UCI Student Performance Dataset: Comprehensive Analysis and Modeling Framework

Kendall Freese [1 2] **Phillip Bonsu** [1 2] **Nick Moir** [2]

Abstract

This document provides a complete description and analysis of the UCI Student Performance dataset. It explores its educational relevance, modeling potential, and methodological implications using regression and classification—particularly k-Nearest Neighbors (kNN)—to predict student academic outcomes from behavioral and background variables.

1. Milestone 1

Dataset link: `https://archive.ics.uci.edu/dataset/320/student%2Bperformance`

1.1. Introduction

Educational achievement is a central focus of both social science and data science research, as it reflects not only individual effort but also the broader impact of family background and social environment. While national testing statistics provide aggregate insights, they often overlook how family characteristics and study habits shape the performance of individual students. The UCI Student Performance Dataset, available publicly from the University of California Irvine, addresses this by offering a collection of variables that capture parental education, student behavior, and academic outcomes.

This dataset is particularly well-suited for predictive modeling in the classroom. It provides a structured environment in which regression and classification can be compared directly, and where the consequences of model choices, such as the value of k in k-nearest neighbors (kNN), can be demonstrated with clarity. As such, it stands not only as a data source for analyzing student outcomes but also as a teaching tool for machine learning methods.

1.2. Dataset Description

The dataset contains a few hundred student-level records (295 math students, 649 Portuguese students, with 33 variables each). The key variables fall into three broad categories:

1. Background characteristics – parental education levels, household context, and family support, etc.
2. Behavioral indicators – weekly study time, prior course failures, and school absences, etc.
3. Outcomes – Continuous academic scores for a year of math and Portuguese, including individual semester grades, etc.

From a machine learning perspective, this structure offers two pathways. Continuous grade outcomes allow for regression models, where we quantify the relationship between predictors (study hours, parental education, absences) and numeric outcomes (grades). Alternatively, grades can be transformed into categorical outcomes such as pass/fail or high/medium/low performance, enabling classification models. This dual potential is particularly useful when introducing the distinction between regression and classification in coursework.

The dataset is also quite extensive in variables, yet relatively small in scale. Missing values are minimal, and much of the wrangling tasks involve recoding categorical variables (e.g., U $\rightarrow$ Urban, R $\rightarrow$ Rural, LE3 $\rightarrow$ family size 3) and handling semicolon delimiters. These features make it ideal for implementing algorithms like kNN regression and kNN classification, where normalization and distance metrics play a critical role.

1.3. Research Value and Methods

The dataset provides both substantive and methodological value. Substantively, it allows us to evaluate how family background and student behavior interact to produce outcomes. For instance, does parental education predict higher performance independent of study time, or do study habits mediate that relationship? Methodologically, it provides a sandbox for testing concepts like bias-variance tradeoff and overfitting versus underfitting.

For example, when applying kNN regression to predict student grades, a very small k may lead to overfitting: the model captures noise from individual students rather than the general trend, producing low training error but poor test performance. In contrast, a very large k may lead to under-

fitting, smoothing across too many students and obscuring meaningful variation. By iterating over different values of k and evaluating the sum of squared error (SSE) on training versus test data, students can observe how model complexity affects predictive accuracy.

In classification exercises, the dataset can also be used to generate confusion tables, which compare actual versus predicted categories of student performance. This provides insight not only into overall accuracy but also into whether the model tends to misclassify specific groups, such as high performers being labeled average or vice versa.

1.4. Broader Significance

The educational significance of this dataset lies in its ability to highlight actionable patterns. Policymakers may find value in identifying whether increased study time offsets disadvantages in parental education, or whether absenteeism is a stronger predictor of failure than background characteristics. While the dataset does not provide institutional factors such as teacher quality or funding, its individual-level detail offers a window into the dynamics of student success.

For students of data science, the dataset is equally important as a pedagogical tool. It allows learners to practice end-to-end workflows: data wrangling, visualization, normalization, fitting regression and classification models, and evaluating results with SSE or accuracy metrics. Because the subject matter is intuitive and socially relevant, the technical lessons of distance metrics, probability distributions, and model validation become more concrete.

1.5. Conclusion

The UCI Student Performance Dataset is a compact, versatile, and socially meaningful resource for both analysis and teaching. It captures key elements of family background, student effort, and outcomes, making it useful for substantive research questions about educational inequality. At the same time, it is small and structured enough to demonstrate machine learning techniques in practice.

In regression settings, it illustrates how continuous outcomes can be predicted and evaluated using SSE. In classification, it supports accuracy analysis through confusion tables and probability distributions. And in kNN, it demonstrates vividly the trade-offs between underfitting and overfitting. These properties make it an ideal dataset for DS 3001, as it aligns directly with the concepts students are learning while grounding them in an applied, real-world context.

2. Milestone 2

2.1. Problem Introduced by Dataset

The overarching problem presented by the dataset is to model and understand how students' background characteristics and behavioral indicators influence their academic performance. For example, given factors like parental education level, study time, absences, etc., we will attempt to determine how much each factor plays a role in predicting a student's final grade.

However, to accomplish this, we must first overcome both a prediction and a modeling problem. Both of these can be solved if the right model is chosen and trained in the correct way, but due to the dataset's smaller size and heavily correlated variables, we must make sure not to overfit the data by choosing too complex a model, while also not choosing a simple model that will not understand all the patterns in the data.

Conceptually, the challenge is to use our model to uncover whether school performance is primarily affected by individual study habits, family background, or a combination of the two.

2.2. Model Used and Justification

For this project, our team will use the k-Nearest Neighbors (kNN) model to predict and analyze students' academic performance based on their background and behavioral characteristics. This model was chosen because of its ability to provide a strong balance between interpretability and flexibility, which fits well with the dataset's smaller size and mix of numeric and categorical variables.

The kNN model operates by predicting a student's final grade based on the average performance of the k most similar students. Similarity is determined by distance in feature space, using characteristics such as study habits, absences, and parental education levels. This allows us to visualize how proximity among students in these features translates into predicted academic outcomes.

kNN is especially well-suited for this dataset because it does not assume a linear relationship between predictors and outcomes, making it ideal for identifying local, nonlinear patterns that might be missed by other models. The model's sensitivity to feature scaling and the choice of k also makes it a valuable tool for exploring the bias-variance tradeoff, which is essential in understanding model performance. To address these sensitivities, our team will use cross-validation and normalize continuous variables to determine the optimal value of k.

Overall, the kNN model offers an intuitive and effective framework for understanding how different student characteristics cluster and influence performance, while rein-

forcing core principles of predictive modeling and model evaluation.

2.3. Model Training Procedure

Before we start training the model, we must prepare the data to be ingested by the model, rather than using its raw, uncleaned state. To start, we will analyze the dataset for any missing values. If there are missing values, the observation they belong to will be ignored to ensure that we only analyze complete records, and don't have to assume any values. Next, we can fix the categorical variables, like parental job or school name, that need to be one-hot encoded to turn them into numeric values that can be analyzed by the model.

After encoding, we will scale all of the numeric variables so no single feature dominates the distance calculations. This step is especially important for kNN since it relies on measuring how close students are to one another in feature space, and features like absences or study time could otherwise outweigh smaller-scaled variables. Once the data is scaled, we will split it into training and testing sets, using 80% of the data for training and 20% for testing to evaluate how well the model generalizes.

During training, the kNN algorithm will store the training data and make predictions for new students by locating the k most similar students based on Euclidean distance. The predicted grade for each student will be the average of those nearest neighbors' grades. We will experiment with several k values to find the one that produces the lowest error and avoids both overfitting and underfitting. Model performance will be evaluated using Mean Squared Error (MSE) and Root Mean Squared Error (RMSE), allowing us to compare accuracy across different k values and identify the most reliable configuration for predicting academic performance.

2.4. Model Validation Plan

Model validation is essential in ensuring the model generalizes to new data beyond training data. The correct validation will avoid overfitting, help compare similar models, and give us confidence that our model's predictions are accurate. The framework we will follow is a k-fold cross-validation on the training set. We will estimate the out-of-sample performance of a kNN model while balancing overfitting. This can support either regression or classification, thus we will validate on metrics specific to both regression and classification.

For regression we will focus on RMSE that is sensitive to scale and R-squared. We can analyze residual vs. fitted and QQ plots for visualizing the respective impact. For classification we will focus on accuracy and ROC-AUC. We can analyze the respective confusion matrix for data visualization.

As a final test to our model, we can report the validation regression and classification metrics, confidence intervals, and key plots like confusion matrices or residuals. We'll summarize the difference between CV and test and conclude if the model over or under performs.

2.5. Implementation Notes

We can use certain implementations to combat potential roadblocks in data quality, partitioning, interpretability, and reproducibility. Data may be missing, have strong outliers, and feature mixed types that impede us from making a quality analysis. We can address this by imputing numeric values with the median and categorical values with most frequent; for outliers we can scale by the median so extreme values don't overpower the data, and for mixed types of data we can keep numeric and categorical preprocessing separate.

There could be partitioning problems between the test and training data; this can be minimized via time-based splits to ensure time periods don't overlap between the train/test sets. For interpretability we can use local neighbor inspection to check close data points and how they influenced the model. For reproducibility we can change seed amounts, record thresholds chosen, and replicate our process to ensure there is no human error. Ultimately, we want to ensure our model is accurate and a successful predictor.

3. Conclusion

The UCI Student Performance Dataset provides a powerful platform for modeling how social and behavioral factors influence academic outcomes. By combining regression and classification, it allows both predictive analysis and interpretive understanding of which variables most strongly drive performance.

Using the kNN model, we are able to explore the relationships among student characteristics and visualize how proximity in feature space corresponds to similar outcomes. The model's simplicity, interpretability, and flexibility make it an ideal teaching tool for demonstrating key concepts such as bias-variance tradeoff, feature scaling, and cross-validation.

From a pedagogical standpoint, this dataset captures the entire modeling process—data cleaning, transformation, scaling, model fitting, and evaluation—while maintaining social relevance and interpretability. As such, it not only introduces technical competence in predictive modeling but also reinforces the broader connection between data science and educational research. The structured yet compact nature of the dataset makes it particularly suitable for courses like DS 3001, where clarity, reproducibility, and conceptual grounding are essential.

4. Milestone 3

4.1. Dataset Preparation

For this project, we used the UCI Student Performance dataset, which provided a small but information-dense set of 649 student records with 32 observed features. These variables ranged from basic demographics (age, sex, family size) to parental education levels, weekday and weekend alcohol consumption, study time, absences, and each student's semester grades (G1 and G2). The target we aimed to predict was the final course grade (G3), recorded as a numeric value between 0 and 20. In our code, we accessed and downloaded the dataset using the `ucimlrepo` package and extracted the features and target into `X` and `y`, keeping the entire dataset intact since there were no missing values to remove. Because the dataset mixed categorical, ordinal, and continuous variables, we had to clean everything up before feeding it into the model. We converted all categorical features using pandas' `get_dummies`, which broke things like school, parental job, romantic status, and internet access into standard one-hot encoded columns the model could actually use. After that, we scaled all of the numeric features with scikit-learn's `StandardScaler` so that no single variable dominated the distance calculations. This mattered a lot for kNN because features like absences or previous grades sit on much larger numeric ranges and would have completely warped which students were considered "neighbors" if we left them unscaled. Once everything was encoded and standardized, we ran an 80/20 train–test split to make sure the model was always evaluated on data it never saw during training. With the dataset being fairly small and most variables heavily correlated, this setup gave us peace of mind as to whether kNN was actually learning useful local structure or just memorizing noise.

4.2. Model Implementation

Our baseline model was the kNN regression. The advanced models we pursued were Decision Tree, Random Forest, Gradient Boosting, Bagging, and Lasso regressions, giving us a diverse set of model classes for comparison. We implemented feedback from previous milestones to incorporate more complex models, allowing us to build toward a final model and results for the final paper.

After training the kNN model, we evaluated the optimal value of k for predicting the final grade (G3) based on RMSE and R^2. The baseline results for kNN were an RMSE of 2.840 and an R^2 of 0.173, which we used as our reference point for comparing the more advanced models.

We prepared the advanced models through preprocessing, implementing train/test splits, fitting each model, and evaluating their predictive strength. We then implemented the Decision Tree, Random Forest, Gradient Boosting, Bagging, and Lasso models, computing their respective RMSE and R^2 values. Comparing these metrics showed that Gradient Boosting performed best, achieving the lowest RMSE of 2.771 and the highest R^2 of 0.213.

4.3. Performance Metrics

Since our task was predicting a continuous numeric outcome, we evaluated the performance of our kNN model using regression metrics, specifically RMSE and R^2. RMSE gave us a straightforward sense of how far our predictions were from the true final grades, measured in the same units as the target variable. Because it punished large mistakes more than small ones, RMSE pushed the model to avoid big misses rather than just being "close on average." The R^2 score complemented this by measuring how much of the variation in student performance the model actually explained. An R^2 near 1 meant the model captured most of the underlying structure, while values near 0 meant it was not doing much better than predicting the mean grade for everyone. Using both metrics gave us a balanced idea of how the model performed: RMSE told us the raw prediction accuracy, and R^2 told us whether the model was actually learning patterns instead of just memorizing or averaging the data. These metrics were also what we used when testing different k values, adjusting hyperparameters, and figuring out whether kNN was genuinely the right fit for this dataset.

4.4. Model Results and Comparisons

To validate our predictive models for student final grades (G3), we first evaluated K-Nearest Neighbors across a range of k values and examined how performance changed using RMSE and R^2. The RMSE curve showed a steep decrease from $k = 1$ to about $k = 7$, after which the error stabilized, reaching its lowest point around $k = 20$–35. This pattern indicates that very small k values overfit the training data, while moderate k values generalize better.

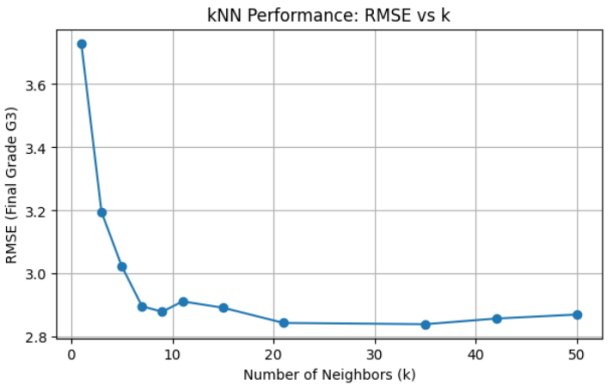

Figure 1. kNN RMSE Performance

Similarly, the R^2 visualization showed that $k = 1$ performed worse than a baseline mean predictor, but performance steadily improved as k increased, peaking near 0.17. Together, these diagnostics confirm that kNN captures some structure in the data but has limited explanatory power for predicting student performance.

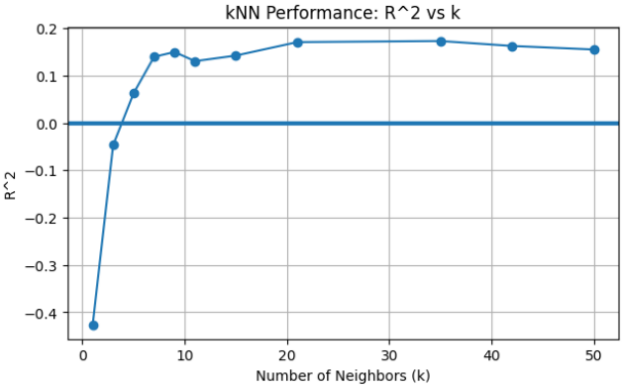

Figure 2. kNN R^2 Performance

Next, we compared all advanced models—including Decision Tree, Random Forest, Gradient Boosting, Bagging, Lasso, and the tuned kNN model—against one another and against a baseline that predicts the mean training G3. Gradient Boosting achieved the best performance, with the lowest RMSE (2.77) and highest R^2 (0.21), followed by Lasso, Random Forest, and kNN. All of these models outperformed the baseline, while the standalone Decision Tree performed substantially worse, confirming that simple trees are unstable and prone to overfitting on this dataset. These results demonstrate that ensemble methods meaningfully improve predictive accuracy, even though the overall R^2 values remain modest.

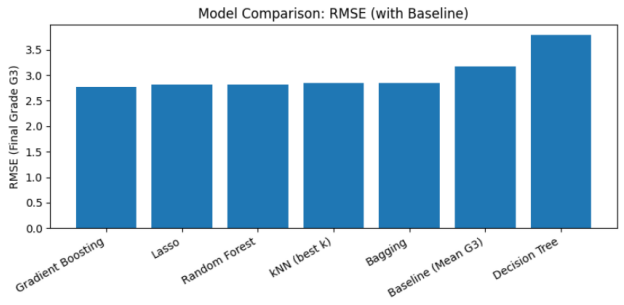

Figure 3. Model RMSE Comparison

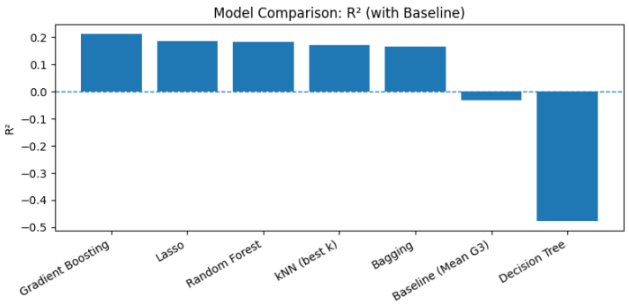

Figure 4. Model R^2 Comparison

To further validate the best-performing model, we ran cross-validation on Gradient Boosting. The RMSE values across folds showed only mild variation, suggesting that the model is reasonably stable and not overly dependent on a particular train/test split.

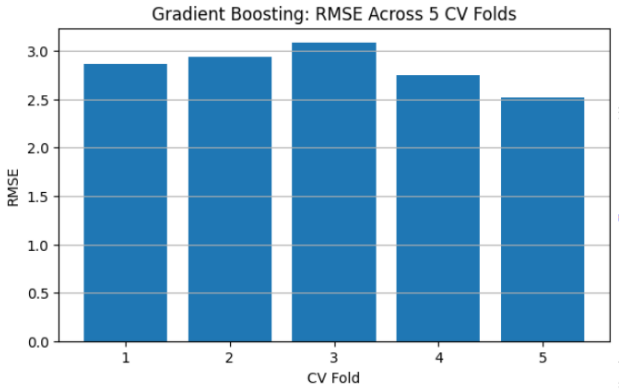

Figure 5. Gradient Boosting RMSE

We then examined residual diagnostics to assess model assumptions. The residuals-versus-fitted plot showed residuals centered around zero with no major patterns, indicating low systematic bias.

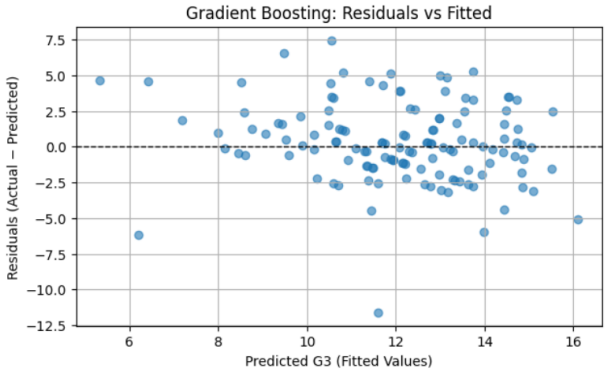

Figure 6. Gradient Boosting Residuals vs Fitted

The QQ plot showed that the residuals follow a roughly normal distribution with slight deviations in the tails, which is acceptable for a dataset of this size. Taken together, these visual checks confirm that Gradient Boosting not only performs best quantitatively but also behaves consistently and predictably across validation techniques.

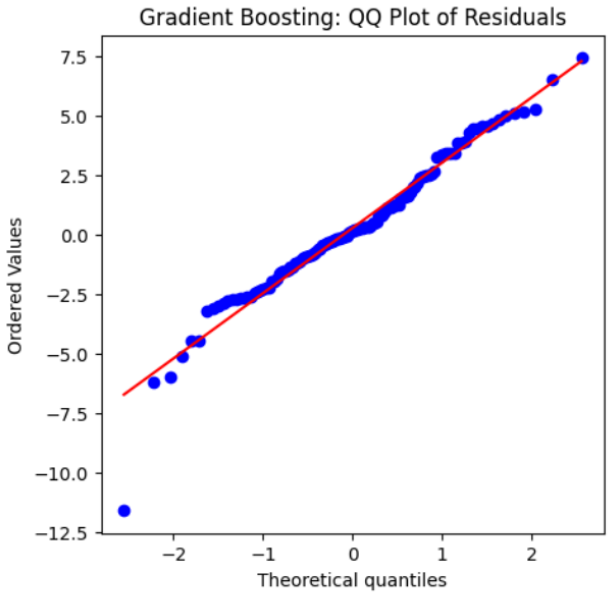

Figure 7. Gradient Boosting Residuals QQ Plot

Overall, the initial validation results show that Gradient Boosting is the strongest and most reliable model for predicting student G3 scores, while simpler or less robust models either underperform or fail to provide stable predictions. These findings guide our next steps as we refine the model and interpret which features contribute most to student performance.