

Research Article

Domain-Driven Feature Engineering and Optimized Stacking Ensemble Learning for Used Car Price Prediction

Dhani Ian Maulana ¹, De Rosal Ignatius Moses Setiadi ^{1,2,*}, Yew Kee Wong ³, and Zahir Edrees ⁴

¹ Faculty of Computer Science, Universitas Dian Nuswantoro, Semarang, Central Java 50131, Indonesia; e-mail : ianmaulana204@gmail.com; moses@dsn.dinus.ac.id

² Research Group for Quantum Computing and Materials Informatics, Universitas Dian Nuswantoro, Semarang, Central Java 50131, Indonesia

³ Hong Kong Chu Hai College, Castle Peak Bay, Tuen Mun, N.T. Hong Kong; e-mail : yewkeewong.eric@gmail.com

⁴ Software Engineering Department, International Science and Technology University, Janiszowska 8, 02-264 Warsaw, Poland; e-mail : zahir.edrees@istu.edu.pl

* Corresponding Author : De Rosal Ignatius Moses Setiadi

Abstract: Accurate used car price prediction remains a challenging task because vehicle values are governed by complex nonlinear relationships among depreciation, engine characteristics, and market perception that are not adequately represented by the original predictor attributes. While recent studies have primarily focused on developing increasingly sophisticated regression models, comparatively limited attention has been given to systematically integrating domain-driven feature engineering, hyperparameter optimization, and interpretable ensemble learning within a unified prediction framework. This study proposes a domain-driven feature engineering and optimized stacking ensemble learning framework for used car price prediction. The proposed feature engineering pipeline enriches the original predictor space by constructing interaction features, engine performance indicators, brand- and market-related representations, and clustering-based features. These engineered features are integrated within an optimized stacking ensemble comprising CatBoost, LightGBM, and XGBoost as base learners and Ridge Regression as the meta-learner, with hyperparameters optimized using Optuna. The proposed framework is evaluated on the publicly available Used Car Price Prediction benchmark dataset containing 4,009 vehicle records using both an independent hold-out test set and 5-fold cross-validation. Experimental results demonstrate that the proposed framework achieves an R^2 of 0.9112 on the hold-out test set and 0.9023 ± 0.0025 under cross-validation, consistently outperforming the individual gradient boosting models while exhibiting stable generalization across both evaluation protocols. Ablation experiments further reveal that domain-driven feature engineering contributes more substantially to predictive performance than hyperparameter optimization alone, with interaction features providing the largest performance gain. Moreover, SHAP analysis confirms that engineered interaction features dominate the prediction process, highlighting vehicle age-mileage interactions as the most influential factors affecting resale value. These findings demonstrate that integrating domain-driven feature engineering with optimized stacking ensemble learning provides a robust, interpretable, and effective framework for accurate used car price prediction.

Keywords: Domain-Driven Feature Engineering; Explainable Artificial Intelligence (XAI); Gradient Boosting; Stacking Ensemble Learning; Sustainable Transportation; Used Car Price Prediction; Vehicle Valuation.

Received: April, 16th 2026

Revised: July, 7th 2026

Accepted: July, 14th 2026

Published: July, 15th 2026

Curr. Ver.: July, 15th 2026



Copyright: © 2026 by the authors.
Submitted for possible open access publication under the terms and conditions of the Creative Commons Attribution (CC BY SA) license (<https://creativecommons.org/licenses/by-sa/4.0/>)

1. Introduction

The used car market has become one of the fastest-growing segments of the automotive industry, driven by increasing consumer demand for affordable vehicles and the rapid expansion of online vehicle marketplaces [1], [2]. Accurate vehicle valuation, however, remains a

challenging task because resale prices are influenced by multiple heterogeneous factors, including brand, model, vehicle age, mileage, engine specifications, vehicle condition, and ownership history. The absence of an objective pricing mechanism often leads to substantial price discrepancies among vehicles with similar characteristics, increasing uncertainty for both buyers and sellers [1], [3].

Machine learning has become the dominant approach for used car price prediction because vehicle values are governed by complex nonlinear relationships that are difficult to capture using conventional statistical models [1], [2], [4], [5]. Previous studies have investigated a wide range of regression algorithms, including Linear Regression, Support Vector Machines (SVM), K-Nearest Neighbors (KNN), Decision Trees, Random Forests, Gradient Boosting, and XGBoost [1], [2], [4]. Nevertheless, predictive performance depends not only on the learning algorithm but also on how effectively vehicle characteristics are represented. Used vehicle datasets typically contain numerical, categorical, and semi-structured attributes, such as engine specifications, whose predictive information cannot be fully exploited without appropriate feature transformation [2], [4].

To improve predictive capability, ensemble learning has been widely adopted because it combines complementary prediction patterns from multiple models to enhance robustness and generalization [5], [6]. Among existing approaches, stacking has demonstrated promising performance by integrating predictions from multiple base learners through a meta-learner [5], [7]. For example, Fayyaz et al. [6] combined Support Vector Regression, Random Forest, and XGBoost within a stacking framework, whereas Onserio and Mukudi [8] proposed a stacking ensemble for the Kenyan used car market. Despite these encouraging results, existing studies have primarily emphasized algorithm comparison or ensemble design, while the contribution of domain-driven feature engineering has received comparatively limited systematic investigation. Moreover, model evaluation is often restricted to a single experimental setting, making it difficult to quantify the individual contributions of feature engineering, model optimization, and ensemble architecture [5], [7].

Among modern machine learning algorithms, gradient boosting methods consistently achieve strong performance on tabular datasets because they effectively model nonlinear relationships and complex feature interactions [9], [10]. CatBoost efficiently handles categorical variables [11], LightGBM emphasizes computational efficiency through histogram-based learning [12] and XGBoost incorporates explicit regularization to improve generalization [13]. Their complementary learning characteristics make them suitable candidates for integration within a stacking framework [14]. Furthermore, because the performance of these algorithms is highly sensitive to hyperparameter configuration, efficient optimization techniques such as Optuna provide a practical alternative to conventional search methods by exploring large hyperparameter spaces more effectively while reducing computational cost [15]–[17].

Although previous studies have demonstrated the effectiveness of stacking ensembles and gradient boosting algorithms, three important challenges remain. First, domain-driven feature engineering has rarely been investigated as a primary contributor to predictive performance. Second, the integration of multiple complementary gradient boosting algorithms within an Optuna-optimized stacking framework has received limited attention. Third, relatively few studies have systematically quantified the individual contributions of engineered feature groups, model optimization, and ensemble components through comprehensive ablation analysis and explainable artificial intelligence.

To address these limitations, this study proposes a domain-driven feature engineering and optimized stacking ensemble learning framework for used car price prediction. The proposed framework integrates CatBoost, LightGBM, and XGBoost as base learners, with hyperparameters optimized using Optuna and Ridge Regression employed as the meta-learner. A domain-driven feature engineering pipeline is developed to construct informative representations through engine specification extraction, interaction feature construction, clustering-based market segmentation, and transformation of vehicle condition attributes. The proposed framework is evaluated using both an independent hold-out test set and 5-fold cross-validation, while SHAP analysis is employed to interpret feature contributions and support model transparency. The main contributions of this study are summarized as follows.

- An optimized stacking ensemble learning framework integrating CatBoost, LightGBM, XGBoost, and Ridge Regression is proposed.
- A domain-driven feature engineering pipeline is developed to enrich the original vehicle attributes through engine-, interaction-, clustering-, and condition-based features.

- A comprehensive ablation and SHAP-based analysis is conducted to quantify the contribution of feature engineering and the proposed ensemble architecture.

The remainder of this paper is organized as follows. Section 2 reviews related studies on used car price prediction. Section 3 presents the dataset, preprocessing procedures, feature engineering pipeline, and the proposed methodology. Section 4 discusses the experimental results, including predictive performance, ablation analysis, and SHAP-based interpretation. Section 5 compares the proposed framework with previous studies, and Section 6 concludes the paper and outlines directions for future research.

2. Related Work

Machine learning has been extensively applied to used car price prediction to overcome the limitations of conventional statistical and mathematical modeling approaches. Serrat and Djebbar [18] demonstrated that mathematical modeling provides a useful foundation for representing the relationship between vehicle characteristics and selling price. However, the pricing mechanism of used vehicles is inherently nonlinear and depends on complex interactions among multiple attributes, limiting the capability of traditional analytical models to accurately capture market behavior. Consequently, machine learning algorithms have become the dominant approach for modeling used vehicle prices because of their ability to learn complex nonlinear relationships directly from data.

Early machine learning studies primarily focused on identifying the most suitable prediction algorithm. Lin [4] compared several regression models, including Linear Regression, Decision Tree, Random Forest, XGBoost, and neural networks, using relatively simple preprocessing based on one-hot encoding. Similarly, Khotimah et al. [5] reported that machine learning models consistently outperformed conventional linear regression after basic data cleaning. Bhatnagar et al. [19] further demonstrated that competitive prediction performance could be achieved using only a subset of highly relevant features with Random Forest, whereas Budiono et al. [20] identified vehicle brand, model, production year, mileage, fuel type, transmission, and engine specifications as the primary factors influencing used vehicle prices. Collectively, these studies indicate that although algorithm selection affects predictive performance, the quality of feature representation remains equally important.

As research progressed, increasing attention was directed toward feature engineering and ensemble learning. Dutulescu et al. [21] proposed a deep learning framework integrating multiple neural architectures to improve vehicle price estimation, while Bergmann and Feuerriegel [22] demonstrated that incorporating detailed vehicle equipment information significantly improved prediction accuracy compared with relying solely on standard vehicle attributes. From the perspective of feature engineering, Andrade-Girón et al. [2] showed that feature construction and selection substantially improve predictive performance by generating more informative data representations. These findings suggest that enhancing feature representation can be as important as improving the prediction algorithm itself.

Ensemble learning has also become an effective strategy for improving model generalization in used car price prediction. Fayyaz et al. [6] combined Support Vector Regression, Random Forest, and XGBoost within a stacking architecture and reported improved prediction accuracy after applying feature engineering. Likewise, Onserio and Mukudi [8] demonstrated that a stacking ensemble composed of Random Forest, Support Vector Machine, K-Nearest Neighbor, and Gradient Boosting outperformed individual regression models on the Kenyan used vehicle market. In contrast, Guo and Zhang [1] showed that a single XGBoost model could also achieve highly competitive performance on a substantially larger dataset. These studies collectively indicate that both gradient boosting models and stacking ensembles are highly effective for tabular vehicle valuation tasks, although their performance remains dependent on model configuration and dataset characteristics.

Recent studies have further emphasized model interpretability and hyperparameter optimization. Sakib et al. [16] demonstrated that combining ensemble learning with genetic algorithm-based feature selection improves predictive performance by constructing a more informative feature subset. Bussmann et al. [23] highlighted the importance of explainable machine learning through SHAP analysis, enabling the contribution of individual predictors to be interpreted more transparently. Meanwhile, Winarso et al. [2] reported that systematic hyperparameter optimization consistently outperformed manually configured models,

indicating that model configuration plays a critical role in achieving reliable predictive performance.

Despite these advances, several research gaps remain. First, previous studies have predominantly focused on algorithm comparison or ensemble design, whereas the contribution of domain-driven feature engineering has rarely been investigated systematically. Second, existing stacking frameworks generally combine heterogeneous base learners or employ manually configured hyperparameters, while the integration of Optuna-based hyperparameter optimization with a stacking architecture composed entirely of complementary gradient boosting algorithms has received limited attention [3]. Third, most studies emphasize overall predictive accuracy without investigating the contribution of individual feature groups, ensemble components, or feature representations through comprehensive ablation analysis. Finally, although SHAP has increasingly been adopted for model interpretation, its use to explain how domain-driven feature engineering influences prediction performance remains limited in the context of used car price prediction. Motivated by these research gaps, this study integrates domain-driven feature engineering, Optuna-based hyperparameter optimization, a gradient boosting stacking ensemble consisting of CatBoost, LightGBM, and XGBoost, together with comprehensive ablation experiments and SHAP-based interpretation within a unified evaluation framework.

3. Materials and Methods

The proposed methodology consists of four major stages: data preprocessing, domain-driven feature engineering, model development, and model evaluation. Data preprocessing aims to improve data quality through cleaning, missing value imputation, attribute standardization, and target transformation. Subsequently, a domain-driven feature engineering pipeline is employed to construct informative feature representations prior to model training. The predictive model is then developed using the proposed optimized stacking ensemble learning framework. Finally, model performance is evaluated using both an independent hold-out testing set and 5-fold cross-validation, complemented by comprehensive ablation experiments and SHAP-based interpretation to assess the contribution of individual feature groups and model components. The overall workflow of the proposed framework is illustrated in Fig. 1.

3.1. Dataset

This study employs the Used Car Price Prediction Dataset, a publicly available benchmark dataset published on <https://www.kaggle.com/datasets/taeefnajib/used-car-price-prediction-dataset> [7]. The dataset has been adopted in several previous studies on used car price prediction [1], [2], [7], making it suitable for benchmarking the proposed approach. In addition to providing sufficient observations, the dataset contains a combination of numerical, categorical, and semi-structured attributes, enabling a comprehensive evaluation of domain-driven feature engineering, hyperparameter optimization, and ensemble learning techniques.

Table 1. Description of the attributes used in the used car price prediction dataset.

No.	Attribute	Type	Description
1	brand	Predictor	Vehicle manufacturer (e.g., Ford, BMW, Toyota).
2	model	Predictor	Vehicle model designation.
3	model_year	Predictor	Manufacturing year of the vehicle.
4	mileage	Predictor	Total vehicle mileage (miles).
5	fuel_type	Predictor	Fuel type (e.g., gasoline, diesel, hybrid, or electric).
6	engine	Predictor	Semi-structured engine specification containing horsepower, engine displacement, cylinder configuration, and related information.
7	transmission	Predictor	Transmission type (e.g., automatic, manual, or CVT).
8	ext_col	Predictor	Exterior color of the vehicle.
9	int_col	Predictor	Interior color of the vehicle.
10	accident	Predictor	Vehicle accident history indicator.
11	clean_title	Predictor	Vehicle ownership title status.
12	price	Target	Vehicle selling price (USD).

The dataset comprises 4,009 used vehicle records collected from online marketplaces in the United States. Consequently, the data reflect the characteristics of the U.S. used vehicle market, and the relationships learned by the prediction models should be interpreted within this context. For example, vehicles with higher accumulated mileage generally exhibit lower resale values than comparable vehicles with lower mileage. The original dataset contains 12 attributes, including 11 predictor variables and one target variable. The predictor variables describe vehicle identity, technical specifications, vehicle condition, and ownership information, whereas price represents the target variable expressed in U.S. dollars (USD). A summary of the original dataset attributes is presented in Table 1.

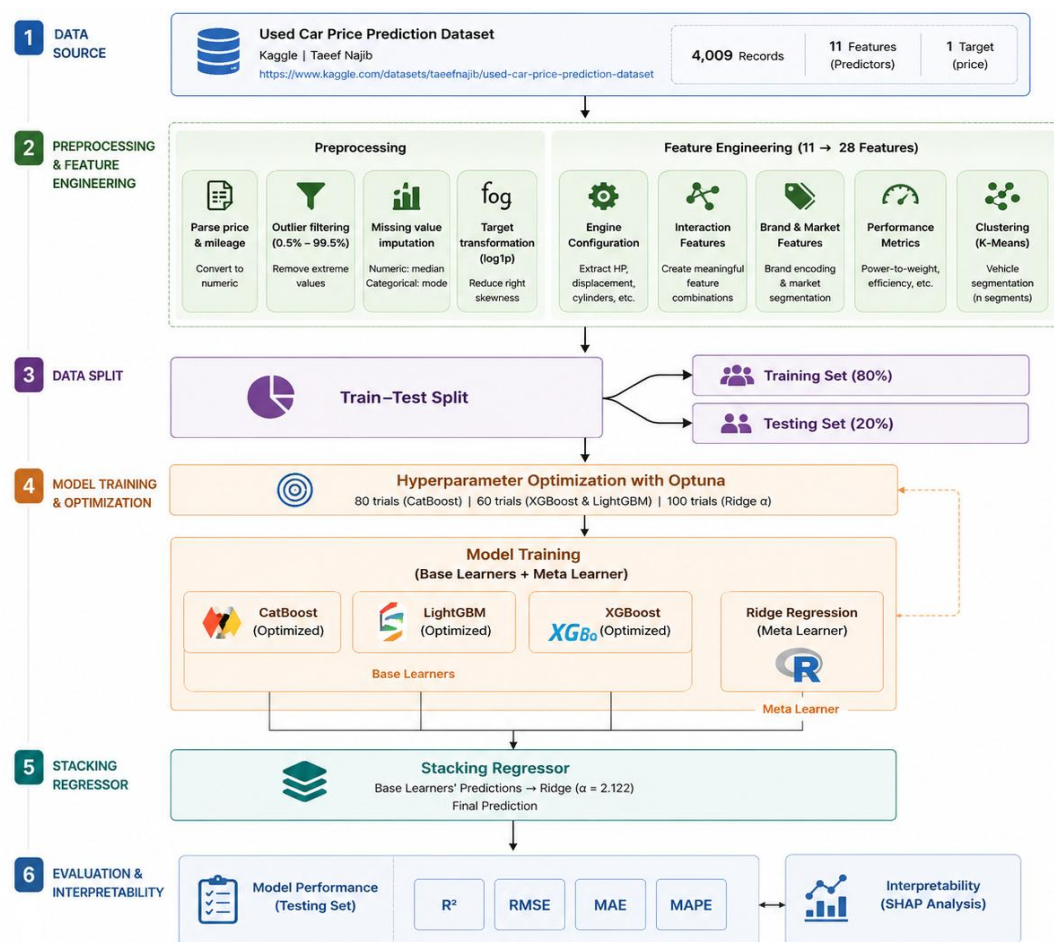


Figure 1. Workflow of the proposed stacking-based framework for used car price prediction.

3.2. Data Preprocessing

Data preprocessing was performed to improve data quality prior to feature engineering and model training. This stage aims to reduce data inconsistencies, handle missing values, mitigate the influence of outliers, and transform the target variable into a representation that is more suitable for gradient boosting-based regression models. To prevent data leakage, all preprocessing statistics were computed exclusively from the training set and subsequently applied to the testing set.

3.2.1. Data Cleaning and Engine Information Extraction

The original dataset contains several attributes stored as formatted text, preventing their direct use by machine learning algorithms. Therefore, the price and mileage attributes were converted into numerical values by removing currency symbols, thousand separators, and measurement units. Invalid entries in the fuel_type, accident, and clean_title attributes, such as empty strings, placeholder symbols (" "), and not supported values, were normalized as missing values for subsequent processing.

The engine attribute contains semi-structured textual descriptions that combine multiple engine specifications within a single field (e.g., "275HP 3.5L V6 Turbo"). Since such representations cannot be directly utilized by regression models, a rule-based parsing procedure was developed to extract structured information. Five engine-related attributes were obtained, namely horsepower, engine displacement, number of cylinders, turbo indicator, and engine configuration (V-Type, Inline, Rotary, or Unknown). To improve data reliability, validation rules were applied by restricting horsepower to the range of 50–2000 HP and engine displacement to 0.5–8.5 L. Values outside these ranges were treated as parsing errors and subsequently regarded as missing values.

3.2.2. Missing Value Treatment

Most missing values were introduced after engine parsing because many vehicle descriptions did not explicitly provide complete engine specifications. To preserve vehicle characteristics while minimizing information loss, missing values in the numerical engine attributes (horsepower, engine displacement, and cylinders) were imputed using a hierarchical imputation strategy.

The imputation process was performed sequentially using (1) the median value of vehicles with the same brand-model combination, (2) the median value of the corresponding brand, and finally (3) the global median when no representative observations were available. This hierarchical strategy was adopted to preserve brand-specific characteristics while maintaining robustness for rare vehicle models. All imputation statistics were estimated exclusively from the training set and then applied to the testing set to eliminate information leakage.

3.2.3. Outlier Detection and Target Transformation

Used car prices typically exhibit a highly right-skewed distribution because a relatively small number of premium vehicles possess substantially higher prices than the majority of observations. Such extreme values may disproportionately influence model training and reduce prediction stability. Therefore, outlier treatment was performed using percentile-based filtering with lower and upper thresholds of 0.5% and 99.5%, respectively, applied to the price attribute.

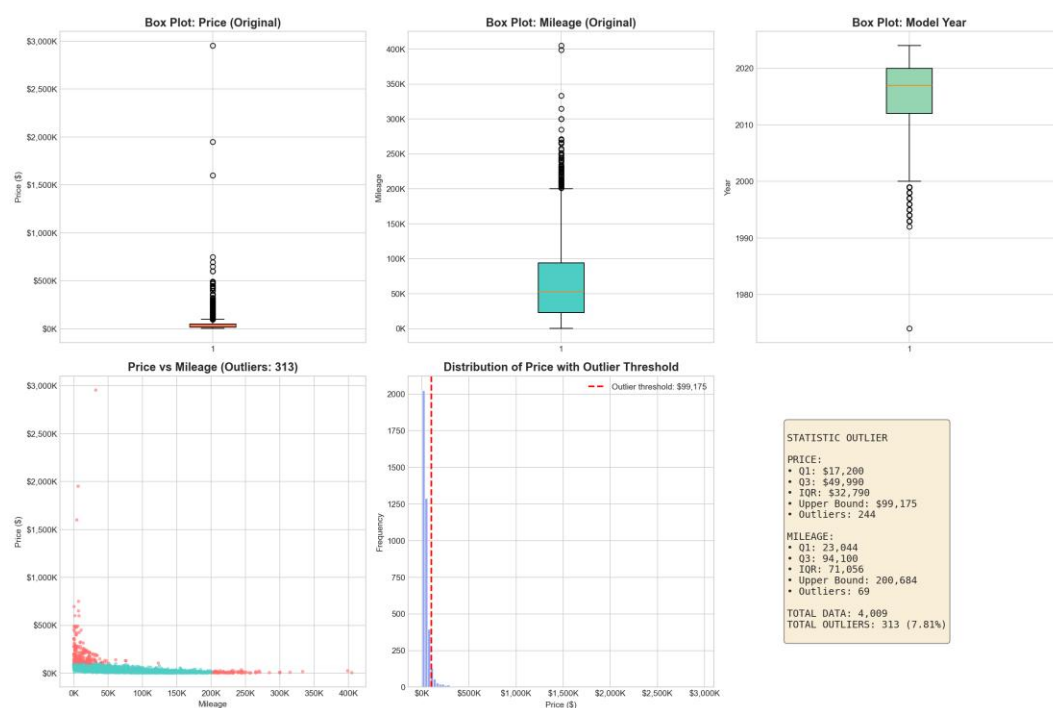


Figure 2. Distribution of the original numerical attributes before preprocessing, illustrating the presence of outliers in price, mileage, and model year.

Percentile-based filtering was selected instead of IQR-based methods because the price distribution exhibits a naturally long-tailed characteristic, where extremely expensive vehicles may still represent valid market observations. The percentile thresholds were estimated solely

from the training set and subsequently applied to the testing set to ensure a fair evaluation procedure. Samples with invalid or non-positive prices were also removed. After preprocessing, the final dataset consisted of 3,176 training samples and 789 testing samples. To further reduce distribution skewness and stabilize variance, the target variable (price) was transformed using the logarithmic $\log_{1p}$ function according to Equation (1):

$$y' = \log(1 + y) \quad (1)$$

where y denotes the original vehicle price and y' represents the transformed value. The inverse transformation was performed using the $\expm1$ function as defined in Equation (2):

$$\hat{y} = \exp(y') - 1 \quad (2)$$

The logarithmic transformation facilitates learning by reducing the influence of extreme prices and producing a target distribution that is more suitable for regression modeling. The distributions of the original numerical attributes before preprocessing are illustrated in Figure 2, highlighting the presence of outliers in price, mileage, and model_year.

3.3. Feature Engineering

Feature engineering was performed to enhance the representational capacity of the original predictor set by constructing features that better capture the underlying relationships between vehicle characteristics and market prices [24]–[26]. Although the original dataset contains essential information such as vehicle age, mileage, engine specifications, and brand identity, these attributes alone may not sufficiently characterize the complex interactions governing vehicle depreciation. In practice, used car prices are determined by the combined influence of multiple factors, including vehicle usage, engine performance, market perception, and ownership characteristics, rather than by individual attributes in isolation. Consequently, relying solely on the original predictors may limit the ability of machine learning models to capture these nonlinear relationships.

Based on this observation, this study adopts a domain-driven feature engineering strategy to transform the original predictor space into more informative representations. Rather than merely increasing the number of variables, the proposed approach aims to construct features that explicitly encode vehicle depreciation, engine efficiency, market characteristics, and latent vehicle segments, thereby providing richer information for subsequent learning algorithms. Following feature engineering, the predictor space was expanded from 11 original predictors to 30 engineered predictors.

3.3.1. Motivation and Overview

To further motivate the proposed feature engineering strategy, Pearson correlation analysis was first conducted using the original predictors after preprocessing. As illustrated in Figure 3, most original attributes exhibit only weak-to-moderate linear correlations with the target variable. Among the original predictors, mileage shows the strongest negative correlation with vehicle price, whereas model_year presents only a moderate positive relationship. Other categorical attributes demonstrate relatively weak linear associations with the target. These observations suggest that the original feature representation alone is insufficient to characterize the complex relationships governing used car prices and motivate the construction of additional domain-specific features capable of capturing nonlinear interactions among vehicle characteristics.

The proposed feature engineering pipeline therefore constructs five complementary categories of features derived from domain knowledge of the automotive market. These categories represent different aspects of vehicle valuation, including depreciation patterns, engine characteristics, market information, and latent customer segments. Collectively, the engineered features are intended to complement the original predictor set by providing representations that are more closely aligned with real-world vehicle valuation mechanisms. A summary of the constructed feature categories is presented in Table 2, while the detailed formulation of each feature group is described in the following subsections.

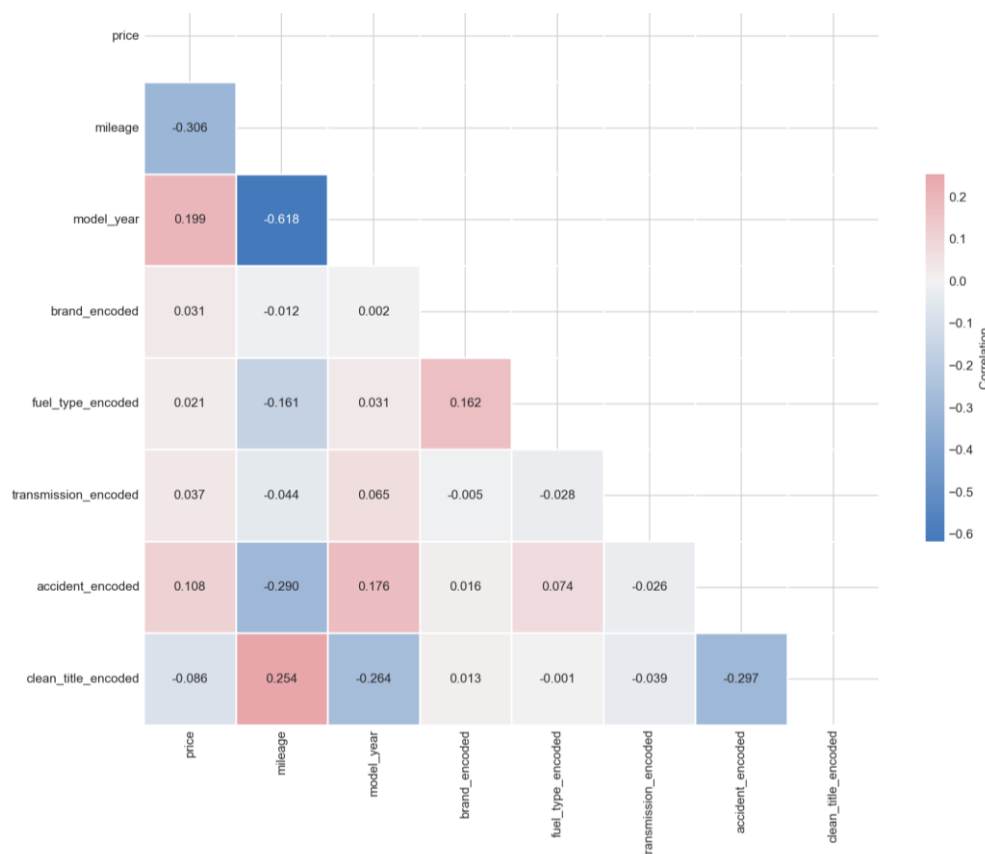


Figure 3. Pearson correlation matrix of the original predictors before feature engineering.

Table 2. Summary of the engineered feature categories.

Category	Number of Features	Features
Interaction Features	5	age_mileage_interaction, age_mileage_product, power_age_interaction, performance_age_interaction, horsepower_squared
Engine Performance	10	horsepower, engine_displacement, cylinders, has_turbo, power_per_liter, power_per_liter_log, power_per_cylinder, displacement_per_cylinder, efficiency_score, power_to_weight_estimate
Engine Configuration	5	engine_config_encoded, engine_config_Unknown, engine_config_Inline, engine_config_V-Type, engine_config_Rotary
Brand and Market	6	brand_frequency, brand_count, model_frequency, model_count, brand_reliability_score, luxury_age_interaction
Clustering	2	cluster, cluster_encoded

3.3.2. Interaction Features

Vehicle depreciation is governed by the combined effects of vehicle age, accumulated mileage, and engine performance rather than by individual attributes alone. To explicitly capture these nonlinear relationships, five interaction features were constructed based on domain knowledge of vehicle depreciation. Vehicle age was first computed as:

$$Age = CurrentYear - ModelYear \quad (3)$$

where *CurrentYear* denotes the year in which the dataset was collected and *ModelYear* represents the manufacturing year of the vehicle.

The interaction between vehicle age and accumulated mileage was computed as:

$$AgeMileageInteraction = \frac{Age \times Mileage}{10000} \quad (4)$$

where the scaling factor of 10,000 was introduced to maintain numerical stability during model training.

To reduce the influence of extreme mileage values while preserving nonlinear relationships, a logarithmic interaction feature was additionally constructed as:

$$AgeMileageProduct = \ln(1 + Age) \times \ln\left(1 + \frac{Mileage}{10000}\right) \quad (5)$$

Engine performance was further combined with vehicle age to represent the gradual depreciation of engine capability over time. The first interaction feature was calculated as:

$$PowerAgeInteraction = \frac{Horsepower}{Age + 1} \quad (6)$$

To account for the combined influence of engine power and engine displacement, a performance-based interaction feature was defined as:

$$PerformanceAgeInteraction = \frac{Horsepower \times EngineDisplacement}{Age + 1} \quad (7)$$

where larger engines with higher power output are assumed to retain greater market value than smaller engines with comparable horsepower, while the denominator models the depreciation associated with increasing vehicle age.

Finally, a quadratic horsepower feature was introduced to capture potential nonlinear effects of engine power on vehicle price, particularly for premium and high-performance vehicles:

$$HorsepowerSquared = Horsepower^2 \quad (8)$$

Collectively, these interaction features provide explicit representations of vehicle depreciation by modeling the joint effects of age, accumulated usage, and engine performance, thereby complementing the original predictor variables.

3.3.3. Engine-, Brand-, and Market-Based Features

Beyond vehicle depreciation, resale value is also influenced by engine capability, manufacturer reputation, and market perception. To capture these complementary aspects, additional features were constructed from engine specifications together with brand- and market-related information.

- **Engine Performance Features**

Although the original dataset provides engine specifications such as horsepower, engine displacement, and cylinder count, these variables describe engine characteristics individually and may not adequately represent overall engine performance. Therefore, several derived indicators were constructed to capture normalized engine characteristics. Engine power generated per unit of displacement was calculated as:

$$PowerPerLiter = \frac{Horsepower}{EngineDisplacement} \quad (9)$$

To reduce distribution skewness while preserving relative differences among vehicles, a logarithmic transformation was subsequently applied:

$$PowerPerLiterLog = \ln(1 + PowerPerLiter) \quad (10)$$

Engine output per cylinder was calculated as:

$$PowerPerCylinder = \frac{Horsepower}{Cylinders} \quad (11)$$

while the average displacement per cylinder was computed as:

$$DisplacementPerCylinder = \frac{EngineDisplacement}{Cylinders} \quad (12)$$

An engine efficiency indicator was further defined to measure the relationship between engine power, displacement, and cylinder count:

$$EfficiencyScore = \frac{Horsepower}{EngineDisplacement \times Cylinders} \quad (13)$$

Because vehicle weight is unavailable in the dataset, a simplified power-to-weight approximation was estimated using horsepower:

$$PowerToWeightEstimate = \frac{Horsepower}{\left(\frac{Horsepower \times 30}{2000}\right) + 3} \quad (14)$$

Although this formulation does not represent the actual physical power-to-weight ratio, it serves as a surrogate indicator for distinguishing vehicles with relatively higher performance characteristics.

- **Brand and Market Features**

Market perception was represented through categorical information describing vehicle brands and models. Brand and model attributes were transformed using both frequency encoding and count encoding, producing the features `brand_frequency`, `brand_count`, `model_frequency`, and `model_count`. Frequency encoding captures the relative occurrence of each category within the training data, whereas count encoding preserves its absolute frequency. These representations are intended to characterize market prevalence, which may influence pricing behavior.

Two additional domain-specific features were also introduced. The first, `brand_reliability_score`, assigns higher scores to manufacturers widely recognized for vehicle reliability, including Toyota, Honda, Mazda, Lexus, Acura, and Subaru, while the remaining manufacturers receive lower scores. This feature reflects the assumption that vehicles produced by manufacturers with stronger reliability reputations generally retain higher resale values.

The second feature, `luxury_age_interaction`, models the interaction between vehicle age and premium-brand ownership to account for differences in depreciation patterns between luxury and non-luxury vehicles:

$$LuxuryAgeInteraction = IsLuxury \times Age \quad (15)$$

where *IsLuxury* equals 1 for premium manufacturers (BMW, Mercedes-Benz, Audi, Lexus, Porsche, Jaguar, Land Rover, Cadillac, Tesla, Maserati, Bentley, Rolls-Royce, Ferrari, Lamborghini, and Aston Martin) and 0 otherwise.

Finally, the extracted engine configuration was represented using two complementary encoding strategies. Ordinal encoding was applied to preserve the ordered representation of engine layouts (Unknown = 0, Inline = 1, V-Type = 2, Rotary = 3), producing the feature `engine_config_encoded`. In addition, one-hot encoding generated four binary variables (`engine_config_Unknown`, `engine_config_Inline`, `engine_config_V-Type`, and `engine_config_Rotary`) to allow the learning algorithms to model configuration-specific effects without assuming ordinal relationships among engine layouts. Collectively, these engine-, brand-, and market-based features provide complementary information on technical performance and consumer perception that is not explicitly represented by the original predictor variables.

3.3.4. Clustering-Based Features

Vehicle valuation is often influenced by latent market segments that cannot be adequately represented using individual vehicle attributes alone. To capture these hidden structures, K-Means clustering was performed using horsepower, engine displacement, mileage, and vehicle age after feature standardization. The resulting clusters group vehicles with similar technical characteristics and usage patterns, thereby providing an additional representation of market segmentation. To transform the cluster assignments into predictive features, target encoding with smoothing was applied according to Equation (16):

$$ClusterEncoded = \frac{\bar{p}_c \times n_c + m \times \bar{p}_{global}}{n_c + m} \quad (16)$$

where $\bar{p}_c$ denotes the mean transformed price ($\ln(1 + price)$) of cluster c , n_c represents the number of observations belonging to cluster c , $\bar{p}_{global}$ is the global mean transformed

price computed from the training set, and m is the smoothing parameter, which was empirically set to 10.

The smoothing mechanism reduces the influence of clusters containing relatively few observations by shrinking their encoded values toward the global mean. Consequently, the resulting feature preserves cluster-level pricing information while reducing the risk of overfitting. Together with the interaction-, engine-, and market-based features, the clustering-based representation complements the original predictor space by incorporating latent market characteristics that are not explicitly captured by individual vehicle attributes.

3.3.5. Feature Representation Analysis

To assess the effectiveness of the proposed feature engineering strategy, Pearson correlation analysis was performed using the engineered predictors. As illustrated in Figure 4, several engineered features exhibit substantially stronger correlations with the target variable than the original predictors. In particular, `age_mileage_interaction`, `age_mileage_product`, and `cluster_encoded` demonstrate considerably stronger associations with vehicle price, indicating that the engineered features successfully capture relationships that are not explicitly represented by the original attributes.

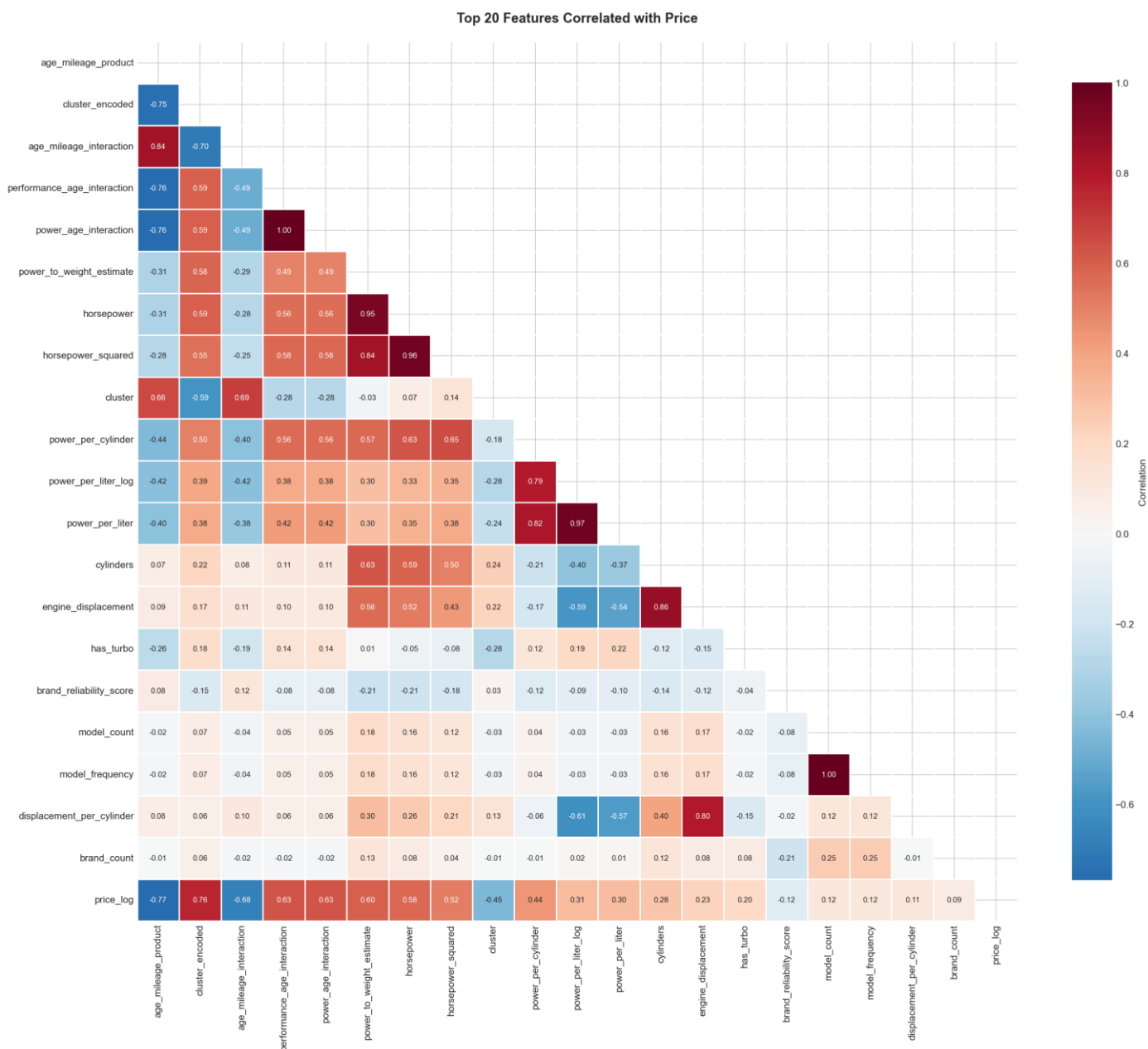


Figure 4. Pearson correlation matrix of the engineered predictors after feature engineering.

Compared with the correlation matrix shown in Figure 3, the engineered feature space provides richer predictive information by incorporating interactions among vehicle age, accumulated mileage, engine characteristics, and market segmentation. These observations support the hypothesis that domain-driven feature engineering produces a more informative representation for used car price prediction.

Nevertheless, correlation analysis alone does not necessarily reflect predictive importance. Features exhibiting relatively weak linear correlations may still contribute substantially to nonlinear learning algorithms, whereas highly correlated features may become redundant after model optimization. Therefore, the contribution of each engineered feature group is further evaluated through the comprehensive ablation study and SHAP-based interpretation presented in Section 4.

3.4. Experimental Setup

This section describes the experimental configuration adopted to develop and evaluate the proposed used car price prediction framework. The experimental setup consists of three major components: data partitioning, stacking ensemble architecture, and hyperparameter optimization. The overall configuration was designed to ensure reproducibility while minimizing information leakage throughout the learning process.

3.4.1 Data Partitioning

The dataset was randomly divided into training and testing subsets using an 80:20 hold-out strategy implemented through the `train_test_split` function from the Scikit-learn library. Data partitioning was performed before any preprocessing operation to ensure that all preprocessing statistics—including missing-value imputation, outlier filtering, feature encoding, target transformation, and feature engineering—were estimated exclusively from the training data and subsequently applied to the testing data. This strategy prevents information leakage and avoids overly optimistic performance estimation.

A fixed random seed (`random_state = 42`) was adopted to ensure reproducibility. The initial partition generated 3,207 training samples and 802 testing samples. After applying the preprocessing pipeline described in Section 3.2, including percentile-based outlier filtering on the target variable, the final dataset consisted of 3,176 training samples and 789 testing samples.

Although the final predictive performance was evaluated using the hold-out testing set, model selection and hyperparameter optimization were conducted using 5-fold cross-validation within the training set. This evaluation protocol provides a more reliable estimate of model generalization while reducing model-selection bias during hyperparameter optimization.

3.4.2 Proposed Stacking Ensemble

The proposed framework employs a two-level stacking ensemble consisting of three gradient boosting models as base learners and Ridge Regression as the meta-learner, as illustrated in Figure 1. The three base learners—CatBoost, LightGBM, and XGBoost—were selected because they employ different gradient boosting strategies, enabling the ensemble to exploit complementary learning characteristics. The underlying hypothesis is that combining these heterogeneous learners produces more robust and accurate predictions than relying on any individual model. The overall configuration of the proposed stacking ensemble is summarized in Table 3.

Predictions generated by the three base learners are subsequently integrated using Ridge Regression, which serves as the second-level learner. Ridge Regression was selected because predictions produced by multiple gradient boosting models are typically highly correlated. By incorporating L2 regularization, the meta-learner stabilizes the regression coefficients and prevents the final prediction from relying excessively on a single base learner, thereby improving generalization performance [27], [28]. During stacking, the meta-learner receives only the out-of-fold (OOF) predictions generated through 5-fold cross-validation (`passthrough=False`) rather than the original predictor variables. This configuration enables the meta-learner to learn the optimal combination of the three base learners while minimizing the risk of overfitting.

Table 3. Configuration of the proposed stacking ensemble framework

Component	Configuration	Rationale
Ensemble architecture	Two-level stacking regressor	Hierarchically combines multiple predictive models through a meta-learner [4], [6]
Base learners	CatBoost, LightGBM, XGBoost	Complementary gradient boosting algorithms with different learning mechanisms [3], [6]
Meta learner	Ridge Regression	Combines correlated base learner predictions using L2 regularization
Cross-validation	5-fold	Generates robust out-of-fold predictions for meta-learner training
Meta-learner input	Out-of-fold predictions	Prevents information leakage between learning stages
Passthrough	False	Restricts the meta-learner to learning only from base learner predictions
Hyperparameter optimization	Optuna (TPE sampler)	Efficient Bayesian optimization of model parameters [3]
Random seed	42	Ensures reproducibility

3.4.3. Hyperparameter Optimization

The predictive performance of gradient boosting models is highly dependent on hyperparameter configuration. Conventional optimization approaches, such as Grid Search, become computationally inefficient as the number of tunable parameters increases, whereas Random Search explores the search space without utilizing information obtained from previous evaluations. Therefore, this study employs Optuna [3] to perform automatic hyperparameter optimization. Optuna uses the Tree-structured Parzen Estimator (TPE), a Bayesian optimization algorithm that iteratively constructs probabilistic models of promising and non-promising hyperparameter configurations based on the outcomes of previous trials [29]. Compared with exhaustive search methods, TPE adaptively focuses the search on promising regions of the parameter space, thereby improving optimization efficiency while substantially reducing computational cost [30].

Hyperparameter optimization was performed independently for each base learner using 5-fold cross-validation, where the mean coefficient of determination (R^2) served as the optimization objective. All optimization experiments employed TPESampler with a fixed random seed (seed = 42) to ensure reproducibility. The hyperparameter search spaces were determined according to the characteristics of each learning algorithm and are summarized in Tables 4. Continuous parameters were sampled using uniform or logarithmic distributions, whereas discrete parameters were sampled from predefined integer ranges.

3.4.4. Performance Evaluation and Model Interpretation

The predictive performance of the proposed framework was evaluated using both a hold-out testing set and 5-fold cross-validation on the training data. The hold-out evaluation measures the generalization capability of the final optimized model on previously unseen data, whereas cross-validation provides a more robust estimate of model stability during training and hyperparameter optimization. The combination of these two evaluation strategies reduces the likelihood of drawing overly optimistic conclusions from a single train-test partition.

Four complementary performance metrics were employed to assess prediction accuracy, namely the coefficient of determination (R^2), Root Mean Squared Error (RMSE), Mean Absolute Error (MAE), and Mean Absolute Percentage Error (MAPE). These metrics provide complementary perspectives on model performance by simultaneously measuring goodness-of-fit, absolute prediction error, and relative prediction error. The coefficient of determination (R^2) is defined as:

$$R^2 = 1 - \frac{\sum_{i=1}^n (y_i - \hat{y}_i)^2}{\sum_{i=1}^n (y_i - \bar{y})^2} \quad (17)$$

where y_i denotes the observed vehicle price, $\hat{y}_i$ represents the predicted value, and $\bar{y}_i$ is the mean of the observed prices.

Table 4. Hyperparameter search space used for Optuna optimization of the base learners.

Hyperparameter	CatBoost	LightGBM	XGBoost
Optimization trials	60	60	60
Objective function	R ² (5-fold CV)	R ² (5-fold CV)	R ² (5-fold CV)
Optimizer	TPE	TPE	TPE
Estimators / Iterations	1200–2500	800–2500	800–2500
Learning rate	0.01–0.05	0.005–0.10	0.005–0.10
Maximum tree depth	6–10	5–12	5–12
Subsample	0.70–1.00	0.60–0.95	0.60–0.95
Feature sampling	–	0.60–0.95 (colsam- ple_bytree)	0.60–0.95 (colsam- ple_bytree)
L1 regularization	–	1e–8–10 (reg_alpha)	1e–8–10 (reg_alpha)
L2 regularization	1–10 (l2_leaf_reg)	1e–8–10 (reg_lambda)	1e–8–10 (reg_lambda)
Minimum samples / child constraint	1–20 (min_data_in_leaf)	5–50 (min_child_sam- ples)	1–10 (min_child_weight)
Additional parameter 1	0–1 (bagging_tempera- ture)	20–200 (num_leaves)	1e–8–1 (gamma)
Additional parameter 2	0.5–5 (random_strength)	–	–
Additional parameter 3	128–255 (border_count)	–	–
Tree construction	Ordered boosting	Leaf-wise growth	Histogram-based tree (tree_method='hist')
Random seed	42	42	42

Note: All hyperparameters were optimized independently using Optuna with the Tree-structured Parzen Estimator (TPE) sampler. Continuous parameters were sampled using either uniform or log-uniform distributions according to their respective search spaces, whereas integer-valued parameters were sampled from discrete ranges. Parameters marked with "–" are not applicable to the corresponding learning algorithm.

The RMSE is computed as:

$$RMSE = \sqrt{\frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2} \quad (18)$$

which measures the average prediction error while assigning greater penalties to larger errors.

The MAE is calculated as:

$$MAE = \frac{1}{n} \sum_{i=1}^n |y_i - \hat{y}_i| \quad (19)$$

providing a more robust estimate of prediction error because it is less sensitive to extreme observations than RMSE.

Finally, the MAPE is defined as:

$$MAPE = \frac{100}{n} \sum_{i=1}^n \left| \frac{y_i - \hat{y}_i}{y_i} \right| \quad (20)$$

which expresses prediction error as a percentage of the observed vehicle price, thereby facilitating practical interpretation.

Although predictive accuracy is the primary objective of this study, understanding the contribution of individual features is equally important. Therefore, model interpretability was investigated using SHapley Additive exPlanations (SHAP) [31]–[33], a game-theoretic approach that quantifies the contribution of each feature to the final prediction.

The SHAP value of feature j is computed as:

$$\phi_j = \sum_{S \subseteq \{j\}} \frac{|S|! (n - |S| - 1)!}{n!} [f(S \cup \{j\}) - f(S)] \quad (21)$$

where S denotes a subset of predictor variables, N represents the complete feature set, and $f(\cdot)$ is the prediction function. Because the proposed framework is a stacking ensemble that combines multiple regression models, the KernelExplainer implementation was adopted to provide model-agnostic explanations. To reduce computational cost while maintaining representative explanations, SHAP values were computed using a randomly selected subset of 100 testing samples. The SHAP analysis complements the quantitative evaluation by identifying the relative importance of engineered features and explaining how individual predictors influence vehicle price estimation.

4. Results and Discussion

All experiments were conducted using Python 3.10 in the Jupyter Notebook environment. The proposed framework was implemented using Scikit-learn 1.3.0 for data preprocessing and model evaluation, CatBoost 1.2.2, LightGBM 4.1.0, and XGBoost 2.0.0 as the gradient boosting libraries, Optuna 3.4.0 for Bayesian hyperparameter optimization, and SHAP 0.48.0 for model interpretation. The experiments were executed on an Acer Swift SFX14-41G equipped with an AMD Ryzen 5 5600U processor, 16 GB RAM, and an NVIDIA GeForce RTX 3050 Laptop GPU running Windows 11.

The experiments were performed using the Used Car Price Prediction dataset described in Section 3.1. After the preprocessing pipeline, the final dataset consisted of 3,176 training samples and 789 testing samples. To provide a reliable assessment of model generalization, all experimental results are reported using two complementary evaluation protocols. The first is a hold-out testing set, which measures the predictive performance of the final optimized model on unseen data. The second is 5-fold cross-validation, which serves as the primary estimator of model robustness by reporting both the mean performance and its standard deviation. Combining these two evaluation protocols enables cross-validation of the experimental findings and reduces the possibility that the reported performance is influenced by a particular data partition.

This section first compares the predictive performance of the proposed stacking framework with the individual gradient boosting models. Subsequently, a comprehensive ablation study is presented to quantify the contribution of feature engineering, feature groups, stacking components, hyperparameter optimization, and meta-learner selection. Finally, SHAP analysis is employed to interpret the learned prediction mechanism and identify the most influential factors affecting used car prices.

4.1. Overall Prediction Performance

Table 5 presents the predictive performance of all evaluated models, including the three individual gradient boosting algorithms (CatBoost, LightGBM, and XGBoost) and the proposed Stacking Regressor. Each model was evaluated using both its baseline configuration and the Optuna-optimized configuration. Across both the hold-out test set and 5-fold cross-validation, hyperparameter optimization consistently improved predictive performance, yielding higher R^2 values and lower prediction errors than the corresponding baseline models. Among the individual learners, CatBoost achieved the best performance, obtaining an R^2 of 0.9021 ± 0.0021 under cross-validation and 0.9104 on the hold-out test set, suggesting that it effectively models the heterogeneous characteristics of the used car dataset.

The proposed Stacking Regressor further improved prediction performance by combining the complementary strengths of the three optimized gradient boosting models. It achieved the highest overall performance, reaching an R^2 of 0.9023 ± 0.0025 under 5-fold cross-validation and 0.9112 on the hold-out test set. Although the improvement over the optimized CatBoost model was modest, the consistent results across both evaluation protocols and the relatively small cross-validation standard deviation indicate that the proposed ensemble provides stable generalization while effectively exploiting complementary prediction patterns among the base learners.

Table 5. Comprehensive performance comparison of baseline and Optuna-optimized models on the hold-out test set and 5-fold cross-validation.

Model	Type	Test Set (20% hold-out)				5-Fold Cross-Validation			
		R ²	RMSE	MAE	MAPE	R ²	RMSE	MAE	MAPE
CatBoost	Baseline	0.9091	0.2414	0.1707	17.91%	0.9011	0.2529	0.1749	17.79%
CatBoost	Optuna	0.9104	0.2397	0.1689	17.82%	0.9021	0.2517	0.1736	17.66%
XGBoost	Baseline	0.9056	0.2461	0.1732	18.14%	0.8894	0.2674	0.1835	18.79%
XGBoost	Optuna	0.9089	0.2417	0.1686	17.91%	0.8991	0.2554	0.1765	18.00%
LightGBM	Baseline	0.9035	0.2487	0.1770	18.72%	0.8910	0.2655	0.1843	18.84%
LightGBM	Optuna	0.9052	0.2465	0.1748	18.54%	0.8939	0.2619	0.1825	18.66%
Stacking	Baseline	0.9096	0.2407	0.1719	18.10%	0.9004	0.2537	0.1764	17.96%
Stacking	Optuna	0.9112	0.2387	0.1678	17.77%	0.9023	0.2514	0.1731	17.63%

The benefit of Optuna is particularly evident for XGBoost, whose cross-validation performance increases from 0.8894 to 0.8991, representing the largest relative improvement among the evaluated models. In contrast, CatBoost exhibits only a marginal improvement after optimization, implying that its default configuration is already close to an optimal solution for this dataset. These observations indicate that the impact of hyperparameter optimization depends on the intrinsic characteristics of each learning algorithm rather than being uniformly distributed across models.

The superior performance of the proposed stacking framework can be attributed to the diversity of its constituent learners. Although each gradient boosting algorithm individually achieves competitive prediction accuracy, they exhibit different prediction biases and error distributions. By learning an optimal linear combination of their out-of-fold predictions, the Ridge Regression meta-learner effectively compensates for individual model errors, resulting in consistently lower prediction errors than those of any single base learner. To further evaluate the stability of the proposed framework, Table 6 summarizes the performance obtained in each fold of the 5-fold cross-validation procedure. The fold-wise evaluation demonstrates a highly consistent predictive performance across all five folds, with R² values ranging from 0.8993 to 0.9056 and a standard deviation of only 0.0025. Similarly, the variation in MAPE remains below 0.5%, indicating that the proposed framework is not sensitive to variations in the training-validation partition. These findings confirm that the performance reported on the hold-out testing set is representative of the overall generalization capability of the proposed model rather than the result of a favorable data split.

Table 6. Fold-wise performance of the optimized stacking ensemble under 5-fold cross-validation.

Fold	R ²	RMSE	MAE	MAPE (%)
Fold 1	0.9007	0.2538	0.1747	17.53
Fold 2	0.9056	0.2465	0.1731	18.35
Fold 3	0.8993	0.2560	0.1756	17.80
Fold 4	0.9007	0.2549	0.1743	17.58
Fold 5	0.9050	0.2458	0.1677	16.91
Mean $\pm$ Std	0.9023 $\pm$ 0.0025	0.2514 $\pm$ 0.0044	0.1731 $\pm$ 0.0028	17.63 $\pm$ 0.47

4.2. Ablation Studies

4.2.1. Contribution of Feature Engineering

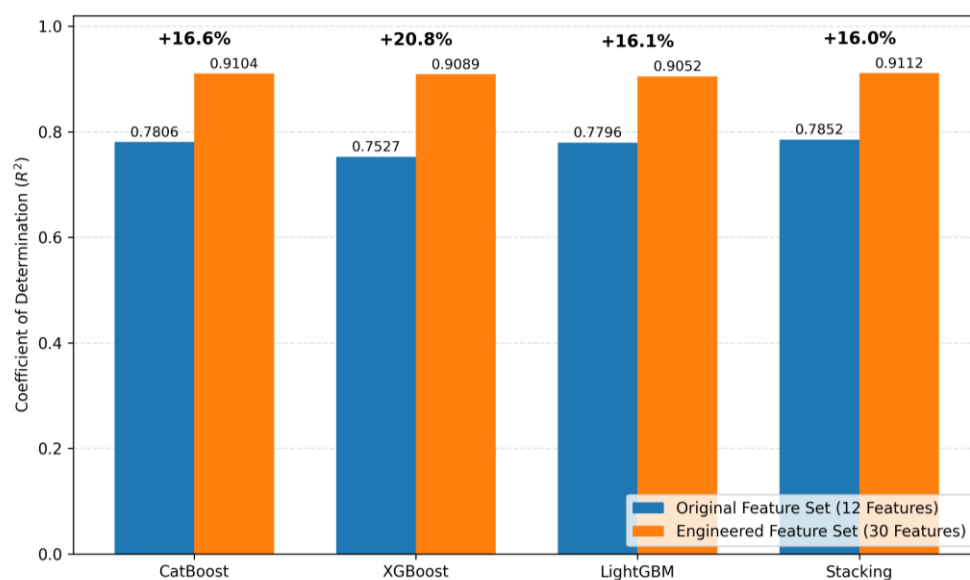
Feature engineering represents one of the primary contributions of the proposed framework. To evaluate its effectiveness, two complementary analyses were conducted. First, the predictive performance obtained using the original predictor set was compared with that obtained after applying the proposed feature engineering pipeline. Second, an ablation study was performed by sequentially removing individual feature groups to quantify their relative contributions to the final prediction model. Table 7 summarizes the predictive performance obtained before and after feature engineering for all evaluated learning algorithms.

Table 7. Performance comparison before and after applying the proposed feature engineering strategy.

Model	Feature Set	R^2	RMSE (log)	MAE (log)	MAPE
CatBoost	Original (12 features)	0.7806	0.3750	0.2843	31.71%
	Engineered (30 features)	0.9104	0.2397	0.1689	17.82%
XGBoost	Original (12 features)	0.7527	0.3982	0.3013	34.16%
	Engineered (30 features)	0.9089	0.2417	0.1686	17.91%
LightGBM	Original (12 features)	0.7796	0.3759	0.2795	31.55%
	Engineered (30 features)	0.9052	0.2465	0.1748	18.54%
Stacking	Original (12 features)	0.7852	0.3711	0.2792	31.29%
	Engineered (30 features)	0.9112	0.2387	0.1678	17.77%

The engineered feature set consistently improves the predictive performance of every regression model. Compared with the original predictor set, the coefficient of determination (R^2) increases by approximately 16–21%, while the prediction error is substantially reduced across all evaluation metrics. The largest improvement is observed for XGBoost, whose R^2 increases from 0.7527 to 0.9089, whereas the proposed stacking framework achieves the highest overall prediction accuracy ($R^2 = 0.9112$). These results indicate that improving feature representation contributes more substantially to predictive performance than model selection alone because the predictive limitations of the original dataset arise primarily from insufficient feature representation rather than inadequate learning capacity. By explicitly modeling vehicle depreciation, engine efficiency, and market-related characteristics, the proposed feature engineering pipeline exposes informative relationships that are not directly accessible from the original attributes. Consequently, the gradient boosting models can exploit richer predictive information before hyperparameter optimization is applied, explaining why improvements in feature representation yield larger performance gains than increasing model complexity alone.

The overall improvement achieved by the proposed feature engineering strategy across different regression models is illustrated in Figure 6. Although the magnitude of improvement varies among the evaluated algorithms, a consistent increase in predictive performance is observed, indicating that the engineered features provide informative representations that are broadly beneficial rather than being tailored to a particular learning algorithm. To further quantify the contribution of individual feature groups, a feature group ablation study was performed by removing one feature category at a time while keeping the remaining feature groups unchanged. The corresponding results are summarized in Table 8.

**Figure 6.** Comparison of predictive performance before and after applying the proposed feature engineering strategy across different regression models.

The ablation results reveal that Interaction Features constitute the most influential component of the proposed feature engineering pipeline. Removing this feature group decreases the cross-validation R^2 from 0.9023 to 0.8453, corresponding to a performance reduction of approximately 5.7%, which is substantially larger than that observed for any other feature category. The Brand & Market and Engine Performance feature groups represent the second and third largest contributors, reducing prediction performance by 2.29% and 2.06%, respectively. These results indicate that market-related characteristics and engine-derived descriptors provide complementary information that is not captured by the conventional vehicle attributes alone. By contrast, the contribution of Engine Configuration is relatively limited, while removing the Clustering features produces a marginal improvement that remains within the cross-validation standard deviation.

Table 8. Feature group ablation study of the optimized stacking framework using 5-fold cross-validation.

Ablation Setting	Remaining Features	CV $R^2 \pm \text{Std}$	ΔR^2	CV RMSE
Full feature set	30	0.9023 $\pm$ 0.0025	—	0.2514
w/o Interaction Features	25	0.8453 $\pm$ 0.0091	−5.70%	0.3162
w/o Engine Performance	20	0.8816 $\pm$ 0.0058	−2.06%	0.2766
w/o Brand & Market	24	0.8794 $\pm$ 0.0050	−2.29%	0.2792
w/o Engine Configuration	25	0.9004 $\pm$ 0.0033	−0.18%	0.2537
w/o Clustering	28	0.9031 $\pm$ 0.0023	+0.08%	0.2503

This observation suggests that the latent vehicle segments intended to be captured by clustering are already adequately represented through the interaction-based and market-oriented features. Consequently, introducing additional clustering variables provides little additional predictive information. Overall, the ablation study demonstrates that the effectiveness of the proposed feature engineering strategy is primarily driven by interaction-based features, followed by market-oriented and engine-derived representations. Increasing the number of engineered variables alone does not necessarily improve prediction performance; rather, meaningful improvements are achieved by constructing features that explicitly represent the underlying mechanisms governing vehicle depreciation.

4.2.2. Contribution of the Stacking Architecture

The proposed framework combines three gradient boosting algorithms through a stacking ensemble. To quantify the contribution of each base learner, an ablation study was conducted by sequentially removing one learner while maintaining the remaining experimental configuration unchanged. In addition, the predictive performance of each individual learner was evaluated independently. All experiments were performed using the engineered feature set under 5-fold cross-validation. Table 9 summarizes the results of the stacking component ablation.

The results indicate that CatBoost constitutes the dominant contributor to the proposed ensemble. Removing CatBoost decreases the cross-validation performance from 0.9023 to 0.8987, representing the largest performance degradation among all base learner removals. By contrast, excluding either LightGBM or XGBoost produces only marginal changes, indicating that the majority of the predictive capability is already captured by CatBoost.

Table 9. Contribution of individual base learners to the performance of the proposed stacking ensemble under 5-fold cross-validation.

Configuration	Base Learners	CV $R^2 \pm \text{Std}$	ΔR^2
Full Stacking	3	0.9023 $\pm$ 0.0025	—
Without LightGBM	2	0.9023 $\pm$ 0.0024	+0.00%
Without XGBoost	2	0.9022 $\pm$ 0.0026	−0.01%
Without CatBoost	2	0.8987 $\pm$ 0.0035	−0.36%
CatBoost only	1	0.9021 $\pm$ 0.0021	−0.02%
XGBoost only	1	0.8991 $\pm$ 0.0029	−0.32%
LightGBM only	1	0.8939 $\pm$ 0.0054	−0.84%

Although the complete stacking framework achieves the highest overall performance, the performance difference between the optimized stacking model ($R^2 = 0.9023$) and the optimized CatBoost model ($R^2 = 0.9021$) is relatively small. This observation suggests that the principal advantage of the stacking framework is not a substantial increase in prediction accuracy but rather a modest and consistent improvement in robustness obtained by integrating complementary prediction patterns from multiple learners.

Interestingly, the optimized CatBoost model alone achieves performance comparable to that of the complete ensemble. This finding indicates that the quality of the engineered feature representation plays a more important role than increasing ensemble complexity. Nevertheless, the stacking framework consistently produces the best overall results across both the hold-out testing set and cross-validation, supporting its selection as the final predictive model.

4.2.3. Meta-Learner Analysis

The choice of meta-learner influences how predictions generated by the base learners are integrated into the final prediction. To evaluate this effect, several linear and nonlinear meta-learners were compared while maintaining identical base learners and optimized hyperparameters. Table 10 presents the comparison of different meta-learning strategies.

Table 10. Performance comparison of alternative meta-learners for the proposed stacking framework.

Meta Learner	Test R^2	CV $R^2 \pm \text{Std}$	CV RMSE	CV MAPE
Ridge (Optuna-tuned, $\alpha = 0.603$)	0.9112	0.9023 ± 0.0025	0.2514	17.63%
Ridge ($\alpha = 1.0$)	0.9112	0.9023 ± 0.0026	0.2514	17.63%
Ridge ($\alpha = 2.0$)	0.9114	0.9023 ± 0.0026	0.2514	17.63%
Huber Regressor	0.9114	0.9021 ± 0.0024	0.2516	17.54%
Bayesian Ridge	0.9111	0.9023 ± 0.0025	0.2514	17.63%

Overall, the evaluated meta-learners exhibit relatively similar predictive performance, indicating that the quality of the base learners contributes more substantially to the final prediction than the choice of the second-level learner. Nevertheless, Ridge Regression consistently achieves the highest cross-validation performance while maintaining the lowest prediction variance.

The superior performance of Ridge Regression can be attributed to the strong correlation among predictions generated by the three gradient boosting models. By incorporating L2 regularization, Ridge Regression effectively stabilizes the regression coefficients and reduces the risk of overfitting without introducing unnecessary model complexity. Consequently, Ridge Regression provides an appropriate balance between predictive accuracy, computational efficiency, and model stability, making it well suited as the meta-learner of the proposed stacking framework.

4.3. SHAP-Based Model Interpretation

To complement the quantitative evaluation, SHAP analysis was performed to investigate how the proposed stacking framework utilizes the engineered features during prediction. The optimized stacking model was interpreted using Kernel SHAP, where 100 background samples and 100 testing instances were employed to estimate feature contributions. The resulting SHAP values provide both global feature importance and local feature effects, allowing the prediction mechanism of the proposed model to be interpreted beyond conventional performance metrics. Figure 7 (a) presents the global feature importance based on the mean absolute SHAP values, whereas Figure 7 (b) illustrates the corresponding SHAP beeswarm plot, which summarizes the distribution of feature contributions across all testing samples.

The SHAP analysis demonstrates that the proposed framework relies predominantly on the engineered interaction features rather than the original vehicle attributes. The two most influential predictors, `age_mileage_interaction` and `age_mileage_product`, contribute substantially more than all remaining variables, indicating that the interaction between vehicle age and accumulated mileage represents the dominant factor governing used vehicle depreciation. This finding is consistent with the feature group ablation presented in Section 4.2, where removing the interaction features resulted in the largest reduction in predictive performance.

Beyond the interaction features, several engine-derived variables—including power_to_weight_estimate, horsepower, engine_displacement, and displacement_per_cylinder—also exhibit relatively high importance, suggesting that vehicle performance remains a key determinant of resale value. Likewise, the relatively high importance of cluster_encoded indicates that the clustering procedure successfully captures latent market characteristics that are not fully represented by the original vehicle descriptors.

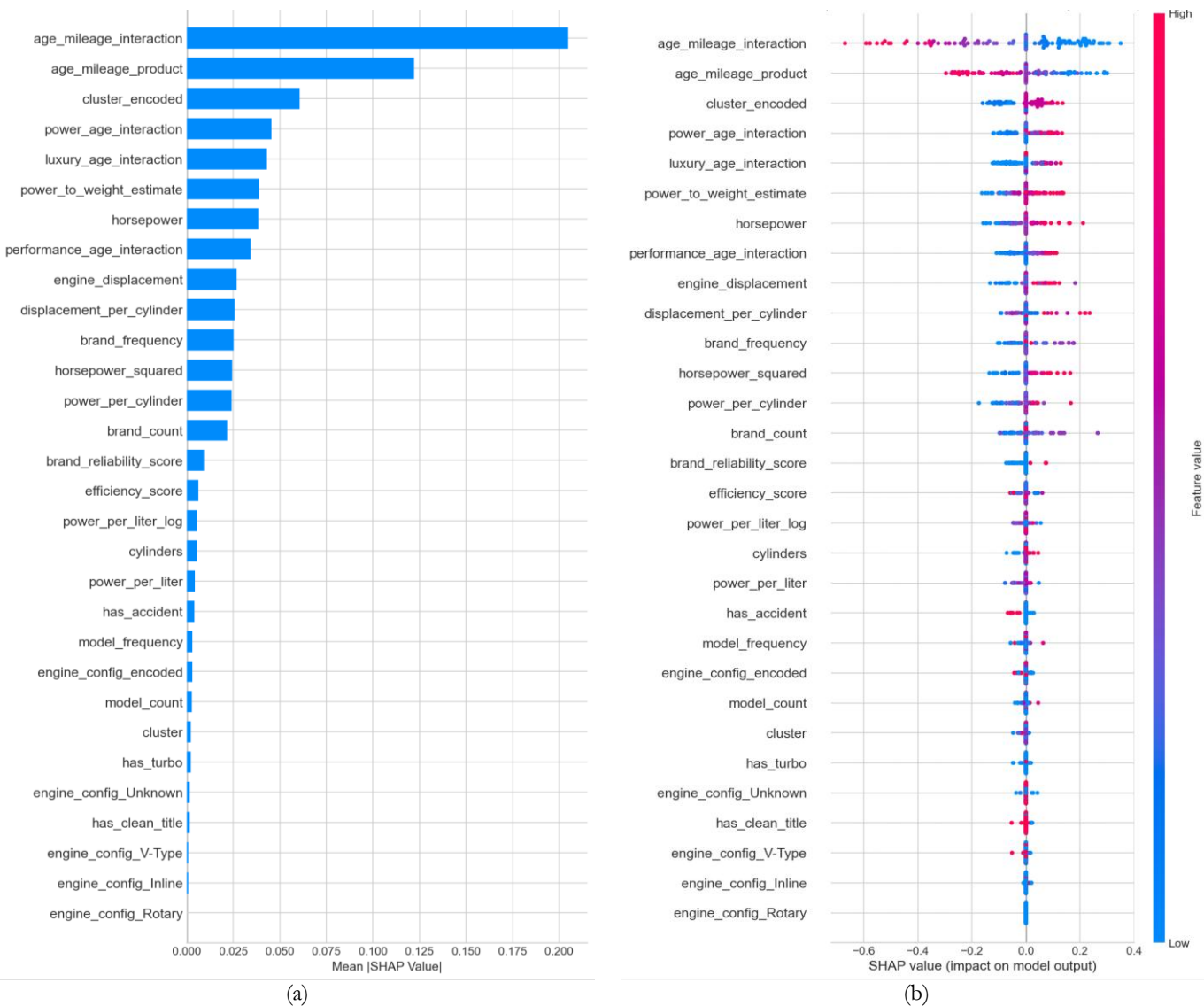


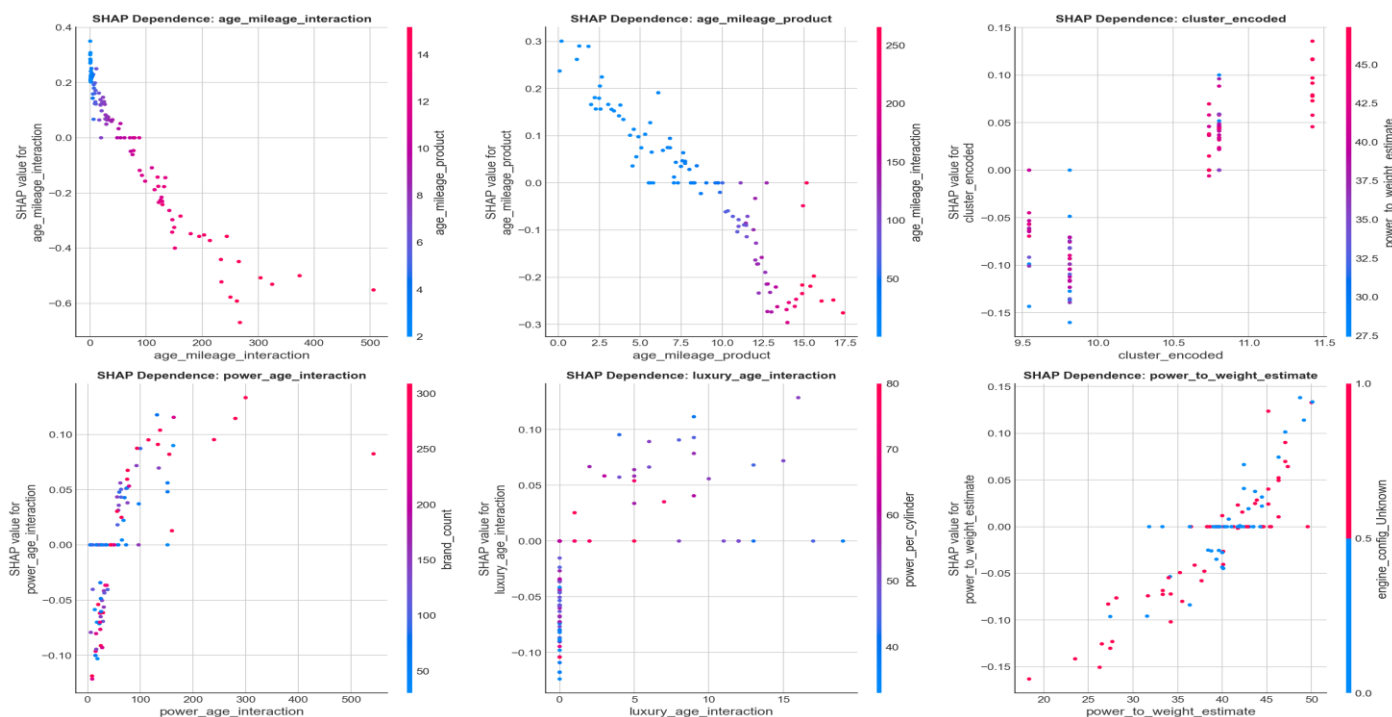
Figure 7. Global SHAP interpretation of the proposed stacking model. (a) Feature importance based on the mean absolute SHAP values. (b) SHAP beeswarm plot illustrating the distribution and direction of feature contributions across the testing samples.

To investigate whether hyperparameter optimization changes the interpretation of the model, the SHAP importance obtained from the optimized stacking framework was compared with that of the baseline configuration. The comparison is summarized in Table 11.

The comparison indicates that hyperparameter optimization produces only minor changes in the overall ranking of the most influential predictors. The two dominant interaction features remain ranked first and second in both models, demonstrating that Optuna primarily improves predictive accuracy without fundamentally altering the learned prediction mechanism. The largest relative increase is observed for luxury_age_interaction, whereas cluster_encoded exhibits the largest reduction in SHAP importance. Nevertheless, the overall ranking remains highly consistent, indicating that the proposed feature engineering strategy, rather than hyperparameter tuning, is the principal factor determining model behavior. To further investigate how individual feature values influence model predictions, Figure 8 presents SHAP dependence plots for the six most influential predictors.

Table 11. Comparison of SHAP feature importance between the baseline and Optuna-optimized stacking models

Feature	Rank (Optuna)	Rank (Baseline)	Mean SHAP (Optuna)	Mean SHAP (Baseline)	Relative Change
age_mileage_interaction	1	1	0.2050	0.1898	+8.0%
age_mileage_product	2	2	0.1221	0.1299	-6.0%
cluster_encoded	3	3	0.0606	0.0783	-22.6%
power_age_interaction	4	4	0.0455	0.0521	-12.7%
luxury_age_interaction	5	7	0.0430	0.0339	+26.6%

**Figure 8.** SHAP dependence plots for the six most influential features identified by the proposed stacking model.

The dependence plots provide additional insight into the learned pricing mechanism. Increasing values of `age_mileage_interaction` and `age_mileage_product` consistently decrease the predicted vehicle price, reflecting the expected depreciation associated with older vehicles and greater accumulated mileage. In contrast, higher values of `power_to_weight_estimate` generally produce positive SHAP values, indicating that superior engine performance contributes positively to vehicle valuation.

Several predictors also exhibit nonlinear response patterns. In particular, `cluster_encoded` forms distinct groups of SHAP values, suggesting that the clustering procedure captures discrete market segments with different pricing characteristics. Similarly, `power_age_interaction` and `luxury_age_interaction` demonstrate heterogeneous effects across the feature space, implying that the influence of engine performance depends on vehicle age and market segment rather than acting independently. So, the SHAP analysis corroborates the findings of the ablation study. Both analyses consistently identify interaction-based features as the primary source of predictive performance, while engine-related and market-oriented features provide complementary contributions. These results indicate that the performance improvement achieved by the proposed framework originates mainly from informative, domain-driven feature representations rather than increased model complexity or hyperparameter optimization alone.

5. Comparison with Previous Studies

To better position the proposed framework within the existing literature, the comparison is presented from two complementary perspectives. First, a qualitative comparison highlights

the methodological differences between the proposed framework and representative studies on used car price prediction. Second, a quantitative comparison is conducted using the most comparable benchmark study that employs the same publicly available dataset.

5.1. Methodological Comparison

Table 12 summarizes the methodological characteristics of representative studies on used vehicle price prediction. The comparison focuses on the principal components of the prediction framework, including feature engineering, hyperparameter optimization, stacking ensemble learning, model interpretability, ablation analysis, and evaluation strategy.

Table 12. Methodological comparison between the proposed framework and representative studies on used vehicle price prediction.

Study	Dataset	Feature Engineering	Hyperparameter Optimization	Stacking Ensemble	Model Interpretation	Ablation Analysis	Evaluation Strategy
Guo and Zhang [1]	China	X	Limited	X	X	X	Hold-out
Onserio and Mukudi [8]	Kenya	Limited	Limited	✓	X	X	5-fold CV
Fayyaz et al. [6]	US (Kaggle)	✓	Grid Search	✓	SHAP	Partial	5-fold CV
This study	US (Kaggle)	✓	Optuna	✓	SHAP	Comprehensive	Hold-out + 5-fold CV

Table 12 illustrates the evolution of recent methodologies for used vehicle price prediction. Early studies mainly emphasized algorithm selection or the application of a single high-performing regression model, whereas more recent research has increasingly incorporated ensemble learning and feature engineering. Nevertheless, most existing approaches focus on only a subset of the prediction pipeline, typically emphasizing either feature engineering, hyperparameter optimization, or ensemble learning independently.

Among the reviewed studies, Fayyaz et al. [6] provides the most appropriate benchmark because it employs the same publicly available Kaggle dataset used in this work. Both studies adopt a stacking ensemble and incorporate feature engineering; however, several methodological differences remain. The proposed framework employs a homogeneous gradient boosting ensemble consisting of CatBoost, LightGBM, and XGBoost optimized using Optuna, whereas Fayyaz et al. combine heterogeneous base learners optimized through Grid Search. Furthermore, the present study extends the evaluation by conducting comprehensive feature-group and model-component ablation analyses together with SHAP-based interpretation to systematically quantify the contribution of each component within the prediction pipeline.

Compared with the remaining studies, the proposed framework integrates domain-driven feature engineering, adaptive hyperparameter optimization, stacking ensemble learning, comprehensive ablation analysis, SHAP-based interpretation, and dual evaluation using both hold-out testing and 5-fold cross-validation within a unified experimental framework. Rather than introducing a fundamentally new learning algorithm, the main contribution of this work lies in providing a more systematic investigation of how feature representation and model configuration jointly influence predictive performance and model interpretability.

5.2. Quantitative Performance Comparison

A direct quantitative comparison is only meaningful when identical benchmark datasets are employed because differences in vehicle markets, feature distributions, and available predictor variables may substantially affect prediction performance. Among the representative studies summarized in Table 12, only Fayyaz et al. [6] evaluated their model using the same publicly available Kaggle dataset containing 4,009 used vehicles from the United States. Consequently, this study provides the most appropriate benchmark for a direct comparison. The quantitative comparison between the proposed framework and Fayyaz et al. [6] is presented in Table 13.

As shown in Table 13, the proposed framework achieves an R^2 of 0.9112, representing an improvement of approximately 2.4% over the previously reported stacking-based approach. Although the numerical improvement appears modest, it should be interpreted in the context of an already highly optimized benchmark. Achieving further performance gains on the same publicly available dataset becomes increasingly difficult because the baseline reported by previous studies is already close to the practical performance ceiling. Since both

studies employ the identical benchmark dataset, the observed improvement can be attributed primarily to differences in the proposed learning framework rather than variations in the underlying data.

Table 13. Quantitative comparison with the most comparable benchmark study using the same publicly available dataset.

Metric	Fayyaz et al. [6]	This Study
R^2	0.8899	0.9112
RMSE	0.1208	0.2387
MAE	—	0.1678

Note: RMSE and MAE were computed on the log-transformed target variable. Since Fayyaz et al. [6] did not report all evaluation metrics using the same evaluation protocol, direct quantitative comparison is primarily based on the coefficient of determination (R^2).

The observed performance improvement is supported by three complementary aspects of the proposed methodology. The primary novelty of the proposed framework lies in the unified integration of domain-driven feature engineering, Bayesian hyperparameter optimization using Optuna, and a gradient boosting stacking ensemble. Unlike previous studies that primarily emphasize algorithm selection, the proposed framework systematically combines feature representation, model optimization, and ensemble learning within a single prediction pipeline. Overall, the qualitative and quantitative comparisons consistently demonstrate that the proposed framework extends previous work not only by achieving competitive predictive performance but also by providing a more comprehensive learning and evaluation framework. The integration of domain-driven feature engineering, systematic hyperparameter optimization, robust evaluation through cross-validation, comprehensive ablation analysis, and SHAP-based interpretation offers a more transparent and reproducible approach to used vehicle price prediction.

6. Conclusions

This study proposed a stacking-based framework for used vehicle price prediction by integrating CatBoost, LightGBM, and XGBoost within a two-level ensemble architecture optimized using Optuna. A domain-driven feature engineering strategy was introduced to enrich the original dataset through interaction features, engine-derived descriptors, market-oriented variables, and clustering-based representations. The experimental results demonstrate that the proposed framework consistently achieves the highest predictive performance among the evaluated models, obtaining an R^2 of 0.9112 on the independent testing set and 0.9023 ± 0.0025 under 5-fold cross-validation.

The experimental analysis further demonstrates that the proposed feature engineering strategy contributes substantially to predictive performance. Expanding the original feature set through domain-driven feature engineering consistently improved the predictive performance of all evaluated regression models. Ablation experiments reveal that interaction-based features provide the largest contribution, whereas engine configuration and clustering features contribute comparatively smaller improvements. The stacking component analysis further indicates that CatBoost provides the strongest individual predictive capability among the base learners, while Ridge Regression serves as an effective meta-learner for integrating their complementary predictions.

SHAP analysis provides complementary evidence supporting the ablation study. Both analyses consistently identify interaction-based features, particularly age_mileage_interaction and age_mileage_product, as the dominant predictors of vehicle price. These findings indicate that carefully designed feature representations contribute more substantially to predictive performance than increasing model complexity alone. Overall, the findings demonstrate that jointly optimizing feature representation and ensemble learning provides a robust and interpretable solution for used vehicle price prediction. Nevertheless, several limitations should be acknowledged. The proposed framework was evaluated using a single publicly available dataset collected from the United States, and therefore its generalizability to other regional vehicle markets requires further investigation. In addition, temporal variations in vehicle prices and broader contextual factors, such as macroeconomic conditions, fuel prices, and regional market demand, were not explicitly incorporated into the prediction process. Future

research may address these limitations by validating the proposed framework on datasets collected from different countries, incorporating temporal and market-related information, and investigating advanced learning paradigms such as transformer-based tabular models, multi-modal learning with vehicle images, graph neural networks for modeling inter-vehicle relationships, and federated learning for privacy-preserving vehicle valuation.

Author Contributions: Conceptualization: D.I.M. and D.R.I.M.S.; Methodology: D.I.M. and D.R.I.M.S.; Software: D.I.M.; Validation: D.R.I.M.S., Y.K.W., and Z.E.; Formal analysis: D.I.M. and D.R.I.M.S.; Investigation: D.R.I.M.S., Y.K.W., and Z.E.; Resources: D.I.M.; Writing—original draft preparation: D.I.M.; Writing—review and editing: D.R.I.M.S., Y.K.W., and Z.E.; Visualization: X.X.; Supervision: D.R.I.M.S.; Project administration: D.I.M.; Funding acquisition: none. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: The dataset used in this study is openly available on Kaggle at <https://www.kaggle.com/datasets/taefnajib/used-car-price-prediction-dataset>. The source code and trained models are available from the corresponding author upon reasonable request.

Acknowledgments: The authors would like to acknowledge the use of artificial intelligence (AI)-assisted writing tools during the preparation of this manuscript. These tools were used solely to improve language quality, grammar, clarity, and readability. All research ideas, methodology, experimental design, data analysis, interpretation of results, and final editorial decisions were developed and verified by the authors, who take full responsibility for the content of this manuscript.

Conflicts of Interest: The authors declare no conflict of interest.

References

- [1] S. Guo and B. Zhang, “Revolutionizing the used car market: Predicting prices with XGBoost,” *Appl. Comput. Eng.*, vol. 48, no. 1, pp. 173–180, Mar. 2024, doi: 10.54254/2755-2721/48/20241349.
- [2] D. Winarso, Edo Arribe, Syahril, Aryanto, Muhandi, and Shahrulniza Musa, “Comparing Optimization Algorithms in ANN Models for House Price Prediction in Pekanbaru,” *J. RESTI (Rekayasa Sist. dan Teknol. Informatika)*, vol. 9, no. 4, pp. 781–787, Aug. 2025, doi: 10.29207/resti.v9i4.6619.
- [3] L. Cudennec, “Scaling Up Optuna: P2P Distributed Hyperparameters Optimization,” *Concurr. Comput. Pract. Exp.*, vol. 37, no. 4–5, Feb. 2025, doi: 10.1002/cpe.70008.
- [4] J. Lin, “Predicting Used Car Price Based on Machine Learning,” in *Proceedings of the 1st International Conference on E-commerce and Artificial Intelligence*, 2024, pp. 553–560. doi: 10.5220/0013270500004568.
- [5] E. K. Khotimah, D. I. Swasono, and G. W. Fajarianto, “Forecasting Used Car Prices Using Machine Learning,” *IT J. Res. Dev.*, vol. 9, no. 2, pp. 123–139, Mar. 2025, doi: 10.25299/itjrd.2025.18031.
- [6] I. Fayyaz, G. G. M. N. Ali, and S. S. Khairunnisa, “Advanced Feature Engineering and Machine Learning Techniques for High Accurate Price Prediction of Heterogeneous Pre-Owned Cars,” *Vehicles*, vol. 7, no. 3, p. 94, Sep. 2025, doi: 10.3390/vehicles7030094.
- [7] T. Najib, “Used Car Price Prediction Dataset,” *Kaggle*, 2025. <https://www.kaggle.com/datasets/taefnajib/used-car-price-prediction-dataset>
- [8] M. Onserio and F. Mukudi, “Ensemble Machine Learning Model for Predicting Prices of Used Cars in Kenya,” *Sci. Front.*, vol. 6, no. 3, pp. 96–105, Aug. 2025, doi: 10.11648/j.sf.20250603.15.
- [9] I. D. Mienye and Y. Sun, “A Survey of Ensemble Learning: Concepts, Algorithms, Applications, and Prospects,” *IEEE Access*, vol. 10, no. August, pp. 99129–99149, 2022, doi: 10.1109/ACCESS.2022.3207287.
- [10] H. N. Kudagama, S. M. Arachchi, K. G. A. Yohan, and C. L. I. S. Fonseka, “Hourly UV Index Forecasting Across Climatic Zones in Sri Lanka: A Residual-Correction Hybrid Machine Learning and Deep Learning Approach,” *J. Futur. Artif. Intell. Technol.*, vol. 3, no. 2, pp. 234–256, Jun. 2026, doi: 10.62411/faith.3048-3719-372.
- [11] B. So, “Enhanced gradient boosting for zero-inflated insurance claims and comparative analysis of CatBoost, XGBoost, and LightGBM,” *Scand. Actuar. J.*, vol. 2024, no. 10, pp. 1013–1035, Nov. 2024, doi: 10.1080/03461238.2024.2365390.
- [12] T. R. Noviandy, G. M. Idroes, and I. Hardi, “An Interpretable Machine Learning Strategy for Antimalarial Drug Discovery with LightGBM and SHAP,” *J. Futur. Artif. Intell. Technol.*, vol. 1, no. 2, pp. 84–95, Aug. 2024, doi: 10.62411/faith.2024-16.
- [13] D. R. I. M. Setiadi, K. Nugroho, A. R. Muslikh, S. W. Iriananda, and A. A. Ojugo, “Integrating SMOTE-Tomek and Fusion Learning with XGBoost Meta-Learner for Robust Diabetes Recognition,” *J. Futur. Artif. Intell. Technol.*, vol. 1, no. 1, pp. 23–38, May 2024, doi: 10.62411/faith.2024-11.
- [14] R. Szczepanek, “Daily Streamflow Forecasting in Mountainous Catchment Using XGBoost, LightGBM and CatBoost,” *Hydrology*, vol. 9, no. 12, p. 226, Dec. 2022, doi: 10.3390/hydrology9120226.

- [15] B. Bischl *et al.*, “Hyperparameter optimization: Foundations, algorithms, best practices, and open challenges,” *WIREs Data Min. Knowl. Discov.*, vol. 13, no. 2, Mar. 2023, doi: 10.1002/widm.1484.
- [16] M. Sakib, T. Siddiqui, S. Mustajab, R. M. Alotaibi, N. M. Alshareef, and M. Z. Khan, “An ensemble deep learning framework for energy demand forecasting using genetic algorithm-based feature selection,” *PLoS One*, vol. 20, no. 1, p. e0310465, Jan. 2025, doi: 10.1371/journal.pone.0310465.
- [17] R. Garine, “Enhanced E-Commerce Demand Prediction through Ensemble Models and Optuna-Based Hyperparameter Optimization,” in *2024 2nd DMIHER International Conference on Artificial Intelligence in Healthcare, Education and Industry (IDICAIEI)*, Nov. 2024, pp. 1–7. doi: 10.1109/IDICAIEI61867.2024.10842873.
- [18] A. Serrat and B. Djebbar, “Solving Symmetrical Drop Suspended Equilibrium Equation by Artificial Bee Colony Programming,” *Math. Model. Eng. Probl.*, vol. 9, no. 2, pp. 507–514, Apr. 2022, doi: 10.18280/mmep.090229.
- [19] P. Bhatnagar, G. H. Lokesh, J. Shreyas, F. Flammini, and S. Gautam, “An Analysis of Car Price Prediction using Machine Learning,” in *2024 9th International Conference on Machine Learning Technologies (ICMLT)*, May 2024, pp. 11–15. doi: 10.1145/3674029.3674032.
- [20] D. A. Budiono, K. S. Utomo, K. J. Wibowo, and M. J. Wiradinata, “Used Car Price Prediction Model: A Machine Learning Approach,” *Int. J. Comput. Inf. Syst.*, vol. 5, no. 1, pp. 59–66, Mar. 2023, doi: 10.29040/ijcis.v5i1.147.
- [21] A. Dutulescu *et al.*, “Car Price Quotes Driven by Data-Comprehensive Predictions Grounded in Deep Learning Techniques,” *Electronics*, vol. 12, no. 14, p. 3083, Jul. 2023, doi: 10.3390/electronics12143083.
- [22] S. Bergmann and S. Feuerriegel, “Machine learning for predicting used car resale prices using granular vehicle equipment information,” *Expert Syst. Appl.*, vol. 263, p. 125640, Mar. 2025, doi: 10.1016/j.eswa.2024.125640.
- [23] N. Bussmann, P. Giudici, A. Tanda, and E. P.-Y. Yu, “Explainable machine learning to predict the cost of capital,” *Front. Artif. Intell.*, vol. 8, Apr. 2025, doi: 10.3389/frai.2025.1578190.
- [24] N. S, S. R. R, T. Thameem Ansari S, A. R. S, V. A, and N. D. R, “An Intelligent Machine Learning based System for Car Price Estimation,” in *2025 5th International Conference on Expert Clouds and Applications (ICOECA)*, Mar. 2025, pp. 556–561. doi: 10.1109/ICOECA66273.2025.00101.
- [25] Jesutofunmi E. Fagbamila, Abass A. Agbaje, and Ganiyu O. Okubadejo, “Predictive analytics for pricing strategy in the automobile industry using machine learning models,” *World J. Adv. Res. Rev.*, vol. 24, no. 3, pp. 3543–3550, Dec. 2024, doi: 10.30574/wjarr.2024.24.3.3920.
- [26] Y. Zheng, “Machine Learning Optimization and Challenges in Used Car Price Prediction,” *ITM Web Conf.*, vol. 70, p. 04032, Jan. 2025, doi: 10.1051/itmconf/20257004032.
- [27] G. Rohini, T. Mariprath, S. Al Bustanji, and I. Zaitsev, “A stacked Gradient Boosting–XGBoost ensemble with ridge meta-learner for accurate short-term solar PV power forecasting in smart grids,” *Sci. Rep.*, vol. 16, no. 1, p. 16871, Apr. 2026, doi: 10.1038/s41598-026-47042-3.
- [28] R. Shen and C. Li, “A Machine Learning-Based Ridge-Meta Ensemble Model for Electrical Load Forecasting in Integrated Energy Systems,” in *2026 9th International Conference on Energy, Electrical and Power Engineering (CEEPE)*, Apr. 2026, pp. 1261–1267. doi: 10.1109/CEEPE69795.2026.11552101.
- [29] Y. Ozaki, Y. Tanigaki, S. Watanabe, M. Nomura, and M. Onishi, “Multiobjective Tree-Structured Parzen Estimator,” *J. Artif. Intell. Res.*, vol. 73, pp. 1209–1250, Apr. 2022, doi: 10.1613/jair.1.13188.
- [30] S. Watanabe, “Tree-Structured Parzen Estimator: Understanding Its Algorithm Components and Their Roles for Better Empirical Performance,” *arXiv*. May 31, 2023. [Online]. Available: <http://arxiv.org/abs/2304.11127>
- [31] S. M. Lundberg and S.-I. Lee, “A Unified Approach to Interpreting Model Predictions,” in *NIPS’17: Proceedings of the 31st International Conference on Neural Information Processing Systems*, Nov. 2017, pp. 4768–4777. [Online]. Available: <https://dl.acm.org/doi/10.5555/3295222.3295230>
- [32] A. Prashanthan and J. Prashanthan, “Optimized Explainable Predictive Models for Risk-Based Prioritization in Type 2 Diabetes Prevention among Women with Prior Gestational Diabetes,” *J. Futur. Artif. Intell. Technol.*, vol. 2, no. 4, pp. 734–759, Feb. 2026, doi: 10.62411/faith.3048-3719-321.
- [33] Q. Wu, T. F. Ma, Y.-H. Kuo, C. Y. Yau, and T. Cheung, “Explainable Used Car Pricing: A Machine Learning Framework With Analysis of Marginal Effects,” *IEEE Trans. Eng. Manag.*, vol. 73, pp. 305–321, 2026, doi: 10.1109/TEM.2025.3635464.