Ingénierie des Systèmes d'Information
Vol. 31, No. 4, April, 2026, pp. 1217-1230
Journal homepage: http://iieta.org/journals/isi

Hybrid Logistic Regression Stacking Framework for Accurate and Interpretable Student Outcome Prediction

Wijiyanto[1,2*], Aris Marjuni[1], Ahmad Zainul Fanani[1], Ruri Suko Basuki[1]

[1] Department of Informatics Engineering, Dian Nuswantoro University, Semarang 50131, Indonesia
[2] Faculty of Computer Science, Universitas Duta Bangsa Surakarta, Surakarta 57154, Indonesia

Corresponding Author Email: wijiyanto@udb.ac.id

https://doi.org/10.18280/isi.310417

Received: 12 January 2026
Revised: 3 March 2026
Accepted: 15 March 2026
Available online: 30 April 2026

Keywords:
educational data mining, student outcome prediction, hybrid stacking, logistic regression, SMOTE, binary and multiclass classification, decision-support systems

ABSTRACT

This study presents Hybrid Logistic Regression Stacking (HLR-STACK), a hybrid stacking framework with logistic regression (LR) as the meta-learner, for predicting student outcomes in higher education. The framework integrates leakage-safe preprocessing, in-fold Synthetic Minority Over-sampling Technique (SMOTE) for class balancing, and out-of-fold stacking over tree-based and boosting base learners. Two tasks are considered: binary classification (Graduate vs. Dropout) and multiclass classification (Dropout, Enrolled, Graduate). The Top-K ensemble selection ensures a compact and deployment-friendly model. Evaluation on a 4,424-record dataset demonstrates that HLR-STACK achieves 0.981 accuracy for the binary task and 0.844 accuracy for the multiclass task, outperforming single-model baselines. Post-hoc calibration with Maximum Calibration Error (MCE) and reliability diagrams supports interpretable probability-based risk scoring. Recursive Feature Elimination (RFE) identifies the top 30 predictors, highlighting semester pass rates, study load, grades, and course structure as primary contributors. The results confirm that a simple, transparent meta-learner effectively integrates complementary base learners while providing interpretable outputs. HLR-STACK operationalizes a practical, auditable decision-support workflow for early identification of at-risk students, facilitating targeted interventions and evidence-based educational policies. This study contributes a scalable, interpretable framework that can be extended to other higher education institutions and datasets, offering insights into improving student retention and academic performance prediction.

1. INTRODUCTION

Student success and dropout rates in higher education affect social mobility, workforce readiness, and the effectiveness of the education system. Recent studies report persistently high dropout rates across various programs and socioeconomic groups. Key drivers are typically academic, demographic, and institutional. From an information-systems engineering perspective, an institutional early-warning system (EWS) is more than a predictive model; it is an end-to-end data-processing and decision-support pipeline that must remain leakage-safe, interpretable to stakeholders, and auditable at the probability level (e.g., through risk bands) to support consistent decisions across cohorts [1]. Contemporary Educational Data Mining (EDM) focuses on fine-grained feature engineering, dealing with class imbalance and exploiting strong ensemble modeling strategies to facilitate early detection and improve predictive accuracy over simple baselines. Precise modeling is also emphasized by institutional studies [2]. Institutional-based work highlights openness in modeling and the transfer of insights between degrees [3].

Viewed as an information system artifact, Hybrid Logistic Regression Stacking (HLR-STACK) operationalizes a scalable workflow that can be embedded into institutional analytics: ingestion of administrative/academic attributes, leakage-safe preprocessing and in-fold imbalance handling, out-of-fold (OOF) stacking to produce robust probabilistic outputs, and probability reliability auditing (Maximum Calibration Error (MCE) and reliability diagrams) to support risk scoring and triage. This framing aligns the study with information systems engineering goals, repeatable data pipelines, scalable scoring, and governance-ready audit trails, rather than treating prediction as an isolated modeling exercise.

Tree-based ensembles are a practical workhorse for structured higher-education data [4]. Multi-metric reporting facilitates guarding against excessive use of a specific metric (in terms of accuracy) under imbalanced settings [5]. The results suggest that Random Forest and boosting methods can be stable for predicting grades and engagement [6]. This aligns with the characteristics of educational data, which typically mixes categorical and numerical variables and exhibits non-linear relationships and occasional missingness [7]. Notably, recent advances draw attention to meta-learning as a means of merging synergistic inductive biases while also preventing overfitting [8]. A pipeline that is leakage-safe, when combined with stacking at the output, gives more reliable probabilities for operation [9]. Post-hoc calibration has been reported to

improve the consistency of risk scores and enhance their generalizability [10].

Robustness also relies on feature engineering/selection, incorporating academic history and study load, efficiency metrics, as well [11]. This is further supported by results on interaction characteristics and quantile discretization (q-binning), which can model non-linear threshold effects, such as high but incomplete study loads [12]. Following these recommendations, we tune hyperparameters and select model configurations on the training set using stratified 7-fold cross-validation, and report final performance on a held-out 20% test to ensure portability and reproducibility [13].

Although considerable progress has been made, the task of predicting student performance is still a hard one. Imbalance in class may decrease the recall for the minority classes; therefore, it is necessary to rebalance to be able to make valid conclusions [14]. Furthermore, drawing attention to the overall accuracy can hide error asymmetries [15]. This is why analyzing per-class and confusion matrix errors is important for diagnosing errors and staging interventions. In operational deployments, model outputs serve as risk scores that affect institutional policy; therefore, transparent audit trails are necessary for accountability, review, and governance [16]. The stacking can improve generalization by combining complementary base learners through OOF meta-features, while mitigating overfitting. When paired with a pipeline that does not leak and whose in-fold rebalancing scheme is independent of the learners' decision functions, supervised stacking can end up with a strong meta-learner (e.g., Logistic Regression (LR)) capable of summarizing predictive evidence. Hybrid/meta-ensemble models are also investigated to enhance the generalization ability, and consistent results for stacked learners from previous work on university performance prediction under strict validation settings have been proven [17].

Recent works confirm the need for feature encoding/selection to alleviate redundancy and stabilize learning, thereby improving interpretability. Metaheuristic methods enhance the performance further if used in a combination of ensembles and careful validations [18]. Careful pre-processing is required for high-quality signals [19]. These methods show promise, but their use is limited by methodological gaps. Digital Learning Systems (LMS) have explored combining deep features with machine learning to extract interaction patterns. This improves accuracy when integrated with other methods [20]. Studies highlight contextual differences and the need for portable models [21]. Recent review papers position calibrated risk-aware decision support as a new standard for publication [22]. There is a need for comprehensive prediction frameworks that support both binary and multiclass outcomes within a unified institutional workflow. Such frameworks should enforce leakage-safe and imbalance handling during evaluation, while generating interpretable probability outputs suitable for downstream risk scoring.

Due to these gaps, this study poses three main research questions. RQ1: How well does the HLR-STACK framework predict graduation and dropout under a leakage-safe evaluation protocol? RQ2: How well can HLR-STACK distinguish between three study-status categories in a more challenging, imbalanced multiclass task? RQ3: What factors do the logistic-regression learners identify as the main contributors to graduation and dropout probabilities, and how can these findings be interpreted through Recursive Feature Elimination (RFE)-based factor analysis?

The contributions of this study are summarized as follows. First, we propose HLR-STACK, a hybrid ensemble framework designed to predict student performance across two tasks, binary and multiclass, using a large-scale higher education dataset. Second, the framework integrates leakage-safe preprocessing, quantile-based feature engineering, and SMOTE-based class rebalancing, together with a logistic regression meta-learner and an optional post-hoc calibration step that is audited using MCE and reliability diagrams. Third, we perform a systematic factor analysis of the top thirty features selected by RFE to examine how predictive performance relates to pass rates, study load, and family context. Fourth, the method is verified by comparing it with baseline algorithms as well as previous studies that build on the same or closely related datasets. Apart from model performance, the paper presents a decision-support framework that prioritizes prediction accuracy and early detection within higher education policy and institutional contexts. The framework also becomes a handy template for machine-learning pipelines and data shapes. In this way, HLR-STACK operationalizes a decision-support workflow that translates validated predictive performance into actionable, auditable risk outputs for higher-education decision-making.

2. RELATED WORK

Recent research on predicting student performance focuses on improving prediction accuracy using various algorithms. Studies on binary or multiclass classification tasks have shown that decision trees and Random Forest perform well for pass/fail prediction and course-level outcomes, often without considering feature engineering or probability calibration [23]. Other studies use Random Forest, combined with handling class imbalance, for multiclass trajectories, showing that label distribution and temporal structure affect model choice [24]. Longitudinal analysis of administrative data over several years reveals that linear and tree-based ensembles can achieve robust accuracy, although the behavior of minority classes is often under-examined [25]. Other contributions use regression or binary classification problems with neural networks or voting ensembles, which outperform single models but rarely examine the impact of class imbalance or error asymmetry [26, 27]. This study proposes a hybrid model that integrates class-imbalance handling and a Top-K stacking scheme, and it further audits probability reliability using MCE and reliability diagrams (with calibration treated as an optional post-processing step).

In this study [28], a hybrid approach combining data sources and applying imputation to improve data quality before classification is proposed. However, the feature-selection procedures were not fully described, which may have introduced bias. Other contributions include data cleaning and chi-square-based screening before classification, but without embedding these processes into cross-validated pipelines [29]. Pipeline-oriented approaches that combine data export, attribute selection, and training of multiple classifiers similarly fail to clearly specify how missing values, normalization, class balancing, and feature-selection integration are handled, which can introduce leakage [30]. In comparison, our study employs a strictly leakage-safe protocol, including imputation, encoding, and scaling; resampling is restricted to the training folds.

The third line of work focuses on evaluation design and often reports accuracy and precision. However, these studies sometimes rely on repeated hold-out splits and do not consider the reliability of probability or the ROC-AUC [31]. Other frameworks use cross-validation and hyperparameter optimization to evaluate multiple algorithms, increasing transparency but omitting calibration analysis [32]. Some studies enrich the evaluation with additional regression-style metrics, but remain limited to small cohorts or individual modules [33]. Robust schemes use k-fold cross-validation with an external test set and a set of performance indicators, though probability reliability and confusion-matrix inspection are not always discussed [34]. In contrast, our work reports accuracy, precision, recall, F1-score, and ROC-AUC for both binary and multiclass tasks, alongside confusion matrices and sigmoid or isotonic calibration.

The fourth study group examines feature selection and class balancing strategies. Hybrid models reduce redundancy and improve accuracy, but are usually evaluated with a single train-test split without stability analysis [35]. One study on a large labeled dataset incorporates data cleaning, feature selection, and class balancing before training models, yielding high accuracy. However, feature selection details and balancing techniques were not fully documented [36]. Furthermore, other works apply the Synthetic Minority Over-sampling Technique (SMOTE) algorithm to increase minority classes to better Random Forest capability; however, do not analyze robustness or stability of selected sets [37]. This analysis uses RFE for factorization and in-fold SMOTE for handling class imbalance and interpretability of the components.

Compared with representative studies on the same dataset (Table 1), our contribution is less about proposing yet another classifier and more about tightening the end-to-end, deployable workflow: leakage-safe preprocessing with in-fold resampling, OOF stacking to avoid meta-level contamination, and probability reliability auditing (MCE and reliability diagrams) for risk-oriented use. While prior works largely emphasize model comparisons or hybrid fusion under standard splits, our design frames student-outcome prediction as a governed decision-support pipeline with reproducible evaluation and auditable probability outputs, rather than relying only on point accuracy.

Table 1. Design-level comparison on the same dataset

Study	Leakage-Safe Evaluation	OOF Stacking Safety	Probability Audit (MCE + Diagrams)	Interpretability
[24]	Partial / not explicit	No	No	Partial
[38]	Not specified	No	No	Partial
[39]	Partial / not explicit	Partial	No	No / limited
This work	Yes	Yes	Yes	Yes

Note: OOF = out-of-fold; MCE = Maximum Calibration Error

In brief, these four families of studies confirm that although lots of progress has been achieved in the design of student performance prediction models, there is a lack of frameworks that (i) explicitly handle both binary and multiclass tasks; (ii) incorporate genuinely leakage-safe preprocessing and class-balancing protocols; (iii) explicitly audit probability reliability (e.g., via calibration evidence), rather than assuming calibration always helps; and (iv) linking model's performance with interpretable factor analysis. The remainder of this section describes the design of HLR-STACK, which has been designed to overcome these shortcomings.

3. METHODOLOGY

3.1 Dataset description

The 4,424-record dataset from a public repository includes 34 predictor features and one target feature. The predictors in the HLR-STACK experiments fall into four categories: six demographic predictors, three macroeconomic predictors, seventeen academic predictors, and eight socioeconomic predictors. A complete description of the dataset is provided in this research [40]. The target feature is an imbalanced categorical variable. Classes include Graduate (2,209), Dropout (1,421), and Enrolled (794). The dataset has no missing values in the selected attributes used in this study. This study predicts student performance under two output scenarios. In the multiclass scenario, all three target labels are used on the full set of 4,424 records. In the binary scenario, we use a subset of 3,630 records that includes only the "Graduate" and "Dropout" classes. The "Enrolled" class is excluded to simulate end-of-study risk detection.

3.2 Preprocessing

All preprocessing steps are executed within a leakage-safe pipeline. Each transformation is fitted on the training data within each fold and then applied to the validation data. Principal Component Analysis (PCA) is executed outside the main training pipeline and used solely to visualize the class structure before and after applying SMOTE, without influencing model learning.

3.2.1 Data cleaning

This process harmonizes feature types and performs basic data quality checks (e.g., type consistency and constant feature removal). Although the selected attributes contain no missing values, an imputation component is retained inside the cross-validation pipeline as a leakage-safe safeguard for robustness and extensibility. Finally, it eliminates constant or dominated features to reduce redundancy.

3.2.2 Feature engineering

The feature engineering stage uses derived variables to measure performance and risk. We examine indicators such as total enrolled and approved credits in the first two semesters. These are supplemented with level and dynamics indicators, as well as efficiency measures. We also encode additional features, such as age and study scheduling. We combine these with financial risk and macroeconomic indicators to create a risk-aware learning system with an interpretable logistic regression learner.

3.2.3 Encoding and scaling

All categorical variables, including those created via quantile binning, are transformed using One-Hot Encoding, while numerical variables are standardized with a StandardScaler. These transformations are implemented through a unified pipeline with a ColumnTransformer that performs encoding and scaling before class balancing. Each transformation is only fitted to the training portion of each fold. Then, it is applied to the corresponding validation fold to prevent information leakage.

3.2.4 Recursive Feature Elimination

RFE is a factor analysis tool that identifies the most important attributes in prediction models. The base estimator is LR with ℓ_2 regularization; RFE is executed outside the HLR-STACK pipeline. In each training fold, the LR model is fitted on standardized data. Its coefficients are used to remove unimportant features, leaving only 30 features. Since only training data is used, the resulting rankings do not incorporate validation data. RFE does not constrain the features used; it is reported as a complementary factor analysis step to rank variables and summarize dominant predictive factors. This ranked subset is used for interpretability and to motivate future work on model simplification. Section 4.6 presents detailed interpretations of the factor analysis features.

3.2.5 Handling class imbalance

Class imbalance, particularly the under-representation of the Dropout class, is addressed using SMOTE applied only to the training portion of each cross-validation fold. Synthetic minority samples are generated in the cleaned and standardized feature space, with SMOTE using the default neighborhood size (k_neighbors = 5) to avoid over-synthesizing in sparse minority regions and to keep the rebalancing behavior reproducible. The evaluation protocol prioritizes minority-class Recall and F1-score, complemented by ROC-AUC under a one-vs-rest (OVR) formulation, so that improvements in dropout detection are not obscured by majority-class performance. The empirical impact of this rebalancing strategy on class separation and error patterns is analyzed in the Results section.

3.2.6 Leakage control

All preprocessing and resampling operations adhere to a strict 'fit-on-train-only' rule within the stratified 7-fold cross-validation scheme. In each fold, imputation, encoding, scaling, and SMOTE are fitted exclusively on the training subset, and the learned transformations are then applied to the validation subset. OOF predictions from the base learners are collected and used as input features for the LR meta-learner, ensuring that the meta-learner never accesses target values from outside its own fold and that probability calibration is learned on data that were not used to fit the corresponding base models. To further reduce the risk of temporal leakage, the feature space is restricted to information from the first two semesters, excluding variables that could encode post-outcome conditions. Additional auditing is conducted through sanity checks and inspection of cross-fold consistency to detect any indication of data contamination or protocol violations.

We also consider feature-cross leakage, where derived variables may inadvertently encode target information (e.g., post-outcome administrative flags). To mitigate this, we restrict the feature set to first-year/first-two-semester attributes and exclude variables that may be downstream consequences of the target outcome. Our causal/time-order assumption is that all predictors precede the outcome definition; therefore, any feature that could reflect later academic status is treated as leakage-prone and removed.

3.3 Experimental methodology

The experiments follow the leakage-safe cross-fold pipeline illustrated in Figure 1. We consider two supervised learning tasks: (i) a multiclass classification task that distinguishes Dropout, Enrolled, and Graduate, and (ii) a binary classification task that separates Graduate from Dropout. For both tasks, stratified 7-fold cross-validation serves as the main evaluation framework, ensuring that class proportions are preserved across folds and that RQ1 and RQ2 are addressed under comparable conditions.

The division of the dataset is 80% training and 20% for testing. The test data is excluded from model training and used once for final evaluation. The 80% training part is further divided into 7 folds stratified cross-validation for model selection & hyperparameter optimization. For each fold, all preprocessing and SMOTE from (Section 3.2) are fit on the training subset only, while the validation subset is kept unseen until transformation and prediction. Once the best configuration has been determined, the chosen model is then retrained on all training data and tested against a test set. The model's generalization performance is accurately reflected in the reported metrics.

Hyperparameters are selected on the 80% training portion using stratified 7-fold cross-validation. We rank candidate configurations using a balanced metric set (Accuracy, Precision, Recall, F1-score, and ROC-AUC), with additional emphasis on minority-class Recall/F1 to avoid majority-class dominance under class imbalance. When two configurations show comparable Accuracy, we prioritize the one with stronger minority-class Recall/F1 and more stable cross-fold behavior (i.e., fewer extreme fold outliers), then retrain the chosen setting on the full training split before evaluating once on the held-out 20% test set.

We tuned hyperparameters using a compact, literature-informed search on the training folds. Specifically, we varied the most influential capacity and regularization controls for each learner (e.g., tree depth, number of estimators, learning rate, subsampling, and L2 regularization for boosting models; and number of trees, minimum leaf size, and feature subsampling for Random Forest). Candidate settings were evaluated by mean cross-validation performance under the leakage-safe pipeline, and the final setting per learner (reported in Tables 2-3) was selected based on the best cross-fold score and stability (i.e., avoiding configurations with highly volatile fold behavior).

Within each fold, we train a set of candidate base learners (XGBoost, LightGBM, Gradient Boosting, CatBoost, Random Forest, and AdaBoost) and collect their OOF predicted probabilities. These OOF predictions form the meta-features for the LR meta-learner. We optionally apply post-hoc calibration on the validation fold (sigmoid for binary; isotonic for multiclass) and evaluate its impact using MCE and reliability diagrams before deciding whether calibrated probabilities are used for downstream risk analysis. Finally, we construct Top-K stacks with K ranging from 2 to 6 by selecting the best-performing base learners. The final configuration is chosen using a balanced metric set (Accuracy, Precision, Recall, F1-score, and ROC-AUC), with particular

attention to minority-class Recall/F1 under class imbalance.

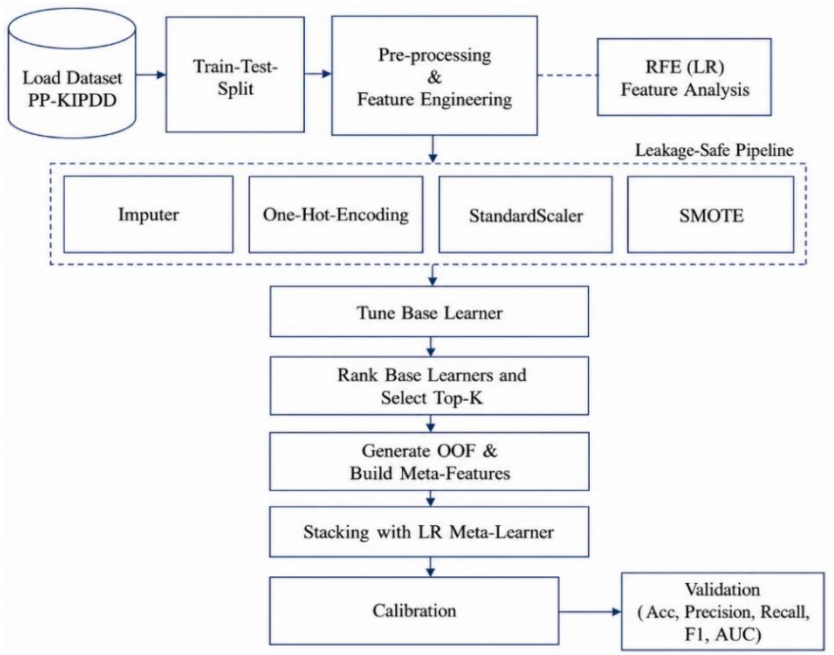


Figure 1. Experimental architecture

3.4 Proposed Hybrid Logistic Regression Stacking model

HLR-STACK is a stacked generalization scheme with LR as a meta-learner on top of a compact set of robust tree-based and boosting learners. The training data is $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^{n}$, where $y_i \in \{0,1\}$ for the binary task and $y_i \in \{1, \dots, C\}$ for the multiclass task. The architecture has two levels. At Level-0, a set of base learners $\{h_m\}_{m=1}^{M}$ is trained within a leakage-safe pipeline. At Level-1, an LR model is trained on stacking features constructed from OOF predicted probabilities produced by the base learners.

The leakage-safe pipeline (Figure 2) applies the transformation T_θ to the training subset, including median imputation for numerical and mode imputation for categorical variables, one-hot coding for categorical (and quantile-binned) variables, and standardization for numerical variables. All components of T_θ are fitted on the training portion of the fold. LR-based RFE identifies the top 30 most contributive features without modifying the main HLR-STACK pipeline.

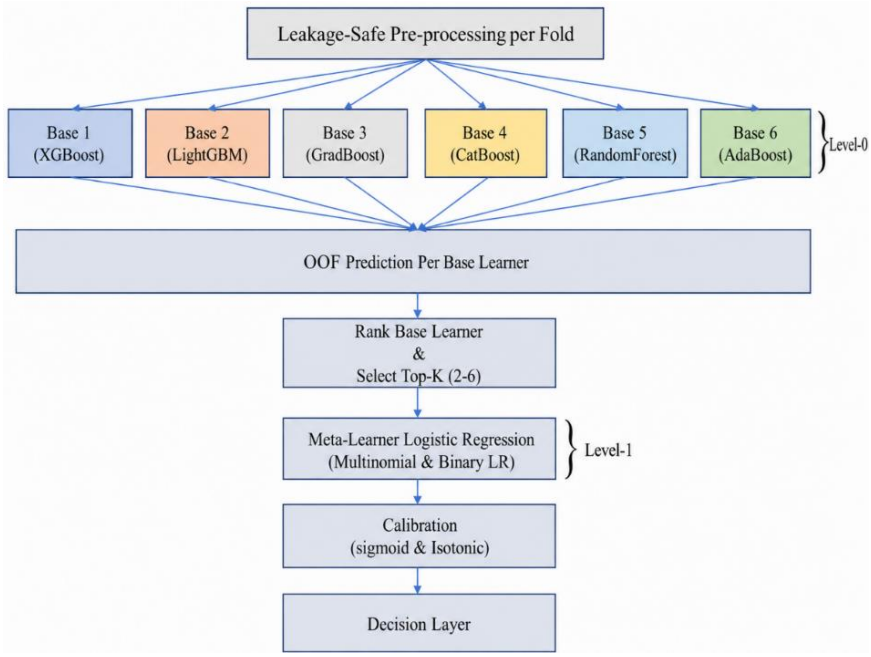


Figure 2. Architecture HLR-STACK model
Note: HLR-STACK = Hybrid Logistic Regression Stacking.

Class imbalance is handled by SMOTE, which is part of the same leakage-safe pipeline. Synthetic minority samples are generated in the cleaned and standardized feature space according to

$$\tilde{x} = x_i + \lambda \left(x_i^{(NN)} - x_i\right), \lambda \sim U(0,1) \quad (1)$$

where, $x_i^{(\mathrm{NN})}$ denotes a nearest neighbor of the minority instance x_i. Because SMOTE is wrapped together with T_θ and refitted on the training fold at each iteration, no information from the validation or test portions can leak into the resampling process. Each base learner h_m is then trained on the resampled training data in every fold. For the binary task, each base learner outputs either a logit or a positive-class probability $p_m(x) \in [0,1]$. For the multiclass task, it outputs a probability vector $p_m(x) = [p_{m1}(x), \ldots, p_{mC}(x)]\top$. For instance i, the stacking feature vector is defined as $z_i = [p_1(x_i), \ldots, p_K(x_i)]\top$ in the binary case (dimension K), or $z_i = vec([p_1(x_i), \ldots, p_k(x_i)])$ in the multiclass case (dimension $K \times C$), where K denotes the number of selected base learners in the stack. All vectors z_i are constructed in an OOF manner: the prediction for x_i is always produced by models that have never been trained on its label y_i. This mechanism ensures that the LR meta-learner captures only cross-model patterns that can genuinely generalize to new data.

At Level-1, LR models the meta-level probability. For the binary task, the meta-learner is defined as

$$\hat{p}\left(y=1 \mid z\right) = \sigma\left(w^\top z + b\right), \ \sigma(t) = \frac{1}{1+e^{-t}} \quad (2)$$

with parameters (w, b) obtained by minimizing the regularized log-loss with an ℓ_2 penalty. For the multiclass task, LR with a softmax output is used,

$$\hat{p}\left(y=c \mid z\right) = softmax(W^\top z + b)_c \quad (3)$$

with parameters

$$(\mathrm{W}, \mathrm{b})^{opt} = \arg\min_{\mathrm{W,b}} \left\{ -\sum_{\mathrm{i}} \log \hat{p}(y_i \mid z_i) + \lambda \parallel W \parallel_2^2 \right\} \quad (4)$$

LR is the meta-learner, so evidence across models remains stable and can be interpreted through its coefficients.

Finally, the meta-learner's output probabilities are adjusted in the meta-feature space. For the binary task, we use sigmoid calibration $\tilde{p} = \sigma(a \cdot \mathrm{logit}(p) + b)$, with (a, b) learned from validation data. For the multiclass task, we apply OVR isotonic regression. Each component p_c is mapped to $\tilde{p}_c = f_c(p_c)$ using a non-decreasing function f_c estimated from validation data. This two-level design, which involves OOF stacking followed by calibration, enables HLR-STACK to provide risk scores that should be verified while maintaining and utilizing the performance of the best Top-K base learners.

3.5 Algorithm

Algorithm 1: HLR-STACK model algorithm

Input: dataset $D = \{(\mathrm{x}_i, y_i)\}_{i=1}^N$; task $T \in$ {binary, multiclass}
Output: final model $\mathcal{M}$; selected Top-K
Set the seed and split $\mathcal{D}$ into training and test sets
Use stratified cross-validation for tuning, OOF stacking, and calibration
Define leakage-safe preprocess $\mathcal{P}$: imputer, OHE, scale (train only)
Define bases $\mathcal{B}$ = {XGB, LGBM, GB, Cat, RF, Ada}
For each base $b \in \mathcal{B}$: build pipeline $P_b = \mathrm{P} \rightarrow$ SMOTE (train in fold only) $\rightarrow b$
Evaluate each pipeline P_b with CV on the training set and rank all bases
Define the set of Top-K candidates as {2,3,4,5,6}
For each candidate, K in {2,3,4,5,6}
Select the Top-K pipelines $\{P_b\}$ with the highest scores
Generate out-of-fold (OOF) predictions and construct the meta feature
Train the LR meta-learner
Optionally calibrate the LR meta-learner on validation meta features (sigmoid/isotonic) and retain calibration only if it reduces MCE on validation evidence.
Refit the selected Top-K stack and apply the selected probability output (raw or calibrated) to the test set.
Evaluate the resulting HLR-STACK
Select the value of K that achieves the best overall performance
Return $\mathcal{M}$

3.6 Evaluation

Model performance is measured in this study by several metrics for the binary and multiclass classification tasks: Accuracy, Precision, Recall, F1-score, and ROC-AUC. Accuracy measures the proportion of correctly classified instances over all observed cases:

$$\mathrm{ACC} = \frac{(\mathrm{TP}+\mathrm{TN})}{(\mathrm{TP}+\mathrm{TN}+\mathrm{FP}+\mathrm{FN})} \quad (5)$$

Precision is the fraction of instances predicted as positive that are actually positive:

$$\mathrm{Precision} = \frac{\mathrm{TP}}{(\mathrm{TP}+\mathrm{FP})} \quad (6)$$

Recall is used to measure the model's ability to detect all true positive cases:

$$\mathrm{Recall} = \frac{\mathrm{TP}}{(\mathrm{TP}+\mathrm{FN})} \quad (7)$$

F1-score aggregates Precision and Recall into a single balanced measure:

$$\mathrm{F1} = \frac{2\,(\mathrm{Precision} \times \mathrm{Recall})}{(\mathrm{Precision}+\mathrm{Recall})} \quad (8)$$

Class separability is assessed for ROC-AUC via the Receiver Operating Characteristic (ROC) curve, where the true positive rate (TPR) and false positive rate (FPR) are defined as follows:

$$\mathrm{TPR} = \frac{\mathrm{TP}}{(\mathrm{TP}+\mathrm{FN})}, \ \mathrm{FPR} = \frac{\mathrm{FP}}{(\mathrm{FP}+\mathrm{TN})} \quad (9)$$

Because the multiclass task is imbalanced, we additionally report Balanced Accuracy and Matthews Correlation Coefficient (MCC) to better reflect minority-class performance beyond overall Accuracy. Balanced Accuracy is computed as the average of per-class recalls, while MCC provides a correlation-based summary that remains

informative under imbalance.

4. RESULTS AND DISCUSSION

All results in this section were computed using the 20% test set held out after model selection and tuning were performed on the training data via stratified 7-fold cross-validation.

4.1 Base learner performance (binary)

In predicting whether a student will graduate or drop out, all six base models performed well, with accuracies between about 0.88 and 0.90 (Table 2). XGBoost had the highest accuracy at 0.895, with LightGBM and Gradient Boosting close behind at 0.890. Random Forest and AdaBoost did not perform quite as well. The small gaps among the top three models suggest that boosting-based methods capture semester progress, workload, and approval ratios better than bagging methods. The top three models also exhibit different error patterns, which a meta-learner can exploit to further improve performance.

Table 2. Base learner performance (binary task)

Model	ACC	Parameter
Ada	0.886	'clf_n_estimators': 300, 'clf_learning_rate': 0.5 'clf_subsample': 1.0, 'clf_reg_lambda': 1.0,
XGB	0.895	'clf_n_estimators': 900, 'clf_min_child_weight': 3, 'clf_max_depth': 6, 'clf_learning_rate': 0.03, 'clf_gamma': 0, 'clf_colsample_bytree': 0.7
LGBM	0.890	'clf_subsample': 0.8, 'clf_reg_lambda': 2.0, 'clf_num_leaves': 63, 'clf_n_estimators': 900, 'clf_learning_rate': 0.03, 'clf_colsample_bytree': 0.7
GBoost	0.890	'clf_n_estimators': 600, 'clf_max_depth': 2, 'clf_learning_rate': 0.03
RF	0.883	'clf_n_estimators': 1200, 'clf_min_samples_split': 2, 'clf_min_samples_leaf': 2, 'clf_max_features': 'sqrt'
Cat	0.889	'clf_rsm': 1.0, 'clf_learning_rate': 0.04, 'clf_l2_leaf_reg': 2.0, 'clf_iterations': 800, 'clf_depth': 4

4.2 Base learner performance (multiclass)

For the multiclass classification task, the same model family provides a consistent foundation (Table 3). XGBoost, CatBoost, and LightGBM yield near-identical accuracy scores (0.764, 0.762, and 0.761, respectively), while Gradient Boosting and Random Forest demonstrate marginally lower results. The negligible difference among the top three models indicates that tree-boosting algorithms effectively discern nonlinear relationships among workload, pass rates, and financial indicators. Still, the moderate accuracy reveals that differentiating all three classes is more complex than binary classification, suggesting opportunities for enhancement through model stacking.

4.3 Performance Hybrid Logistic Regression Stacking model (binary)

To answer RQ1, the proposed HLR-STACK model was compared with single-based models on binary tasks. XGBoost alone produced a test accuracy of 0.895, while the Top-K = 3 stack (XGB + LGBM + Gradient Boosting) improved the test accuracy to 0.981, accompanied by precision, recall, and F1-score of 0.964, 0.972, and 0.968, respectively, as well as ROC-AUC of 0.999 (Table 4). Using more than three base learners (K = 4–6) did not significantly improve performance over K = 3; in some settings (e.g., K = 4 or K = 6), the F1-score actually decreased slightly, suggesting diminishing returns when adding less complementary models. Confusion matrix analysis in Figure 3 shows that stacking Top-K = 3 reduces classification errors from Graduate to Dropout and Dropout to Graduate compared to smaller or larger K values. Therefore, the K = 3 configuration offers the best balance between performance and complexity, keeping the stack simple, interpretable, and easier to audit for downstream use.

Table 3. Base learner performance (multiclass task)

Model	ACC	Parameter
Ada	0.726	'clf_n_estimators': 300, 'clf_learning_rate': 0.5 'clf_subsample': 0.8, 'clf_reg_lambda': 1.0,
XGB	0.764	'clf_n_estimators': 1200, 'clf_min_child_weight': 3, 'clf_max_depth': 6, 'clf_learning_rate': 0.06, 'clf_gamma': 1, 'clf_colsample_bytree': 0.9
LGBM	0.761	'clf_subsample': 0.8, 'clf_reg_lambda': 2.0, 'clf_num_leaves': 63, 'clf_n_estimators': 900, 'clf_learning_rate': 0.03, 'clf_colsample_bytree': 0.7
GBoost	0.755	'clf_n_estimators': 800, 'clf_max_depth': 2, 'clf_learning_rate': 0.03
RF	0.745	'clf_n_estimators': 800, 'clf_min_samples_split': 2, 'clf_min_samples_leaf': 1, 'clf_max_features': 'sqrt'
Cat	0.762	'clf_rsm': 1.0, 'clf_learning_rate': 0.08, 'clf_l2_leaf_reg': 2.0, 'clf_iterations': 800, 'clf_depth': 8

Table 4. Performance of HLR-STACK (binary task)

Top-K	Bases	ACC	Pr	Rc	F1	ROC-AUC
3	XGB+LGBM+GBoost	0.981	0.964	0.972	0.968	0.999
5	XGB+LGBM+GBoost+ Cat+Ada	0.981	0.968	0.968	0.968	0.999
4	XGB+LGBM+GBoost+Cat	0.979	0.963	0.968	0.966	0.999
2	XGB+ LGBM	0.977	0.947	0.977	0.962	0.999
6	XGB+LGBM+Boost +Cat+Ada+RF	0.975	0.946	0.972	0.959	0.999

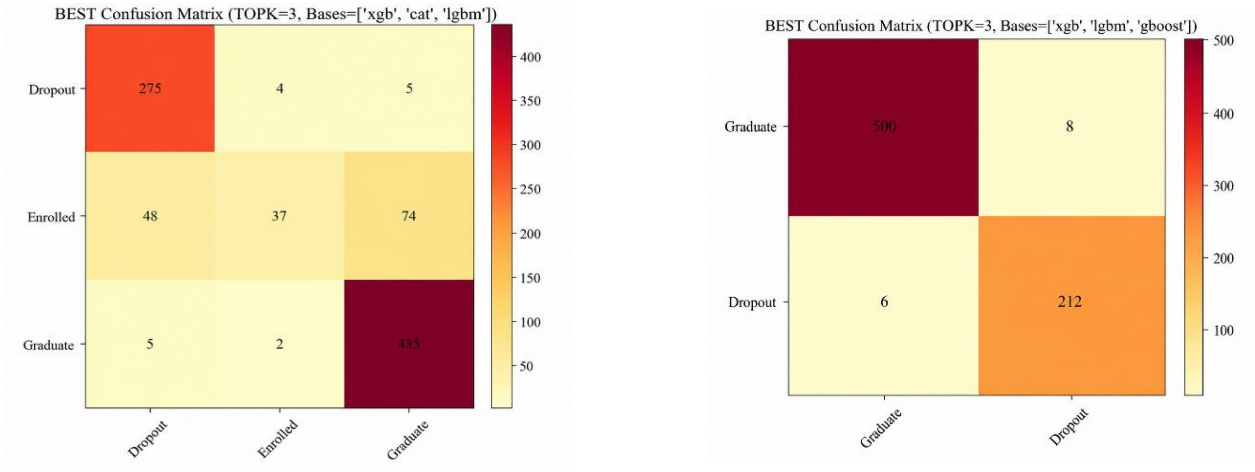
4.4 Performance Hybrid Logistic Regression Stacking model (multiclass)

To answer RQ2, we analyzed how well HLR-STACK separated the three study status categories for multiclass tasks. The Stack Top-K = 3 configuration (XGBoost + CatBoost + LightGBM) achieved an accuracy of 0.844, precision of 0.848, recall of 0.728, F1-score of 0.725, and ROC-AUC of 0.915 on the test data, surpassing the best baseline model, XGBoost, which had an accuracy of 0.764 (Table 5). The K = 4 configuration achieved the same accuracy and only a marginal change in F1-score (0.727 vs. 0.725), indicating that K = 3

already captures most of the complementary signal while keeping the stack simpler. Overall, Top-K = 3 offers a practical performance–complexity trade-off for deployment.

Table 5. Performance of HLR-STACK (multiclass task)

Top-K	Bases	Acc	Pr	Rc	F1	ROCAUC	Bal-Acc	MCC
3	XGB+Cat+ LGBM	0.844	0.848	0.728	0.725	0.915	0.728	0.752
4	XGB+Cat+ LGBM+ GBoost	0.844	0.844	0.729	0.727	0.916	0.729	0.751
6	XGB+Cat+ LGBM+RF+ GBoost+ Ada	0.841	0.833	0.724	0.720	0.916	0.724	0.745
5	XGB+Cat+ LGBM+RF+ GBoost	0.840	0.841	0.720	0.712	0.915	0.720	0.745
2	XGB+Cat	0.838	0.839	0.717	0.709	0.916	0.717	0.743



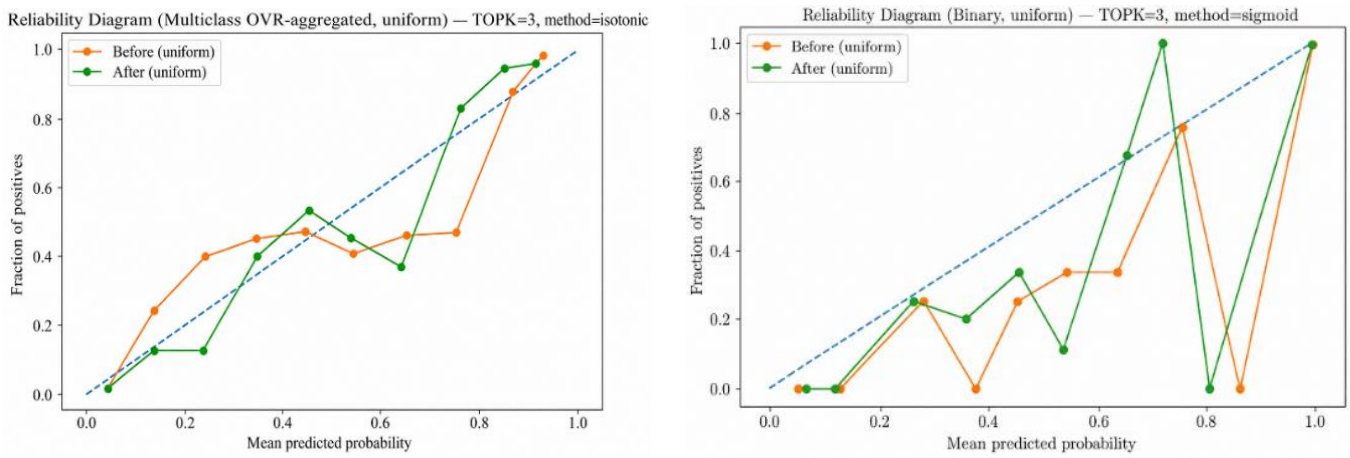
(a) Multiclass (b) Binary

Figure 3. Confusion matrix

Table 6. Calibration evidence on the held-out test (before vs after calibration)

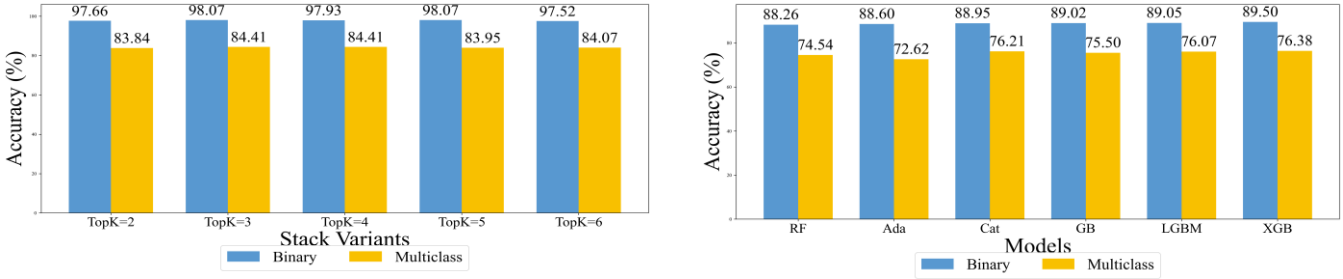
Setting	MCE (Uniform, 10 Bins) Binary	MCE OVR-Macro (Uniform, 10 Bins) Multiclass
Before Calibration	0.862004	0.546830
After Calibration (sigmoid/isotonic)	0.806986	0.498270

When isotonic calibration is applied, it can reduce worst-case bin deviations in probability outputs (as reflected by MCE), which helps make risk profiles more stable in difficult boundary regions such as the Enrolled class. However, we treat calibration as an evidence-driven post-processing step and rely on MCE plus reliability diagrams to judge whether calibrated probabilities are preferable for risk-related use. To complement Accuracy and macro-F1 under multiclass imbalance, we also report Balanced Accuracy and MCC in Table 5; these metrics follow the same overall trend as Acc/F1 across Top-K settings and provide an additional, imbalance-aware view of performance.



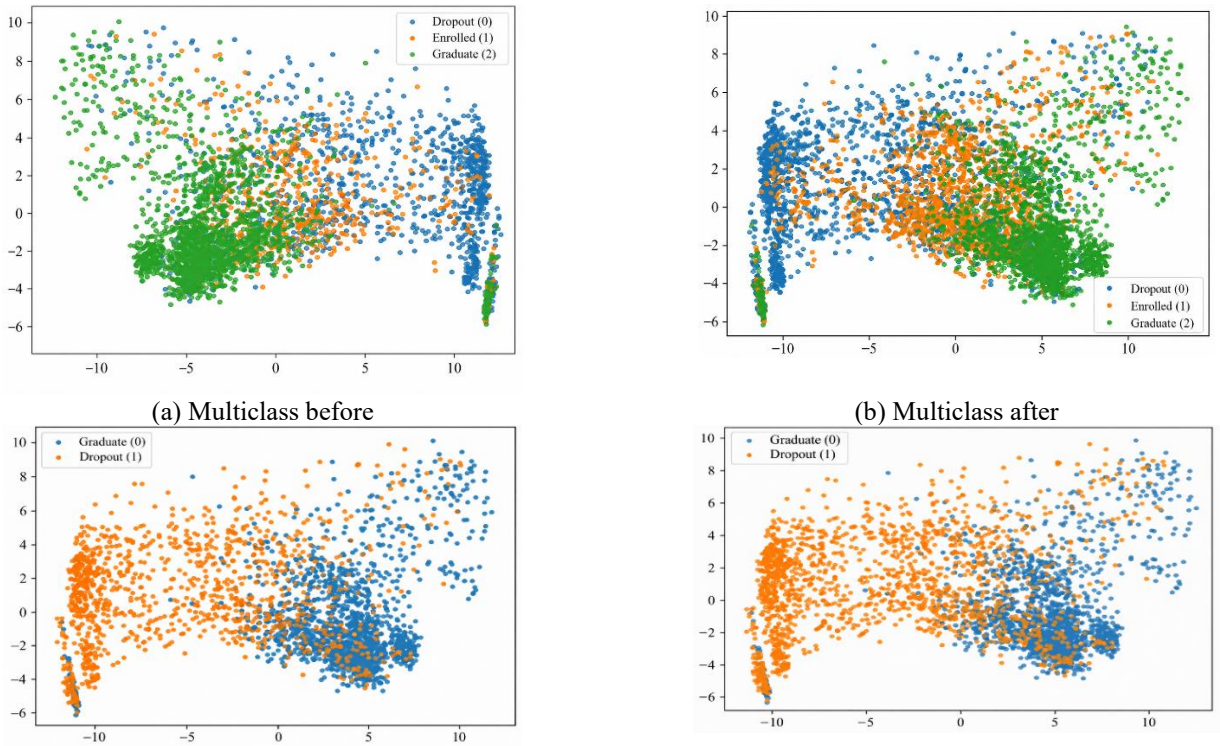
(a) Multiclass (b) Binary

Figure 4. Reliability diagrams comparing uncalibrated vs calibrated probabilities



(a) Stack Variants (b) Base Models

Figure 5. Performance comparison

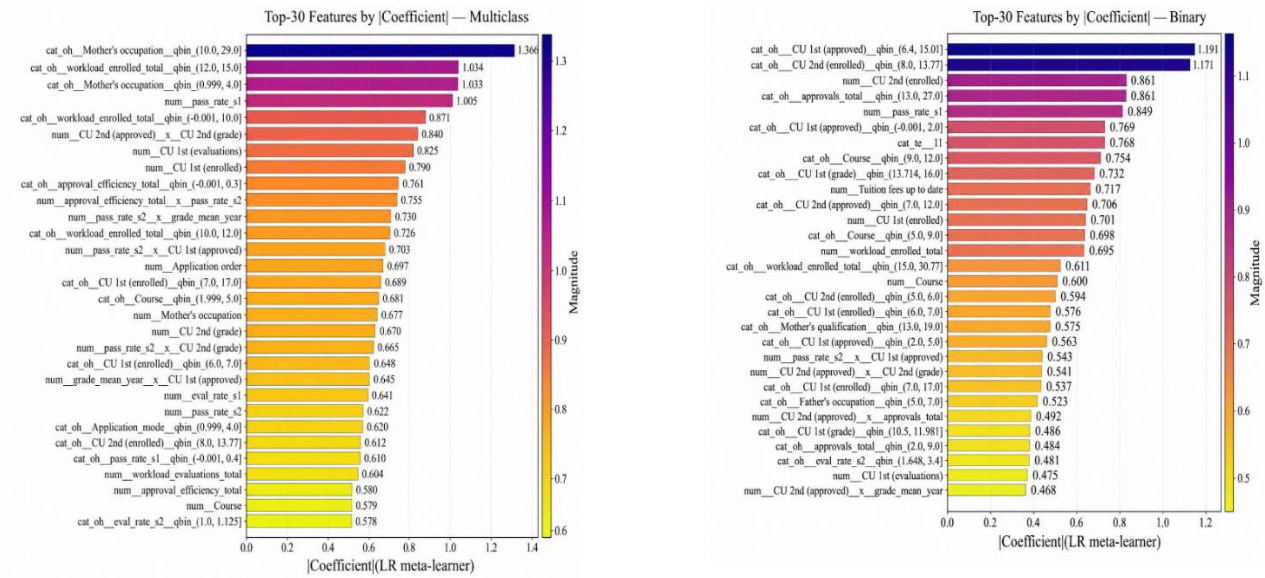


(a) Multiclass before (b) Multiclass after

(c) Binary before (d) Binary after

Figure 6. PCA before and after SMOTE

Note: PCA = Principal Component Analysis; SMOTE = Synthetic Minority Over-sampling Technique.



(a) Multiclass task (b) Binary task

Figure 7. Top-30 feature coefficients from LR-based Recursive Feature Elimination (RFE)

To assess whether predicted probabilities are suitable for risk scoring, we report the Maximum Calibration Error (MCE; uniform 10-bin scheme) and reliability diagrams (Table 6 and Figure 4); calibration reduces worst-case bin deviations for both the binary task and the OVR-macro multiclass task. In the binary task, sigmoid calibration reduced the worst deviation between bins in the uniform scheme from 0.8620 to 0.8070, which means that the most extreme gap between confidence and accuracy became smaller; the pattern in the reliability diagram also shows that some points "after" are closer to the identity line, although the improvement is not uniform across the entire probability range. In the multiclass task (OVR-aggregated), isotonic calibration provides a more consistent reduction in MCE, from 0.5468 to 0.4983 (uniform), and the "after" curve in the reliability diagram tends to move closer to the identity line, especially in the medium to high probability range. Overall, this evidence suggests that calibration primarily helps reduce worst-case errors across various probability ranges. Therefore, we report MCE and reliability diagrams as the primary basis for assessing the feasibility of probability in downstream risk analysis. The performance trend across Top-K values (K = 2–6) is summarized in Figure 5.

The Enrolled class is inherently harder to separate because it often reflects a transitional academic state between persistent progression (Graduate) and disengagement (Dropout). This 'boundary' nature makes its feature distribution overlap with both neighboring classes, which explains why Recall/F1 for Enrolled typically lags behind the other classes. In practice, this suggests that improving Enrolled detection may require class-aware strategies (e.g., tuned SMOTE neighborhood size, class-weighted learning, or thresholding policies focused on Enrolled), which we discuss as targeted improvement directions in Future Work.

From an operational perspective, the audited probabilities can be turned into simple risk bands for capacity-aware interventions. For example, institutions can define a High-Risk band (top-k% highest predicted risk), a Medium-Risk band (next tranche), and a Low-Risk band (remainder), then map each band to concrete actions (e.g., academic mentoring and remedial support for High-Risk; study-plan review for Medium-Risk; automated check-ins for Low-Risk). In this setting, MCE is useful as a worst-case reliability signal: lower MCE means the most extreme confidence–outcome mismatch becomes smaller, which reduces the chance of overconfident mis-triage in the riskiest bins.

4.5 Validation Hybrid Logistic Regression Stacking model

We analyze in this section the test results summarized in Section 4 and compare how HLR-STACK performs against its base learners. The test accuracies of all models are shown in Figure 5. For binary classification (graduate versus dropout), the accuracy of base learners is 88.26% to 89.50%. On the multiclass dropout–enrolled–graduate classification task, their accuracies vary between 72.62% and 76.38%. For the binary classification task, 98.07% accuracy is achieved by the HLR-STACK model with Top-K = 3 base learners. Increasing K to values between 4 and 6 does not consistently improve the results; therefore, we adopt K = 3 as the recommended configuration to keep the stack simple and interpretable; larger K values are reported for completeness but are not used as the default.

Under the same leakage-safe protocol, HLR-STACK consistently outperforms all base learners in accuracy, precision, recall, F1-score, and ROC-AUC. In the binary classification scenario, this model produces 96.36% precision, 97.25% recall, 96.80% F1-score, and 99.89% ROC-AUC. Meanwhile, in the multiclass classification scenario, HLR-STACK achieved a precision of 84.84%, a recall of 72.84%, an F1-score of 72.50%, and an ROC-AUC of 91.53%. Relative to the strongest single baseline (XGBoost), stacking improves accuracy by 8.6 points (from 89.50% to 98.07%) for binary classification and by 8.0 points (from 76.38% to 84.41%) for multiclass classification. This improvement can be explained by the role of the logistic regression meta-learner, which integrates OOF predictions from several complementary, powerful learners, thereby reducing the specific error bias of each underlying model. Visually, the PCA projection after applying SMOTE (Figure 6) shows a clearer interclass separation. Additionally, the Confusion Matrix (Figure 3) indicates that classification errors occur predominantly within the Enrolled class, which mostly has an intersection with the Graduate and Dropout classes. Taken as a whole, these findings endorse the validity of HLR-STACK as a solid model that performs well and is useful for decision-makers in higher education settings.

4.6 Student performance factor analysis

To understand the HLR-STACK model and answer RQ3, we selected the top 30 features using Logistic Regression-based RFE for multiclass as well as binary tasks (Figure 7). The interpretation should focus on the sign and magnitude of the Logistic Regression coefficients produced by the LR-based RFE ranking, to summarize how academic, workload, financial, and family background factors jointly relate to student outcomes.

In the multiclass setting, these are first and second semester pass rates, study load, and interactions between efficiency and performance (e.g., number of courses passed by average grade). These factors are direct measurements of students' performance. High scores represent students who are successfully moving forward through the curriculum, while low scores indicate individuals who progress through credits more slowly or do not pass assessments. Some quantile-related administrative and socio-economic standards, such as payment regularity and parents' education or employment, are also ranked high because these elements can affect the patterns of student performance.

The same trend holds for the binary task, graduate vs dropout. Academic components such as pass rates, course structure (registration, approval, and grading), study hours, and total course completions are still dominating forces. Higher unapproved workloads without specific criteria are one of the predictors of higher dropout potential. These additional financial and programmatic variables, such as tuition payment status, degree program, and parental background, provide further context.

To make the interpretation actionable, we map feature groups to intervention themes: low pass rates and low approval ratios suggest immediate academic remediation and study-plan adjustments, high workload with low completion points to workload rebalancing and closer mentoring, tuition-payment irregularity and socioeconomic indicators suggest proactive financial counseling and aid referral, and program/course-structure effects motivate course-level policy reviews (e.g., prerequisite alignment and assessment pacing).

This translation helps institutions move from 'prediction' to 'decision support' using interpretable signals.

Overall, these RFE patterns are consistent with student persistence theories: early academic integration (pass rates and approvals), manageable workload (enrolled vs completed credits), and adequate institutional/financial support jointly shape continuation risk. In practice, this means that academic indicators should trigger early academic support (e.g., tutoring and course-level remediation), while financial and family-context signals are better used to tailor support pathways (e.g., flexible payment plans, targeted advising, or connecting students to financial aid), rather than being treated as exclusionary criteria.

4.7 Discussion

The results show that the HLR-STACK framework performs better than individual models in both binary (graduate versus dropout) and multiclass (dropout, enrolled, and graduate) classification tasks. In binary classification, the Top-K = 3 stack, which includes XGB, LGBM, and GBoost, reaches an accuracy of nearly 0.98, with balanced precision, recall, F1-score, and ROC-AUC. For multiclass classification, the same Top-K = 3 stack achieves about 0.84 accuracy and an ROC-AUC above 0.91. These results answer RQ1 and RQ2, showing that HLR-STACK delivers strong predictive performance in both settings, while probability reliability is explicitly audited using MCE-based calibration evidence and reliability diagrams to inform downstream risk scoring.

Table 7. Comparative results with prior studies

References	Method	Classification	Accuracy
	With the same dataset		
[45]	Classical ensemble + LIME	Binary	0.785
[24]	Imbalance handling	Multiclass	0.749
[38]	Model comparison	Multiclass	0.773
[39]	Hybrid Fusion	Multiclass	0.805
[47]	HLRNN	Binary	0.960
[46]	Dataset distillation + SHAP	Binary	0.924
	With another dataset		
[41]	Ensemble technique	Binary	0.897
[42]	CNN-LSTM	Binary	0.940
[43]	LGBM	Binary (F1)	0.874
[44]	Gradient-Boosting (XGB)	Binary	0.850
Proposed Method	HLR-STACK Model	Multiclass Binary	0.844 0.981

Compared to prior studies on the same dataset and related higher-education cohorts, which typically report multiclass accuracies in the 0.70–0.80 range (e.g., [24, 38, 39]) and binary accuracies around 0.90 (e.g., [41-47]), HLR-STACK achieves an accuracy of 0.844 for the multiclass task and 0.981 for the binary task (see Table 7). The gains of approximately 8–9 percentage points over the best single models and previously published ensembles indicate that the combination of leakage-safe stacking and the in-fold SMOTE method explains most of the performance improvement. Calibration evidence is used to evaluate whether calibration is useful for downstream risk applications. At the same time, using a logistic regression meta-learner preserves interpretability by providing a transparent aggregation mechanism rather than a "black box" second-level model.

Factor analysis based on LR-driven RFE shows that per-semester pass rates, study load, efficiency indicators, average grades, and curricular unit structure are dominant predictors of persistence, complemented by financial and family background variables. This finding is consistent with the theory on retention, which maintains that decreasing the likelihood of attrition requires academic integration from an early stage, a manageable workload, and financial solvency. The LR meta-learner yields interpretable coefficients in the meta-feature space, while LR-based RFE provides a complementary ranking of variables for factor analysis. It will help close the distance from very high-dimensional ensemble models to educational needs for fair decision support systems. The relevance of pass rates and workload efficiency measures points to the importance of early interventions that target course completion and study load rather than just achievement.

4.7.1 Implications and limitations

A leakage-safe stacking pipeline paired with calibration auditing (MCE + reliability diagrams) can serve as the analytical core of an institutional early-warning decision-support system. Its probability estimates can be converted into stable and actionable risk brackets, enabling targeted interventions (e.g., remedial clinics for specific courses, adaptive study load planning, mentoring programs for students with poor pass rates). Financial indicators can be used to inform flexible payment plans, emergency grants, and counseling, whereas demographic and family context variables are best for support profiling but should not constitute exclusion criteria.

At the governance level, the proposed end-to-end solution with pre-processing, cross-fold balanced class distribution, and OOF predictions based on a stacking-designed mechanism in collaboration with post-training calibration contributes to a solid machine learning workflow for higher education. This framework allows for performing reproducible model behavior audits, like confusion matrix analysis, reliability diagrams, and performance stability across fold tests. It is furthermore of reasonable adaptability for other study programs or universities. As such, it can enforce responsible model governance and align with new recommendations on privacy-aware learning analytics.

However, although this study has its merits, it has several limitations. First, the data is taken from only one university in only one country, so these limitations of the dataset need to be tested empirically with other universities. Second, the predictors in our analysis are limited to administrative, academic, and socioeconomic features from the first two semesters. Although this design choice was made to avoid temporal leakage, it also leaves out risk factors that may emerge later in the program or in extracurricular activities. Finally, no digital traces of student behavioral data (e.g., LMS clickstreams or logged student interactions) were utilized, and attitude measures were not considered either; thereby limiting conclusions about learning processes beyond formal academic records.

4.7.2 Future research

Findings from the study may be utilized to stimulate further investigation by extending the validation of the HLR-STACK

model on other datasets and universities that have different educational programs and policies for reliability testing. Behavioral indices could provide more efficient methods for the early identification of dropout students in e-learning environments, serving as a solid basis for the design of an EWS together with other administrative and academic variables (e.g., activity in e-learning systems, homework submission degree, and participation in an online forum). From a modeling perspective, exploring alternative base-learner families beyond tree boosting, such as additive models, kernel-based approaches, Gaussian process classification, or ordinal and hierarchical formulations of study-status transitions, may further improve calibration and interpretability. Finally, implementing the HLR-STACK pipeline into a decision support system (DSS) that integrates real-time data with visualized risk bands and recommended concrete actions will run the framework as a practical model in data-driven prevention strategies in higher education.

5. CONCLUSION

This paper introduces HLR-STACK, a leakage-safe hybrid stacking approach that uses Logistic Regression as an interpretable meta-learner over a compact set of tree-based and boosting models. Beyond predictive accuracy, we present the method as a deployable information-systems pipeline with auditable probability outputs, supported by MCE-based reliability evidence and reliability diagrams for risk scoring. Logistic regression serves as an interpretable meta-learner on top of robust tree-based and boosting base models, while probability reliability is audited using MCE evidence and reliability diagrams to support risk-oriented use. HLR-STACK is designed for two operational decision-support tasks in higher education: (i) Graduate vs. Dropout identification and (ii) Dropout–Enrolled–Graduate differentiation. A strictly leakage-safe evaluation protocol was used, and HLR-STACK achieved accuracy gains of approximately 8–9 percentage points over the best single learners in both settings. These results confirm that a simple, transparent meta-learner can effectively exploit complementary error patterns among boosted ensembles.

In various tasks, HLR-STACK achieved an accuracy of 0.844 for multiclass configurations and 0.981 for binary configurations, with consistently high precision, recall, F1-score, and ROC-AUC. Factor analysis based on LR-driven RFE shows that first and second semester pass rates, workload and efficiency indicators, average grades, and curricular unit structure are the main predictors of persistence. These variables are complemented by financial and family context variables.

Overall, the Top-K = 3 configuration offers a strong performance–complexity trade-off, making it a practical analytical core for institutional decision support and risk-band-based early warning deployment. The probabilities and factor structures generated support interventions that promote efficient course completion, high academic achievement, adaptive workload planning, and targeted non-academic support.

In practical deployment, the proposed pipeline can be integrated into institutional decision-support systems as a repeatable scoring service: periodic batch scoring, risk-band stratification for capacity-aware interventions, and audit logs for governance and accountability. While the current study is based on a single-institution dataset, the workflow is designed to be portable, provided that feature definitions and time-order constraints are preserved and external validation across institutions and data modalities (e.g., LMS/behavioral traces) remains an important next step.

REFERENCES

[1] Latif, G., Abdelhamid, S.E., Fawagreh, K.S., Brahim, G.B., Alghazo, R. (2023). Machine learning in higher education: Students' performance assessment considering online activity logs. IEEE Access, 11: 69586-69600. https://doi.org/10.1109/ACCESS.2023.3287972

[2] Jain, A., Dubey, A.K., Khan, S., Panwar, A., Alkhatib, M., Alshahrani, A.M. (2025). A PSO weighted ensemble framework with SMOTE balancing for student dropout prediction in smart education systems. Scientific Reports, 15(1): 17463. https://doi.org/10.1038/s41598-025-97506-1

[3] Pham, N., Ngoc, H.P., Nguyen-Duc, A. (2025). Fairness for machine learning software in education: A systematic mapping study. Journal of Systems and Software, 219: 112244. https://doi.org/10.1016/j.jss.2024.112244

[4] Wang, S., He, J. (2025). Evaluating and forecasting undergraduate dropouts using machine learning for domestic and international students. Technologies, 13(11): 480. https://doi.org/10.3390/technologies13110480

[5] Adnan, M., Alarood, A.A.S., Uddin, M.I., ur Rehman, I. (2022). Utilizing grid search cross-validation with adaptive boosting for augmenting performance of machine learning models. PeerJ Computer Science, 8: e803. https://doi.org/10.7717/PEERJ-CS.803

[6] López-García, A., Blasco-Blasco, O., Liern-García, M., Parada-Rico, S.E. (2023). Early detection of students' failure using Machine Learning techniques. Operations Research Perspectives, 11: 100292. https://doi.org/10.1016/j.orp.2023.100292

[7] Berens, J., Diem, A., Rumert, L., Schneider, K., Wolter, S.C. (2025). Crossing individual university boundaries: A comprehensive approach to predicting dropouts in the higher education system. Higher Education. https://doi.org/10.1007/s10734-025-01509-w

[8] Sahlaoui, H., Nayyar, A., Agoujil, S., Jaber, M.M. (2021). Predicting and interpreting student performance using ensemble models and shapley additive explanations. IEEE Access, 9: 152688-152703. https://doi.org/10.1109/ACCESS.2021.3124270

[9] Tamada, M.M., Giusti, R., Netto, J.F.D.M. (2022). Predicting students at risk of dropout in technical course using LMS logs. Electronics, 11(3): 468. https://doi.org/10.3390/electronics11030468

[10] Alsubhi, B., Alharbi, B., Aljojo, N., Banjar, A., Tashkandi, A., Alghoson, A., Al-Tirawi, A. (2023). Effective feature prediction models for student performance. Engineering, Technology & Applied Science Research, 13(5): 11937-11944. https://doi.org/10.48084/etasr.6345

[11] Adekitan, A.I., Salau, O. (2020). Toward an improved learning process: The relevance of ethnicity to data mining prediction of students' performance. SN Applied Sciences, 2(1): 8. https://doi.org/10.1007/s42452-019-

1752-1
[12] Xue, H., Niu, Y. (2023). Multi-output based hybrid integrated models for student performance prediction. Applied Sciences, 13(9): 5384. https://doi.org/10.3390/app13095384
[13] Vaarma, M., Li, H. (2024). Predicting student dropouts with machine learning: An empirical study in Finnish higher education. Technology in Society, 76: 102474. https://doi.org/10.1016/j.techsoc.2024.102474
[14] Ouatik, F., Erritali, M., Ouatik, F., Jourhmane, M. (2022). Predicting student success using big data and machine learning algorithms. International Journal of Emerging Technologies in Learning, 17(12): 236-251. https://doi.org/10.3991/ijet.v17i12.30259
[15] Badal, Y.T., Sungkur, R.K. (2023). Predictive modelling and analytics of students' grades using machine learning algorithms. Education and Information Technologies, 28(3): 3027-3057. https://doi.org/10.1007/s10639-022-11299-8
[16] Maphosa, M., Doorsamy, W., Paul, B. (2024). Improving academic advising in engineering education with machine learning using a real-world dataset. Algorithms, 17(2): 85. https://doi.org/10.3390/a17020085
[17] Sun, D., Luo, R., Guo, Q., Xie, J., et al. (2023). A university student performance prediction model and experiment based on multi-feature fusion and attention mechanism. IEEE Access, 11: 112307-112319. https://doi.org/10.1109/ACCESS.2023.3323365
[18] Alhazmi, E., Sheneamer, A. (2023). Early predicting of students performance in higher education. IEEE Access, 11: 27579-27589. https://doi.org/10.1109/ACCESS.2023.3250702
[19] Angeioplastis, A., Aliprantis, J., Konstantakis, M., Tsimpiris, A. (2025). Predicting student performance and enhancing learning outcomes: A data-driven approach using educational data mining techniques. Computers, 14(3): 83. https://doi.org/10.3390/computers14030083
[20] Abuzinadah, N., Umer, M., Ishaq, A., Al Hejaili, A., et al. (2023). Role of convolutional features and machine learning for predicting student academic performance from MOODLE data. Plos One, 18(11): e0293061. https://doi.org/10.1371/journal.pone.0293061
[21] Holicza, B., Kiss, A. (2023). Predicting and comparing students' online and offline academic performance using machine learning algorithms. Behavioral Sciences, 13(4): 289. https://doi.org/10.3390/bs13040289
[22] Mastour, H., Dehghani, T., Moradi, E., Eslami, S. (2023). Early prediction of medical students' performance in high-stakes examinations using machine learning approaches. Heliyon, 9(7): e18248. https://doi.org/10.1016/j.heliyon.2023.e18248
[23] Khairy, D., Alharbi, N., Amasha, M.A., Areed, M.F., Alkhalaf, S., Abougalala, R.A. (2024). Prediction of student exam performance using data mining classification algorithms. Education and Information Technologies, 29(16): 21621-21645. https://doi.org/10.1007/s10639-024-12619-w
[24] Martins, M.V., Baptista, L., Machado, J., Realinho, V. (2023). Multi-class phased prediction of academic performance and dropout in higher education. Applied Sciences, 13(8): 4702. https://doi.org/10.3390/app13084702
[25] Alamgir, Z., Akram, H., Karim, S., Wali, A. (2024). Enhancing student performance prediction via educational data mining on academic data. Informatics in Education, 23(1): 1-24. https://doi.org/10.15388/infedu.2024.04
[26] Baashar, Y., Hamed, Y., Alkawsi, G., Capretz, L.F., Alhussian, H., Alwadain, A., Al-Amri, R. (2022). Evaluation of postgraduate academic performance using artificial intelligence models. Alexandria Engineering Journal, 61(12): 9867-9878. https://doi.org/10.1016/j.aej.2022.03.021
[27] Karalar, H., Kapucu, C., Gürüler, H. (2021). Predicting students at risk of academic failure using ensemble model during pandemic in a distance learning system. International Journal of Educational Technology in Higher Education, 18(1): 63. https://doi.org/10.1186/s41239-021-00300-y
[28] Kustitskaya, T.A., Esin, R.V., Vainshtein, Y.V., Noskov, M.V. (2024). Hybrid approach to predicting learning success based on digital educational history for timely identification of at-risk students. Education Sciences, 14(6): 657. https://doi.org/10.3390/educsci14060657
[29] Wang, D., Lian, D., Xing, Y., Dong, S., Sun, X., Yu, J. (2022). Analysis and prediction of influencing factors of college student achievement based on machine learning. Frontiers in Psychology, 13: 881859. https://doi.org/10.3389/fpsyg.2022.881859
[30] Elbourhamy, D.M., Najmi, A.H., Elfeky, A.I.M. (2023). Students' performance in interactive environments: An intelligent model. PeerJ Computer Science, 9: e1348. https://doi.org/10.7717/peerj-cs.1348
[31] Orrego Granados, D., Ugalde, J., Salas, R., Torres, R., López-Gonzales, J.L. (2022). Visual-predictive data analysis approach for the academic performance of students from a Peruvian University. Applied Sciences, 12(21): 11251. https://doi.org/10.3390/app122111251
[32] Bellaj, M., Dahmane, A.B., Boudra, S., Sefian, M.L. (2024). Educational data mining: Employing machine learning techniques and hyperparameter optimization to improve students' academic performance. International Journal of Online & Biomedical Engineering, 20(3): 55-74. https://doi.org/10.3991/ijoe.v20i03.46287
[33] Wakelam, E., Jefferies, A., Davey, N., Sun, Y. (2020). The potential for student performance prediction in small cohorts with minimal available attributes. British Journal of Educational Technology, 51(2): 347-370. https://doi.org/10.1111/bjet.12836
[34] Alruwais, N., Zakariah, M. (2023). Evaluating student knowledge assessment using machine learning techniques. Sustainability, 15(7): 6229. https://doi.org/10.3390/su15076229
[35] Begum, S., Padmannavar, S.S. (2022). Student performance prediction with BPSO feature selection and CNN classifier. International Journal of Advanced and Applied Sciences, 9(11): 84-92. https://doi.org/10.21833/ijaas.2022.11.010
[36] Nafea, A.A., Mishlish, M., AL-Ani, M.M., Alheeti, K.M.A., Mohammed, H.J. (2023). Enhancing Student's performance classification using ensemble modeling. Iraqi Journal for Computer Science and Mathematics, 4(4): 16. https://doi.org/10.52866/ijcsm.2023.04.04.016
[37] Kaensar, C., Wongnin, W. (2023). Predicting new student performances and identifying important attributes of admission data using machine learning techniques with hyperparameter tuning. Eurasia Journal of Mathematics, Science and Technology Education,

19(12): em2369. https://doi.org/10.29333/ejmste/13863
[38] Maheshwari, A., Malhotra, A., Hada, B.S., Ranka, M., Basha, M.S.A. (2024). Comparative analysis of machine learning models in predicting academic outcomes: Insights and implications for educational data analytics. In 2024 International Conference on Smart Systems for applications in Electrical Sciences (ICSSES), Tumakuru, India, pp. 1-7. https://doi.org/10.1109/ICSSES62373.2024.10561260
[39] Kannan, K.R., Abarna, K.T.M., Vairachilai, S. (2024). Enhancing student academic performance forecasting in technical education: A cutting-edge hybrid fusion method. International Journal of Electronics and Communication Engineering, 11(12): 146-153. https://doi.org/10.14445/23488549/IJECE-V11I12P114
[40] Realinho, V., Machado, J., Baptista, L., Martins, M.V. (2022). Predicting student dropout and academic success. Data, 7(11): 146. https://doi.org/10.3390/data7110146
[41] Rabelo, A.M., Zárate, L.E. (2025). A model for predicting dropout of higher education students. Data Science and Management, 8(1): 72-85. https://doi.org/10.1016/j.dsm.2024.07.001
[42] Talebi, K., Torabi, Z., Daneshpour, N. (2024). Ensemble models based on CNN and LSTM for dropout prediction in MOOC. Expert Systems with Applications, 235: 121187. https://doi.org/10.1016/j.eswa.2023.121187
[43] Seo, E.Y., Yang, J., Lee, J.E., So, G. (2024). Predictive modelling of student dropout risk: Practical insights from a South Korean distance university. Heliyon, 10(11): e30960. https://doi.org/10.1016/j.heliyon.2024.e30960
[44] Bravo, D.P., Alves, M.A.Z., Ensina, L.A., de Oliveira, L.E.S. (2023). Evaluating strategies to predict student dropout of a bachelor's degree in computer science. In Symposium on Knowledge Discovery, Mining and Learning (KDMiLe), pp. 1-8. https://doi.org/10.5753/kdmile.2023.232763.
[45] Hai, N.T., Phuong, L., Thi, N.T.T., Kim, S.A. (2024). A multivariate analysis of the early dropout using classical machine learning and local interpretable model-agnostic explanations. CTU Journal of Innovation and Sustainable Development, 16(Special issue: ISDS): 98-106. https://doi.org/10.22144/ctujoisd.2024.327
[46] Liu, H., Mao, M., Li, X., Gao, J. (2025). Model interpretability on private-safe oriented student dropout prediction. Plos One, 20(3): e0317726. https://doi.org/10.1371/journal.pone.0317726
[47] Mustofa, S., Emon, Y.R., Mamun, S.B., Akhy, S.A., Ahad, M.T. (2025). A novel AI-driven model for student dropout risk analysis with explainable AI insights. Computers and Education: Artificial Intelligence, 8: 100352. https://doi.org/10.1016/j.caeai.2024.100352