UCI Student Performance Dataset: Comprehensive Analysis and Modeling Framework

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

Abstract

This project provides a complete description and analysis of the UCI Student Performance dataset and examines how demographic, behavioral, and academic factors influence students' final grade outcomes (G3). The dataset includes a collection of variables that capture parental education, student behavior, household context, and academic performance, making it well suited for predictive modeling in the classroom. Its mixture of categorical, ordinal, and continuous variables, along with minimal missingness, offers 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.

To prepare the data for modeling, we performed comprehensive preprocessing including one-hot encoding, normalization, and an 80/20 train–test split, ensuring that no single feature dominated distance calculations and that heavily correlated variables did not overwhelm the modeling process. Our exploration of kNN highlighted the importance of tuning. Smaller k values tended to overfit the training data, while moderate values revealed more stable and generalizable predictions. Although kNN captured some structure in the dataset, its predictive power remained modest, reinforcing the need to evaluate more flexible models.

Building on this foundation, we implemented advanced models such as Decision Tree, Random Forest, Gradient Boosting, Bagging, and Lasso, and compared them against both the tuned kNN model and a baseline predictor. Gradient Boosting achieved the lowest RMSE and highest R2, consistently outperforming the other models. Ensemble methods, including Random Forest and Bagging, also surpassed the baseline and the standalone Decision Tree, demonstrating that aggregated learners are better suited for datasets with mixed feature types and nonlinear relationships. Five-fold cross-validation on Gradient Boosting showed stable RMSE values across folds, indicating dependable generalization. Diagnostic plots further confirmed model stability, with residuals centered around zero and QQ-plot patterns reflecting approximate normality.

Overall, the project illustrates that ensemble methods, especially Gradient Boosting, offer the strongest and most reliable framework for predicting student performance in this dataset, while also providing a methodological foundation for future work such as feature importance analysis, hyperparameter optimization, and examining interventions linked to influential predictors.

1. Data

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 (Cortez and Silva, 2008), 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 vari-

ables 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. 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.

2. Methods and Results

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 correctly, 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 (Fix and Hodges, 1951) 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, non-linear 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 (Stone, 1974) 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 reinforcing 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. Next, we can fix the categorical variables using pandas' `get_dummies` (McKinney, 2010), 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 using scikit-learn's `StandardScaler` (Pedregosa et al., 2011) 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. Once the data is scaled, we will split it into training and testing sets using an 80/20 train–test split 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 RMSE and R^2.

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 approach on the training set. 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 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.

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.

2.6. Model Implementation

Our baseline model was the kNN regression. The advanced models we pursued were Decision Tree (Breiman et al., 1984), Random Forest (Breiman, 2001), Gradient Boosting (Friedman, 2001), Bagging (Breiman, 1996), and Lasso regression (Tibshirani, 1996), giving us a diverse set of model classes for comparison.

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.

2.7. Performance Metrics

Since our task was predicting a continuous numeric outcome, we evaluated the performance of our kNN model using 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. The R^2 score complemented this by measuring how much of the variation in student performance the model actually explained. Using both metrics gave us a balanced idea of how the model performed.

2.8. Model Results and Comparisons

To validate our predictive models for student final grades (G3), we first evaluated kNN 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, indicating that very small k values overfit the training data, while moderate k values generalize better.

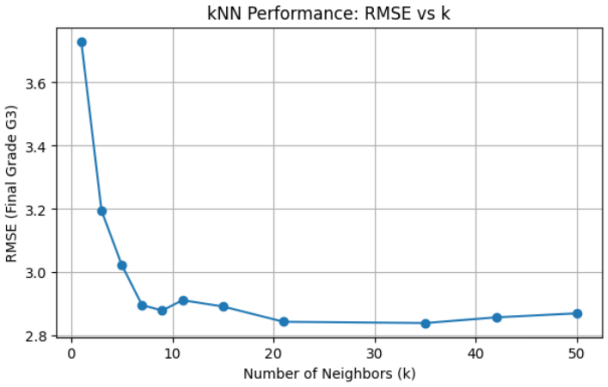

Figure 1. kNN RMSE performance across different values of k.

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.

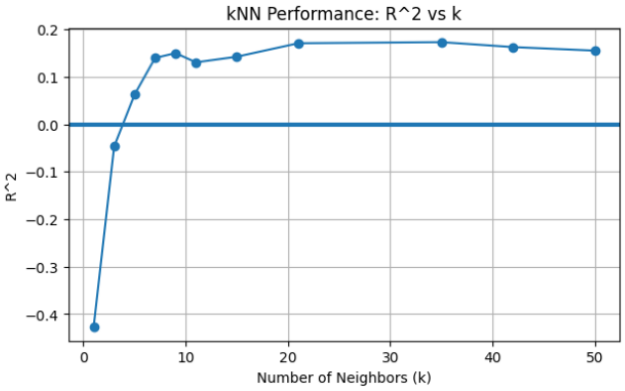

Figure 2. kNN R^2 performance across different values of k.

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.

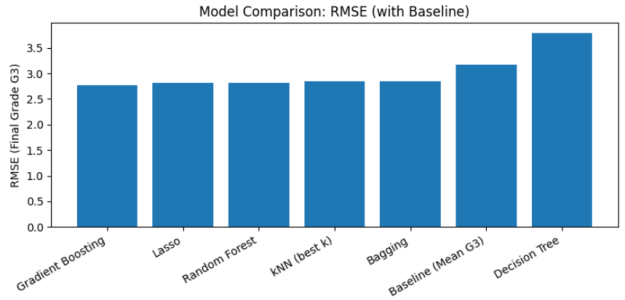

Figure 3. Model RMSE comparison across baseline, kNN, Decision Tree, Random Forest, Gradient Boosting, Bagging, and Lasso.

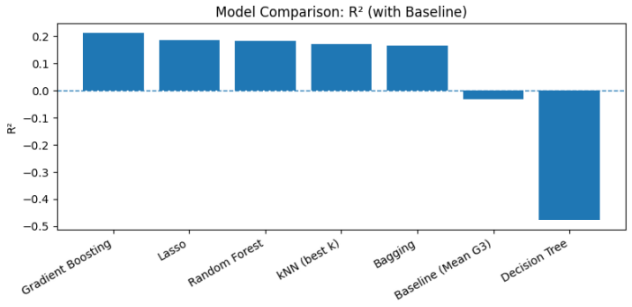

Figure 4. Model R^2 comparison across baseline, kNN, Decision Tree, Random Forest, Gradient Boosting, Bagging, and Lasso.

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.

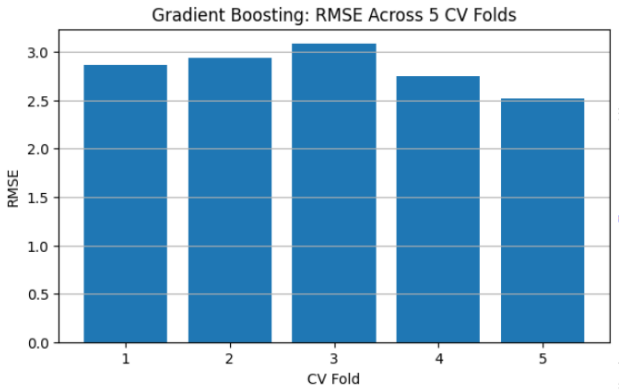

Figure 5. Gradient Boosting RMSE across cross-validation folds.

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.

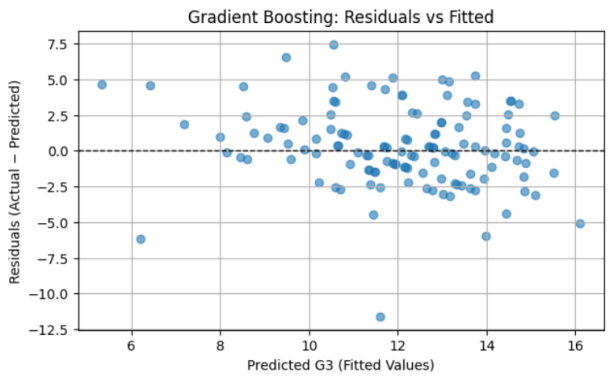

Figure 6. Gradient Boosting residuals vs. fitted values.

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.

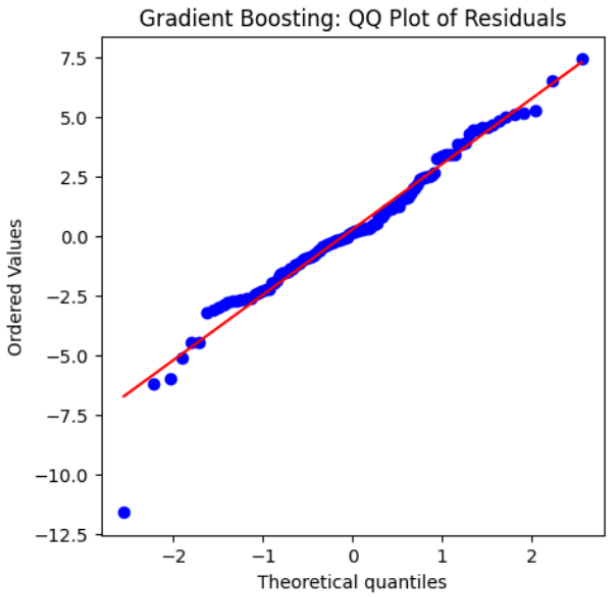

Figure 7. Gradient Boosting residuals QQ plot.

2.9. Feature Analysis

Using our selected Gradient Boosting model, we evaluate the given parameters to determine which features are most predictive of final grades. For interpretation, importance values below 0.01 are considered negligible; values between 0.01–0.03 are weak; 0.05–0.06 are moderate; and values near 0.25 represent very strong predictive power. Based on these thresholds, prior failures emerge as the strongest predictor of final grade. The variable `num_failures` has an importance value of 0.251, which is more than five times higher than the next most influential feature. This indicates that students who have previously failed classes are substantially more likely to earn lower final grades, suggesting that past academic struggles provide the strongest signal of future performance.

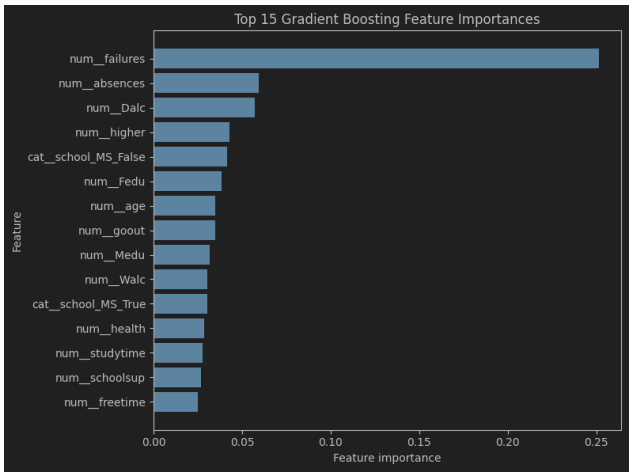

Figure 8. Top 15 Gradient Boosting Features

Absences are the second most influential parameter with an importance value of 0.059, indicating a moderate negative relationship with final grades. This shows that increased absences have a meaningful association with reduced academic outcomes and are more predictive than the social or demographic variables in the dataset.

Alcohol consumption during the work week (`num_Dalc`) is nearly as influential as absences, with an importance value of 0.057. This suggests that lifestyle factors such as weekday drinking are moderately associated with lower academic performance.

Higher education aspirations (`num_higher`) predict better academic outcomes, with an importance value of 0.043. Students who express plans to pursue higher education tend to perform better, indicating that motivation and long-term goals positively correlate with current academic success.

School-level differences also contribute moderately. The variables `cat_school_MS_False` and `cat_school_MS_True` have importance values of 0.042 and 0.030, respectively. Parental education shows a similar moderate influence: `num_Fedu` has an importance of 0.038 and `num_Medu` an importance of 0.032. These results imply that while parental education and school differences matter, they are not as impactful as behavioral and academic history–related predictors.

Age, health, going out, and weekend alcohol consumption have weak but non-negligible predictive power, with importance values ranging from 0.025 to 0.035. This indicates that social habits and general well-being have small but negative associations with academic performance.

Study time (`num_studytime`) has an importance value of 0.027, suggesting a positive but relatively minor contribution compared to absences, failures, or alcohol use.

Overall, the model reveals that the strongest predictors of final grades stem from behavioral patterns and past academic outcomes rather than demographic characteristics. Prior failures, absences, and alcohol consumption dominate in predictive power, while factors often assumed to strongly influence performance exhibit only moderate or weak associations.

2.10. Limitations

Our study has several limitations that should be noted. The dataset represents a single set of students in one educational context, so the findings may not generalize broadly to other schools or populations. The predictive power of all models remains moderate, which makes sense given both the limited number of observed features and the complexity of academic performance. The analysis is also observational, meaning the models can identify associations but cannot establish causal relationships between student characteristics and final grades. These limitations suggest that additional variables and broader data sources would be useful for strengthening future work.

3. Conclusion

This project set out to evaluate how demographic, behavioral, and academic factors influence student's final grade outcomes (G3) using the UCI Student Performance dataset. Through comprehensive data cleaning, categorical encoding, and feature scaling we prepared a reliable modeling environment that allowed both distance-based and ensemble models to operate effectively. Our exploration of k-Nearest Neighbors highlighted the importance of model tuning. Small k values overfit the training data, whereas moderate values highlighted more stable and generalizable predictions. Although kNN captured some structure in the data, its predictive power remained modest, demonstrating the need for more flexible models.

Building upon this foundation, our team implemented and compared several advanced models such as Decision Tree, Random Forest, Gradient Boosting, Bagging, and Lasso, alongside the tuned kNN model and a mean-predicted baseline. Gradient Boosting clearly outperformed all alternatives, achieving the lowest RMSE and the highest R2, followed up by the Lasso and Random Forest which trailed closely. Ensemble approaches consistently surpassed both the baseline and the standalone Decision Tree, underscoring the aggregated learners are better suited for datasets with mixed feature types and nonlinear relationships. To ensure the reliability of our conclusions, we conducted a detailed validation of the Gradient Boosting model. Five-fold cross-validation revealed stable RMSE values across folds, indicating strong generalization rather than dependence on a single train-test split. Diagnostic plots further

solidified and supported this stability. Residuals were centered around zero with no major structure, and the QQ-plot showed approximate normality with minor tail deviations. Taken together, these checks confirm that Gradient Boosting is not only the leading performer but also the most dependable model for this prediction task.

Our model revealed that the strongest predictors of final grades stem from behavioral patterns and past academic outcomes rather than demographic characteristics. Prior failures, absences, and alcohol consumption dominate in predictive power, while factors often assumed to strongly influence performance exhibit only moderate or weak associations.

Our team's findings demonstrated and highlighted that ensemble methods, specifically Gradient Boosting offers the strongest and most reliable approach for predicting student performance in this dataset. It should be noted that while we understand that the predictive power is moderate due to the inherent complexity and variability of human academic outcomes, the project establishes a clear framework that is methodological and highlights promising directions for future work, such as feature importance analysis, hyperparameter optimization, and exploring interventions linked to the most influential predictors.

References

Breiman, L. (1996). Bagging predictors. *Machine Learning*. https://doi.org/10.1007/BF00058655

Breiman, L. (2001). Random forests. *Machine Learning*. https://doi.org/10.1023/A:1010933404324

Breiman, L., Friedman, J. H., Olshen, R. A., & Stone, C. J. (1984). *Classification and Regression Trees*. Wadsworth. https://doi.org/10.1201/9781315139470

Cortez, P., & Silva, A. (2008). *UCI Machine Learning Repository: Student Performance Data Set*. University of California, Irvine. Retrieved from `https://archive.ics.uci.edu/dataset/320/student%2Bperformance`

Fix, E., & Hodges, J. L. (1951). Discriminatory analysis. USAF School of Aviation Medicine. Reprinted report available at: https://digital.library.unt.edu/ark:/67531/metadc17066/

Friedman, J. H. (2001). Greedy function approximation: A gradient boosting machine. *Annals of Statistics*. https://doi.org/10.1214/aos/1013203451

McKinney, W. (2010). Data structures for statistical computing in Python. *Proceedings of the 9th Python in Science Conference*. https://doi.org/10.25080/Majora-92bf1922-00a

Pedregosa, F., et al. (2011). Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research*. https://www.jmlr.org/papers/v12/pedregosa11a.html

Stone, M. (1974). Cross-validatory choice and assessment of statistical predictions. *Journal of the Royal Statistical Society, Series B*. https://www.jstor.org/stable/2984809

Tibshirani, R. (1996). Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society, Series B*. https://www.jstor.org/stable/2346178

Appendix A. Data Summary

This project used the UCI Student Performance Dataset, specifically the combined Portuguese-language dataset. The dataset includes:

- 649 observations (students)
- 32 predictors describing:
  - Demographics (age, sex, family size, address type)
  - Parental background (education, occupations)
  - Social/behavioral factors (study time, absences, alcohol consumption, free time, going out)
  - School support indicators (internet access, tutoring, paid classes)
  - Interim grades (G1, G2)
- Target variable: G3 (final grade), numerical (0–20)
- Explicitly removed G1 and G2 to prevent target leakage.

Appendix B. Data Cleaning and Feature Preparation

Our team applied a consistent preprocessing pipeline before modeling:

Row Handling

Dropped any duplicate rows. Confirmed there were no missing values, but the code is written to drop missing rows if present.

Encoding and Transformation

Converted binary "yes"/"no" features (schoolsup, famsup, paid, activities, nursery, higher, internet, romantic) to 0/1.

Encoded simple categories:

- famsize: "LE3" $\rightarrow$ 0, "GT3" $\rightarrow$ 1

- address: "R" $\rightarrow$ 0, "U" $\rightarrow$ 1

Clipped extreme absences values at 40 to reduce the influence of outliers.

Grouped rare categories (fewer than 10 rows) in Mjob, Fjob, reason, and guardian into "other".

Applied one-hot encoding with `pd.get_dummies(..., drop_first=True)` to remaining categorical variables.

Scaling

For kNN, all features were scaled using StandardScaler, and we used an 80/20 train–test split with `random_state=42`.

For advanced models, we used a ColumnTransformer inside a Pipeline to scale numeric columns and one-hot encode any remaining categoricals.

Appendix C. Models and Hyperparameters

We implemented and compared the following regression models to predict G3:

k-Nearest Neighbors (kNN) Regressor

Evaluated multiple k values: 1, 3, 5, 7, 9, 11, 15, 21, 35, 42, 50. Selected the best k based on lowest RMSE on the test set.

Decision Tree Regressor

Basic CART regression tree with `random_state=42`.

Random Forest Regressor

`n_estimators = 300`, `random_state = 42`.

Gradient Boosting Regressor

`n_estimators = 300`, `learning_rate = 0.05`, `max_depth = 3`, `random_state = 42`.

Bagging Regressor

`n_estimators = 100`, `random_state = 42`.

Lasso Regression

Linear model with L1 regularization, `alpha = 0.01`.

Performance was evaluated primarily using RMSE and R^2 on the held-out test set, and compared against a baseline model that predicts the mean training G3.

Appendix D. Validation Plots and Diagnostics

To validate and interpret our models, we created the following visualizations:

kNN Model Diagnostics

- RMSE vs k plot to see how predictive error changes as neighborhood size increases.
- R^2 vs k plot with a horizontal line at $R^2 = 0$ to compare against the baseline mean predictor.

These plots showed that very small k values overfit, while k in the range of about 20–35 gave the best and most stable performance.

Model Comparison Plots

Bar chart of RMSE for: Baseline (Mean G3), kNN (best k), Decision Tree, Random Forest, Gradient Boosting, Bagging, and Lasso.

Bar chart of R^2 for the same models, including the baseline.

These figures highlight that Gradient Boosting has the lowest RMSE and highest R^2, with Lasso, Random Forest, and tuned kNN also clearly outperforming the baseline.

Gradient Boosting Validation

5-fold cross-validation RMSE bar plot for Gradient Boosting: Shows similar RMSE across folds with a small standard deviation, indicating stable performance across different splits.

Residuals vs Fitted plot for Gradient Boosting: Residuals are roughly centered around zero with no strong pattern or funnel shape, suggesting low systematic bias.

QQ plot of residuals: Residuals mostly follow the reference line with mild deviations in the tails, consistent with approximately normal error behavior for this dataset.