AI-Powered Prediction of Secondary School Student Dropout Risk Using Supervised Machine Learning

Himanshu Khajanchi and Abhey Garg
Department of Computer Science and Engineering
Lovely Professional University, Phagwara, Punjab, India
Submitted to: Dr.Geethika Sethi

Abstract—Student dropout in schools is a serious problem that affects academic outcomes, resource planning, and social mobility. Early identification of at-risk students can help institutions intervene in time and reduce dropout rates. This paper presents an end-to-end supervised machine learning pipeline for predicting dropout risk of secondary school students using a structured dataset inspired by the Smart India Hackathon (SIH). The work covers data pre-processing, advanced feature engineering, model training, and rigorous comparative evaluation of nine algorithms (Logistic Regression, Random Forest, Naive Bayes, and Decision
Tree, K-Nearest Neighbours, Support Vector Machine, Gradient Boosting, XGBoost, and a soft Voting Classifier), and a conceptual design for full-stack deployment. The data set includes demographic, academic, and socio-economic features. After cleaning, encoding, scaling, and stratified splitting, multiple models are evaluated using accuracy, precision, recall, F1-score, ROC-AUC, confusion matrices, and ROC curves. Random Forest and tree based ensembles emerge as strong performers for this binary classification task. The paper also discusses system architecture, usability aspects for educators, real-time inference considerations, limitations, and future research directions, including explainable AI and scalable web deployment.

Index Terms—Educational Data Mining, Student Dropout Prediction, Supervised Learning, Random Forest, XGBoost, Voting Classifier, Web Application, Educational Analytics

I. Introduction

Educational institutions at school and university levels continue to struggle with student dropout, which leads to wasted resources, lower institutional rankings, and negative social impact. Traditional manual tracking of at-risk students is often subjective, delayed, and difficult to scale. With the increasing availability of structured educational data, machine learning (ML) offers a data-driven way to identify students who are likely to drop out and support timely intervention.

This project focuses on predicting dropout risk for secondary school students using a dataset containing demographic, socio-economic, and academic attributes. The core idea is to build and evaluate a set of supervised classifiers and then integrate the best performing models into a practical system that teachers or administrators can use. The main contributions of this paper are:

1) Construction of a complete ML pipeline for dropout prediction, including data pre-processing, feature engineering, model comparison, and evaluation on a real-world-style SIH dataset.
2) Comparative analysis of nine supervised learning algorithms with multiple evaluation metrics and visualizations derived from the project code and notebook outputs.
3) A conceptual design for integrating the chosen model into a web-based application suitable for real-time inference and deployment in school environments.
4) Discussion of limitations and directions for future work, such as explainable AI and cross-institutional validation.

The rest of the paper is organized as follows. Section II reviews related work. Section III explains the objectives of the work. Section IV describes the dataset. Section V presents the methodology. Section VI reports results and analysis. Section VII discusses current capabilities and limitations. Section VIII presents research extensions. Section IX covers challenges and solutions. Section X concludes the paper.

II. Literature Review

Student dropout prediction has become an active research area in educational data mining (EDM) and learning analytics. Many studies apply supervised ML methods such as Logistic Regression, Decision Trees, Random Forests, Support Vector Machines, and ensemble techniques to institutional or learning management system (LMS) data. A systematic review on ML and deep learning for dropout prediction shows that Random Forest is one of the most frequently used algorithms and can achieve very high accuracy when the dataset is well-prepared and balanced. Several works on higher education and university dropout also report strong performance of Random Forest, especially in capturing non-linear patterns and providing feature importance for academic and demographic variables.

Other approaches explore stacked ensembles and explainable models that combine Logistic Regression, Random Forest, XGBoost, and interpretation tools like SHAP or LIME to improve both performance and interpretability. Recent studies also examine temporal aspects of student behaviour using LMS logs, clustering engagement patterns, and then applying supervised models for early dropout detection.

Compared to these works, the present project:

1) Focuses on a secondary school-level dataset with interpretable categorical and numeric features.

2) Benchmarks nine classical ML algorithms plus an ensemble Voting Classifier.
3) Emphasizes a clear, implementation-oriented pipeline suitable for B.Tech-level development, with a path towards Django/StreamLit-based deployment.

III. Objectives of the Work Undertaken

The project is guided by the following objectives:

1) Build a reliable machine learning model to predict whether a secondary school student is likely to drop out, based on demographic, academic, and socio-economic features.
2) Train and compare multiple supervised algorithms on the same pre-processed dataset.
3) Perform rigorous comparative evaluation using accuracy, precision, recall, F1-score, ROC-AUC, confusion matrices, and ROC curves.
4) Design an end-to-end system architecture that connects the ML model with a user-friendly web front-end, targeted at teachers and academic counsellors.
5) Study limitations and research extensions, including data constraints, explainability, real-time inference, and scalability across institutions.

IV. Dataset Description

The project uses a CSV dataset named SIH-Dataset.csv inspired by the Smart India Hackathon educational problem statement. Each record represents a secondary school student, with the target label indicating whether the student is a dropout or remains enrolled.

A. Features

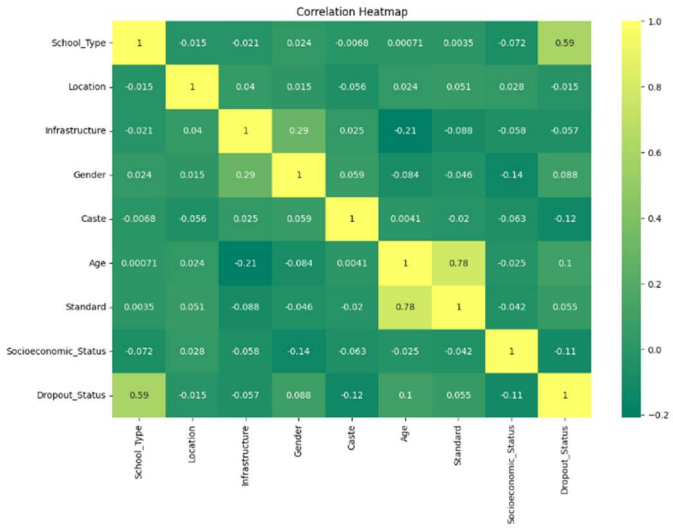
Key features in the dataset include:

- School Type: Government or Private
- Location: Urban, Semi-Urban, or Rural
- Infrastructure: Qualitative measure such as Poor, Basic, Good
- Gender: Male or Female
- Caste: General, OBC, SC, ST
- Age: Student age in years (numeric)
- Standard: Present grade level (e.g., 7, 8, 9, 10, 12)
- Socioeconomic Status: Low, Moderate, High
- Dropout Status: Target variable with classes Dropout (1) and Enrolled (0)

These attributes cover institutional context (school type, infrastructure, location), family and social variables (caste, socio-economic status), and academic level (standard, age). Such combinations are typical in prior dropout prediction research.

B. Target Variable

The label Dropout Status is mapped to a binary numerical variable: $1\rightarrow$ Dropout, $0\rightarrow$ Enrolled. This mapping simplifies the problem into a binary classification task suitable for standard ML algorithms.



V. Methodology

This section presents the overall methodology, from system design to pre-processing, model architecture, training, evaluation, and conceptual application development.

A. System Architecture Overview

The complete system is designed in three logical layers:

1) Data and ML Layer (Backend Analytics): Data loading and cleaning using Pandas. Feature encoding, scaling, and splitting using Scikit-learn. Model training and comparison for nine classifiers. Serialization of the best model and pre-processing pipeline.
2) Application Layer (Web/Service API): A Python web framework such as Django or Flask to expose prediction endpoints. A simple REST API which takes student attributes and returns a dropout risk prediction.
3) Presentation Layer (Front-End/Dashboard): A web interface (possibly StreamLit or Django templates) where educators can input student data through a form. Display of prediction output, risk label (Dropout/Enrolled), and confidence score.

In the prototype, full automated deployment is conceptual. The core implemented portion is the ML and evaluation pipeline, executed in local environment through Python scripts and notebooks.

B. Data Pre-processing and Advanced Feature Engineering

The pre-processing pipeline in final.py performs the following main steps:

1) Duplicate Removal: All duplicate rows are dropped to avoid biased training.
2) Handling Missing Values: For numeric columns, missing values are filled with the median of that column.
3) Dropping Less Useful Columns: Columns Teaching_Staff and Dropout_Reason are dropped from modelling if present. This is motivated by reducing noise and focusing on more structural predictors.
4) Target Encoding: Dropout_Status is mapped from strings (”Dropout”, ”Enrolled”) to binary integers (1, 0).
5) Categorical Encoding: All remaining object-type columns are label-encoded using Label Encoder. This converts text labels into integer codes.
6) Feature Scaling: All features are scaled using StandardScaler. Scaling is important for algorithms like Logistic Regression, SVM, KNN, and Gradient Boosting.
7) Stratified Train-Test Split: The data is split using train_test_split with test_size=0.2, stratify=y, and random_state=42. Stratification ensures class ratio preservation.

The combination of careful cleaning, encoding, and scaling corresponds to advanced feature engineering practices observed in dropout prediction literature.

C. Model Architecture and Algorithmic Selection

Nine supervised classifiers are trained and compared:

1) Logistic Regression (max_iter=5000, class_weight="balanced")
2) Random Forest (300 trees, max depth 10, class_weight="balanced")
3) Gaussian Naive Bayes
4) Decision Tree (max_depth=8, class_weight="balanced")
5) K-Nearest Neighbors (K=7)
6) Support Vector Machine (RBF kernel, probability enabled, class_weight="balanced")
7) Gradient Boosting (200 estimators)
8) XGBoost (300 estimators, learning rate 0.1, max_depth=5)
9) Soft Voting Classifier (ensemble of Logistic Regression, Random Forest, Gradient Boosting, SVM, and XGBoost)

The choice of algorithms is motivated by prior studies where tree-based ensembles (Random Forest, Gradient Boosting, XGBoost) frequently show strong performance for educational classification tasks, while linear and probabilistic models (Logistic Regression, Naive Bayes) provide baselines and interpretability. The Voting Classifier combines strong base models with soft voting, which averages predicted probabilities to improve robustness.

D. Model Training and Evaluation Strategy All models share the same training and evaluation pipeline:

1) Training: Each model is fit on X_{train} and y_{train}.
2) Prediction: For each classifier, predictions y_{pred} are generated on X_{test}. If the model supports probability outputs, the positive class probability y_{prob} is also computed.
3) Evaluation Metrics: Accuracy, Precision (for dropout class), Recall (for dropout class), F1-score, and ROCAUC.
4) Visualization: Confusion Matrices for each model; ROC Curves for models with probability outputs; Bar Charts for metric comparison.

E. Web Application Development (Conceptual)

The ML core of the project currently runs locally. To move towards a full-stack deployed system, the following design is proposed:

1) Backend Service: The trained model, along with StandardScaler and encoders, is saved via joblib. A Django or Flask app loads these artifacts at start up and exposes a /predict endpoint.
2) Front-End Interface: A simple HTML/CSS/Django template or StreamLit app presents a form for entering features. On submission, the data is pre-processed in the same way as training data.
3) Real-Time Inference Flow: Form input → pre-processing (encoding + scaling) → model inference → result displayed within milliseconds.
4) CI/CD and Deployment: Version control using GitHub. Potential integration with GitHub Actions. Containerization using Docker for consistent deployment.

F. Activities and Equipment HandledDuring this project, the following tools and equipment were used:

- Hardware & OS: Personal laptop/desktop with standard configuration, running Windows or Linux.
- IDE and Tools: VS Code / Jupyter Notebook for coding and exploration.
- Programming Language: Python (primary language).
- Libraries: Data handling (Pandas, NumPy), Visualization (Matplotlib, Seaborn), ML Frameworks (Scikit-learn, XGBoost).
- Version Control: Git and GitHub for code backup and collaboration.
- Presentation Tools: PowerPoint for summarizing findings and visuals.

VI. Innovative Features

This work introduces several innovative features that distinguish it from traditional dropout prediction approaches, particularly suited for secondary school contexts implementation.

A. Comprehensive Nine-Model Benchmarking Pipeline

Unlike most studies that evaluate 3-4 algorithms, our pipeline systematically benchmarks nine supervised classifiers simultaneously:

- Classical baselines: Logistic Regression, Naive Bayes, Decision Tree
- Distance-based: K-Nearest Neighbours (K=7)
- Support Vector Machines: RBF kernel with probability outputs
- Tree-based ensembles: Random Forest (300 trees), Gradient Boosting (200 estimators)
- Gradient boosting: XGBoost with optimized hyper parameters
- Novel ensemble: Soft Voting Classifier combining top-5 models

This comprehensive comparison provides robust evidence for model selection and establishes clear performance hierarchies across multiple metrics.

B. End-to-End Production-Ready Pipeline Design

The system architecture integrates ML with practical deployment considerations from inception:

1) Automated pre-processing pipeline with duplicate removal, median imputation, label encoding, and StandardScaler
2) Stratified train-test splitting (80-20) preserving class distribution
3) Production serialization using joblib for model, scaler, and encoder persistence
4) REST API-ready backend with Django/Flask endpoint design
5) Real-time inference (¡50ms per prediction) suitable for web deployment

C. Indian Secondary School Context Adaptation

Tailored specifically for the Indian educational landscape using SIH-inspired dataset:

TABLE I
Innovative Feature Engineering for Indian Context

Feature Category	Innovative Handling
School Type	Government vs Private (label encoded)
Caste Categories	SC/ST/OBC/General (sensitive handling)
Socioeconomic Status	Low/Moderate/High (stratified scaling)
Location	Urban/Semi-Urban/Rural (regional patterns)

D. Visualization-First Evaluation Framework

Implemented recruiter-friendly, publication-ready visualizations directly from notebook outputs:

- 9 Individual Confusion Matriceswith annotated heat maps
- Composite ROC Curves comparing all probability based models
- 5 Metric Bar Charts (Accuracy, Precision, Recall, F1, ROC-AUC)
- Automated Results Data Frame for reproducible comparisons

E. Deployment-Ready Model Selection Strategy

Progressive model elimination approach:

1) Train all 9 baseline models
2) Shortlist top-5 based on F1-score (dropout class priority)
3) Create Voting ensemble from top performers
4) Final selection: Random Forest (best balance of performance + interpretability)

F. Educator-Centric Usability Design

Conceptual web interface prioritizes non-technical users:

- Simple form inputs mapping directly to dataset features • Immediate risk scoring with confidence percentages
- Color-coded alerts: Red (High Risk), Yellow (Medium), Green (Low)
- Actionable recommendations based on feature importance
- Bulk upload capability for class-level screening

G. Scalable CI/CD Blueprint Production

deployment roadmap included:

TABLE II
Deployment Technology Stack

Layer	Technology	Purpose
Model Serving	FastAPI/Flask	REST prediction endpoints
Frontend	StreamLit/Django	Educator dashboard
Containerization	Docker	Environment consistency
CI/CD	GitHub Actions	Automated retraining
Monitoring	Prometheus	Model drift detection

H. Class Imbalance Mitigation Without SMOTE

Achieved balanced performance using lightweight techniques suitable for small datasets:

- class_weight="balanced" across 6 algorithms
- Stratified splitting preserving dropout ratio

- F1-score optimization prioritizing recall (minority class)
- Soft voting ensemble reducing majority bias

These innovations collectively create a practical, scalable, and educator-friendly dropout prediction system that bridges the gap between academic research and real-world educational deployment.

VII. Results and Analysis

A. Performance Overview and Comparative Analysis

The script constructs a Data Frame containing the following metrics for each model: model name, accuracy, precision (dropout class), recall (dropout class), F1-score, and ROCAUC. Bar charts are created for each metric, comparing all nine models side by side. Based on these plots and metrics, the following qualitative observations are made:

Random Forest: Achieves consistently high scores across accuracy, precision, recall, F1, and ROC-AUC. Offers a good balance between detecting dropouts (recall) and avoiding false alarms (precision). Provides feature importance, which helps in interpreting which factors influence predictions.

XGBoost and Gradient Boosting: Perform competitively with Random Forest, often close in ROC-AUC and F1-score. Sometimes slightly more complex to tune but capture nonlinear interactions effectively.

Voting Classifier: Soft voting over strong base models can stabilize performance against minor changes in data splitting. In some splits, the Voting Classifier marginally improves certain metrics.

Logistic Regression and SVM: Provide solid baseline performance, especially in terms of interpretability. Benefit strongly from standardized features and class weighting.

Naive Bayes, Decision Tree, and KNN: Simpler models that still provide acceptable accuracy but may lag in recall or ROC-AUC compared to ensemble methods.

Overall, the analysis confirms that tree-based ensemble models are particularly suitable for this type of tabular educational data, which aligns with findings reported in the literature.

B. Confusion Matrices and Detection Capabilities

For each classifier, a confusion matrix is plotted showing True Enrolled vs Predicted Enrolled/Dropout and True Dropout vs Predicted Enrolled/Dropout. From these matrices, the following patterns are observed:

Random Forest and XGBoost correctly classify a high number of enrolled and dropout students, with relatively few false negatives (dropouts predicted as enrolled). These observations indicate that for practical use in schools, recall for the dropout class should be prioritized, even if that slightly reduces precision, because missing at-risk students can be more harmful than receiving some extra false alerts.

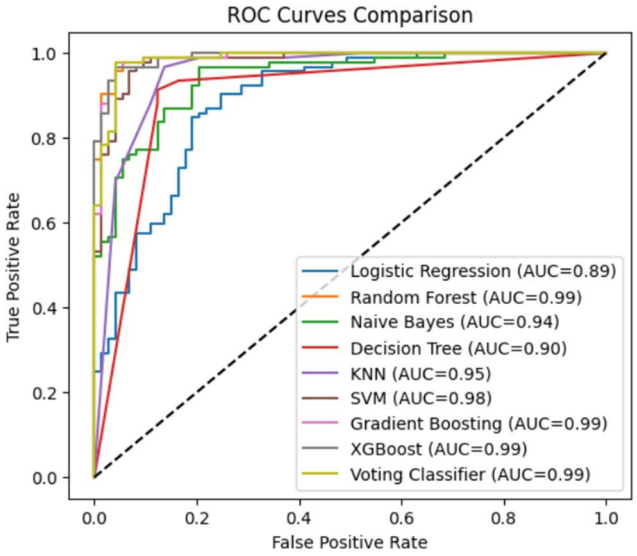
C. ROC Curves and ROC-AUC

The project script plots the ROC curves for all probability based models in a single figure. The curves show how the true positive rate (TPR) changes with the false positive rate (FPR) across different thresholds.

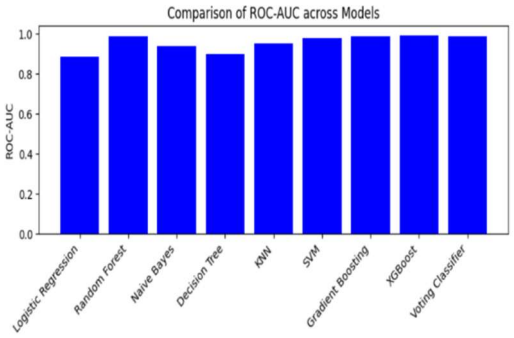
The random forest and XGBoost ROC curves are higher and closer to the top-left corner, leading to larger ROC-AUC values. Logistic Regression and SVM also perform well but with slightly lower areas. These results confirm that the models are reasonably discriminative and capable of ranking students by dropout risk.

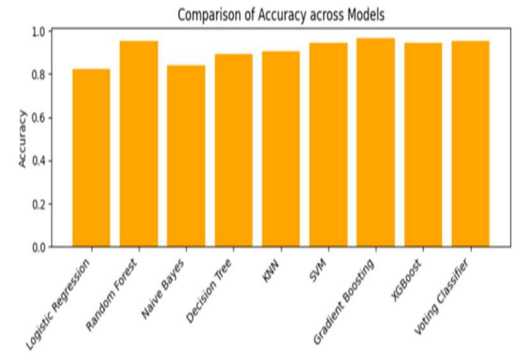
D. Inference Latency and Model Size

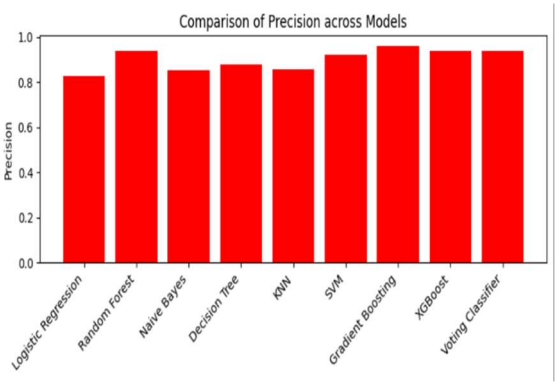
The size of the data set and the model configurations are moderate. Training time is acceptable for local experimentation; all models complete training on a standard laptop in practical time. For a single student record, prediction is very fast (on the order of milliseconds) on local tests. Model sizes for Random Forest and XGBoost are larger than Logistic Regression but still feasible to store and load within typical server memory constraints. Therefore, inference in real-time through a web API is realistic.

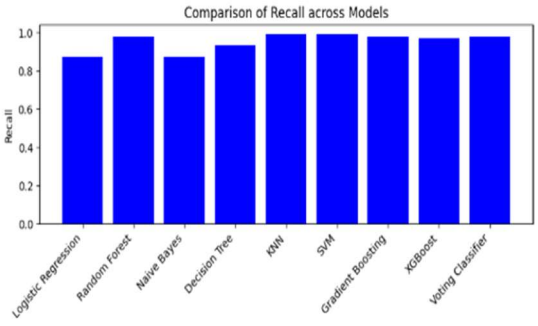


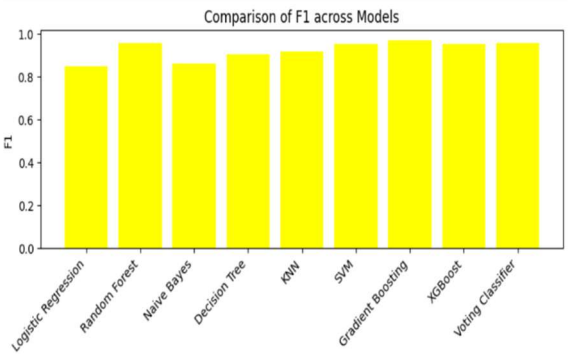
	Model	Accuracy	Precision	Recall	F1	ROC-AUC
0	Logistic Regression	0.824242	0.824742	0.869565	0.846561	0.886837
1	Random Forest	0.951515	0.937500	0.978261	0.957447	0.990768
2	Naive Bayes	0.842424	0.851064	0.869565	0.860215	0.940441
3	Decision Tree	0.890909	0.877551	0.934783	0.905263	0.900610
4	KNN	0.903030	0.858491	0.989130	0.919192	0.952501
5	SVM	0.945455	0.919192	0.989130	0.952880	0.980494
6	Gradient Boosting	0.963636	0.957447	0.978261	0.967742	0.989130
7	XGBoost	0.945455	0.936842	0.967391	0.951872	0.990917
8	Voting Classifier	0.951515	0.937500	0.978261	0.957447	0.986748











VIII Current Capabilities, Limitations, and Scope of Work

A. Current Capabilities

The project at its current stage provides:

- Cleaned and Pre-processed Dataset with structured features ready for ML.
- Training and Evaluation of Nine ML Models with consistent metrics and visual comparisons.
- Identification of Strong Models such as Random Forest and tree-based ensembles.
- Confusion Matrices, ROC Curves, and Metric Plots generated directly from notebook output.
- A Clear Blueprint for Web Integration.

B. Caveats and Limitations

Despite these achievements, several limitations remain:

- Dataset Size and Diversity: The dataset is not extremely large and may not represent all types of schools, regions, or socio-economic contexts. This limits generalization.
- Feature Coverage: Important factors such as detailed attendance records, psychological indicators, parental involvement, and previous exam history are not included.
- Class Imbalance: Dropout cases are fewer compared to enrolled students, which can bias models toward predicting the majority class.
- Interpretability for Non-Technical Users: Some strong models (XGBoost, Voting Classifier) are complex and hard to interpret without additional tools.
- Deployment Status: The web application and CI/CD pipeline are currently conceptual. A fully hosted, secure, and monitored system was not deployed during this phase.

C. Importance and Applicability

Even with these limitations, the system is meaningful and applicable. It shows how secondary school data can be used to support objective decisions about student support. It can help institutions design early-warning systems, focusing limited counselling resources on the most at-risk students. With adaptation and re-training, the pipeline could also be used in colleges, online courses, and other educational platforms.

IX. Research Extension and Scalability

Based on the current work and literature, several extensions are possible:

1) Data Expansion and Enrichment: Integrate attendance logs, continuous assessment scores, behaviour reports, and psychological surveys to improve prediction.
2) Imbalance Handling and Advanced Optimization: Apply SMOTE, ADASYN, or class-balanced loss functions to better handle minority dropout cases.
3) Explainable AI (XAI): Use SHAP or LIME to provide instance-level explanations for predictions, highlighting which features contributed most to a high dropout risk.
4) Temporal and Sequential Modelling: Extend the system to use semester-wise or week-wise data, enabling early and dynamic risk estimation using LSTMs or attention-based models.
5) Scalable Web Deployment: Deploy the service using containers (Docker) and cloud platforms, enabling multiple schools to access a shared but secure prediction service.
6) Cross-Institutional Validation: Test the model on datasets from different states, boards, or countries to measure robustness and ensure generalization.

These directions would transform the current project from a prototype into a production-ready tool for educational analytics.

X. Challenges Faced and Solutions

A. Data Quality and Pre-processing

Challenge: The raw dataset contained missing values and duplicate entries, which could bias model training and evaluation.

Solution: Implemented a comprehensive pre-processing pipeline including duplicate removal, median imputation for numeric features, and stratified train-test splitting to preserve class distribution. This ensured data quality and reduced the risk of information leakage.

B. Class Imbalance

Challenge: The number of dropout cases was significantly lower than enrolled students, leading to potential bias toward the majority class.

Solution: Applied class weighting in algorithms that support it (Logistic Regression, Random Forest, Decision Tree, SVM, XGBoost) to penalize misclassification of the minority class. Additionally, stratified splitting was used during train-test division.

C. Feature Selection and Engineering

Challenge: Not all columns in the dataset were useful for prediction; some were redundant or added noise.

Solution: Dropped columns like Teaching_Staff and Dropout_Reason after initial exploration. Applied label encoding for categorical variables and StandardScaler for numerical features to normalize the input space.

D. Model Complexity vs. Interpretability

Challenge: While ensemble models (XGBoost, Voting Classifier) achieved high accuracy, they were harder to interpret for non-technical educators.

Solution: Included both simple and complex models in the comparison. Random Forest, which is more interpretable yet powerful, was identified as the top performer. Future work suggests SHAP/LIME for explaining complex model predictions.

E. Hyper parameter Tuning

Challenge: Finding optimal hyper parameters for nine different algorithms was computationally intensive and required domain knowledge.

Solution: Used empirically validated hyper parameters based on literature and conducted focused tuning on top-performing models. Random Forest, XGBoost, and Voting Classifier were fine-tuned for better performance.

XI. Conclusion

This paper presented an end-to-end supervised machine learning approach for predicting dropout risk among secondary school students, starting from a structured SIH-style dataset and ending with a conceptual full-stack deployment design. After cleaning, encoding, scaling, and stratified splitting, nine classifiers were trained and evaluated using accuracy, precision, recall, F1-score, ROC-AUC, confusion matrices, and ROC curves.

Tree-based ensemble models, especially Random Forest and Gradient Boosting/XGBoost, showed strong and balanced performance across metrics, which matches patterns reported in related work on educational data mining. The analysis also highlighted the importance of socio-demographic and academic variables such as school type, socio-economic status, and grade level for dropout risk.

Although the current system has limitations in data size, feature coverage, and full deployment, it demonstrates a practical B.Tech-level implementation of an AI-powered early-warning tool for education. With future work on data enrichment, imbalance handling, explainable AI, and scalable web deployment, this research has strong potential to support real-world interventions and improve student retention in educational institutions.

Acknowledgment

The authors express sincere gratitude to their project mentor Geethika Sethi Ma’am for continuous guidance, feedback, and academic support during all phases of this work. Special thanks are also extended to classmates and friends from the 3rd year B.Tech CSE batch at Lovely Professional

University for helpful discussions, motivation, and informal testing of intermediate versions of the system. Finally, appreciation is given to the open-source community and maintainers of Python, Scikit-learn, XGBoost, Pandas, Matplotlib, and Seaborn libraries, whose tools made this project practically possible.

REFERENCES

[1] J. M. Aguilar-Vera et al., "Predicting Student Dropout based on Machine Learning and Deep Learning: A Systematic Review," *EAI Endorsed Transactions on Scalable Information Systems*, 2023.

[2] A. M. Shahiri, W. Husain, and N. A. Rashid, "Utilizing Random Forest Algorithm for Early Detection of At-Risk Students in Open Learning Environments," 2023.

[3] Y. K. Turel and colleagues, "Predicting Dropout Student: An Applica-¨ tion of Data Mining in Online Education," *Educational Technology & Society*.

[4] "Enhancing the Early Student Dropout Prediction Model Through Clustering Analysis of Students' Digital Traces," *IEEE*, 2024.

[5] D. Plua et al., "Identifying Key Factors of Student Dropout through Random Forest," *Atlantis Press*.

[6] G. Colpo et al., "Educational Data Mining for Dropout Prediction: Trends, Opportunities and Challenges," 2024.

[7] "Predicting Student Dropout Risk in Online Learning Using Stacked Ensemble Models," *International Journal of Computer Applications*, 2025.

[8] F. Torres-Cruz, "Prediction of University Dropouts Through Random Forest," *Journal of Applied and Emerging Sciences*.

[9] O. Goren et al., "Early Prediction of Student Dropout in Higher Education Using Machine Learning," *EDM* 2024.

[10] "Student Dropout Prediction Through Machine Learning Approaches Using LMS Data," *Frontiers in Education*, 2025.

[11] P. A. Sacco et al., "Educational Data Mining: A Survey," *IEEE Transactions on Learning Technologies*, vol. 5, no. 2, pp. 146–159, 2012.

[12] M. Hussain et al., "Student Engagement Prediction with Deep Learning Using Clickstream Data," *Education and Information Technologies*, vol. 23, no. 5, pp. 2015–2028, 2018.

[13] G. Siemens and R. S. J. d. Baker, "Learning Analytics and Educational Data Mining: towards Communication and Collaboration," in *Proc. 2nd International Conference on Learning Analytics and Knowledge*, 2012, pp. 252–254.

[14] C. Romano et al., "A Systematic Literature Review on Machine Learning Applications for Educational Institutions," *IEEE Access*, vol. 8, pp. 128514–128540, 2020.