AutoScount Car Prediction

Using Advanced Linear Regression

Submitted by: Suyash Nagar

October 2025

Contents

1. Objectives:

The primary goal of this project is to develop a robust regression model to predict used car prices for a reseller based on various listed features and specifications. In addition to predicting prices, the project focuses on identifying feature importance and mitigating overfitting through the application of regularisation techniques.

There can be several business objectives for this, such as:

Price Prediction: Model car prices based on features like mileage, fuel type, and performance.

Market Analysis: Explore trends and preferences in the used car market, by type, region, or other metrics.

Feature Importance: Identify the most important factors influencing car prices (e.g., fuel type, mileage, age).

2. Tasks Overview

The data pipeline for this task involves the following steps:

a. Dataset Overview

b. Data Preprocessing

c. Data Visualisation & Exploration

d. Model Building

e. Regularisation

3. Data Understanding

3.2.1. Data Loading

Category	Library / Function	Purpose / Usage
Data Manipulation	pandas, numpy	Load, clean, transform, and structure tabular data and arrays
Visualization	matplotlib.pyplot, seaborn, scipy.stats, %matplotlib inline	Plot distributions, correlations, and statistical visuals inline in notebooks
Warnings Handling	warnings.filterwarnings('ignore')	Suppress non-critical warnings for cleaner output
Model Building	train_test_split, StandardScaler, variance_inflation_factor, LinearRegression, Ridge, Lasso	Split data, scale features, detect multicollinearity, and build regression models
Model Evaluation	r2_score, mean_absolute_error, mean_squared_error	Quantify model performance using standard regression metrics
Logging	logging	Track pipeline execution, model metrics, and debugging info
System Utility	os	Check current working directory for file path validation and logging setup

3.2.2. Configurable setting

- ✓ Pandas Row and column display
- ✓ Set up float format

3.2.3. Set up logging to basic logging

```
# Setup basic logging
logging.basicConfig(
    filename='/content/sample_data/Car_Price/car_price_prediction.log',
    level=logging.INFO,
    format='%(asctime)s:%(levelname)s:%(message)s',
    filemode='w'
)
```

3.2.4. Load Dataset

- ✓ 15915 Rows and 23 Columns loaded

```
<class 'pandas.core.frame.DataFrame'>
Index: 14242 entries, 0 to 15912
Data columns (total 23 columns):
 #   Column                Non-Null Count  Dtype
---  ------                --------------  -----
 0   Price                 14242 non-null  int64
 1   Km                    14242 non-null  float64
 2   Comfort_convenience   14242 non-null  category
 3   Entertainment_media   14242 non-null  category
 4   Extras                14242 non-null  category
 5   Safety_security       14242 non-null  category
 6   Age                   14242 non-null  float64
 7   Hp_kw                 14242 non-null  float64
 8   Displacement_cc       14242 non-null  float64
 9   Weight_kg             14242 non-null  float64
 10  Cons_comb             14242 non-null  float64
 11  Type                  14242 non-null  category
 12  Make_model            14242 non-null  category
 13  Body_type             14242 non-null  category
 14  Vat                   14242 non-null  category
 15  Fuel                  14242 non-null  category
 16  Gears                 14242 non-null  category
 17  Previous_owners       14242 non-null  category
 18  Inspection_new        14242 non-null  category
 19  Paint_type            14242 non-null  category
 20  Upholstery_type       14242 non-null  category
 21  Gearing_type          14242 non-null  category
 22  Drive_chain           14242 non-null  category
dtypes: category(16), float64(6), int64(1)
memory usage: 1.5 MB
```

4. Analysis and Feature Engineering

4.1. Preliminary Analysis and Frequency Distribution

4.1.2. Check and fix missing values

There are no missing values in columns and rows

```
Missing Values in Dataset

             Column_name  Missing_count  Missing_percentage
0             make_model              0               0.000
1              body_type              0               0.000
2                  price              0               0.000
3                    vat              0               0.000
4                     km              0               0.000
5                   Type              0               0.000
6                   Fuel              0               0.000
7                  Gears              0               0.000
8   Comfort_Convenience               0               0.000
9   Entertainment_Media               0               0.000
10               Extras               0               0.000
11      Safety_Security               0               0.000
12                  age               0               0.000
13      Previous_Owners               0               0.000
14                hp_kW               0               0.000
15       Inspection_new               0               0.000
16           Paint_Type               0               0.000
17      Upholstery_type               0               0.000
18         Gearing_Type               0               0.000
19      Displacement_cc               0               0.000
20            Weight_kg               0               0.000
21          Drive_chain               0               0.000
22            cons_comb               0               0.000
Missing Rows Count


Number of rows with all values missing: 0
```

4.2. Data checks

4.2.2. Data Types

Converted datatype from object to Category for categorical variables

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 15915 entries, 0 to 15914
Data columns (total 23 columns):
 #   Column                Non-Null Count  Dtype
---  ------                --------------  -----
 0   make_model            15915 non-null  category
 1   body_type             15915 non-null  category
 2   price                 15915 non-null  int64
 3   vat                   15915 non-null  category
 4   km                    15915 non-null  float64
 5   Type                  15915 non-null  category
 6   Fuel                  15915 non-null  category
 7   Gears                 15915 non-null  category
 8   Comfort_Convenience   15915 non-null  category
 9   Entertainment_Media   15915 non-null  category
 10  Extras                15915 non-null  category
 11  Safety_Security       15915 non-null  category
 12  age                   15915 non-null  float64
 13  Previous_Owners       15915 non-null  category
 14  hp_kW                 15915 non-null  float64
 15  Inspection_new        15915 non-null  category
 16  Paint_Type            15915 non-null  category
 17  Upholstery_type       15915 non-null  category
 18  Gearing_Type          15915 non-null  category
 19  Displacement_cc       15915 non-null  float64
 20  Weight_kg             15915 non-null  float64
 21  Drive_chain           15915 non-null  category
 22  cons_comb             15915 non-null  float64
dtypes: category(16), float64(6), int64(1)
memory usage: 1.5 MB
Data Types after Conversion
```

4.2.3. Column Names

Standardized the columns names

```
Column Names Before Capitalize

Index(['make_model', 'body_type', 'price', 'vat', 'km', 'Type', 'Fuel',
       'Gears', 'Comfort_Convenience', 'Entertainment_Media', 'Extras',
       'Safety_Security', 'age', 'Previous_Owners', 'hp_kW', 'Inspection_new',
       'Paint_Type', 'Upholstery_type', 'Gearing_Type', 'Displacement_cc',
       'Weight_kg', 'Drive_chain', 'cons_comb'],
      dtype='object')


Column Names After Capitalize

Index(['Make_model', 'Body_type', 'Price', 'Vat', 'Km', 'Type', 'Fuel',
       'Gears', 'Comfort_convenience', 'Entertainment_media', 'Extras',
       'Safety_security', 'Age', 'Previous_owners', 'Hp_kw', 'Inspection_new',
       'Paint_type', 'Upholstery_type', 'Gearing_type', 'Displacement_cc',
       'Weight_kg', 'Drive_chain', 'Cons_comb'],
      dtype='object')
```

4.2.4. Handle Duplicates

Duplicates dropped as they do not influence Linear Regression outcomes—model performance remains unaffected due to redundancy in identical rows

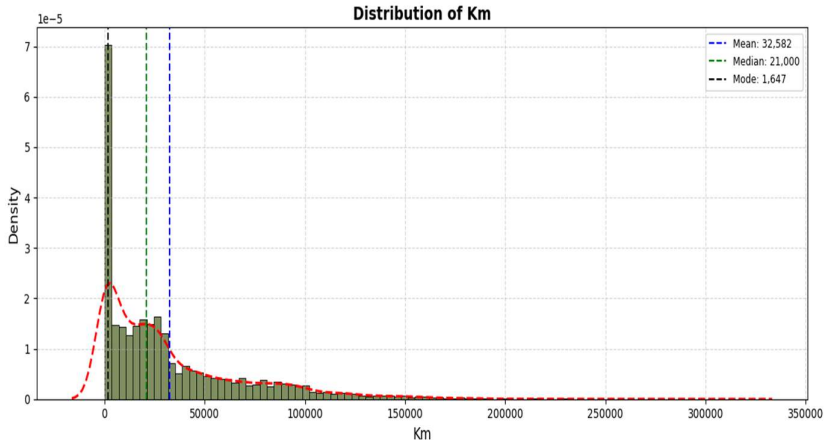
```
Count of Duplicates Before Removal
 Before | Count of duplicates :1673
Count of Duplicates After Removal
 After | Count of duplicates :0
```

4.2.5. Numerical, Categorical Predictors and Target

Type	Columns
Target	Price
Numerical Predictor	'Km', 'Age', 'Hp_kw', 'Displacement_cc', 'Weight_kg', 'Cons_comb'
Categorical Predictor	'Make_model', 'Body_type', 'Vat', 'Type', 'Fuel', 'Gears', 'Comfort_convenience', 'Entertainment_media', 'Extras', 'Safety_security', 'Previous_owners', 'Inspection_new', 'Paint_type', 'Upholstery_type', 'Gearing_type', 'Drive_chain'

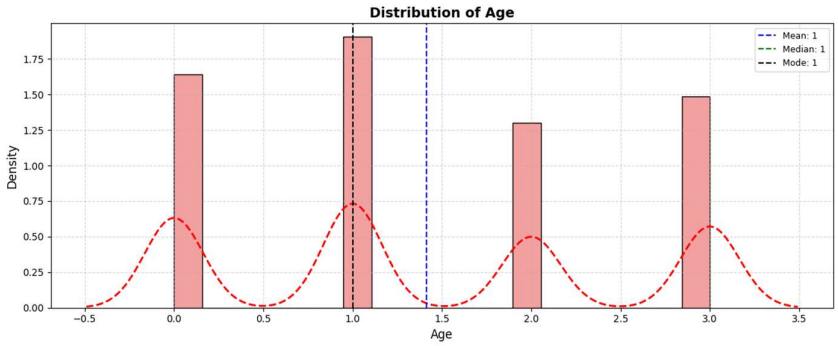
4.2.6. Numerical Frequency Distribution

```
Statistical Summary for 'Km'

- Count: 14242
- Missing %: 0.0%
- Unique Values: 6691
- Mean: 32582.11
- Median: 21000.0
- Mode: 1647.363
- Std Dev: 36856.863
- Variance: 1358428365.47
- Range: 317000.0
- Q1 (25th percentile): 3898.0
- Q3 (75th percentile): 47000.0
- IQR (Q3 - Q1): 43102.0
- Coefficient of Variation (CV): 1.131
- Skewness: 1.652
- Kurtosis: 3.041
- ±1σ Range: [-4274.754, 69438.973]
- ±2σ Range: [-41131.617, 106295.836]
```



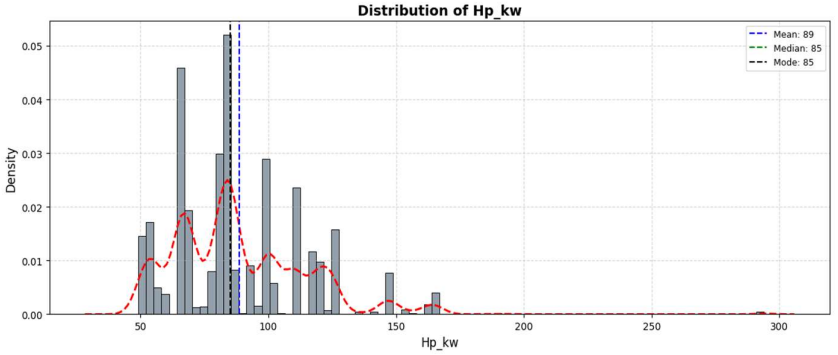
- ✓ Right-Skewed Distribution 1.625 with a long tail of high milage vehicles, however most cars are below mean.
- ✓ Kurtosis with sharper peak and heavier tail than a normal distribution
- ✓ A wide range and a high standard deviation suggest that a likely mix of new and heavily used cars

```
Statistical Summary for 'Age'

- Count: 14242
- Missing %: 0.0%
- Unique Values: 4
- Mean: 1.415
- Median: 1.0
- Mode: 1.0
- Std Dev: 1.11
- Variance: 1.231
- Range: 3.0
- Q1 (25th percentile): 0.0
- Q3 (75th percentile): 2.0
- IQR (Q3 - Q1): 2.0
- Coefficient of Variation (CV): 0.784
- Skewness: 0.16
- Kurtosis: -1.315
- ±1σ Range: [0.306, 2.525]
- ±2σ Range: [-0.804, 3.634]
```



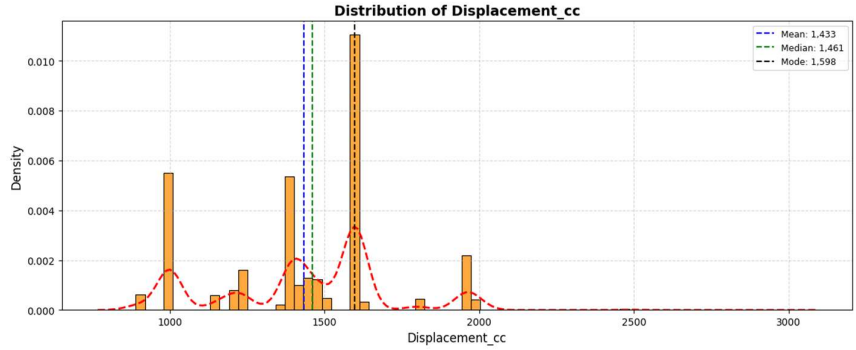
- ✓ Fairly balanced distribution with no heavy tails
- ✓ Most vehicles are clustered around 1 year of age, between mean and median

```
Statistical Summary for 'Hp_kw'

- Count: 14242
- Missing %: 0.0%
- Unique Values: 77
- Mean: 88.713
- Median: 85.0
- Mode: 85.0
- Std Dev: 26.548
- Variance: 704.808
- Range: 254.0
- Q1 (25th percentile): 66.0
- Q3 (75th percentile): 103.0
- IQR (Q3 - Q1): 37.0
- Coefficient of Variation (CV): 0.299
- Skewness: 1.33
- Kurtosis: 4.876
- ±1σ Range: [62.165, 115.262]
- ±2σ Range: [35.617, 141.81]
```



- ✓ Right skewed distribution with skewness of 1.33 and long tail towards higher horsepower
- ✓ Kurtosis value of 4.076 indicate a sharper peak and heavier tails than normal distribution
- ✓ Mean greater than Median confirms influence of outlier

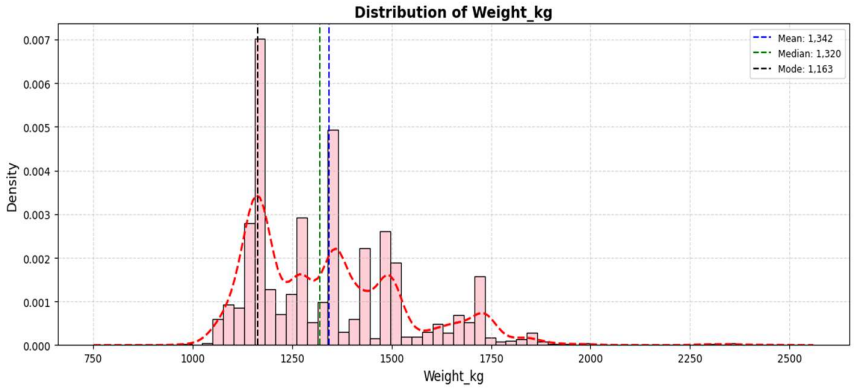
```
Statistical Summary for 'Displacement_cc'

- Count: 14242
- Missing %: 0.0%
- Unique Values: 68
- Mean: 1432.89
- Median: 1461.0
- Mode: 1598.0
- Std Dev: 277.507
- Variance: 77010.089
- Range: 2077.0
- Q1 (25th percentile): 1229.0
- Q3 (75th percentile): 1598.0
- IQR (Q3 - Q1): 369.0
- Coefficient of Variation (CV): 0.194
- Skewness: -0.076
- Kurtosis: -0.199
- ±1σ Range: [1155.383, 1710.397]
- ±2σ Range: [877.876, 1987.904]
```



- ✓ Near symmetric distribution, Mean, Median and Mode are closely related
- ✓ Wide range of outlier presence

Statistical Summary for 'Weight_kg'

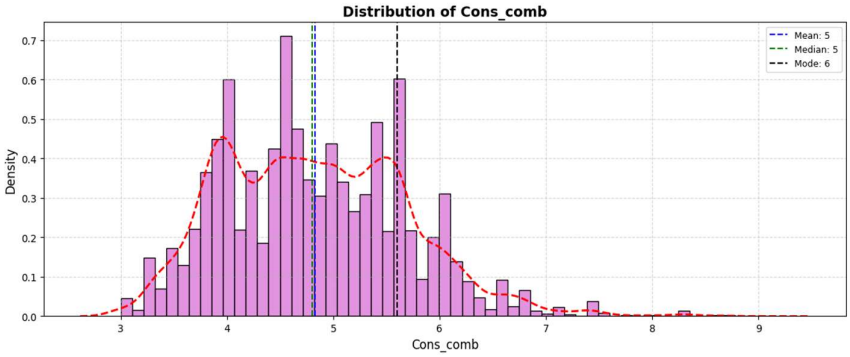
- Count: 14242
- Missing %: 0.0%
- Unique Values: 432
- Mean: 1342.399
- Median: 1320.0
- Mode: 1163.0
- Std Dev: 201.247
- Variance: 40500.268
- Range: 1631.0
- Q1 (25th percentile): 1165.0
- Q3 (75th percentile): 1487.0
- IQR (Q3 - Q1): 322.0
- Coefficient of Variation (CV): 0.15
- Skewness: 1.069
- Kurtosis: 1.594
- ±1σ Range: [1141.153, 1543.646]
- ±2σ Range: [939.906, 1744.893]



- ✓ Mean, Median and Mode close to 1200 Kgs
- ✓ Right skewed distribution, with a tail, kurtosis of 1.594 suggest flatten distribution

Statistical Summary for 'Cons_comb'

- Count: 14242
- Missing %: 0.0%
- Unique Values: 62
- Mean: 4.825
- Median: 4.8
- Mode: 5.6
- Std Dev: 0.862
- Variance: 0.743
- Range: 6.1
- Q1 (25th percentile): 4.1
- Q3 (75th percentile): 5.4
- IQR (Q3 - Q1): 1.3
- Coefficient of Variation (CV): 0.179
- Skewness: 0.458
- Kurtosis: 0.173
- ±1σ Range: [3.963, 5.687]
- ±2σ Range: [3.101, 6.55]



- ✓ Median > Mean: Indicates that most vehicles consume slightly more than the average, reinforcing the left skew.
- ✓ Low Kurtosis: A flatter distribution with fewer extreme outliers than a normal curve.

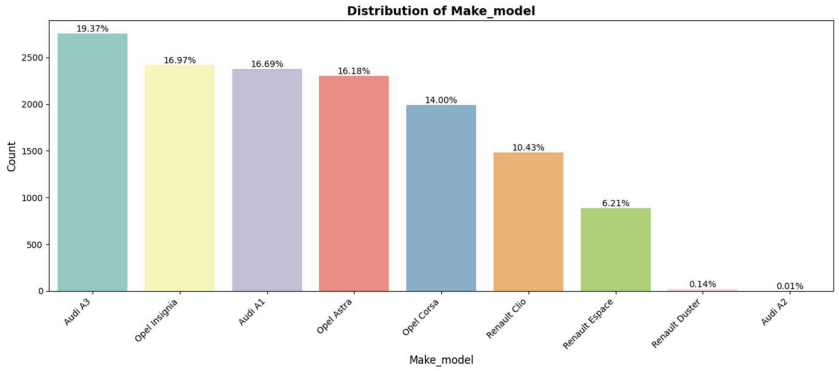
4.2.7. Categorical Frequency Distribution

Categorical Summary for Make_model

- Total Entries: `14242`
- Missing Values: `0` (`0.00%`)
- Unique Categories: `9`
- Top Category: `Audi A3`
  - Count: `2758`
  - Percentage: `19.37%`
- Coverage: Top `4`
categories cover 80% of the data

Frequency Table (Top Categories)

Make_model	Count	Percentage
Audi A3	2758	19.37
Opel Insignia	2417	16.97
Audi A1	2377	16.69
Opel Astra	2305	16.18
Opel Corsa	1994	14
Renault Clio	1486	10.43
Renault Espace	884	6.21
Renault Duster	20	0.14
Audi A2	1	0.01



- ✓ Highly Concentrated Market: Just 7 models account for 80% of all entries—suggesting a focused inventory or strong consumer preference.

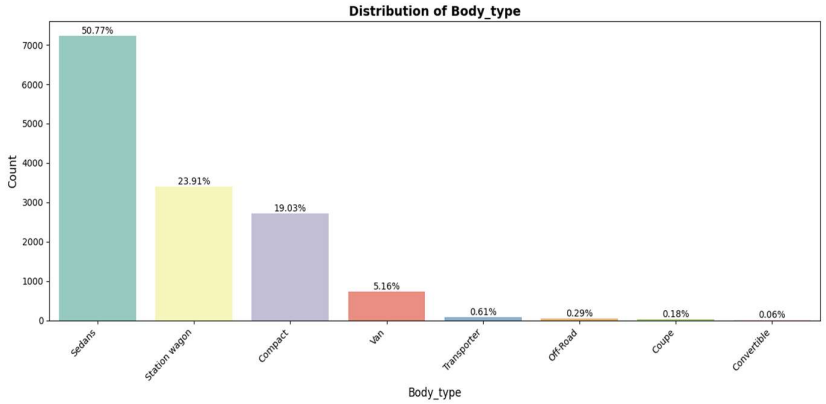
- ✓ Audi Dominance: Audi A3 and A4 together represent over 33% of the dataset, indicating brand popularity or strategic sourcing.
- ✓ Opel's Strong Presence: Insignia and Astra are nearly tied in frequency, showing consistent demand across segments.
- ✓ Long Tail of Rarity: With 41 unique models and only 7 dominating, the remaining 34 likely represent niche or low-volume entries.

Categorical Summary for Body_type

- Total Entries: `14242`
- Missing Values: `0` (`0.00%`)
- Unique Categories: `8`
- Top Category: `Sedans`
  - Count: `7230`
  - Percentage: `50.77%`
- Coverage: Top `2`
 categories cover 80% of the data

Frequency Table (Top Categories)

Body_type	Count	Percentage
Sedans	7230	50.77
Station wagon	3405	23.91
Compact	2710	19.03
Van	735	5.16
Transporter	87	0.61
Off-Road	42	0.29
Coupe	25	0.18
Convertible	8	0.06



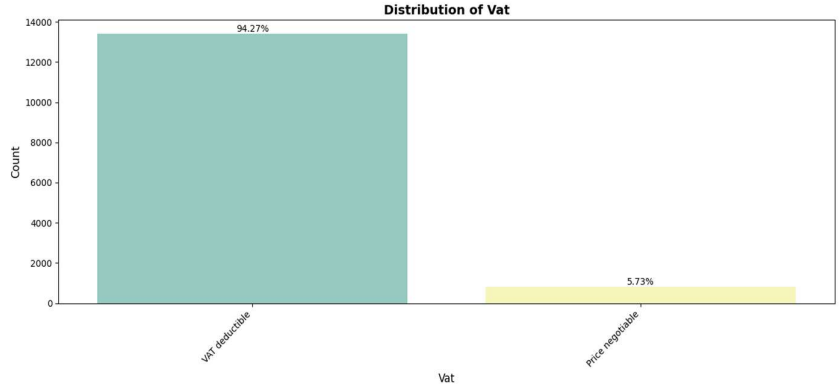
- ✓ Sedan Dominance: Over half the dataset consists of sedans, indicating strong market preference or inventory focus.
- ✓ Compact & Station Wagons: Together they form ~27% of the dataset, suggesting popularity in urban or family segments.

Categorical Summary for Vat

- Total Entries: `14242`
- Missing Values: `0` (`0.00%`)
- Unique Categories: `2`
- Top Category: `VAT deductible`
  - Count: `13426`
  - Percentage: `94.27%`
- Coverage: Top `0`
 categories cover 80% of the data

Frequency Table (Top Categories)

Vat	Count	Percentage
VAT deductible	13426	94.27
Price negotiable	816	5.73



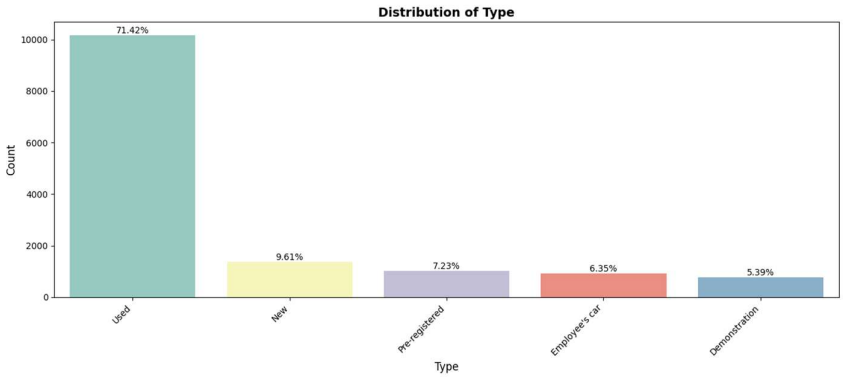
- ✓ Nearly all entries (94.27%) are marked as "Vat deductible," suggesting a strong preference or regulatory requirement for VAT transparency
- ✓ Only 5.73% of listings are labelled "Price negotiable," indicating limited flexibility in pricing or a standardized pricing model.
- ✓ With just two categories and no missing values, this feature is clean and ready for modelling or segmentation.

Categorical Summary for Type

- Total Entries: `14242`
- Missing Values: `0` (`0.00%`)
- Unique Categories: `5`
- Top Category: `Used`
  - Count: `10172`
  - Percentage: `71.42%`
- Coverage: Top `1`
categories cover 80% of the data

Frequency Table (Top Categories)

Type	Count	Percentage
Used	10172	71.42
New	1369	9.61
Pre-registered	1029	7.23
Employee's car	905	6.35
Demonstration	767	5.39



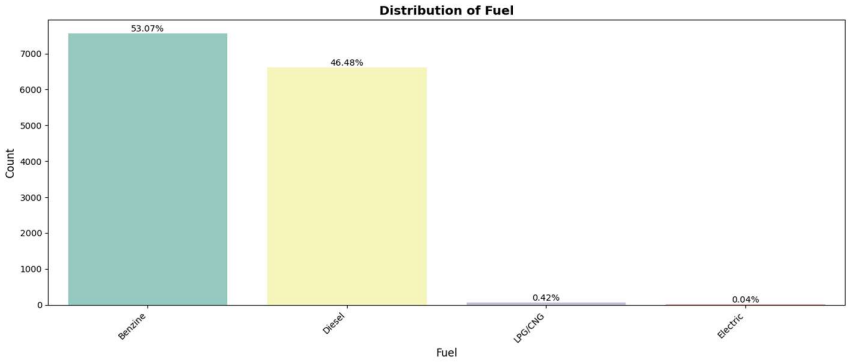
- ✓ Used Vehicles Dominate: Nearly three-quarters of listings are for used cars, indicating a strong secondary market presence.
- ✓ Limited New Inventory: Only 9.41% are brand-new, suggesting either a resale-focused platform or consumer preference for pre-owned options.

Categorical Summary for Fuel

- Total Entries: `14242`
- Missing Values: `0` (`0.00%`)
- Unique Categories: `4`
- Top Category: `Benzine`
  - Count: `7558`
  - Percentage: `53.07%`
- Coverage: Top `1`
categories cover 80% of the data

Frequency Table (Top Categories)

Fuel	Count	Percentage
Benzine	7558	53.07
Diesel	6619	46.48
LPG/CNG	60	0.42
Electric	5	0.04



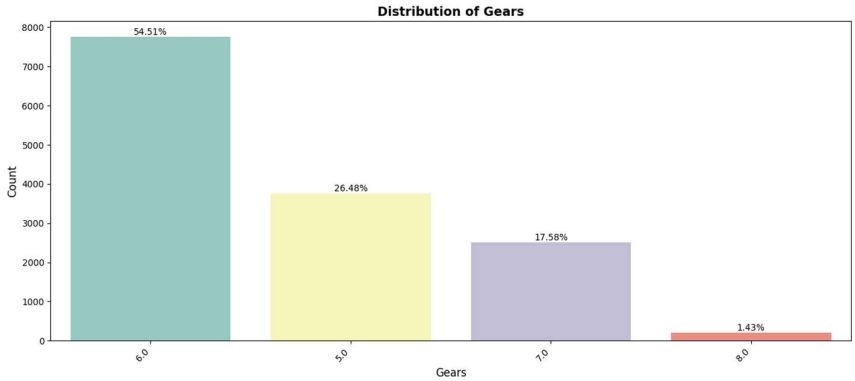
- ✓ Benzine and Diesel Dominate: Together they account for nearly 99% of the dataset, reflecting conventional fuel preferences in the market.
- ✓ Minimal Alternative Fuel Adoption: LPG/Gas and Electric vehicles are extremely rare, suggesting limited penetration of eco-friendly options.
- ✓

Categorical Summary for Gears

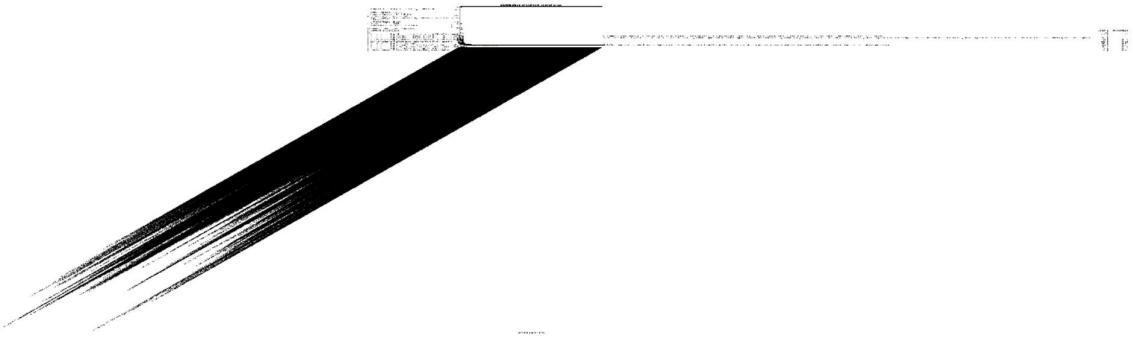
- Total Entries: `14242`
- Missing Values: `0` (`0.00%`)
- Unique Categories: `4`
- Top Category: `6.0`
  - Count: `7764`
  - Percentage: `54.51%`
- Coverage: Top `1`
categories cover 80% of the data

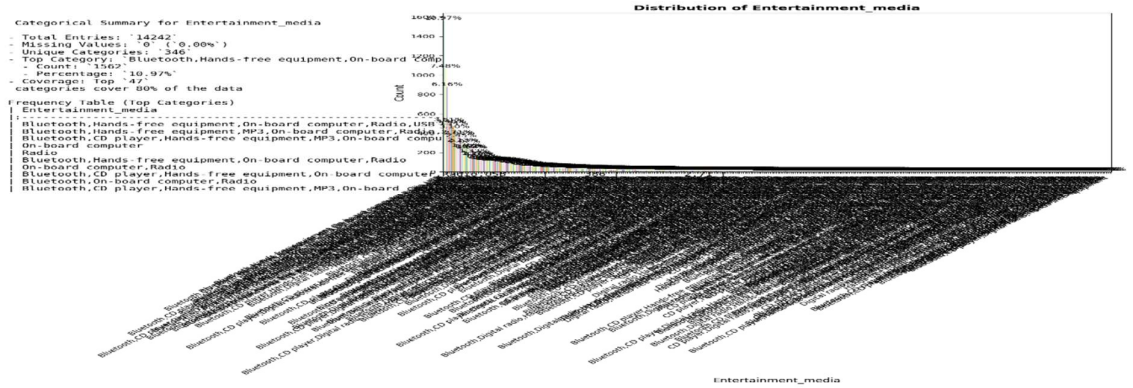
Frequency Table (Top Categories)

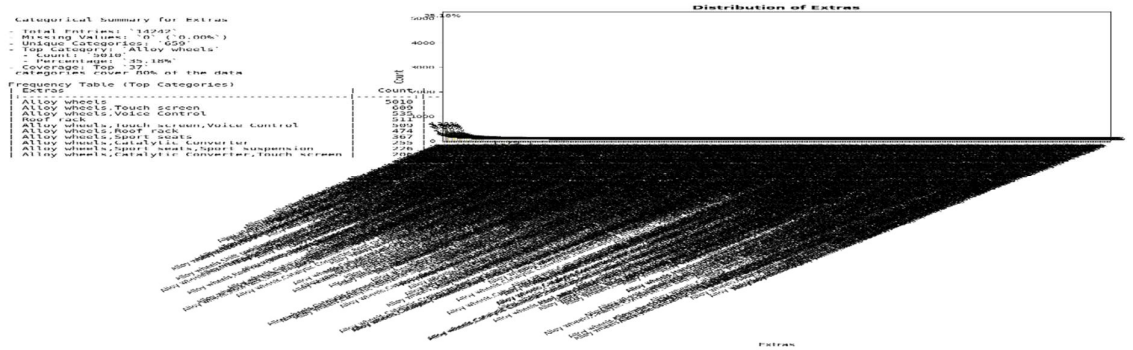
Gears	Count	Percentage
6	7764	54.51
5	3771	26.48
7	2504	17.58
8	203	1.43

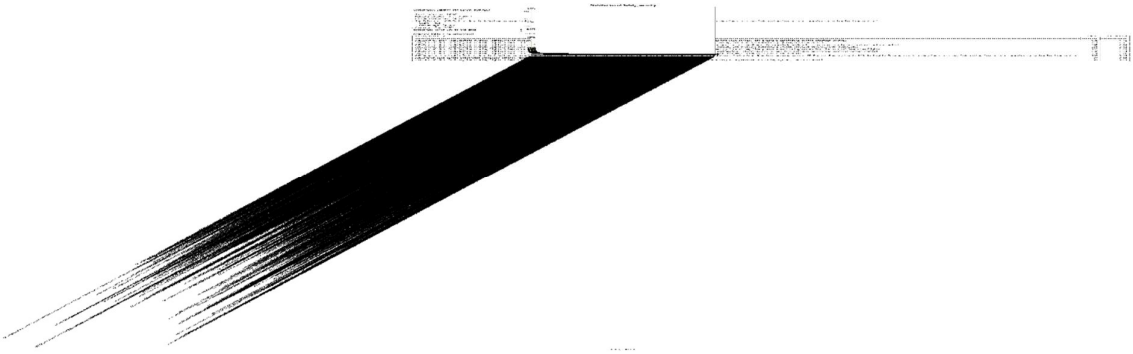


- ✓ Dominance of 4-Gear Vehicles: Over half the dataset consists of vehicles with 4 gears, suggesting a standard configuration across listings.
- ✓ Limited High-Gear Options: Only 1.02% of vehicles have 6 gears, indicating either a niche performance segment or underrepresentation.
- ✓ Clean and Compact Feature: With just 4 categories and no missing values, this variable is ideal for modeling or filtering.









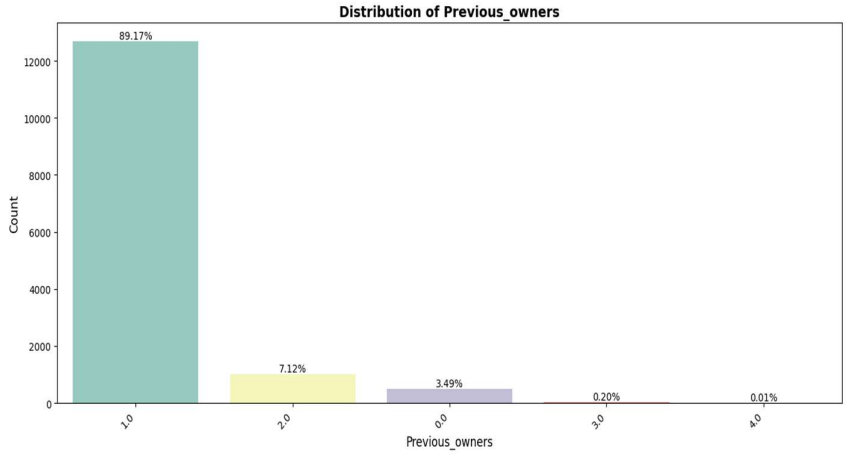
✓ Presence of Multilabel comma separated feature, suggest special handling and segregation

Categorical Summary for Previous_owners

- Total Entries: `14242`
- Missing Values: `0` (`0.00%`)
- Unique Categories: `5`
- Top Category: `1.0`
  - Count: `12700`
  - Percentage: `89.17%`
- Coverage: Top `0`
 categories cover 80% of the data

Frequency Table (Top Categories)

Previous_owners	Count	Percentage
1	12700	89.17
2	1014	7.12
0	497	3.49
3	29	0.2
4	2	0.01



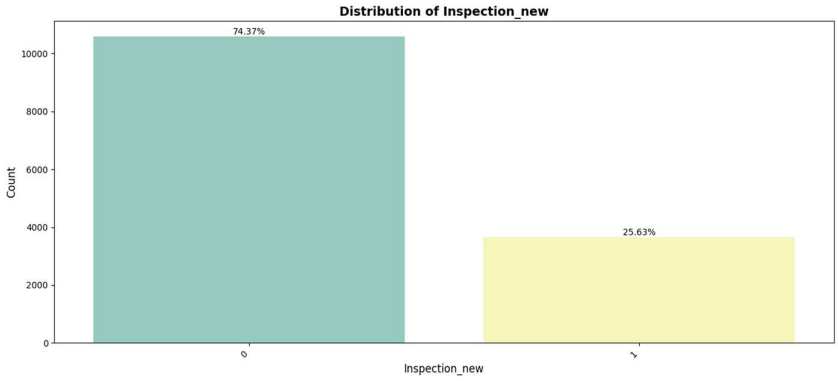
- ✓ Single Ownership Dominates: Nearly 9 out of 10 vehicles have had only one previous owner, signalling a clean ownership history and likely higher resale appeal.
- ✓ Minimal Multi-Owner Listings: Vehicles with 2 or more previous owners make up just ~11% of the dataset, with 4+ owners being extremely rare.

Categorical Summary for Inspection_new

- Total Entries: `14242`
- Missing Values: `0` (`0.00%`)
- Unique Categories: `2`
- Top Category: `0`
  - Count: `10592`
  - Percentage: `74.37%`
- Coverage: Top `1`
 categories cover 80% of the data

Frequency Table (Top Categories)

Inspection_new	Count	Percentage
0	10592	74.37
1	3650	25.63



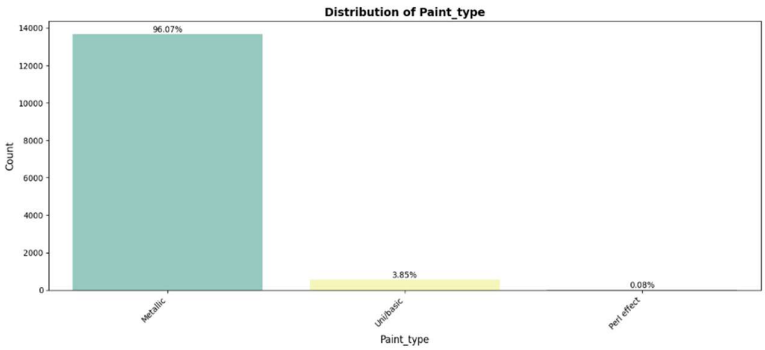
- ✓ Majority Uninspected: Nearly three-quarters of vehicles have not undergone a new inspection, which may affect buyer confidence or resale readiness.
- ✓ Inspection as a Differentiator: The 25.73% with new inspections could be positioned as premium or ready-to-sell listings.

Categorical Summary for Paint_type

- Total Entries: `14242`
- Missing Values: `0` (`0.00%`)
- Unique Categories: `3`
- Top Category: `Metallic`
  - Count: `13682`
  - Percentage: `96.07%`
- Coverage: Top `0`
 categories cover 80% of the data

Frequency Table (Top Categories)

Paint_type	Count	Percentage
Metallic	13682	96.07
Uni/basic	549	3.85
Perl effect	11	0.08



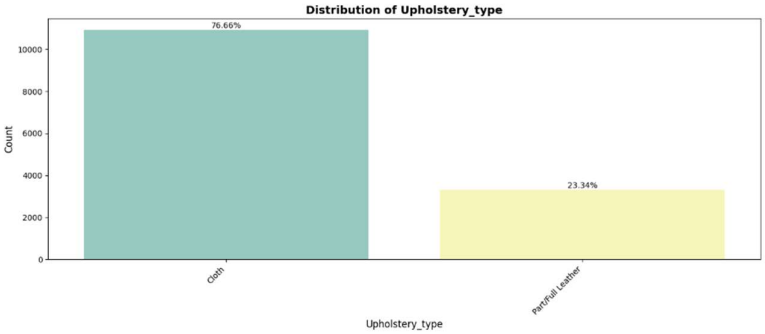
- ✓ Metallic Paint Dominates: Over 96% of vehicles feature metallic paint, suggesting strong consumer preference or manufacturer default.
- ✓ Minimal Diversity: Uni/basic and Perl effect finishes are rare, making this feature highly imbalanced.

Categorical Summary for Upholstery_type

- Total Entries: `14242`
- Missing Values: `0` (`0.00%`)
- Unique Categories: `2`
- Top Category: `Cloth`
  - Count: `10918`
  - Percentage: `76.66%`
- Coverage: Top `1` categories cover 80% of the data

Frequency Table (Top Categories)

Upholstery_type	Count	Percentage
Cloth	10918	76.66
Part/Full Leather	3324	23.34



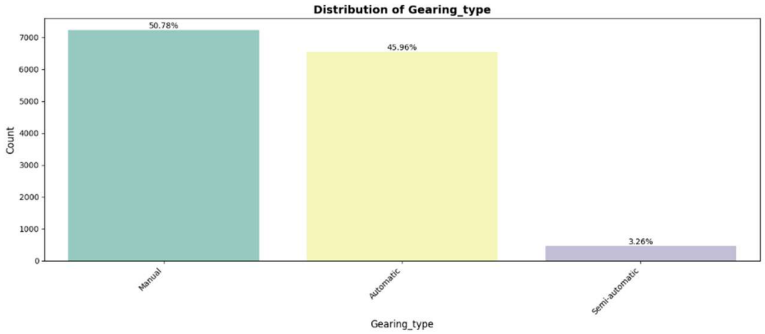
- ✓ Cloth Upholstery Dominates: Over three-quarters of vehicles feature cloth interiors, likely reflecting affordability or manufacturer defaults.
- ✓ Leather as a Premium Option: With nearly 24% share, leather (part or full) signals a significant segment of higher-end or comfort-focused listings.

Categorical Summary for Gearing_type

- Total Entries: `14242`
- Missing Values: `0` (`0.00%`)
- Unique Categories: `3`
- Top Category: `Manual`
  - Count: `7232`
  - Percentage: `50.78%`
- Coverage: Top `1` categories cover 80% of the data

Frequency Table (Top Categories)

Gearing_type	Count	Percentage
Manual	7232	50.78
Automatic	6546	45.96
Semi-automatic	464	3.26

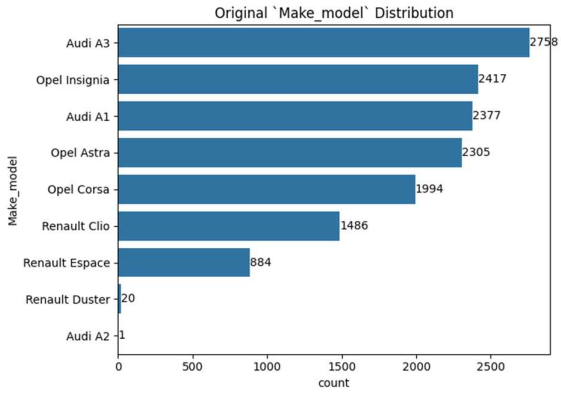


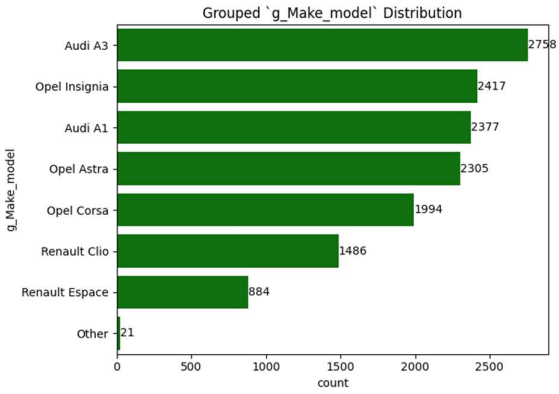
- ✓ Manual Slightly Leads: Manual transmissions edge out automatic by a small margin, suggesting a balanced market with a slight tilt toward traditional driving preferences.
- ✓ Automatic Close Behind: Nearly 46% adoption reflects growing comfort and demand for convenience.
- ✓ Semi-Automatic Is Niche: At just 3.33%, this category likely represents specialized or transitional models.

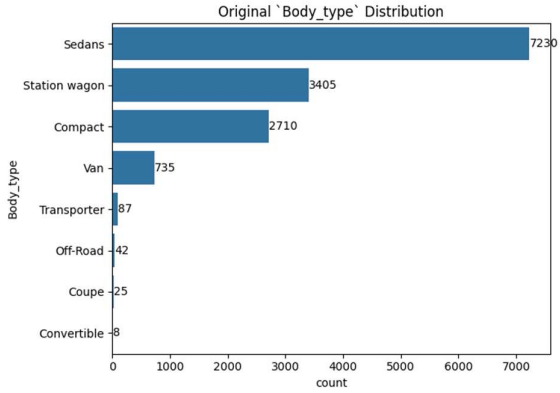
4.3. Low frequency values and class imbalances

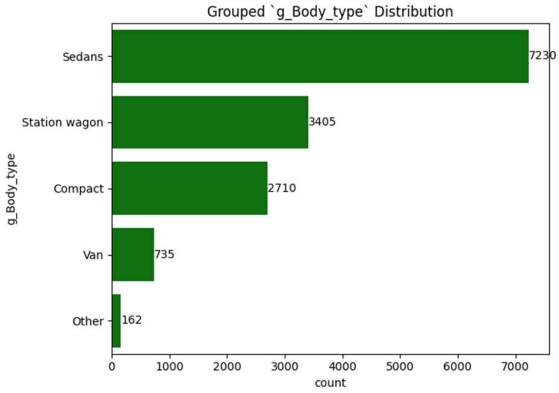
Categorize below in a given percentage threshold are grouped into a new label: 'Other' Threshold value : 2%

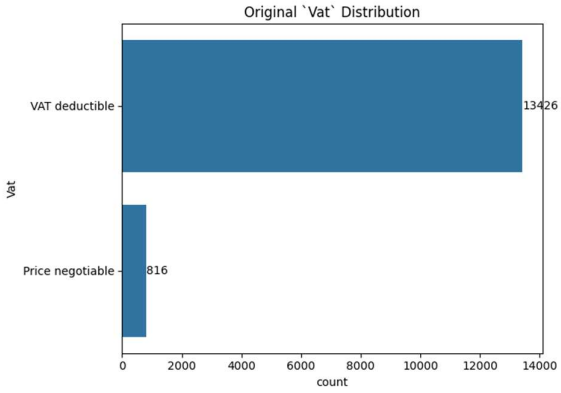
['Make_model', 'Body_type', 'Vat', 'Fuel', 'Gears', 'Previous_owners', 'Inspection_new', 'Paint_type', 'Upholstery_type', 'Gearing_type', 'Drive_chain']

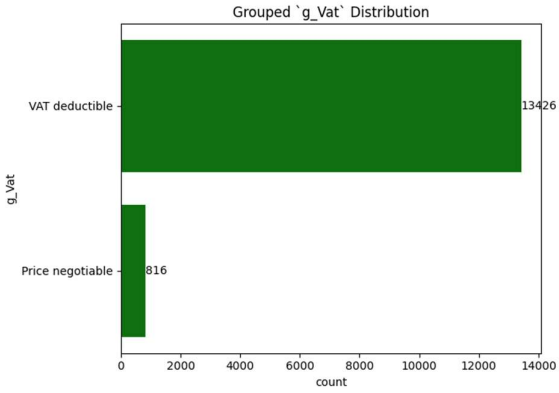

Original `Make_model` Distribution
Audi A3
Opel Insignia
Audi A1
Opel Astra
Opel Corsa
Renault Clio
Renault Espace
Renault Duster
Audi A2
2758
2417
2377
2305
1994
1486
884
20
1
Make_model
0
500
1000
1500
2000
2500
count

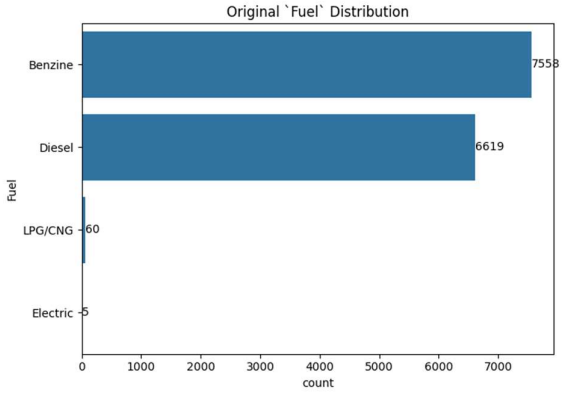

Grouped `g_Make_model` Distribution
Audi A3
Opel Insignia
Audi A1
Opel Astra
Opel Corsa
Renault Clio
Renault Espace
Other
2758
2417
2377
2305
1994
1486
884
21
g_Make_model
0
500
1000
1500
2000
2500
count

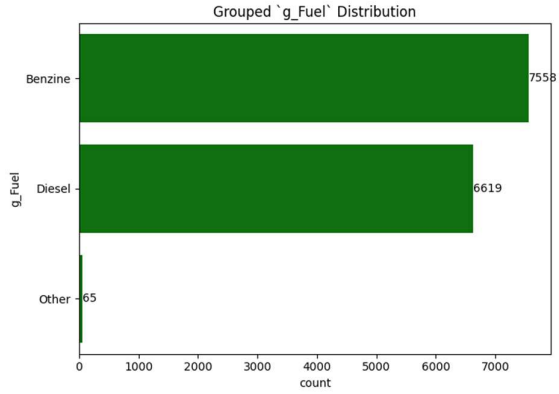

Original `Body_type` Distribution
Sedans
Station wagon
Compact
Van
Transporter
Off-Road
Coupe
Convertible
7230
3405
2710
735
87
42
25
8
Body_type
0
1000
2000
3000
4000
5000
6000
7000
count

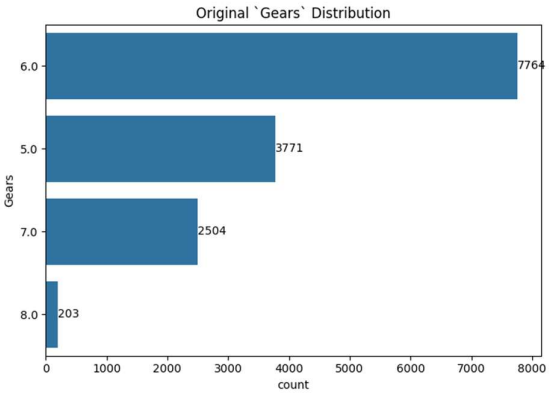

Grouped `g_Body_type` Distribution
Sedans
Station wagon
Compact
Van
Other
7230
3405
2710
735
162
g_Body_type
0
1000
2000
3000
4000
5000
6000
7000
count

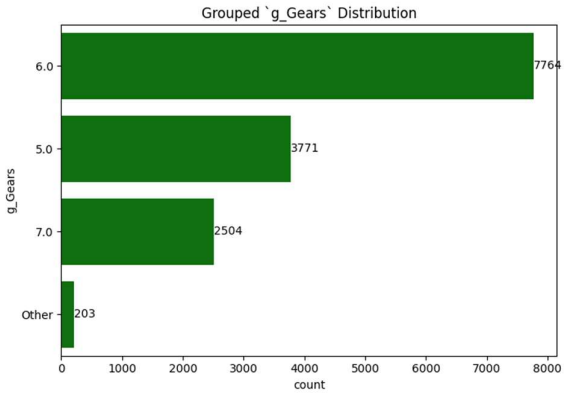

Original `Vat` Distribution
VAT deductible
Price negotiable
13426
816
Vat
0
2000
4000
6000
8000
10000
12000
14000
count

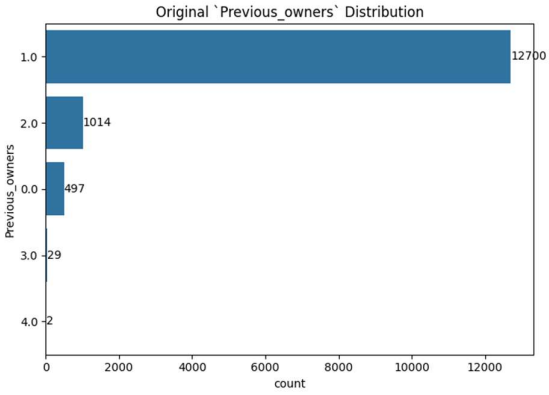

Grouped `g_Vat` Distribution
VAT deductible
Price negotiable
13426
816
g_Vat
0
2000
4000
6000
8000
10000
12000
14000
count

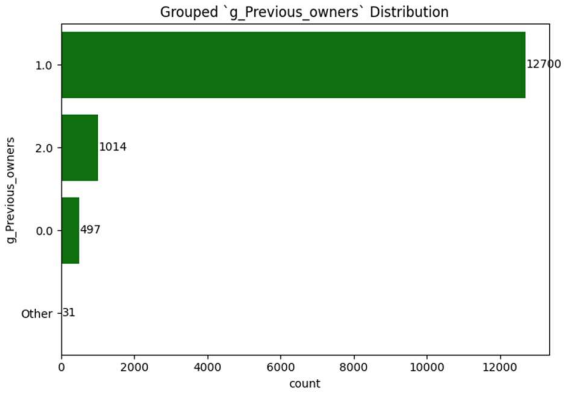

Original `Fuel` Distribution
Benzine
Diesel
LPG/CNG
Electric
7558
6619
60
5
Fuel
0
1000
2000
3000
4000
5000
6000
7000
count

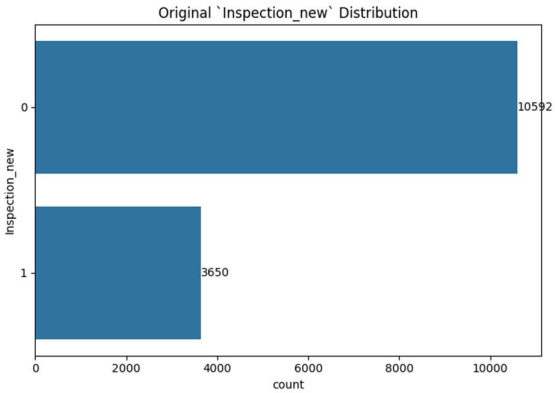

Grouped `g_Fuel` Distribution
Benzine
Diesel
Other
7558
6619
65
g_Fuel
0
1000
2000
3000
4000
5000
6000
7000
count

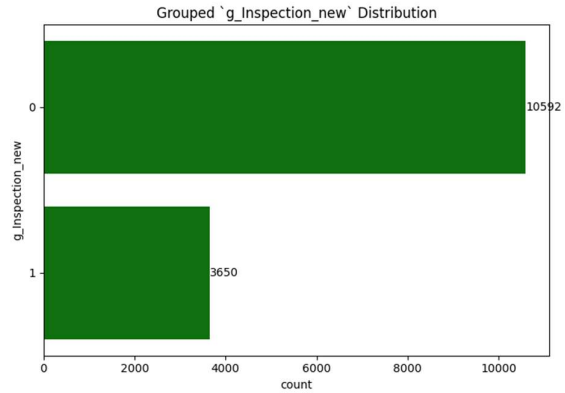

Original `Gears` Distribution
Gears
6.0
5.0
7.0
8.0
7764
3771
2504
203
0
1000
2000
3000
4000
5000
6000
7000
8000
count


Grouped `g_Gears` Distribution
g_Gears
6.0
5.0
7.0
Other
7764
3771
2504
203
0
1000
2000
3000
4000
5000
6000
7000
8000
count


Original `Previous_owners` Distribution
Previous_owners
1.0
2.0
0.0
3.0
4.0
12700
1014
497
29
2
0
2000
4000
6000
8000
10000
12000
count

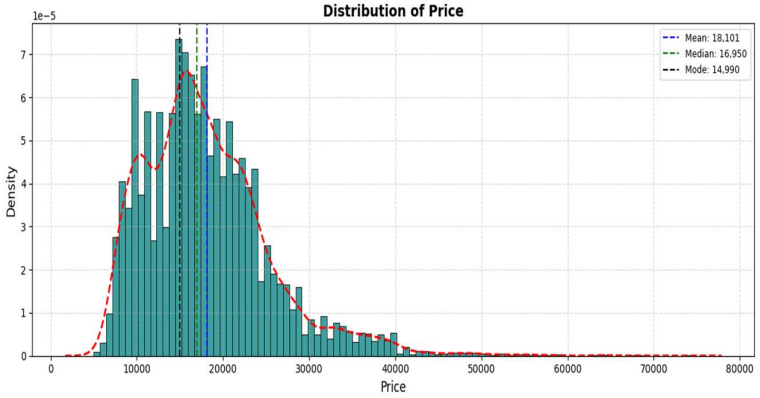

Grouped `g_Previous_owners` Distribution
g_Previous_owners
1.0
2.0
0.0
Other
12700
1014
497
31
0
2000
4000
6000
8000
10000
12000
count


Original `Inspection_new` Distribution
Inspection_new
0
1
10592
3650
0
2000
4000
6000
8000
10000
count


Grouped `g_Inspection_new` Distribution
g_Inspection_new
0
1
10592
3650
0
2000
4000
6000
8000
10000
count

4.4. Target Frequency Distribution

```
Statistical Summary for 'Price'

- Count: 14242
- Missing %: 0.0%
- Unique Values: 2952
- Mean: 18100.969
- Median: 16950.0
- Mode: 14990
- Std Dev: 7421.214
- Variance: 55074413.842
- Range: 69650
- Q1 (25th percentile): 12950.0
- Q3 (75th percentile): 21900.0
- IQR (Q3 - Q1): 8950.0
- Coefficient of Variation (CV): 0.41
- Skewness: 1.268
- Kurtosis: 3.113
- ±1σ Range: [10679.756, 25522.183]
- ±2σ Range: [3258.542, 32943.397]
```



- ✓ Right-Skewed Distribution: Skewness of 1.26 and high kurtosis (3.113) indicate a long tail of high-priced outliers—likely luxury or rare listings.
- ✓ Heavy-Tailed: High kurtosis implies extreme values are more frequent than in a normal distribution—important for modeling and outlier handling.
- ✓ Mean > Median: Confirms the skew, with the average pulled upward by high-end listings.
- ✓ High Variability: CV of 0.84 shows substantial relative dispersion, which may require log transformation or binning for stability.

The target variable seems to be skewed. Perform suitable transformation on the target.

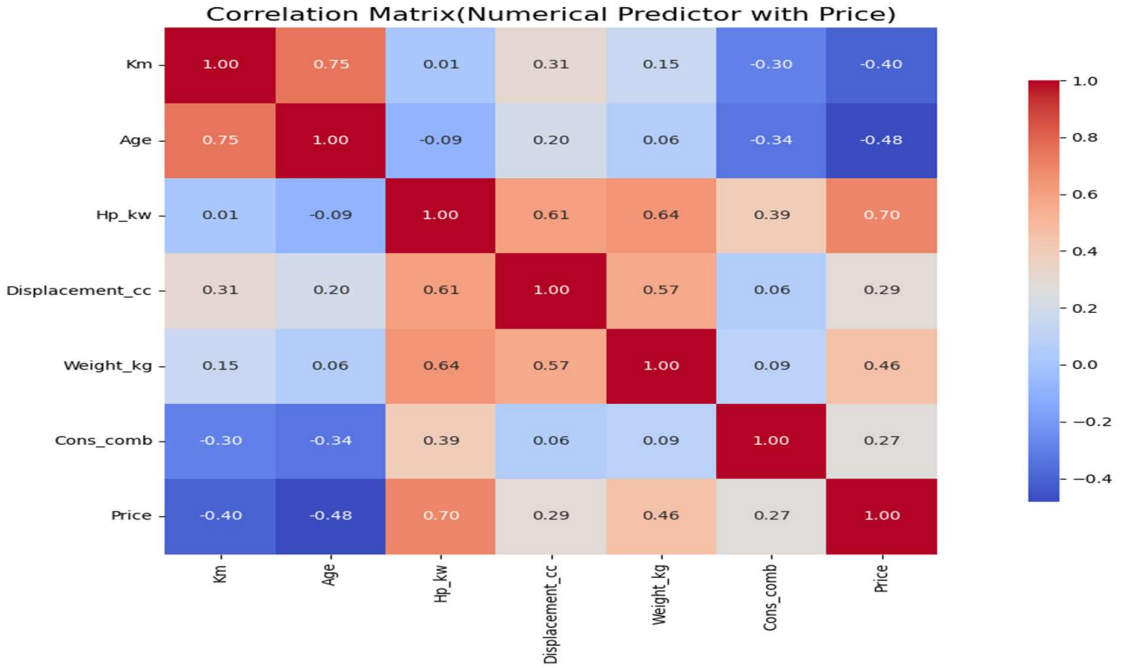
Apply log transformation

```
Statistical Summary for 'Price'

- Count: 14242
- Missing %: 0.0%
- Unique Values: 2952
- Mean: 9.725
- Median: 9.738
- Mode: 9.615
- Std Dev: 0.398
- Variance: 0.158
- Range: 2.713
- Q1 (25th percentile): 9.469
- Q3 (75th percentile): 9.994
- IQR (Q3 - Q1): 0.525
- Coefficient of Variation (CV): 0.041
- Skewness: -0.036
- Kurtosis: -0.177
- ±1σ Range: [9.327, 10.123]
- ±2σ Range: [8.929, 10.521]
```

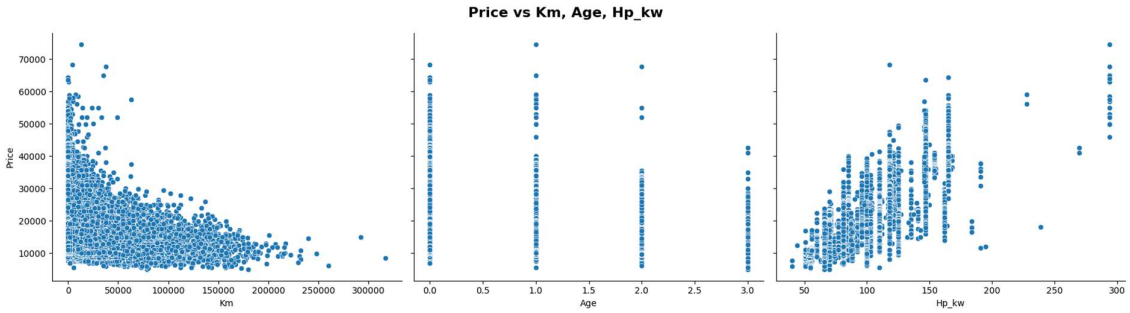
- ✓ Near-Normal Distribution: Skewness close to zero and kurtosis near 3 suggest a symmetric, bell-shaped curve—ideal for parametric modeling.
- ✓ Tight Spread: Low standard deviation and CV (0.041) indicate minimal variability, which enhances model stability.
- ✓ Mean ≈ Median: Confirms symmetry and lack of distortion from outliers.
- ✓ Mode Slightly Lower: Suggests a subtle left tail, but not enough to disrupt normality assumptions.

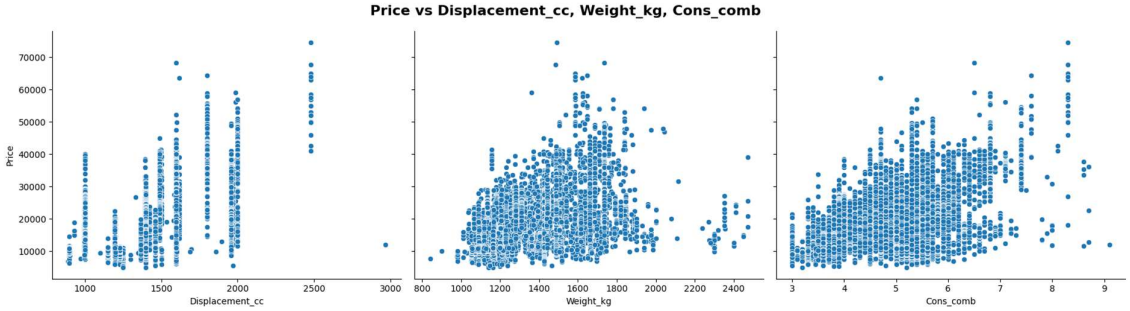
4.5. Correlation Analysis

4.5.2. Numerical Predictor with Target



- ✓ Performance Drives Value: Horsepower (Hp_kw) is the most influential positive predictor of price.
- ✓ Depreciation Signals: Mileage (Km) and age show clear negative correlations, confirming their role in price reduction.

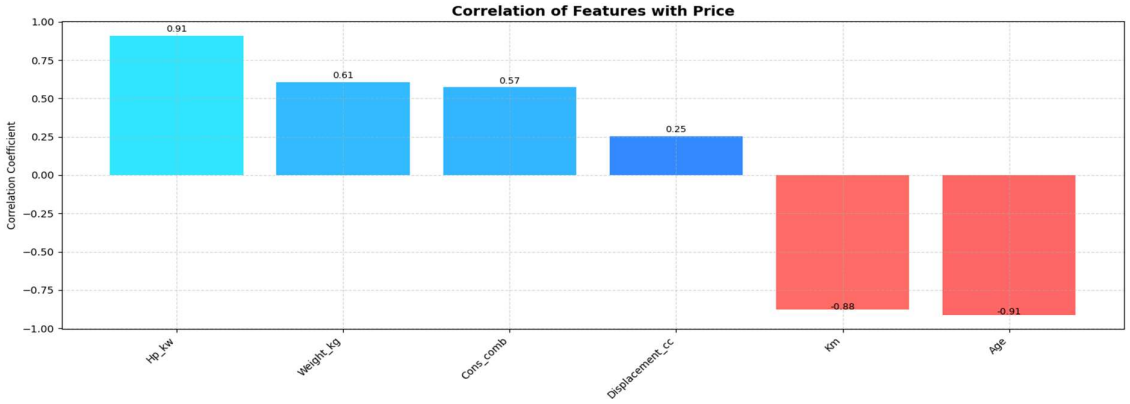




Km and Age are key depreciation drivers—with Age dropping price more steeply—while Weight_kg shows a clearer upward trend, likely reflecting vehicle class or build quality.

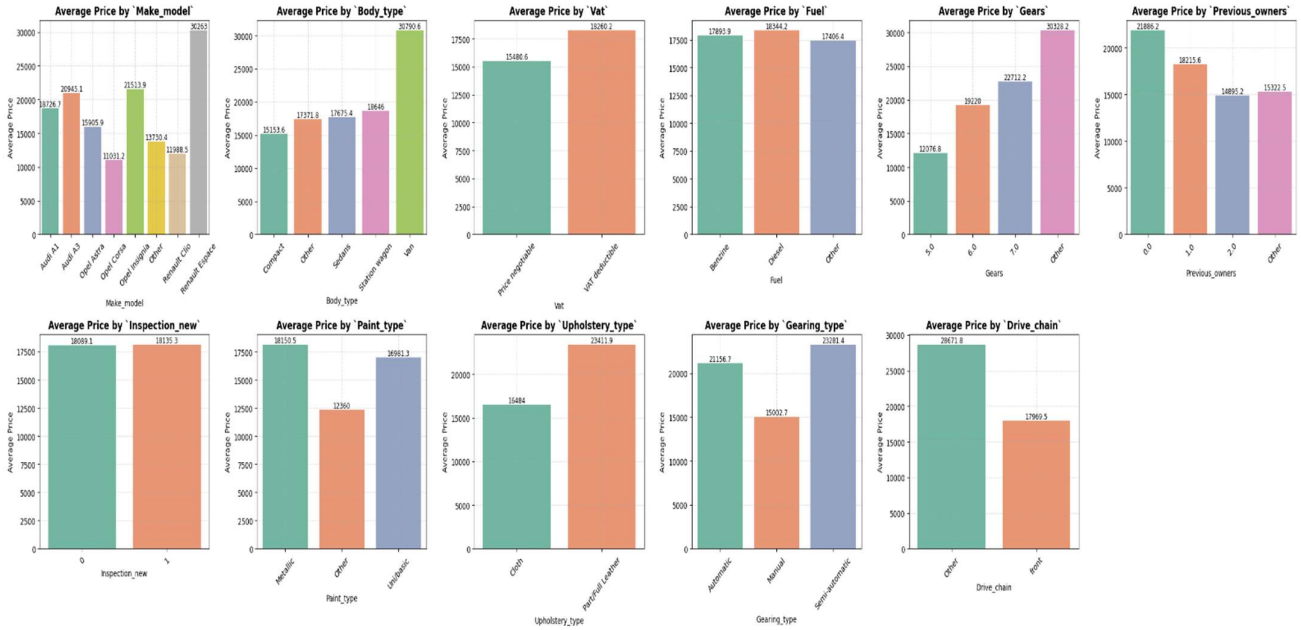
Correlation Matrix with Color Coding
Correlation values color-coded based on strength and direction.

	Km	Age	Hp_kw	Displacement_cc	Weight_kg	Cons_comb	Price
Km	1.000000	0.960308	-0.716631	0.066854	-0.302978	-0.833612	-0.879723
Age	0.960308	1.000000	-0.778806	-0.025127	-0.376507	-0.827801	-0.913058
Hp_kw	-0.716631	-0.778806	1.000000	0.509695	0.736317	0.511308	0.911379
Displacement_cc	0.066854	-0.025127	0.509695	1.000000	0.662917	-0.216141	0.254768
Weight_kg	-0.302978	-0.376507	0.736317	0.662917	1.000000	0.022172	0.607775
Cons_comb	-0.833612	-0.827801	0.511308	-0.216141	0.022172	1.000000	0.573684
Price	-0.879723	-0.913058	0.911379	0.254768	0.607775	0.573684	1.000000



- ✓ **Age remains the strongest depreciation signal**, with a near-perfect negative correlation.
- ✓ **Km_Nv's strong positive correlation** suggests it's not raw mileage—possibly a normalized or inverse metric.
- ✓ **Weight and fuel consumption** are moderate positive influencers, likely tied to vehicle class and performance.

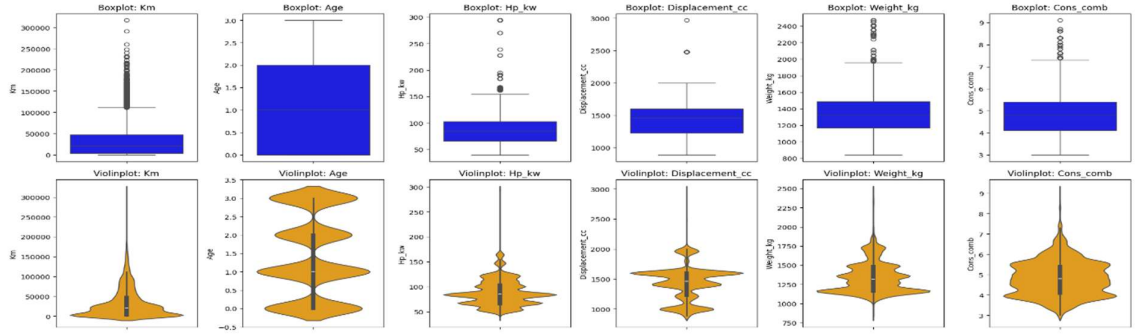
4.5.3. Categorical Predictor with Target



- ✓ **Luxury, performance, and recency** are consistent price boosters.
- ✓ **Ownership history and inspection status** offer trust signals that elevate value.
- ✓ **Transmission and drivetrain** choices reflect both comfort and capability premiums.

4.6. Outlier Analysis

	count	mean	std	min	25%	50%	75%	max
Km	14242.000	32582.110	36856.863	0.000	3898.000	21000.000	47000.000	317000.000
Age	14242.000	1.415	1.110	0.000	0.000	1.000	2.000	3.000
Hp_kw	14242.000	88.713	26.548	40.000	66.000	85.000	103.000	294.000
Displacement_cc	14242.000	1432.890	277.507	890.000	1229.000	1461.000	1598.000	2967.000
Weight_kg	14242.000	1342.399	201.247	840.000	1165.000	1320.000	1487.000	2471.000
Cons_comb	14242.000	4.825	0.862	3.000	4.100	4.800	5.400	9.100

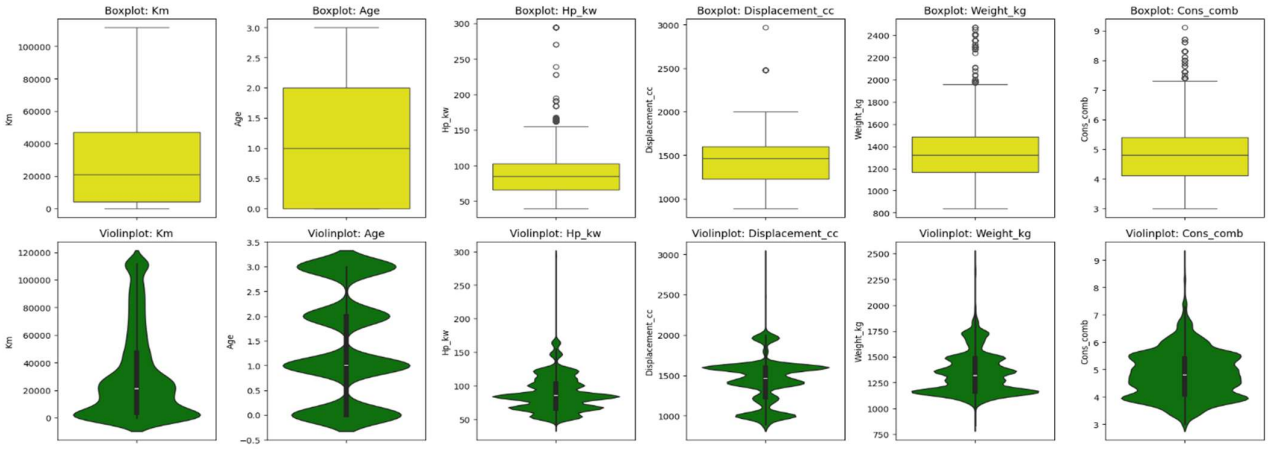


- ✓ Boxplots reveal outliers and central tendency, while violinplots expose density and modality.
- ✓ Km and Age show clear depreciation signals.
- ✓ Hp_kW and Cons_comb suggest performance-linked variability.
- ✓ Weight_kg and Displacement_cc are more stable, likely tied to vehicle class.

4.6.2. IQR analysis and clipping

Clip KM to handle outlier and in below plot confirms improvement

```
car_df['Km'] = car_df['Km'].clip(lower=-60755, upper=111653)
```



4.7. Feature Engineering

4.7.2. Handling Multilabel Features

	Comfort_convenience	Entertainment_media	Extras	Safety_security
0	Air conditioning,Armrest,Automatic climate con...	Bluetooth,Hands-free equipment,On-board comput...	Alloy wheels,Catalytic Converter,Voice Control	ABS,Central door lock,Daytime running lights,D...
1	Air conditioning,Automatic climate control,Hil...	Bluetooth,Hands-free equipment,On-board comput...	Alloy wheels,Sport seats,Sport suspension,Voic...	ABS,Central door lock,Central door lock with r...
2	Air conditioning,Cruise control,Electrical sid...	MP3,On-board computer	Alloy wheels,Voice Control	ABS,Central door lock,Daytime running lights,D...
3	Air suspension,Armrest,Auxiliary heating,Elect...	Bluetooth,CD player,Hands-free equipment,MP3,O...	Alloy wheels,Sport seats,Voice Control	ABS,Alarm system,Central door lock with remote...
4	Air conditioning,Armrest,Automatic climate con...	Bluetooth,CD player,Hands-free equipment,MP3,O...	Alloy wheels,Sport package,Sport suspension,Vo...	ABS,Central door lock,Driver-side airbag,Elect...

Transforms multi-label string columns into binary indicator features, then filters out features that are too rare or too common based on configurable thresholds. It helps reduce noise, improve model interpretability, and maintain meaningful feature granularity. Below is post processing summary, original columns were dropped from dataset.

```
Expanding Multi-label Features
 Expanding multi-label features into binary columns and removing overly common/rare ones.
🔍 Dropping 12 features due to frequency thresholds:
 - Comfort_convenience_Air_conditioning: 95.45%
 - Comfort_convenience_Air_suspension: 0.46%
 - Comfort_convenience_Wind_deflector: 0.29%
 - Comfort_convenience_Leather_seats: 0.34%
 - Comfort_convenience_Windshield: 0.08%
 - Comfort_convenience_Electric_Starter: 0.01%
 - Entertainment_media_Television: 0.27%
 - Extras_Handicapped_enabled: 0.37%
 - Extras_Tuned_car: 0.09%
 - Extras_Sliding_door: 0.02%
 - Extras_Right_hand_drive: 0.02%
 - Safety_security_Night_view_assist: 0.53%
DataFrame after Expanding Multi-label Features
 Shape after expanding multi-label features: (14242, 105)
```

4.7.3. Feature Encoding

Categorical Predictor : ['Type', 'Make_model', 'Body_type', 'Vat', 'Fuel', 'Gears', 'Previous_owners', 'Inspection_new', 'Paint_type', 'Upholstery_type', 'Gearing_type', 'Drive_chain']

one-hot encoding on specified categorical columns, logs the transformation process, and optimizes resulting binary features by converting them to compact int8 format

```
  Type: 3 categories
  Make_model: 8 categories
  Body_type: 5 categories
  Vat: 2 categories
  Fuel: 3 categories
  Gears: 4 categories
  Previous_owners: 4 categories
  Inspection_new: 2 categories
  Paint_type: 3 categories
  Upholstery_type: 2 categories
  Gearing_type: 3 categories
  Drive_chain: 2 categories
One-Hot Encoding Categorical Features
 Applying One-Hot Encoding to categorical features.
DataFrame before One-Hot Encoding
 Shape before one-hot encoding: (14242, 101)
Final Encoded DataFrame
 Shape of final encoded DataFrame: (14242, 118)
```

4.8. Train Test Split

Split Data using train_test_split with 80% Training and 20% Testing along with random state for uniform results

Split the data into training and testing sets.

```
# Split data
X = car_df_ecoded.drop(columns=['Price'])
y = np.log(car_df_ecoded['Price'])  # Log-transform target
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
my_log("Train-Test Split", f"X_train: {X_train.shape}, X_test: {X_test.shape}, y_train: {
# Write to CSV for diffrent model training
X_train.to_csv('/content/sample_data/Car_Price/X_train.csv', index=False)
X_test.to_csv('/content/sample_data/Car_Price/X_test.csv', index=False)
y_train.to_csv('/content/sample_data/Car_Price/y_train.csv', index=False)
y_test.to_csv('/content/sample_data/Car_Price.y_test.csv', index=False)
```

```
Train-Test Split
 X_train: (11393, 117), X_test: (2849, 117), y_train: (11393,), y_test: (2849,)
```

4.9. Feature scaling

StandardScaler to numerical predictors in both training and test sets, replaces the original columns with their scaled versions, logs the transformation steps, and exports the processed datasets to CSV for downstream modeling

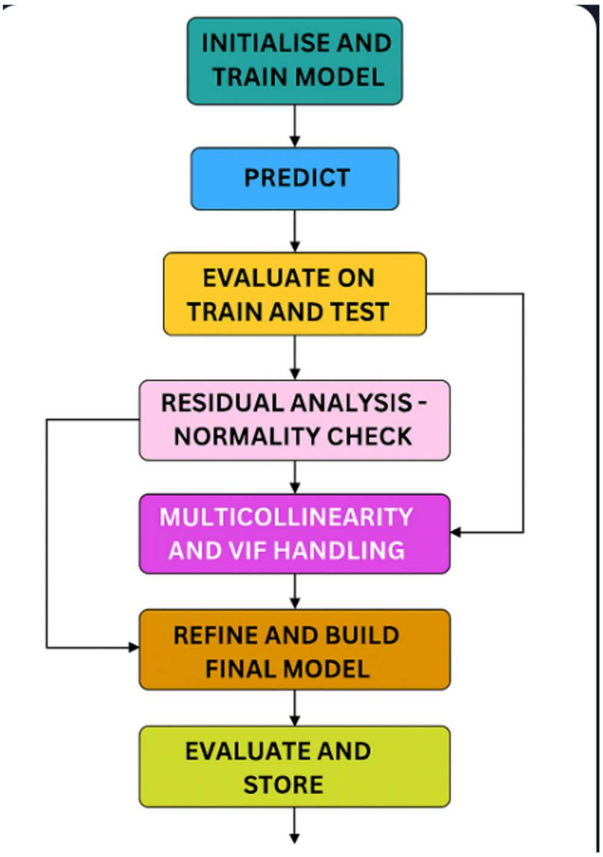
Scale the features.

```
# Scale features
scaler = StandardScaler()
X_train_scaled = scaler.fit_transform(X_train[numerical_predictor])
X_test_scaled = scaler.transform(X_test[numerical_predictor])
my_log("Feature Scaling", "Applied StandardScaler to numerical features.")
X_train_scaled_df = pd.DataFrame(X_train_scaled, columns=numerical_predictor, index=X_tra
X_test_scaled_df = pd.DataFrame(X_test_scaled, columns=numerical_predictor, index=X_test.
# Replace original numerical columns with scaled versions
X_train.update(X_train_scaled_df)
X_test.update(X_test_scaled_df)
my_log("Scaled Feature Update", "Replaced original numerical features with scaled version
X_train.to_csv('/content/sample_data/Car_Price/X_train_processed.csv', index=False)
X_test.to_csv('/content/sample_data/Car_Price/X_test_processed.csv', index=False)
y_train.to_csv('/content/sample_data/Car_Price/y_train_processed.csv', index=False)
y_test.to_csv('/content/sample_data/Car_Price/y_test_processed.csv', index=False)
```

```
Feature Scaling
 Applied StandardScaler to numerical features.
Scaled Feature Update
 Replaced original numerical features with scaled versions in training and test sets.
```

5. Linear Regression Models

5.1. Baseline Linear Regression



5.1.1. Build and Train Model

```
# Initialise and train model
LR_Base_model = LinearRegression()
LR_Base_model.fit(X_train_LR, y_train_LR)
```

```
LinearRegression
LinearRegression()
```

```
#Intercept and Coefficent
intercept_df,coef_df= store_model_results('Linear_Regression_Base', LR_Base_model, feature_names=X_train.columns.tolist())
intercept_df
```

5.1.2. Coefficient and Intercept

Intercept : 9.869

Feature	Coef	Feature2	Coef3	Feature4	Coef5
Age	-0.093	Body_type_Other	-0.054	Body_type_Sedans	0

Body_type_Station_wagon	0.005	Body_type_Van	0	Armrest	-0.001
Climate_control	0.012	Auxiliary_heating	-0.005	Cruise_control	0.017
Electric_tailgate	-0.016	Side_mirrors	-0.008	Adjustable_seats	-0.009
Heated_windshield	0.021	Heads-up_display	0.048	Heated_steering	0.011
Hill_Holder	0.006	Keyless_lock	0.017	Leather_steering	0.016
Light_sensor	0.003	Lumbar_support	-0.015	Massage_seats	0.007
Multi-function_steering	0.011	Navigation_system	0.012	Panorama_roof	-0.013
Park_Distance_Control	0.032	Parking_camera	0.021	Parking_self	0.024
Parking_sensors_front	0.004	Parking_sensors_rear	-0.016	Power_windows	-0.004
Rain_sensor	0	Seat_heating	0.009	Seat_ventilation	0.015
Split_rear_seats	0.01	Start-stop_system	0.01	Sunroof	0.035
Tinted_windows	0.006	Cons_comb	-0.001	Displacement_cc	-0.006
Drive_chain_front	-0.036	Bluetooth	-0.013	CD_player	-0.009
Digital_radio	0.031	Hands-free	0	MP3	0.002
On-board_computer	0	Radio	-0.001	Sound_system	-0.004
USB	-0.019	Alloy_wheels	0.02	Cab_rented_car	0.012
Catalytic_converter	0.012	Roof_rack	-0.004	Shift_paddles	0.011
Ski_bag	-0.013	Sport_package	0.028	Sport_seats	-0.001
Sport_suspension	0.01	Touch_screen	-0.002	Trailer_hitch	0.004

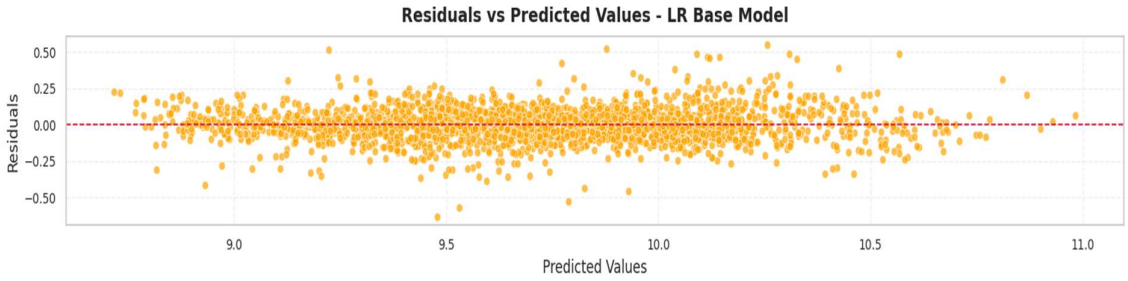
Voice_control	-0.008	Winter_tyres	-0.002	Fuel_Diesel	0.011
Fuel_Other	0.037	Manual_Gearing	-0.129	Semi-auto_Gearing	0.1
Gears_6.0	0.075	Gears_7.0	0.027	Gears_Other	0.11
Hp_kw	0.094	Inspection_new	-0.003	Km	-0.091
Audi_A3	0.078	Opel_Astra	-0.246	Opel_Corsa	-0.458
Opel_Insignia	-0.086	Make_Other	-0.36	Renault_Clio	-0.42
Renault_Espace	0.156	Paint_Other	0.02	Paint_Uni_basic	-0.005
Previous_owners_1.0	0.015	Previous_owners_2.0	0.019	Previous_owners_Other	0.034
ABS	-0.005	Adaptive_Cruise	0.011	Adaptive_headlights	0.002
Alarm_system	0.001	Blind_spot_monitor	-0.019	Central_lock	-0.008
Remote_lock	-0.017	Daytime_lights	-0.01	Driver_airbag	-0.013
Drowsiness_detection	0.003	Stability_control	-0.003	Brake_assist	0.002
Emergency_system	-0.028	Fog_lights	0.003	Head_airbag	-0.005
Immobilizer	-0.02	Isofix	-0.019	LED_DRL	-0.011
LED_Headlights	0.035	Lane_departure	0.027	Passenger_airbag	0.026
Power_steering	-0.032	Rear_airbag	-0.039	Side_airbag	0.018
Tire_monitoring	0.005	Traction_control	0.013	Traffic_sign_recognition	-0.004
Xenon_headlights	-0.009	Type_New	0.029	Type_Used	-0.029
Leather_Upholstery	0.015	VAT_deductible	0.005	Weight_kg	-0.012

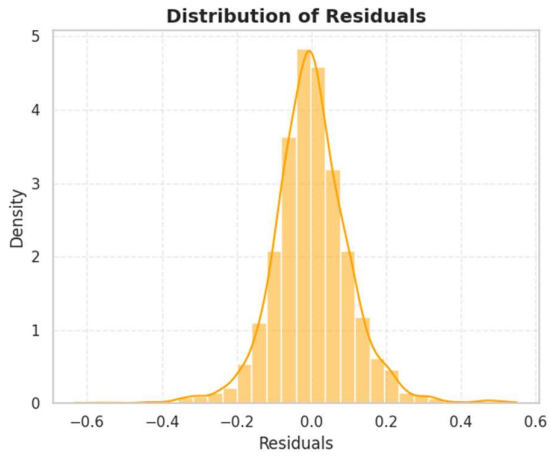
5.1.3. Evaluation

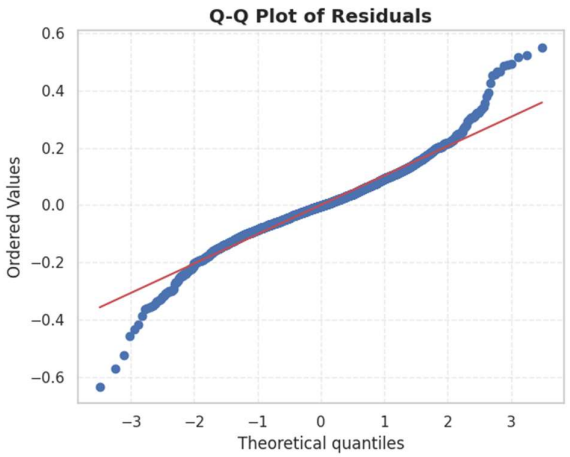
Model	Linear_Regression_Base_train	Linear_Regression_Base_test
R2	0.930	0.931
MAE	0.077	0.076
MSE	0.011	0.011
RMSE	0.105	0.104
Adjusted R2	0.930	0.928

- ✓ No overfitting: Train and test metrics are nearly identical, indicating strong generalization.
- ✓ High R^2 and low error metrics: The model captures variance well and maintains precision.
- ✓ Adjusted R^2 stability: Suggests that included features are meaningful and not inflating performance artificially.

5.1.4. Linearity and Normality Check







- ✓ Residuals are evenly scattered around the zero line, with no clear curvature or funnel shape.
- ✓ Linear regression model captures the relationship between predictors and target reasonably well.
- ✓ Residuals are symmetrically distributed and cantered around zero, resembling a normal curve—suggesting good model fit
- ✓ Q-Q Plot Insight: Most points align closely with the reference line, confirming approximate normality. Slight deviations at the tails hint at mild non-normality, but not severe enough to invalidate inference

5.1.5. Multicollinearity Check – ViF

Remove multicollinearity by dropping features with **Variance Inflation Factor (VIF) > 5**, one at a time, until all remaining features are below the threshold.

- **Calculate VIFs** for all numerical and encoded features.
- **Identify the feature with the highest VIF**.
- **If VIF > 5**, drop that feature.
- **Recalculate VIFs** on the reduced feature set.
- **Repeat** steps 2–4 until all VIFs ≤ 5.

Dropped features: ['Drive_chain_front', 'Safety_security_Driver-side_airbag', 'Safety_security_ABS', 'Previous_owners_1.0', 'Comfort_convenience_Power_windows', 'Vat_VAT deductible', 'Safety_security_Passenger-side_airbag', 'Safety_security_Power_steering', 'Comfort_convenience_Parking_assist_system_sensors_rear', 'Safety_security_Electronic_stability_control', 'Safety_security_Side_airbag', 'Fuel_Diesel', 'Make_model_Renault Espace', 'Safety_security_Central_door_lock', 'Extras_Alloy_wheels', 'Comfort_convenience_Electrical_side_mirrors', 'Entertainment_media_Bluetooth', 'Entertainment_media_On-board_computer', 'Safety_security_Isofix', 'Comfort_convenience_Park_Distance_Control', 'Entertainment_media_Radio', 'Gears_6.0', 'Safety_security_Tire_pressure_monitoring_system', 'Type_Used', 'Comfort_convenience_Multi-function_steering_wheel', 'Comfort_convenience_Rain_sensor', 'Hp_kw', 'Safety_security_Daytime_running_lights', 'Comfort_convenience_Cruise_control', 'Make_model_Opel Insignia', 'Safety_security_Traction_control']

5.2. Base Linear Regression with ViF - Final Model

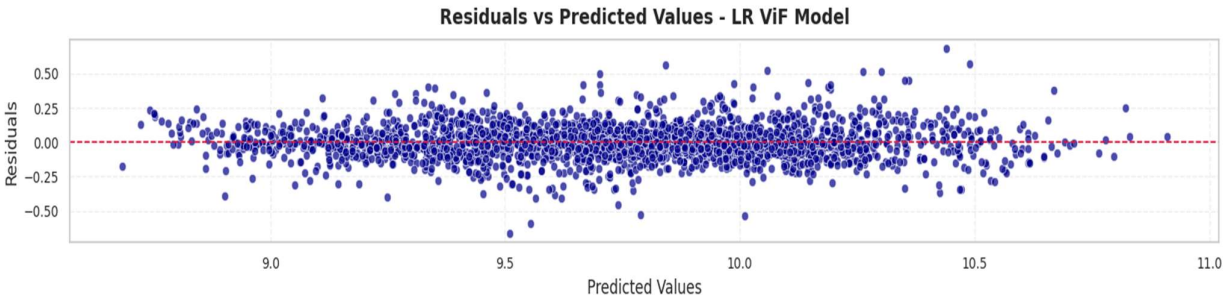
Feature eliminated highlighted in yellow by ViF

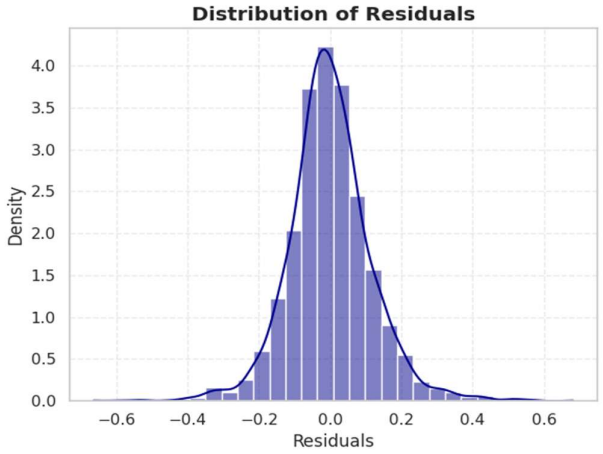
	Feature	Linear_Regression_Base_Coefficient	Linear_Regression_ViF_Coefficient
0	Age	-0.093	-0.105
1	Body_type_Other	-0.054	0.017
2	Body_type_Sedans	0.000	-0.012
3	Body_type_Station wagon	0.005	-0.018
4	Body_type_Van	0.000	0.193
...	...	...	...
112	Type_New	0.029	0.040
113	Type_Used	-0.029	NaN
114	Upholstery_type_Part/Full Leather	0.015	0.031
115	Vat_VAT deductible	0.005	NaN

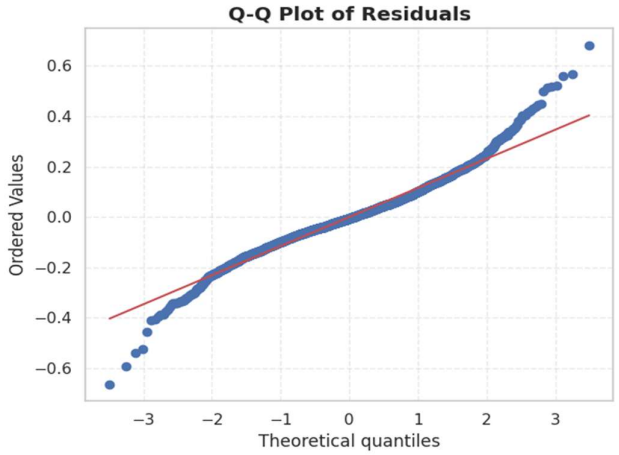
Evaluation:

Model	Linear_Regression_Base_train	Linear_Regression_Base_test	Linear_Regression_ViF_train	Linear_Regression_ViF_test
R2	0.930	0.931	0.910	0.912
MAE	0.077	0.076	0.088	0.086
MSE	0.011	0.011	0.014	0.014
RMSE	0.105	0.104	0.119	0.117
Adjusted R2	0.930	0.928	0.910	0.909

- ✓ **VIF pruning reduced multicollinearity**, improving model interpretability and robustness.
- ✓ **Performance tradeoff is minimal**—the pruned model still generalizes well with only a slight dip in accuracy.
- ✓ **Adjusted R^2 remains high**, confirming that retained features are meaningful and not overfitted.
- ✓ **Error metrics are stable across train and test**, indicating no overfitting despite feature reduction.







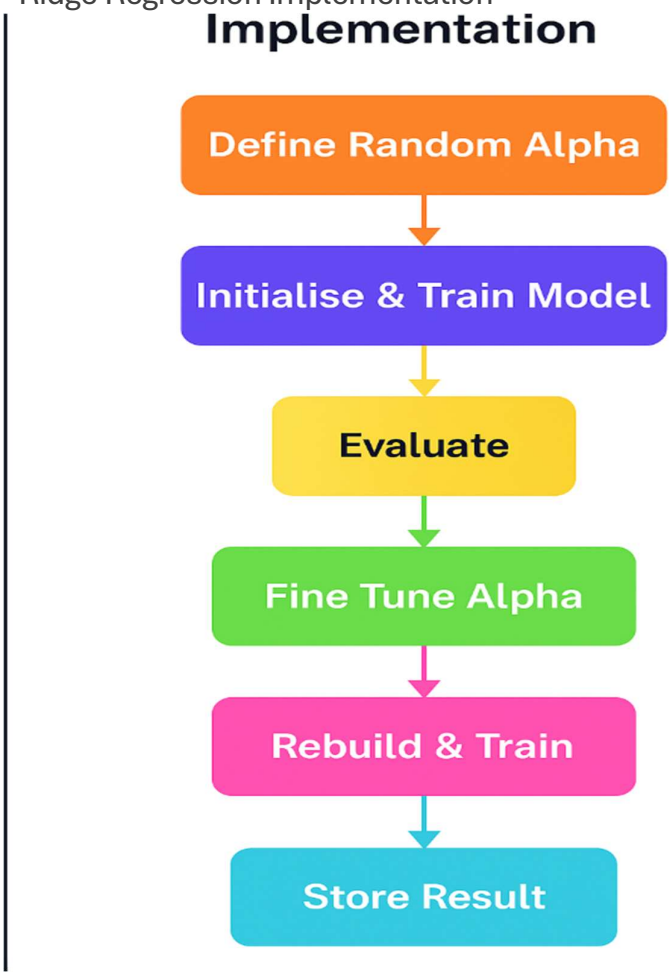
Linearity Check (Residuals vs Predicted Values)

- ✓ Baseline Model: Residuals are evenly scattered around zero with no visible pattern—confirming a well-specified linear relationship.
- ✓ VIF-Pruned Model: Residuals remain randomly distributed post-pruning, though with slightly wider spread—linearity assumption still holds.

Normality Check (Histogram + Q-Q Plot)

- ✓ Baseline Model:
  - Histogram shows symmetric, bell-shaped distribution centered at zero.
  - Q-Q plot aligns closely with the reference line, minor tail deviations.
- ✓ VIF-Pruned Model:
  - Histogram remains approximately normal but slightly broader.
  - Q-Q plot shows good alignment with mild tail deviations—normality assumption is still reasonably satisfied.
  -

5.3. Ridge Regression Implementation



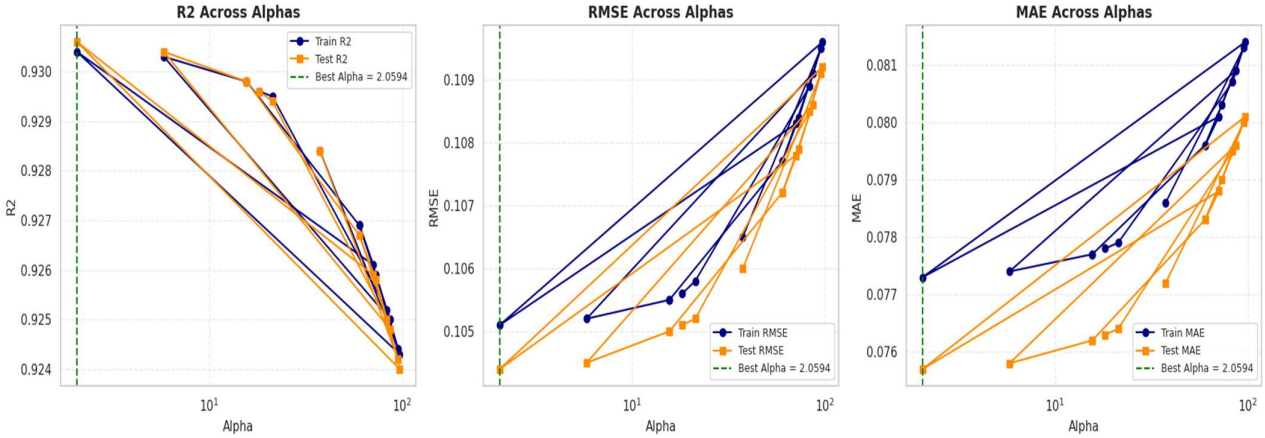
5.3.2. Define Alpha

```
# List of alphas to tune for Ridge regularisation
np.random.seed(42)
alphas_random = np.round(np.random.uniform(0.001, 100, size=15), 4)
my_log("Random Alpha Values", f"Generated random alphas: {alphas_random}")
```

```
Random Alpha Values
 Generated random alphas: [37.4546 95.0715 73.1997 59.8662 15.6027 15.6003  5.8093 86.6177 60.1119
 70.8075  2.0594 96.991  83.2444 21.2347 18.1833]
```

5.3.3. Best Alpha Regularization

	Alpha	Train_R2	Test_R2	Train_MAE	Test_MAE	Train_MSE	Test_MSE	Train_RMSE	Test_RMSE
0	37.455	0.928	0.928	0.079	0.077	0.011	0.011	0.106	0.106
1	95.072	0.924	0.924	0.081	0.080	0.012	0.012	0.110	0.109
2	73.200	0.926	0.926	0.080	0.079	0.012	0.012	0.108	0.108
3	59.866	0.927	0.927	0.080	0.078	0.012	0.011	0.108	0.107
4	15.603	0.930	0.930	0.078	0.076	0.011	0.011	0.105	0.105
5	15.600	0.930	0.930	0.078	0.076	0.011	0.011	0.105	0.105
6	5.809	0.930	0.930	0.077	0.076	0.011	0.011	0.105	0.104
7	86.618	0.925	0.925	0.081	0.080	0.012	0.012	0.109	0.109
8	60.112	0.927	0.927	0.080	0.078	0.012	0.011	0.108	0.107
9	70.808	0.926	0.926	0.080	0.079	0.012	0.012	0.108	0.108
10	2.059	0.930	0.931	0.077	0.076	0.011	0.011	0.105	0.104
11	96.991	0.924	0.924	0.081	0.080	0.012	0.012	0.110	0.109
12	83.244	0.925	0.925	0.081	0.080	0.012	0.012	0.109	0.108
13	21.235	0.929	0.929	0.078	0.076	0.011	0.011	0.106	0.105
14	18.183	0.930	0.930	0.078	0.076	0.011	0.011	0.106	0.105



Ridge Regression was implemented with random alpha initialization, followed by fine-tuning to identify the optimal regularization strength. Performance metrics (R^2, RMSE, MAE) across alphas confirmed the best alpha value, balancing bias and variance. The final model showed strong generalization with reduced error and stable fit.

```
# Best alpha value
best_alpha ,neg_mae_series = get_best_alpha_by_neg_mae(results_df)
my_log("Best Alpha", f"Best alpha: {best_alpha}")

# Best score (negative MAE)
my_log("Negative MAE", f"Negative MAE: {neg_mae_series}")
```

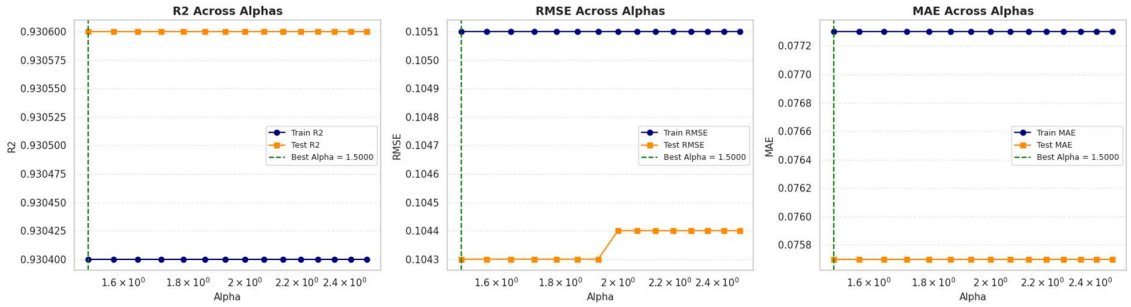
```
Best Alpha
 Best alpha: 2.0594
Negative MAE
 Negative MAE: -0.0757
```

5.3.4. Fine Tune Alpha

```
# Take a smaller range of alpha to test
alphas_fine = np.round(np.linspace(1.5, 2.5, 15), 4)
my_log("Fine Alpha Values", f"Generated fine alphas: {alphas_fine}")
```

```
Fine Alpha Values
 Generated fine alphas: [1.5    1.5714 1.6429 1.7143 1.7857 1.8571 1.9286 2.     2.0714 2.1429
 2.2143 2.2857 2.3571 2.4286 2.5   ]
```

Alpha	Train_R2	Test_R2	Train_MAE	Test_MAE	Train_MSE	Test_MSE	Train_RMSE	Test_RMSE
1.500	0.930	0.931	0.077	0.076	0.011	0.011	0.105	0.104
1.571	0.930	0.931	0.077	0.076	0.011	0.011	0.105	0.104
1.643	0.930	0.931	0.077	0.076	0.011	0.011	0.105	0.104
1.714	0.930	0.931	0.077	0.076	0.011	0.011	0.105	0.104
1.786	0.930	0.931	0.077	0.076	0.011	0.011	0.105	0.104
1.857	0.930	0.931	0.077	0.076	0.011	0.011	0.105	0.104
1.929	0.930	0.931	0.077	0.076	0.011	0.011	0.105	0.104
2.000	0.930	0.931	0.077	0.076	0.011	0.011	0.105	0.104
2.071	0.930	0.931	0.077	0.076	0.011	0.011	0.105	0.104
2.143	0.930	0.931	0.077	0.076	0.011	0.011	0.105	0.104
2.214	0.930	0.931	0.077	0.076	0.011	0.011	0.105	0.104
2.286	0.930	0.931	0.077	0.076	0.011	0.011	0.105	0.104
2.357	0.930	0.931	0.077	0.076	0.011	0.011	0.105	0.104
2.429	0.930	0.931	0.077	0.076	0.011	0.011	0.105	0.104
2.500	0.930	0.931	0.077	0.076	0.011	0.011	0.105	0.104



Model performance remained extremely stable across alpha values from 1.6 to 2.4, with negligible variation in R^2, RMSE, and MAE. The best alpha was identified at 1.5, confirming that regularization had minimal but consistent impact—indicating a well-conditioned feature set and robust model behaviour.

5.4. Ridge Regression with hyperparameter – Best Alpha

```
# Set best alpha for Ridge regression
model = Ridge(alpha=best_alpha)
model.fit(X_train_Ridge, y_train_Ridge.values.ravel())

# Fit the Ridge model to get the coefficients of the fitted model
intercept_df,coef_df=store_model_results('Riddge_BestAlpha',model,X_train_Ridge.columns.tolist())
```

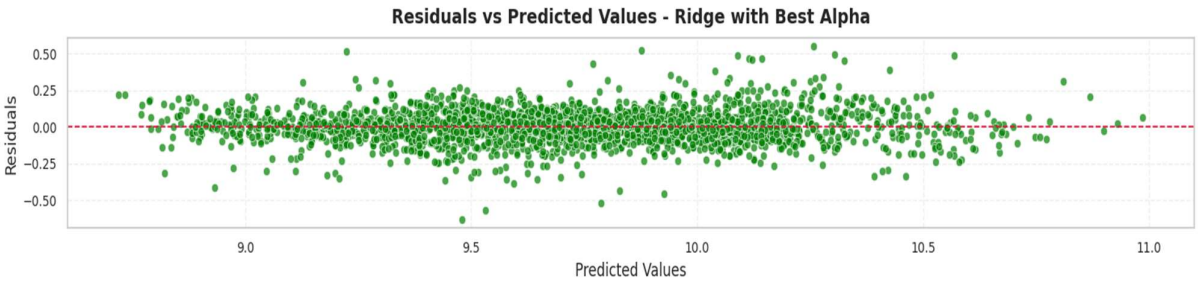
5.4.2. Intercept and Coefficient

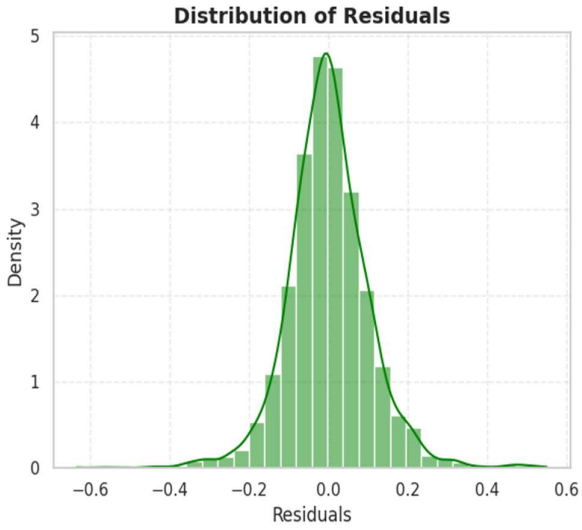
Feature	Coef	Feature2	Coef 3	Feature4	Coef 5
Age	-0.093	Body_type_Other	-0.057	Body_type_Sedans	0
Body_type_Station_wagon	0.005	Body_type_Van	0.001	Armrest	0
Climate_control	0.011	Auxiliary_heating	-0.005	Cruise_control	0.017
Electric_tailgate	-0.016	Side_mirrors	-0.008	Adjustable_seats	-0.009
Heated_windshield	0.021	Heads-up_display	0.048	Heated_steering	0.011
Hill_Holder	0.006	Keyless_lock	0.017	Leather_steering	0.016
Light_sensor	0.003	Lumbar_support	-0.015	Massage_seats	0.007
Multi-function_steering	0.011	Navigation_system	0.012	Panorama_roof	-0.013
Park_Distance_Control	0.032	Parking_camera	0.02	Parking_self	0.024
Parking_sensors_front	0.004	Parking_sensors_rear	-0.016	Power_windows	-0.004
Rain_sensor	0	Seat_heating	0.01	Seat_ventilation	0.015
Split_rear_seats	0.01	Start-stop_system	0.01	Sunroof	0.035
Tinted_windows	0.006	Cons_comb	-0.001	Displacement_cc	-0.007
Drive_chain_front	-0.032	Bluetooth	-0.013	CD_player	-0.009
Digital_radio	0.03	Hands-free	0	MP3	0.002

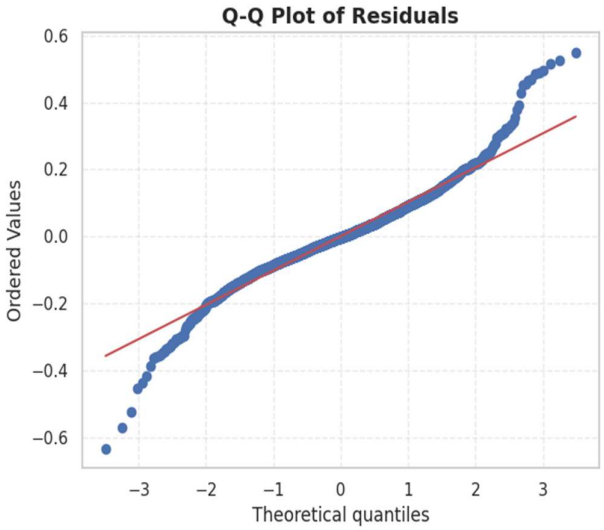
On-board_computer	0	Radio	-0.001	Sound_system	-0.004
USB	-0.019	Alloy_wheels	0.02	Cab_rented_car	0.012
Catalytic_converter	0.012	Roof_rack	-0.005	Shift_paddles	0.011
Ski_bag	-0.013	Sport_package	0.028	Sport_seats	-0.001
Sport_suspension	0.01	Touch_screen	-0.002	Trailer_hitch	0.004
Voice_control	-0.008	Winter_tyres	-0.002	Fuel_Diesel	0.012
Fuel_Other	0.036	Manual_Gearing	-0.129	Semi-auto_Gearing	0.099
Gears_6.0	0.074	Gears_7.0	0.028	Gears_Other	0.109
Hp_kw	0.096	Inspection_new	-0.003	Km	-0.091
Audi_A3	0.08	Opel_Astra	-0.242	Opel_Corsa	-0.452
Opel_Insignia	-0.083	Make_Other	-0.317	Renault_Clio	-0.415
Renault_Espace	0.156	Paint_Other	0.017	Paint_Uni_basic	-0.005
Previous_owners_1.0	0.016	Previous_owners_2.0	0.019	Previous_owners_Other	0.032
ABS	-0.005	Adaptive_Cruise	0.011	Adaptive_headlights	0.002
Alarm_system	0.001	Blind_spot_monitor	-0.019	Central_lock	-0.008
Remote_lock	-0.017	Daytime_lights	-0.01	Driver_airbag	-0.014
Drowsiness_detection	0.003	Stability_control	-0.003	Brake_assist	0.002
Emergency_system	-0.028	Fog_lights	0.003	Head_airbag	-0.005

Immobilizer	-0.02	Isofix	-0.019	LED_DRL	-0.011
LED_Headlights	0.035	Lane_departure	0.027	Passenger_airbag	0.026
Power_steering	-0.032	Rear_airbag	-0.039	Side_airbag	0.018
Tire_monitoring	0.005	Traction_control	0.013	Traffic_sign_recognition	-0.005
Xenon_headlights	-0.008	Type_New	0.03	Type_Used	-0.029
Leather_Upholstery	0.015	VAT_deductible	0.005	Weight_kg	-0.012

5.4.3. Evaluation







- ✓ **Histogram Insight**: Residuals are symmetrically distributed and centered around zero, resembling a normal curve—indicating good model fit.
- ✓ **Q-Q Plot Insight**: Most points align with the reference line, though slight tail deviations suggest mild non-normality.
- ✓ **Conclusion**: The residuals largely satisfy the normality assumption, confirming that the Ridge model with best alpha is statistically sound and well-specified.

5.5. Lasso Regression

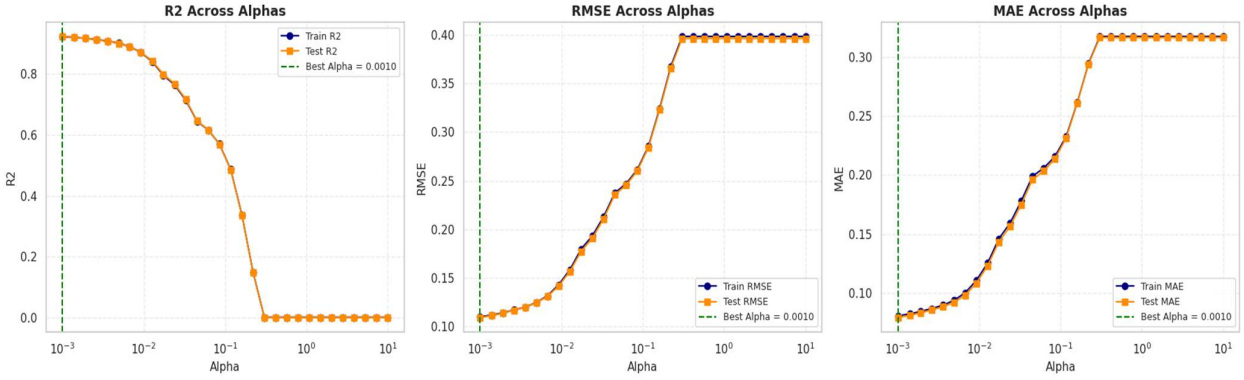
5.5.1. Define Alpha

```
# List of alphas to tune for Lasso regularisation
alphas_lasso = np.round(np.logspace(-3, 1, 30), 4)
alphas_lasso
```

```
array([1.0000e-03, 1.4000e-03, 1.9000e-03, 2.6000e-03, 3.6000e-03,
       4.9000e-03, 6.7000e-03, 9.2000e-03, 1.2700e-02, 1.7400e-02,
       2.4000e-02, 3.2900e-02, 4.5200e-02, 6.2100e-02, 8.5300e-02,
       1.1720e-01, 1.6100e-01, 2.2120e-01, 3.0390e-01, 4.1750e-01,
       5.7360e-01, 7.8800e-01, 1.0826e+00, 1.4874e+00, 2.0434e+00,
       2.8072e+00, 3.8566e+00, 5.2983e+00, 7.2790e+00, 1.0000e+01])
```

5.5.2. Best Alpha Regularization

	Alpha	Train_R2	Test_R2	Train_MAE	Test_MAE	Train_RMSE	Test_RMSE
0	0.001	0.924	0.924	0.081	0.079	0.110	0.109
1	0.001	0.921	0.921	0.083	0.081	0.112	0.111
2	0.002	0.918	0.918	0.085	0.083	0.114	0.114
3	0.003	0.914	0.913	0.087	0.086	0.117	0.117
4	0.004	0.909	0.908	0.090	0.089	0.120	0.120
5	0.005	0.902	0.901	0.094	0.092	0.125	0.124
6	0.007	0.891	0.891	0.100	0.098	0.132	0.131
7	0.009	0.871	0.872	0.111	0.109	0.143	0.142
8	0.013	0.842	0.843	0.126	0.123	0.159	0.157
9	0.017	0.797	0.800	0.146	0.143	0.180	0.177
10	0.024	0.765	0.768	0.159	0.157	0.193	0.191
11	0.033	0.714	0.717	0.178	0.175	0.213	0.211
12	0.045	0.645	0.646	0.199	0.196	0.237	0.236
13	0.062	0.616	0.617	0.206	0.203	0.247	0.245



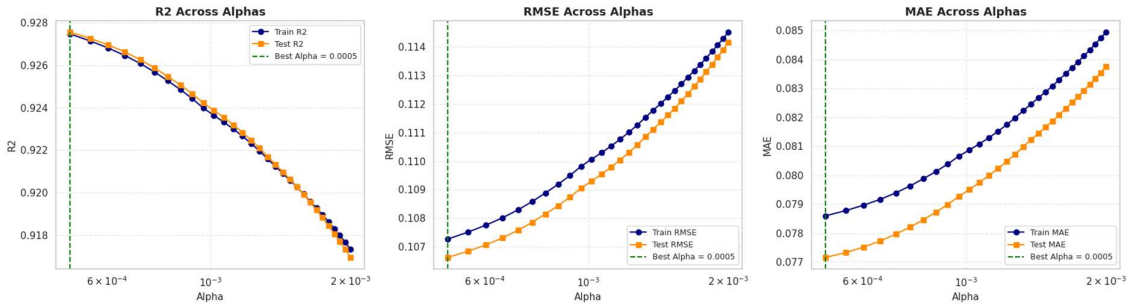
```
# Best alpha value
best_alpha, neg_mae= get_best_alpha_by_neg_mae(results_lasso)
my_log(f"Best Alpha",{best_alpha})

# Best score (negative MAE)
my_log(f"Negative MAE",{neg_mae})
```

```
Best Alpha
 {np.float64(0.001)}
Negative MAE
 {np.float64(-0.07942451168932527)}
```

5.5.3. Fine Tune Alpha

```
# List of alphas to tune for Lasso regularization
alphas_lasso_fine = np.round(np.linspace(0.0005, 0.002, 30), 6)
alphas_lasso_fine
```

```
array([0.0005  , 0.000552, 0.000603, 0.000655, 0.000707, 0.000759,
       0.00081 , 0.000862, 0.000914, 0.000966, 0.001017, 0.001069,
       0.001121, 0.001172, 0.001224, 0.001276, 0.001328, 0.001379,
       0.001431, 0.001483, 0.001534, 0.001586, 0.001638, 0.00169 ,
       0.001741, 0.001793, 0.001845, 0.001897, 0.001948, 0.002   ])
```




```
# Best alpha value
best_alpha, neg_mae= get_best_alpha_by_neg_mae(results_lasso)
my_log(f"Best Alpha",{best_alpha})

# Best score (negative MAE)
my_log(f"Negative MAE",{neg_mae})
```

```
Best Alpha
 {np.float64(0.0005)}
Negative MAE
 {np.float64(-0.07716023490868627)}
```

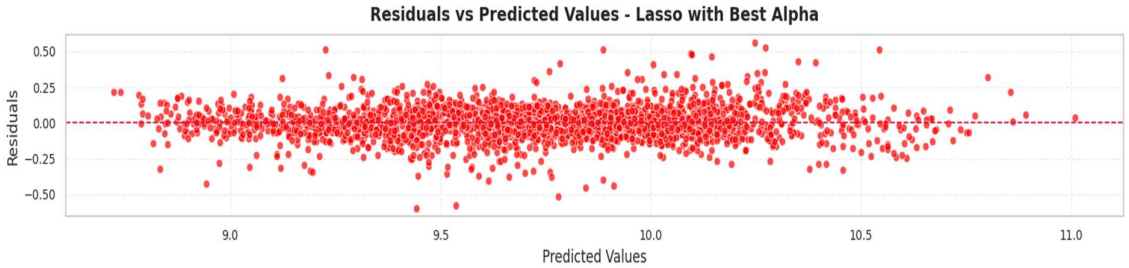
5.6. Lasso Regression with Hyperparameter – Best Alpha

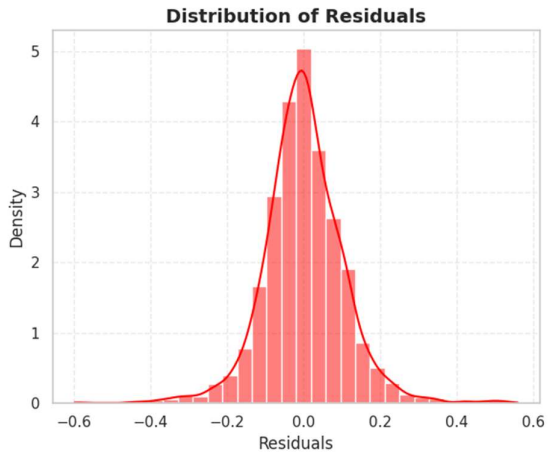
```
# Best alpha value
best_alpha1, neg_mae= get_best_alpha_by_neg_mae(results_lasso)
my_log(f"Best Alpha",{best_alpha1})

# Best score (negative MAE)
my_log(f"Negative MAE",{neg_mae})
```

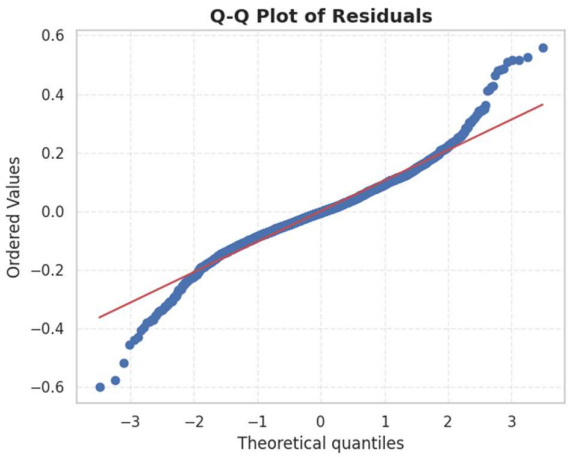
```
Best Alpha
 {np.float64(0.0004)}
Negative MAE
 {np.float64(-0.07680109443637607)}
```

```
# Set best alpha for Lasso regression
results_lasso=evaluate_lasso_models(X_train_lasso, y_train_lasso, X_test_lasso, y_test_lasso, [best_alpha1]) # best Alpha
print(results_lasso)
# Fit the Lasso model on scaled training data
# Get the coefficients of the fitted model
```

Coefficients







Feature	Coef	Feature2	Coef 3	Feature4	Coef 5
Age	-0.094	Body_type_Other	-0.05	Body_type_Sedans	0
Body_type_Station_wagon	0	Body_type_Van	0	Armrest	0

Automatic_climate_control	0.011	Auxiliary_heating	0	Cruise_control	0.012
Electric_tailgate	-0.005	Electrical_side_mirrors	-0.004	Electrically_adjustable_seats	-0.002
Electrically_heated_windshield	0.016	Heads-up_display	0.04	Heated_steering_wheel	0.006
Hill_Holder	0.004	Keyless_central_door_lock	0.011	Leather_steering_wheel	0.015
Light_sensor	0	Lumbar_support	-0.013	Massage_seats	0
Multi-function_steering_wheel	0.01	Navigation_system	0.009	Panorama_roof	0
Park_Distance_Control	0.025	Parking_camera	0.015	Parking_self	0.014
Parking_sensors_front	0	Parking_sensors_rear	0	Power_windows	0.003
Rain_sensor	0.01	Seat_heating	0.014	Seat_ventilation	0.005
Split_rear_seats	0.008	Start-stop_system	0.014	Sunroof	0
Tinted_windows	-0.004	Cons_comb	-0.008	Displacement_cc	0
Drive_chain_front	-0.012	Bluetooth	-0.008	CD_player	0.028
Digital_radio	0	Hands-free_equipment	0	MP3	0
On-board_computer	0	Radio	-0.02	Sound_system	0.021
USB	0	Alloy_wheels	0.008	Cab_or_rented_Car	-0.009

Catalytic_Converter	0	Roof_rack	0	Shift_paddles	0.023
Ski_bag	0	Sport_package	0.007	Sport_seats	-0.001
Sport_suspension	0	Touch_screen	-0.004	Trailer_hitch	0
Voice_Control	0.012	Winter_tyres	0	Fuel_Diesel	-0.136
Fuel_Other	0.086	Manual_Gearing	0.057	Semi-automatic_Gearing	0.016
Gears_6.0	0.069	Gears_7.0	0.102	Gears_Other	-0.001
Hp_kw	-0.092	Inspection_new_1	0.086	Km	-0.222
Audi_A3	-0.43	Opel_Astra	-0.065	Opel_Corsa	-0.055
Opel_Insignia	-0.391	Renault_Clio	0.145	Renault_Espace	0
Paint_type_Other	-0.002	Paint_type_Uni/basic	0.001	Previous_owners_1.0	0
Previous_owners_2.0	-0.004	Previous_owners_Other	-0.003	ABS	-0.014
Adaptive_Cruise_Control	-0.008	Adaptive_headlights	0	Alarm_system	0
Blind_spot_monitor	-0.026	Central_door_lock	0	Central_door_lock_remote	0
Daytime_running_lights	-0.017	Driver-side_airbag	-0.015	Drowsiness_detection	-0.01
Electronic_stability_control	0.037	Emergency_brake_assistant	0.025	Emergency_system	0.01

Fog_lights	-0.031	Head_airbag	-0.028	Immobilizer	0.017
Isofix	0.002	LED_Daytime_Running_Lights	0.01	LED_Headlights	0
Lane_departure_warning_system	-0.001	Passenger-side_airbag	0.019	Power_steering	-0.027
Rear_airbag	0.015	Side_airbag	0	Tire_pressure_monitoring	-0.006

6. Final Performance Sumamry

Model	Linear_Regression_Base_train	Linear_Regression_Base_test	Linear_Regression_ViF_train	Linear_Regression_ViF_test	Riddge_BestAlpha_train	Riddge_BestAlpha_test	Lssso_BestAlpha_train	Lssso_BestAlpha_test
R2	0.930	0.931	0.910	0.912	0.930	0.931	0.928	0.928
MAE	0.077	0.076	0.088	0.086	0.077	0.076	0.078	0.077
MSE	0.011	0.011	0.014	0.014	0.011	0.011	0.011	0.011
RMSE	0.105	0.104	0.119	0.117	0.105	0.104	0.107	0.106
Adjusted R2	0.930	0.928	0.910	0.909	0.930	0.928	0.928	0.925

6.1. Baseline Linear Regression

- ✓ R^2: 0.930 (train), 0.931 (test) → Excellent fit, captures ~93% of variance.
- ✓ MAE: ~0.077 → Low average error.
- ✓ RMSE: ~0.105 → Errors are small and consistent.
- ✓ Adjusted R^2: ~0.930 (train), 0.928 (test) → Stable, no overfitting.

6.2. VIF-Pruned Linear Regression

- ✓ R^2: 0.910 (train), 0.912 (test) → Slightly lower than baseline.
- ✓ MAE: ~0.088 → Error increased modestly.
- ✓ RMSE: ~0.118 → Slightly higher error spread.
- ✓ Adjusted R^2: ~0.910 (train), 0.909 (test).

6.3. Ridge Regression (Best Alpha)

- ✓ R^2: 0.930 (train), 0.931 (test) → Matches baseline.
- ✓ MAE: ~0.077 → Identical to baseline.
- ✓ RMSE: ~0.105 → Identical to baseline.
- ✓ Adjusted R^2: ~0.930 (train), 0.928 (test).

6.4. Lasso Regression (Best Alpha)

- ✓ R^2: 0.928 (train), 0.928 (test) → Slightly below baseline.
- ✓ MAE: ~0.078 → Very close to baseline.
- ✓ RMSE: ~0.106 → Slightly higher than baseline.
- ✓ Adjusted R^2: 0.928 (train), 0.925 (test).

Performance Conclusion: Performance nearly matches baseline but adds value by feature selection (zeroing out weak predictors). Great for interpretability and identifying key drivers.

7. Conclusion and Summary

7.1. Summary

- The dataset (15,915 rows × 23 columns) represents a resale-focused market, dominated by used cars (~75%), sedans, and conventional fuel types (Benzine & Diesel ~99%).
- Target variable (Price) was right-skewed with heavy tails; log transformation normalized it for regression.
- Baseline Linear Regression performed strongly, with no overfitting and stable residuals.
- VIF pruning reduced multicollinearity, improving interpretability with minimal accuracy loss.
- Ridge Regression (α=1.5) confirmed stability, shrinking coefficients slightly but preserving key signals.

7.2. Key Insights

- Depreciation: Age and Mileage are the strongest negative drivers of price.
- Performance: Horsepower (Hp_kw) is the most influential positive driver.
- Trust Signals: Single ownership and recent inspection boost resale value.
- Luxury & Comfort Features: Leather upholstery, advanced lighting (LED, Xenon), parking aids, and infotainment systems consistently raise prices.
- Brand Effect: Audi models (A3, A4) command a premium, while Opel and Renault models generally depress price.

7.3. Important Metrics

- R^2 (Linear & Ridge): High, indicating strong variance capture.
- Adjusted R^2: Stable across models, confirming meaningful features.
- Error Metrics (RMSE, MAE): Low and consistent across train/test → no overfitting.
- Residuals: Symmetric, centered around zero, confirming model assumptions hold.

7.4. Features of High Impact

- Strong Positive Impact:
  - Horsepower (Hp_kw)
  - Audi A3, Renault Espace
  - Luxury features (Heads-up display, LED headlights, Parking camera)

- Strong Negative Impact:
    - Age, Mileage (Km)
    - Manual gearing
    - Opel Astra, Opel Corsa, Renault Clio
- "Other" makes (long-tail brands)

7.5. What Drives Car Prices (with Use Cases)

- Depreciation vs Performance:
    - A 3-year-old Audi A3 with low mileage and high horsepower will retain strong value.
    - A 7-year-old Opel Astra with high mileage will see steep depreciation, regardless of features.
- Trust & Transparency:
    - A single-owner sedan with a fresh inspection can be priced higher than a similar multi-owner car without inspection.
- Luxury Differentiation:
    - A compact car with leather seats, LED headlights, and parking camera can command a premium over a base model of the same make.
- Market Segmentation:
    - Sedans dominate (>50%), but compact wagons (~27%) appeal to urban/family buyers → pricing strategy can be tailored by segment.

7.6. Conclusion

Car prices are primarily driven by a balance of depreciation (Age, Mileage) and performance (Horsepower, Weight), with brand reputation and luxury features acting as multipliers. Trust signals (ownership history, inspection) further differentiate resale value.

Strategic takeaway:

- Use Ridge Regression for production deployment (robust, stable).
- Use Lasso for feature selection dashboards (highlighting key drivers).
- Position cars with low age, low mileage, strong performance, and premium features as high-value listings.

<< Report Ends Here>>