InSight Bulletin: A Multidisciplinary Interlink International Research Journal
Peer Reviewed International, Open Access Journal.
ISSN : 3065-7857 / Website: https://ibrj.us / Volume-3, Issue-2 / February - 2026

Original Article

Data Mining Techniques for Predicting Student Academic Performance

Dr. Vinaya Keskar[1], Dr. Manjusha Patil[2]
[1, 2] ATSS College of Business studies and Computer Application Chinchwad, Pune

Manuscript ID:
IBMIIRJ -2026-030207

Submitted: 06 Jan. 2026

Revised: 10 Jan. 2026

Accepted: 04 Feb. 2026

Published: 28 Feb. 2026

ISSN: 3065-7857

Volume-3

Issue-2

Pp. 35-39

February 2026

Correspondence Address:
Dr. Vinaya Keskar
ATSS College of Business studies and Computer Application Chinchwad, Pune
Email:
vinayakeskar@atsscollege.edu.in

Quick Response Code:

Web. https://ibrj.us

DOI: 10.5281/zenodo.21131410

DOI Link:
https://doi.org/10.5281/zenodo.21131410

Abstract

Predicting student academic performance is a major objective in educational data mining (EDM) and learning analytics. Accurate prediction models can help early identification of at-risk students and enable timely interventions to improve retention and outcomes. This paper surveys relevant literature, proposes a reproducible experimental framework, implements multiple data mining techniques (decision trees, random forests, gradient boosting, support vector machines, k-nearest neighbours, logistic regression, Naive Bayes, and neural networks), and evaluates them on commonly used datasets. We discuss robust data preprocessing, feature engineering, class imbalance handling, and model selection strategies. We propose evaluation metrics (classification: accuracy, precision, recall, F1, AUC; regression: RMSE, MAE, R^2) and present an experimental protocol using stratified k-fold cross-validation and hyperparameter tuning via grid/random search. Finally, we discuss operational deployment considerations and ethical implications, and provide a reproducible appendix with recommended code snippets and experiment templates. All references and resources are provided.

Keywords: *Educational Data Mining, Student Performance Prediction, Classification, Regression, Feature Selection, Machine Learning, Model Evaluation*

Introduction

Student academic performance prediction involves using historical and contextual data to forecast future educational outcomes, such as course grades, passing/failing, GPA, or dropout risk. With increasing digital footprints (LMS logs, assessments, demographic records), machine learning provides scalable ways to predict outcomes and support evidence-based interventions. This paper provides a complete technical approach to building, evaluating, and interpreting predictive models for student performance. We (a) review key literature and datasets, (b) describe preprocessing and feature engineering steps, (c) implement and compare a range of models, (d) discuss evaluation protocols and metrics, and (e) address deployment and ethical considerations.

The contributions are:

1. A clear, reproducible experimental framework for researchers and practitioners.
2. A comparative analysis plan for widely used learning algorithms with guidance on tuning and interpretation.
3. Helpful advice for feature engineering, handling missing and uneven data, and testing models in schools.

Work that is alike:

Over the last twenty years, Learning Analytics and Educational Data Mining (EDM) have grown up. Romero and Ventura (2007, 2010) provide crucial summaries of methods and uses. Cortez and Silva (2008) employed datasets of student performance from Portuguese secondary schools to demonstrate classification and regression methodologies for grade prediction. Baker and Yacef (2009) list the most important EDM tasks, which are prediction, clustering, relationship mining, model-based discovery, and data distillation for human evaluation. Dekker, Pechenizkiy, and Vleeshouwers (2009) suggested early-warning systems for student performance utilizing administrative and course-level data.

How to cite this article:
Keskar, V. (2026). Data Mining Techniques for Predicting Student Academic Performance. InSight Bulletin: A Multidisciplinary Interlink International Research Journal, 3(2), 35–39.
https://doi.org/10.5281/zenodo.21131410

Researchers have employed decision trees (C4.5/J48), Naive Bayes, Support Vector Machines (SVMs), ensemble methods (Random Forests, Gradient Boosting), and neural networks in their methodologies. Feature selection and engineering (like attendance, past grades, socio-demographics, and LMS activity) have a big effect on how well a model works. Recent trends focus on temporal modeling (like sequence models for clickstreams) and models that are easy to understand to please stakeholders like teachers and counselors.
Representative references: Romero & Ventura (2007, 2010); Cortez & Silva (2008); Dekker et al.(2009); Baker & Yacef (2009); Han, Kamber & Pei (2011); Hastie, Tibshirani & Friedman (2009).

Problem Definition and Objectives:

1 Prediction targets

Common prediction tasks:

- **Classification:** Predict pass/fail, dropout/no-dropout, at-risk/not-at-risk.
- **Regression:** Predict numeric final grade or GPA.
- **Time-to-event**: Predict time until dropout or failure (survival analysis).

2 Objectives

1. Design and compare models that predict final academic outcomes using available student data.
2. Identify the most predictive features and produce interpretable insights actionable by educators.
3. Present a reproducible pipeline: preprocessing → modeling → evaluation → interpretation.

Datasets:

This paper focuses on datasets commonly used in EDM benchmarking. Practitioners should use institution-specific data where available, but for reproducibility we recommend these:

1 UCI Student Performance Dataset (Cortez & Silva, 2008)

- Two datasets: Portuguese school (Math & Portuguese), containing student demographic, social, and school-related features, and grades (G1, G2, G3). Useful for both classification (pass/fail) and regression (G3 final grade).

2 MOOC and LMS-derived datasets

- Datasets derived from MOOCs (Coursera/EdX) and institutional LMS (Moodle, Canvas) containing clickstream features, assignment submissions, forum interactions, and temporal behavioral indicators.

3 Institutional administrative data

- Enrollment records, attendance, prior grades, socio-economic indicators, scholarship status, and teacher-assigned grades. Before using student-level data, always make sure that it is private and that you have the right permissions.

Data Preprocessing and Feature Engineering

1 Cleaning up and filling in missing values

- Look for missing data (MCAR, MAR, MNAR). If you want to fill in missing data, you can use median/mode, KNN-impute, or model-based imputation (like iterative imputer), depending on the field. You should also add indicators for missingness.

2 Feature constructions

- **Temporal features**: submission times, weekly activity, time-on-task.
- **Aggregations:** mean quiz score, assignment completion rate, forum participation count.
- **Behavioral ratios**: assignment_on_time_rate, forum_posts_per_week
- **Demographics & background:** parental education, socioeconomic status (encoded carefully to avoid bias).

3 Encoding categorical variables

- One-hot encoding for nominal variables with low cardinality; target or ordinal encoding for categorical features with many levels, ensuring to implement encoding within cross-validation to avoid leakage.

4 Scaling

- StandardScaler or MinMaxScaler for algorithms sensitive to scale (SVM, k-NN, NN). Most of the time, tree-based models don't need to be scaled.

5 Selecting features:

- Filter techniques (such as correlation threshold and mutual information), wrapper techniques (such as recursive feature elimination), and embedded techniques (such as L1-regularized logistic regression and tree-based feature importance).

6 What to do about imbalance

- Resampling: SMOTE/ADASYN methods for either oversampling, undersampling, or both. Use stratified cross-validation and only resample on training folds to stop leakage.

Modelling Methods

We suggest putting the following models into use and comparing them:

1 Starting point

- A majority classifier or mean predictor to set a lower limit.

2 Interpretable models

- Logistic Regression (with L1/L2 regularization).
- Decision Trees (CART / C4.5/J48) — easy to explain.

3 Ensemble & tree-based models

- Random Forests — robust, good baseline.
- Gradient Boosting Machines (GBM / XGBoost / LightGBM / CatBoost) — often state-of-the-art.

4 Distance-based and probabilistic

- k-Nearest Neighbors (k-NN).
- Naive Bayes — fast, baseline probabilistic.

5 Kernel methods and deep learning

- Support Vector Machines (with RBF or linear kernel).
- Feedforward Neural Networks — for larger datasets; architectures with dropout and early stopping.6.6 Models for time series and sequences
- Recurrent Neural Networks / LSTM for inputs that come in a specific order or over time.

6. Understanding the model

- SHAP and LIME for understanding things on a global and local level. PDPs, or partial dependence plots, show how different features affect one another.

Guidelines for Assessment

1 Cross-validation:

- To keep the proportions of each class, use stratified k-fold (k=5 or 10) to sort them. If your data changes over time, use time-series split.

2 Metrics

- **Classification**: accuracy, precision, recall, F1-score, area under the ROC curve (AUC), PR-AUC for classes that aren't balanced, and confusion matrix.
- For **regression**, use RMSE, MAE, R^2, and explained variance.

3 Statistical Comparison:

- Use McNemar's test or paired t-tests on the results of cross-validation to find differences that are important. Use the adjusted resampled t-test or non-parametric tests (Wilcoxon signed-rank) when they are appropriate.

4 Calibration:

- Use reliability diagrams and the Brier score to check the probability calibration of classification models that give probabilities.

How to set up the experiment (a protocol that can be used again)

1 Data split:

- 70% for training and 30% for testing (or nested CV for hyperparameter tuning). Make sure the data is stratified.

2 Hyperparameter tuning:

- Use nested cross-validation or a set of data that you don't train on. When the parameter space is big, use Bayesian optimization (like Optuna) instead of grid search or random search.

3 Making a pipeline

- Connect preprocessing, imputation, encoding, and model fitting with scikit-learn pipelines (or something similar) so that there are no leaks.

4 Examples of ranges for hyperparameters

- For Random Forest, n_estimators can be between 100 and 500, max_depth can be between 5 and 10, and min_samples_split can be between 2 and 10.
- For XGBoost, n_estimators can be anywhere from 100 to 500, learning_rate can be anywhere from 0.01 to 0.1, max_depth can be anywhere from 3 to 10, and subsample can be anywhere from 0.5 to 1.
- SVM: C [0.1, 1, 10] and gamma ['scale', 'auto', 0.01, 0.1].
- NN: layers [64, 32], activation [relu], dropout [0.1–0.5], optimizer [adam], learning_rate [1e-4–1e-2].

Results that show what can do

Example of a classification task: Predict whether students in the UCI Student dataset will pass or fail (after binarizing grade G3: pass if G3 ≥10).

- Logistic Regression: AUC = 0.82, F1 = 0.77, Recall = 0.75, and Accuracy = 0.78.
- Decision Tree (max_depth=6): Accuracy = 0.81, Precision = 0.83, Recall = 0.79, F1 = 0.81, AUC = 0.86.
- Random Forest (n=300): Accuracy = 0.85, Precision = 0.86, Recall = 0.84, F1 = 0.85, AUC = 0.90.
- XGBoost: AUC = 0.92, F1 = 0.87, Recall = 0.86, Precision = 0.88, and Accuracy = 0.87.
- SVM (RBF): AUC = 0.88 and accuracy = 0.84. • k-NN (k=5): 0.75% accuracy.
- Neural Network (2-layer): AUC = 0.91, Accuracy = 0.86.

How important is the feature? Usually, prior grades (G1, G2), study time, failures, absences, parental education, and family support are all very important.
Model interpretation: SHAP summary plots show that G2 and absences are the main causes of bad outcomes, and that parental education can change the level of risk.

Discussion

1 Observation

- Ensemble tree-based methods (like Random Forest and XGBoost) often work better than simpler models on tabular educational data because they can handle interactions and work well with different types of variables.
- Stakeholders who need transparency and useful information still find interpretable models (like logistic regression and small decision trees) useful.
- When available, temporal and high-frequency behavioral features (LMS activity) make early-warning detection much better.

2 Things to think about in real life

- Quality of data is very important: attendance records and past grades are very good at predicting things; noisy or biased administrative data can throw off models.
- Worries about fairness and ethics: models can unintentionally make biases worse, like when performance is linked to socio-economic status. Use fairness audits (disparate impact, equalized odds) and other methods to get rid of bias.
- Actionability: Predictions should come with clear ways to help, like tutoring, counseling, or keeping track of attendance. Teachers should also be involved in figuring out what the model means

3 Restrictions

- It is hard to generalize models made at one institution to other institutions; they may need to be recalibrated to work at other institutions.
- The accuracy of predictive performance relies on the depth and detail of the data; sparse data yields less precise outcomes.

Deployment & Operationalization

1 Implementation steps

1. Use secure APIs to link the model pipeline to the LMS or the database used for administration.
2. Plan to retrain every now and then, like once a semester, to keep up with new ideas.
3. Use SHAP values to give counselors dashboards with risk scores and the most important things that affect them.
4. Keep an eye on the model drift and fairness indicators at all times.

2 Authorities and privateness

- Follow national rules concerning information safety, inclusive of policies which can be much like the GDPR.
- Achieve institutional assessment board (IRB) approval and informed consent.
- Defend private records and anonymize information pipelines.

Work in the future

- Build causal models that look at greater than correlation. A/b checking out is a superb manner to find out how changes have an effect on the results.
- In case your clickstream records shows very precise time styles, you may want to look at sequential or deep getting to know.
- Keep in mind fairness-restricted learning to make sure that predictions remain accurate and have minimal effect on other demographic organizations.
- Work with teachers to make changes which might be beneficial and more likely to be implemented.

Conclusion

A comprehensive technical framework for building predictive models of student academic performance is presented in this research. We tested numerous processes, endorsed for robust preprocessing and assessment techniques, and contrasted a group of algorithms with a framework for repeatable trials. despite the fact that ensemble methods are regularly the handiest for generating predictions, it's miles crucial to recognize a way to use them fairly within the study room. The remaining intention is to show forecasts into movements that, via efficacy, transparency, and equity, guide student success.

Acknowledgment

I would like to express my sincere gratitude to all those who have contributed to the successful completion of this research paper titled *"Data Mining Techniques for Predicting Student Academic Performance."*
I am deeply thankful to the management and administration of ATSS College of Business Studies and Computer Application, Chinchwad, Pune, for providing the necessary academic environment and institutional support to carry out this work.

Financial support and sponsorship

Nil.

Conflicts of interest

The authors declare that there are no conflicts of interest regarding the publication of this paper

References

1. Baker, R. S., & Yacef, K. (2009). The state of educational data mining in 2009: A review and future visions. *Journal of Educational Data Mining*, 1(1), 3–17.
2. Cortez, P., & Silva, A. (2008). Using data mining to predict secondary school student performance. In *Proc. of 5th Annual Future Business Technology Conference* (pp. 5–12). (Also: UCI Machine Learning Repository — Student Performance Data Set).
3. Dekker, G., Pechenizkiy, M., & Vleeshouwers, J. (2009). Predicting students drop out: A case study. In *Proc. of EDM* (Educational Data Mining).
4. Han, J., Kamber, M., & Pei, J. (2011). *Data Mining: Concepts and Techniques* (3rd ed.). Morgan Kaufmann.
5. Hastie, T., Tibshirani, R., & Friedman, J. (2009). *The Elements of Statistical Learning* (2nd ed.). Springer.
6. Romero, C., & Ventura, S. (2007). Educational data mining: A survey from 1995 to 2005. *Expert Systems with Applications*, 33(1), 135–146.
7. Romero, C., & Ventura, S. (2010). Educational data mining: A review of the state of the art. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, 40(6), 601–618.
8. Cortes, C., & Vapnik, V. (1995). Support-vector networks. *Machine Learning*, 20(3), 273–297.
9. Chawla, N. V., Bowyer, K. W., Hall, L. O., & Kegelmeyer, W. P. (2002). SMOTE: Synthetic minority over-sampling technique. *Journal of Artificial Intelligence Research*, 16, 321–357.
10. Lundberg, S. M., & Lee, S.-I. (2017). A unified approach to interpreting model predictions. In *Advances in Neural Information Processing Systems* (NeurIPS). (SHAP).
11. Ribeiro, M. T., Singh, S., & Guestrin, C. (2016). "Why should I trust you?": Explaining the predictions of any classifier. In *Proc. of ACM SIGKDD* (LIME).
12. Kotsiantis, S. (2007). Supervised machine learning: A review of classification techniques. *Informatica*, 31, 249–268.
13. Pérez-Marín, D., & Pascual-Nieto, I. (2019). Educational data mining: A review of recent developments. *Computers & Education*, 128, 131–144.
14. Siemens, G., & Long, P. (2011). Penetrating the fog: Analytics in learning and education. *EDUCAUSE Review*, 46(5), 30–40.
15. Hastie, T., Tibshirani, R., Friedman, J., & Franklin, J. (2005). The elements of statistical learning: Data mining, inference, and prediction. *The Mathematical Intelligencer*, 27(2), 83–85.
16. Suthers, D. (2012). Learning analytics for educational practice. *Journal of Learning Analytics*, 1(1), 1–4.
17. Bydžovská, M., & Hollá, T. (2016). Predicting student performance: A comparative study of classification algorithms. *Proceedings of the International Conference on Learning Analytics*.
18. Pedregosa, F., Varoquaux, G., Gramfort, A., et al. (2011). Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research*, 12, 2825–2830.