V International Scientific and Technical Conference Actual Issues of Power Supply Systems

Predicting school students’ suitability for grade levels using the Light GBM model

AIPCP25-CF-ICAIPSS2025-00274 | Article

Predicting school students' suitability for grade levels using the Light GBM model

Erkaboy Samandarov [1,a)], Dostonbek Abduraimov[2], Ulmasbek Yuldashev[2], Farangiz Xakimova[3], Sayid Islikov[2]

[1] National Research University under "Tashkent Institute of Irrigation and Agricultural Mechanization Engineers", Tashkent, Uzbekistan

[2] Gulistan State University, Gulistan, Uzbekistan

[3] Gulistan State Pedagogical Inistitute, Gulistan, Uzbekistan

[a)] Corresponding author: samandaroverka09@gmail.com

Abstract. Currently, the usage of machine learning algorithms in the education system is increasing. Machine learning algorithms are used in every stage of education. This article considers the usage of the LightGBM machine learning algorithm to predict the grade level of school students in subjects. A web platform based on the LightGBM machine learning algorithm was created, and through this platform, 214 students from grades 6 to 11 of a school in the Khorezm region of the Republic of Uzbekistan took a test to determine their grade level in mathematics. The LightGBM model predicted the test results of the school students with high accuracy. The using of the LightGBM model in the educational process led to an increase in the quality of education, because this model determined the student's general level of knowledge in mathematics based on mathematical models, taking into account the answers given by each student to each test question. The results of this experiment are discussed in the article. This case helps determine what level of learning materials to provide to each student and what teaching methods to use.

INTRODUCTION

As a result of the development of modern technologies, machine learning algorithms are increasingly entering the education system. These technologies help to make the educational process effective and personalized for each student.

The goal of using machine learning algorithms in the educational process is to create educational programs tailored to the individual needs of each student. The artificial intelligence system analyzes the student's level of knowledge, mastery of subjects, and the topics that the student has difficulty mastering. As a result, each student receives a curriculum that is tailored to him. For example, a student who has difficulty mastering mathematics receives additional in-depth lessons, while a student with a high level of knowledge deals with more complex issues.

Machine learning algorithms allow for a more accurate assessment of the level of knowledge of students in each subject. Unlike traditional exams in the education system, educational systems based on machine learning algorithms constantly monitor of each student's educational activities. Machine learning-based learning platforms analyze student test scores, task completion time, and types of errors to determine the student's actual level of knowledge. This scenario allows teachers to improve their knowledge and provide each student with the necessary assistance in a timely manner [1].

To save teachers' time, machine learning-based learning platforms are used to check students' test scores and evaluate essays written by students. Using natural language processing (NLP) technologies, the learning platform analyzes written work, detects grammatical errors, and evaluates the quality of essay content. This allows teachers to spend more time directly interacting with students in the learning process [2].

Machine learning algorithms can predict students' future performance and the risk of academic failure with high accuracy in the learning process. The platform, built on machine learning algorithms, analyzes historical data and identifies which students need additional tutoring. This allows for early intervention in the educational process and reduces the likelihood of students dropping out.

Artificial intelligence-based virtual assistants conduct lessons with students 24/7 and answer their questions. These chatbots are capable of performing a wide range of tasks, from simple questions to complex explanations. Students can get help at any time and continue their learning process uninterrupted.

Machine learning algorithms are revolutionizing the field of education. Machine learning algorithms are making education more efficient, convenient, and tailored to the learning needs of each student. In the future, these technologies will develop and become an integral part of the education system.

LightGBM (Light Gradient Boosting Machine) is an efficient gradient boosting algorithm developed by Microsoft. The algorithm is designed to process large amounts of data quickly and accurately. LightGBM is a histogram-based algorithm that uses a leaf-wise strategy for building trees. This makes it faster than XGBoost and other algorithms. The algorithm is memory-efficient when working with large data sets and performs parallel calculations.

LightGBM machine learning algorithm is used in education for the following purposes: LightGBM is used to predict student performance, test scores, and academic success. It has the ability to identify students at risk based on attendance, task performance, and past performance [3].

Analyzes each student’s learning and recommends individual learning paths. This helps create adaptive learning systems. It offers the most suitable courses based on the student’s interests, past academic achievements, and career goals. It identifies students at high risk of dropping out of universities early and allows them to receive timely assistance.

Helps in grading essays and saves teachers time. LightGBM is an important tool in education for making data-based decisions and improving the quality of teaching.

Now, articles on the application of the LightGBM algorithm at different stages of education will be reviewed.

The paper transforms time series and behavioral data from the learning process into vectors using a “learning process model”. Then, it predicts the probability of a student dropping out of the course using the LightGBM machine learning algorithm. The authors first divide the behavior into stages of session duration, video viewing, Q&A activity, bookmarking and annotation. Then, the LightGBM machine learning algorithm trains the dataset and uses SHAP/feature-importance techniques to interpret the model, and the proposed approach shows small but consistent improvements in AUC, F1 and recall compared to the previous ones [4].

The paper is written to predict course outcomes for MOOC students and develop online course suggestions based on the predicted outcomes. First, the authors create a large feature set of behavioral and demographic features and model them using the LightGBM machine learning algorithm. The main focus is on the efficiency and speed of the LightGBM algorithm and its interpretation using SHAP. The article investigates the possibility of determining which lesson modules or types of lesson activities (video, forum, quiz) have the greatest impact on the result. Experiments are conducted on data from several MOOC courses. The LightGBM machine learning algorithm shows that it provides faster and more competitive results compared to other traditional methods. The authors present the results as practical recommendations for feedback to teachers and improvement of course design [5].

In this paper, the authors compare the use of AdaBoost, XGBoost, CatBoost, and LightGBM boosting algorithms in gradeifying students' mathematics performance. The focus of the paper is on the feature-selection stage. The optimal features for the LightGBM algorithm are selected using a combination of Fisher score and information gain with Recursive Feature Elimination (RFE). Experimental results show that the LightGBM machine learning algorithm shows significantly higher accuracy and F1 scores when selecting features correctly than other boosting methods. The paper also highlights the support for sparse features, speed, and low memory requirements of the LightGBM algorithm, and discusses its application to real-time recommendation and early warning systems in educational systems [6].

This paper in the MDPI Mathematics journal presents a comprehensive dataset that combines spatiotemporal, that is, time and space features of students. Based on this dataset, 6 models, namely XGBoost, LightGBM, RF, AdaBoost, DecisionTree, SVM models, are used to predict the final grade of a student. The authors determine the influence of features using SHAP to apply teachers' intervention strategies. As a result, although the LightGBM algorithm is faster and more efficient than its competitors, XGBoost performs better in some cases. An important aspect of the study is that the data is reliably estimated using KNN imputation, SMOTE balancing, and 20-fold hold-out re-randomization. In particular, the paper shows that semester grades are the most influential features and the integrated dataset gives the best results. This paper provides practical results for educational statistics and automated pedagogical recommendations [7]. In this paper, the authors propose to optimize the hyperparameters of the LightGBM model using nature-inspired metaheuristic algorithms FOX, GTO, PSO, SCSO, SSA. The authors' approach was tested with 5-fold cross-validation and 20 independent trials. The best results were observed with the combination of SCSO-LightGBM algorithms. SHAP was used for the interpretation of the model, and the most influential features were identified: attendance, hours studied, previous scores, and parental involvement. The study demonstrated the effectiveness of the LightGBM algorithm on real-world data and its more stable performance through metaheuristic optimization. The article provides practical conclusions on which factors should be used to intervene in educational management [8].

In the article, the authors compare the algorithms from the Decision Tree, SVM, Random Forest and boosting family (GB, XGBoost, CatBoost, LightGBM) using data from a university. Optuna was used for hyperparameter tuning and Isolation Forest was used for outlier detection. The results show that boosting algorithms, in particular LightGBM and CatBoost, performed better than traditional methods when optimized using Optuna. The article also identified important features using SHAP. The study provides practical recommendations for early detection systems and educational programs for student dropout [9]. In the article, the authors proposed an approach to predicting and interpreting student dropout by combining multiple source data with high privacy requirements. The authors used SMOTE to balance the grade and the LightGBM machine learning algorithm to express the local and global explanation of the model through SHAP. The article presents a detailed SHAP graph to show how specific features of the model affect dropout. [10].

In the paper, the authors used the k-means clustering algorithm to group students and then build a LightGBM model for each cluster to predict the outcomes of newly admitted students. The advantage of this hybrid approach is that it creates a model that is adapted to similar features and dynamic characteristics within each cluster. This method helps to protect against overfitting and heterogeneity compared to a single general model. The LightGBM algorithm has been shown to be effective on large data sets due to its speed and histogram splits, and to provide more accurate, stable predictions when combined with clustering. This work is useful for admissions profiling and resource planning for educational institutions [11]. The study examines which types of data can best predict dropout by comparing LMS (Moodle) logs, transcripts, and demographic data in a Finnish university dataset. The authors used several models, including the LightGBM algorithm, to test and evaluate the importance of LMS data. The results show that LMS can be a high-value feature for the model and can be well exploited by fast LightGBM algorithms. The paper proposes early warning and resource prioritization systems for universities from a policy and practice perspective [12].

A web-application was created based on the LightGBM machine learning algorithm. Through this web-application, 214 students from grades 6 to 11 of a school in the Khorezm region of the Republic of Uzbekistan took a test to determine their grade level in mathematics in the 2026 academic year. The results showed high efficiency. The paper also discusses the results of the experiment.

METHODS

Initially, the LightGBM machine learning algorithm is trained on data from the training dataset and the model is brought to a state where it predicts the knowledge level of a new student.

In the next stage, the LightGBM model is predicted based on the input data of the new student. The main goal of the algorithm is to assess the knowledge level of the student based on the answers given to 30 test questions by grade level from 6 to 11.

The input values of the LightGBM machine learning algorithm are the answers given by the student to 30 test questions for each student, that is, binary answers are given to the questions Q1, Q2, Q3, ..., Q30. Each answer accepts 1 - the correct answer or 0 - the incorrect answer [13].

In mathematical form, the vector of answers of each student is represented by the formula 1.

$$X = \left[x_1, x_2, x_3, ... x_{30}\right] \tag{1}$$

where:

$x_i \in \{0,\ 1\},\ i\ =\ 1,\ 2,\ ...,\ 30$.

Target variable, i.e. the range of values for the Grade grade level [6, 11] ∈ ℤ.

The mathematical representation of the Grade column is: $Y = Grade \in \{6, 7, 8, 9, 10, 11\}$.

The structure of the training dataset file used to gradeify the problem under consideration for the LightGBM machine learning algorithm is shown in Table 1.

TABLE 1. Structure of training dataset for LightGBM

Q1	Q2	Q3	…	Q30	Grade
0/1	0/1	0/1	…	0/1	6-11

Each row contains the data of one student. The dataset consists of N rows and can be written mathematically as follows:

$$D=\{(X_1,Y_1),(X_2,Y_2),...(X_n,Y_n)\} \tag{2}$$

where:

$X_i -$ $i -$ student feature vector;

$Y_i -$ $i -$ the student's actual grade level;

$N -$ The total number of students in the training dataset.

The LightGBM machine learning algorithm has strict rules for determining the level of knowledge of the learner. These rules are based on dividing the test questions into blocks of 5. To express this rule mathematically, 30 questions are divided into 6 blocks, and these questions are expressed as follows:

- Block 1: Q1 - Q5 (first 5 questions)
- Block 2: Q6 - Q10 (second 5 questions)
- Block 3: Q11 - Q15 (third 5 questions)
- Block 4: Q16 - Q20 (fourth 5 questions)
- Block 5: Q21 - Q25 (fifth 5 questions)
- Block 6: Q26 - Q30 (sixth 5 questions)

Formula 3 is used to calculate the number of correct answers for each block

$$S_k = \sum_{i=(k-1)*5+1}^{k*5} x_i \tag{3}$$

where:

$k = 1, 2, 3, 4, 5, 6$ - block number.

If at least 4 values in each block are 1 (true), the LightGBM algorithm predicts the value in the Grade column.

$Grade = 11 = (S_1 \geq 4) \wedge (S_2 \geq 4) \wedge (S_3 \geq 4) \wedge (S_4 \geq 4) \wedge (S_5 \geq 4) \wedge (S_6 \geq 4)$

i.e. Grade=11=[1,1,1,1,1,1,1,1,1,0,0,1,1,1,1,1,0,1,1,1,1,1,0,1,1,1,1,1,1,0]

We will also touch on the mathematical foundations of the LightGBM algorithm. We will consider the prediction of the knowledge level of students and their gradeification into gradees using the above mathematical expressions through the LightGBM machine learning algorithm. First, we will consider the theoretical foundations of the LightGBM algorithm. LightGBM is based on the gradient boosting algorithm. Gradient boosting is a method of creating a strong predictive model by sequentially combining weak learners. Each new model corrects the errors of previous models.

The function representing the LightGBM machine learning algorithm is:

$$F(X) = \sum_{m=1}^{M} \gamma_m h_m(X), \tag{4}$$

where:

$X -$ input data ($Q_1, Q_2, Q_3, ...Q_n$ questions);

$F(X) -$ final predictive function;

$h_m(X) -$ The response of the m th tree based on data X;

$\gamma_m -$ weight of m the tree

$M -$ number of trees.

For multi-grade gradeification problems, the cross-entropy loss function is commonly used. The mathematical expression of the loss function is:

$$L = -\sum_{i=1}^{n}\sum_{c=1}^{C} y_i^c \log(p_i^c), \tag{5}$$

where:

$n -$ number of samples;

$C -$ number of gradees (in our case $C = 6$, because Sinf ∈ {6,7,8,9,10,11});

$y_i^c -$ the $c-$grade belonging index of the i th sample (0 or 1);

$p_i^c -$ the probability that the model predicts that the i th sample belongs to grade c.

The job of the loss function is to measure how poorly the model is performing. It gives the amount of error as a number. The model predicts, for example, that the student's Grade Level is eight. But in reality, it's nine. The loss function calculates this difference and outputs a number - the bigger the difference, the bigger the loss.

At each iteration, the gradient is calculated. The formula for calculating the gradient is:

$$g_i = \frac{\partial L}{\partial F(X_i)}, \qquad h_i = \frac{\partial^2 L}{\partial F(X_i)^2}, \tag{6}$$

where:

$g_i -$ first-order gradient;

$h_i -$ second-order gradient (Hessian).

The gradient computation task tells the model which direction to move. It finds the best direction to minimize the error. If the model makes a prediction error, the gradient is computed and the error is reduced. For example, if the student's actual Grade level is nine, but the model predicts eight, the error is corrected by computing the gradient. LightGBM model uses a leaf-wise strategy. The optimal split point for each updated tree is found using the following formula:

$$Gain = \frac{1}{2}\left[\frac{G_L^2}{H_L+\lambda} + \frac{G_R^2}{H_R+\lambda} - \frac{\left(G_L+G_R\right)^2}{H_L+H_R+\lambda}\right] - \gamma, \tag{7}$$

where:

$G_L = \sum_{i\in I_L} g_i -$ gradient sum of the left side;

$G_R = \sum_{i\in I_R} g_i -$ the gradient sum of the right side;

$H_L = \sum_{i\in I_L} h_i -$ Hessian sum of the left side;

$H_R = \sum_{i\in I_R} h_i -$ Hessian sum of the right-hand side;

$\lambda -$ regularization parameter;

$\gamma -$ minimum division parameter.

$$w_j^* = \frac{\sum_{i\in I_j} g_i}{\sum_{i\in I_j} h_i + \lambda}, \tag{8}$$

In formula 8 $I_j - j -$ A collection of samples related to the leaf.

The value of each leaf of the tree is calculated. After the tree is built, it is determined how many students are collected at each endpoint (leaf) and how many errors they make. If most of the students in that leaf should have a high Grade column, the leaf weight will be a positive number. If it should be lower, it will be negative. Each leaf should have its own weight, because students in different leaves are corrected differently. For some students, a large correction is needed, for some, a small correction is enough. The weight is adjusted in this order. Thus, each leaf has its own weight and increases the accuracy of the model [14].

The model makes an initial prediction. For multi-grade gradeification:

$$F_0\left(X\right) = \arg\max_c \sum_{i=1}^{n} 1\left(y_i = c\right), \tag{9}$$

i.e., it takes the most common grade as the initial prediction.

In our case, an iteration means building a new decision tree and adding it to the model. The training of the model starts with an initial prediction and then gradually improves it over a hundred or so iterations. At each iteration, the model sees and corrects its mistakes from the previous iteration.

In each $m-$ iteration:

Step 1: The gradient and Hessian for each sample are calculated:

$$g_i^m = \frac{\partial L\left(y_i, F_{m-1}(X_i)\right)}{\partial F_{m-1}\left(X_i\right)}, \qquad h_i^m = \frac{\partial^2 L\left(y_i, F_{m-1}\left(X_i\right)\right)}{\partial F_{m-1}\left(X_i\right)^2}. \tag{10}$$

Step 2: A new decision tree h_m is constructed that minimizes the gradients.

Step 3: The optimal step length γ_m is found:

$$\gamma_m = \arg\min_{\gamma} \sum_{i=1}^{n} L\left(y_i, F_{m-1}\left(X_i\right) + \gamma h_m\left(X_i\right)\right). \tag{11}$$

Step 4: The model is updated:

$$F_m\left(X\right) = F_{m-1}\left(X\right) + \eta \gamma_m h_m\left(X\right), \tag{12}$$

where:
η – learning rate

Each iteration improves the model slightly, and in the end, the combined effect of all the iterations creates a very strong model.

Regularization is used to prevent the model from overfitting to the training dataset:

$$\Omega\left(F\right) = \lambda T + \frac{1}{2}\lambda \sum_{j=1}^{T} w_j^2, \tag{13}$$

where:
$T -$ number of leaves;
$w_j - j -$ leaf weight;

General purpose function:

$$L'\left(F\right) = \sum_{i=1}^{n} L\left(y_i, F\left(X_i\right)\right) + \sum_{m=1}^{M} \Omega\left(h_m\right). \tag{14}$$

The trained model calculates the probability of each grade for the given input values. For multi-grade gradeification, the softmax function is used:

$$p\left(y = c \mid X\right) = \frac{e^{(F_c(X))}}{\sum_{k=1}^{11} e^{F_k(X)}}, \tag{15}$$

where:

$F_c(X) - c$ - the model's prediction for the grade (logit);

The sum is taken over all possible grades (6 to 11). The grade with the highest probability is selected for the final prediction:

$$\hat{y} = \arg_{c \in \{6,7,8,9,10,11\}} p\left(y = c \mid X\right). \tag{16}$$

RESULTS

In the modern education system, it is important to accurately assess the level of knowledge of students. Therefore, a web application was created based on the LightGBM machine learning algorithm. This software product analyzes the level of knowledge of schoolchildren in mathematics, determines in a baschart which grade level the students' actual level of knowledge corresponds to. The software divides students into appropriate gradees and predicts their future results in mastering subjects.

The Django framework was used in the development of the web application. Django is a high-level web framework written in the Python programming language, allowing you to create fast and secure web applications. The main application in Django uses the TensorFlow machine learning library. TensorFlow is an open-source and widely used artificial intelligence library developed by Google. This library provides extensive opportunities for the practical application of complex mathematical calculations, deep learning models, and various machine learning algorithms.

Figure 1 shows the architecture of a web application based on the LightGBM machine learning algorithm. The architecture of the application consists of four main functional parts. Each part performs its own specific tasks and is closely interconnected. This modular structure increases the efficiency of the system, facilitates error detection, and allows for the addition of new functions in the future.

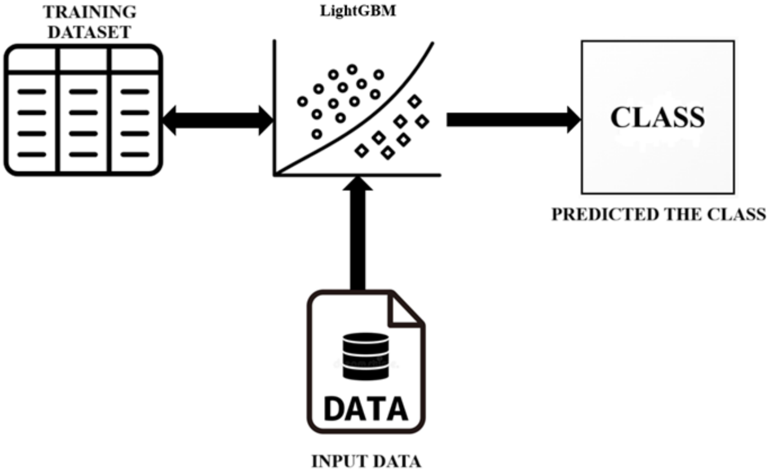

FIGURE1. The architecture of application

At the initial stage, the LightGBM machine learning algorithm reads data from the training dataset to determine which grade level the students' knowledge level in mathematics corresponds to. During this training process, the algorithm learns complex relationships between the level of knowledge in various mathematics topics and their suitability for the grade, based on the results of past students' test results, the problems they solved, the time they spent, the errors they made, and their final grades. The LightGBM algorithm creates a high-accuracy prediction model by sequentially building many decision trees based on the principle of gradient boosting [15].

After the model is fully trained, the next step is for the system to receive the new student’s data, test scores, and academic performance. The trained model analyzes this new data and assigns the new student to appropriate gradees based on their level of math proficiency. The system can also predict the student’s future learning dynamics. The algorithm uses the learned patterns in the prediction process to provide individual recommendations for each student, providing detailed information about the student’s strengths and weaknesses. This information helps teachers create individualized instructional plans for each student.

DISCUSSIONS

A web application was created based on the LightGBM machine learning algorithm. Through this web application, 214 students from grades 6 to 11 of a school in the Khorezm region of the Republic of Uzbekistan took a test in 2026 to determine their level of knowledge in mathematics. In terms of grades, 35 students in grade 6, 32 students in grade 7, 39 students in grade 8, 36 students in grade 9, 34 students in grade 10, and 38 students in grade 11 participated in the experiment [16].

TABLE 2. Confusion matrix for LightGBM

True/False	Grade 6	Grade 7	Grade 8	Grade 9	Grade 10	Grade 11
Grade 6	33	1	1	0	0	0
Grade 7	1	30	1	0	0	0
Grade 8	1	2	36	0	0	0
Grade 9	0	1	2	33	0	0
Grade 10	0	0	0	1	32	1
Grade 11	0	0	0	1	2	35

We will consider confusion matrix results for grade 6 to grade 11.

Grade 6: Additional metrics are calculated from the matrix above. Accuracy = 97.14%. For grade 6, Precision =96.97%, for grade 7, Precision =100%, for grade 8, Precision =100%. For grade 6, Recall = 100%, for grade 7, Recall = 100%. For grade 6, F1=98.46%, for grade 7, F1=100%, for grade 8, F1=100%.

Grade 7: Overall accuracy = 96.88%. For grade 6, Precision = 100%, Recall = 100%, F1 = 100%. For grade 7, Precision = 96.67%, Recall =100%, F1 = 98.31%. For grade 8, Precision = 100%, Recall =100%, F1 = 100%.

Grade 8: Overall accuracy = 89.74%. For grade 6, Precision = 100%, Recall = 100%, F1 = 100%. For grade 7, Precision = 100%, Recall = 96.4%, F1 = 100%. For grade 8, Precision = 94.44%, Recall = 94.44%, F1 =94.44%.

Grade 9: Accuracy = 94.44%. For grade 7, Precision = 100%, Recall =100%, F1 = 100%. For grade 8, Precision = 100%, Recall =100%, F1 = 100%. For grade 9, Precision =93.9%, Recall =100%, F1 = 96.87%.

Grade 10: Accuracy = 91.18%. For grade 9, Precision =100%, Recall =100%, F1 = 100%. For grade 10, Precision = 93.75%, Recall =100%, F1 =96.82%. For grade 11, Precision = 100%, Recall = 100%, F1 = 100%.

Grade 11: Accuracy =94.74%. For grade 9, Precision = 100%, Recall =100%, F1 = 100%. For grade 10, Precision 100%, Recall =100%, F1 = 100%. For grade 11, Precision =94.29%, Recall = 100%, F1 = 97.06% [17].

CONCLUSIONS

The confusion matrix results presented in the previous section demonstrated the performance of the machine learning model for grades 6–11. Below, we present the main conclusions and an overall analysis for each grade. We present the conclusions by class distribution. For grade 6, the model showed the highest results with 97.14% accuracy. All metrics are very high. Only one sample belonging to the "Other" class was misclassified, which is not very significant. For grade 7, the almost perfect result is 96.88% accuracy. Grade 7 students were completely identified with 100% Recall. Only one out of 30 predictions was wrong, which is a very good indicator. For grade 8, the results are good, but slightly lower with 89.74% accuracy. For grade 8, the Precision and Recall are 94.44%, which is a satisfactory result but can be improved. High accuracy for 9th grade is 94.44%. This shows that the model does not learn this class well. Good result for 10th grade is 91.18%. 100% Recall for 10th grade, but 2 additional samples were incorrectly classified as 10th grade. Grades 9 and 11 were identified with high accuracy.Very high accuracy for 11th grade is 94.74%. 100% Recall for 11th grade, all 33 students were correctly identified. Only 2 students from other grades were incorrectly classified as 11th grade. Strengths of the model The model classified all major classes with high accuracy (89–97%). Recall rates are very high, i.e. 96–100% in most classes, and almost all students belonging to their own class were correctly identified. F1-scores are balanced and in the high range (94–100%). The model’s weaknesses include a relatively low accuracy for grade 8 (89.74%), where errors are more common. In addition, in some classes there is confusion with neighboring classes, for example, grade 8 with grades 7 or 9. The model generally performed well and can be considered suitable for use in real-world conditions, but the model can be further improved by eliminating the weaknesses mentioned above.

REFERENCES

1. Mandal L. et al. Edu Vault: An Interactive, Multilingual, and Intelligent Topic-Conscious Video Discovery System for Enhanced Conceptual Learning Using Advanced NLP Techniques //Computación y Sistemas. – 2025. – T. 29. – №. 3. https://doi.10.13053/cys-29-3-5888

2. Samandarov E. Classification of the psychological condition of students using XGBoost machine learning model //AIP Conference Proceedings. – AIP Publishing LLC, 2025. – T. 3377. – №. 1. – C. 040004. https://doi.org/10.1063/5.0299675

3. Kalandarov, A., Kalandarov, A., Abduraimov, D., & Anorbayev, M. (2024, November). Mathematical model of the coupled problem of thermoelasticity in stresses. In *AIP Conference Proceedings* (Vol. 3244, No. 1, p. 020013). AIP Publishing LLC.

4. H. Nie, Y. Wen, B. Cao, B. Liang. MOOC Dropout Prediction Using Learning Process Model and LightGBM Algorithm. *Computer Supported Cooperative Work and Social Computing — ChineseCSCW 2023*, Communications in Computer and Information Science, vol. 2012 (2024), pp. 121–136. https://doi.org/10.1007/978-981-99-9637-7_9

5. Y. Ren, J. Wang, J. Hao, J. Gan, K. Chen. MOOC Performance Prediction and Online Design Instructional Suggestions Based on LightGBM. *Proceedings of ML4CS*, Lecture Notes / Conference Proceedings (2022). https://doi.org/10.1007/978-3-031-20102-8_39

6. T. Hamim, F. Benabbou, N. Sael. Student Profile Modeling Using Boosting Algorithms. *International Journal of Web-Based Learning and Teaching Technologies (IJWLTT)*, Vol. 17, No. 5 (2022), pp. 1–13. https://doi.org/10.4018/IJWLTT.20220901.oa4

7. Z. Luo, J. Mai, C. Feng, D. Kong, J. Liu, Y. Ding et al. A Method for Prediction and Analysis of Student Performance That Combines Multi-Dimensional Features of Time and Space. *Mathematics* (MDPI) 12(22) (2024): 3597. https://doi.org/10.3390/math12223597

8. Abukader, A. Alzubi, O. R. Adegboye. Intelligent System for Student Performance Prediction: An Educational Data Mining Approach Using Metaheuristic-Optimized LightGBM with SHAP-Based Learning Analytics. *Applied Sciences* 15(20) (2025): 10875. https://doi.org/10.3390/app152010875

9. Villar, C. Robledo Velini de Andrade. Supervised machine learning algorithms for predicting student dropout and academic success: a comparative study. *Discover Artificial Intelligence* 4 (2024): Article 2. https://doi.org/10.1007/s44163-023-00079-z

10. H. Liu, M. Mao, X. Li, J. Gao. Model interpretability on private-safe oriented student dropout prediction. *PLOS ONE* 20(3) (2025): e0317726. https://doi.org/10.1371/journal.pone.0317726

11. T. Shen, Y. Zhang. A Performance Prediction Model for Newly Enrolled University Students Based on The Fusion of K-Means and LightGBM Algorithms. *Proceedings of the 2024 3rd International Conference on Cloud Computing, Big Data Application and Software Engineering (CBASE)*. DOI: 10.1109/CBASE64041.2024.10824518.

12. M. Vaarma, H. Li. Predicting student dropouts with machine learning: An empirical study in Finnish higher education. *Technology in Society* 76 (2024) 102474. https://doi.org/10.1016/j.techsoc.2024.102474

13. Samandarov, E., & Abduraimov, D. (2025, September). Classification the level of knowledge of students at school using the Naive Bayes machine learning algorithm. In *AIP Conference Proceedings* (Vol. 3356, No. 1, p. 030001). AIP Publishing LLC.

14. Samandarov, E., Abduraimov, D., Xudayberdiyev, A., Normatova, M., Butaboyev, A., To'ychiyeva, Z., & Taniberdiyev, A. (2025, June). Comprehensive review of educational platform for assessing and classifying students' knowledge levels utilizing machine learning. In *Fourth International Conference on Digital Technologies, Optics, and Materials Science (DTIEE 2025)* (Vol. 13662, pp. 232-242). SPIE.

15. Saidov, J., Qudratov, A., Islikov, S., Normatova, M., & Monasipova, R. (2023). Problems of Competency Approach in Developing Students' Creativity Qualities for Creating a Database. Journal of Higher Education Theory and Practice, 23(1). https://doi.org/10.33423/jhetp.v23i1.5786

16. Eshbaevich, T. D., Bakhronovich, N. M., Abdubanopovich, Y. U., & Sherali's, S. I. (2020). Resource support of distance course information educational environment. Journal of Critical Reviews, 7(5), 399-400.

17. Samandarov E. THE ARCHITECTURE OF EDUCATIONAL PLATFORM BASED ON MACHINE LEARNING //Journal of Mathematics, Mechanics & Computer Science. – 2024. – T. 124. – №. 4. https://doi.org/10.26577/JMMCS2024-v124-i4-a7