Exploratory Data Analysis

CAR PRICE PREDICTION

Students

Cucos Marianita(408), Dilirici Mihai(411), Huțan Mihai-Alexandru(407), Militaru Mihai(411), Cosmin Moarcas(411)

Bucharest, January 2025

Contents

1 Scraper

Machine learning, especially deep learning are considered ”data-driven” approaches, meaning that the data is the most important factor, regardless of how much we fine-tune or optimize our approaches.

At the foundation of our experiments lays the scraping process, that was conducted manually using tools such Python [1], BeautifulSoup4 [2], pandas [3] and lxml [4].

1.1 Choosing our target

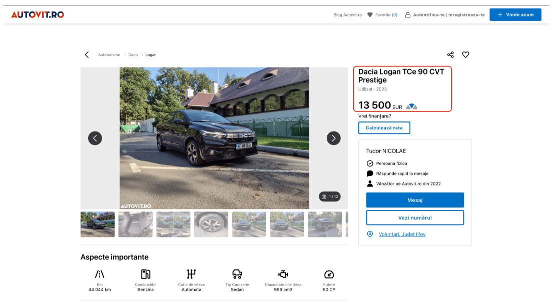
In order to obtain our dataset, we decided to scrape Autovit.ro, the biggest website in Romania for second hand cars listings. We primarily chose this website instead of other competitors on the market for the fact that they have a very rich listing configuration, and therefore, more data to scrape that might give us valuable insights for the pricing models.

1.2 What we searched

The first thing we did was filter out the damaged cars by using a specific URL, for the website. The reasoning behind this decision is that damaged cars are a different subject, their price is heavily influenced by how much a car is damaged, which elements are destroyed, etc.. A complete approach would require a more complex modelling, in order to also detect the damages on the cars, and also way more data than Auotvit can provide us at the moment.

```
https://www.autovit.ro/autoturisme?search%5Bfilter_enum_damaged%5D=0
```

In Figure 1 we listed 4 subfigures, each one representing one of the parts that we targeted during our scraping process. These represent all the data available, and relevant for a seconda hand car listing.



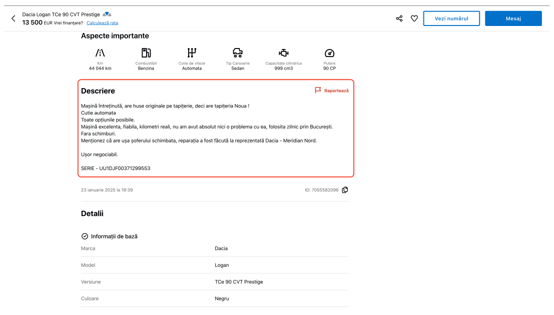
Dacia Logan TCe 90 CVT Prestige
Utilizat · 2023
13 500 EUR

Tudor NICOLAE
Persoana fizica
Răspunde rapid la mesaje
Vânzător pe Autovit.ro din 2022

Aspecte importante

Km	Combustibil	Cutie de viteze	Tip Caroserie	Capacitate cilindrica	Putere
44 044 km	Benzina	Automata	Sedan	999 cm3	90 CP

(a) Target price


Dacia Logan TCe 90 CVT Prestige
13 500 EUR Vrei finanțare? | Calculează rata
Vezi numărul
Mesaj
Raportează

Aspecte importante

Km	Combustibil	Cutie de viteze	Tip Caroserie	Capacitate cilindrica	Putere
44 044 km	Benzina	Automata	Sedan	999 cm3	90 CP

Descriere

Mașină întreținută, are huse originale pe tapițerie, deci are tapițeria Noua !
Cutie automata
Toate opțiunile posibile.
Mașină excelenta, fiabila, kilometri reali, nu am avut absolut nici o problema cu ea, folosita zilnic prin București.
Fara schimburi.
Menționez că are ușa șoferului schimbata, reparația a fost făcută la reprezentată Dacia - Meridian Nord.

Ușor negociabil.

SERIE - UU1DJF00371299553

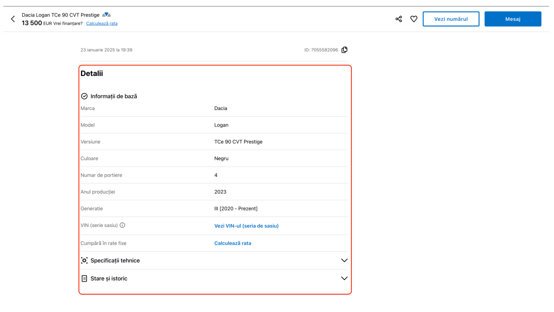
23 ianuarie 2025 la 19:39 ID: 7055582096

Detalii

Informații de bază

Marca	Dacia
Model	Logan
Versiune	TCe 90 CVT Prestige
Culoare	Negru

(b) Description


Dacia Logan TCe 90 CVT Prestige
13 500 EUR Vrei finanțare? | Calculează rata
Vezi numărul
Mesaj

23 ianuarie 2025 la 19:39 ID: 7055582096

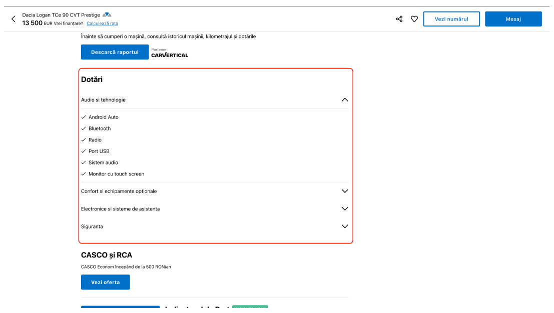
Detalii

Informații de bază

Marca	Dacia
Model	Logan
Versiune	TCe 90 CVT Prestige
Culoare	Negru
Numar de portiere	4
Anul producției	2023
Generatie	III [2020 - Prezent]
VIN (serie sasiu)	Vezi VIN-ul (seria de sasiu)
Cumpără în rate fixe	Calculează rata

Specificații tehnice
Stare și istoric

(c) Details


Dacia Logan TCe 90 CVT Prestige
13 500 EUR Vrei finanțare? | Calculează rata
Vezi numărul
Mesaj
Descarcă raportul
Partener CARVERTICAL
Vezi oferta

Înainte să cumperi o mașină, consultă istoricul mașinii, kilometrajul și dotările

Dotări

Audio si tehnologie
- ✓ Android Auto
- ✓ Bluetooth
- ✓ Radio
- ✓ Port USB
- ✓ Sistem audio
- ✓ Monitor cu touch screen

Confort si echipamente optionale
Electronice si sisteme de asistenta
Siguranta

CASCO și RCA
CASCO Econom începând de la 500 RON/an

(d) Optional features

Figure 1: Scraped information from Autovit.ro

1.3 Challenges

During the implementation phase, we faced one major issue, the website has a rate limiter implemented. The easiest solution that we found, was using a sleep in order to simulate a more humanly behaviour, and not get blocked by the website.

Another small issue was that the website expects an actual browser as it's user, curl or requests package for python not working out of the box when hitting their endpoints. We found an easy solution for this problem by adding the following headers to each request.

```
"User-Agent": "Mozilla/5.0 (Windows NT 10.0; Win64; x64)
    AppleWebKit/537.36 (KHTML, like Gecko) Chrome/119.0.0.0
    Safari/537.36"
```

We also faced some network issue, but a simple retry mechanism, of retrying maximum three times, solved all these issues.

1.4 Limitations

The biggest limiting factor for the dataset we have scraped, is the target variable. Since we are no experts, we could not fill the actual correct price for the entries, and therefore, we consider the listings inside autovit as correct. This might skew our models performances, due to overpriced or underpriced listings. One solution that we did not explore in our approach, would be outlier detection, or watching closely the predictive variance in our model in order to see the uncertainty, or maybe even using Bayesian

Neural Networks for such detection tasks. The limiting factor that stopped us from such an approach was the small size of the dataset.

1.5 Results

At this step we did not focus on cleaning, or formatting the data, but only on getting as many entries as possible. The details about our obtained dataset are presented in Section 2.

2 Dataset Analysis

2.1 Dataset Structure

2.1.1 Dataset Composition

This dataset comprises almost 50.000 samples, each representing a car characterized by 61 features. These features describe various attributes such as the car's make, model, production year, mileage, fuel type, engine capacity, power, CO_2 emissions, and other technical specifications. The target variable, *price*, is a continuous numerical value representing the market price of the car. This dataset provides an excellent foundation for developing regression models aimed at predicting car prices based on their attributes. Furthermore, the diversity in features facilitates exploring the relationships between different car characteristics and their influence on pricing.

To ensure a focused and relevant analysis, we performed pre-processing steps to clean and refine the data set. Specifically, we excluded cars that are electric, vintage, tuned, or in lease, as these categories often exhibit unique pricing patterns and characteristics that could introduce noise into the analysis. The code for these pre-processing steps is included at the beginning of the "reformat_and_plots" file, prior to the generation of plots. By making these adjustments, the dataset is better suited to identify trends and patterns in mainstream car markets, leading to more robust and accurate modeling results. After cleaning the data, we are left with around 20.000 samples, enough to train our machine learning models later.

2.2 Insights

Extensive exploratory data analysis (EDA) was conducted to identify patterns, distributions, and relationships among the features. Below are the detailed findings for each visualization:

2.2.1 Firm vs Private Sales

To better understand the distribution of sellers, a bar plot was used to compare the number of vehicles sold by firms versus private individuals (see Figure 2). The x-axis represents the seller type (firm or private), while the y-axis shows the number of vehicles sold in each category. This visualization highlights whether sales are dominated by professional sellers or private owners, potentially revealing market dynamics and consumer preferences. A significant imbalance between these categories may also indicate different pricing strategies or vehicle conditions offered by the two groups.

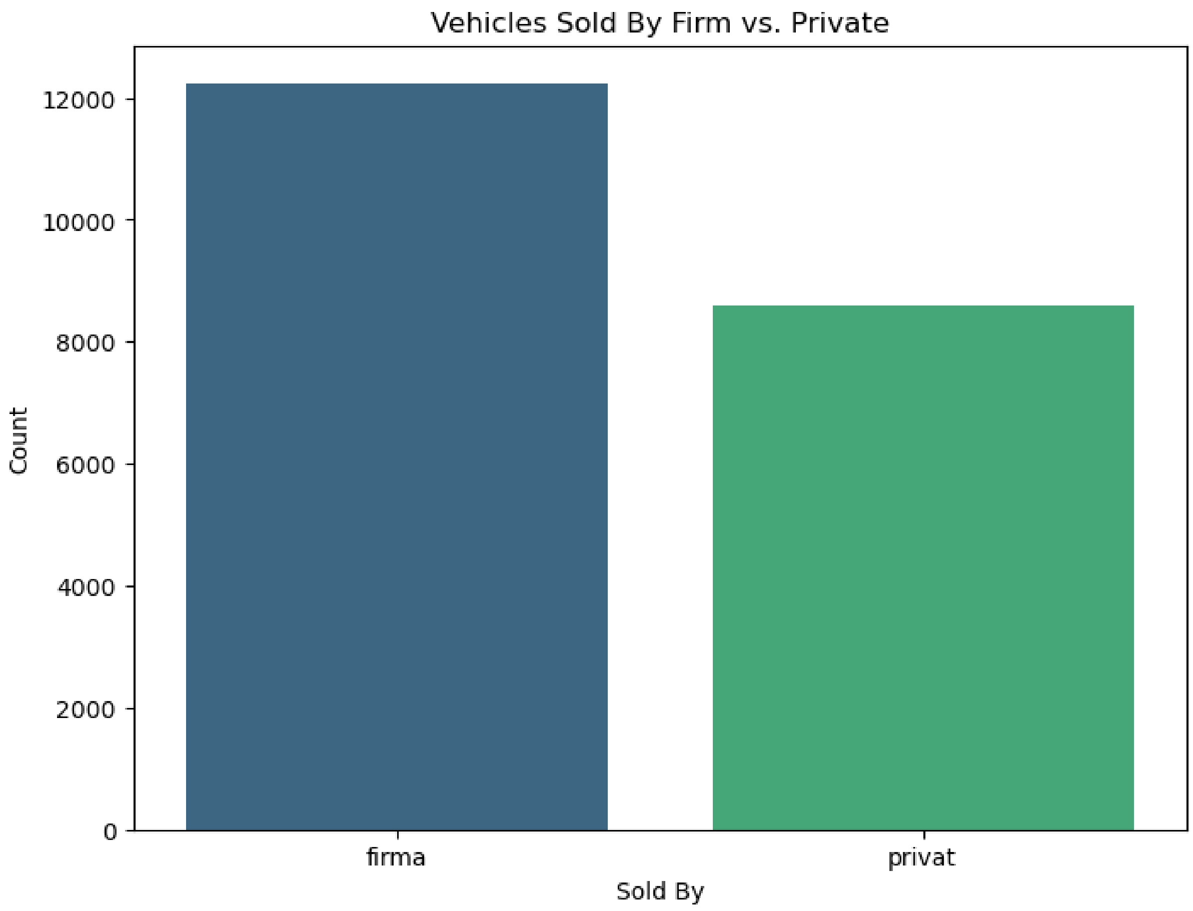


Figure 2: Comparison of vehicles sold by firms and private sellers.

2.2.2 Top Manufacturers and Models

The dataset's most frequently occurring manufacturers and car models were analyzed to determine market leaders. A two-panel bar plot (Figure 3) displays the top 10 manufacturers (left panel) and top 10 car models (right panel) based on the number of vehicles sold. This analysis sheds light on brand and model popularity, indicating which manufacturers and models dominate the market. Identifying these patterns helps in understanding pricing trends and customer preferences, which can influence predictive modeling for car prices.

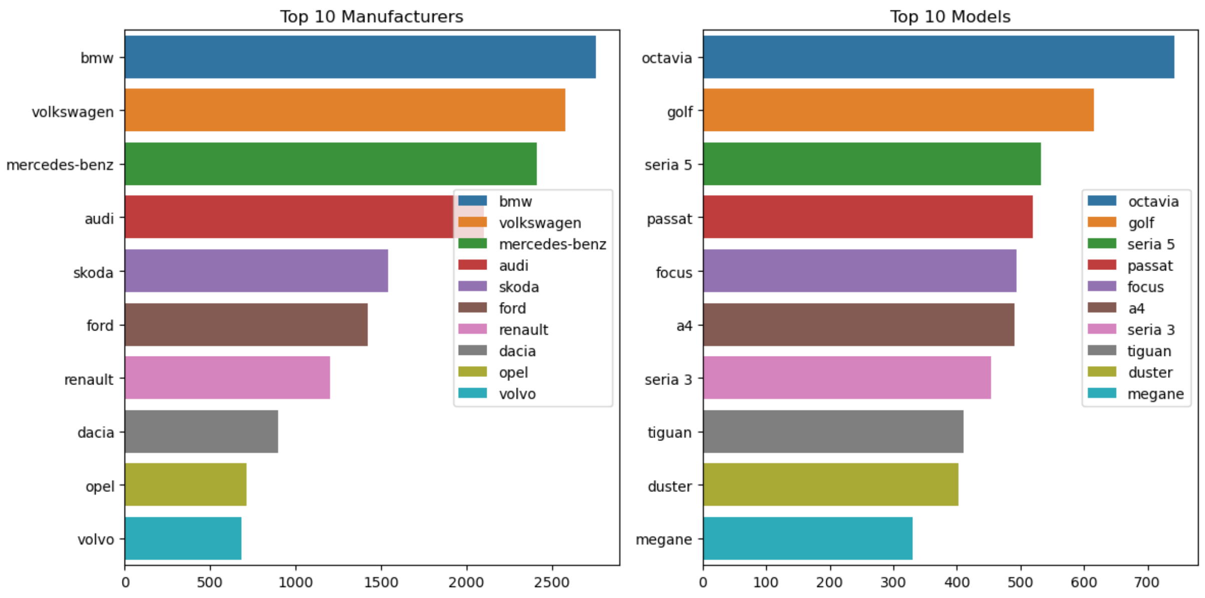


Figure 3: Top 10 manufacturers (left) and top 10 models (right) by vehicle count.

2.2.3 Car Price Distribution

The car price distribution was visualized using a histogram to explore the range, frequency, and overall spread of prices (see Figure 4). The x-axis represents the price ranges, while the y-axis shows the number of cars within each range. This analysis reveals key characteristics such as the skewness of the price distribution, the presence of high-priced outliers, and any clusters around specific price points. These insights are critical for understanding the dataset's overall pricing structure and identifying anomalies.

We chose to look only at the cars with price smaller than 100.000 euro because after that, there are a lot of outliers that decreases the performance of our models later discussed. One example is the Rolls Royce Phantom model. As there are fewer than 10 cars of this type, even our best model couldn't predict anything close to the actual price (actual price : 623.000, predicted : 344.000).

2.2.4 Correlation Heatmap of Car Features

A correlation heatmap (see Figure 5) was generated to examine the relationships between numerical features, including engine capacity, power, mileage, year of manufacture, CO_2 emissions, and price. Strong correlations between certain features (e.g., engine capacity and power, or year and price) can provide valuable insights for feature selection in predictive modeling. This visualization also highlights which features may have minimal influence on car price predictions, aiding in dimensionality reduction.

We are solely interested in the correlation between price and other features, and after analysing each of them, we found that we have five important ones (with value greater than +-0.4). Four of them have a positive value, meaning that when the price rises, the

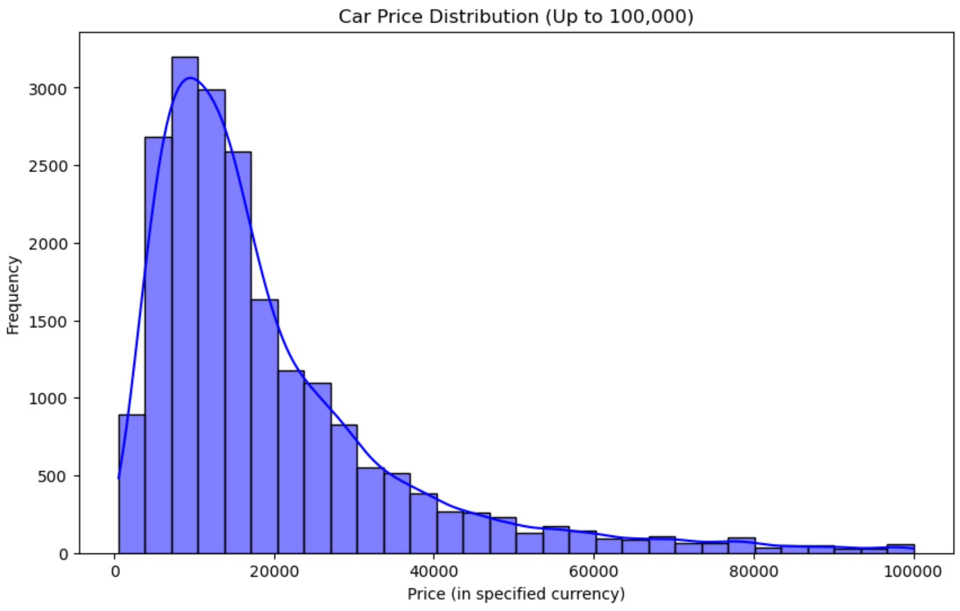


Figure 4: Distribution of car prices in the dataset.

engine capacity is also bigger, or the co2 emissions are greater. One negative example is between price and kilometers, as the first one decreases when the former increases.

A bad example of correlation is between price and urban consumption. At first, we thought that this should be more important, but after examining the dataset manually, we found out why there is no association between them. For example, if we look at a small car like Opel Corsa or Renault Megane, they have small urban consumption, unrestricted by the year of manufacture. On the other hand, newer models of big and powerful cars, like Audi Q5 (2018) or BMW 5 series (2023) have similar consumption, around 5-6 L/100 Km .

2.2.5 Relationship Between Engine Capacity and Price

The relationship between engine capacity and car price was visualized using a scatter plot with a regression line (see Figure 6). The x-axis represents engine capacity (in cubic centimeters), while the y-axis represents price (in EUR). This plot reveals whether larger engine capacities are associated with higher prices, providing insights into how technical specifications influence market value. Outliers or clusters in this plot may also indicate unique market behaviors or niche car types.

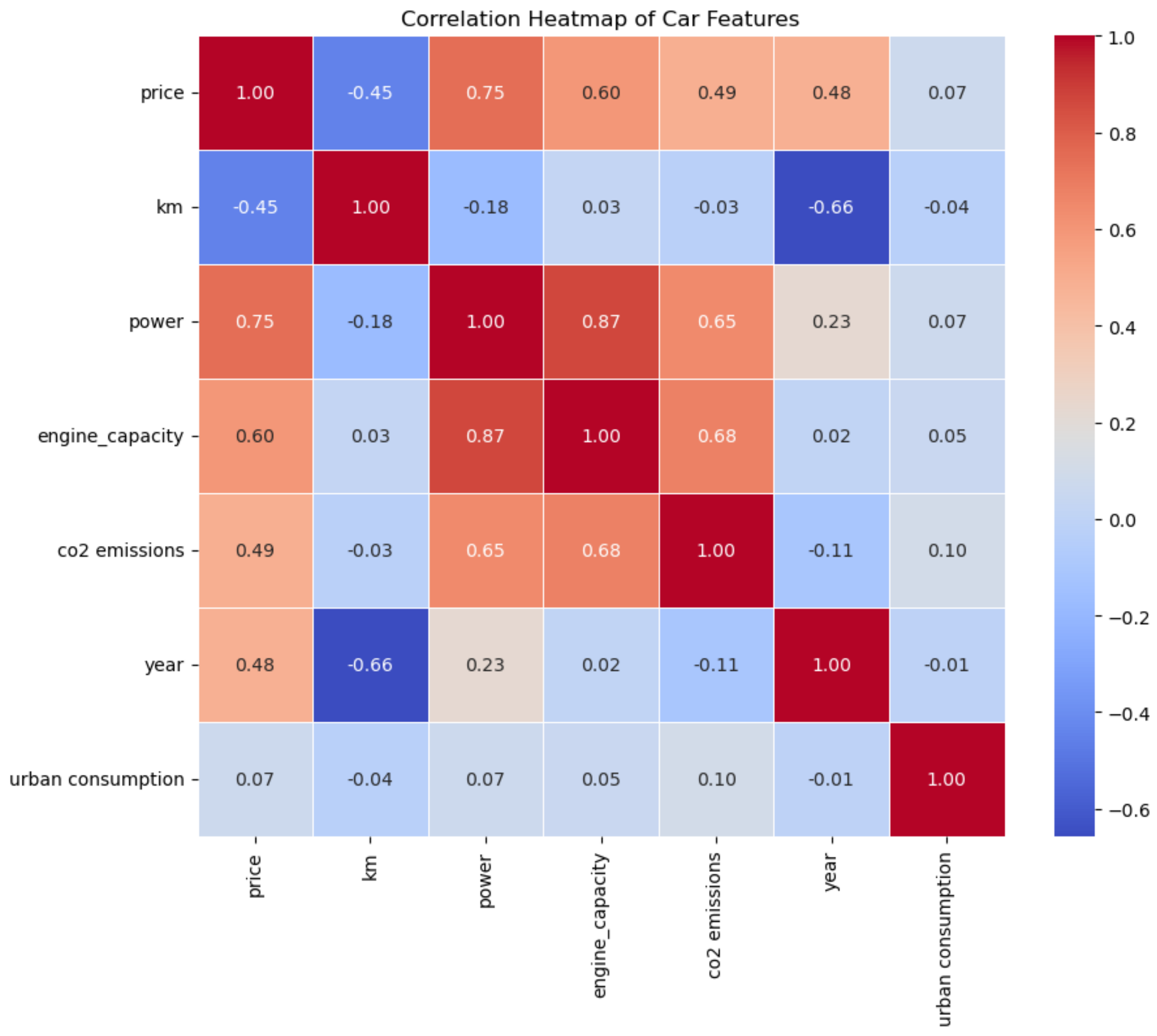


Figure 5: Correlation heatmap of numerical car features.

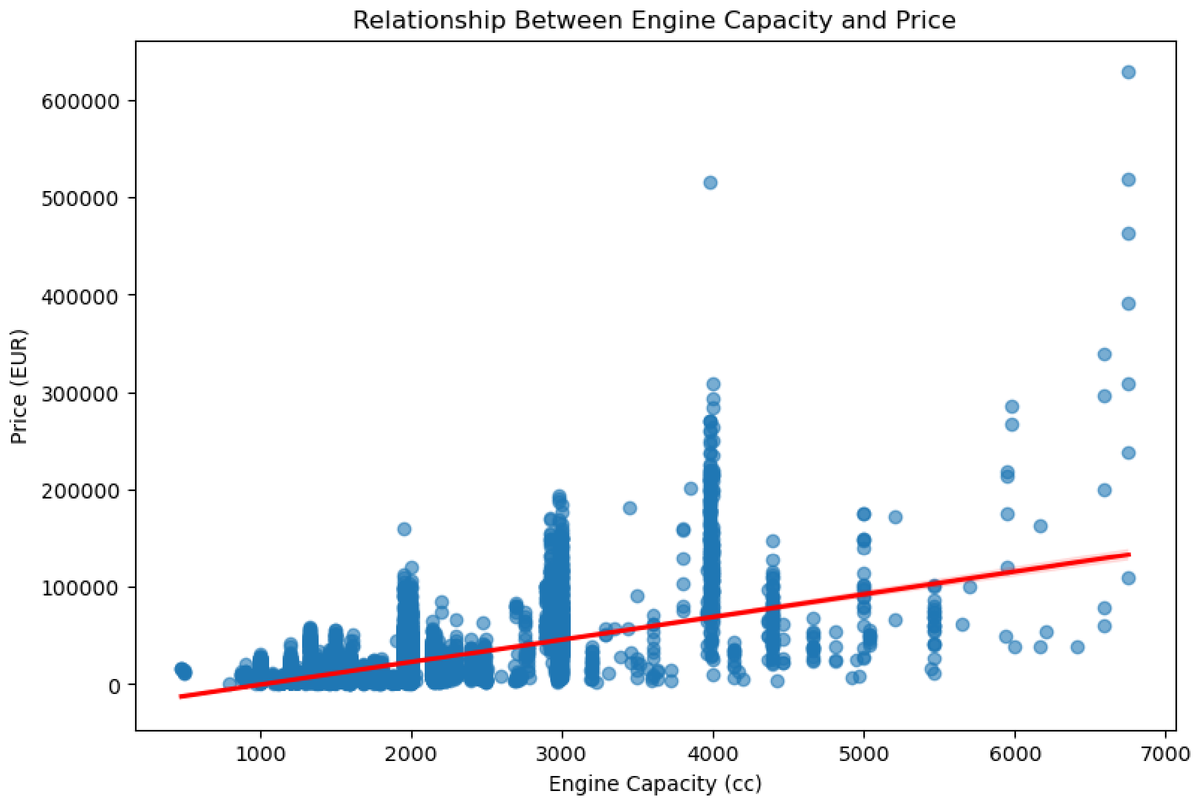


Figure 6: Relationship between engine capacity and price.

2.2.6 Average Car Price by Engine Horsepower Category

The relationship between engine horsepower and average car price was analyzed by categorizing horsepower into predefined ranges and plotting the average price for each category (see Figure 7). This analysis highlights how increasing engine performance affects market prices, offering valuable insights into customer willingness to pay for higher-powered vehicles.

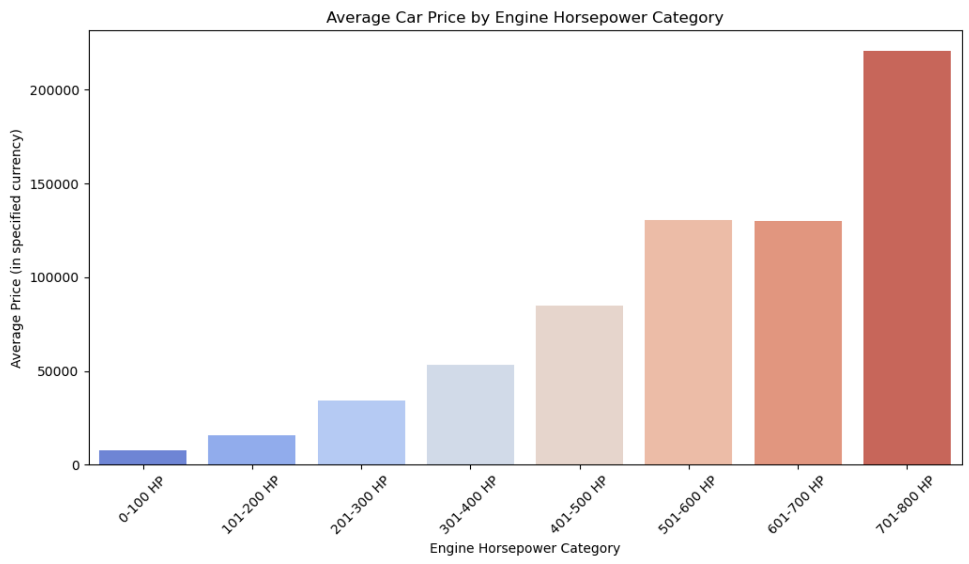


Figure 7: Average car price by engine horsepower category.

2.2.7 Number of Cars by Chassis Type

Figure 8 illustrates the distribution of cars across various chassis types, including sedans, SUVs, hatchbacks, and others. The x-axis categorizes the chassis types, while the y-axis represents the number of vehicles in each category. This visualization provides valuable insights into consumer preferences and market trends, highlighting the dominance of specific chassis types. It also helps identify the popularity of vehicle body styles, which may influence pricing and market segmentation strategies.

2.2.8 Evolution of CO2 Emissions by Production Year

Figure 9 showcases the changes in average CO2 emissions over production years. This plot reveals the automotive industry's response to environmental policies and technological advancements aimed at reducing emissions. It helps identify periods of significant reductions in CO2 emissions, which may align with regulatory shifts or innovations in engine efficiency and design.

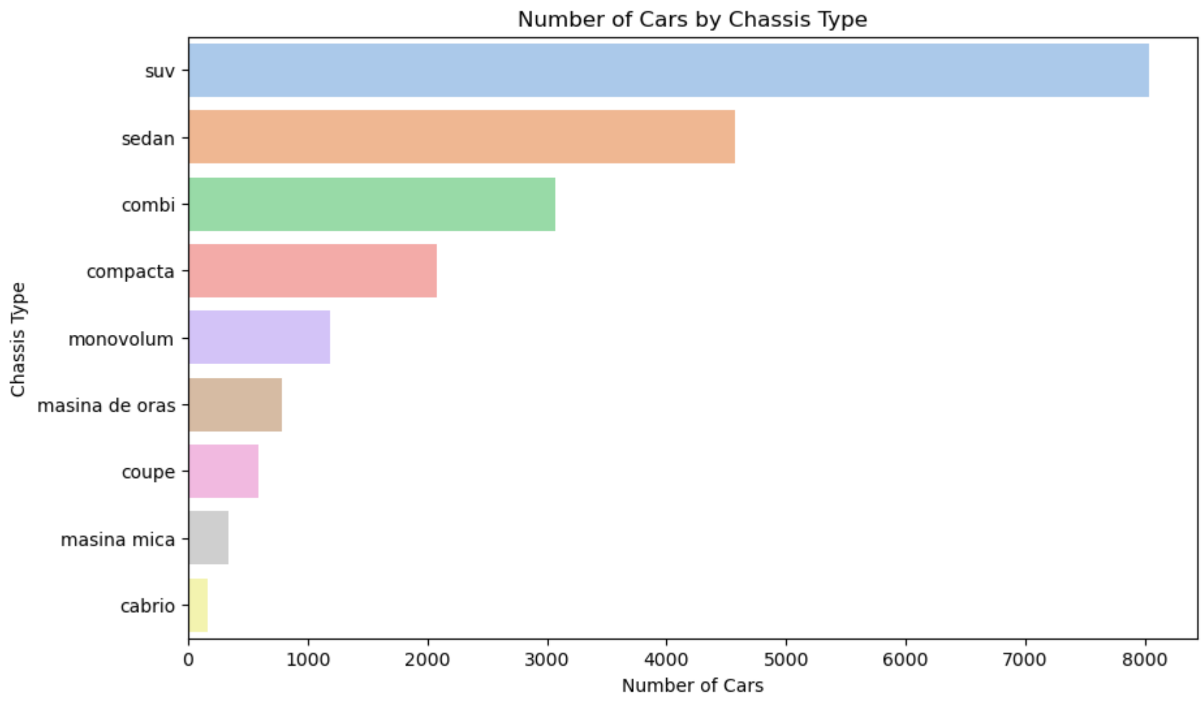


Figure 8: Number of cars by chassis type.

2.2.9 Price Range for Each Manufacturer

Figure 10 visualizes the price range of vehicles across different manufacturers. The plot provides an overview of the pricing strategies employed by each brand, highlighting the spread of low, mid, and high-priced vehicles. This helps understand the market positioning of manufacturers and how their product offerings cater to various customer segments.

2.2.10 Models With the Highest Number of Optional Features

Figure 11 identifies car models that offer the highest number of optional features. These models often provide extensive customization opportunities, appealing to consumers who prioritize personalization and added functionality. This analysis also reveals the correlation between the number of optional features and the car's price and target demographic.

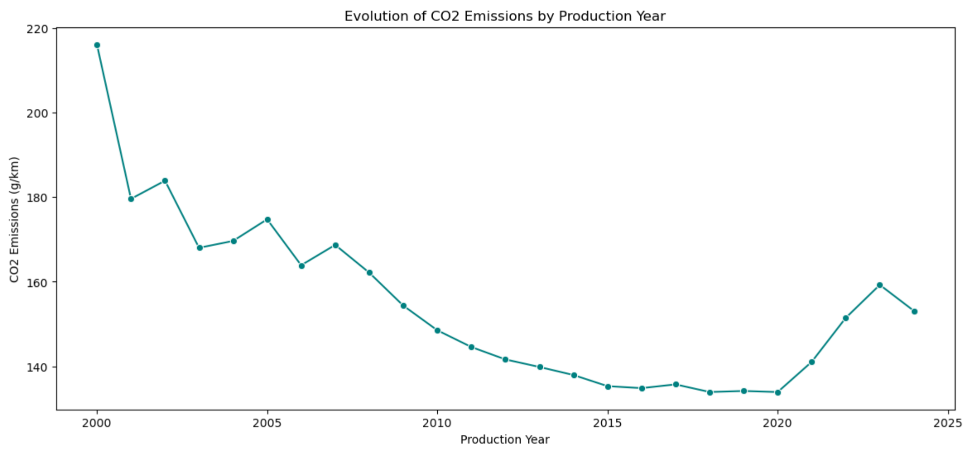


Figure 9: Evolution of CO2 emissions by production year.

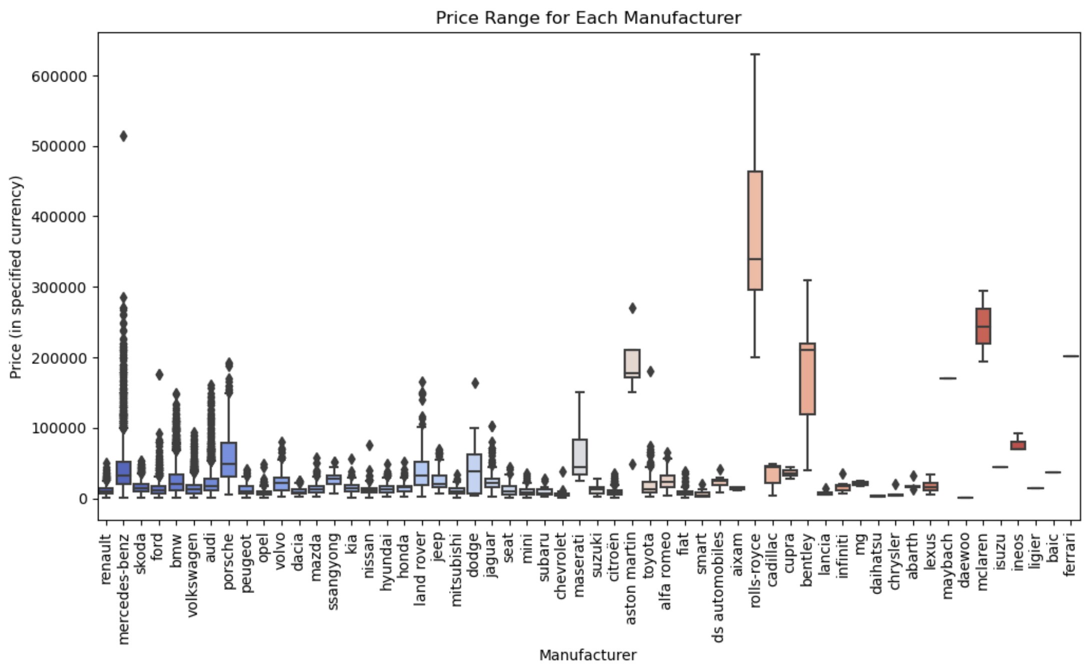


Figure 10: Price range for each manufacturer.

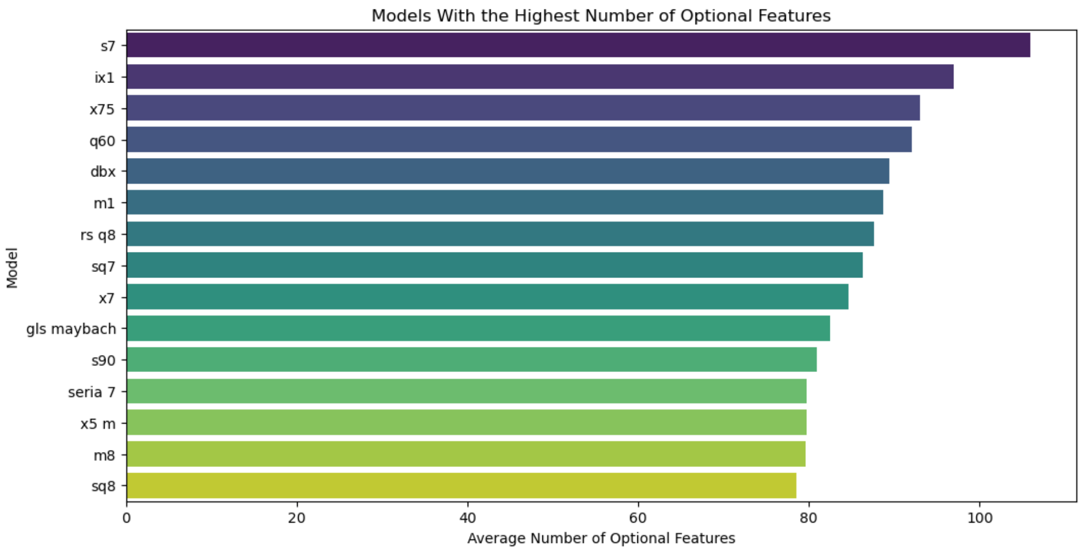


Figure 11: Models with the highest number of optional features.

2.2.11 Models With the Lowest Number of Optional Features

In contrast, Figure 12 highlights car models with the lowest number of optional features. These models may be positioned as budget-friendly or entry-level options, catering to cost-conscious buyers. This analysis helps understand how limited customization options influence pricing and target market preferences.

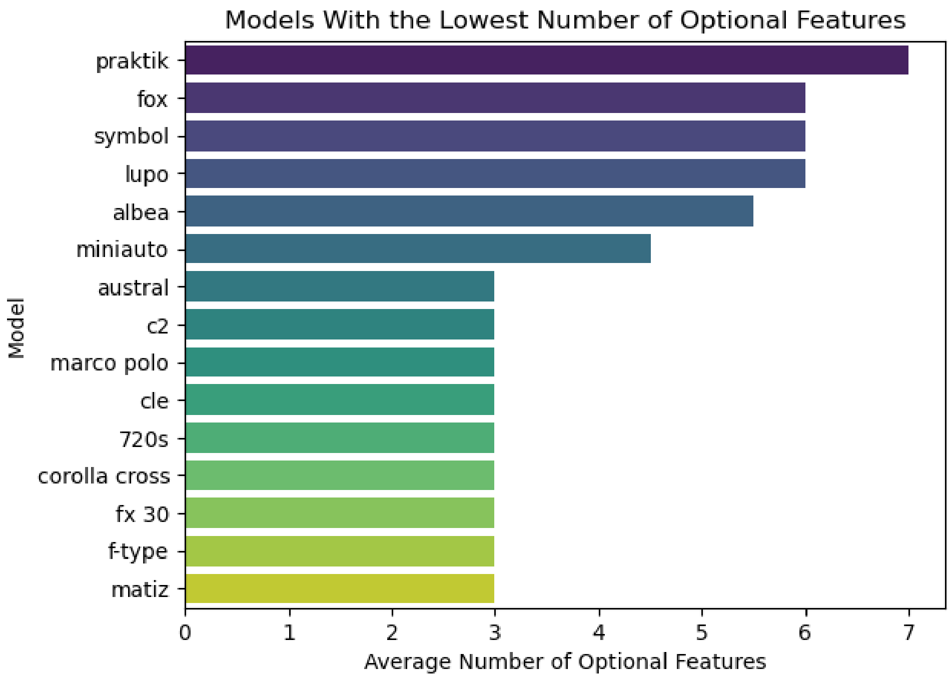


Figure 12: Models with the lowest number of optional features.

2.2.12 Cars Available by Country of Origin

Figure 13 presents the distribution of cars based on their country of origin. This analysis offers insights into the global automotive market, showcasing which countries dominate vehicle production and exports. It also helps understand regional preferences and the availability of specific brands or models in different markets.

2.2.13 Evolution of Engine Power by Year

Figure 14 tracks the evolution of average engine power over production years. This visualization reflects advancements in automotive technology, changes in consumer demand for performance, and the impact of regulations. It highlights how engine power has evolved to balance performance with efficiency and environmental considerations.

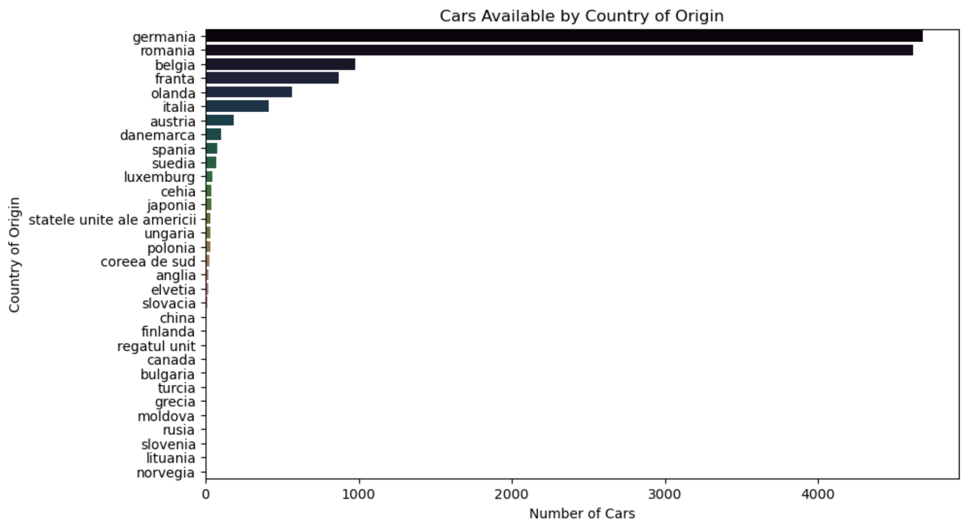


Figure 13: Cars available by country of origin.

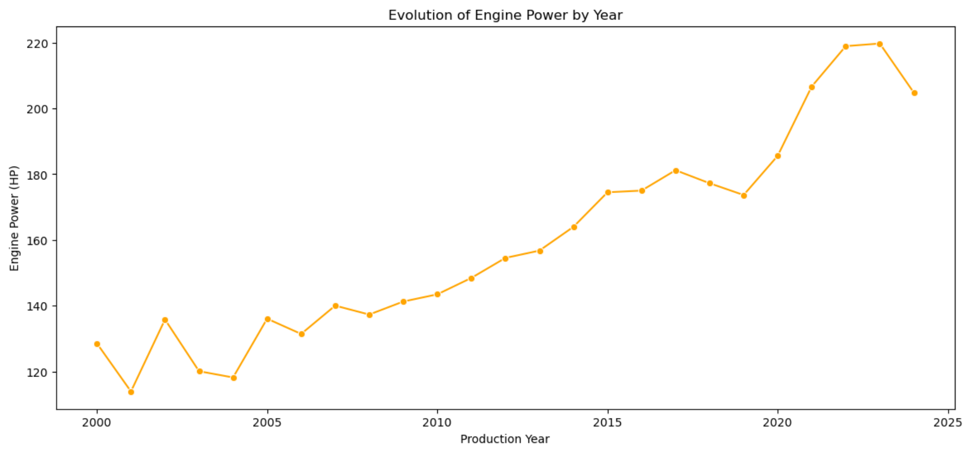


Figure 14: Evolution of engine power by production year.

2.2.14 Relationship Between Kilometers Driven and Price

Figure 15 explores the relationship between the number of kilometers a car has been driven and its market price. This scatter plot demonstrates how higher mileage typically leads to lower prices, reflecting wear and tear. It provides insights into depreciation trends and helps predict pricing for used vehicles.

2.2.15 Relationship Between Year of Manufacturer and Price

Figure 16 examines how the production year of a vehicle impacts its price. This plot reveals the depreciation pattern, showing how older vehicles generally lose value over time. It also highlights any exceptions where classic or vintage cars may retain or increase their value.

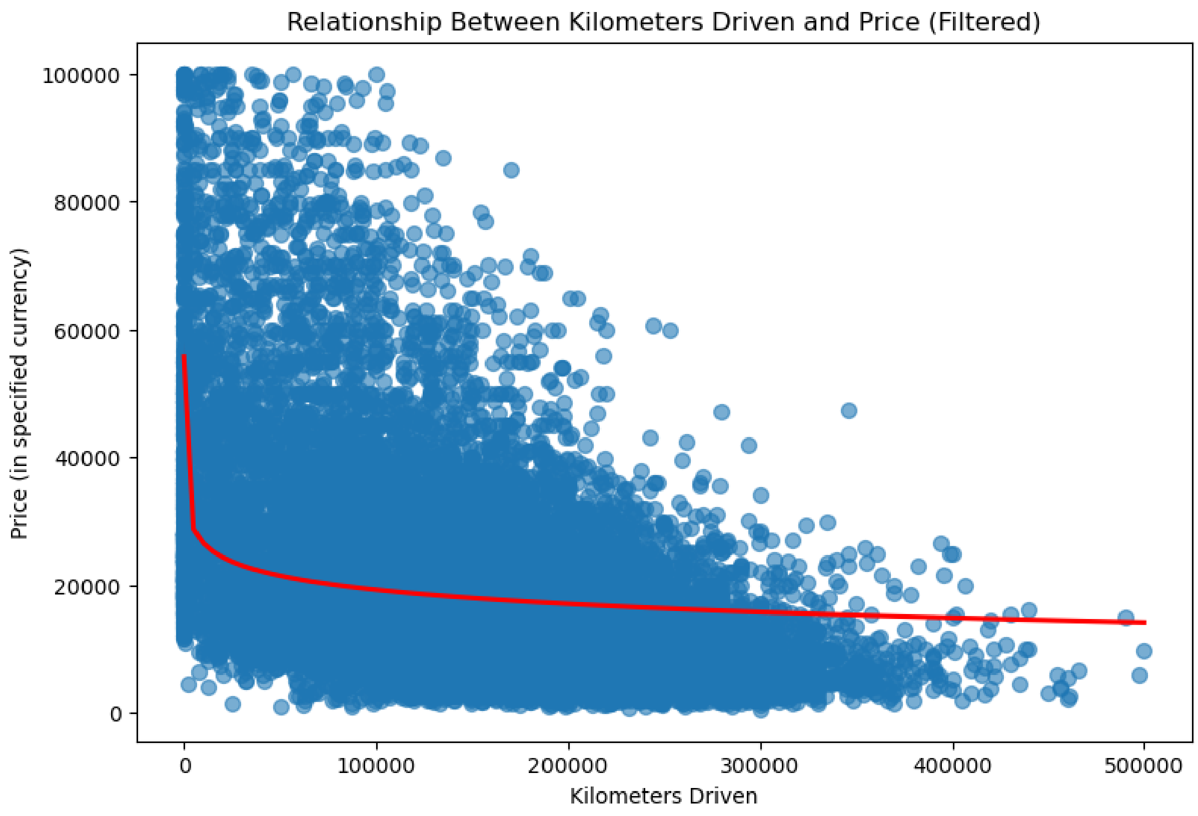


Figure 15: Relationship between kilometers driven and car price.

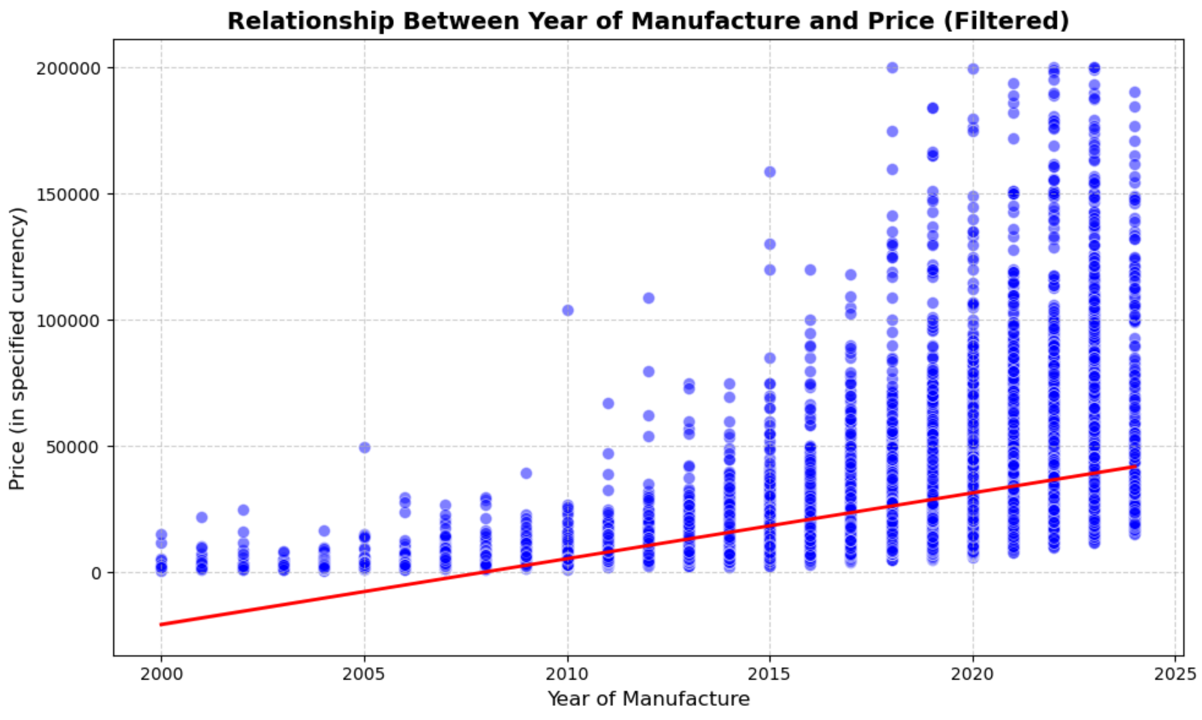


Figure 16: Relationship between year of manufacturer and car price.

2.2.16 Total Number of Cars Per Year (2000 and Later)

Figure 17 visualizes the total number of cars produced each year since 2000. This analysis highlights trends in car production, such as growth in demand, the impact of economic downturns, or industry shifts towards specific vehicle types.

Also, most of the cars on sale have the fabrication year between 2017-2020, most probably because 5 year (or 7 in some cases like KIA) warranty expiration is a major factor for putting a car for sale.

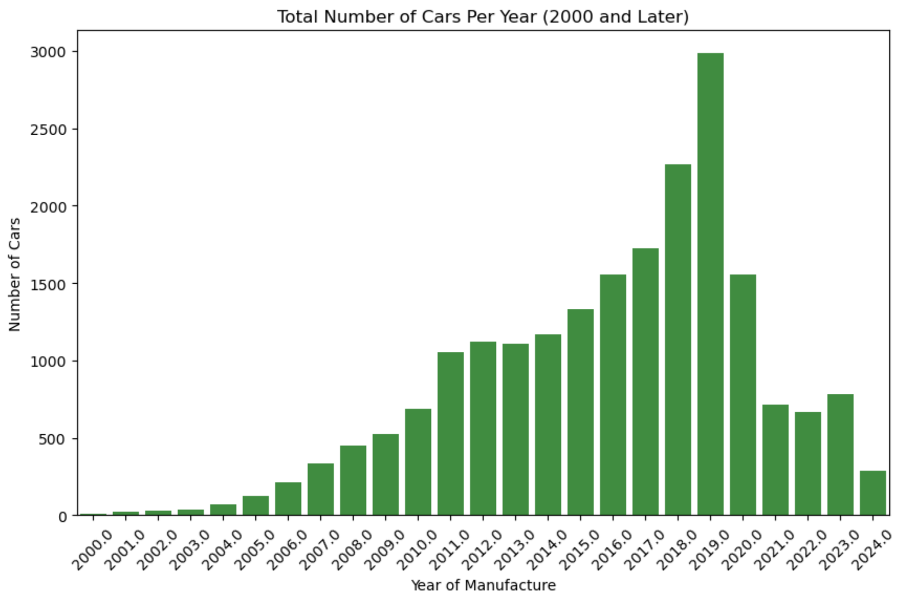


Figure 17: Total number of cars produced per year (2000 and later).

2.2.17 Average Car Prices by Year

Figure 18 shows the average prices of cars for each production year. This visualization provides insights into how market trends, technological advancements, and inflation have influenced car prices over time. It also highlights the premium associated with newer models.

Similar to figure 16, we can see that the price drastically increases if the car has available warranty (not older than 7 years).

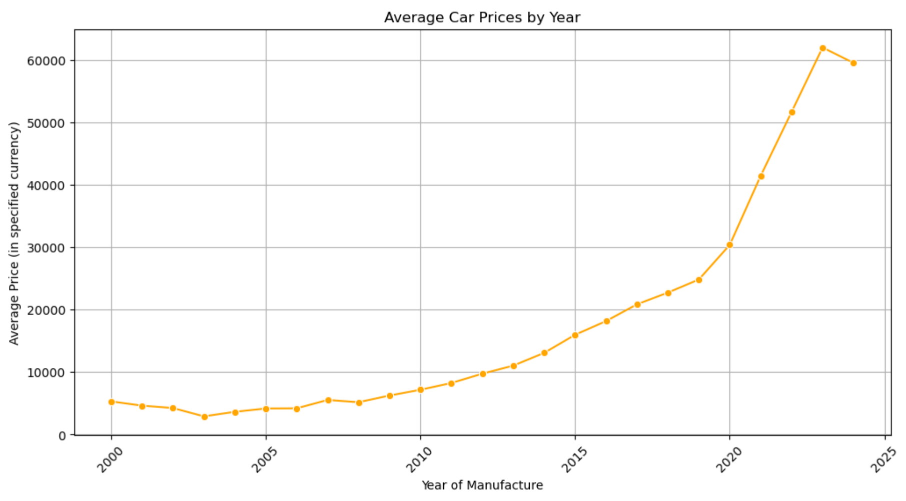


Figure 18: Average car prices by year.

2.2.18 Average Car Prices by Transmission Type

Figure 19 compares the average prices of cars with different transmission types, such as manual, automatic, or semi-automatic. This analysis reveals how transmission preferences influence pricing and provides insights into consumer trends in various markets.

Usually, the higher-end vehicles have 4x4 transmission to increase safety on slippery roads, compared to cheaper cars that don't have that much power and thus, they don't need so many safety features.

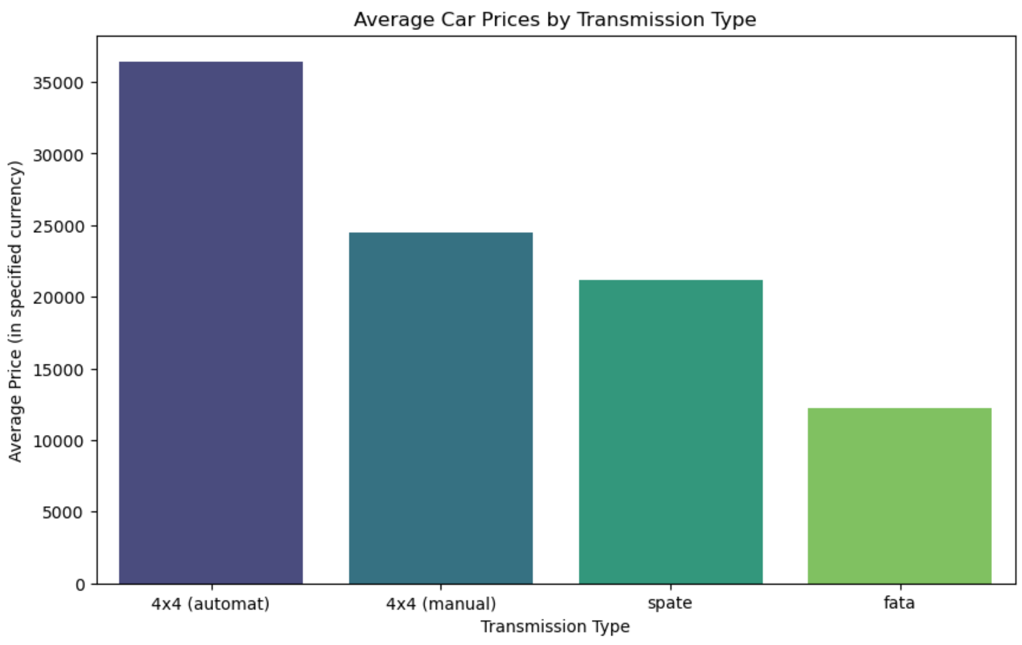


Figure 19: Average car prices by transmission type.

2.2.19 Average Engine Horsepower by Car Maker

Figure 20 visualizes the average engine horsepower for vehicles produced by each car manufacturer. This analysis highlights how brands position their vehicles in terms of performance, targeting different market segments, such as economy, luxury, or sports vehicles.

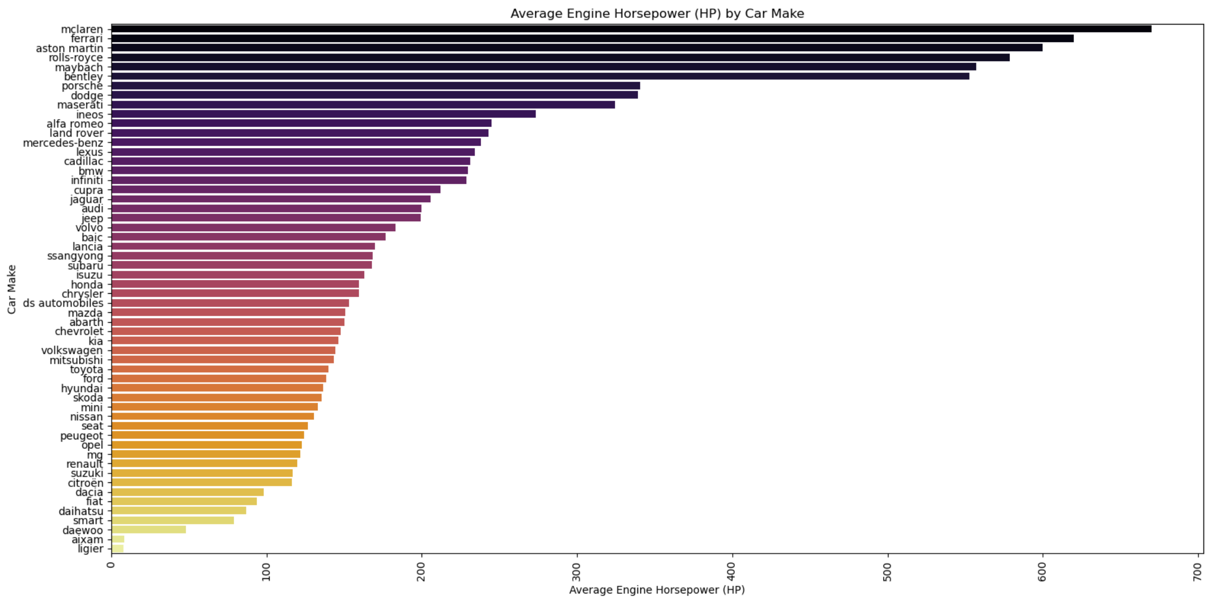


Figure 20: Average engine horsepower by car maker.

3 Experiments

3.1 Gradient Boosting Regressor

The Gradient Boosting Regressor was trained and tuned using a grid search approach. The hyperparameters tested are summarized in Table 1, with the best values highlighted in **bold**.

Table 1: Hyperparameter Tuning Results for Gradient Boosting Regressor

Hyperparameter	**Best Values**	**Values in Grid Search**
Learning Rate	**0.1**	0.01, 0.1, 0.2
Maximum Depth	**7**	3, 5, 7
Maximum Features	**sqrt**	sqrt, log2
Minimum Samples per Leaf	**1**	1, 2, 4
Minimum Samples per Split	**10**	2, 5, 10
Number of Estimators	**200**	50, 100, 200
Subsample	**1.0**	0.8, 1.0

The performance of the model was evaluated using the R^2 score, Mean Absolute Error (MAE), and Mean Squared Error (MSE). The results for both the validation and test datasets are summarized in the tables below.

Validation Set Results

Metric	**Value**
R^2 Score	0.9304
Mean Absolute Error (MAE)	1596.37
Mean Squared Error (MSE)	5395878.48

Table 2: Validation Set Results

Test Set Results

Metric	**Value**
R^2 Score	0.9224
Mean Absolute Error (MAE)	1695.48
Mean Squared Error (MSE)	6212566.81

Table 3: Test Set Results

3.2 LightGBM Model

The LightGBM Regressor was also trained and tuned using a grid search approach. The best hyperparameters found are as follows:

Table 4: Hyperparameter Tuning Results for LightGBM Regressor

Hyperparameter	**Best Values**	**Values in Grid Search**
Colsample by Tree	**0.6**	0.6, 0.8
Learning Rate	**0.05**	0.05, 0.1
Maximum Depth	**10**	5, 10
Minimum Child Samples	**20**	20, 50
Number of Estimators	**500**	400, 500
Number of Leaves	**50**	30, 50
Subsample	**0.6**	0.6, 0.8

This model proved to be the best among those tested, achieving the smallest error in predicting the prices of used cars. The superior performance can be attributed to the following factors:

- **Flexible Handling of Nonlinear Relationships**: LightGBM's gradient boosting framework excels in capturing complex nonlinear relationships between features and the target variable, making it well-suited for predicting car prices, which are influenced by a variety of interdependent attributes.
- **High Efficiency with Large Datasets**: LightGBM is designed to handle large datasets and high-dimensional feature spaces efficiently. This is particularly important for the car price prediction dataset, which contains a substantial number of samples and features.
- **Regularization Techniques**: The use of parameters like `colsample_bytree` and `subsample` helped prevent overfitting by reducing the reliance on specific samples or features during tree construction.
- **Robustness to Feature Interactions**: LightGBM's ability to construct deep trees and manage interactions between features enabled it to capture subtle patterns that other models might have missed.

Evaluation Metrics for LightGBM

The performance of the LightGBM model was evaluated using the R^2 score, Mean Absolute Error (MAE), and Mean Squared Error (MSE). The results for both the validation and test datasets are summarized in the following tables.

Metric	**Value**
R^2 Score	0.9354
Mean Absolute Error (MAE)	1528.19
Mean Squared Error (MSE)	5003588.43

Table 5: Validation Set Results for LightGBM

Validation Set Results

Test Set Results

Metric	**Value**
R^2 Score	0.9299
Mean Absolute Error (MAE)	1592.94
Mean Squared Error (MSE)	5613138.10

Table 6: Test Set Results for LightGBM

3.3 Multilayer Perceptron (MLP) Regressor

The Multilayer Perceptron (MLP) model was trained and optimized using the Adam optimizer with a learning rate of 0.001. The MLP model consists of three fully connected layers with ReLU activation functions, as well as dropout layers for regularization to prevent overfitting. The model was trained for 900 epochs, with a batch size of 128. The architecture is as follows:

- **Input Layer:** 128 units
- **Hidden Layer 1:** 128 units with ReLU activation
- **Hidden Layer 2:** 64 units with ReLU activation
- **Output Layer:** 1 unit (for regression output)
- **Dropout Rate:** 0.2 after each hidden layer

The performance of the model was evaluated using the R^2 score, Mean Absolute Error (MAE), and Mean Squared Error (MSE). The results for the test dataset are summarized in the table below.

Metric	**Value**
R^2 Score	0.8720
Mean Absolute Error (MAE)	2192.89
Mean Squared Error (MSE)	10250744.0

Table 7: Test Set Results for MLP Regressor

Test Set Results

3.4 Random Forest Regressor

The Random Forest Regressor was also trained and tuned using a grid search approach. The hyperparameters tested are summarized in Table 8, with the best values highlighted in **bold**.

Table 8: Hyperparameter Tuning Results for Random Forest Regressor

Hyperparameter	**Best Values**	**Values in Grid Search**
Bootstrap	**False**	True, False
Maximum Depth	**None**	None, 10, 20, 30
Maximum Features	**sqrt**	sqrt, log2
Minimum Samples per Leaf	**1**	1, 2, 4
Minimum Samples per Split	**2**	2, 5, 10
Number of Estimators	**150**	50, 100, 150

Evaluation Metrics for Random Forest Regression

The performance of the Random Forest model was evaluated using the R^2 score, Mean Absolute Error (MAE), and Mean Squared Error (MSE). The results for both the validation and test datasets are summarized in the tables below.

Validation Set Results

Metric	**Value**
R^2 Score	0.9119
Mean Absolute Error (MAE)	1731.25
Mean Squared Error (MSE)	6829695.27

Table 9: Validation Set Results for Random Forest Regression

Metric	**Value**
R^2 Score	0.9026
Mean Absolute Error (MAE)	1840.96
Mean Squared Error (MSE)	7803029.18

Table 10: Test Set Results for Random Forest Regression

Test Set Results

3.5 Lasso Regression

The optimal hyperparameter for the Lasso regression model was determined to be:

- **Alpha:** 0.01

During the grid search, several values of the regularization parameter α were tested, including 0.1, 1, 10, and 100. Among these, $\alpha = 0.01$ provided the best performance for the given dataset, as it offered the most effective balance between bias and variance.

However, the overall performance of the Lasso regression model was worse compared to the other models tested. This can be attributed to the following reasons:

- **Linear Assumptions:** Lasso regression assumes a linear relationship between features and the target variable, which might not adequately capture complex patterns or interactions present in the data.

- **High Dimensionality:** The dataset contains a large number of features, and other models like Gradient Boosting and Random Forests are better suited for handling high-dimensional and non-linear relationships in data.

These limitations highlight that while Lasso regression is a powerful tool for feature selection and regularization, it may not be the best choice for datasets with complex, non-linear relationships.

Evaluation Metrics for Lasso Regression

The performance of the Lasso regression model was evaluated using the R^2 score, Mean Absolute Error (MAE), and Mean Squared Error (MSE). The results for both the validation and test datasets are summarized in the tables below.

Validation Set Results

Metric	**Value**
R^2 Score	0.7945
Mean Absolute Error (MAE)	3012.51
Mean Squared Error (MSE)	15926031.68

Table 11: Validation Set Results for Lasso Regression

Test Set Results

Metric	**Value**
R^2 Score	0.7885
Mean Absolute Error (MAE)	3067.87
Mean Squared Error (MSE)	16938690.06

Table 12: Test Set Results for Lasso Regression

3.6 XGBoost Regressor

The XGBoost Regressor was also trained and tuned using a grid search approach. The hyperparameters tested are summarized in Table 13, with the best values highlighted in **bold**.

This model proved to be the best among those tested, along with `LightGBM`, achieving a small error in predicting the prices of used cars. The performance can be attributed to the following factors:

- **Flexible Handling of Nonlinear Relationships**: Similar to LightGBM, XGBoost's gradient boosting framework effectively captures complex nonlinear relationships between features and the target variable, making it suitable for predicting car prices, which are influenced by multiple interrelated factors.
- **High Efficiency with Large Datasets**: XGBoost is highly efficient in handling large datasets and high-dimensional feature spaces. This is crucial for the car price prediction dataset, which consists of a large number of samples and features.
- **Regularization Techniques**: XGBoost employs L1 and L2 regularization to prevent overfitting, ensuring that the model does not rely too heavily on specific features or data points during training.
- **Robustness to Feature Interactions**: XGBoost's ability to model complex feature interactions allows it to identify subtle patterns in the data, similar to LightGBM, making it effective at capturing the intricate relationships in car price prediction.

Table 13: Hyperparameter Tuning Results for RXGBoost Regressor

Hyperparameter	**Best Values**	**Values in Grid Search**
Learning rate	**0.1**	0.01, 0.1, 0.2
Maximum Depth	**5**	3, 5, 7
Colsample bytree	**0.8**	0.8, 1
Number of Estimators	**300**	100, 200, 300
Reg Alpha	**1**	0, 0.1, 0.5
Reg Lambda	**1**	1, 2, 5
Subsample	**1**	0.8, 1

Evaluation Metrics for XGBoost Regressor

The performance of the XGBoost Regressor model was evaluated using the R^2 score, Mean Absolute Error (MAE), and Mean Squared Error (MSE). The results for both the validation and test datasets are summarized in the tables below.

Validation Set Results

Metric	**Value**
R^2 Score	0.9326
Mean Absolute Error (MAE)	1583.66
Mean Squared Error (MSE)	5223175.23

Table 14: Validation Set Results for XGBoost Regressor

Test Set Results

Metric	**Value**
R^2 Score	0.9251
Mean Absolute Error (MAE)	1669.05
Mean Squared Error (MSE)	6001517.24

Table 15: Test Set Results for XGBoost Regressor

4 Conclusion

Based on the experiments conducted, the models were evaluated using the R^2 score, Mean Absolute Error (MAE), and Mean Squared Error (MSE). The results are summarized in Table 16 to identify the best-performing model.

Model	R^2 **Score (Test)**	**MAE (Test)**	**MSE (Test)**
Gradient Boosting Regressor	0.9224	1695.48	6212566.81
LightGBM Regressor	**0.9299**	**1592.94**	**5613138.10**
MLP	0.8720	1695.48	6212566.81
Random Forest Regressor	0.9026	1840.96	7803029.18
Lasso Regression	0.7885	3067.87	16938690.06
XGBoost Regressor	0.9251	1669.05	6001517.24

Table 16: Performance Summary of Models

From Table 16, the LightGBM Regressor demonstrated the best overall performance. It achieved the highest R^2 score of 0.9299, indicating that it explained approximately 93% of the variance in the target variable. Furthermore, the LightGBM model also produced the lowest Mean Absolute Error (MAE) of 1592.94 and the lowest Mean Squared Error (MSE) of 5613138.10 on the test dataset, highlighting its accuracy in predicting car prices with minimal deviation from actual values.

In comparison to the other models, LightGBM outperformed alternatives such as the Gradient Boosting Regressor (which had a R^2 score of 0.9224) and the XGBoost Regressor (with a R^2 score of 0.9251). Although XGBoost and Gradient Boosting also showed strong performance, the LightGBM Regressor stood out with consistently lower error metrics, suggesting it is better suited to the dataset at hand.

The results of the Multi-Layer Perceptron (MLP) and the Random Forest Regressor were also promising, but their performance was notably inferior compared to the boosting algorithms, with higher error metrics and lower R^2 scores. The Lasso Regression model showed the weakest performance, with the lowest R^2 score of 0.7885, indicating that it was less effective in capturing the complexity of the data.

Given its strong performance across multiple evaluation metrics, the LightGBM Regressor is recommended as the optimal model for this task. Its ability to handle large datasets and model complex, nonlinear relationships makes it highly suitable for predicting car prices in this context.

A Technologies Used

This section mentions the main technologies used in this project:

- **Python** [1]
- **BeautifulSoup4** [2]
- **lxml** [4]
- **Pandas** [3]
- **Matplotlib** [5]
- **Seaborn** [6]
- **Numpy** [7]
- **Scikit-learn** [8]
- **TensorFlow** [9]

References

[1] Python Software Foundation. *Python Programming Language*. Version 3.x used. 1991. URL: https://www.python.org/.

[2] Leonard Richardson. *Beautiful Soup Documentation*. Version 4.x used. 2004. URL: https://www.crummy.com/software/BeautifulSoup/.

[3] Wes McKinney et al. *pandas: a foundational Python library for data analysis and statistics*. Version 1.x used. 2010. URL: https://pandas.pydata.org/.

[4] Stefan Behnel, Martijn Faassen, and Ian Bicking. *lxml: XML and HTML with Python*. Version 4.x used. 2004. URL: https://lxml.de/.

[5] J. D. Hunter. *Matplotlib: A 2D Graphics Environment*. 2007. DOI: 10.1109/MCSE.2007.55. URL: https://matplotlib.org/.

[6] Michael L. Waskom. *Seaborn: statistical data visualization*. 2021. DOI: 10.21105/joss.03021. URL: https://seaborn.pydata.org/.

[7] Charles R. Harris, K. Jarrod Millman, Stéfan J. van der Walt, Ralf Gommers, Pauli Virtanen, David Cournapeau, Eric Wieser, Julian Taylor, Sebastian Berg, Nathaniel J. Smith, Robert Kern, Matti Picus, Stephan Hoyer, Marten H. van Kerkwijk, Matthew Brett, Allan Haldane, Jaime Fernández del Río, Mark Wiebe, Pearu Peterson, Pierre Gérard-Marchant, Kevin Sheppard, Tyler Reddy, Warren Weckesser, Hameer Abbasi, Christoph Gohlke, and Travis E. Oliphant. *Array programming with NumPy*. 2020. DOI: 10.1038/s41586-020-2649-2. URL: https://numpy.org/.

[8] Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, Jake Vanderplas, Alexandre Passos, David Cournapeau, Matthieu Brucher, Matthieu Perrot, and Édouard Duchesnay. *Scikit-learn: Machine Learning in Python*. 2011. URL: https://scikit-learn.org/.

[9] Martín Abadi, Paul Barham, Jianmin Chen, Zhifeng Chen, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Geoffrey Irving, Michael Isard, Manjunath Kudlur, Josh Levenberg, Rajat Monga, Sherry Moore, Derek G. Murray, Benoit Steiner, Paul Tucker, Vijay Vasudevan, Pete Warden, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng. *TensorFlow: A System for Large-Scale Machine Learning*. 2016. URL: https://www.tensorflow.org/.