Predicting Average Price of Used Cars

DISSERTATION

Submitted in partial fulfillment
of the requirements for the course of

CSE 587 - Data Intensive Computing

By
Shri Gayathri Chandrasekar (ESDS - 50495167)
Sree Lekshmi Prasannan (ESDS - 50495144)
Jaskirat Singh (ESDS - 50495170)

Under the guidance of:

Dr. Shamsad Parvin
Department of Computer Science and Engineering

The State University of New York at Buffalo

Phase 1

Problem Statement

The market for secondhand cars is enormous and thriving, which defines the automotive sector. Making accurate projections about the average cost of used automobiles is becoming increasingly important as buyers look for trustworthy information to help them make informed selections.

In this project, we will use a data-driven approach to create a reliable predictive model that can calculate the average cost of used automobiles based on a variety of traits and attributes. We will create a series of regression models that forecast used automobile prices while taking into account both auction bid prices and potential repair expenses, in order to help buyers make budget-conscious purchases.

Background:

The used car industry is enormous and diverse with vehicles ranging in age, condition, and price. Due to hidden expenses, including potential repair costs that might not be immediately apparent during an auction or sale, it can be difficult for prospective buyers to determine a car's genuine value. A buyer's budget may be strained and regretful financial choices may be made as a result of overbidding on a vehicle that needs significant repairs. This difficulty is especially important at auctions, where the excitement and competitiveness can obscure a car's actual value. By accurately predicting the average price, inclusive of repair costs, buyers can navigate the second-hand car market with confidence, ensuring they obtain value for money and avoid financial pitfalls.

Importance / Potential

The potential for resolving the data science challenge of estimating used car prices in the USA is enormous, and it has the ability to significantly benefit both consumers and other players in the automotive sector.

Some key benefits of solving this problem are:

1. Accurate price prediction of a used car can provide consumers with knowledge of fair market values, which can enable prospective purchasers to make more informed choices, ensuring they pay a fair price.

2. Predictive models can help dealerships and individual sellers improve their pricing tactics. This gives companies the ability to set rates that are both competitive and profitable, attracting potential customers while maximizing revenue.

3. The chance of unfair negotiations will decrease since buyers and sellers will have access to a more transparent and data-driven pricing process.

4. Dealerships, producers, and policymakers can use this market trend information to better understand consumer preferences, demand patterns, and economic factors affecting the automotive market.

5. Advanced prediction models can provide vendors and purchasers with personalized recommendations. This can include recommendations for buyers based on their preferences and financial limitations. For sellers, the model might suggest pricing adjustments based on the state of the market.

Solving this problem could make a more open, effective, and fair automobile market for used cars. By delivering useful information and enhancing decision-making processes, it can benefit buyers, sellers, financial institutions, and other stakeholders.

Data Acquisition

We took data from *data.world* website and it is available in CSV format - *https://data.world/data-society/used-cars-data*
The dataset has approx 370,000 rows and 20 columns of used cars scraped with Scrapy from Ebay-Kleinanzeigen. Some of the categorical values are in German, so we have to translate those into English.

List of features in the dataset:

- *dateCrawled*: when this ad was first crawled
- *name:* `name` of the car
- *seller:* private or dealer
- *offerType:* `Gesuch` means `request` and `Angebot` means `listing`
- abtest: Tells whether the car is being modified or not
- vehicletype: Type of vehicle, SUV, coupe, etc

- yearOfRegistration: At which year the car was first registered
- gearbox: Car is automatic or manual
- powerPS: The horsepower of the car in PS
- model: Model of the car
- kilometer: How many kilometers the car has driven
- monthOfRegistration: At which month the car was first registered
- fuelType: Diesel, CNG, electric, etc
- Brand: Company of the car
- notRepairedDamage: If the car has damage and is not repaired yet
- dateCreated: The date for which the ad at eBay was created
- nrOfPictures: Number of pictures in the ad
- postalCode: Postal code of car location
- lastSeenOnline: When the crawler saw this ad last online

Data Cleaning

Data cleaning is one of the most crucial steps in any Data Science project to get quality data. It is important to make sure that your data is trustworthy, accurate, and ready for analysis.

1. Handling missing values - In the original dataset there were **371,824** rows and a lot of features were having null values. In the EDA section, we can visualize the number of null values for each feature. We decided to drop those instead of filling them by using domain knowledge or by some descriptive analysis like mean, mode, etc. After dropping null values, we still managed to preserve a good amount of rows, which is **261,161**.

```
Null value in each coloums are as follows:-
dateCrawled                0
name                       0
seller                     0
offerType                  0
price                      0
abtest                     0
vehicleType            37869
yearOfRegistration         0
gearbox                20209
powerPS                    0
model                  20484
kilometer                  0
monthOfRegistration        0
fuelType               33386
brand                      0
notRepairedDamage      72060
dateCreated                0
nrOfPictures               0
postalCode                 0
lastSeen                   0
dtype: int64
```

2. Removing duplicate records - After removing null values, we check for the number of records that are duplicates. Fortunately, there are not a lot of observations where all feature values are the same. After removing them we got **261,139** rows.

3. Resolving schema - Since our dependent variable `price` should be a decimal value or float, because the regression model would give us a floating point prediction. It is better to make the `price` column a float or decimal.

```
root
 |-- name: string (nullable = true)
 |-- seller: string (nullable = true)
 |-- offerType: string (nullable = true)
 |-- price: double (nullable = true)
 |-- abtest: string (nullable = true)
 |-- vehicleType: string (nullable = true)
 |-- yearOfRegistration: integer (nullable = true)
 |-- gearbox: string (nullable = true)
 |-- powerPS: integer (nullable = true)
 |-- model: string (nullable = true)
 |-- kilometer: integer (nullable = true)
 |-- fuelType: string (nullable = true)
 |-- brand: string (nullable = true)
 |-- notRepairedDamage: string (nullable = true)
 |-- dateCreated: timestamp (nullable = true)
 |-- nrOfPictures: integer (nullable = true)
 |-- postalCode: integer (nullable = true)
 |-- lastSeen: timestamp (nullable = true)
```

4. Handling outliers: There were a couple of features that had outliers. There are multiple ways to handle outliers like Interquartile Range (IQR), Z Score, etc. For this dataset, we use the IQR technique and set 1.5 as a factor. We will view that graphically in the EDA section.
After removing outliers, there were **201,726** rows left.

5. Handling inconsistencies in features: There are two features where all observations are in German language and we have converted that in English for consistency.
Changing values of 2 features from German to English

1. offerType - `Gesuch` and `Angebot` which means `request` and `listing`
2. notRepairedDamage - "ja" and "nein" which means `yes` and `no`

```
0     no
1     no
2    yes
3     no
4     no
Name: notRepairedDamage, dtype: object
```

```
0    nein
1    nein
2      ja
3    nein
4    nein
Name: notRepairedDamage, dtype: object
```

6. Feature Extraction: We create a new feature called `age_of_car` from two existing features - `yearOfRegistration` and `lastSeen`.

yearOfRegistration	lastSeen	lastSeen_year	age_of_car
1997	2016-04-07 12:45:02	2016	19
1999	2016-04-07 10:17:16	2016	17
2007	2016-03-26 15:16:14	2016	9
1998	2016-04-06 17:46:14	2016	18
2004	2016-04-06 03:17:34	2016	12
1997	2016-03-12 04:46:51	2016	19
1996	2016-03-06 03:44:30	2016	20
2008	2016-03-23 11:16:59	2016	8
1991	2016-04-06 02:15:30	2016	25
2001	2016-03-12 21:44:42	2016	15

only showing top 10 rows

7. Filtering data by date: Our dataset has used cars records where year if registration of a car was in 1919 and some of the cars are 50 or 60 years old.
We don't have lot of records for those observations, so we decided to take observations where age of car is less than or equal to 20 years.
After filtration, we were left with **183,445** rows.

8. Normalizing numeric features: Since most of our dataset has categorical columns and 2 columns are numeric which we scaled - `powerps` and `kilometer`

The new column names are: `powerps_scaled` and `kilometer_scaled`.

```
 #   Column              Non-Null Count   Dtype
---  ------              --------------   -----
 0   name                183445 non-null  object
 1   seller              183445 non-null  object
 2   offerType           183445 non-null  object
 3   price               183445 non-null  float64
 4   abtest              183445 non-null  object
 5   vehicleType         183445 non-null  object
 6   yearOfRegistration  183445 non-null  int32
 7   gearbox             183445 non-null  object
 8   powerPS             183445 non-null  int32
 9   model               183445 non-null  object
 10  kilometer           183445 non-null  int32
 11  fuelType            183445 non-null  object
 12  brand               183445 non-null  object
 13  notRepairedDamage   183445 non-null  object
 14  dateCreated         183445 non-null  datetime64[ns]
 15  nrOfPictures        183445 non-null  int32
 16  postalCode          183445 non-null  int32
 17  lastSeen            183445 non-null  datetime64[ns]
 18  lastSeen_year       183445 non-null  int32
 19  age_of_car          183445 non-null  int32
 20  powerPS_scaled      183445 non-null  float64
 21  kilometer_scaled    183445 non-null  float64
```

9. Removing features that have 0 values: We noticed that there is one feature called `nrofpictures` which means number of pictures of cars uploaded for the advertisement. All the observations has 0 values for this, so we decided to drop this feature.
We have also removed the `name` of car feature because it has junk values and 100k unique values which will be hard to encode. For our project we can identify a car by the `brand` name, `type of car`, and `model` of the car.

10. Removing bad values from the features: There are few columns which has a mixture of German and English words, this is different from the above two columns where all observations were having German words.
We have made changes to make everything in English.

```
['privat' 'gewerblich']
['andere' 'kombi' 'limousine' 'bus' 'kleinwagen' 'cabrio' 'coupe' 'suv']
['benzin' 'diesel' 'lpg' 'cng' 'hybrid' 'elektro' 'andere']
['manuell' 'automatik']
```

↓

```
['private' 'commercial']
['others' 'kombi' 'limousine' 'bus' 'kleinwagen' 'cabrio' 'coupe' 'suv']
['benzine' 'diesel' 'lpg' 'cng' 'hybrid' 'electric' 'others']
['manual' 'automatic']
```

11. Feature Selection: After changing scale, removing unimportant columns, changing observations and checking correlation, we did the final feature selection. We are left with 15 columns.

```
Data columns (total 15 columns):
 #   Column               Non-Null Count   Dtype
---  ------               --------------   -----
 0   seller               184445 non-null  object
 1   offerType            184445 non-null  object
 2   abtest               184445 non-null  object
 3   vehicleType          184445 non-null  object
 4   yearOfRegistration   184445 non-null  int32
 5   gearbox              184445 non-null  object
 6   model                184445 non-null  object
 7   fuelType             184445 non-null  object
 8   brand                184445 non-null  object
 9   notRepairedDamage    184445 non-null  object
 10  lastSeen_year        184445 non-null  int32
 11  age_of_car           184445 non-null  int32
 12  powerPS_scaled       184445 non-null  float64
 13  kilometer_scaled     184445 non-null  float64
 14  price                184445 non-null  float64
```

12. Encoding Categorical values: For the final dataset, we did one-hot encoding for categorical values and clean dataset will be attached in the zip file. Final shape of our data after Data Cleaning is: **(184445, 311)**

Exploratory Data Analysis (EDA)

1. **Summary of original dataset:**

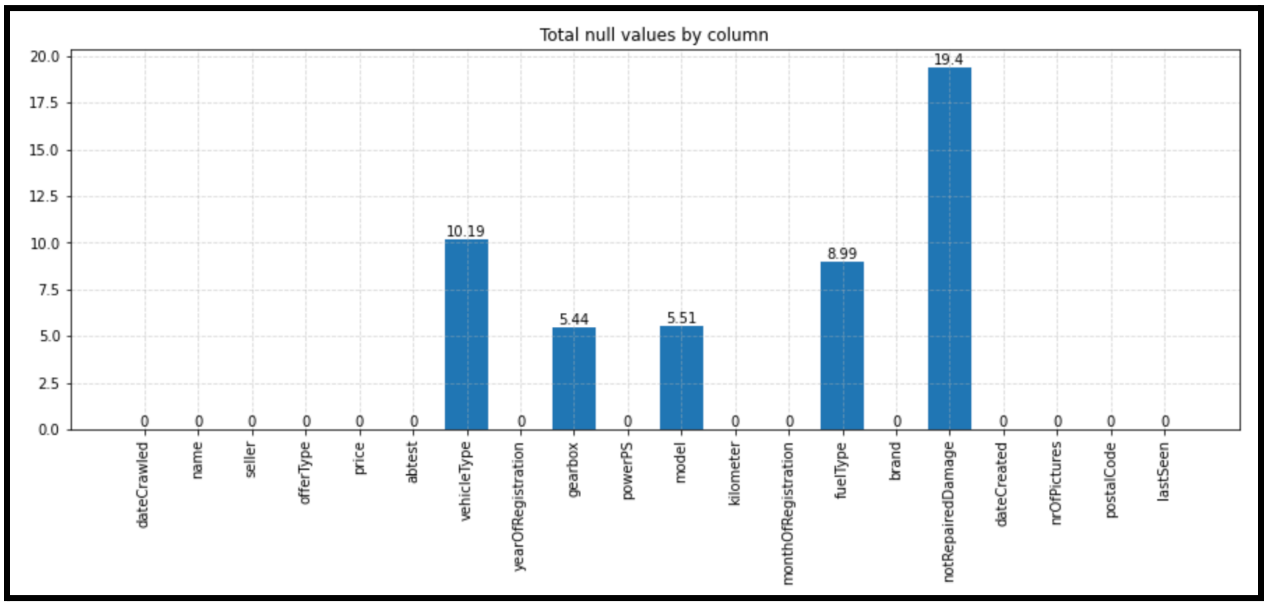
summary	yearofregistration	powerps	kilometer	price	nrOfPictures
count	371823	371823	371823	371823	371823
mean	2004.5767206439623	115.54146462160759	125618.56044408226	17286.338865535483	0.0
stddev	92.8299905430405	192.07254710096788	40111.620164944565	3586530.1840677853	0.0
min	1000	0	5000	0	0
max	9999	20000	150000	2147483647	0

Summary of dataset after removing outliers:

summary	yearofregistration	powerps	kilometer	price	nrOfPictures
count	202740	202740	202740	202740	202740
mean	2002.0737594949196	116.0988655420736	141323.93706224722	4286.3965127749825	0.0
stddev	5.45614528828709	52.03065222991242	17375.625524412106	4103.757017614489	0.0
min	1919	0	90000	0.0	0
max	2018	270	150000	20850.0	0

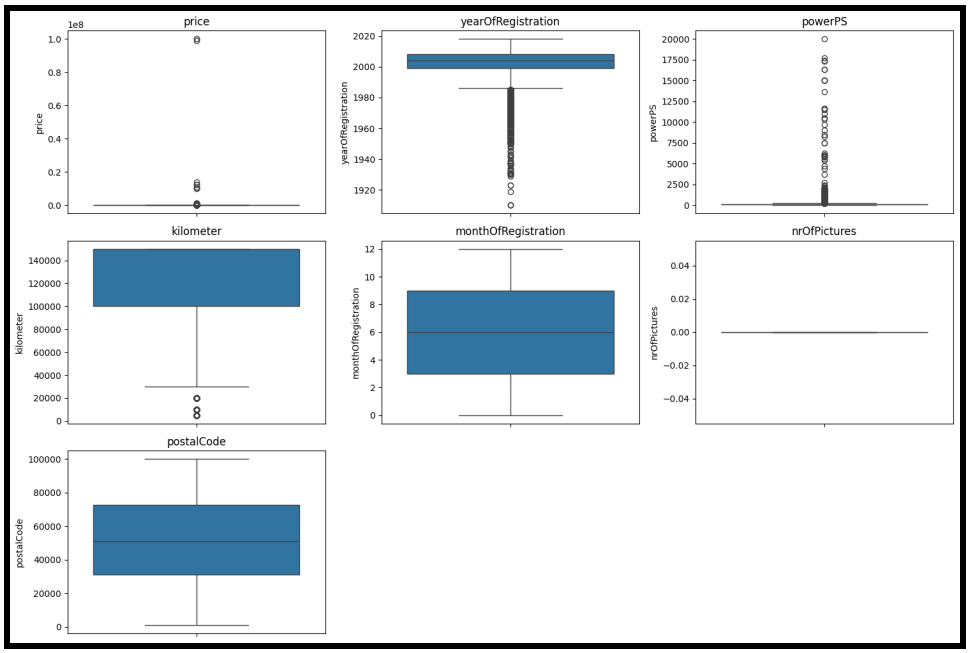
We can notice that in the original dataset there were a couple of problems, min year is 1000 and max year is 9999, PowerPS which is horsepower of a car is 20,000 which is not possible, also some of the prices are really huge and minimum price is 0.
All these issues were majorly handled by removing outliers.

2. **Percentage of null value in original data by columns:**



We can notice from the chart that notRepairDamage has the maximum percentage of null values, vehicleType, fuelType, etc.

3. **Outliers Detection on original dataset:** Boxplots helped us to determine the central tendency, variability, and any unusual numbers in our numerical data by visually inspecting these boxplots. Using the IQR technique we noticed a lot of outliers, which we removed in Data Cleaning process.



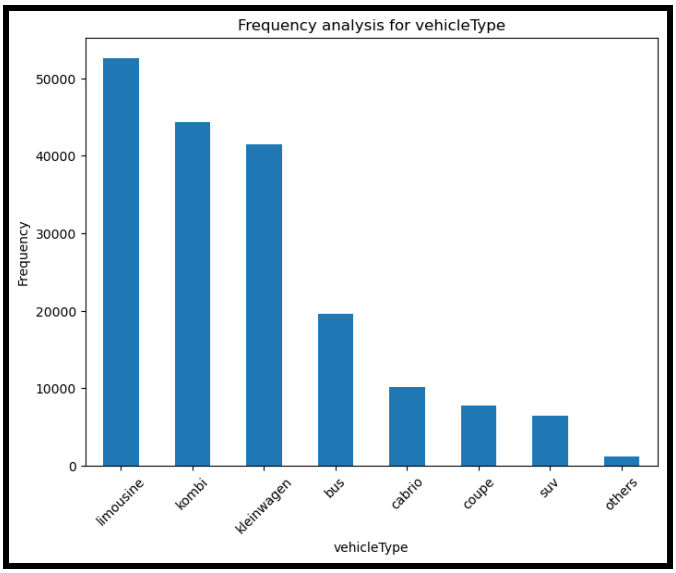
4. Univariate Analysis:

Univariate analysis is a statistical and data analysis technique that focuses on analyzing a single feature in isolation. In univariate analysis, you examine the distribution, characteristics, and properties of a single feature without considering its relationship with other features.

Univariate analysis was performed on some of the features like vehicleType, gearbox, fuelType, brand and age_of_car.

a. VehicleType:

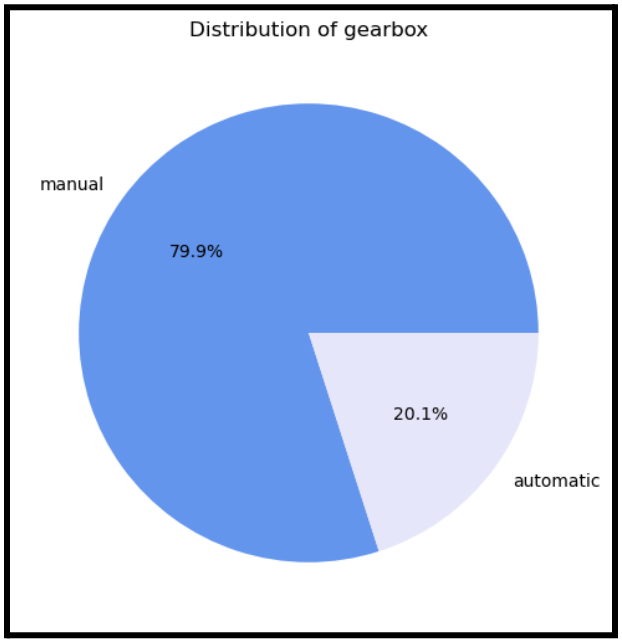
In this univariate analysis, we focused on a single variable, which is "vehicleType." The analysis aims to understand and describe the distribution and characteristics of unique vehicle types in the dataset. The count of the vehicle type limousine has the highest count and suv is the lowest



b. GearBox:

We focused on the variable "gearbox," which represents different gearbox types (e.g., manual and automatic) in the dataset. The analysis aims to understand and visualize the distribution of these gearbox types.

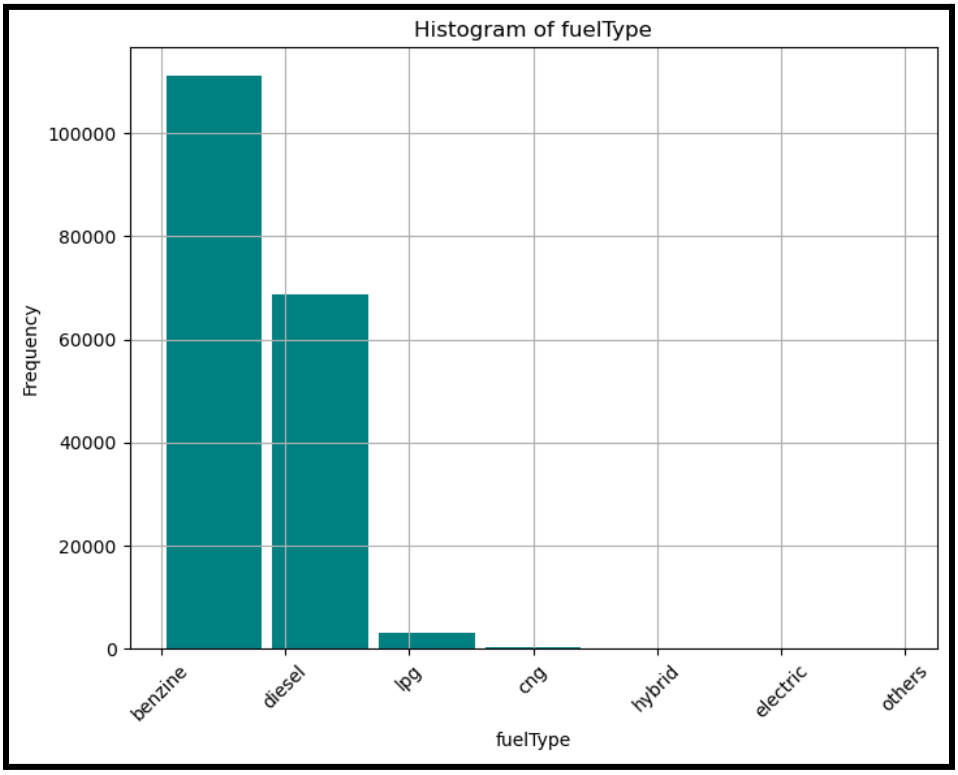
As you can see from the plot manual cars from our dataset have the highest count with 79.9% and automatic with only 20.1%



c. **FuelType:**

You created a histogram to examine the frequency distribution of the "fuelType" variable. A histogram divides the range of values into bins and counts how many data points fall into each bin. In this case, each bin represents a different fuel type, and the height of each bar indicates the frequency (number of occurrences) of that fuel type. of these gearbox types.

Cars with benzine fuel type have the highest count and we don't have any cars with hybrid, electric and the others left.

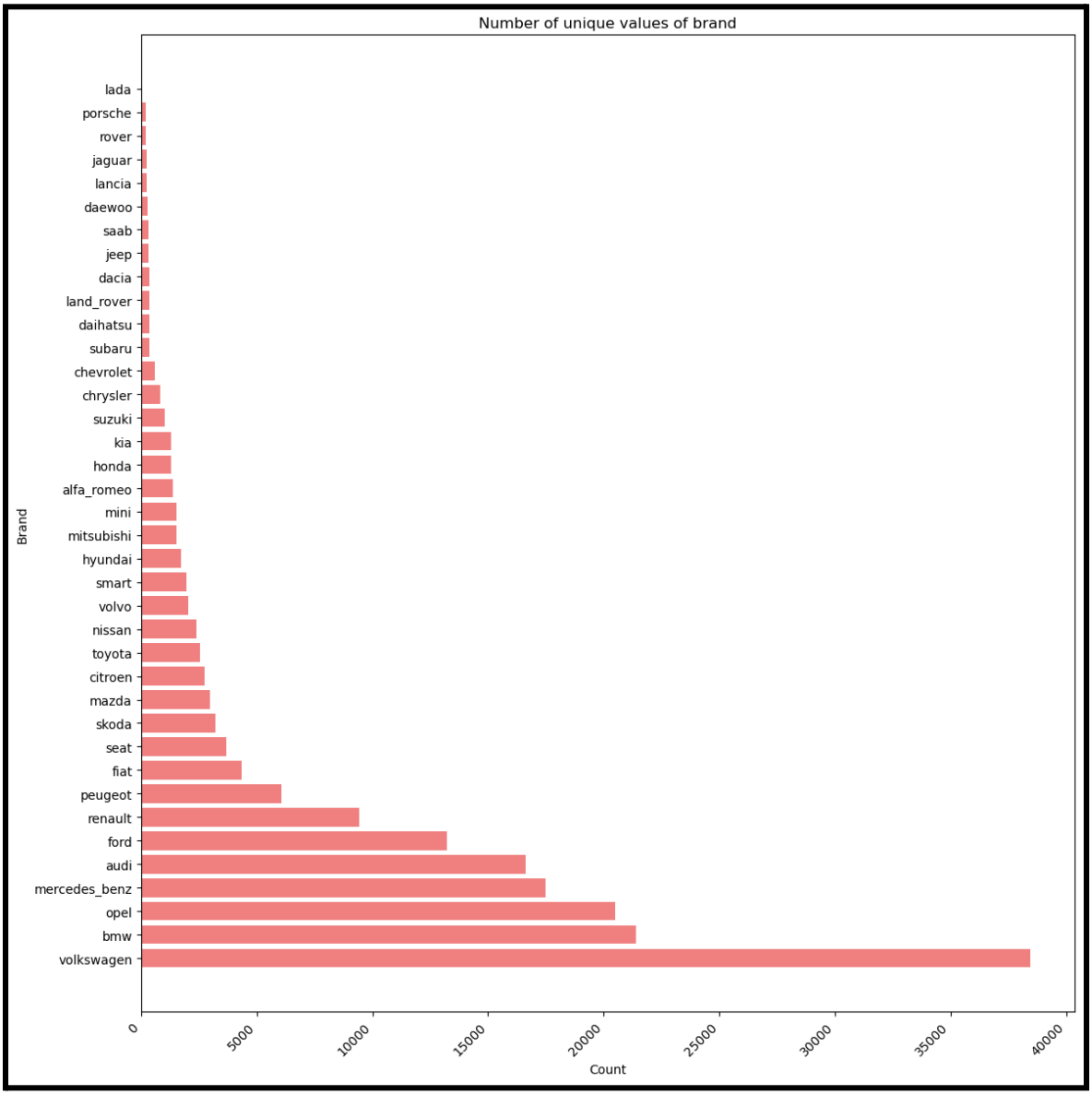


d. **Brand:**

In this univariate analysis, the focus is on the "brand" variable, which represents different vehicle brands in the dataset. The analysis aims to understand and visualize the distribution of these brands.

We calculated the frequency or count of each category within the "brand" variable. This provides insights into the distribution of vehicle brands in the dataset, indicating how many instances of each brand are present.

We have the brand Volkswagen with the highest count in cars, second with the bmw, opel being the third.

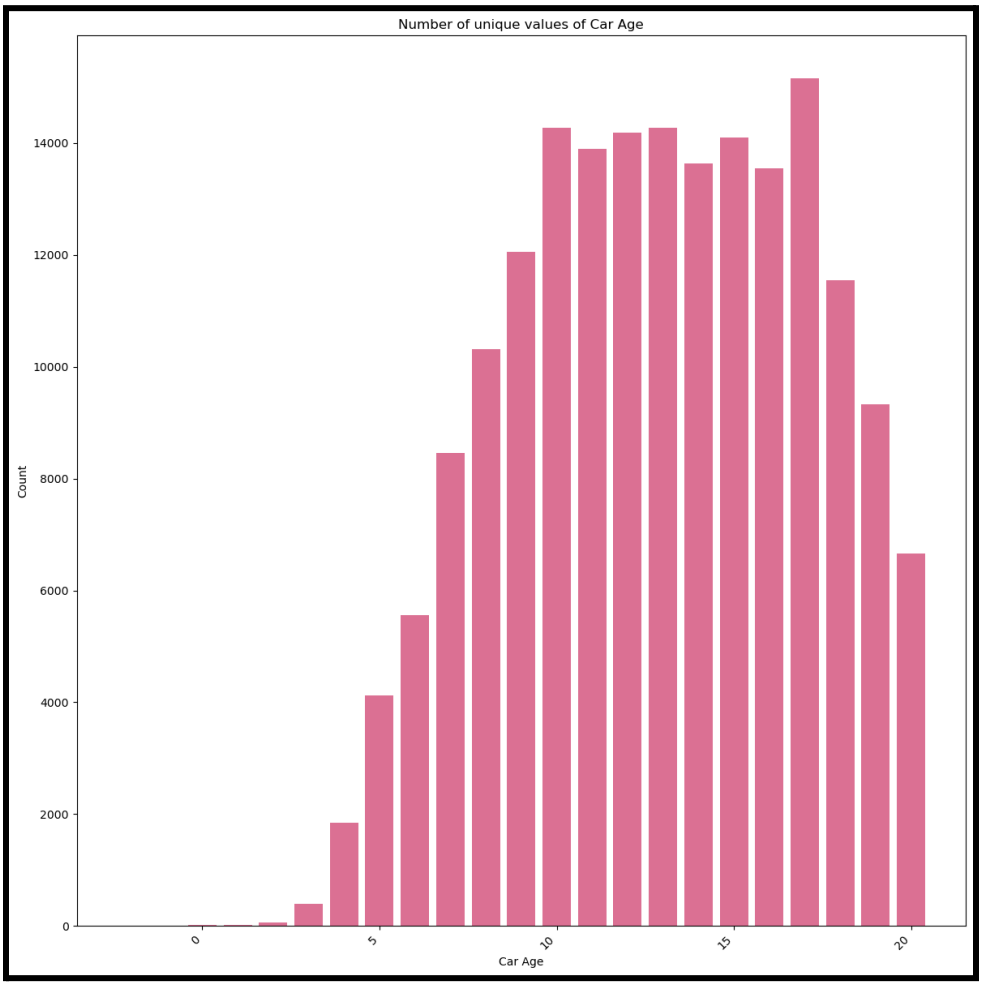


e. Age_of_car:

The "age_of_car" variable represents the ages of vehicles in the dataset. The analysis aims to understand and visualize the distribution of the ages of cars.

Calculated the frequency or count of each unique value within the "age_of_car" variable. This provides insights into the distribution of

car ages in the dataset, indicating how many vehicles fall into each age category.
Frequency of cars between 9 to 17 years old is the most.

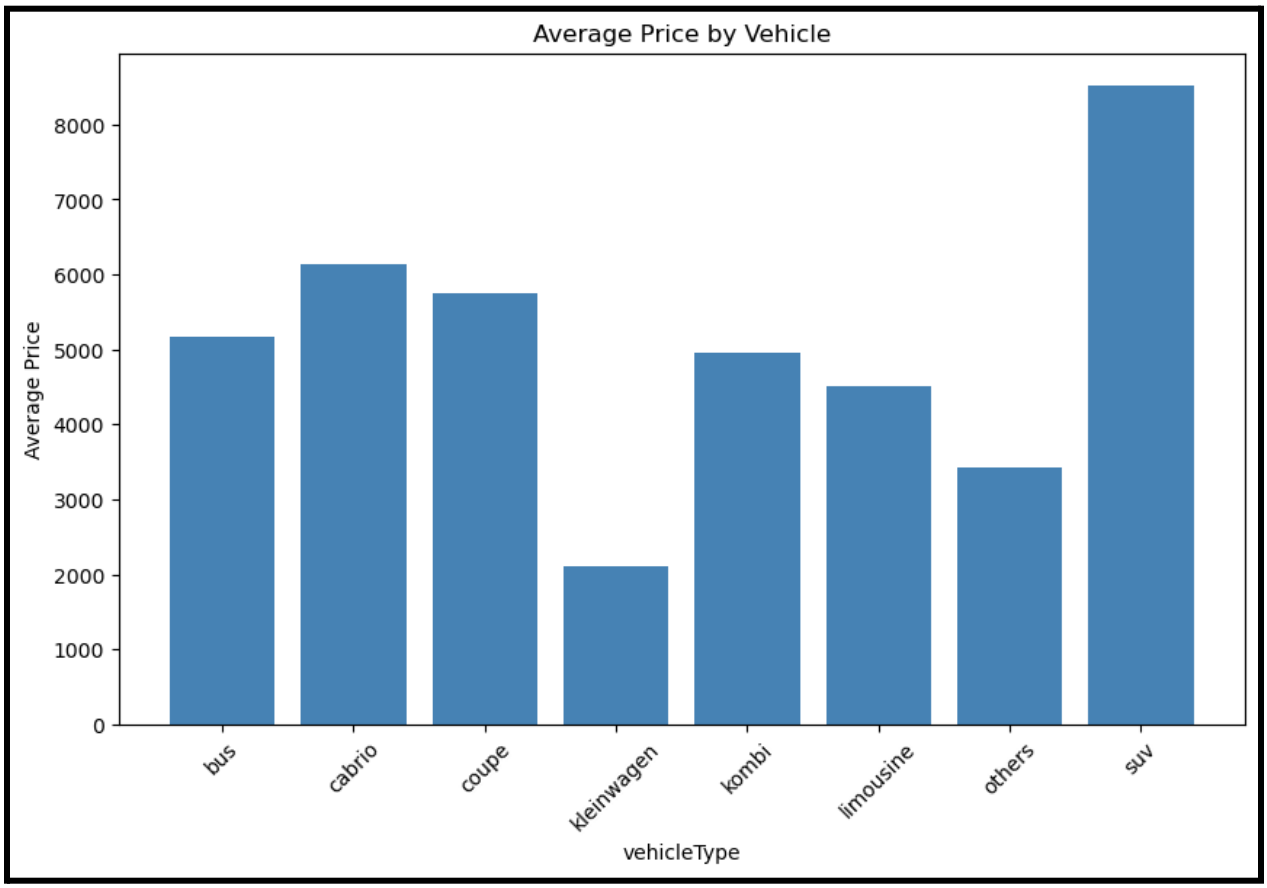


5. Bivariate Analysis:

Bivariate analysis is a fundamental step in understanding the association between different aspects of data. It involves the simultaneous analysis of two variables to understand their relationships and interactions. Unlike univariate analysis, which focuses on a single variable, bivariate analysis explores how two variables are related to each other. This analysis can help reveal patterns, correlations, and dependencies between the two variables.

a. VehicleType and Average Price:

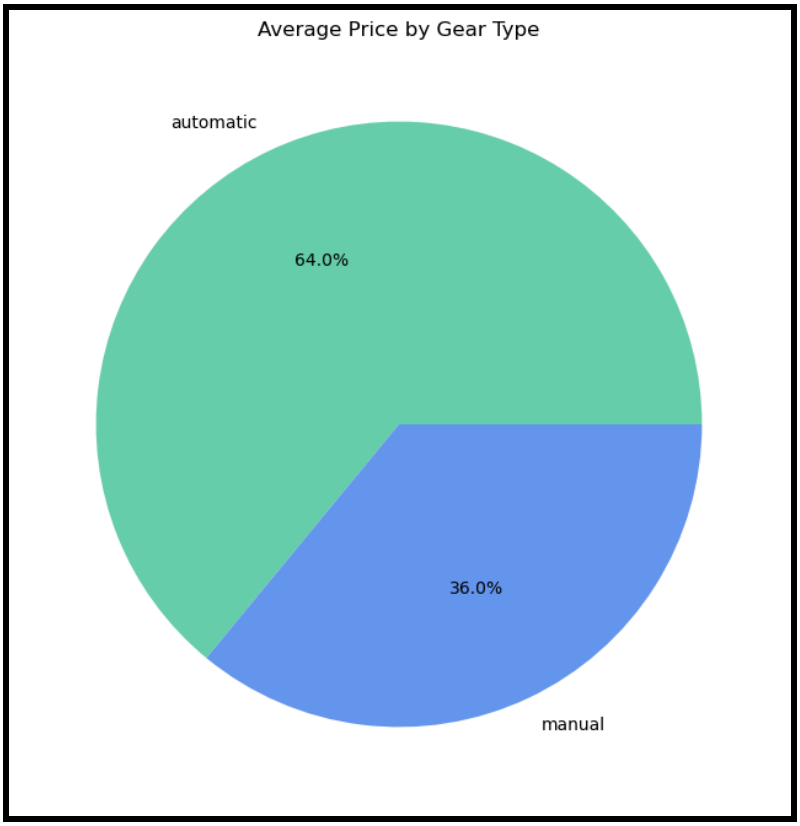
In the given bivariate analysis, we examined the relationship between the "vehicleType" (categories of vehicle types) and the "price" (average price). By visualizing this relationship in a bar chart, we observed how the average price varies across different vehicle types. The analysis provides insights into the price differences among vehicle categories, aiding in better understanding the pricing trends for various vehicle types.



b. Gearbox and Average Price:

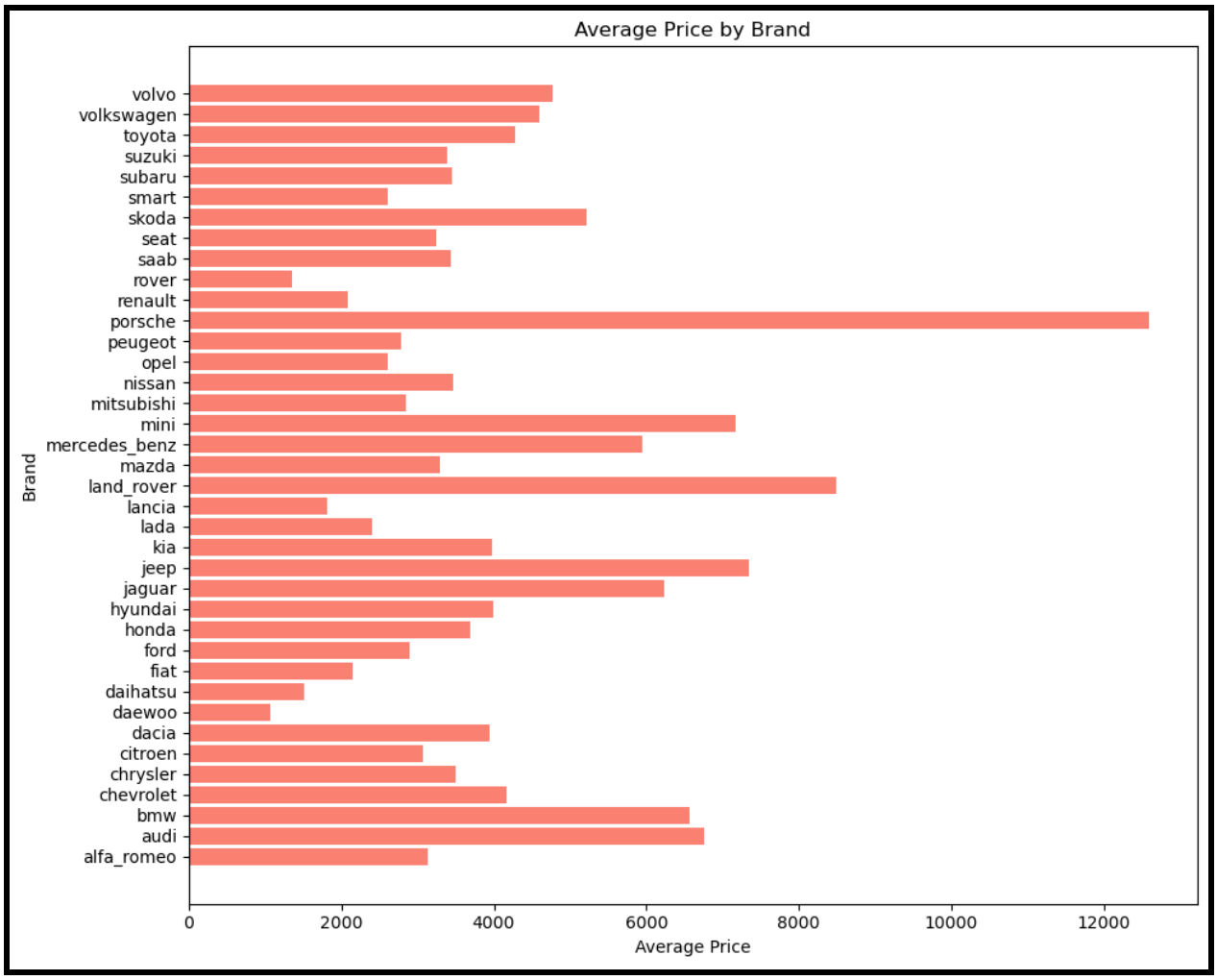
In the provided bivariate analysis, we investigated the relationship between the "gearbox" types (e.g., manual and automatic) and the "price" (average price) of vehicles. This analysis is visualized using a pie chart, where each slice of the pie represents a different gearbox type. The chart allows us to observe the distribution of average prices within each gearbox category. By

presenting the data in this manner, we gain insights into the relative pricing patterns for manual and automatic gearboxes.



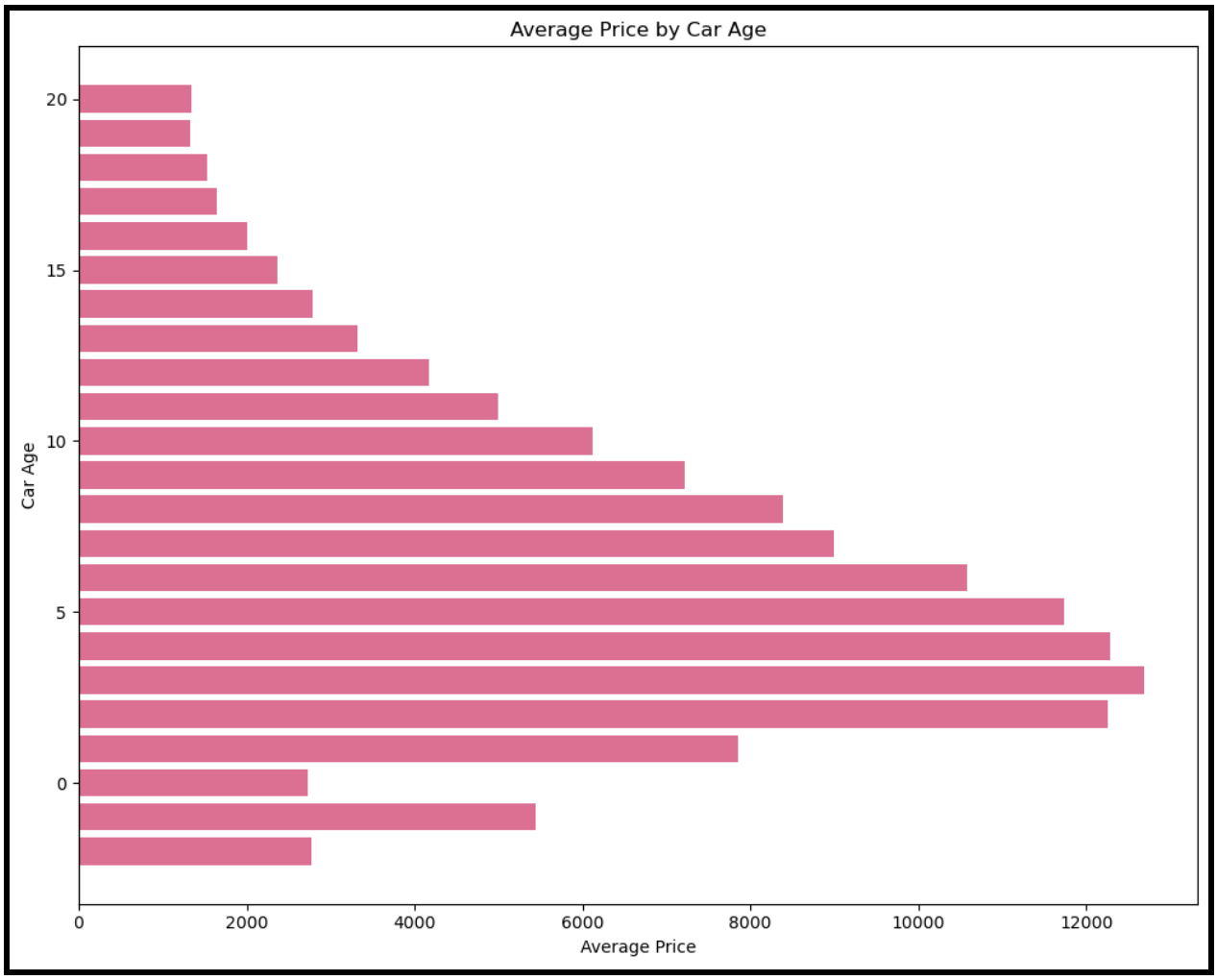
c. Brand and Average Price:

In this bivariate analysis, we explored the relationship between different "brands" of vehicles and their "average price." The analysis is depicted through a horizontal bar chart, with each brand represented on the y-axis and its corresponding average price on the x-axis. The chart's horizontal layout allows for a clear comparison of the average prices across various vehicle brands. By visualizing this data, we gain valuable insights into how different vehicle brands are priced relative to one another, which can be informative for decision-making and market understanding.



d. Age_of_car and Average Price:

In this bivariate analysis, we examined the relationship between the "age_of_car" (the age of vehicles) and their corresponding "average price." The analysis is visually presented using a horizontal bar chart, where the age of the car is displayed on the y-axis and its associated average price is shown on the x-axis. The horizontal orientation of the chart allows for an intuitive comparison of how the average prices vary with different car ages. By visualizing this data, we gain insights into how the age of a vehicle influences its average price, providing valuable information for understanding pricing trends in the used car market.



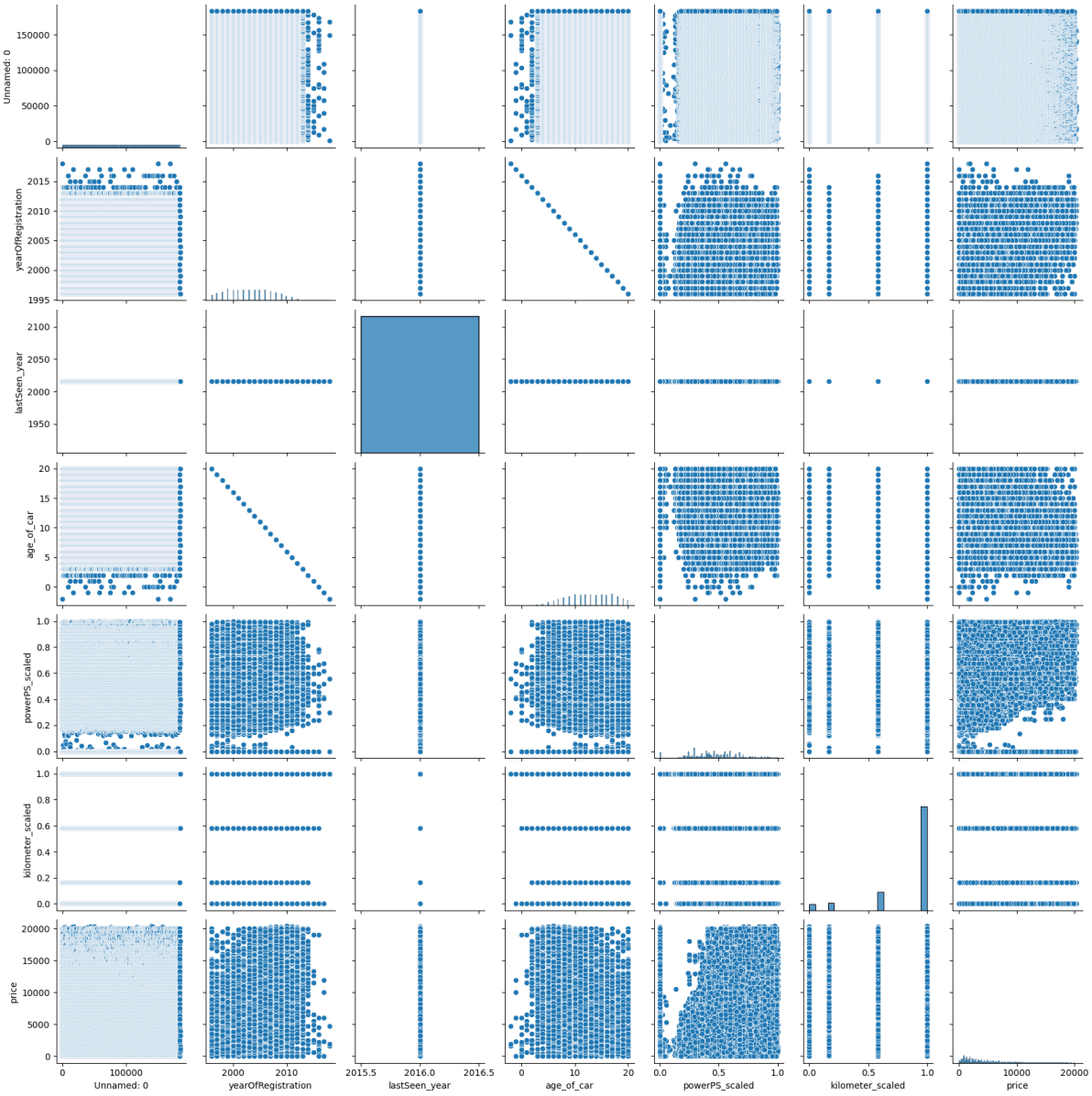
6. Multivariate Analysis:

Multivariate analysis is a statistical and data analysis technique that involves the simultaneous examination and analysis of multiple variables to understand their relationships, dependencies, and interactions. Unlike univariate analysis (which deals with a single variable) or bivariate analysis (which focuses on two variables), multivariate analysis considers three or more variables simultaneously. The primary goal of multivariate analysis is to gain a deeper and more comprehensive understanding of complex relationships within a dataset.

a. Pairplot of features:

A pairplot is a type of multivariate analysis that allows you to explore the pairwise relationships between multiple numeric variables in a dataset. It

allows you to visualize and explore the relationships between multiple numeric variables in your dataset. It's a useful tool for identifying patterns, correlations, and outliers in your data, helping you gain a better understanding of how the variables are related to each other.



7. CHI SQUARE

Chi-square is a statistical test that is used to look at differences between categorical variables in a random sample and see how well the findings meet the predictions.

Chi square ststistics with respect to price

	p-value
name	0.0000
seller	1.0000
offerType	1.0000
abtest	0.1205
vehicleType	0.0000
gearbox	0.0000
model	0.0000
fuelType	0.0000
brand	0.0000
notRepairedDamage	0.0000
dateCreated	0.0000
lastSeen	1.0000

Observations:

1. Car features (like model, gasoline type, brand, etc.) significantly affect the price categories.

2. The creation date of the listing has a sporadic correlation with pricing categories.

3.Offer type and price category have a weak relationship with a p-value of 0.0199.

4. Abtest groups and price distribution have a tenuous relationship (p-value 0.4091).

5.Price categories are not significantly impacted by seller type (p-value 0.8113).

6.The "last seen" component and price category do not correlate (p-value 1.0000).

8. Descriptive Statistics.

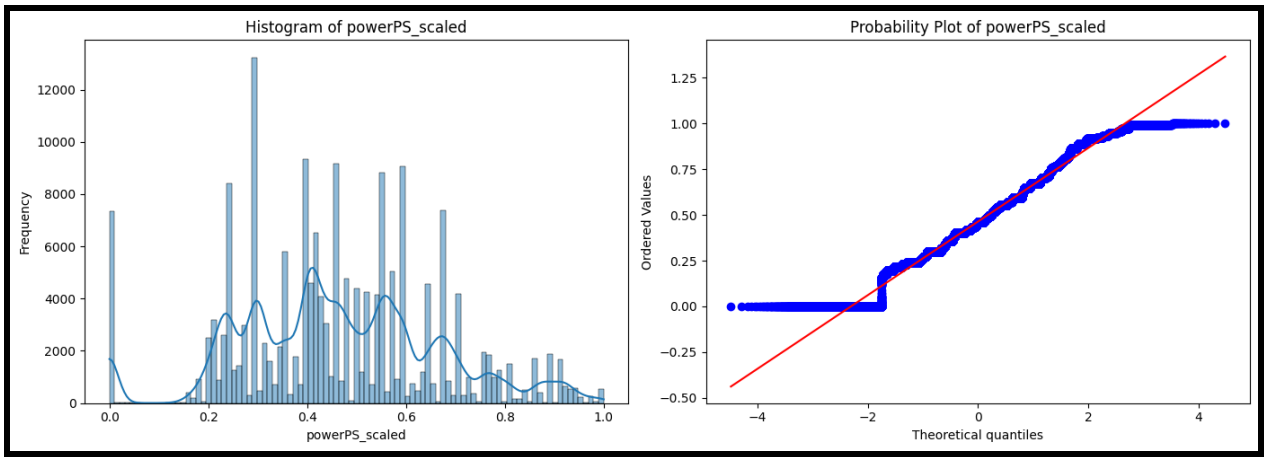
The patterns in the data can be summarized, described, and understood with the aid of descriptive statistics. They reveal information on the data distribution's central tendency, dispersion, and shape.

	powerPS_scaled	kilometer_scaled	price
Mean	0.464087	0.851898	4.421810e+03
Median	0.460317	1.000000	2.999000e+03
Mode	0.297619	1.000000	1.500000e+03
Variance	0.041211	0.085409	1.666654e+07
Standard Deviation	0.203005	0.292248	4.082467e+03
Skewness	0.100740	-1.842444	1.443364e+00
Kurtosis	-0.008991	2.096021	1.699489e+00

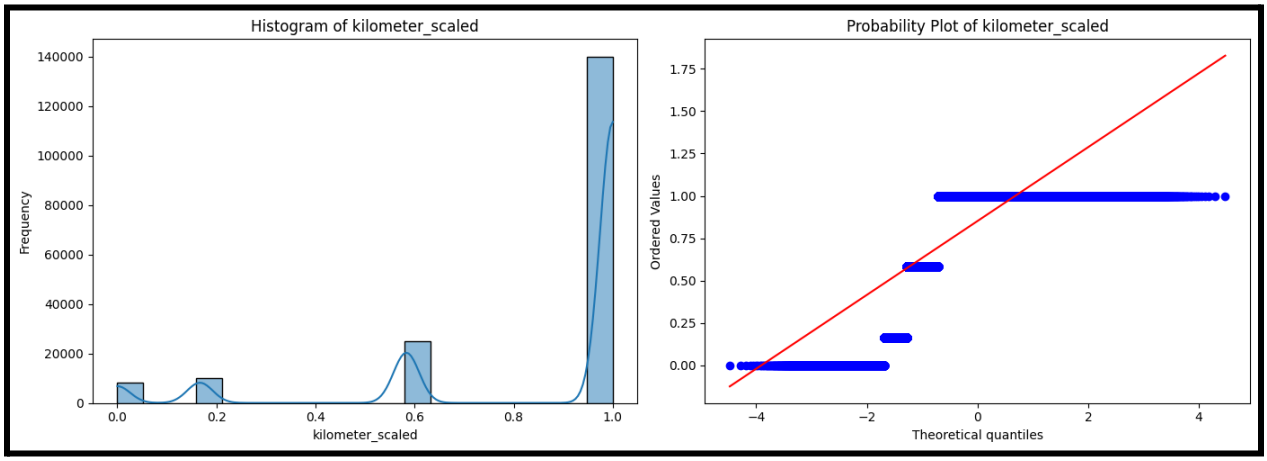
- **Price:**
  Average price is 4476.38 units, median is 2999 units, right-skewed with a kurtosis indicating heavier tails.
- **Kilometer:**
  Average distance is 141073.47 km, median is 150000 km, left-skewed with a pronounced peak and heavier tails.
- **PowerPS:**
  Average power is 117.3 PS, median is 116 PS, roughly symmetrical distribution with near-normal shape.

9. From Histograms to Quantiles: Interpreting Data Distributions.

1. **PowerBS:** The distribution of the powerPS_scaled data is depicted on the left by the histogram, and the distribution of the data relative to a normal distribution is shown on the right by the probability plot (or Gaussian plot). The probability plot indicates that the data are roughly normally distributed if a straight line connects all the data points. It appears that the powerPS_scaled data in this instance deviates from a normal distribution, particularly at the lower and upper tails.



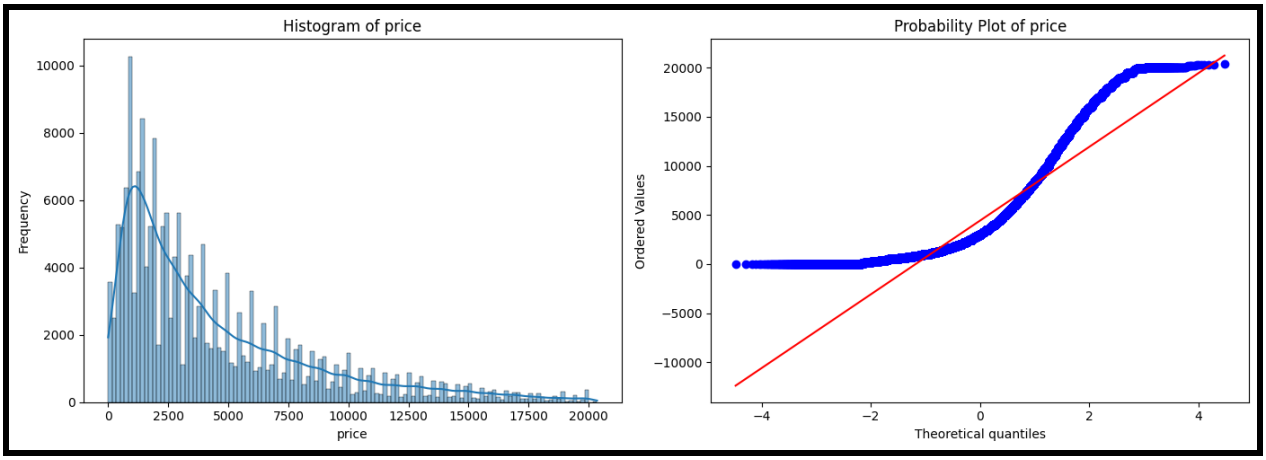
2.Kilometer



The kilometer_scaled data's distribution is displayed in the histogram on the left. It is clear that many cars in the sample have scaled kilometer values of 1.0 because we can see a big peak at this number.

A visual comparison of the distribution of the data to a normal distribution is again provided by the probability plot on the right. Since the kilometer-scaled data deviates from a straight line, it is possible that the distribution is not normal.

3. Price



The pricing data distribution is shown in the histogram on the left. The majority of automobile prices are concentrated in the lower range, with fewer high-priced vehicles, as seen by the right-skewed distribution that is visible.

A different visualization to evaluate the normalcy of the price data is provided by the probability plot on the right. It is possible that the price data is not properly distributed because of the data points' deviation from the straight line, especially towards the higher values.

10. Normalization

By converting values measured on various scales to a single scale, normalization ensures that no characteristic unfairly dominates due to its range. The columns powerPS_scaled, kilometer_scaled, and price had their values changed to range from 0 to 1 using the Min-Max Scaler. This approach increases algorithm efficiency and gives optimisation tasks a stable workspace.

11. Correlation Matrix

The correlation matrix provides information about how different variables relate to one another. We found through investigation that there is a positive association between price and powerPS_scaled, indicating that more powerful cars frequently fetch higher prices. Price and kilometer_scaled, on the other hand, show a negative connection, suggesting that cars with higher mileage may be more affordable. Notably, there was no discernible linear link between kilometer_scaled and powerPS_scaled, which was shown by their low correlation. For modeling and analyzing the interaction between diverse aspects,

it is essential to comprehend these dynamics.

```
price                1.000000
yearOfRegistration   0.696852
powerPS_scaled       0.543172
Unnamed: 0          -0.002701
kilometer_scaled    -0.252118
age_of_car          -0.696852
lastSeen_year             NaN
Name: price, dtype: float64
```

Heatmap

1. The price and powerPS_scaled have a positive relationship, which is logical given that more powerful cars frequently cost more.
2. The price and kilometer_scaled have a negative association, which implies that cars with more kilometers
(greater usage) may be less expensive.
3. There is only a minimum linear association between powerPS_scaled and kilometer_scaled, which is close to zero.