Predicting Used Car Prices with Machine Learning: A Data-Driven Approach

Author : ERAHOUTEN ILIASS **Supervisor : P. KHAMJANE AZIZ**

iliass.erahouten@etu.uae.ac.ma akhamjane@uae.ac.ma

Department of Mathematics & Computer Science, Abdelmalek Essaadi Univ.
ENSAH College, Alhoceima, Morocco.

Abstract

Predicting used car prices is a complex task, especially as the car market has become highly competitive and the growing demand for specialized vehicles has made used cars an attractive option for buyers. This adds to the challenge of accurately estimating their value in the current market. That's why it's necessary to use a machine learning model to predict the price of cars, helping buyers and sellers determine fair prices.

In this study, we examine various features that impact used car prices, such as mileage, year, make, power, and other characteristics. For our dataset, I scraped data from the Carvago website, a popular platform for buying used cars. Based on this data, a predictive model was created.

For the machine learning implementation, we evaluated several algorithms including Linear Regression, Ridge Regression, Lasso Regression, ElasticNet, Random Forest Regressor, Decision Tree Regressor, AdaBoost Regressor, Gradient Boosting Regressor, and XGB Regressor. The performance of each algorithm was evaluated using mean squared error (MSE) and R^2 scores. Our results showed that XGB Regressor and Random Forest Regressor performed better at predicting prices.

1. Introduction

1.1. Background & Related Work

Have you ever tried to buy or sell a used car? If so, you've probably faced the challenging question: "What's the right price?" This question has become increasingly complex in today's automotive market, where a car's value depends on countless factors beyond just its age and mileage.

Consider this scenario: Two identical cars, same model, same year, but one costs significantly more than the other. Why? The answer lies in the intricate web of features that influence a car's value - its maintenance history, specific options, market demand, and even color can all play crucial roles. This complexity makes it difficult for both buyers and sellers to determine fair prices, often leading to lengthy negotiations and potential mistrust in the marketplace.

The challenge has become even more relevant recently. With new car prices reaching record highs and supply chains facing disruptions, more people are turning to the used car market. However, traditional pricing methods - whether it's checking local listings or relying on basic depreciation calculations - often fall short in capturing the true market value of a vehicle.

This is where machine learning enters the picture. Think of it as having an expert who can instantly analyze thousands of car sales, identifying patterns that humans might miss. Our research harnesses this power by developing a data-driven approach to predict used car prices accurately. Using data collected from Carvago, a popular used car marketplace, we've built models that can process numerous vehicle characteristics simultaneously to estimate market values.

Why does this matter? Because accurate price predictions can benefit everyone in the used car ecosystem:

- Buyers can shop with confidence, knowing they're paying a fair price
- Sellers can list their cars competitively, avoiding the pitfall of over- or under-pricing
- Dealerships can make better inventory decisions
- Banks and insurance companies can provide more accurate vehicle valuations

By bringing together machine learning and automotive expertise, our work aims to make the used car market more transparent and efficient for all participants. After all, whether you're buying your first car or managing a dealership's inventory, having accurate price predictions can make a significant difference in making informed decisions.

1.2. Objective

Our goal is to predict car prices in Europe, addressing the challenge of determining fair prices for both buyers and sellers. This initiative has two main objectives: First, to build a precise prediction model utilizing machine learning techniques based on extensive data from the European car market.
Second, to develop a useful tool that assists market participants in making well-informed decisions regarding car valuations. By applying different machine learning algorithms and examining essential vehicle features, we aim to offer a data-driven approach that minimizes pricing uncertainty in the European used car market.

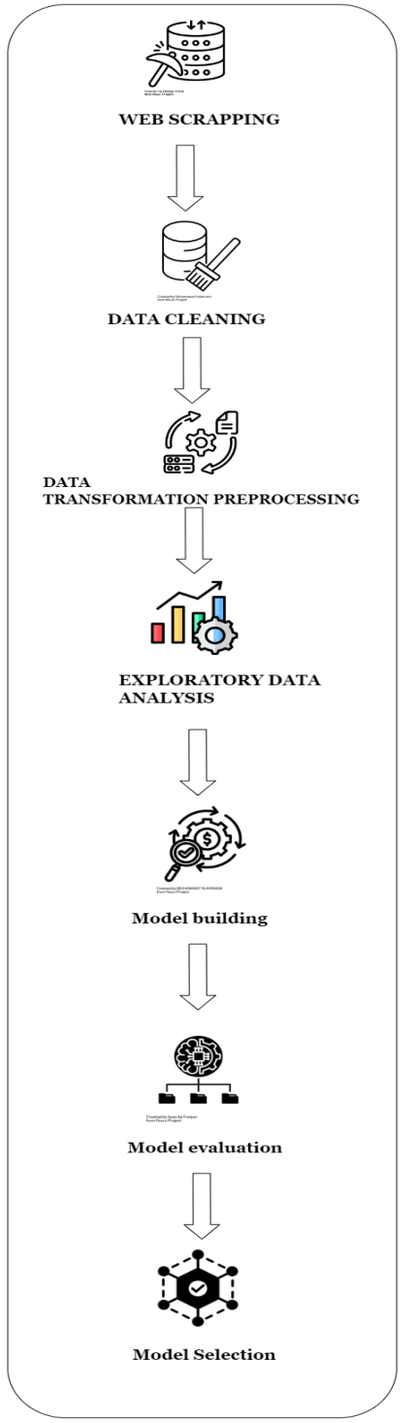

Fig 1. Proposed methodology

2. Data

Our data has been collected from the website "Carvago.com". In total there are 30001 cars in the first file and another one for recommendation with 47110 cars with variable links that don't exist in the first file, in this table we show the names of the variables and their corresponding meaning.

Variable name	Description
Mileage	The total distance the vehicle has traveled.
First_registration	The date when the vehicle was first registered for use.
Power	The engine power of the vehicle.
Transmission	The type of gearbox, such as manual, automatic.
Fuel	The type of fuel the vehicle uses, such as petrol, diesel, electric, or hybrid.
Drive_type	The drivetrain of the vehicle. (4x4 or 4x2)
Location	Location where the vehicle is available.
Vehicle_ID	A unique identifier for the vehicle.
Make	brand of the vehicle.
Body_color	The exterior color of the vehicle.
Type_of_finish	The type of paint finish
Interior_color	The color of the vehicle's interior.
Interior_material	The material used for the interior.
Body	The body type of the vehicle.
Doors	The number of doors on the vehicle.
Seats	The seating capacity of the vehicle.
Consumption	The fuel consumption rate.
CO2_emissions	The carbon dioxide emissions produced by the vehicle.
Engine_capacity	The carbon dioxide emissions produced by the vehicle.
Emission_class	The vehicle's compliance with emission standards, e.g., Euro 6, Euro 5.
Speeds	The number of gears or speed settings in the transmission.
Price	The total price of the vehicle
Price_without_vat	The price of the vehicle excluding VAT.
Links	Links to a web site.

Table 1. Names and description of each variable.

3. Methodology

3.1. Data preprocessing and transformation

First, we removed non-essential columns such as Vehicle_ID, Location, and price_without_vat, as they had no direct impact on price prediction. We also dropped columns with excessive missing values (Speeds

and Type_of_finish). And removed rows with missing value in critical features like Milelage, Doors and Body color , and standardaized the numerical features, as they contained various units. Milelage measured in 'km', power with 'hp' and price with '€'; We cleaned these by removing units and converting all values to appropriate numeric formats. To improve model reliability, we filtered out car makes with less than 10 instances in our dataset. This helped us avoid potential bias from brands represented in too few examples. We normalized all the fuel consumption values to one common unit, namely, L/100km, so that the comparisons are possible between different types of fuel. (at the least I dropped this column because I didn't trust this standardization of KWH/100km to L/100km even is include in the analysis)

For missing values in other features:

- Interior features (material and color) were filled with the most common values.
- Missing seat numbers were set to 5 (the most common configuration).
- Registration dates were converted to just the year.
- Rare fuel types (LPG and CNG) were removed to prevent overfitting.

- we handled missing technical specifications (CO2 emissions, engine capacity) using median values based on fuel type and power groups.

For missing value in emission class I did some analysis depends of the type of fuel and Year of registration and I fill nan value in it.

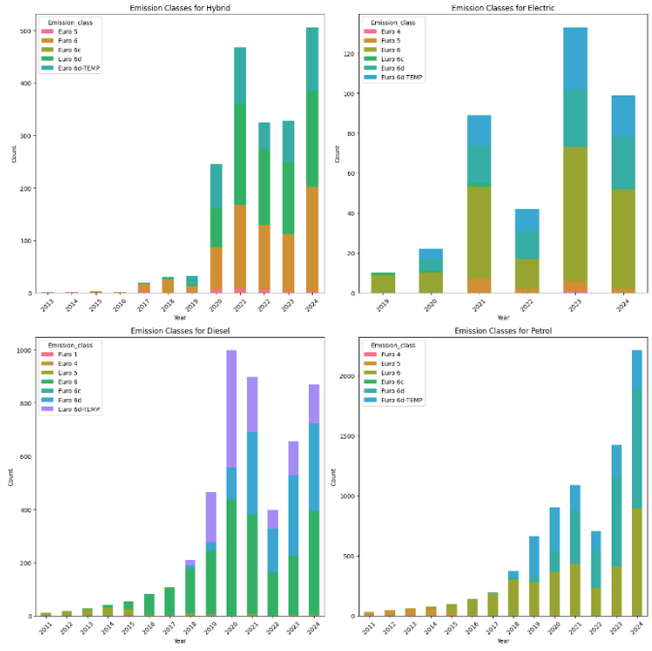

Fig 2. Distribution of Emission class depends of type of fuel and year

For each group, we filled missing emission class values using the most common class within that specific fuel type and year combination .

Below are some of the transformation and feature engineering processes that were done to cleanse the dataset:

- We applied one-hot encoding to convert categorical features (Make, Interior color, Body color, Body type, and Interior material) into binary columns suitable for machine learning analysis.
- Remove cars with 6/7 doors because they are so Rare.
- We removed cars with Euro 1, Euro 4, and Euro 6c emission classes, as they are very rare in the dataset and could bias the results.
- Mapping categorical values of Doors, Transmission, Fuel, and Drive_type columns

into numerical representations for easier processing.

- Cars with atypical or very infrequent seat configurations in the dataset were removed to avoid bias in the analysis.
- Duplicates values were removed.

3.2. Data Analysis & EDA

3.2.1 Numerical Features

We initiated our data analysis phase by seen The distribution of numerical features

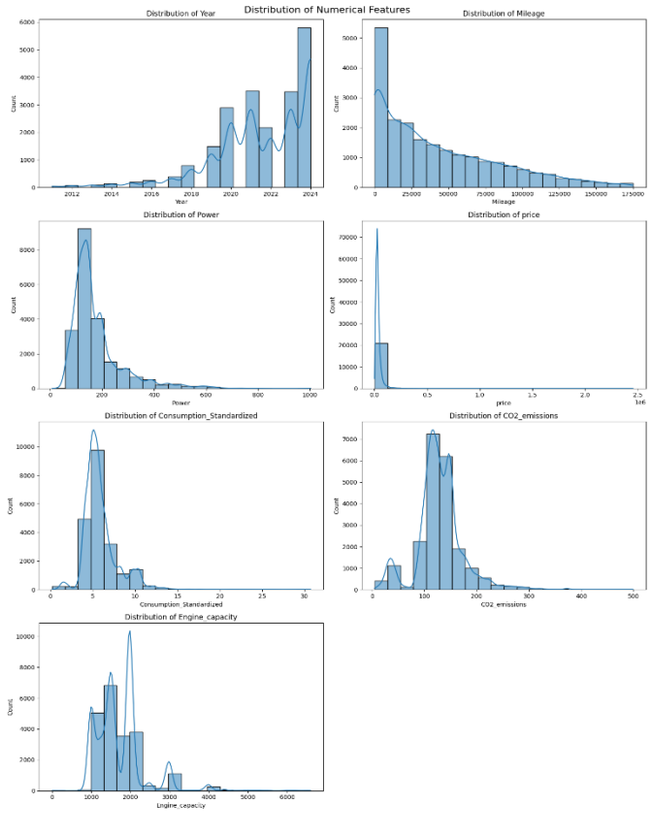

Fig 3 Distribution of numerical features

We can see that there are some outliers in our data, so we will use the boxplots for a good visualisation and investigation.

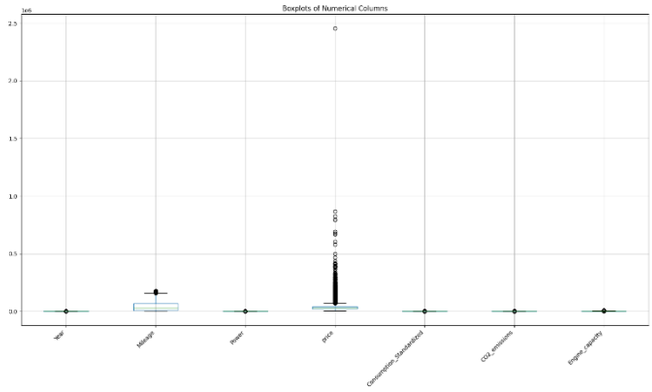

Fig 4 Boxplots of numerical features

Year

- Distribution: Most of the data is concentrated around the median, with little variability.
- Outliers: A few values lie outside the whiskers, indicating atypical years of manufacture (perhaps very old or very recent compared with the majority of cars in the dataset).

Mileage

- Distribution: Vehicle mileage is relatively concentrated around the median.
- Outliers : A few vehicles have extremely high or low mileage compared to the majority, which could indicate very worn or very little used vehicles.

Power

- Distribution: The power of the vehicles is concentrated around the median, with a few extreme values.
- Outliers: There are some very high or very low power values, suggesting extremely powerful or underpowered vehicles.

Price

- Distribution: The price distribution shows great variability, with many extreme values.
- Outliers: A significant number of vehicles have prices much higher than the majority, reaching up to around 2.5 million, which could indicate luxury cars .

Engine_capacity

- Distribution: Engine capacity shows a tight distribution around the median.
- Outliers: Some extreme values for engine capacity, suggesting vehicles with exceptionally large or small engines.

After that, we're going to study the relationship between numerical column to understand the strength and direction of the linear relationship between variables.

Fig 5. A heatmap of numerical variable

All the columns in our data seem to play an important role and are significant in understanding the relationship between vehicle characteristics. We note a low negative correlation (-0.73) between the variables 'year' and 'mileage', while we note a high correlation (+0.77) between 'power' and 'engine capacity'; we cannot remove either of these, as price and mileage are very important columns in the characteristics of a vehicle.

Power and engine capacity, which are highly correlated with price. If one of these columns is removed, there could be a loss of information that explains the price.

The correlation coefficient is calculated using the following formula:

$$r_{xy} = \frac{\sum_{i=1}^{n}(xi-x)(yi-\bar{y})}{\sqrt{\sum_{i=1}^{n}(xi-x)^2(yi-\bar{y})^2}}$$

- $(r_{xy})$is the Pearson correlation coefficient between x and y,
- (x_i) and $(y_i)$are the individual sample points indexed with (i),
- $(\bar{x})$ and $(\bar{y})$ are the mean of (x) and (y) respectively,
- n is the total number of samples.

3.2.2 Categorical features :

The distribution of some categorical features

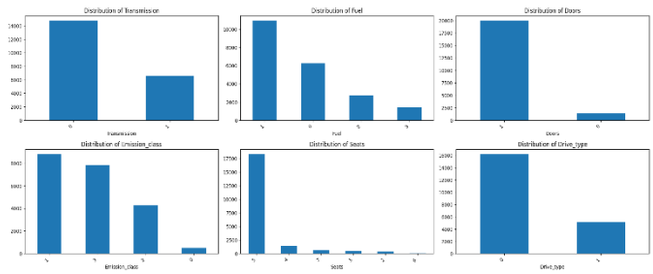

Fig 6. Distribution of categorical column

"Transmission”,"Fuel", "Doors", "Emission_class", "Seats", and "Drive_type." The distribution looks logical and well-balanced for most of these categories. Just in the distribution of seats we have a lot of values of 5 seats then other .

Know we gonna study the correlation between categorical columns that have already been one-hot encoded and price and we will remove the categorical column that they had a low correlation with the price view this figures

Of some categorical that will removed because they are low corelated with the price

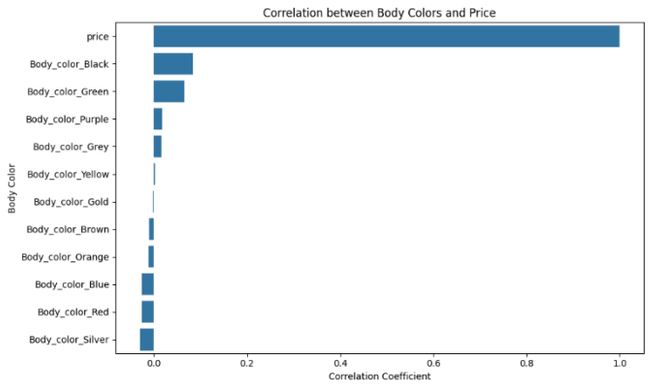

Fig 7. Correlation between body color and price

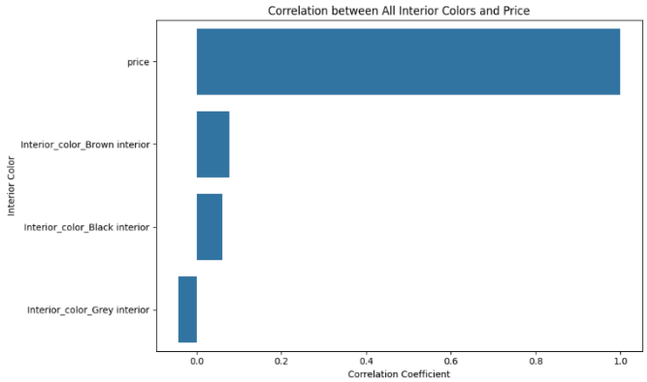

Fig 8. Correlation between Interior colors and price

This two column “Body colors” and “interior colors“ will be removed.

3.2. Model Selection

In addressing the used car price prediction regression problem, we employed various algorithms to identify the most suitable one for this task and we split data into training and testing sets with a test size of 0.2. then we used a BayesSearchCV to find the best parameter of each model.

In the following step, we will dive into the fundamental concepts of each algorithm.

3.2.1. Linear regression

Linear Regression is a foundational algorithm in supervised learning used for predicting a continuous target variable. It assumes a linear relationship between the input features X and the target variable y. The model is defined as:

$$y = \beta 0 + \beta 1 * x1 + \beta 2 * x2 + \ldots + \beta n * Xn + \epsilon y$$

where **β** represents the coefficients, and **ϵ** epsilon is the error term. The objective of linear regression is to minimize the sum of squared residuals (errors) between the predicted and actual values, often solved using Ordinary Least Squares (OLS). Regularization techniques like Lasso and Ridge are applied to improve generalization and handle multicollinearity.

3.2.2. Decision Tree Regression

Decision Tree Regression (DTR) is a powerful supervised learning algorithm predominantly used for solving regression problems. It operates by partitioning the input space into regions, and then fitting a simple model (like a constant) in each one. Key parameters such as max_depth, min_samples_leaf, and min_samples_split control the complexity and size of the tree and can be tuned to prevent overfitting and underfitting. These parameters make DTR a flexible and interpretable model for regression tasks.

3.2.3. Random Forest Regression

Random Forest Regression is a meta-estimator that fits a number of decision trees on various sub-samples of the dataset and uses averaging to improve the predictive accuracy and control overfitting. It’s an extension of the bagging method and utilizes both bagging and feature randomness to create an uncorrelated forest of decision trees. At each node during the construction of the tree, a random subset of z features is selected out of the total n features. Moreover, using bootstrapping or

other methods, it also selects p examples from the m total available examples at each node.

The key parameters of a Random Forest include:

- n_estimators: Specifies the number of trees in the forest.
- criterion: Determines the function to measure the quality of a split.
- max_depth: Controls the maximum depth of the tree.
- min_samples_split and min_samples_leaf: Manage the minimum number of samples required to split an internal node and to be at a leaf node, respectively.

3.2.4. Gradient Boosting Regression

Gradient Boosting is a technique that progressively builds a sophisticated regression model, referred to as H, by incorporating simple models in an iterative manner. Its principle is the following:

1. Train a simple model **h0** on the training data **(xi,yi)**
   Set **H=h0**
   Calculate the residuals **{r1,r2,…,rm}** for **H.**
2. Fit a model **h1** to the residuals (using the examples (**xi,ri**)).
3. Update H to be **H = h0 + λ*h1.**
4. Calculate the residuals **ri = ri – λ*h1(xi)** where **λ** is the learning rate.
5. Repeat steps 2-4 until a stopping condition is met.

Gradient Boosting effectively balances bias and variance and works well with hyperparameter tuning, such as adjusting the learning rate or tree depth.

3.2.5 XGBRegressor :

The **XGBRegressor** is a regression implementation of the Gradient Boosting framework, developed by the XGBoost library. It is known for its efficiency, scalability, and performance. XGBoost introduces innovations like:

- **Tree Pruning**: Uses a depth-first approach with cost complexity pruning to control overfitting.
- **Handling Missing Data**: Automatically learns the best direction for missing values in splits.
- **Regularization**: Includes L1 and L2 regularization for better generalization.

Key parameters include:

- learning_rate: Controls the weight of each weak model.
- max_depth: Limits the maximum depth of trees.
- n_estimators: Specifies the number of boosting rounds.
- subsample: Determines the fraction of samples to be used for fitting individual models.

3.2.6 Lasso Regression :

Lasso Regression, or **Least Absolute Shrinkage and Selection Operator**, is a linear model enhanced with L1 regularization. It minimizes the residual sum of squares while constraining the sum of the absolute values of the coefficients:

Objective: $\min\{\|y-X\beta\|^2+\alpha\|\beta\|^2\}$
The L1 penalty induces sparsity by shrinking some coefficients to zero, making Lasso a powerful tool for feature selection in high-dimensional datasets.

3.2.7 Ridge Regression :

Ridge Regression applies L2 regularization to Linear Regression, penalizing the sum of the squared coefficients to reduce model complexity:

Objective: min{‖y−Xβ‖²+α‖β‖²}

This technique helps mitigate issues of multicollinearity and overfitting by shrinking coefficients, but unlike Lasso, it does not enforce sparsity.

3.2.8 ElasticNet :

ElasticNet combines the strengths of both Lasso (L1) and Ridge (L2) regression by adding a weighted sum of the two penalties to the objective function:
Objective: min{‖y−Xβ‖²+αρ‖β‖+α((1−ρ)/2)‖β‖²}
The parameter **ρ** controls the mix between L1 and L2 regularization, making ElasticNet ideal for datasets with correlated predictors and feature selection requirements.

3.2.9 AdaBoostRegressor :

AdaBoostRegressor uses the Adaptive Boosting methodology for regression tasks. It sequentially trains weak learners, typically decision trees, where each successive learner focuses more on the samples that were mispredicted by previous ones. The ensemble prediction is the weighted sum of the predictions from all learners.

Key parameters include:

- n_estimators: Controls the number of weak learners in the ensemble.
- learning_rate: Adjusts the contribution of each learner to the final prediction.

AdaBoost is effective in reducing both bias and variance, making it a versatile tool for regression problems.

4. Results

4.1. Evaluation Metrics

The evaluation metrics used in this study are the Mean Squared Error (MSE) and the Coefficient of Determination, denoted as R^2. These two metrics are explained as follows:

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

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

Where y_i represents the observed values, $\hat{y}_i$ represents the predicted values, $\bar{y}$ represents the mean and n is the total number of observations.

4.2. Models Performance

After the BayesSearchCV and model training, we evaluated the models' performance using the R^2 and MSE metrics. **Fig 9** illustrate the results obtained for each model.

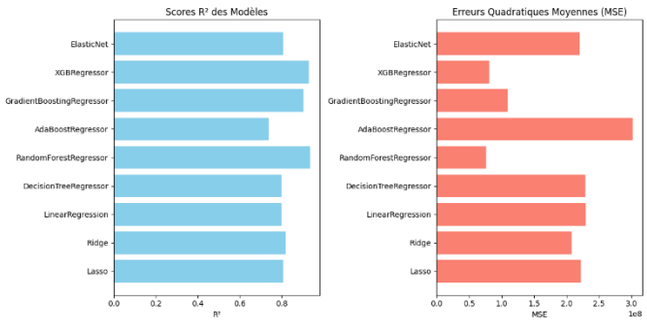

Fig 9. R^2 and MSE comparison of the hyperparametertuned models

Among all the models evaluated, the **RandomForestRegressor** emerged as the best-performing model, achieving the lowest Mean Squared Error (MSE) of **76,069,304.66** and the highest R^2 score of **0.9336**. Its optimal parameters included a maximum depth of 20, a minimum samples split of 2, and 50

estimators. This demonstrates the model's ability to effectively

capture complex patterns in the data while maintaining generalization. Its performance surpassed other models, such as the XGBRegressor and GradientBoostingRegressor, which also showed strong results but with slightly higher errors and lower R^2 scores.

5. Discussion & Limitations

The evaluation of our models highlighted the strong performance of ensemble methods, particularly the RandomForestRegressor, which achieved the best predictive accuracy. However, the scope of this analysis is constrained by the dataset's coverage. At present, the dataset may not encompass information from **all countries worldwide**, **all car models**, or **all fuel types**, which limits the model's generalizability and applicability to diverse contexts.

To develop a truly universal model, it is essential to integrate data from **every country**, ensuring a broad representation of regional variations in automotive trends, policies, and infrastructure. Additionally, incorporating **all car models**, ranging from compact cars to luxury vehicles and commercial trucks, would provide a more holistic view of the automotive landscape. Furthermore, including **all fuel types**—such as gasoline, diesel, electric, hybrid, hydrogen, and alternative fuels—would enable the model to account for differences in energy efficiency, environmental impact, and adoption trends across technologies.

6. Conclusion

This paper explores the development of machine learning models for predicting used car prices using a data-driven approach. Leveraging a comprehensive dataset, the study identifies key factors influencing car prices and evaluates multiple regression techniques to construct an accurate predictive model. The methodology encompasses data preprocessing, exploratory analysis, and the application of advanced machine learning algorithms.

References :

[1] : Jiang, Xianshun. *Research for Car Price Prediction Base on Machine Learning*. Rose-Hulman Institute of Technology, Terre Haute, IN, 47803, the United States, Jiangx6@rose-hulman.edu.

[2] : Li, Chenguang. *Machine Learning-Based Models for Accurate Car Prices Prediction*. Department of FinTech, University College Dublin, Dublin, Ireland. li.chenguang@ucdconnect.ie

[3] : Gao, Jiaying. *Second-hand Car Price Prediction Based on Multiple Linear Regression and Random Forest*. School of Mathematics and Artificial Intelligence, Chongqing University of Arts and Sciences, Chongqing, 400000, China.