Comparative Analysis of Supervised Learning Models for Diamond Price Prediction

Chiao-Ting Tseng
Department of Big Data Analytics
National Chung Hsing University
Taichung, Taiwan
u3814520@smail.nchu.edu.tw

***Abstract*—This study aims to reproduce and compare supervised machine learning models for diamond price prediction, based on three peer-reviewed studies. Models including Linear Regression, Lasso Regression, Random Forest, and Extra Trees are applied to a dataset containing over 50,000 diamond records. After preprocessing and feature selection, models are evaluated based on metrics such as RMSE and R^2. Results show that ensemble models like Extra Trees and Random Forest consistently achieved higher accuracy, while minimalist models such as k-NN also performed well under feature-reduced conditions. These findings validate the effectiveness and reproducibility of the selected methodologies.**

***Index Terms*—Diamond price prediction, supervised learning, regression, Random Forest, Extra Trees, Lasso**

I. Introduction

The price of diamonds is influenced by multiple attributes. While carat weight is commonly regarded as one of the primary determining factors, whether its relationship with price is consistent and linear remains subject to empirical validation. Traditionally, diamond pricing in the market is based on the "4Cs" standard—Cut, Color, Clarity, and Carat—which categorizes and estimates value. However, this approach has limitations in capturing the complex interactions among multiple variables, making it difficult to reflect the increasingly sophisticated mechanisms behind market valuation.

With the growing maturity of machine learning techniques, recent studies have increasingly adopted supervised learning models to enhance the accuracy and interpretability of diamond price prediction. This paper references three studies that apply supervised learning for diamond price forecasting: (1) Sharma et al. (2021), "Comparative Analysis of Supervised Models for Diamond Price Prediction," presented at the IEEE Confluence conference; (2) Kapil Amadavadi et al. (2024), "Diamond Price Prediction using Machine Learning Techniques," presented at IEEE ICOSEC; and (3) Fitriani et al. (2022), "LASSO and k-NN Algorithm Analysis Based on Feature Selection for Diamond Price Prediction," presented at IEEE ISMODE.

Although all three adopt supervised learning approaches, their motivations and methodological strategies differ. Sharma et al. focus on comparing the performance of various regression models, aiming to identify the most accurate and stable predictor. Kapil et al. argue that the traditional 4Cs are insufficient to fully capture price variation and therefore incorporate additional features and diverse machine learning algorithms. Fitriani et al., on the other hand, emphasize the importance of feature reduction, using statistical methods to select core variables and comparing the predictive performance of LASSO and k-NN models under a simplified input setting.

II. Data Description and Preprocessing

The dataset used in this study is the publicly available "Diamonds" dataset from the Kaggle platform, comprising a total of 53,940 diamond records. As illustrated by Amadavadi et al. (2024), the dataset documents the variation in diamond prices under different attribute conditions, making it highly valuable for analytical purposes.

The main features include: Cut, Color, Clarity, Carat, Length (X), Width (Y), Height (Z), Total Depth, Table size, and an extended feature derived for Market Demand classification. Among them, several features such as Cut, Color, and Clarity are categorical variables, while the remaining ones are continuous numerical data, making them suitable inputs for regression models.

Fig. 1 illustrates the geometric structure of a diamond, highlighting key geometric parameters that influence diamond quality and pricing. These include table size, total depth, crown height, pavilion depth, crown and pavilion angles, girdle thickness, and culet. Variables such as Depth and Table in our dataset correspond to the geometrical dimensions shown in the diagram, providing a basis for understanding the relationship between shape and price.

To facilitate quick understanding of the dataset features and their respective value ranges, Fig. 2 summarizes the names and ranges of key variables. Carat weight ranges from 0.2 to 5.01 carats, while prices range from $326 to 18{,}823. Other attributes include dimensional and classification indicators relevant to diamond size, shape, and market demand.

III. Related Work

A. Sharma et al. (2021)

1) Data Preprocessing: This study used the publicly available diamond dataset from Kaggle and performed basic preprocessing steps. Categorical features such as `cut`, `color`, and `clarity` were converted to numeric form using label encoding, enabling compatibility with regression algorithms. Continuous features were standardized to ensure consistent

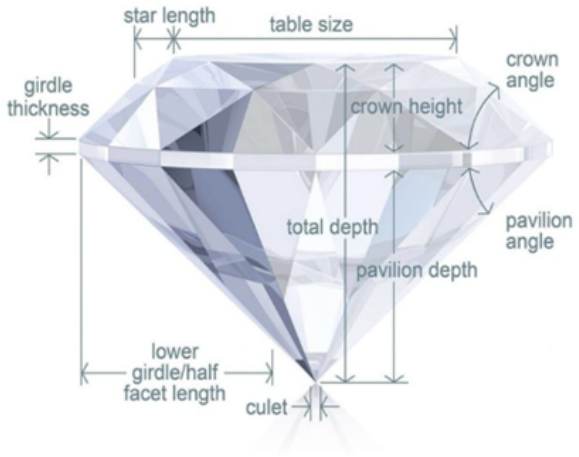

Fig. 1. Illustration of diamond geometric structure.

Features	***Range***
Cut	(Fair, Good, Very Good, Premium, Ideal)
Color	J (worst) - D (best)
Clarity	(I1 (worst), SI2, SI1, VS2, VS1, VVS2, VVS1, IF (best))
Carat	0.2 - 5.01 ct
Depth	(0 - 31.8) mm
Table	(43 - 95) mm
Price	($326--$18,823)
X(length)	(0 - 10.74) mm
Y(width)	(0 - 58.9) mm
Z(depth)	(0 - 31.8) mm
Market demand	(high, low, Medium)

Fig. 2. Feature summary table of the dataset.

input scale across variables. Unlike some studies that filter out anomalies, this research preserved all data—including potential outliers and missing values—for model training.

A correlation matrix was utilized to preliminarily assess the linear relationship between features and price. Fig. 3 illustrates the scatterplot of `carat` versus `price`, revealing a clear positive relationship. While a heavier diamond generally corresponds to a higher price, the chart indicates increasing price volatility in the high-carat region. This suggests that exceptionally large diamonds may exhibit unpredictable pricing due to factors such as rarity or quality variance.

To further evaluate linear relationships among numerical variables, a feature correlation heatmap was constructed, as

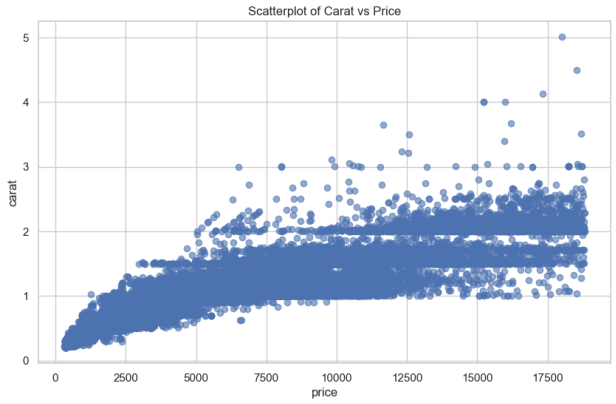

Fig. 3. Scatterplot of Carat vs Price

shown in Fig. 4. The heatmap highlights a strong correlation (0.92) between `carat` and `price`, the highest among all features. The dimensions `x`, `y`, and `z` (length, width, and height) also showed strong positive correlations with price (0.86–0.88), indicating that diamond size plays a critical role in pricing. Conversely, `depth` and `table` exhibited weaker correlations, suggesting lower predictive influence.

Fig. 4. Correlation Heatmap of Features

In replicating this paper's method, I implemented the same preprocessing pipeline in Python. This included label encoding for categorical features, data standardization, and visual correlation analysis for feature evaluation. To maintain consistency with the original study, I refrained from performing further feature selection and retained all variables in the model training phase to enable a fair comparison of different regression models.

2) Models: In this study, eight supervised regression models were applied for diamond price prediction: Linear Regression, Lasso Regression, Ridge Regression, Decision Tree Regressor, Random Forest Regressor, ElasticNet, AdaBoost

Regressor, and Gradient Boosting Regressor. The dataset was split into 80% training and 20% testing using stratified sampling based on `carat` to preserve the distribution of diamond weights. Performance was evaluated using two key metrics: Root Mean Square Error (RMSE) and Cross-Validation (CV) RMSE scores.

3) Results: Results showed that the Random Forest Regressor achieved the best performance, with the highest accuracy (approximately 97.96%) and the lowest prediction error, demonstrating its strong capacity to model continuous target variables. This is shown in Fig. 5, where Random Forest is positioned at the top in terms of accuracy among all models tested. Decision Tree and Gradient Boosting Regressors also performed well, ranking second and third respectively, indicating their effectiveness in handling structured numerical data.

	Model	RMSE	Cross-Validation RMSE	Accuracy
0	Random Forest	569.237693	582.791666	97.961658
1	Decision Tree	630.164799	665.212760	97.501968
2	Gradient Boosting	655.694773	684.031070	97.295462
3	AdaBoost	1260.560387	1339.595915	90.004213
4	Ridge Regression	1351.262011	1355.212906	88.513999
5	Linear Regression	1351.263480	1355.234769	88.513974
6	Lasso Regression	1351.268941	1355.152417	88.513881
7	ElasticNet	1444.055914	1457.803767	86.882299

Fig. 5. Accuracy comparison of eight supervised regression models.

In my implementation, I followed the same modeling pipeline described in the paper and trained the same eight models using Python. I used RMSE as the primary error metric for consistency and maintained the original feature set without additional selection. As shown in Fig. 6, my experimental results were consistent with those reported by Sharma et al. (2021), with Random Forest again outperforming all other models across RMSE, Cross-Validation RMSE, and accuracy. Gradient Boosting and Decision Tree also demonstrated strong performance, ranking closely behind Random Forest. These findings reinforce the stability and generalizability of Sharma et al.'s model ranking and highlight the robust predictive power of ensemble-based models, particularly in datasets involving multivariate interactions and non-linear relationships.

B. Amadavadi et al. (2024)

1) Data Preprocessing: This study also utilized the same publicly available diamond dataset from Kaggle as used in the other studies. The dataset includes attributes such as carat weight, cut, color, clarity, dimensions (x, y, z), depth, table, price, and an additional feature, market demand. The authors performed a series of meticulous preprocessing steps, including data cleaning, outlier removal, label encoding, and correlation analysis.

Specifically, they excluded records with missing or erroneous values, then applied statistical methods such as Z-score and IQR to detect and remove outliers, aiming to minimize

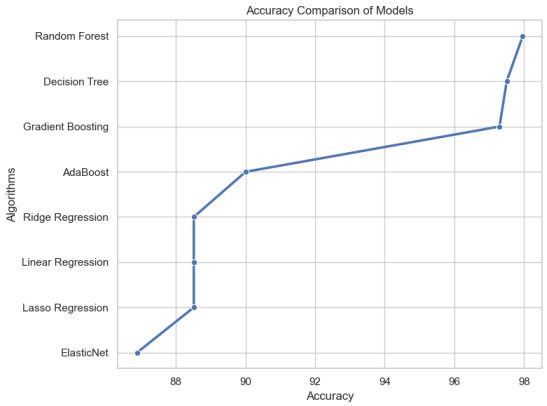

Fig. 6. Detailed performance metrics including RMSE, CV RMSE, and accuracy.

noise that might interfere with model training. Categorical features like `cut`, `color`, and `clarity` were transformed into numerical format via label encoding to make them compatible with machine learning models.

To understand relationships among features, the authors computed a correlation matrix, visualized in Fig. 7, which showed strong positive correlations between `carat`, `x`, `y`, `z` and `price`. These variables were prioritized and retained for model training.

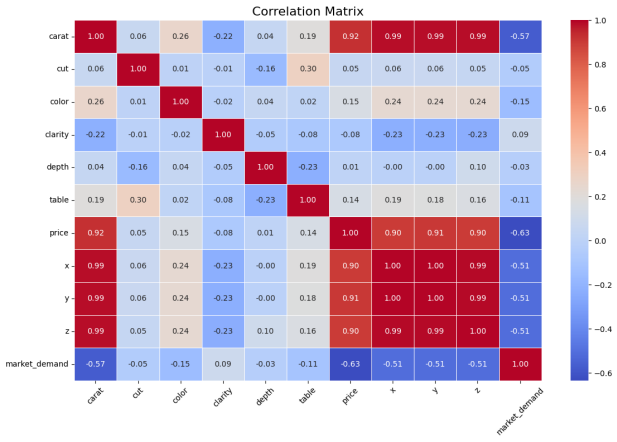

Fig. 7. Correlation matrix of features in Amadavadi et al. (2024).

Compared to Sharma et al. (2021), this study demonstrated greater rigor in data quality management, especially with respect to outlier handling.

In my own implementation, I followed the outlined preprocessing steps. I used Pandas to load the dataset, applied label encoding for categorical variables, and employed Seaborn and Matplotlib to generate scatterplots and heatmaps to verify variable correlations. I also implemented outlier detection and removal using Z-score and IQR methods, consistent with the methodology described in the paper, to reduce the influence of extreme values on model performance. Outliers were identified based on a Z-score threshold of 3 and IQR fences, and

the corresponding rows were removed to ensure data consistency. Furthermore, I standardized all numerical features using `StandardScaler` to eliminate scale-related biases before model training. This comprehensive preprocessing approach enabled me to faithfully replicate the procedures of the original study and contributed to more stable and accurate predictive outcomes.

2) Models: The study compared multiple supervised machine learning models for diamond price prediction. Models tested included: Linear Regression, Decision Tree Regression, Random Forest Regression, K-Nearest Neighbors (KNN), XGBoost Regressor, Extra Trees Regressor, Gradient Boosting Regressor, and Multi-Layer Perceptron (MLP). An 80/20 train-test split was used for all experiments. The authors evaluated performance using three key metrics: R-squared (R^2), Root Mean Square Error (RMSE), and Mean Absolute Error (MAE).

3) Results: The results indicated that the Extra Trees Regressor outperformed all other models, achieving the highest R^2 value of 0.9869 and the lowest prediction error. Ensemble methods such as Random Forest, XGBoost, and Gradient Boosting also delivered strong performance, consistently surpassing traditional single models like Linear Regression and KNN. Fig. 8 illustrates the R^2 values reported for each model.

In my implementation, I reproduced the eight models mentioned in the study and evaluated them using the same metrics. Consistent with the findings of Amadavadi et al. (2024), Extra Trees and Random Forest achieved the best results. Fig. 9 presents the accuracy of each model from my implementation. These outcomes not only confirm the reproducibility and stability of the model rankings proposed in the original paper but also emphasize the advantage of ensemble techniques in modeling complex, high-dimensional data.

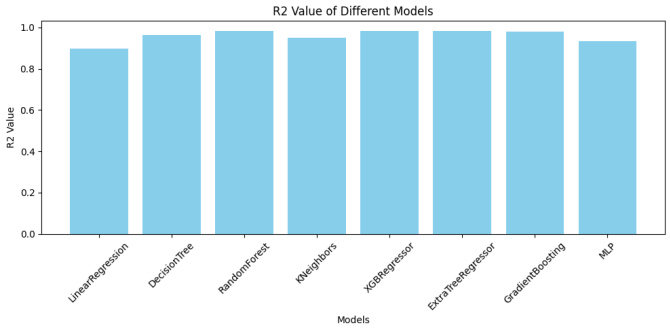

Fig. 8. R^2 values of different models reported by Amadavadi et al. (2024).

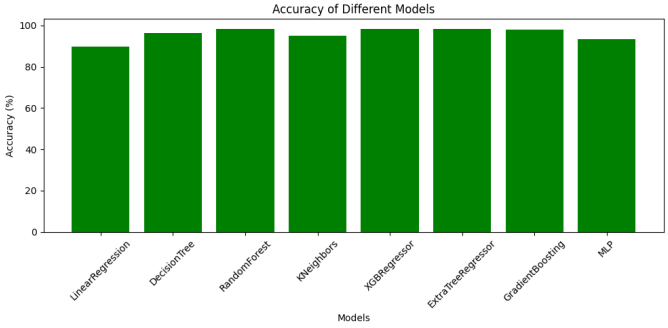

Fig. 9. Accuracy comparison of different models in my implementation.

C. Fitriani et al. (2022)

1) Data Preprocessing: The third study focuses on comparing the predictive performance of two models: LASSO and k-Nearest Neighbors (k-NN). The authors emphasize the importance of data preprocessing and feature selection for improving model accuracy. Like the previous two studies, this work also utilized the publicly available diamond dataset from Kaggle, ensuring consistency across all three studies.

The preprocessing steps involved several stages. First, data cleaning was performed to check for missing or duplicate entries. Since the dataset contained no missing values, the primary focus was on removing duplicates to ensure data integrity. Categorical variables such as `cut`, `color`, and `clarity` were converted into numerical form via label encoding. Subsequently, all numerical features were standardized using `StandardScaler` to eliminate scale-based biases.

Feature selection was a core part of this study. The authors applied the `SelectKBest` function with the `f_regression` scoring method to assess feature relevance. Fig. 10 shows the distribution of the top four features: `carat`, `x`, `y`, and `z`, which were identified as having the highest correlation with price. These four features were retained as model inputs, effectively removing less informative variables such as `depth` and `table`.

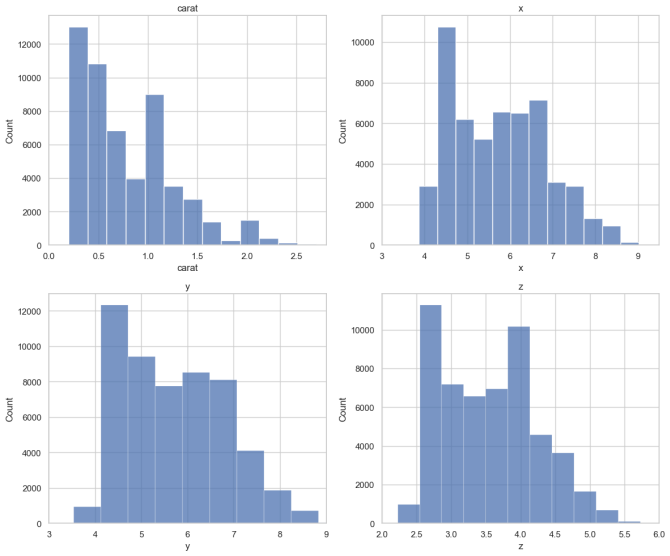

Fig. 10. Top 4 selected features (carat, x, y, z) using SelectKBest.

In my implementation, I followed a similar preprocessing pipeline. Label encoding and standardization were applied to categorical and numerical features respectively. I also used `SelectKBest` to retain the top four features, matching the original study.

2) Models: The study evaluates and compares the performance of LASSO and k-NN models on the selected features. For the k-NN model, the authors used the Elbow Method to determine the optimal number of neighbors k. Fig. 11 illustrates that $k = 11$ yields the best performance.

To further support this result, Table I presents the RMSE and R^2 scores obtained from a Grid Search over k values ranging from 1 to 15. The model achieved its lowest RMSE (995.16) and highest R^2 (0.9350) at $k = 11$, confirming it as the optimal configuration for this task.

TABLE I
GRID SEARCH RESULTS FOR K-NN ($k = 1$ TO 15)

k	RMSE	R^2
1	1287.63	0.8912
2	1131.29	0.9160
3	1078.39	0.9237
4	1047.00	0.9281
5	1029.35	0.9305
6	1022.21	0.9314
7	1010.70	0.9330
8	1005.37	0.9337
9	999.77	0.9344
10	996.62	0.9348
11	995.16	0.9350
12	995.64	0.9350
13	995.41	0.9350
14	996.58	0.9348
15	996.81	0.9348

For the LASSO model, hyperparameter tuning was conducted using GridSearchCV across a predefined range of α values. Table II summarizes the RMSE and R^2 scores for different α values tested.

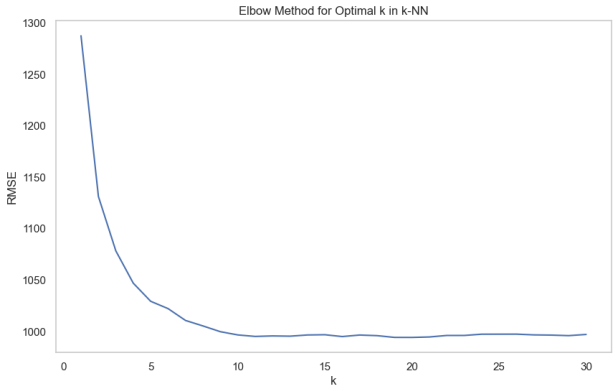

Fig. 11. Elbow Method for optimal k selection in k-NN.

TABLE II
LASSO REGRESSION GRIDSEARCHCV RESULTS

α	RMSE	R^2
5	1313.99	0.8867
10	1317.79	0.8861
20	1330.06	0.8839
30	1347.07	0.8809
40	1349.53	0.8805
45	1350.99	0.8803
50	1352.59	0.8800
55	1354.35	0.8797
100	1376.05	0.8758

3) Results: According to the paper, k-NN with $k = 11$ achieved the best result with $R^2 = 0.9066$ and RMSE = 926.06. The LASSO model performed best when $\alpha = 5$, achieving $R^2 = 0.8801$ and RMSE = 1049.59.

Fig. 12 clearly compares the predictive performance of the two models based on their test RMSE values. The bar chart reveals that the k-NN model with $k = 11$ significantly outperforms the LASSO model with $\alpha = 5$. The RMSE for k-NN is 995.16, while LASSO's RMSE is notably higher at 1313.99. This nearly 300-point difference in error indicates that, under the selected features and tuned parameters, k-NN achieves a more accurate fit for the diamond price prediction task.

Furthermore, the visualization shows that the performance gap is not marginal—k-NN maintains a substantial advantage in minimizing prediction error. This suggests that for the given dataset and simplified input feature space, the distance-based approach of k-NN is better suited than the regularized linear formulation used in LASSO. These findings reinforce the paper's conclusion that k-NN is more effective under the described preprocessing and feature selection strategy.

In my implementation, I also applied `SelectKBest` to isolate the same four input features and used GridSearchCV to tune both models. I found that $k = 11$ was also optimal for k-NN, while $\alpha = 5$ was the best setting for LASSO, closely replicating the original study's results. The outcomes reinforce the reliability of feature selection and parameter tuning strategies proposed in the paper.

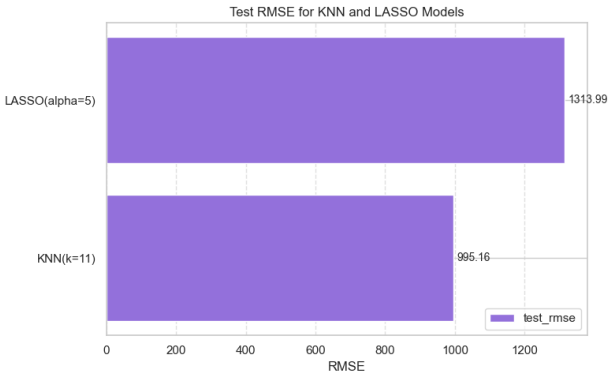

Fig. 12. Final test RMSE comparison for KNN and LASSO models.

IV. Comparative Summary

Although all three studies adopted supervised learning approaches for diamond price prediction, they differ significantly in data preprocessing strategies, model architecture, and feature utilization.

Sharma et al. (2021) conducted a comprehensive comparison of eight regression models without performing any feature selection. All variables were retained in the modeling process, allowing the authors to evaluate how different models respond to the full feature set. Their findings highlighted that ensemble models such as Random Forest and Gradient Boosting Regressor outperformed others in terms of accuracy and stability.

In contrast, Amadavadi et al. (2024) emphasized data quality and variable correlation. They included outlier treatment and correlation heatmaps to evaluate the linear relationships between features and price, establishing a feature importance hierarchy. Although they retained all variables in the final model training, they benchmarked a diverse set of supervised models and found that Extra Trees Regressor achieved the best performance, showcasing its strength in high-dimensional settings.

Fitriani et al. (2022), on the other hand, focused on the impact of feature reduction on model performance. Using the SelectKBest function with the f_regression method, they retained only four features—carat, x, y, and z—for modeling. Under this simplified input space, the k-NN model outperformed LASSO, demonstrating strong predictive capability with minimal features.

In summary, the three studies represent distinct modeling philosophies. Sharma et al. approached the task from a model-centric perspective, comparing diverse algorithms using the full dataset. Amadavadi et al. incorporated feature relevance and data quality considerations to validate ensemble models' robustness in complex settings. Fitriani et al. explored the trade-off between model simplicity and accuracy by aggressively reducing input dimensions. Together, they exemplify three common supervised learning strategies—comprehensive model comparison, correlation-driven refinement, and minimalist feature selection—providing complementary insights into effective diamond price prediction.

V. Conclusion

This study aimed to explore the effectiveness of supervised learning models in predicting diamond prices using the widely adopted Kaggle dataset. Through a comprehensive review and reproduction of three peer-reviewed studies—Sharma et al. (2021), Amadavadi et al. (2024), and Fitriani et al. (2022)—we examined diverse modeling philosophies and evaluated their respective preprocessing strategies, feature selection techniques, and model performance.

Sharma et al. (2021) conducted a broad comparison of eight regression algorithms without applying feature selection. Their findings highlighted the strong predictive performance of ensemble models like Random Forest and Gradient Boosting when using the full set of features. Amadavadi et al. (2024) incorporated more rigorous preprocessing steps, including outlier removal and correlation-based feature prioritization, and demonstrated that Extra Trees Regressor outperformed other models across all metrics. Fitriani et al. (2022) emphasized feature reduction by applying SelectKBest with f_regression, narrowing the input space to the most relevant variables. Under this minimalist setting, the k-NN model delivered better results than LASSO.

In my implementation, I adopted and adapted methodologies from these three studies. For Sharma et al., I replicated the full-feature modeling process and confirmed Random Forest's superior performance. For Amadavadi et al., I implemented outlier detection and ensemble model training, again confirming the effectiveness of Extra Trees. For Fitriani et al., I performed feature selection and hyperparameter tuning for LASSO and k-NN models using a slightly modified feature set, validating the impact of feature simplification on performance.

Overall, the comparative analysis underscores the versatility and adaptability of supervised regression techniques in handling structured pricing problems. It also highlights that model performance depends not only on algorithm choice but also on thoughtful preprocessing and feature engineering. These insights are valuable for future research and practical applications in data-driven pricing strategies.

References

[1] G. Sharma, V. Tripathi, M. Mahajan, and A. K. Srivastava, "Comparative Analysis of Supervised Models for Diamond Price Prediction," in *Proc. 11th Int. Conf. on Cloud Computing, Data Science & Engineering (Confluence)*, 2021, pp. 1019–1022.

[2] K. Amadavadi, R. Rane, and R. Patankar, "Diamond Price Prediction Using Machine Learning Techniques," in *Proc. 5th Int. Conf. on Smart Electronics and Communication (ICOSEC)*, 2024.

[3] S. A. Fitriani, Y. Astuti, and I. R. Wulandari, "LASSO and k-NN Algorithm Analysis Based on Feature Selection for Diamond Price Prediction," in *Proc. Int. Seminar on Machine Learning, Optimization, and Data Science (ISMODE)*, 2022.