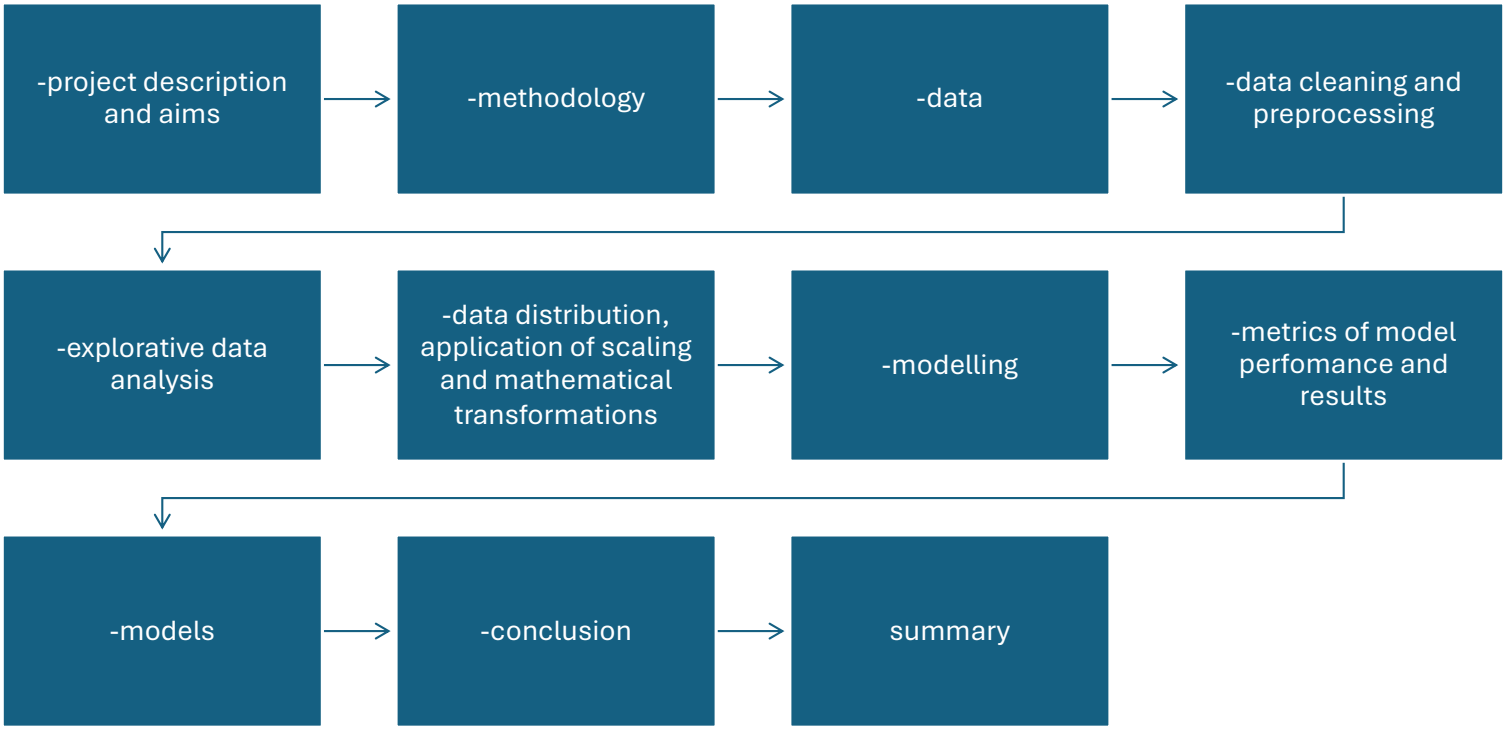
Data-Driven Analysis and Price Prediction of Germany's Used Car Market (2023)

Valeria Solozobova

Project report(2023-11-10)

Project URL: https://github.com/ValeriaMalin/Used-Car-Market-2023-

Contents


-project description and aims
-methodology
-data
-data cleaning and preprocessing
-explorative data analysis
-data distribution, application of scaling and mathematical transformations
-modelling
-metrics of model perfomance and results
-models
-conclusion
summary

Project description and data

In this project, I conducted a comprehensive analysis of the **German used car market of year 2023**, focusing on key factors that influence the pricing of used cars. The primary objective was to build machine learning models capable of accurately predicting car prices based on various features, while also gaining insights into the significance of these features through exploratory data analysis (EDA).

The dataset was downloaded from kaggle:
https://www.kaggle.com/datasets/wspirat/germany-used-cars-dataset-2023/data
About Dataset: Collection of car offers from one of Germany's largest car sales websites from the year 2023. This scraped dataset contains a wide range of information about car offers, covering a cars manufactured from 1995 to 2023. It contains data about 251079 cars.

Features:
Brand: The brand or manufacturer of the car.
Model: The specific model of the car.
Color: The color of the car's exterior.
Registration Date: The date when the car was registered (Month/Year).
Year of Production: The year in which the car was manufactured.
Price in Euro: The price of the car in Euros.
Power: The power of the car in kilowatts (kW) and horsepower (ps).
Transmission Type: The type of transmission (e.g., automatic, manual).
Fuel Type: The type of fuel the car requires.
Fuel Consumption: Information about the car's fuel consumption in L/100km ang g/km.
Mileage: The total distance traveled by the car in km.
Offer Description: Additional description provided in the car offer.

Methodology

1. Data Collection and Preprocessing

•Data preprocessing steps included converting the data to correct type (int, float), handling **missing values**, possible correcting of not correct data , addressing **outliers**, and performing **feature encoding** (e.g., encoding categorical variables like car brands and models). My dataset revealed at least two variants of data preprocessing, thus creating two pre-processed datasets for the tests, in order to compare them and choosing the best way..

2.Exploratory Data Analysis (EDA)

•An extensive **EDA** was conducted to understand the distribution of key features and identify trends in the dataset:

- The overall situation on the market of the used cars: brands, models, transmission type, fuels, etc
- Scatter plots to find correlated variables, as mileage and price, power and price or fuel consumption and price
- Visualized the distribution of key variables such as car prices, mileage, and engine power, revealing **skewness** and the presence of **outliers**.
- **Correlation analysis** was performed to assess the relationships between features and the target variable (price), with a special focus on factors like car **color** and their potential impact on car prices.

3.Feature Distribution and Transformation

•I analysed the data distribution creating boxplot, histograms, qqplots, calculated skewness and curtosis, number of outliers. I performed the same after application **log** or **square root transformations** to highly skewed variables (e.g., car price, engine power, and mileage) to improve model performance.

Methodology

4.Pipeline Models

•Several machine learning models were developed and tested for predicting car prices on datasets (preprocessed in different way):

- **Linear Regression** with log-transformed features.
- **K-Nearest Neighbors (KNN)** with robust scaling to handle feature variability.
- **Random Forest Regressor** and **Decision Tree Regressor**, which naturally handle complex interactions between features without the need for scaling.
- **Decision Trees**
- **XGBoost Regressor**, a powerful gradient boosting model, was included for enhanced performance on structured data.

•The models were fine-tuned using **GridSearchCV** with cross-validation to identify the best-performing hyperparameters.

5. Evaluation and Results

•The best models were evaluated more detailed using key performance metrics such as **Mean Squared Error (MSE)**, **Mean Absolute Error (MAE)**, and **R^2 Score** to compare their predictive accuracy. Additionally, **feature importance and SHAP values** was studied for each model. For the strongest models I analysed the errors

6. Additional Models

•I further explored the following advanced models:

- **Neural Network**
- **CatBoost**
- **LightGBM (LGBoost)**

Data and cleaning/preprocessing

1

```
Index: 251079 entries, 0 to 251078
Data columns (total 14 columns):
 #   Column                  Non-Null Count   Dtype
---  ------                  --------------   -----
 0   brand                   251079 non-null  object
 1   model                   251079 non-null  object
 2   color                   250913 non-null  object
 3   registration_date       251075 non-null  object
 4   year                    251079 non-null  object
 5   price_in_euro           251079 non-null  object
 6   power_kw                250945 non-null  object
 7   power_ps                250950 non-null  object
 8   transmission_type       251079 non-null  object
 9   fuel_type               251079 non-null  object
 10  fuel_consumption_l_100km 224206 non-null  object
 11  fuel_consumption_g_km   251079 non-null  object
 12  mileage_in_km           250927 non-null  float64
 13  offer_description       251078 non-null  object
```

→

```
Non-Null Count   Dtype
--------------   -----
244726 non-null  object
244726 non-null  object
244642 non-null  object
244722 non-null  object
244546 non-null  float64
244542 non-null  float64
244593 non-null  object
244519 non-null  float64
244726 non-null  object
244726 non-null  object
218134 non-null  object
244726 non-null  object
244585 non-null  float64
244725 non-null  object
```

Dtypes changing and duplicates/obvious NaNs removal

2 Work on the ‘fuel_consumption’ – the most ‘dirty’ variable in the dataset:

fuel_consumption_l_100km
10,9 l/100 km
NaN
NaN
9,5 l/100 km
7,2 l/100 km
9,5 l/100 km

→

fuel_consumption_l_100km
10.9
NaN
9.5
7.2
9.5

Removing of all values not related to the fuel consumtion ,
Removing the units (l/100km), transforming the values to floats.
Work on extremely big for fuel consumption values (for example 224 to 22.4, which happened definitely as mistake during scraping),
Conception of hadling too small values (less than 1.6 l/100km) and handling NaN:

1) to use the mean or mode of fuel consumption for the model of corresponding year, resulting in dataset with 244567 entries

2)to drop all the NaNs and too small data (which could not be a true), resulting in an alternative dataset with 219735 entries

Data cleaning and preprocessing

3 Creating new variable ‘age’: 2023-’year’

4 Deleting the variables ‘registration_date’, ‘fuel_consumption_g_km’ (less informative than ‘fuel_consumption_l_100k’), ‘power_kw’ (‘power_ps’ is enough)

5 Considering outliers: The dataset is rich on outliesrs, especially outstanding are ‘price_in_euro’ (some luxury cars) and ’mileage_in_km’. I left them for now , in order to deside directly during the models test: to leave them or delete

6 Categorical values: ‘fuel_type’, ‘transmission_type’, ‘color’ were transformed with One_hot Encoder

Value counts:

```
transmission_type
Automatic          131181
Manual             117765
Unknown              1140
Semi-automatic        317
Name: count, dtype: int64
```

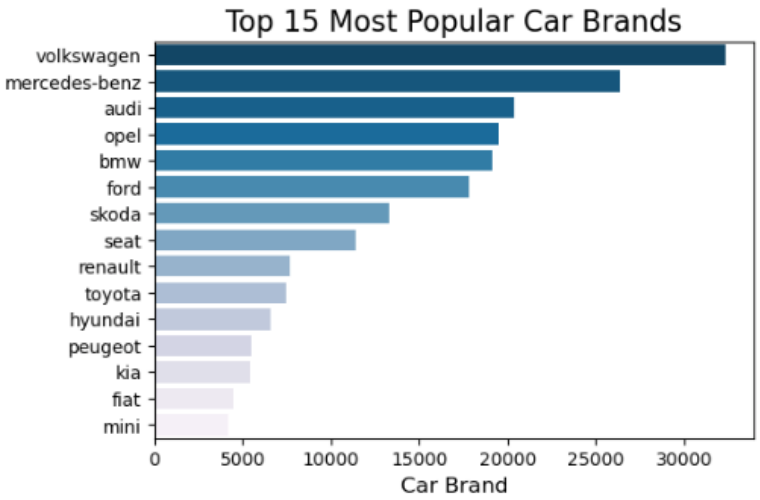
```
fuel_type
Petrol          143280
Diesel           86421
Hybrid           12470
Electric          5624
LPG               1255
CNG                508
Diesel Hybrid      474
Other              171
Unknown             96
Hydrogen            82
Ethanol             10
```

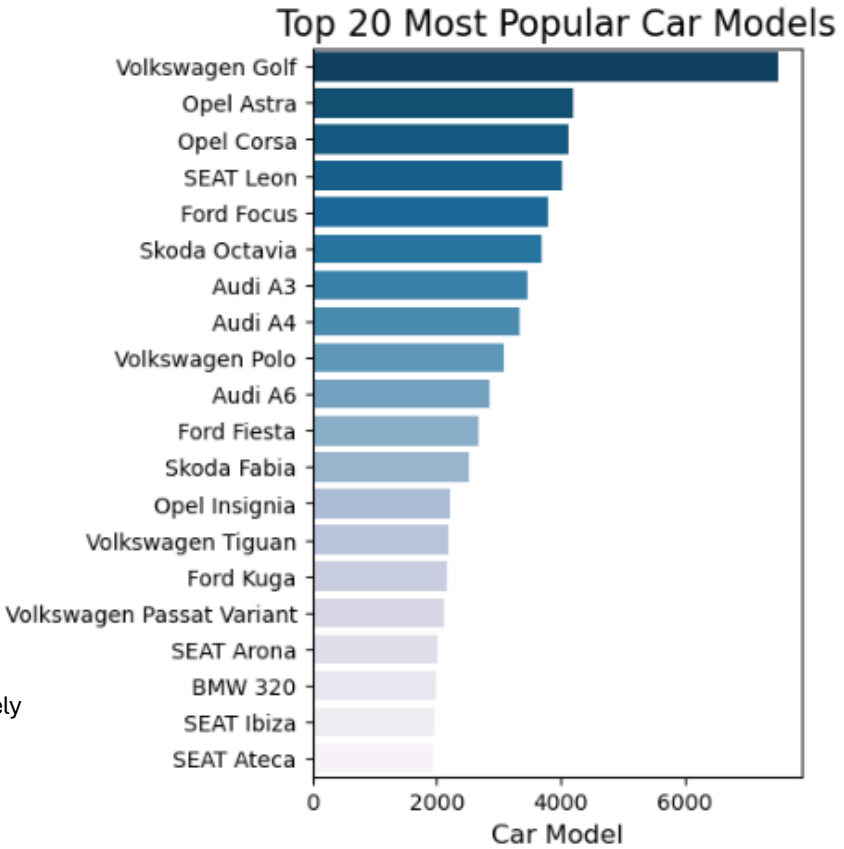
```
color
black     58598
grey      46657
white     40486
silver    34293
blue      32003
red       21192
brown      4406
green      3485
orange     3359
beige      2413
yellow     1775
bronze      584
gold        579
violet      407
```

7 Categorical values: ‘model’ and ‘brand’ were tranformed with Target Encoder. I have 67 brands and 1310 models In the dataset, which to much for the One-hot encoder

8 Deleting of resting Nans and ‘Unknown’ values

Explorative Data Analysis

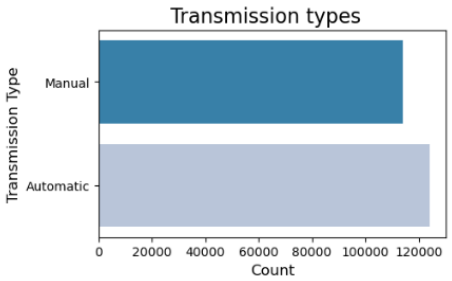




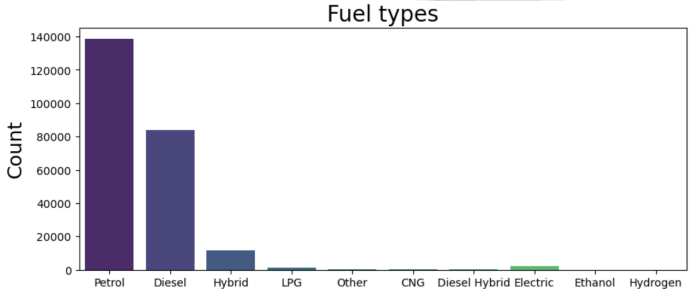
Brand Popularity:
•German brands dominate the used car market in Germany.
•The dataset includes over **30,000 Volkswagen**, more than **25,000 Mercedes-Benz**, and over **20,000 Opel, BMW, and Audi** vehicles.
•**Ford** is the next most common brand, showing the strong presence of both local and widely recognized brands in the market.

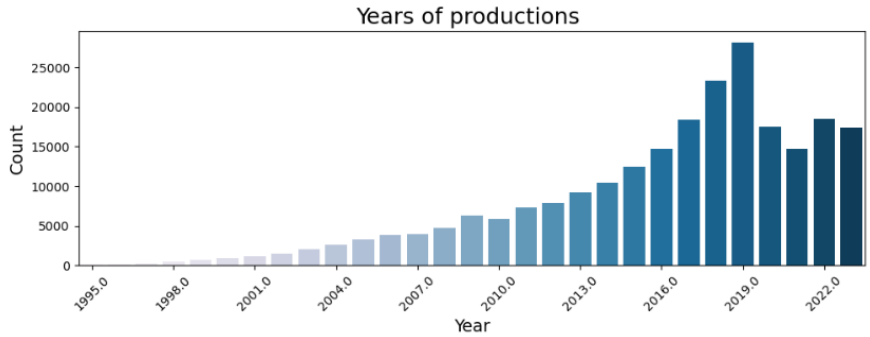
Most Popular Car Models:
•The **Volkswagen Golf** is the most popular used car model in Germany.
•Following the Golf, the top five models are the **Opel Astra, Opel Corsa, Seat Leon,** and **Ford Focus**.
•This reflects a trend towards smaller, practical cars on the secondary market.

Explorative Data Analysis



Transmission Types:
•**Automatic transmissions** are more common, but **manual transmission** vehicles are still highly present in the German used car market.
•This balance may reflect cultural preferences or specific market demands in Germany.

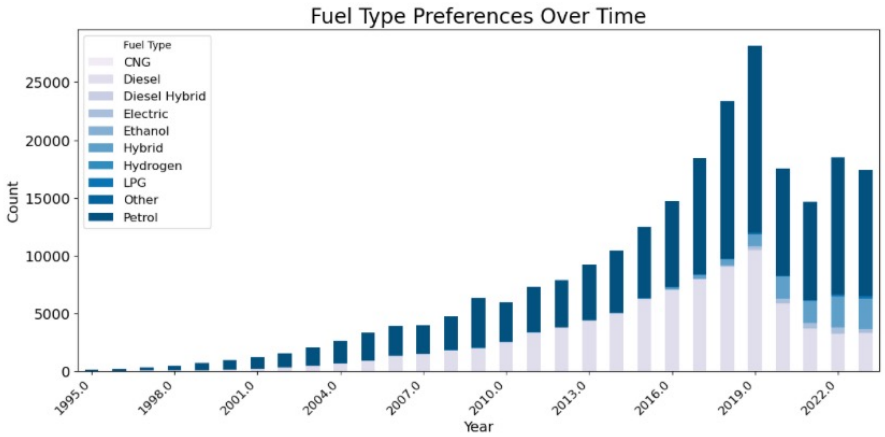




Fuel Types:
•**Petrol** cars are the most prevalent fuel type, followed by **diesel**.
•Although electric cars are gaining popularity, they still represent a small fraction of the used car market as of 2023. It would be interesting to compare these numbers with future datasets, as the market for electric vehicles continues to grow.

Car Age and Model Years:
•Most cars in the secondary market were produced in **2018-2019**, followed by very new cars from **2023**.
•There's a noticeable dip in the availability of cars produced during **2020-2021**. This could be due to the COVID-19 pandemic, which affected car production, imports, and demand globally.

Explorative Data Analysis: Fuel and transmission across the years



Fuel Type Trends:

•**Diesel and Petrol** are still the dominant fuel types in the secondary car market as of 2023.

•Up to **2018-2019**, diesel cars were on the rise, nearly matching petrol vehicles in volume (approaching a 50:50 ratio).

•After 2019, we observe a general decrease in production/import of vehicles. Diesel cars, in particular, start to decline in proportion.

•At the same time, **electric and hybrid vehicles** have begun to gain traction, reflecting a shift towards greener alternatives. This change is gradual but visible, marking a potential trend for future years.

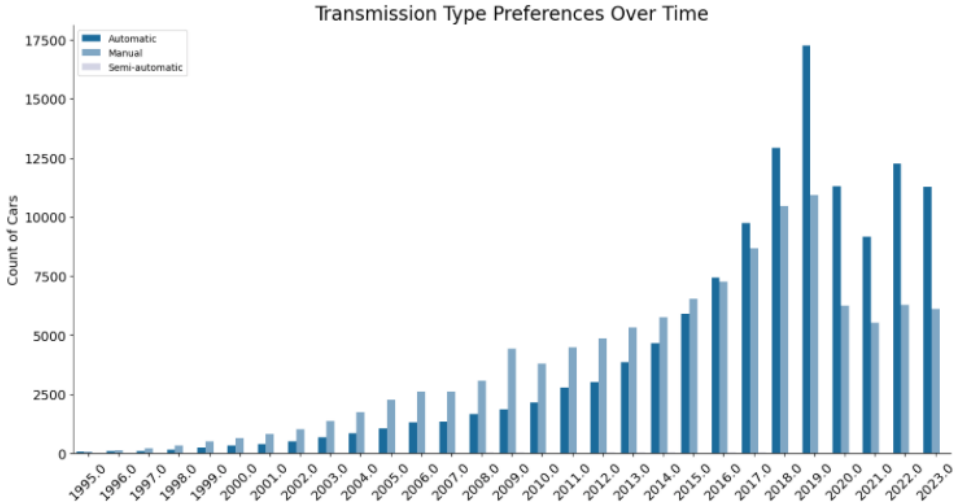
•**Key Insight**: The shift away from diesel cars and the rise in hybrid and electric vehicles likely reflects evolving consumer preferences and possibly new regulations.

Transmission Type Trends:

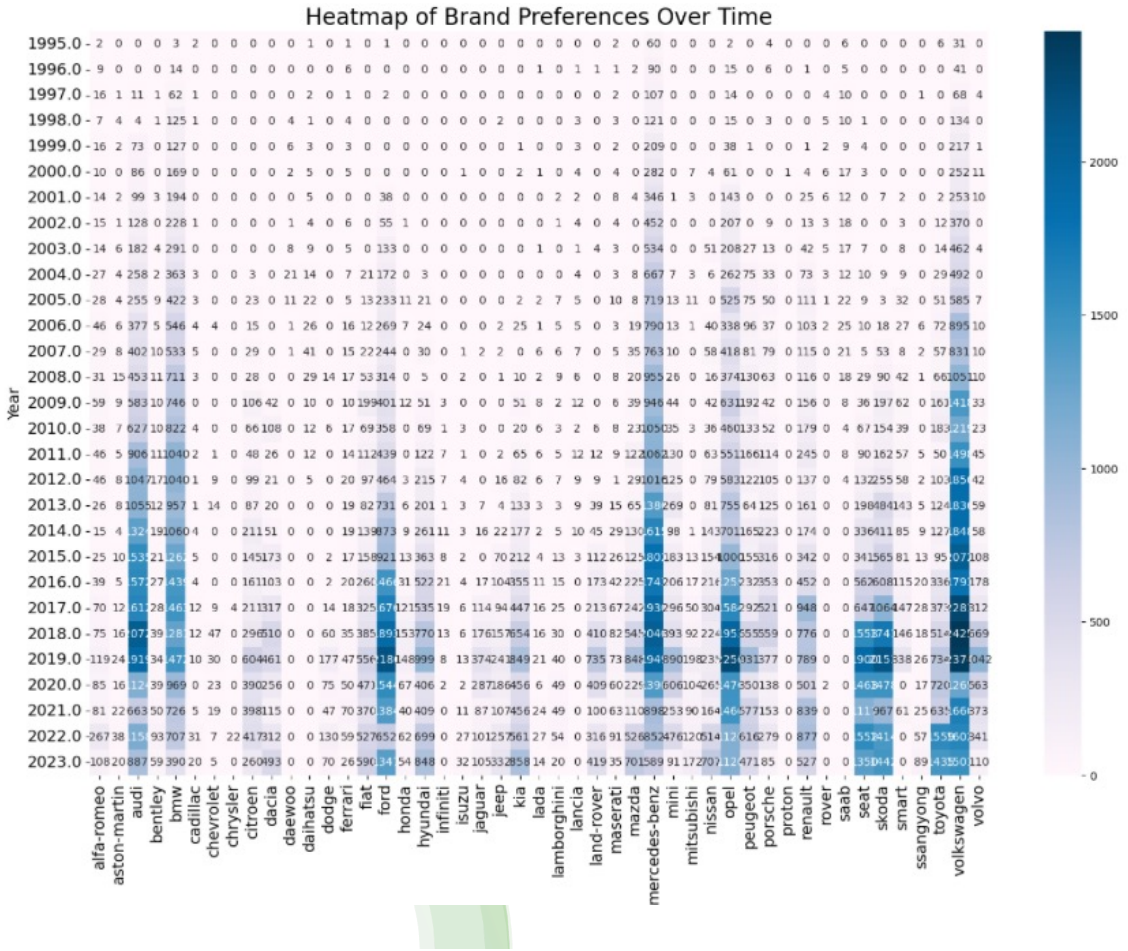
•Historically, **manual transmissions** were more prevalent in the German market. However, over time, there has been a noticeable shift towards **automatic transmissions**.

•This change is clearly observable, indicating that consumers may now prioritize convenience and ease of use over traditional preferences for manual transmissions.

•**Key Insight**: The rise in automatic transmissions could also relate to the growing presence of electric and hybrid cars, which are often available only with automatic transmissions.



Explorative Data Analysis: Brands across the years



Heat Map: Brand Imports Over Time

- The heat map reveals a decline in car imports on the German secondary market after the COVID-19 pandemic, showing the pandemic's impact on the availability of certain brands.
- While major brands like **BMW, Mercedes-Benz, Opel, and Volvo** have yet to recover to pre-pandemic levels in the secondary market, some brands have shown signs of recovery or even growth, notably **Seat, Skoda, Ford, Toyota, and Nissan**.
- **Volkswagen, Audi, Renault, Dacia, Kia, and Mazda** show a moderate recovery, suggesting a mixed impact of the pandemic on different segments and price tiers.
- These patterns highlight the shifting dynamics in the secondary market, with certain brands able to adapt more quickly to disruptions.

Explorative Data Analysis: Brands across the years (examples)

Brand Distribution Across Production Years (1995, 2010, and 2022)

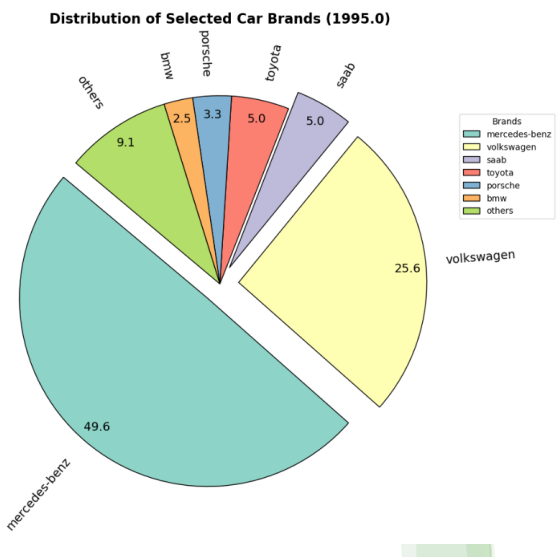
•**1995**: The market was heavily dominated by **Mercedes-Benz (49.5%)** and **Volkswagen (25%)**, with smaller shares from brands like **Saab, Toyota, Porsche, and BMW**.

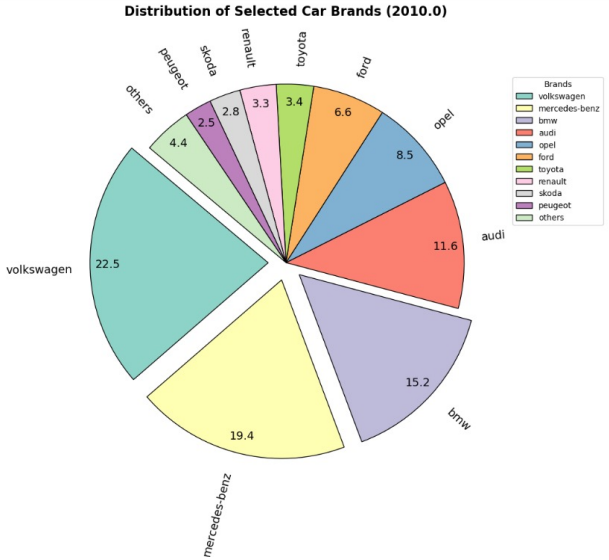
•**2010**: Distribution became more diversified, with **Volkswagen (22.5%)**, **Mercedes-Benz (19.4%)**, **BMW (15.2%)**, and **Audi (13.6%)** being the most common. Newer entrants like **Ford, Toyota, Renault, and Skoda** also gained a significant presence.

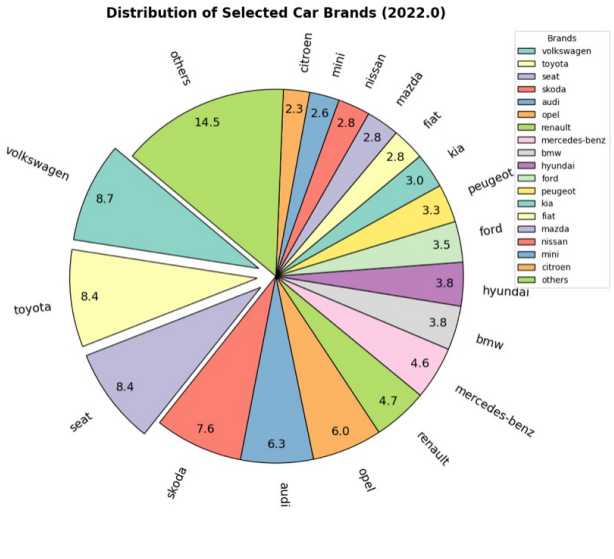
•**2022**: The brand landscape has further diversified, with **Volkswagen (8.7%)**, **Toyota (8.4%)**, **Seat (8.4%)**, and **Skoda (7.6%)** among the leaders. The "others" category (14.5%) indicates a broad range of brands now present in the secondary market.

•**Conclusion**: Over time, the dominance of traditional German brands like Mercedes-Benz, BMW, and Volkswagen has diminished, with newer brands gaining ground. This trend suggests a shift in consumer preferences and perhaps a response to the rising availability of international and more affordable brands.

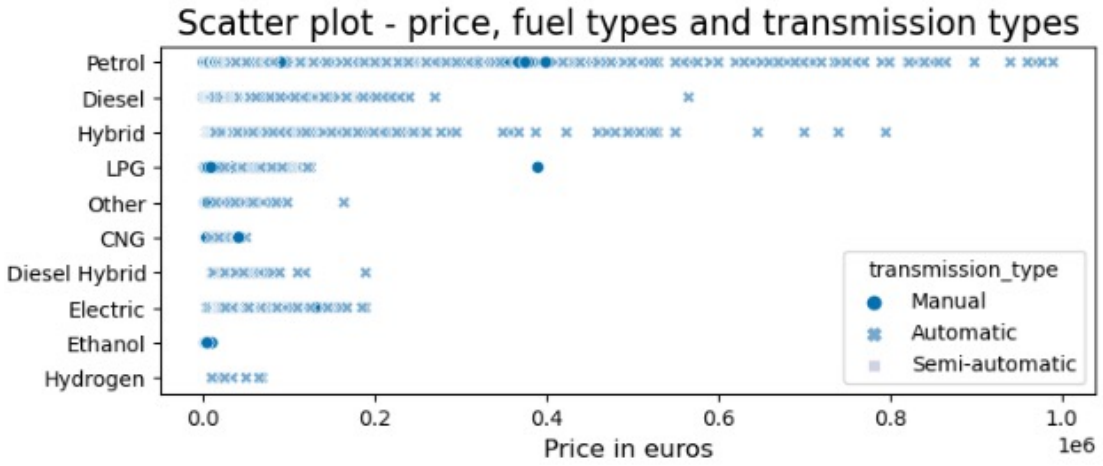
•The data indicates a marked impact of the pandemic on import patterns, with some brands adapting better than others. The recovery seen in brands like Seat, Skoda, and Toyota suggests resilience and perhaps strategic alignment with consumer demand in Germany.

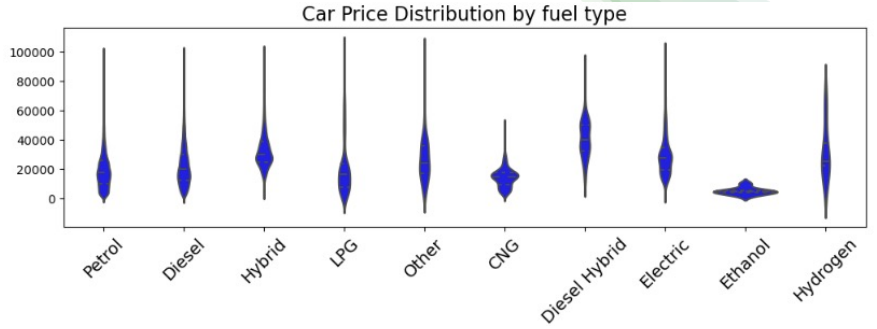






Explorative Data Analysis: transmission, fuel and the price





Price Analysis by Fuel and Transmission Type:

•Fuel Type and Price:

- Analysis of price distributions by fuel type shows that **petrol cars** have the largest price range—from very affordable to extremely expensive.
- **Diesel cars** can also be expensive but generally don't reach the highest price points seen in petrol vehicles.
- **Ethanol cars** are consistently among the least expensive options, indicating limited demand or lower resale value.
- **Hybrid and electric vehicles** tend to hold high prices on the secondary market, likely due to their advanced technology and lower supply.

Violin Plot: Price Distribution by Fuel Type

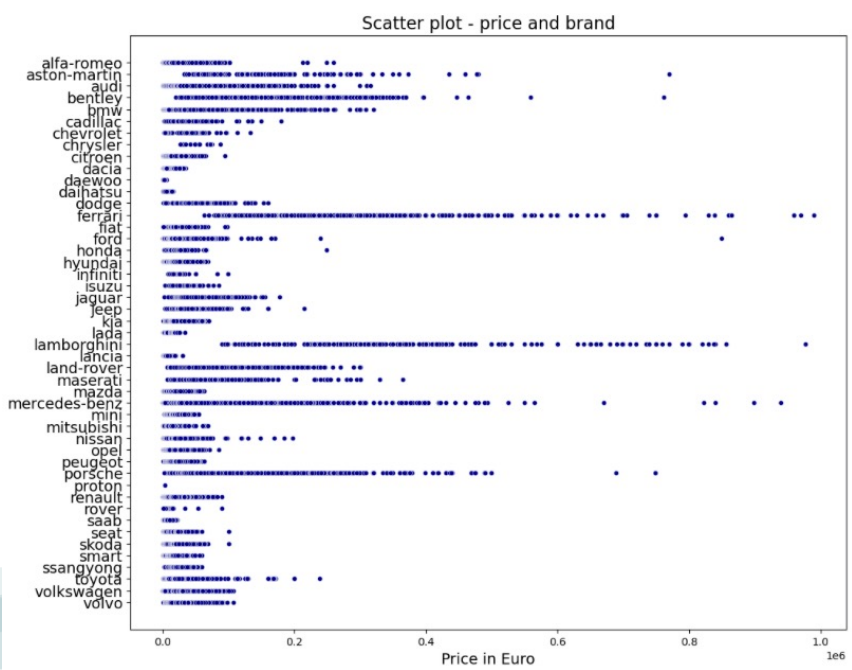
•Price distributions across fuel types reveal that **hybrid cars** tend to be the most expensive, while **LPG and CNG vehicles** are among the cheapest options, indicating a demand for more affordable and fuel-efficient alternatives.

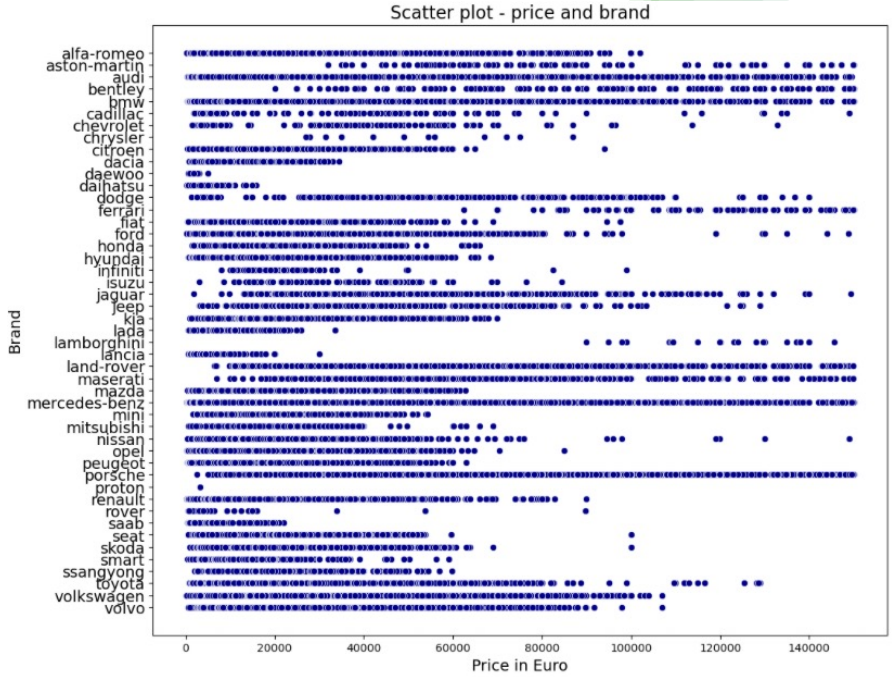
•**Diesel hybrids** also rank high in price, highlighting their value for long-distance drivers seeking fuel efficiency and performance.

•**Electric cars** show price levels similar to **petrol cars**, suggesting that electric vehicles are gaining affordability, likely due to increased market supply and consumer acceptance.

•**Conclusion**: Hybrid and diesel hybrid vehicles are generally more expensive, reflecting their perceived value and efficiency. Ethanol and LPG options, however, appeal to budget-conscious consumers.

Explorative Data Analysis: price and brand





The cars with a price < 150 000

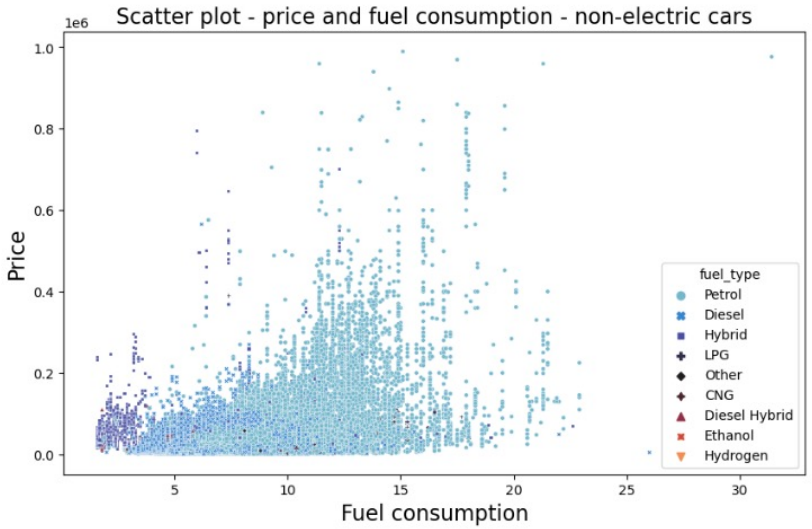
Brand and Price Analysis:

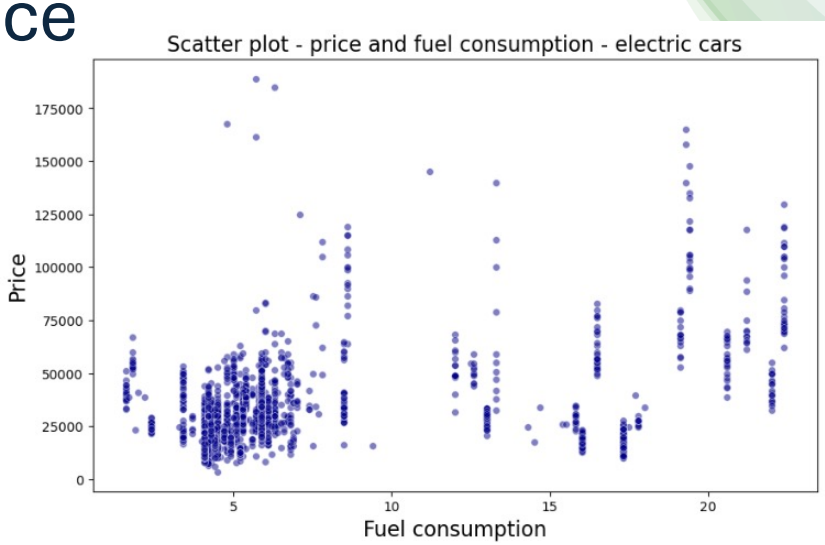
•Scatter plots of **price vs. brand** illustrate the diversity in brand positioning within the market:

- Some brands, like **Mercedes-Benz and BMW**, have a higher price span and more premium offerings, reflecting their brand value.
- **Volkswagen and Ford** cover a broader price range, with affordable to mid-range options, making them accessible across various budgets.
- Brands like **Opel** and **Fiat** are generally on the lower end, indicating that they're seen as more economical choices.

•**Key Insight**: Certain brands have a strong association with either high or low prices, which can influence consumer perceptions and expectations in the secondary market.

Explorative Data Analysis: fuel, its consumption and the price





Observations:

•In general, **price increases with fuel consumption up to around 15 liters per 100 km**. This initial increase is likely because powerful, high-performance cars—often luxury models—tend to have higher fuel consumption.

•After reaching this threshold, **price decreases as fuel consumption rises beyond 15 liters per 100 km**. This decline may reflect a diminishing demand for cars with excessive fuel consumption due to running costs or environmental concerns.

High fuel consumption is not exclusively linked to older vehicles; it also occurs in powerful, newer models with high-performance engines, which contributes to the higher price in this range. **Electric Cars**:

•The trend for electric vehicles is less clear due to the smaller sample size, mostly consisting of models like the Renault Zoe, Volkswagen e-Golf, and Hyundai electric models.

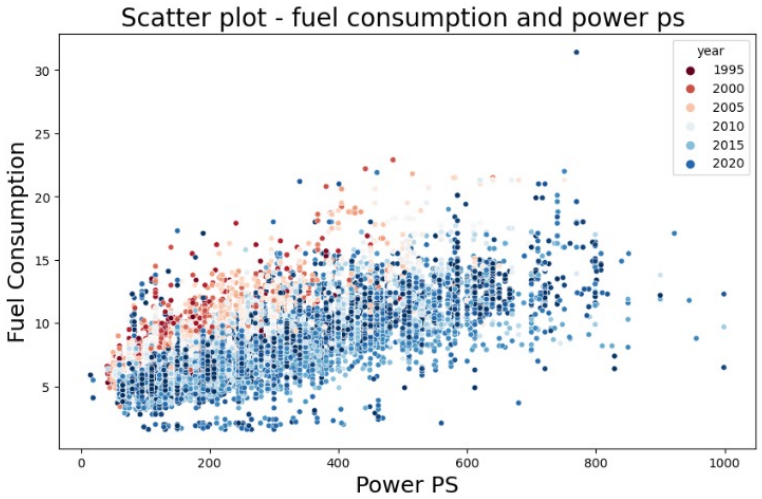
•With an increase in **electric car diversity**, including new high-performance options entering the market, we may see clearer trends in the future. Models like the Volkswagen ID.4, Hyundai Ioniq 5, and others could offer more insights into how consumption impacts pricing in the electric segment.

Insights:

•The observed trends suggest that **fuel consumption is a significant factor affecting price**, particularly in traditional fuel types.

•Consumers seem willing to pay more for high-performance cars up to a point, after which high fuel consumption appears to negatively impact perceived value.

Explorative Data Analysis: power , fuel_consumption across the years (Petrol)

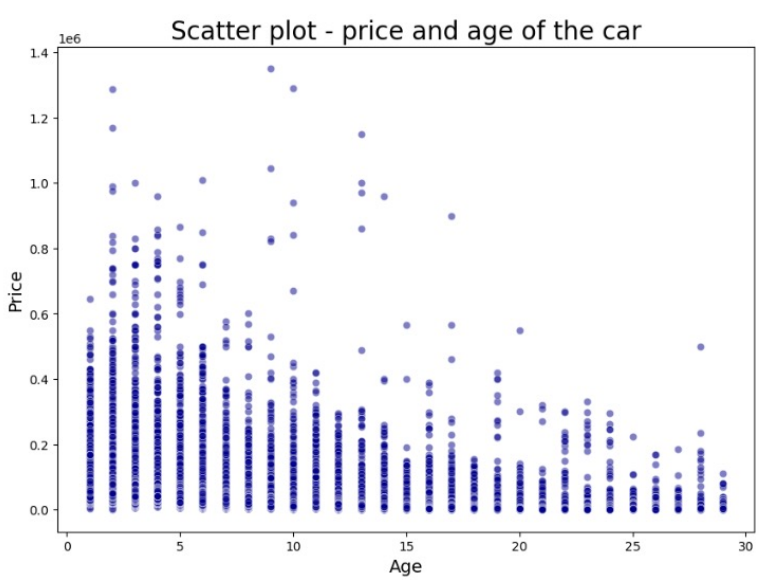


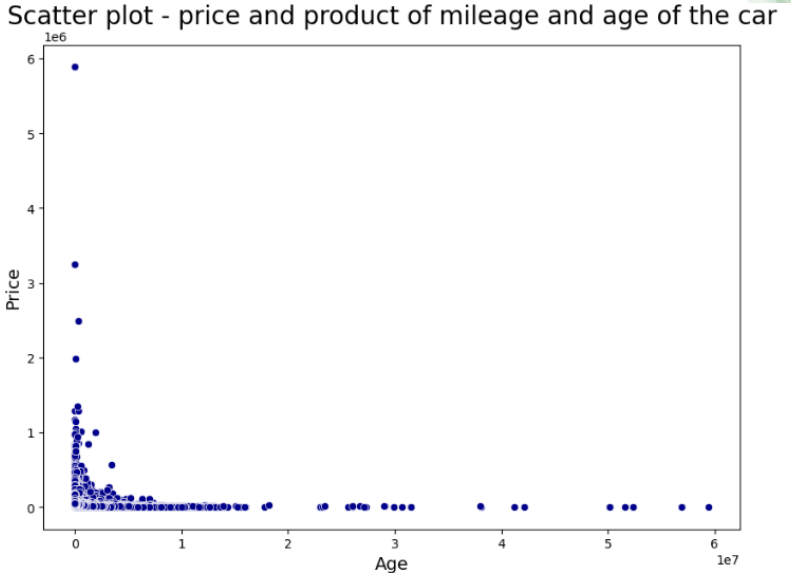
Fuel Consumption and Power:

•**Higher fuel consumption often correlates with higher engine power**. More powerful cars tend to consume more fuel, which can increase their price.

•Occasionally, **high fuel consumption also correlates with car age**, particularly in older, less efficient models.

Explorative Data Analysis: age , mileage and the price

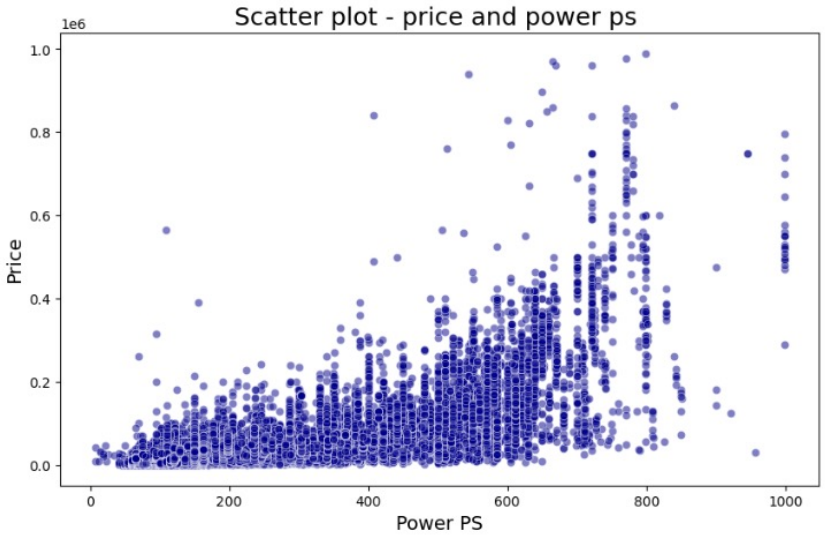


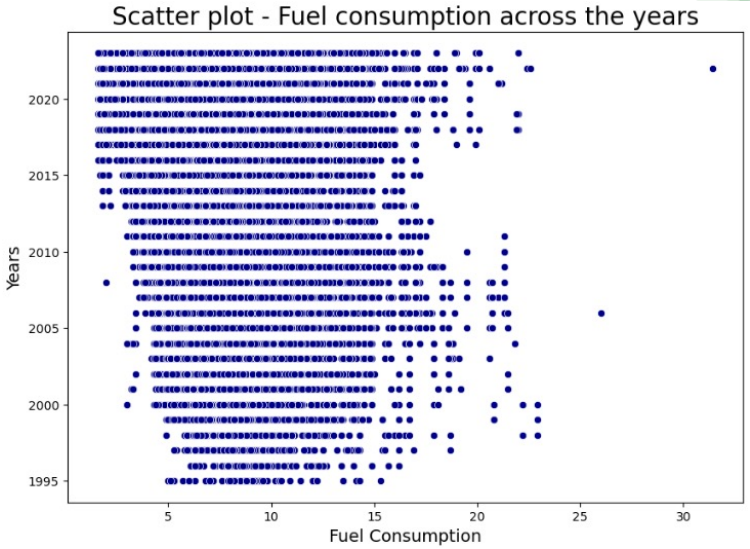


Age and Mileage vs. Price:

•**Older cars and those with higher mileage are generally cheaper**. This is a standard depreciation effect in the used car market, where age and usage reduce value.

Explorative Data Analysis: power (PS) and the price,

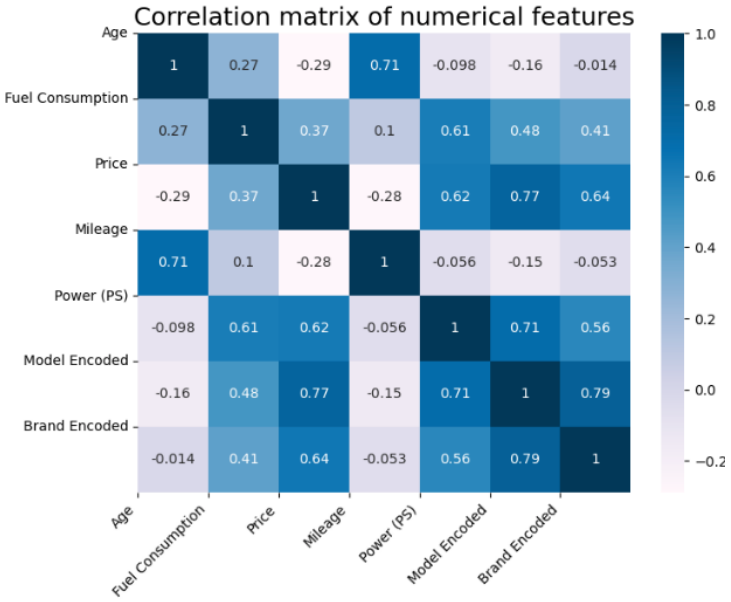




Power vs. Price:

•**Higher engine power generally means a higher price**. Most high-power cars are relatively new, indicating that powerful models tend to retain value.

Explorative Data Analysis: correlation matrix

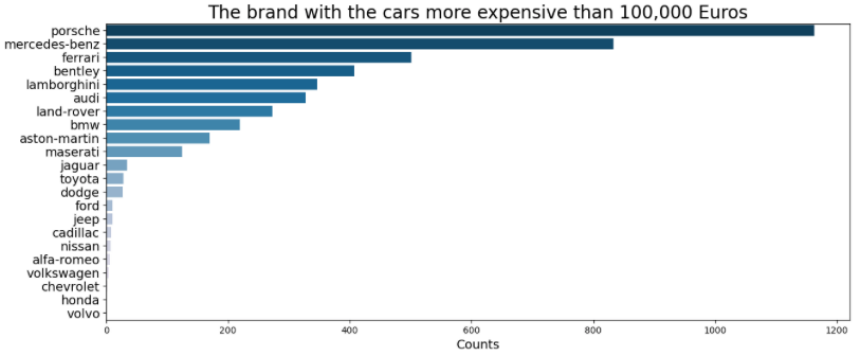


Correlation Matrix - Numerical Features vs. Price

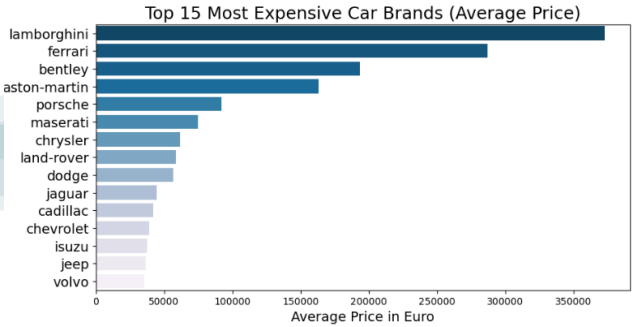
•Price Correlations:

- **Power (0.62), Model (0.77), and Brand (0.64)** have the highest positive correlations with price, suggesting these features are strong predictors.
- **Fuel Consumption (0.37)** also positively correlates with price, though less strongly.
- **Age (-0.29) and Mileage (-0.29)** show negative correlations with price, reflecting the impact of depreciation.

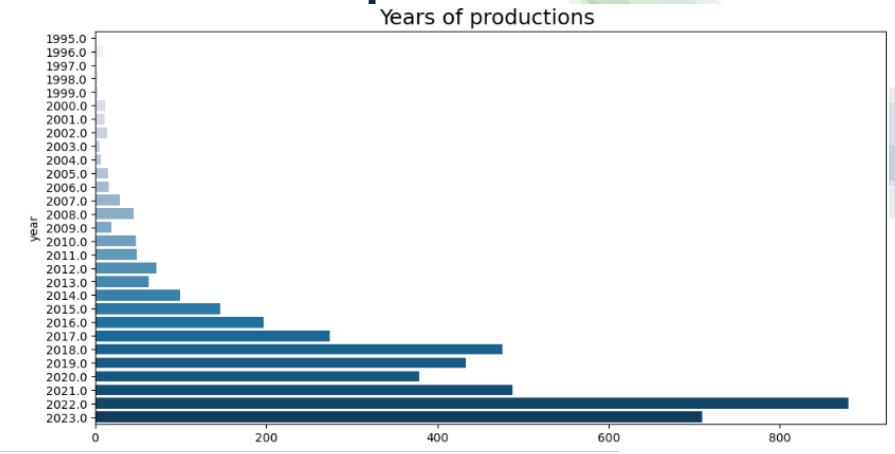
Explorative Data Analysis: the most expensive

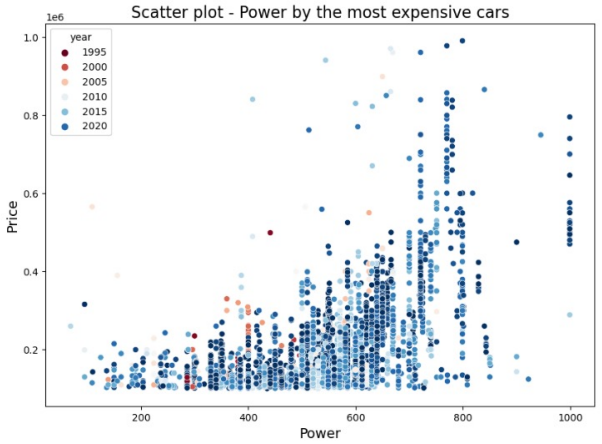


Among the most expensive cars (priced above €100,000), **Porsche, Mercedes-Benz, Ferrari, and Bentley** are the most represented brands, signaling their luxury status and demand on the secondary market.



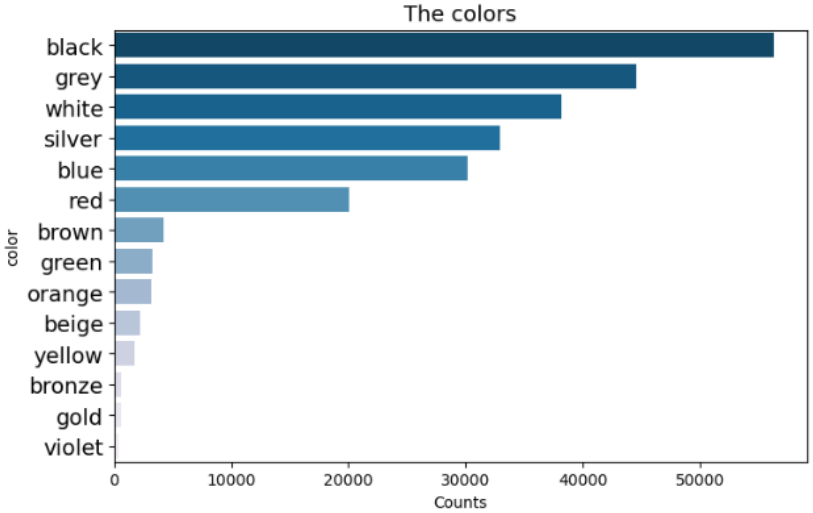
Average Price Leaders: **Lamborghini** leads with the highest average price, followed by **Ferrari** and **Bentley**, confirming their exclusive market positioning.

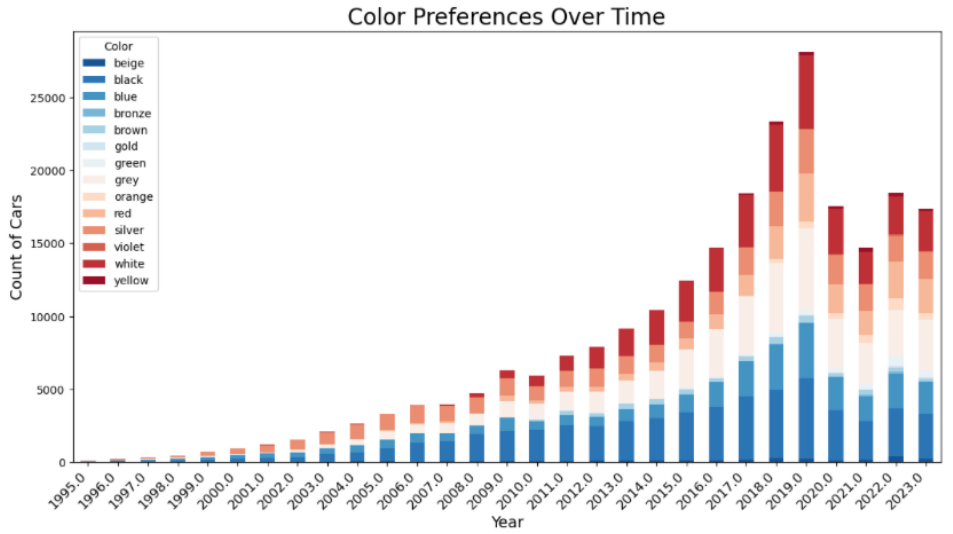




Additionally, the most expensive cars tend to be relatively new and feature powerful engines, as buyers in this segment value performance and modernity.

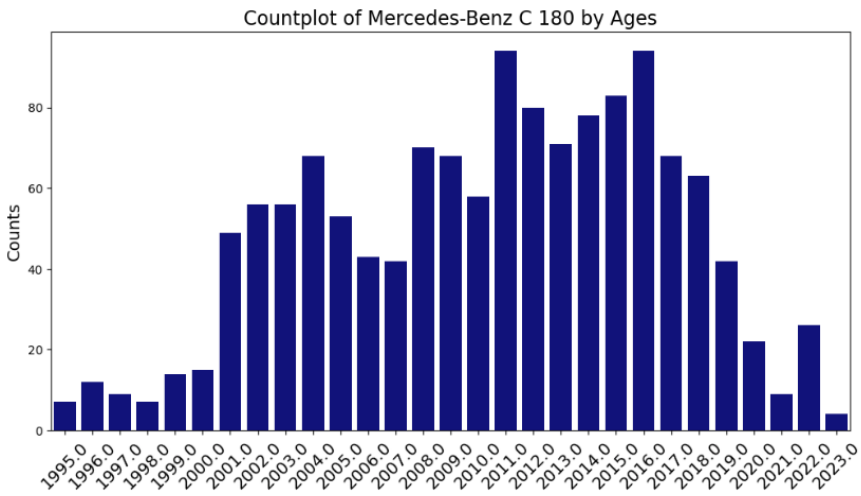
Explorative Data Analysis: colors

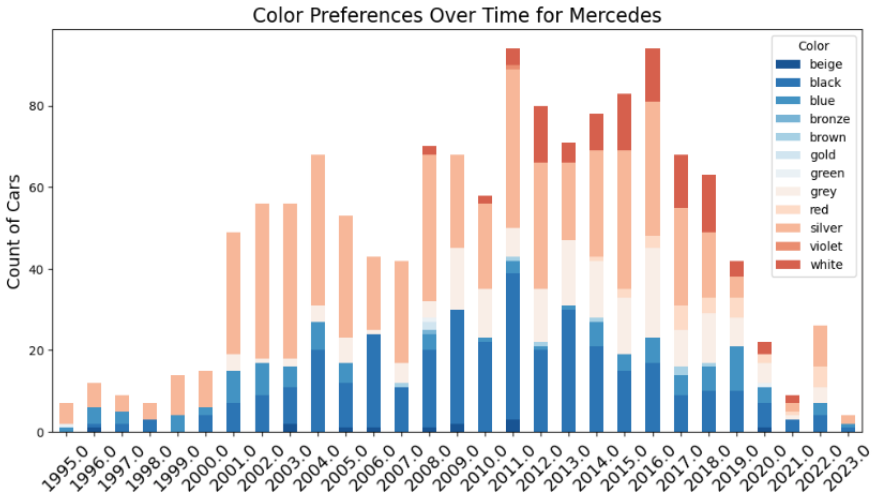




The count plot shows that **black cars** are the most common on the secondary market, followed by **grey, white, silver, blue, and red**. However, color trends have shifted over time.Up until 2007, black made up a large portion of the market (up to half of all cars). Since then, a shift has occurred, with **white, grey, blue, silver, and red** cars increasing in share. This evolution may reflect broader changes in consumer aesthetics and brand styling choices.

Explorative Data Analysis: colors (example Mercedes-Benz C 180)

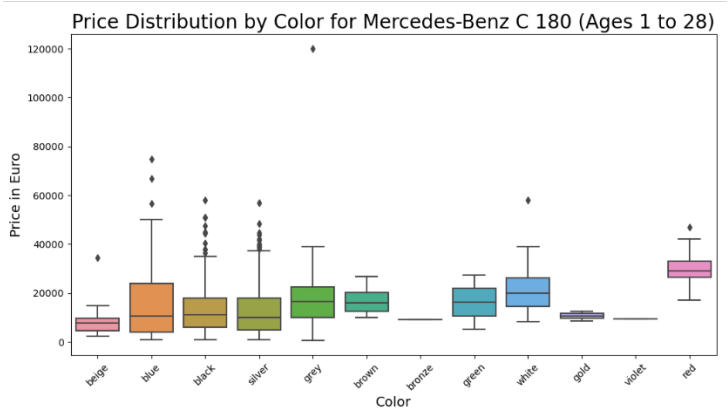




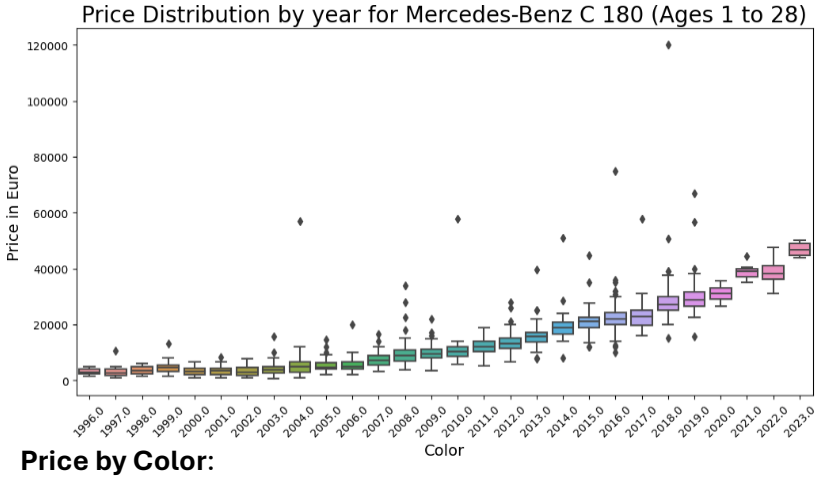
•For the Mercedes-Benz C180, there are **consistent high production volumes from 2002 to 2018** with a peak in recent years. **Color shifts** over time:Black and silver were dominant initially.

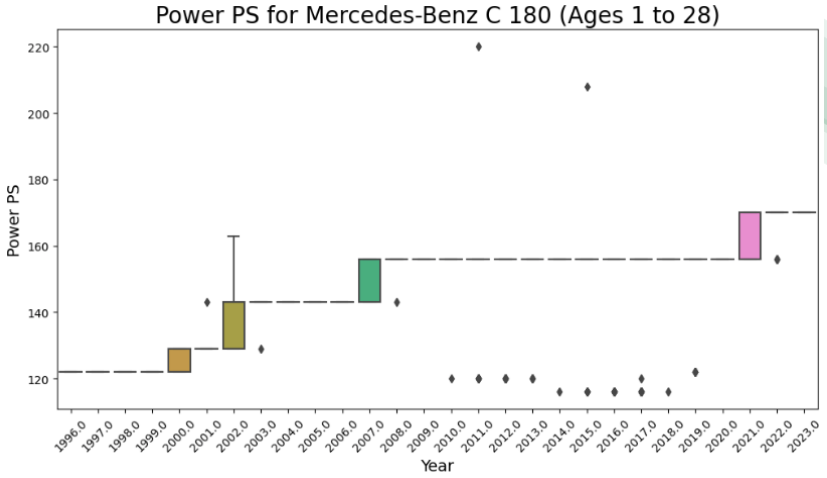
•Grey increased in later years, while unique colors like bronze, brown, and green appeared briefly around 2008.

•**Blue was more common in earlier and recent years**, while **red became more popular in later years**.



Explorative Data Analysis: colors (example Mercedes-Benz C 180)





Price by Color:

•Red Mercedes-Benz C180s are generally **more expensive**, but this is largely due to their being newer and more powerful models.

•**Power variations (box plot of power by color)** indicate that color alone is not enough to influence price; age and engine power also play critical roles

Whether color is important factor for the price? To isolate color's impact, **ANOVA tests were conducted** on specific models from the same year, controlling for age and power.

The ANOVA test results - colors, age for the price of 7-9 year old Mercedes-Benz C180

	sum_sq	df	F	PR(>F)
color_brown	1.888184e+06	1.0	0.056576	0.812195
color_grey	2.991986e+07	1.0	0.896495	0.344676
color_red	9.733874e+07	1.0	2.916580	0.088967
color_white	1.525834e+07	1.0	0.457189	0.499592
color_blue	2.104274e+08	1.0	6.305079	0.012698
color_silver	2.732696e+07	1.0	0.818803	0.366438
age	1.689663e+08	1.0	5.062771	0.025352
Residual	8.009826e+09	240.0	NaN	NaN

The ANOVA test results - colors, age for the price of 8 year old Mercedes-Benz C180

	sum_sq	df	F	PR(>F)
color_blue	6.897688e+08	1.0	16.142094	0.000122
color_grey	6.495555e+07	1.0	1.520101	0.220815
color_red	1.014478e+08	1.0	2.374099	0.126872
color_silver	1.095695e+08	1.0	2.564164	0.112815
color_white	3.703246e+07	1.0	0.866640	0.354378
Residual	3.845796e+09	90.0	NaN	NaN

BLUE color seems to have a **significant effect on the price** of the Mercedes-Benz C180, meaning BLUE cars are priced differently (likely higher or lower depending on the direction of the effect) compared to other colors.

Explorative Data Analysis: colors (other examples)

The ANOVA test results - colors, age for the price of 6 year old Mercedes-Benz E200

	sum_sq	df	F	PR(>F)
color_black	3.473418e+09	1.0	29.524426	7.354370e-07
color_blue	9.678574e+08	1.0	8.226892	5.430995e-03
color_bronze	2.974793e+08	1.0	2.528606	1.162422e-01
color_brown	3.664204e+08	1.0	3.114613	8.189344e-02
color_grey	1.380571e+09	1.0	11.735001	1.023189e-03
color_red	1.622852e+09	1.0	13.794418	4.028578e-04
color_silver	2.174963e+09	1.0	18.487424	5.356345e-05
color_white	1.421897e+09	1.0	12.086281	8.708101e-04
Residual	8.352835e+09	71.0	NaN	NaN

The ANOVA test results - colors, age for the price of 1 year old Volkswagen Golf

	sum_sq	df	F	PR(>F)
color_blue	1.971840e+08	1.0	2.266735	0.134384
color_grey	3.884295e+08	1.0	4.465204	0.036329
color_red	8.746243e+07	1.0	1.005427	0.317695
color_silver	3.083110e+08	1.0	3.544200	0.061785
color_white	4.000830e+08	1.0	4.599167	0.033678
Residual	1.243961e+10	143.0	NaN	NaN

The ANOVA test results - colors, age for the price of 5 years old Ford Focus

	sum_sq	df	F	PR(>F)
color_black	1.104018e+08	1.0	4.825082	0.028446
color_blue	1.046222e+08	1.0	4.572488	0.032909
color_bronze	2.276968e+07	1.0	0.995143	0.318907
color_brown	6.462171e+07	1.0	2.824276	0.093390
color_grey	1.466306e+08	1.0	6.408456	0.011622
color_orange	3.389722e+08	1.0	14.814696	0.000132
color_red	1.114606e+08	1.0	4.871359	0.027699
color_silver	3.762212e+07	1.0	1.644265	0.200256
color_violet	2.276968e+07	1.0	0.995143	0.318907
color_white	6.437618e+07	1.0	2.813545	0.094013
color_yellow	2.276968e+07	1.0	0.995143	0.318907
Residual	1.320222e+10	577.0	NaN	NaN

The ANOVA test results - colors, age for the price of 5 years old Opel Astra

	sum_sq	df	F	PR(>F)
color_blue	4.661771e+06	1.0	0.695802	0.404540
color_brown	3.895810e+05	1.0	0.058148	0.809533
color_green	8.341494e+05	1.0	0.124503	0.724329
color_grey	1.862763e+07	1.0	2.780304	0.095967
color_red	3.470575e+06	1.0	0.518008	0.471981
color_silver	5.172836e+07	1.0	7.720821	0.005635
color_white	6.498658e+06	1.0	0.969970	0.325096
Residual	3.906014e+09	583.0	NaN	NaN

The ANOVA test results - colors, age for the price of 10 years old Audi A3

	sum_sq	df	F	PR(>F)
color_blue	8.950592e+07	1.0	8.847459	0.003190
color_brown	2.794534e+07	1.0	2.762334	0.097622
color_grey	5.476948e+07	1.0	5.413840	0.020689
color_red	1.999829e+07	1.0	1.976786	0.160834
color_silver	8.045229e+07	1.0	7.952527	0.005144
color_white	1.752746e+07	1.0	1.732549	0.189158
Residual	2.842756e+09	281.0	NaN	NaN

The ANOVA test results - colors, age for the price of 8 years Porsche Cayenne

	sum_sq	df	F	PR(>F)
color_blue	8.852848e+07	1.0	0.228530	0.633532
color_grey	6.348352e+08	1.0	1.638779	0.203092
color_silver	3.831779e+09	1.0	9.891450	0.002118
color_white	2.819896e+08	1.0	0.727935	0.395342
Residual	4.416166e+10	114.0	NaN	NaN

In some cases, color showed a statistically significant impact on price for certain models and years. For this reason, the colors stay in dataset for machine learning.

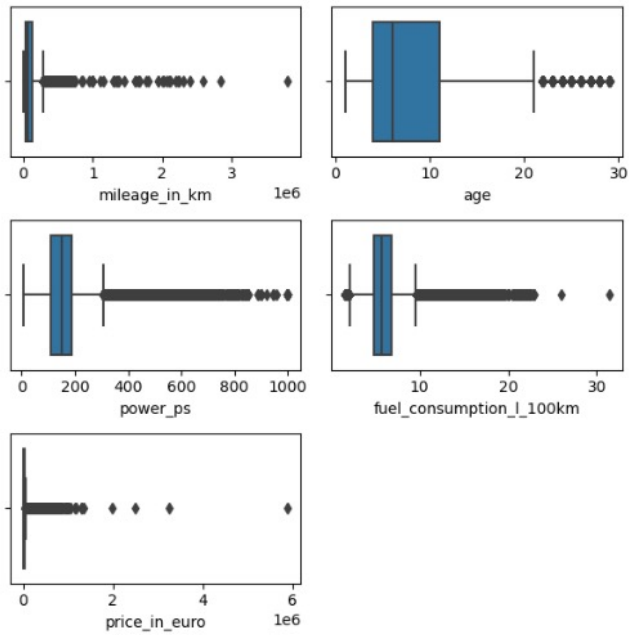
Data Distribution: numerical features (mileage, price, power (PS), fuel consumption, age)

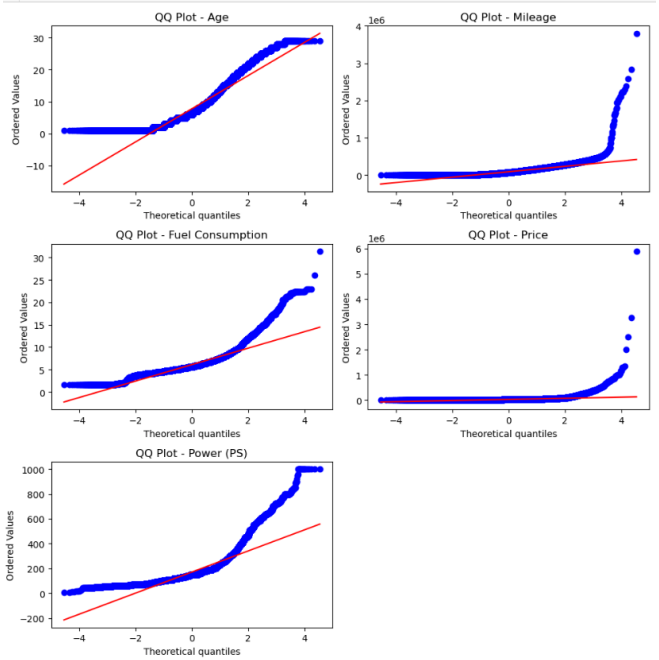
Boxplot:

•A boxplot visually represents the distribution of data and highlights **summary statistics** such as the. **median**, **quartiles** (25th and 75th percentiles), and **potential outliers**.

Q-Q Plot (Quantile-Quantile Plot):

•A Q-Q plot compares the **quantiles** of the data's distribution against the quantiles of a theoretical distribution (often **normal distribution**) to assess whether the data follows that distribution.





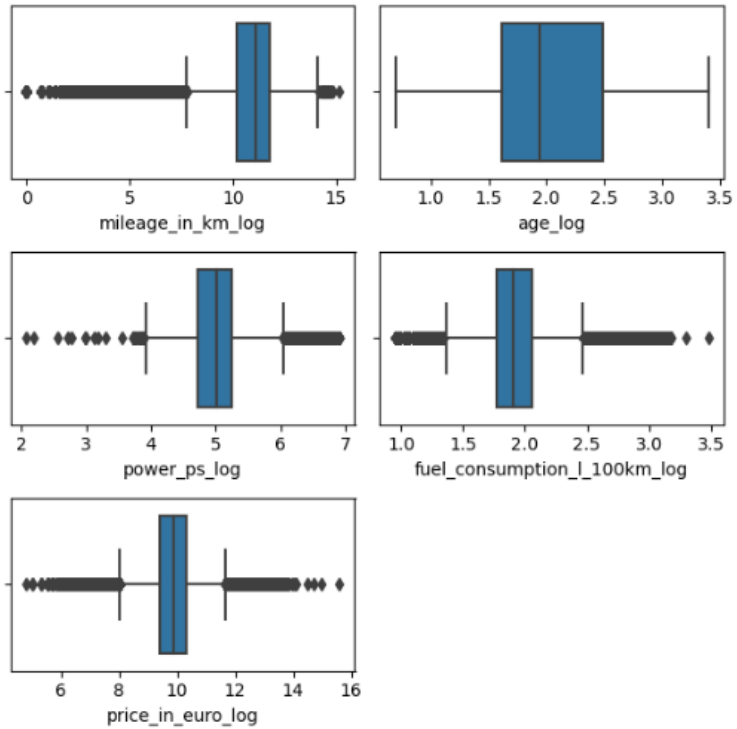
All these numerical values have outliers: especially heavy for price, mileage, power and fuel consumption.

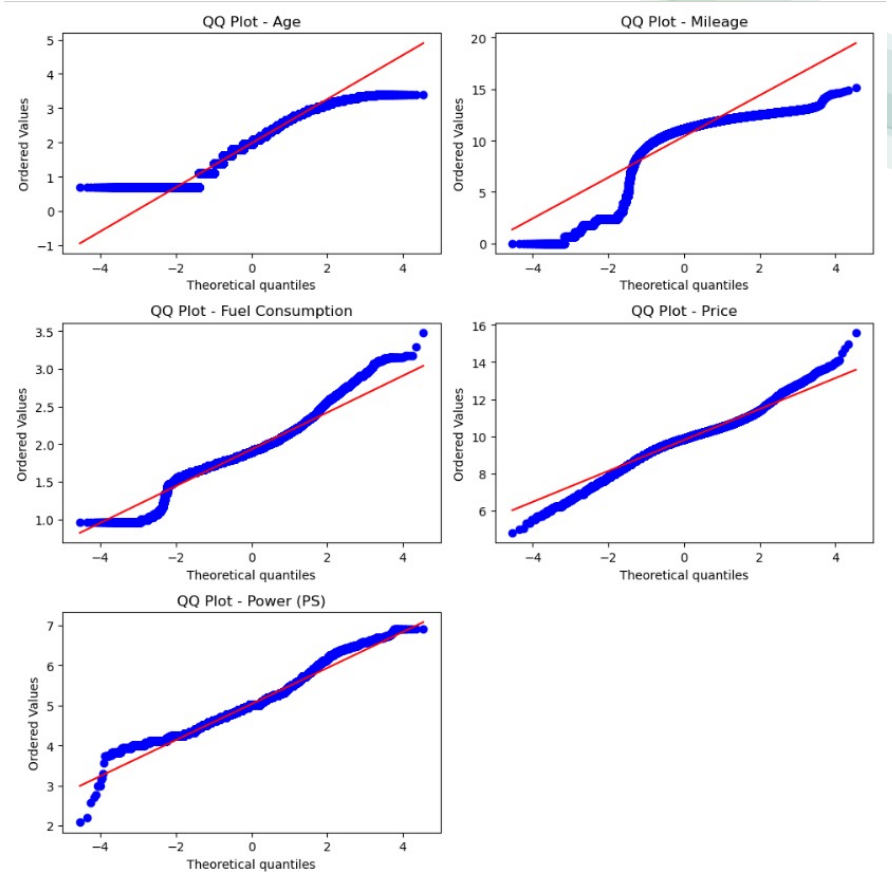
Q-Q plots additionally demonstrate

This – see the deviations from the line, which shows theoretical Distribution.

Especially, mileage and price have the long tail outside the red line, indicating massive outliers on high values.

Data Distribution: application of log transformation

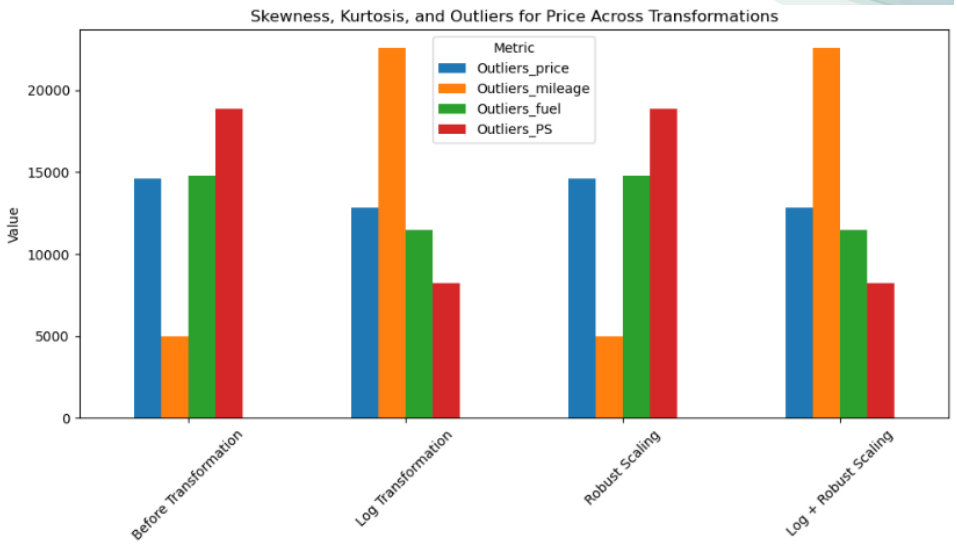




Log transformation changed the data distribution. Only for mileage it has worked strangely. It moved the mileage outliers to the other side.

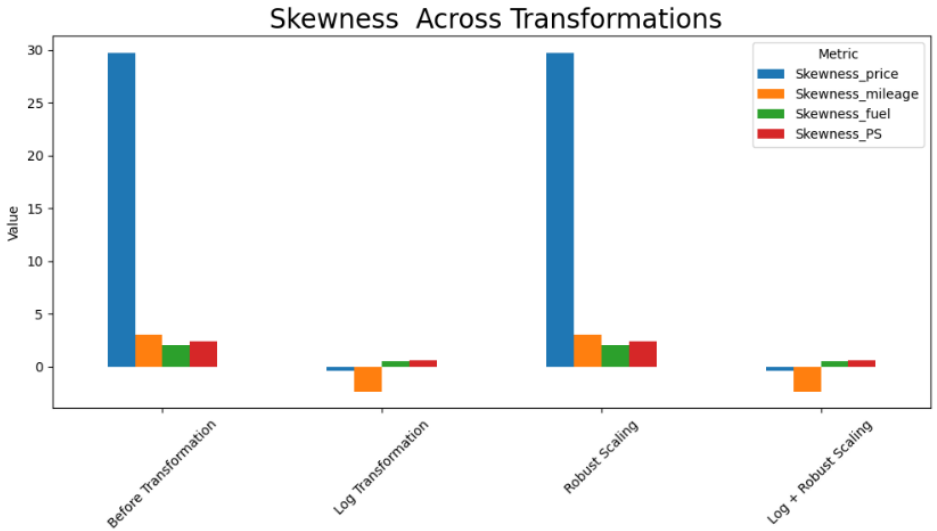
Data Distribution: numerical features (mileage, price, power (PS), fuel consumption, age) – skewness, kurtosis and outliers number

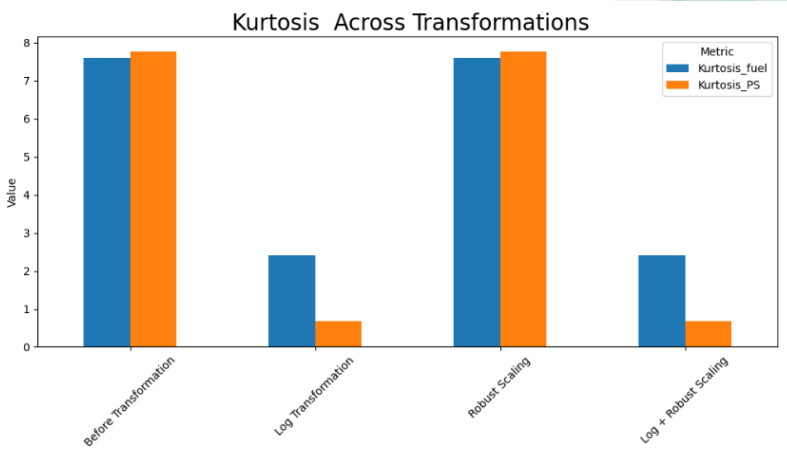
	Metric	Before Transformation	Log Transformation	Robust Scaling	Log + Robust Scaling
0	Skewness_price	29.6955	-0.4048	29.6955	-0.4048
1	Kurtosis_price	3362.5264	1.606	3362.5264	1.606
2	Outliers_price	14628	12835	14628	12835
3	Skewness_mileage	3.0727	-2.3221	3.0727	-2.3221
4	Kurtosis_meliage	61.7246	4.998	61.7246	4.998
5	Outliers_mileage	4980	22553	4980	22553
6	Skewness_fuel	2.0493	0.4915	2.0493	0.4915
7	Kurtosis_fuel	7.59	2.418	7.59	2.418
8	Outliers_fuel	14793	11456	14793	11456
9	Skewness_age	1.0306	-0.2635	1.0306	-0.2635
10	Kurtosis_age	0.6615	-0.5772	0.6615	-0.5772
11	Outliers_age	N/A	N/A	N/A	N/A
12	Skewness_PS	2.4044	0.6157	2.4044	0.6157
13	Kurtosis_PS	7.7664	0.6835	7.7664	0.6835
14	Outliers_PS	18871	8227	18871	8227



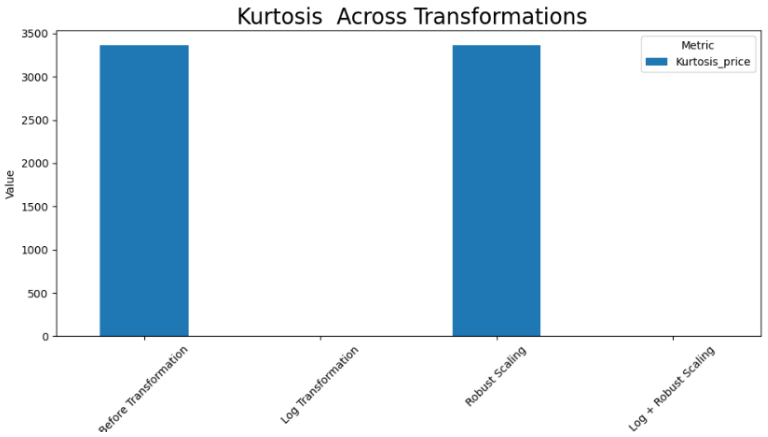
The number of outliers, skewness and kurtosis were tested as well, before log transformation, afterwards, and after scaling. Only mileage outliers were increased after log transformation

Data Distribution: numerical features (mileage, price, power (PS), fuel consumption, age) – skewness, kurtosis and outliers

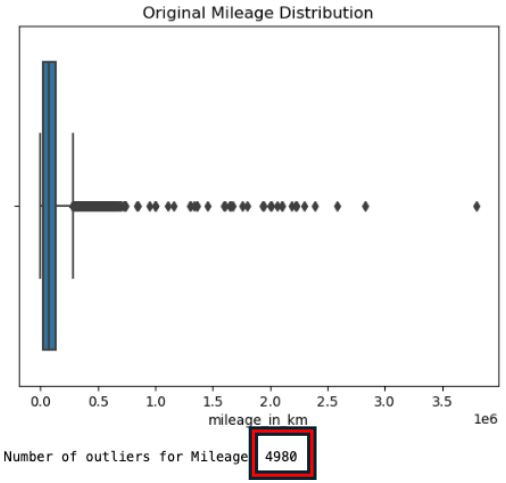


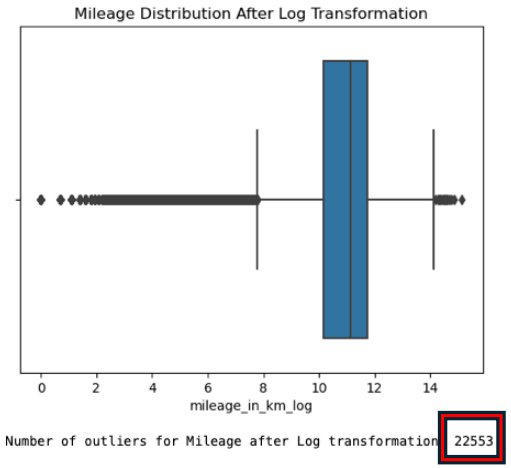


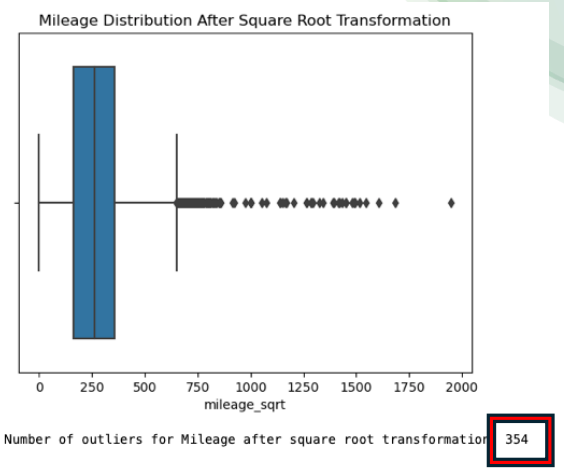
Skewness of price, mileage, fuel consumption and power_ps reduced drastically after log transforamtion, the same for kurtosis

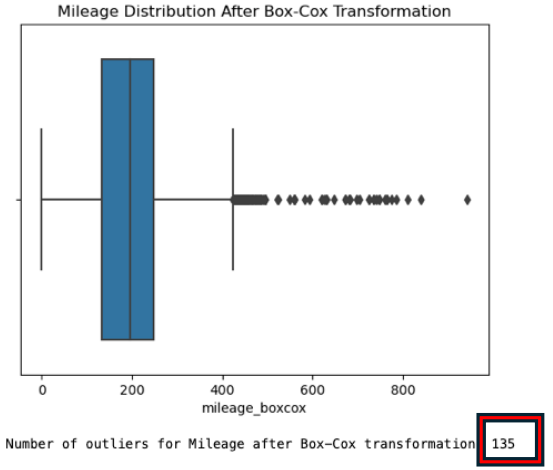


Data Distribution: mileage – other transformations









For the milieage was chosen another type of transformation,
square root transformation, which worked better and improved distribution
(the number of outliers was reduced)
of mileage considerably

Modelling

Data Preparation and Modeling Approach
To prepare the data for modeling, two cleaned versions of the dataset were created, with specific handling for **fuel consumption** and **extreme values**:
1.Dataset Version 1: For entries with missing or abnormally low fuel consumption values (below 1.6), these values were imputed. Missing values were replaced with either the average fuel consumption for cars of the same model and year or, if unavailable, the mode for the respective category.
2.Dataset Version 2: In this version, entries with missing fuel consumption data or values below 1.6 were removed entirely to ensure a more conservative dataset without imputation.
Both datasets retained **all outliers** related to mileage and price, as these extremes might carry valuable insights for certain car models. However, the degree and handling of extreme values for each model (e.g., mileage and price limits) were determined empirically, based on preliminary evaluations to assess their impact on model performance.

Modeling and Hyperparameter Tuning
To evaluate these datasets and gain an initial understanding of model performance, each dataset was tested in a pipeline with various machine learning models (Linear Regression. With log and square root transformation, kNN with Robust Scaling, Decision Trees, random Fporest and XGBost), using **GridSearchCV** for hyperparameter tuning. This allowed for systematic screening of model parameters and provided preliminary insights into expected model accuracy and robustness.

This structured approach ensures that both the impact of outliers and the handling of missing data are thoroughly considered in the model selection and tuning process.

Metrics of Model Perfomance Evaluation:

1. <u>Mean Squared Error (MSE)</u>
•**Description**: Measures the average of the squares of the errors (the average squared difference between the predicted and actual values).
•**Formula**: MSE=1n∑i=1n(yi−y^i)2MSE=n1∑i=1n(yi−y^i)2
•**Use Case**: MSE gives higher weight to larger errors due to the squaring effect, making it useful if you want to penalize large errors significantly.
•**Scikit-learn Scoring Parameter**: Use 'neg_mean_squared_error' (the negative is used because GridSearchCV tries to maximize the score).

2. <u>Root Mean Squared Error (RMSE)</u>
•**Description**: The square root of the mean squared error, which provides an error metric in the same units as the target variable.
•**Formula**: RMSE=1n∑i=1n(yi−y^i)2RMSE=n1∑i=1n(yi−y^i)2
•**Use Case**: RMSE is also sensitive to large errors but is interpretable in the same units as the target variable.

•**Scikit-learn Scoring Parameter**: Scikit-learn does not have a built-in scoring function for RMSE, but you can use 'neg_root_mean_squared_error'.
3. <u>Mean Absolute Error (MAE)</u>
•**Description**: Measures the average absolute difference between the predicted and actual values.
•**Formula**: MAE=1n∑i=1n|yi−y^i|MAE=n1∑i=1n|yi−y^i|
•**Use Case**: MAE treats all errors equally, making it less sensitive to outliers compared to MSE.
•**Scikit-learn Scoring Parameter**: Use 'neg_mean_absolute_error'.

4. <u>R^2 Score (Coefficient of Determination)</u>
•**Description**: Measures how well the predicted values explain the variance of the actual values. R^2 ranges from 0 to 1, with higher values indicating better model performance.
•**Use Case**: While R^2 is a useful measure of how well the model fits the data, it does not directly measure error and can be misleading if used alone, especially for models with non-linear relationships.
•**Scikit-learn Scoring Parameter**: Use 'r2'.

Model Screening Using GridSearchCV and Pipelines

	Best Estimator	n_estimators	max_depth	n_neighbors	learning_rate	Score	MAE	MSE
Full dataset	KNeighborsRegressor	NA	NA	3	NA	0.8030	4110	172252763
Full dataset without outliers	XGBRegressor	200	20	NA	0.1	0.9107	2790	33357651
Dropped dataset	KNeighborsRegressor	NA	NA	3	NA	0.7600	3956	115936049
Dropped dataset without outliers	XGBRegressor	100	10	NA	0.1	0.9070	2771	31503649

I screened a range of models to evaluate predictive performance for price prediction.

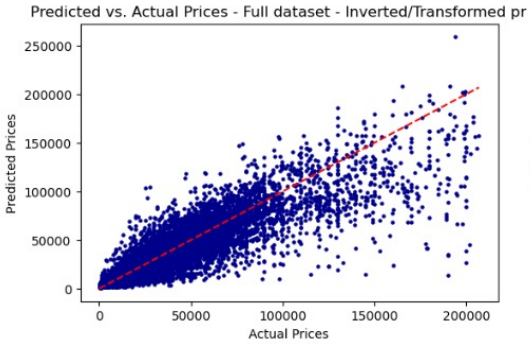
•A pipeline with GridSearchCV was used to streamline hyperparameter tuning and ensure consistency across models.
•The best performing model on the dataset without outliers was **XGBoost**, which minimized error metrics, particularly MAE.
•For the datasets with outliers, **k-Nearest Neighbors (k-NN)** demonstrated the best performance, suggesting robustness to extreme values in the data.
•In overall, all the models performed better on the datasets without outliers, however, outliers represent luxury segment of cars, they are real values, and I wanted to keep them

•As a next step, all these models were studied separately, the parameters were additionally fine-tuned, the allowance of extreme values was defined , and error analysis was performed

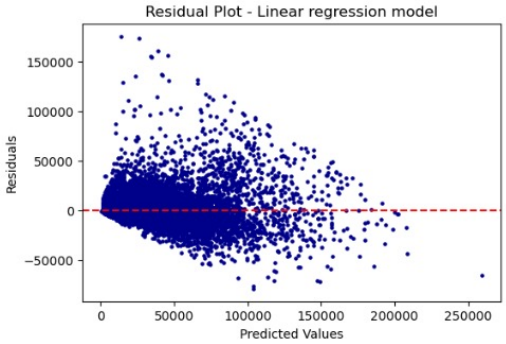
Linear Regression Model

•Linear regression served as a baseline for comparison. It is simple, interpretable, and provided an initial performance benchmark.
•**Best Metrics**: R^2 = 0.819
• MAE = 4510, 105
• MSE = 76914723
Price Range : up to 200 000 Euro

•The model was quick to fit, but residual plots show a trend of underperforming with high prices, indicating limitations in capturing non-linear relationships in the data. It works as well on quite narrow price range, and at the same having MAE higher than the other models.



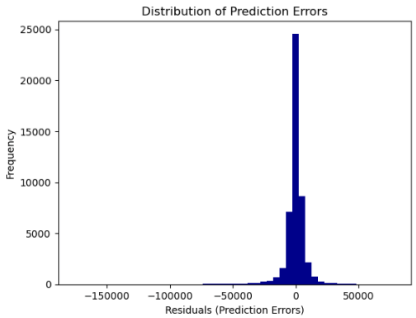
The **Predicted vs. Actual Price Plot** is a very useful diagnostic tool in regression analysis. It visualizes how well the model's predicted values align with the true values (actual prices) in the dataset.



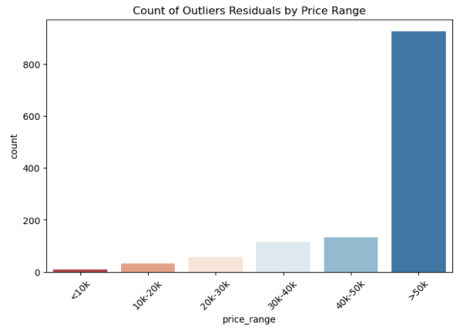
This **residual plot** for the **Linear Regression model** shows the difference between the actual and predicted values (residuals) plotted against the predicted values. The residuals are scattered around the horizontal line at zero, but there is a noticeable fan-shaped pattern as predicted values increase, indicating **heteroscedasticity**. This suggests that the model's performance decreases as the predicted values get larger

	Actual Price	Predicted Price	Residuals	Abs_residuals	Modell_encoded	Year	Power	model
0	47990.0	34478.601412	-13511.398588	13511.398588	10.173946	2021.0	200.0	Peugeot 508
1	12999.0	12903.994202	-95.005798	95.005798	9.531071	2015.0	132.0	Toyota Auris
2	28940.0	25287.635736	-3652.364264	3652.364264	10.231388	2021.0	122.0	Toyota Corolla
3	11870.0	8385.519322	-3484.480678	3484.480678	9.114380	2017.0	69.0	Citroen C1
4	8000.0	8724.687809	724.687809	724.687809	9.470520	2011.0	105.0	Alfa Romeo Giulietta
...	...	...	...	...	...	...	...	...
47273	32380.0	33802.837297	1422.837297	1422.837297	9.940437	2022.0	122.0	Volkswagen Caddy
47274	29490.0	36657.417421	7167.417421	7167.417421	10.239262	2022.0	150.0	SEAT Ateca
47275	17490.0	12474.274572	-5015.725428	5015.725428	9.307559	2018.0	101.0	Ford C-Max
47276	53990.0	51818.468708	-2171.531292	2171.531292	10.516170	2022.0	177.0	Toyota Proace
47277	15100.0	18187.734886	3087.734886	3087.734886	9.908221	2017.0	150.0	BMW 218

47278 rows × 8 columns



The **histplot** of prediction errors provides an overview of how the errors are distributed. A normally distributed error would indicate that the model's predictions are, on average, unbiased and the error variance is consistent



The most of extreme errors were for price over 50000. The number of outliers for errors was 1271.

Linear regression assumes a linear relationship, which might not capture the true patterns in the data (e.g., the relationship between car age, mileage, and price isn't always linear).

k-Nearest Neighbors (k-NN) – Non-Parametric Model

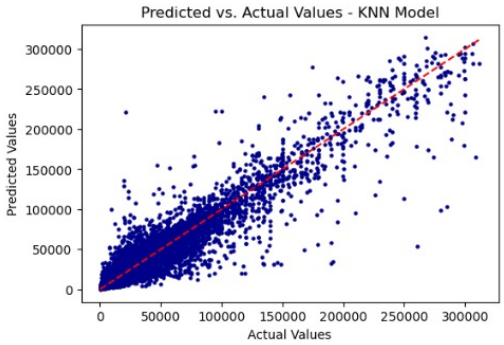
The **KNN** model often performs better than **Linear Regression** in scenarios where the relationship between features and the target variable (in this case, price) is **non-linear** or where the data does not follow a normal distribution. **KNN** is a **non-parametric model** that doesn't assume a specific form of the relationship between features and the target variable. Instead, it looks at the **k-nearest neighbors** and uses their labels (in this case, price) to predict the value for the new data point. This flexibility allows **KNN** to adapt better to the **non-linear patterns** in the data.

•**Metrics**: R^2 = 0.908
MAE = 3396,97
MSE = 55301256

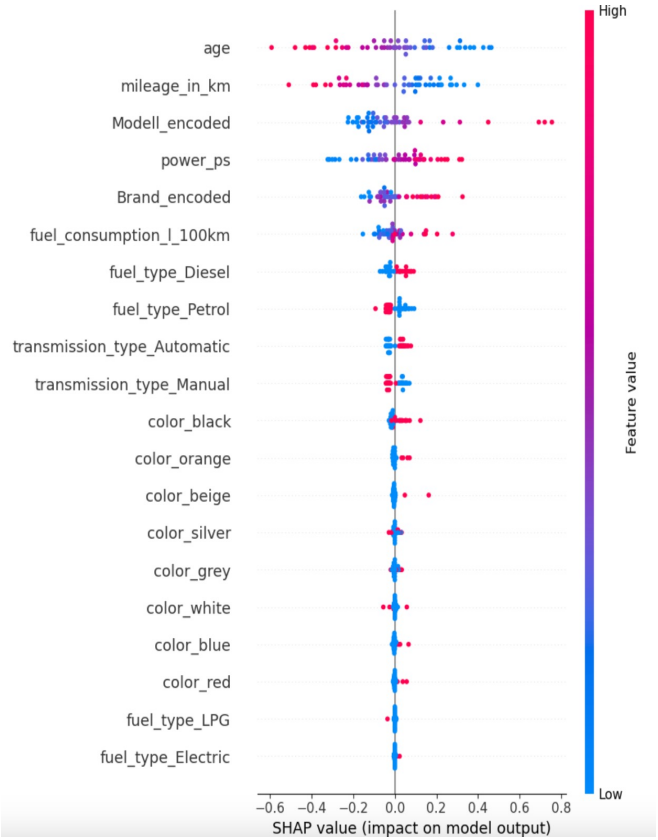
Price Range up to 320 000 Euro

	Actual Price	Predicted Price	Residuals	Abs_residuals	Modell_encoded	Age	Power	model
0	59900.0	59982.452352	82.452352	82.452352	113987.316378	17.0	385.0	Aston Martin V8
1	13750.0	13885.732900	135.732900	135.732900	9972.500162	6.0	75.0	Volkswagen Polo
2	7500.0	6704.944311	-795.055689	795.055689	12268.885175	14.0	101.0	NaN
3	19470.0	22240.541297	2770.541297	2770.541297	28201.139419	5.0	150.0	Volkswagen T-Roc
4	64450.0	60986.774941	-3463.225059	3463.225059	37938.626692	1.0	204.0	Land Rover Discovery Sport
...	...	...	...	...	...	...	...	...
47467	19990.0	19010.238752	-979.761248	979.761248	24236.931264	8.0	192.0	BMW 220
47468	16900.0	17823.089452	923.089452	923.089452	12938.947857	2.0	91.0	Renault Clio
47469	14990.0	16045.427007	1055.427007	1055.427007	18916.626652	5.0	131.0	Peugeot 308
47470	7990.0	9068.298880	1078.298880	1078.298880	17230.525216	14.0	120.0	Citroen Grand C4 Picasso
47471	12499.0	17898.931784	5399.931784	5399.931784	14343.185637	3.0	71.0	Fiat 500

47472 rows × 8 columns



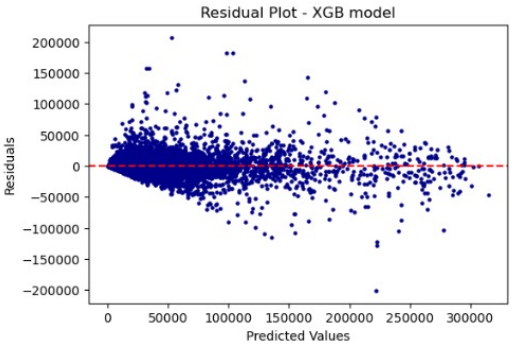
This plot show how k-NN performs better than the linear regression. The points are not so far away from the red line and distributed quite strictly along it.



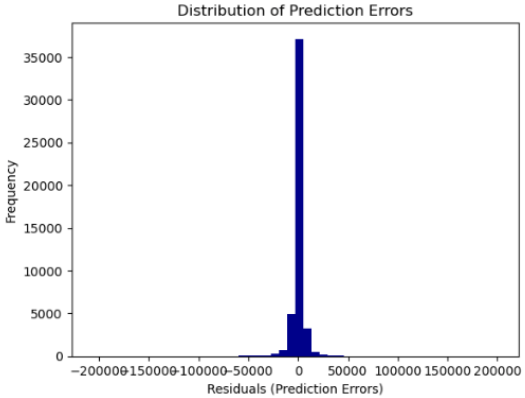
k-NN metrics is very good, it is similar to that obtained with Random Forest and XGBoost, however, the predictions price range is lower. At the same time it outperformed linear regression, as expected, and deciosion tree. **KNN** is a **lazy learner** that doesn't involve explicit training, and it can naturally handle **noisy data**better. It simply memorizes the dataset and makes predictions based on the nearest neighbors without constructing complex decision boundaries. Since it makes predictions based on local patterns, it's less likely to **overfit** compared to Decision Trees. On the other hand, **Decision Trees** may become very sensitive to outliers, especially if the tree grows deep without regularization, leading to overfitting or biased splits.

The SHAP plot indicates that age, mileage, model, power, and brand are the primary factors affecting car price predictions.
Secondary factors like transmission type, fuel type, and color have some influence but are relatively minor.

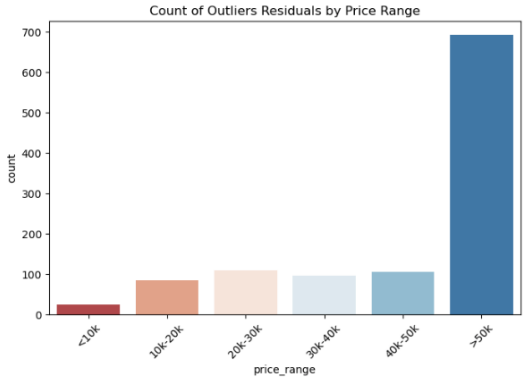
k-Nearest Neighbors (k-NN) – Errors



The **residuals** (the differences between predicted and actual values) for **KNN** are likely less **spread** out than those from **Linear Regression**.



The **histplot** of residuals for **KNN** show a **mores symmetric and centered** distribution around zero, indicating a more balanced error across all data points.



The **count of outliers (1100)** revealed that a relatively small percentage of the predictions were extreme, but these outliers were important to investigate further. A significant finding from the outlier analysis was that most of the **outliers in prediction errors** came from the **high-priced cars**. A **count plot** of these outliers showed a clear concentration of errors in the higher price range, which could indicate that the model struggles with predicting prices for expensive cars

Decision Trees – Interpretable Model with Non-Linear Boundaries

- Decision Trees captured non-linear relationships effectively, providing easily interpretable splits and feature importance.
- **Metrics**: R2 = 0.87
  MAE = 3940,12
  MSE = 55208830

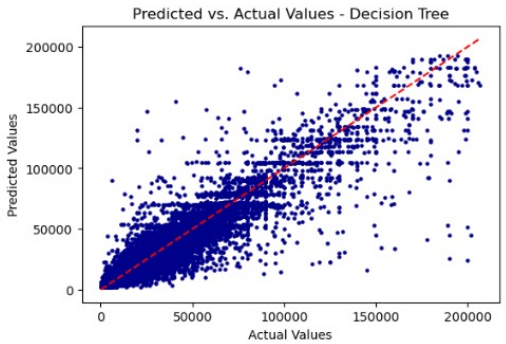
Range price up to 200000

- Limitation: High variance risk (overfitting) in larger trees, leading to poor generalization.

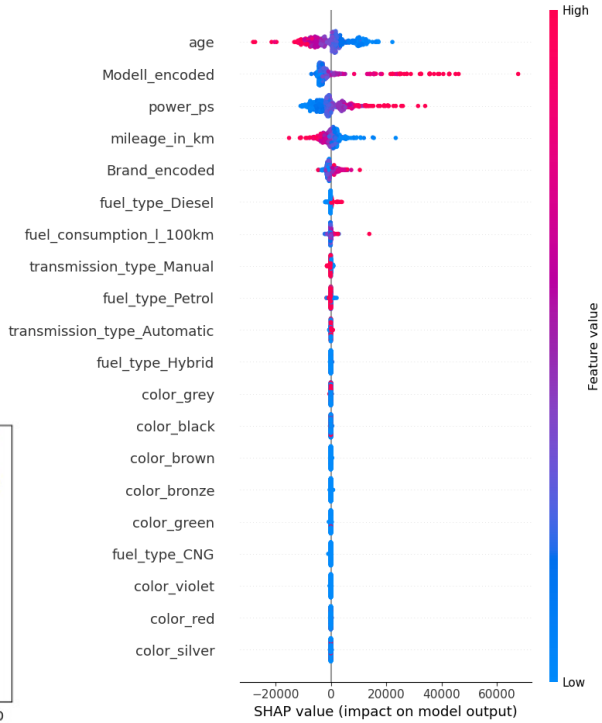
	Actual Price	Predicted Price	Year	Power	Mileage	Modell_encoded	model
0	47990.0	30482.580411	2021.0	200.0	47300.0	26210.293333	Peugeot 508
1	12999.0	14058.421403	2015.0	132.0	51959.0	13780.348168	Toyota Auris
2	28940.0	29289.631214	2021.0	122.0	2000.0	27760.009862	Toyota Corolla
3	11870.0	8444.576271	2017.0	69.0	84000.0	9084.004119	Citroen C1
4	8000.0	7132.576318	2011.0	105.0	139000.0	12970.635106	Alfa Romeo Giulietta
...	...	...	...	...	...	...	...
47273	32380.0	38449.471074	2022.0	122.0	12200.0	20751.802191	Volkswagen Caddy
47274	29490.0	34650.776402	2022.0	150.0	15500.0	27979.466908	SEAT Ateca
47275	17490.0	16711.819506	2018.0	101.0	50890.0	11020.018018	Ford C-Max
47276	53990.0	52161.494048	2022.0	177.0	7193.0	36906.486726	Toyota Proace
47277	15100.0	16654.043557	2017.0	150.0	133000.0	20093.894558	BMW 218

47278 rows × 7 columns

Definitely, Decision Trees performs better that the linear regression. Decision trees can model complex, non-linear relationships between features and the target (car price). Decision trees tend to overfit more and miss finer, local relationships that KNN can capture by looking at the nearest neighbors. KNN is simpler in concept and doesn't require explicit model training. It works by storing the data and making predictions based on proximity, making it more flexible. A decision tree requires building a model structure, and its complexity grows with more features and deeper trees, which can lead to overfitting.

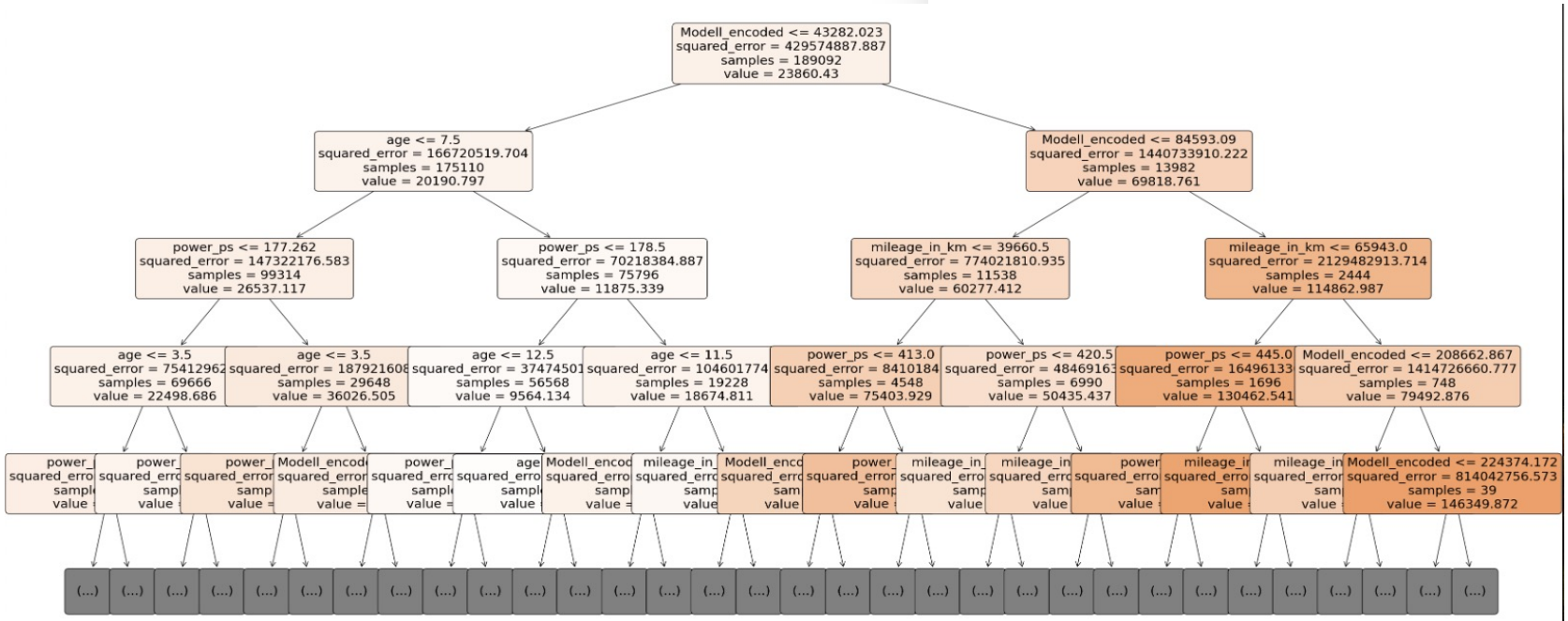


This plot show how decision trees performs . The points are quite far away from the red line , however distributed better along than the linear regression



The main predicting features according to
SHAP plot are ‘age’, ‘model’, ‘power’, ‘mileage’, ‘brand’. The secondary features have little effect
on prediction.

Decision Tree- how to it works



At the top of the tree, we can see that the first decision is made based on the **Model** feature. If the value of the **Model** is less than **43282**, the tree branches down to one group, whereas values greater than **43282** lead to another group. This is the first step in segmenting the dataset into different regions based on car model price thresholds.

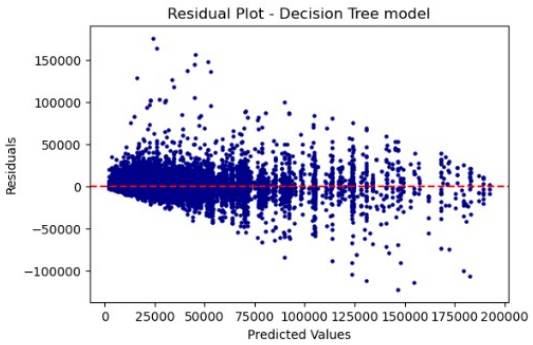
For the next split:

•In the **Model < 43282** group, the tree then checks the **Model** again but with a more specific threshold of **84593**. This decision further divides the dataset based on different **car model categories**.

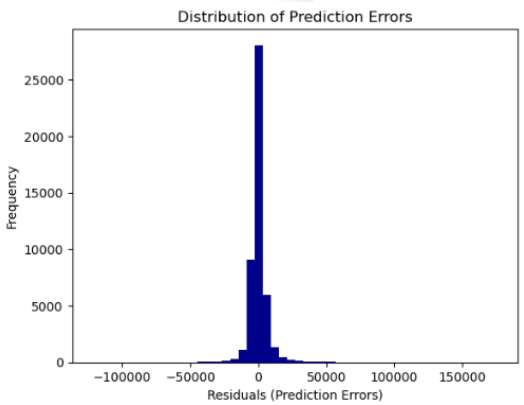
•After this, the tree looks at **Age** and further divides cars with **Age < 7.5 years** into one branch, showing that newer cars tend to be priced differently from older ones.

This process of **recursive splitting** continues by checking additional features and thresholds (such as more price-based splits or other characteristics like mileage or condition), ultimately leading to terminal **leaf nodes** where predictions of car prices are made.

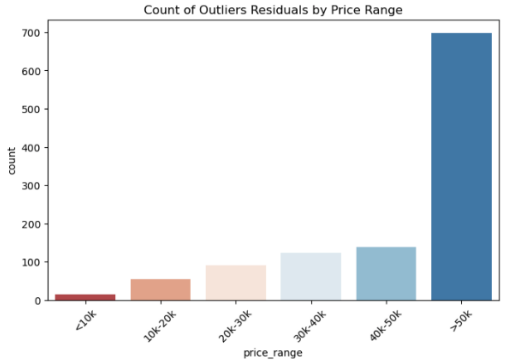
Decision Tree – Errors



The residuals have too many points far away from zero line. However, the distribution is symmetrical. It shows some level of heteroscedasticity as well.



The **histplot** of prediction errors provides an overview of how the errors are distributed. The distribution is normal, however, less centered, than the same for the KNN model.



The **count of outliers (1026)** revealed that a relatively small percentage of the predictions were extreme, but these outliers were important to investigate further. A significant finding from the outlier analysis was that most of the **outliers in prediction errors** came from the **high-priced cars**. A **count plot** of these outliers showed a clear concentration of errors in the higher price range, which could indicate that the model struggles with predicting prices for expensive cars

Random Forest

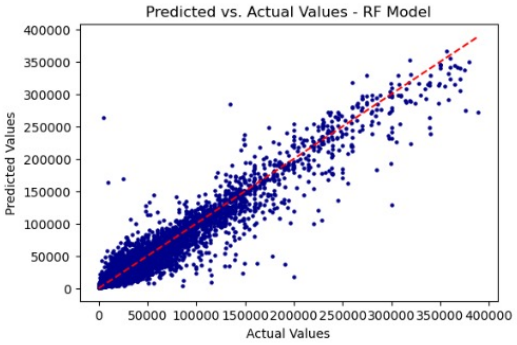
•Random Forest - Reducing Variance with Ensemble Averaging, an ensemble of decision trees, provided a balance between bias and variance by averaging multiple trees. This imoporves accuracy and robustness compared to a single decision tree.
•Random Forest model allowed to set up the higher threshold for the extreme values, without increasing the mean average errors

•**Metrics**: R2 = 0.9238
MAE = 2997.7
MSE = 44979125
Price range: up to 420 000 Euro

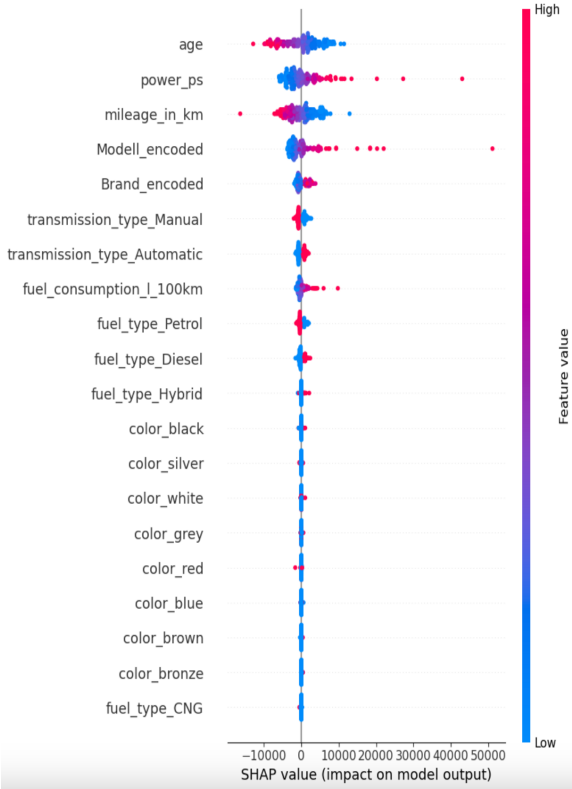
•Benefits: Improved generalization and robustness to overfitting compared to single trees.

	Actual Price	Predicted Price	Year	Power	Mileage	Modell_encoded	model	brand
0	14990.0	14464.158825	2017.0	116.0	117000.0	20922.626683	SEAT Leon	seat
1	29650.0	31789.467354	2019.0	190.0	69000.0	20016.239382	BMW 520	bmw
2	57000.0	55130.440738	2009.0	345.0	164000.0	130226.361345	Porsche 911	porsche
3	67500.0	74528.696494	2019.0	306.0	45000.0	69445.221987	Land Rover Range Rover Sport	land-rover
4	14800.0	12146.543024	2015.0	95.0	89000.0	15396.244337	MINI One	mini
...	...	...	...	...	...	...	...	...
47424	15800.0	18544.513775	2019.0	150.0	92000.0	15577.207905	Ford Focus	ford
47425	19490.0	20477.747166	2019.0	150.0	63000.0	18912.095890	Mazda CX-3	mazda
47426	27990.0	26159.460245	2019.0	190.0	99990.0	22101.846119	Volkswagen Tiguan	volkswagen
47427	17445.0	19350.128733	2022.0	83.0	100.0	13393.097126	Lada Niva	lada
47428	18950.0	18119.626581	2019.0	121.0	90000.0	18912.095890	Mazda CX-3	mazda

47429 rows × 8 columns

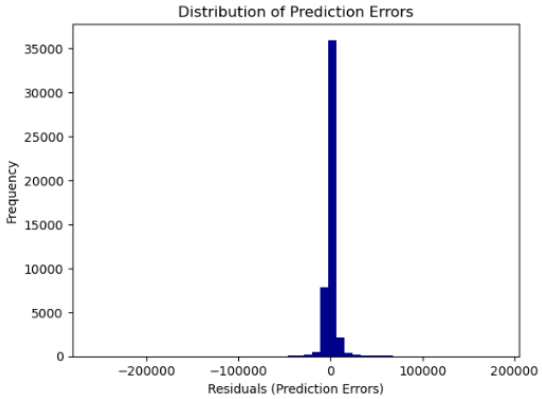


This plot show Random Forest performs better than the other models. The points are not so far away from the red line and distributed quite strictly along it. And the price range for prediction is impressive.

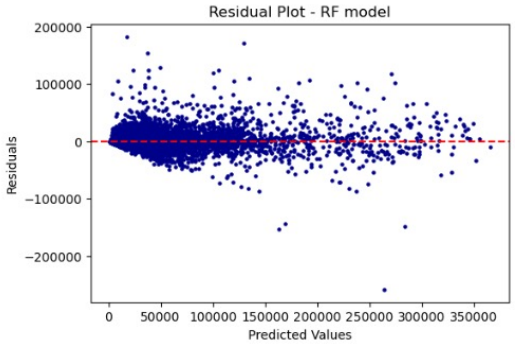


According SHAP values –
the most predictin features are ‘age’, ‘power_ps’, ‘mileage’, ‘model’. Brand, transmission types, fuel_types and fuel
consumption have less predicting values. From the colors the strongest is black, but its predictive values is too small.

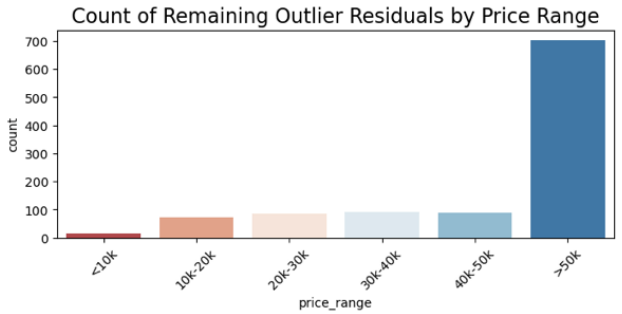
Random Forest: errors analysis



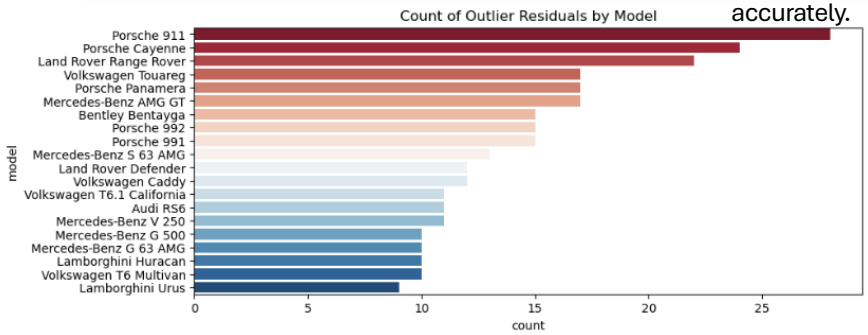
The **histplot** of residuals for **Random Forest** shows a **more symmetric and centered** distribution around zero, indicating a more balanced error across all data points.



The plot shows a **better fit** overall with fewer indications of bias in the residuals, especially compared to the Linear Regression model.While there is still some **increasing spread** in the residuals for higher predicted values, this might be inherent in the nature of predicting prices for high-end cars, where variability can be large and harder to predict accurately.



The **count of outliers (987)** revealed that a relatively small percentage of the predictions were extreme, but these outliers were important to investigate further. A significant finding from the outlier analysis was that most of the **outliers in prediction errors** came from the **high-priced cars**. A **count plot** of these outliers showed a clear concentration of errors in the higher price range, which could indicate that the model struggles with predicting prices for expensive cars



Upon further analysis, most extreme errors are concentrated among high-priced vehicles, particularly those priced over €50,000. Specific models with high error counts include luxury and performance brands like Porsche, Bentley, Mercedes AMG, and Land Rover. Interestingly, more standard models, such as the Volkswagen Caddi, also display some outliers. In the Notebook I performed additional analysis of the errors (similar to those made for XGBoost)

XGBoost – Gradient Boosted Trees

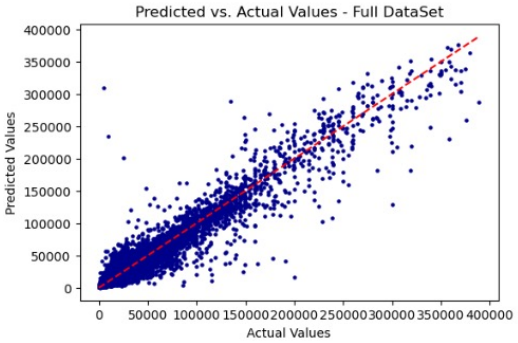
XGBoost - Optimized Model with Best Performance. **It** stands out as a powerful machine learning model for car price prediction due to its ability to handle complex, non-linear relationships in the data. XGBoost can capture more intricate patterns by building an ensemble of decision trees through boosting.

- XGBoost emerged as the top-performing model on the dataset, benefiting from gradient-boosting techniques.
- **Metrics**: R2 = 0.93
  MAE = 3007.37
  MSE = 47959433

Price range: up to 420 000 Euro

	Actual Price	Predicted Price	Residuals	Abs_residuals	Year	Power	Mileage	Modell_encoded	Age	Fuel	model	brand
0	8685.0	13012.257812	-4327.257812	4327.257812	2014.0	150.0	143623.0	21352.507317	10.0	4.5	Skoda Octavia	skoda
1	29990.0	26777.845703	3212.154297	3212.154297	2017.0	190.0	172661.0	23930.188571	7.0	5.8	Mercedes-Benz Vito	mercedes-benz
2	11900.0	14108.957031	-2208.957031	2208.957031	2009.0	306.0	275000.0	35995.277272	15.0	6.9	BMW 740	bmw
3	35980.0	34635.402344	1344.597656	1344.597656	2023.0	150.0	100.0	27979.466908	1.0	6.5	SEAT Ateca	seat
4	4990.0	4750.384766	239.615234	239.615234	2005.0	75.0	30000.0	3234.673082	19.0	6.1	Renault Modus	renault
...	...	...	...	...	...	...	...	...	...	...	...	...
47511	23300.0	21125.623047	2174.376953	2174.376953	2015.0	258.0	175100.0	23632.863071	9.0	6.0	BMW X3	bmw
47512	15980.0	16753.482422	-773.482422	773.482422	2015.0	110.0	54000.0	17472.635914	9.0	5.4	Volkswagen Golf Sportsvan	volkswagen
47513	21995.0	19905.697266	2089.302734	2089.302734	2014.0	184.0	63773.0	17731.760717	10.0	6.0	Mercedes-Benz C 200	mercedes-benz
47514	9990.0	11318.509766	-1328.509766	1328.509766	2016.0	86.0	19000.0	11795.147679	8.0	4.7	Kia Picanto	kia
47515	37900.0	17617.437500	20282.562500	20282.562500	2014.0	170.0	201000.0	24085.488166	10.0	4.5	Mercedes-Benz E 220	mercedes-benz

47516 rows × 12 columns



This plot demonstrates XGBoost very strong performance. The points are not so far away from the red line and distributed quite strictly along it. And the price range for prediction is impressive.

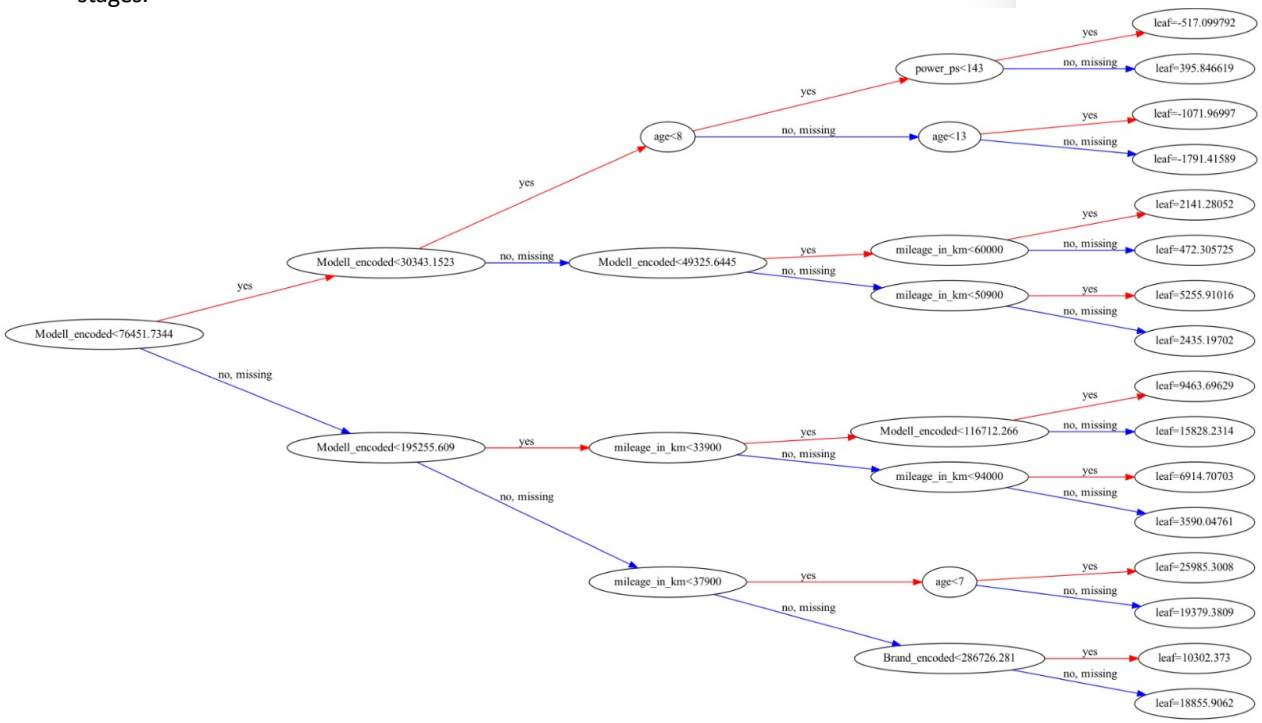
Strong Performance: XGBoost model's metrics are quite solid for such a diverse dataset. XGBoost and Random Forest are performing well here, as expected, due to their robustness with varied data and ability to capture non-linear relationships.

SHAP: Top Predictors: The most influential features in predicting car prices are Modell_encoded, age, power_ps, mileage_in_km, and Brand_encoded. This is expected, as these features directly relate to a car's model, age, power, mileage, and brand, which are all critical factors in determining a car's value.

Impact of Categorical Features: The SHAP plot shows that categorical features like transmission_type, fuel_type, and various color values are included, but they have a relatively smaller impact on the model's predictions. This aligns with the idea that while color might show significance in certain contexts (as ANOVA indicated), it does not provide substantial predictive power for price when looking at the overall model.

XGBoost – How does it work

Decision Tree visualization: The tree visualization offers an intuitive view of the decision rules used by the XGBoost model at different stages.



Insight: Observing these rules helps in understanding how the model segments the dataset based on key thresholds in features like mileage, power, and age. This provides a visual intuition about the model's decision-making process and how features interact to predict car prices.

Interpretation of XGBoost Decision Tree Plot

1.Root Node (Level 1):

1. **Condition**: Model Encoded < 76541
    1. This is the first feature that the model splits on. If the encoded value of the model is less than 76541, the decision-making process proceeds down one path.

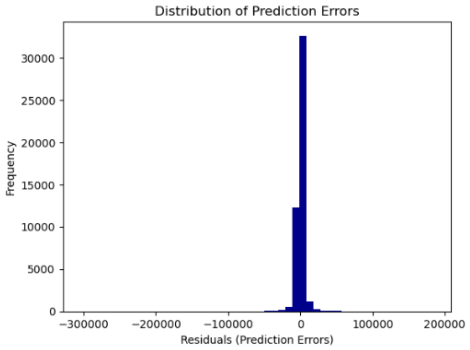
2.Second Layer (Level 2):

1. For the branch where Model Encoded < 76541, the model then considers two other conditions:
    1. **Condition 1**: Model Encoded < 30343
    2. **Condition 2**: Model Encoded < 195550

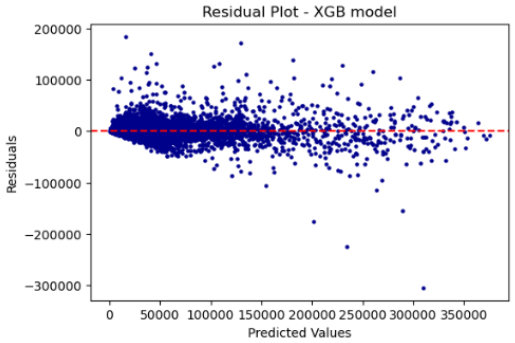
3.Third Layer (Level 3):

1. If Model Encoded < 30343 holds true, the tree further splits based on:
    1. **Condition 1**: Age < 8
    2. **Condition 2**: Model Encoded < 49325
    3. These nodes indicate that the model is checking the age and another encoded value to split the data into subgroups.
2. For the path where Mileage < 33900, the model also checks conditions such as:
    1. **Mileage < 47900**
    2. Additional conditions like these create increasingly granular subgroups.

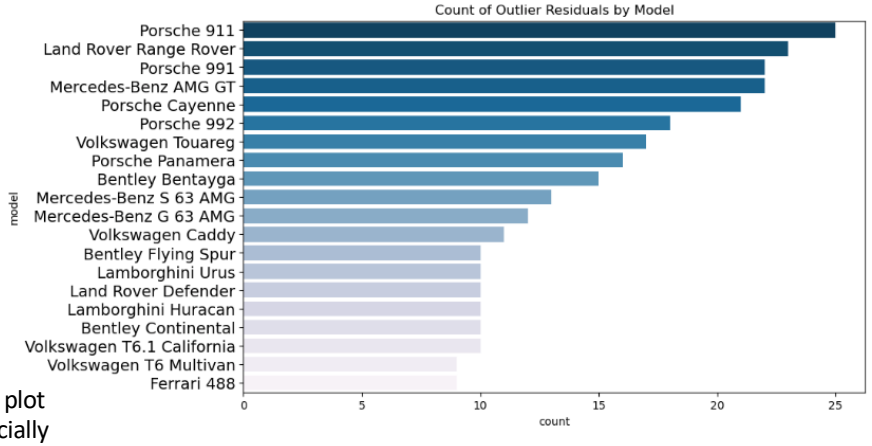
XGBoost: errors analysis



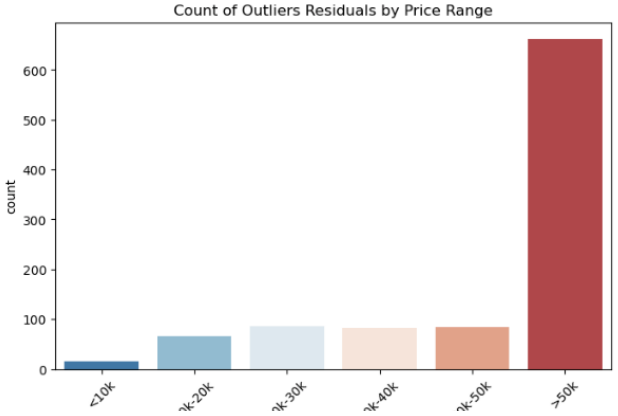
The **histplot** of residuals for **XGBoost** shows a **good symmetric and centered** distribution around zero, indicating a more balanced error across all data points.



XGBoost model is already fine-tuned, the residual plot still shows some slight spread in the residuals, especially at higher predicted values. However, the model is clearly performing better than the **Linear Regression/Decision Tree** model, as the residuals are more evenly distributed around zero and show less evidence of heteroscedasticity.



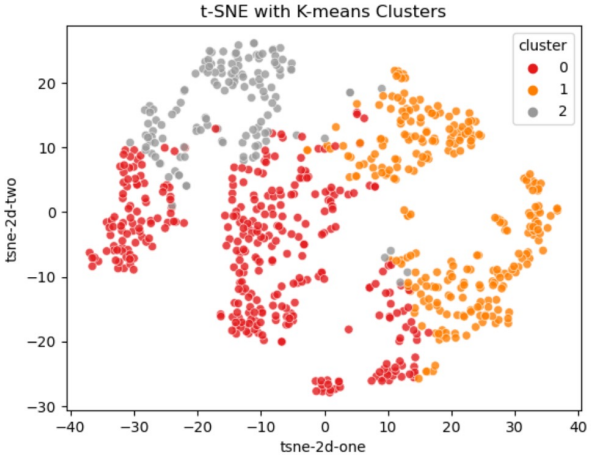
Like Random Forest Mode and other models, the errors show a normal distribution, as illustrated in the histogram. Upon further analysis, the majority of extreme errors are concentrated among high-priced vehicles, particularly those priced over €50,000. Specific models with high error counts include luxury and performance brands like Porsche, Bentley, Mercedes AMG, Ferrari and Land Rover. Interestingly, more standard models, such as the Volkswagen Caddy, also display some outliers. It is almost identical to that obtained by Random Forest.



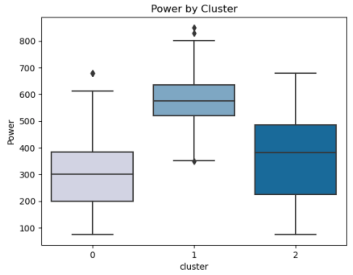
The **count of outliers (1999)** revealed that a relatively small percentage of the predictions were extreme, but these outliers were important to investigate further. A significant finding from the outlier analysis was that most of the **outliers in prediction errors** came from the **high-priced cars**. A **count plot** of these outliers showed a clear concentration of errors in the higher price range, which could indicate that the model struggles with predicting prices for expensive cars

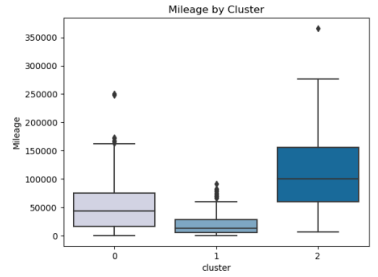
XGBoost: errors analysis – clustering of extreme errors

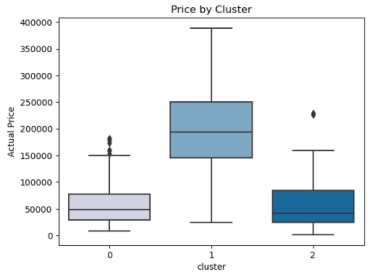
I used K-means clustering and t-SNE dimensionality reduction to analyze the patterns among the extreme prediction errors, opting for three clusters. To define the number of clusters I used elbow method.Here's a breakdown of the clusters and their characteristics:

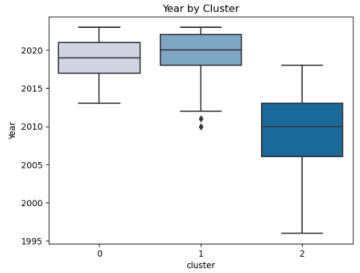


•**Cluster 0**: Contains 300 errors, primarily among cars with relatively low prices, low mileage, moderate power, and recent model years (around 2020).
•**Cluster 1**: Contains 200 errors, featuring high-priced cars with very low mileage, high power, and very recent model years.
•**Cluster 2**: Contains 120 errors, mostly for lower-priced cars with high mileage, moderate power, and older model years (around 2010).
Interestingly, the residuals appear similarly distributed across all three clusters. This suggests that while the clusters capture distinct car characteristics associated with prediction errors, the model's prediction discrepancies remain consistent across these categories.





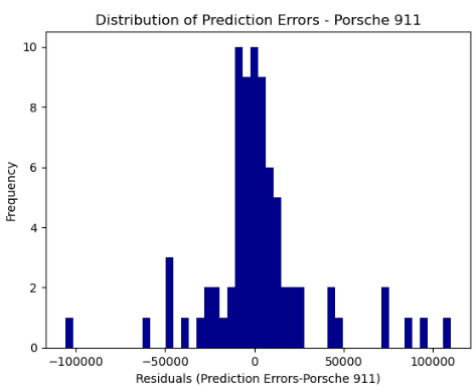


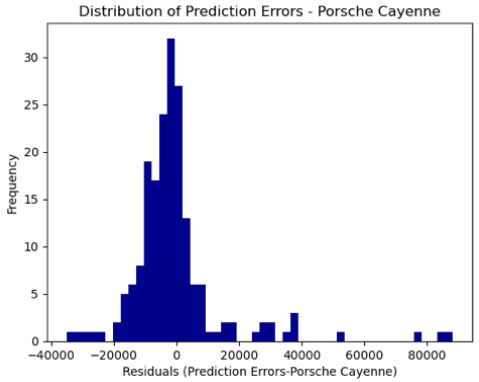


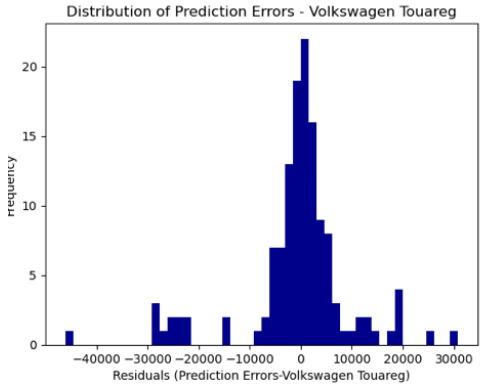
Model Performance on Price Ranges and Mileage Variability
•The presence of errors across all three clusters, despite their distinct characteristics, indicates that the model may be generalizing well overall but struggles to accurately predict prices in cases of:

- **High-value vehicles** (Cluster 1), where features like rarity or unique conditions can influence pricing.
- **Older, high-mileage vehicles** (Cluster 2), where additional factors like wear, potential damage, or custom modifications can significantly affect market price.
- **Newer, moderate vehicles** (Cluster 0), where price might be affected by competitive market pricing or incentives.

XGBoost: errors analysis







In examining the residual distributions for models such as the **Porsche 911**, **Porsche Cayenne**, and **Volkswagen Touareg**, we observe that the residuals—representing the differences between the predicted and actual car prices—tend to follow a **normal distribution** centered around zero. This indicates that, on average, the models are making unbiased predictions, with errors symmetrically distributed around the true values.

However, we also notice a significant number of **extreme values** or **outliers** on both sides of the distribution. These outliers represent cases where the model either significantly overestimates or underestimates the price. Such extreme values may arise due to unique characteristics or rare conditions in certain car listings that are not well-captured by the model, such as unusual mileage, unique configurations, or special editions of the cars.

Key Insights:

•**Centered, Symmetrical Residuals**: The normal, centered distribution suggests that overall, the models are well-calibrated, with no consistent bias towards over- or underprediction.

•**Presence of Outliers**: The outliers on both sides highlight that while the models capture typical price patterns well, there are instances with more substantial deviations, possibly due to specific, uncommon attributes of the vehicles.

Conclusion:

Model Performance Summary

1.Model Ranking:

1. **Performance Order**: Linear Regression < Decision Trees < K-Nearest Neighbors (K-NN) < Random Forest ≈ XGBoost
2. **Top Models**: Random Forest and XGBoost performed best overall, especially on high price ranges (up to €420,000), maintaining relatively low Mean Absolute Error (MAE).

2.Error Analysis:

1. **High-Priced Cars**: All models showed the most significant errors for high-priced cars (prices over €50,000).
2. **Outliers**:
    1. Approximately 25% of the prediction errors were linked to price outliers in the dataset.
    2. The remaining outliers were harder to explain and might stem from factors not in the dataset, such as unique features, marketing influences, usage history, or other characteristics specific to luxury cars.

3.Feature Importance:

1. **SHAP Analysis**: Across all models, the following features consistently had the highest predictive impact:
    1. **Age**: Older cars generally have lower prices.
    2. **Mileage**: Higher mileage tends to reduce the car's value.
    3. **Model and Brand**: Specific models and brands influence price significantly.
    4. **Power (PS)**: More powerful cars tend to have higher prices.
2. **Secondary Features**: Occasionally, certain categorical variables like fuel type and transmission type also showed relative importance, though they were less impactful than the core features above.

4.Color Feature Impact:

1. Removing color data for Random Forest and XGBoost models did not affect performance, suggesting that color does not significantly enhance predictive accuracy in this context.

Conclusion:

Additional Points to Consider:

•**Practical Implications**: The model's effectiveness at predicting high-priced cars' values, though with some error, shows potential for pricing guidance in the luxury segment.
•**Data Limitations**: Some factors affecting car prices (e.g., accident history, modifications, detailed condition) were not available in the dataset, which could account for unexplained errors. (There was some information in the 'offer_description', it could be possible to work with it additionally).

•**Future Improvements**:

- Incorporating additional data on car condition, maintenance history, or market trends could reduce unexplained errors.
- Exploring ensemble techniques combining multiple models might further enhance predictive accuracy, particularly for luxury vehicles.

•**2024 Dataset**: Including an updated dataset for 2024, where electric vehicles (EVs) are expected to have a stronger presence, would allow for deeper insights. This could reveal shifts in feature importance, such as **fuel type** or **power**, and assess how EV-specific attributes impact car pricing.

Additional Data Points: Adding information on car condition, accident history, and maintenance records could improve model accuracy, particularly in the high-end segment.

Exploring Hybrid Models: Combining different algorithms (e.g., ensemble methods) may further enhance accuracy, especially for luxury and electric vehicles where pricing patterns might differ from traditional cars

Summary of the project:

Summary of Car Price Analysis and Modeling

This study utilized a dataset of approximately 250,000 cars, which, after data cleaning, was reduced to about 238,000 entries. The analysis focused on understanding key factors influencing car prices in the secondary market. Key areas of analysis included:

•**Market Trends and Model Popularity**: I examined the most popular car models, the distribution of production years, and patterns in the presence of specific brands and models over time.

•**Correlation Analysis**: Relationships between price and both numerical and categorical features (such as brand, model, transmission, fuel type, and color) were analyzed to identify significant predictors. Specific examples, like the Mercedes-Benz C180, demonstrated how color impacts price, verified using ANOVA tests.

•**Feature Distribution Analysis**: Distributions of all features were evaluated, with particular attention to normality, linearity, skewness, kurtosis, and the presence of outliers. Most features displayed non-linear distributions, high skewness, high kurtosis, and substantial outliers, reflecting the diverse and often extreme values in the car market.

Modeling Approach

•**Data Transformation and Scaling**: Linear regression models required transformation to address skewness, while k-nearest neighbors (KNN) models used robust scaling. Each model was tested on the full dataset and compared against versions where extreme values were removed to varying degrees, depending on the model requirements.

•**Model Tuning**: Hyperparameters for each model were optimized using GridSearchCV, enabling refined performance.

•**Model Performance**: Random Forest and XGBoost models emerged as the top-performing models, capable of accurately predicting prices across a wide range (up to €450,000). These models achieved low Mean Absolute Error (MAE), Mean Squared Error (MSE), and high R^2 values, indicating strong predictive power.

•**Outlier Analysis**: Around 1,000 outliers were observed in the predictions, with one-third of these being true price outliers within the dataset. The largest errors occurred with high-priced vehicles (over €50,000), likely due to unique, unobserved characteristics such as special features, damage history, or marketing influence in luxury models, which are challenging to capture with conventional features.

Conclusion

This analysis successfully identified the most impactful factors for car prices in the secondary market and demonstrated the effectiveness of Random Forest and XGBoost in handling complex price variations. While these models performed well, further precision in predicting luxury car prices would require additional data on unique features and potential incidents impacting value.