Predicting Diamond Outcomes: A Regression Study

José Pires — Student ID: 23115639

https://github.com/jpires0405/MachineLearning-Coursework1.git

1 Exploratory Data Analysis

The dataset consists of 10,000 training samples and 1,000 test samples, each described by 30 input features. Three are categorical—`cut`, `color`, `clarity`—representing ordered quality grades. The remaining 27 are continuous: seven physical properties (`carat`, `depth`, `table`, `price`, `x`, `y`, `z`) and twenty pre-computed transformations (`a1`–`a10`, `b1`–`b10`).

No missing values were found. Figure 1 shows Pearson correlations between `outcome` and the ten most linearly correlated numeric features. The strongest individual correlations remain moderate ($|\rho| \leq 0.65$), indicating that no single feature is linearly sufficient and that complex, non-linear interactions govern the target. This motivated abandoning Ridge regression in favour of gradient-boosted trees. Continuous features were standardised; categorical features were one-hot encoded with unknown-category handling.

2 Model Selection

A Ridge baseline ($R^2 = 0.282$) confirmed substantial non-linear structure. Two ensemble candidates were then evaluated: Random Forest and `HistGradientBoostingRegressor`.

`HistGradientBoosting` was preferred for two technical reasons. First, it discretises each continuous feature into up to 255 histogram bins before tree construction, reducing the split-search cost from $O(n)$ to $O(B)$ per node ($B \ll n$), yielding faster training on the 10,000-row dataset. Second, its additive, stage-wise fitting naturally captures high-order non-linear interactions—precisely the structure indicated by the moderate pairwise correlations in Figure 1. All models were sourced exclusively from `scikit-learn`.

Table 1: Cross-validated performance comparison (5-Fold, seed = 123).

Model	Configuration	CV Mean R^2	CV Std Dev
Ridge	Baseline ($\alpha = 1.0$)	0.282	0.014
Random Forest	Default	0.452	0.014
HistGradientBoosting	Default	0.460	0.015
HistGradientBoosting	Tuned	**0.472**	—

The tuned configuration used: learning rate = 0.05, max iterations = 300, max depth = 3, and ℓ_2 regularisation = 0.1. Figure 2 shows predicted vs. actual values on a 20% hold-out split, demonstrating the model captures the general trend with residual spread attributable to the non-linear, unlabelled `a`/`b` features.

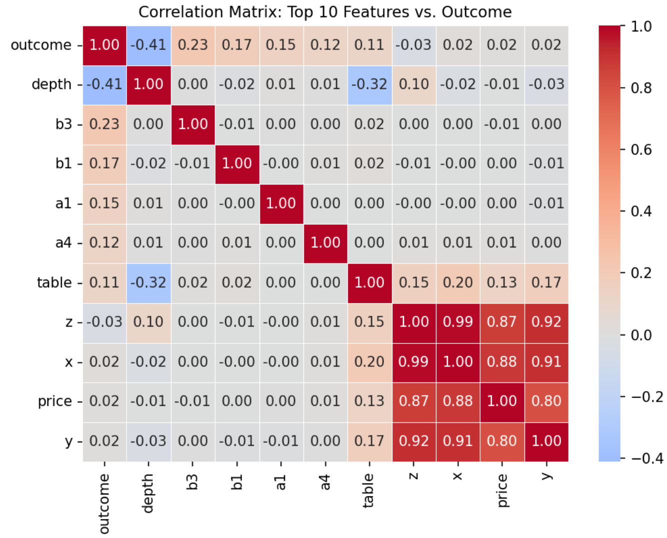

Figure 1: Pearson correlations: top 10 numeric features vs. `outcome`. Moderate $|\rho|$ values justify non-linear modelling.

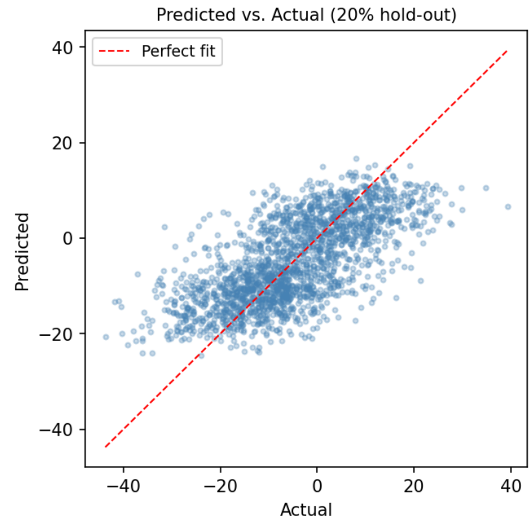

Figure 2: Predicted vs. actual on the 20% hold-out set (tuned HistGBR).

3 Training & Evaluation

All models were evaluated using 5-Fold cross-validation (`KFold`, `shuffle=True`, `random_state=123`) with R^2 as the scoring metric. Preprocessing was embedded within an `sklearn.pipeline.Pipeline` coupled with a `ColumnTransformer`, ensuring that scaling and encoding were fitted exclusively on each training fold, preventing data leakage.

Hyperparameter tuning was performed via `RandomizedSearchCV` (15 iterations) over discrete grids for learning rate, iteration count, tree depth, and ℓ_2 regularisation. The best configuration raised mean R^2 from 0.460 (default) to 0.472, after which the estimator was refit on the full training set before generating test predictions.

4 Code Supplement

The full codebase is hosted at `https://github.com/jpires0405/MachineLearning-Coursework1.git` in a modular structure: `src/features.py` defines the preprocessing pipeline, `src/models.py` registers candidates, `src/evaluate.py` implements CV logic, and `src/train.py` orchestrates training, tuning, and submission generation.

All random seeds are fixed at 123. The final submission is a single-column CSV (`yhat`, 1,000 rows) validated by programmatic assertions before writing to disk. Feature branches were used throughout development (`feature/linear-baselines`, `feature/ensemble`, `feature/tuning`), with `main` reserved for the final deliverable.