5CCSAMLF Coursework 1 Report

Lucas Perez Reis Lobo (k23075501) — February 2026

1 Exploratory Data Analysis

The dataset provided in the coursework contains 10,000 training samples and 1,000 test samples with 30 features and a continuous target variable. No missing values or duplicates were found.

Target Variable The target is approximately symmetric (mean = -5.0, std = 12.7, range $[-44.9, 39.7]$) with near-zero skewness and few extreme outliers. Thus, not requiring target transformation.

Feature Taxonomy The features in this data set were divided into four groups (Table 1).

Table 1: Feature groups and key properties.

Group	Count	Key Property
Categorical	3	Balanced ($\hat{H} \in [0.85, 0.96]$)
Interpretable	7	Multicollinear ($r \leq 0.99$)
Latent Uniform	10	Independent ($\lvert r\rvert < 0.05$)
Latent Gaussian	10	Correlated block ($\lvert r\rvert \leq 0.3$)

The interpretable features, {carat, price, x, y, z} form a cluster with $r > 0.95$, measured via Pearson correlation $r_{xy} = \mathrm{Cov}(X, Y)/(\sigma_X \sigma_Y)$.

Category balance was quantified using normalised Shannon entropy $\hat{H} = -(\log_2 K)^{-1} \sum_{k=1}^{K} p_k \log_2 p_k$, where values near 1 indicate uniform class distributions. High entropy ($\hat{H} > 0.85$) confirmed no category merging or rebalancing was needed.

Latent features show stronger marginal correlations with the target ($|r|$ up to 0.22) than interpretable features ($|r| < 0.10$), although all are weak. This suggests that there is a **nonlinear interaction**.

Preprocessing Decisions Based on what was found in the EDA phase we preprocess the data as follows: (1) one-hot encode categoricals with `drop_first` (balanced categories require no special handling); (2) drop price, x, y, z (redundant with carat at $r > 0.95$, reducing noise without losing information); (3) engineer 23 features from latent variables to capture nonlinear interactions suggested by weak marginal but potentially strong joint effects (Section 3); (4) no target transformation (symmetric distribution; log/sqrt transforms tested and hurt R^2 by 2–6%).

Dropping these variables reduced variance without losing accuracy, as tree-based models exploit relative ordering rather than absolute scale.

2 Model Selection

Ten algorithms from five different families were evaluated using 5-fold cross-validation with $R^2 = 1 - \sum_i (y_i - \hat{y}_i)^2 / \sum_i (y_i - \bar{y})^2$, consistent random seed (123), and appropriate preprocessing (StandardScaler for linear/distance models; passthrough for trees).

Table 2: Model comparison (5-fold CV, default hyperparameters).

Family	Algorithm	CV R^2	Std
Linear	Ridge	0.283	0.013
Linear	Lasso	0.286	0.012
Distance	KNN	0.085	0.016
Neural Net	MLP	0.391	0.023
Bagging	Random Forest	0.455	0.014
Boosting	GradientBoosting	0.469	0.018
Boosting	XGBoost	0.383	0.018
Boosting	LightGBM	0.449	0.017

Analysis **Linear models** ($R^2 \approx 0.28$): The ceiling confirms nonlinear structure. Similar Ridge/Lasso performance suggests many small, non-zero effects (no sparse solution). **KNN** ($R^2 = 0.085$): The high-dimensionality of the dataset, with 30 features, rendered local averaging ineffective. **MLP** ($R^2 = 0.39$): Outperforms linear models but underperforms trees, $n = 10{,}000$ (training samples) is insufficient for a 3-layer network to learn interactions that tree splits capture structurally. **Tree ensembles** ($R^2 \approx 0.45$–0.47): Boosting dominates via sequential residual correction $F_m(x) = F_{m-1}(x) + \eta \cdot h_m(x)$.

Selection A stacking ensemble of GradientBoosting + XGBoost + LightGBM was selected, with a Ridge regression to learn the optimal weights for each model's predictions: $\hat{y} = \beta_0 + \beta_1 \hat{y}_{\mathrm{GB}} + \beta_2 \hat{y}_{\mathrm{XGB}} + \beta_3 \hat{y}_{\mathrm{LGB}}$. Three boosting implementations provide diversity through different splitting strategies and regularisation approaches. Stacking was chosen to reduce estimation variance by combining independently regularised boosting learners.

3 Model Training and Evaluation

Feature Engineering Starting from 26 numerical features (after dropping multicollinear ones), 23 engineered features were added (Table 3). **Pair interactions** ($a_i \cdot b_i$) capture joint effects between matched latent pairs, motivated by weak marginal but strong combined signals. **Group aggregations** ($\sum a_i$, $\sum b_i$, difference) encode overall magnitude and direction of latent groups.

Squared Gaussian terms (x_j^2) capture symmetric nonlinear effects where both $x > 0$ and $x < 0$ may predict similarly.

Rejected approaches: (1) polynomial expansion on Gaussian features (degree 2, adding 55 terms) increased dimensionality without improving signal, dropping R^2 to 0.465; (2) target transformations (shifted-log, signed-sqrt) distorted the symmetric distribution the model learned effectively from; (3) extended 69-feature set with cross-group interactions added noise ($R^2 = 0.478$, worse than 49 features).

Table 3: Engineered features (26 original + 23 = 49 total).

Technique	n	Form
Pair interactions	10	$a_i \cdot b_i$
Group aggregations	3	$\sum a_i$, $\sum b_i$, diff
Squared Gaussian	10	x_j^2

Hyperparameter Tuning RandomizedSearchCV (100 iterations, 5-fold CV) was applied to each base learner. Initial experiments in draft notebooks compared grid search (1,296 combinations, 67 min) versus randomised search (100 iterations, 2 min) for GradientBoosting, finding equivalent final R^2 (<0.1% difference) in a fraction of the time.

Table 4: Tuned model performance.

Model	CV R^2	Key Parameters
GradientBoosting	0.478	η=0.05, depth=2, n=200
XGBoost	0.480	η=0.03, depth=2, n=400, λ=5
LightGBM	0.480	η=0.03, depth=2, n=500, λ=5
Stacking	**0.481**	Ridge meta-learner (α=1.0)

Cross-Model Patterns All models converge on: (1) **shallow trees** (depth=2, $\leq$4 leaves per tree), acting as weak learners that boosting aggregates; (2) **low learning rates** ($\eta \in [0.03, 0.05]$) implementing shrinkage $F_m(x) = F_{m-1}(x) + \eta \cdot h_m(x)$ to prevent overfitting; (3) **subsampling** (0.7–0.9), acting as stochastic regularisation; (4) **strong L_2 penalties** ($\lambda = 5$ in XGBoost/LightGBM), confirming moderate signal-to-noise.

Evaluation **Final CV $R^2 = 0.481 \pm 0.018$.** The learning curve shows validation R^2 flattening beyond ~6,000 samples while the train-validation gap remains moderate, indicating we approach the irreducible noise ceiling rather than being data-limited. Residuals are centred at zero with approximately symmetric distribution and mild heteroskedasticity at extreme predictions.

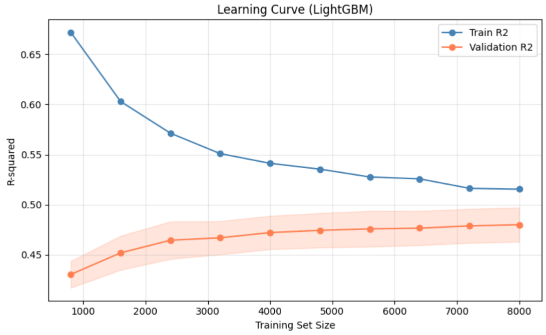

Figure 1: Learning curve showing validation R^2 flattening beyond ~6,000 samples, indicating approach to irreducible noise ceiling.

Performance Progression The largest gain (+0.186) came from the model family choice (linear → boosting). Tuning and ensembling yielded incremental improvements, consistent with the ~0.48 noise ceiling.

Table 5: Improvement across stages

Stage	CV R^2	Δ
Ridge baseline	0.283	—
GB (default)	0.469	+0.186
GB (tuned)	0.478	+0.010
Stacking (tuned)	0.481	+0.003

Conclusion Model family choice (linear → boosting) was the predominant factor in performance, accounting for 95% of the total R^2 improvement. Tuning and ensembling provided smaller but gains, with all approaches converging near $R^2 \approx 0.48$, suggesting this represents the noise ceiling of the dataset.

4 Code Supplement

Repository: `https://github.com/LucasPRLobo/ml-cw1`

- `labs/1_EDA.ipynb`: Data quality, distributions, correlations, preprocessing decisions (8 visualisations).
- `labs/2_Model_Selection.ipynb`: 10 algorithms across 5 families, feature engineering, stacking.
- `labs/3_Model_Training.ipynb`: Hyperparameter tuning, ensemble evaluation, residual analysis, submission.

All notebooks are self-contained and reproducible (`random_state=123`). Additional draft notebooks documenting intermediate experiments (grid vs random search comparison, rejected feature engineering approaches) are available in `labs/drafts/`.