UNIVERSITÀ DI BOLOGNA

School of Engineering

Master Degree in Automation Engineering

Optimization and Machine Learning M

PROJECT REPORT

Professor: **Andrea Lodi**

Students: Vittorio Caputo Federico Collepardo

Academic year 2024/2025

Contents

Chapter 1

Introduction

1.1 Project Overview

This project developed a robust machine learning pipeline for predicting sports car prices using the Sport_car_price.csv dataset. The process involved comprehensive data cleaning, feature engineering, exploratory analysis, model selection, tuning and evaluation. The outcome was a high-performing regression model, particularly a tuned CatBoost model, capable of accurately predicting car prices based on a wide range of attributes.

1.2 Objectives

The main objectives of this project were:

- To develop an accurate predictive model for sports car prices
- To implement a complete machine learning pipeline from data cleaning to model deployment
- To compare different regression approaches and select the optimal one
- To provide insights into the factors most influencing sports car prices

Chapter 2

Data Preprocessing and Exploration

2.1 Data Loading and Initial Inspection

The project began with loading the dataset and conducting an initial inspection of its structure, data types, and sample records. This early step helped identify non-numeric entries and placeholders within key columns such as Engine Size (L) and Torque (lb-ft) and formatting characters or non-numeric values (such as dashes, commas, plus/less-than signs) in numeric columns. These irregularities prompted a need for in-depth data cleaning and transformation.

2.2 Data Cleaning

2.2.1 Placeholder Handling

Placeholder values such as '0', '-', '', and non-numeric strings like 'Electric', 'Hybrid' etc. in column 'Engine Size (L)' were converted to NaN to standardize missing value representation or numeric values from strings that indicate both the hybrid type and the engine size in liters for hybrid vehicles were extracted.

2.2.2 String Cleaning and Type Conversion

Unwanted characters including dashes, commas, plus/minus signs were removed from numerical features and common placeholders to proper missing values ('NaN'). This allowed for the successful conversion of those fields into numeric types, critical for machine learning algorithms that require numerical input.

2.2.3 Duplicate and Outlier Treatment

Duplicate records were removed. In particular, from boxplot analysis, two 'Roadster' noise outlier models were identified since they had inconsistent and extreme values in 'Horsepower' and 'Torque (lb-ft)' columns. These two outliers were corrected by looking at other 'Roadster' entries and, after correction, they resulted in duplicates rows, so they were eliminated. The 1965 Shelby, being a significant temporal outlier among otherwise recent car models, was removed to maintain consistency. On the other hand, structural outliers, such as very high-priced vehicles, were retained as they reflect genuine real-world data. These outliers were handled using tree-based models, which are naturally more robust to such anomalies and by applying power transformation.

2.2.4 Missing Values Handling

Rows missing Torque (lb-ft) values were dropped, as they were few and unlikely to impact the model. However, missing values in the 'Engine Size (L)' column are more complex: they represent a significant portion of the data and correspond to electric cars, which do not have a traditional engine-size they are structural missing values. Dropping or imputing these with mean/median is not appropriate, so they were left as 'NaN', (to be handled by tree-based models) or, in case of Linear Regression that doesn't deal with NaNs natively, imputed with a sentinel value (e.g., 0) in conjunction with a new categorical indicator column, Engine Type.

2.3 Exploratory Data Analysis

Exploratory data analysis was conducted to better understand feature distributions and relationships.

2.3.1 Visual Analysis

Histograms, boxplots, pairplots, and correlation heatmaps were created to identify skewness, outliers, and inter-feature relationships.

2.3.2 Skewness Identification

Several numeric features including Horsepower, Torque (lb-ft), and Price (in USD) exhibited skewness. These distributions were transformed using power transformations to improve symmetry and reduce the impact of extreme values.

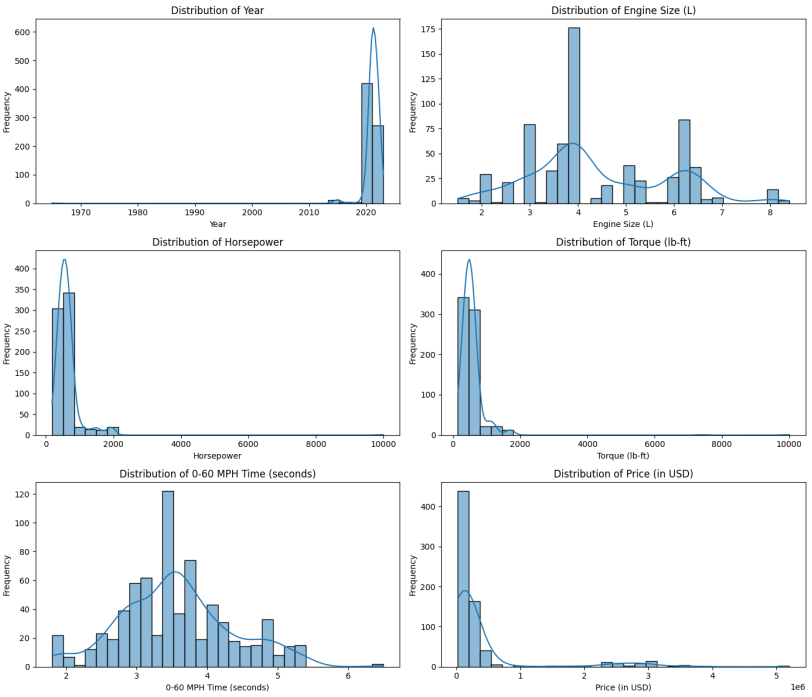


Figure 2.1: Distributions

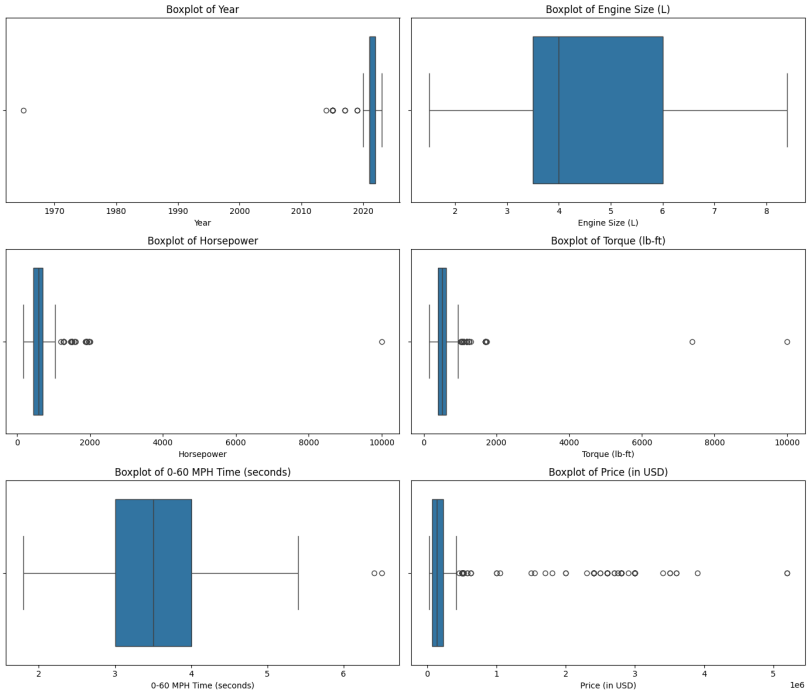


Figure 2.2: Boxplots

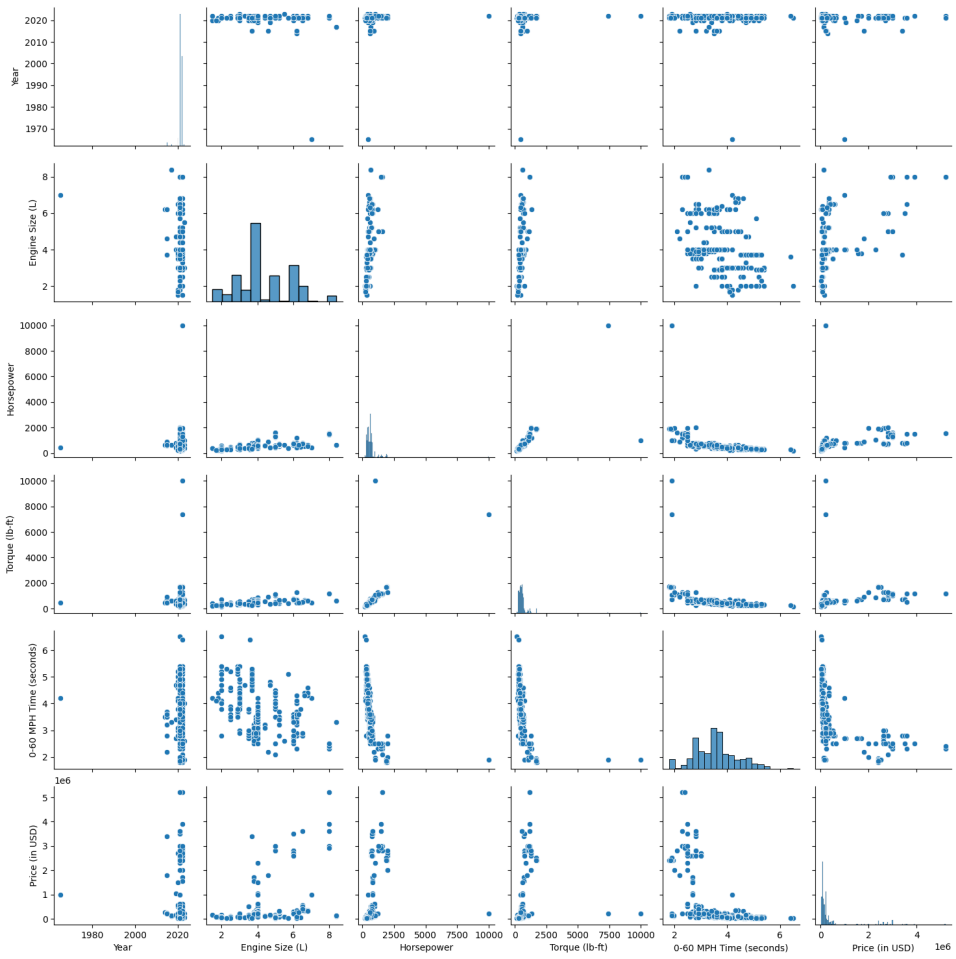


Figure 2.3: Pairplots

Chapter 3

Feature Engineering

3.1 New Feature Creation

A new categorical variable, Engine Type, was introduced to classify vehicles as Electric, Hybrid, or Combustion. This feature helped in explaining missing engine size values and capturing the technological aspect of the vehicle.

3.2 Data Transformation

To enhance model performance, power transformations (e.g., Yeo-Johnson) were applied to continuous variables with skewed distribution to reduce skewness and mitigate the influence of structural outliers.

3.3 Feature Analysis

Notably, the features 'Horsepower', 'Torque (lb-ft)', and 'Price (in USD)' after Feature Engineering exhibit distributions that are closer to normal, thanks to the applied power transformations. Additionally, it's also more evident the linear relationship between 'Horsepower', 'Torque (lb-ft)', '0-60 MPH Time (seconds)' and 'Price (in USD)' in scatterplots and correlation heatmap.
After Feature Engineering, an high correlation of 0.93 between 'Horsepower' and 'Torque (lb-ft)' features was identified, but dropping one of the two didn't lead to performance improvements, so they were both considered for prediction. The feature correlation analysis showed that 'Year' has the lowest correlation with price at -0.13, followed by 'Engine Size' at 0.38, while all other features have correlations above 0.65 with the target variable.

In addition, the categorical features 'Car Make', 'Car Model', and 'Engine Type' appeared to be quite important for predicting price. Given these observations, it was reasonable to retain all features, both numerical and categorical, in the modeling process to ensure that the model can leverage all relevant information. Furthermore, this conclusion is reinforced by additional attempts to improve performance, which did not yield better results.

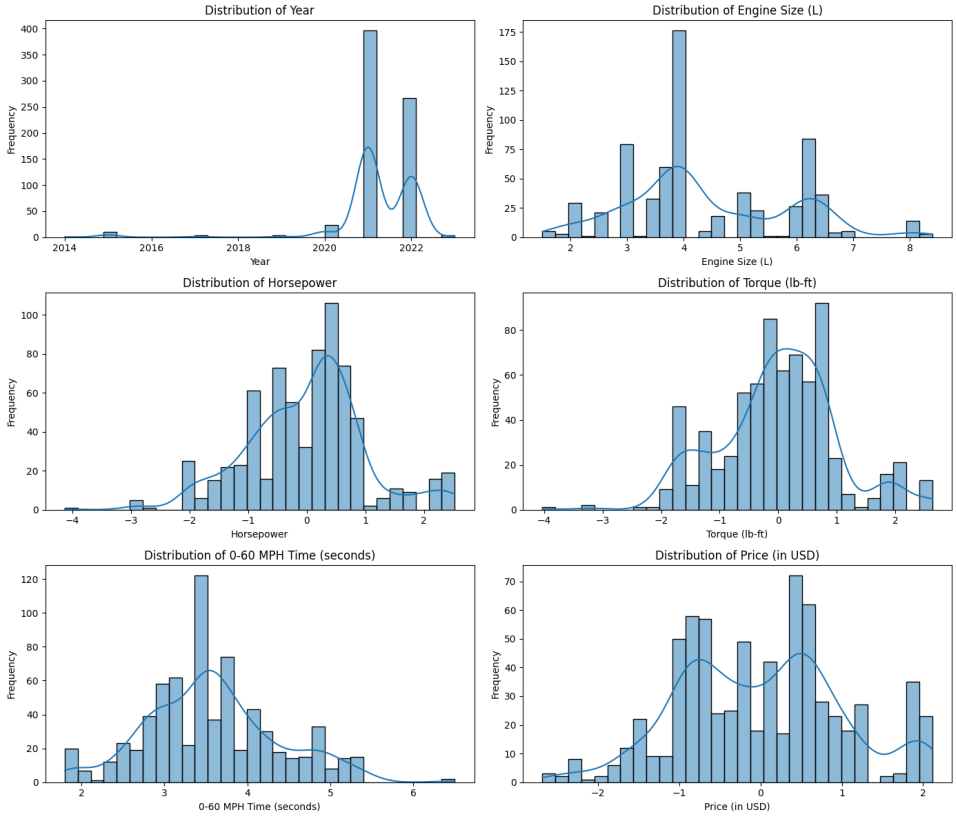


Figure 3.1: Distributions

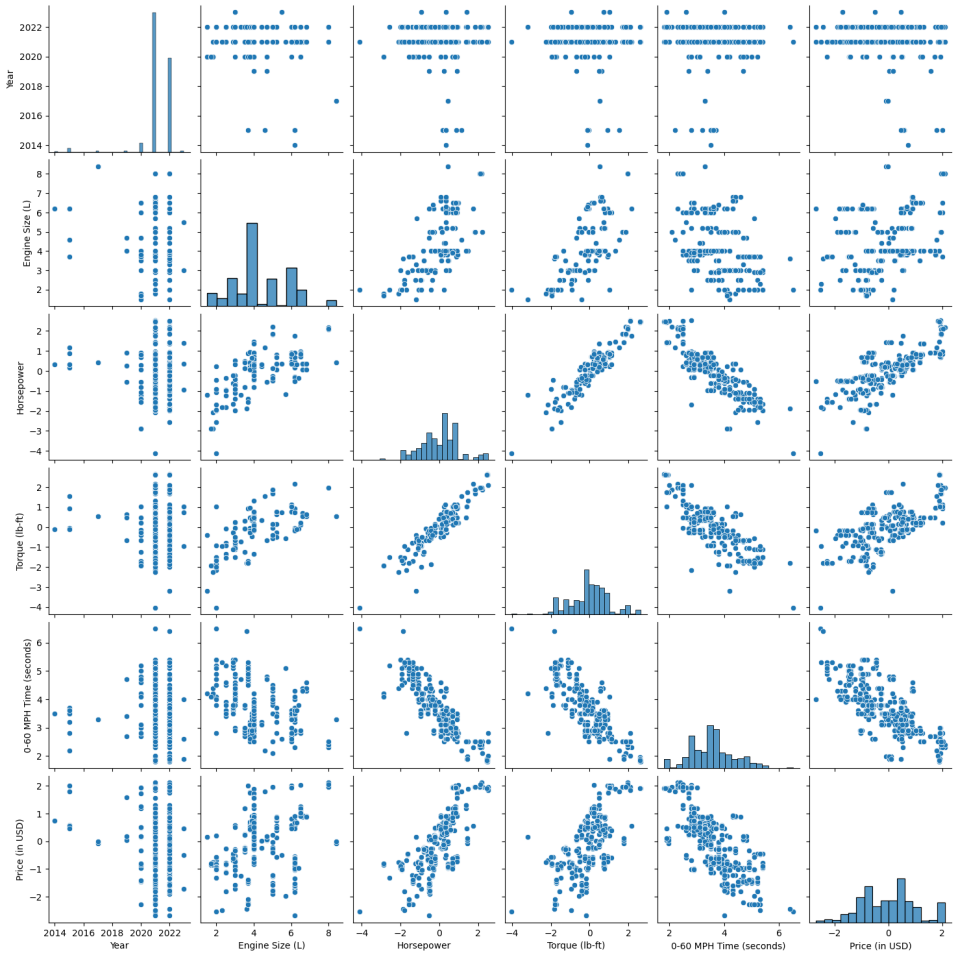


Figure 3.2: Boxplots

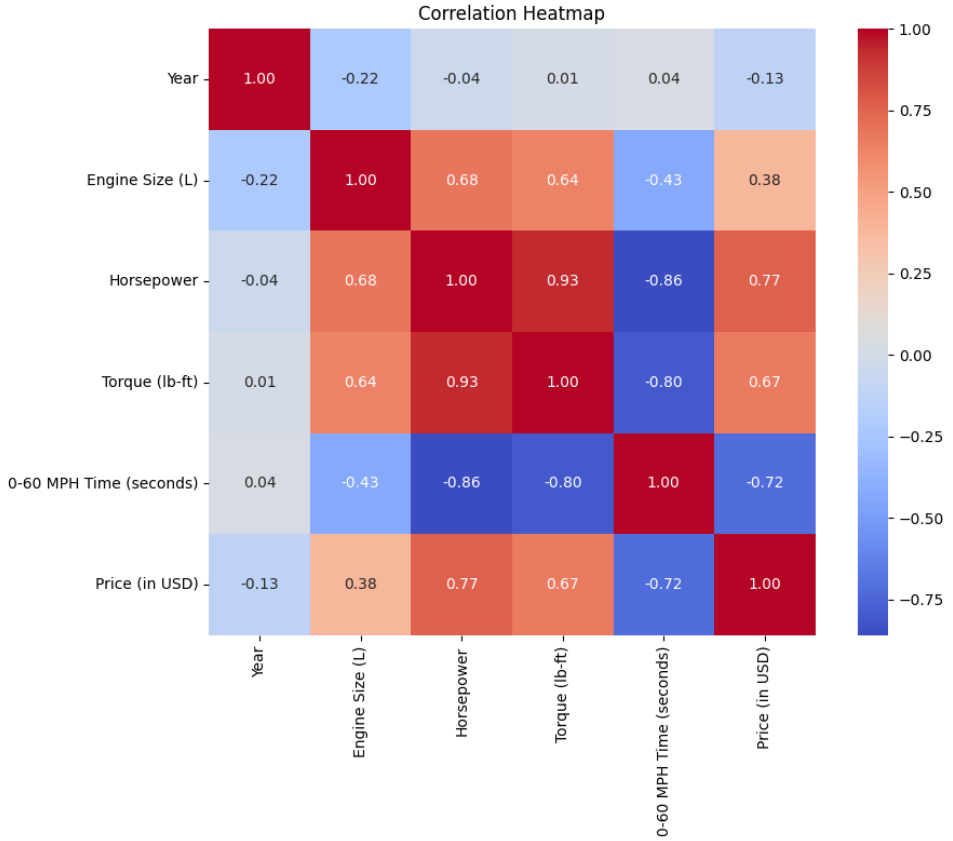


Figure 3.3: Correlation Heatmap

3.4 Encoding and Preparation

Categorical variables (Car Make, Car Model, Engine Type) were encoded using label encoding to convert them into numeric form, a prerequisite for most machine learning algorithms.
The target variable was set to 'Price (in USD)', and the rest of the columns formed the feature matrix. The dataset was split into training (70%), validation (15%), and test (15%) sets using train_test_split.

Chapter 4

Regression Model for Price Prediction

4.1 Model Selection

The following regression models were evaluated:

- **Linear Regression**
- **HistGradientBoostingRegressor**
- **RandomForestRegressor**
- **XGBRegressor**
- **CatBoostRegressor**

Each model was trained on the training set and evaluated on the validation set using R^2, Mean Squared Error (MSE), and Mean Absolute Error (MAE). Linear Regression performed poorly, achieving an R^2 score of only **0.591** on the validation set. In contrast, all tree-based models exceeded **0.8** in R^2, with CatBoost achieving the highest score **0.932**. These results underscored the superior capability of tree-based models to capture non-linear interactions and handle complex feature relationships, making them more suitable for this prediction task. The best model was selected based on the highest R^2 score on the validation set.

4.2 Hyperparameter Tuning

CatBoost, the best-performing model, was further optimized using RandomizedSearchCV with 5-fold cross-validation. The training and validation sets were combined during this step. The following hyperparameters were found to be optimal in case of 'random_state = 42' :

- learning_rate = 0.2
- iterations = 300
- depth = 5

4.3 Model Evaluation

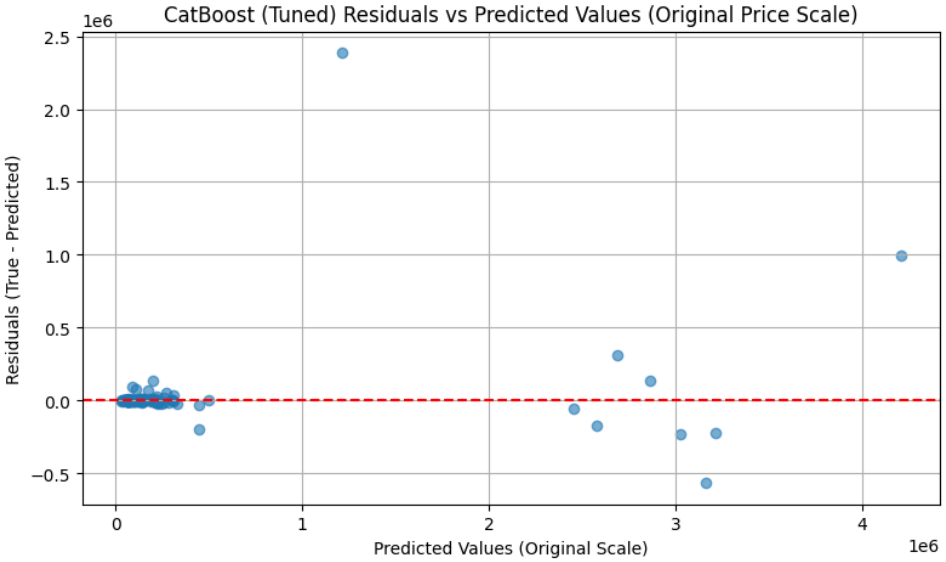
The performance of the default and tuned versions of CatBoost were evaluated on the test set, using the original price scale for meaningful interpretation by applying , inverse power transformations to both the predicted and actual test values. With 'random_state=42' set throughout the workflow, the tuned CatBoost model achieved an R^2 score of 0.909 on the test set, compared to 0.850 for the default model. This represents a 6.9% increase in R^2, along with a reduction in mean squared error (MSE) from approximately 111,449,492,212 to 67,876,712,481 and a decrease in mean absolute error (MAE) from about $67,689 to $58,056. These results indicate a substantial improvement in predictive accuracy after hyperparameter tuning. An R^2 of 0.909 suggests that the tuned model explains over 90% of the variance in the target variable, which is considered very good performance for a regression task. The final CatBoost model was evaluated by residual plots and scatter plots of predicted vs. actual values, providing a comprehensive visual assessment of how well the tuned model fits the test data

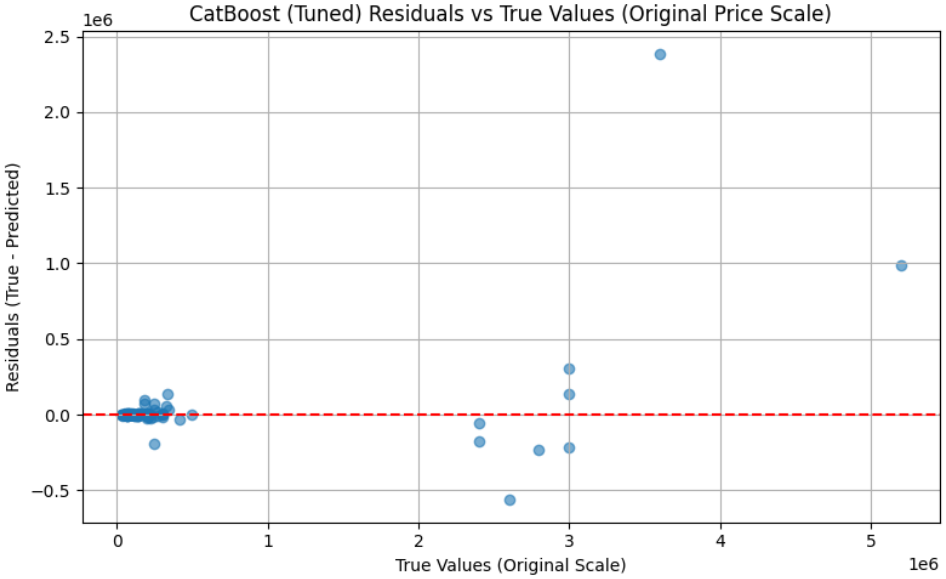
4.3.1 Residual Analysis

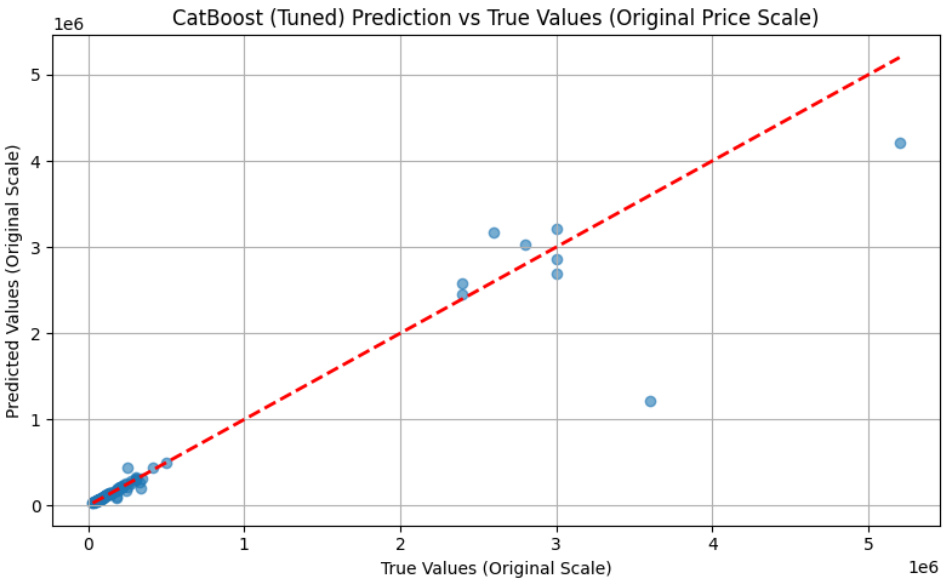
The majority of residuals are concentrated around zero, indicating that the model provides accurate predictions for most price ranges. However, there are some noticeable outliers characterized by larger residuals at higher true price values. This pattern implies that the model tends to be less precise for the most expensive cars, because such high-value cases are underrepresented in the dataset.

4.3.2 Predicted vs Actual Values

Most points lie close to the diagonal line, which represents perfect predictions, demonstrating that the model's predictions closely match the actual values for the majority of cases. There is a slight increase in deviation from the line at the highest price levels, further supporting the observation that the model struggles more with extreme values. .


CatBoost (Tuned) Residuals vs Predicted Values (Original Price Scale)
1e6
2.5
2.0
1.5
1.0
0.5
0.0
−0.5
Residuals (True - Predicted)
0
1
2
3
4
1e6
Predicted Values (Original Scale)


CatBoost (Tuned) Residuals vs True Values (Original Price Scale)
1e6
2.5
2.0
1.5
1.0
0.5
0.0
−0.5
Residuals (True - Predicted)
0
1
2
3
4
5
1e6
True Values (Original Scale)



Overall, these plots demonstrate that the tuned CatBoost model delivers strong predictive accuracy for most vehicles, particularly within the common price ranges. The model's performance remains robust throughout the dataset, though it exhibits some difficulty in precisely estimating prices for the most expensive, less frequent cars in the Dataset.

4.4 Key Insights

- Data cleaning,especially correcting inconsistent values and handling duplicates, was crucial for accurate modeling.
- Tree-based models significantly outperformed linear regression, highlighting the importance of selecting models aligned with the data's characteristics.
- Power transformations and engineered features, such as Engine Type, enhanced the dataset's expressiveness and supported better model learning.
- Hyperparameter tuning of the CatBoost model led to a notable performance improvement, validating the importance of model optimization.

4.5 Recommendations for Future Work

- **Dataset expansion for luxury cars**: Gather additional data, particularly for luxury and high-end sports cars, to improve model generalization or consider data augmentation.

- **Additional feature engineering**: Incorporate additional features and explore feature interactions.
- **Advanced modeling techniques**: Investigate ensemble strategies (e.g., stacking or blending) or deep learning architectures for further performance gains.
- **Pipeline automation**: Develop automated data pipelines for cleaning, feature engineering, and model selection, making the process scalable and reproducible.
- **Model deployment**: Build an API or web interface for real-time predictions and implement monitoring for model drift.
- **Continuous Learning**: Incorporate new data as it becomes available and retrain the model periodically to reflect evolving market conditions.

Conclusions

This project demonstrates a complete machine learning lifecycle. The final CatBoost model, with a test R^2 of **0.909**, delivers high-accuracy predictions and reflects best practices in data science and machine learning engineering. With further refinement and deployment, this solution can serve real-world use cases in automotive valuation platforms and dealership analytics.