Student Performance Prediction Using Machine Learning

Anuja Medhe[1], Sakshi Mahale[2], Smita Ghorpade[3]
[1,2]*Student, KRT Arts BH Commerce AM Science College Nashik, Maharashtra, India*
[3]*Assistant Professor, KRT Arts BH Commerce AM Science College Nashik, Maharashtra, India*
doi.org/10.64643/IJIRT211376

***Abstract*—Predicting student academic performance can help educators identify learners who may benefit from timely academic support. This study compares five supervised machine-learning classifiers—Logistic Regression, Decision Tree, Random Forest, K-Nearest Neighbors (KNN), and Support Vector Machine (SVM)—using the mathematics portion of the UCI Student Performance dataset. The target variable, final grade (G3), is grouped into three classes: Low (0–9), Average (10–13), and High (14–20). Two experiments are conducted: Experiment A includes the earlier period grades G1 and G2, while Experiment B excludes them to examine prediction using other available attributes. The supplied experimental output reports 395 records and a class distribution of 130 Low, 165 Average, and 100 High instances. In Experiment A, Random Forest achieved the highest test accuracy (86.08%) and Macro F1-score (0.8684), with a cross-validation Macro F1 mean of 0.8479. In Experiment B, SVM achieved the highest test accuracy (53.16%) and Macro F1-score (0.5222), while Logistic Regression had a test Macro F1-score of 0.4813. Overall, removing G1 and G2 substantially reduced performance, indicating that earlier grades contain strong predictive information in this dataset. Permutation-importance output identified failures as the most positively important feature in Experiment B, although most reported importance values were close to zero or negative and should be interpreted cautiously. The results are specific to this dataset and require validation on independent student populations before practical deployment.**

***Index Terms*—student performance prediction, educational data mining, machine learning, classification, Random Forest, SVM, Macro F1-score, feature importance**

I. INTRODUCTION

Educational institutions collect different types of student information, including academic grades, attendance, study habits, and demographic attributes. Analysing these data can help identify patterns associated with academic outcomes. Educational data mining applies computational and statistical methods to educational data to support the study of learning and educational processes.

Student performance prediction is useful when it provides timely information for supportive educational decisions. A predictive model may help flag students who could benefit from tutoring, feedback, or additional academic guidance. However, model predictions should not be treated as permanent labels or as replacements for teachers' judgement. Predictions can be inaccurate, and their reliability depends on the dataset, the definition of the target, the quality of the features, and the evaluation procedure. This study uses the UCI Student Performance dataset described by Cortez and Silva [1]. The dataset includes student-related variables and grades from Portuguese secondary schools. The final grade, G3, is the target for the proposed classification task. Because earlier grades G1 and G2 are closely related to the final grade, their inclusion may make prediction easier but may be less useful for an early-warning setting in which such grades are unavailable. The study therefore compares two feature settings: with G1 and G2, and without G1 and G2. The main contribution of this project is a reproducible comparison of five commonly used classification algorithms using the same data split, preprocessing pipeline, cross-validation strategy, and evaluation metrics. The study also examines feature importance when earlier grades are excluded. It does not assume in advance that a particular algorithm will be the best; conclusions will be based on the observed results.

1.1. Problem Statement

Different Machine Learning algorithms can produce different predictions, and results reported across separate studies are difficult to compare when the datasets, target definitions, and evaluation metrics differ. In addition, a model that relies heavily on

previous grades may be less suitable for identifying students at risk before those grades are available. This research addresses these issues by comparing five classification algorithms under a consistent experimental setup and by evaluating the effect of excluding G1 and G2.

1.2. Research Objectives

1. To develop classification models for predicting student performance as Low, Average, or High.
2. To compare Logistic Regression, Decision Tree, Random Forest, KNN, and SVM using consistent evaluation procedures.
3. To investigate the effect of previous grades (G1 and G2) on model performance.
4. To evaluate models using Accuracy, Macro Precision, Macro Recall, Macro F1-score, Weighted F1-score, and Balanced Accuracy.
5. To identify useful features in the experiment that excludes previous grades.
6. To discuss the responsible use and limitations of student performance predictions.

II. LITERATURE REVIEW

Student performance prediction has become an important research area in educational data mining. Machine Learning techniques can be used to analyze students' academic records and identify patterns associated with their performance. Researchers have explored different algorithms, including Decision Tree, Random Forest, Support Vector Machine (SVM), Logistic Regression, and K-Nearest Neighbors (KNN), for predicting student outcomes.

Previous studies have investigated the use of academic scores, attendance, study habits, and other student-related attributes to predict academic performance. However, the effectiveness of a model may vary depending on the dataset, selected features, preprocessing techniques, and evaluation methods.

A comparative analysis of existing studies can help identify suitable algorithms, understand the limitations of previous approaches, and determine the research gap addressed by the present study.

Kalmani et al. (2023) proposed a Student Performance Prediction System using machine learning. The study examined approximately 1,023 student records and compared techniques including XGBoost, Random Forest, and Support Vector Machine. The research also described the deployment of a prediction system using Flask. The reported results included 95% accuracy for XGBoost in one section, while other reported results included 92.64% for Random Forest and 92.18% for SVM. The study demonstrates the potential of machine learning for academic prediction, although the reported scores require verification against the original experimental tables.

Rathod and Vora (2025) conducted a study titled Predictive Analysis of Student Performance using Machine Learning Models. Their dataset contained 174 student records collected through Google Forms. The features included weekly study hours, attendance, previous grades, extracurricular activities, participation, and parental education. Random Forest and Gradient Boosting reportedly achieved 97% accuracy, while XGBoost achieved 94% and Support Vector Classifier achieved 80%. The study highlights the usefulness of academic and behavioural features, but its relatively small dataset limits the generalisation of the findings.

Thorat et al. (2025) investigated student performance prediction using supervised machine learning techniques. The reported comparison included XGBoost, Random Forest, Decision Tree, SVM, and a logistic-regression-based model. Their comparative results reported accuracy values of 95.1% for XGBoost, 92.3% for Random Forest, 89.2% for Decision Tree, 88.7% for SVM, and 84.5% for the logistic-regression-based model. The study also discussed data privacy, algorithmic bias, and interpretability as important challenges.

Rohit M. N., Mohan, and Vishwakiran (2025) focused on predicting student performance in programming education. Their work discussed the use of educational data mining, decision trees, Random Forest, SVM, neural networks, and learning-management-system data. Potential features included attendance, assignment scores, quiz results, online participation, programming exercise attempts, debugging behaviour, and task-completion time. The supplied review describes this work primarily as a review or proposed research direction rather than a completed experiment with a clearly reported final accuracy.

Vamsi and Bidikar (2025) presented a data-driven approach to academic success forecasting using the Portuguese Student Performance dataset. The study used a Multilayer Perceptron Regressor to predict the

final grade, G3. The described methodology included missing-value treatment, categorical encoding, feature scaling, and Principal Component Analysis. Flask and Gradio interfaces were also discussed for deployment. The supplied review identifies MAE, MSE, and R^2 as evaluation measures but does not provide clearly extractable numerical results, so the model's performance cannot be ranked reliably from the supplied document.

Aher (2026) compared Decision Tree and Random Forest classifiers using 200 student records. The features included internal examination marks, study time, absences, and final marks. Random Forest reportedly achieved 88.33% accuracy, compared with 81.66% for Decision Tree. The study suggests that combining multiple decision trees can improve prediction compared with a single tree. However, the small dataset and limited evaluation metrics restrict the strength of the conclusions.

Kumari and Khorwal (2026) proposed a machine learning approach for predicting academic performance using attendance percentage, daily study hours, previous examination marks, assignment submission rate, and class participation. Their system classified student outcomes into Pass, Average, and Fail categories. Random Forest reportedly achieved 91% accuracy, Decision Tree achieved 85%, and Linear Regression was reported at 78%. The study identified previous marks and attendance as important predictors. However, the small test set and uneven class distribution indicate that additional evaluation using recall, F1-score, and confusion matrices would be useful.

Dev, Kumar, and Gupta (2026) compared Random Forest, Decision Tree, SVM, KNN, and Naïve Bayes for predicting undergraduate engineering students' performance. The study considered academic, behavioural, psychological, and engagement-related features and used SHAP to explain feature contributions. Random Forest reportedly achieved 91% accuracy, followed by SVM at 86%, KNN at 84%, Decision Tree at 81%, and Naïve Bayes at 76%. Assignment submission rate, laboratory attendance, and daily self-study hours were identified as influential features. The study demonstrates the value of explainability, although feature importance does not establish a causal relationship.

Mahesh and Harwalkar (publication year not established in the supplied document) investigated student performance prediction in higher education using records associated with IIT Roorkee. The dataset covered more than 1,000 students from academic years 2006–2017. Random Forest, SVM, XGBoost, and Artificial Neural Networks were compared. The supplied review reported test accuracies of 91.2% for Random Forest, 87.6% for SVM, 92.4% for XGBoost, and 93.8% for the neural network. Backlog count, attendance, and previous-semester GPA were identified as influential features. However, the publication details need verification, and independent external validation was not reported in the supplied review.

Sharma et al. (2026 issue; year metadata requires verification) presented an ID3 Decision Tree approach for classifying student performance into four grade categories. The study used a synthetic dataset containing 500 records. It reportedly achieved 57% accuracy, 60% precision, and a 57% F1-score, compared with 38% accuracy for a majority-class baseline. The work illustrates the benefit of interpretable decision rules, but its synthetic dataset and modest performance limit conclusions about real-world effectiveness.

Dheeraj et al. (2026) evaluated six classification approaches for predicting students' academic performance using 649 secondary-school records. The study considered Logistic Regression, Decision Tree, SVM, AdaBoost, XGBoost, and a bagged tree ensemble. A stacking approach used Logistic Regression as a meta-learner. The stacker reportedly achieved 93% test accuracy, compared with 92% for the bagged ensemble and 91% for XGBoost. A Flask-based application was also described. The study provides a comparison of several models, although validation on independent datasets remains an important requirement.

Vashist and Singh (2026) proposed EduXplain Pro, an explainable ensemble framework for student performance prediction. The framework used a synthetic dataset of 20,000 records and compared Logistic Regression, Random Forest, XGBoost, LightGBM, and CatBoost. A soft-voting ensemble and SHAP explanations were incorporated, with stratified five-fold cross-validation described in the supplied review. The ensemble reportedly achieved 91.8%

accuracy, an AUC-ROC of 0.926, precision of 0.931, recall of 0.954, and an F1-score of 0.942. A Streamlit dashboard supported presentation of the results. However, synthetic data do not establish the same performance on real student records.

Anusha Rani et al. (2026) examined student performance prediction using Gradient Boosting and SHAP analysis. The supplied review describes an approximately 80,000-record dataset with 31 features, including study time, attendance, sleep time, internet usage, parental assistance, and test performance. Gradient Boosting reportedly achieved 92% accuracy, 0.90 precision, 0.91 recall, and an F1-score of 0.905. Attendance percentage and internal assessment scores were identified as important features. The results suggest that feature interpretation can complement predictive modelling, although the reliability of the conclusions depends on data quality and validation.

Thakur et al. (2026) proposed a Random Forest regression approach using the UCI Student Performance dataset to predict the numerical final grade, G3, on a scale from 0 to 20. The features included demographic and family information, previous academic performance, study habits, absences, and behavioural variables. The supplied review reported an RMSE of 4.21 and an R^2 of 0.86. A Flask web interface was described for displaying predicted grades. Since this is a regression task, its results should be evaluated using regression metrics rather than compared directly with classification accuracy.

III. RESEARCH METHODOLOGY

This study follows a quantitative experimental design. A publicly available secondary dataset is processed, transformed into a classification target, and used to train and evaluate five supervised learning algorithms. All models are assessed using a common train-test split and a consistent set of evaluation metrics.

3.1 Dataset Description

The proposed dataset is the Student Performance dataset from the UCI Machine Learning Repository [1]. The mathematics file, student-mat.csv, is used for the initial experiment. It contains student attributes and grades, with G3 representing the final mathematics grade on a 0–20 scale. The dataset was collected from Portuguese secondary schools; therefore, the results should not automatically be generalised to Indian colleges or other educational systems.

3.2 Target Variable and Categories

The original target G3 is a numerical grade. For this classification experiment, it is converted into three project-defined categories: Low for grades 0–9, Average for grades 10–13, and High for grades 14–20. These thresholds are defined for this study and are not official categories supplied by UCI.

3.3 Input Features and Leakage Prevention

The target column G3 and the derived Performance label are excluded from the input features. Experiment A includes G1 and G2. Experiment B excludes both G1 and G2 while retaining the other available variables. The same stratified training and testing indices are used for both experiments so that the comparison is easier to interpret.

Preprocessing is performed inside a scikit-learn pipeline. Numeric features are imputed using the median and standardised. Categorical features are imputed using the most frequent value and transformed using one-hot encoding. Fitting preprocessing only on each training fold reduces the risk of information leakage from validation or test data.

3.4 Algorithms

Five classification algorithms are evaluated:

- Logistic Regression: a linear baseline classifier.
- Decision Tree: learns rule-based splits from input features.
- Random Forest: combines predictions from multiple decision trees.
- K-Nearest Neighbors (KNN): predicts from nearby examples in feature space.
- Support Vector Machine (SVM): learns a decision boundary that separates classes.

The implementation uses scikit-learn. Class weighting is enabled for Logistic Regression, Decision Tree, Random Forest, and SVM as a potential response to class imbalance. KNN does not use the same class_weight parameter; therefore, class-wise metrics remain important when interpreting its performance.

3.5 Experimental Design

The dataset is divided into 80% training data and 20% testing data using stratified sampling with a fixed

random seed (42). Five-fold StratifiedKFold cross-validation is performed on the training data. The mean cross-validation Macro F1-score is used to select a model within each experiment. The held-out test set is then used for final evaluation and is not used to select the best model.

Experiment A (With G1 and G2): all eligible features are used except G3 and the derived target label.
Experiment B (Without G1 and G2): G1 and G2 are removed from the eligible input features.
The experiment is intended to quantify how the availability of earlier grades changes prediction quality. Because G1 and G2 are earlier grades from the same academic year, Experiment A may reflect a later prediction setting rather than a genuinely early-stage warning system.

3.6 Evaluation Metrics
Accuracy is the proportion of correctly classified records. Precision measures the proportion of correct predictions among records predicted as a particular class. Recall measures the proportion of actual members of a class that are correctly identified.
Macro F1-score computes the F1-score for each class and gives each class equal weight. Weighted F1-score weights class-level F1-scores by the number of observations in each class. Balanced Accuracy averages recall across classes. A confusion matrix is used to inspect the types of classification errors. Macro F1-score is the primary model-selection metric because it gives equal weight to Low, Average, and High categories. Accuracy is also reported but is not used alone. Cross-validation mean and standard deviation are reported to show average training-fold performance and its variability.

3.7 Feature Importance
Permutation importance is calculated for the selected Experiment B model on the held-out test set using Macro F1-score. For each feature, its values are shuffled and the change in model performance is measured. A larger decrease in Macro F1-score indicates that the model relies more on that feature for its predictions under this evaluation setup. Importance values may be close to zero or negative and should be interpreted with caution, especially when features are correlated. Feature importance is not evidence of causality.

3.8 Tools and Environment
The implementation uses Python, Pandas, NumPy, scikit-learn, Matplotlib, and Seaborn. Google Colab or Jupyter Notebook can be used as the execution environment. The Python version and library versions should be recorded from the actual environment before final submission for reproducibility.

IV. RESULTS AND DISCUSSION

4.1 Dataset Description
The dataset used in this study contains 395 student records and 33 features, including demographic, family, lifestyle, and academic information. The target variable, Performance, was divided into three categories: Low, Average, and High. The dataset contains 130 students in the Low category, 165 students in the Average category, and 100 students in the High category.

4.2 Model Performance Comparison
Five machine learning algorithms were evaluated: Logistic Regression, Decision Tree, Random Forest, K-Nearest Neighbors (KNN), and Support Vector Machine (SVM). Two experiments were conducted. Experiment A included the previous academic grade features G1 and G2, whereas Experiment B excluded these features.

In Experiment A, Random Forest achieved the highest accuracy of 86.08% and a macro F1-score of 0.8684. Logistic Regression achieved an accuracy of 84.81%, followed by Decision Tree at 83.54%, SVM at 78.48%, and KNN at 70.89%.
In Experiment B, where G1 and G2 were excluded, SVM achieved the highest accuracy of 53.16%, followed by Logistic Regression at 48.10%, KNN at 44.30%, Random Forest at 41.77%, and Decision Tree at 39.24%.
These results indicate that previous academic grades contributed substantially to student performance prediction in the selected dataset.

4.3 Classification Report Analysis
The classification report for Experiment A showed that Random Forest achieved an F1-score of 0.85 for Low performance, 0.84 for Average performance, and 0.92 for High performance. Its precision for the High category was 0.95, with a recall of 0.90.

Logistic Regression also performed well, achieving an F1-score of 0.94 for Low performance, 0.81 for Average performance, and 0.79 for High performance. These findings suggest that Random Forest provided the strongest overall classification performance among the five evaluated models when G1 and G2 were included.

In Experiment B, the models showed substantially lower performance across the three categories, indicating that predicting student performance without previous academic grades was more challenging.

4.4 Confusion Matrix Analysis

The confusion matrix of the Random Forest model in Experiment A showed that 22 out of 26 Low-performance students, 28 out of 33 Average-performance students, and 18 out of 20 High-performance students were correctly classified.

The model misclassified four Low-performance students as Average, four Average-performance students as Low, one Average-performance student as High, and two High-performance students as Average. These results demonstrate that the Random Forest model classified the High-performance category particularly well, although some confusion remained between the Low and Average categories.

4.5 Feature Importance Analysis

In Experiment B, the feature importance analysis identified failures as the feature with the highest positive importance score of 0.0391. The next highest positive scores were obtained by romantic status and Pstatus, both with an importance score of 0.0105.

Several features had negative importance scores, including reason (-0.0515), absences (-0.0375), paid (-0.0290), and Walc (-0.0284).

Negative importance values should not be interpreted as proof that these features negatively affect student performance. Their interpretation depends on the importance method used and indicates that these features did not improve predictive performance under the evaluated procedure.

4.6 Discussion of Findings

The experimental results show a clear difference between the two feature settings. Random Forest achieved the best overall results when G1 and G2 were included, with an accuracy of 86.08%. Without these features, SVM achieved the highest accuracy of 53.16%.

The comparison suggests that previous academic grades are highly informative for predicting student performance in this dataset. However, prediction without these grades remains important because it may help identify students who need academic support before their grades are available.

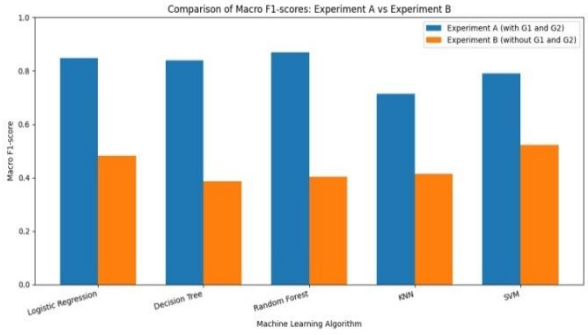
Further research could evaluate additional datasets, compare alternative feature-selection techniques, tune model hyperparameters, and assess whether the models generalize to students from other institutions. Future studies should also investigate interpretable prediction methods and evaluate the fairness of predictions across student groups.

Table

Model	Experiment A Accuracy	Experiment B Accuracy
Logistic Regression	84.81%	48.10%
Decision Tree	83.54%	39.24%
Random Forest	86.08%	41.77%
KNN	70.89%	44.30%
SVM	78.48%	53.16%

Graph:

Model Accuracy Comparison



V. LIMITATIONS AND ETHICAL CONSIDERATIONS

- The dataset represents Portuguese secondary-school mathematics students; results cannot automatically be generalized to Indian schools or colleges.
- The grade cutoffs (0–9, 10–13, 14–20) are researcher-defined and alternative thresholds may change class distribution and metrics.

- A single hold-out split can produce variable estimates. Repeated or external validation would provide stronger evidence.
- The supplied output does not include complete hyper parameter settings, software versions, full code, or all preprocessing details; these should be included for reproducibility
- Permutation importance values are exploratory and should not be interpreted as causal effects.
- Student data should be handled confidentially. Predictions should support, not replace, educator judgment and should not be used to stigmatize or restrict students.

VI. CONCLUSION

This research presents a comparative analysis of five machine learning algorithms for predicting student academic performance: Logistic Regression, Decision Tree, Random Forest, K-Nearest Neighbors (KNN), and Support Vector Machine (SVM). The experiments were conducted using a dataset containing 395 student records and 33 features. Student performance was classified into three categories: Low, Average, and High.

Two experiments were performed to evaluate the effect of previous academic grades on prediction accuracy. In Experiment A, which included the G1 and G2 grade features, the Random Forest classifier achieved the highest accuracy of 86.08%, with a macro F1-score of 0.8684. In Experiment B, where G1 and G2 were excluded, the Support Vector Machine achieved the highest accuracy of 53.16%.

The results indicate that previous academic grades are important predictive features in the selected dataset. Although the models performed better when these grades were included, predicting student performance without previous grades remains challenging. The feature importance analysis in Experiment B also identified previous failures as the feature with the highest positive importance score.

Overall, this study demonstrates the potential of machine learning techniques for student performance prediction. Such predictive models may help educational institutions identify students who could benefit from additional academic support. However, the results are based on a specific dataset, and further evaluation using additional datasets is necessary before applying the models in real educational environments.

VII. FUTURE SCOPE

1. In the future, this study can be conducted using a larger dataset of students.
2. More machine learning algorithms can be tested to improve prediction accuracy.
3. Additional factors, such as attendance, study time, and family support, can be used for prediction.
4. The model can be improved by tuning its parameters.
5. A simple web application can be developed to predict student performance.
6. In the future, this model can help teachers identify students who need extra support.

REFERENCES

[1] A. Rathod and N. Vora, “Predictive analysis of student performance using machine learning models,” *J. Inf. Technol. Digit. World*, vol. 7, no. 1, pp. 12–24, 2025, doi: 10.36548/jitdw.2025.1.002. [Online]. Available: https://irojournals.com/itdw/article/view/1427

[2] A. Dev, G. Kumar, and D. Gupta, “Student performance prediction using machine learning,” *Int. J. Multidiscip. Res.*, vol. 8, no. 3, 2026. [Online]. Available: https://www.ijfmr.com/research-paper.php?id=81587

[3] P. Thorat, P. Pareek, A. A. Khan, and R. S. Deshpande, “Student performance prediction using machine learning,” *Int. J. Creative Res. Thoughts*, vol. 13, no. 7, pp. b283–b286, 2025. [Online]. Available: https://ijcrt.org/viewpaperforall.php?paper=IJCRT2507142

[4] M. Anusha Rani, K. Bhagya Lakshmi, K. Bhargavi Latha Reddy, S. Nikhat Fathima, and G. Jyothika, “Student performance prediction using gradient boosting and SHAP analysis,” *Int. J. Novel Res. Develop.*, vol. 11, no. 5, pp. f326–f333, 2026. [Online]. Available: https://www.ijnrd.org/viewpaperforall.php?paper=IJNRD2605536

[5] S. Kumari and G. Khorwal, “A machine learning approach for student academic performance

prediction,” *Int. J. Creative Res. Thoughts*, vol. 14, no. 4, pp. f272–f275, 2026. [Online]. Available: https://ijcrt.org/viewpaperforall.php?paper=IJCRT2604614

[6] H. Thakur, A. Mansuri, H. K. Shrivastav, and A. Sahu, “Student performance prediction system,” *Int. J. Novel Res. Develop.*, vol. 11, no. 4, pp. d533–d539, 2026. [Online]. Available: https://www.ijnrd.org/viewpaperforall.php?paper=IJNRD2604347

[7] R. M. N., S. H. Mohan, and K. Vishwakiran, “Enhancing programming education: Machine learning models for predicting student performance,” *Int. J. Creative Res. Thoughts*, vol. 13, no. 1, pp. d965–d970, 2025. [Online]. Available: https://ijcrt.org/viewpaperforall.php?paper=IJCRT2501445

[8] B. Y. Reddy, B. H. Reddy, S. Dheeraj, and R. Bharathi, “Predicting academic performance of students using machine learning,” *Int. J. Creative Res. Thoughts (IJCRT)*, vol. 14, no. 4, pp. j196–j200, Apr. 2026. [Online]. Available: https://ijcrt.org/papers/IJCRT26A4069.pdf

[9] H. Vashist and M. Singh, “EduXplain Pro: An explainable ensemble machine learning framework for student performance prediction using academic and behavioral learning analytics,” *Int. J. Novel Res. Develop. (IJNRD)*, vol. 11, no. 7, pp. a884–a889, Jul. 2026, doi: 10.56975/ijnrd. v11i7.326639.

[10] A. Mahesh and D. Harwalkar, “Predictive modeling of student performance in higher education using machine learning algorithms,” SSRN, Apr. 11, 2025, doi: 10.2139/ssrn.6746338. [Online]. Available: https://ssrn.com/abstract=6746338