Online-Journals.org

iJIM International Journal of Interactive Mobile Technologies

iJIM | eISSN: 1865-7923 | Vol. 20 No. 17 (2026) | OPEN ACCESS

https://doi.org/10.3991/ijim.v20i17.62574

PAPER

Comparative Evaluation of Machine Learning Models for Early Prediction of Student Academic Performance Using the Open University Learning Analytics Dataset

Assel Omarbekova, Aizhan Nazyrova, Gulmira Bekmanova, Zhanar Lamasheva, Rakhila Turebayeva (✉), Zhainagul Abzal

L.N. Gumilyov Eurasian National University, Astana, Kazakhstan

turebayeva_rd@enu.kz

ABSTRACT

Early identification of students at risk of academic failure is a central problem in learning analytics, yet the multi-table, behaviorally heterogeneous, and class-imbalanced nature of educational data complicates reliable prediction. This study develops and comparatively evaluates four supervised machine learning models—logistic regression, support vector machine (SVM) with a radial basis function kernel, random forest, and extreme gradient boosting (XGBoost)—for multi-class prediction of final student outcome on the Open University Learning Analytics Dataset (OULAD), comprising 32,593 anonymized student records and more than ten million virtual-learning-environment interaction logs. A reproducible, leakage-controlled feature-engineering pipeline is proposed: final examination records are excluded, and virtual-learning-environment activity is restricted to the first 120 days of each presentation, yielding 85 demographics, assessment, and behavioral features. All models are trained on an identical stratified 80/20 split and assessed within a unified evaluation framework using accuracy, weighted and macro F1-score, balanced accuracy, per-class recall, and stratified cross-validation. XGBoost achieved the strongest overall performance (accuracy 0.7302, macro F1 0.6658, balanced accuracy 0.6555, and cross-validated accuracy 0.7314), followed closely by random forest and logistic regression, while SVM, despite the weakest aggregate scores, produced the highest recall for the practically critical Fail class (0.4026). The analysis demonstrates that aggregate accuracy substantially overstates the usefulness of these models for risk detection and that the fail class remains the principal bottleneck. A deployed Streamlit prototype illustrates operationalization. The contribution lies in the leakage-aware early-window pipeline and the imbalance-sensitive, algorithm-aware comparison that together expose where predictive value is gained and lost.

KEYWORDS

learning analytics, student performance prediction, machine learning, extreme gradient boosting (XGBoost), class imbalance, OULAD, educational data mining, early-warning systems

Omarbekova, A., Nazyrova, A., Bekmanova, G., Lamasheva, Z., Turebayeva, R., Abzal, Z. (2026). Comparative Evaluation of Machine Learning Models for Early Prediction of Student Academic Performance Using the Open University Learning Analytics Dataset. *International Journal of Interactive Mobile Technologies (iJIM)*, 20(17), pp. 110–136. https://doi.org/10.3991/ijim.v20i17.62574

Article submitted 2026-05-06. Revision uploaded 2026-06-24. Final acceptance 2026-06-24.

1 INTRODUCTION

The digital transformation of higher education has fundamentally expanded the capacity of institutions to collect and process large volumes of data about the learning process. Universities now accumulate detailed records of student enrollment, demographic background, continuous-assessment outcomes, and fine-grained interaction with virtual learning environments. The systematic analysis of these data underpins the field of learning analytics, whose objective is to improve teaching and learning by transforming raw educational traces into actionable insight [1]. Within this field, the prediction of academic performance has emerged as both a scientifically demanding and a practically consequential task, because the timely identification of students who are likely to struggle enables institutions to intervene before academic difficulty becomes irreversible [2].

Academic performance is a multidimensional construct. Although it is most often operationalized through grades, grade point average, or standardized examination results, the scholarly literature emphasizes that performance also reflects cognitive skills, the depth of knowledge acquisition, critical thinking, and the capacity to apply learning in practice. For predictive modeling, however, a measurable and consistently recorded outcome is required, and the final result of a module—typically distinguishing distinction, pass, fail, and withdrawal—provides such a target. Predicting this outcome from data available early in a course is valuable precisely because it can be acted upon: at-risk students can be offered academic counseling, supplementary tuition, or psychological support, and teaching can be reorganized to address emerging difficulties.

Machine learning offers a natural methodology for this problem [3]. Supervised algorithms can learn complex, multifactorial relationships between learning behavior, demographic characteristics, and assessment outcomes on the one hand, and final performance on the other. A substantial body of research has applied classification methods—ranging from interpretable linear models to ensemble and kernel-based methods—to educational data, frequently reporting encouraging predictive accuracy [4]. Nonetheless, the complexity and multifactorial character of educational data continue to make accurate prediction a non-trivial scientific problem. Three difficulties recur. First, educational datasets are typically distributed across multiple relational tables that must be integrated without introducing information that would not be available at prediction time. Second, outcome classes are markedly imbalanced, with passing and withdrawing students far outnumbering those who fail, so that high aggregate accuracy can mask poor detection of the minority but most important classes. Third, the diversity of algorithmic families means that no single model is uniformly superior, and a fair comparison requires identical data, identical splits, and a common, imbalance-sensitive evaluation protocol.

This paper addresses these difficulties through a systematic comparative study. The principal aim is to construct machine learning models for predicting student academic performance on a single, well-characterized dataset, to evaluate their effectiveness, and through comparative analysis, to identify the model that delivers the highest predictive quality. The Open University Learning Analytics Dataset (OULAD) is used as the empirical basis, and four algorithms—logistic regression, support vector machine (SVM), random forest, and extreme gradient boosting (XGBoost)—are trained and compared under identical conditions.

The study makes the following contributions. (i) A reproducible, leakage-controlled feature-engineering pipeline is proposed in which final examination records are removed and behavioral activity is restricted to an early window of the

first 120 days of each presentation, so that prediction reflects genuinely early-available information rather than proxies of the outcome. (ii) Four representative machine learning families are compared on an identical stratified split within a unified evaluation framework that combines aggregate metrics with imbalance-sensitive metrics—macro F1, balanced accuracy, and per-class recall—thereby exposing differences that aggregate accuracy conceals. (iii) The analysis isolates the recall of the Fail class as the central practical bottleneck of the entire model family and connects this finding to concrete mitigation strategies. (iv) A working web prototype demonstrates how the trained models can be operationalized for use by teaching staff. The intended contribution does not lie in the four algorithms themselves—logistic regression, SVM, random forest, and XGBoost are standard methods and are not claimed as novel—nor in the headline result that XGBoost performs best. It lies rather in the evaluation design: the combination of a leakage-controlled early-prediction window with an imbalance-sensitive, per-class protocol that demonstrates how aggregate accuracy conceals weak detection of failing students and that makes explicit the trade-off between overall performance and at-risk-student detection. It is this framing, rather than the model comparison in isolation, that the introduction, related work, discussion, and conclusion are organized around.

The motivation for such systems is reinforced by the wider maturation of learning analytics within institutional practice. As universities increasingly integrate digital technologies into teaching and assessment processes, the prediction of academic outcomes is evolving from a research topic into a practical instrument for monitoring and supporting student success. Recent studies have explored intelligent systems for student performance evaluation [5] and AI-enhanced educational environments that improve student engagement and academic outcomes [6], highlighting the growing demand for predictive models that are both accurate and actionable. At the same time, the value of a predictive model is realized only when its outputs are timely and actionable, which places a premium on early prediction from information available well before final assessment—a requirement that directly shapes the methodological choices made in this study.

2 RELATED WORK

The prediction of student performance has been studied extensively within educational data mining and learning analytics. Systematic reviews of the field report that classification is the dominant task formulation, that demographic, assessment, and behavioral features are the most informative predictor groups, and that ensemble and tree-based methods, SVMs, and neural networks are the most frequently applied algorithms [2]. These reviews also note a persistent methodological concern: studies differ so widely in their datasets, feature sets, target definitions, and evaluation metrics that direct comparison of reported accuracies across papers is rarely meaningful. Recent work has further demonstrated the growing role of intelligent educational systems and AI-supported learning environments in monitoring and improving academic performance [5], [6]. This observation motivates controlled, single-dataset comparisons such as the one presented here.

A recurrent finding in the literature is that algorithmic complexity does not guarantee superior performance. Simple, interpretable models such as logistic regression and linear regression often perform competitively with more elaborate ensembles, and in some studies linear models have matched or exceeded random forests and nearest-neighbor classifiers on particular educational datasets [7]. The relative ranking of models is therefore strongly dependent on the structure of

the data, the quality of feature engineering, and the treatment of class imbalance, rather than on the nominal sophistication of the algorithm. Consequently, comparative studies that hold the dataset and preprocessing constant are essential for drawing reliable conclusions about which methods are appropriate for a given problem.

Behavioral data drawn from virtual learning environments have proven particularly valuable for early prediction [8]. The frequency with which students access materials, their participation in forums, the number of resources visited, and the timing of activity all carry a signal about engagement and, by extension, about eventual outcomes [9]. However, the use of behavioral features introduces a methodological hazard: activity recorded late in a course is partly a consequence of the outcome itself, and incorporating it inflates apparent accuracy while undermining genuine generalization. The careful temporal scoping of behavioral features is therefore a methodological prerequisite for credible early prediction and is a design principle adopted explicitly in this study.

A further methodological theme concerns the treatment of class imbalance, which is pervasive in educational outcome data. The literature on learning from imbalanced data establishes that classifiers trained to maximize overall accuracy tend to favor majority classes [10] and that resampling techniques such as synthetic minority oversampling can materially improve the recovery of minority classes [11]. In educational settings the minority classes—failing or withdrawing students—are frequently the ones of greatest interest, so the choice of evaluation metric is not a technical detail but a determinant of whether a study answers the question it purports to address. Reviews of supervised techniques for examination-performance prediction likewise stress that the comparative ranking of methods can invert depending on whether evaluation emphasizes aggregate accuracy or per-class performance [12].

The OULAD has become a widely used benchmark for this line of research because it is large, openly licensed, rigorously anonymized, and well documented [13]. Released in 2017, it integrates demographic, assessment, and interaction data for tens of thousands of students across multiple module presentations, and its reliability has been examined statistically against later cohorts. Its relational structure, however, requires substantial integration and feature construction before modeling, and its outcome distribution is imbalanced—properties that make it a realistic testbed for the difficulties described in the Introduction.

Prior OULAD-based prediction studies vary considerably along the dimensions that matter most for early warning, and the present study can be positioned against them on seven axes. On temporal leakage, many OULAD studies draw on whole-course clickstream activity and on final-assessment information, whereas the pipeline used here excludes final-examination records and confines behavior to an early window. On the early-prediction window, studies differ in how much of a presentation they observe; the present work fixes a single early 120-day window rather than using the full presentation. On assessment features, only intermediate tutor- and computer-marked results are aggregated, avoiding outcome-coupled examination scores. On VLE clickstream aggregation, behavior is summarized into volume, intensity, regularity, and breadth features restricted to the early window. On class imbalance, the present study foregrounds macro F1, balanced accuracy, and per-class recall instead of reporting aggregate accuracy alone. On per-class performance, Fail and Withdrawn recall are reported separately, and the Fail class is identified as the dominant bottleneck. On the comparison protocol, all four models share an identical stratified split, an algorithm-appropriate but common preprocessing scheme, and a single evaluation framework. Taken together, these choices distinguish the

present study from OULAD work that maximizes aggregate accuracy on whole-course data, and they delimit what it adds. This comparison is set out in Table 1.

publication of your article.

Table 1. Positioning of the present study against representative prior OULAD-based prediction studies

Study	Models Compared	Early Window/ Leakage Control	Class-Imbalance Handling	Per-Class (Fail) Reporting	Reported Result
Adnan et al. (2021) [14]	Multiple classic ML (incl. RF, SVM, gradient boosting, neural nets)	Predicts at several percentages of course length (0–100%); early windows are time-sliced	Oversampling to balance classes	Dropout and result prediction by window (AUC); Fail-class recall is not isolated	VLE-interaction features are strongest; gradient boosting AUC is up to 0.91 (dropout) 0.93 (result)
Torkhani and Rezgui (2025) [15]	LR, DT, RF, SVM; CNN, RNN, LSTM	Whole-course MOOC data; leakage not explicitly controlled	Not explicitly addressed	Aggregate accuracy, precision, recall, and F1	LSTM best (accuracy 83.4%, precision 82.2%)
Shala Riza et al. (2025) [16]	BiLSTM-RNN with attention	Early identification via temporal sequence modeling	Advanced resampling (notes SMOTE may not preserve temporal patterns)	At-risk detection; accuracy/ precision/recall/F1	BiLSTM-RNN is best across metrics.
This study	LR, SVM (RBF), RF, XGBoost	Fixed early 120-day window; final-examination records excluded	Stratification + imbalance-sensitive metrics; no data-level rebalancing in main experiments	Fail and Withdrawn recall were reported explicitly; Fail was identified as the bottleneck	XGBoost is best overall (acc. 0.730, macro F1 0.666); Fail recall is low (0.34–0.40); SVM has the best Fail recall (0.40)

Table 2 presents a summary of representative strands of prior research on student performance prediction and highlights how the present study is positioned within this literature.

Table 2. Summarizes representative strands of prior work and situates the present study within them

Research Focus	Typical Data/Context	Methods Reported	Principal Finding
Systematic reviews of performance prediction	Heterogeneous institutional and MOOC datasets	Decision trees, Random Forest, SVM, ANN	Classification dominates; comparability across studies is limited
Algorithm comparison studies	Course-level academic records	Linear/logistic regression, Random Forest, KNN	Simple models often rival complex ensembles; ranking is data-dependent
Behavior-based early prediction	VLE interaction logs	Logistic regression, ensembles	Engagement features are predictive but prone to temporal leakage
OULAD-based studies	32,593 students, 7 relational tables	Various supervised classifiers	Strong benchmark; imbalance and integration are recurring challenges
Operational early-warning systems	Institutional digital profiles	BI and analytics pipelines	Predictive value depends on timely, actionable features

In summary, prior work establishes that supervised classification on integrated educational data is an effective approach, that model ranking is contingent on data and preprocessing rather than on algorithmic complexity alone, that behavioral features are powerful but must be temporally controlled, and that class imbalance demands evaluation beyond aggregate accuracy. The present study responds to each of these themes by combining a leakage-controlled early-window pipeline with an imbalance-sensitive, algorithm-aware comparison on a single benchmark dataset.

3 MATERIALS AND METHODS

The study follows a data-driven methodology composed of several interdependent stages: preparation and integration of the raw data, preprocessing, construction of the feature space, training of the machine learning models, and evaluation and comparison of their results. The problem is formulated as a supervised multiclass classification task in which the target is the final module result. All stages were implemented in Python using the pandas, NumPy, scikit-learn, XGBoost, Matplotlib, and Streamlit libraries [17], and the project was organized into clearly separated components for raw data, processed data, analysis notebooks, trained models, and results [18]. The overall architecture of the system is shown in Figure 1.

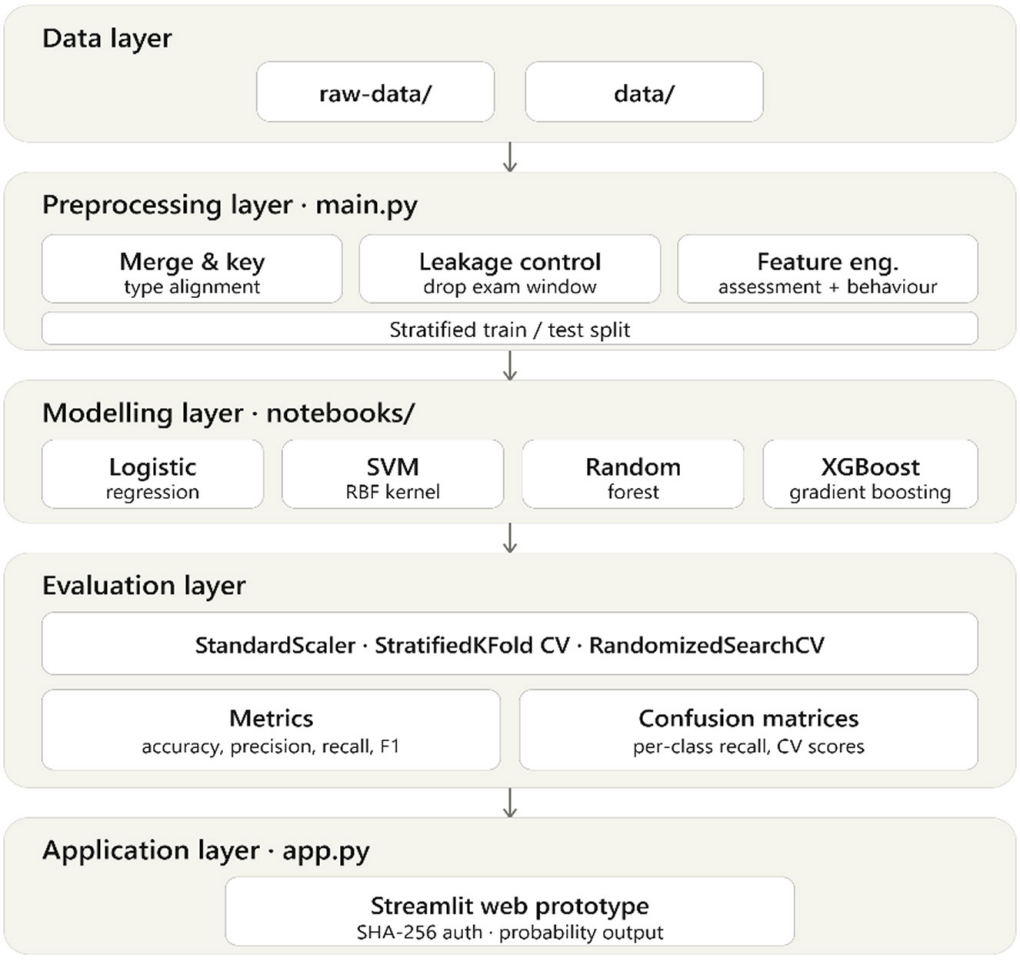

Fig. 1. Layered architecture of the proposed system, from raw OULAD tables through the leakage-controlled preprocessing pipeline and the four-model training layer to the unified evaluation layer and the Streamlit application prototype

The architecture separates concern into five layers. The data layer stores the raw OULAD tables. The preprocessing layer, implemented in a single orchestration script, integrates the tables, aligns key types, applies the leakage controls described below, engineers behavioral and assessment features, and encodes categorical variables. The modeling layer trains the four algorithms on the resulting feature matrix using a common training procedure. The evaluation layer computes a fixed set of metrics, confusion matrices, and cross-validation scores for every model, ensuring that

comparisons are made under identical conditions. The application layer exposes the trained models through a web interface for practical testing.

3.1 Dataset

The empirical basis of the study is the OULAD, a structured collection of anonymized data about students of the Open University, United Kingdom, released in 2017 under a CC-BY 4.0 license. Teaching at the institution is delivered through a virtual learning environment, and all student interactions with learning materials are systematically recorded. The dataset was constructed through a documented sequence of stages—data collection from a centralized repository, selection of representative modules, anonymization, and certified publication—designed to guarantee both data quality and compliance with privacy requirements. Anonymization employed k-anonymity with k = 5, a suppression limit of 0.7, and a re-identification risk capped at 0.05, after which the population comprised 32,593 students.

Seven modules were selected according to criteria requiring more than 500 students per presentation, at least two presentations, available VLE data, and a non-trivial number of unsuccessful students; four belong to the STEM domain and three to the social sciences. The selected modules and their enrolments are summarized in Table 3.

Table 3. Modules included in the OULAD and their characteristics

Module	Domain	Presentations	Students
AAA	Social sciences	2	748
BBB	Social sciences	4	7,909
CCC	STEM	2	4,434
DDD	STEM	4	6,272
EEE	STEM	3	2,934
FFF	STEM	4	7,762
GGG	Social sciences	3	2,534

The dataset is organized around the student as the unit of analysis and comprises seven interrelated tables linked by student and module identifiers. Together they describe demographic characteristics, registration, assessment definitions, assessment results, and day-level interaction with the virtual learning environment, the latter amounting to more than 10.6 million records. The tables used in this study and their sizes are listed in Table 4.

Table 4. Relational tables of the OULAD used in the study

Table	Description	Rows
studentInfo	Demographic information and final module result	32,593
courses	Available modules and their presentations	22
studentRegistration	Registration and de-registration timing	–
assessments	Assessment definitions, type, and weight	206
studentAssessment	Student assessment results	173,912
studentVle	Day-level VLE interaction (clicks)	10,655,280
vle	Catalogue of available VLE materials	6,364

A reliability analysis reported by the dataset authors compared the 2013–2014 cohorts with a later 2015 cohort using the chi-squared and Wilcoxon rank-sum tests; the resulting p-values, ranging from 0.15 to 0.93, indicated no statistically significant distributional differences and supported the dataset as a representative basis for modeling.

3.2 Preprocessing and leakage control

Because the OULAD consists of several large interrelated tables, integration, cleaning, transformation, and conversion to a model-ready format were performed systematically. Five source tables—studentInfo, studentAssessment, assessments, studentVle, and vle—were loaded, and the presence of all required files was verified automatically before processing. To manage the large data volume, column types were optimized: integers and floats were down-cast, and object columns with a unique-value ratio below 0.5 were converted to the categorical type, reducing memory usage and improving processing efficiency. Key columns used for joins—code_module, code_presentation and id_student—were type-aligned to prevent merge errors, with the student identifier cast to a nullable integer and the remaining keys cast to object type.

Two deliberate methodological controls were imposed to prevent data leakage, that is, the inadvertent use of information that would not be available at the time an early prediction is required. First, records corresponding to final examinations were removed from the assessment data, because examination outcomes are directly coupled to the final result and their inclusion would allow a model to infer the target indirectly rather than learning genuine patterns. Only intermediate assessment results were therefore used. Second, virtual-learning-environment activity was restricted to the first 120 days of each presentation through an early-window limit. Activity recorded near the end of a presentation partly reflects the outcome itself; restricting the window to early activity preserves the objectivity and reliability of early prediction. The 120-day boundary was chosen to balance two competing requirements. OULAD presentations run for approximately nine months, so a 120-day window corresponds to roughly the first half of a typical presentation and precedes the bulk of the summative assessment schedule; this leaves a substantial portion of the course during which institutional interventions—academic counseling, supplementary tuition, or targeted outreach—can still influence a student's trajectory. A shorter window would yield an earlier warning but rest on sparser behavioral evidence, whereas a longer window would approach mid- or late-course prediction and erode the practical value of advance notice. Rather than assume that this single window is adequate, its sensitivity is examined directly: all four models are additionally evaluated on early windows of 30, 60, 90, and 120 days, and the results are reported in Section 4.7. That analysis quantifies how predictive performance, and in particular Fail-class recall, varies with the length of the observation window and shows that the 120-day choice does not materially disadvantage early detection. These controls are central to the validity of the study and distinguish it from analyses that use whole-course behavioral data.

3.3 Feature engineering

From the filtered assessment data, aggregate features describing each student's assessment behavior were computed by grouping on the module, presentation,

and student keys. These include the number of assessments attempted and submitted, the mean, maximum, minimum, and standard deviation of scores, and the mean and total assessment weight; pivoting by assessment type produced separate features for tutor-marked and computer-marked assessments. From the early-window VLE data, behavioral features were computed: total, mean, and maximum clicks, the standard deviation of clicks, the number of active days; and the number of unique resources visited, with pivoting by activity type adding resource-specific click counts. The principal engineered features are summarized in Table 5.

Table 5. Principal engineered assessment and behavioral features

Group	Feature	Description
Assessment	assessment_count	Total number of assessments attempted
Assessment	submitted_assessments_count	Number of assessments submitted
Assessment	mean_score/max_score/min_score	Central tendency and extremes of scores
Assessment	std_score	Standard deviation of scores
Assessment	mean_weight/total_weight	Mean and total assessment weight
Assessment	tma_score_mean	Mean score on tutor-marked assessments
Behavioral	total_clicks/mean_clicks/max_clicks	Volume and intensity of VLE activity
Behavioral	std_clicks	Variability of daily click counts
Behavioral	activity_days	Number of distinct active days
Behavioral	unique_resources_visited	Number of distinct VLE resources accessed

The assessment and behavioral feature sets were merged with the demographic table to form a single dataset in which each student is described by demographic, academic, and digital-activity attributes. Missing values were handled according to a principled rule: numeric columns were filled with zero, reflecting the genuine absence of an activity or record for some students, while categorical columns were filled with an explicit 'Unknown' category. The student identifier was removed because it carries no predictive meaning and could harm generalization. Categorical predictors were converted to numeric form by one-hot encoding, retaining all categories as separate binary columns. The pipeline produced a final feature matrix of 32,593 records described by 85 features.

3.4 Target variable and data partitioning

The target variable is the final module result, comprising four classes: Distinction, Pass, Fail, and Withdrawn. The dataset is partitioned into training and test subsets in an 80/20 ratio using stratified sampling with a fixed random seed of 42, so that the class proportions of the target are preserved in both subsets and the results are reproducible. This yields 26,074 training records and 6,519 test records. The class distribution is markedly imbalanced: Pass and Withdrawn dominate, whereas Distinction is comparatively rare. This imbalance, preserved consistently across the splits by stratification (refer to Table 6), justifies the inclusion of imbalance-sensitive evaluation metrics introduced in Section 3.7.

Table 6. Distribution of the target variable across the full dataset and the stratified training and test partitions

Class (Final_result)	Full Dataset	Training Set	Test Set
Pass	12,361	9,889	2,472
Withdrawn	10,156	8,125	2,031
Fail	7,052	5,641	1,411
Distinction	3,024	2,419	605
Total	32,593	26,074	6,519

3.5 Machine learning models

Four supervised algorithms were selected to represent distinct families: a linear probabilistic model, a kernel-based margin classifier, a bagging ensemble of decision trees, and a gradient-boosting ensemble. The selection is motivated by their complementary properties—logistic regression for interpretability, SVM for effectiveness in high-dimensional spaces, random forest for capturing non-linear relationships, and XGBoost for high predictive accuracy.

Logistic Regression is one of the most widely used supervised methods for classification [19]. It estimates the probability that an observation belongs to a class by applying the logistic (sigmoid) function to a linear combination of the input features so that the model output lies between 0 and 1 and is interpretable as a class probability:

$$P(y=1|x)=\frac{1}{1+e^{-z}} \tag{1}$$

where the linear term z is defined as

$$z=w_0+w_1x_1+w_2x_2+\cdots+w_nx_n \tag{2}$$

and the coefficients w_i quantify the influence of each feature x_i on the outcome, giving the model its characteristic interpretability. L1 and L2 regularization are used to limit overfitting.

The Support Vector Machine seeks the optimal separating hyperplane that maximizes the margin between classes, which improves generalization to unseen data [20]. For linearly separable data the model satisfies the constraint

$$y_i(w^Tx_i+b)\geq 1 \tag{3}$$

and the decision boundary is given by

$$f(x)=\text{sign}\,(w^Tx+b) \tag{4}$$

When the data are not linearly separable, kernel functions map them into a higher-dimensional space in which linear separation becomes possible; the radial basis function kernel is used here. SVMs are effective in high-dimensional feature spaces, although their computational cost grows with dataset size and they are sensitive to parameter selection.

Random forest is an ensemble method that builds many decision trees and combines their outputs [21]. For classification, the final prediction is obtained by majority vote across the trees:

$$\hat{y}=\text{mode}\{T_1(x),T_2(x),\ldots,T_M(x)\} \tag{5}$$

Bootstrap sampling and random feature selection make the individual trees largely independent, which improves generalization and robustness to noise; the method also yields measures of feature importance and captures complex non-linear relationships.

XGBoost is an advanced form of gradient boosting distinguished by high predictive accuracy [22]. It adds new trees iteratively, each correcting the errors of its predecessors:

$$\hat{y}^{(t)} = \hat{y}^{(t-1)} + f_t(x) \tag{6}$$

and the objective function combines a loss term with a regularization term that controls model complexity:

$$\mathcal{L} = \sum_i l(y_i, \hat{y}_i) + \sum_k \Omega(f_k) \tag{7}$$

where l is the loss function and Ω penalizes model complexity. By combining gradient boosting with L1 and L2 regularization and supporting parallel computation, XGBoost achieves strong results on structured data, which motivates its inclusion as the most powerful candidate in the comparison.

3.6 Unified training architecture

A single training architecture was applied to all four models so that their results could be compared objectively. The architecture proceeds through seven stages: initialization of project directories; loading of the identical training and test partitions; algorithm-appropriate preprocessing; training of a baseline model; unified evaluation; stratified cross-validation; and hyperparameter optimization followed by retraining. Because the train/test split is performed once and reused, every model is compared under identical initial conditions on a training matrix of 26,074 by 85 and a test matrix of 6,519 by 85.

Preprocessing is adapted to the nature of each algorithm while remaining within the common framework. Logistic Regression, SVM, and XGBoost are trained on standardized features using a scaler fitted on the training set and applied to the test set because these models are sensitive to feature scale; for XGBoost, standardization is not strictly required but is applied for consistency and stability. Random Forest, being a tree-based method that is invariant to monotonic feature scaling, is trained on the unscaled features. This scientifically grounded distinction respects the assumptions of each algorithm while preserving comparability.

The rationale for comparing exactly these four algorithms is to span the principal trade-offs encountered in tabular classification rather than to enumerate every available method. Logistic Regression contributes transparency and a strong, low-variance baseline; SVM tests whether margin maximization with a non-linear kernel offers advantages on a high-dimensional feature space; Random Forest probes the benefit of variance reduction through bagging and random feature selection; and XGBoost evaluates the bias-reduction and regularization properties of gradient boosting. Comparing one linear model, one kernel model, and two ensemble paradigms on identical data isolates the effect of the learning principle itself, which is the question of scientific interest, and avoids conflating algorithmic differences with differences in preprocessing or evaluation.

A baseline model is first trained for each algorithm with default or lightly specified parameters to establish a reference level of performance. Logistic Regression uses a fixed random seed and a sufficiently high iteration limit; SVM uses an RBF kernel with the regularization parameter C set to one and an automatically scaled kernel coefficient, with probability estimates enabled; Random Forest uses default parameters with a fixed seed; and XGBoost uses a multi-class objective with a label-encoded target. The baseline serves as a benchmark against which the optimized models are subsequently compared.

3.7 Evaluation metrics

A unified evaluation function computes a fixed set of metrics for every model, which is the prerequisite for a fair comparison. Accuracy measures the overall proportion of correctly classified records; precision and recall describe, respectively, the correctness of positive predictions and the ability to recover true positives; and the F1-score is their harmonic mean. Because the dataset is imbalanced, two additional metrics are essential: the macro F1-score, which averages the per-class F1 values without weighting by frequency, and balanced accuracy, which averages per-class recall. The metrics are defined as follows.

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN}$$

$$\text{Precision} = \frac{TP}{TP + FP}$$

$$\text{Recall} = \frac{TP}{TP + FN}$$

$$F1 = 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}}$$

$$\text{Macro-F1} = \frac{1}{C} \sum_{c=1}^{C} F1_c$$

$$\text{Balanced Accuracy} = \frac{1}{C} \sum_{c=1}^{C} \text{Recall}_c$$

where TP, TN, FP, and FN denote true positives, true negatives, false positives, and false negatives, and C is the number of classes. Reliance on accuracy alone would overstate model quality on imbalanced data; macro F1 and balanced accuracy give equal weight to minority classes and therefore better reflect the practical objective of detecting at-risk students. For the same reason, per-class recall—in particular the recall of the Fail and Withdrawn classes—is reported separately, alongside per-class precision, recall, and F1 in classification reports and confusion matrices, both raw and normalized.

3.8 Cross-validation and hyperparameter optimization

To assess the stability and generalization of the models, stratified k-fold cross-validation was applied uniformly. Stratification preserves the class proportions within each fold, which is important under imbalance, and increases the reliability

of the resulting estimates; for each model the mean cross-validated accuracy and its dispersion were recorded. Hyperparameter optimization was then performed using randomized search over the parameter space [23], which samples promising combinations rather than exhaustively enumerating them, thereby reducing computation while retaining quality. For logistic regression and SVM, the search was embedded in a pipeline that combined scaling with the estimator; random forest, requiring no scaling, was tuned directly; and XGBoost was tuned from a predefined base configuration. In every case the best parameter combination was selected, recorded, and used to retrain an optimized model, whose metrics, classification report, and confusion matrices were recomputed on the test set. This procedure ensures that each model is evaluated not only with default settings but also in a configuration favorable to it, increasing the scientific validity of the comparison.

3.9 Reproducibility

All experiments were implemented in Python 3.11 using scikit-learn 1.5.1, XGBoost 2.1.1, pandas 2.2.2, and NumPy 1.26.4. The train/test split (80/20, stratified) used a fixed random seed of 42; additional seeds used were none. Within hyperparameter optimization, stratified 5-fold cross-validation was used, and all preprocessing steps (scaler, one-hot encoder) were fitted only on the training portion of each fold and applied to the validation portion, preventing information leakage across folds. Hyperparameters were selected by randomized search with 50 iterations per model, optimizing macro F1-score. The search spaces and the selected configurations are reported in Tables 7 and 8.

Table 7. Hyperparameter search spaces by model

Model	Hyperparameter	Search Space
Logistic Regression	C	[0.001, 0.01, 0.1, 1, 10, 100]
Logistic Regression	penalty	[l1, l2]
Logistic Regression	solver	[liblinear, saga]
SVM (RBF)	C	[0.1, 1, 10, 100]
SVM (RBF)	gamma	[scale, auto, 0.001, 0.01, 0.1]
Random Forest	n_estimators	[100, 200, 300, 500]
Random Forest	max_depth	[10, 20, 30, None]
Random Forest	min_samples_split	[2, 5, 10]
Random Forest	min_samples_leaf	[1, 2, 4]
Random Forest	max_features	[sqrt, log2]
XGBoost	n_estimators	[100, 200, 300, 500]
XGBoost	max_depth	[3, 5, 7, 9]
XGBoost	learning_rate	[0.01, 0.05, 0.1, 0.2]
XGBoost	subsample	[0.7, 0.8, 0.9, 1.0]
XGBoost	colsample_bytree	[0.7, 0.8, 0.9, 1.0]
XGBoost	reg_alpha	[0, 0.01, 0.1, 1]
XGBoost	reg_lambda	[0.1, 1, 10]

Table 8. Selected (best) hyperparameters by model

Model	Selected Hyperparameters
Logistic Regression	C = 1.0, penalty = l2, solver = liblinear
SVM (RBF)	C = 10, gamma = scale
Random Forest	n_estimators = 300, max_depth = 20, min_samples_split = 2, min_samples_leaf = 1, max_features = sqrt
XGBoost	n_estimators = 300, max_depth = 5, learning_rate = 0.1, subsample = 0.8, colsample_bytree = 0.8, reg_alpha = 0.01, reg_lambda = 1.0

4 RESULTS

4.1 Class distribution and exploratory observations

The distribution of the target variable, shown in Figure 2, confirms the imbalance noted during data preparation. The Pass class is the largest (37.9% of students), followed by Withdrawn (31.2%), Fail (21.6%), and Distinction (9.3%). The combination of a large withdrawal class and a comparatively small distinction class is characteristic of distance-learning cohorts and has direct consequences for modeling: classifiers can attain high aggregate accuracy by performing well on the majority classes while neglecting the minority Fail and Distinction classes. This observation frames the interpretation of all subsequent results and motivates the imbalance-sensitive metrics introduced in Section 3.7.

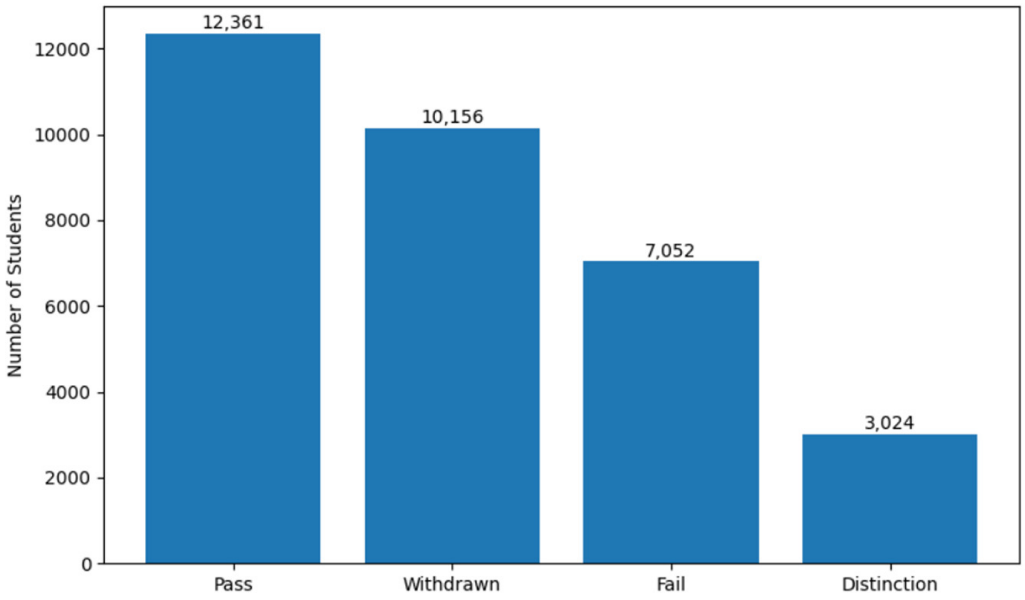

Fig. 2. Distribution of the four target classes across the full dataset, illustrating the imbalance that conditions model behavior

4.2 Baseline and optimized models

Each algorithm was first evaluated as a baseline and then as an optimized model after randomized hyperparameter search. Across all four algorithms, optimization produced modest rather than dramatic improvements in aggregate metrics, indicating that the baseline configurations were already close to the performance ceiling attainable on this feature set and that the principal limits on accuracy arise from the

data—in particular the class imbalance and the partial overlap of the Fail class with neighboring classes—rather than from suboptimal parameters. The optimized models were retained for the comparative analysis, and their stability was confirmed by stratified cross-validation, whose mean accuracies closely tracked the single-split test accuracies, indicating that the reported results are not artifacts of a particular partition.

4.3 Comparative evaluation

The optimized models were compared within a unified evaluation framework using accuracy, precision, recall, F1-score, macro F1, balanced accuracy, per-class recall for the Fail and Withdrawn classes, and mean cross-validated accuracy. The complete results are presented in Table 9.

Table 9. Comparative evaluation of the optimized models on the test set. The best value in each metric is highlighted and serves as the basis for the analysis presented in the text

Metric	XGBoost	Random Forest	Logistic Regression	SVM (RBF)
Acc.	0.7302	0.7200	0.7181	0.6967
Prec.	0.7158	0.7027	0.7024	0.6814
Rec.	0.7302	0.7200	0.7181	0.6967
F1	0.7150	0.6982	0.6974	0.6849
Macro F1	0.6658	0.6393	0.6374	0.6266
Bal. Acc.	0.6555	0.6281	0.6244	0.6184
Rec. Fail	0.3926	0.3430	0.3614	0.4026
Rec. Wdn	0.8385	0.8360	0.8370	0.7917
CV Mean	0.7314	0.7274	0.7174	0.7070

Notes: Abbreviations: Acc. = accuracy; Prec. = weighted precision; Rec. = weighted recall; F1 = weighted F1-score; Macro F1 = macro-averaged F1-score; Bal. Acc. = balanced accuracy; Rec. Fail = recall for the Fail class; Rec. Wdn = recall for the Withdrawn class; CV Mean = mean stratified cross-validated accuracy.

Several conclusions follow from Table 9. XGBoost achieved the highest overall performance, with accuracy 0.7302, F1-score 0.7150, macro F1 0.6658, balanced accuracy 0.6555, and mean cross-validated accuracy 0.7314—the strongest values across the board. This indicates that XGBoost not only predicts accurately in aggregate but also behaves more evenly across classes and generalizes more reliably, since its cross-validated accuracy is not dependent on a single test partition. Random forest ranked second on aggregate metrics (accuracy 0.7200, F1-score 0.6982), but its lower macro F1 and balanced accuracy show that it predicts the dominant classes well while performing less uniformly across all classes. Logistic regression produced results close to random forest (accuracy 0.7181, F1-score 0.6974), a notable outcome given its simplicity, confirming its value as an efficient and interpretable baseline, although it lagged behind XGBoost on the rarer classes. SVM recorded the weakest aggregate metrics (accuracy 0.6967, F1-score 0.6849, macro F1 0.6266, and balanced accuracy 0.6184); yet it attained the highest recall for the Fail class (0.4026), a point of practical significance discussed below.

The cross-validation results corroborate the single-split ranking and speak to the robustness of the conclusions. The mean cross-validated accuracies—0.7314 for XGBoost, 0.7274 for random forest, 0.7174 for logistic regression and 0.7070 for SVM—preserve the same ordering observed on the held-out test set and differ from

the corresponding test accuracies by no more than approximately one percentage point. This concordance indicates that the relative performance of the models is not an artifact of a fortunate partition and that XGBoost's advantage in generalization is genuine rather than incidental. It also confirms that the small margins separating the top three models are stable so that the practical choice among them can legitimately be informed by secondary considerations such as interpretability or per-class behavior rather than by aggregate accuracy alone.

Fig. 3. Optimized model performance across the four primary metrics

Note: XGBoost leads on every aggregate metric, while random forest and logistic regression are closely matched and SVM trails.

The ranking by aggregate metrics is therefore consistent: XGBoost, random forest, logistic regression, and SVM. Figure 3 visualizes this ordering and shows that the gap between XGBoost and the next two models is small, whereas SVM is clearly separated below them. The heat map in Figure 4 summarizes all nine metrics simultaneously and reinforces two patterns: XGBoost dominates the majority of metrics, and SVM, despite low aggregate scores, stands out on Fail recall. The proximity of the random forest and logistic regression rows confirms the similarity of their behavior.

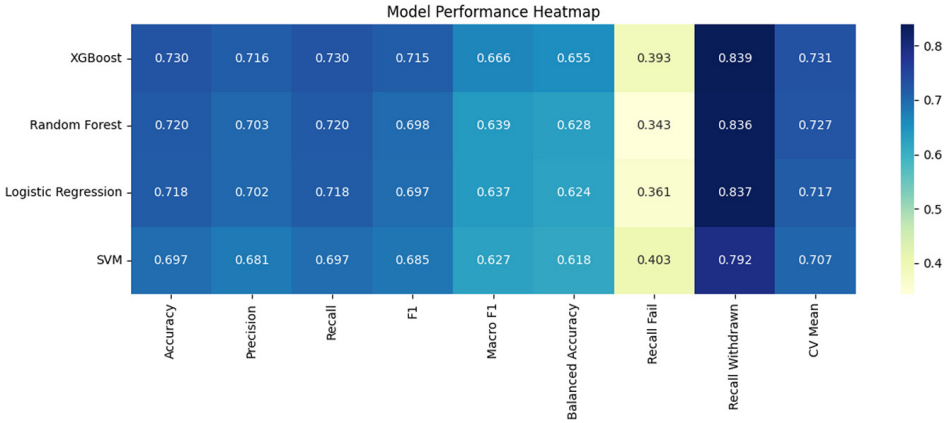

Fig. 4. Heat map of all nine-evaluation metrics by model

Notes: Darker cells denote higher values; XGBoost is strongest across most metrics, while the Fail-recall column reveals the cross-model weakness on the at-risk class.

4.4 Per-class analysis and the fail-class gap

Because the Fail class identifies students at academic risk, its recall is the most consequential per-class metric. Across all four models, however, Fail recall is low—between 0.3430 and 0.4026—whereas Withdrawn recall is consistently high, between 0.7917 and 0.8385. Figure 5 contrasts the two. Three factors explain this asymmetry. First, the imbalance of the dataset biases models towards the more frequent classes, since optimizing aggregate accuracy rewards correct prediction of Pass and Withdrawn. Second, the Fail class is structurally complex: some failing students resemble withdrawing students in their low activity, while others resemble marginal passes, so the Fail class partially overlaps with its neighbors in feature space and is intrinsically harder to separate. Third, the Withdrawn class has a distinctive signature—markedly reduced learning activity—that is comparatively easy to detect, which accounts for its uniformly high recall.

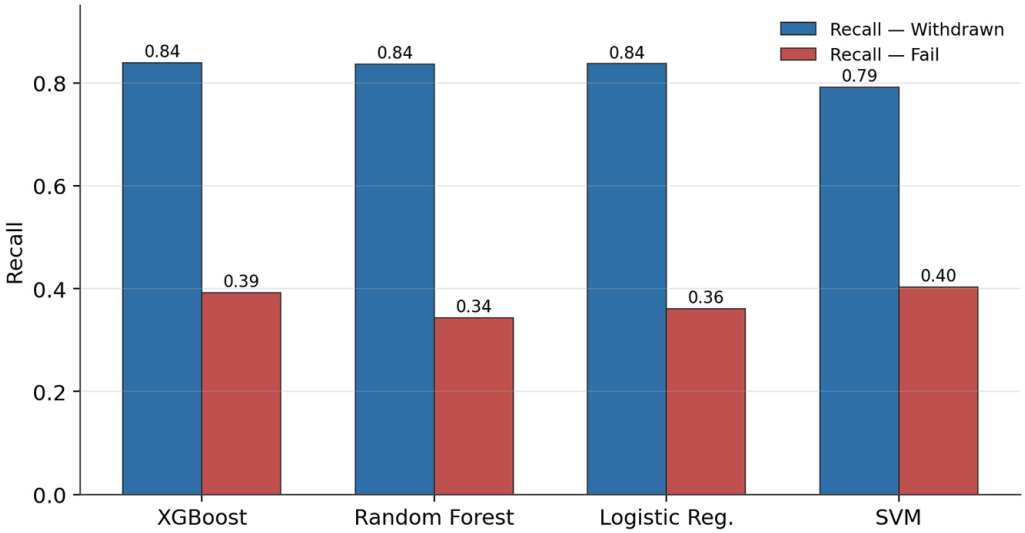

Fig. 5. Per-class recall for the Withdrawn and Fail classes

Notes: Withdrawn is detected reliably by all models, whereas Fail recall remains low, with SVM providing the best detection of failing students.

This pattern qualifies the headline ranking. While XGBoost is the best overall model, SVM is the most sensitive detector of failing students, and an institution whose primary objective is to maximize the recovery of at-risk students might weight this property heavily. The result also demonstrates concretely why aggregate accuracy is an insufficient criterion for this problem: a model can rank first on accuracy yet detect fewer than two in five failing students.

To support the analysis above, the per-class precision, recall, and F1 of all four optimized models across the four outcome classes are reported in Table 10, and the corresponding confusion matrices follow. All analyses in this and the following subsections were produced by the same reproducible pipeline as the main comparison: an identical 80/20 stratified split (random seed 42), identical preprocessing (scaling and one-hot encoding fitted on the training portion only), and the identical feature-construction code were used throughout, and no alternative feature sets were introduced. The headline figures in Table 9 correspond to the hyperparameter-optimized models, whereas the per-window and class-weighting experiments (refer to Tables 15 and 16) use the common baseline configuration so

that the single manipulated factor—observation-window length or class weighting—is isolated. The resulting differences from the Table 9 aggregates are therefore minor (within approximately one percentage point; for example, XGBoost accuracy is 0.7302 for the optimized model and 0.7208 at the baseline 120-day setting) and stem solely from this difference in model configuration, not from any difference in data, split, features, or preprocessing.

Table 10 presents the class-specific precision, recall, and F1 scores of the four optimized models, providing a more detailed view of predictive performance across the outcome categories.

Table 10. Per-class precision, recall, and F1 for all four outcome classes (optimized models)

Model	Class	Precision	Recall	F1
XGBoost	Distinction	0.64	0.48	0.55
XGBoost	Pass	0.75	0.89	0.81
XGBoost	Fail	0.58	0.40	0.47
XGBoost	Withdrawn	0.76	0.81	0.78
Random Forest	Distinction	0.68	0.41	0.52
Random Forest	Pass	0.73	0.92	0.82
Random Forest	Fail	0.59	0.35	0.44
Random Forest	Withdrawn	0.75	0.82	0.78
Logistic Regression	Distinction	0.66	0.42	0.51
Logistic Regression	Pass	0.73	0.87	0.80
Logistic Regression	Fail	0.55	0.37	0.44
Logistic Regression	Withdrawn	0.75	0.82	0.78
SVM (RBF)	Distinction	0.52	0.44	0.48
SVM (RBF)	Pass	0.75	0.82	0.78
SVM (RBF)	Fail	0.50	0.41	0.45
SVM (RBF)	Withdrawn	0.73	0.77	0.75

The confusion matrices for the four optimized models are presented in Tables 11–14. These matrices provide additional insight into the types of classification errors made by each model and reveal how the outcome classes are confused with one another.

Table 11. Confusion matrix—XGBoost

True\Pred	Distinction	Pass	Fail	Withdrawn
Distinction	289	315	1	0
Pass	156	2188	107	21
Fail	3	330	567	511
Withdrawn	3	68	305	1655

Note: Rows are true classes, columns predicted.

Table 12. Confusion matrix—random forest

True\Pred	Distinction	Pass	Fail	Withdrawn
Distinction	251	354	0	0
Pass	111	2271	69	21
Fail	2	385	490	534
Withdrawn	4	85	278	1664

Note: Rows are true classes, columns predicted.

Table 13. Confusion matrix—logistic regression

True\Pred	Distinction	Pass	Fail	Withdrawn
Distinction	254	339	7	5
Pass	125	2155	152	40
Fail	4	360	526	521
Withdrawn	2	95	271	1663

Note: Rows are true classes, columns predicted.

Table 14. Confusion matrix—SVM (RBF)

True\Pred	Distinction	Pass	Fail	Withdrawn
Distinction	269	325	3	8
Pass	223	2038	176	35
Fail	13	302	575	521
Withdrawn	8	62	400	1561

Note: Rows are true classes, columns predicted.

4.5 Operational prototype

To verify the practical applicability of the models, a web prototype was developed using the Streamlit framework. Access is restricted through a teacher authentication mechanism in which passwords are stored and compared as SHA-256 hashes rather than as plain text. User-entered information is transformed by the same preprocessing logic used in training—deriving activity features, constructing a feature row aligned to the training schema and applying one-hot encoding—so that the model receives input of the correct structure. The selected model and, where applicable, its scaler and label encoder are loaded, a prediction is generated, and the per-class probability distribution is returned and visualized. The prototype is presented strictly as a proof-of-concept interface demonstrating how a trained model could be operationalized and not as a validated or production-ready system [24]. No evaluation with teachers, students or administrators—such as a usability test, expert walkthrough, or scenario-based assessment—was conducted in this study, and its usability, pedagogical usefulness, and institutional feasibility therefore remain to be established. Accordingly, the prototype should not be interpreted as evidence that the system is ready for real-world educational decision-making.

4.6 Imbalance-aware modeling

To investigate whether the limited recovery of failing students could be improved, the best-performing model, XGBoost, was re-trained using class-weighted learning. The weighting scheme assigned greater importance to minority classes, particularly the Fail class, during optimization. Performance was then compared with the original unweighted model using the same train/test split and evaluation protocol. The results are summarized in Table 15.

Table 15. Effect of imbalance handling on Fail-class recall and overall metrics

Configuration	Accuracy	Macro F1	Balanced Acc.	Fail Recall	Fail Precision
Baseline (no rebalancing)	0.7208	0.6552	0.6449	0.4018	0.58
+ Balanced class weights (XGBoost)	0.6934	0.6633	0.6928	0.5287	0.54

Applying balanced class weighting to XGBoost substantially improved the recovery of failing students: Fail-class recall rose from 0.4018 to 0.5287 (+12.7 percentage points) and balanced accuracy from 0.6449 to 0.6928, while macro F1 also increased slightly. These gains came at the cost of lower overall accuracy (0.7208 to 0.6934) and a small reduction in Fail precision (0.58 to 0.54), reflecting a larger number of false-positive risk alerts. This trade-off is characteristic of imbalance-aware learning: the model becomes more sensitive to at-risk students but less conservative in assigning the Fail label. For institutions whose primary objective is early intervention, balanced accuracy and Fail recall are the more appropriate selection criteria, and under these criteria, the class-weighted configuration is preferable despite its modest reduction in aggregate accuracy.

4.7 Sensitivity to the early-prediction window

To assess how predictive performance depends on the length of the observation window, all four models were re-evaluated on early windows of 30, 60, 90, and 120 days (refer to Table 16). All windows were evaluated with the common baseline configuration on the same stratified split and pipeline, so the 120-day row corresponds to the baseline values in Table 15 rather than to the hyperparameter-optimized figures of Section 4.3 (see the reproducibility note in Section 4.4); the remaining cells are obtained from the corresponding per-window re-runs.

Table 16. Model performance under early windows of 30, 60, 90, and 120 days

Window (Days)	Model	Accuracy	Macro F1	Balanced Acc.	Fail Recall
30	XGBoost	0.7243	0.6597	0.6482	0.4089
30	Random Forest	0.7150	0.6330	0.6196	0.3494
30	Logistic Regression	0.7032	0.6281	0.6158	0.3671
30	SVM (RBF)	0.6828	0.6176	0.6113	0.4132
60	XGBoost	0.7225	0.6573	0.6471	0.4103
60	Random Forest	0.7144	0.6344	0.6217	0.3494

(Continued)

Table 16. Model performance under early windows of 30, 60, 90, and 120 days *(Continued)*

Window (Days)	Model	Accuracy	Macro F1	Balanced Acc.	Fail Recall
60	Logistic Regression	0.7044	0.6293	0.6171	0.3721
60	SVM (RBF)	0.6823	0.6160	0.6099	0.4139
90	XGBoost	0.7210	0.6561	0.6460	0.3997
90	Random Forest	0.7154	0.6358	0.6232	0.3466
90	Logistic Regression	0.7062	0.6330	0.6205	0.3806
90	SVM (RBF)	0.6792	0.6133	0.6079	0.4089
120	XGBoost	0.7208	0.6552	0.6449	0.4018
120	Random Forest	0.7173	0.6377	0.6250	0.3473
120	Logistic Regression	0.7053	0.6333	0.6208	0.3728
120	SVM (RBF)	0.6815	0.6162	0.6113	0.4075

Across the windows examined, overall predictive performance changed very little: accuracy remained close to 0.72 for XGBoost, and the model ordering was stable at every window length. Fail-class recall likewise stayed in a narrow band (approximately 0.40 for XGBoost and SVM) rather than rising substantially with more behavioral data. This near-constant behavior indicates that, in the present feature set, the demographic and intermediate-assessment features carry most of the predictive signal, while the volume of early VLE activity contributes comparatively little; the Fail-class gap therefore persists undiminished even at the shortest 30-day window. SVM retained the highest Fail recall at every window, and XGBoost the strongest aggregate performance, consistent with the main results. A practical implication is that useful early predictions are already available from the first 30 days, but that lengthening the observation window alone does not close the Fail-class gap: improving the detection of failing students requires imbalance-aware modeling (Section 4.6) or richer, window-specific assessment features rather than simply more days of clickstream data.

5 DISCUSSION

The comparative evaluation supports three principal findings. First, XGBoost is the most effective model for predicting student academic performance on the OULAD, leading in accuracy, F1-score, macro F1, balanced accuracy, and cross-validated accuracy. Its advantage reflects the strengths of regularized gradient boosting on structured, heterogeneous tabular data, where iterative error correction and complexity control allow it to capture non-linear interactions among demographic, assessment, and behavioral features without overfitting. Second, random forest and logistic regression form a closely matched second tier; the competitiveness of Logistic Regression, despite its linear form, echoes the broader finding in the literature that simple models often rival complex ones on educational data and underlines the continued value of interpretable baselines. Third, SVM, although weakest in aggregate, achieves the best recall on the Fail class, which means that model selection cannot be reduced to a single aggregate score and must instead be guided by the institutional objective.

The most important substantive result is the persistent gap between the high recall of the Withdrawn class and the low recall of the Fail class. From a learning-analytics perspective, this is the crux of the problem, because failing students are precisely those whom an early-warning system most needs to identify. The gap arises from the interaction of class imbalance with the structural ambiguity of the Fail class, whose members occupy a region of feature space that overlaps with both withdrawing and marginally passing students. The implication is methodological as well as practical: aggregate accuracy is a misleading optimization target for this task, and evaluation must foreground per-class recall and balanced metrics. The same reasoning suggests that improvements are more likely to come from addressing imbalance and enriching the representation of the Fail class than from further hyperparameter tuning, which yielded only marginal gains.

These results should be read in light of several limitations. The handling of missing data, although principled, cannot fully preserve the natural distribution of the variables and may introduce some distortion. The class imbalance, while mitigated by stratification and addressed through imbalance-sensitive metrics, was not corrected at the data level in the present experiments, and the low Fail recall is a direct consequence. The models were trained solely on the OULAD, which covers specific modules and cohorts, so their transferability to other institutions or later periods remains to be established. The feature matrix is static and does not represent the temporal dynamics of learning across a presentation; activity that evolves over time is summarized rather than modeled sequentially, which limits the use of methods designed for temporal dependence. Finally, the predictive quality of the deployed prototype depends on the accuracy and completeness of the data entered by users, and the relevance of any trained model will decay as curricula, assessment policies, and student behavior change, implying a need for periodic retraining.

Each limitation points to a concrete avenue for improvement. The class imbalance can be addressed at the data level using synthetic minority oversampling or controlled under sampling, which prior work shows can substantially improve minority-class recall; in such settings, the F1-score and the area under the ROC curve are more informative than accuracy. Feature engineering can be deepened by partitioning behavioral activity into early, middle, and late phases and by adding indicators such as submission timing and the earliest assessment outcomes, which bear most directly on the Fail-class gap identified above. Sequential models, including recurrent neural networks [25], temporal convolutional networks [26], and temporal attention-based hybrid deep learning models [27], could capture learning dynamics where sufficient sequential data are available. In parallel, interpretability methods such as SHAP [28] and LIME [29] could be used to explain individual predictions and improve transparency. It is important, however, to distinguish prediction from explanation and intervention: the models developed here estimate the probability of an outcome, but they neither establish why a particular student fails nor identify which support measure would change that outcome. Closing the gap between an accurate early warning and an effective response therefore, requires causal and pedagogical evidence beyond the predictive modeling reported in this study.

Beyond predictive quality, responsible deployment raises considerations that bear directly on how such systems should be used. Because the models are trained on demographic as well as behavioral features, predictions could encode or amplify disparities associated with age, region, prior education, or socio-economic background; institutions should therefore audit predictions for differential error rates across protected groups and treat outputs as decision support rather than as deterministic judgements. These concerns echo wider questions about the validity, reliability, and

fairness of AI-supported assessment in higher education, where automated judgements must be benchmarked carefully against human evaluation and examined for differential effects before they are allowed to inform consequential decisions [30]. The relatively low recall on the Fail class is itself an equity concern, since a system that misses the majority of failing students cannot equitably allocate support. Transparency is equally important: the integration of interpretability methods, the clear communication of uncertainty through the per-class probabilities already produced by the prototype, and the framing of predictions as early warnings to be verified by human advisers are all necessary safeguards against over-reliance. These considerations reinforce the study's central methodological message and should accompany any operational use of the models.

Looking further ahead, the practical value of performance-prediction systems will grow as they are integrated into the digital infrastructure of teaching. Expanding the range of data sources to include richer behavioral and motivational signals, moving from retrospective scoring towards real-time monitoring that issues automatic alerts, integrating predictions with learning-management systems and electronic gradebooks, and using predictions to support personalized learning trajectories would transform such systems from evaluation tools into intelligent mechanisms for managing the educational process [31].

6 CONCLUSION

This study developed and comparatively evaluated four machine learning models—logistic regression, SVM, random forest, and XGBoost—for the multiclass prediction of student academic performance on the OULAD. A reproducible, leakage-controlled feature-engineering pipeline was proposed in which final examination records were excluded and behavioral activity was restricted to the first 120 days of each presentation, yielding 85 demographic, assessment, and behavioral features for 32,593 students. All models were trained on an identical stratified split and assessed within a unified, imbalance-sensitive evaluation framework.

XGBoost delivered the strongest overall performance, with accuracy 0.7302, macro F1 0.6658, balanced accuracy 0.6555, and cross-validated accuracy 0.7314, and is therefore the best-performing model under the unweighted experimental setup. It should not, however, be regarded as a universally preferred model: the choice depends on the institutional objective. Where overall accuracy matters most, the unweighted XGBoost is the natural choice, whereas where the primary goal is early intervention, the class-weighted XGBoost variant (Section 4.6) is preferable since it raises Fail-class recall from 0.40 to 0.53 and balanced accuracy from 0.64 to 0.69 at the cost of a modest reduction in aggregate accuracy. SVM, though weakest in aggregate, offers a further high-recall alternative for the Fail class, and Random Forest and Logistic Regression form a closely matched and efficient second tier. The analysis showed that aggregate accuracy substantially overstates the practical usefulness of these models, because the recall of the academically critical Fail class remained low across all of them, and it identified this gap—rooted in class imbalance and the structural ambiguity of the Fail class—as the principal target for future work. A Streamlit prototype demonstrated that the trained models can be operationalized within an early-warning workflow.

The contribution of the work lies in combining a leakage-aware early-window pipeline with an imbalance-sensitive, algorithm-aware comparison on a single benchmark, thereby clarifying both which model performs best and why aggregate

metrics can mislead. Future research should address the Fail-class gap through data-level rebalancing, richer temporal feature engineering, ensemble and sequential models, and integrated interpretability and should examine transferability across institutions and cohorts. Such developments would strengthen the role of machine learning as a basis for the early identification of at-risk students and for improving the quality of teaching and learning in higher education.

7 FUNDING

This study has been/was/is funded by the Committee of Science of the Ministry of Science and Higher Education of the Republic of Kazakhstan (Grant №BR28713531 Intelligent digital system of higher and postgraduate education organizations Smart.EDU).

8 CONFLICTS OF INTEREST

The authors declare no conflicts of interest.

9 INFORMED CONSENT STATEMENT

This study relied exclusively on the OULAD, a publicly available dataset released in 2017 under a CC-BY 4.0 license and fully anonymized by its providers prior to publication using k-anonymity (k = 5). No personally identifiable information was accessed, and no new data were collected from human participants by the authors. Because the work constitutes secondary analysis of an openly licensed, anonymized dataset and did not involve the recruitment of participants or the administration of surveys, additional informed consent and institutional ethics approval were not required.

10 REFERENCES

[1] C. Romero and S. Ventura, "Educational data mining and learning analytics: An updated survey," *WIREs Data Mining and Knowledge Discovery*, vol. 10, no. 3, p. e1355, 2020. https://doi.org/10.1002/widm.1355

[2] A. Namoun and A. Alshanqiti, "Predicting student performance using data mining and learning analytics techniques: A systematic literature review," *Applied Sciences*, vol. 11, no. 1, p. 237, 2021. https://doi.org/10.3390/app11010237

[3] I. H. Sarker, "Machine learning: Algorithms, real-world applications and research directions," *SN Computer Science*, vol. 2, no. 3, p. 160, 2021. https://doi.org/10.1007/s42979-021-00592-x

[4] A. Serek and D. Polat, "Forecasting student academic performance using machine learning," *Journal of Emerging Technologies and Computing*, vol. 2, no. 2, 2025. https://doi.org/10.47344/w85rct27

[5] A. Herayono, M. Anwar, E. Tasrif, and Q. N. M. Batubara, "Intelligent mobile system for student performance evaluation: Model testing using structural equation modeling," *International Journal of Interactive Mobile Technologies (iJIM)*, vol. 20, no. 4, pp. 23–36, 2026. https://doi.org/10.3991/ijim.v20i04.60101

[6] N. A. Dahri, W. M. Al-Rahmi, K. A. Alhashmi, and F. Bashir, "Enhancing mobile learning with AI-powered Chatbots: Investigating ChatGPT's impact on student engagement and academic performance," *International Journal of Interactive Mobile Technologies (iJIM)*, vol. 19, no. 11, pp. 17–38, 2025. https://doi.org/10.3991/ijim.v19i11.54643

[7] S. B. Kotsiantis, "Use of machine learning techniques for educational purposes: A decision support system for forecasting students' grades," *Artificial Intelligence Review,* vol. 37, no. 4, pp. 331–344, 2012. https://doi.org/10.1007/s10462-011-9234-x

[8] H. Waheed, S. U. Hassan, N. R. Aljohani, J. Hardman, S. Alelyani, and R. Nawaz, "Predicting academic performance of students from VLE big data using deep learning models," *Computers in Human Behavior*, vol. 104, p. 106189, 2020. https://doi.org/10.1016/j.chb.2019.106189

[9] M. Hussain, W. Zhu, W. Zhang, S. M. R. Abidi, and S. Ali, "Student engagement predictions in an e-learning system and their impact on student course assessment scores," *Computational Intelligence and Neuroscience,* vol. 2018, 2018. https://doi.org/10.1155/2018/6347186

[10] H. He and E. A. Garcia, "Learning from imbalanced data," *IEEE Transactions on Knowledge and Data Engineering*, vol. 21, no. 9, pp. 1263–1284, 2009. https://doi.org/10.1109/TKDE.2008.239

[11] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, "SMOTE: Synthetic minority over-sampling technique," *Journal of Artificial Intelligence Research*, vol. 16, pp. 321–357, 2002. https://doi.org/10.1613/jair.953

[12] N. Tomasevic, N. Gvozdenovic, and S. Vranes, "An overview and comparison of supervised data mining techniques for student exam performance prediction," *Computers & Education*, vol. 143, p. 103676, 2020. https://doi.org/10.1016/j.compedu.2019.103676

[13] J. Kuzilek, M. Hlosta, and Z. Zdrahal, "Open university learning analytics dataset," *Scientific Data*, vol. 4, p. 170171, 2017. https://doi.org/10.1038/sdata.2017.171

[14] M. Adnan *et al.*, "Predicting at-risk students at different percentages of course length for early intervention using machine learning models," *IEEE Access*, vol. 9, pp. 7519–7539, 2021. https://doi.org/10.1109/ACCESS.2021.3049446

[15] W. Torkhani and K. Rezgui, "OULAD MOOC student performance prediction using machine and deep learning techniques," in *Proceedings of the International Conference on Data Analytics and Insights (ICODAI 2024)*. Atlantis Press, 2025. https://doi.org/10.2991/978-94-6463-654-3_18

[16] L. Shala Riza, L. Abazi Bexheti, and J. Zoroja, "Early identification of at-risk students in online education: A deep learning approach to predictive modelling. Business systems research," *International Journal of the Society for Advancing Innovation and Research in Economy*, vol. 16, no. 2, pp. 69–91, 2025. https://doi.org/10.2478/bsrj-2025-0019

[17] F. Pedregosa *et al.*, "Scikit-learn: Machine learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.

[18] J. VanderPlas. *Python Data Science Handbook: Essential Tools for Working with Data.* O'Reilly Media, 2016.

[19] D. R. Cox, "The regression analysis of binary sequences," *Journal of the Royal Statistical Society: Series B (Methodological)*, vol. 21, no. 2, pp. 238–238, 1959. https://doi.org/10.1111/j.2517-6161.1959.tb00334.x

[20] C. Cortes and V. Vapnik, "Support-vector networks," *Machine Learning*, vol. 20, no. 3, pp. 273–297, 1995. https://doi.org/10.1007/BF00994018

[21] L. Breiman, "Random forests," *Machine Learning*, vol. 45, no. 1, pp. 5–32, 2001. https://doi.org/10.1023/A:1010933404324

[22] T. Chen and C. Guestrin, "XGBoost: A scalable tree boosting system," in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '16)*, 2016, pp. 785–794. https://doi.org/10.1145/2939672.2939785

[23] J. Bergstra and Y. Bengio, "Random search for hyper-parameter optimization," *Journal of Machine Learning Research*, vol. 13, no. 2, pp. 281–305, 2012.

[24] M. Hlosta, Z. Zdrahal, and J. Zendulka, "Ouroboros: Early identification of at-risk students without models based on legacy data," in *Proceedings of the Seventh International Learning Analytics & Knowledge Conference (LAK '17)*, 2017, pp. 6–15. https://doi.org/10.1145/3027385.3027449

[25] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997. https://doi.org/10.1162/neco.1997.9.8.1735

[26] S. Bai, J. Z. Kolter, and V. Koltun, "An empirical evaluation of generic convolutional and recurrent networks for sequence modeling," *arXiv preprint arXiv:1803.01271*, 2018. https://doi.org/10.48550/arXiv.1803.01271

[27] A. Omarbekova *et al.,* "A temporal attention-based hybrid deep learning model for student performance and academic risk prediction," *Frontiers in Artificial Intelligence*, vol. 9, 2026. https://doi.org/10.3389/frai.2026.1811886

[28] S. M. Lundberg and S.-I. Lee, "A unified approach to interpreting model predictions," in *Advances in Neural Information Processing Systems*, I. Guyon et al., Eds., vol. 30, 2017.

[29] M. T. Ribeiro, S. Singh, and C. Guestrin, "'Why should I trust you?': Explaining the predictions of any classifier," in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '16)*, 2016, pp. 1135–1144. https://doi.org/10.1145/2939672.2939778

[30] G. Zacharis and S. Papadakis, "Can AI grade like a human? Validity, reliability, and fairness in university coursework assessment," *Educational Process: International Journal*, vol. 19, 2025. https://doi.org/10.22521/edupij.2025.19.591

[31] B. Sujin, M. S. Mani, and K. Kumar, "Implementation of a personalised learning system for at-risk students using adaptive techniques," *International Journal of Interactive Mobile Technologies (iJIM)*, vol. 20, no. 3, pp. 121–133, 2026. https://doi.org/10.3991/ijim.v20i03.60059

11 AUTHORS

Assel Omarbekova is an Associate Professor at the Institute of Digital Sciences and Artificial Intelligence, L.N. Gumilyov Eurasian National University, Kazakhstan. Her research interests include information security, digital technologies, cybersecurity, and information systems. She is the author of more than 250 scientific publications, including papers indexed in Scopus and Web of Science (E-mail: omarbekova_as@enu.kz).

Aizhan Nazyrova is a PhD researcher at the Institute of Digital Sciences and Artificial Intelligence, L.N. Gumilyov Eurasian National University, Kazakhstan. Her research interests include informatics, artificial intelligence, software reliability, data analysis, optimization methods, and intelligent systems. She has authored several scientific publications in these areas (E-mail: nazyrova_aye_1@enu.kz).

Gulmira Bekmanova is a PhD and an Associate Professor at the Institute of Digital Sciences and Artificial Intelligence, L.N. Gumilyov Eurasian National University, Kazakhstan. Her research interests include information systems, digital technologies, digital transformation, and higher education management. She has authored more than 120 scientific publications and has held several academic and administrative positions at the university, including Vice-Rector for Digitalization (E-mail: bekmanova_gt@enu.kz).

Zhanar Lamasheva received her PhD in Information Systems from Satbayev University, Almaty, Kazakhstan, in 2015. She is currently a Lecturer at the Institute

of Digital Sciences and Artificial Intelligence, L.N. Gumilyov Eurasian National University, Astana, Kazakhstan. Her research interests include simulation modeling, data analysis, machine learning, and natural language processing (NLP). She has contributed to research in intelligent systems and data-driven technologies (E-mail: lamasheva_zhb@enu.kz).

Rakhila Turebayeva is a researcher at the Institute of Digital Sciences and Artificial Intelligence, L.N. Gumilyov Eurasian National University, Kazakhstan. Her research interests include machine learning, educational data mining, learning analytics, artificial intelligence, predictive modeling, and data-driven approaches to improving student academic performance. She has contributed to research on the application of intelligent systems and analytics in education (E-mail: turebayeva_rd@enu.kz).

Zhainagul Abzal completed her Bachelor's degree at the Institute of Digital Sciences and Artificial Intelligence, L.N. Gumilyov Eurasian National University, Kazakhstan, in 2026. This publication is based on a collaborative research project conducted under the supervision of the second and fourth authors. Her research interests include educational data mining, learning analytics, machine learning, and the application of artificial intelligence in education (E-mail: abzal_zha@enu.kz).