Decision Tree and Random Forest for Student Performance Prediction

Shaan Ali Remani, Jose Pedro Pessoa Dos Santos, Chen Chin-Lan, Poh Har Yap

4 March 2025

Word Count: 1034 words[1]

[1] Excluding footnotes, captions, and appendices.

Table of Contents

Table of Figures

1. Introduction

This report explores *Decision Tree* and *Random Forest* approaches to classifying the '*Mathematics*' Student Performance dataset, Cortez & Silva (2008).[2] We successfully predict the final student grade, G3, with an out-of-sample testing accuracy of **72.27%** for Decision Tree and **66.39%** for Random Forest.[3] Both approaches are visualised through an interactive *Shiny* app for our non-expert clients.

2. Decision Trees vs Random Forests

Classification tasks in machine learning aim to understand the relationship between the explanatory variables (feature space) and the response variable (target) by fitting a model that maps the feature space to the response (James et al. 2013). Decision trees and random forests are a subset of such tasks. In these *supervised learning* algorithms a target, Y, is split into several classes and a feature matrix, X, trains the model.

Decision trees split the dataset into regions by recursively choosing the best feature and corresponding threshold that minimises impurity, measured by the *Gini Index* (**Appendix I**). Each split divides the data into two subregions, making the best decision at each step. This results in a tree structure with internal *nodes* where decisions are made, *branches* representing pathways from one decision to the next, and *terminal nodes* which provide the final predictions by majority voting.

Instead of growing a single decision tree, random forests sample multiple trees (i.e. a *forest*) using *bootstrapped* training data (i.e. *random*).[4] Each tree produces an independent prediction, which is averaged to determine final model performance. To reduce correlation among trees, random forests introduce further randomness: when splitting a node, each tree considers only a random subset of features (often m=$p^{1/2}$, where p is the number of features in the dataset (James et al. 2013)).

3. Visualisation with Shiny App

Shiny for Python is used to visualise the impact of *hyperparameter* selection on tree structure. Our interactive app allows clients to tune hyperparameters for decision trees and random forests, and compare model outputs in real time. Clients can toggle between decision trees and random forests, and adjust sliders for α and *Max Depth*.[5] As these parameters change, the app dynamically updates the tree structure and displays the tree's training and test accuracy.

[2] Cortez & Silva (2008) examine student performance in Mathematics and their native language Portuguese. We redact the *'Portuguese'* dataset (645 observations) since it exceeds the 500 observation boundary, and limit our analysis to the *'Mathematics'* dataset.

[3] *Weighted F1 Score* is used to evaluate performance as it performs better than classification error rate on imbalanced data.

[4] Bootstrapping is a resampling technique in which random samples of the dataset are drawn with replacement, allowing some observations to be selected multiple times and some to be excluded entirely in a given sample. See Dikta and Scheer (2021).

[5] We also include the option to visualise derivative models of Random Forest, *AdaBoost* and *Gradient Boost*, through Shiny, even though these models were not included in our own analysis.

Hyperparameter selection balances the trade-off between model complexity and model fit. A model which is too complex tends to overfit the training data, capturing noise rather than patterns and performing poorly out-of-sample (James et al. 2013). Similarly, a simple model fails to capture the underlying relationship between X and Y. We tune 2 hyperparameters in our analysis:[6]

> α: Regulates *pruning* by removing branches that offer little improvement. Higher values lead to smaller, simpler trees. **Range: [0,0.05]**
>
> Max Depth: Defines the maximum depth of the tree, representing the longest path from the root to a leaf node. Deeper trees have higher training accuracy but risk overfitting. **Range: [0,20]**

4. Data Processing and Results Analysis

Our dataset, '*Mathematics*', examines secondary school students' grades in Mathematics. The features, broadly categorised into demographics, social factors, and school factors, are used to predict the target G*3* (**Appendix II**). Mid-year assessment 1, G*1*, and mid-year assessment 2, G*2*, are excluded as features due to their extraordinarily high correlation with G*3* (*0.826* and *0.919*, respectively), compared to the next highest correlation (*0.39).*[7]

We classify G*3* into two bands:

$$G3 \geq 10 \text{ pass,}$$
$$G3 < 10 \text{ fail.}$$

where *10* is chosen as the boundary to maintain consistency with the Portuguese grading system (Government of Portugal 2005, art. 15(2)(a)).

The Shapley values in *Figure 1* indicate that student absences, and family dynamics are among the most influential predictors of G*3*.[8] Prima facie, socio-economic background and attendance play key roles in student performance. Conversely, Shapley values with dispersion close to 0 suggest redundancy in the dataset, as they exhibit near-zero correlation with G3 and a lack of predictive power. To counteract this, we combine *Mother's education* and *father's education* into the new features *Average parental education* and *Maximum parental education*, and redact *Address*.

Decision Trees and Random Forest algorithms require numerical values as inputs so categorical features are transformed through industry-standard one-hot encoding.

[6] Our app allows clients to adjust additional hyperparameters not tuned in our own analysis. These highlight the challenge of preventing Random Forest and its derivative models from overfitting training data. These additional hyperparameters are explained in (**Appendix IV**).

[7] Cortez and Silva (2008) note that including G1 and G2 results in a classification model that relies predominantly on these variables. The exclusion forces the model to rely on demographic, social, and school-related factors, which can provide insights into areas where teachers might intervene.

[8] Shapley values, deriving from game theory, display the correlation of features with G3. These can be thought of as a measure of the individual influence each feature has on G3. See SHAP (2025).

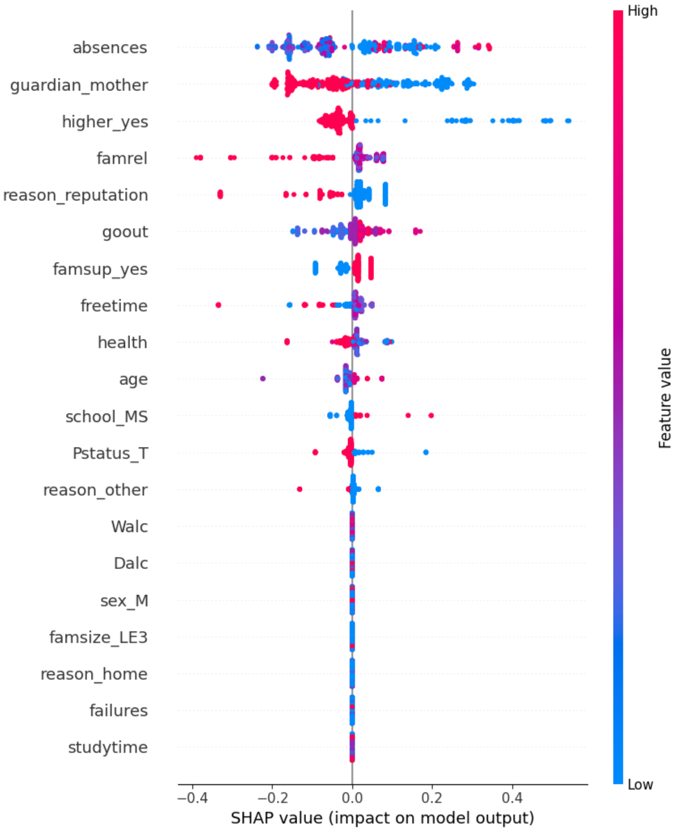


Figure 1 Shapley values for all features in the original dataset.

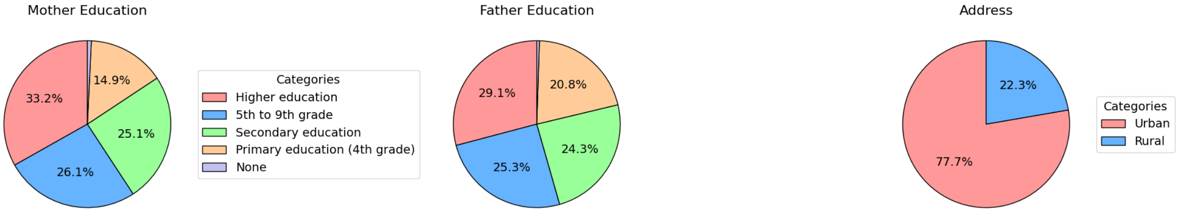


Figure 2 Pie charts illustrating the distribution of Mother's education and Father's education, as well as Address.

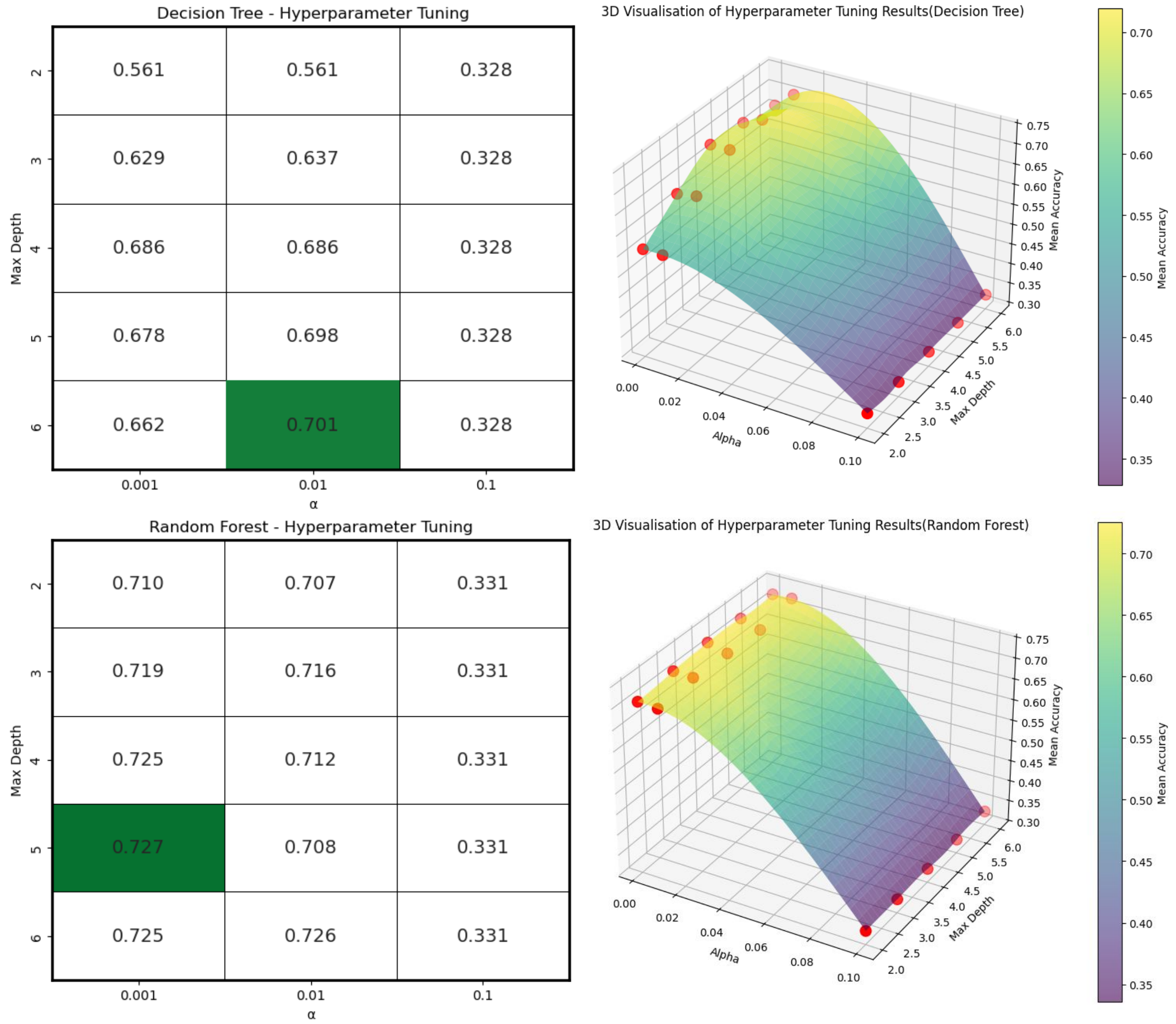
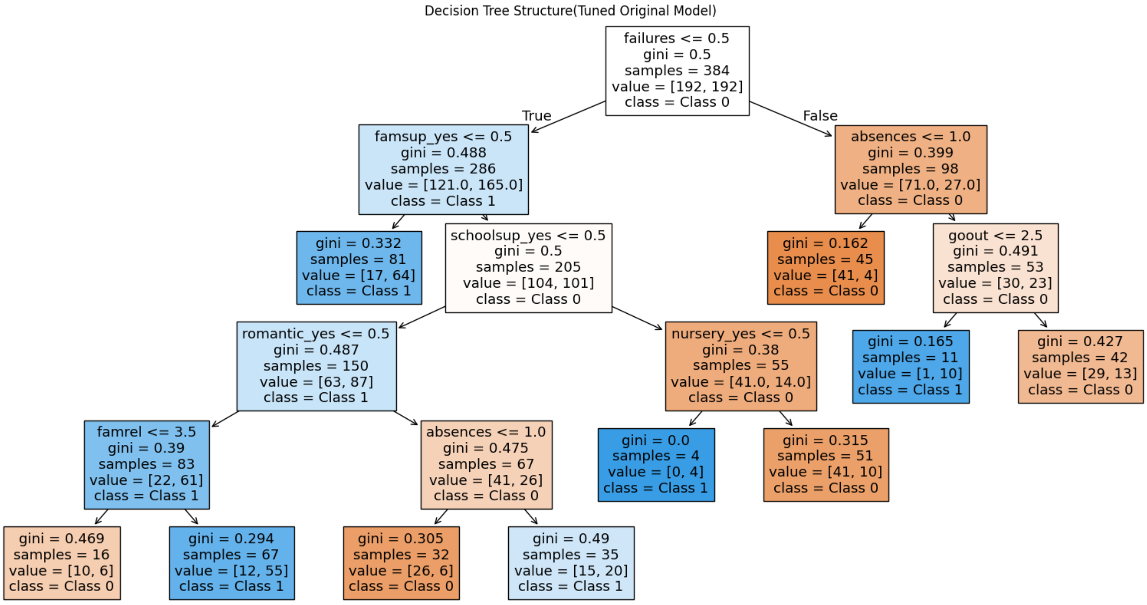


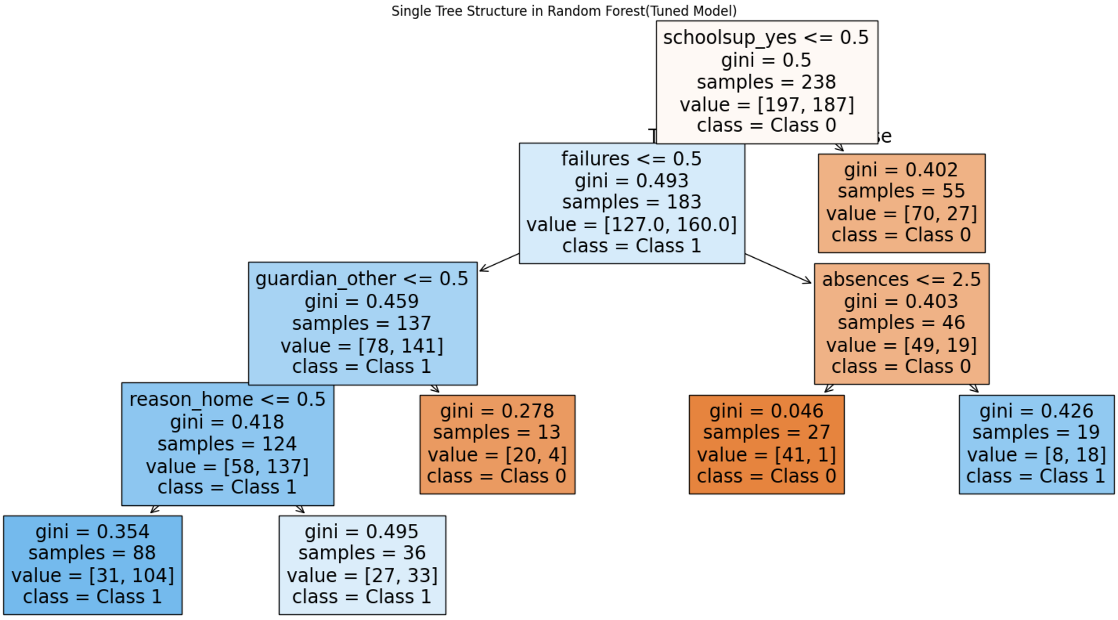
Figure 3 (a) Upper panels depict Decision Tree results and (b) lower panels depict Random Forest results. Left panels show how α and Max Depth affect the accuracy of our model. Right panel visualises this in a 3D surface; as α decreases, the penalty term is reduced and so model performance increases, as Max Depth increases, the model has more branches and nodes leading to a stronger performance. Also, it is clear to see that small changes in alpha have a larger impact on model performance than changes in Max Depth.

Using *GridSearchCV* from *Scikit-Learn*, we search for the optimal hyperparameters and perform 10-fold cross-validation (**Appendix III**) to evaluate model performance. *Split* and *leaf* parameters are untuned in our analysis, however, we set higher values for random forest to increase regularisation/penalty since this is a more complex model and easily leads to overfitting. We determine the best decision tree cross-validation accuracy is *70.1%*, achieved with *α*=*0.01* and *Max Depth*=6 (Figure 3a). Similarly, the best random forest cross-validation accuracy is *72.7%*, also with *α*=*0.001* and *Max Depth*=*5* (Figure 3b). These tuned, and therefore optimised, decision trees are shown in Figure 4 and Figure 5.



Decision Tree Structure(Tuned Original Model)

Figure 4 Visualisation of Best Decision Tree Model. Key Predictors include number of past failures, absences, and support at home.



Single Tree Structure in Random Forest(Tuned Model)

Figure 5 Visualisation of Best Random Forest Model. Key Predictors include school support, failures, and support at home

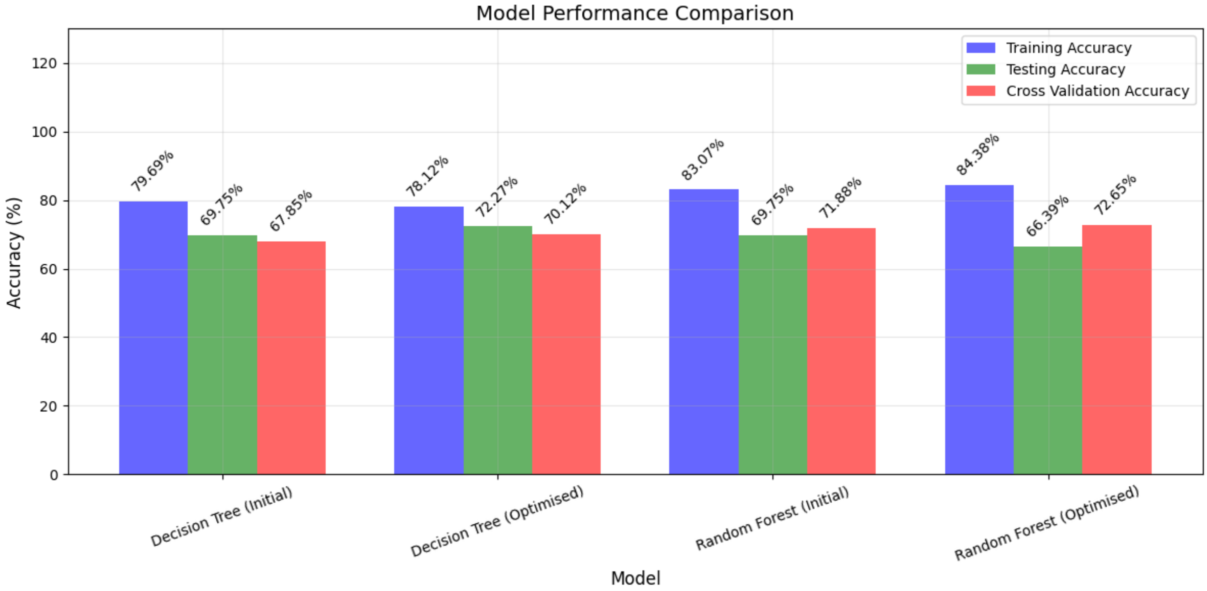


Figure 6 Model Performance across tuned and untuned Decision Trees and Random Forests

Our optimisation improves Decision Tree cross-validation accuracy by 2.27% and Random Forests' by 0.77%. Out-of-sample testing performances of 72.27% for Decision Tree and 66.39% for Random Forest indicates that our decision tree generalises well whereas our random forest does not. This is confirmed by the ROC curves in Figure 7, where the decision tree curve is smoother than the jagged random forest curve.

Notably, the Random Forest model exhibits a larger gap between training and testing accuracies compared to the Decision Tree model. This indicates that while the Random Forest achieves higher performance on the training data, it tends to overfit, capturing noise rather than generalisable patterns. In contrast, Figure 8 shows that Decision Tree's smaller accuracy gap reflects better generalisation on unseen data, making it a more reliable choice for this particular task. As a result, we conclude that the optimised decision tree model performs best for this task.

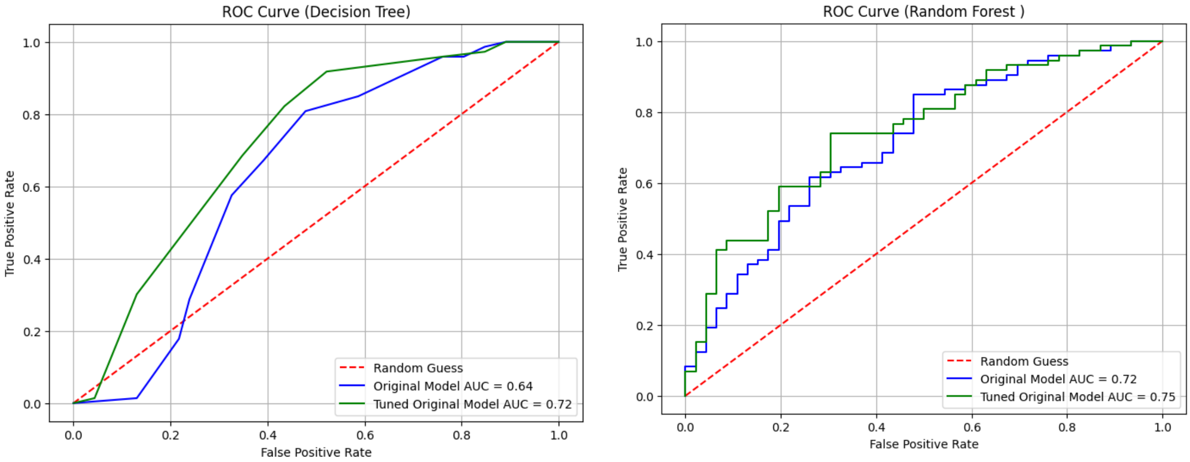


Figure 7 ROC Curves showing a smoother performance for Decision Tree than Random Forest

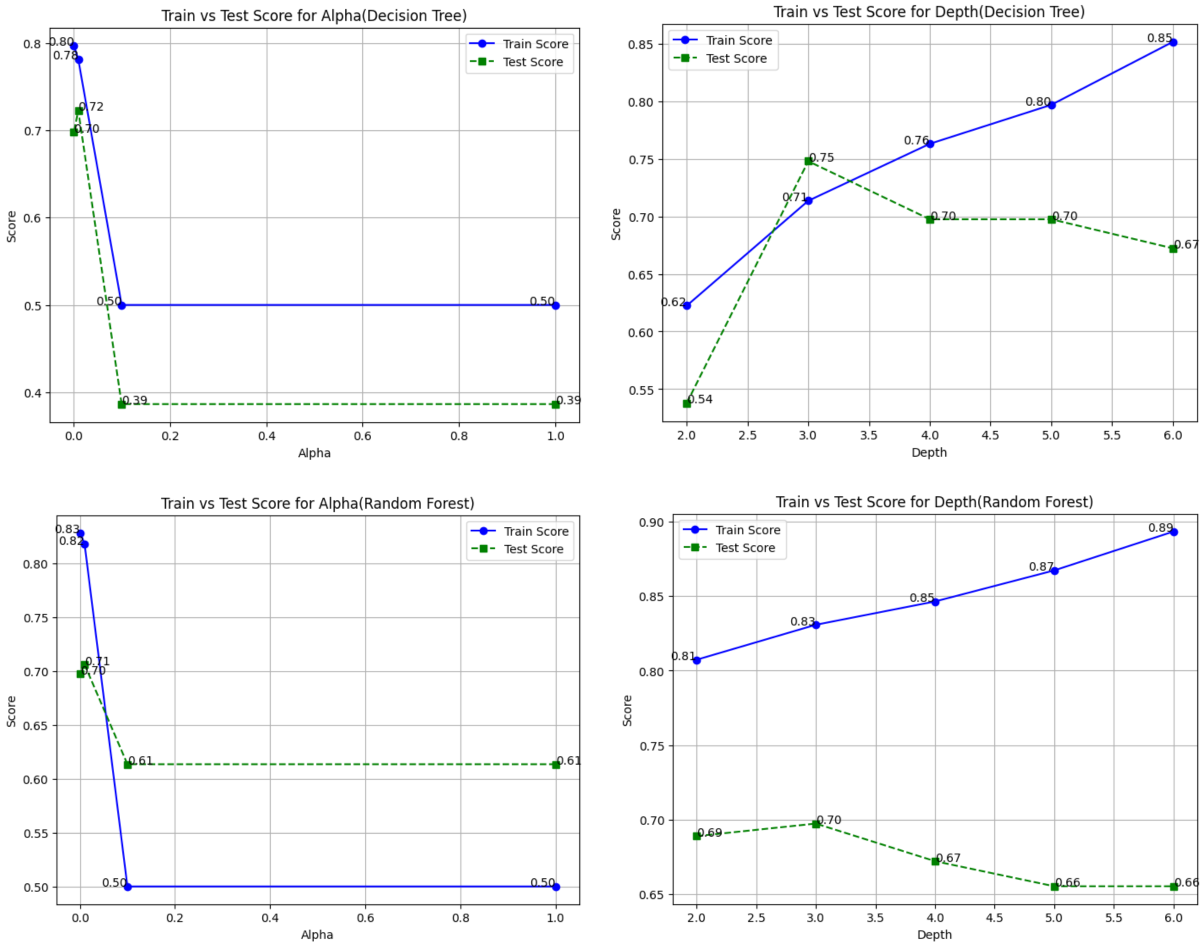


Figure 8 Line graphs illustrating testing and training performance as α and Max Depth varies. Note, in particular, the divergence in train and test score in the lower right panel (overfitting).

5. Conclusion

The availability of open-source machine learning algorithms, including Decision Trees and Random Forests, has sparked a flurry of statistical innovation. Whilst previously reserved for mathematicians, statisticians and some professionals, machine learning is now ubiquitous in data analysis – from healthcare and school performance tracking to governance and finance. This report and corresponding Shiny app demonstrate not only the possibilities of building one's own model but also the intricacies of finetuning model hyperparameters and avoiding overfitting training data. We have shown that it is possible to predict student pass or fail with an out-of-sample accuracy of **72.27%** using Decision Trees. We invite current and prospective clients to speak to a representative to discuss our machine learning solutions.

References

Includes References in Appendices

Ceriani, L. and Verme, P. (2012) The origins of the Gini index: extracts from Variabilità e Mutabilità (1912) by Corrado Gini. *The Journal of Economic Inequality*. [Online] 10 (3), 421–443. Available at: doi:10.1007/s10888-011-9188-x (Accessed: 3 March 2025).

Cortez, P. and Silva, A.M.G. (2008) Using data mining to predict secondary school student performance. In: *Proceedings of 5th Annual Future Business Technology Conference*. [Online]. 1 April 2008 Porto, Portugal. pp. 5–12. Available at: http://repositorium.sdum.uminho.pt/bitstream/1822/8024/1/student.pdf (Accessed: 1 March 2025).

Dikta, G. and Scheer, M. (2021) *Bootstrap Methods: With Applications in R*. Cham, Springer International Publishing.

Hastie, T., Tibshirani, R. and Friedman, J. (2009) *The Elements of Statistical Learning*. Springer Series in Statistics. [Online]. New York, NY, Springer.

James, G., Witten, D., Hastie, T. and Tibshirani, R. (2013) *An Introduction to Statistical Learning*. Springer Texts in Statistics. New York, NY, Springer.

Government of Portugal (2005) *Decreto-Lei n.º 42/2005, de 22 de Fevereiro*. Ministério da Ciência, Inovação e Ensino Superior. Diário da República, Série I-A, n.º 38, pp. 1533-1542. Available at: https://files.diariodarepublica.pt/1s/2005/02/037a00/14941499.pdf (Accessed: 1 March 2025).

SHAP (2025) *An introduction to Explainable AI with Shapley Values – SHAP Latest Documentation*. [Online]. 2025. Available at: https://shap.readthedocs.io/en/latest/example_notebooks/overviews/An%20introduction%20to%20explainable%20AI%20with%20Shapley%20values.html (Accessed: 3 March 2025).

Appendix I – Gini Index

Ceriani and Verme (2012 p421) note that 'Gini proposed no less than 13 formulations of his index, none of which is known today to the large public.' For a technical introduction, see Hastie et al. (2009).

Gini Index:

$$G = \sum_{k=1}^{K} \widehat{p_{mk}}(1 - \widehat{p_{mk}})$$

where p_{mk} denotes the proportion of training observations in the mth node that belong to the kth class. G is a measure of total variance across the K classes. If p_{mk} is close to *0* or *1*, G is close to *0*.

Appendix II – Feature Space

This table is reproduced with modifications from Cortez & Silva (2008, p7). The original structure and content belong to the authors. *G1* and *G2* are redacted for this version and *Parent_Edu* and *Max_Parent_Edu* replace Mother's Education and Father's Education.

Attribute	Description (Domain)
sex	student's sex (binary: female or male)
age	student's age (numeric: from 15 to 22)
school	student's school (binary: Gabriel Pereira or Mousinho da Silveira)
Pstatus	parent's cohabitation status (binary: living together or apart)
Parent_Edu[a]	average parental education (numeric: from 0 to 4)
Mjob[b]	mother's job (nominal)
Max_Parent_Edu[a]	maximum parental education (numeric: from 0 to 4)
Fjob[b]	father's job (nominal)
guardian	student's guardian (nominal: mother, father or other)
famsize	family size (binary: ≤ 3 or > 3)
famrel	quality of family relationships (numeric: from 1 – very bad to 5 – excellent)
reason	reason to choose this school (nominal: close to home, school reputation, course preference or other)
traveltime	home to school travel time (numeric: 1 – < 15 min., 2 – 15 to 30 min., 3 – 30 min. to 1 hour, or 4 – > 1 hour)
studytime	weekly study time (numeric: 1 – < 2 hours, 2 – 2 to 5 hours, 3 – 5 to 10 hours or 4 – > 10 hours)
failures	number of past class failures (numeric: n if 1 ≤ n < 3, else 4)
schoolsup	extra educational school support (binary: yes or no)
famsup	family educational support (binary: yes or no)
activities	extra-curricular activities (binary: yes or no)
paidclass	extra paid classes (binary: yes or no)
internet	Internet access at home (binary: yes or no)
nursery	attended nursery school (binary: yes or no)
higher	wants to take higher education (binary: yes or no)
romantic	with a romantic relationship (binary: yes or no)
freetime	free time after school (numeric: from 1 – very low to 5 – very high)
goout	going out with friends (numeric: from 1 – very low to 5 – very high)
Walc	weekend alcohol consumption (numeric: from 1 – very low to 5 – very high)
Dalc	workday alcohol consumption (numeric: from 1 – very low to 5 – very high)
health	current health status (numeric: from 1 – very bad to 5 – very good)
absences	number of school absences (numeric: from 0 to 93)
G3 (TARGET)	final grade (numeric: from 0 to 20)

Figure 9 Table of Features

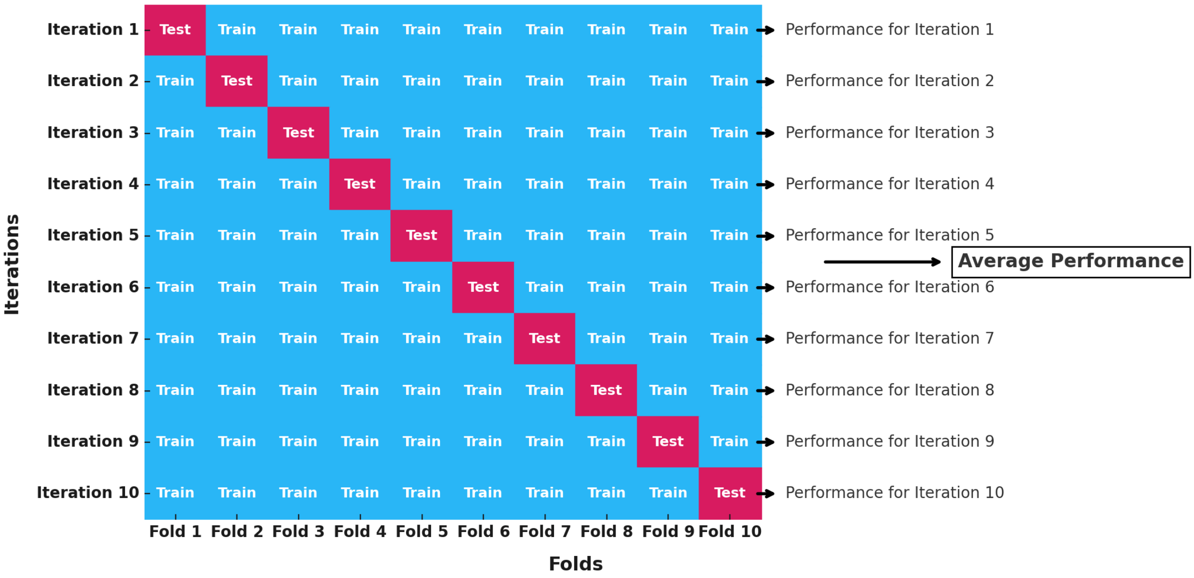
[a] 0 – none, 1 – primary education (4th grade), 2 – 5th to 9th grade, 3 – secondary education, 4 – higher education.

[b] Teacher, health care related, civil services (e.g., administrative or police), at home, or other.

Appendix III – 10-Fold Cross-Validation

K-fold Cross-Validation is a technique used to estimate the performance of machine learning models to reduce overfitting and data variability. The training dataset is divided into *K* sections. The model is subsequently trained and tested *K* times, each time using one fold for testing and the remaining *K-1* sections for training. In our analysis, we choose *K=10* (a commonly used value) as it has been shown to result in a model skill estimate with low bias and reasonable variance (James et al. 2013). Choosing a high *K* value leads to high computational cost, while a low *K* value has higher variance.

10-Fold Cross Validation Visualisation



	Fold 1	Fold 2	Fold 3	Fold 4	Fold 5	Fold 6	Fold 7	Fold 8	Fold 9	Fold 10	
Iteration 1	Test	Train	Train	Train	Train	Train	Train	Train	Train	Train	Performance for Iteration 1
Iteration 2	Train	Test	Train	Train	Train	Train	Train	Train	Train	Train	Performance for Iteration 2
Iteration 3	Train	Train	Test	Train	Train	Train	Train	Train	Train	Train	Performance for Iteration 3
Iteration 4	Train	Train	Train	Test	Train	Train	Train	Train	Train	Train	Performance for Iteration 4
Iteration 5	Train	Train	Train	Train	Test	Train	Train	Train	Train	Train	Performance for Iteration 5
Iteration 6	Train	Train	Train	Train	Train	Test	Train	Train	Train	Train	Performance for Iteration 6
Iteration 7	Train	Train	Train	Train	Train	Train	Test	Train	Train	Train	Performance for Iteration 7
Iteration 8	Train	Train	Train	Train	Train	Train	Train	Test	Train	Train	Performance for Iteration 8
Iteration 9	Train	Train	Train	Train	Train	Train	Train	Train	Test	Train	Performance for Iteration 9
Iteration 10	Train	Train	Train	Train	Train	Train	Train	Train	Train	Test	Performance for Iteration 10

Figure 10 10-Fold Cross Validation Illustration

Appendix IV – Tuneable Features in Shiny App

Further to the hyperparameters detailed in the main report (α and M*ax Depth*), our interactive Shiny app permits clients to explore and tune model behaviour by adjusting the parameters below. Tuning these allows clients to interactively explore the sensitivity between model complexity and out-of-sample testing performance, that is, just how easy it is to overfit training data.

Number of Estimators: Controls the minimum number of samples at leaf nodes. *Range:* [1, 100]
Max Features: Determines the number of features considered when determining the best split at each node. *Range:* [1, 60]
Tree Selection: Allows users to select an individual decision tree from the random forest for visualisation. *Range:* Tree 1 to Tree 50
Learning Rate: Adjusts the contribution of each *weak learner* to the final prediction. *Range:* [0, 1]