INFO-H410 Techniques of artificial intelligence

Used Car Price Estimator

Rida Belkhiri - (000609769) - rida.belkhiri@ulb.be

Imad El Harrouti - (000609770) - imad.el.harrouti@ulb.be

El Mamoune Benmassaoud - (000608995) - el.mamoune.benmassaoud@ulb.be

2024-2025

Contents

Introduction

The main aim of this project is to predict the price of used vehicles based on their technical characteristics (mileage, power, fuel type, etc.). This prediction can help users to assess whether the price proposed for a vehicle is fair or not, and thus avoid potential scams when buying or selling.

The idea is to build a machine learning model capable of giving a realistic estimate of the price of a given vehicle. To achieve this, we used a set of real data found on Kaggle from the AutoScout24 platform, bringing together several tens of thousands of car advertisements.

The project consists of several stages: loading and exploratory analysis of the data, cleaning and pre-processing, selection and training of several models, then performance evaluation and implementation via a user interface.

1 Data Visualization

1.1 Dataset Selection

Initially, we considered using a dataset containing more than 250,000 rows from the AutoTrader Canada website [2]. Although very large, this dataset had many missing values in key columns essential to our analysis, such as the vehicle's year, the number of previous owners (*dealer*), and other important technical specifications. Moreover, several key attributes that influence the price of a vehicle such as fuel type, transmission, or body type were absent. After evaluation, we concluded that this dataset was not complete enough and not exploitable for the purpose of our project.

We then opted for a second, higher-quality dataset from Kaggle [1], containing around 16,000 rows. Although smaller, this dataset provides rich and diverse information about the vehicles:

Column	Description
make_model	Make and model of the vehicle.
body_type	Body type of the vehicle.
price	Price of the vehicle.
vat	Indicates whether VAT is included in the price.
km	Vehicle mileage.
type	Type of vehicle.
fuel	Fuel type.
gears	Number of gears.
comfort_convenience	Comfort equipment.
entertainment_media	Multimedia equipment.
extras	Additional options.
safety_security	Safety equipment.
age	Vehicle age in years.
previous_owners	Number of previous owners.
hp_kw	Engine power in kilowatts.
inspection_new	Recent technical inspection.
paint_type	Type of paint.
upholstery_type	Type of upholstery.
gearing_type	Transmission type.
displacement_cc	Engine displacement in cm^3.
weight_kg	Vehicle weight in kg.
drive_chain	Type of drivetrain.
cons_comb	Combined fuel consumption.

Table 1.1: Description of the columns in the AutoScout24 dataset

1.2 Initial Data Loading and Cleaning

The raw dataset was imported using `pandas`. A cleaning step was carried out to harmonize column names, convert numerical variables to the correct format, and handle missing values:

- Rows with missing values in essential columns (`price`, `km`, `age`) were removed.
- Median imputation was used for missing values in remaining numerical columns.
- Column names were standardized (spaces removed, converted to lowercase, replaced with underscores).

After these operations, the data was ready for exploratory analysis.

1.3 Exploratory Analysis

We began by visualizing the distribution of vehicles by make and model (`make_model`). A bar plot was used to identify the most represented models. To ensure statistical robustness for further analysis, only models appearing at least 100 times were retained.

A crosstab analysis between the `make_model` and `body_type` columns was then used to study the available body types for each model. Rare combinations (fewer than 10 occurrences) were removed to reduce noise in the dataset.

1.4 Export of Cleaned Dataset

The final dataset, after cleaning and filtering, contains approximately 15,820 rows. It was saved in a file named `autoscout_clean.csv` for use in subsequent modeling steps.

2 Exploratory Data Analysis (EDA)

After the initial cleaning phase, we performed an exploratory data analysis to understand the structure of our dataset and identify patterns, correlations, and potential issues before model training.

2.1 Price Distribution and Transformation

The raw price distribution was highly skewed to the right, indicating the presence of luxury vehicles with very high prices. To better understand the distribution and reduce the influence of outliers, we applied a logarithmic transformation to the `price` column.

- We visualized both the original and log-transformed price distributions.
- The `log(price)` plot appeared more symmetrical and suitable for regression models.

2.2 Feature Relationships

We explored how various numerical and categorical variables relate to vehicle price (and log-price).

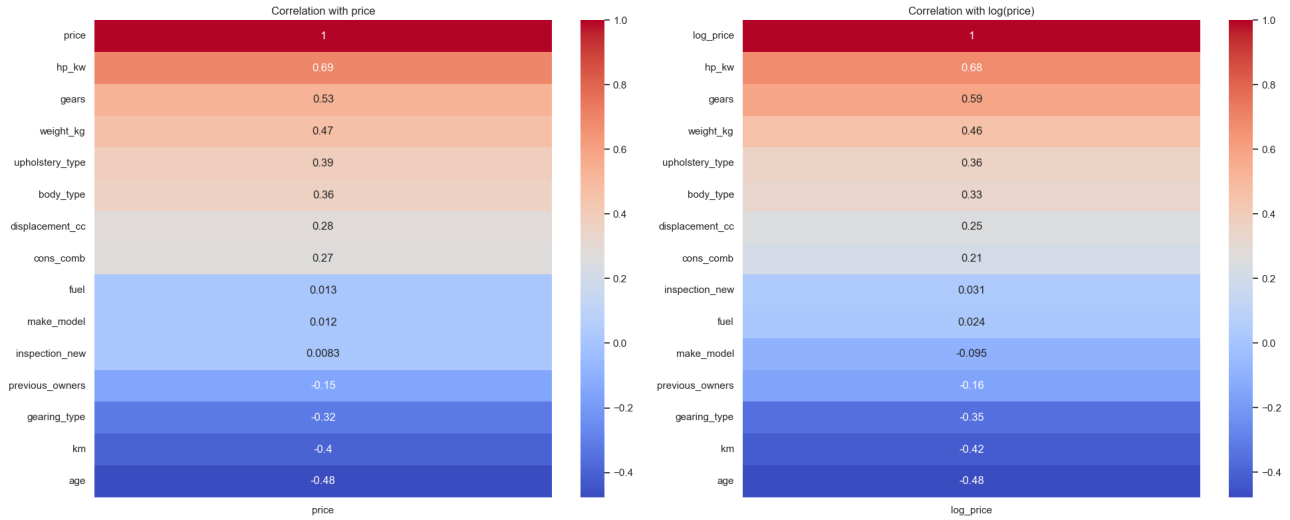
- **Mileage:** A negative correlation was observed; vehicles with higher mileage tend to cost less.
- **Age:** Using a boxplot, we confirmed that older vehicles generally have lower prices.
- **Power (kW):** Scatterplots revealed a mild positive correlation with price.

2.3 Correlation Analysis

To quantify the strength of relationships between features and price, we computed a Pearson correlation matrix.

- We encoded categorical and boolean columns using `LabelEncoder` before correlation.
- A full 23x23 correlation heatmap was generated.

- We compared correlations with both `price` and `log(price)` to highlight relevant predictors.



2.4 Final Feature Selection

Based on the EDA, we selected a subset of relevant features for modeling:

- Numerical: `age`, `km`, `hp_kw`, `gears`, `weight_kg`, `displacement_cc`
- Categorical: `body_type`, `upholstery_type`, `gearing_type`, `fuel`, `make`, `model`
- Target: `log_price`

The dataset was then one-hot encoded for the categorical columns, and the input/output matrices `X` and `y` were exported for model training.

This exploratory phase helped us identify important relationships, mitigate the impact of skewed variables, and prepare our data for effective learning.

3

Model Evaluation and Final Selection

3.1 Initial Model Comparison

After training on a cleaned dataset, five regression algorithms were compared: Linear Regression, Ridge Regression, Decision Tree, Random Forest, and XGBoost. The following table summarizes their performance:

Model	**MAE (€)**	**RMSE (€)**	**R^2**	**R^2 CV Mean**	**R^2 CV Std**
Random Forest	**1033.59**	**1711.85**	**0.9454**	0.4076	0.2703
XGBoost	1086.27	1721.63	0.9448	**0.5600**	**0.2089**
Decision Tree	1205.98	2065.55	0.9205	0.0593	0.6955
Linear Regression	1636.66	2537.28	0.8801	0.7221	0.1708
Ridge Regression	1636.48	2537.80	0.8801	0.7231	0.1696

Table 3.1: Initial regression model performance

Random Forest and XGBoost clearly outperform the others in terms of low MAE and RMSE, and high R^2. However, their higher cross-validation standard deviation suggests they are more sensitive to training set variations.

3.2 Hyperparameter Optimization with GridSearchCV

A targeted hyperparameter search was conducted using GridSearchCV on XGBoost, with 5-fold cross-validation.

- **Best hyperparameters found:** :
  {colsample_bytree: 0.8, learning_rate: 0.05, max_depth: 3, n_estimators: 300, subsample: 1.0}

3.3 Before/After Grid Search Comparison

Metric	Before GridSearch	After GridSearch
MAE (€)	**1086.27**	1237.24
RMSE (€)	**1721.63**	1906.86
R^2 (test set)	**0.9448**	0.9323
R^2 CV Mean	0.5600	**0.7174**
R^2 CV Std	0.2089	**0.1262**

Table 3.2: XGBoost performance before and after hyperparameter optimization

3.4 Discussion and Final Model Selection

Although the test-set performance slightly decreased after optimization (higher MAE and RMSE), the optimized model shows:

- **Improved generalization stability** (lower R^2 CV Std),
- **Higher average cross-validation performance**, indicating increased robustness,
- Overall, still very competitive performance.

Conclusion: We select the **optimized XGBoost model** as our final model. It offers the best trade-off between *accuracy* and *generalization*, and will be used for deployment and result interpretation.

Although further improvements could still be pursued (e.g., using advanced optimization methods or adding features such as vehicle options or equipment), the current cost-benefit ratio does not justify more complex work at this stage. The selected model already provides sufficiently accurate predictions for reliable vehicle price estimation.

4

Web Interface Implementation with Streamlit

To provide an interactive experience for users, we developed a web interface using **Streamlit**, allowing real-time estimation of car prices based on vehicle characteristics.

4.1 Dynamic Input Filtering

- The interface starts by extracting available **brands** and **models** directly from the dataset.
- Models are filtered dynamically based on the selected brand to prevent invalid combinations.
- Similarly, categorical features such as *body type*, *fuel type*, *transmission*, and *upholstery* are restricted to values that actually exist for the chosen brand/model.

4.2 User Input

The user is asked to provide:

- Technical specifications: age, mileage, engine power, weight, displacement, number of previous owners.
- A checkbox to indicate whether the car recently passed a technical inspection.

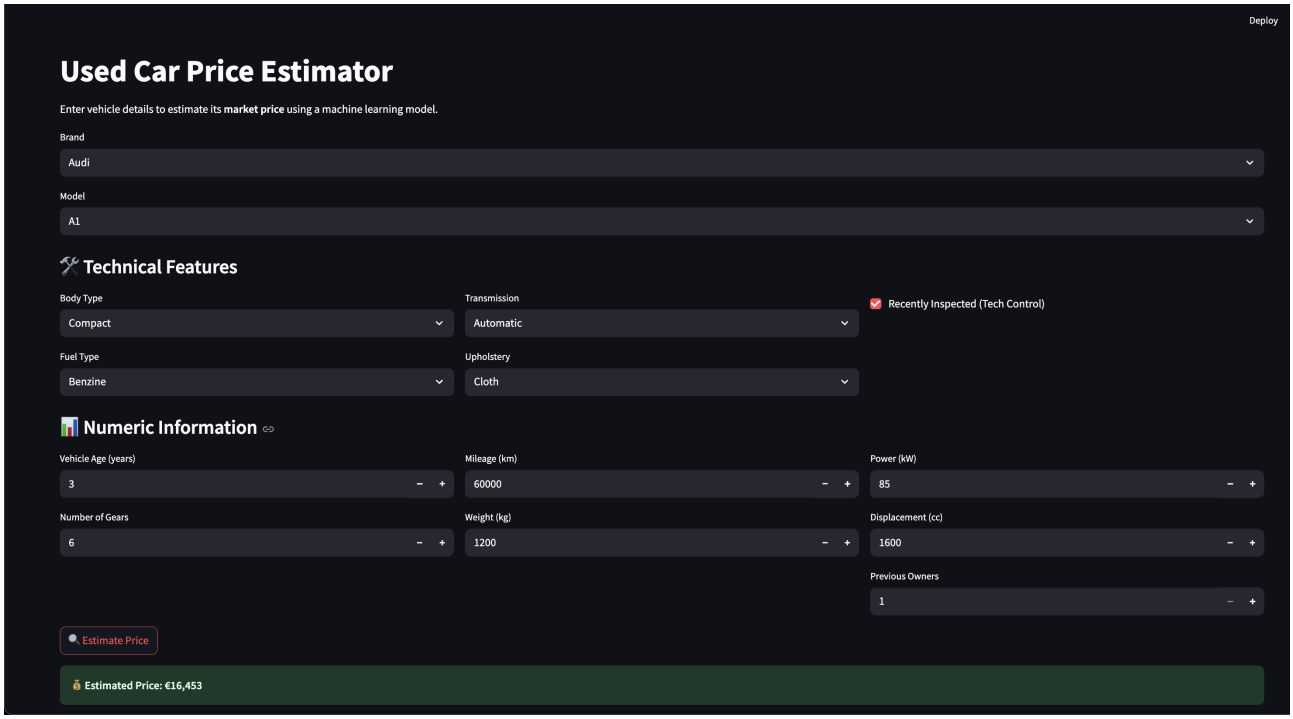
4.3 Prediction Pipeline

1. All user inputs are compiled into a structured data row.
2. Categorical values are **one-hot encoded** based on the training features.
3. Any missing features (not selected by the user) are added with default values to match the model's expected input.
4. The trained model predicts the *log-price*, which is then exponentiated to return the final estimated market value.

4.4 Output Display

The estimated price is presented in a clear and user-friendly format, offering instant feedback and enhancing the overall experience.

This interface played a key role in making the model usable by non-technical users, while strictly adhering to the data constraints established during training.



Conclusion

In this project, we developed a machine learning model to estimate the price of used vehicles based on their technical characteristics. This tool can assist users in evaluating the fairness of a proposed price when buying or selling a car, thus reducing the risk of scams or overpayments.

We began by selecting and cleaning a high-quality dataset from the AutoScout24 platform, retaining around 15,820 entries after preprocessing. We performed exploratory analysis to understand key features influencing car prices and applied various data transformation techniques to prepare the data for modeling.

Multiple models were trained and evaluated, including linear regression and tree-based methods. The most accurate model was selected based on its performance metrics and displayed via a user interface, allowing easy access to price predictions.

Even though the current model is limited to a subset of vehicle brands and models, and primarily covers cars from 0 to 3 years old (can predict older cars but will be less effective), the available information is already rich and detailed enough to generate accurate and realistic price estimates for a wide range of common vehicles. This level of performance makes the tool practical and effective for many real-world use cases.

Overall, this project illustrates the potential of data science and machine learning in the automotive market. Future improvements could include integrating additional features (e.g., accident history, maintenance records) or using larger and more diverse datasets to increase model robustness.

Bibliography

[] M. Maxwell. *AutoScout Data*. https://www.kaggle.com/datasets/mexwell/autoscout-data/data. Accessed: 2025-05-29. 2023.

[] Samuel Sullivan-Delgobbo. *AutoTraderCA Vehicle Listings with Price (250k)*. https://www.kaggle.com/datasets/samsullivandelgobbo/autotraderca-vehicle-listings-w-price-250k/data. Accessed: 2025-05-29. 2022.