

ClassCircle: Design and Verification of a Multi-Institution Learning Analytics Platform with Explainable Early Warnings

Aqdus Shaikh Zahoorahmed, Dr Manoj Singh

Department of Computer Science, SIES College of Arts, Commerce and Science, Mumbai, Maharashtra, India
aqdusshaikh357@gmail.com

Abstract—

Early-warning dashboards are now common in higher education, but many published systems either stop at a bare risk label or remain prototypes that never meet the practical demands of a real college: separate institutions, staff roles, secure sign-in and data-protection law. This paper presents ClassCircle, a web platform that grew out of the author's earlier browser prototype and now serves several institutions from a single Rocky Linux virtual machine. Each institution has its own SQLite database behind one sign-in page, and the server alone decides what each of five roles may see. Risk is estimated with an L2-regularised logistic regression over seven academic and engagement indicators. Each 0–100 score is shown with its exact per-feature contributions and with rule-based next steps, and every summary number on a dashboard opens the list of students behind it. With no real student data available, the evaluation covers only measured results: 189 back-end tests, 85 browser checks, 25 acceptance scenarios and 34 attack checks passed; on a synthetic cohort of 120 students the model reached a mean AUC of 0.934 over 100 stratified splits, against 0.849 for the rules alone; and a burst of 120 simulated students writing at the same moment completed 360 requests without an error on a two-core test machine. The paper makes no claim that any student improved. Instead, it fixes in advance how improvement and possible harms will be measured in a staged pilot, and it maps the platform's controls to India's Digital Personal Data Protection framework.

Index Terms—Learning analytics, early warning systems, explainable models, logistic regression, multi-tenant architecture, educational dashboards, data protection, software verification.

I. INTRODUCTION

Most colleges already hold the raw material for an early-warning system. Attendance registers, internal assessment marks, assignment records and, more and more often, logs from a learning management system (LMS) are collected every week. What is usually missing is one place where a teacher can see which students may need a conversation this week, and on what grounds. Learning analytics dashboards were proposed to fill this gap more than a decade ago [1], and the volume of work on them has grown steadily since [2].

The evidence that such dashboards change outcomes is much thinner than that volume suggests. A recent review of 38 impact studies found that about three quarters of the reported effects on achievement were negligible or small, and it found no well-powered randomised trial [3]. Early-alert programmes have a similar history. A retention programme that looked effective in simple comparisons showed no effect once mixed-effects models accounted for who had been flagged [4], and a randomised trial of proactive advising across 11 universities found the same graduation rate in both arms [5]. A common reading of this record is that a flag does nothing by itself. It helps only when a person acts on it, and a person is more likely to act sensibly when the reason for the flag is visible.

The author's earlier work [6] explored the second half of that reading. It was a browser-only prototype that trained a logistic regression on generated records, turned the predicted probability into a 0–100 score, and showed the linear contribution of each indicator next to a short list of rule-based

suggestions. The prototype demonstrated the idea, but it could not be put in front of a real college. It had no user accounts, no roles, no way of keeping one institution's records apart from another's, no audit trail and no route for the spreadsheets that colleges actually keep. This paper reports the work needed to close that gap, and what could and could not be verified without real student data.

The paper makes four contributions:

- 1) An architecture for a small multi-institution platform, ClassCircle, in which every institution has its own database file, all users share one sign-in page, and the server enforces five roles, including an operator role that cannot open student records (Section IV).
- 2) An early-warning pipeline that keeps the prototype's model unchanged but ties each score to its exact additive explanation, to editable next-step rules, and to drill-down links from every dashboard number to the students behind it (Section V).
- 3) A verification record taken from the released code: automated tests, browser and acceptance checks, attack checks, a model comparison on synthetic data with a known structure, and a simultaneous-write load test (Section VII).
- 4) A privacy mapping and a staged pilot protocol that state, before any student is involved, which legal duties will apply and how improvement will be measured (Sections VIII and IX).

One limit should be stated plainly at the outset. ClassCircle has not yet been used with real students, so this paper reports

no improvement in student performance. All model figures come from a synthetic cohort produced by a generator whose rule is known, and they are reported only to show that the pipeline behaves as designed. How many students improve, if any, is the question that the pilot in Section IX is designed to answer.

II. RELATED WORK

A. Learning Analytics Dashboards

Reviews of dashboard research tell a consistent story. Verbert *et al.* [1] compared a set of dashboard applications and noted that evaluations rarely tested whether a dashboard changed what learners or teachers did. Schwendimann *et al.* [2] screened 346 papers, kept 55, and found that most described exploratory or proof-of-concept systems. Bodily and Verbert [7] asked for experiments on behaviour and achievement to add to a small body of evidence, and a later review found that only 31% of the dashboards it covered let users interact with the data [8]. Matcha *et al.* [9] added that dashboards were seldom grounded in learning theory and seldom told students what to do next.

The question of impact has now been asked directly. Kaliisa *et al.* [3] reviewed 38 studies published between 2015 and 2023. Across 34 achievement studies, 76.5% of the effects were negligible or small, and several of the larger effects came from comparisons of users with non-users. Ferguson and Clow [10] had reached a similar conclusion earlier from a public evidence hub. For teacher-facing dashboards, which is where ClassCircle sits, Kaliisa, Jivet and Prinsloo [11] reviewed 50 studies and found that most systems raised awareness but offered few insights a teacher could act on. Only one of those studies treated privacy as a design requirement.

B. Early Warning and At-Risk Prediction

Purdue's Course Signals [12] is the best-known early-warning system. It combined grades, past academic history, demographic data and LMS effort into a traffic-light signal that students saw together with a message from their instructor. Its multi-year retention claims were later questioned on the grounds that students who stay enrolled take more courses and therefore collect more signals [13]. Dawson *et al.* [4] followed 11,160 enrolments in a retention programme and found that a small positive association disappeared under mixed-effects modelling. The largest documented teacher-facing deployment, the Open University's OU Analyse, gives the most encouraging evidence: across 559 teachers and more than 14,000 students, students of teachers who used the weekly predictions moderately performed better once prior attainment was controlled for [14]. A companion study, however, found that teachers rated the tool useful yet seldom used it [15]. At the far end, the MAAPS trial randomised 10,489 students at 11 universities to proactive, technology-supported advising, and graduation was 70% in both arms [5].

Reviews of performance prediction show which inputs and reporting habits matter. Namoun and Alshanqiti [16] found in-semester assessment grades and online activity to be the most useful predictors, yet 61% of the 62 studies they read used no baseline and 34% named no evaluation metric. Albreiki *et al.* [17] stressed that prediction is useful only when it is tied to an intervention plan. Two reviews of predictive dashboards found

that most stop at identifying at-risk students and do not interpret the model for the person who has to act [18], [19].

C. Interpretable Models and Fairness

Rudin [20] argued that high-stakes decisions should rely on models that are interpretable by design rather than on after-the-fact explanations of black boxes. Education data support this position to a degree. On 331,254 community-college students, Bird *et al.* [21] found that with a full feature set the c-statistic ranged only from 0.884 for ordinary least squares to 0.903 for XGBoost, but with 14 basic predictors the gap widened to roughly 0.81–0.82 against 0.85–0.86. They also found that the students ranked as highest risk changed noticeably from one model to another. Jayaprakash *et al.* [22] found logistic regression the best of four classifiers in an open early-alert project, but with a precision near 33%, which means about two in three flags were false alarms; when the model was moved to one of the partner colleges, recall fell to about 52%. Kung and Yu [23] reported that interpretable models gave up neither accuracy nor fairness when predicting college success.

Explanations bring risks of their own. Khosravi *et al.* [24] warn that an incomplete explanation can create a false sense of understanding, and Alamri and Alharbi [25] call for studies that measure accuracy and explainability together. In a field experiment, Kizilcec [26] found that a moderate amount of transparency preserved learners' trust when a grade was unexpected, while too much information eroded it. On bias, Baker and Hawn [27] note that most of the evidence comes from the United States, and Gardner *et al.* [28] proposed comparing groups by the area between their ROC curves.

D. Privacy, Governance and Multi-Tenancy

Slade and Prinsloo [29] framed learning analytics as a moral practice and named consent, interpretation and data management as core concerns. The DELICATE checklist [30] and the SHEILA framework [31] turn such concerns into institutional steps; SHEILA was built from interviews with senior managers at 51 institutions. In India, the Digital Personal Data Protection (DPDP) Act, 2023 [32] and the DPDP Rules, 2025 [33] now set the legal baseline, and Section VIII returns to them.

Software-as-a-service design offers named patterns for keeping customers apart. Krebs *et al.* [34] describe multi-tenancy as tenants sharing one application instance while staying isolated in performance and data. Microsoft's guidance rates one database per tenant as high in isolation and low in development complexity [35], and Amazon calls a shared application tier over dedicated data stores a bridge model [36]. SQLite's documentation suggests that sites with fewer than 100,000 hits a day should work well with SQLite and mentions a separate database per user as a way to improve concurrency [37]; in write-ahead-log (WAL) mode, readers and writers do not block each other, provided all processes run on the same host [38]. Broken access control remains the first risk in the OWASP list of web-application risks [39].

E. Positioning of This Work

Table I places the present system beside the earlier prototype. The model and its explanation are unchanged; what is new is everything a college needs before an early-warning tool can be used responsibly. To the author's knowledge, no published learning analytics system combines exact per-feature

explanations, drill-down from every summary figure, one database per institution on a single small server, and a control-by-control mapping to India's DPDP framework. The contribution lies in this combination and in its verification, not in any single component.

TABLE I
EARLIER PROTOTYPE AND CLASSCIRCLE COMPARED

Aspect	Earlier prototype [6]	ClassCircle (this paper)
Where the model runs	In the browser	On the server, separately for each institution
Users and roles	One anonymous user	Five roles, checked by the server on every request
Institutions	One generated cohort	Many; one database file each
Data entry	Generated records	Forms and seven kinds of CSV import
Explanation	Linear contributions	Same, plus rule-based next steps and drill-down lists
Sign-in	None	script hashes, lockout, one-time codes for staff
Audit	None	Append-only log per institution and for the platform
Evaluation support	Metric views	Metric views, SUS survey, outcome recording
Deployment	Front-end web app only	Nginx, Gunicorn and systemd on Rocky Linux with SELinux

III. DESIGN GOALS

Seven goals guided the design, and later sections refer back to them.

G1. Every score carries its reasons. No risk number appears without the indicators that produced it.

G2. Every summary number is a door. A count on a dashboard opens the list it counts, so a principal can move from "28 students need attention" to the 28 names.

G3. Isolation by construction. One institution's records live in a different file from another's, and the platform operator has no route to student-level data.

G4. The server decides. The browser may hide a button, but every permission check happens on the server.

G5. Small footprint. The whole system runs on one virtual machine that a college's own IT staff can manage, with no paid cloud service and no GPU.

G6. Honest fallbacks. When there are too few records to train the model, the system says so and uses transparent rules instead.

G7. Ready for evaluation. The data a pilot will need, such as usability ratings and recorded outcomes, can be collected inside the application.

IV. SYSTEM ARCHITECTURE

Fig. 1 shows the deployed structure. A React and TypeScript single-page application talks to the server only over HTTPS. Nginx terminates TLS, adds security headers, limits the rate of sign-in attempts from each address and serves the static files. It passes application requests to Gunicorn on the loopback interface, which runs a Flask application as a systemd service with two worker processes of four threads each. The application is a modular monolith: sign-in and roles, tenancy, academic data, early warning, workflows, and audit and notification code sit in separate modules but in one process. Keeping a single

process put deployment and debugging within the reach of one developer, in line with G5.

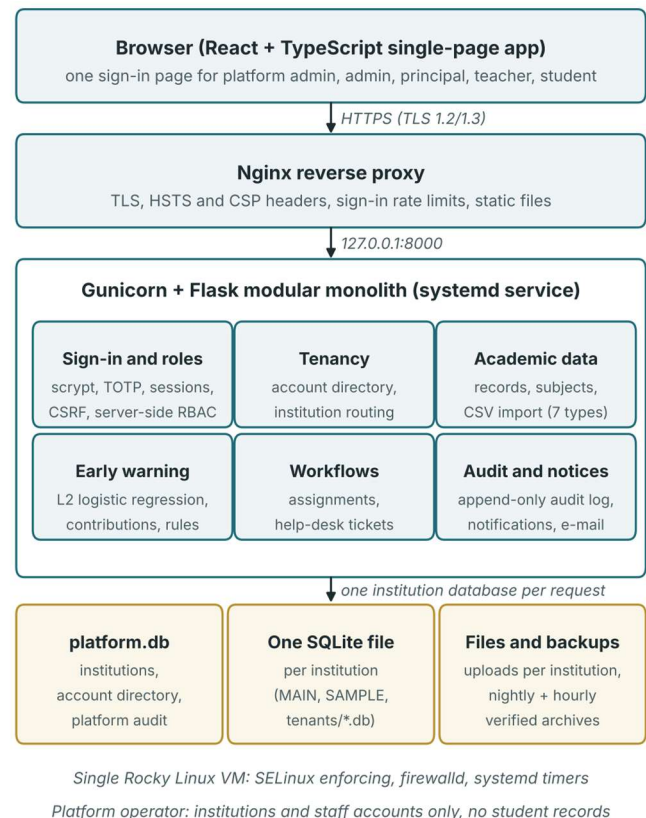


Fig. 1. Deployed architecture of ClassCircle. Every request is served from the database of exactly one institution; the platform database holds only the list of institutions, the account directory and the platform's own audit log.

A. One Sign-In Page and Server-Side Roles

All users, from students to the platform operator, sign in on the same page (Fig. 2). The username is first looked up in a directory kept in the platform database, which records the institution that owns it; the password is then checked against a salted script hash stored in that institution's database. An unknown username and a wrong password produce the same message, so the page cannot be used to discover accounts. Five failed attempts lock an account for 15 minutes, and wrong two-step codes count towards the same limit. The session cookie is bound to the institution, and a test confirms that a cookie carried over to another institution is refused.

ClassCircle
Every student in the circle

Sign in
Students, teachers, principals and administrators all sign in here. Your dashboard opens automatically.

Invalid username or password.

Username
nobody-at-all

Password

Sign in

[Forgot your password?](#)

Fig. 2. The single sign-in page of the sample college. The same message is shown for an unknown username and for a wrong password.

Administrators and principals must also enter a time-based one-time code from a standard authenticator app (six digits, 30-second steps, one step of clock drift allowed). A code that has already been accepted is refused, which meets the single-use requirement of NIST SP 800-63B-4; the same guidance notes that such codes are not phishing-resistant, so the combination should be read as an AAL2-level control and not as a stronger one [40]. Ten single-use recovery codes are issued at enrolment, and only their hashes are stored.

Roles are read from the database on every request, and a role sent by the browser is ignored. Permissions are checked in two layers. A matrix decides whether a role may use a capability at all, and an object-scope check decides whether this user may touch this particular student, assignment or ticket. The second layer addresses insecure direct object references, a common form of broken access control [39]. Table II summarises what each role may do.

TABLE II
ROLES AND WHAT THE SERVER ALLOWS

Role	Allowed	Refused
Platform operator	Create institutions and their staff accounts, import CSV files on request, reset staff passwords; every action is logged in both audit logs	Opening any student record, dashboard, mark, assignment or ticket; resetting student passwords
Administrator	All accounts, CSV import, rules and thresholds, institution dashboards, audit log	Anything in another institution
Principal	Institution dashboards, students and subjects, CSV import, rules and thresholds, resetting teacher and student accounts	Managing administrator accounts; reading the audit log
Teacher	Students and marks of assigned classes, assignments, help-desk work	Students outside assigned classes; settings
Student	Own dashboard and explanation, assignment submission, help-desk requests	Any other student's data

B. One Database per Institution

Each institution is a separate SQLite file. The platform database holds the list of institutions, the account directory and the platform's audit log; it holds no marks or attendance. A sample institution with invented data lives in its own file, so demonstrations never touch real records, and a sample username can never block a real person's username. This arrangement matches the database-per-tenant pattern in Microsoft's guidance [35] and what Amazon calls a bridge model [36]. It gives three practical benefits. A faulty query cannot return another institution's rows, because those rows are not in the open file. One institution can be backed up, restored or erased as a unit. And a burst of writes at one college does not queue behind writes at another, since SQLite allows one writer per database file at a time [37], [38]. The costs are real as well. Schema changes must be applied to every file, backups must be scripted (ClassCircle takes nightly and hourly archives that include every institution and verifies them), and all files must stay on the same host as the application.

C. Getting Data In

Colleges keep most of their records in spreadsheets, so data enter ClassCircle either through forms or through seven kinds of CSV import: students, subjects, subject results, semester

records, attendance, teaching assignments and accounts. Each file is validated before it is applied, unknown fields are refused, and the import history is kept. The account import has no password column; it generates a temporary password for each new account, shows the list once, and forces a new password at first sign-in. Templates for all seven kinds ship with the release.

D. Workflows Around the Score

An early warning is only useful if it appears where teachers already work. ClassCircle therefore also carries assignments, which teachers create, students submit and teachers review; a help desk for student requests with categories, priorities and response targets; in-app notifications; and e-mail. These features are ordinary, but they give staff a reason to open the platform every day, and the OU Analyse experience suggests that regular use is where the difficulty lies [15].

V. EXPLAINABLE EARLY-WARNING PIPELINE

A. Indicators and Outcome Label

The model uses the seven indicators of the earlier prototype, taken from each student's latest semester: attendance, assignment completion, assessment average, LMS engagement, number of late submissions, weekly study hours and number of prior failures. No demographic field is used. The training label is the recorded outcome (at risk or not) when the institution has recorded one. Otherwise a fallback label marks a student as at risk when the GPA is below 2.0; GPA itself is never an input, so the fallback cannot leak the label into the features. The model is trained only when at least 20 students have complete records and each class has at least four members. Below that, the rule-based indicator of Section V-D is used, and the dashboard says so (G6).

B. Model and Score

For student i and indicator j , each value is standardised with the mean and standard deviation of the training partition,

$$z_{ij} = (x_{ij} - \mu_j) / \sigma_j \quad (1)$$

and the model is a logistic regression on the standardised values:

$$p_i = 1 / [1 + \exp(-\eta_i)], \quad \eta_i = b + \sum_j w_j z_{ij} \quad (2)$$

The weights w_j and the bias b minimise the L2-regularised log-loss

$$L = -(1/n) \sum_i \ell_i + (\lambda/2) \|w\|^2 \quad (3)$$

where $\ell_i = y_i \ln p_i + (1 - y_i) \ln(1 - p_i)$ is the log-likelihood of student i with label y_i . They are found by batch gradient descent over 1,500 iterations with a learning rate of 0.2 and $\lambda = 0.01$, after an 80/20 split stratified by class with a fixed seed. These are the prototype's settings [6]; they were not tuned on the synthetic data. The probability then becomes a score,

$$s_i = \text{round}(100 p_i) \quad (4)$$

which falls in the High band when $s_i \geq 60$, in the Medium band when $35 \leq s_i < 60$ and in the Low band otherwise. Both cut-offs are settings that an administrator or principal can change. They are design choices rather than calibrated thresholds, a point taken up in Section X.

C. Exact Explanation of Each Score

Because the model is linear in the standardised indicators, the log-odds split exactly into one term per indicator:

$$\varphi_{ij} = w_j z_{ij}, \quad \eta_i = b + \sum_j \varphi_{ij} \quad (5)$$

A positive φ_{ij} pushes student i towards the at-risk class and a negative one pulls the student away from it. Unlike sampling-based explainers, (5) is not an approximation: the terms add up to the model's own log-odds, which Section VII-C checks numerically. The student's home page (Fig. 3) turns the largest positive terms into short sentences, each naming the indicator, the student's value, its position relative to the training average and the size of the term. For a student in the Low band, the sentences instead name the indicators that keep the risk down. Since too much detail can reduce trust [26], at most three reasons are listed, under the heading "guidance, not a judgement". A global view averages $|\varphi_{ij}|$ over all students to show which indicators matter most to the current model.

D. Rules for Next Steps

The second half of each explanation is a set of next steps drawn from eight rules (Table III). Each rule tests one observable condition, and when it fires the dashboard shows the exact condition, for example "Attendance 68% is below 75%", together with a suggested action. Rules are sorted by priority. They are prompts for a conversation, not decisions: low attendance can come from illness, a job, family duties or a timetable clash, and only a person who talks to the student can tell which. The model and the rules are kept apart on purpose. The model estimates risk, while the rules carry the institution's own support policy, which an administrator or principal can change without retraining anything. When the model cannot be trained, the number of rules that fire gives a rule-based attention level: High for three or more, Medium for one or two and Low for none.

Academic attention

Where should you focus? (guidance, not a judgement)

High Logistic Regression (paper method)

- Assessment average 29.3% is below the training average and raises risk (+4.86)
- Prior failures 3 is above the training average and raises risk (+1.72)
- Assignment completion 77% is below the training average and raises risk (+0.26)

Suggested next steps

- Targeted assessment tutoring** — Review weak assessment areas and consider focused tutoring.
- Resit support pathway** — Review previously failed subjects and available academic support.
- Structured assignment planning** — Break upcoming assignments into smaller scheduled tasks.

Fig. 3. Explanation on a synthetic student's home page. The reasons are the

largest positive terms of (5); the suggested next steps come from the rules in Table III.

TABLE III
NEXT-STEP RULES (DEFAULT THRESHOLDS, EDITABLE PER INSTITUTION)

Condition	Suggested next step	Priority
Attendance below 75%	Attendance recovery plan	High
Assessment average below 55%	Targeted assessment tutoring	High
One or more prior failures	Resit support pathway	High
Trend is Declining	Early-warning tutor intervention	High
Assignment completion below 80%	Structured assignment planning	Medium
LMS engagement below 50%	Increase learning-platform engagement	Medium
Three or more late submissions	Time-management coaching	Medium
Study hours below 7 per week	Guided study schedule	Medium

E. Trends and Drill-Down

A student's trend is Improving or Declining when the semester marks move by at least five points from the previous semester, and Stable otherwise; a Declining trend is itself one of the rules. Every count on a dashboard links to the filtered list behind it (G2). Fig. 4 shows the path in the sample college: the principal's "Requiring attention" card shows 28 and opens a student directory filtered to the same 28 students. An automated test checks, for the principal, teacher and administrator views, that every count equals the length of the list it opens, and that students cannot use the filters to reach other students.

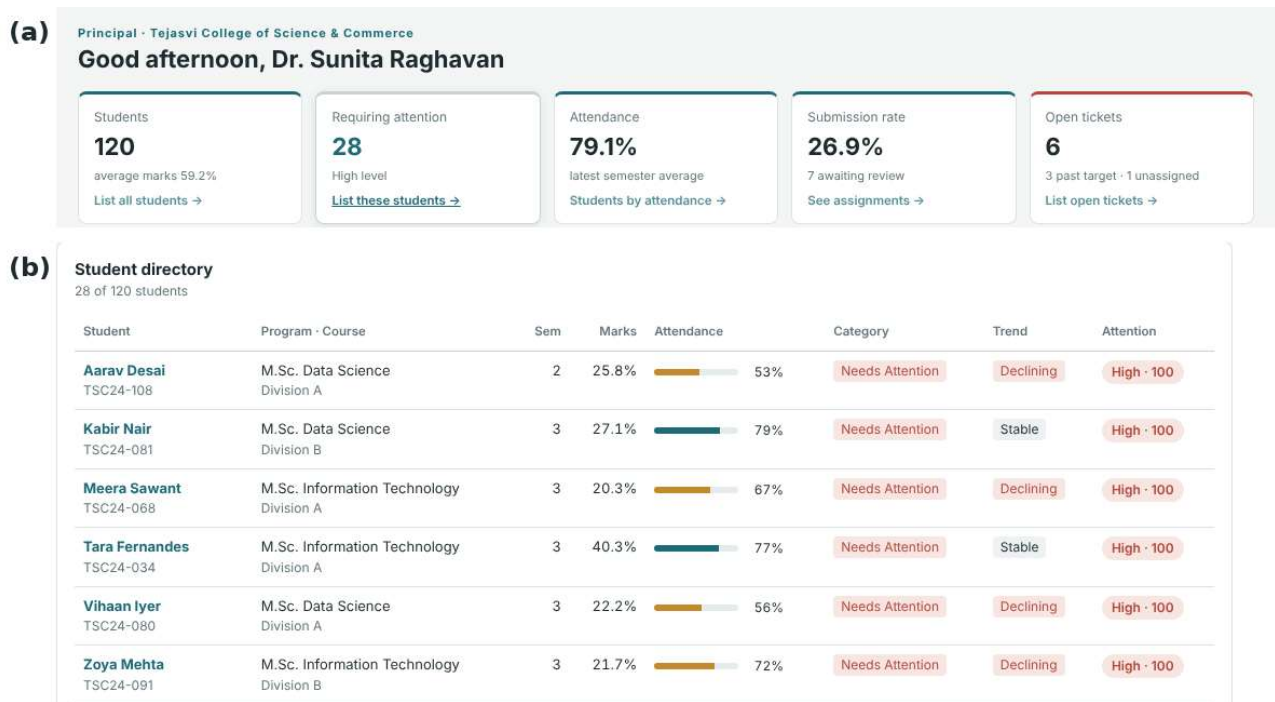


Fig. 4. Drill-down in the sample college (synthetic data). (a) Part of the principal's dashboard; each card links to a list. (b) The list opened from "Requiring attention", showing the same 28 students with marks, attendance, trend and attention level (first rows shown).

VI. IMPLEMENTATION AND DEPLOYMENT

The back end is written in Python with Flask and NumPy and runs under Gunicorn, with SQLite in WAL mode. The model, its evaluation metrics and the rules together take fewer than 300 lines of Python and use no machine-learning library, so every step can be read and checked. The front end uses React 19 and TypeScript and is bundled with esbuild into static files that Nginx serves. Counted by the Bandit analyser, the Python code is about 7,200 lines, excluding a vendored QR-code library used for authenticator enrolment; the TypeScript front end is about 5,400 lines.

The target system is Rocky Linux 10 with SELinux in enforcing mode and firewalld. systemd runs the application, a nightly backup, an hourly database snapshot and a mail queue; nightly archives are kept for 14 days and hourly ones for 48 hours. On a private network with no public domain, the release creates a small certificate authority that is restricted by name constraints to the platform's own host name, so a browser that trusts it will not accept certificates it might issue for other sites. A built-in self-check reports 53 deployment conditions, such as file permissions, certificate validity, backup age and SELinux state, as pass, warning or failure. Finally, a pilot module collects the ten-item System Usability Scale (SUS) [41], reports results by role without names, and gives teachers and principals one screen on which to record the known outcome of each student in a class. These are the inputs that Stages 1 and 2 of Section IX will need (G7).

VII. VERIFICATION AND MEASURED RESULTS

A. Test Environment

The measurements were taken in the build workspace: an Ubuntu 24.04 container with two virtual CPU cores and about 7 GB of memory, running Python 3.13, Flask 3.1.3, Gunicorn 26.2.2, NumPy 2.5.3 and Nginx 1.31.6 with the release's own configuration files. A local host name pointed to the container and the browser trusted the release's certificate authority, so no certificate errors were ignored. Rocky Linux could not be booted in that workspace; the author separately ran the back-end test suite on a Rocky Linux 10 virtual machine. The model and load scripts ship with the release, so every number in this section can be regenerated.

B. Functional and Security Checks

Table IV lists the test suites and their results for release 4.1.1. The attack checks are scripted requests sent through Nginx that try what a curious or hostile user might try: sending a different role with a request, carrying a session cookie over to another institution, calling platform endpoints that would expose student data, and requesting files through path traversal. All 34 were refused as intended. Several things were not tested, and the table says so instead of leaving them out.

C. Model Behaviour on Synthetic Data

The sample college comes from a seeded generator. Each synthetic student has two hidden traits, ability and engagement, drawn from a standard normal distribution, and the seven indicators are noisy linear functions of these traits. A student is labelled at risk when a noisy weighted sum of the two traits passes a fixed cut-off. The generator gives 120 students with

297 semester records, of whom 35 (29.2%) are at risk in their latest semester. Because the indicators are linear in the hidden traits by construction, these data favour a linear model. The results below therefore test that the pipeline is correct, not that it will work on real students.

TABLE IV
VERIFICATION RESULTS FOR RELEASE 4.1.1

Suite	What it covers	Result
Back-end automated tests (18 files)	Roles, isolation between institutions, sign-in, one-time codes, CSV import, drill-down counts, backups, model and rules	189/189 passed; also 189/189 on the author's Rocky Linux 10 VM
Type check and production build	TypeScript front end	Passed
Browser checks (Chromium; 1366×900 and 390×844)	Sign-in, landing page per role, drill-down links, phone layout	85/85 passed
Acceptance scenarios via Nginx and trusted HTTPS	20 planned scenarios and 5 added ones	25/25 passed
Attack checks via Nginx	Role tampering, cookie reuse across institutions, platform limits, path traversal	34/34 passed
Static analysis (Bandit)	About 7,200 lines of Python	0 issues
Deployment self-check	53 conditions	43 pass, 5 warnings, 5 skipped, 0 failures
Not tested	Rocky Linux commands in the workspace; browsers other than Chromium; full screen-reader audit; many institutions under load; network scanners re-run for this release	—

On the default seeded split (96 training and 24 test students), the model reached an AUC of 0.882 with 5 true positives, 2 false positives, 2 false negatives and 15 true negatives, giving an accuracy of 0.833 and a precision, recall and F1 of 0.714 each. A test set of 24 students is too small for comparing methods, so Table V repeats the stratified 80/20 split with 100 different seeds. Logistic regression reached a mean AUC of 0.934, against 0.849 when students were ranked by the share of rules that fired. A random forest of 200 trees, included as the tree-based reference that reviews ask for [16], reached 0.936; the difference from logistic regression is far smaller than one standard deviation, as the linear generator would lead one to expect. The rules found slightly more of the at-risk students (recall 0.729 against 0.674) but at a cost in precision (0.661 against 0.811), so more of their flags were false alarms.

TABLE V
MEAN (SD) OVER 100 STRATIFIED 80/20 SPLITS, SYNTHETIC COHORT OF 120 STUDENTS

Metric	Logistic regression	Rules only ^a	Random forest ^b	Majority class
AUC	0.934 (0.041)	0.849 (0.080)	0.936 (0.047)	0.500
Precision	0.811 (0.146)	0.661 (0.129)	0.868 (0.144)	— ^c
Recall	0.674 (0.166)	0.729 (0.147)	0.701 (0.156)	0.000
F1	0.720 (0.121)	0.684 (0.114)	0.758 (0.108)	— ^c
Accuracy	0.851 (0.060)	0.805 (0.072)	0.873 (0.055)	0.708

^aScore = share of the 8 rules that fired; flagged at 3 or more rules.

^bscikit-learn 1.9.1, 200 trees, default settings, raw indicators.

^cUndefined: no student is flagged.

Two further checks concern the explanation. Over all 120 students, the largest gap between the model's log-odds and the bias plus the sum of the contributions in (5) was 8.9×10^{-16} , which is floating-point rounding, and every displayed score could be rebuilt exactly from the contributions. The fitted weights also show why contributions must be read with care. Assessment average had by far the largest weight (−2.303) and the largest mean absolute contribution (1.880), followed by prior failures (+0.820) and study hours (−0.629). Late submissions, however, received a negative weight (−0.408) even though their correlation with the at-risk label was positive (+0.372). In the synthetic data, late submissions move closely with study hours ($r = -0.69$) and attendance ($r = -0.62$); once those two are in the model, late submissions act as a small correction rather than as a risk factor. Contributions therefore describe how the model uses an indicator, not what causes a student's risk, and a teacher reading them should treat them that way.

Training took a median of 20.3 ms, and a full recomputation for the sample college, covering training, scoring, rules and summary figures, took 34.2 ms. With the deployed model, 28 students fall in the High band, 14 in Medium and 78 in Low. The generator also produces trends: in the latest semester 12 students are Improving, 72 Stable and 36 Declining. These counts describe the generator, not real progress, and they must not be read as evidence that any student improved.

D. Load on One Institution

Two loads were sent through Nginx over HTTPS (Table VI). In the read test, one principal requested the institution dashboard and the student list 200 times each, one request after another. In the write test, N students who were already signed in waited at a barrier and then, at the same moment, each opened a help-desk ticket, added a comment to it and reloaded their home page. The sign-ins were done beforehand and directly against Unicorn, because many sign-ins from one address would trip the per-address limit in Nginx. With 60 students, 180 requests finished in 1.14 s; with 120 students, 360 requests finished in 1.99 s. Neither run produced an error, and the slowest single request took 1.39 s. Median read times stayed below 10 ms in both runs (9.5 and 9.6 ms for the dashboard).

Write times grow with N because every write in one institution passes through SQLite's single writer. This is the expected cost of the design, and it would be spread across files if the students belonged to different institutions. The test did not cover many institutions at once, longer sustained load or slower disks.

TABLE VI
RESPONSE TIMES THROUGH NGINX ON TWO CPU CORES (MS)

Request	n	Median	p95	Max
<i>Read path: one principal, sequential</i>				
Principal dashboard	200	9.5	14.1	59.3
Student list	200	6.3	9.4	15.9
<i>60 students at once: 180 requests, 0 errors, 1.14 s</i>				
Open ticket	60	302.8	473.9	682.1
Add comment	60	235.3	418.9	465.3
Reload home page	60	238.9	415.7	428.1
<i>120 students at once: 360 requests, 0 errors, 1.99 s</i>				
Open ticket	120	556.7	793.1	1388.4
Add comment	120	443.1	518.4	690.9
Reload home page	120	492.7	595.9	643.4

E. What These Results Do Not Show

None of the numbers above says anything about real students. The model results come from data built to suit a linear model, the load test used one institution on a two-core machine, and the security checks cover the attacks that were scripted, not every possible one. Above all, the paper has no measure of how many students improved, because no student has yet used the system. The next two sections describe what must be in place before that question can be asked.

VIII. PRIVACY, ETHICS AND COMPLIANCE

The DPDP Act, 2023 [32] requires a notice with every request for consent (s. 5); consent that is free, specific, informed, unconditional and unambiguous and limited to the data needed for the purpose (s. 6); reasonable security safeguards, breach reporting and erasure once the purpose has been served (s. 8); and verifiable parental consent for anyone under 18, together with a bar on tracking or behavioural monitoring of children (s. 9). The DPDP Rules, 2025 were notified on 13 November 2025 [33]. Their substantive duties, including Rule 6 on security safeguards, which covers access control, logging and monitoring and the keeping of logs for one year, will apply from 13 May 2027. The date should be checked again before any pilot, as the timeline may still change.

In a multi-institution platform, each college would most likely act as the Data Fiduciary and the platform operator as its Data Processor under a contract. This is the author's reading, not legal advice. Table VII maps the duties most relevant to ClassCircle onto the controls that exist today and the work that remains. Two gaps stand out: the database files and backups are not encrypted at rest, and there is no written breach-response procedure. Both must be closed before real data are loaded.

TABLE VII
DPDP DUTIES MAPPED TO CLASSCIRCLE (AUTHOR'S READING)

Duty	Control in place	Still needed
Notice and consent (s. 5, s. 6)	Only the seven indicators, identifiers and course data are stored; no demographic fields	College notice in the students' languages; consent records for the pilot
Security safeguards (s. 8(5); Rule 6)	Database per institution; server-side roles and object scope; script, lockout and one-time codes; TLS; SELinux; root-only backups	Encryption of database files and backups at rest
Logs and one-year retention (Rule 6)	Append-only audit log per institution and for the platform	A written retention period of at least one year and a review routine
Breach notice (s. 8(6); Rule 7)	Audit trail to reconstruct events	A procedure naming who informs the Board and the students
Erasure (s. 8(7))	Student records can be deleted; an institution's file can be removed as a unit; backups expire	A retention schedule agreed with each college
Children (s. 9; Rule 10)	Indicators limited to academic records; staff, not the system, decide any action	Verifiable parental consent for students under 18 unless an exemption is confirmed
Operator as processor	Operator role cannot open student records; its actions are logged twice	A written contract between each college and the operator

The law sets a floor, and the ethics frameworks add points it does not cover. In the spirit of Slade and Prinsloo [29] and DELICATE [30], ClassCircle shows students their own view in non-judgemental language (Fig. 3), keeps the invented sample college in a separate file that never mixes with real records, and leaves every action to a member of staff. Following SHEILA [31], a college should adopt a written learning analytics policy before a pilot begins, covering who may see scores, how students can question them and when they are deleted. Because the model uses no demographic data, group gaps cannot be seen from the model alone. The pilot plan therefore includes a per-group check on attributes that the college already holds, analysed separately from the model [27], [28].

IX. PLANNED PILOT EVALUATION

The question most readers will ask, how many students improve, cannot be answered from the present data. This section therefore fixes in advance how it will be answered. Table VIII lists four stages, each with a condition for moving on.

Stage 0 needs no college data. The model will be trained and tested on the public Open University Learning Analytics Dataset (OULAD), which holds 32,593 students in 22 course presentations with assessment results and daily LMS click summaries [42]. Assessment scores, submission dates and clicks map onto three of ClassCircle's indicators; the others will be left out rather than invented. The model will be compared with a marks-only rule and with a tree-based model, and precision and recall will be reported for each band, not only overall, so that the 60 and 35 cut-offs can be judged.

TABLE VIII
STAGED PILOT PLAN

Stage	Who and what	Main measures	Condition to proceed
0. Offline check	Public OULAD data [42]; no college data	AUC; precision and recall per band; calibration; per-group gaps [28]	Model beats a marks-only rule
1. Usability	About 8–12 teachers and principals; set tasks on the sample college	SUS [41]; usefulness and ease-of-use items [43]	Mean SUS of 70 or more [44]
2. One term	Dashboard sections against comparison sections in one college	Band change of flagged students; marks as Hedges' g with 95% CI [45]; harms; use	Primary outcome positive and no rise in harms
3. Wider use	Several colleges	Matched or regression-discontinuity designs	Governance approval at each college

Stage 1 asks teachers and principals to complete set tasks in the sample college, for example finding the students in one class who need attention and naming the main reason for one of them, and then to fill in the SUS. A mean SUS of 70 or more, commonly treated as acceptable [44], is the condition for moving to Stage 2. Short questionnaires on perceived usefulness and ease of use, adapted from the Technology Acceptance Model [43] as Rienties *et al.* did for teachers [46], will be added.

Stage 2 is a one-term comparison within one college. Whole sections, not individual students, will be assigned either to use the dashboard or to continue as usual, as in the design of Jayaprakash *et al.* [22]; where possible, the same teacher will teach one section of each kind. Scores will be computed for every section but shown only in the dashboard sections. Improvement is defined before any data are seen:

- **Primary outcome:** among students in the High or Medium band in the fourth week of term, the share whose band is lower at the end of term, compared between dashboard and comparison sections.
- **Secondary outcomes:** end-of-term marks, change in attendance and pass rate, each reported as Hedges' g with a 95% confidence interval [45].
- **Harms:** withdrawal rate, students' reported stress about being flagged, and teacher workload, measured as flags per teacher per week and the share of flags that a teacher marks as not needed.
- **Use:** whether teachers opened a flagged student's page and recorded an action, because having access to a dashboard is not the same as using it [3].

The analysis will use a mixed-effects model with section as a random effect, since a simple comparison can show an effect that vanishes once the grouping is modelled [4]. All outcomes will be reported, including null results. Only a positive primary outcome without a rise in harms would support the statement that students improved, and even then a single term in one college would not justify a causal claim. Stage 3, across several colleges, would allow matched designs or a regression-discontinuity design at the High cut-off. No stage involving real

students will begin without the college's approval, a notice and consent process that meets Section VIII, and parental consent for any student under 18.

X. DISCUSSION AND LIMITATIONS

The main lesson from turning the prototype into a platform is how little of the work concerned the model. The model, its explanation and the rules took fewer than 300 lines and did not change. Almost everything else, from sign-in, roles and isolation between colleges to imports, audit, backups, certificates and the evidence that each of these works, was needed before the model could be shown to a real teacher at all. This fits the reviews that find many prototypes and few deployments [2], [11]: the distance between the two is largely a matter of engineering and governance, not of prediction.

Exact contributions turned out to be both more useful and more delicate than expected. They let a teacher check a score by hand, which is the property Rudin argues for [20], yet the late-submissions weight in Section VII-C shows that a correct explanation of a model can still mislead a reader who takes it as a statement about causes. The student-facing wording and the limit of three reasons are attempts to manage this; whether they work is a question for Stage 1. The main limitations are the following.

- **Synthetic data.** All model results come from a generator with a linear structure. Real data will be noisier, may be non-linear, and may show a larger gap to tree models, as Bird *et al.* found when few predictors were available [21].
- **Uncalibrated bands.** The 60 and 35 cut-offs are inherited design choices. Rankings near a cut-off are sensitive to the choice of model [21], so the bands should be calibrated for each institution in Stages 0 and 2.
- **Transfer between colleges.** Each institution trains its own model on its own file, which avoids carrying a model from one college to another, a step known to lose recall [22]. The price is that a small college may stay below the training minimum and remain on the rules.
- **One machine.** One writer per database, the same-host condition of WAL mode [38] and a single virtual machine together set a ceiling. The load test covered one institution; many institutions at once were not tested.
- **Test coverage.** Apart from the back-end suite, measurements were taken in a container rather than on Rocky Linux; only Chromium was tested; no full screen-reader audit was done; and one-time codes are not phishing-resistant [40].
- **Fixed rules.** The rules encode expert judgement and do not learn which action helps which student.
- **One developer.** The system was built and tested by a single author with AI assistance, and it has not yet had an independent code review.

XI. CONCLUSION

This paper described ClassCircle, a multi-institution learning analytics platform that turns an earlier browser prototype into a system that a college could run on one virtual machine. Each institution keeps its records in its own database, the server alone

decides what each role sees, every risk score is shown with its exact additive explanation and practical next steps, and every dashboard figure opens the list behind it. Automated, browser, acceptance and attack checks passed on the released code; on synthetic data the model behaved as its construction predicts; and the platform handled 120 simultaneous student writes without an error on modest hardware. The paper deliberately makes no claim about student improvement. The next steps are the offline check on public data and a usability study, followed, with a college's approval, by a one-term comparison designed to show whether the flags lead to help that makes a difference.

ACKNOWLEDGMENT

The author used Claude (Anthropic), an AI assistant, during this work: to help write and test parts of the ClassCircle source code, to write and run the measurement and figure scripts behind Section VII and Figs. 1–4, to search for and summarise the literature in Section II, and to draft the text of all sections of this paper. The author set the direction of the work, reviewed and edited the text, checked the reported figures against the measurement outputs, and takes full responsibility for the content. No real student data were used; all names and records shown belong to an invented sample college.

REFERENCES

- [1] K. Verbert, E. Duval, J. Klerkx, S. Govaerts, and J. L. Santos, "Learning analytics dashboard applications," *Amer. Behav. Sci.*, vol. 57, no. 10, pp. 1500–1509, Oct. 2013, doi: 10.1177/0002764213479363.
- [2] B. A. Schwendimann et al., "Perceiving learning at a glance: A systematic literature review of learning dashboard research," *IEEE Trans. Learn. Technol.*, vol. 10, no. 1, pp. 30–41, 2017, doi: 10.1109/TLT.2016.2599522.
- [3] R. Kaliisa, K. Misiejuk, S. López-Pemas, M. Khalil, and M. Saqr, "Have learning analytics dashboards lived up to the hype? A systematic review of impact on students' achievement, motivation, participation and attitude," in *Proc. 14th Learn. Analytics Knowl. Conf. (LAK '24)*, 2024, pp. 295–304, doi: 10.1145/3636555.3636884.
- [4] S. Dawson, J. Jovanović, D. Gašević, and A. Pardo, "From prediction to impact: Evaluation of a learning analytics retention program," in *Proc. 7th Int. Learn. Analytics Knowl. Conf. (LAK '17)*, 2017, pp. 474–478, doi: 10.1145/3027385.3027405.
- [5] D. Rossman, M. Kurzweil, and B. Lewis, "MAAPS advising experiment: Evaluation findings after six years," Ithaca S+R, New York, NY, USA, May 2023, doi: 10.18665/sr.318895.
- [6] A. S. Zahoorahmed, "Student performance analytics and early-risk dashboard: A lightweight explainable learning analytics prototype," unpublished manuscript, Dept. of Computer Science, SIES College of Arts, Commerce and Science, Mumbai, India, 2026.
- [7] R. Bodily and K. Verbert, "Review of research on student-facing learning analytics dashboards and educational recommender systems," *IEEE Trans. Learn. Technol.*, vol. 10, no. 4, pp. 405–418, 2017, doi: 10.1109/TLT.2017.2740172.
- [8] R. Bodily et al., "Open learner models and learning analytics dashboards: A systematic review," in *Proc. 8th Int. Conf. Learn. Analytics Knowl. (LAK '18)*, Sydney, NSW, Australia, 2018, pp. 41–50, doi: 10.1145/3170358.3170409.
- [9] W. Matcha, N. A. Uzir, D. Gašević, and A. Pardo, "A systematic review of empirical studies on learning analytics dashboards: A self-regulated learning perspective," *IEEE Trans. Learn. Technol.*, vol. 13, no. 2, pp. 226–245, 2020, doi: 10.1109/TLT.2019.2916802.
- [10] R. Ferguson and D. Clow, "Where is the evidence? A call to action for learning analytics," in *Proc. 7th Int. Learn. Analytics Knowl. Conf. (LAK '17)*, 2017, pp. 56–65, doi: 10.1145/3027385.3027396.
- [11] R. Kaliisa, I. Jivet, and P. Prinsloo, "A checklist to guide the planning, designing, implementation, and evaluation of learning analytics dashboards," *Int. J. Educ. Technol. High. Educ.*, vol. 20, 2023, doi: 10.1186/s41239-023-00394-6.
- [12] K. E. Arnold and M. D. Pistilli, "Course Signals at Purdue: Using learning analytics to increase student success," in *Proc. 2nd Int. Conf. Learn. Analytics Knowl. (LAK '12)*, 2012, pp. 267–270, doi: 10.1145/2330601.2330666.
- [13] M. Caulfield, "Why the Course Signals math does not add up," *Hapgood* (blog), Sep. 26, 2013. [Online]. Available: <https://hapgood.us/2013/09/26/why-the-course-signals-math-does-not-add-up/>
- [14] C. Herodotou, M. Hlosta, A. Boroowa, B. Rienties, Z. Zdrahal, and C. Mangafa, "Empowering online teachers through predictive learning analytics," *Br. J. Educ. Technol.*, vol. 50, no. 6, pp. 3064–3079, 2019, doi: 10.1111/bjet.12853.
- [15] C. Herodotou, B. Rienties, A. Boroowa, Z. Zdrahal, and M. Hlosta, "A large-scale implementation of predictive learning analytics in higher education: The teachers' role and perspective," *Educ. Technol. Res. Dev.*, vol. 67, no. 5, pp. 1273–1306, Oct. 2019, doi: 10.1007/s11423-019-09685-0.
- [16] A. Namoun and A. Alshanqiti, "Predicting student performance using data mining and learning analytics techniques: A systematic literature review," *Appl. Sci.*, vol. 11, no. 1, 2021, Art. no. 237, doi: 10.3390/app11010237.
- [17] B. Albreiki, N. Zaki, and H. Alashwal, "A systematic literature review of student' performance prediction using machine learning techniques," *Educ. Sci.*, vol. 11, no. 9, 2021, Art. no. 552, doi: 10.3390/educsci11090552.
- [18] T. Susnjak, G. S. Ramaswami, and A. Mathrani, "Learning analytics dashboard: A tool for providing actionable insights to learners," *Int. J. Educ. Technol. High. Educ.*, vol. 19, 2022, Art. no. 12, doi: 10.1186/s41239-021-00313-7.
- [19] G. Ramaswami, T. Susnjak, A. Mathrani, and R. Umer, "Use of predictive analytics within learning analytics dashboards: A review of case studies," *Technol. Knowl. Learn.*, vol. 28, no. 3, pp. 959–980, 2023, doi: 10.1007/s10758-022-09613-x.
- [20] C. Rudin, "Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead," *Nature Mach. Intell.*, vol. 1, no. 5, pp. 206–215, May 2019, doi: 10.1038/s42256-019-0048-x.
- [21] K. A. Bird, B. L. Castleman, Z. Mabel, and Y. Song, "Bringing transparency to predictive analytics: A systematic comparison of predictive modeling methods in higher education," *AERA Open*, vol. 7, 2021, doi: 10.1177/23328584211037630.
- [22] S. M. Jayaprakash, E. W. Moody, E. J. M. Lauria, J. R. Regan, and J. D. Baron, "Early alert of academically at-risk students: An open source analytics initiative," *J. Learn. Anal.*, vol. 1, no. 1, pp. 6–47, 2014, doi: 10.18608/jla.2014.11.3.
- [23] C. Kung and R. Yu, "Interpretable models do not compromise accuracy or fairness in predicting college success," in *Proc. 7th ACM Conf. Learning @ Scale (L@S '20)*, 2020, pp. 413–416, doi: 10.1145/3386527.3406755.
- [24] H. Khosravi et al., "Explainable artificial intelligence in education," *Comput. Educ.: Artif. Intell.*, vol. 3, 2022, Art. no. 100074, doi: 10.1016/j.caeai.2022.100074.
- [25] R. Alamri and B. Alharbi, "Explainable student performance prediction models: A systematic review," *IEEE Access*, vol. 9, pp. 33132–33143, 2021, doi: 10.1109/ACCESS.2021.3061368.
- [26] R. F. Kizilcec, "How much information? Effects of transparency on trust in an algorithmic interface," in *Proc. CHI Conf. Human Factors Comput. Syst. (CHI '16)*, San Jose, CA, USA, 2016, pp. 2390–2395, doi: 10.1145/2858036.2858402.
- [27] R. S. Baker and A. Hawn, "Algorithmic bias in education," *Int. J. Artif. Intell. Educ.*, vol. 32, no. 4, pp. 1052–1092, Dec. 2022, doi: 10.1007/s40593-021-00285-9.
- [28] J. Gardner, C. Brooks, and R. Baker, "Evaluating the fairness of predictive student models through slicing analysis," in *Proc. 9th Int. Conf. Learn. Analytics Knowl. (LAK19)*, Tempe, AZ, USA, 2019, pp. 225–234, doi: 10.1145/3303772.3303791.

- [29] S. Slade and P. Prinsloo, "Learning analytics: Ethical issues and dilemmas," *Amer. Behav. Sci.*, vol. 57, no. 10, pp. 1510–1529, Oct. 2013, doi: 10.1177/0002764213479366.
- [30] H. Drachsler and W. Greller, "Privacy and analytics: It's a DELICATE issue—A checklist for trusted learning analytics," in *Proc. 6th Int. Conf. Learn. Analytics Knowl. (LAK '16)*, Edinburgh, U.K., 2016, pp. 89–98, doi: 10.1145/2883851.2883893.
- [31] Y.-S. Tsai et al., "The SHEILA framework: Informing institutional strategies and policy processes of learning analytics," *J. Learn. Anal.*, vol. 5, no. 3, pp. 5–20, 2018, doi: 10.18608/jla.2018.53.2.
- [32] *The Digital Personal Data Protection Act, 2023*, Act No. 22 of 2023, Government of India, Aug. 11, 2023. [Online]. Available: <https://www.indiacode.nic.in/bitstream/123456789/22037/2/a2023-22.pdf>
- [33] Ministry of Electronics and Information Technology, "Digital Personal Data Protection Rules, 2025," G.S.R. 846(E), *The Gazette of India: Extraordinary*, Nov. 13, 2025. [Online]. Available: <https://www.meity.gov.in/static/uploads/2025/11/53450e6e5dc0bfa85ebd78686cadad39.pdf>
- [34] R. Krebs, C. Momm, and S. Kounev, "Architectural concerns in multi-tenant SaaS applications," in *Proc. 2nd Int. Conf. Cloud Comput. Services Sci. (CLOSER)*, Porto, Portugal, 2012, pp. 426–431, doi: 10.5220/0003957604260431.
- [35] Microsoft, "Multitenant SaaS database tenancy patterns," Azure SQL Database documentation, Aug. 2025. [Online]. Available: <https://learn.microsoft.com/en-us/azure/azure-sql/database/saas-tenancy-app-design-patterns> (accessed Oct. 11, 2026).
- [36] Amazon Web Services, "Silo, pool, and bridge models," in *SaaS Lens—AWS Well-Architected Framework*. [Online]. Available: <https://docs.aws.amazon.com/wellarchitected/latest/saas-lens/silo-pool-and-bridge-models.html> (accessed Oct. 11, 2026).
- [37] SQLite Development Team, "Appropriate uses for SQLite." [Online]. Available: <https://www.sqlite.org/whentouse.html> (accessed Oct. 11, 2026).
- [38] SQLite Development Team, "Write-ahead logging." [Online]. Available: <https://www.sqlite.org/wal.html> (accessed Oct. 11, 2026).
- [39] OWASP Foundation, "A01:2021—Broken access control," *OWASP Top 10:2021*, 2021. [Online]. Available: https://owasp.org/Top10/A01_2021-Broken_Access_Control/ (accessed Oct. 11, 2026).
- [40] D. Temoshok et al., "Digital identity guidelines: Authentication and authenticator management," Nat. Inst. Standards Technol., Gaithersburg, MD, USA, NIST SP 800-63B-4, Jul. 2025, doi: 10.6028/NIST.SP.800-63b-4.
- [41] J. Brooke, "SUS: A 'quick and dirty' usability scale," in *Usability Evaluation in Industry*, P. W. Jordan et al., Eds. London, U.K.: Taylor & Francis, 1996, pp. 189–194.
- [42] J. Kuzilek, M. Hlosta, and Z. Zdrahal, "Open University Learning Analytics dataset," *Sci. Data*, vol. 4, 2017, Art. no. 170171, doi: 10.1038/sdata.2017.171.
- [43] F. D. Davis, "Perceived usefulness, perceived ease of use, and user acceptance of information technology," *MIS Quart.*, vol. 13, no. 3, pp. 319–340, 1989, doi: 10.2307/249008.
- [44] A. Bangor, P. Kortum, and J. Miller, "Determining what individual SUS scores mean: Adding an adjective rating scale," *J. Usability Stud.*, vol. 4, no. 3, pp. 114–123, May 2009.
- [45] D. Lakens, "Calculating and reporting effect sizes to facilitate cumulative science: A practical primer for t-tests and ANOVAs," *Front. Psychol.*, vol. 4, 2013, Art. no. 863, doi: 10.3389/fpsyg.2013.00863.
- [46] B. Rienties, C. Herodotou, T. Olney, M. Schencks, and A. Boroowa, "Making sense of learning analytics dashboards: A technology acceptance perspective of 95 teachers," *Int. Rev. Res. Open Distrib. Learn.*, vol. 19, no. 5, 2018, doi: 10.19173/irrodl.v19i5.3493.