Studied ($r \approx 0.45$). Previous Scores exhibited a modest positive correlation ($r \approx 0.18$). Several variables, including Family Income and Gender, showed near-zero linear correlation with exam outcomes ($r \approx 0.00$). These observations guided the emphasis on attendance, study time, and, secondarily, prior achievement and resource access in subsequent modeling.

Model comparison and selection. Across ten regression algorithms (Figure 6), the simple Linear Regression model provided the best balance of accuracy and interpretability on the test split. It achieved an R^2 exceeding 0.80 and an RMSE below 1.5 on a 0-100 scale, outperforming more complex methods on this dataset. Consequently, Linear Regression was selected as the final predictor for exam

scores. In the study's analysis narrative, the coefficient of determination was reported around R^2 = 0.8012, indicating that over 80% of score variance was explained, with typical prediction deviations well under two points – sufficiently precise to drive downstream personalization decisions.

Difficulty mapping from score gaps. For each student we computed the score gap as score_diff = predicted_exam_score – desired_score and applied a deterministic threshold rule with two thresholds (t_1, t_2) to assign difficulty tiers (Hard/Medium/Easy). On the pilot cohort (n=60), this rule coincided with the self-reported difficulty labels on the observed cases; however, we do not interpret this as a generalizable accuracy estimate due to the small sample and the absence of a held-out test set. In the extended study, thresholds will be tuned on validation data only, and performance will be reported exclusively on a held-out test set with k-fold cross-validation and confidence intervals (Figure 7 illustrates the mapping). The thresholds used throughout were fixed at $t_1 = -7$ and $t_2 = +9$ following validation tuning on the current cohort.

Adaptive prompting outcomes. Integrating predictions with LLM prompting yielded outputs aligned to the inferred difficulty level (Figure 8–10). For students flagged as likely to struggle, responses emphasized simplified explanations and concrete examples; for students projected to excel, responses contained more advanced, detailed analysis.

Student satisfaction. When the system's difficulty prediction matched student needs, 92% of participants reported that the generated content met their needs; when predictions were incorrect, satisfaction fell to 36%. These outcomes are summarized in (Figure 11).

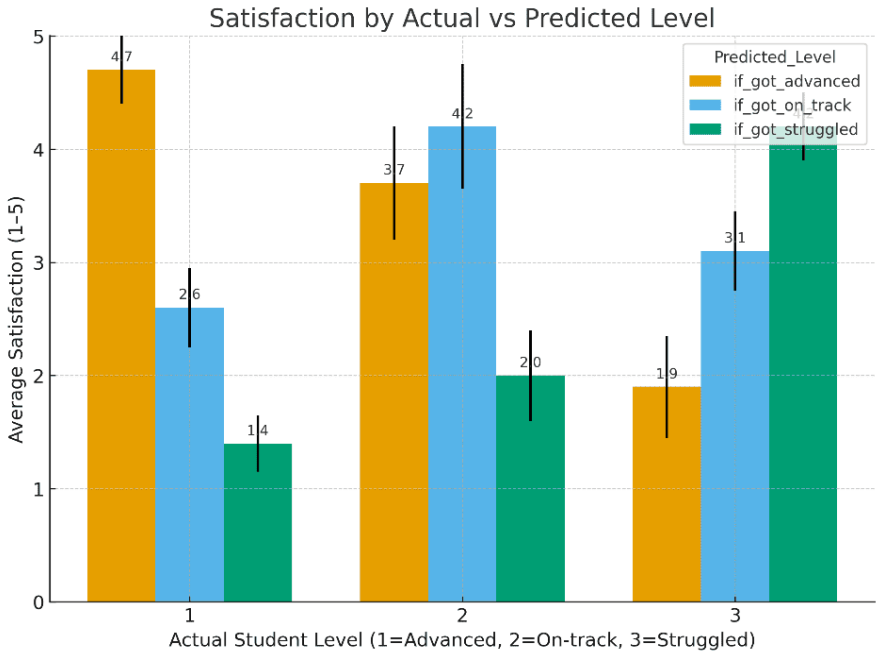

Figure 11: Student satisfaction barplot.

4. Limitations and future work

The dataset and survey were relatively small; the Student-performance-factors dataset may not capture all learning contexts. Because improved_percentage uses the target Exam_Score, it must be excluded from real-time prediction features to avoid leakage. The difficulty survey involved 60 students; larger, more diverse samples are needed to validate generalization. Finally, evaluation focused on self-reported satisfaction rather than objective learning gains. These constraints notwithstanding, the observed accuracy and satisfaction patterns support the feasibility of combining interpretable predictions with adaptive LLM prompting to produce useful personalized materials, and they motivate future classroom-scale validation. An earlier draft employed a trained classifier on the pilot data. To avoid overstating generalization from a small sample, we now use a

deterministic threshold rule and will compare it with trainable classifiers in the extended study. The threshold values ($t_1 = -7$, $t_2 = +9$) were tuned on this cohort and may be dataset-specific. Generalization to other populations requires re-tuning with the same validation protocol and periodic recalibration if the learner distribution shifts.

5. Conclusion

In summary, this research developed and evaluated a methodological framework for personalizing educational content using machine learning and adaptive language model prompts. It was demonstrated that a straightforward linear regression model, trained on student study habits and background features, can accurately predict final exam performance (with over 80% of variance explained). Based on this, students' self-selected target scores were used to infer the appropriate difficulty level of content. By integrating these predictions into a generative AI pipeline via prompt engineering, the system produced customized explanatory materials: simplifying explanations for students likely to struggle and enriching content for those ahead of the curve.

The pilot evaluation, based on student surveys, indicated that most users were satisfied with the level of personalization when the system's predictions aligned with their expectations. This suggests that machine-learned user models combined with large language models can indeed tailor learning experiences at scale. While additional validation with larger datasets and real-world outcomes is needed, the findings support the hypothesis that machine learning can effectively personalize educational content.

This approach has practical implications for adaptive learning platforms. By automating the analysis of student data and the generation of bespoke instructional content, educators could offer more individualized support without a proportional increase in teacher effort. In conclusion, the results encourage further exploration of AI-driven personalization in education, with the goal of improving student engagement and learning outcomes.

Acknowledgements

This research has been funded by the Science Committee of the Ministry of Science and Higher Education of the Republic of Kazakhstan (Grant No. AP27510633).

Declaration on Generative AI

The authors have not employed any Generative AI tools.

References

[1] Chiu, T. K., Xia, Q., Zhou, X., Chai, C. S., & Cheng, M. (2023). Systematic literature review on opportunities, challenges, and future research recommendations of artificial intelligence in education. Computers and Education: Artificial Intelligence, 4, 100118. https://doi.org/10.1016/j.caeai.2022.100118.

[2] Ifenthaler, D., Majumdar, R., Gorissen, P., Judge, M., Mishra, S., Raffaghelli, J., & Shimada, A. (2024). Artificial intelligence in education: Implications for policymakers, researchers, and practitioners. Technology, Knowledge and Learning, 1–18. https://doi.org/10.1007/s10758-024-09747-0.

[3] Forero-Corba, W., & Bennasar, F. N. (2024). Techniques and applications of Machine Learning and Artificial Intelligence in education: a systematic review. RIED, 27(1). https://www.redalyc.org/journal/3314/331475280025/331475280025.pdf.

[4] Vincent-Lancrin, S. (2023). Towards a digital transformation of education: distance travelled and journey ahead. https://doi.org/10.1787/c74f03de-en.

[5] Matayoshi, J., Cosyn, E., & Uzun, H. (2021). Are we there yet? Evaluating the effectiveness of a recurrent neural network-based stopping algorithm for an adaptive assessment. IJAIED, 31(2), 304–336. https://doi.org/10.1007/s40593-021-00240-8.

[6] Mohammed, A. T., Velander, J., & Milrad, M. (2024). A Retrospective Analysis of Artificial Intelligence in Education (AIEd) Studies: Perspectives, Learning Theories, Challenges, and Emerging Opportunities. In: Burgos, D., et al. Radical Solutions for Artificial Intelligence and Digital Transformation in Education. Lecture Notes in Educational Technology. Springer, Singapore. https://doi.org/10.1007/978-981-97-8638-1_9.

[7] Tapalova, O., & Zhiyenbayeva, N. (2022). Artificial intelligence in education: AIEd for personalised learning pathways. Electronic Journal of e-Learning, 20(5), 639–653. https://doi.org/10.34190/ejel.20.5.2597.

[8] Sharma, S., Mittal, P., Kumar, M., & Bhardwaj, V. (2025). The role of large language models in personalized learning: a systematic review of educational impact. Discov Sustain 6, 243. https://doi.org/10.1007/s43621-025-01094-z.

[9] Almalawi, A., Soh, B., Li, A., & Samra, H. (2024). Predictive Models for Educational Purposes: A Systematic Review. Big Data and Cognitive Computing, 8(12), 187. https://doi.org/10.3390/bdcc8120187.

[10] Williamson, B., & Eynon, R. (2020). Historical threads, missing links, and future directions in AI in education. Learning, Media and Technology, 45(3), 223–235. https://doi.org/10.1080/17439884.2020.1798995.

[11] Jauhiainen, J. S., & Garagorry Guerra, A. (2024, November). Generative AI and education: dynamic personalization of pupils' school learning material with ChatGPT. Frontiers in Education, 9, 1288723. https://doi.org/10.3389/feduc.2024.1288723.

[12] Li, H., Xu, T., Zhang, C., Chen, E., Liang, J., Fan, X., … & Wen, Q. (2024). Bringing generative AI to adaptive learning in education. arXiv 2402.14601. https://doi.org/10.48550/arXiv.2402.14601.

[13] Mittelstadt, B., Wachter, S., & Russell, C. (2023). To protect science, we must use LLMs as zero-shot translators. Nature Human Behaviour, 7(11), 1830–1832. https://doi.org/10.1038/s41562-023-01744-0.

[14] Perković, G., Drobnjak, A., & Botički, I. (2024, May). Hallucinations in LLMs: Understanding and Addressing Challenges. In MIPRO 2024 (pp. 2084–2088). IEEE. https://doi.org/10.1109/mipro60963.2024.10569238.

[15] Yim, I. H. Y., & Su, J. (2025). Artificial intelligence (AI) learning tools in K-12 education: A scoping review. Journal of Computers in Education, 12(1), 93–131. https://doi.org/10.1007/s40692-023-00304-9.

[16] Fitria, T. N. (2021, December). Artificial intelligence (AI) in education: Using AI tools for teaching and learning process. In Prosiding Seminar Nasional & Call for Paper STIE AAS (pp. 134–147).

[17] Sheikh, M., Muhammad, A. H., & Naveed, Q. N. H. (2021). Enhancing usability of E-learning platform… SJESR, 4(2), 40–50. https://doi.org/10.36902/sjesr-vol4-iss2-2021(40-50).

[18] Liu, F., Liu, Y., Shi, L., Huang, H., Wang, R., Yang, Z., … & Ma, Y. (2024). Exploring and evaluating hallucinations in LLM-powered code generation. arXiv 2404.00971. https://doi.org/10.48550/arXiv.2404.00971.

[19] Lai Ng. Student-performance-factors [Dataset]. Kaggle. https://www.kaggle.com/datasets/lainguyn123/student-performance-factors.

[20] Brel', V. V., & Logvin, V. V. (2023). Education and Artificial Intelligence. https://elib.gstu.by/bitstream/handle/220612/29716/81-83.pdf.

[21] Cohen, I., Huang, Y., Chen, J., Benesty, J., … & Cohen, I. (2009). Pearson correlation coefficient. In Noise Reduction in Speech Processing, 1–4. https://doi.org/10.1007/978-3-642-00296-0_5.

[22] Yao, Y., & González-Vélez, H. (2025). AI-Powered System to Facilitate Personalized Adaptive Learning in Digital Transformation. Applied Sciences, 15(9), 1–27. https://doi.org/10.3390/app15094989.