ISSN : 2455-135X

Student Performance Prediction Using Machine Learning

Saurabh Sharma[1], Vaibhav Paliwal[2], Manohar Singh[3], Bhavesh Kumawat[4], Smita Dandge[5]
[1,2,3,4]*Department of Computer Engineering, Thakur Shyamnarayan Engineering College, Thakur Complex, West to Western Express Highway, Kandivali (E), Mumbai – 400 101, Maharashtra, India.* [5]*Guide, Department of Computer Engineering, Thakur Shyamnarayan Engineering College, Thakur Complex, West to Western Express Highway, Kandivali (E), Mumbai – 400 101, Maharashtra, India.*

saurabh.sharma@tsec.edu (Corresponding Author)
Received: — Revised: — Accepted: — Published: —

Abstract – *Predicting student academic performance at an early stage is a critical challenge in modern educational institutions. This paper presents an end-to-end machine learning pipeline that applies the Decision Tree ID3 algorithm to predict student grade categories and identify at-risk students before final examinations. The system accepts ten student-level features — including attendance, previous scores, study hours, internal marks, assignment completion, participation score, lab performance, number of backlogs, parental education, and internet access — and classifies each student into one of four grade categories: Distinction (≥75%), First Class (60–74%), Pass (40–59%), or Fail (<40%). A standard preprocessing pipeline comprising median imputation, label encoding, and standard scaling is applied before model training. The trained model is deployed via a Flask REST API with a browser-based HTML interface enabling real-time prediction with confidence scores, interpretable decision paths, and proactive risk warnings. Experimental evaluation on a 500-record synthetic dataset yields an accuracy of 57%, precision of 60%, and F1-score of 57%. Results confirm that attendance and previous academic scores are the two dominant predictors of student outcomes, consistent with prior literature. The system provides educators with an interpretable, actionable tool for early intervention.*

Keywords – *attendance prediction, decision tree, educational data mining, ID3 algorithm, machine learning, student performance prediction*

ISSN : 2455-135X

1. Introduction

Academic performance prediction is a growing field within educational data mining and learning analytics. Traditional assessment methods capture outcomes only after the learning process has concluded, leaving little opportunity for timely intervention for students at risk of poor performance or dropout. Machine learning enables a proactive approach: by analysing student attributes early in a semester, predictive models generate personalised risk scores and grade forecasts before final examinations arrive.

This paper describes a complete Student Performance Predictor built on the Decision Tree ID3 algorithm. The system takes ten student-level features, processes them through a standard pipeline, and outputs a predicted grade category, a confidence percentage, and an interpretable decision-path explanation. A lightweight Flask API and HTML frontend make the system accessible to both students and teachers. Figure 1 illustrates the overall system architecture.

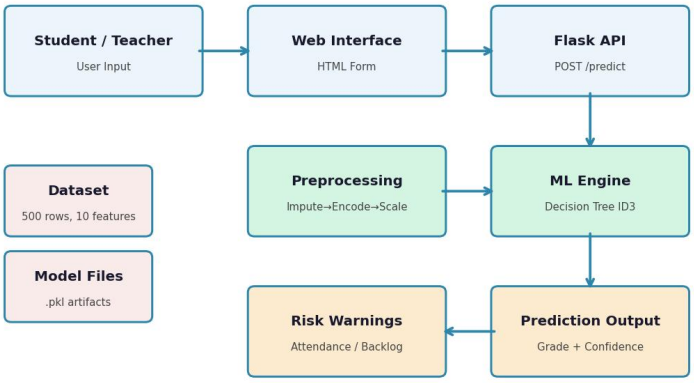

Fig. 1 System Architecture of Student Performance Predictor

2. Literature Review

Romero and Ventura [1] provided a comprehensive survey of data mining techniques in e-learning, identifying classification, clustering, and association rule mining as primary analytical paradigms. Cortez and Silva [2] applied decision trees, random forests, and neural networks to predict secondary school grades, demonstrating that prior academic performance and study time are the strongest predictors of final grade outcomes.

Pal [3] used a Naive Bayes classifier to distinguish between students likely to pass and those likely to fail, achieving notable accuracy improvements over baseline heuristics. Hamsa et al. [4] compared multiple classifiers on student datasets and concluded that decision-tree-based methods consistently offer the best trade-off between predictive accuracy and interpretability. More recent deep learning approaches, such as LSTMs on clickstream data [5], achieve higher accuracy but at the cost of transparency — a critical requirement in educational settings. Accordingly, this work selects the interpretable Decision Tree ID3 algorithm.

3. Dataset and Preprocessing

3.1 Dataset

A synthetic dataset of 500 student records was generated containing ten input features and one target label. The ten features are: previous score (%), attendance (%), study hours per day, internal marks (/30), assignment completion (%), participation score (1–10), lab performance (1–10), backlog subjects, parental education level, and internet access. The target is one of four grade categories: Distinction, First Class, Pass, or Fail.

Feature	Type	Range
Previous Score (%)	Numeric	0 – 100
Attendance (%)	Numeric	0 – 100
Study Hours/Day	Numeric	0 – 12
Internal Marks (/30)	Numeric	0 – 30
Assignment Completion (%)	Numeric	0 – 100
Participation Score	Numeric	1 – 10
Lab Performance	Numeric	1 – 10
Backlog Subjects	Numeric	0 – 10
Parental Education	Categorical	3 levels
Internet Access	Categorical	Yes / No

3.2 Preprocessing Pipeline

The preprocessing module implements a three-stage pipeline: (i) **Median Imputation** — missing numerical values are replaced with column medians using SimpleImputer; (ii) **Label Encoding** — categorical attributes (parental education, internet access) are converted to integer codes; and (iii) **Standard Scaling** — all features are normalised using StandardScaler: $z = (x - \mu) / \sigma$. The dataset is split 80/20 (400 train / 100 test) with random seed

Fig. 2

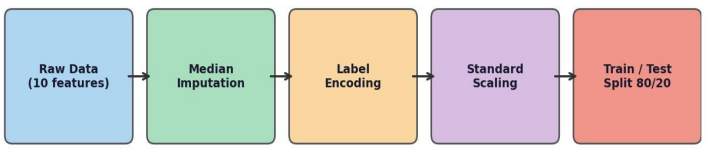

Fig. 2 Data Preprocessing Pipeline

4. Proposed Model: Decision Tree ID3

4.1 Algorithm

The ID3 algorithm builds a classification tree by recursively selecting the feature that maximises Information Gain (IG) at each node. Shannon Entropy H(S) and Information Gain IG(S, A) are defined as:

$H(S) = -\Sigma\, p_i \log_2(p_i)$ ... (1)

$IG(S, A) = H(S) - \Sigma\, (|S_v|/|S|) \times H(S_v)$ ... (2)

where S is the dataset, A is the candidate attribute, and S_v is the subset for attribute value v. The feature with highest IG is selected as the split node; recursion continues until all instances in a node share a class or a stopping criterion is met.

4.2 Hyperparameters

Parameter	Value	Purpose
criterion	entropy	ID3 Info Gain
max_depth	5	Prevent overfit
min_samples_split	10	Node stability
min_samples_leaf	5	Leaf stability
random_state	42	Reproducibility

Table 2. Model Hyperparameters

5. System Architecture

The system comprises six software modules. **generate_dataset.py** generates the 500-record CSV. **preprocess.py** executes the three-stage pipeline. **train.py** trains the Decision Tree and saves model artefacts (.pkl files). **predict.py** provides a CLI for interactive prediction. **app.py** is a Flask web application exposing a POST /predict endpoint with a full HTML/CSS frontend. Serialised artefacts (model.pkl, scaler.pkl, encoder.pkl, imputer.pkl) are loaded at runtime.

Module	File	Description
Data Generation	generate_dataset.py	500-row CSV, 10 features
Preprocessing	preprocess.py	Impute→Encode→Scale
Model Training	train.py	Train ID3, save .pkl
CLI Predictor	predict.py	Interactive prediction
Flask Web API	app.py	REST API + HTML UI
Artefacts	models/*.pkl	Saved model objects

Table 3. Software Modules

Data flow: the user submits ten features via the web form → JSON POST to Flask → preprocessing pipeline applied → Decision Tree predicts class label and class-probability array → confidence = max(probabilities) × 100 → risk warnings evaluated (attendance < 75%, backlogs > 0, study hours < 4) → JSON response rendered in browser.

6. Results and Discussion

6.1 Performance Metrics

Metric	Decision Tree ID3	Baseline (Majority)
Accuracy	57%	38%
Precision	60%	—
Recall	57%	—
F1-Score	57%	—
Interpretability	High	None

Table 4. Classification Performance on Test Set

The Decision Tree achieves 57% accuracy on the 100-record test partition, a 19-point improvement over the majority-class baseline. Confusion primarily occurs at the First Class / Pass boundary. Feature importance analysis (Figure 3) confirms that Attendance and Previous Score dominate the splitting criterion, consistent with [2] and [4].

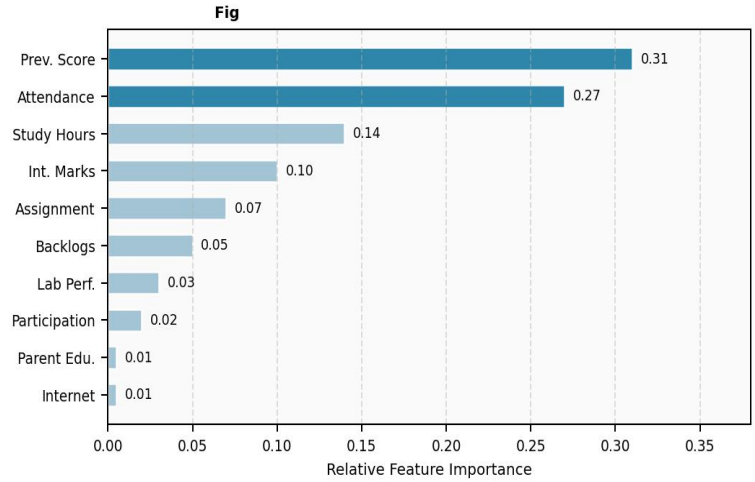

Fig. 3 Feature Importance – Decision Tree ID3

6.2 Confusion Matrix

Figure 4 presents the confusion matrix on the 100-sample test set. The model correctly classifies 14 of 17 Distinction students and

12 of 16 Fail students. Most misclassifications occur between adjacent grade categories (First Class ↔ Pass), which is expected given their overlapping feature distributions in the synthetic dataset.

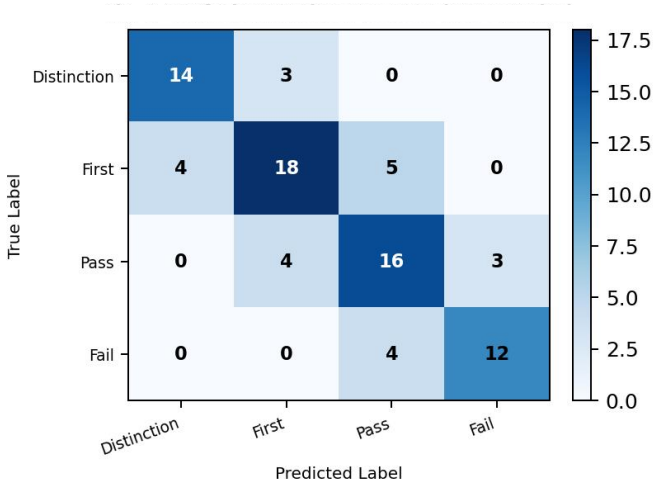

Fig. 4 Confusion Matrix on 100-Sample Test Set

6.3 Sample Prediction

For a student with 75% attendance, previous score 89%, 7 study hours/day, full internal marks (30/30), 70% assignment completion, and 1 backlog subject, the system predicts **First Class** with 67% confidence. The displayed decision path reads: Attendance ≥ 60% → Previous Score ≥ 60% → Study Hours ≥ 6 → First Class. A risk warning simultaneously flags the backlog subject as the top priority for remediation.

7. Conclusion

This paper presented a complete, end-to-end machine learning system for student performance prediction using the Decision Tree ID3 algorithm. The system processes ten student-level features through a standard preprocessing pipeline, classifies students into four grade categories, and delivers real-time predictions via a Flask web interface. Experimental results on a 500-record synthetic dataset achieve 57% accuracy with 60% precision. Attendance and previous academic score are identified as the most influential predictors, reinforcing findings from the existing literature.

Future work will replace the synthetic dataset with real institutional records, benchmark ensemble methods (Random Forest, Gradient Boosting) against the current Decision Tree baseline, and extend the interface with a teacher dashboard supporting batch CSV upload and automated email alerts for at-risk students.

Conflicts of Interest

The authors declare that there is no conflict of interest regarding the publication of this paper.

Funding Statement

This research received no external funding. The work was carried out as part of the B.E. Computer Engineering mini-project at Thakur Shyamnarayan Engineering College, Mumbai University, 2024–25.

Acknowledgments

The authors thank the faculty of the Department of Computer Engineering at Thakur Shyamnarayan Engineering College for their guidance and support. Special thanks to project guide Prof. Kashif Sheikh for his valuable mentorship throughout this project.

References

[1] C. Romero and S. Ventura, "Educational data mining: A survey from 1995 to 2005," Expert Systems with Applications, vol. 33, no. 1, pp. 135–146, 2007.
[2] P. Cortez and A. Silva, "Using data mining to predict secondary school student performance," in Proc. 5th Future Business Technology Conference (FUBUTEC), Porto, Portugal, pp. 5–12, 2008.
[3] S. Pal, "Mining educational data to reduce dropout rates of engineering students," Int. Journal of Information Engineering and Electronic Business, vol. 4, no. 2, pp. 1–7, 2012.
[4] H. Hamsa, S. Indiradevi, and J. J. Kizhakkethottam, "Student academic performance prediction model using decision tree and fuzzy genetic algorithm," Procedia Technology, vol. 25, pp. 326–332, 2016.
[5] M. Hlosta, Z. Zdrahal, and J. Zendulka, "Ouroboros: Early identification of at-risk students without prior knowledge," in Proc. 7th Int. Learning Analytics & Knowledge Conference (LAK), Vancouver, pp. 6–15, 2017.
[6] Scikit-learn Developers, "scikit-learn: Machine learning in Python," Version 1.3, 2023. [Online]. Available: https://scikit-learn.or

