Predicting Student Performance Using Synthetic Big Data and XGBoost

Anonymous Author(s)

Abstract

Predicting student academic performance is critical for early intervention and improving educational outcomes. However, there are many real-world data scarcity and privacy concerns that limit large-scale analysis. To address these challenges, a synthetically generated student performance dataset scaled to 50 million records was used to simulate a big data environment. This dataset, was sourced and expanded from an initial synthetic dataset on Kaggle, which included 21 demographic, socioeconomic, and academic features for each student.

The main objective of this work was to classify students into one of four GPA categories using a XGBoost classifier. The model was trained on the data and was evaluate using multiple performance metrics including accuracy, precision, recall, and F1-score. Extensive exploratory analysis and dimensionality reduction techniques such as PCA and chi-squared tests were sued to prepare the data and optimize model performance. Hyperparameter tuning was conducted using Optuna to improve generalization.

The results demonstrated that XGBoost achieved high predictive performance, with key features such as prior test scores, attendance rates, and parental education contributing significantly to GPA classification. This study highlights the potential of synthetic data to overcome ethical and logistical barriers in education research and validates the use of scalable machine learning frameworks for analyzing massive student datasets while maintaining data privacy.

Keywords

Education, XGBoost, Synthetic Data

1 Overview

Educational institutions increasingly leverage data-driven methods to predict and improve student success. *Educational Data Mining* (EDM) applies machine learning (ML) and data mining techniques to student records to uncover hidden patterns and predict academic results [9]. Previous studies have shown that analyzing characteristics such as demographics, socioeconomic status, and past grades can moderately predict student performance [9].

However, assembling a large-scale real-world student dataset poses challenges due to privacy constraints and data scarcity. High-quality educational data often contain sensitive personal information (e.g., grades, background), raising ethical and legal concerns about student privacy [5]. In addition, collecting millions of student records from various sources is time-consuming and costly.

To address these issues, we utilize a synthetic student performance data set obtained from Kaggle (contributed by the user 'NeuralSorcerer'). Synthetic data are artificially generated records that preserve the statistical patterns of real data without exposing real individuals. Recent research highlights that synthetic data can effectively overcome data scarcity, privacy concerns, and high collection costs in ML applications [7]. In the education domain specifically, synthetic data generation is seen as a promising privacy-preserving approach, allowing researchers to share and analyze student data while protecting confidentiality [7].

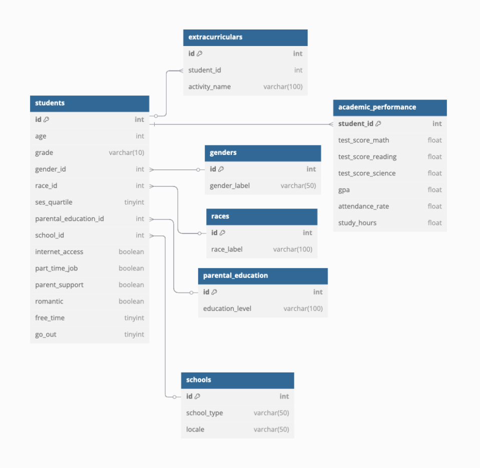
In this project, we will use a synthetic high-school performance dataset to build predictive models of student GPA. We further simulate a big data scenario by scaling the dataset to ~50 million records. The data is loaded into a relational database and accessed via Python in Google Colab, allowing us to handle the volume by querying in chunks. This approach provides a testbed for big data analytics techniques in education, letting us evaluate model performance and scalability on a massive dataset without real student records.

Our primary goal is to classify student academic outcomes (e.g., high or low GPA) based on their input characteristics. We focus on the XGBoost classifier for this task, given its strong track record in tabular data competitions and applications [1]. The following sections discuss relevant literature (Section 2), describe the dataset and pre-processing steps (Section 3), and detail our methodology including the justification for synthetic data and the choice of XGBoost (Section 4).

2 Related Work

Student Performance Prediction: Predicting academic success has been widely studied in EDM. Diverse machine learning techniques have been applied to student datasets, including decision trees, random forests, support vector machines (SVM), and logistic regression [9]. These models use student attributes (e.g. attendance, prior grades, socio-demographics) to forecast outcomes such as course grades or GPA. For instance, earlier work by Cortez and Silva (2008) modeled secondary school grades using demographic and grade features, demonstrating that data-driven models can identify at-risk students early [2]. Recent surveys confirm that EDM has become an effective tool to predict academic achievement and inform interventions [9]. Typical classification accuracies reported range from about 0.5 to 0.8 (50–80%), depending on the algorithms and features used [9]. Moreover, ensemble methods like gradient boosting have shown promise; Fernandes et al. (2019) used Gradient Boosting Machines on demographic and prior performance features to predict student grades, finding that prior grades and attendance were the most important predictors of final achievement [3]. These studies underscore the potential of ML models to improve educational outcomes, but they generally rely on relatively small datasets (hundreds or thousands of students).

Synthetic Data in Education: Due to legitimate privacy concerns, there is a growing interest in using synthetic educational data. Researchers have begun to explore techniques such as generative models and privacy-preserving frameworks to create artificial student records that mirror real data distributions [6]. Liu et al. (2025) introduce a framework combining synthetic data generation with differential privacy to safely share student data for research, demonstrating that such data can maintain utility for analysis while preventing privacy leaks [7].



extracurriculars	
id	int
student_id	int
activity_name	varchar(100)

students	
id	int
age	int
grade	varchar(10)
gender_id	int
race_id	int
ses_quartile	tinyint
parental_education_id	int
school_id	int
internet_access	boolean
part_time_job	boolean
parent_support	boolean
romantic	boolean
free_time	tinyint
go_out	tinyint

academic_performance	
student_id	int
test_score_math	float
test_score_reading	float
test_score_science	float
gpa	float
attendance_rate	float
study_hours	float

genders	
id	int
gender_label	varchar(50)

races	
id	int
race_label	varchar(100)

parental_education	
id	int
education_level	varchar(100)

schools	
id	int
school_type	varchar(50)
locale	varchar(50)

Figure 1: The used entity relationship diagram.

Synthetic data offer the advantage of scale and diversity without infringing on student privacy regulations (e.g. FERPA or GDPR). Our work builds on this concept by using an openly available synthetic dataset from Kaggle. By scaling it up to 50 million samples, we emulate a scenario of "big data" in education. This approach aligns with observations in the literature that synthetic data can address the lack of large real datasets [6]. It also follows best practices suggested in recent overviews of synthetic data research, which emphasize generating realistic, diverse artificial datasets to enable model training at scale [6].

3 Design and Implementation

Dataset Description: The project uses a synthetic Student Performance dataset (initially containing a few thousand records) created by NeuralSorcerer on Kaggle. Each record corresponds to a high school student and includes 21 attributes spanning demographic, socio-economic, and academic factors. Features are shown in table 1

The target variable for this study is the student's cumulative GPA, categorized into four ordinal groups: Low (0.00–2.49), Mid-Low (2.50–2.99), Mid-High (3.00–3.49), and High (3.50–4.00). These categories provide a balanced representation of academic performance levels for classification tasks. Notably, because the data is synthetically generated to resemble realistic student profiles, it contains no personally identifiable information. This alleviates privacy issues and allows unrestricted analysis. The synthetic data is assumed to reflect plausible correlations (for example, one might expect parental education to correlate with student grades, etc.), which our models can learn. To simulate a big data environment, we use Synthetic Data Vault [DataCebo, Inc. 2025] to generate 50 million rows of synthetic data from the original dataset of 10 million rows using a Gaussian Copula synthesizer. This approach enables faster iteration times and allows us to stress-test our database infrastructure in preparation for big data workloads. We use a PostgreSQL [8] database as well as ibis [4] in order to store and work with our data.

As shown in figure 1, the data is normalized into seven separate tables. The Gender, Race, and Parental Education attributes are separated into their own tables as they are often a self-reported category. Isolating them into distinct tables allows for data normalization, consistency and ease of cleaning. The Schools and Extracurricular attributes are separated as they have a one-to-many relationship with individual students. Finally, Academic Performance was stored in its own table to allow for more efficient and flexible queries on student performance.

3.1 Methodology

Predictive Modeling Approach: The task is formulated as a classification problem: given a student's features, predict the category of their GPA (or whether they are likely to perform well or poorly). We will split the data into training and testing sets (for instance, 80% train, 20% test, drawn randomly or by stratifying to maintain class proportions). Due to the large data volume, we may refrain from complex k-fold cross-validation; instead, a single train-test split or a hold-out validation set will be used to evaluate performance, to keep computation feasible. We will measure classification accuracy, and since class imbalance might exist (e.g. fewer failing students than passing), we will also track metrics like precision, recall, F1-score.

Our primary algorithm is **XGBoost** (Extreme Gradient Boosting). XGBoost is an ensemble method that builds a series of decision trees, where each tree corrects errors of the previous ones (boosting), with an efficient implementation that supports parallelism. We chose XGBoost because it has demonstrated state-of-the-art results in many structured data contests and applications [1]. In fact, XGBoost has become a de facto standard for tabular data modeling due to its high accuracy and speed [1]. It is precise and robust across various types of structured datasets and is relatively easy to use, with built-in handling for missing values and flexible tuning options. Importantly for our project, XGBoost is designed with performance optimizations that allow it to scale to very large datasets efficiently. Its core engine uses techniques like histogram-based splitting and cache-aware block structures, enabling it to train on billions of examples using far fewer resources than naive implementations [1]. This makes it well-suited to handle our simulated 50M-record dataset. We will use XGBoost's Python API (xgboost library) in Colab, potentially enabling GPU acceleration if available to further speed up training.

Hyperparameter tuning will be performed on a subset of the data or using automated methods. As detailed in Table 2, the optimization process explores a carefully defined search space of hyperparameters, including tree depth, learning rate, and regularization terms. These parameters are selected for their known impact on XGBoost's predictive performance and their role in controlling overfitting in high-dimensional synthetic data. Our objective function is designed to maximize multi-class classification accuracy by iteratively training and evaluating models across the parameter space outlined in Table 2. This search space is intended to balance broad exploration of configurations with computational feasibility, ensuring the tuning process remains scalable to larger or more complex datasets in future work. In the current implementation, we perform 30 Optuna trials, each training a model and evaluating

Table 1: Dataset Features and Descriptions

Feature	Description
Age	Student's age in years (14–18).
Grade	Grade level (9–12), derived from age.
Gender	Student gender (Female, Male).
Race	Race/ethnicity (White, Hispanic, Black, Asian, Two-or-more, Other).
SES_Quartile	Socioeconomic status quartile (1 = lowest, 4 = highest).
ParentalEducation	Highest education of parent/guardian (<HS, HS, SomeCollege, Bachelors+).
SchoolType	Type of school attended (Public, Private).
Locale	School location (Suburban, City, Rural, Town).
TestScore_Math	Math achievement score (0–100).
TestScore_Reading	Reading achievement score (0–100).
TestScore_Science	Science achievement score (0–100).
GPA	Cumulative Grade Point Average on a 0.0–4.0 scale.
AttendanceRate	Fraction of school days attended (0.70–1.00).
StudyHours	Average self-reported homework/study hours per day (0–4).
InternetAccess	Home internet access (1 = yes, 0 = no).
Extracurricular	Participation in clubs/sports (1 = yes, 0 = no).
PartTimeJob	Holds a part-time job (1 = yes, 0 = no).
ParentSupport	Regular parental help with homework (1 = yes, 0 = no).
Romantic	Currently in a romantic relationship (1 = yes, 0 = no).
FreeTime	Amount of free time after school on a scale from 1 (low) to 5 (high).
GoOut	Frequency of going out with friends on a scale from 1 (low) to 5 (high).

its accuracy on a held-out test set. Based on preliminary experiments, we anticipate that a smaller learning rate combined with a larger number of estimators (see Table 2) may support superior generalization by enabling gradual learning.

Future iterations will explore dynamically adjusting these parameters based on early stopping criteria and cross-validation results to reduce training time without compromising performance. Future tuning may involve:

- Extending trials beyond 30 to explore the hyperparameter space more thoroughly.
- Incorporating early stopping criteria to avoid unnecessary boosting rounds.
- Exploring Bayesian optimization or multi-objective tuning (e.g., balancing accuracy and training time).

This approach provides a systematic and scalable way to fine-tune XGBoost for optimal predictive performance on large, high-dimensional datasets.

In addition to XGBoost, we plan to establish baseline comparisons with simpler models. For example, a logistic regression or a single decision tree can provide a reference for performance and runtime. These baselines will help quantify the value-added by the more complex XGBoost model. We anticipate that XGBoost will outperform simpler models by capturing non-linear feature interactions and handling feature heterogeneity effectively.

Visualization: To assess model performance and interpret results, we generate visualizations focused on key evaluation metrics and predictive behavior. These include:

- Confusion matrices to analyze prediction accuracy across GPA categories and identify class-specific performance issues.
- Bar plots of precision, recall, and F1-scores for each class to highlight imbalances and areas for potential model improvement.
- Feature importance plots from XGBoost to interpret the relative contribution of input features to the final predictions.

These visualizations support both quantitative and qualitative analysis of model behavior. Metrics such as accuracy, precision, recall, and F1-score are used to provide a comprehensive view of classifier performance, especially in the presence of potential class imbalance.

Future iterations may extend result visualization to describe how the model derives its predictions. Advanced metrics such as Cohen's Kappa, Matthews Correlation Coefficient (MCC), and class-weighted F1-scores could also be incorporated to provide a more nuanced evaluation of model performance, particularly in the presence of imbalanced classes or ordinal target variables.

Big Data Implementation: Training on 50 million instances poses computational challenges. We will assess the feasibility of training XGBoost on the entire dataset. However, on Colab we are constrained to a single machine environment. One strategy is to train on a substantial random sample (e.g. 5–10 million rows) to get initial results within a reasonable time, and then, if possible, incrementally train on additional data. We will leverage the relational database to filter or sample data efficiently for this purpose. For example, using SQL queries to fetch only needed columns or random chunks can reduce I/O overhead.

After training the model, we will evaluate it on the held-out test set to report final performance metrics. We will also interpret the model by examining feature importance scores provided by

XGBoost to see which factors had the most influence on GPA prediction. As an additional qualitative check, we plan to construct a small set of hand-crafted synthetic student profiles based on domain knowledge and common-sense expectations. This will allow us to evaluate whether the model's predictions align with intuitive assumptions—for example, whether students with higher parental support and consistent attendance are more likely to be classified into higher GPA categories. This hybrid of quantitative validation and qualitative sanity checks ensures a more comprehensive assessment of model reliability.

4 Analysis

Exploratory Analysis and Cleaning: We perform exploratory data analysis (EDA) to examine feature distributions, validate data integrity, and prepare the dataset for downstream tasks. The process begins with generating summary statistics to provide a high-level overview of numerical and categorical features, supplemented by inspecting value distributions for each categorical variable. These steps allow early detection of irregular patterns such as skewed distributions or sparse categories that may influence modeling performance.

To assess data completeness, we calculate missing value counts for all features and visualize their presence using a heatmap. Although the dataset is designed to have no missing values by the original generator, we explicitly verify this assumption to ensure robustness if future synthetic data iterations introduce missingness or corrupted records. Any rows containing NaN values are dropped as a precautionary step, along with duplicate entries to avoid inflating certain feature distributions.

Future implementations may extend these cleaning steps to handle other potential data quality issues:

- Filtering outliers based on domain knowledge or statistical thresholds (e.g., values beyond 3 standard deviations).
- Consolidating inconsistent or overlapping categories in nominal features to strengthen class balance.
- Normalizing or scaling numerical features if models sensitive to magnitude differences are employed.
- Automating anomaly detection pipelines for large-scale synthetic data where manual inspection is infeasible.

This multi-step EDA and cleaning pipeline ensures that the dataset remains reliable and manageable, while being flexible enough to accommodate future synthetic datasets with potentially different characteristics.

Looking at Figure 2, several interesting patterns emerge. The majority of participants have internet access but do not have a part-time job or are currently in a romantic relationship. Slightly more students report participating in extracurricular activities. Surprisingly, most students do not receive regular parental support with homework.

Most participants report having a moderate amount of free time after school and tend to go out infrequently. Age is fairly evenly distributed across the 14–18 range, though most students are in 12th grade. In terms of academic performance, students score similarly in math and science but tend to perform slightly worse in reading.

Interestingly, test scores and attendance rates all follow approximately normal distributions that are skewed to the right, indicating generally high performance. There is a noticeable bump at the 100% mark across reading, writing, math, science scores, and attendance rate, suggesting a cluster of high achievers. Most students report spending around one hour per day studying.

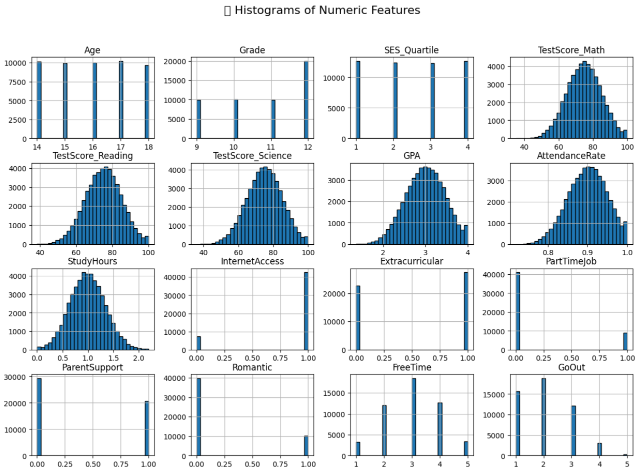


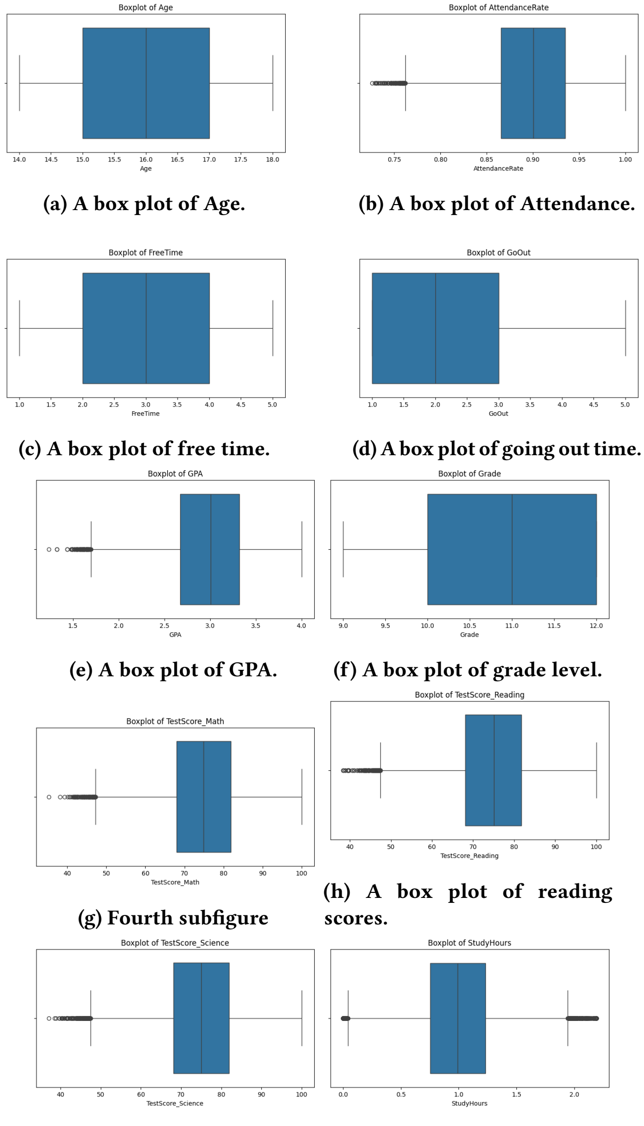
Figure 2: A histogram of count of each category of attribute.

Examining the box plots for math, reading, and science scores reveals a notable presence of low-score outliers, particularly concentrated in the low 40s. These outliers indicate that while the overall distribution of academic performance is skewed toward higher scores, there are many students who are significantly underperforming in each subject area. The presence of these low outliers may reflect factors such as lack of academic support, inconsistent study habits, or broader socioeconomic challenges.

In addition to academic scores, the box plots also highlight a number of outliers in attendance rate. While most students have high attendance, a small subset exhibits noticeably lower attendance, which may be a contributing factor to lower academic performance. These cases suggest a possible link between school engagement and achievement outcomes.

Furthermore, there are several outliers in GPA, with some students reporting cumulative GPAs below 1.75. This subset of students is performing well below the average and may be at risk of not meeting graduation requirements. These low-GPA outliers likely overlap with other negative indicators, such as low test scores and reduced attendance, pointing to a group of students who may require targeted academic interventions or support services.

Another interesting area is the distribution of study hours, which shows outliers on both the low and high ends. Many students spend low time studying, which could correlate with lower academic achievement or disengagement. Conversely, there are outliers on the upper end, where some students greatly exceed the average amount of hours. These students may be highly motivated or under significant academic pressure. The wide range of study behaviors captured by these outliers underscores the variation in time management and learning strategies among students, which may have complex relationships with academic outcomes depending on their context.



(a) A box plot of Age.

(b) A box plot of Attendance.

(c) A box plot of free time.

(d) A box plot of going out time.

(e) A box plot of GPA.

(f) A box plot of grade level.

(g) Fourth subfigure

(h) A box plot of reading scores.

(i) A box plot of science scores.

(j) A box plot of study hours.

Figure 3: Grouped figure showing four related images.

Taken together, the box plots provide valuable insights not just into the central tendencies of student performance, but also into the variability and potential areas of concern at the lower end of the spectrum.

Dimensionality Reduction: With 21 input features, we applied a hybrid dimensionality reduction strategy tailored to numerical and categorical data types.

For numerical features, we applied Principal Component Analysis (PCA) to transform correlated variables into an orthogonal set of principal components that maximize variance retention. PCA was selected because it effectively reduces redundancy in numerical data by concentrating information into fewer uncorrelated dimensions. Before applying PCA, we standardized numerical features using z-score normalization, as PCA assumes all variables are centered and scaled. We retained enough principal components to explain 90

- Improve training speed and reduce memory usage on large datasets.
- Mitigate multicollinearity, which can destabilize models sensitive to correlated inputs.
- Filter out noise by focusing on high-variance directions in the data.

For categorical dummy variables, we performed feature selection using the Chi-Squared (χ^2) test to evaluate their statistical association with the target variable. Instead of selecting a fixed number of top features, we retained all categorical features with a p-value less than 0.1. This threshold was chosen to balance inclusiveness of potentially informative features while filtering out noise from weakly associated variables. Using a p-value criterion ensures that only statistically significant predictors (at the 10% level) are passed to the model. The χ^2 test is particularly suitable for one-hot encoded data, as it measures dependency between binary features and a discrete target.

This hybrid PCA–χ^2 approach was selected because:

- PCA excels at handling numerical data with interdependencies, creating compact representations that preserve key patterns.
- The χ^2 test is efficient for high-dimensional categorical data and provides a statistical basis for feature relevance.
- Together, these methods produce a reduced, noise-resistant feature space optimized for mixed-type data.

Future iterations may explore:

- Adjusting the p-value threshold dynamically using cross-validation to optimize model performance.
- Combining PCA with supervised dimensionality reduction techniques (e.g., Partial Least Squares) for numericals.
- Evaluating embedded feature selection methods (e.g., tree-based importance scores) as an alternative to χ^2.

This dimensionality reduction process enhances model robustness and efficiency while retaining features with meaningful predictive power.

5 Legal Considerations

Use this section to discuss legal issues relevant to your project, especially relating aspects of data that are relevant to your project.

Use the textbook and your readings to guide the legal aspects of your discussion. Look at the laws that have been passed in recent years, and look at legislation that is being proposed in the space covered by your project.

6 Ethical Considerations

Use this section to discuss ethical issues relevant to your project, especially relating aspects of data that are relevant to your project.

Use the ACM Code to guide the ethical aspects of your discussion [?].

7 Conclusions

Use this section to describe the current status of your work and what else needs to be done.

Table 2: Hyperparameter search space for `XGBClassifier`

Parameter	Range	Description
`max_depth`	3–10	Controls tree depth; deeper trees may overfit.
`learning_rate`	0.01–0.3 (log scale)	Smaller rates allow gradual learning; larger rates speed up convergence.
`n_estimators`	100–300	Number of boosting rounds; more rounds improve accuracy but increase training time.
`subsample`	0.5–1.0	Fraction of data sampled per tree; reduces overfitting.
`colsample_bytree`	0.5–1.0	Fraction of features sampled per tree; improves generalization.
`gamma`	0–1.0	Minimum loss reduction required to make a split; adds regularization.
`reg_alpha`	0–1.0	L1 regularization to encourage sparsity in leaf weights.
`reg_lambda`	0–1.0	L2 regularization for weight shrinkage.

Also, discuss what further directions your work can be taken by others.

Finally, present some final words to place your project in perspective.

References

[1] Tianqi Chen and Carlos Guestrin. 2016. XGBoost: A Scalable Tree Boosting System. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '16)*. ACM, 785–794. doi:10.1145/2939672.2939785

[2] Paulo Cortez and Alice Silva. 2008. Using Data Mining to Predict Secondary School Student Performance. *Proceedings of 5th FUture Business Technology Conference (FUBUTEC)* (2008), 5–12.

[3] E. Fernandes, G. Holanda, M. G. Fernandes, and J. M. Carvalho. 2019. Educational data mining: Predicting students' performance using a Gradient Boosting Machine. *International Journal of Information Management* 50 (2019), 287–295. doi:10.1016/j.ijinfomgt.2019.05.003

[4] Ibis Project 2025. *ibis*. Ibis Project. https://ibis-project.org/reference/ Version 10.5.0.

[5] Birgit Ifenthaler and Philipp Schumacher. 2020. Student perceptions of privacy principles for learning analytics. *Educational Technology Research and Development* 68, 1 (2020), 165–183. doi:10.1007/s11423-019-09731-w

[6] Qinyi Liu, Oscar Deho, Farhad Vadiee, Mohammad Khalil, Srecko Joksimovic, and George Siemens. 2025. Can Synthetic Data be Fair and Private? A Comparative Study of Synthetic Data Generation and Fairness Algorithms. In *Proceedings of the 15th International Learning Analytics and Knowledge Conference (LAK '25)*. Association for Computing Machinery, New York, NY, USA, 591–600. doi:10.1145/3706468.3706546

[7] Q. Liu, R. Shakya, J. Jovanovic, and M. Khalil. 2025. Ensuring privacy through synthetic data generation in education. *British Journal of Educational Technology* 56, 3 (2025), 1053–1073. doi:10.1111/bjet.13576

[8] Postgres Project 2025. *PostgreSQL*. Postgres Project. https://ibis-project.org/reference/ Version 17.5.0.

[9] Cristóbal Romero and Sebastián Ventura. 2020. Educational data mining and learning analytics: An updated survey. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery* 10, 3 (2020), e1355. doi:10.1002/widm.1355