

Comparing machine learning models for academic success prediction in online learning

Sofia V. Velychko¹, Olena Y. Chepurna¹ and Olena H. Fedorenko²

¹National University “Odesa Law Academy”, 23 Fontanska Str., Odesa, 65009, Ukraine

²Donbas State Pedagogical University, 19 Henerala Batiuka Str., Sloviansk, 84116, Ukraine

Abstract. Early-warning models must balance the detection of unsuccessful outcomes with the workload of student support. We compare logistic regression, a decision tree, a random forest, and gradient boosting using 32,593 records from the Open University Learning Analytics dataset. Sixteen features are evaluated at 30, 60, and 90 days and over the full course using a student-grouped holdout and repeated grouped cross-validation. Gradient boosting has the highest cross-validation macro-F1; holdout macro-F1 is 0.706, 0.751, and 0.775 in the early windows. Assessment features provide substantial predictive information. Target risk recalls of 0.80–0.90 require flagging 48.3–73.3% of records with precision 0.541–0.662. These operating points are selected on holdout outcomes and require independent validation. Full-course performance is retrospective and uses a different population. Approximate statistical comparisons and unresolved feature-availability times limit inference. The results motivate a prospective evaluation of advisor support rather than the establishment of an effective deployment policy.

Keywords: learning analytics, academic success prediction, student risk, grouped cross-validation, online learning

1. Introduction

Online course records can reveal difficulties before the outcome is known, but a useful warning system must connect prediction to feasible student support. Accuracy alone does not show how many students will be contacted, how many eventual failures will be missed, or whether support improves learning. Experience with predictive learning analytics at the Open University emphasises the roles of teachers, institutional processes, and stakeholder participation in the use of predictions [9]. This motivates evaluating discrimination and advisor workload together.

We study the Open University Learning Analytics Dataset (OULAD) [14], which contains 32,593 module-presentation records from 28,785 distinct students. The outcome is binary: Pass/Distinction is class 1, while Fail/Withdrawn is class 0, termed the at-risk group. This grouping is a benchmark outcome; failure and withdrawal can reflect different circumstances and need not imply the same type of support. We compare Logistic Regression, Decision Tree, Random Forest, and Gradient Boosting at 30, 60, and 90 days from the start of the presentation, along with a full-course retrospective reference.

Educational prediction studies differ in targets, feature timing, evaluation populations, and validation designs. Regression of an examination score [1, 21], classification of grade categories [19], engagement prediction [22], and binary course completion [2] cannot be ranked using their reported accuracy values alone. OULAD supports comparisons under a common task definition, but repeated records from the same student require grouped splitting, and the exclusion of earlier withdrawals changes the population observed at each cutoff.

ORCID: 0000-0002-2728-6796 (S. V. Velychko); 0000-0002-1432-0799 (O. Y. Chepurna); 0000-0002-1897-874X (O. H. Fedorenko)

Email: sofiavelichko33@gmail.com (S. V. Velychko); chepurna@onua.edu.ua (O. Y. Chepurna); fedorenko.elena1209@gmail.com (O. H. Fedorenko)

Received
2026-08-17

Accepted
2026-10-15

Published
2026-10-17



© Copyright for this article by its authors, published by the Academy of Cognitive and Natural Sciences. This is an Open Access article distributed under the terms of the Creative Commons License Attribution 4.0 International (CC BY 4.0), which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

The contribution is an empirical benchmark and workload analysis, rather than a new learning algorithm. It combines (1) common student-level holdout assignments and training-fold preprocessing for four fixed model configurations; (2) feature-group ablation and permutation analysis across observation windows; and (3) descriptive risk-threshold operating points, with explicit boundaries on statistical inference and operational interpretation. Event-date filtering and exclusion of exam records address specific leakage mechanisms, while unverified grade-release times and registration eligibility remain limitations. The research questions are:

- RQ1: How does observed prediction performance vary across the implemented observation windows and evaluation populations?
- RQ2: How large are the observed differences between the four model configurations, and what do exploratory comparisons support?
- RQ3: What predictive information is added by assessment and VLE activity features beyond demographic and course information?
- RQ4: What detection–workload trade-offs appear when risk-score thresholds are varied?

Section 2 positions the study, section 3 specifies the implemented benchmark, and section 4 reports the results. Sections 5 and 6 discuss educational interpretation, limitations, and the questions above.

2. Related work

Ahmed et al. [1] compares regression models on two student-performance datasets and develops an explainable voting regressor; linear regression is a strong standalone model in that study. Its score-prediction task differs from the task of identifying unsuccessful course outcomes. Paul et al. [19] classify cumulative grade-point-average categories using Naive Bayes, JRip, J48, Random Forest, and CatBoost, with class balancing and ten-fold CV. Pronina and Pertsev [21] also distinguish numerical-score prediction from classification of performance levels. These studies motivate retaining a simple baseline and specifying the outcome and evaluation protocol before interpreting differences between models.

Althibyani [2] evaluates Logistic Regression and Random Forest on OULAD using four-outcome and pass/fail formulations, and discusses the information in student activity indicators. Raj and Renumol [22] use an OULAD subset to predict engagement, which is a different target from the binary final course outcome examined here. Tkach’s master’s dissertation [23] studies behavioural prediction and dropout using classifiers and survival models. A recent OULAD comparison by Omarbekova et al. [18] addresses early academic-performance prediction. These related tasks establish context, but their reported metrics are not used as numerical baselines for the present protocol.

Logistic Regression supplies a linear baseline [11]; CART trees provide partition-based classification [5]; Random Forest averages randomised trees [4]; and Gradient Boosting fits successive trees to improve a loss function [8]. These established model families provide a useful basis for comparison of aggregated tabular data without implying that any one family is best in every educational setting.

Cross-validation provides repeated performance estimates, but reuse of observations makes fold results dependent [6, 12]. In OULAD, grouping is also needed because a student can appear in multiple module presentations. Feature availability, training-only preprocessing, and the unit of uncertainty estimation are therefore part of the model comparison, not merely implementation details. Global permutation importance describes the model’s dependence on predictors, whereas local methods such as SHAP address a different explanatory question [15]; neither establishes a causal effect of changing a predictor.

For educational use, a warning must be interpreted in the context of the course and available support resources. Herodotou et al. [9] report implementation experience involving educational stakeholders, complementing model-performance research with the conditions under which predictions are used. The present benchmark examines alert workload but does not evaluate an intervention. Table 1 summarises its positioning.

Table 1

Related prediction settings and their implications for this benchmark. Scores across these tasks are not directly comparable.

Study	Prediction setting	Model or analysis focus	Relevance here
Ahmed et al. [1]	Student-performance scores	Regression and voting ensembles; explanations	Retain a linear baseline; distinguish regression from classification
Paul et al. [19]	CGPA categories	Five classifiers, balancing, ten-fold CV	Target and class distribution affect comparison
Pronina and Pertsev [21]	Scores and performance levels	Regression and classification	Choose metrics to match the outcome
Althibyani [2]	OULAD course outcomes	Logistic Regression and Random Forest	Compare a common binary task and feature windows
Raj and Renumol [22]	OULAD engagement	Activity-based engagement prediction	Engagement and final outcomes are distinct targets
This study	OULAD Fail/Withdrawn versus Pass/Distinction	Four fixed classifiers; grouped evaluation	Report discrimination alongside record-level alert workload

3. Methodology

This section describes the dataset, feature engineering, the formulation of the prediction task, the models compared, the experimental design, and the evaluation metrics.

3.1. Dataset: OULAD

The primary data source is the Open University Learning Analytics Dataset (OULAD) [14], containing anonymised data from the Open University UK for 2013–2014: 7 modules (AAA–GGG) offered in 22 presentations (sessions), 32,593 student-module-presentation records, and 28,785 unique students. Some students took multiple modules or re-enrolled, so the records are not independent – this directly affects the splitting strategy (section 3.5). The dataset comprises seven relational tables (table 2).

Table 2

Structure of OULAD tables and their usage in this study.

Table	Rows	Columns	Content	Use
courses	22	3	Modules, presentations, duration	Loaded for table summary
studentInfo	32,593	12	Demographics and final outcome	Predictors and target
studentRegistration	32,593	5	Registration and unregistration dates	Registration predictor and cohort filtering
assessments	206	6	Types, due dates, weights	Identify non-exam records and count due assessments
studentAssessment	173,912	5	Submissions, dates, scores	Assessment aggregates
studentVle	10,655,280	6	Daily student clicks	VLE aggregates; five columns loaded
vle	6,364	6	VLE resource descriptions	Not loaded

Central entities are studentInfo (demographics and final result) and studentVle (over 10.6M daily activity records). For memory efficiency, studentVle was read with only five required columns and compact data types.

3.2. Missing value handling

The CSV reader recognises standard missing-value representations and additionally treats “?” as missing. Table 3 reports the missing counts recorded by the analysis, rather than assuming that all missing entries use the same literal representation.

Table 3
Missing values in OULAD tables and handling strategy.

Table	Field	Missing	Share	Handling
studentInfo	imd_band	1,111	3.4%	Category “Unknown”
studentRegistration	date_registration	45	0.1%	Training-fold median
studentRegistration	date_unregistration	22,521	69.1%	No recorded unregistration; cohort filtering only
assessments	date	11	5.3%	Exam records excluded
studentAssessment	score	173	0.1%	Missing scores excluded from means; undefined means set to 0

Tables, courses, and studentVle contain no missing values. The missing date_unregistration is structural (the student did not leave the course) and must not be imputed. All other missing values are few and handled without record loss.

3.3. Observation unit and target variable

Each observation is one student in one module presentation. The analysis checks that module code, presentation code, and student identifier jointly form a unique key. From the four values of final_result, a binary target is formed: class 1 (“success”) = Pass + Distinction; class 0 (“at-risk group”) = Fail + Withdrawn. For the full dataset, class 0 has 17,208 records (52.8%), and class 1 has 15,385 (47.2%). The distribution is moderately balanced, so beyond accuracy, per-class metrics are essential.

Records with assessment_type = Exam are excluded from all assessment aggregates because they are close to the outcome and are generally unavailable at an early prediction horizon. Neither final_result nor date_unregistration is a predictor; the former defines the target, and the latter defines the cutoff-specific evaluation cohort. Excluding exam records and separating students’ addresses to address these specific leakage mechanisms, but do not verify the availability time of every retained feature.

3.4. Feature engineering and observation windows

Features are grouped into four functional categories (table 4). There are 16 raw features. The archived impurity-importance output contains 45 encoded feature names; the encoded dimension can vary when a category is absent from a training fold.

Aggregates use the key (module, presentation, student). At a finite cutoff t , VLE clicks satisfy $\text{date} \leq t$ and non-exam submissions satisfy $\text{date_submitted} \leq t$. Pre-start events with negative dates are retained. Let $n_i(t)$ count retained submission records for student i , and let $D_{mp}(t)$ count non-exam assessments in module-presentation (m, p) with due date $\leq t$. The implemented feature is

$$\text{share_submitted}_i(t) = \begin{cases} \min\{n_i(t)/D_{mp}(t), 1\}, & D_{mp}(t) > 0, \\ 0, & D_{mp}(t) = 0. \end{cases} \quad (1)$$

Table 4

Model features: 16 raw features grouped into four categories.

Feature	Group	Type	Description
gender	Demographic	Categorical	Gender
region	Demographic	Categorical	Region of residence
highest_education	Demographic	Categorical	Highest prior education level
imd_band	Demographic	Categorical	Index of multiple deprivation decile
age_band	Demographic	Categorical	Age group
disability	Demographic	Categorical	Declared disability flag
num_of_prev_attempts	Demographic	Numerical	Number of previous module attempts
studied_credits	Demographic	Numerical	Total credits studied concurrently
code_module	Course	Categorical	Module code
date_registration	Course	Numerical	Days from presentation start to registration
total_clicks	VLE Activity	Numerical	Total clicks in VLE
active_days	VLE Activity	Numerical	Number of days with VLE activity
mean_score	Assessment	Numerical	Mean observed non-exam score; 0 if no observed mean
n_assessments	Assessment	Numerical	Number of submitted assessments
share_submitted	Assessment	Numerical	Clipped submission-to-due-count ratio (see text)
has_submissions	Assessment	Numerical	Binary indicator of any submission

The numerator can include early submissions to assignments due after t ; it is not restricted to the denominator's assessment set. Consequently, this is a submission-ratio proxy rather than the literal fraction of due assignments completed. Banked assessments are retained when their submission-date condition is met. Mean score averages the available non-missing scores; an undefined mean is set to zero, including when submitted assessments have no observed score.

Window filtering retains records for which the unregistration date is missing or strictly greater than t . The implementation does not require $\text{date_registration} \leq t$ in addition, so eligibility at the cutoff is approximate for any late registrants. Moreover, a submission timestamp does not establish when its score became available. FULL uses all records and all non-exam events, without a cutoff-specific enrollment restriction. Table 5 reports these implemented cohorts.

Table 5

Sample sizes for each observation window.

Window	Records	Students	Train	Test	Class 0	Class 1	Risk (%)
W30	27,466	24,787	21,966	5,500	12,081	15,385	44.0%
W60	26,361	23,985	21,079	5,282	10,976	15,385	41.6%
W90	25,562	23,398	20,439	5,123	10,177	15,385	39.8%
FULL	32,593	28,785	26,074	6,519	17,208	15,385	52.8%

The archive reports 15,385 successful records across all windows; the excluded records are from the unsuccessful class in this dataset. Relative to FULL, W30, W60, and W90, remove 5,127, 6,232, and 7,031 records, respectively. The unsuccessful-class share consequently falls from 44.0% in W30 to 39.8% in W90, compared with 52.8% in FULL. These are empirical properties of the implemented unregistration filter, not a general claim that successful students can never unregister. Cohort composition must be considered alongside feature availability when comparing windows.

3.5. Train/test splitting

A student may have several module-presentation records, so splitting is grouped by `id_student`. For the full dataset, `StratifiedGroupKFold` uses five folds, shuffling, and seed 42; the first fold is reserved as the holdout set for student IDs. All other students form the training pool. Each cutoff filters records within these fixed assignments, so the actual student subsets evaluated need not be

identical across windows. Student overlap between training and holdout is checked programmatically. FULL contains 26,074 training and 6,519 holdout records.

Importantly, splitting is by student, not by presentation: test students study the same modules and presentations as training students. Thus, the experiment evaluates generalisation to new students, not to new courses.

3.6. Prediction task

Given the window-specific feature vector, the classifier predicts Pass or Distinction (class 1) or Fail or Withdrawn (class 0). Logistic Regression is the baseline; Decision Tree, Random Forest, and Gradient Boosting are alternatives. The task is a retrospective binary-outcome benchmark, with class-specific precision, recall, F1, macro-F1, accuracy, and ROC-AUC reported on the held-out set.

3.7. Models and hyperparameters

Each scikit-learn model [20] is a Pipeline that includes preprocessing. Numerical features use median imputation; Logistic Regression additionally uses standardisation. Categorical features use one-hot encoding with the first level dropped and unknown categories ignored. Preprocessing is fitted separately within each training fold, then refitted on the full training pool for holdout evaluation. This prevents validation and holdout records from being used to determine transformation parameters.

The environment metadata accompanying the tabulated outputs records Python 3.14.8, NumPy 2.5.3, pandas 3.0.6, scikit-learn 1.9.1, and Windows 11 with 16 logical CPUs. A separate top-level environment file records a different Linux setup; it should not be used as a substitute for the metadata accompanying these results. The stored per-model fit times exclude cross-validation, data preparation, and output generation, so they do not establish total experiment runtime.

Table 6 gives the fixed configurations. Estimators use seed 42. The optional randomised hyperparameter search was disabled (TUNE=0); all per-window entries in `best_params.json` are empty dictionaries. Equal search budgets, therefore, mean no tuning for any model, rather than evidence that these settings are optimal. Sensitivity to other configurations was not evaluated in the archived run.

Table 6
Models and their hyperparameters.

Model	Fixed classifier settings	Numeric scaling
Logistic Regression	<code>max_iter=1000, C=1, L2 penalty</code>	Standardization
Decision Tree	<code>max_depth=10</code>	None
Random Forest	<code>n_estimators=100, max_depth=12</code>	None
Gradient Boosting	<code>n_estimators=100, learning_rate=0.1, max_depth=3</code>	None

3.8. Experimental design and evaluation

Evaluation is conducted separately for the four feature formation scenarios (W30, W60, W90, FULL). The first three are designed for early risk detection; FULL is a control showing classification quality with full-course data.

The evaluation protocol for each window is:

1. Fix the student-level training and holdout pools on FULL, then apply the window-specific record filters within each pool.
2. Perform three repeats of grouped five-fold cross-validation on the training pool. The code uses seeds $42 + r$ for repeat $r = 0, 1, 2$, namely 42, 43, and 44. Identical splits are used for all models within a window; preprocessing is fitted within each fold. The resulting 15 estimates are paired but dependent.

3. Final training of each model on the entire training part and a single evaluation on the test set at threshold 0.5.
4. Selection of the best model by mean macro-F1 from cross-validation (not by test result), reducing the risk of model selection by test-set idiosyncrasies.
5. Compute record-level bootstrap intervals for every model. For the CV-selected model, also compute exploratory model comparisons, holdout-derived threshold operating points, and permutation importance.
6. Ablation study for W60 and FULL; model saving and reproducibility verification.

The holdout does not enter model fitting, model family selection, or hyperparameter tuning. Threshold operating points, permutation importance, and ablation summaries are nevertheless examined on holdout predictions; as such, they are descriptive analyses, not an independent validation of a deployment policy. Seed 42 controls estimator initialisation and bootstrap resampling; CV repeats additionally use seeds 43 and 44. The supplied research package contains the experiment script and archived outputs; the generated manuscript tables and figures trace back to those outputs.

For context, a constant reference predicts the majority class in each window’s training pool. Training class counts are obtained by subtracting holdout class counts from the complete cohort totals. Its holdout accuracy and macro-F1 are then calculated analytically; the prediction rule is not selected using holdout outcomes and does not enter the four-model CV comparisons.

3.9. Evaluation metrics

The confusion matrix (with class 1 = success as positive) has elements:

- TP – successful records correctly classified as successful;
- TN – at-risk records correctly classified as risk;
- FP – at-risk records incorrectly classified as success (missed risk);
- FN – successful records incorrectly classified as risk (false alarm).

For class $k \in \{0, 1\}$, we compute:

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN}, \quad (2)$$

$$\text{Precision}_k = \frac{n_{kk}}{\text{number of objects assigned to class } k}, \quad (3)$$

$$\text{Recall}_k = \frac{n_{kk}}{\text{number of actual objects of class } k}, \quad (4)$$

$$F1_k = \frac{2 \cdot \text{Precision}_k \cdot \text{Recall}_k}{\text{Precision}_k + \text{Recall}_k}, \quad (5)$$

$$\text{Macro-F1} = \frac{F1_0 + F1_1}{2}. \quad (6)$$

For class 0, $\text{Recall}_0 = TN/(TN + FP)$ is the fraction of unsuccessful records detected, and $\text{Precision}_0 = TN/(TN + FN)$ is the unsuccessful fraction among records flagged. False alerts correspond to FN under the success-positive convention used here; missed risks correspond to FP. Their relative educational costs depend on the support policy and are not estimated in this study. ROC-AUC describes ranking discrimination across thresholds.

For paired CV macro-F1 differences $d_1, \dots, d_N$, we report the corrected statistic of Nadeau and Bengio [3, 17]:

$$t = \frac{\bar{d}}{\sqrt{(1/N + n_{\text{val}}/n_{\text{fit}})s_d^2}}, \quad N = 15, \quad df = N - 1. \quad (7)$$

The implementation approximates the validation-to-fitting size ratio to 1/4, yielding a variance factor of $1/15 + 1/4$. Both folds and repeats reuse observations, and grouped folds can have unequal

record counts. The statistic is therefore approximate; neither independence across repeats nor guaranteed conservatism is assumed. Comparisons involving the CV-selected winner are exploratory because the same CV scores selected that winner. Absolute macro-F1 differences are emphasised rather than applying conventional Cohen’s d cutoffs to dependent fold scores.

A record-level McNemar calculation compares the two classifiers’ error rates on the same holdout records [16]. Let b count records correct only for Gradient Boosting, and c count records correct only for the comparator. Under the independent-record null, $B \sim \text{Binomial}(b + c, 1/2)$, and the two-sided value is $\min\{1, 2\Pr[B \leq \min(b, c)]\}$. This tests marginal correctness, not differences in macro-F1 or in the errors the models make. Because a student can contribute several records, the nominal exact-binomial p -value does not provide an exact student-clustered test.

Holm’s step-down correction [10] is applied to all 12 comparisons (three comparators across four windows), separately for the CV and McNemar families. Within-window corrections are retained in the accompanying derived-statistics file. These adjustments address the stated multiplicity, but do not repair clustering, approximate dependence corrections, or selection on CV results. They are presented as exploratory diagnostics rather than confirmatory significance claims.

For each fitted model, 1000 bootstrap samples [7] of holdout records provide percentile intervals for accuracy, risk-class recall, and macro-F1. The archived intervals condition on a fixed fitted model and resample records, not students. They omit both training uncertainty and dependence among a student’s records; their coverage for student-level generalisation is not established. No paired interval for an ablation difference is available.

4. Results

This section presents the main experimental results: overall performance across windows and models (section 4.1), detailed per-window analysis (section 4.2), ablation study (section 4.3), threshold analysis (section 4.5), statistical significance (section 4.6), and robustness checks (section 4.7).

4.1. Overall performance

Table 7 presents test-set metrics at threshold 0.5; table 8 shows cross-validation results (mean $\pm$ standard deviation over 15 estimates). The best macro-F1 in each window is highlighted in bold.

Table 7

Test-set metrics at threshold 0.5. Best macro-F1 per window in bold.

Window	Model	Accuracy	AUC	Prec. ₀	Rec. ₀	Prec. ₁	Rec. ₁	Macro-F1
W30	LR	0.7087	0.7748	0.7098	0.5733	0.7082	0.8154	0.6961
W30	DT	0.6962	0.7388	0.6928	0.5576	0.6980	0.8053	0.6829
W30	RF	0.7207	0.7849	0.7401	0.5642	0.7109	0.8440	0.7060
W30	GB	0.7180	0.7862	0.7202	0.5885	0.7168	0.8200	0.7063
W60	LR	0.7590	0.8229	0.7642	0.6113	0.7564	0.8648	0.7431
W60	DT	0.7429	0.7710	0.7341	0.6023	0.7475	0.8437	0.7272
W60	RF	0.7626	0.8285	0.7842	0.5950	0.7526	0.8827	0.7445
W60	GB	0.7651	0.8312	0.7657	0.6299	0.7647	0.8619	0.7508
W90	LR	0.7839	0.8429	0.8023	0.6090	0.7759	0.9002	0.7629
W90	DT	0.7767	0.8045	0.7688	0.6305	0.7806	0.8739	0.7587
W90	RF	0.7911	0.8535	0.8311	0.5987	0.7750	0.9191	0.7685
W90	GB	0.7948	0.8544	0.8179	0.6256	0.7847	0.9074	0.7753
FULL	LR	0.9150	0.9720	0.9487	0.8870	0.8822	0.9464	0.9150
FULL	DT	0.9112	0.9529	0.9331	0.8960	0.8886	0.9282	0.9111
FULL	RF	0.9167	0.9739	0.9606	0.8783	0.8757	0.9597	0.9167
FULL	GB	0.9195	0.9757	0.9509	0.8937	0.8886	0.9483	0.9194

Table 8Macro-F1 from repeated grouped cross-validation (mean $\pm \sigma$, 15 estimates). Best per window in bold.

Model	W30	W60	W90	FULL
Logistic Regression	0.6927 $\pm$ 0.0055	0.7339 $\pm$ 0.0073	0.7557 $\pm$ 0.0077	0.9191 $\pm$ 0.0030
Decision Tree	0.6718 $\pm$ 0.0050	0.7173 $\pm$ 0.0061	0.7416 $\pm$ 0.0098	0.9109 $\pm$ 0.0036
Random Forest	0.7018 $\pm$ 0.0058	0.7435 $\pm$ 0.0050	0.7630 $\pm$ 0.0075	0.9214 $\pm$ 0.0021
Gradient Boosting	0.7049 $\pm$ 0.0046	0.7475 $\pm$ 0.0056	0.7694 $\pm$ 0.0090	0.9236 $\pm$ 0.0026

Gradient Boosting has the highest mean CV macro-F1 in all four windows (0.7049, 0.7475, 0.7694, and 0.9236), so it is selected for detailed analyses. The holdout scores are close to the CV means, but their agreement on a single split does not demonstrate the absence of overfitting or establish robustness to another cohort. The result tables use LR, DT, RF, and GB for Logistic Regression, Decision Tree, Random Forest, and Gradient Boosting.

Figure 1 summarises holdout macro-F1 across the implemented windows. For Gradient Boosting, scores are 0.7063, 0.7508, 0.7753, and 0.9194; the corresponding record-bootstrap intervals are [0.6940; 0.7186], [0.7391; 0.7625], [0.7635; 0.7868], and [0.9129; 0.9257]. The upward sequence is descriptive: the populations differ, and marginal intervals do not constitute a paired cross-window test or separate the effect of additional information from cohort selection.

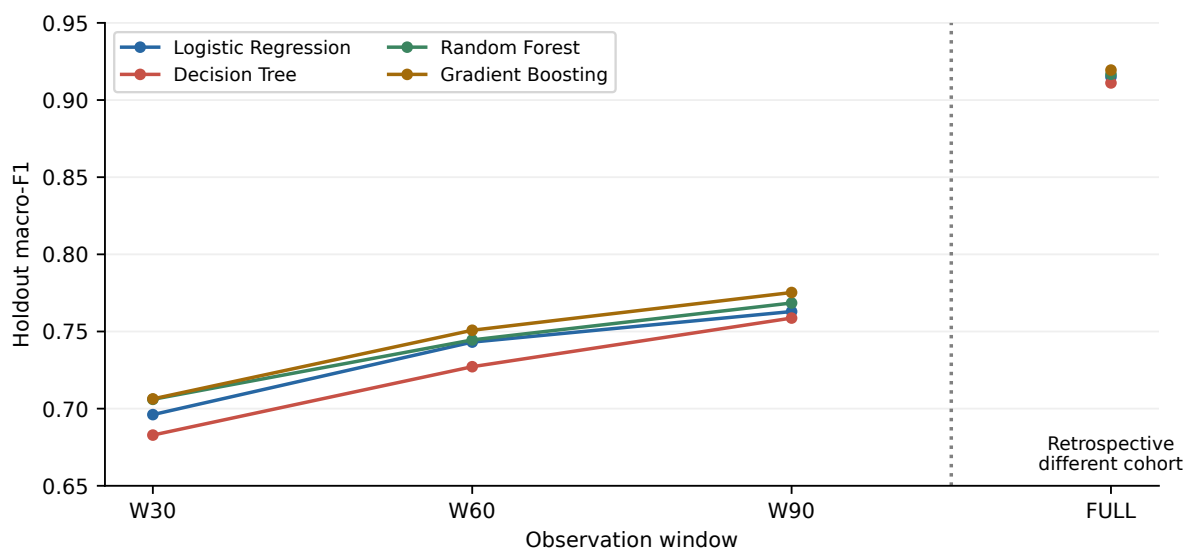


Figure 1: Holdout macro-F1 by model and window. FULL is a retrospective reference to a different population; no line joins it to the earlier windows.

The increase from W90 macro-F1 0.7753 to FULL 0.9194 is not a controlled estimate of the value of waiting longer. FULL uses complete non-exam records and includes early withdrawers excluded from the earlier cohorts. Both information and population change; their separate contributions have not been measured. FULL is therefore a retrospective reference, not an early-warning result or a formal performance upper bound.

Table 9 places the learned models in context. A rule that predicts the training-majority class has holdout macro-F1 of approximately 0.35–0.38, below all four classifiers. Its risk recall is zero in W30–W90 because it predicts success for every record, while in FULL it predicts risk for every record. It therefore provides a discrimination reference, not an advisor-contact policy or an adequate assessment of the value of intervention.

Table 9

Constant training-majority reference, calculated from archived class counts. No classifier is fitted for this reference.

Window	Predicted class	Accuracy	Macro-F1	Recall ₀
W30	1	0.5595	0.3588	0.0
W60	1	0.5825	0.3681	0.0
W90	1	0.6006	0.3752	0.0
FULL	0	0.5280	0.3455	1.0

4.2. Per-window analysis

At 30 days (window W30), Gradient Boosting achieves macro-F1 0.7063 and ROC-AUC 0.7862. Risk-class recall is 0.5885, so 997 of 2,423 at-risk holdout records (41.1%) are missed. Of 3,077 successful records, 554 (18.0%) receive false alerts. These are record counts, not unique students (table 10).

Table 10

Confusion matrices on test set for Gradient Boosting. FP = missed risk (at-risk predicted as success); FN = false alarm (success predicted as risk).

Window	TN	FP (missed risk)	FN (false alert)	TP
W30	1426	997	554	2523
W60	1389	816	425	2652
W90	1280	766	285	2792
FULL	3076	366	159	2918

At W60, Gradient Boosting reaches macro-F1 0.7508 and ROC-AUC 0.8312 (figure 2). Risk-class recall is 0.6299: 816 of 2,205 at-risk records (37.0%) are missed. There are 425 false alerts among 3,077 successful records (13.8%).

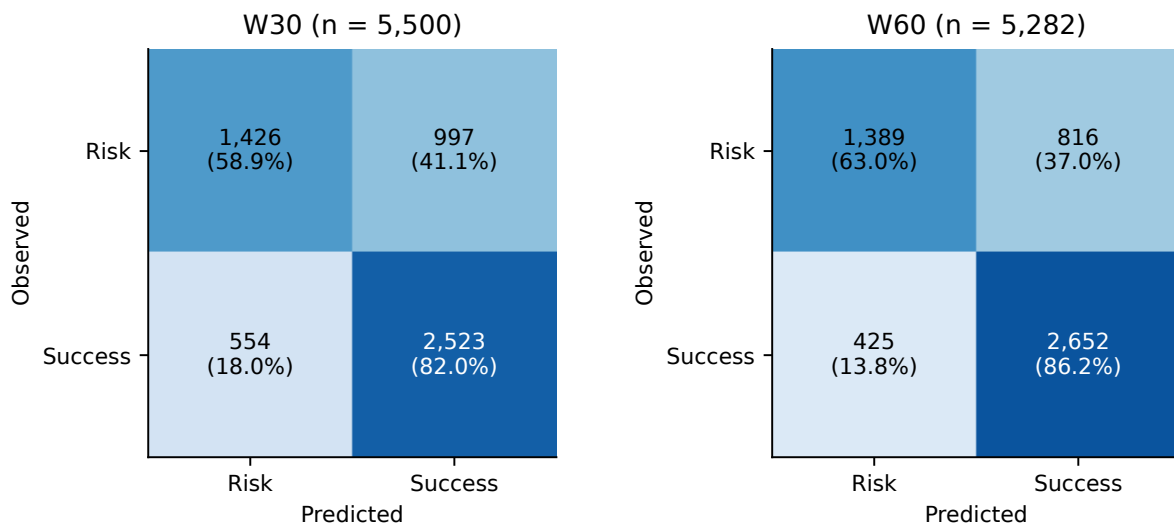


Figure 2: Gradient Boosting holdout confusion matrices for W30 and W60. Cells show record counts and percentages within each observed class; shading uses a common 0–100% row-percentage scale.

At W90, macro-F1 is 0.7753, and ROC-AUC is 0.8544. Risk-class recall is 0.6256: 766 of 2,046 at-risk records (37.4%) are missed. There are 285 false alerts among 3,077 successful records (9.3%). Thus, risk recall does not increase monotonically with the observation window length.

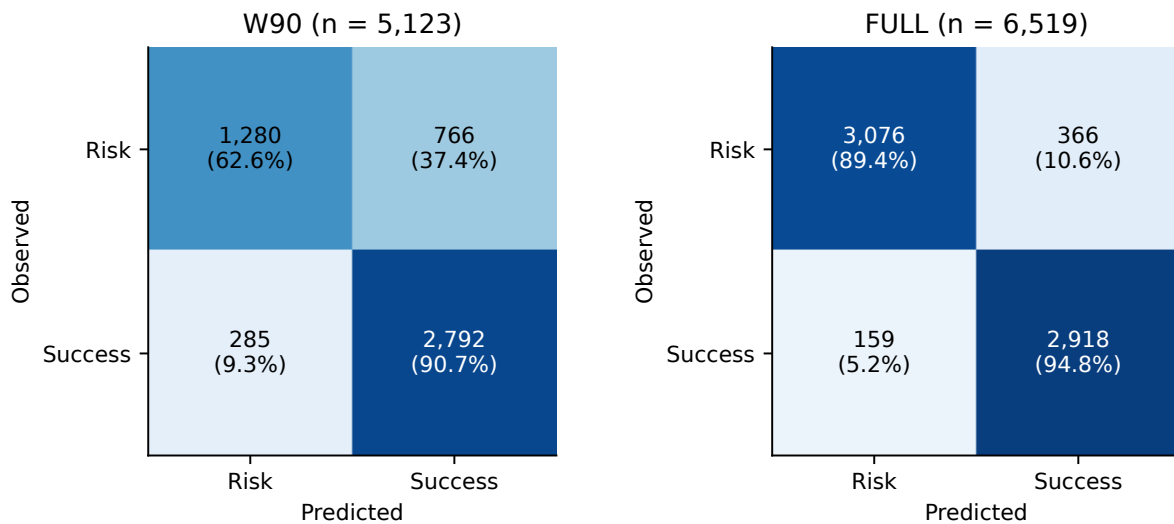


Figure 3: Gradient Boosting holdout confusion matrices for W90 and FULL. Cells show record counts and row percentages; FULL uses a different population.

FULL yields macro-F1 0.9194 and ROC-AUC 0.9757. Of 3,442 at-risk records, 366 (10.6%) are missed; of 3,077 successful records, 159 (5.2%) receive false alerts. These results describe the full retrospective population (figure 3).

4.3. Ablation study: feature group contributions

To quantify the contribution of each feature group, an ablation study was conducted for W60 and FULL, comparing four scenarios:

- A: Demographic + Course (10 raw features)
- B: A + VLE Activity (12 raw features)
- C: A + Assessment (14 raw features)
- D: All features (16 raw / 45 encoded)

Scenarios B and C independently extend A; comparing B, C, and D reveals individual and joint contributions. Ablation uses one cross-validation repeat.

Table 11 shows test-set macro-F1 for all models and scenarios. Table 12 details Gradient Boosting results with Recall₀ and ROC-AUC.

Table 11

Test-set Macro-F1 by ablation scenario.

Window	Scenario	Raw features	LR	DT	RF	GB
W60	A	10	0.5960	0.5634	0.5618	0.5798
W60	B	12	0.6963	0.6545	0.6871	0.6895
W60	C	14	0.7295	0.7289	0.7359	0.7420
W60	D	16	0.7431	0.7272	0.7445	0.7508
FULL	A	10	0.6292	0.6094	0.6218	0.6313
FULL	B	12	0.8340	0.8328	0.8443	0.8462
FULL	C	14	0.9141	0.9110	0.9161	0.9180
FULL	D	16	0.9150	0.9111	0.9167	0.9194

In W60, the combined demographic-and-course scenario A yields macro-F1 0.5634–0.5960 across the four models. For Gradient Boosting, adding VLE activity raises macro-F1 from 0.5798 to 0.6895,

Table 12

Gradient Boosting detailed ablation results.

Window	Scenario	Macro-F1	Recall ₀	ROC-AUC
W60	A	0.5798	0.3338	0.6618
W60	B	0.6895	0.5302	0.7704
W60	C	0.7420	0.6059	0.8248
W60	D	0.7508	0.6299	0.8312
FULL	A	0.6313	0.6819	0.6782
FULL	B	0.8462	0.8196	0.9265
FULL	C	0.9180	0.8960	0.9744
FULL	D	0.9194	0.8937	0.9757

while adding assessment features raises it to 0.7420; the full feature set reaches 0.7508. Scenario A is not a demographic-only control, so it cannot isolate demographic contribution. Macro-F1 also has no universal 0.5 baseline for these class proportions.

In FULL, Gradient Boosting macro-F1 is 0.6313 for A, 0.8462 for B, 0.9180 for C, and 0.9194 for D. The D–C difference is 0.0014 (figure 4). Because no paired confidence interval or test was computed for this contrast, its uncertainty and statistical significance cannot be inferred from the single-model bootstrap intervals.

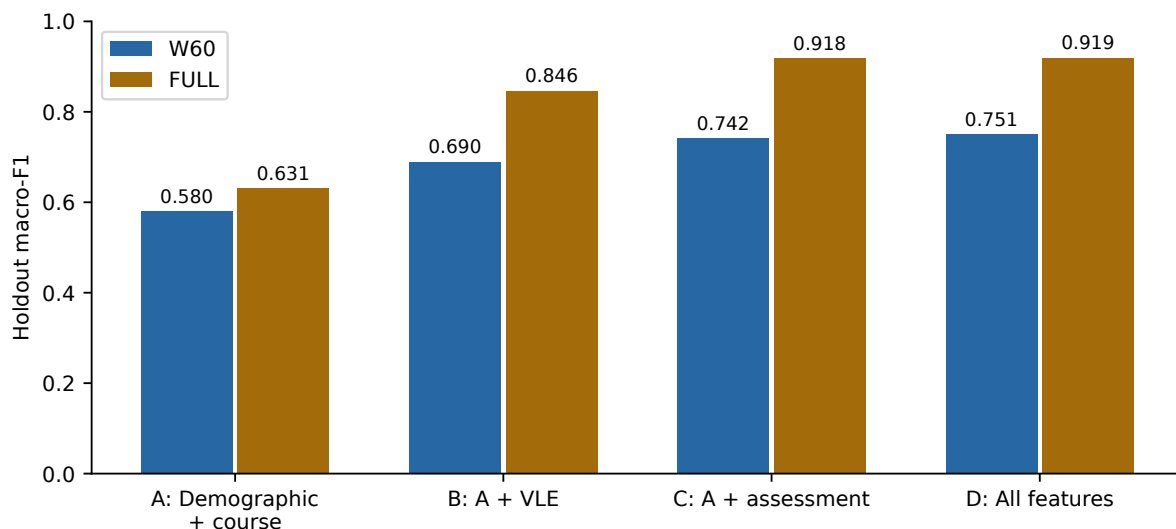


Figure 4: Gradient Boosting holdout macro-F1 by feature scenario. A contains demographic and course features; B and C add VLE and assessment features, respectively; D contains all features. No uncertainty estimates for differences between scenarios are available.

The group comparison supports a substantial predictive contribution from assessment features beyond demographic and course information, with a smaller additional contribution from VLE aggregates in the full model. It does not establish which behaviour leads to success or whether changing a predictor would improve outcomes. The permutation results below, rather than the group ablation alone, describe individual-feature associations.

Features `n_assessments`, `share_submitted`, and `has_submissions` are highly correlated, as are `total_clicks` and `active_days`; permutation importance distributes and attenuates each's contribution (evident from low `n_assessments` importance in FULL despite high `share_submitted`). Therefore, ablation analysis is more reliable for group-level conclusions. Additionally, the first module category (AAA) has no separate feature due to `drop="first"`, which does not affect permutation importance on raw features but complicates the interpretation of impurity-based importance per category.

4.4. Feature importance analysis

Permutation importance for Gradient Boosting (best model in all windows) was computed on the test set: each raw feature's values were randomly shuffled 5 times, and importance is the mean macro-F1 decrease. Results for the original (pre-encoding) features are in table 13 and figure 5.

Table 13

Permutation importance of Gradient Boosting (mean macro-F1 decrease over 5 permutations).

Feature	W30	W60	W90	FULL
mean_score	0.0768	0.1079	0.1068	0.1216
share_submitted	-0.0013	0.0291	0.0835	0.2600
n_assessments	-0.0001	0.0013	0.0150	0.0029
code_module	0.0549	0.0464	0.0203	0.0053
active_days	0.0299	0.0277	0.0146	0.0180
total_clicks	0.0101	0.0020	0.0034	0.0074
highest_education	0.0085	0.0076	0.0060	-0.0006
num_of_prev_attempts	0.0035	-0.0002	0.0005	0.0000

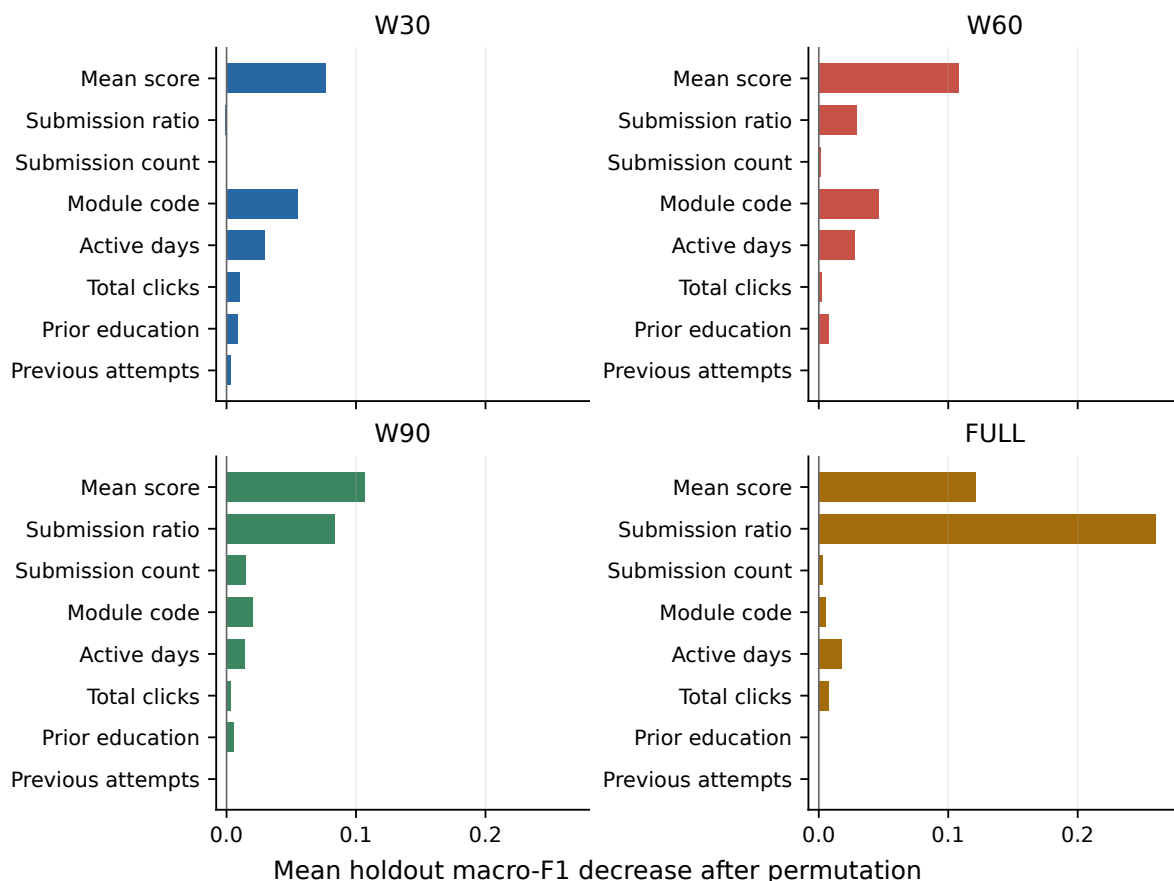


Figure 5: Gradient Boosting permutation importance for eight selected raw features. Negative values are retained; panels share the same horizontal scale. These are predictive associations, not causal effects.

The dominant factor in early windows is mean_score (0.077–0.108 for W30–W60). With increasing window length, share_submitted becomes dominant (0.029 in W60, 0.084 in W90, 0.260 in FULL), becoming the prevailing feature in FULL. Regularity of study (active_days) is consistently more important than action volume (total_clicks): 0.030, 0.028, 0.015, 0.018 vs. 0.010, 0.002, 0.003, 0.007. Module code has a notable influence in the early windows (0.055 in W30), reflecting differ-

ences in module structure (first-assignment deadlines, difficulty), but its role drops to 0.005 in FULL. Remaining features (gender, age, region, disability, credits, registration date, has_submissions) have importance ≤ 0.003 ; only highest_education (0.006–0.009 in W30–W90) and num_of_prev_attempts (0.0035 in W30) show weak influence.

Built-in impurity-based importance of Gradient Boosting gives a consistent picture: in W30–W60, mean_score is highest (0.35 and 0.55); in W90 and FULL, share_submitted dominates (0.53 and 0.88).

4.5. Threshold analysis: risk detection trade-offs

For the CV-selected Gradient Boosting model, define the risk score $R_i = 1 - \hat{P}_i(\text{success})$ and flag a record when $R_i \geq \tau$. The empirical precision–recall calculation selects the highest observed threshold that attains each target recall: 0.80, 0.85, 0.90, and 0.95. Table 14 reports precision, workload, missed-risk records, false alerts, and lift (precision divided by the observed risk prevalence). Holdout prevalences are 44.1%, 41.7%, 39.9%, and 52.8% for W30, W60, W90, and FULL. All counts and workload percentages refer to module-presentation records; contacting unique students would require deduplication.

Table 14

Threshold analysis for Gradient Boosting: risk thresholds achieving target Recall₀.

Window	Target	Cutoff	Prec. ₀	Flagged (%)	Missed	False alerts	Lift
W30	0.80	0.3465	0.6057	58.2	484	1262	1.37
W30	0.85	0.3094	0.5767	64.9	363	1512	1.31
W30	0.90	0.2679	0.5409	73.3	242	1851	1.23
W30	0.95	0.2080	0.4962	84.3	121	2337	1.13
W60	0.80	0.3242	0.6433	51.9	441	978	1.54
W60	0.85	0.2655	0.5907	60.1	330	1299	1.42
W60	0.90	0.2227	0.5541	67.9	219	1598	1.33
W60	0.95	0.1716	0.5024	78.9	110	2075	1.20
W90	0.80	0.3099	0.6622	48.3	409	835	1.66
W90	0.85	0.2598	0.6170	55.0	306	1080	1.54
W90	0.90	0.2018	0.5548	64.8	204	1478	1.39
W90	0.95	0.1447	0.4944	76.8	102	1988	1.24
FULL	0.80	0.9219	0.9960	42.4	688	11	1.89
FULL	0.85	0.7356	0.9829	45.7	516	51	1.86
FULL	0.90	0.4760	0.9477	50.1	344	171	1.79
FULL	0.95	0.2111	0.8623	58.2	172	522	1.63

The table 14 and figure 6 illustrate the workload associated with higher empirical recall. At W60, with a target recall of 0.80, 51.9% of records are flagged, with a precision of 0.643, 441 missed-risk records, and 978 false alerts. At target recall 0.90, 67.9% are flagged, with precision 0.554, 219 missed-risk records, and 1,598 false alerts. Across the early windows and target recalls 0.80–0.90, the observed flagged fraction is 48.3–73.3% and precision is 0.541–0.662. At W60, target recall is 0.95; workload rises further to 78.9%. FULL is retrospective and should not be used to infer early-warning workload.

These operating points are selected and evaluated on the same holdout outcomes. Their performance is descriptive and can be optimistic for a subsequently deployed threshold. A new evaluation should determine the cutoff using training-only out-of-fold predictions or a separate validation cohort, then assess the fixed policy on untouched data. Ranking and empirical precision–recall trade-offs do not require calibrated probabilities; probability calibration matters when scores are interpreted as absolute risks.

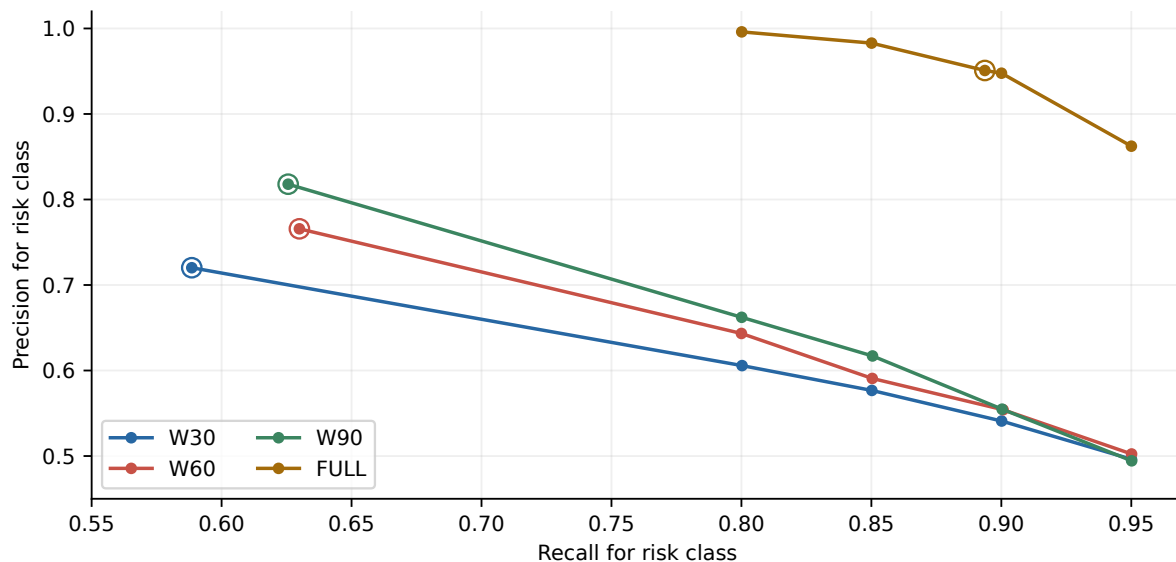


Figure 6: Selected empirical precision–recall operating points for the risk class. Open circles mark the default threshold; other points attain the target recalls in table 14. Connecting lines are visual guides, not complete precision–recall curves.

4.6. Exploratory statistical comparisons

Tables 15 and 16 separate the two different hypotheses. CV comparisons concern mean paired macro-F1; the McNemar calculation concerns marginal correctness on holdout records. LR, DT, RF, and GB denote Logistic Regression, Decision Tree, Random Forest, and Gradient Boosting, respectively, throughout the results tables. Holm adjustment uses 12 comparisons per test family.

Table 15

Exploratory corrected CV comparisons of Gradient Boosting against each comparator. Raw and Holm-adjusted values use the approximate corrected statistic.

Window	vs.	Δ CV macro-F1	t	Raw p	Holm p (12)
W30	LR	0.0121	4.00	0.0013	0.0066
W30	DT	0.0331	8.69	5.2×10^{-7}	5.7×10^{-6}
W30	RF	0.0031	1.29	0.2191	0.2191
W60	LR	0.0136	7.19	4.6×10^{-6}	3.7×10^{-5}
W60	DT	0.0301	9.01	3.4×10^{-7}	4.0×10^{-6}
W60	RF	0.0040	2.03	0.0620	0.1860
W90	LR	0.0138	5.35	0.0001	0.0007
W90	DT	0.0278	8.32	8.6×10^{-7}	8.6×10^{-6}
W90	RF	0.0064	3.32	0.0051	0.0202
FULL	LR	0.0046	4.59	0.0004	0.0025
FULL	DT	0.0128	7.44	3.1×10^{-6}	2.8×10^{-5}
FULL	RF	0.0023	1.91	0.0763	0.1860

The corrected CV diagnostic yields Holm-adjusted values below 0.05 for GB versus DT and LR in every window, and for GB versus RF only in W90 ($p_{\text{Holm}} = 0.0202$). The absolute CV macro-F1 differences versus RF are 0.0023–0.0064. The record-level McNemar calculation gives adjusted values below 0.05 for GB versus DT in all windows, and versus LR only in W90 ($p_{\text{Holm}} = 0.0423$); the FULL comparison against LR no longer meets this threshold after adjustment across windows ($p_{\text{Holm}} = 0.1264$). No GB–RF McNemar comparison meets that threshold. Failure to reject does not demonstrate practical equivalence, and clustered inference is needed before treating these nominal

Table 16

Nominal record-level McNemar comparisons. b and c count records correctly only for Gradient Boosting or only for the comparator. Student clustering is not accounted for.

Window	vs.	b	c	Raw p	Holm p (12)
W30	LR	363	312	0.0542	0.3253
W30	DT	408	288	6.1×10^{-6}	6.8×10^{-5}
W30	RF	156	171	0.4389	0.8777
W60	LR	240	208	0.1429	0.7147
W60	DT	373	256	3.5×10^{-6}	4.2×10^{-5}
W60	RF	129	116	0.4434	0.8777
W90	LR	223	167	0.0053	0.0423
W90	DT	269	176	1.2×10^{-5}	0.0001
W90	RF	106	87	0.1950	0.7147
FULL	LR	85	56	0.0181	0.1264
FULL	DT	174	120	0.0019	0.0175
FULL	RF	87	69	0.1733	0.7147

results as confirmatory.

The intervals in table 17 describe uncertainty under the archived record-resampling scheme. Overlap between two marginal intervals is not a test of a paired model difference.

Table 17

Archived 95% record-bootstrap percentile intervals, conditional on the fitted models. These intervals do not account for student clustering or model-refitting uncertainty.

Window	Model	Accuracy	Recall ₀	Macro-F1
W30	LR	[0.6967; 0.7202]	[0.5530; 0.5919]	[0.6840; 0.7076]
W30	DT	[0.6842; 0.7082]	[0.5390; 0.5763]	[0.6708; 0.6949]
W30	RF	[0.7084; 0.7325]	[0.5441; 0.5837]	[0.6927; 0.7179]
W30	GB	[0.7062; 0.7302]	[0.5684; 0.6072]	[0.6940; 0.7186]
W60	LR	[0.7486; 0.7707]	[0.5912; 0.6321]	[0.7315; 0.7548]
W60	DT	[0.7315; 0.7546]	[0.5823; 0.6228]	[0.7150; 0.7399]
W60	RF	[0.7512; 0.7739]	[0.5743; 0.6153]	[0.7327; 0.7564]
W60	GB	[0.7543; 0.7762]	[0.6100; 0.6493]	[0.7391; 0.7625]
W90	LR	[0.7736; 0.7950]	[0.5885; 0.6302]	[0.7515; 0.7752]
W90	DT	[0.7650; 0.7884]	[0.6086; 0.6511]	[0.7462; 0.7711]
W90	RF	[0.7798; 0.8021]	[0.5765; 0.6200]	[0.7566; 0.7805]
W90	GB	[0.7839; 0.8056]	[0.6045; 0.6476]	[0.7635; 0.7868]
FULL	LR	[0.9083; 0.9212]	[0.8763; 0.8967]	[0.9082; 0.9211]
FULL	DT	[0.9041; 0.9178]	[0.8854; 0.9059]	[0.9040; 0.9177]
FULL	RF	[0.9100; 0.9230]	[0.8672; 0.8889]	[0.9099; 0.9230]
FULL	GB	[0.9129; 0.9258]	[0.8832; 0.9029]	[0.9129; 0.9257]

4.7. Reproducibility and sensitivity boundaries

The archive records successful serialisation checks for all 16 fitted pipelines: predictions immediately after reloading the saved pipeline were identical to those of the fitted object (the `reload_identical` flag is true). This checks persistence within the recorded environment; it does not establish independent reproduction, cross-version compatibility, or sensitivity to a new split. Gradient Boosting files occupy 144.1–144.8 KiB, versus 9.3–14.0 MiB for Random Forest; the CSV column named `size_kb` uses a divisor of 1024.

The benchmark evaluates one outer student-grouped holdout and fixed model hyperparameters. The three CV repeats assess training-pool repartitioning while holding the outer holdout and the

estimator seed fixed; they do not measure sensitivity to different outer splits or configurations. Hyperparameter search is disabled. The companion derivation script recomputes fold summaries, corrected statistics, Holm adjustments, and the constant-reference metrics from archived outputs without refitting the four classifiers.

5. Discussion

5.1. Interpretation of key findings

Gradient Boosting has the highest mean CV macro-F1 in all four archived windows, but the observed differences from Random Forest are small. Logistic Regression remains competitive: the GB–LR holdout macro-F1 differences range from 0.0044 to 0.0124. The corrected CV and nominal McNemar diagnostics address different metrics and do not establish either universal superiority or model equivalence. With fixed configurations and a single outer split, these results describe this benchmark rather than a ranking known to survive optimisation or cohort changes.

Later observation windows have higher observed macro-F1, but risk recall at the default threshold is similar at W60 and W90. Later cohorts also exclude more early withdrawers. A larger macro-F1, therefore, does not, by itself, identify the best time for educational support: a later alert can leave less time to respond, and population changes confound the comparison. FULL cannot supply an operational early-warning estimate because it includes complete event histories and early-withdrawal records.

The feature-group results and permutation analysis indicate that non-exam assessment scores and the submission-ratio proxy carry substantial predictive information. Module identity and VLE activity are also informative. In W30, the mean score already has the largest reported permutation importance; the data do not support describing that window as relying mainly on VLE activity. Low importance for a demographic feature in one fitted model neither proves that the feature is irrelevant in general nor establishes fairness. Correlated features can substitute for one another, and the ablation does not isolate demographics from course information.

5.2. Educational use and advisor workload

The threshold analysis illustrates the workload associated with detecting more eventual failures or withdrawals. Across W30–W90, target recalls of 0.80–0.90 require flagging 48.3–73.3% of holdout records, with precision 0.541–0.662 and lift 1.23–1.66. For example, W60 at target recall 0.80 flags 2,742 of 5,282 records, includes 978 false alerts, and still misses 441 at-risk records. A contact policy would need to combine alerts for the same student and distinguish assignment-specific difficulties from broader support needs.

A prospective evaluation could use these findings to compare an advisor-facing ranking with simpler policies, such as missing-assignment alerts or a capacity-matched baseline. Staff would review the underlying course context before offering support, and students would need an opportunity to correct inaccurate information. A low score should not determine access to learning opportunities. These are proposed conditions for evaluating a support system; no intervention, advisor workflow, resource-allocation outcome, or improvement in learning was tested here.

An operational threshold must be chosen on validation data using local capacity and the consequences of missed risks and false alerts, then evaluated on a new cohort. The evidence does not establish 60–90 days as an optimal window or recall 0.80–0.90 as a universally appropriate objective. Probability calibration is useful for interpreting scores as risks, but choosing a ranking-based operating point does not require calibration.

5.3. Comparison with prior work

The comparison with prior OULAD work is qualitative because outcome definitions, available features, observation windows, and splitting schemes differ. A multiclass Fail label is not interchangeable with the binary Fail/Withdrawn class used here. Likewise, a dropout classifier and a retrospective pass/fail classifier answer different questions. The present study provides a consistent within-task comparison and an explicit workload calculation, without claiming that its numerical scores outperform results from different tasks.

5.4. Limitations and threats to validity

1. *Retrospective feature availability*: Submission dates do not establish when scores were released. Banked scores are retained, pre-start events are included, and the implementation does not explicitly exclude registrations after a cutoff. A timestamp-availability and enrollment audit is needed before interpreting these cohorts as real-time snapshots.
2. *Changing populations*: Unregistration filtering changes the cohort and class prevalence across windows; FULL additionally includes early withdrawers. No matched-cohort comparison separates added information from selection effects. Fail and Withdrawn are combined even though their support needs may differ.
3. *Limited transportability*: The student-grouped holdout contains the same modules and presentations as training. It tests new students within the observed setting, not transfer to new courses, years, institutions, or contemporary learning platforms. The data describe distance learning at one institution in 2013–2014.
4. *Dependent uncertainty estimates*: Bootstrap intervals and the McNemar calculation operate on records rather than student clusters. Corrected CV inference is approximate, and comparisons with the CV-selected winner are exploratory. Holm adjustment addresses the stated families of comparisons, not these other sources of uncertainty.
5. *No independent threshold validation*: Operating points are selected on holdout outcomes; precision and workload cannot be treated as validated deployment guarantees. Probabilities were not calibrated, and no independent assessment of their calibration is available.
6. *Incomplete comparative baselines*: Fixed hyperparameters, one outer split, no verified sensitivity runs, and no assignment-rule or capacity-matched support-policy baseline limit the model comparison. The ablation has no paired uncertainty estimates and does not isolate demographic contribution.
7. *No causal, fairness, or intervention evaluation*: Global importance describes prediction rather than causal drivers. Subgroup error rates, differential access to digital resources, advisor burden, and effects of support on learning or retention were not evaluated. Low demographic importance does not demonstrate equitable treatment.

5.5. Future work

The next evaluation should first audit registration eligibility and actual score availability times, and revise the submission ratio to align submissions with the assignments due. An unchanged-cohort analysis and forward validation on later presentations would then separate horizon effects from cohort selection and assess transportability. Student-level cluster resampling and paired model differences should replace independent-record uncertainty assumptions. Nested training-only tuning, alternative outer splits, subgroup performance, probability calibration, and simple support-policy baselines would assess robustness. Finally, a prospective study should test whether advisor use improves learning or retention at an acceptable workload; predictive associations alone do not establish that benefit.

6. Conclusions

This retrospective OULAD benchmark compares four fixed classical classifiers using 16 raw features, four observation windows, a student-grouped holdout, and repeated grouped cross-validation. It addresses four questions:

- RQ1 (observation window): Gradient Boosting holdout macro-F1 is 0.7063, 0.7508, 0.7753, and 0.9194 for W30, W60, W90, and FULL. Changing cohorts and full-course information prevent the interpretation of this sequence as a controlled gain from waiting longer. The best intervention time was not measured.
- RQ2 (model comparison): Gradient Boosting has the highest archived CV macro-F1, with small observed differences from Random Forest and Logistic Regression. Exploratory corrected CV comparisons favour GB over RF only in W90 after Holm adjustment; record-level McNemar comparisons do not establish a GB–RF difference. Neither model equivalence nor universal superiority follows. GB has fewer saved artefacts than RF, which may matter when comparing implementations.
- RQ3 (features): Non-exam scores and the submission-ratio proxy are informative predictors alongside module identity and VLE activity. The ablation supports assessment features’ added predictive value over demographic and course information; it does not establish causal effects or the absence of demographic disparities.
- RQ4 (threshold trade-offs): At default threshold 0.5, GB risk recall in W30–W90 is 0.5885–0.6299. Target recalls of 0.80–0.90 flag 48.3–73.3% of records with precision 0.541–0.662. These holdout-derived operating points illustrate workload rather than validated contact policies.

The results motivate further evaluation of advisor support based on assessment and activity signals. Deployment claims require verified real-time feature availability, validation on new cohorts, student-clustered uncertainty estimates, and evidence of educational benefit.

Author contributions

Sofiia V. Velychko: Conceptualisation, Methodology, Software, Validation, Formal Analysis, Investigation, Data Curation, Writing – Original Draft, Writing – Review & Editing, Visualisation. **Olena Y. Chepurna:** Conceptualization, Methodology, Formal Analysis, Investigation, Writing – Review & Editing, Supervision. **Olena H. Fedorenko:** Conceptualization, Methodology, Writing – Review & Editing, Supervision, Project Administration.

Acknowledgments

An OpenAI Codex assistant was used to simulate peer review, assist with reference verification, revise text, and prepare tables and figures in LaTeX. No new model training results were generated during this process. The human authors retain responsibility for validating the analyses, checking citations, and deciding the final content.

Data availability statement

OULAD is public under a Creative Commons Attribution 4.0 license at <https://archive.ics.uci.edu/dataset/349/open+university+learning+analytics+dataset> [13, 14]. The companion computational supplement (<https://github.com/ssemerikov/oulad>) contains `oulad.py`, the archived aggregate result tables and environment metadata, and the scripts used to derive the manuscript’s tables, figures, and multiplicity adjustments.

References

- [1] Ahmed, W., Wani, M.A., Plawiak, P., Meshoul, S., Mahmoud, A. and Hammad, M., 2025. Machine learning-based academic performance prediction with explainability for enhanced decision-making in educational institutions. *Scientific Reports*, 15(1), 26879. Available from: <https://doi.org/10.1038/s41598-025-12353-4>.
- [2] Althibyani, H.A., 2024. Predicting student success in MOOCs: a comprehensive analysis using machine learning models. *PeerJ Computer Science*, 10, e2221. Available from: <https://doi.org/10.7717/peerj-cs.2221>.
- [3] Bouckaert, R.R. and Frank, E., 2004. Evaluating the Replicability of Significance Tests for Comparing Learning Algorithms. In: H. Dai, R. Srikant and C. Zhang, eds. *Advances in Knowledge Discovery and Data Mining*. Berlin, Heidelberg: Springer, *Lecture Notes in Computer Science*, vol. 3056, pp.3–12. Available from: https://doi.org/10.1007/978-3-540-24775-3_3.
- [4] Breiman, L., 2001. Random Forests. *Machine Learning*, 45(1), pp.5–32. Available from: <https://doi.org/10.1023/A:1010933404324>.
- [5] Breiman, L., Friedman, J.H., Olshen, R.A. and Stone, C.J., 1984. *Classification and Regression Trees*. Belmont: Wadsworth. Available from: <https://doi.org/10.1201/9781315139470>.
- [6] Dietterich, T.G., 1998. Approximate Statistical Tests for Comparing Supervised Classification Learning Algorithms. *Neural Computation*, 10(7), pp.1895–1923. Available from: <https://doi.org/10.1162/089976698300017197>.
- [7] Efron, B., 1979. Bootstrap Methods: Another Look at the Jackknife. *The Annals of Statistics*, 7(1), pp.1–26. Available from: <https://doi.org/10.1214/aos/1176344552>.
- [8] Friedman, J.H., 2001. Greedy function approximation: a gradient boosting machine. *The Annals of Statistics*, 29(5), pp.1189–1232. Available from: <https://doi.org/10.1214/aos/1013203451>.
- [9] Herodotou, C., Rienties, B., Verdin, B. and Boroowa, A., 2019. Predictive Learning Analytics ‘At Scale’: Guidelines to Successful Implementation in Higher Education. *Journal of Learning Analytics*, 6(1), pp.85–95. Available from: <https://doi.org/10.18608/jla.2019.61.5>.
- [10] Holm, S., 1979. A Simple Sequentially Rejective Multiple Test Procedure. *Scandinavian Journal of Statistics*, 6(2), pp.65–70. Available from: <https://www.jstor.org/stable/4615733>.
- [11] Hosmer, D.W., Lemeshow, S. and Sturdivant, R.X., 2013. *Applied Logistic Regression*, Wiley Series in Probability and Statistics. 3rd ed. Hoboken: Wiley. Available from: <https://doi.org/10.1002/9781118548387>.
- [12] Kohavi, R., 1995. A Study of Cross-Validation and Bootstrap for Accuracy Estimation and Model Selection. *Proceedings of the 14th International Joint Conference on Artificial Intelligence (IJCAI 1995)*. vol. 2, pp.1137–1143. Available from: <https://www.ijcai.org/Proceedings/95-2/Papers/016.pdf>.
- [13] Kuzilek, J., Hlosta, M. and Zdrahal, Z., 2015. Open University Learning Analytics dataset. Available from: <https://doi.org/10.24432/C5KK69>.
- [14] Kuzilek, J., Hlosta, M. and Zdrahal, Z., 2017. Open university learning analytics dataset. *Scientific data*, 4(1), 170171. Available from: <https://doi.org/10.1038/sdata.2017.171>.
- [15] Lundberg, S.M. and Lee, S., 2017. A Unified Approach to Interpreting Model Predictions. In: I. Guyon, U. von Luxburg, S. Bengio, H.M. Wallach, R. Fergus, S.V.N. Vishwanathan and R. Garnett, eds. *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*. pp.4765–4774. Available from: <https://proceedings.neurips.cc/paper/2017/hash/8a20a8621978632d76c43dfd28b67767-Abstract.html>.
- [16] McNemar, Q., 1947. Note on the Sampling Error of the Difference Between Correlated Proportions or Percentages. *Psychometrika*, 12(2), pp.153–157. Available from: <https://doi.org/10.1007/BF02295996>.
- [17] Nadeau, C. and Bengio, Y., 2003. Inference for the Generalization Error. *Machine Learning*, 52(3), pp.239–281. Available from: <https://doi.org/10.1023/A:1024068626366>.
- [18] Omarbekova, A., Nazyrova, A., Bekmanova, G., Lamasheva, Z., Turebayeva, R. and 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. Available from: <https://doi.org/10.3991/ijim.v20i17.62574>.
- [19] Paul, R., Hussain, S., Kannan, K.D., Mondal, B.K., Baruah, A.K. and Gaftandzhieva, S., 2025. Interpreting the Machine Learning Approaches to Predict CGPA of the University Students. *TEM Journal*, 14(4), pp.3438–3447. Available from: <https://doi.org/10.18421/TEM144-50>.
- [20] Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., VanderPlas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M. and Duchesnay, E., 2011. Scikit-learn: Machine Learning in Python. *The Journal of Machine Learning Research*, 12, pp.2825–2830. Available from: <https://doi.org/10.5555/1953048.2078195>.
- [21] Pronina, O.I. and Pertsev, Y.A., 2026. Comparative analysis of machine learning algorithms in decision support tasks for academic success. *Reporter of the Priazovskyi State Technical University. Section: Technical sciences*, 53(1), pp.61–68. Available from: <https://doi.org/10.31498/2225-6733.53.1.2026.359776>.
- [22] Raj, N.S. and Renumol, V.G., 2022. Early prediction of student engagement in virtual learning environments using machine learning techniques. *E-Learning and Digital Media*, 19(6), pp.537–554. Available from: <https://doi.org/10.1177/20427530221108027>.
- [23] Tkach, V.S., 2025. *Povedinkovi modeli otsiniuvannia uspishnosti studentiv u smart-klasakh [Behavioral models for assessing student success in smart classrooms]*. Master's thesis. National Technical University of Ukraine Igor Sikorsky Kyiv Polytechnic Institute, Kyiv, Ukraine. Available from: <https://ela.kpi.ua/items/21594864-0f65-470b-81a7-59957a67a6cd>.