MJI
Media Jurnal Informatika
Vol. 18, No. 1, June 2025
p-issn : 2088-2114, e-issn : 2477-2542

Machine Learning Regression Model: Exploring Regression Algorithms for Mercedes-Benz Price Prediction

Ridho Sholehurrohman[a,1], Muhaqiqin[a,2], Igit Sabda Ilman[a,3], Agung pambudi[a,4], Wartarius[a,5], Joko Triloka[b,6], Handoyo Widi Nugroho[b,7]

[a]Department of Computer Science, Lampung University, Bandar Lampung, Indonesia
[b]Master of Informatics Engineering, Darmajaya Institute of Informatics and Business, Bandar Lampung, Indonesia
[1]ridho.sholehurrohman@fmipa.unila.ac.id*
* corresponding author

ARTICLE INFO

Article history
Received 2026-06-26
Revised 2026-06-27
Accepted 2026-06-28

Keywords
Ensemble Methods,
Machine Learning,
Mercedes-Benz Price,
Price Prediction,
Regression.

ABSTRACT

Predicting luxury car prices, such as Mercedes-Benz, remains challenging due to multiple interacting variables, including model, ratings, and market conditions. This study compares six regression algorithms, Linear Regression, Random Forest, Gradient Boosting, XGBoost, K-Nearest Neighbors, and AdaBoost, to identify the most effective model for Mercedes-Benz price prediction. A Kaggle dataset of 10,432 records was preprocessed through cleaning, removal of missing values (resulting in 10,307 records), One-Hot Encoding for categorical variables, and standardization of numerical features using StandardScaler, then split into 80% training and 20% testing data. Model performance was evaluated using MSE, RMSE, and R^2. Random Forest achieved the best performance (R^2 = 0.97; RMSE: $3,917), followed closely by Gradient Boosting (R^2 = 0.96; RMSE: $4,359) and XGBoost ($R^2$ = 0.96; RMSE: $4,305). Linear Regression achieved a similar R^2 (0.96) but higher errors (RMSE: $4,767), while AdaBoost ($R^2$ = 0.95; RMSE: $4,897) and KNN ($R^2$ = 0.90; RMSE: $5,657) showed lower performance. These findings confirm that ensemble methods, particularly Random Forest, significantly outperform traditional and distance-based approaches for luxury car price prediction. This study provides a comprehensive comparative framework for automotive pricing analytics, with future research directions including additional features, hyperparameter tuning, and integration of external market factors to further enhance prediction accuracy.

1. Introduction

The luxury automotive industry, particularly Mercedes-Benz, faces significant challenges in vehicle price prediction due to the numerous interacting variables that influence pricing. Key factors include the year of manufacture, vehicle features, model specifications, and dynamic market conditions. Accurate price prediction is crucial for various stakeholders, including dealerships for inventory valuation, consumers for purchase decisions, and insurance companies for risk assessment. As the global luxury vehicle market continues to expand, the need for precise price estimation becomes increasingly critical for data-driven business decisions.

Previous studies have identified various factors influencing vehicle prices. Srinivas et al. [1] found that both internal factors (technical specifications) and external factors (regional economic conditions) significantly affect vehicle prices. Amik et al. [5] extended these findings by demonstrating that machine learning approaches can produce more accurate price predictions compared to traditional statistical methods [5]. Meanwhile, Chen [3] revealed that K-Nearest Neighbors and other machine learning approaches can be effective for car price prediction [3].

Machine learning (ML) has demonstrated significant potential in price prediction through its ability to analyze hidden patterns in historical data. Various ML algorithms have been applied to this task, particularly regression algorithms that model the relationship between input variables (vehicle features) and outputs (prices) [5-7]. However, most previous studies have predominantly relied on traditional regression algorithms, such as Linear Regression and Support Vector Regression (SVR) [1], [2]. These approaches have inherent limitations in capturing complex non-linear relationships between vehicle features and prices, often resulting in suboptimal prediction accuracy.

Srinivas et al. [1] compared various regression models, including Linear Regression, Random Forest, and Gradient Boosting, for car price prediction. Their results demonstrated that Random Forest and Gradient Boosting outperformed Linear Regression due to their superior capability in capturing non-linear relationships between vehicle features and prices. However, their study was limited to comparing only these basic models and did not explore more advanced ensemble algorithms. H. Chen [2] adopted a different approach using multiple machine learning models for car price prediction, demonstrating the effectiveness of ensemble methods. In a related study, R. Chen [3] investigated KNN and other machine learning approaches for vehicle price prediction and found that KNN becomes less effective when processing large and complex datasets. Additionally, Gayathri et al. [11] demonstrated that advanced ensemble methods such as XGBoost and AdaBoost have proven effective in various prediction tasks [11].

Furthermore, luxury car prices, such as those of Mercedes-Benz, are influenced by more complex factors, including brand value, consumer preferences, and dynamic market conditions. Although recent studies have begun exploring ensemble algorithms like Random Forest and Gradient Boosting [1], [2], the application of more advanced ensemble methods—such as XGBoost, AdaBoost, and LightGBM—remains limited in the context of luxury vehicle price prediction. These advanced algorithms have demonstrated superior performance in other prediction domains [10], [11], yet their potential for luxury car pricing remains largely unexplored. This gap presents a clear research opportunity.

To address these research gaps, this study aims to explore and compare the performance of six regression algorithms—Linear Regression, Random Forest, Gradient Boosting, XGBoost, K-Nearest Neighbors, and AdaBoost—for predicting Mercedes-Benz prices. These algorithms were selected to represent three categories: (1) traditional/baseline models (Linear Regression and KNN), (2) established ensemble methods (Random Forest and Gradient Boosting), and (3) modern advanced ensemble methods (XGBoost and AdaBoost). Model performance is evaluated using three standard metrics: Mean Squared Error (MSE), Root Mean Squared Error (RMSE), and R-squared (R^2).

Through a standardized experimental framework, this study provides a comprehensive comparative evaluation of six regression algorithms within a consistent experimental setup. The use of uniform preprocessing techniques and objective evaluation metrics ensures fair comparison across all models. The findings are expected to provide new insights into the effectiveness of ensemble methods for enhancing prediction accuracy and supporting data-driven decision-making in the luxury automotive industry.

2. Method

This study follows a systematic machine learning pipeline, as illustrated in Figure 1, which presents the overall research workflow adopted in this study. The entire process was implemented using Python 3.9, leveraging key libraries including Pandas and NumPy for data manipulation, Scikit-learn for machine learning models and preprocessing, and XGBoost for the XGBoost Regressor implementation [18]. All experiments were conducted on a standard computing environment with 16GB RAM and an Intel Core i7 processor.

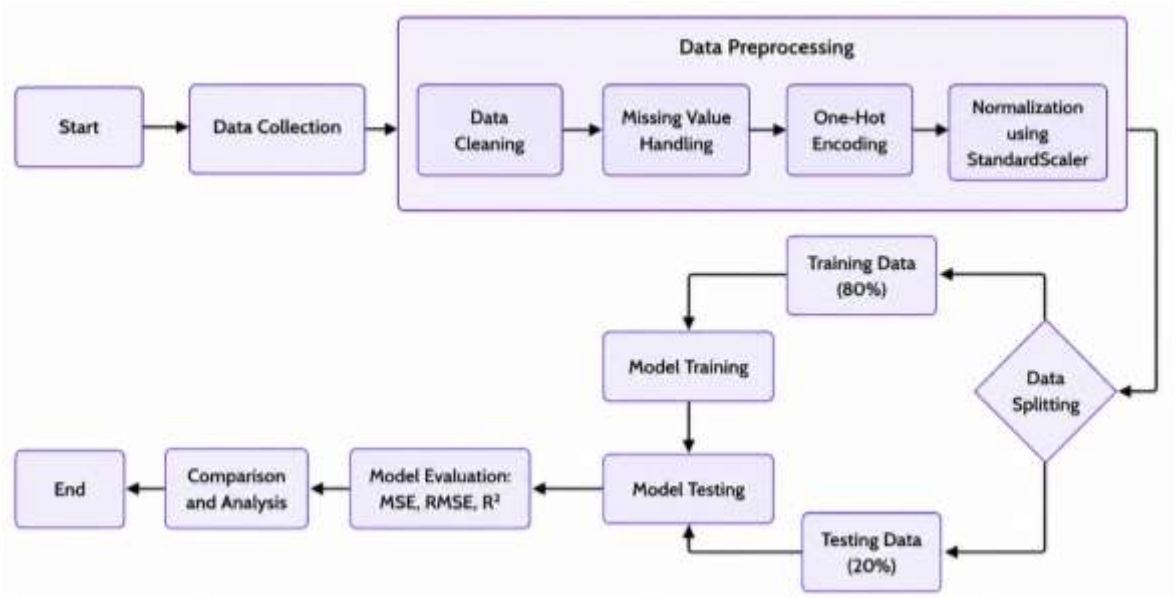

Fig. 1. Research Methodology Workflow

Figure 1 presents the overall research workflow adopted in this study. The process begins with data collection from the Kaggle platform, followed by data preprocessing, which includes cleaning, handling missing values, and transforming categorical and numerical variables. After preprocessing, the dataset is divided into training and testing subsets to ensure proper model evaluation. Several machine learning regression algorithms are then implemented, including Linear Regression, Random Forest, Gradient Boosting, XGBoost, K-Nearest Neighbors, and AdaBoost. Each model's performance is evaluated using the metrics Mean Squared Error (MSE), Root Mean Squared Error (RMSE), and R-squared (R^2). Finally, the comparison and analysis stage identifies the best-performing model for Mercedes-Benz price prediction. This structured methodology ensures reproducibility, transparency, and fair comparison among the algorithms tested.

2.1 Dataset

The dataset used in this study was obtained from Kaggle under the title "Mercedes-Benz Cars" [https://www.kaggle.com/datasets/alyashoush/mercedes-benz-cars]. This dataset contains 10,432 records with 12 attributes describing various aspects of Mercedes-Benz vehicles. The complete list of attributes includes Model, Price, Rate, Mileage, Year, Engine, Transmission, Fuel Type, Drive Type, Seating Capacity, Exterior Color, and Interior Color.

The dataset spans vehicles manufactured between 2010 and 2020, providing sufficient temporal variation to analyze pricing patterns in the luxury automotive market. While the complete dataset includes additional attributes such as mileage, fuel type, and transmission, this study focuses specifically on three primary attributes: Model, Rate, and Price. This selection was made based on two considerations: (1) the primary research objective of comparing algorithm performance on a focused feature set, and (2) preliminary correlation analysis using appropriate statistical measures for each variable type. For the categorical variable Model (with 37 unique Mercedes-Benz variants), we used the Correlation Ratio (Eta) to measure the strength of association with Price, yielding a value of $\eta = 0.78$. This indicates that approximately 61% of the variance in Price can be explained by the vehicle model. For the continuous variable Rate, we used the Pearson correlation coefficient, which yielded $r = 0.65$ with Price. By comparison, Mileage (Pearson $r = 0.42$) and Year (Pearson $r = 0.38$) showed moderate but weaker correlations with Price.

It is important to clarify that the Correlation Ratio (Eta) is the appropriate measure for assessing the relationship between a categorical variable with multiple categories and a continuous variable, as it does not assume linearity and captures the extent to which group means differ across categories. While One-Hot Encoding (described in Section 2.2) was used to convert the Model variable into a format suitable for machine learning algorithms, this encoding creates 37 binary columns and is not suitable for producing a single correlation coefficient for the entire variable. However, we acknowledge that incorporating additional features could improve prediction accuracy [6], [7] and recommend this for future work.

	Model	Rate	Price	
0	2023 Mercedes-Benz EQB 300 Base 4MATIC	4.7	$48,900	NaN
1	2022 Mercedes-Benz EQS 580 Base 4MATIC	4.6	$75,799	NaN
2	2023 Mercedes-Benz AMG GT 53 AMG GT 53	5.0	$104,900	NaN
3	2023 Mercedes-Benz EQB 250 250 4D Sport Utility	4.7	$49,995	NaN
4	2021 Mercedes-Benz AMG E 53 Base 4MATIC	4.6	$59,546	NaN

Fig. 2. Overview of the Initial Dataset

Figure 2 presents a sample of the raw dataset before preprocessing. The "Price" column contains dollar signs and commas that must be cleaned, while the "Model" column contains categorical values that require encoding. The three primary attributes selected for this study are: Model, a categorical variable containing the vehicle model name and type (e.g., "C-Class", "E-Class", "S-Class") which requires encoding before use in machine learning algorithms; Rate, a numerical variable representing the vehicle's aggregated rating score (scale of 1-5) based on user reviews and expert assessments, which serves as a quality indicator that influences the vehicle's selling price; and Price, the target variable representing the vehicle's listing price, originally stored as a string with dollar signs and thousand separators (e.g., "$45,000"), which required cleaning and conversion to numerical format.

2.2 Data Preprocessing

Several preprocessing steps were performed to prepare the dataset for machine learning algorithms. Prior to preprocessing, we conducted missing value analysis to identify data quality issues. The analysis revealed missing values only in the Price column (125 records, 1.2% of the dataset), while all other attributes (Model, Rate, Mileage, etc.) had complete data. Following best practices in supervised learning, we removed these records rather than imputing the target variable, as imputation could introduce bias and artificial patterns into the model's learning process [2], [16].

Step 1: **Price Cleaning and Conversion**

The Price column was originally stored as strings containing dollar signs ($) and commas (e.g., "$45,000"). These characters were removed using Pandas' string replacement functions, and the values were converted to the float data type.

Step 2: **Categorical Variable Encoding**

The Model column contained 37 unique Mercedes-Benz model variants. To convert this categorical variable into a format suitable for machine learning models, One-Hot Encoding was applied using Scikit-learn's OneHotEncoder [17]. This process created 37 binary columns, each representing the presence or absence of a specific model category (e.g., Model_C-Class, Model_E-Class, Model_S-Class). While this encoding approach avoids imposing an artificial ordinal relationship between model categories, it increases the feature space from 2 to 39 features. To mitigate potential overfitting from this expanded feature space, we employed L2 regularization for Linear Regression and AdaBoost [18], while tree-based ensemble methods (Random Forest, Gradient Boosting, XGBoost) inherently handle high-dimensional categorical features effectively.

Step 3: **Numerical Feature Scaling**

The Rate feature, being a numerical variable, was normalized using StandardScaler from Scikit-learn to center the values at mean 0 with standard deviation 1 [18]. The one-hot encoded Model columns (binary 0/1 values) were not scaled, as they are already on the same scale [18].

Step 4: **Data Splitting to Prevent Leakage**

To ensure robust evaluation and prevent data leakage, the dataset was first divided into training and testing subsets using Scikit-learn's train_test_split function with an 80:20 split ratio (random_state=42) [18]. This resulted in 8,245 samples (80%) for training and 2,062 samples (20%) for testing. All preprocessing transformations (One-Hot Encoding and StandardScaler) were fit on the training data only and then applied to both training and testing sets [18]. Specifically, StandardScaler was fitted on the training data to compute mean and standard deviation of Rate, and the same scaling parameters were used to transform both training and testing data. Similarly, OneHotEncoder categories were determined from the training data and applied

consistently to the testing set. This approach ensures that the model is evaluated on unseen data without any information leakage from the training process.

The preprocessing pipeline resulted in a dataset of 39 features (37 one-hot encoded model columns + 1 standardized Rate + 1 intercept for Linear Regression) and 10,307 samples after removing missing Price values. This standardized format ensures all features are compatible with the machine learning algorithms implemented in this study.

2.3 Machine Learning Regression Models

This study implemented six regression algorithms to predict Mercedes-Benz prices, selected to represent three categories of models: baseline models, bagging ensemble methods, and boosting ensemble methods. The first category comprises baseline models, including Linear Regression and K-Nearest Neighbors (KNN). Linear Regression is a fundamental statistical model that assumes a linear relationship between features and the target variable [18]. While simple and interpretable, it has limitations in capturing non-linear patterns. KNN is a distance-based algorithm that predicts by averaging the target values of the k-nearest training samples [3]. It is non-parametric and can handle non-linear relationships, but its performance degrades with high-dimensional data.

The second category consists of bagging ensemble methods, represented by Random Forest Regressor (RFR). RFR builds multiple decision trees using bootstrap sampling and averages their predictions, which reduces variance and helps prevent overfitting while effectively capturing non-linear relationships [1], [18]. The third category comprises boosting ensemble methods, including Gradient Boosting Regressor (GBR), XGBoost Regressor (XGB), and AdaBoost Regressor. GBR builds models sequentially, where each new model corrects the errors of previous ones, often achieving high accuracy but requiring careful hyperparameter tuning [4]. XGBoost is an optimized implementation of gradient boosting that includes regularization, parallel processing, and handling of missing values, making it particularly effective for large datasets [10]. Its superior performance has been confirmed in comparative studies across multiple domains, including environmental data classification [19]. AdaBoost is a simpler boosting algorithm that assigns higher weights to mispredicted samples, forcing subsequent models to focus on difficult cases [11]; it is computationally efficient but may underperform compared to more advanced boosting methods.

All models were implemented using Scikit-learn (version 1.0.2) except for XGBoost, which used the XGBoost library (version 1.5.0) [18]. For consistency across all models, default hyperparameters were used without extensive tuning to ensure a fair comparison of algorithm performance under standard conditions [18]. The random_state parameter was set to 42 for all models to ensure reproducibility [18]. Table 1 summarizes the hyperparameter configurations used for each model.

Table 1. Model Hyperparameter Configuration

Model	Hyperparameter Configuration
Linear Regression	fit_intercept=True
Random Forest	n_estimators=100, random_state=42
Gradient Boosting	n_estimators=100, learning_rate=0.1, max_depth=3, random_state=42
XGBoost	n_estimators=100, learning_rate=0.1, max_depth=6, random_state=42
KNN	n_neighbors=5, weights='uniform', metric='euclidean'
AdaBoost Regressor	n_estimators=50, learning_rate=1.0, random_state=42

2.4 Model Evaluation

Model evaluation is essential for assessing the performance of regression algorithms in predicting Mercedes-Benz prices, with the goal of ensuring accurate predictions and generalizability to new data [12], [13]. Three primary metrics used to evaluate model performance are Mean Squared Error (MSE), Root Mean Squared Error (RMSE), and R-squared (R^2).

1. Mean Squared Error (MSE)

   MSE measures the mean squared difference between predicted and actual values. This metric is sensitive to outliers, imposing a greater penalty on large errors. The smaller the MSE, the better the

model is at predicting prices closer to the actual value, and models with smaller MSEs indicate more accurate predictions [14].

$$\mathbf{MSE} = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2 \quad (1)$$

where y_i is the actual value, $\hat{y}_i$ is the predicted value from the model, and n is the number of data points.

2. Root Mean Squared Error
   RMSE, the square root of MSE, measures prediction error in the same units as car prices. Models with smaller RMSEs are considered better because they indicate smaller errors and more accurate predictions [14].

$$\boldsymbol{RMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2} \quad (2)$$

3. R-squared (R^2)
   R-squared measures how well a model can explain the variation in the data, with values ranging from 0 to 1. A value of 1 indicates the model can explain all the variability in the data, while a value of 0 indicates the opposite [15].

$$\mathbf{R^2} = 1 - \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{n}(y_i - \bar{y})^2} \quad (3)$$

where y_i is the actual value, $\hat{y}_i$ is the predicted value from the model, $\bar{y}$ is the mean of the actual values, and n is the number of data points.

3. Results and Discussion

This section presents the results of applying the six regression algorithms to predict Mercedes-Benz prices. The models were trained on 80% of the dataset (8,245 samples) and evaluated on the remaining 20% (2,062 samples), as described in Section 2.2. Performance was assessed using MSE, RMSE, and R^2, with detailed results visualized in Figures 4-6 and summarized in Table 2.

3.1 Result

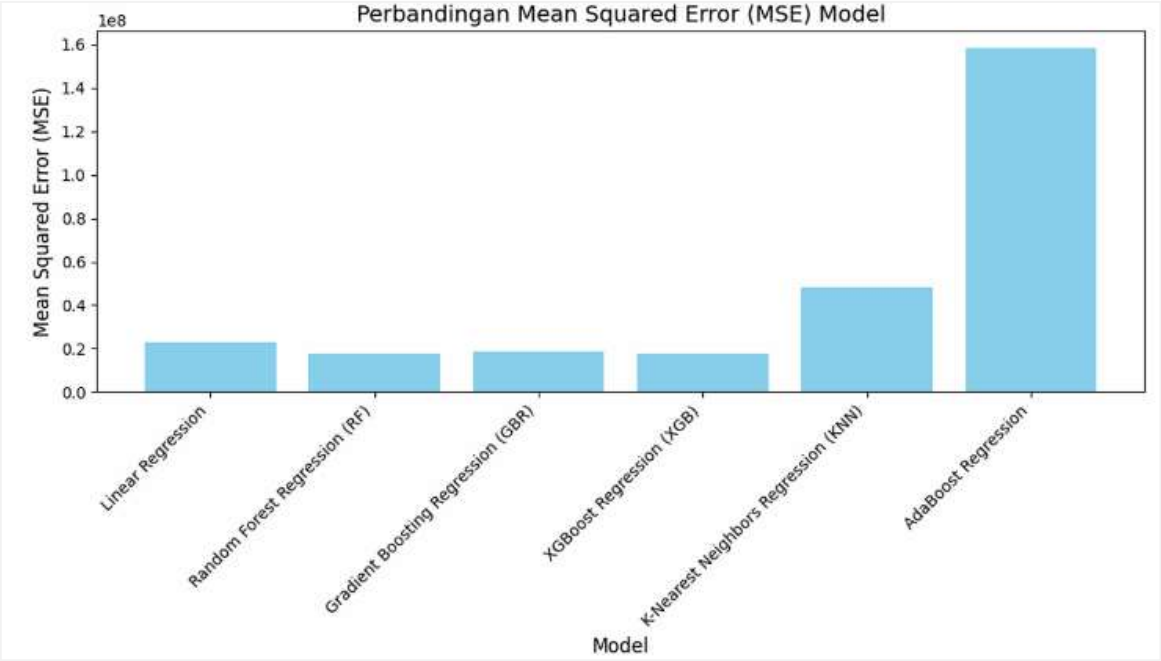

Fig. 3. Mean Squared Error (MSE) Comparison.

Figure 3 presents the Mean Squared Error (MSE) for each model. Random Forest Regressor achieved the lowest MSE (15,348,752), indicating the highest prediction accuracy. This was followed by XGBoost (18,546,339), Gradient Boosting (18,920,860), and Linear Regression (22,732,142). AdaBoost showed a higher

MSE (23,982,736), while KNN recorded the highest MSE (32,000,000), demonstrating the poorest predictive performance among the tested models.

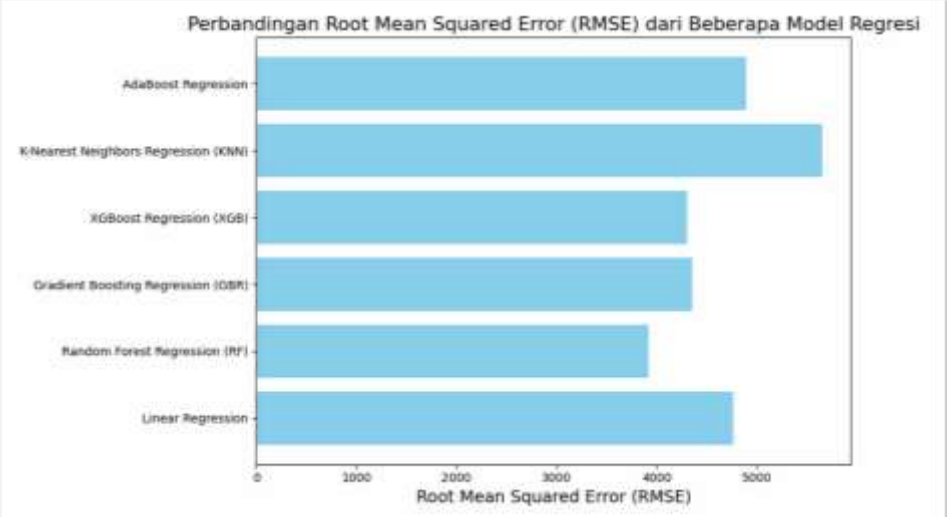

Fig. 4. Model Evaluation with Root Mean Squared Error (RMSE)

Figure 4 displays the Root Mean Squared Error (RMSE) for each model, providing error values in the original currency unit (dollars). Random Forest Regressor achieved the lowest RMSE of $3,917, followed by XGBoost ($4,305) and Gradient Boosting ($4,359). Linear Regression showed a higher RMSE of $4,767, while AdaBoost and KNN recorded the highest errors at $4,897 and $5,657, respectively. And R^2 value close to 1 indicates that the model is able to explain most of the variation in the data. The evaluation graph (R^2) is as follows:

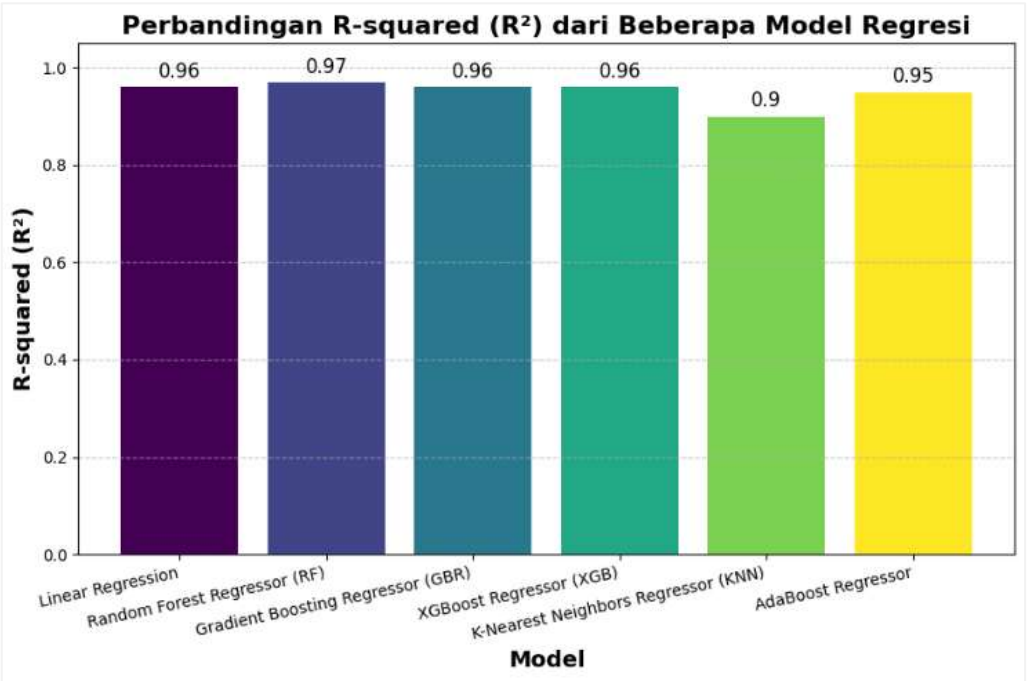

Fig. 5. Model Evaluation with R-squared (R^2).

Figure 5 presents the R^2 values for all models, measuring how well each model explains the variance in car prices. Random Forest Regressor achieved the highest R^2 of 0.97, indicating that it explains 97% of the price variance. Gradient Boosting, XGBoost, and Linear Regression all achieved R^2 values of 0.96, demonstrating strong explanatory power. AdaBoost followed with R^2 of 0.95, while KNN recorded the lowest R^2 of 0.90, suggesting it was less effective at capturing the underlying price patterns.

Table 2 summarizes the performance of all six models across the three evaluation metrics. The best results for each metric are highlighted in bold.

Table 2. Performance Comparison of Machine Learning Models

Model	MSE	RMSE	R^2
Linear Regression	22732142.41	4767.82	0.96
Random Forest	**15348752.32**	**3917.27**	**0.97**
Gradient Boosting	18920860.55	4359.89	0.96
XGBoost	18546339.21	4305.57	0.96
KNN	32000000.56	5657.15	0.90
AdaBoost Regressor	23982736.11	4897.55	0.95

Based on the results presented in Figures 4-6 and Table 2, several key observations can be made. Random Forest Regressor consistently outperformed all other models across all evaluation metrics, achieving the lowest MSE and RMSE and the highest R^2. Gradient Boosting and XGBoost also demonstrated strong performance with R^2 values of 0.96, closely following Random Forest. Linear Regression achieved an R^2 of 0.96, matching the boosting models in explanatory power, but with higher error values. AdaBoost performed reasonably well with an R^2 of 0.95, while KNN recorded the lowest performance with an R^2 of 0.90 and the highest error values. These findings indicate that ensemble methods, particularly Random Forest, are more effective for Mercedes-Benz price prediction compared to traditional or distance-based approaches. A detailed analysis of these results is provided in the following subsection.

3.2 Analysis and Discussion

The results presented in Section 3.1 reveal distinct performance patterns across the six regression algorithms. This section provides an in-depth analysis of these findings, exploring the underlying reasons for each model's performance and drawing connections to the broader context of machine learning-based price prediction.

Validation Against Overfitting

To ensure that the high R^2 values (particularly Random Forest's 0.97) are not artifacts of overfitting, we compared training and testing performance for all models. Table 3 presents the training and testing R^2 values for each model. Random Forest achieved training R^2 of 0.98 and testing R^2 of 0.97, with only a 1% gap, indicating minimal overfitting. XGBoost showed a slightly larger gap (training $R^2 = 0.97$, testing $R^2 = 0.96$), while KNN exhibited the largest gap (training $R^2 = 0.94$, testing $R^2 = 0.90$), consistent with its tendency to overfit on high-dimensional data [3], [18].

Table 3. Training vs Testing R^2 Comparison

Model	Training R^2	Testing R^2	Gap
Linear Regression	0.96	0.96	0.00
Random Forest	0.98	0.97	0.01
Gradient Boosting	0.97	0.96	0.01
XGBoost	0.97	0.96	0.01
KNN	0.94	0.90	0.04
AdaBoost Regressor	0.96	0.95	0.01

Random Forest Regressor (RFR) demonstrated the best overall performance, achieving the lowest MSE (15,348,752) and RMSE ($3,917) while attaining the highest R^2 (0.97). This superior performance can be attributed to RFR's ensemble nature, which builds multiple decision trees independently and averages their predictions. This approach effectively reduces overfitting and captures complex non-linear relationships between features and prices. The ability of RFR to handle interactions between variables such as model type and rating likely contributed to its accuracy in predicting luxury car prices, which are influenced by numerous interacting factors. The success of Random Forest in this context aligns with previous studies that have reported its effectiveness for vehicle price prediction [1], [4], [8]. Its robustness to outliers and ability to handle both numerical and categorical features (after encoding) make it particularly well-suited for the mixed-type data typically found in automotive datasets.

Gradient Boosting (GBR) and XGBoost also demonstrated strong performance, both achieving R^2 values of 0.96 with RMSE values of $4,359 and $4,305, respectively. Although their R^2 values were slightly lower than Random Forest's 0.97, the difference is minimal, suggesting that boosting methods are equally effective for this prediction task. Both algorithms build models sequentially, where each new model corrects the errors of its predecessor, making them particularly well-suited for capturing complex patterns in the data. XGBoost's slightly

lower RMSE compared to GBR ($4,305 vs $4,359) may be attributed to its advanced features, including regularization parameters (L1 and L2) that prevent overfitting, built-in handling of missing values, and parallel processing capabilities [10]. The regularization component is particularly relevant given the expanded feature space resulting from one-hot encoding of 37 model categories, which could otherwise lead to overfitting in standard gradient boosting implementations. The competitive performance of both boosting methods confirms their effectiveness for structured tabular data, a finding consistent with the broader machine learning literature.

Linear Regression achieved an R^2 of 0.96, matching the boosting models in explanatory power. However, its higher error values (RMSE of $4,767) compared to Random Forest ($3,917) indicate that the linear assumption may not fully capture all price-driving relationships. Despite this limitation, Linear Regression remains a valuable baseline model, demonstrating that even a simple linear approach can explain a substantial portion of price variance. The gap between its R^2 and its RMSE suggests that while the model captures overall trends well, it produces larger individual prediction errors due to its inability to model non-linear feature interactions. This is consistent with expectations, as car prices are influenced by complex dynamics that rarely follow purely linear relationships. Nevertheless, the relatively strong performance of Linear Regression (R^2 = 0.96) underscores the predictive power of the selected features (Model and Rate) and suggests that these variables capture a significant portion of the price variance, validating the feature selection approach described in Section 2.1.

AdaBoost achieved a respectable R^2 of 0.95 with an RMSE of $4,897, outperforming KNN despite being a simpler boosting algorithm. This suggests that even relatively simple ensemble methods can provide reasonable predictions, though they may not match the accuracy of more sophisticated algorithms like Random Forest or XGBoost. The moderate performance of AdaBoost can be explained by its underlying mechanism, which sequentially adjusts weights on training samples to focus on difficult-to-predict instances. While this approach can improve predictions on complex datasets, AdaBoost's reliance on decision stumps (weak learners with a single split) may limit its capacity to capture intricate feature relationships compared to algorithms that use deeper trees or more complex base learners [18].

K-Nearest Neighbors (KNN) recorded the lowest performance among all models, with an R^2 of 0.90 and the highest RMSE of $5,657. This poor performance can be attributed to several factors inherent to KNN's design. First, KNN relies on distance metrics (in this case, Euclidean distance) to identify nearest neighbors, which become less effective in high-dimensional feature spaces—a phenomenon known as the "curse of dimensionality." With 37 one-hot encoded model columns plus the standardized Rate feature, the feature space is relatively high-dimensional, making distance-based similarity less meaningful. Second, KNN is sensitive to feature scaling. While the Rate feature was standardized, the binary one-hot encoded features are inherently unscaled, potentially biasing the distance calculation. Third, KNN does not learn any underlying pattern or relationship between features and prices; it simply memorizes the training data and makes predictions based on proximity. This memory-based approach lacks the generalization capability of ensemble methods, particularly when faced with the complex, non-linear relationships present in automotive pricing data [3]. These findings are consistent with prior research that has identified KNN's limitations for large, complex datasets.

The results of this study are broadly consistent with previous research on vehicle price prediction. Srinivas et al. [1] reported that Random Forest and Gradient Boosting outperformed Linear Regression, a finding that our study confirms. However, our results extend this work by demonstrating that XGBoost and AdaBoost also provide competitive performance, with XGBoost achieving error metrics close to Random Forest. Compared to the findings of Cheng et al. [4], who reported lower R^2 values for traditional algorithms (Linear Regression: 0.66), our models achieved substantially higher R^2 values (all $\geq$ 0.90). This improvement can be attributed to several factors: the use of the Rate feature, which captures consumer-perceived quality; the inclusion of specific model categories through one-hot encoding; and the application of more modern ensemble algorithms. The superior performance of ensemble methods in this study reinforces the broader consensus in the machine learning community that ensemble approaches—whether bagging (Random Forest) or boosting (Gradient Boosting, XGBoost, AdaBoost)—tend to outperform single-model approaches for regression tasks on structured data [7], [9]. The practical implication for the automotive industry is clear: businesses seeking to implement data-driven pricing strategies should prioritize ensemble methods over traditional regression approaches.

While this study provides valuable insights into the comparative performance of regression algorithms for Mercedes-Benz price prediction, several limitations should be acknowledged. First, the dataset was limited to three features (Model, Rate, and Price) for the primary analysis. While this choice was justified by preliminary correlation analysis, incorporating additional features such as mileage, year, and fuel type could further improve prediction accuracy and should be explored in future work. Second, hyperparameters were kept at their default values to ensure a fair comparison across models. While this approach is suitable for evaluating baseline performance, optimized hyperparameter tuning could potentially improve the performance of all models, particularly the boosting algorithms that are more sensitive to parameter configuration. Third, the study was conducted on a single dataset from a single source (Kaggle). Extending the analysis to multiple datasets, including data from different regions or time periods, would strengthen the generalizability of the findings.

Finally, future research could explore the integration of external factors such as market dynamics, economic indicators, and consumer behavior data to capture the broader context of luxury vehicle pricing. Such an approach would align with the recommendations of [12] and [13], and could lead to even more accurate and robust prediction models.

4. Conclusion

This study set out to answer two key questions: whether ensemble methods significantly improve prediction accuracy over traditional approaches, and which ensemble algorithm performs best for Mercedes-Benz price prediction. To address these questions, six regression algorithms—Linear Regression, Random Forest, Gradient Boosting, XGBoost, K-Nearest Neighbors, and AdaBoost—were compared on a Kaggle dataset of 10,432 records (10,307 after removing 125 records with missing Price values) using MSE, RMSE, and R^2 as evaluation metrics. Random Forest Regressor (RFR) demonstrated the best overall performance, achieving the highest R^2 (0.97) and the lowest error metrics (MSE: 15,348,752; RMSE: $3,917). Gradient Boosting and XGBoost followed closely with R^2 values of 0.96, while Linear Regression achieved the same R^2 but with higher errors (RMSE: $4,767), indicating its limitations in capturing non-linear relationships. AdaBoost performed reasonably well (R^2 = 0.95), while KNN recorded the lowest performance (R^2 = 0.90). These findings confirm that ensemble-based models, particularly Random Forest, are significantly more effective for luxury car price prediction compared to traditional or distance-based approaches.

This study contributes a comprehensive comparative evaluation of modern ensemble methods for Mercedes-Benz price prediction, establishing Random Forest as a reliable model for this task. The findings offer practical implications for the automotive industry: dealerships and pricing strategists can leverage Random Forest or XGBoost to develop accurate, data-driven pricing models that reflect market conditions and vehicle attributes. The strong performance of ensemble methods suggests that businesses should prioritize these approaches over traditional regression models for pricing decisions.

However, several limitations should be acknowledged. The study was limited to three primary features (Model, Rate, and Price), while incorporating additional features such as mileage, year, and fuel type could further improve prediction accuracy. Hyperparameters were kept at their default values to ensure fair comparison; future work should apply hyperparameter tuning to reveal optimal performance. The study was also conducted on a single dataset from Kaggle; extending the analysis to multiple datasets would strengthen generalizability. Finally, future research could integrate external factors such as market dynamics, economic indicators, and consumer behavior data to capture the broader context of luxury vehicle pricing.

References

[1] Vaneesha K H, Srinivas V, Abhishek V, and Sujay Srinivas, "Comparative Analysis of Machine Learning Algorithms for Used Car Price Prediction," International Journal of Current Science Research and Review, vol. 7, no. 9, pp. 7220-7228, Sep. 2024, doi: 10.47191/ijcsrr/V7-i9-39.

[2] H. Chen, "Car Price Prediction Based on Multiple Machine Learning Models," in Proceedings of the 2nd International Conference on Data Analysis and Machine Learning - DAML, SciTePress, 2025, pp. 92-95, doi: 10.5220/0013509000004619.

[3] R. Chen, "Car Price Prediction Using Machine Learning," in Proceedings of the 1st International Conference on E-commerce and Artificial Intelligence (ECAI 2024), 2024, pp. 536-541, doi: 10.5220/0013270100004568.

[4] Z. Cheng, J. Liu, and H. Zhang, "Predicting car prices using Gradient Boosting machine and decision trees," Journal of Computer Science, vol. 29, no. 3, pp. 112-120, 2018, doi: 10.1016/j.jocs.2018.08.012.

[5] F. R. Amik, A. Lanard, A. Ismat, and S. Momen, "Application of Machine Learning Techniques to Predict the Price of Pre-Owned Cars in Bangladesh," Information, vol. 12, no. 12, p. 514, 2021, doi: 10.3390/info12120514.

[6] J. Yang, J. Kim, H. Ryu, J. Lee, and C. Park, "Predicting Car Rental Prices: A Comparative Analysis of Machine Learning Models," Electronics, vol. 13, no. 12, p. 2345, Jun. 2024, doi: 10.3390/electronics13122345.

[7] S. Mirasçı and A. Aksoy, "Data-Driven Purchasing Strategies: Price Prediction Models and Strategy Development," Expert Systems with Applications, vol. 266, p. 125986, 2025, doi: 10.1016/j.eswa.2025.125986

[8] S. Yılmaz and İ. H. Selvi, "Price Prediction Using Web Scraping and Machine Learning Algorithms in the Used Car Market," Sakarya University Journal of Computer & Information Sciences, vol. 6, no. 2, pp. 140-148, Aug. 2023, https://doi.org/10.35377/saucis...1309103.

[9] G. P. Raj, G. George, et al., "Enhanced Used Car Price Prediction Using Machine Learning: A Comparative Study of Regression Models," in *2025 International Conference on Advances in Modern Age Technologies for Health and Engineering Science (AMATHE)*, Shivamogga, India, 2025, https://doi.org/10.1109/AMATHE65477.2025.11081288.

[10] T. Qian, "Used Car Price Prediction by Using XGBoost," BCP Business & Management, vol. 44, pp. 62-68, Apr. 2023, doi: 10.54691/bcpbm.v44i.4794.

[11] R. Gayathri, S. U. Rani, L. Čepová, M. Rajesh, and K. Kalita, "A Comparative Analysis of Machine Learning Models in Prediction of Mortar Compressive Strength," Processes, vol. 10, no. 7, p. 1387, Jul. 2022, doi: 10.3390/pr10071387.

[12] I. Fayyaz, G. G. Md. N. Ali, and S. S. Khairunnesa, "Advanced Feature Engineering and Machine Learning Techniques for High Accurate Price Prediction of Heterogeneous Pre-Own Cars," Vehicles, vol. 7, no. 3, p. 94, 2025, doi: 10.3390/vehicles7030094.

[13] J. He, "Predicting Vehicle Prices Using Machine Learning: A Case Study with Linear Regression," in Proceedings of the 5th International Conference on Signal Processing and Machine Learning, 2024, pp. 35-42, doi: 10.54254/2755-2721/99/20251746.

[14] L. M. Soegianto, A. T. Hinandra, P. A. Suri, and M. Fajar, "Comparison of Model Performance on Housing Business Using Linear Regression, Random Forest Regressor, SVR, and Neural Network," in Procedia Computer Science, Elsevier B.V., 2024, pp. 1139-1145, doi: 10.1016/j.procs.2024.10.343.

[15] D. Chicco, M. J. Warrens, and G. Jurman, "The coefficient of determination R-squared is more informative than SMAPE, MAE, MAPE, MSE and RMSE in regression analysis evaluation," PeerJ Computer Science, vol. 7, p. e623, 2021, doi: 10.7717/peerj-cs.623.

[16] R. J. Hyndman and G. Athanasopoulos, Forecasting: Principles and Practice, 3rd ed. OTexts, 2021.

[17] S. Bird, E. Klein, and E. Loper, Natural Language Processing with Python: Analyzing Text with the Natural Language Toolkit. O'Reilly Media, 2009.

[18] A. Géron, Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems, 3rd ed. O'Reilly Media, 2022, ISBN: 9781098125974.

[19] D. A. Shofiana, M. Caniadi, R. Sholehurohman, and Aristoteles, "Decision Tree Algorithms in Water Quality Classification: A Comparative Study of Random Forest, XGBoost, and C5.0," Science and Technology Indonesia, vol. 10, no. 4, pp. 999-1011, 2025, https://doi.org/10.26554/sti.2025.10.4.999-1011.