Machine Learning Choices for Diamond Price Prediction, A Controlled Empirical Study

Zekai Hu

International Business School Suzhou, Xi'an Jiaotong-Liverpool University, Suzhou, 215123, China
Zekai.Hu23@student.xjtlu.edu.cn

Abstract. Diamond price prediction is an attribute-based regression problem in which observable diamond characteristics explain a substantial but incomplete part of market price. This paper converts a public Diamonds-dataset baseline into a controlled empirical study of modelling choices. The workflow reproduces the baseline preprocessing and evaluates five hypotheses concerning log price transformation, volume feature engineering, categorical encoding, model family, and simple ensembling. RMSE is used as the primary metric, with MAE, R-squared, and cross-validation RMSE as supporting metrics. Results show that tree-based and boosting models outperform linear regression and support vector regression. The best individual extended model is CatBoost, while a simple ensemble of Random Forest, XGBoost, LightGBM, and CatBoost achieves the lowest test RMSE, reducing RMSE from 554.73 for the reproduced XGBoost baseline to 521.16. Segment-level diagnostics show broadly similar relative MAE across low-, mid-, and high-price groups, but dollar-denominated errors increase for high-price diamonds and repeated splits show mild high-price underestimation. These findings indicate that attribute-only valuation is accurate on average but remains limited in the upper price range, where richer market, certification, and transaction variables are likely needed.

Keywords: Diamond price prediction, tabular regression, supervised learning, gradient boosting, error analysis.

1 Introduction

Diamond valuation can be formulated as a supervised tabular regression task in which the model receives observable diamond attributes and predicts market price. Each observation in the Diamonds dataset describes one diamond through static features including carat, cut, color, clarity, depth, table, and geometric dimensions, while price is the target variable. The problem is therefore different from financial time-series forecasting because it focuses on how far a structured set of measurable product attributes can explain a continuous valuation target.

Accurate diamond price prediction is of substantial practical significance because consistent price estimation supports retail pricing, inventory management, customer comparison, and automated appraisal. The task is also difficult because the 4Cs and

© The Author(s) 2026
L. C. Yoong (ed.), *Proceedings of the 2026 2nd International Conference on Financial Innovation and Marketing Management (FIMM 2026)*, Advances in Economics, Business and Management Research 408,
https://doi.org/10.2991/978-94-6239-801-6_142

physical dimensions provide a strong objective basis for price prediction, while market prices may also reflect certification quality, brand reputation, retail channel, local demand, scarcity, and other premiums that are not observed in the dataset.

Recent studies support the feasibility of supervised learning for diamond price prediction. Non-linear machine learning models can exploit relationships among carat, cut, clarity, depth, table, and price, and broader benchmarks show that multiple regression models can be compared for price estimation under a shared experimental setting [1-3]. At the methodological level, recent tabular-learning research shows that gradient-boosted tree models remain highly competitive for structured data, while comparative work on boosting explains why XGBoost, LightGBM, and CatBoost are strong candidates for this task [4-6]. Related attribute-based valuation studies in housing also show that structured variables can support predictive modelling, although their market mechanisms differ from the diamond domain [7, 8].

Despite this progress, the literature leaves three related gaps. First, many studies report the best-performing estimator but do not systematically isolate the contribution of target transformation, feature engineering, encoding, and model family within the same pipeline. Second, performance is often summarized by aggregate metrics, which can hide price-segment heterogeneity. Third, comparisons may blur the distinction between a demonstration-oriented baseline workflow and a reproducible empirical study with explicit research objectives. This paper addresses these gaps by converting a publicly available baseline implementation into a controlled research workflow. The research objective is to identify which modelling choices most improve diamond price prediction and where the final model remains systematically less accurate.

2 Methodology

2.1 Dataset and Preprocessing

The experiment uses the classic Diamonds dataset employed by the public Kaggle baseline notebook [9]. The raw dataset contains 53,940 observations and 10 substantive variables, namely carat, cut, color, clarity, depth, table, price, x, y, and z. Price is the prediction target, and the remaining variables are predictors. Considering that price is continuous, the task is defined as a supervised regression problem rather than a classification task. Because this dataset is a public attribute-based benchmark rather than a transaction record, the analysis emphasizes comparative modelling choices rather than causal interpretation of market premiums.

The preprocessing procedure follows the baseline logic and records each filtering step to keep the sample construction transparent. The index column is removed. Meanwhile, observations with zero physical dimensions are excluded because a diamond cannot have zero length, width, or depth, and extreme values in depth, table, and geometric dimensions are filtered according to the reproduced baseline rules. Categorical variables are encoded for modelling, while a label-preserving copy of the cleaned dataset is kept for exploratory plots. Table 1 summarizes the sample sizes and column counts at each stage.

Table 1. Dataset preprocessing summary

Stage	Observations	Columns	Notes
Raw Diamonds dataset	53,940	11	Original data loaded from diamonds.csv
After removing index column	53,940	10	Dropped unnamed index column
After removing zero physical dimensions	53,920	10	Removed rows with x, y, or z equal to zero
After baseline outlier filtering	53,907	10	Applied baseline thresholds for depth, table, x, y, and z
Final encoded modelling dataset	53,907	10	Categorical variables encoded for model training
Training split	40,430	9	75% split with random_state=7
Test split	13,477	9	25% split with random_state=7

2.2 Experimental Design

The workflow has three layers. The first layer reproduces the baseline XGBRegressor pipeline to establish a comparable starting point. The second layer performs controlled design comparisons involving raw versus log-transformed target values, baseline features versus baseline features plus volume, and ordinal versus one-hot encoding. The third layer benchmarks broader model families and evaluates a simple ensemble of strong tree-based models. This design separates preprocessing decisions from model-family comparisons and reduces the risk that a performance difference is incorrectly attributed to the wrong modelling choice. Table 2 maps each hypothesis to its corresponding experiment and decision rule.

The engineered feature is volume = x × y × z. This feature is motivated by geometry, but it is treated as a hypothesis rather than an assumed improvement. Because volume is a deterministic within-row transformation of the predictor variables x, y, and z and does not use price, fitted statistics, or information from other observations, it introduces no train-test contamination. The final ensemble averages Random Forest, XGBoost, LightGBM, and CatBoost predictions. More complex stacking or weighted averaging could further optimize validation performance, but would introduce additional tuning choices and a higher risk of validation-specific overfitting. The simple average is therefore kept as a transparent robustness test rather than an optimized ensemble architecture.

Table 2. Mapping between hypotheses, experiments, and decision rules

Hypothesis	Experiment	Decision rule
H1	Raw price vs. log-transformed price	Test RMSE, with MAE and R^2 as support
H2	Baseline features vs. baseline + volume	Test RMSE
H3	Ordinal vs. one-hot encoding	RMSE primary; MAE trade-off noted
H4	Model-family benchmark	CV RMSE and test RMSE
H5	Simple ensemble vs. best single model	RMSE improvement over best single model

2.3 Models, Metrics, and Evaluation Protocol

The main reported benchmark compares Linear Regression, Random Forest, XGBoost, LightGBM, CatBoost, SVR, and the simple ensemble. Linear regression provides an interpretable low-complexity reference, while SVR acts as a non-tree nonlinear comparator. Decision Tree and Random Forest capture non-linear splits and interactions, and boosting methods provide strong tabular benchmarks. XGBoost, LightGBM, and CatBoost are included as representative gradient-boosting methods for structured data [10-12]. CatBoost is used here as a modern boosting method on encoded features, while native categorical handling is not claimed as a separate contribution because the controlled encoding experiments use a shared encoded matrix.

Performance is measured using RMSE as the primary metric because large errors are economically important in price prediction. MAE and R-squared are supporting metrics. For segment-level diagnostics, Relative MAE is reported as MAE divided by the mean actual price of the segment. The dataset is split into training and test sets using a 75% to 25% ratio with random_state=7 for reproducibility. The reproduced baseline reports 10-fold cross-validation RMSE, while the model family and ensemble comparisons use five-fold cross-validation on the training set. Cross-validation RMSE is used for model-level comparison, and controlled design experiments are compared on the same held-out test split.

3 Results

3.1 Controlled Design Comparisons

The controlled comparisons in this section hold the learner fixed within each comparison. The target-transformation and volume experiments use XGBRegressor, following the reproduced baseline configuration, while the encoding comparison uses Random Forest with the same train/test split for both encoding schemes. Thus, each reported difference is attributable to the design choice under test rather than to changing model families within that comparison. The first results evaluate preprocessing and representation decisions under the same train/test split. Table 3 shows that the log-

transformed target improves all reported test metrics, reducing RMSE from 554.73 to 548.24 and MAE from 280.17 to 273.37. This result is consistent with H1 and with the right-skewed price distribution. The volume feature, however, provides no evidence of improvement under the primary metric, as RMSE increases slightly from 554.73 to 556.01. This result is consistent with H2, although the effect size is small. Moreover, the encoding comparison is mixed. Ordinal encoding gives better RMSE and R-squared, while one-hot encoding gives a marginally lower MAE. Because RMSE is the primary criterion, ordinal encoding is preferred with this qualification.

Table 3. Controlled design comparisons on the held-out test set

Design choice	Candidate	Test R^2	Test MAE	Test RMSE
Target transformation	Raw price	0.9804	280.17	554.73
Target transformation	Log-transformed price	0.9808	273.37	548.24
Feature engineering	Baseline features	0.9804	280.17	554.73
Feature engineering	Baseline + volume	0.9803	279.49	556.01
Encoding	Ordinal encoding	0.9809	271.07	547.96
Encoding	One-hot encoding	0.9807	270.90	550.11

3.2 Model Benchmark Comparison

The model-level benchmark confirms that model family matters strongly in this experimental setting. Linear regression obtains a test RMSE of 1319.54 and R-squared of 0.8890, indicating that an approximately linear function cannot capture the full pricing structure. SVR performs worse under the encoded default configuration used here, with test RMSE of 2774.75 and R-squared of 0.5092; this result should be interpreted as a baseline comparator rather than as a claim that all tuned SVR variants would fail. In contrast, tree-based and boosting models remain strong. CatBoost is the best individual extended model, with test RMSE of 527.28 and R-squared of 0.9823. As shown in Table 4, the simple ensemble gives the best overall result, reducing test RMSE to 521.16, improving MAE to 263.11, and increasing R-squared to 0.9827. This supports H4 and H5 because the final improvement does not come from a more complex theoretical model, but from selecting appropriate tabular learners and averaging complementary predictions.

Table 4. Model benchmark comparison

Model	CV RMSE	Test R^2	Test MAE	Test RMSE
Baseline XGBRegressor	547.65	0.9804	280.17	554.73
CatBoost	526.91	0.9823	273.73	527.28
LightGBM	545.44	0.9815	285.94	538.96
XGBRegressor	552.31	0.9804	280.17	554.73
Random Forest	553.91	0.9809	271.08	547.90
Linear Regression	1348.42	0.8890	849.35	1319.54
SVR	2961.80	0.5092	1360.72	2774.75
Simple ensemble	521.76	0.9827	263.11	521.16

3.3 Error Analysis by Price Segment

Aggregate performance is high, but it does not show whether errors are evenly distributed across the target range. The final diagnostic model is therefore evaluated across low-, mid-, and high-price segments, defined as equal-frequency tertiles based on actual test-set prices. These segments are diagnostic groups rather than externally validated market tiers. Table 5 reports segment-level errors and Relative MAE. Fig. 1 summarizes the final diagnostic evidence. Panel a shows absolute-error distributions across price segments, panel b compares actual and predicted prices, and panel c shows residuals versus predicted prices. MAE rises from 56.39 in the low-price segment to 165.62 in the mid-price segment and 567.44 in the high-price segment, while Relative MAE remains close across segments.

This residual pattern provides a useful diagnostic signal, but it should not be overinterpreted. Because the high-price mean residual is much smaller than the segment MAE, it indicates only a directional pattern rather than strong evidence of systematic bias. Because the dataset does not include brand, certificate issuer, retailer, channel, or temporal information, the observed underestimation cannot be attributed to any specific market mechanism. A plausible explanation is feature-space sparsity, where rare combinations of carat, clarity, color, and cut occur less frequently and make upper-range prices harder to estimate from the available attributes alone. However, this explanation is not directly testable with the present dataset. Future studies using enriched transaction-level data could examine whether additional market, certification, or channel variables reduce this residual pattern. The present evidence is therefore best understood as a diagnostic signal rather than proof of a specific premium effect.

Table 5. Segment-level error analysis for the final diagnostic model

Price segment	n	Mean actual price	MAE	RMSE	Mean residual	Relative MAE
Low-price	4,496	773.23	56.39	74.18	8.64	7.29%
Mid-price	4,489	2,518.24	165.62	250.77	29.20	6.58%
High-price	4,492	8,420.08	567.44	864.01	-16.54	6.74%

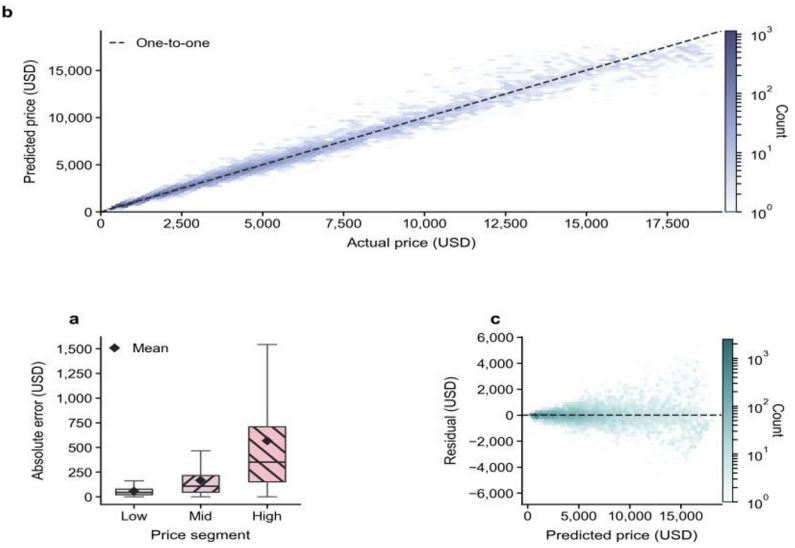


Fig. 1. Diagnostic evidence for the final simple ensemble. (Picture credit: Original)

3.4 Robustness Check across Repeated Splits

To test whether the main conclusion depends on a single data partition, the strongest model families are reevaluated over five train/test splits using random states 7, 23, 73, 101, and 2026. This repeated-split experiment is a robustness check rather than a replacement for the main random_state=7 benchmark. Table 6 shows that the simple ensemble remains the strongest method by mean RMSE, with 524.12 +/- 3.90, while CatBoost remains the strongest individual model with 529.09 +/- 3.06. The reproduced XGBRegressor baseline has a higher mean RMSE of 557.01 and the largest RMSE standard deviation among the compared models. These results indicate that the ensemble advantage is robust to the tested data partitions, although they do not constitute formal statistical evidence of superiority. Table 7 summarizes segment-level residual stability across the same five splits. The high-price segment has a negative mean residual in every split, ranging from -28.79 to -16.54 with an average of -24.18. By contrast, low- and mid-price residuals remain positive, ranging from 8.64 to 11.07 and from 29.20 to 44.00, respectively. This strengthens the interpretation that the high-price underestimation in Table 5 is not merely a single-partition artifact, while Fig. 1 further shows wider residual dispersion at higher predicted prices.

Table 6. Robustness check across five repeated train/test splits

Model	R^2 mean	R^2 std	MAE mean	MAE std	RMSE mean	RMSE std
Simple ensemble	0.9826	0.0004	264.03	1.45	524.12	3.90
CatBoost	0.9823	0.0003	273.73	1.14	529.09	3.06

LightGBM	0.9813	0.0004	286.39	0.85	543.27	3.69
Random Forest	0.9807	0.0004	272.43	1.41	553.12	3.71
XGB baseline	0.9804	0.0008	281.37	2.21	557.01	8.22

Table 7. Segment-level residual stability across repeated splits

Price segment	Count mean	Mean actual price	MAE mean	RMSE mean	Mean residual	Relative MAE (%)
Low-price	4,495.6	780.47	57.23	75.41	9.92 (8.64 to 11.07)	7.33 +/- 0.07
Mid-price	4,489.2	2,550.21	170.00	256.94	35.96 (29.20 to 44.00)	6.67 +/- 0.09
High-price	4,492.2	8,436.01	564.93	867.43	-24.18 (-28.79 to -16.54)	6.70 +/- 0.05

4 Discussion

4.1 Challenges

The main challenge is that strong aggregate accuracy does not remove uncertainty from attribute-only diamond valuation. Tree-based and boosting models are the strongest candidates in this setting, with CatBoost as the best individual extended model and the simple ensemble as the lowest-RMSE method across repeated partitions. However, the benchmark is not fully optimized and therefore supports a practical ranking under the present protocol rather than a universal claim about all possible hyperparameter settings.

The error analysis identifies a bounded residual failure mode. Attribute-only models achieve strong aggregate performance and comparable Relative MAE across price segments, but high-price diamonds carry larger dollar-denominated errors and are mildly underestimated across partitions. Together with the residual-versus-predicted diagnostic in Fig. 1, this pattern suggests that absolute prediction errors increase at higher price levels while proportional accuracy remains broadly stable. Since brand, certificate issuer, retailer, sales channel, time, and local demand are absent from the data, the residual pattern cannot be attributed to a specific market mechanism.

4.2 Future Directions

Future work should first use richer transaction-level data that include certificate issuer, brand or retailer information, sales channel, date, region, fluorescence, polish, symmetry, and market conditions. These variables would allow the model to separate

physical quality from market premiums and would make high-price residual analysis more interpretable.

A second direction is to extend the experimental design beyond local ablations. More systematic hyperparameter optimization, nested cross-validation, uncertainty intervals, calibration analysis, and explainability methods such as permutation importance or SHAP values would strengthen both predictive performance and interpretability. The controlled comparisons in this study show that plausible modelling changes require empirical validation, because log transformation improves the main split, the volume feature does not improve RMSE, and encoding choices produce a metric trade-off.

5 Conclusion

This paper presents a controlled empirical study of machine learning choices for diamond price prediction. Starting from a public baseline implementation, the workflow is reconstructed as a reproducible supervised regression pipeline with explicit preprocessing, feature representation, model benchmarking, ensembling, and error analysis. The study evaluates not only which model performs best, but also which design decisions materially affect prediction accuracy.

The results show that log-transforming price improves performance in the main split, whereas adding a volume feature does not improve RMSE. Ordinal encoding is preferred under the primary RMSE criterion, although one-hot encoding has a marginal MAE advantage. At the model level, CatBoost is the best individual extended model, while a simple ensemble of Random Forest, XGBoost, LightGBM, and CatBoost achieves the strongest main-split performance, with test RMSE of 521.16 and R^2 of 0.9827. Repeated-split evaluation further shows that the ensemble remains the lowest-RMSE method across five partitions, with RMSE 524.12 ± 3.90. Supplementary residual diagnostics indicate that high-price mean residuals remain negative across the tested partitions. Overall, attribute-only valuation is accurate on average and proportionally stable, but remains incomplete when dollar-denominated uncertainty in the upper price range is considered. Future work should incorporate richer market, certification, and transaction-level variables beyond the classic dataset.

References

1. Kigo, S.N., Omondi, E.O., Omolo, B.O.: Assessing predictive performance of supervised machine learning algorithms for a diamond pricing model. Scientific Reports 13, 17315 (2023)
2. Basha, M.S.A., Oveis, P.M.: Predictive modeling and benchmarking for diamond price estimation, integrating classification, regression, hyperparameter tuning and execution time analysis. International Journal of System Assurance Engineering and Management 15(11), 5279-5313 (2024)
3. Alsuraihi, W., Al-Hazmi, E., Bawazeer, K., AlGhamdi, H.: Machine learning algorithms for diamond price prediction. In: Proceedings of the 2020 2nd International Conference on Image, Video and Signal Processing, pp. 150-154. ACM, New York (2020)

4. Shwartz-Ziv, R., Armon, A.: Tabular data, deep learning is not all you need. Information Fusion 81, 84-90 (2022)
5. Grinsztajn, L., Oyallon, E., Varoquaux, G.: Why tree-based models still outperform deep learning on typical tabular data. In: Advances in Neural Information Processing Systems 35, pp. 507-520. Curran Associates, Red Hook (2022)
6. Bentejac, C., Csorgo, A., Martinez-Munoz, G.: A comparative analysis of gradient boosting algorithms. Artificial Intelligence Review 54(3), 1937-1967 (2021)
7. Rico-Juan, J.R., Taltavull de La Paz, P.: Machine learning with explainability or spatial hedonics tools, an analysis of the asking prices in the housing market in Alicante, Spain. Expert Systems with Applications 171, 114590 (2021)
8. Xu, X., Zhang, Y.: House price forecasting with neural networks. Intelligent Systems with Applications 12, 200052 (2021)
9. Kapoor, K.: Diamond Price Prediction. www.kaggle.com/code/karnikakapoor/diamond-price-prediction, last accessed 2026/04/29
10. Chen, T., Guestrin, C.: XGBoost, a scalable tree boosting system. In: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 785-794. ACM, New York (2016)
11. Ke, G., Meng, Q., Finley, T., Wang, T., Chen, W., Ma, W., Ye, Q., Liu, T.Y.: LightGBM, a highly efficient gradient boosting decision tree. In: Advances in Neural Information Processing Systems 30, pp. 3146-3154. Curran Associates, Red Hook (2017)
12. Prokhorenkova, L., Gusev, G., Vorobev, A., Dorogush, A.V., Gulin, A.: CatBoost, unbiased boosting with categorical features. In: Advances in Neural Information Processing Systems 31, pp. 6639-6649. Curran Associates, Red Hook (2018)

Open Access This chapter is licensed under the terms of the Creative Commons Attribution-NonCommercial 4.0 International License (http://creativecommons.org/licenses/by-nc/4.0/), which permits any noncommercial use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license and indicate if changes were made.

The images or other third party material in this chapter are included in the chapter's Creative Commons license, unless indicated otherwise in a credit line to the material. If material is not included in the chapter's Creative Commons license and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder.

