Car Price Prediction Using Machine Learning Regression Models

Hakan ONAY

Erciyes University, Software Engineering Department

Introduction to Data Science

Instructor: Rukiye Nur KAÇMAZ

December 30, 2025

Abstract

Accurately estimating vehicle prices is crucial in the automotive industry for buyers, sellers, and market analysts. This study presents a machine learning approach to predict used car selling prices using the CarDekho dataset. After cleaning and removing outliers from the target variable using the interquartile range (IQR) rule, the dataset contained 14,025 records. Features included numerical attributes (vehicle_age, km_driven, mileage, engine, max_power, seats) and categorical attributes (brand, seller_type, fuel_type, transmission_type). The high-cardinality model feature (92 unique values) was label-encoded as model_encoded. Five regression models (Linear Regression, Ridge, Lasso, Random Forest, Gradient Boosting) were trained with a unified preprocessing pipeline. Models were evaluated using R^2, RMSE, MAE, and 5-fold cross-validation. Results show that Gradient Boosting achieved the best test performance (Test R^2 = 0.901, RMSE = 87,475 currency units, MAE = 63,975 currency units), followed closely by Random Forest (Test R^2 = 0.896, RMSE = 89,783 currency units). Feature importance analysis highlighted max_power, vehicle_age, and engine as the most influential predictors. Overall, ensemble models provided substantially better predictive accuracy than linear baselines for this tabular pricing task.

Keywords: Car price prediction, machine learning, regression, random forest, gradient boosting, CarDekho

Car Price Prediction Using Machine Learning Regression Models

1. Introduction

Online used-car marketplaces require fast and consistent valuation of listings. Manual valuation is often subjective, while supervised machine learning can learn pricing patterns from historical data. This study builds and compares multiple regression models to predict a vehicle's selling price from structured attributes. The analysis uses the CarDekho dataset distributed via Kaggle (Kaggle, n.d.).

Problem Definition

Used-car prices depend on interacting factors such as brand, age, mileage, engine size, and power. Key challenges include extreme price outliers, heterogeneous feature types (numerical and categorical), and overfitting risk in high-capacity models. This work addresses these issues using a strict preprocessing pipeline, robust evaluation metrics, and cross-validation.

2. Aim of the Project

- Build a reliable regression model to estimate used-car selling prices from structured vehicle attributes.
- Compare multiple regression algorithms (Linear, Ridge, Lasso, Random Forest, Gradient Boosting) under the same preprocessing pipeline.
- Evaluate performance using R^2, RMSE, MAE, and 5-fold cross-validation to assess generalization.
- Interpret the best model with feature importance and analyze errors by price segment to understand where the model performs well/poorly.

3. Literature Review

Used-car price prediction is typically formulated as a supervised regression problem where the target is the transaction/selling price and inputs include vehicle age, mileage, engine specifications, brand, and transmission. Prior work on the CarDekho dataset compared linear models (Linear/Ridge) against tree ensembles and reported that Random Forest achieved superior predictive performance, highlighting the strength of non-linear ensemble methods for this domain (Lin, 2024).

Recent studies also emphasize the value of richer feature sets and ensemble learning. For example, Bergmann and Feuerriegel demonstrate that granular vehicle equipment information can materially improve used-car resale value prediction with machine learning models (Bergmann & Feuerriegel, 2025). In the Turkish context, Gülmez and Kulluk compared multiple algorithms using Apache Spark and likewise found Random Forest to be among the strongest performers (Gülmez & Kulluk, 2023). These findings motivated our inclusion of ensemble models and our focus on both accuracy metrics and error analysis.

4. Materials

Dataset: CarDekho Used Car Dataset (Kaggle), containing 15,411 records and 14 columns before cleaning, including brand/model information and numerical vehicle attributes. After IQR-based outlier filtering on the target variable, 14,025 records were used for modeling. Tools: Python (NumPy, pandas), scikit-learn for preprocessing/modeling, and Matplotlib/Seaborn for visualization.

5. Methods

Dataset

The raw dataset contained 15,411 records and 14 columns. Two non-informative columns (Unnamed: 0 and car_name) were removed, leaving 12 columns. No missing values were observed in the provided dataset, so no rows were removed during missing-value cleaning. The target variable was selling_price (in currency units).

Preprocessing

Outliers were detected in selling_price using the IQR method. With Q1 = 385,000 currency units and Q3 = 825,000 currency units, the upper bound was 1,485,000 currency units (1.5×IQR). Listings above this upper bound were removed (n = 1,386; 9.0%), resulting in 14,025 records. Categorical variables (brand, seller_type, fuel_type, transmission_type) were one-hot encoded. The model feature (92 unique values) was label-encoded into model_encoded and treated as numerical. Numerical features were standardized with StandardScaler. All transformations were implemented with ColumnTransformer and Pipeline to avoid data leakage during evaluation. Monetary values are reported in the dataset's original currency; plots label the unit as "TL" due to a formatting helper and should be interpreted as currency units.

Models and Evaluation

Five regression algorithms were trained: Linear Regression, Ridge Regression (alpha = 1.0), Lasso Regression (alpha = 0.01, max_iter = 10,000), Random Forest Regressor (n_estimators = 100, max_depth = 15, min_samples_split = 5, min_samples_leaf = 2), and Gradient Boosting Regressor (n_estimators = 100, learning_rate = 0.1, max_depth = 5, subsample = 0.8). The dataset was split into training (80%) and test (20%) sets with random_state = 42 (n_train = 11,220; n_test = 2,805). Performance was evaluated using R^2, root

mean square error (RMSE), and mean absolute error (MAE) on the held-out test set. Additionally, 5-fold cross-validation (KFold, shuffle = True) was applied on the training set. The difference between training and test R^2 was used as an overfitting indicator. Data preparation used pandas (McKinney, 2010), and modeling was implemented using scikit-learn (Pedregosa et al., 2011).

Implementation Snippets (Code Screenshots)

Figure A1

Preprocessing pipeline (ColumnTransformer + Pipeline).

```
# Preprocessing
categorical_cols = ["brand", "seller_type", "fuel_type", "transmission_type"]
numerical_cols = ["vehicle_age", "km_driven", "mileage", "engine", "max_power", "seats", "model_encoded"]

preprocessor = ColumnTransformer(
    transformers=[
        ("num", StandardScaler(), numerical_cols),
        ("cat", OneHotEncoder(handle_unknown="ignore", sparse_output=False), categorical_cols),
    ]
)

# Example pipeline
gb_model = Pipeline([
    ("preprocessor", preprocessor),
    ("regressor", GradientBoostingRegressor(
        n_estimators=100, learning_rate=0.1, max_depth=5, subsample=0.8, random_state=42
    ))
])
```

Note. Screenshot-style snippet illustrating the preprocessing design used in the implementation.

Figure A2

Training and evaluation loop (metrics + cross-validation).

```
# Training & evaluation loop (core idea)
for name, model in models.items():
    model.fit(X_train, y_train)
    y_test_pred = model.predict(X_test)

    test_r2 = r2_score(y_test, y_test_pred)
    rmse = np.sqrt(mean_squared_error(y_test, y_test_pred))
    mae = mean_absolute_error(y_test, y_test_pred)

    cv_scores = cross_val_score(
        model, X_train, y_train,
        cv=KFold(n_splits=5, shuffle=True, random_state=42),
        scoring="r2", n_jobs=-1
    )
```

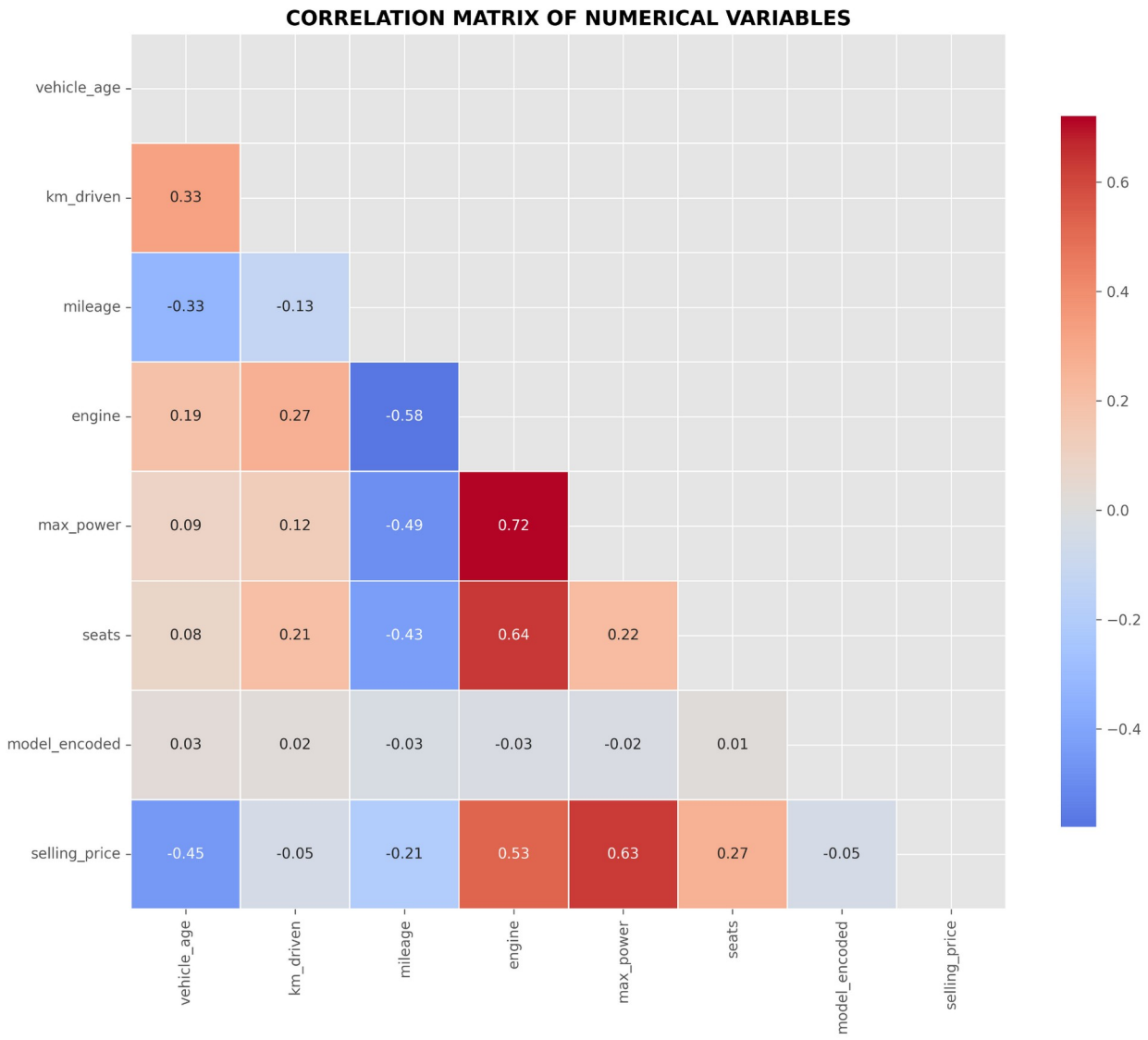
Note. Screenshot-style snippet showing how metrics and 5-fold CV are computed consistently across models.

Exploratory Data Analysis

A correlation analysis of numerical variables showed that selling_price correlated most strongly with max_power (r = 0.63) and engine (r = 0.53), and negatively with vehicle_age (r = -0.45). Seats had a weaker positive relationship (r = 0.27), and mileage had a modest negative relationship (r = -0.21).

Figure 1

Correlation matrix of numerical variables (including selling_price).



Note. Correlation coefficients are computed on the cleaned dataset (n = 14,025).

6. Results

Note. This is a regression task; therefore, classification metrics such as accuracy and the confusion matrix are not applicable. Model performance is reported using R^2, RMSE, MAE, cross-validation scores, and residual/error distributions.

Table 1 summarizes the model performance. Ensemble models substantially outperformed linear baselines. Gradient Boosting achieved the best test performance (Test R^2 = 0.901; RMSE = 87,475 currency units; MAE = 63,975 currency units) and had a small train-test gap (0.016), indicating good generalization. Random Forest was a close second (Test R^2 = 0.896; RMSE = 89,783 currency units) but showed a larger train-test gap (0.057), suggesting higher overfitting risk. Linear, Ridge, and Lasso models achieved Test $R^2 \approx 0.777–0.778$, indicating lower performance on this dataset.

Table 1

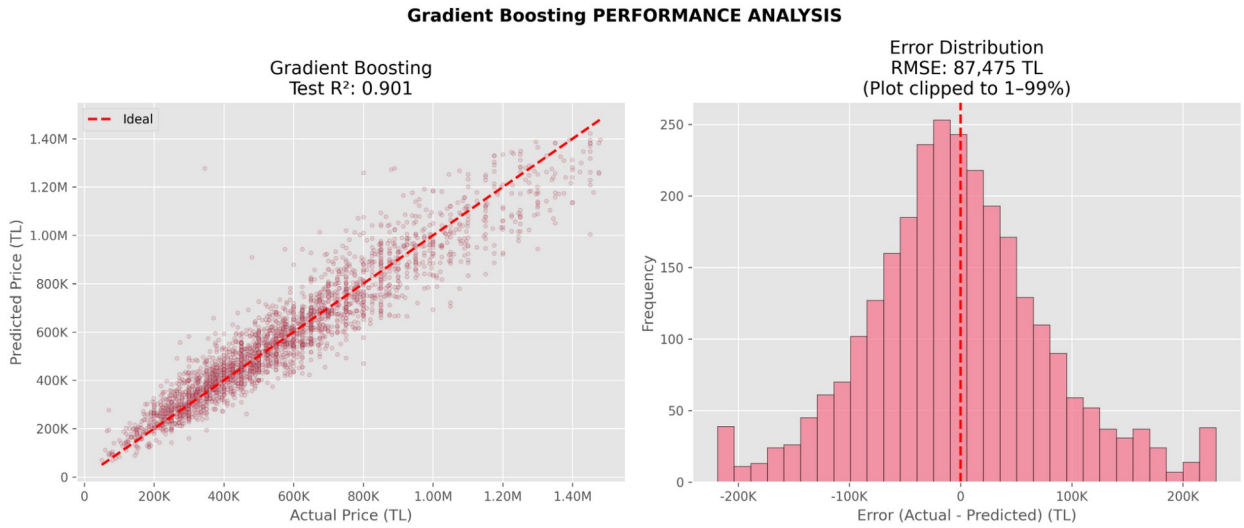
Model performance on the test set and 5-fold cross-validation.

Model	**Train R^2**	**Test R^2**	**R^2 Diff**	**CV R^2 Mean**	**CV R^2 SD**	**RMSE (currency units)**	**MAE (currency units)**
Gradient Boosting	0.9170	0.9010	0.0160	0.8927	0.0022	87,475	63,975
Random Forest	0.9529	0.8957	0.0573	0.8882	0.0051	89,783	64,529
Linear Regression	0.7794	0.7778	0.0017	0.7769	0.0078	131,040	97,516
Lasso Regression	0.7794	0.7778	0.0017	0.7769	0.0077	131,040	97,516

Ridge Regression	0.7793	0.7774	0.0019	0.7770	0.0074	131,140	97,624

Figure 2

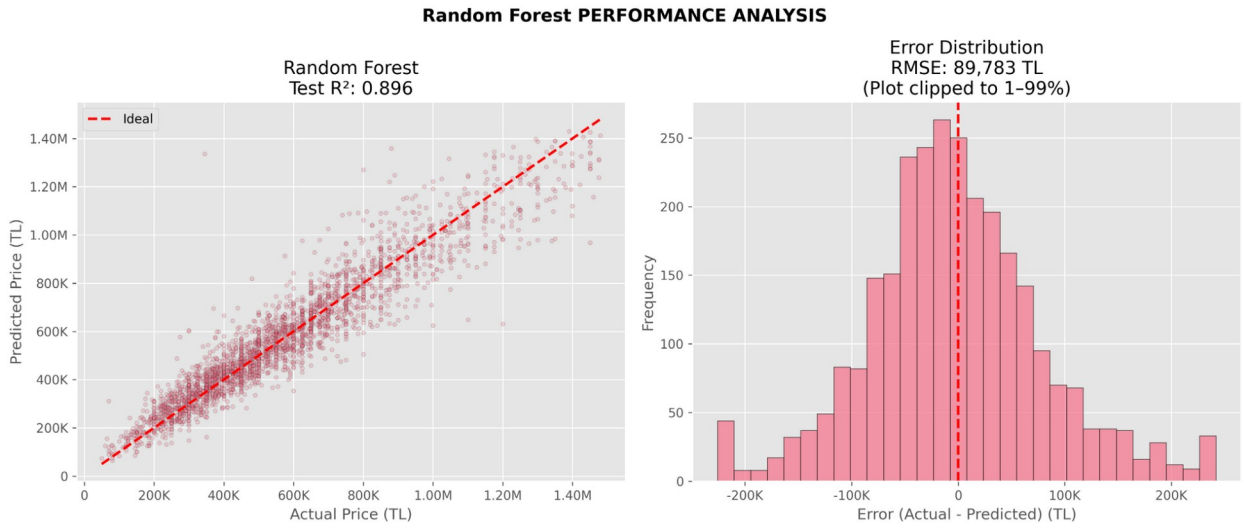
Gradient Boosting performance: actual vs. predicted and error distribution.



Note. Error is defined as Actual - Predicted. Histogram is clipped to the 1st-99th percentiles for readability.

Figure 3

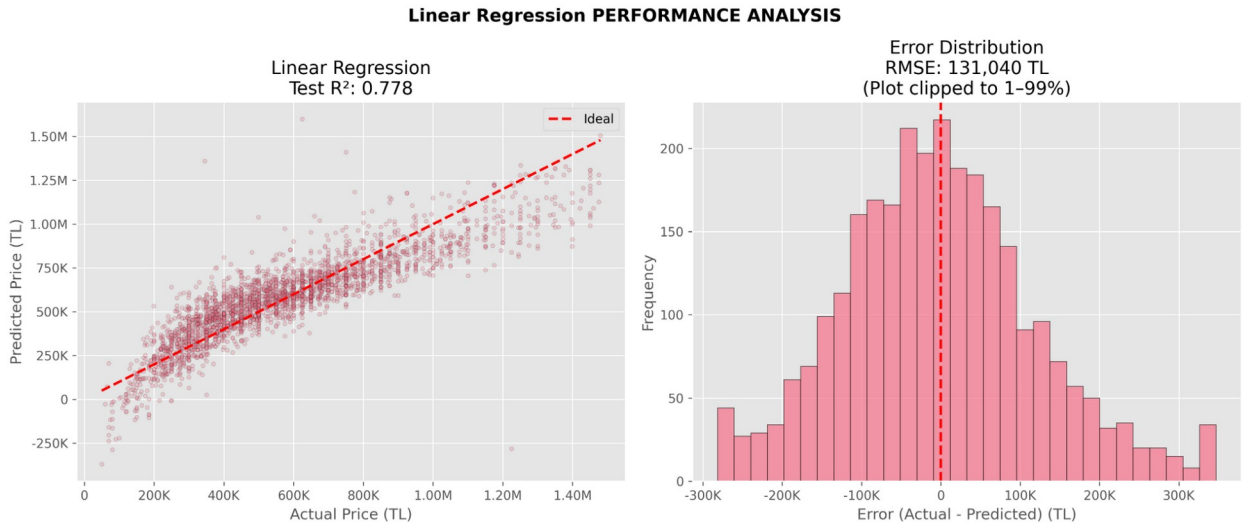
Random Forest performance: actual vs. predicted and error distribution.



Note. Error is defined as Actual - Predicted. Histogram is clipped to the 1st-99th percentiles for readability.

Figure 4

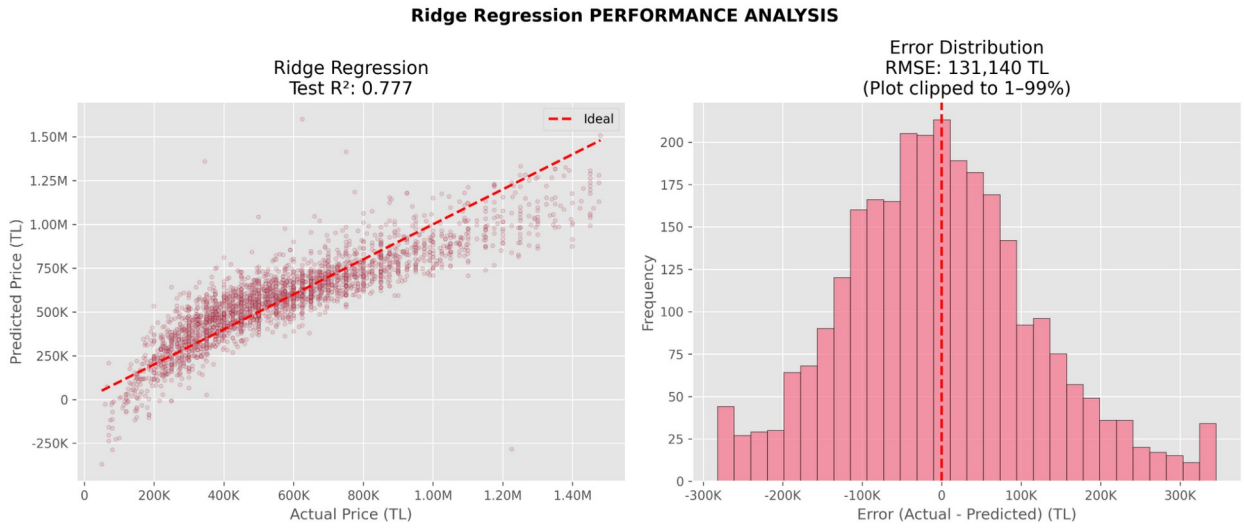
Linear Regression performance: actual vs. predicted and error distribution.



Note. Error is defined as Actual - Predicted. Histogram is clipped to the 1st-99th percentiles for readability.

Figure 5

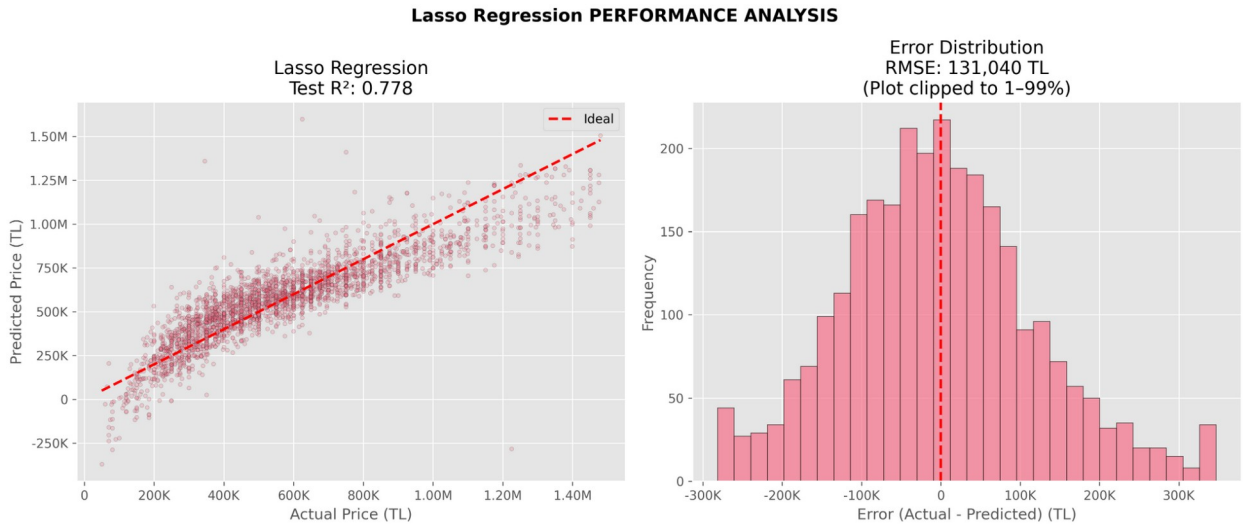
Ridge Regression performance: actual vs. predicted and error distribution.



Note. Error is defined as Actual - Predicted. Histogram is clipped to the 1st-99th percentiles for readability.

Figure 6

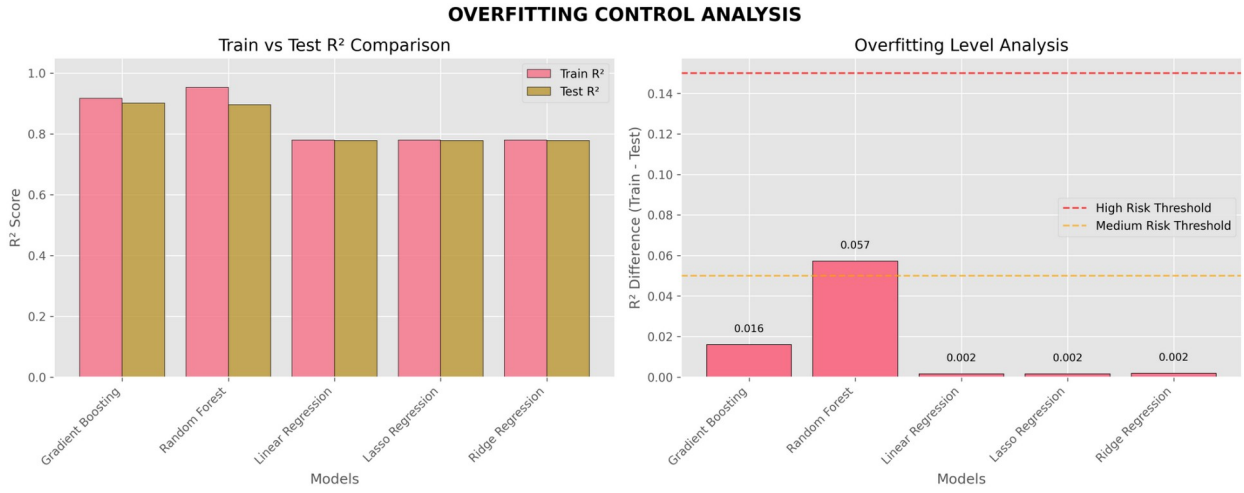
Lasso Regression performance: actual vs. predicted and error distribution.



Note. Error is defined as Actual - Predicted. Histogram is clipped to the 1st-99th percentiles for readability.

Figure 7

Overfitting control: train vs. test R^2 and train-test R^2 difference.



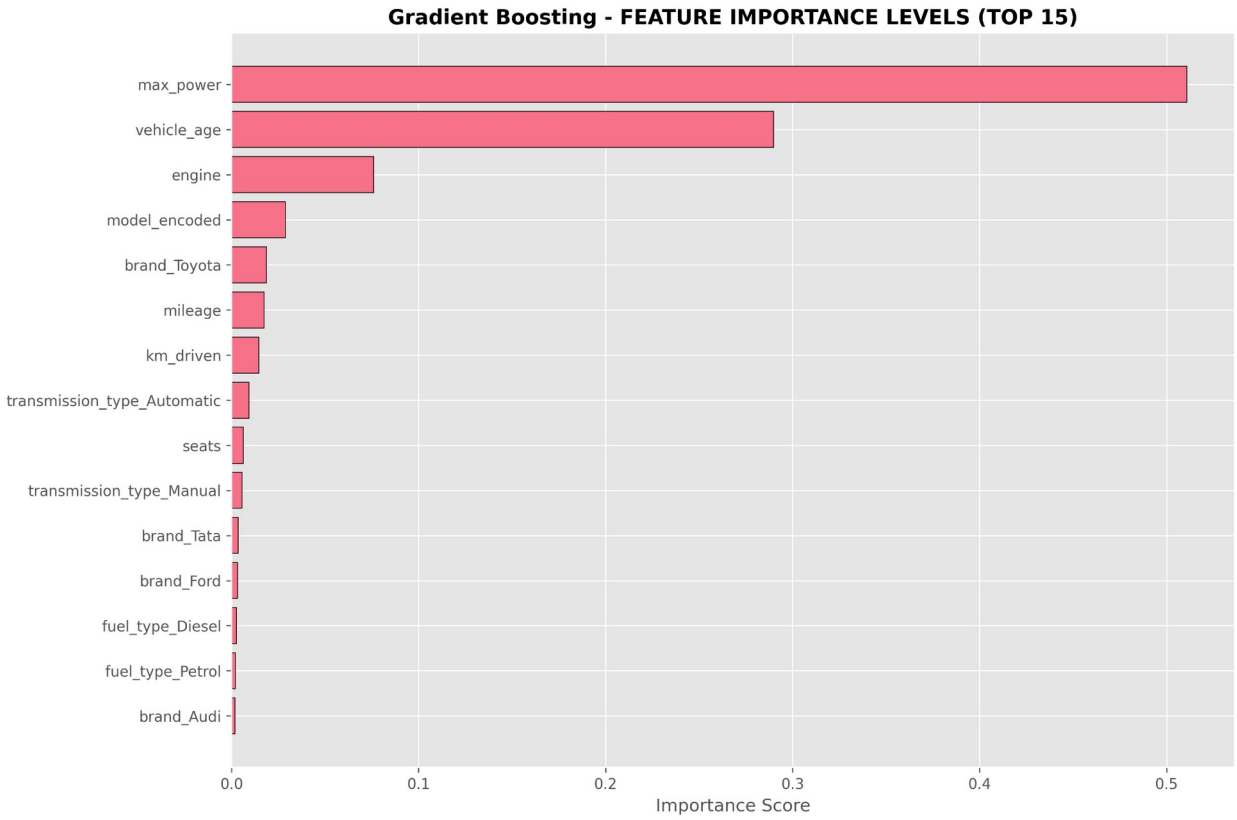
Note. Higher values indicate greater overfitting risk. Dashed lines denote medium and high risk thresholds.

Feature Importance

Feature importance values were extracted from the Gradient Boosting model. The most influential predictors were max_power (0.511), vehicle_age (0.290), and engine (0.076). These results align with domain expectations: higher power and larger engines are typically associated with higher prices, while older vehicles tend to have lower prices.

Figure 8

Top-15 feature importance values for Gradient Boosting.



Note. Importance values are normalized to sum to 1 across all features.

Price Segment Analysis

To inspect performance across price ranges, test samples were grouped into price segments: Very Low (0-300k), Low (300k-600k), Medium (600k-1M), and High (1M-2M currency units). Mean percentage errors indicate a tendency to overestimate in the Very Low segment (mean = -14.98%) and to underestimate in the High segment (mean = 6.88%). The Medium segment showed the smallest bias (mean = 0.74%). Very High and Premium segments contained no samples because prices above the IQR upper bound (1.485M) were removed as outliers.

7. Discussion

Overall, ensemble models captured nonlinear interactions in vehicle pricing better than linear baselines. The Random Forest model achieved high accuracy but exhibited a larger train-test gap, consistent with overfitting risk. Gradient Boosting provided the best balance of accuracy and generalization, supported by strong test performance and stable cross-validation results. The use of preprocessing pipelines ensured consistent transformations and minimized evaluation leakage.

Limitations and Future Work

This study has limitations. First, the dataset reflects a particular market context and time period; generalization to other regions may be limited. Second, important external factors such as location, accident history, and macroeconomic conditions were not included. Third, hyperparameters were selected manually rather than tuned systematically. Future work could evaluate modern gradient-boosted tree libraries (e.g., XGBoost or LightGBM), perform hyperparameter optimization (e.g., GridSearchCV), and deploy the final model as a web service for interactive price estimation.

8. Conclusion

This project implemented an end-to-end workflow for used car price prediction using machine learning regression models. After IQR-based outlier removal, five models were trained with a unified preprocessing pipeline and evaluated using multiple metrics. Gradient Boosting achieved the best overall results (Test $R^2 = 0.901$; RMSE = 87,475 currency units) and identified max_power, vehicle_age, and engine as the most influential predictors. These findings demonstrate the effectiveness of ensemble learning for structured vehicle pricing data.

9. References

Bergmann, S., & Feuerriegel, S. (2025). Machine learning for predicting used car resale prices using granular vehicle equipment information. Expert Systems with Applications, 263, 125640. https://doi.org/10.1016/j.eswa.2024.125640

Gülmez, B., & Kulluk, S. (2023). Predicting used car prices with machine learning algorithms using Apache Spark. Journal of the Faculty of Engineering and Architecture of Gazi University, 38(4), 2279–2289. https://doi.org/10.17341/gazimmfd.980840

Kaggle. (n.d.). CarDekho used car dataset [Data set]. https://www.kaggle.com/datasets/manishkr1754/cardekho-used-car-data

Lin, J. (2024). Predicting used car price based on machine learning. In Proceedings of the 1st International Conference on E-commerce and Artificial Intelligence (ECAI 2024) (pp. 553–560). SCITEPRESS. https://doi.org/10.5220/0013270500004568

McKinney, W. (2010). Data structures for statistical computing in Python. In S. van der Walt & J. Millman (Eds.), Proceedings of the 9th Python in Science Conference (pp. 51–56).

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., & Duchesnay, É. (2011). Scikit-learn: Machine learning in Python. Journal of Machine Learning Research, 12, 2825–2830. https://jmlr.org/papers/v12/pedregosa11a.html