IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 4, November 2026
Impact Factor: 8.2

Learnalytix: An Intelligent Analytical Modeling System for Academic Outcome Classification and Dropout Risk Assessment Using Supervised Learning Techniques

Gauravi Patil, Trisha Pujari, Vibha Patil
Department of Computer Science and Technology
Usha Mittal Institute of Technology, S.N.D.T. Women's University, Mumbai, India
gauravipatil365@gmail.com; trishapujari05@gmail.com; patilvibha98@gmail.com;

Abstract: *Higher-education institutions collect substantial student data, yet identifying academically at-risk students often remains manual, retrospective, and hard to interpret. This paper presents Learnalytix, an explainable machine-learning platform that predicts binary Pass/Fail outcomes and three-class dropout risk (Low, Medium, High) using semester-wise SGPA, attendance, backlog history, and engagement attributes. After identifier removal and categorical encoding, Gradient Boosting and Random Forest models are trained and combined into a unified risk score, accompanied by global feature importance, SHAP-style explanations, and rule-based intervention suggestions. The platform is implemented with React JS, Flask REST APIs, MongoDB, and JWT role-based access control. Evaluated on 500 undergraduate engineering records (60:40 split), the dropout-risk model achieved 86.50% test accuracy and 83.80% five-fold cross-validated accuracy, while Medium-risk recall (0.21) and Pass/Fail target leakage highlight the need for larger, multi-institutional validation..*

Keywords: Academic Analytics, Student Dropout Prediction, Educational Data Mining, Supervised Learning, Explainable AI, Random Forest, Gradient Boosting

I. INTRODUCTION

Increasing enrolment, diverse academic programmes, and institutional accountability have strengthened the need for data-driven approaches to student success and retention. Conventional practices—including manual grade registers, spreadsheets, ad hoc counselling, and end-of-semester reviews—often identify struggling students only after academic difficulties have accumulated. Low attendance, declining academic performance, and disengagement are commonly reported early-warning signals [2], [5], [9].

Educational data mining and learning analytics provide opportunities to transform institutional records into actionable insight. Despite this potential, many workflows do not combine multiple academic and engagement indicators into a unified, quantified, and explainable risk estimate. Students may also lack a self-service mechanism to understand their own academic risk.

Learnalytix addresses these gaps through an end-to-end platform that estimates Pass/Fail outcomes and dropout risk, explains contributing factors, and presents results through separate student and administrator portals. The principal contributions are: (1) an integrated academic prediction pipeline; (2) a combined risk score derived from two model outputs; (3) global feature importance and per-student SHAP-style explanations with rule-based recommendations; (4) a full-stack implementation with JWT-based role access control; and (5) a preliminary evaluation on 500 engineering student records, including discussion of limitations.

Copyright to IJARSCT
www.ijarsct.co.in
DOI: 10.48175/IJARSCT-40971

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 4, November 2026
Impact Factor: 8.2

II. RELATED WORK

A. Educational Data Mining

Romero and Ventura surveyed educational data mining research and identified prediction as a dominant task, with classification and ensemble approaches frequently applied to educational data [1]. Their review supports the use of supervised classification to estimate student outcomes.

B. Student Dropout Prediction

Mduma et al. examined machine-learning methods for student dropout prediction and reported the use of Decision Trees, Support Vector Machines, Logistic Regression, and ensemble models [2]. Arnold and Pistilli demonstrated how learning analytics can provide actionable early-warning signals to support student success [4]. Marburger reported a relationship between absenteeism and examination performance, motivating the inclusion of attendance indicators [5]. More recently, Niyogisubizo et al. [11] used stacked ensembles for university class dropout prediction, while Kim et al. [12] combined resampling with boosted trees to improve dropout precision and recall on imbalanced institutional records.

C. Explainable Prediction

Lundberg and Lee introduced SHAP, a unified framework for interpreting model predictions through additive feature contributions [3]. In educational contexts, Krüger et al. [13] demonstrated that explainable machine-learning pipelines help staff understand the factors associated with a dropout risk estimate before deciding whether support or intervention is appropriate.

D. Engagement and Socio-Academic Factors

Engagement indicators, attendance, and socio-academic factors are considered relevant to academic performance and dropout prediction [10]. Learnalytix includes project, hackathon, and internship-related attributes alongside academic records. In the present dataset, however, these engagement features contributed comparatively little to the fitted dropout model's feature importance.

III. PROBLEM STATEMENT AND FORMAL DEFINITION

A. Problem Motivation

Student academic and behavioural information is often reviewed manually, resulting in delayed and inconsistent insights. Existing workflows may provide descriptive Pass/Fail status or rough dropout estimates, but lack predictive capability, explanations, integrated risk scoring, student self-service, and scalable role-separated access.

B. Formal Definition

Let $D = \{(x_i, y_i^{pf}, y_i^{dr})\}$ for $i = 1, \ldots, N$ represent a dataset. Each $x_i \in R^d$ is a feature vector containing academic, attendance, backlog, and engagement attributes. The Pass/Fail label $y_i^{pf} \in \{0, 1\}$ and dropout-risk label $y_i^{dr} \in \{0, 1, 2\}$ correspond to the encoded classes. The objective is to learn $f^{pf}: R^d \rightarrow \{0, 1\}$ and $f^{dr}: R^d \rightarrow \{0, 1, 2\}$ while producing interpretable feature contributions for individual predictions.

IV. EXISTING AND PROPOSED SYSTEMS

A. Existing System

The existing workflow relies on manual analysis of student records, spreadsheets, and static reports. It is time-consuming, reactive, inconsistent, and offers limited predictive or explanatory capability. It also lacks a centralised database and does not provide students with access to their own predictions.

B. Proposed System

Learnalytix automates data processing and prediction through a three-tier web platform. Students register, submit semester-wise information, and receive their Pass/Fail prediction, dropout-risk category, and explanation. Administrators can upload or review records, view branch- and year-wise analytics, inspect student records, and export at-risk reports. The platform combines a React frontend, Flask REST API, MongoDB persistence, ensemble models, explainability, and server-side JWT role checks.

Copyright to IJARSCT
www.ijarsct.co.in
DOI: 10.48175/IJARSCT-40971

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 4, November 2026
Impact Factor: 8.2

V. SYSTEM ARCHITECTURE

The system is organised into three tiers: (1) React JS frontend with Student and Admin portals; (2) Flask REST API for authentication, submission, prediction, and analytics; and (3) MongoDB for users, student records, and predictions. The tiers are independently deployable, and the document-oriented database supports variable semester counts and optional attributes.

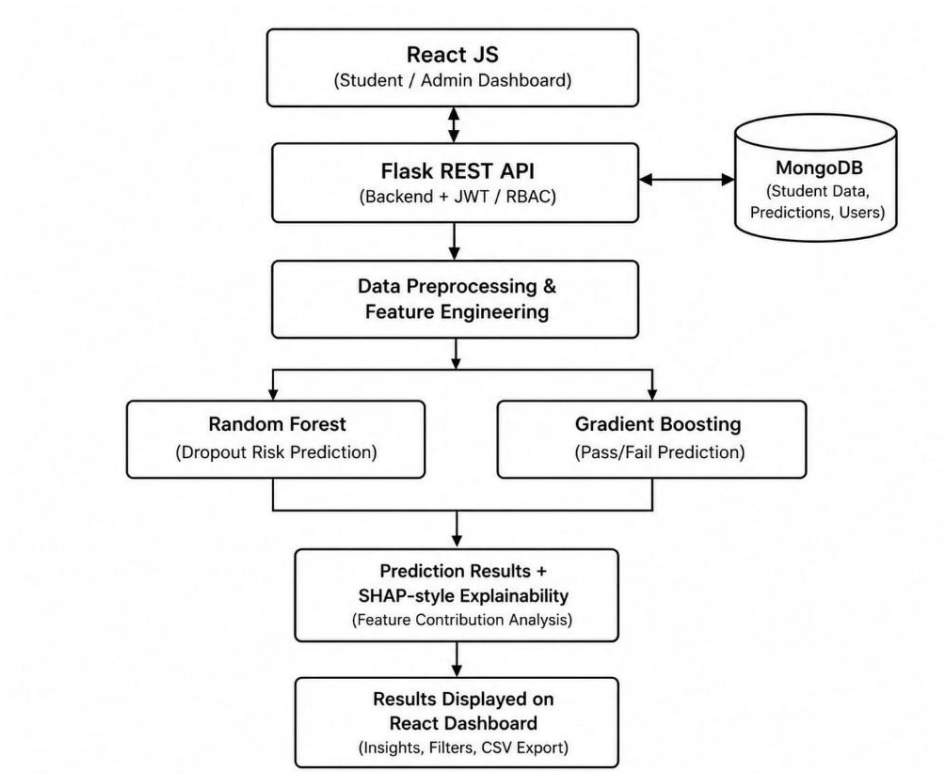


Fig. 1. Learnalytix three-tier architecture and prediction pipeline.

A. Functional Modules

The platform includes authentication and role-based access control, student and administrator frontend modules, the Flask API, data preprocessing, the prediction engine, explainability, MongoDB storage, and report generation for at-risk students.

B. Workflow

An administrator uploads a CSV containing student identifiers, branch, year, semester-wise SGPA, attendance, CGPA, backlog counts, internship status, projects, and hackathons. The backend validates and preprocesses the records, aligns the feature vector with the training feature order, runs the trained models, and stores predictions. Students can also submit their own data through the portal, triggering the same prediction path. The overall workflow is: Login → Data Input → Preprocess → Predict → Store → Dashboard.

VI. ALGORITHMS AND METHODOLOGY

A. Preprocessing

The preprocessing pipeline removes identifier columns such as student ID and name, derives CGPA where needed, label-encodes categorical attributes including branch and internship, and passes numerical features to the models. The resulting feature vector contains 23 attributes: CGPA, overall attendance, semester-wise SGPA and attendance,

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 4, November 2026
Impact Factor: 8.2

backlogs, year, current semester, projects, and hackathons. While the pipeline supports Min–Max scaling for non-tree baseline comparisons, tree ensembles in the reported experiments operate directly on unscaled features. Derived features such as SGPA trend, SGPA standard deviation, attendance trend, and an aggregate engagement score were considered but were not included in the reported experiments.

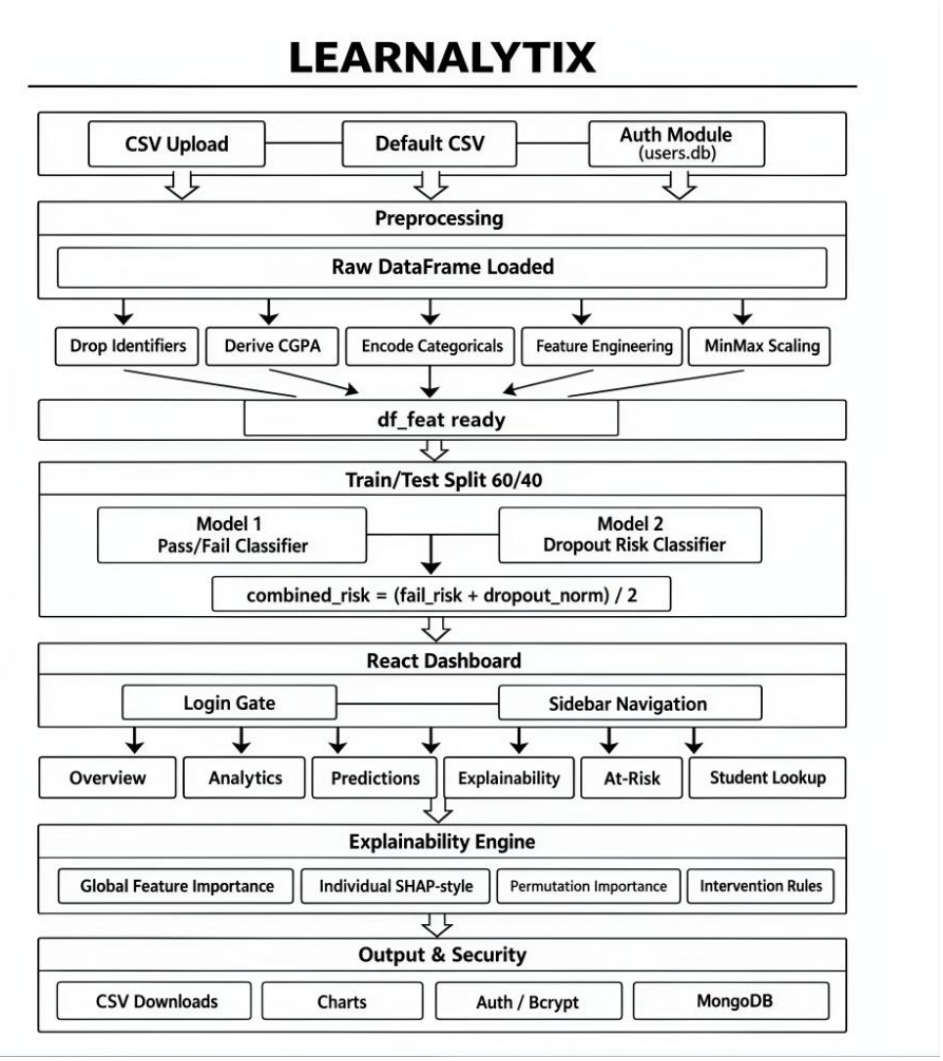


Fig. 2. End-to-end Learnalytix preprocessing, modeling, dashboard, and explainability workflow.

B. Gradient Boosting Classifier

Gradient Boosting builds an ensemble of decision trees sequentially, with each learner fitted to the negative gradient of the current loss. The ensemble update is expressed as $F_m(x) = F_{m-1}(x) + \eta h_m(x)$, where h_m is the m-th weak learner and η is the learning rate [6]. Learnalytix uses Gradient Boosting for binary Pass/Fail classification.

C. Pass/Fail Prediction

The binary classifier estimates the probability of the Fail class. A student is predicted as Fail when the estimated probability is at least 0.5; otherwise, the prediction is Pass.

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 4, November 2026
Impact Factor: 8.2

D. Dropout Risk Estimator

The Random Forest classifier estimates probabilities for Low, Medium, and High dropout risk [7]. A weighted risk value is calculated from the class probabilities: $r_i^{dr} = 0 \cdot p_L + 0.5 \cdot p_M + 1.0 \cdot p_H$, where p_L, p_M, and p_H are the predicted probabilities for the three risk categories.

E. Combined Risk Score

The two model outputs are fused using simple averaging: Combined $Risk_i = (p_i^{pf} + r_i^{dr}) / 2$. This provides an unweighted composite indicator combining immediate academic failure likelihood with longer-term dropout risk; individual model outputs and explanations are stored alongside the fused score in the predictions collection.

F. Explainability

The explainability module presents global feature importance and per-student SHAP-style contributions based on Shapley values [3]. It also applies rule-based intervention suggestions, for example, recommendations triggered by low attendance or declining SGPA. Explanation values are serialised to MongoDB and displayed on the student results view.

VII. IMPLEMENTATION

A. Technology Stack

The backend and machine-learning pipeline use Python 3.9+, Flask 2.3+, PyMongo, Pandas, NumPy, Scikit-learn 1.3+ [8], and SHAP 0.42+. Charts use Matplotlib/Seaborn, while authentication uses PyJWT and bcrypt. The frontend is built with React JS 18+ and Axios, with Node.js 18+. MongoDB 6.0+ is used for persistence. The stated minimum hardware is a dual-core 1.8 GHz CPU, 4 GB RAM, and 500 MB storage.

B. Authentication and Access Control

Login issues a JWT containing a role claim. Protected Flask routes validate the token and enforce the role server-side. Passwords are hashed with bcrypt. Students can access their own submissions, predictions, explanations, and history; administrators can access institution-wide analytics, records, and at-risk exports.

C. Data Model

The users collection stores identifiers, email, password hash, role, and creation time. student_records stores the student reference, semester SGPA array, attendance percentages, backlog count, internship flag, project and hackathon counts, branch, and year. predictions stores the student reference, Fail probability, dropout-risk class, combined risk, explanation values, and prediction timestamp.

D. REST API

TABLE I. PRINCIPAL FLASK API ENDPOINTS

Method	Endpoint	Description
POST	/api/auth/register	Register student or administrator
POST	/api/auth/login	Authenticate and return JWT
POST	/api/student/submit	Submit data and trigger prediction
GET	/api/student/results	Own prediction and explanation results
GET	/api/admin/students	View all records (admin only)
GET	/api/admin/analytics	Branch/year analytics
GET	/api/admin/atrisk	Download at-risk report

E. Machine-Learning Pipeline

Training is performed offline using a script that loads the 500-student dataset, removes identifiers, encodes branch and internship, and trains both classifiers on the 60% training split. The script reports test metrics and five-fold cross-validated accuracy and serialises the models, label encoders, and metadata. At inference, the API loads these artefacts,

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 4, November 2026
Impact Factor: 8.2

applies the saved encoders, aligns submitted features to the training order, predicts, fuses the outputs, computes explanations, stores the results, and returns JSON to the frontend.

F. Frontend

The Student Portal supports registration, login, multi-step data entry, a prediction results view, a dropout-risk gauge, and an explanation view with feature contributions and recommendations. The Admin Portal provides an overview dashboard, branch- and year-wise analytics, a sortable and filterable student table, and an at-risk export.

VIII. EVALUATION

A. Dataset and Experimental Setup

The evaluation used 500 undergraduate engineering student records ([specify: anonymized institutional records / synthetic dataset]) from a single institution. The dataset contains 270 Low-risk students (54%), 160 High-risk students (32%), and 70 Medium-risk students (14%). Pass/Fail labels comprise 312 Fail (62.4%) and 188 Pass (37.6%). A 60:40 split yielded 300 training and 200 test records. The test set contained 108 Low, 64 High, and 28 Medium dropout-risk students, and 125 Fail and 75 Pass students. Twenty-three input features were used. Performance was evaluated using accuracy, weighted precision, recall, and F1-score, per-class metrics, confusion matrices, and five-fold cross-validated accuracy.

TABLE II. MODEL PERFORMANCE ON THE 200-STUDENT TEST SPLIT

Metric	Dropout Risk (RF)	Pass/Fail (GB)
Accuracy	86.50%	100.00%
Weighted Precision	83.71%	100.00%
Weighted Recall	86.50%	100.00%
Weighted F1-score	83.85%	100.00%
5-fold CV Accuracy	83.80%	100.00%

B. Per-Class Dropout-Risk Performance

The dropout-risk model performed strongly for the Low and High classes but was less effective for the Medium-risk class. Only 6 of the 28 Medium-risk test students were correctly identified; 16 were classified as Low and 6 as High. No Low-risk student was misclassified, and no High-risk student was predicted as Low, although five High-risk students were classified as Medium.

TABLE III. PER-CLASS RESULTS FOR DROPOUT RISK

Class	Precision	Recall	F1-score	Support
High	0.91	0.92	0.91	64
Low	0.87	1.00	0.93	108
Medium	0.55	0.21	0.31	28
Macro Average	0.77	0.71	0.72	200
Weighted Average	0.84	0.86	0.84	200

TABLE IV. CONFUSION MATRIX FOR DROPOUT RISK (ACTUAL ROWS; PREDICTED COLUMNS)

Actual / Predicted	High	Low	Medium
High	59	0	5
Low	0	108	0
Medium	6	16	6

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 4, November 2026
Impact Factor: 8.2

C. Feature Importance

The leading features in the Random Forest dropout model were CGPA, overall attendance, early-semester SGPA, and backlog count. These results align with the importance of academic performance and attendance reported in prior work [2], [5]. Projects, hackathons, and internship attributes each had importance below 0.01; year and current semester were each below 0.005 in this dataset.

TABLE V. TOP FEATURE IMPORTANCES

Feature	Importance
CGPA	0.1715
Overall attendance (%)	0.1184
Semester 2 SGPA	0.1038
Backlogs	0.0995
Semester 1 SGPA	0.0824
Semester 1 attendance (%)	0.0662
Semester 4 SGPA	0.0566
Semester 3 attendance (%)	0.0521
Semester 2 attendance (%)	0.0521
Semester 3 SGPA	0.0358

D. System Testing

Functional testing covered eight reported test cases, all marked Passed: administrator login with JWT, student login and role enforcement, student data submission triggering prediction, Pass/Fail and dropout-risk output, React dashboard rendering, explanation display, student restriction from administrator endpoints, and MongoDB persistence.

TABLE VI. SYSTEM TEST CASES

ID	Test Case	Status
TC-01	Admin login (JWT)	Passed
TC-02	Student login; role enforced	Passed
TC-03	Data submission triggers prediction	Passed
TC-04	Pass/Fail and dropout output	Passed
TC-05	React dashboard rendering	Passed
TC-06	Explanation display	Passed
TC-07	Student blocked from admin endpoints	Passed
TC-08	MongoDB persistence	Passed

IX. DISCUSSION AND LIMITATIONS

The results suggest that tree-based ensembles can distinguish academic-risk categories using academic and engagement-related inputs. The dropout-risk model achieved 86.50% test accuracy and 83.80% five-fold cross-validated accuracy. Nevertheless, weighted metrics conceal uneven class performance: Medium-risk recall was 0.21, indicating that the model missed a substantial portion of students in this category. The Medium class is the smallest (14%) and lies between Low and High, which makes it harder to separate without class-imbalance techniques.

The Pass/Fail classifier achieved 100% on the test split and cross-validation. This result should not be interpreted as proof of generalisation; the label appears to be largely determined by input features such as CGPA and backlogs, creating a target-leakage or label-dependence concern. Because the combined risk score currently averages this Pass/Fail probability with the dropout risk estimate, future iterations should decouple Pass/Fail rule checks from predictive dropout modeling. The dataset also contains 500 records from a single institution, limiting external validity.

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429 Volume 6, Issue 4, November 2026 Impact Factor: 8.2

Additional limitations include the use of a single hold-out split, class imbalance, and the absence of visual SHAP/UI evaluation figures in this preliminary report. Future evaluation should use larger, multi-institution datasets, repeated or nested validation, semester-specific checkpoints that restrict features to information available at prediction time, class-imbalance strategies (such as class weighting or SMOTE [12]), and calibration of the combined score.

X. CONCLUSION AND FUTURE WORK

This paper presented Learnalytix, an explainable academic analytics platform for Pass/Fail classification and dropout-risk assessment. The system combines preprocessing, a Gradient Boosting Pass/Fail classifier, a Random Forest dropout-risk estimator, a fused risk score, feature explanations, and role-separated web portals using React JS, Flask, and MongoDB. Preliminary results on 500 undergraduate engineering records show 86.50% dropout-risk accuracy, while highlighting weak Medium-risk recall and the need for cautious interpretation of perfect Pass/Fail performance.

Future work includes evaluating on larger and multi-institution datasets; repeated cross-validation; engineered SGPA and attendance trends; class-imbalance methods to improve Medium-risk recall; multi-checkpoint prediction; calibration and decoupling of the combined risk score; and an enhanced student dashboard with progress tracking, goal setting, personalised nudges, and tailored recommendations.

ACKNOWLEDGMENT

The authors thank the Department of Computer Science and Technology, Usha Mittal Institute of Technology, S.N.D.T. Women's University, Mumbai, for institutional support and resources.

REFERENCES

[1] C. Romero and S. Ventura, "Educational data mining: A review of the state-of-the-art," IEEE Trans. Syst., Man, Cybern. C, Appl. Rev., vol. 40, no. 6, pp. 601–618, Nov. 2010.

[2] N. Mduma, K. Kalegele, and D. Machuve, "A survey of machine learning approaches and techniques for student dropout prediction," Data Science Journal, vol. 18, no. 14, pp. 1–14, 2019.

[3] S. M. Lundberg and S.-I. Lee, "A unified approach to interpreting model predictions," in Adv. Neural Inf. Process. Syst. (NeurIPS), vol. 30, 2017, pp. 4765–4774.

[4] K. E. Arnold and M. D. Pistilli, "Course signals at Purdue: Using learning analytics to increase student success," in Proc. 2nd Int. Conf. Learning Analytics and Knowledge (LAK), 2012, pp. 267–270.

[5] D. R. Marburger, "Absenteeism and undergraduate exam performance," J. Economic Education, vol. 32, no. 2, pp. 99–109, 2001.

[6] J. H. Friedman, "Greedy function approximation: A gradient boosting machine," Ann. Statist., vol. 29, no. 5, pp. 1189–1232, 2001.

[7] L. Breiman, "Random forests," Mach. Learn., vol. 45, no. 1, pp. 5–32, 2001.

[8] F. Pedregosa et al., "Scikit-learn: Machine learning in Python," J. Mach. Learn. Res., vol. 12, pp. 2825–2830, 2011.

[9] V. Tinto, "Dropout from higher education: A theoretical synthesis of recent research," Rev. Educ. Res., vol. 45, no. 1, pp. 89–125, 1975.

[10] A. Alyahyan and D. Düsterdör, "Predicting academic success in higher education: Literature review and best practices," Int. J. Educ. Technol. High. Educ., vol. 17, art. no. 3, 2020.

[11] J. Niyogisubizo, L. Liao, E. Nziyumva, E. Murwanashyaka, and P. C. Nshimyumukiza, "Predicting student's dropout in university classes using two-layer ensemble machine learning approach: A novel stacked generalization," Comput. Educ. Artif. Intell., vol. 3, art. no. 100066, 2022.

[12] S. Kim, E. Choi, Y.-K. Jun, and S. Lee, "Student dropout prediction for university with high precision and recall," Appl. Sci., vol. 13, no. 10, art. no. 6275, May 2023, doi: 10.3390/app13106275.

[13] J. G. C. Krüger, A. de S. Britto, and J. P. Barddal, "An explainable machine learning approach for student dropout prediction," Expert Syst. Appl., Jul. 2023.