Modeling of Diamond Prices: A Data-Driven Analysis

Miguel Ángel Bravo Martínez del Valle

MIS 720: E-Business Infrastructure: Data Science and Big Data

Instructor: Dr. Aaron Elkins

San Diego State University

December 19, 2024

Contents

1 Executive Summary

This report explores predictive modeling of diamond prices using the "Diamonds" dataset, which comprises 53,940 observations and 10 variables, including both numerical and categorical data. The project aims to assess the performance of various machine learning models and the impact of skewness transformation on predictive accuracy.

The dataset was prepared through rigorous preprocessing steps. Skewness in numerical variables was evaluated and corrected where necessary, while highly correlated variables were removed to address multicollinearity. Categorical variables were transformed into dummy variables, and standardization was applied to ensure all features were on a comparable scale.

Six models were employed: Linear Regression, K-Nearest Neighbors (KNN), Decision Tree, Random Forest, Gradient Boosting, and Neural Network. These models were evaluated using 10-fold cross-validation and Root Mean Squared Error (RMSE) as the primary performance metric. The analysis was conducted on both the original dataset and a skewness-transformed version, enabling direct comparisons of model performance under different data distributions.

The results revealed varying impacts of skewness transformation. For most models, including Linear Regression, KNN, Decision Tree, and Random Forest, the transformation had minimal effect on RMSE. For instance, Decision Tree achieved RMSEs of 693 and 683 for the non-transformed and transformed datasets, respectively. However, the Neural Network demonstrated a significant improvement, with RMSE decreasing from 1584 on the non-transformed dataset to 547.83 on the transformed dataset. This highlights the importance of preprocessing for models that rely heavily on well-distributed input data.

Table 1: Comparison of Model RMSE Results for Both Datasets

Model	**Non-Transformed RMSE**	**Transformed RMSE**
Linear Regression	1156	1182
K-Nearest Neighbors (KNN)	914	913
Decision Tree	693	683
Random Forest	640	644
Gradient Boosting	1867	1872
Neural Network	1584	547.83

The best overall performance was achieved by the Neural Network on the skewness-transformed dataset, making it the recommended approach for scenarios where computational resources and expertise are available. For simpler implementations, Random Forest on the non-transformed dataset provided a strong balance between performance (RMSE: 640) and interpretability.

The findings underscore the importance of tailoring preprocessing techniques to the specific characteristics of the model and dataset. Skewness transformation is particularly valuable for complex models like Neural Networks but may be unnecessary for tree-based models. This nuanced approach to data preparation and model selection ensures robust, actionable insights, offering valuable guidance for practitioners in the diamond industry and beyond.

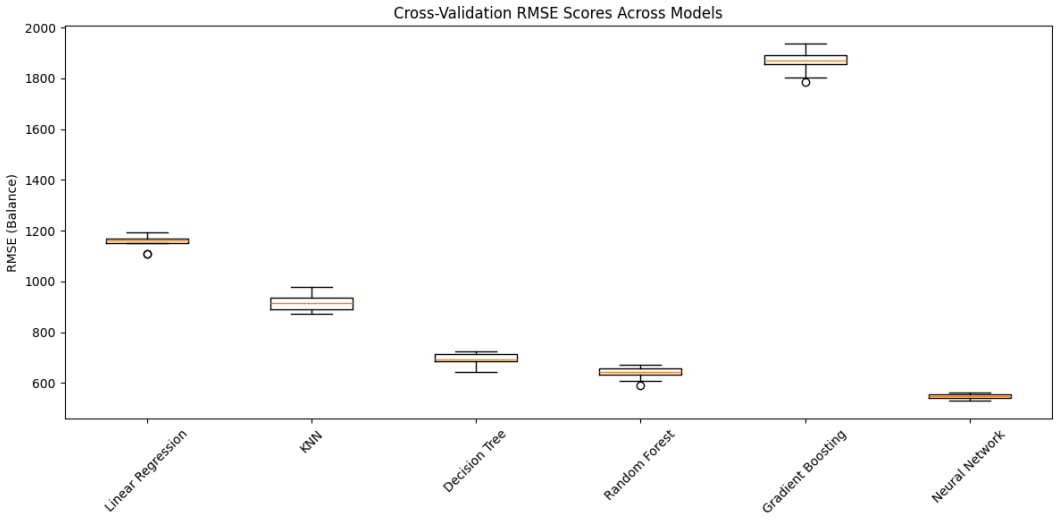

Figure 1: Comparison of Model Performances Across Datasets.

2 Discovery and Data Preparation

2.1 Data Discovery

The dataset used for this project originates from Kaggle and is titled *"Diamonds"* [**?**]. The dataset consists of 53,940 observations (rows) and 10 initial variables (columns). These variables include both numerical and categorical data, as described in the table below:

Table 2: Dataset Variables and Ranges

Variable	**Description**	**Range / Categories**
price	Price in US dollars	$326 – $18,823
carat	Weight of the diamond	0.2 – 5.01
cut	Quality of the cut	Fair, Good, Very Good, Premium, Ideal
color	Diamond color	J (worst) to D (best)
clarity	Clarity level	I1, SI2, SI1, VS2, VS1, VVS2, VVS1, IF
x	Length (mm)	0 – 10.74
y	Width (mm)	0 – 58.9
z	Depth (mm)	0 – 31.8
depth	Total depth percentage	43 – 79
table	Width of top relative to widest point	43 – 95

Although the dataset contains 10 initial variables, the presence of categorical variables (such as *cut*, *color*, and *clarity*) allows for the expansion of attributes through techniques like one-hot encoding. This ensures the dataset fulfills the project's requirement of having more than 10 attributes.

The selected Diamonds dataset provides a robust foundation for analyzing key factors influencing diamond prices, such as weight, cut, color, and clarity. This data is highly relevant for understanding market dynamics, as it reflects real-world pricing variations based on measurable attributes. By leveraging this dataset, we can develop predictive models to assist jewelers, buyers, and sellers in pricing strategies, inventory management, and value assessment, thereby driving data-driven decision-making in the diamond industry.

2.2 Data Preparation

Removal of Irrelevant Variables. The column `Unnamed: 0` was removed from the dataset as it only served as an index and did not provide any relevant information for the analysis.

Identification of Numerical and Categorical Columns. The dataset variables were classified as follows:

- **Numerical**: `carat`, `depth`, `table`, `price`, `x`, `y`, `z`.
- **Categorical**: `cut`, `color`, `clarity`.

The categorical variables were later converted into *dummy variables* for analysis.

Skewness Evaluation. The skewness of the numerical variables was evaluated to identify asymmetric distributions. As an example, the `price` variable showed significant positive skewness ($skewness = 1.618$) in its original distribution. After transformation, skewness was successfully reduced to 0.026 (see Figure 2).

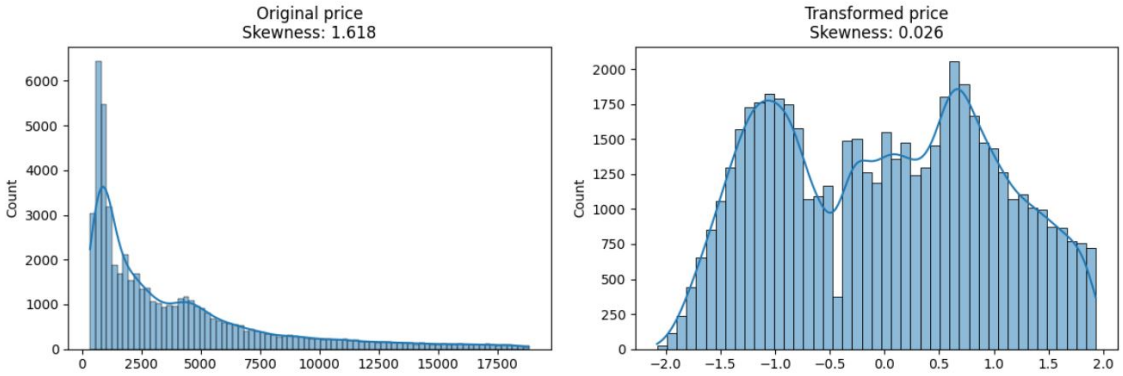

Figure 2: Skewness of `price` before and after transformation.

A similar process was applied to all numerical variables, resulting in significant improvements in skewness. However, these transformations were **not applied to the final dataset**, as their impact will be evaluated in later stages. Specifically, when studying the predictive models, we will assess whether applying transformations to reduce skewness improves model performance.

Correlation Analysis. The correlation matrix initially revealed strong correlations between certain variables:

- `carat` and `x` (0.975), `y` (0.952), `z` (0.953)
- `x` and `y` (0.975), `z` (0.971)

These high correlations indicated potential issues of *multicollinearity*, which could negatively impact the performance and interpretability of the predictive model. To address this, the variables `x`, `y`, and `z` were removed from the dataset.

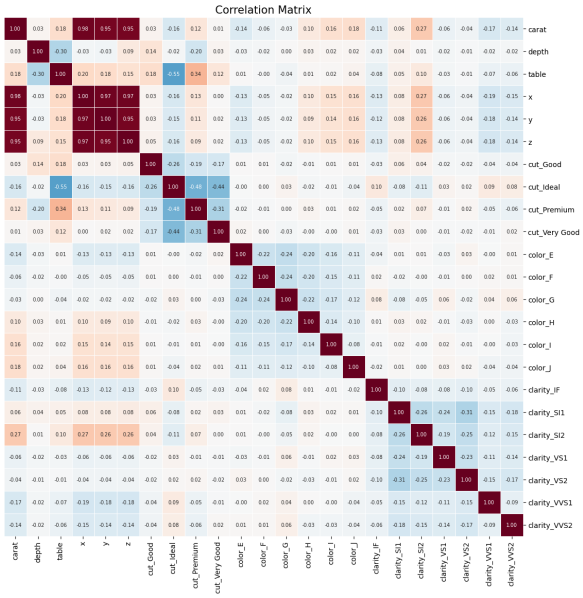

Figure 3: Correlation Matrix before

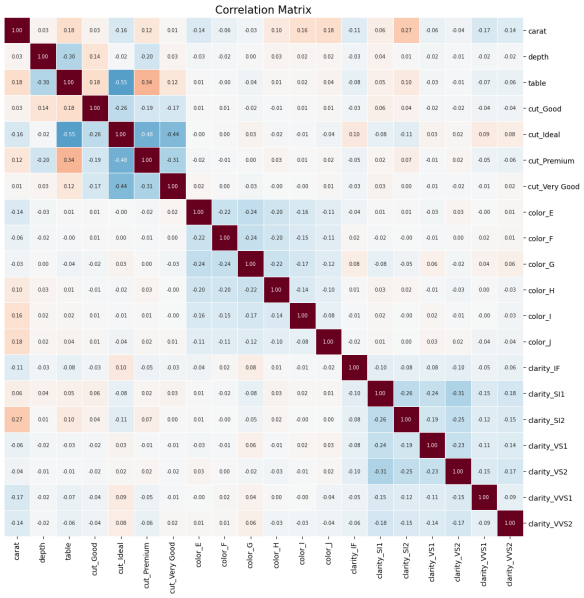

Figure 4: Correlation Matrix after

Conversion of Categorical Variables. The categorical variables (`cut`, `color`, and `clarity`) were converted into *dummy variables*, resulting in a total of 22 new boolean columns.

Near Zero Variance Analysis. A near-zero variance analysis was conducted to identify predictors with minimal variability, which typically contribute little to the model due to their lack of dispersion. Many categorical dummy variables were flagged, as well as some numerical predictors with repeated values.

While the results suggested that several predictors, such as `cut_Good`, `clarity_VVS1`, and `carat`, exhibited low variability, no variables were removed at this stage. A deeper evaluation will be conducted during the modeling process to assess their actual contribution to the prediction performance.

Standardization of Predictors. Finally, the numerical variables were standardized using **Z-score normalization** to ensure all predictors are on the same scale. This process

transforms the values into standard deviations from the mean, allowing the model to handle variables with different ranges more effectively. The resulting dataset is now standardized and ready for analysis.

3 Model Planning and Building

This section outlines the technical and analytical strategies employed for modeling the diamond price prediction. Six predictive models were selected to evaluate their performance: Linear Regression, K-Nearest Neighbors (KNN), Decision Tree, Random Forest, Gradient Boosting, and Neural Network. Each model was evaluated using 10-fold cross-validation to ensure robust performance estimates.

3.1 Evaluation Strategy

Root Mean Squared Error (RMSE) was used as the primary performance metric. RMSE measures the average magnitude of prediction errors, penalizing larger deviations more heavily due to its quadratic nature. This makes it particularly useful for assessing models in regression tasks, where understanding the scale of errors in the same units as the target variable (price in dollars) is crucial. The metric allows for a direct interpretation of how well the model's predictions align with actual values.

Additionally, 10-fold cross-validation was employed to validate model performance. By splitting the data into 10 subsets and rotating through them for training and testing, this approach minimizes bias and variance in the evaluation process, ensuring a robust assessment of each model's ability to generalize to unseen data.

3.2 Model Descriptions

Linear Regression

Linear Regression serves as the baseline model for this analysis. It assumes a linear relationship between predictors and the target variable. Regularization techniques, such as Ridge and Lasso, were considered to prevent overfitting and improve generalizability.

K-Nearest Neighbors (KNN)

KNN is a non-parametric model that predicts based on the similarity of data points. Hyperparameters such as the number of neighbors (*n_neighbors*), distance metrics (*metric*), and weighting schemes (*weights*) were optimized using GridSearchCV to identify the best configuration.

Decision Tree

Decision Trees are interpretable models that split the data based on features to minimize prediction error. The model was tuned by adjusting parameters such as maximum depth (*max_depth*), minimum samples per split (*min_samples_split*), and minimum samples per leaf (*min_samples_leaf*).

Random Forest

Random Forests, an ensemble method based on Decision Trees, combine multiple trees to improve prediction accuracy and reduce overfitting. Hyperparameters such as the number of trees (*n_estimators*) and tree-specific parameters (*max_depth*, *min_samples_split*, *min_samples_leaf*) were optimized.

Gradient Boosting

Gradient Boosting is another ensemble method that builds trees sequentially, optimizing residual errors from previous iterations. Key parameters such as the learning rate (*learn-*

ing_rate), number of estimators (*n_estimators*), and tree depth (*max_depth*) were tuned.

Neural Network

A feed-forward Neural Network with multiple hidden layers was employed for this analysis. Parameters such as the architecture (*hidden_layer_sizes*) and activation functions (*activation*) were tuned using GridSearchCV. While computationally intensive, Neural Networks offer flexibility in capturing complex relationships.

4 Results and Performance

4.1 Model Results for Non-Transformed Data

Table 3 presents the RMSE results for all models applied to the original dataset (without skewness transformation).

Table 3: Model RMSE Results for Non-Transformed Data

Model	**RMSE**	**Std Dev**
Linear Regression	1156	27.47
K-Nearest Neighbors (KNN)	914	32.24
Decision Tree	693	23.70
Random Forest	640	23.67
Gradient Boosting	1867	42.96
Neural Network	1584	1076.04

4.2 Model Results for Skewness-Transformed Data

Table 4 shows the RMSE results for the skewness-transformed dataset after reversing the standardization for comparability.

Table 4: Model RMSE Results for Skewness-Transformed Data

Model	**RMSE**
Linear Regression	1182
K-Nearest Neighbors (KNN)	913
Decision Tree	683
Random Forest	644
Gradient Boosting	1872
Neural Network	547.83

4.3 Performance of Individual Models

Each model's performance is analyzed by comparing its results for both the non-transformed and skewness-transformed datasets. The figures present the results side by side for visual comparison.

Linear Regression Linear Regression showed similar performance across both datasets, with an RMSE of 1156 for the non-transformed dataset and 1182 for the skewness-transformed dataset. The model demonstrated limited sensitivity to the skewness transformation, as expected for a linear model.

K-Nearest Neighbors (KNN) KNN achieved an RMSE of 914 for the non-transformed dataset and 913 for the skewness-transformed dataset. The optimal parameters for the non-transformed dataset were `metric = euclidean`, `n_neighbors = 5`, and `weights = distance`. For the skewness-transformed dataset, the optimal parameters changed slightly to `metric = manhattan`, `n_neighbors = 5`, and `weights = distance`.

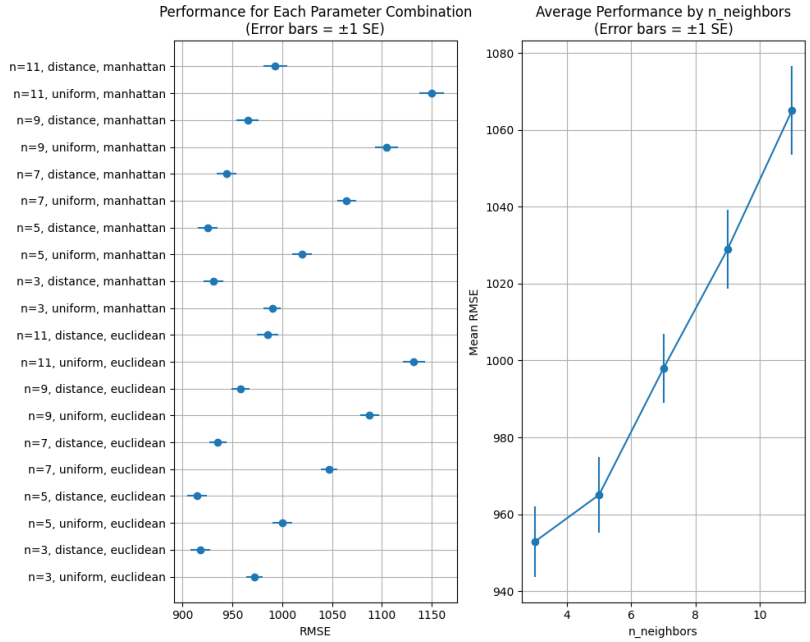

Figure 5: KNN on Non-Transformed Data.

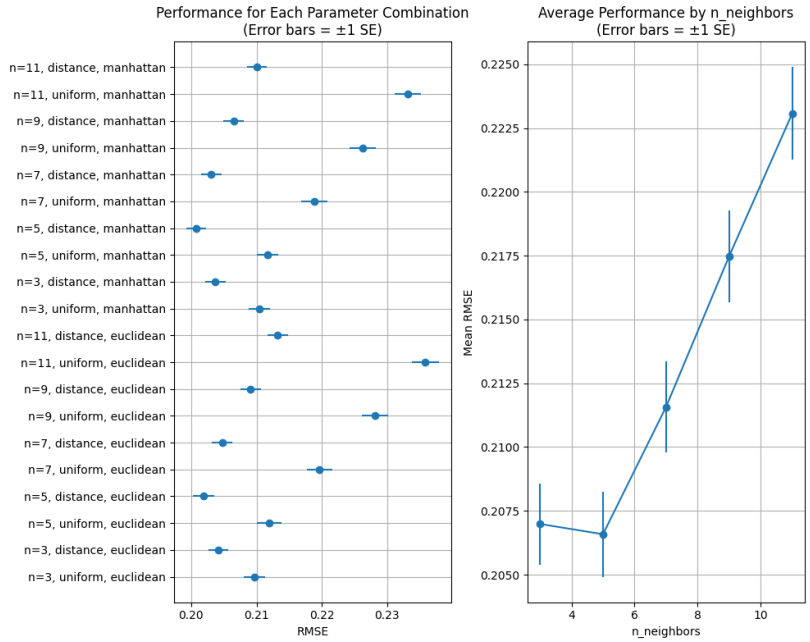

Figure 6: KNN on Skewness-Transformed Data.

Decision Tree Decision Tree showed improvement with skewness transformation, achieving an RMSE of 693 for the non-transformed dataset and 683 for the skewness-transformed

dataset. The optimal parameters for both datasets included `max_depth = 15`, `min_samples_leaf = 2`, and `min_samples_split = 10`.

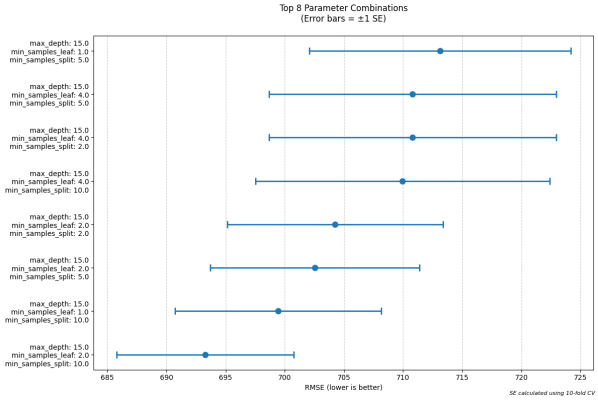

Figure 7: Decision Tree on Non-Transformed Data.

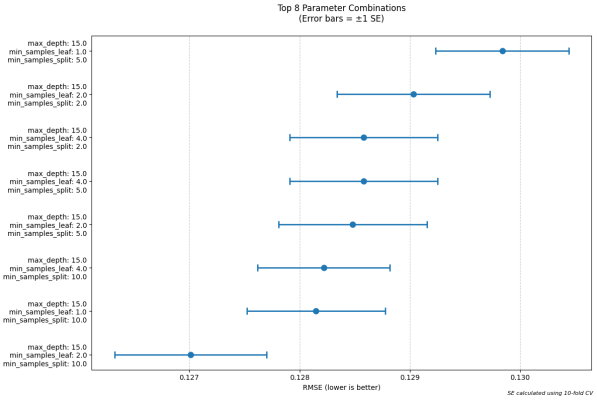

Figure 8: Decision Tree on Skewness-Transformed Data.

Random Forest Random Forest remained consistent across datasets, with an RMSE of 640 for the non-transformed dataset and 644 for the skewness-transformed dataset. This reflects the model's robustness in handling both distributions.

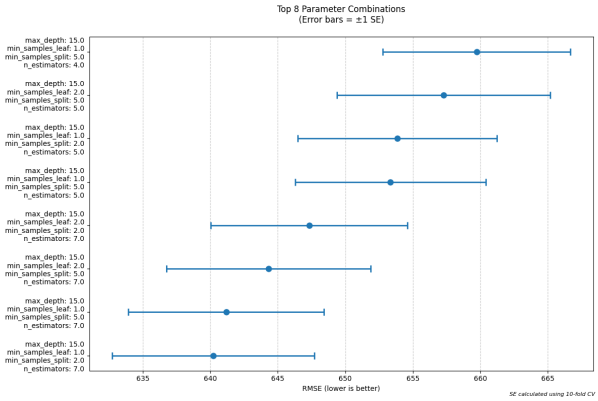

Figure 9: Random Forest on Non-Transformed Data.

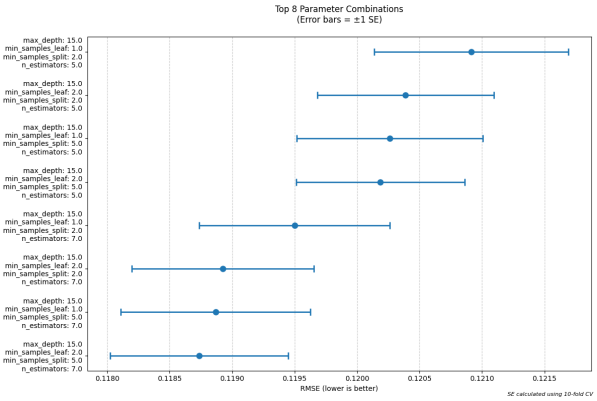

Figure 10: Random Forest on Skewness-Transformed Data.

Gradient Boosting Gradient Boosting showed slightly worse performance on the skewness-transformed dataset, with an RMSE of 1867 for the non-transformed dataset and 1872 for the skewness-transformed dataset. This indicates the model's limited sensitivity to the transformation.

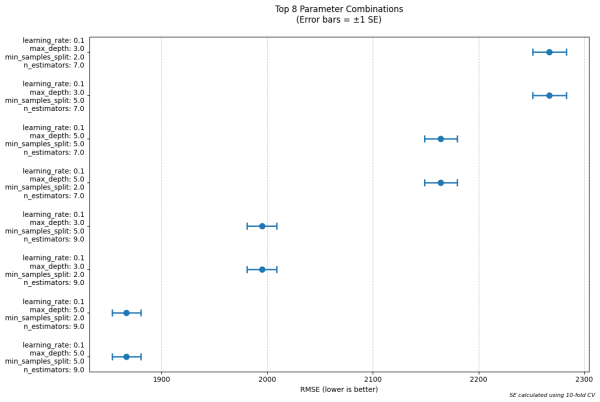

Figure 11: Gradient Boosting on Non-Transformed Data.

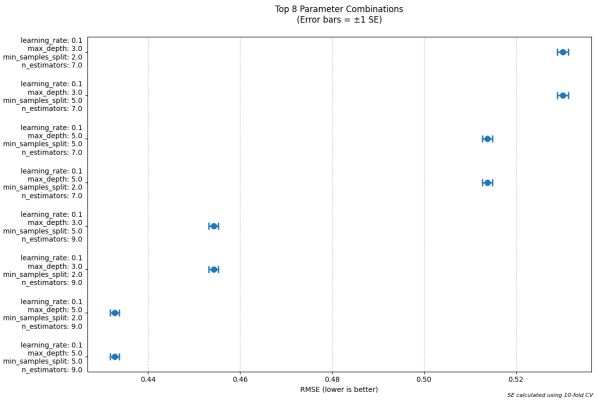

Figure 12: Gradient Boosting on Skewness-Transformed Data.

Neural Network Neural Network showed significant improvement with the skewness-transformed dataset, achieving an RMSE of 1584 for the non-transformed dataset and 547.83 for the skewness-transformed dataset. This highlights the model's sensitivity to data distribution and the importance of preprocessing.

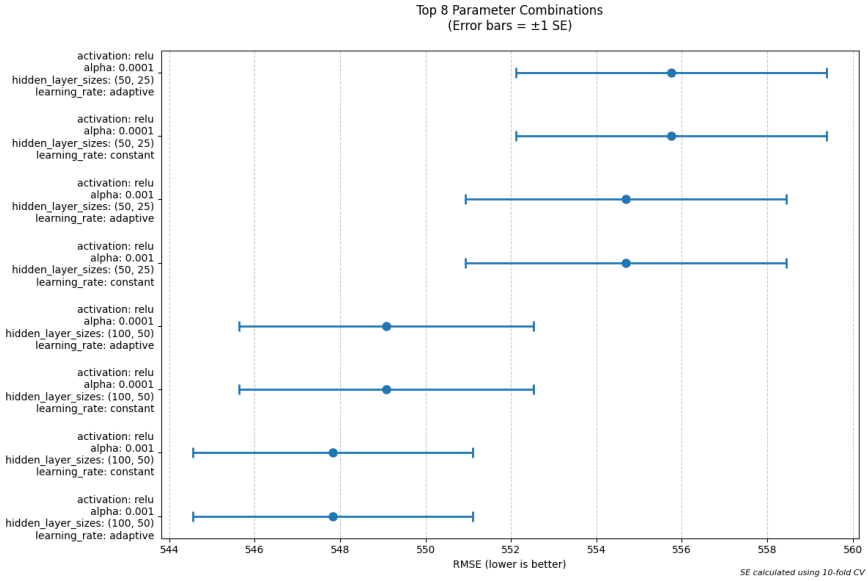

Figure 13: Neural Network on Skewness-Transformed Data.

4.4 Comparison of Results

Table 5 summarizes the RMSE results for all models across both datasets, allowing for a direct comparison.

Table 5: Comparison of Model RMSE Results for Both Datasets

Model	**Non-Transformed RMSE**	**Transformed RMSE**
Linear Regression	1156	1182
K-Nearest Neighbors (KNN)	914	913
Decision Tree	693	683
Random Forest	640	644
Gradient Boosting	1867	1872
Neural Network	1584	547.83

5 Discussion and Recommendations

The comparison of RMSE values across both datasets, as shown in Table 5, provides key insights into the effectiveness of skewness transformation and model performance. The results reveal that the impact of the transformation varies depending on the model employed, as well as the nature of the dataset.

Impact of Skewness Transformation

The skewness transformation primarily aimed to normalize the distribution of the numerical variables, potentially enhancing model performance by addressing issues with asymmetric data distributions. However, the results suggest that its benefits are not universal across all models:

- Linear Regression exhibited a slight increase in RMSE after the transformation (1156 to 1182). This behavior aligns with the nature of linear models, which are less sensitive to skewness when features are appropriately scaled.

- K-Nearest Neighbors (KNN) showed negligible improvement, with RMSE values remaining almost identical (914 for the non-transformed dataset vs. 913 for the transformed dataset). This suggests that the model's performance depends more on the proximity metric and feature scaling than on skewness correction.

- Decision Tree and Random Forest displayed marginal differences, with RMSE slightly lower for the transformed dataset in the case of Decision Tree (693 vs. 683) and slightly higher for Random Forest (640 vs. 644). These tree-based models are inherently robust to skewness due to their splitting mechanism.

- Gradient Boosting exhibited minimal change, with RMSE values slightly increasing post-transformation (1867 to 1872). The model's sensitivity to transformations may depend on hyperparameter settings, but the difference observed here is negligible.

- Neural Network was the most impacted model, achieving a dramatic reduction in RMSE (1584 to 547.83) with the skewness-transformed dataset. This highlights the importance of preprocessing for neural networks, which rely on well-distributed data for effective gradient optimization.

Best Performing Model and Recommendations

The best performing model overall is the Neural Network on the skewness-transformed dataset, with an RMSE of 547.83. This substantial improvement underscores the value of skewness transformation for models that depend heavily on data normalization. However, this comes at the cost of increased computational complexity and the need for careful tuning.

For practical applications:

- If computational resources and expertise are available, the Neural Network with skewness transformation is recommended, as it provides the lowest error.
- For scenarios where interpretability and simplicity are critical, Random Forest on the non-transformed dataset offers a robust alternative, balancing performance (RMSE: 640) and ease of implementation.
- Skewness transformation is most beneficial when working with models like Neural Networks, where distribution normalization plays a critical role. For tree-based models, the transformation offers minimal advantage and may not be necessary.

Conclusion

The decision to apply skewness transformation should be informed by the specific model and the computational resources available. For this dataset, the transformation significantly benefited the Neural Network but had limited or negligible effects on other models. As a result, skewness transformation is recommended primarily for use cases involving advanced machine learning models that are sensitive to feature distributions.

Appendix

A Code Overview

The project code is organized into two main sections, reflecting the key stages of the analysis process. Each section is outlined below, with a brief description of its purpose and structure.

A.1 Discovery and Data Preparation

This section contains the steps for exploring and preparing the dataset, ensuring it is ready for modeling. The key tasks include:

- Loading the dataset and inspecting its structure.
- Identifying numerical and categorical variables.
- Addressing skewness in numerical variables through transformation.
- Removing highly correlated features to reduce multicollinearity.
- Creating dummy variables for categorical data.
- Standardizing numerical variables using Z-score normalization.

A.2 Model Planning and Building

This section focuses on the implementation and evaluation of predictive models. It is divided into two key subsections:

- **Models without Skewness Transformation:** This subsection explores the performance of models on the original dataset without any transformations applied to address skewness.

- **Models with Skewness Transformation:** This subsection evaluates the models on a transformed version of the dataset, where skewness in numerical variables has been corrected.

The following models were implemented and evaluated in both subsections:

- Linear Regression
- K-Nearest Neighbors (KNN)
- Decision Tree
- Random Forest
- Gradient Boosting
- Neural Network

B Additional Details

For further details on the preprocessing steps, hyperparameter tuning, and evaluation metrics, refer to the full code submitted alongside this report. The code is organized in a clear and logical manner, corresponding to the sections outlined above.